

Telemetria veicular e análise de dados aplicados a veículos Baja

Proposta do Projeto

Júlio Ohfugi Yamaguti
FGA
UNB – Universidade de Brasília
Gama, Brasil
julio.ohfugi21@gmail.com

Guilherme Felix De Andrade
FGA
UNB – Universidade de Brasília
Gama, Brasil
guilhermefelixandrade@gmail.com

Resumo- Este projeto tem como finalidade a criação de uma plataforma embarcada para o monitoramento em tempo real do carro baja da equipe de competição Unbaja da universidade de Brasília, fazendo o mapeamento da pista e aquisição de dados para o melhor dimensionamento do veículo.

I. INTRODUÇÃO

Os veículos Baja são protótipos de um veículo off road com a função de uso fora de estrada, com quatro ou mais rodas e motor padrão, estabelecido pelo regulamento da competição, de 10 HP, que devem ser capazes de transportar pilotos com até 1,90 m de altura, e com peso de até 113,4 kg. Os sistemas de eletrônica, suspensão, transmissão e freios, assim como o próprio chassi, são projetados e construídos pelas equipes [1].

O projeto Baja SAE foi criado na Universidade da Carolina do Sul, Estados Unidos, sob a direção do Dr. John F. Stevens, e a primeira competição ocorreu em 1976. O ano de 1991 marcou o início das atividades da SAE BRASIL, e em 1994 lançou o Projeto Baja SAE BRASIL [2].

Em 1995, realizou-se a primeira competição nacional, na pista Guido Caloi, bairro do Ibirapuera, cidade de São Paulo. No ano seguinte a

competição foi transferida para o Autódromo de Interlagos, onde permaneceu até o ano de 2002. A partir de 2003 a competição passou a ser realizada em Piracicaba, interior de São Paulo, no ECPA – Esporte Clube Piracicabano de Automobilismo [2].

Desde 1997 a SAE BRASIL também apoia a realização de eventos regionais do Baja SAE BRASIL, por meio de suas Competições Regionais. Desde então dezenas de eventos foram realizados em vários estados do país como Rio Grande do Sul, São Paulo, Minas Gerais e Bahia [2].

O programa Baja SAE BRASIL é um desafio proposto aos estudantes de Engenharia que oferece a aplicação na prática dos conhecimentos adquiridos em sala de aula, com o objetivo de incrementar sua preparação para o mercado de trabalho. Ao participar do programa Baja SAE, o aluno se envolve com um caso real de desenvolvimento de um veículo off road, desde sua concepção, projeto detalhado, construção e testes. Além disso, as equipes vencedoras da etapa

nacional são convidadas a participar da competição internacional, nos Estados Unidos [2].



Figura 1 - Prova de conforto, Regional Nordeste, 2017.



Figura 2 - Prova Enduro de resistência, Regional Nordeste, 2017.

Além da aplicação prática das matérias passadas em sala de aula, a competição valoriza muito a parte teórica realizada e as simulações, que são necessárias para justificar todo o dimensionamento do projeto [3].

As avaliações e notas são dadas da seguinte maneira:

Avaliação de Projeto - 320 pontos	
Relatório de Projeto	120
Avaliação de Projeto Dinâmico	20
Apresentação de Projeto e Finais de Apresentação de Projeto	180
Eventos dinâmicos - 280 pontos	
Aceleração	45
Velocidade Máxima	45
Tração	45
Lama	45
Suspensão	70
Manobrabilidade	30
Enduro de Resistência - 400 pontos	
Pontuação total	1000

Figura 3 - Pontuações da competição[3].

II. OBJETIVOS

Este projeto almeja a modelagem de um sistema embarcado que possa ser aplicado em veículos de competição, tendo como foco o veículo baja da equipe de competição Unbaja da Universidade de Brasília campus Gama, com o intuito de informar para os integrantes os dados qualitativos e quantitativos sobre o veículo e seu posicionamento em tempo real durante o período que antecede a corrida, para que seja possível a realização de uma análise e dimensionamento adequado e resistente de cada subsistema do projeto.

Será realizado o mapeamento do carro na pista a partir dos dados adquiridos com o módulo gps e módulo giroscópio e acelerômetro (latitude, longitude, altitude, velocidade, inclinação do chassi), ocorrerá o estabelecimento de uma comunicação com um computador externo para a exibição e avaliação dos dados em tempo real, a partir da criação de zonas de calor em um mapa. Com isso será possível a obtenção dos perfis de terreno que possibilitará ter um conhecimento em relação ao tipo de comportamento que o veículo realizará sobre diferentes trechos de pistas. Além disso, os dados serão essenciais para o cálculo de esforços que serão exercidos nos componentes do carro.

