

---

*Desenvolvimento de um sistema de aquisição e tratamento de  
dados para veículo baja SAE*

William Pereira

---

Joinville

2017

William Pereira

*Desenvolvimento de um sistema de aquisição e tratamento de  
dados para veículo baja SAE*

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

**Orientador: Roberto Silvio Ubertino Rosso Jr.**

**Doutor**

Joinville

2017

William Pereira

*Desenvolvimento de um sistema de aquisição e tratamento de  
dados para veículo bája SAE*

Relatório 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

## Resumo

### **PLACEHOLDER**

Eu gosto de carros. Desde criança minha paixão era correr nos circuitos virtuais de Gran Turismo, nas pistas noturnas de Need For Speed Underground e nas maluquices de Top Gear 3000. Quando menor, sabia que para ser piloto era necessário muito dinheiro e minha família não possuía tais recursos, então eu sonhava em ser um mecânico de carros, para poder mexer com os carros mais rápidos e os tornar-lós ainda mais rápidos, fazer com que a corrida não acabasse.**FAZER UM RESUMO DE VERDADE** Palavras-chave: *Performance, Sensoriamento, Telemetria.*

# Abstract

## PLACEHOLDER

I like cars. Since I was a kid my passion was to run in the virtual race tracks of Gran Turismo, in the night circuits of Need For Speed Underground and in the craziness of Top Gear 3000. When little, I knew it was needed a lot of money to be a motorsport race pilot and my family didn't had the resourcer to, so I dreamed to be a car mechanic, so I would be able to customize the fastest cars and make them go even faster, making sure the race never stop. **MAKE A REAL ABSTRACT**

**Keywords:** *Performance, Data Acquisition, Telemetry*

# Conteúdo

<b>Lista de Abreviaturas</b>	<b>7</b>
<b>1 Introdução</b>	<b>8</b>
1.1 Estrutura do trabalho . . . . .	10
<b>2 Fundamentação Teórica</b>	<b>11</b>
2.1 Baja SAE . . . . .	11
2.2 Sensores . . . . .	12
2.2.1 Temperatura do motor . . . . .	13
2.2.2 Frequência de rotação do motor . . . . .	14
2.2.3 Velocidade do veículo . . . . .	15
2.2.4 Nível do combustível . . . . .	16
2.2.5 Temperatura do câmbio CVT . . . . .	17
2.2.6 Deslocamento do amortecedor . . . . .	18
2.3 Microcontroladores . . . . .	19
2.3.1 Placas de desenvolvimento . . . . .	22
2.4 Software . . . . .	24
2.4.1 Aquisição de dados . . . . .	24
2.4.2 Tratamento de dados . . . . .	26
2.4.3 Engenharia de software . . . . .	26
<b>3 Trabalhos Relacionados</b>	<b>30</b>
<b>4 Projeto/Proposta de Solução</b>	<b>39</b>
4.1 Objetivos . . . . .	39

4.1.1	Objetivo Geral . . . . .	39
4.1.2	Objetivos Específicos . . . . .	40
<b>5</b>	<b><i>Pré Ensaio</i></b>	<b>42</b>
5.1	Pré Ensaio . . . . .	42
<b>6</b>	<b><i>Considerações Finais</i></b>	<b>43</b>
6.1	Fortemente considerável . . . . .	43
	<b>Bibliografia</b>	<b>44</b>

# **Lista de Figuras**

2.1	Diagrama do sistema elétrico usados atualmente no baja. . . . .	13
2.2	Sensor termistor NTC e um exemplo do sensor específico para automóveis. . . . .	14
2.3	Vela automotiva. . . . .	15
2.4	Sensor de velocidade MTE 73020. . . . .	16
2.5	Sensor de velocidade conectado próximo ao freio traseiro. . . . .	16
2.6	<i>Shield</i> criado para sensores de efeito Hall US1881. . . . .	17
2.7	Câmbio CVT. . . . .	18
2.8	Sensor MLX90614 e seu funcionamento. . . . .	19
2.9	Sensor de deslocamento do amortecedor SHARP GP2Y0A21YK0F. . . . .	19
2.10	Diagrama de blocos do microcontrolador ATmega328/ATmega328p. . . . .	20
2.11	Foto do painel dianteiro atual do baja Velociraptor. . . . .	23
2.12	Diagrama exemplificando funcionamento do sistema de aquisição. . . . .	25
2.13	Tabela com atributos desejáveis em um <i>software</i> . . . . .	27
2.14	Desenho exemplificando uma aplicação genérica do modelo evolucionário. . . .	28
2.15	Exemplo de um quadro genérico de <i>SCRUM</i> . . . . .	29
3.1	Exemplo de entrada <i>On-Board Diagnostics</i> . . . . .	35
4.1	Diagrama com o esquema geral do sistema. . . . .	40

