
*Sistema de aquisição e tratamento de dados para veículo baja
SAE*

William Pereira

Joinville

2018

William Pereira

*Sistema de aquisição e tratamento de dados para veículo baja
SAE*

Monografia de Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Graduação em Ciência da Computação, da Universidade do Estado de Santa Catarina (UDESC), como requisito parcial da disciplina de Trabalho de Conclusão de Curso.

Orientador: Roberto Silvio Ubertino Rosso Jr.

Doutor

Joinville

2018

William Pereira

*Sistema de aquisição e tratamento de dados para veículo baja
SAE*

Monografia de Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Ciência da Computação da UDESC, como requisito parcial para a obtenção do grau de BACHAREL em Ciência da Computação.

BANCA EXAMINADORA

Roberto Silvio Ubertino Rosso Jr.

Doutor

Charles Christian Miers

Doutor

Douglas Wildgrube Bertol

Doutor

*"Computer are like Old Testament gods:
lots of rules and no mercy."*

–Joseph Campbell

Agradecimentos

Agradeço primeiramente aos meus pais, Lourenil e Benvinda, pelo suporte incondicional. Ao meu orientador Roberto Silvio Ubertino Rosso Jr. pela paciência e sabedoria transmitidas ao longo deste trabalho. A minha namorada Camila Will por sempre acreditar em mim mesmo quando eu não acreditava. Ao grupo de Baja da UDESC Velociraptor pela oportunidade de trabalhar com pessoas incríveis como eles e aos meus amigos que estiveram sempre ao meu lado nas horas mais difíceis.

Resumo

Nas últimas décadas a indústria automotiva vem adicionando eletrônica para monitorar seus veículos em funcionamento urbano e para fins de alto desempenho em corridas automobilísticas. Este trabalho demonstra o passo a passo do desenvolvimento de um sistema de aquisição e tratamento de dados de um veículo *off-road* de alto desempenho com a utilização de protocolos de rede sem fio para realização de telemetria. Também é feita uma revisão bibliográfica do que é estado da arte para este assunto em artigos, livros e outras monografias. Além disto é realizado um estudo de sensores e microcontroladores a fim de criar este sistema, bem como uma discussão sobre o significado dos sistemas propostos por este trabalho. São realizados testes com o sistema criado em conjunto com a equipe para validar a eficácia da proposta e da execução, demonstrando os resultados e discutindo sobre os mesmos. Por fim é refletido sobre o estado atual do sistema e possíveis caminhos de trabalhos futuros na área.

Palavras-chave: *Microcontroladores, Sensores, ZigBee, Telemetria, Aquisição de Dados, Baja SAE, Automobilismo.*

Abstract

In the last decades the automotive industry have been acquiring electronic devices for monitoring dynamic and comfort functions at urban areas and with performance focus on racetracks. This work portrays a step by step of the development of an acquisition and treatment system for data acquired from a off-road performance vehicle using wireless protocols for telemetry. An literature review of state of the art content is made from articles, books and thesis sources. Besides this, a study of sensors and micro controllers with the objective of making this system is created, as well as a discussion of the meaning of the systems proposed by this work. Tests with the system created are performed with the stockholder team to evaluate the efficiency and execution of the project, showing and arguing about the results. By the end of it, a discussion about the current state of the system is made, aiming for possible ways of research in future works.

Keywords: *Microcontrollers, Sensors, Zigbee, Telemetry, Data Acquisition, baja SAE, Motorsport*

Conteúdo

Listas de Abreviaturas	10
1 Introdução	11
1.1 Estrutura do trabalho	14
2 Fundamentação Teórica	15
2.1 Baja SAE	15
2.2 Sensores	16
2.2.1 Temperatura do motor	17
2.2.2 Frequência de rotação do motor	18
2.2.3 Velocidade do veículo	19
2.2.4 Nível do combustível	20
2.2.5 Temperatura do câmbio CVT	21
2.2.6 Célula de carga	23
2.2.7 Deslocamento do amortecedor	23
2.3 Microcontroladores	24
2.3.1 Placas de desenvolvimento	26
2.4 ZigBee	28
2.4.1 Estrutura de envio	30
2.5 Software	31
2.5.1 Aquisição de dados	31
2.5.2 Tratamento de dados	32
3 Trabalhos Relacionados	34

4 Projeto/Proposta de Solução	43
4.1 Objetivos	43
4.1.1 Objetivo Geral	43
4.1.2 Objetivos Específicos	45
4.1.3 Mudanças de escopo	45
5 Desenvolvimento da solução	47
5.1 Bibliotecas e Softwares utilizados	47
5.2 Hardware utilizado	49
5.3 Software desenvolvido	51
6 Testes e análise de resultados	55
6.1 Taxa de atualização	58
6.2 Validação dos dados	59
6.3 Distância de funcionamento dos módulos ZigBee	59
6.4 Consumo de bateria	60
7 Conclusão e Trabalhos Futuros	61
Referências	63

Lista de Figuras

1.1	Veículo Fênix da equipe Baja Velociraptor.	12
2.1	Diagrama do sistema elétrico usados atualmente no Baja.	17
2.2	Variante do termistor para aplicação automotiva.	18
2.3	Vela automotiva.	19
2.4	Sensor de velocidade MTE 73020.	19
2.5	Sensor de velocidade conectado próximo ao freio traseiro.	20
2.6	<i>Shield</i> criado para sensores de efeito Hall US1881.	21
2.7	Câmbio CVT.	22
2.8	Sensor MLX90614 e seu funcionamento.	22
2.9	Sensor de deslocamento do amortecedor SHARP GP2Y0A21YK0F.	24
2.10	Diagrama de blocos do microcontrolador ATmega328/ATmega328p.	25
2.11	Foto do painel dianteiro atual do Baja Velociraptor.	27
2.12	Diagrama com camadas do protocolo ZigBee.	30
2.13	Alguns quadros de envio dos modelos XBee.	31
2.14	Diagrama exemplificando funcionamento do sistema de aquisição.	32
3.1	Exemplo de entrada <i>On-Board Diagnostics</i>	39
4.1	Diagrama com o esquema geral do sistema atual e o esquema geral do sistema proposto.	44
5.1	Sistema de controle <i>on-board</i> montado no veículo.	50
5.2	Tela principal do programa.	51
5.3	Tela de extensometria.	52
5.4	Tela de seleção de gráficos de teste e monitoramento.	53

5.5	Tela com exemplo de gráfico de temperatura do motor por tempo em segundos.	53
5.6	Tela de configurações.	54
6.1	Gráficos gerados a partir de dados da célula de carga com o veículo parado. . .	56
6.2	Gráficos gerados a partir de dados da célula de carga com o veículo andando. .	56
6.3	Gráficos gerados a partir de dados do sensor de temperatura.	57

Lista de Quadros

2.1	Comparação de microcontroladores.	26
2.2	Comparação entre placas de desenvolvimento	27
2.3	Comparação entre <i>Single Board Computer</i>	28
2.4	Comparação de alguns parâmetros dos protocolos 802.11b e 802.15.4.	29
4.1	Conceitos do cenário estudado	43

Lista de Abreviaturas

ADC	<i>Analogic to Digital Converter</i>
ECU	<i>Engine Control Unit</i>
RPM	<i>Rotação/Revoluções por Minuto</i>
SCOB	<i>Sistema de Controle On-Board</i>
SD	<i>Secure Digital</i>
UDESC	<i>Universidade do Estado de Santa Catarina</i>
TCC	<i>Trabalho de Conclusão de Curso</i>

1 Introdução

Veículos automotores estão presente no nosso dia a dia há mais de 100 anos (HISTORY, 2017) e nunca param de evoluir. Tais máquinas que por anos eram peças de engenharia mecânica tem recebido uma mudança de paradigma. Os carros tem acumulado eletrônica para desempenhar funções simples e complexas que algumas décadas atrás eram impossíveis. Sensores, atuadores e centrais eletrônicas de processamento podem ser usados para auxílio na segurança, com detecção de faixas de pista, auxílio em frenagem antecipada e controle de estabilidade (KATZOURAKIS, 2012), em aplicações de conforto com sensores de temperatura do ambiente para controlar aparelhos condicionadores de ar digital e também em aplicações de alto desempenho, onde os sistemas atuam para entregar um melhor tempo de volta para o veículo, de acordo com sua situação em que o mesmo se encontra na pista.

Ainda na área de alto desempenho a área de telemetria, aquisição de dados e tratamento de dados de um veículo são usadas de forma extensiva em competições automobilísticas onde a amostragem em tempo real é necessária para centenas de sensores que permitem aos engenheiros manterem o rastreamento do desempenho do *powertrain*, parâmetros de dirigibilidade como configurações de suspensão e temperatura dos pneus (ANDRIA, 2015). Estes dados são fundamentais para que os engenheiros e pilotos consigam acertar o carro a fim de obter tempo de voltas mais rápidos, além de melhor manutenção do carro de forma geral.

Dentro deste contexto, a equipe escolar de automobilismo Velociraptor¹ foi procurada para realização de um projeto que envolvesse automobilismo e ciência da computação. Depois de diversas reuniões com a equipe foi encontrada uma área em comum, a aquisição de dados e telemetria do veículo da categoria Baja SAE codinome Fênix, que pode ser visto na Figura 1.1.

O projeto consiste na criação de um *software* de tratamento de dados e também na atualização e/ou criação de sistemas que auxiliariam este *software*. O sistema de tratamento de dados deve receber informações de grandezas provenientes do veículo durante sua execução e mostrar em um computador nos boxes da equipe os dados mais importantes. Para realização deste sistema foi utilizado de telemetria com uma rede sem fio do protocolo ZigBee conectada

¹URL: <<https://www.facebook.com/velociraptor.baja/>>

Figura 1.1: Veículo Fênix da equipe Baja Velociraptor.



Fonte: Autor.

ao sistema de controle *on-board* (ou SCOB) que consiste em uma unidade de processamento para dados recebidos de sensores espalhados pelo carro com um microcontrolador.

Em relação ao sistema de tratamento de dados proposto, a equipe deseja obter melhores informações sobre o veículo de forma intuitiva com visualização dos dados com gráficos e separação das informações por setores automotivos, de forma simplificada com a utilização de um único programa para visualização das informações, com um programa feito de forma modular, ou seja, que esteja preparado para receber atualizações futuras de componentes e tecnologias e com armazenamento para melhor confiabilidade e comparações de informações de diversas provas de teste.

Inicialmente alguns problemas encontrados no cenário de veículos *off-road* podem ser analisados. Em Araújo (2006) alguns avisos sobre problemas que são comumente encontrados em situações de extremo esforço para o conjunto mecânico e eletrônico, como em uma prova de enduro do Baja SAE. Estes problemas devem ser levados em consideração na hora de montagem do sistema de aquisição de dados e também do *software* de tratamento de dados. Os problemas encontrados são:

- Interferência eletro-magnética: Os picos de tensão produzidos pela centelha da vela podem interferir no microcontrolador se o mesmo não for bem isolado;
- Temperatura do *cockpit*: Depende muito da hora de realização do enduro e local, mas em

regiões mais quentes como no nordeste e em horários como 12 e 14 horas, a temperatura interna do *cockpit* pode ficar muito elevada e esta pode comprometer qualquer circuito eletrônico que seja sensível a calor e não esteja protegido;

- Vibração: Pela natureza de uma prova *off-road*, existe muita vibração. Este problema pode acarretar em mal contato entre circuitos e placas;
- Lama e água: Devido ao contexto do projeto *off-road* existem muitos obstáculos com lama e água no percurso dos testes e estes dois elementos podem danificar as placas eletrônicas.

Foram então levantados junto com a grupo Velociraptor várias grandezas físicas que a equipe deseja monitorar. Alguns componentes para sensoriamento já estão instalados no veículo e fazem parte dos requisitos do projeto, na parte de grandezas que devem ser monitoradas. Os seguintes itens foram levantados:

- Frequência de rotação do motor;
- Velocidade do veículo;
- Nível do combustível;
- Relação de transmissão;
- Temperatura do câmbio CVT;
- Rolagem da carroceria;
- Deslocamento do amortecedor;
- Deslocamento da suspensão; e
- Temperatura do disco de freio.

Enquanto algumas grandezas não fazem parte do escopo inicial deste projeto, as que fazem são analisadas na fundamentação teórica. As grandezas serão recebidas no computador dos boxes a partir do SCOB e transmitidas via protocolo ZigBee para a plataforma de tratamento de dados. Com este trabalho é esperado que a equipe consiga manter informações valiosas sobre o veículo que não poderiam ser obtidas de outra forma. A utilização de um *software* de monitoramento é um passo importante para atingir um melhor desempenho na hora da manutenção, construção e realização de novos projetos para os veículos focados em automobilismo.

1.1 Estrutura do trabalho

O trabalho a seguir é estruturado da seguinte forma. O Capítulo 2 traz uma revisão de conceitos que serão utilizados no trabalho. Os conceitos abordados incluem o funcionamento e organização da prova da categoria Baja SAE, com alguns detalhes das regras impostas para produção do veículo *off-road*, uma revisão sobre vários sensores utilizados para a captura de grandezas monitoradas, um estudo sobre microcontroladores para adequação de novas tecnologias sobre o sistema embarcado do veículo, uma análise sobre o protocolo de transmissão de dados sem fio e por fim um estudo sobre os objetivos dos sistemas de tratamento e aquisição de dados. No Capítulo 3 é realizado um estudo sobre o que existe de estado da arte em aquisição de dados na área automotiva, com foco em trabalhos voltados para o automobilismo. No Capítulo 4 é apresentada a proposta de trabalho, junto com os objetivos específicos que devem ser alcançados no percurso de sua realização. No Capítulo 5 é exposto o desenvolvimento do sistema de tratamento e de aquisição de dados. No Capítulo 6 são discutidos os testes realizados no sistema para validação dos resultados. Por último no Capítulo 7 estão as considerações finais sobre o trabalho e algumas ideias para trabalho futuros sobre o sistema especialmente para seu uso nas competições.

2 Fundamentação Teórica

Nesta sessão serão descritos alguns dos conceitos essenciais para a compreensão do trabalho. Inicialmente será explicado como funciona a organização Baja SAE e as provas as quais os carros são submetidos, além de um breve resumo da história da equipe Velociraptor. Também são descritos alguns detalhes técnicos de quais sensores são/devem ser aplicados no percurso do trabalho além de informações sobre os microprocessadores estudados para servir como base do SCOB vão ser discutidas. Uma análise sobre o protocolo ZigBee é feita e por último são discutidos alguns conceitos dos sistemas de tratamento e aquisição de dados.

2.1 Baja SAE

A categoria Baja, organizado pela SAE (*Society of Automotive Engineers*), é uma categoria de *motorsport* feita para estudantes de engenharia aprofundarem seu conhecimento na área com um projeto real, no qual toda a construção do veículo deve ser realizada, bem como sua documentação e busca por patrocinadores para viabilização do projeto. Os carros a serem montados devem, por regulamento, (SAE, 2017) serem feitos de uma estrutura tubular de aço, fibra de vidro, carbono ou kevlar (ARAÚJO, 2006), com quatro ou mais rodas e motor padrão de 10HP. Também segundo o regulamento o carro deve suportar uma pessoa de até um metro e noventa de altura e 113,4 quilogramas de peso. Todo o sistema de suspensão, freio, transmissão e chassi é projetado e executado pela equipe participante.

As provas realizadas pelos veículos em um torneio , segundo (SAE, 2016) são:

- Segurança;
- Motor;
- Conforto;
- Frenagem;
- Suspensão;
- Capacidade de tração;

- Dirigibilidade; e
- Enduro.

Além destas provas a equipe também deve realizar uma apresentação com o projeto completo do veículo, contando pontos para o torneio. A equipe que obtiver a maior quantidade de pontos nas provas citadas acima ganha o torneio.

2.2 Sensores

Os sensores, junto com atuadores, são os métodos em que circuitos elétricos conseguem se comunicar com o mundo físico realizando algumas tarefas como fazer uma medida de alguma grandeza. Em Fraden (2016) o sensor é definido como "um dispositivo que recebe um estímulo e responde com um sinal elétrico". Existem diversos tipos de sensores e tipos de variações específicas para cada área de uso, sensores de posição como os potenciômetros, sensores de velocidade angular como tacômetros com rotores dentados, sensores de proximidade com uso de sensores ópticos e assim por diante (KILIAN, 2001).