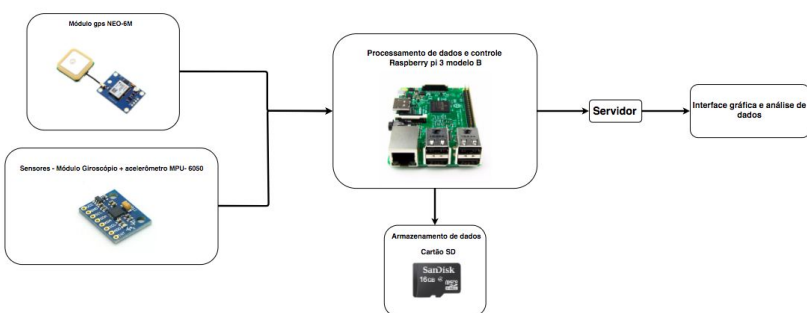


Figura 4 – Diagrama de funcionamento do sistema

III. JUSTIFICATIVA E BENEFÍCIOS

Para realização de um projeto e construção de um veículo é necessário a análise e estudo de inúmeros fatores, como por exemplo, as características de terreno em que ele será utilizado e o tipo e a quantidade de esforços que cada parte do veículo sofrerá. Essas informações são essenciais para a realização de um dimensionamento de cada subsistema do carro, de forma coerente e apta para um devido funcionamento do veículo [5].

Temos como exemplo um sistema de suspensão veicular, que possui como objetivo absorver as irregularidades do solo e dissipa-las sem afetar o desempenho do veículo e o conforto dos ocupantes. Hoje em dia diversas bibliografias apresentam como efetuar analiticamente o projeto de uma suspensão tendo como meta as características citadas. Mas é necessário, com o projeto construído, validar tais características. Para isso os projetistas precisam de ferramentas que durante o deslocamento do veículo forneçam informações precisas sobre o trajeto percorrido pelo veículo, e as respostas da carroceria a tais deslocamentos. Informações como velocidade em determinado trecho de pista, e deslocamento angular lateral da carroceria, são dados de entrada para modelos matemáticos e softwares de análise em multi corpos que são capazes, em ambiente virtual, de simular e otimizar o comportamento de um veículo antes de seu desenvolvimento final. Com isso é possível economizar recursos e gastos com prototipagem [6].

Empregou-se o uso do Raspberry Pi para realizar o projeto que se diferencia de microcontroladores na questão de permitir múltiplas tarefas do software nele embarcado e o monitoramento em tempo real. Com isso é possível gerenciar os sistemas que serão empregados para cumprir as tarefas a serem resolvidas no projeto.

Para a captação dos dados de latitude, longitude e altitude, necessitou-se a utilização do módulo gps NEO-6M, pois este atende as necessidades do projeto com um custo baixo no mercado, suas dimensões são compactas, podendo ser colocada no painel do Baja e comunica-se com a raspberry pi 3 através da comunicação UART, fazendo a interpretação dos dados no padrão NMEA, conexão de internet e envio das informações para o servidor que com o banco de dados de posicionamento, é capaz de disponibilizar remotamente as informações coletadas, gerando um ambiente de fácil entendimento para o usuário.

V. Materiais utilizados

Processamento de dados e controle Raspberry pi 3 modelo B



Armazenamento de dados

Cartão SD



VII. Descrição de Software

Módulo GPS NEO-6M



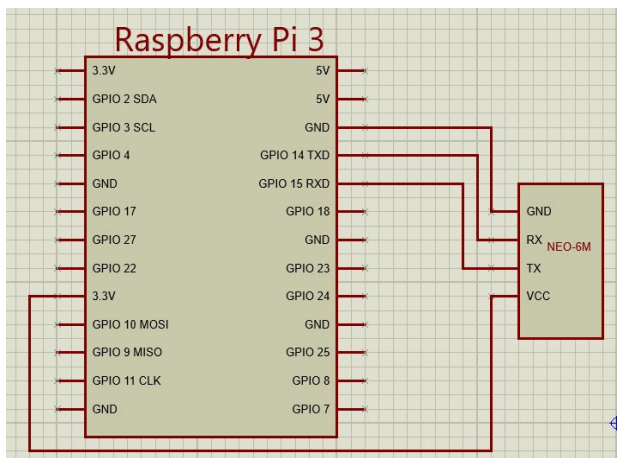
Sistema

Linux e Raspbian



VI. Descrição de Hardware

A raspberry é conectada diretamente ao módulo gps NEO-6M, com os grounds ligados, o pino vcc do gps ligado à saída de 3.3v, as comunicações Rx ligado ao Tx e Tx ligado ao Rx da raspberry.