## **Lista de Tabelas**

2.1	Comparação de microcontroladores . . . . .	21
2.2	Comparação entre placas de desenvolvimento . . . . .	22
2.3	Comparação entre <i>Single Board Computer</i> . . . . .	23
4.1	Conceitos do cenário estudado . . . . .	39

## **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>Solid Disk</i>
UDESC	<i>Universidade do Estado de Santa Catarina</i>
TCC	<i>Trabalho de Conclusão de Curso</i>

# 1 Introdução

A área de telemetria, aquisição de dados e sensoriamento em geral de um veículo é usada de forma extensiva competições automobilísticas aonde amostragem em tempo real é necessária para centenas de sensores que permitem aos engenheiros manterem o rastreamento da performance 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.

Alguns problemas do cenário de veículos *off-road* podem ser levantados, (ARAÚJO, 2006) trás 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*, além de levantados nos processos de engenharia de *software* quando se comenta sobre os requisitos do sistema (como 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 circuitaria que seja sensível a calor e não esteja protegida;
- Vibração: Pela natureza de uma prova *off-road*, existe muita vibração. Este problema pode acarretar em mal contato entre circuitos e placas;
- Lama

Foram então levantados junto com o grupo Velociraptor os componentes que seriam monitorados. Os seguintes itens foram levantados:

- Frequência de rotação;

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

Os dados serão recebidos no computador de boxe a partir do *SCOB* e transmitidos do cartão *SD* interno para a plataforma para atualização dos dados.

## **1.1 Estrutura do trabalho**

## 2 Fundamentação Teórica

Nesta sessão serão descritos alguns dos conceitos essenciais para a compreensão do trabalho. Inicialmente sera explicado como funciona a organização Baja SAE e as provas a 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 alem de informações sobre os microprocessadores estudados para servir como base do *SCOB* vão ser discutidas. Por último é feito uma análise de práticas de engenharia de *software* que podem ajudar na construção do sistema que será proposto.

### 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, (BRASIL, 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 (BRASIL, 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).

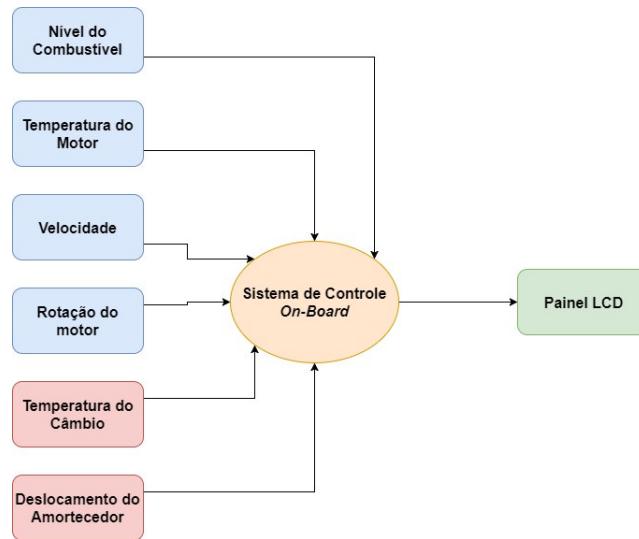
Os sensores presentes em um veículo se dividem em duas categorias (CHANDRAMANI; BHANDARI; HARIPRASAD, 2014) - aqueles que são feitos para o conforto do passageiro e aqueles feitos para garantir o funcionamento devido da parte dinâmica do veículo. O primeiro é importante para a lista de opcionais do carro 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.

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



Fonte: Autor.

As subseções a seguir explicam o funcionamento destes sensores.

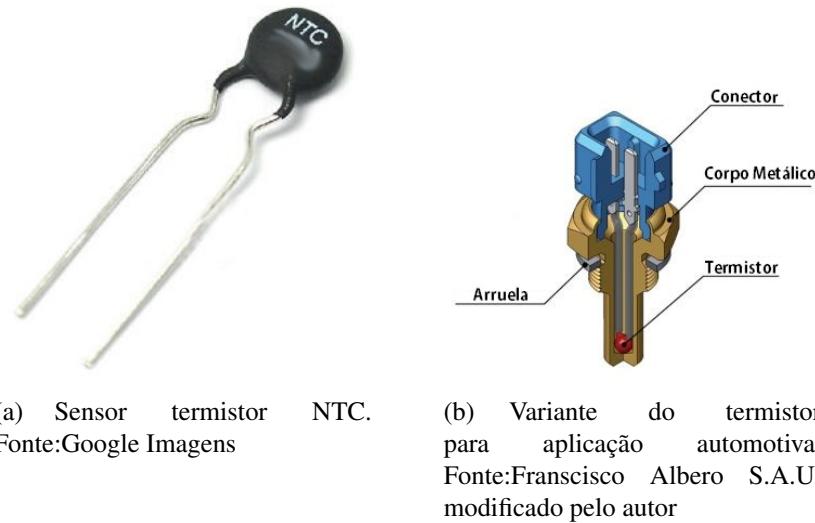
### 2.2.1 Temperatura do motor

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 espessura 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 normalmente encontrado