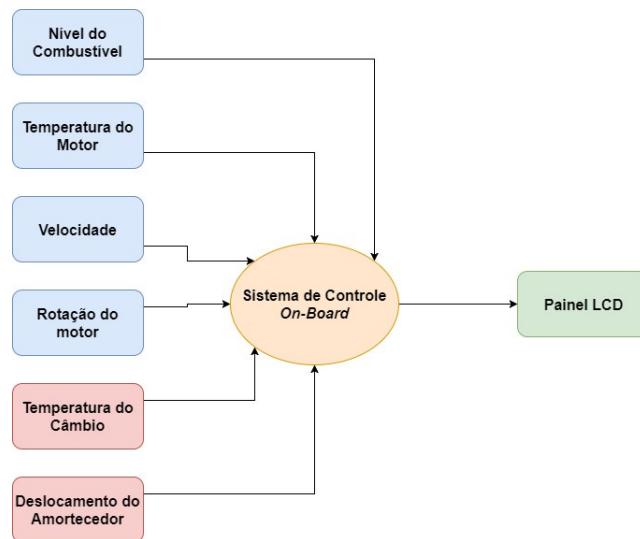
Os sensores presentes em um veículo se dividem em duas categorias (CHANDRAMANI; BHANDARI; HARIPRASAD, 2014) - aqueles que são feitos para o conforto do passageiro e aqueles feitos para garantir o funcionamento devido da parte dinâmica do veículo. O primeiro é importante para a lista de opcionais do carro consequentemente agradar ao consumidor comum. O segundo é crítico ao engenheiro em alto nível, sem impacto imediato para o público em geral.

Um bom exemplo de como o sensoriamento pode ajudar no desempenho de uma equipe de automobilismo em tempo real é dado por Araújo (2006). Quando superaquecido, o motor fica fora do seu regime ideal de trabalho, causando um maior desgaste dos componentes internos e por consequência uma perda de rendimento, o autor continua e comenta que quando o motor está numa temperatura muito abaixo da ideal, o rendimento também sofre alteração, porque o consumo de combustível aumenta e o torque é menor em relação a um motor ajustado. Para manter o processo dentro de uma faixa desejada, sensores de temperatura do óleo devem ser instalados, com eles é possível manter um histórico da faixa de funcionamento do motor e além de manter o piloto constantemente atualizado da temperatura do motor do seu veículo, o que permite que o piloto tenha mais liberdade em forçar o carro para melhorar o tempo de volta

ou dirigir mais cautelosamente, a fim de evitar desgaste excessivo nas peças e melhor consumo de gasolina, fatores que devem ser levados em consideração em uma prova de automobilismo.

O projeto do veículo *off-road* do grupo Velociraptor já conta atualmente com alguns sensores. A Figura 2.1 mostra o esquema do sistema elétrico usado atualmente no veículo. Os espaços em azul são sensores utilizados na competição, os espaços em vermelho são sensores usados para testes de bancada.

Figura 2.1: Diagrama do sistema elétrico usados atualmente no Baja.



Fonte: Autor.

As subseções a seguir explicam o funcionamento destes sensores.

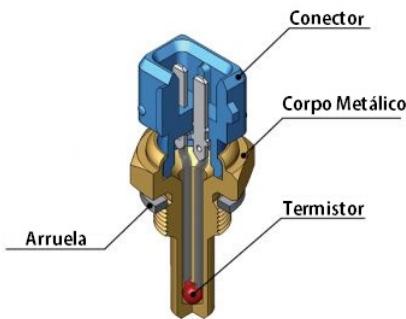
2.2.1 Temperatura do motor

O motor deve operar em uma temperatura ideal abaixo de 110°C para desempenhar bem o seu papel. Se sobreaquecidas, as peças do conjunto começam a operar de forma incorreta e isto pode ocasionar rompimento da junta de vedação da galeria de óleo, diminuição da viscosidade do óleo e em casos de exposição prolongada a sobreaquecimento, travamento do pistão dentro da câmara do motor. Para evitar casos como estes citados acima é importante ter a informação sobre a temperatura do motor, com a informação o piloto pode forçar menos o motor na prova para manter a temperatura estável ou o carro pode ser parado para uma manutenção de emergência.

Para realizar a aquisição desta grandeza é utilizado um sensor termistor de coeficiente negativo de temperatura (conhecido como NTC). Este sensor é facilmente encontrado em

lojas de eletrônica, porém o utilizado no projeto Baja Velociraptor é preparado para o ambiente automotivo com um corpo metálico para proteção contra ruídos e melhor posicionamento do sensor. A Figura 2.2 é um desenho que demonstra qual é a composição deste sensor. O sensor vai acoplado ao cárter do motor, submerso em óleo. O sensor emite um sinal analógico proporcional a temperatura que ele é exposto, a resistência dele diminui de acordo com o aumento de temperatura e aumenta com temperaturas mais baixas (FRADEN, 2016).

Figura 2.2: Variante do termistor para aplicação automotiva.



Fonte:(FAE, 2017) modificado pelo autor.

2.2.2 Frequência de rotação do motor

A revolução/rotação por minuto (RPM) é uma grandeza que indica quantas rotações por minuto um certo objeto realiza, sendo em torno de si mesmo ou em torno de um outro objeto. No contexto do automobilismo, a RPM é a quantidade de vezes em que um motor realiza um giro completo, movendo o virabrequim que está conectado a uma transmissão. Esta transmissão por sua vez está conectada a um eixo e consequentemente o eixo está conectado a uma roda. Esta é uma ideia básica de como funciona o sistema de conversão de um fenômeno químico (combustão do motor) em força física e o que o RPM trata neste meio.

Para fazer aquisição desta grandeza, é utilizado uma técnica que envolve a vela. A vela, que pode ser vista na Figura 2.3, está presente em todos os carros movidos a motores de combustão a gasolina, sua função é causar a centelha que começa o processo de explosão do combustível misturado com ar na câmara interna do motor.

A vela produz uma centelha a cada ciclo de explosão da gasolina e esta centelha é produzida a partir de uma descarga de energia com tensão de 12KV. A carga de energia é capturada por uma bobina em volta do fio elétrico da vela e conectada a uma entrada digital do microprocessador. Quando ocorrer a centelha é executada uma interrupção no programa

Figura 2.3: Vela automotiva.



Fonte: (BOSCH, 2017).

do microprocessador e incrementado um valor em uma variável reservada para o RPM. Esta variável ainda deve ser tratada antes de ser passada para o painel pois a centelha da vela é ativada uma vez a cada ciclo do motor, porém dentro de um ciclo completo (admissão, compressão, explosão e exaustão) é realizado dois giros do virabrequim, ou seja, duas RPM. Depois de ter o valor multiplicado por dois, o valor é mostrado no painel.

2.2.3 Velocidade do veículo

A velocidade do veículo é a grandeza que indica a quanto rápido um veículo se move de um ponto A para um ponto B. Para aquisição desta grandeza é utilizado um sensor MTE 73020 que pode ser visto na Figura 2.4. Este sensor vai acoplado perto do eixo do veículo e captura a rotação do mesmo graças ao efeito Hall, esta rotação é transformada em uma onda quadrada (MTE-THOMSON, 2017) e que deve ser tratada de acordo com a sua aplicação. Este tipo de sensor é utilizado em carros da montadora General Motors.

Figura 2.4: Sensor de velocidade MTE 73020.



Fonte: (MTE-THOMSON, 2017).

No Baja Velociraptor o sensor está acoplado ao eixo traseiro, próximo ao motor, e captura a rotação do freio traseiro que está conectado ao eixo, como pode ser visto na Figura

2.5. Toda vez que um dente do freio passa próximo ao sensor o mesmo emite um sinal que está conectado a entrada digital do microprocessador no SCOB. A frequência recebida pelo SCOB é tratada, levando em consideração o tamanho da roda e número de dentes no freio, resultando na fórmula matemática a seguir:

Figura 2.5: Sensor de velocidade conectado próximo ao freio traseiro.



Fonte: UDESC Baja Velociraptor (2017).

$$V = \frac{S_{vel}}{N_{freio}} \times 2\pi r \times \frac{3.6km}{h},$$

Sendo V é a velocidade final mostrada no painel, S_{vel} é o número de vezes que o sensor produziu um sinal quadrático, N_{freio} é a quantidade de dentes presente no freio customizado do Baja Velociraptor, $2\pi r$ é fórmula do comprimento da circunferência e $\frac{3.6km}{h}$ transforma o resultado de metros por segundo para quilômetros por hora.

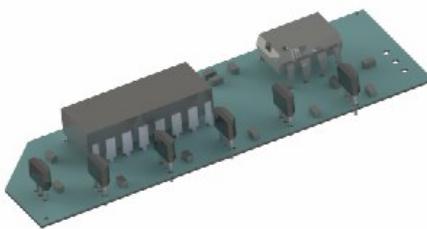
2.2.4 Nível do combustível

O combustível é utilizado para causar a explosão controlada dentro do motor de combustão. Medir a quantidade ainda disponível no tanque de gasolina dele é essencial para que possam ser tomadas decisões como manter o veículo na pista ou chamar o veículo para os boxes. A técnica mais comum utilizada para medir esta grandeza é a boia, como pode ser visto em alguns trabalhos analisados ((NUNES, 2016) e (ARAÚJO, 2006)) mas esta técnica não pode mais ser aplicada nas competições devido a novas regras de competição que não permitem uso de fios dentro de motores ou tanques de combustível (SAE, 2017).

Para se adequar a estes novos regulamentos, foi utilizada uma nova técnica de medida da grandeza. Esta técnica consiste em utilizar uma boia dentro do tanque de gasolina com um imã e 5 sensores de efeito Hall US1881 fixados na parte externa do tanque. Os sensores de efeito Hall tem seu valor de saída alterado quando a boia presente dentro do tanque passa por

eles, desta forma é possível medir a quantidade de gasolina dentro do tanque sem que exista a necessidade de fios na parte interna do mesmo. A sinal de saída desses sensores é digital, mas para melhor manutenção dos chicotes elétricos este sinal é convertido para um sinal analógico, no qual a frequência recebida pelo microcontrolador é proporcional a quantidade de gasolina presente no tanque. A Figura 2.6 mostra como é montado o *shield* com os sensores de efeito Hall enfileirados.

Figura 2.6: *Shield* criado para sensores de efeito Hall US1881.



Fonte: UDESC Baja Velociraptor (2017).

2.2.5 Temperatura do câmbio CVT

O motor a combustão de um veículo gera torque e potência, porém esta entrega é linear e pode ser ineficaz dependendo do terreno em que o veículo se encontra. Para contornar esta situação e ter sempre o torque desejado, é adicionada uma transmissão ao veículo. A transmissão converte a saída de torque e potência com o objetivo de manter o motor o mais próximo possível da hipérbole de tração ideal (NAUNHEIMER, 2011).

O tipo de transmissão escolhida para o Baja Velociraptor foi uma caixa de câmbio CVT (*Continuously Variable Transmission*). Para a equipe, a informação da temperatura da transmissão é importante para o projeto do veículo pois o câmbio CVT possui componentes sensíveis como a correia utilizada no sistema. Na Figura 2.7 é possível visualizar um exemplo de câmbio CVT, a correia é o objeto pintado de vermelho e o rompimento da mesma por fatores como desgaste excessivo causaria falha total do sistema.

Para obter essas informações do câmbio foi utilizado um sensor MLX90614 (Figura 2.8(a)). Este sensor, da categoria termopilha, opera utilizando o mesmo princípio de um sensor termopar. Um termopar simples é um dispositivo de baixa sensibilidade que responde a mudanças da casa de 1°C com uma mudança de dezenas de μV , um sensor de termopilha é

Figura 2.7: Câmbio CVT.



Fonte: (NISSAN, 2017).

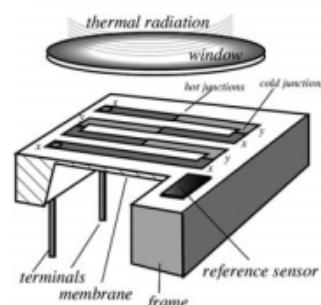
uma cadeia de termopares conectados, tipicamente 50 a 100 junções, para obter um sinal de 50 a 100 vezes mais forte. O sensor opera com uma membrana com uma capacidade térmica baixa devido a seu tamanho, e esta membrana varia de temperatura de acordo com a radiação térmica para que desta forma a temperatura seja calculada gerando uma tensão correspondente (FRADEN, 2016). A Figura 2.8(b) exemplifica este processo descrito. A saída deste sensor é dada em valores analógicos porém vai conectada a uma entrada digital do microcontrolador devido ao uso do protocolo I²C.

Figura 2.8: Sensor MLX90614 e seu funcionamento.



(a) Sensor

MLX90614.

Fonte:(MELEXIS,
2017)

(b) Exemplo do funcionamento.

Fonte:(FRADEN, 2016)

2.2.6 Célula de carga

A célula de carga é um sensor que transforma grandeza física em sinal elétrico. Ele funciona a partir do princípio da variação da resistência ôhmica de um extensômetro quando este é submetido a alguma força (FRADEN, 2016). Esta deformação então é transformada em uma saída de tensão que é capturada pela porta analógica do microcontrolador.

Este sensor foi utilizado para verificar a ação de forças sobre o *link* da suspensão, uma das peças responsáveis por manter os pneus na cambagem correta. O modelo utilizado foi o Z-20.

2.2.7 Deslocamento do amortecedor

O amortecedor, que está contido no conjunto da suspensão, tem o papel de controlar o carro de forma a manter as rodas do veículo em constante contato com o solo, trazendo melhor desempenho para o mesmo. Este sensor de deslocamento de amortecedor é utilizado em bancada para testes e para realização da aquisição da grandeza, neste caso a distância de deslocamento, foi criado um instrumento semelhante a um amortecedor em cano PVC e este instrumento fica acoplado ao lado do amortecedor. Quando o amortecedor sofre tensão o esforço também é transmitido para o instrumento e nele é realizada a aquisição da grandeza. Estes dados adquiridos ajudam na tomada de decisão da configuração da suspensão, escolhendo entre uma suspensão mais rígida ou mais suave dependendo do terreno, calibragem dos pneus, etc.

É utilizado um sensor GP2Y0A21YK0F da SHARP como pode ser visto na Figura 2.9 para fazer a aquisição do deslocamento do amortecedor. Este sensor vai acoplado em uma das pontas do cano, apontado para a outra. A distância efetiva de operação do sensor é de 10cm - 80cm (SHARP, 2017) o que é suficiente para o cenário. O sinal resultante da aquisição é analógico e proporcional a distância do sensor em relação ao objeto em que ele está apontado.

Figura 2.9: Sensor de deslocamento do amortecedor SHARP GP2Y0A21YK0F.



Fonte: (SHARP, 2017).

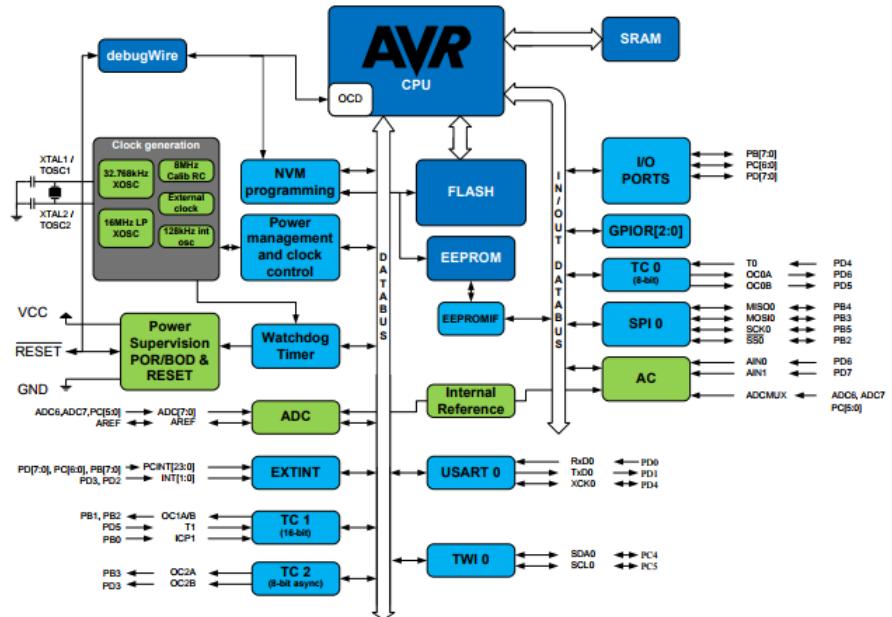
2.3 Microcontroladores

Os microcontroladores são os dispositivos de maior importância dentro de um sistema embarcado, sendo considerados o coração de um projeto que envolva soluções remotas (RANI, 2010). O microcontrolador possui várias funções integradas dentro de um único *chip* e com baixo consumo de energia. Dentro dele é possível encontrar uma unidade de processamento, uma quantidade fixa de memória RAM, memória ROM, pinagem para operações gerais de entrada e saída, entradas/saídas para protocolos seriais como I^2C além de um *timer* (MAZIDI; MCKINLAY; CAUSEY, 2008). A Figura 2.10 é um exemplo a arquitetura de um microcontrolador usado em diversas placas de desenvolvimento, o ATmega328/ATmega328p. Ele utiliza arquitetura Harvard, ou seja, o acesso a memória de dados é separado da memória de programa.