A. Interface gráfica

Para a construção da interface gráfica e realização dos primeiros testes, utilizou-se das linguagens de programação HTML, em conjunto com CSS e JavaScript. Além disso, utilizou-se do Google Maps JavaScript AP e da biblioteca google.maps.visualization, para a geração do mapa e de uma camada de mapa de calor, respectivamente, utilizando-se de suas classes, construtores e métodos[10].

```
1 <!DOCTYPE html>
2
3 <html>
4
5 <head>
6
7 <title>Mapa UNBAJA</title>
8 <meta name="viewport" content="initial-scale=1.0">
9 <meta charset="utf-8">
10
11 <style>
12
13 #map {
14   height: 100%;
15 }
16
17 html, body {
18   height: 100%;
19   margin: 0;
20   padding: 0;
21 }
22 </style>
23
24 </head>
```

Figura 5 - Código HTML, primeira parte.

A primeira parte do código se dispõe de acordo com a figura 5. Na primeira linha do código é declarado que o documento a ser escrito é do tipo HTML5. Logo após, na linha 3, é declarado o início da tag html, e tudo que estiver descrito abaixo, até o fechamento da tag, será considerado um documento html.

Já a descrição da linha 5 a linha 22, está representado a parte do cabeçalho, ou seja, é responsável pelas configurações comportamentais do site. As primeiras informações do programa são o título da página(linha 7), as instruções de controle das dimensões e escala da página(linha 8) e a definição do uso dos conjuntos de caracteres latinos(linha 9). Além disso, da linha 11 a 22, é

realizado a definição do estilo de informação para a página HTML, em que é configurado para que a página ocupe todo o navegador[8].

```
27 <body>
28
29 <div id="map"></div>
30
31 <script>
32 var map, heatmap;
33 function initMap() {
34   map = new google.maps.Map(document.getElementById('map'), {
35     center: {lat: 37.775, lng: -122.434},
36     zoom: 13
37   });
38
39   heatmap = new google.maps.visualization.HeatmapLayer({
40     data: getPoints(),
41     map: map
42   });
43 }
44
```

Figura 6 - Código HTML, segunda parte.

A segunda parte do código é apresentado de acordo com a figura —. Na linha 27 do código, é declarado o início da tag body, em que tudo que vai aparecer, visualmente, tem q estar na área body. Logo depois, na linha 29, é criado um elemento div, que é nomeado e obtém uma referência para o elemento map, no modelo de objetos do documento (DOM) do navegador. Isso é feito para reservar um lugar para mapa, para que o mesmo seja exibido em uma página. Além disso, utiliza-se do CSS para definir a altura do div do mapa para “100%”[7].

A partir da linha 31 até a linha 550, encontra-se a tag script, em que tudo que tiver entre <script> e </script> é considerado um código em linguagem JavaScript, responsável por manipular objetos que estejam dentro do navegador[8].

Logo em seguida, é declarado uma variável “map” e “heatmap”, linha 32, e declara-se uma função “initMap”, linha 33. Dentro dessa função é criado um objeto “map”, pertencente a classe “google.maps.Map”, e logo em seguida, é chamado o

construtor “Map(document.getElementById('map'), { center: {lat: 37.775, lng: -122.434}, zoom: 13

});”. Esse construtor é responsável por criar um novo mapa dentro do dado HTML, com o zoom escolhido, e a latitude e longitude, utilizadas, para determinar o local, que para a fase de teste, utilizou-se um local em São Francisco[7].

Ainda dentro da função, é criado o objeto “heatmap”, linha 39, e chama-se o construtor “heatmap = new google.maps.visualization.HeatmapLayer({ data: getPoints(), map: map });”. Esse construtor é responsável por criar o mapa de calor, a partir dos dados de latitude e longitude retornados pela função “getPoints()”, que para teste, foram determinados valores de exemplo, fornecidos pelo Google API[9].

```
552 <script async defer
553   src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDSbHv2gnqaJTBxgBjaGviONytVz8A99VA6libraries=visualization&callback=initMap">
554 </script>
555
556 <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDSbHv2gnqaJTBxgBjaGviONytVz8A99VA6libraries=visualization&callback=initMap"
557   async defer></script>
558
559 </body>
560
561 </html>
562
```

Figura 7 - Código HTML, terceira parte.

A terceira parte do código é apresentado de acordo com a figura —. Da linha 552 a linha 557 são carregados a Google Maps JavaScript API e a biblioteca de Visualization, para que possamos utilizar de seus conteúdos.