em lojas de eletrônica da forma demonstrada na Figura 2.3(a), porém, o utilizado no projeto baba Velociraptor é preparado para o ambiente automotivo com um corpo metálico para proteção contra ruídos e melhor posicionamento do sensor. A Figura 2.3(b) 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: Sensor termistor NTC e um exemplo do sensor específico para automóveis.



### 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 esta conectado a uma transmissão. Esta transmissão esta conectada a um eixo e consequentemente o eixo está conectado a 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, sua função é causar a centelha que começa o processo de explosão do combustível misturado com ar na câmara interna do motor.

Figura 2.3: Vela automotiva.



Fonte: Google Imagens.

A vela produz uma centelha a cada ciclo de explosão da gasolina e esta centelha é produzida a partir de uma carga de energia de 15V. 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 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 demonstrado 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 quadrática com frequência proporcional a velocidade do veículo (MTE-THOMSON, 2017) e que deve ser tratada de acordo com a sua aplicação. Este sensor é utilizado em carros da marca GM.

No bala 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 quadrático 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,

Figura 2.4: Sensor de velocidade MTE 73020.



Fonte: MTE-THOMSON.

resultando na fórmula matemática a seguir:

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



Fonte: UDESC Baja Velociraptor.

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

Aonde  $V$  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 da área da roda e  $\frac{3.6km}{h}$  transforma o resultado de metros por segundo para quilômetros por hora.

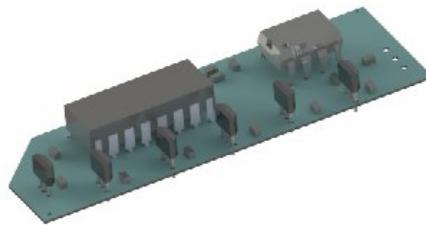
## 2.2.4 Nível do combustível

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 tirar 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 (BRASIL, 2017).

Para se adequar a estes novos regulamentos, foi utilizada uma nova técnica de medida da grandeza. Esta técnica consistem em utilizar uma boia dentro do tanque de gasolina com um imã e 5 sensores de efeito Hall US1881 fixados pelo lado de fora do tanque. Os sensores de efeito Hall tem seu valor de saída alterado quando a boia presente dentro do tanque passa por ele, desta forma é possível medir a quantidade de gasolina dentro do tanque sem que 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.

### 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 o torque sempre 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.

Figura 2.7: Câmbio CVT.



Fonte: Google Imagens.

Para obter essas informações do câmbio foi utilizado um sensor MLX90614 (Figura 2.9(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  $1^{\circ}\text{C}$  com uma mudança de dezenas de  $\mu\text{V}$ , um sensor de termopilha é 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 voltagem correspondente (FRADEN, 2016). A Figura 2.9(b) exemplifica este processo descrito. A saída deste sensor é analógica e vai conectada a entrada analógica do microcontrolador.

### 2.2.6 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

Figura 2.8: Sensor MLX90614 e seu funcionamento.



(a) Sensor MLX90614.

Fonte:Google Imagens

(b) Exemplo do funcionamento.

Fonte:(FRADEN, 2016)

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.



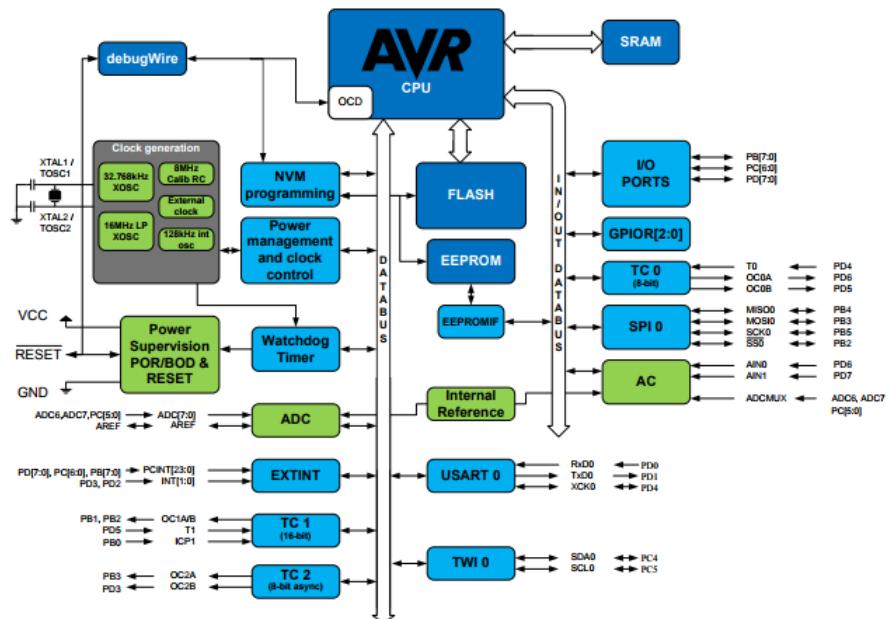
Fonte: SHARP.