Em Rani (2010) e em Mazidi, McKinlay e Causey (2008) são citados alguns fatores que são levados em conta na escolha do microcontrolador, porém tais critérios podem ser revisados e redimensionados para a aplicação no Baja Velociraptor. Os critérios de maior relevância então são:

- Baixo consumo de energia;
- Quantidade de RAM e ROM suficiente para a rotina desejada;
- Facilidade de desenvolvimento;
- Quantidade suficiente de entradas/saídas para periféricos;
- Programável em C; e

Figura 2.10: Diagrama de blocos do microcontrolador ATmega328/ATmega328p.



Fonte: (ATMEL, 2016).

- Custo.

A equipe Velociraptor atualmente já trabalha com um microcontrolador da Atmel, o ATmega328 em uma placa de desenvolvimento Arduino Nano V3. Este microcontrolador supre todos os critérios citados acima para o cenário atual, com os sensores citados na seção 2.2, porém com a criação de um novo *software* para tratamento de dados e em conjunto com os objetivos futuros da equipe de atualizações nas tecnologia utilizadas, o uso deste microcontrolador deve ser revisado para ser mantido ou descartado. Outro fator discutido com a equipe é o uso de microcontroladores que são suportados por placas de desenvolvimento, revisadas na seção 2.3.1. As placas de desenvolvimento facilitariam a criação do SCOB, e servem para o escopo deste projeto já que são placas desenvolvidas para um ambiente de prototipagem.

O Quadro 2.1 faz uma comparação entre microcontroladores encontrados em placas de desenvolvimento no mercado das marcas Intel¹, Atmel²³⁴⁵ e NXP⁶. O microcontrolador com maior poder de processamento é o AT91SAM3X8E com velocidade de *clock* 5.25 vezes maior que o microcontrolador mais lento e 48 vezes mais memória. O microcontrolador da Intel

¹Quark SE Microcontroller C1000 Datasheet

²ATmega16U4/ATmega32U4 Datasheet

3 ATmega328/P Datasheet

⁴ Atmel ATmega640/V-1280/V-1281/V-2561/V Datasheet

⁵SAM3X / SAM3A Series Datasheet

⁶NXP Kinetis KL25 Sub-Family Datasheet

possui uma boa quantidade de memória RAM, porém sua pinagem é menor que do ATmega328, o microcontrolador de menor desempenho da família Atmel visto neste comparativo.

Quadro 2.1: Comparação de microcontroladores.

Modelo	Fabricante	Clock	I/O Digital	I/O Analógica	RAM
AT91SAM3X8E	Atmel	84 MHz	54	14	96 KB SRAM
ATmega2560	Atmel	16 MHz	54	16	8 KB SRAM
ATmega328	Atmel	16 MHz	14	8	2 KB SRAM
ATmega32u4	Atmel	16 MHz	20	12	2.5 KB SRAM
MKL25Z128VLK4	NXP	48 MHz	60	6	16 KB SRAM
Quark SE C1000	Intel	32 MHz	14	6	80 KB SRAM

Fonte: Autor.

2.3.1 Placas de desenvolvimento

O mercado atual de sistemas embarcados oferece uma opção para melhor produtividade e facilidade de desenvolvimento, a custo da especificidade envolvida nos projetos. As placas de desenvolvimento são componentes criados para suportar um ambiente com uma unidade de processamento, geralmente um microcontrolador, um cristal gerador de *clock* e uma entrada para interfaceamento com o computador (USB ou mini-USB), outros periféricos também podem ser encontrados de fábrica dependendo do modelo, como acelerômetros e giroscópios.

Estas placas de desenvolvimento possuem diversos microcontroladores, fabricantes, tamanhos, poder de processamento, poder de armazenamento e assim por diante. O Quadro 2.2 traz uma comparação entre várias placas de desenvolvimento que utilizam microcontroladores vistos no Quadro 2.1. A dimensão é dada em milímetro, todas as placas tiveram seus preços verificados no mês de novembro do ano 2017 em um *e-commerce* de componente eletrônicos⁷. A dimensão de altura das placas Arduino são estimativas, visto que oficialmente estes números não são divulgados.

É possível tirar algumas análises deste quadro, por exemplo, o Arduino Nano V3 possui o mesmo poder de processamento de um Arduino Uno R3 porém utiliza uma área aproximadamente 4.5 vezes menor do que o mesmo, custando uma pequena margem a mais. Levando em conta o cenário do projeto, as **dimensões** de uma placa de desenvolvimento é um fator que também deve ser levado em conta na tomada de decisão, o que não se aplicava a microcontroladores pois estes possuem um tamanho pequeno em comparação com placas de desenvolvimento

⁷<https://www.filipeflop.com/>

Quadro 2.2: Comparaçao entre placas de desenvolvimento

Modelo	Fabricante	Microcontrolador	Dimensões	Preço
FRDM-KL25Z	NXP	MKL25Z128VLK4	80 x 53 x 6	R\$169,90
Genuino 101	Intel	Quark SE	70 x 55 x 20	R\$229,90
Uno R3	Arduino	ATmega328	68.6 x 53.4 x 25	R\$44,90
Nano V3	Arduino	ATmega328	45 x 18 x 7.6	R\$49,90
Mega 2560 R3	Arduino	ATmega2560	102 x 54 x 25	R\$74,90
Due	Arduino	AT91SAM3X8E	101.52 x 53.3 x 25	R\$139,90
Leonardo R3	Arduino	ATmega32u4	68.6 x 53.3 x 25	R\$59,90

Fonte: Autor.

que possuem alguns componentes extras já citados no início da seção. Algumas placas como Genuino 101, Due e FRDM-KL25Z possuem um preço que varia acima da casa dos R\$100, isto pode ser justificado com desempenho superior em relação as demais placas e alguns periféricos extras já inclusos nestas placas, como acelerômetro em ambas FRDM-KL25Z e Genuino 101. Placas com dimensões como o modelo Mega 2560 R3 não são a melhor opção para ser utilizadas no veículo para uso durante a prova visto a impossibilidade de criar um *shield* para proteção da mesma e acoplamento no painel dianteiro do veículo. A Figura 2.11 mostra como está estruturado o painel atualmente com um Arduino Nano V3. Uma mudança de tamanho requisitaria uma revisão do projeto de eletrônica do veículo.

Figura 2.11: Foto do painel dianteiro atual do Baja Velociraptor.



Fonte: Autor.

Todavia, além das placas de desenvolvimento citadas, Andria (2015) traz uma solução de aquisição de dados para veículos (mais informações do trabalho no capítulo 3) com o uso de placas de melhor desempenho, com maior número de periféricos porém ainda em uma

escala reduzida. Este tipo de placa é comumente chamado de *Single Board Computer* por serem capazes de rodar um sistema operacional completo, e não apenas um programa como nas placas analisadas no Quadro 2.2. O Quadro 2.3 tem a comparação de algumas destas *Single Board Computer* na qual as dimensões são dadas em milímetros, todas as placas tiveram seus preços verificados no mês de novembro do ano 2017 em um *e-commerce* de componente eletrônicos⁸.

Quadro 2.3: Comparação entre *Single Board Computer*

Modelo	Fabricante	Processador	Clock	I/O Digital	I/O Analógica	RAM	Dimensões	Preço
3 Model B	Raspberry Pi	BCM2837	1200 MHz	40	N/A	1 GB LPDDR2	85 x 56 x 17	R\$299,90
Zero W	Raspberry Pi	BCM2835	1000 MHz	40	N/A	512 MB LPDDR2	65 x 30 x 5	R\$79,90
Green	BeagleBone	AM3358	1000 MHz	65	7	512 MB DDR3	86 x 53 x 19	R\$329,90
Black Rev.C	BeagleBone	AM3358	1000 MHz	65	7	512 MB DDR3	86 x 53 x 19	R\$399,90

Fonte: Autor.

Os modelos 3 Model B, Green e Black Rev.C possuem um preço acima da faixa de R\$250 e dificilmente serão escolhidos para o projeto, porém de todas as *Single Board Computer* comparadas, o modelo Zero W tem a proposta mais atraente. Com memória RAM de 512 MB, velocidade de *clock* de 1.000 MHz, dimensões 65 x 30 x 5 milímetros e preço R\$79,90 esta *Single Board Computer* possui poder de processamento e memória acima dos modelos vistos no Quadro 2.2 por um preço abaixo da casa de R\$100. A utilização dessas placas ainda trazem vantagens como a possibilidade de rodar um sistema operacional dentro do veículo, o que pode ser explorado para melhor processamento de dados dentro do SCOB, utilização de linguagens interpretadas e utilização de sistemas gerenciadores de bancos de dados para adição de uma camada extra de confiabilidade nos dados entregues pelo sistema. A desvantagem da utilização deste sistema é o custo, que aumenta uma margem pequena (cerca de 1.6 vezes o valor atual da placa de desenvolvimento), o aumento da complexidade do sistema com a adição de um sistema operacional dentro do veículo, o aumento das dimensões da placa de desenvolvimento (de 45 x 18 x 7.6 para 65 x 30 x 5) e a falta de entradas analógicas no modelo Zero W sendo assim necessário o uso de módulos ADC externos.

2.4 ZigBee

O protocolo ZigBee de conexão sem fio foi criado em 1998 e padronizado em 2003. Atualmente é mantido pela *ZigBee Alliance*¹ e o sistema tem como foco atingir as metas de (GISLASON, 2008):

⁸<https://www.filipeflop.com/>

¹Site: <<http://www.zigbee.org/>>

Quadro 2.4: Comparação de alguns parâmetros dos protocolos 802.11b e 802.15.4.

Parâmetro	Wifi (802.11b)	ZigBee (802.15.4)
Faixa de Frequência	2.4Ghz	2.4Ghz
Alcance	35-140m	~100m
Consumo de energia em uso	400mA	~30mA
Consumo de energia em standby	20mA	3µA
Taxa de transferência	11Mbps	250Kbps

Fonte: Autor.

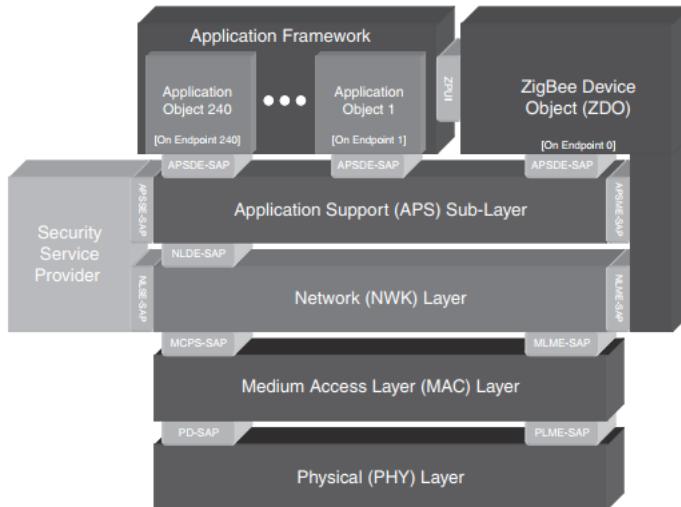
- Ser altamente confiável;
- Ter custo benefício alto;
- Execução em baixa energia;
- Conseguir reter alto padrão de segurança;e
- Ter um padrão global de execução.

Estas metas são atingidas com um porém. O ZigBee não atua em taxas de transferência de dados de outros protocolos como o WiFi por exemplo. O Quadro 2.4 apresenta uma comparação entre alguns parâmetros destes dois protocolos, o que deixa explícito que ambos possuem enfoques diferentes e são utilizados para aplicações diferentes.

O ZigBee possui uma arquitetura própria que se encaixa parcialmente no modelo de rede OSI de 7 camadas(GISLASON, 2008). Alguns elementos são os mesmos como a camada física, a camada de enlace e a camada de rede. As outras camadas restantes do modelo OSI são remodeladas nas camadas *ZigBee Device Object* e *Application Framework*, a Figura 2.12 é um diagrama exemplificando o funcionamento das camadas do protocolo com algumas informações a mais como a comunicação entre as camadas por *Service Access Points*.

Quanto a topologia de rede, o ZigBee opera em topologias de estrela, malha e árvore. Entretanto, independente de topologia, o ZigBee atua sobre uma *Personal Area Network* (ou PAN) criada em um *identifier* (ID) específico, esta PAN possui os nós (módulos físicos do ZigBee) onde cada nó da topologia pode ser um dos três tipos de nós do protocolo, sendo eles o coordenador, o roteador e o *end-point* (ELAHI; GSCHWENDER, 2009). O coordenador tem a responsabilidade de criar a rede PAN, o roteador repassa pacotes enviados de *end-points* e os ditos *end-points* são os nós que possuem os sensores e atuadores da rede, uma característica

Figura 2.12: Diagrama com camadas do protocolo ZigBee.



Fonte: (GISLASON, 2008).

específica deste tipo de nós é que os mesmos podem entrar em *stand-by* para consumir menos energia quando não estão operando.

2.4.1 Estrutura de envio

O ZigBee possui uma estrutura definida para o envio de dados sem fio. O quadro de dados do protocolo possui alguns *stamps* para envio dos dados de forma que o destino das mensagens possa ser definido no envio, possua um indicador de força do sinal com o byte chamado de RSSI (*Received Signal Strength Indicator*) e possibilita o envio de mensagens de sucesso de transmissão.

A Figura 2.13 exibe alguns quadros que podem ser enviados nos módulos XBee, da empresa *Digi International*. É possível observar na estrutura dos quadros alguns *stamps* obrigatório que permitem que as características citadas sejam possíveis. O primeiro byte é o delimitador de início do quadro, ele indica que a partir daquele byte existe um quadro válido, os dois próximos bytes são referentes ao início das informações do quadro (*Most Significant Byte*) e ao fim das informações (*Least Significant Byte*). Os próximos N bytes são referentes aos dados do quadro, que variam de acordo com o tipo de quadro enviado, contudo, esta parte inicia com um byte de identificação e prossegue com seus N bytes de dados. Por fim é enviado um byte de *checksum*, este byte é utilizado para verificar a integridade dos componentes (ERDMANCZYK, 2014).

Figura 2.13: Alguns quadros de envio dos modelos XBee.

	1	2	3	4		n	n+1	
Generic	0x7E	MSB	LSB	Frame Data		Chksm		
				API cmd	cmdData			
	1	2	3	4	5	6		
Modem Status	0x7E	MSB	LSB	0x8A	Status	Chksm		
	1	2	3	4	5	6	7	8 .. n n+1
AT Command	0x7E	MSB	LSB	0x08	Frame ID	AT Command	Parameter Value	Chksm
	1	2	3	4	5	6	7	8 .. n n+1
AT Command Response	0x7E	MSB	LSB	0x88	Frame ID	AT Command	Status	Parameter Value Chksm
	1	2	3	4	5	6	7	8 .. n (<100) n+1
Transmit Request (16-bit addr)	0x7E	MSB	LSB	0x01	Frame ID	Dest Addr	Options	RF Data Chksm
	1	2	3	4	5	6	7	
Transmit Status (16-bit addr)	0x7E	MSB	LSB	0x89	Frame ID	Status	Chksm	
	1	2	3	4	5	6	7	
Received Message (16-bit addr)	0x7E	MSB	LSB	0x81	Source Addr	RSSI	Options	RF Data Chksm
	1	2	3	4	5	6	7	

Fonte: (ERDMANCZYK, 2014).

2.5 Software

O *software* é parte essencial deste trabalho. Todas as grandezas adquiridas pelos sensores apresentados na seção 2.2 e concentrados nos microcontroladores revisados na seção 2.3 devem ser submetidas a um *software* para transformação destes dados em informações. As seções a seguir revisam qual tarefa cada sub sistema deve realizar para alcançar os objetivos propostos na seção 4.1.