Ao abrir o arquivo em um Browser, obteve-se a interface de acordo com a figura 8.

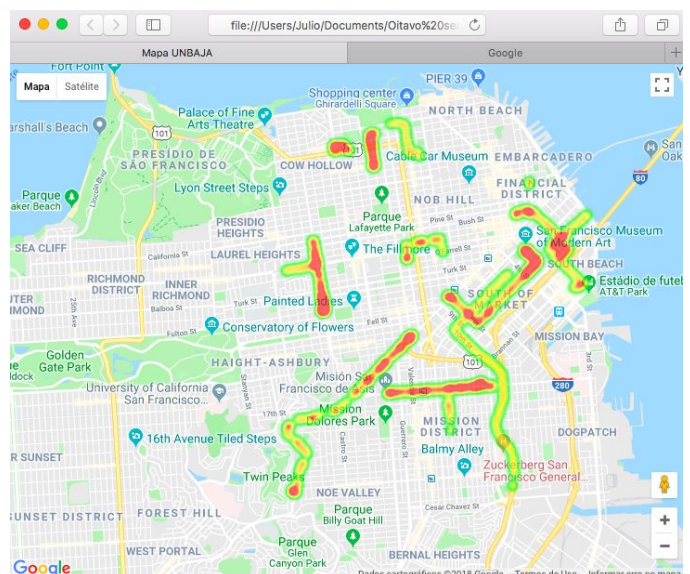


Figura 8 - Interface gráfica..

B. Padrão NMEA

O módulo gps NEO-6M comunica-se através de um protocolo definido pela National Marine Electronics Association(NMEA) [4].

Os dados provenientes deste dispositivo são fornecidos em pacotes chamados de sentenças NMEA. Existem vários tipos diferentes de sentenças com finalidades distintas, tais como latitude, longitude, altitude, velocidade, dentre outras [4].

Todas as sentenças de NMEA consistem em até 82 caracteres em ASCII. Cada sentença começa com o símbolo \$ e termina com um carriage return e o consequente começo de nova linha. Todos os campos de dados são delimitados por vírgulas (,) e têm comprimento variável. Os campos nulos também são delimitados por vírgulas (,) mas não contêm nenhuma informação. O primeiro campo de dados é um campo de endereço, ao passo que o último campo de dados é um controle de soma (checksum) [4].

C. Comunicação UART

O módulo gps e a raspberry pi 3 comunicam-se a partir da comunicação UART. O Raspian aplica a porta serial assíncrona (UART) para acesso remoto ao terminal e para utilizá-la é necessário desabilitar este acesso remoto a partir dos seguintes comandos.

```
$ sudo rasp-config
```

Advanced options -> Serial, e desabilite o acesso

```
$ sudo cat /boot/config.txt
```

Neste arquivo trocar o 0 pelo 1 em:

```
enable_uart = 1
```

```
$sudo reboot
```

```
$sudo nano /boot.cmdline.txt
```

Verificar se o arquivo contém :

```
dwc_otg.lpm_enable=0 root=/dev/mmcblk0p2  
rootfstype=ext4 elevator=deadline fsck.repair=yes  
rootwait quiet splash  
plymouth.ignore-serial-consoles
```

Instalar o programa que interpreta os dados NMEA

```
$ sudo apt-get install gpsd gpsd-clients
```

Após a reinicialização é necessário desabilitar os serviços ainda pendentes na 'serialS0', utilizando o novo serviço da versão Raspbian conhecido como serial-getty.

```
$ sudo systemctl stop serial-getty@serialS0.service
```

```
$ sudo systemctl disable  
serial-getty@serialS0.service
```

Com o programa instalado utiliza-se os seguintes comandos para a visualização dos dados no terminal.

```
$sudo gpsd /dev/ttyS0 -F /var/run/gpsd.sock
```

```
$ sudo gpsmon /dev/ttyS0
```

Oteve-se o seguinte resultado ao ser testado o
gps.

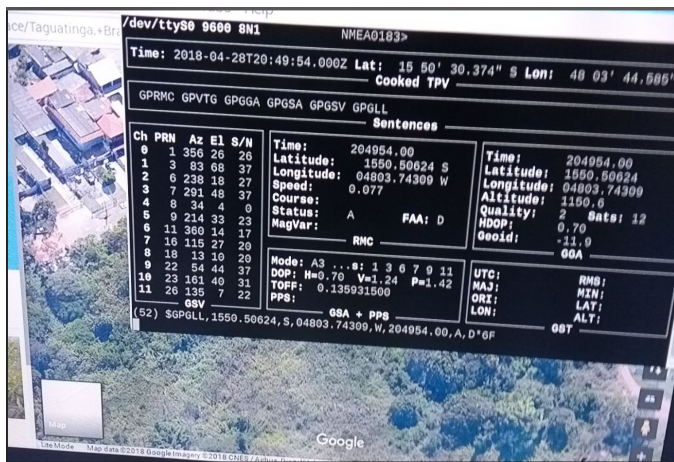


Figura 9 - Dados do módulo gps.