## 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.

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



Fonte: Atmel.

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 bája 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

- 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.

a Tabela 2.1 faz uma comparação entre microcontroladores encontrados em placas de desenvolvimento no mercado das marcas Intel<sup>1</sup>, Atmel<sup>2345</sup> e NXP<sup>6</sup>. O microcontrolador com maior poder de processamento é o AT91SAM3X8E com velocidade de *clock* 5.25 vezes maior que o microcontrolador com mais lento e 48 vezes mais memória. O microcontrolador da Intel possui uma boa quantidade de memoria RAM, porém sua pinagem é menor que do ATmega328, o microcontrolador de menor desempenho da família Atmel visto neste comparativo.

Tabela 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.

<sup>1</sup>Quark SE Microcontroller C1000 Datasheet

<sup>2</sup>ATmega16U4/ATmega32U4 Datasheet

<sup>3</sup>ATmega328/P Datasheet

<sup>4</sup>Atmel ATmega640/V-1280/V-1281/V-2561/V Datasheet

<sup>5</sup>SAM3X / SAM3A Series Datasheet

<sup>6</sup>NXP Kinetis KL25 Sub-Family Datasheet

### 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. A Tabela 2.2 traz uma comparação entre várias placas de desenvolvimento que utilizam microcontroladores vistos na Tabela 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 eletrônicos<sup>7</sup>. A dimensão de altura das placas Arduino são estimativas, visto que oficialmente estes números não são divulgados.

Tabela 2.2: Comparação 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.

É possível tirar algumas análises desta tabela, 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 que possuem alguns componentes extras já citados no início da seção de fábrica. 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

<sup>7</sup><https://www.filipeflop.com/>

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 da uma demonstração de 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) trás 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 na Tabela 2.2. A Tabela 2.3 tem a comparação de algumas destas *Single Board Computer* na qual as dimensões são dadas em milímetro, todas as placas tiveram seus preços verificados no mês de novembro do ano 2017 em um *e-commerce* de componente eletrônicos<sup>8</sup>.

Tabela 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.

<sup>8</sup><https://www.filipeflop.com/>

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 todas as placas comparadas, o modelo Zero W tem a proposta mais atraente. Com memória RAM de 512 MB, velocidade de *clock* de 1000 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 na Tabela 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 Software