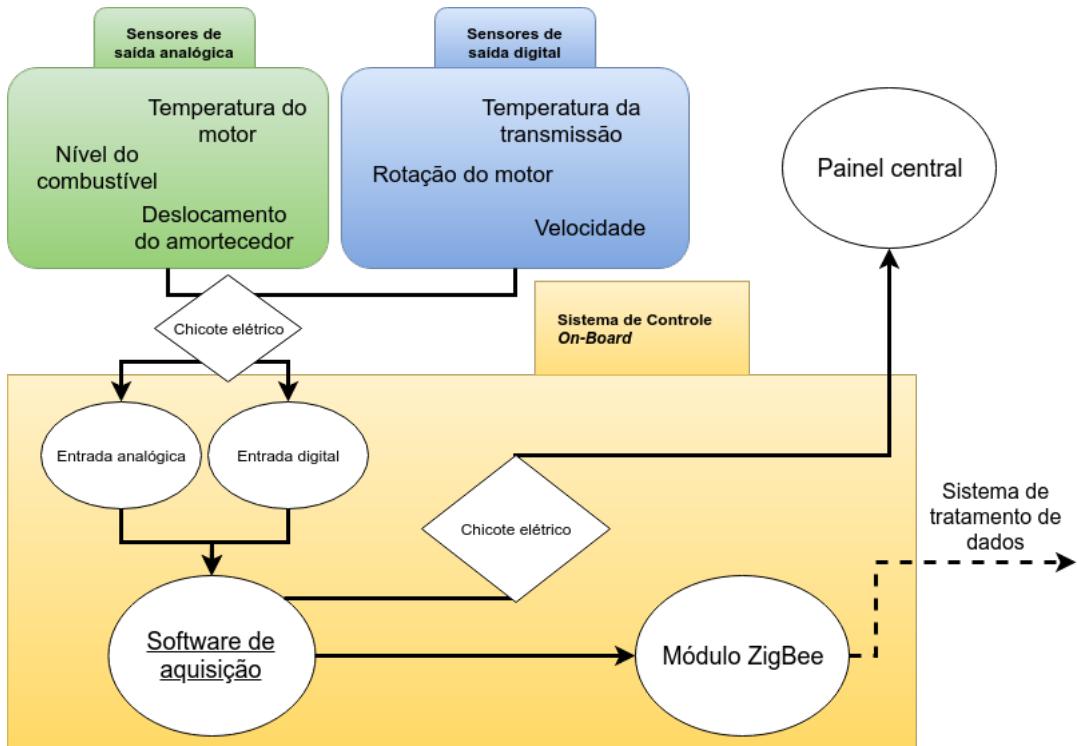
2.5.1 Aquisição de dados

O *software* de aquisição de dados é a primeira interação dos dados vindos dos sensores com algum tipo de tratamento. Este subsistema não é o foco deste trabalho, porém ele deve ser revisado pois caso o *software* principal traga alguma mudança substancial na forma de tratamento dos dados, os subsistemas de aquisição dentro dos microcontroladores deve ser atualizado para satisfazer quaisquer novos requisitos.

Este subsistema é produzido por três motivos: ele deve realizar a comunicação entre múltiplos microcontroladores em caso de uso de arquitetura distribuída, realizar qualquer cálculo inicial para demonstração de informações no painel e ele deve preparar os dados para passagem para o subsistema de tratamento de dados (KATZOURAKIS, 2012)(ARAÚJO, 2006)(ANDRIA, 2015)(DIAS, 2010)(NUNES, 2016). A Figura 2.14 demonstra como este subsistema funciona, os sensores capturam as grandezas desejadas e enviam os dados para o SCOB via chicote elétrico. As saídas dos sensores são conectadas nas entradas digitais ou analógicas do microcontrolador. O *software* atual é feito em C e utiliza a biblioteca padrão do Arduino,

visto o uso do Arduino Nano V3. Os dados são armazenados em variáveis e a cada ciclo do programa, os mesmos são enviados via ZigBee.

Figura 2.14: Diagrama exemplificando funcionamento do sistema de aquisição.



Fonte: Autor.

2.5.2 Tratamento de dados

O sistema de tratamento de dados deve receber as informações fornecidas pelo meio de transporte de informação, neste caso o módulo ZigBee, e com esses dados atualizar o ambiente e informar a equipe dos boxes dos dados mais importantes sobre o veículo. Este sistema pode ser feito para dados recebidos posteriormente a testes ou com telemetria, sendo feita análises em tempo real. Algumas análises de dados que podem ser realizadas neste *software* seriam torção do chassi, temperatura do motor em relação a velocidade do veículo, tempo de prova, entre outras informações.

As equipes de automobilismo não divulgam muitas informações sobre seus sistemas de tratamento de dados. A exceção é quando estes sistemas utilizam de plataformas de empresas privadas para codificação destes *software*, nestes casos os artigos comentam que estas plataformas são utilizadas para criação da GUI (Graphical User Interface) e geralmente usadas em conjunto com dispositivos de aquisição de dados também adquiridos de empresas

privadas (TAHA, 2010)(CHANDIRAMANI; BHANDARI; HARIPRASAD, 2014)(ANDRIA, 2015)(ANDRIA, 2016). Contudo estas informações não são relevantes para o trabalho proposto na área de tratamento de dados, pois o sistema que deve ser desenvolvido para este fim deve ser projetado e executado junto a equipe Velociraptor com um requisito de baixo custo monetário. Um artigo revisado produziu sua própria plataforma de tratamento de dados em C e C++ (KAT-ZOURAKIS, 2012) e utiliza de alguns conceitos comentados na Seção 2.5.1, como utilização de *comma-separated values* para formatação dos arquivos de dados.

3 Trabalhos Relacionados

Nesta seção são discutidos alguns trabalhos que possuem um objetivo similar ao sistema produzido ou utilizam tecnologias similares.

O trabalho de Calderón, Ruiz e Bohórquez (2013) propõe uma solução de telemetria para um veículo de competição elétrico. A proposta se assemelha ao trabalho proposto no quesito de manter a equipe em questão atualizada dos dados provindos do carro, a diferença é que o veículo é movido a energia limpa, elétrica. Este requisito altera também algumas das grandezas a serem analisadas pelo sistema, nele são verificados fatores como amperes por hora, voltagem, velocidade e distância percorrida. Infelizmente o trabalho não comenta como é feita a coleta dos dados pelos sensores, apenas comentando que existe um equipamento que faz o mesmo. Para parte de telemetria, os desenvolvedores comentam que trabalharam com foco em resolver dois problemas: robustez sobre todo o circuito de provas e segurança dos dados, além de redução do ruído. Portanto duas tecnologias foram analisadas para a transmissão dos dados embarcados do veículo até os computadores da equipe, a de rádio frequência e a rede móvel de celular (GSM). A última foi escolhida sobre a outra revisada devido a "(...) Impossibilidade de garantir a comunicação entre todos os circuitos devido aos formatos e obstáculos encontrados no terreno..."(CALDERÓN; RUIZ; BOHÓRQUEZ, 2013), também é comentado que foram realizados testes e todos os circuitos possuem cobertura de sinal móvel. Por último é desenvolvido um *software* para receber os dados provenientes do veículo. O sistema de aquisição de dados teve seu funcionamento dividido em quatro blocos, sendo eles: configuração do aplicativo e do canal transmissão de dados usado; informações específicas da direção do piloto e do circuito percorrido; valores numéricos dos dados técnicos mais importantes para a manutenção do veículo em tempo real; representação gráfica de toda a informação recebida durante todo o processo da prova. Com isto e o tratamento das informações, o programa apresenta para a equipe de boxes informações como:

- Consumo de energia;
- Voltagem da bateria;
- Velocidade;

- Distância percorrida;
- Eficiência energética.

Todas estas informações disponíveis são relevantes, porém o artigo não se entra em detalhes quanto a construção do sistema de aquisição, o que poderia aumentar em muito a relevância deste trabalho para o projeto proposto. Como é comentado, existe um indício de emprego de técnicas de engenharia de *software* na criação do sistema de aquisição de dados, porém por não ser o foco do trabalho, as mesmas não são citadas.

Outro trabalho que também possui um carro movido a energia limpa e propõe um sistema de aquisição de dados é o de Taha (2010). O sistema é desenvolvido para um carro que utilizada um motor elétrico e deve ser capaz de percorrer distâncias de mais de 3000 quilômetros no evento *World Solar Challenge*. O equipamento utilizado pela equipe foi o CompactRIO em conjunto com o *software* Labview da National Instrument para aquisição dos dados e a criação da plataforma de tratamento de dados, respectivamente, além do módulo de transmissão por rádio frequência MaxStream (atualmente Digi) Xtream. Como todos os dispositivos usados pela equipe são fornecidos por uma fabricante externa, pouco é discutido sobre como os sensores operam no ambiente. O artigo demonstra um pouco sobre a arquitetura do sistema montado, utilizando os equipamentos citados e discute os resultados, também não comentando sobre como foi feita a abordagem de criação do *software* e que requisitos ele deve suprir. Um dado interessante visto neste artigo é que para coletar dados de seis termopares, dois transdutores de corrente, um grupo de bateria e um tacômetro, foi necessário 363,3 kilobytes de dados por hora, assim um cartão SD de dois gigabytes seria suficiente para armazenar uma longa bateria de treinos.

O artigo de Katzourakis (2012) é mais abrangente, ele utiliza um sistema de aquisição de dados e telemetria para estudar o comportamento de motoristas ao volante de carros convencionais. O estudo de comportamento visto no artigo não é verificado por não ser o foco, porém a parte de instrumentação faz algumas menções relevantes. Os testes feitos para as análises de dirigibilidade possuem alguns sensores construídos pelos autores, como o de posição do acelerador e sensores para cada roda a fim medir sua velocidade individual (útil em casos de derrapagem), e outra parte dos dados são capturados com um equipamento chamado Racelogic VBOX, ele mede alguns dados como aceleração de 0 a 100 e distância percorrida utilizando GPS. Para as entradas analógicas foi comprado um sistema de aquisição de dados da National Instruments modelo USB-6211 USB M Series (ou, como é chamado no artigo, NI-DAQ), onde

tais entradas são direcionadas a ele. Para os sensores de velocidade das rodas foi utilizada uma placa de prototipagem AVR-P40-USB-8535 da Olimex em conjunto com um microcontrolador Microchip da família ATMega32. O *software* criado tem o código produzido na linguagem C/C++, e o programa tem como objetivo se comunicar com a placa de prototipagem AVR e o sistema de aquisição de dados NI-DAQ, além de outros periféricos citados no texto do artigo. Tendo estes dados, o *software* trata esses valores e disponibiliza para o usuário em tempo real, além de armazenar os dados em um arquivo de texto para análises posteriores. O *software* tem taxa de atualização de 100 Hz, sendo a taxa de amostragem do NI-DAQ de 100Hz e a taxa de amostragem dos sensores de roda de 20Hz no qual o valor final mostrado ao usuário é o último dado recebido dos sensores. O artigo apresenta um diagrama, esquematizando o funcionamento do *software*, além da parte de sensoriamento ainda existem os algoritmos criados para o estudo de comportamento, aumentando o nível de complexidade total do sistema. Apenas o diagrama não seria suficiente para manter a equipe de desenvolvimento atualizada do progresso do *software* proposto, mas como o foco do texto não está na engenharia de *software*, esta parte não se encontra discutida por completo. As taxas de atualização apresentadas nesse artigo dão uma ideia de um problema que será encontrado no trabalho proposto, alguns sensores podem ter taxas de atualização diferentes de outros (como temperatura do motor e RPM) o que deve ser ajustado com filtros no *firmware* do SCOB que será desenvolvido em conjunto com a equipe Velociraptor.

Outro artigo revisado foi Chandiramani, Bhandari e Hariprasad (2014), este traz uma solução de aquisição de dados com um SCOB customizado para fórmula SAE, porém o sistema é de uso geral na área veicular, podendo ser instalado em quaisquer outros modelos. O artigo não é muito específico em como e quais sensores são usados, ele coloca alguns exemplos como um sensor para temperatura (LM35) e como é feito um filtro matemático a partir da mediana de 100 valores para obter um resultado mais confiável. O foco do artigo é na parte de telemetria, nesta área é discutida uma solução sobre qual protocolo *wireless* utilizar para o fim de sensoriamento veicular. Algumas opções são descartadas no começo, como *Bluetooth* e infravermelho, devido ao alcance limitado e a necessidade de manter contato direto entre os nós, o que é inviável em um circuito automobilístico. Então foram estudados dois outros protocolos, o WiFi e ZigBee. O artigo comenta que ambos possuem alcance mínimo para o cenário, ambos trabalham na frequência 2.4GHz e podem ter seus dados encriptados. Contudo o ZigBee foi escolhido devido a melhor relação de consumo de energia, além de segundo o autor, ser mais simples de se instalar uma malha de rede ZigBee. A vantagem do WiFi é que o mesmo possui

maiores taxas de transferência, porém no sistema analisado este requisito não era prioritário.

Um estudo feito por Araújo (2006) traz várias informações sobre o sistema eletrônico do Paraibaja, equipe de Baja da Universidade Federal de Campina Grande. O artigo não é focado em telemetria e tratamento de dados, portanto não existe *software* para ser analisado nestes dois campos, mas este artigo comenta como a equipe fez para montar seu painel de instrumentos e o *firmware* que está contido nele, este mesmo possui alguns sensores e técnicas de utilização interessantes que apesar da data devem ser levados em consideração. A equipe trabalhou com o velocímetro, tacômetro, termômetro, medidor de nível de combustível e indicador de nível de óleo. Todos esses sensores foram ligados a um microcontrolador AT89S8252 para manipulação dos dados e exibição das medidas no *display* do painel. Para a visualização destas mesmas informações o sistema conta com um *display* de sete segmentos e LEDs indicadores (para RPM e superaquecimento do motor). O artigo então comenta como são utilizados os sensores para retirada de informação, o termômetro por exemplo, utiliza uma sonda de alumínio acoplada ao cárter do motor, mesmo processo utilizado atualmente pela equipe Velociraptor. Essa sonda, além de ser feita de alumínio, também tem seu interior preenchido com pasta térmica, o que garante uma boa condutividade térmica e uma medição mais próxima do valor real. Outro dado interessante é que o valor considerado para superaquecimento do motor é 120°C, caso este valor seja ultrapassado, um LED indicativo no painel é acendido para informar o piloto. Saber este valor é essencial para a escolha dos sensores de temperatura, pois os sensores possuem uma faixa de medição que deve ser levada em consideração antes de sua escolha. Ambas as medição de combustível e do nível do óleo são feitas com boias dentro dos seus respectivos recipientes. Esta técnica é comumente utilizada em carros de rua, porém novas regulamentações do Baja SAE (SAE, 2017) não permitem o uso de cabos e fios na parte interna de recipientes como o tanque de gasolina, inviabilizando esta técnica em veículos atuais da categoria. Para fazer a medida das revoluções por minuto do motor para tacômetro, é utilizada uma técnica que tira proveito da vela, equipamento que libera a centelha para iniciar a combustão interna do motor. Como o motor possui quatro ciclos definidos (admissão, compressão, explosão e exaustão), a vela libera a centelha para a queima no momento em que o combustível está comprimido, assim causando a explosão que movimenta o pistão que consequentemente movimenta o virabrequim. Neste processo a vela necessita de uma tensão de aproximadamente 12KV e este pico pode ser capturado por uma bobina, este sinal capturado pode ser convertido em RPM se estiver dentro de uma janela de tempo adequada. O sistema foi testado em pista nas provas e é reportado que ocorreram mal contato e vários *bugs* no *firmware*. Os autores então

fazem algumas recomendações sobre as adversidades encontradas em um evento da Baja SAE (estes detalhados anteriormente na seção 1) e recomendam que o sistema de acoplamento com os sensores deve ser isolado e a caixa que acomoda o SCOB deve possuir blindagem eletrostática, junto com vedação contra água e lama.