VIII. Conclusões e possíveis melhorias

Fez-se a comunicação da raspberry com o módulo gps NEO-6M a partir da comunicação UART, sendo gerado uma string no padrão NMEA e o programa instalado interpreta esse padrão gerando os dados de latitude e longitude necessários.

Para a próxima fase do projeto será feita a codificação do padrão NMEA para valores que serão interpretadas pelo código html, e serão apresentados pela interface gráfica de mapas. Além disso, será construído um servidor, onde os dados obtidos serão salvos em um arquivo, para que possa ser acessado pela equipe, e para a realização das análises de perfil de pista e aceleração lateral.

VII. Referências

[1]<http://www.petrobras.com.br/fatos-e-dados/equipes-e-dados/equipes-brasileiras-de-sae-baja-se-destacam-em-competicacao-internacional.htm>

[2]<http://portal.saebrasil.org.br/>

[3]<http://portal.saebrasil.org.br/Portals/0/PE/Baja%20Nacional%202018/RATBSB.pdf>

[4]Evandro Rui Fernandes. Sistema On-Line de Localização e Roteamento Urbano. Curitiba, Novembro de 2006.

[5] Brito, Matheus Hoffmann. *Desenvolvimento e Dimensionamento Do Projeto Do Subsistema De Direção De Veículo Fora-de-estrada*. Departamento de Engenharia Mecânica, 2002.

[6]William, F; Douglas, L. Race Car Vehicle Dynamics. SAE International, 1994.

[7]<https://developers.google.com/maps/documentation/javascript/tutorial>

[8]<https://www.youtube.com/watch?v=epDCjksKMo>

[9]<https://developers.google.com/maps/documentation/javascript/heatmaplayer>

[10]<https://developers.google.com/maps/documentation/javascript/reference#HeatmapLayerOptions>

Apêndice

Códigos principais do sistema

A. Código HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Mapa UNBAJA</title>
```

```
<meta name="viewport" content="initial-scale=1.0">
```

```
<meta charset="utf-8">
```

```
<style>
```

```
#map {
  height: 100%;
}
```

```
html, body {
  height: 100%;
```

```

margin: 0;
padding: 0;
}
</style>

</head>

<body>

<div id="map"></div>

<script>
var map, heatmap;
function initMap() {
  map = new
google.maps.Map(document.getElementById('map'), {
  center: {lat: 37.775, lng: -122.434},
  zoom: 13
});

  heatmap = new
google.maps.visualization.HeatmapLayer({
  data: getPoints(),
  map: map
});
}

function getPoints() {
  return [
    new google.maps.LatLng(37.782551, -122.445368),
    new google.maps.LatLng(37.782702, -122.400470),
    new google.maps.LatLng(37.782915, -122.400192),
    new google.maps.LatLng(37.783137, -122.399887),
    new google.maps.LatLng(37.783414, -122.399519),
    new google.maps.LatLng(37.783629, -122.399237),
    new google.maps.LatLng(37.783688, -122.399157),
    new google.maps.LatLng(37.783716, -122.399106),
    new google.maps.LatLng(37.764457, -122.405094),
    new google.maps.LatLng(37.762932, -122.405398),
    new google.maps.LatLng(37.762126, -122.405813),
    new google.maps.LatLng(37.761344, -122.406215),
    new google.maps.LatLng(37.760556, -122.406495),
    new google.maps.LatLng(37.759732, -122.406484),
    new google.maps.LatLng(37.758910, -122.406228),
    new google.maps.LatLng(37.758182, -122.405695),
    new google.maps.LatLng(37.757676, -122.405118),
    new google.maps.LatLng(37.757039, -122.404346),

```

```

    new google.maps.LatLng(37.756335, -122.403719),
    new google.maps.LatLng(37.755503, -122.403406),
    new google.maps.LatLng(37.754665, -122.403242),
    new google.maps.LatLng(37.753837, -122.403172),
    new google.maps.LatLng(37.751266, -122.403355)
  ];
}
</script>

<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDS
bHv2gnqaJT8xBjaGviCNytVz8A99VA&libraries=visualizat
ion&callback=initMap">
</script>

<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDS
bHv2gnqaJT8xBjaGviCNytVz8A99VA&callback=initMap"
async defer></script>

</body>

</html>

```