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

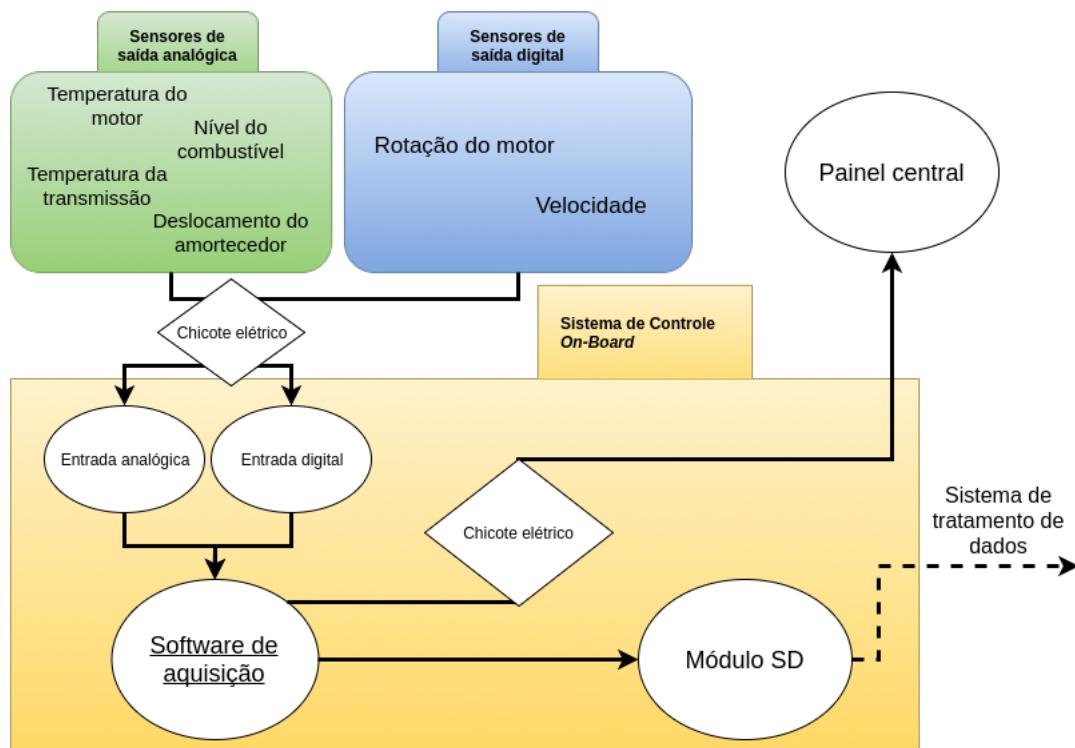
### 2.4.1 Aquisição de dados

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 possui é produzido por dois motivos: ele deve realizar a comunicação entre múltiplos microcontroladores em caso de uso de arquitetura distribuída, realizar qual-

quer 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 2.12 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 salvos em um cartão SD. A forma atual de armazenamento dos dados no cartão é com criação de arquivos de texto sem formatação específica, o que dificulta a leitura de dados em outros nós. Contudo será estudado a aplicação de alguns formatos de arquivo como *comma-separated values* e *javascript object notation*, ou a utilização de um banco de dados de modelo relacional caso seja possível a utilização de *Single Board Computers* como visto na Seção 2.3.1.

Figura 2.12: Diagrama exemplificando funcionamento do sistema de aquisição.



Fonte: Autor.

## 2.4.2 Tratamento de dados

O sistema de tratamento de dados deve receber as informações fornecidas pelo meio de transporte de informação, neste caso o cartão SD, e com esses dados atualizar o ambiente do programa.

As equipes de automobilismo não divulgam muitas informações sobre seus sistemas de tratamento de dados, principalmente aqueles que envolvem telemetria. 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++ (KATZOURAKIS, 2012) e utiliza de alguns conceitos comentados na Seção 2.4.1, como utilização de *comma-separated values* para formatação dos arquivos de dados.

Apesar da falta de literatura neste campo, este problema deve ser contornado com a utilização de métodos de engenharia de *software*. Na Seção 2.4.3 são discutidos modelos que podem ajudar no levantamento de requisitos junto a equipe e impulsionar alguns atributos do sistema positivamente.

## 2.4.3 Engenharia de software

A engenharia de *software* é fundamentalmente o uso de processos para atingir uma meta com melhor qualidade, previsível e econômico. A engenharia de *software* se baseia em princípios da engenharia para que suas atividades proposta, a fim de alcançar um objetivo final, sejam: interdependentes; com responsáveis (pessoas encarregadas de tarefas); e com entradas e saídas definidas (WAZLAWICK, 2013). Os processos de engenharia de *software* são feitos para o *software* de forma genérica, sendo assim o cunho do projeto não deve ser um empecilho para a adoção das técnicas, processos e atividades. Em Pereira (2012) é produzida uma tabela com um conjunto de atributos que é desejável em qualquer sistema de *software*. A Figura 2.13 mostra

esta tabela, todos os atributos comentados na tabela são relevantes para o trabalho proposto e alcançar-los é um objetivo específico deste trabalho, o que impulsiona o uso de alguns conceitos de tais processos.

Figura 2.13: Tabela com atributos desejáveis em um *software*.

COMPONENTE	DESCRIÇÃO
Facilidade de manutenção	O software deve ser escrito de modo que possa evoluir para atender às necessidades de mudança dos clientes. É um atributo fundamental, pois a mudança de software é uma consequência inevitável de um ambiente de negócios em constante mutação.
Confiança	O nível de confiança do software tem uma série de características, incluindo confiabilidade, proteção e segurança. Um software confiável não deve causar danos físicos ou econômicos no caso de falha no sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, a eficiência inclui tempo de resposta, tempo de processamento, utilização de memória etc.
Usabilidade	O software deve ser usável, sem esforço excessivo, pelo tipo de usuário para o qual ele foi projetado. Isso significa que ele deve apresentar uma interface com o usuário e documentação adequadas.