Dois artigos do mesmo autor foram estudados ((ANDRIA, 2015) e (ANDRIA, 2016)) e possuem temas semelhantes e pertinentes, porém Andria (2015) será focado no texto a seguir. Este sistema de aquisição de dados e monitoramento com telemetria foi especificamente produzido para carros convencionais, pois ele utiliza uma entrada de leitura de dados padrão na maioria dos carros convencionais de rua. A *On-Board Diagnostics* (Figura 3.1) é uma entrada presente geralmente na parte inferior do painel dos veículos e fornece informações diretamente da unidade de controle do motor (*Engine Control Unit* ou ECU) do mesmo. Além de ser uma entrada padrão, os protocolos de interfaceamento para a aquisição dos dados também são padronizados e independente de montadora, sendo esta porta muito utilizada em oficinas para aquisição de diagnósticos preliminares do veículo. Além desta fonte de informações, o autor também utiliza um sensor de movimento MPU6050 para medir aceleração lateral e velocidade angular. Todos estes dados são enviados para um SCOB que diferente da maioria dos outros artigos aqui citados, consiste em um Raspberry Pi. Isto é importante pois o artigo cita algumas vantagens de se ter uma *PC Board* para processamento dos dados recebidos. Uma das vantagens é o poder de processamento superior em relação a um microcontrolador o que dependendo dos requisitos do projeto pode ser fundamental. Outra vantagem é a possibilidade de utilização de linguagens interpretadas como Python e Ruby, tudo graças ao ambiente que suporta um sistema operacional completo. Uma última vantagem comentada é a possibilidade de uso de um sistema gerenciador de bancos de dados integrados com o SCOB para *backup* de informações e otimização do uso de memória para armazenamento de dados, além de aumento na confiabilidade dos mesmos. Porém uma desvantagem que não é levantada mas deve ser levada em consideração é o aumento do custo monetário do projeto, com o preço de um Raspberry Pi 3 Model B sendo aproximadamente dez vezes o preço de um Arduino Nano V3 (Novembro de 2017). O *software* desenvolvido para este projeto utiliza Labview e funciona via internet com uma interface web. Os dados são guardados dentro do SCOB e quando requisitado pelo computador são baixados, caso o computador não consiga fazer o carregamento das informações do SCOB, ele utiliza os dados presentes no servidor local. Existe um sistema de usuários para controle das informações e os dados dos sensores são divididos em categorias e podem ser visualizados em tabelas e gráficos. Nos testes, a taxa de amostragem de dados foi fixada em 100Hz e é comentado no artigo

que o gargalo do sistema nesse quesito é o sistema *On-Board Diagnostics*, que varia de carro para carro. Não é comentado nenhum tipo de planejamento ou análise de requisitos na parte do *software* e também não é apresentado nenhum diagrama explicando o sistema corrente.

Figura 3.1: Exemplo de entrada *On-Board Diagnostics*.



Fonte: Autor.

Uma discussão mais profunda sobre o uso destas placas e de outras é feita na seção 2.3.1.

Os últimos trabalhos a serem revisados são os que tiveram propostas mais relevantes e similares às propostas neste trabalho. Os trabalhos de conclusão de curso analisados ((DIAS, 2010), (NUNES, 2016) e (PEREIRA, 2012)) tem propostas para criação de um sistema de telemetria para a modalidade Baja/fórmula SAE com um cenário similar ao deste trabalho. Primeiramente, Dias (2010) tem como foco a pesquisa, projeto e execução da parte de *hardware* do sistema de telemetria, deixando a parte de *software* para um segundo trabalho. Já Nunes (2016) utiliza parte do que já foi projetado em outros anos na equipe Car-Kará para projetar um sistema completo de telemetria com duas ECUs, incluindo *software* e *hardware*. Por último Pereira (2012) traz uma abordagem diferenciada na qual é proposta uma revisão do projeto de elétrica da equipe de fórmula SAE da USP para introdução de conceitos de engenharia de *software*.

Em Dias (2010) é construído um sistema protótipo para toda a parte de eletrônica, instrumentação e telemetria do veículo UFVbaja. Os sensores que são instalados no veículo

são de temperatura, combustível, bateria, velocidade e rotação do motor. O texto traz uma revisão de conceitos de eletrônica, principalmente da parte de microcontroladores, visto que essa é a área abordada pelo trabalho. É comentado sobre conversores analógico-digital, comunicação serial assíncrona, microcontroladores, USB e transceptores sem fio. Este último possui algumas informações sobre o protocolo ZigBee, este que é usado para enviar e receber dados entre os sistemas do veículo e dos boxes. É comentado que o protocolo oferece um excelente imunidade a interferências e a pode hospedar mais de 65 mil dispositivos, além de possuir taxas de transferências de dados entre 20Kbps e 250Kbps. Depois de explicar estes conceitos iniciais, o trabalho explica quais e como funcionam os transdutores utilizados no mesmo. Para grandeza de velocidade e RPM foi escolhido um transdutor indutivo acoplado no eixo do motor para a rotação e na roda dianteira para a velocidade. Um fato curioso é que no veículo Baja da equipe Velociraptor, a medida da grandeza é feita na roda traseira, não na dianteira, isto é interessante pois aqui é comentado que a medida é tirada da roda dianteira devido ao fato do motor estar acoplado ao eixo traseiro e em casos de derrapagem o medidor de velocidade iria indicar um falso valor, porém também existe o problema de manutenção na roda dianteira, visto que acoplar sensores no eixo frontal é mais complicado por questões de apoio para os sensores. Para o nível de combustível é utilizado um transdutor de pressão diferencial, este fica acoplado na derivação da mangueira de combustível entre o tanque e o carburador do motor. O cálculo do nível de combustível é feito levando em conta a pressão que a gasolina exerce no fundo do tanque. Para medir a temperatura do motor é utilizado um sensor LM35 na carcaça externa do motor. Para se obter a tensão da bateria não existe um sensor específico, porém o trabalho utiliza o sinal proveniente da bateria, conectado diretamente em um canal ADC do microcontrolador. No *firmware* do microcontrolador é realizado um cálculo com uma constante restauradora e assim é obtido o decaimento da tensão. Este cálculo de tensão da bateria não é gerado atualmente no Velociraptor e pode ser adicionado ao *software* de tratamento de dados porém como um extra para testes, pois a bateria usada é calculada previamente para durar o tempo de prova. A configuração do SCOB possui um microcontrolador PIC16F877A e um PIC16F628A, ambos da Microship, a programação dos mesmos é feita em linguagem C. Para utilizar o protocolo ZigBee, é empregado o módulo Xbee-PRO. Este módulo tem potência de transmissão de 100mW e alcance de 1,6 quilômetros sem contar com obstáculos segundo o autor. O sistema de boxes é independente e portátil, contando com um painel LCD de duas linhas e 16 colunas e LEDs para alertar sobre alguns dados específicos. Apesar de funcionar sem um computador, os dados recebidos pelo modulo Xbee-PRO podem ser enviados pelo microcontrolador para um computador via USB, e lá eles podem

ser tratados para análise. Após estabelecida conexão com USB, são enviados pacotes com letras (que indicam a grandeza) e números (que indicam os valores medidos) e cada grandeza é separada por um caractere especial "\$". O trabalho é finalizado com uma discussão de resultados e algumas sugestões de aprimoramento como troca dos microcontroladores citados por outros com maior numero de periféricos além de maior poder de processamento e melhoria de precisão da medida do nível do combustível. Ambas as sugestões devem ser acatadas com a utilização dos microcontroladores mais recentes disponíveis no mercado e um método de medida de nível de combustível diferente do citado no trabalho (detalhes na seção 2.2.4).

O trabalho de Nunes (2016) cria um sistema para aquisição de dados de um veículo Baja, utilizando protocolo de comunicação CAN entre os múltiplos SCOBs presentes. O sistema tem uma proposta de utilização de arquitetura distribuída na qual são utilizados dois microcontroladores para aquisição de dados, diferentemente da maioria dos trabalhos revisados até agora. Um fica na parte dianteira e outro na traseira, também existe um terceiro microcontrolador, porém este só reune as informações mais importantes e disponibiliza em um painel digital. O trabalho cita algumas vantagens de se usar um sistema distribuído como diminuição no tamanho do chicote elétrico devido ao SCOB estar próximo aos sensores, facilidade de inserir novos módulos e maior robustez contra falhas elétricas. O último fator citado (robustez) é questionável pois a adição de novos sistemas aumentam a complexidade da manutenção do equipamento. Também são citados pontos negativos na utilização desta arquitetura, estes seriam o maior custo monetário para o projeto e maior complexidade de preparo do equipamento com uso do protocolo CAN. Os sensores utilizados no trabalho trabalham com grandezas como RPM, temperatura, nível do combustível, tensão da bateria e velocidade. As técnicas de medida das grandezas dos sensores de RPM, temperatura do motor, tensão da bateria e velocidade do veículo são as mesma das citadas no trabalho de Dias (2010), a única diferença encontrada está no nível de combustível, que é medido com um boia automotiva. Depois é comentado sobre o uso do protocolo CAN, tempo de bit e quadro de dados de uma mensagem do protocolo. O autor então comenta sobre a implementação e experimento, falando sobre os testes realizados sobre o sistema em bancada e em campo. O código dos SCOBs é feito em C e são utilizados dois Arduino Uno para a aquisição de dados dos sensores e um Arduino Mega para disponibilização dos dados no painel principal. O trabalho não possui um *software* para tratamento dos dados recebidos e análise dos mesmos e também, contrário ao que o título indica, não existe um sistema de telemetria para os dados, apenas da aquisição dos mesmo. A ideia de utilização de múltiplos microprocessadores pode ser levada em consideração, principalmente devido ao ta-

manho de algumas plataformas como Arduino Nano e Raspberry Pi Zero, uma discussão sobre as plataformas de prototipagem e microcontroladores utilizados pode ser vista na seção 2.3.

O último trabalho é de Pereira (2012). Diferentemente dos outros, este não tem como objetivo criar um sistema de aquisição de dados de sensores de forma alguma, mas ele utiliza um sistema já concebido previamente e sugere conceitos de engenharia de *software* que podem ser empregados neste meio do automobilismo. O objetivo em utilizar estes conceitos é adotar um padrão para ser seguido e facilitar a adequação do *software* a evoluções. São apresentados alguns modelos de processo como modelo em cascata, desenvolvimento evolucionário e engenharia baseada em componentes, e por fim é demonstrado modelo feito especificamente para atender as necessidades desse projeto. O modelo criado possui uma execução de processos em círculo, aonde é inicialmente feita uma especificação do *software*, em cima dela é produzido um código, depois são realizados testes e por último é realizada uma manutenção, para logo após reiniciar o ciclo para implementar novas funcionalidade. As quatro fases citadas acima são especificadas no texto, a fase de especificações do *software* tem os requisitos do mesmo levantados e avaliados, ela pode ser detalhada em quatro outras fase, de estudo de viabilidade, de elicitação e análise de requisitos, especificações dos requisitos e validação dos requisitos. A etapa de projeto de *software* faz uma análise também dividida em quatro fases, primeiro é feito um projeto de arquitetura, depois projeto de componentes, projeto de estrutura de dados e por último o projeto do algoritmo. A etapa de testes é na qual o sistema é avaliado, verificando se ele esta de acordo com os requisitos estudados. Por último a etapa de manutenção consiste em adequar e manter o ambiente criado nas últimas etapas, reparando defeitos de *software*, adaptando-o para novos ambientes operacionais e adicionando funcionalidades. O trabalho especifica alguns pontos dentro das etapas comentadas acima, porém elas são relativas ao modelo adotado pelo autor e não são de uso geral. Finalizando, o autor conclui que para trabalhos futuros é possível atingir padrões como ISO9126 de qualidade, além de controle de custos através da verificação de diversas propostas de arquitetura.

4 Projeto/Proposta de Solução

Os conceitos revisados até este momento foram utilizados para a criação de uma proposta de solução computacional para a equipe baja Velociraptor. Revisando, a equipe tem como principal problema não conseguir obter informações que interessam a cada setor de projeto do veículo de maneira **intuitiva, simplificada, modular e com armazenamento**. Estes três conceitos são detalhados no Quadro 4.1 a seguir:

Quadro 4.1: Conceitos do cenário estudado

Conceito	Descrição
Intuitiva	Demostração dos dados com gráficos e valores absolutos
Simplificada	Um sistema para aquisição para todos os dados dos sensores reunindo as informações de todos os setores do veículo
Modular	Preparado para expansão e atualização com novas tecnologias
Armazenamento	Armazena os dados retirados dos sensores para análises futuras

Fonte: Autor.

4.1 Objetivos

Os objetivos que devem nortear o trabalho a fim de sanar os problemas levantados podem ser subdivididos em um objetivo geral, que é a meta final do trabalho, e objetivos específicos, que servem como diretrizes menores que quando concluídas resultaram na conclusão do trabalho como um todo.

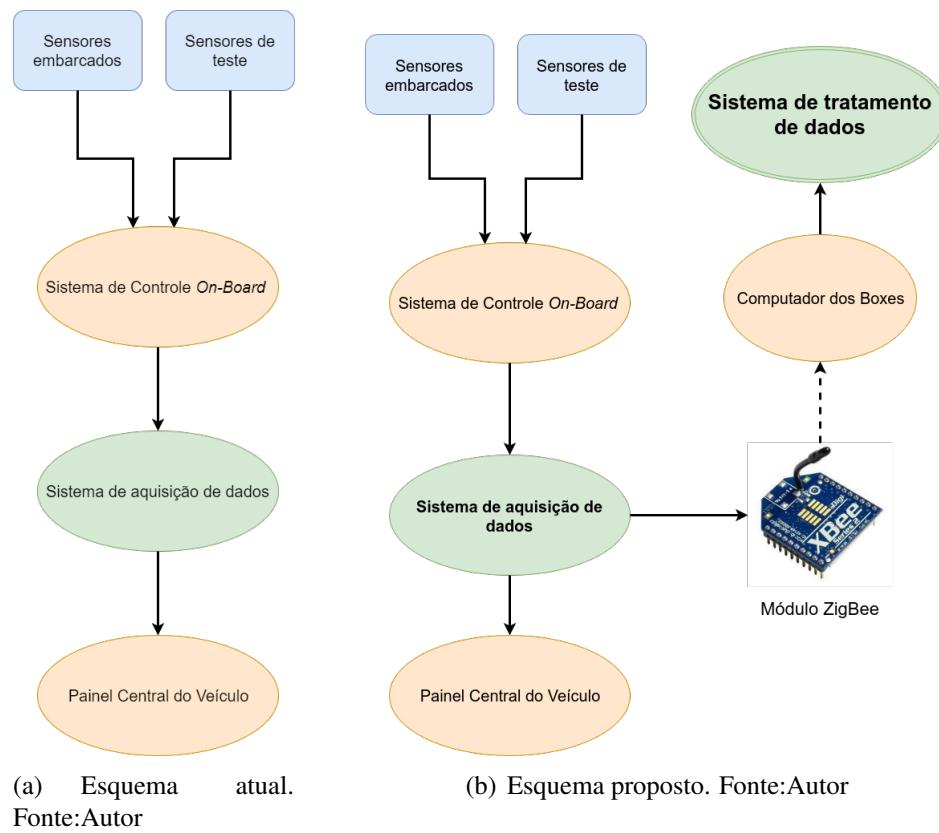
4.1.1 Objetivo Geral

Produzir um sistema que forneça a equipe informações que irão ajudar aos setores de projeto do veículo como suspensão dianteira, suspensão traseira, freio, transmissão e eletrô-

nica em testes de bancada, bem como fornecer dados de uso geral da equipe como média do consumo de gasolina em prova para análises posteriores.

O sistema é dividido em duas frentes: a parte de **aquisição** dos dados, feito junto ao SCOB com base nos dados recebidos via sensores revisados na seção 2.2, esta parte já existe atualmente no baja Velociraptor, porém ela foi atualizada para acompanhar os avanços do novo *software*; a parte de **tratamento** dos dados é o foco deste trabalho, nesta parte os dados serão transmitidos por um módulo de transmissão de dados sem fio ZigBee e lidos por um *software* que permite visualização de gráficos e mostra de valores absolutos de maior importância. A Figura 4.1 possui dois diagramas, o diagrama da Figura 4.1(a) traz um esquema geral que atualmente é utilizado pelo Velociraptor no seu projeto, já a Figura 4.1(b) traz um esquema geral proposto para este trabalho de conclusão de curso, desde a aquisição dos dados pelos SCOBs até o tratamento dos dados nos boxes com o *software*. Em ambas as figuras os balões em azul indicam os sensores, os balões em laranja indicam *hardware* de controle e balões em verde indicam *software*, os balões com texto em negrito indicam modificação/criação de um sistema.

Figura 4.1: Diagrama com o esquema geral do sistema atual e o esquema geral do sistema proposto.