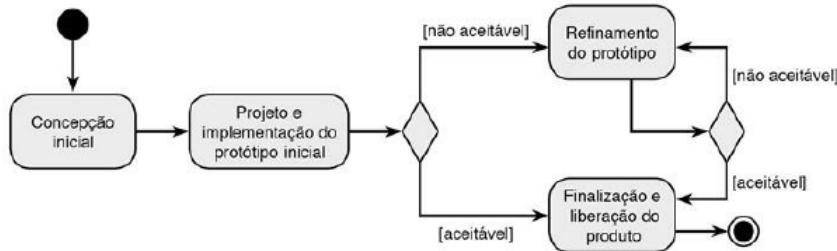
Fonte: (PEREIRA, 2012)

Existem dois tipos principais de abordagem quando se comenta em usar métodos de engenharia de *software*, a primeira é a utilização de modelos de processos prescritivos. Os modelos prescritivos são aqueles que se baseiam em uma descrição, geralmente em requisitos levantados junto com o cliente, para criar processos a serem seguidos durante a execução do projeto (WAZLAWICK, 2013). Alguns modelos que utilizam processos prescritivos são: o modelo cascata que introduz fases bem definidas e produção de documentação durante o processo; o modelo V que possui a execução de testes em diversos momentos em decorrer do projeto, o que enfatiza a importância dos mesmos para a construção do *software*; o modelo espiral proposto por Boehm e com forte orientação a redução de riscos do projeto.

O modelo evolucionário parece possuir maior relevância com o projeto proposto neste trabalho de conclusão de curso, nele o projeto é baseado em uma técnica de prototipagem conhecida como *cornerstone*. Nesta técnica são construídos protótipos no decorrer do processo, para avaliar os aspectos do sistema e entender melhor os requisitos e reduzir riscos. A Figura 2.14 produzida por Wazlawick (2013) demonstra um diagrama de atividades UML simplificado para um protótipo genérico utilizando o modelo de prototipagem evolucionária. Uma desvantagem citada no texto é em relação a dificuldade de gerência do projeto, pois é difícil manter uma boa previsão de tempo para o mesmo devido a volatilidade das fases. Tal desvantagem pode ser

superada com a ajuda de outro modelo sobre a prototipagem evolucionária, porém ao custo de aumento de complexidade.

Figura 2.14: Desenho exemplificando uma aplicação genérica do modelo evolucionário.



Fonte: (WAZLAWICK, 2013)

O outro tipo de abordagem para os métodos da engenharia de *software* é os métodos ágeis. Criados para suprir a necessidade da indústria de lançar produtos com mudança constante de requerimentos e impescindibilidade de desenvolvimento rápido utilizando uma metodologia que melhora a produtividade geral dos seus usuários. Os métodos ágeis se apoiam em uma atitude de incremento sobre mudanças, sejam elas de requisitos, desenvolvimento ou entrega (SOMMERRVILLE, 2010). Contudo a melhor forma de explicar a filosofia do método ágil é com o manifesto para desenvolvimento ágil de *software* (AGILE, 2011), este comenta que os indivíduos e interações devem estar acima dos processos e ferramentas, *software* em funcionamento sobre documentação abrangente, colaboração com o cliente e resposta constante a mudanças.

Existem vários modelos de métodos ágeis e o escolhido para ser focado neste trabalho foi o método *SCRUM*. Esta escolha foi feita devido ao foco em produtividade do método e experiência prévia do autor no mesmo.

#### 2.4.3.1 SCRUM

O *SCRUM* é um método que foca em trazer o cliente para perto da equipe de desenvolvimento e com isto fazer entregas assertivas em ciclos mensais ou semanais, dependendo do tamanho da equipe de desenvolvimento. Existem três perfis importantes no método, eles são os seguintes (DESENVOLVIMENTOAGIL, 2014)(WAZLAWICK, 2013):

- **SCRUM master:** É um facilitador na tomada de decisão em relação as mecânicas do *SCRUM*;

- **Product owner:** O interessado no produto, quem irá utilizar os resultados, este papel tem maior peso na decisão de tarefas a serem feitas; e
- **SCRUM team:** Esta é a equipe de desenvolvimento, entre 6 a 10 pessoas.

O método trabalha em ciclos de desenvolvimento de semanas ou até mesmo um mês. Em cada ciclo, tarefas são levantadas da necessidade do cliente e colocadas no *sprint* da equipe. As tarefas ficam em um *product backlog* e são avaliadas em todo início de ciclo, se forem aprovadas elas são adicionadas ao *sprint backlog* e trabalhadas durante o ciclo. A Figura 2.15 trás um exemplo de um quadro genérico de SCRUM produzido por Wazlawick (2013). Os quadros de SCRUM e algumas outras mecânicas do modelo podem ser alterados para se encaixar melhor com o cenário em que elas são empregadas (SOMMERVILLE, 2010), na Seção 4.1 é proposto uma utilização de um quadro específico para o cenário do trabalho.

Figura 2.15: Exemplo de um quadro genérico de SCRUM.

História	Tarefas a fazer	Em andamento	Para verificar	Terminadas
Como usuário, eu... 6 pontos	Codificar... Testar... Testar...	Codificar... Testar...	Codificar...	Codificar... Codificar... Codificar... Testar...
Como usuário, eu... 5 pontos	Codificar... Codificar... Testar... Testar...	Testar...	Codificar... Testar...	Codificar... Testar... Testar...

Fonte: (WAZLAWICK, 2013)

### **3 Trabalhos Relacionados**

Nesta seção será comentado sobre alguns trabalhos que possuem um objetivo similar 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 a este trabalho proposto no quesito de manter a equipe em questão atualizada dos dados provindos do carro, a diferença é que o veículo é movida 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 radio 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...", 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 Taha (2010). O sistema é feito para um carro que utiliza 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 modulo de transmissão por radio 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 fala dos 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 Race-logic VBOX, ele tira 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 um 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*, e 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 para qual protocolo *wireless* utilizar para o fim 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 maiores taxas de transferência, porém no sistema analisado este requisito não era prioritário.

Estes dados não estão mais no escopo de projeto do trabalho proposto, porém este era um dos requisitos comentados pela equipe de baja SAE da UDESC, este artigo tem informações que podem ser muito úteis para um trabalho futuro no campo da telemetria veicular.

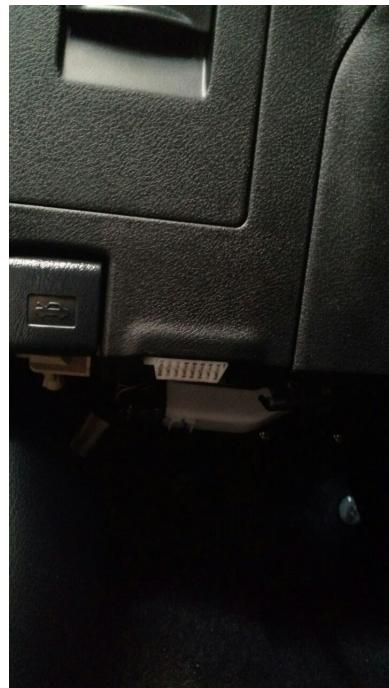
Um estudo feito por Araújo (2006) traz várias informações sobre o sistema eletrônico do Paraibaja, equipe de baja da Universidade Federal de Campina Grande. O artigo não é focado em telemetria e tratamento de dados, portanto não existe *software* para ser analisado nestes dois campos, mas este artigo comenta como a equipe fez para montar seu painel de instrumentos e o *firmware* que esta 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 levado 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 (BRASIL, 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, e 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 detalhado 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*) 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 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) trás uma abordagem diferenciada no 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 trás uma revisão de conceitos de elétrica, 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 Velo-ciraptor e pode ser adicionado ao *software* de tratamento de dados porém como um extra para testes, pois a bateria usada em provas 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 uso do 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 microprocessador 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 no 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 mesmas 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 tamanho 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 dos *software*, em cima dela é produzido um código, depois são realizados teste 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 é 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. Este projeto trouxe uma nova perspectiva ao trabalho, a ideia de adicionar conceitos de engenharia de *software* ao projeto adiciona em muito o que pode ser oferecido a equipe de baja da UDESC Velociraptor. A discussão feita sobre os conceitos de engenharia de *software* podem ser vistas na seção 2.4.

## 4 Projeto/Proposta de Solução

Os conceitos revisados até este momento seriam 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 do veículo de maneira **intuitiva, simplificada, modular e com armazenamento**. Estes três conceitos são detalhados na Tabela 4.1 a seguir:

Tabela 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

### 4.1 Objetivos

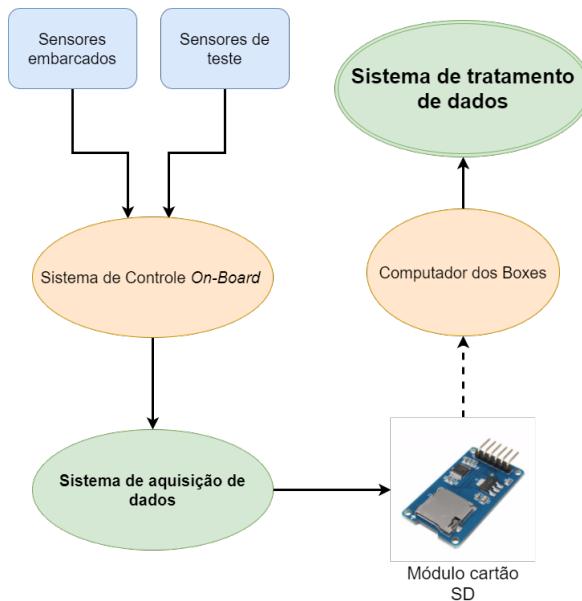
Os objetivos que devem nortear o trabalho a fim de sanar os problemas levantados podem ser subdivididos em objetivo principal, que é a meta final do trabalho, e objetivos secundários, que servem para medir a acertividade das medidas tomadas no curso do trabalho.