4.1.2 Objetivos Específicos

- O sistema deve ser independente em relação ao meio em que os dados são transportados do SCOB para os boxes, desta forma em atualizações futuras o método atual de transferência de dados pode ser substituído;
- O *software* deve ser independente de sistema operacional;
- O *software* deve possuir informações específicas para cada área da engenharia automobilística;
- O *software* deve ser construído de forma a facilitar a manutenção evolutiva, preparado para chegada de novos sensores;
- Deve se instaurar uma cultura de utilização de plataformas de versionamento para facilitar manutenções adaptativas e manutenções corretivas.

Os objetivos específicos devem nortear a produção deste sistema. Atingindo as metas propostas se espera que o objetivo geral do projeto seja alcançado, trazendo essas informações do veículo para a equipe presente nos boxes.

4.1.3 Mudanças de escopo

O projeto teve uma mudança importante no escopo durante seu desenvolvimento e tais mudanças serão apresentadas aqui nesta seção.

Inicialmente, o projeto englobaria telemetria, a criação de um sistema de controle *on-board* e a criação de um *software* de tratamento dos dados provenientes do veículo. Esta ideia possuía um escopo muito amplo para ser feito por apenas uma pessoa em dois semestres, então uma nova proposta foi realizada na qual o foco seria no *software* de tratamento de dados recebendo estes mesmos dados via um *Secure Digital Card*, ou como é normalmente conhecido cartão SD. Este segundo escopo era mais realista em relação ao tempo de execução do projeto, levando em conta o trabalho para a criação do sistema levantando os dados necessários para uma melhor forma de demonstrar as grandezas do sistema. Contudo durante reuniões do projeto 2018 do sub-sistema de eletrônica veicular do Velociraptor, foi criado um terceiro escopo de projeto em conjunto com os interesses da equipe e suas prioridades. Neste escopo, que é o atual desenvolvido e testado neste trabalho, o sistema recebe as informações dos sensores via rede

sem fio utilizando um par de módulos ZigBee. Este escopo se assemelha muito ao primeiro, porém com o auxílio da equipe de eletrônica para fabricação das placas de teste, a ideia enfim se tornou viável.

5 Desenvolvimento da solução

Neste capítulo é discutido sobre o desenvolvimento do *software* de tratamento de dados proposto por este trabalho.

O programa feito para receber as informações provindas do veículo é implementado em sua totalidade na linguagem interpretada Python v3.6, esta foi escolhida devido a facilidades que a mesma proporciona no uso de bibliotecas de suporte ao sistema e de interfaceamento gráfico além de ser uma linguagem comum ao autor. Outras linguagens como o C++ foram consideradas para a aplicação, porém como o fluxo de dados não é feito em grande escala a velocidade de processamento não é uma questão primária. O sistema que atua no microcontrolador é programado em C e C++, pois este é o padrão de desenvolvimento para os microcontroladores utilizados.

5.1 Bibliotecas e Softwares utilizados

O sistema de tratamento de dados utiliza de diversos recursos disponíveis na linguagem Python sejam eles bibliotecas ou até mesmo *frameworks*, a seguir é discutido sobre tais recurso e como são utilizados no sistema.

Um dos principais componentes do *software* é a interface gráfica, ou interface do usuário. É por meio dela que a equipe presente nos boxes pode ter acesso as informações do veículo, observando os gráficos de tração, velocidade, extensometria, etc... Para a concepção da interface gráfica foi utilizada um *framework* chamada Qt¹. O Qt é nativamente utilizado para o C++ na construção de interfaces gráficas, para utilizar-lo no Python, é necessário a utilização de uma biblioteca auxiliar chamada PyQt². Tanto o Qt quanto o PyQt estão disponíveis em Windows, distribuições Linux e MacOS, um fator importante visto que a equipe utiliza o sistema operacional Windows, porém o autor desenvolveu o sistema utilizando a distribuição Manjaro XFCE e Ubuntu 16.04. Este é um tema recorrente neste trabalho, todas as bibliotecas utilizadas devem ser portáveis para ambos os sistemas operacionais Windows ou distribuições Linux como declarado nos objetivos específicos. As versões utilizadas das bibliotecas foram o Qt 5.5.1 e o

¹Site: <<https://www.qt.io/>>

²URL: <<https://riverbankcomputing.com/software/pyqt/intro>>

PyQt 5

Para o desenvolvimento foi utilizada a ferramenta Designer da *framework* Qt. A ferramenta permite a criação de telas com auxílio de interface gráfica e após a construção das mesmas telas, se utiliza um comando de linha (*pyuic5*) para transformar o código bruto da tela de XML para Python. Este novo código em Python é utilizado para se referenciar os objetos dentro da interface, necessário para união da parte lógica com a parte gráfica.

Uma característica específica deste sistema é a execução da parte gráfica em paralelo a recepção de dados. Isto implica que a parte gráfica deve ser executada em conjunto com o recebimento dos dados enviados pelo veículo, tendo isto em mente, é necessário utilizar de processamento paralelo para fazer ambas as tarefas. A solução encontrada para tal foi a biblioteca *multiprocessing* do Python. Esta biblioteca possui funções para criação de *pipes*, estrutura que retorna dois objetos representando um "tubo" de passagem de informação entre dois processos, ambos os processos tem acesso a comandos de envio e recepção de dados para conversa entre duas aplicações em tempo real. A função *pipe* opera de forma *duplex*, onde ambos os lados podem enviar e receber dados, porém devido ao escopo atual do projeto o sistema será utilizado como *simplex*.

Para a conexão do sistema de boxes com o microcontrolador é utilizado o protocolo de rede sem fio ZigBee. O protocolo é definido como uma solução para sistemas de baixa taxa de transmissão de dados e baixo consumo de energia, ele opera no alcance de uma PAN e tem uma taxa de transferência de até 250Kbps (ELAHI; GSCHWENDER, 2009). Estas características e outras informações revisadas na Seção 2.4 levaram a utilização deste protocolo para a comunicação sem fio entre os sistemas citados. Para realizar o processamento dos pacotes recebidos pelo módulo ZigBee no computador foi utilizada a biblioteca *python-xbee*. Esta biblioteca permite ler e escrever pacotes para os módulos que estão na rede PAN sem necessidade de utilizar comandos diretos para as entradas da porta USB. Para utilizar a biblioteca é configurado o *baud rate* e qual entrada está localizado o modulo ZigBee.

Um *software* auxiliar foi utilizado para a configuração dos módulos ZigBee. O XCTU³ faz a configuração inicial entre os módulos a fim de ordenar quem terá qual papel na rede. Além de configurar quem será o coordenador e quem será o roteador/*end-point*, o XCTU também configura qual será o ID da PAN, em qual modo o módulo irá operar seja ele API ou AT, permite envio de pacotes de teste entre os envolvidos e também de sequencias de pacotes.

³URL: <<https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>>

5.2 Hardware utilizado

Para o desenvolvimento do sistema foram utilizados alguns equipamentos de *hardware*. Devida a proposta de solução foi necessária a utilização de alguns equipamentos específicos para o desenvolvimento do sistema , estes equipamentos utilizados no sistema são listados a seguir:

- Xbee ZigBee PRO series 2;
- Xbee ZigBee series 2;
- XBee Explorer USB Adapter;
- Arduino Nano V3;
- Sensores automotivos;
- *Shield* para acoplar as placas;e
- Computador para receber os dados.

Os módulos Xbee ZigBee series 2/PRO series 2 (DIGI, 2018) foram adicionados no decorrer do projeto. A telemetria foi uma das aspirações iniciais desse trabalho de conclusão de curso, porém devido ao tamanho do escopo do projeto a execução desta parte do sistema foi retirada. Contudo durante as reuniões feitas em conjunto com o sub-sistema de eletrônica veicular do Baja Velociraptor, foi decidido que a telemetria entraria no escopo anual do projeto 2018 do veículo. Com a entrada deste projeto, alguns dos membros se comprometeram a ajudar na construção desse sistema e também na compra dos módulos para uso do protocolo ZigBee. Tendo todo este cenário, foi decidido entre o autor e orientador pela opção de produção deste sistema pronto para receber dados via telemetria, no qual o sistema que adquire os dados é executado ao mesmo tempo que o sistema que trata esses mesmos dados.

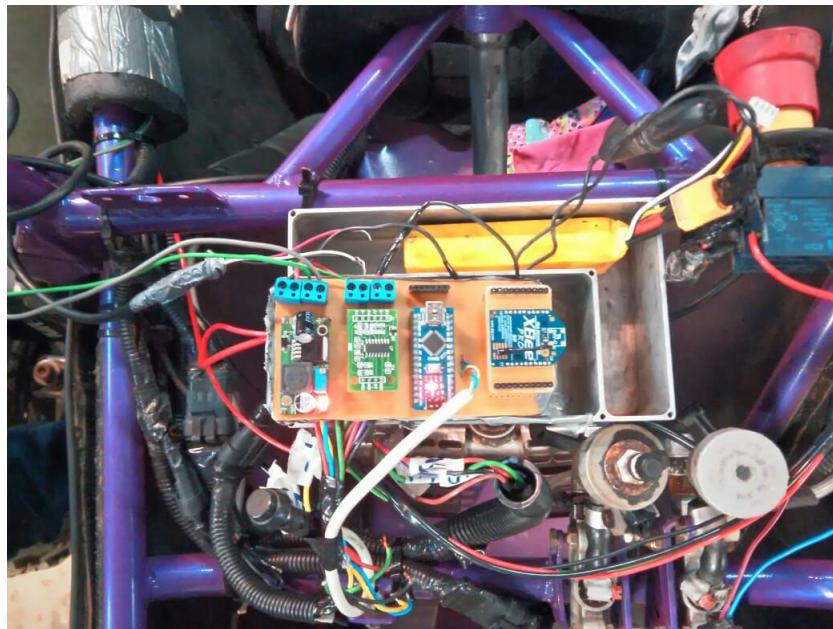
Em relação ao módulo utilizado, ele possui as características base do protocolo ZigBee porém a versão comprada (PRO) possui algumas característica melhoradas. Por exemplo: O alcance urbano do Xbee ZigBee series 2 é de 40 metros, já da versão PRO é de 90 metros, o alcance em área rural/aberta, com pouca interferência de sinal, é de 120 metros na versão padrão e 3.200m na versão PRO (DIGI, 2018). Para a realizao dos testes foram utilizados um modulo Xbee ZigBee series 2 e um Xbee ZigBee PRO series 2, ambos são compatíveis sem necessidade de uso de bibliotecas auxiliares ou modificação de *firmware*.

O Xbee Explorer USB Adapter¹ é um adaptador USB para utilizar o módulo ZigBee ligado direto aos computadores para configurar. Ele permite a utilização do *software* XCTU de forma eficiente, sem a necessidade de conectar o módulo ZigBee a um outro módulo FTDI², que faz conexão com o computador.

O Arduino Nano V3 foi utilizado por ser mais prático de realizar os testes, as configurações desta placa de desenvolvimento são as mesmas apresentadas na Seção 2.3.1. Nenhuma biblioteca exclusiva das plataformas Arduino foi utilizada para enviar os dados do microcontrolador para o computador. O módulo Xbee ZigBee recebe os dados tratados e envia para a outra ponta de forma serial pelos pinos *Rx* e *Tx*, a outra ponta dos dados deve se receber os mesmos e tratar-los de acordo com o protocolo.

O computador utilizado nos testes possui um processador Intel I5 7200U e 8GB de memória RAM. Estes são os atributos de maior relevância para a execução do sistema já que os dados são recebidos diretamente dos módulos ZigBee e apenas gravados em HD no formato de gráficos. Na Figura 5.1 é demonstrado o sistema embarcado no veículo, pronto para transmitir os dados dos testes realizados no Capítulo 6.

Figura 5.1: Sistema de controle *on-board* montado no veículo.



Fonte: Autor.

¹URL: <<https://www.filipeflop.com/produto/xbee-explorer-usb-adapter/>>

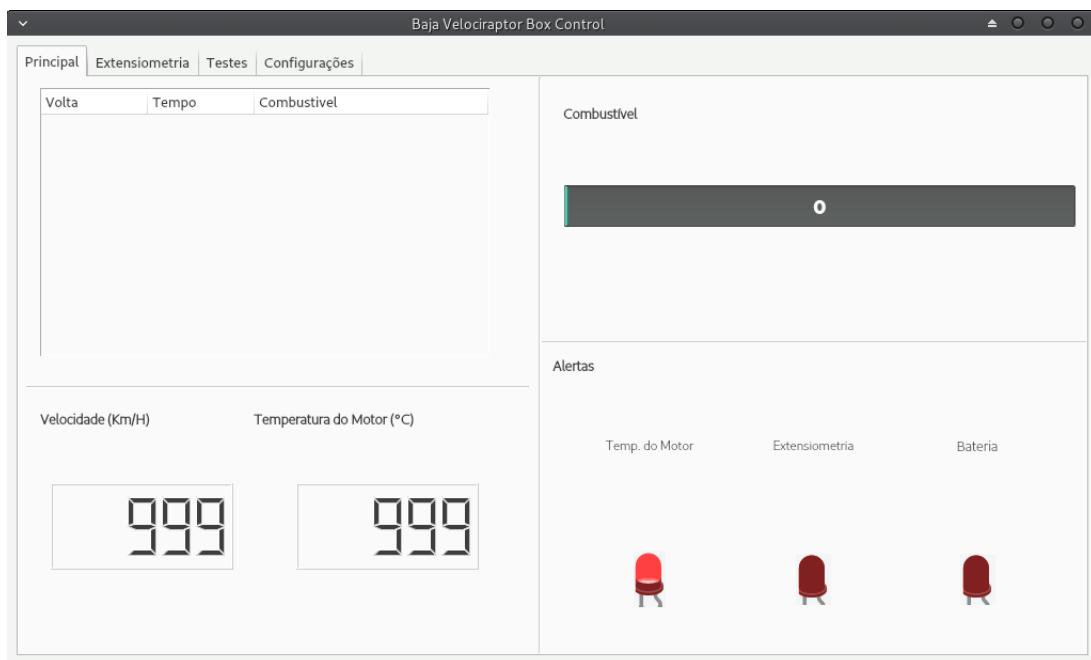
²URL: <<http://www.ftdichip.com/>>

5.3 Software desenvolvido

Nesta seção será apresentado o *software* que foi criado para tratamento dos dados provindos do veículo. Na seção será demonstrada tela a tela quais são as funcionalidades do sistema.

Inicialmente temos a tela principal na Figura 5.2. Nela é possível ir para as outras telas de funcionalidades do sistema na barra superior, além de ter as funcionalidades padrões para provas de enduro, uma das modalidades dos campeonatos de baja SAE. Na região superior direita é possível visualizar o contador do nível do combustível, ele inicia em 8 marcações quando o tanque está cheio e progride até 0 quando o tanque de combustível está vazio. No canto inferior direito estão presentes os alertas importantes para a equipe durante a prova, entre os alertas existem o de temperatura do motor que é ligado quando o óleo do mesmo passa de 125°C, o de extensiometria que é acionado quando algum ponto do chassis sofre um esforço de tensão maior do que um valor X e o da bateria que é acionado quando a corrente da bateria está baixa. No canto inferior esquerdo está disposto os dados de velocidade do veículo em quilômetros por hora e a temperatura do motor na ultima coleta de dados. Por último no canto superior esquerdo é disposta a tabela de dados de volta, onde a cada volta são mostrados os dados de número da volta, tempo de volta e combustível restante após a mesma.

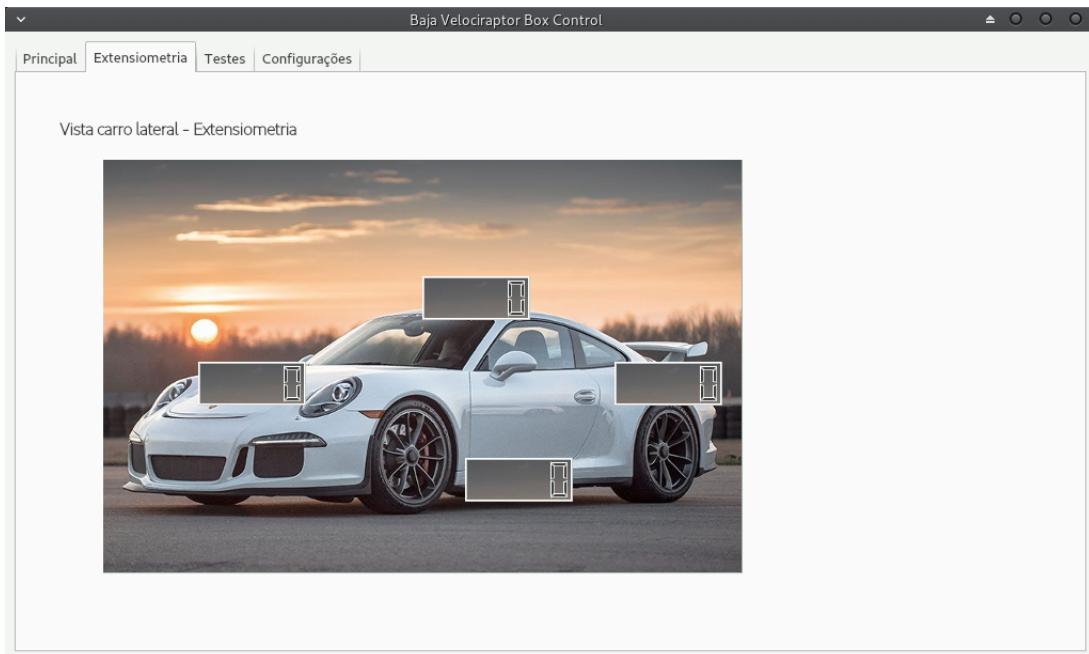
Figura 5.2: Tela principal do programa.