#### 4.1.1 Objetivo Geral

Produzir um sistema que forneça a equipe informações que irão ajudar aos setores do veículo (suspensão dianteira, suspensão traseira, freio, transmissão, eletrônica) 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 bája Velociraptor, porém ela deve ser atualizada para acompanhar os avanços do novo *software*; a parte de **tratamento** dos dados é o foco deste trabalho, nesta parte os dados serão armazenados de um cartão SD no SCOB e quando conectados a computadores de boxes, seus dados ali contidos são lidos por um *software* que permite visualização de gráficos e mostra de valores absolutos de maior importância. A Figura 4.1 é um diagrama explicando o esquema geral do sistema, desde a aquisição dos dados pelos SCOBs e tratamento nos boxes com o *software*. Os balões em azul indicam os sensores, os balões em laranja indicam *hardware* e balões em verde indicam *software*.

Figura 4.1: Diagrama com o esquema geral do sistema.



Fonte: Autor.

### 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 por telemetria;
- 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 ser criada e atualizada uma documentação do *software*;
- Deve se instaurar uma cultura de utilização de engenharia de *software* para facilitar manutenções adaptativas e manutenções corretivas.

## **5 Pré Ensaio**

### **5.1 Pré Ensaio**

aqui vai um ensaio sobre o projeto final =D

## **6 Considerações Finais**

### **6.1 Fortemente considerável**

aqui vai alguma consideração final =D

## Bibliografia

AGILE. *Manifesto for Agile Software Development*. 2011. Disponível em: <<http://agilemanifesto.org>>. Acesso em: nov. 2017.

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.

BRASIL, S. *Estudantes de engenharia constroem 30 carros off-road para competição*. ago 2016. Disponível em: <<http://portal.saebrazil.org.br/noticia/noticia-sae-brasil/Post/4151/Estudantes-de-engenharia-constroem-30-carros-off-road-para-competi>>

BRASIL, S. *Regulamento*. jun 2017. Disponível em: <<http://portal.saebrazil.org.br/programas-estudantis/baja-sae-brasil/regras>>.

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.

DESENVOLVIMENTOAGIL. *SCRUM*. 2014. Disponível em: <<http://desenvolvimentoagil.com.br>>. Acesso em: nov. 2017.

DIAS, J. E. A. Trabalho de conclusão de curso, *Eletrônica, Instrumentação e Telemetria do Veículo UFVBAJA*. Viçosa: UFV, dezembro de 2010. 50 p.

FRADEN, J. *Handbook of modern sensors: Physics, designs, and applications*. 2016. 1–758 p. ISSN 1098-6596. ISBN 9783319193038.

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>>.

- MTE-THOMSON. *Catálogo Eletrônico - 73020*. nov 2017. Disponível em: <<https://cate.mte-thomson.com.br/produto/detalhes/73020>>.
- NAUNHEIMER, H. et al. *Automotive transmissions: Fundamentals, selection, design and application*. 2011. ISSN 14337851. ISBN 9783642162138.
- NUNES, T. F. Trabalho de conclusão de curso, *Telemetria de um veículo Baja SAE através de rede CAN*. Natal: UFRG, junho de 2016. 49 p.
- PEREIRA, R. M. R. Trabalho de conclusão de curso, *Processos de Software para Implementações de Especificações Relacionadas ao Projeto Fórmula SAE/EESC/USP*. São Carlos: USP, novembro de 2012. 87 p.
- 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.
- SHARP. *GP2Y0A21YK0F Datasheet*. nov 2017. Disponível em: <[http://www.mouser.com/ds/2/321/gp2y0a21yk\\_e-3493.pdf](http://www.mouser.com/ds/2/321/gp2y0a21yk_e-3493.pdf)>.
- SOMMERVILLE, I. *Software Engineering*. 2010. 790 p. ISSN 0014-2972. ISBN 9780137035151.
- 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.
- WAZLAWICK, R. *ENGENHARIA DE SOFTWARE: CONCEITOS E PRÁTICAS*. Elsevier Brasil, 2013. ISBN 9788535260854. Disponível em: <<https://books.google.com.br/books?id=Qtg4VUkE0V0C>>.