Fonte: Autor.

A segunda tela possui as informações da extensometria. Estes dados são provindos de quatro sensores espalhados no carro que dão informação sobre a torção exercida sobre o chassis em aceleração, saltos e curvas. A Figura 5.3 demonstra essa tela.

Figura 5.3: Tela de extensometria.

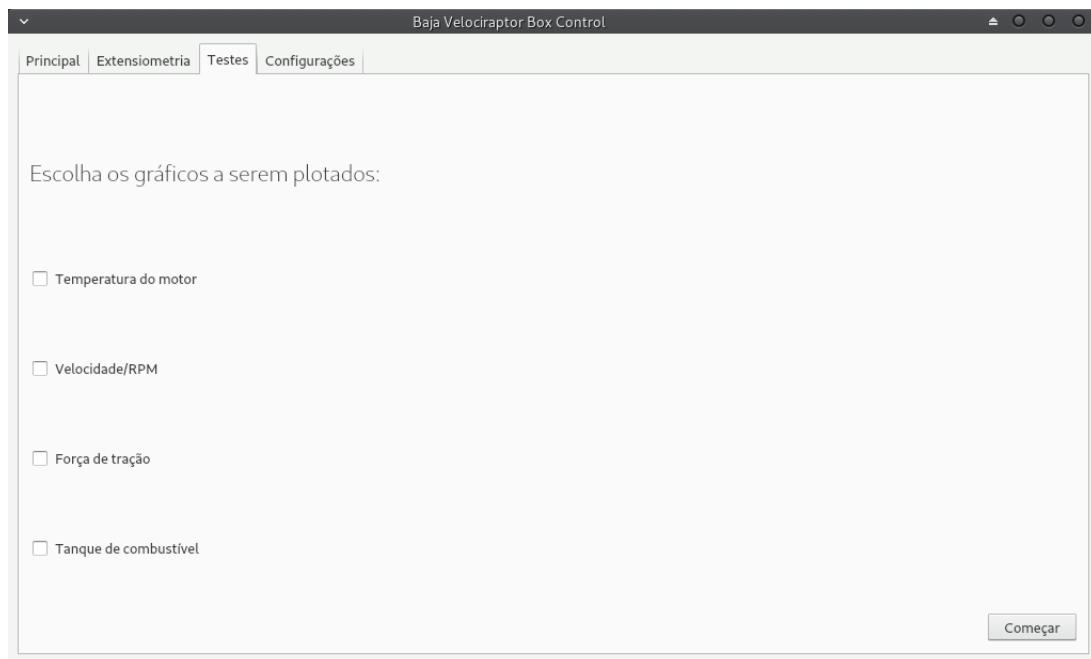


Fonte: Autor.

A tela da Figura 5.4 possui as opções de gráficos que se deseja visualizar. O principal objetivo é gerar gráficos para o monitoramento do veículo em testes ou em prova. O gráfico de força de tração tem como foco os testes do veículo, já os de velocidade/RPM, temperatura do motor e de nível do combustível são utilizados em tempo de prova. Mais de um gráfico pode ser aberto ao mesmo tempo, acrescentando modularidade a disposição das telas, a Figura 5.5 exibe um exemplo de como o gráfico é gerado, os dados são provenientes do veículo.

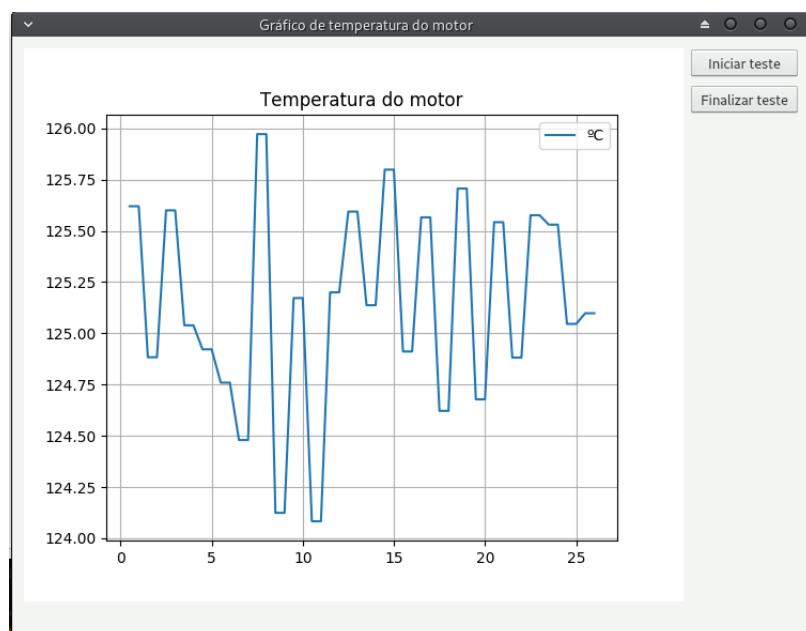
Por ultimo a Figura 5.6 exibe as configurações do sistema. Atualmente a única opção é desligar o sistema, fechando as portas, *pipes* e processos abertos. Estas são as telas presentes nos programas, elas foram construídas em conjunto com a equipe Velociraptor para atender as demandas geradas pelos sensores presentes nos veículos.

Figura 5.4: Tela de seleção de gráficos de teste e monitoramento.



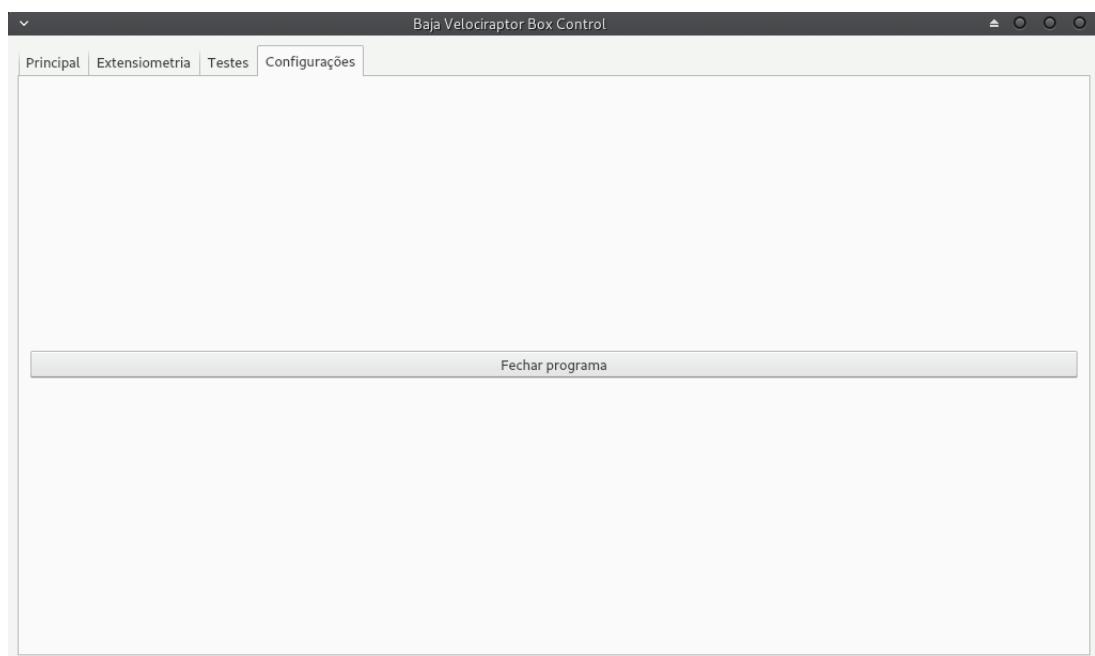
Fonte: Autor.

Figura 5.5: Tela com exemplo de gráfico de temperatura do motor por tempo em segundos.



Fonte: Autor.

Figura 5.6: Tela de configurações.



Fonte: Autor.

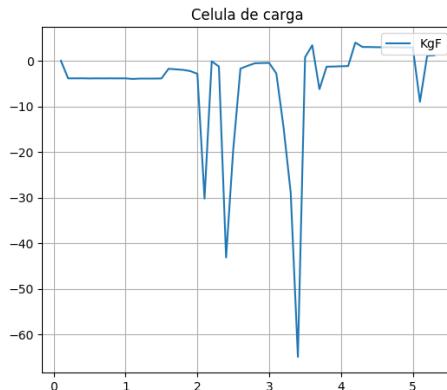
6 Testes e análise de resultados

Nesta seção serão comentados sobre os testes realizados sobre o *software*, em seguida será comentado um pouco dos resultados obtidos. Os testes do sistema foram feitos durante testes do veículo do sub-sistema de suspensão monitorando dois elementos do veículo, uma célula de carga foi posicionada no *link* superior esquerdo da suspensão traseira para retirar dados sobre a ação de forças sobre o mesmo e um sensor de temperatura foi acoplado ao cárter do motor para retirada da temperatura do motor. Os outros sensores do veículo não foram abrangidos nos testes devido a não estarem prontos no período que os testes foram realizados. Alguns sensores são alterados entre o período de competição e precisam ser reimplementados no sistema.

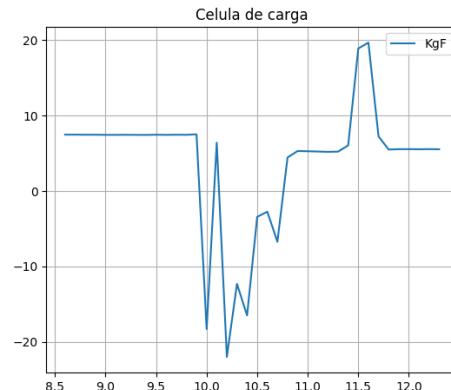
As Figuras 6.1, 6.2 e 6.3 exibem gráficos gerados pelo sistema em tempo de execução dos testes do veículo. É importante salientar que os valores dos gráficos recebidos pelos sensores não são importantes para este trabalho, o que é importante é o fato de que os valores estão condizentes com o que os sensores apresentam. Os dados exibidos são importantes para o projeto do sub-sistema de suspensão no caso do teste com a célula de carga e importantes para o sub-sistema do motor no caso da temperatura do motor. As conclusões dos testes são tiradas sobre os gráficos demonstrados a seguir. Levando em conta que esta é a funcionalidade do sistema que melhor exemplifica seu funcionamento sem uma demonstração ao vivo do programa, as outras funcionalidades de demonstração dos dados em valores absolutos como a temperatura do motor demonstrada na Figura 5.2 foram verificadas e funcionam em conjunto com os gráficos gerados. Consequentemente é possível validar assim a demonstração dos dados em valores absolutos sem necessidade de impressão de um *log* do programa.

Os gráficos exibidos nas Figuras 6.1 e 6.2 indicam a força exercida sobre o *link* superior esquerdo da suspensão traseira em quilograma-força no eixo vertical e o tempo em segundos no eixo horizontal. Os valores flutuam para números negativos quando o *link* é comprimido e flutuam para números positivos quando o *link* é tracionado. Os valores do eixo horizontal dados em segundos não estão corretos, isto se deve a alguns fatores explicados na Sessão 6.1. Já os gráficos exibidos na Figura 6.3 indicam a temperatura do óleo do motor em graus Celsius no eixo vertical e o tempo em segundo no eixo horizontal. A validação dos dados é explicada na seção 6.2.

Figura 6.1: Gráficos gerados a partir de dados da célula de carga com o veículo parado.

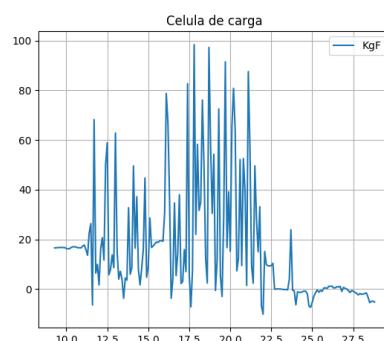


(a) Veículo parado no chão. Fonte: Autor

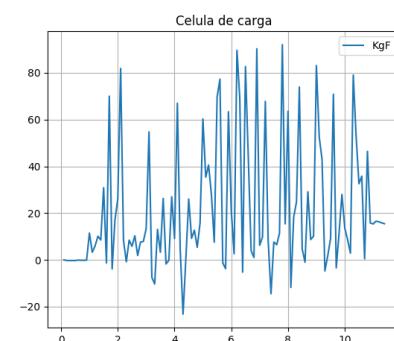


(b) Veículo parado com suspensão traseira suspensa. Fonte: Autor.

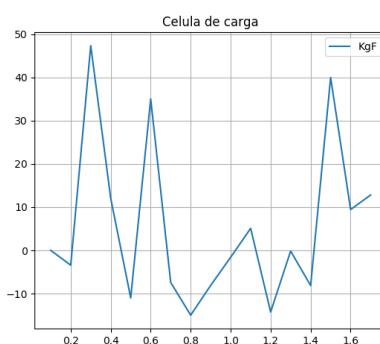
Figura 6.2: Gráficos gerados a partir de dados da célula de carga com o veículo andando.



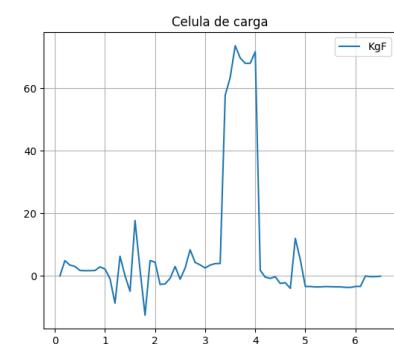
(a) Veículo percorrendo a pista de testes no sentido horário. Fonte: Autor



(b) Veículo percorrendo a pista de testes no sentido anti-horário. Fonte: Autor.

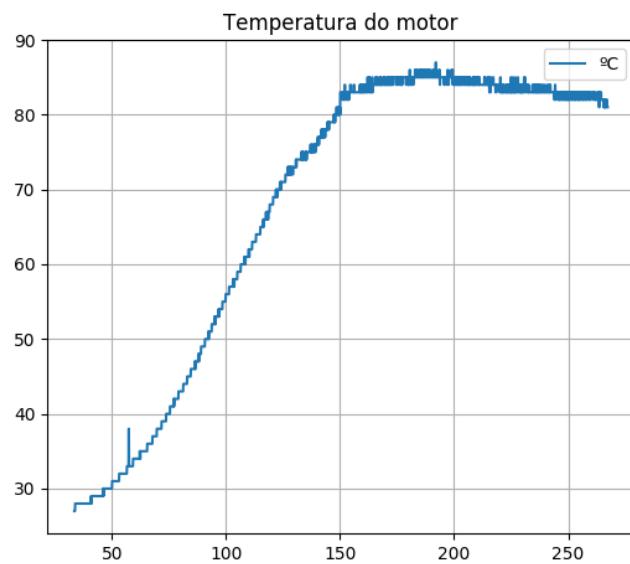


(c) Veículo percorrendo a pista em ritmo de corrida com saltos e tombo. Fonte: Autor.

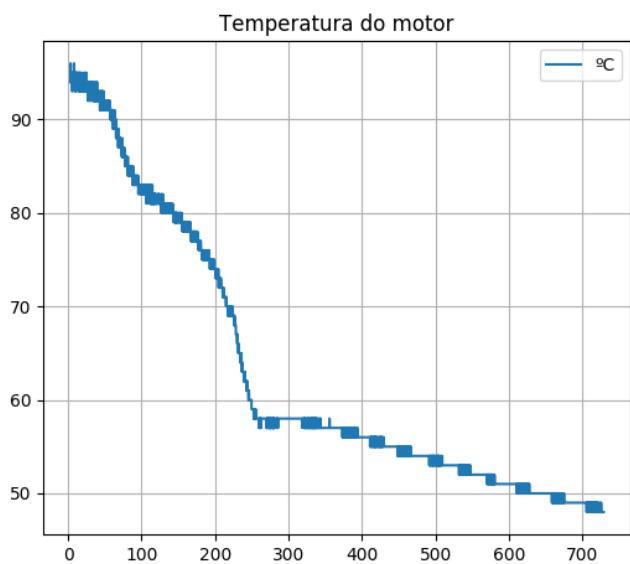


(d) Veículo andado em círculos. Fonte: Autor.

Figura 6.3: Gráficos gerados a partir de dados do sensor de temperatura.



(a) Temperatura do óleo do motor no início dos testes. Fonte: Autor



(b) Temperatura do óleo do motor do fim dos testes até entrada na oficina. Fonte: Autor.

6.1 Taxa de atualização

A taxa de atualização dos dados no sistema é definida pelo tempo em que os dados são atualizados do veículo para o computador. O sistema envia os dados de todos os sensores no mesmo pacote e isto se deve a limitação de recursos do projeto, pois a equipe Velociraptor possui apenas um par de módulos XBee ZigBee S2/PRO S2. Devido a esta característica o sistema possui uma taxa de atualização do sistema de tratamento de dados igual a taxa de atualização dos dados do sensor com maior demora para atualizar. Alguns sensores possuem uma taxa de atualização dos dados mais rápida que a estipulada para o programa originalmente, nos testes realizados foi notado que o sensor de temperatura pode ser atualizado em até 100 milissegundos (ms) sem perda de informação, o que deixa uma margem para ajustar a taxa de atualização do mesmo, contudo também é notado que o sensor de célula de carga demora aproximadamente 2.000 ms para fazer os cálculos necessários para apresentar os dados na resolução escolhida.

Com estas informações é possível concluir que a taxa de atualização do sistema de tratamento de dados é igual a taxa de atualização do sensor de célula de carga. Com isto a taxa de atualização seria de uma atualização a cada 2.000 ms, um valor que não é satisfatório segundo a equipe dos boxes do Velociraptor. Porém o sensor de célula de carga é utilizado apenas para testes e não deve entrar no veículo para provas de enduro ou provas de testes das competições do baja SAE. Portanto os valores alcançados com o outro sensor testado de 100 ms são suficientes para a equipe. Testes futuros com os outros sensores do veículo devem manter ou negar esta taxa de atualização.

Outro fato que deve ser explicado é os valores em segundo dos gráficos demonstrados nas Figuras 6.1, 6.2 e 6.3. Devido a falta de conhecimento da real taxa de atualização dos dados do sensor da célula de carga, os valores aferidos para o sistema eram de 500 ms, no qual os segundos apresentados nos gráficos são escritos em 500 ms porém não são atualizados em 500 ms, mas sim em 2.000 ms. Este problema no código foi percebido durante a realização dos testes e não foi possível realizar novos testes com o código atualizado devido ao calendário de testes da equipe.

6.2 Validação dos dados

Para realizar a validação dos dados apresentados nos gráficos, foram feitos testes em bancada com o sensor de temperatura do motor. Os testes foram realizados com o microcontrolador conectado a um computador, no qual os dados obtidos na porta analógica e após tratamento no microcontrolador eram mostrados na tela do computador conectado ao microcontrolador e também no sistema de tratamento de dados, que recebia os dados via ZigBee.

Foram lidos alguns pacotes de dados do sensor em ambos os receptores e assim foi constatado que os dados não continham divergências, a mesma informação era encontrada em ambos. Os valores dos dados também foram comparados a distância, para verificar se o sistema quando fora do alcance entregava informações falsas. Foi constatado com o teste que as informações não são alteradas mesmo com a distância, elas apenas chegam com atraso ou dão *timeout* como pode ser visto na Seção 6.3.

6.3 Distância de funcionamento dos módulos ZigBee

Um teste para medir a distância de funcionamento dos módulos foi realizado com o auxílio da ferramenta Google Maps. Este teste tinha um intuito de não apenas verificar as distâncias de funcionamento mas também de visualizar o comportamento do sistema quando utilizado no limite de seu alcance de operação.

A distância anunciada (DIGI, 2018) *indoor* ou em área urbana para os módulos Xbee Zigbee series 2 é de 40 metros e *outdoor* ou em área rural é de 120 metros. Os testes foram feitos em uma área urbana aberta em linha reta, e foram notados três tipos de comportamento nos módulos. Até 67 metros os dados eram enviados sem atraso e com consistência, até 94 metros os dados ainda eram enviados para o sistema de tratamento de dados porém alguns quadros chegavam com atraso de algumas centenas de milissegundos e a partir de 94 metros os dados não chegavam no sistema com consistência, levando a atrasos de diversos segundos. Os resultados foram surpreendentes devido a consistência de entrega ser ainda maior que a declarada pelo fabricante, no momento dos testes os módulos estavam operando com tensão de 3.3V a recomendada pelo fabricante e testada com um multímetro.

6.4 Consumo de bateria

Também foi feito um teste para descobrir o consumo de corrente da bateria adicionado no sistema. Este teste consistiu em ligar o SCOB sem o Xbee com a leitura do sensor de temperatura do motor e depois ligar o SCOB nas mesmas condições, porém com o Xbee. Ambos os sistemas estavam com o mesmo código no microcontrolador, rodando nas mesmas condições.

Então foi medido o uso de corrente em miliampere dos dois casos com um multímetro. No primeiro caso todo o SCOB consumia 22 mA e no segundo o valor saltou para 48 mA. Estes dados são coerentes olhando para os valores de consumo de corrente dados pelo fabricante no manual (DIGI, 2018) e levando em conta que nem todas as funções do módulo estavam sendo utilizadas. A adição do Xbee foi a única mudança feita especificamente para o sistema de aquisição e tratamento de dados, além disto, esta mesma mudança já estava no escopo de projeto da equipe Velociraptor, o que indica que o aumento de consumo de corrente elétrica não deve causar grande impacto no sistema final.

7 Conclusão e Trabalhos Futuros

O desenvolvimento de um *software* de tratamento de dados para o veículo da categoria baja SAE se mostrou necessário devido ao nível de complexidade e quantidade de sensores que fazem e devem fazer parte do projeto da equipe Velociraptor. Apenas a utilização de arquivos de texto para gerenciamento das informações pode deixar a tarefa confusa, árdua e dificulta o armazenamento das informações de forma comprehensiva e centralizada. O sistema produzido alcança o objetivo geral estipulado, abrangendo todos os sensores contidos no veículo e alguns que ainda se encontram em desenvolvimento já estão implementados no sistema de tratamento. Ele também auxilia na manutenção indicando possíveis falhas para a equipe caso alguma grandeza saia dos padrões estabelecidos durante testes ou provas. Além da parte de *software* possuir essas características, o sistema foi utilizado em um teste pratico e com resultados reais, onde a equipe pode verificar o funcionamento de algumas grandezas importantes do sistema, os dados da suspensão traseira e temperatura de operação do motor. Tendo isto dito, foi possível verificar que as características propostas no início do Capítulo 4 de demonstração dos dados de forma intuitiva, simplificação de complexidade com a junção dos dados em um único ambiente, modularidade para adição de novas tecnologias e armazenamento dos dados tratados foram atingidas devido aos testes realizados e também ao *feedback* recebido pelos membros da equipe Velociraptor.

Em relação a trabalhos futuros o sistema deve passar por uma etapa de refinamento das funções e de pesquisa de novas funcionalidades. Na etapa de refinamento, é importante destacar que a taxa de atualização deve ser mantida em uma atualização a cada 500 ms quando adicionados novos sensores ao veículo, considerando sensores de uso em competição. Os sensores que não estavam prontos para os testes finais devem ser adicionados ao veículo para realização de um teste completo do sistema antes do inicio do Baja Sul 2018. Os sensores de extensometria serão implementados também ao sistema, logo que a parte de *hardware* seja projetada e implementada pela equipe de eletrônica embarcada. Por último, novos requisitos de testes podem surgir e a criação de novos parâmetros para apresentação de gráficos podem se tornar necessários, um exemplo disto é a criação de uma janela específica para a realização de testes de aceleração do veículo.

Em relação a etapa de pesquisa para novas funcionalidade que podem ser pesquisadas

das e implementadas em trabalhos futuros, deve se tomar como diretriz o que está acontecendo no mundo automotivo atualmente. Um grande salto de complexidade para o sistema e de entrega de valor seria a adição de simulação aerodinâmica, no qual seria possível saber o comportamento do chassis em alta velocidade, melhorando desempenho em tempo de volta e também consumo de combustível (WANG, 2018). Outra possibilidade seria uma implementação de um sistema de conselhos sobre o veículo, utilizando conceitos de inteligência artificial para obter melhores valores em ajustes do veículo. Este sistema viria junto com a implementação de atuadores, onde os mesmos poderiam fazer alterações de parâmetros de peças como por exemplo a relação do câmbio em tempo de prova para melhor tração das rodas.

Este trabalho foi de grande importância para o autor não apenas pelo conhecimento técnico adquirido. A participação no desenvolvimento de um projeto como o Baja SAE trouxe a necessidade de conciliar o que é importante para o projeto e o que os integrantes da equipe desejam fazer, criar um escopo de projeto com duração de um ano e a realização de um trabalho em uma equipe multidisciplinar com várias áreas da engenharia ajudou a expandir o entendimento sobre o que é trabalhar em colaboração de um objetivo maior e mais que isso, são conhecimentos que serão levados muito além deste trabalho.

Referências

- ANDRIA, G. et al. Design and implementation of automotive data acquisition platform. In: *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. 2015. p. 272–277. ISSN 1091-5281.
- ANDRIA, G. et al. Development of an automotive data acquisition platform for analysis of driving behavior. *Measurement*, v. 93, n. Supplement C, p. 278 – 287, 2016. ISSN 0263-2241. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0263224116303943>>.
- ARAÚJO, T. V. et al. Projeto mini baja como estudo de para instrumentação eletrônica. In: *XXXIV Congresso Brasileiro de Ensino de Engenharia*. 2006. p. 1776–1790.
- ATMEL. *ATmega328/P Datasheet Complete*. 2016. Disponível em: <http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>. Acesso em: nov. 2017.
- BOSCH. *Iridium Spark Plugs*. 2017. Disponível em: <<https://www.boschautoparts.com/en/auto/spark-plugs/iridium-spark-plugs>>. Acesso em: out. 2017.
- CALDERÓN, A. G.; RUIZ, G. G.; BOHÓRQUEZ, A. C. G. Gprs telemetry system for high-efficiency electric competition vehicles. In: *2013 World Electric Vehicle Symposium and Exhibition (EVS27)*. 2013. p. 1–7.
- CHANDIRAMANI, J. R.; BHANDARI, S.; HARIPRASAD, S. A. Vehicle data acquisition and telemetry. In: *2014 Fifth International Conference on Signal and Image Processing*. 2014. p. 187–191.
- DIAS, J. E. A. *Eletrônica, Instrumentação e Telemetria do Veículo UFVBAJA*. 50 p. Monografia (Trabalho de Conclusão de Curso) — Engenharia Elétrica na Universidade Federal de Viçosa, Viçosa, 2010.
- DIGI. *DIGI Xbee ZigBee specifications*. 2018. Disponível em: <<https://www.digi.com/products/xbee-rf-solutions/2-4-ghz-modules/xbee-zigbee#specifications>>. Acesso em: mai. 2018.
- DIGI. *XBee RF Modules*. 2018. Disponível em: <<https://www.digi.com/resources/documentation/digidocs/pdfs/90000976.pdf>>. Acesso em: jun. 2018.
- ELAHI, A.; GSCHWENDER, A. *ZigBee Wireless Sensor and Control Network*. Pearson Education, 2009. (Prentice Hall Communications Engineering and Emerging Technologies Series from Ted Rappaport). ISBN 9780137059409. Disponível em: <<https://books.google.com.br/books?id=481PeRPyiEoC>>.
- ERDMANCZYK, S. *XBee API Mode Tutorial Using Python and Arduino*. 2014. Disponível em: <<https://serdmanczyk.github.io/XBeeAPI-PythonArduino-Tutorial/>>. Acesso em: mai. 2018.

- FAE. *Temperature Sensor - Breakdown*. 2017. Disponível em: <<http://www.fae.es/en/products/temperature-sensor>>. Acesso em: nov. 2017.
- FRADEN, J. *Handbook of modern sensors: Physics, designs, and applications*. 2016. 1–758 p. ISSN 1098-6596. ISBN 9783319193038.
- GISLASON, D. *ZigBee wireless networking*. : Newnes, 2008.
- HISTORY. *Último dia de produção do modelo T da Ford*. 2017. Disponível em: <<https://seuhistory.com/hoje-na-historia/ultimo-dia-de-producao-do-modelo-t-da-ford>>. Acesso em: nov. 2017.
- KATZOURAKIS, D. I. et al. Race-car instrumentation for driving behavior studies. *IEEE Transactions on Instrumentation and Measurement*, v. 61, n. 2, p. 462–474, Feb 2012. ISSN 0018-9456.
- KILIAN, C. *Modern Control Technology: Components and Systems*. Delmar Thomson Learning, 2001. (Student Material TV Series). ISBN 9780766823587. Disponível em: <<https://books.google.com.br/books?id=h35SAAAAMAAJ>>.
- MAZIDI, M.; MCKINLAY, R.; CAUSEY, D. *PIC Microcontroller and Embedded Systems: Using Assembly and C for PIC18*. Pearson Prentice Hall, 2008. ISBN 9780136009023. Disponível em: <<https://books.google.com.br/books?id=TVPMGQAACAAJ>>.
- MELEXIS. *MLX90614 family*. 2017. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/MLX90614_rev001.pdf>. Acesso em: nov. 2017.
- MTE-THOMSON. *Catálogo Eletrônico - 73020*. 2017. Disponível em: <<https://cate.mte-thomson.com.br/produto/detalhes/73020>>. Acesso em: nov. 2017.
- NAUNHEIMER, H. et al. *Automotive transmissions: Fundamentals, selection, design and application*. 2011. ISSN 14337851. ISBN 9783642162138.
- NISSAN. *New Nissan Pathfinder Performance*. 2017. Disponível em: <<https://www.nissan.co.kr/vehicles/new/pathfinder/performance.html>>. Acesso em: nov. 2017.
- NUNES, T. F. *Telemetria de um veículo Baja SAE através de rede CAN*. 49 p. Monografia (Trabalho de Conclusão de Curso) — Engenharia Mecatrônica na Universidade Federal do Rio Grande do Norte, Natal, 2016.
- PEREIRA, R. M. R. *Processos de Software para Implementações de Especificações Relacionadas ao Projeto Fórmula SAE/EESC/USP*. 87 p. Monografia (Trabalho de Conclusão de Curso) — Engenharia Elétrica na Universidade de São Paulo, São Carlos, 2012.
- RANI, B. S. et al. A survey to select microcontroller for sathyabama satellite's on board computer subsystem. In: *Recent Advances in Space Technology Services and Climate Change 2010 (RSTS CC-2010)*. 2010. p. 134–137.
- SAE. *Estudantes de engenharia constroem 30 carros off-road para competição*. 2016. Disponível em: <<http://portal.saebrasil.org.br/noticia/noticia-sae-brasil/Post/4151/Estudantes-de-engenharia-constroem-30-carros-off-road-para-compet%C3%A7%C3%A3o>>. Acesso em: ago. 2017.

SAE. *Regulamento*. 2017. Disponível em:
<<http://portal.saebrasil.org.br/programas-estudantis/baja-sae-brasil/regras>>. Acesso em: ago. 2017.

SHARP. *GP2Y0A21YK0F Datasheet*. 2017. Disponível em:
<http://www.mouser.com/ds/2/321/gp2y0a21yk_e-3493.pdf>. Acesso em: nov. 2017.

TAHA, Z. et al. Application of data acquisition and telemetry system into a solar vehicle. In: *2010 Second International Conference on Computer Engineering and Applications*. 2010. v. 1, p. 96–100.

WANG, J. et al. Aerodynamic research of a racing car based on wind tunnel test and computational fluid dynamics. In: EDP SCIENCES. *MATEC Web of Conferences*. 2018. v. 153, p. 04011.