

Entregável Desafio Eletrônica - Aquisição de Dados

21ª Competição Baja SAE BRASIL 2024 – Etapa Sul

DESAFIO TÉCNICO

INTRODUÇÃO

O ensaio de *Coast Down* é um método empírico amplamente utilizado na indústria automotiva visando determinar as forças resistivas que atuam sobre um veículo em movimento. As informações coletadas permitem a análise das forças de resistência, como o arrasto aerodinâmico e o atrito de rolamento, oferecendo dados cruciais para o desenvolvimento e a otimização do desempenho do veículo.

O escopo inicial do projeto é desenvolver um *data logger* que atenda às diretrizes do comitê técnico da SAE Brasil e esteja em conformidade com a norma NBR 10312 (2019). O dispositivo será projetado para utilização no protótipo da equipe e para atender aos requisitos técnicos especificados no informativo 3. O *data logger* será equipado com sensores e sistemas de aquisição de dados que garantam a coleta precisa e confiável das informações durante o teste, contribuindo para a avaliação detalhada das características de resistência do veículo e possibilitando melhorias no design e na performance.

DESENVOLVIMENTO

Fundamentação teórica

O procedimento do teste *coastdown* consiste em acelerar o veículo até atingir uma velocidade constante e, em seguida, desacoplar o sistema de transmissão, permitindo que o veículo desacelere exclusivamente devido às forças resistivas até que pare completamente. Durante esse processo, a velocidade do veículo é registrada em tempo real por meio de instrumentação embarcada.

A composição das forças que se opõem ao movimento é descrita pela equação de *Road Load*, que representa o somatório dos atritos responsáveis pela desaceleração do

veículo, sendo uma função dependente da velocidade (GILLESPIE, 1992). Os principais parâmetros dessa equação são a resistência ao rolamento, associada ao contato dos pneus com o terreno, e o arrasto aerodinâmico, influenciado pela interação da geometria do veículo com o fluxo de ar. Contudo, é essencial reconhecer que diversos outros fatores afetam a desaceleração, como o atrito mecânico na transmissão, a velocidade e direção do vento, o aplanejamento da superfície, as condições ambientais e fatores proporcionais à velocidade.

A equação teórica pode ser entendida ao analisar as forças que agem sobre um veículo se movendo em um plano, sendo a resistência ao rolamento e a resistência aerodinâmica representadas pelas Eq. (1) e (2) respectivamente.

$$Q_r = f * G \quad (1)$$

$$Q_a = \frac{1}{2} * \rho * A_f * C_d * v^2 \quad (2)$$

Onde:

- G : Peso do veículo;
- f : Coeficiente de atrito de rolamento;
- ρ : Massa específica do ar;
- v : Velocidade do veículo;
- A_f : Área projetada da seção transversal do veículo;
- C_d : Coeficiente de resistência aerodinâmica.

Deste modo, o *Road Load* é a soma das equações anteriores e descreve como as resistências ao movimento variam conforme a velocidade do veículo. Assim, fazendo o produto dos termos constantes, pode-se escrever a equação teórica na forma de:

$$RL(v) = A * v^2 + C \quad (3)$$

Onde:

$$\bullet \quad A = \frac{1}{2} * \rho * A_f * C_d \quad (4)$$

$$\bullet \quad C = f * G \quad (5)$$

A aceleração causada pela ação do conjunto de forças resistivas é expressa pela equação (LEAL; ROSA; NICOLAZZI, 2001) :

$$RL(v) = -m * a \quad (6)$$

O fenômeno físico, descrito por uma equação quadrática, será investigado por meio de uma regressão polinomial aplicada aos dados de velocidade coletados pelo sistema eletrônico, com finalidade de determinar os coeficientes que governam esse comportamento.

Escopo detalhado

Com o fim de estabelecer o que abrange o desenvolvimento do *datalogger*, foi estabelecido o escopo de projeto, definido como conjunto de características e funções que identificam um produto, serviço ou resultado, conforme o PMI (2021). Para tal, inicialmente foram definidos os principais requisitos do sistema, levando em conta os seguintes fatores: confiabilidade e frequência de aquisição dos dados; adequação do *hardware* ao sensor proposto pelo desafio; compatibilidade e padronização com o sistema embarcado já utilizado pela equipe; adaptação aos fatores ambientais nocivos ao *hardware*; transmissibilidade, armazenamento, tratamento e amostragem de dados; custo e disponibilidade de componentes. Através desses, foram estabelecidos como requisitos mínimos do sistema:

- Capacidade de aquisição de velocidade do eixo traseiro superior a 230 Hz em uma onda quadrada, considerando a condição de 40 km/h e roda fônica de 36 dentes;
- Adequação da leitura ao sinal de saída do sensor proposto pelo informativo 3 e da alimentação da bateria utilizada pela equipe, conforme Tabela 1;
- Disponibilidade dos componentes utilizados no projeto conforme o estoque disponível na equipe, com custo total inferior a R\$ 150,00;
- Resolução da aquisição inferior a 0,2 km/h e exatidão de 0,4 km/h, conforme a norma NBR 10312;
- Funcionalidade de armazenamento de dados e geração de relatórios em tempo real, com confiabilidade e frequência suficientes para a aplicação;

- Invólucro compacto, peso inferior a 0,30 kg e com estanqueidade adequada ao ambiente de competição;
- Capacidade de comunicação através de protocolo ESP-NOW e WiFi, para adequação com o sistema já existente na equipe.

Requisito	Meta
Resolução da velocidade (km/h)	0,20
Exatidão da velocidade (km/h)	0,40
Frequência de medição (Hz)	230,00
Peso total do case (kg)	0,30
Custo total (R\$)	150,00
Tensão de leitura (V)	3,30
Corrente máxima de leitura (mA)	4,00
Alimentação do datalogger (V)	12,00

Tabela 1: Requisitos do datalogger. Fonte: Autoral.

Com todos os pré-requisitos técnicos estabelecidos, foi definido o fluxo da informação, conforme o fluxograma presente na Figura 1, partindo das medições da roda fônica, conversão dos dados, gravação da informação em cartão SD e envio dos dados para o sistema embarcado presente no protótipo da equipe. Por fim, os dados são processados e disponibilizados em um servidor externo.

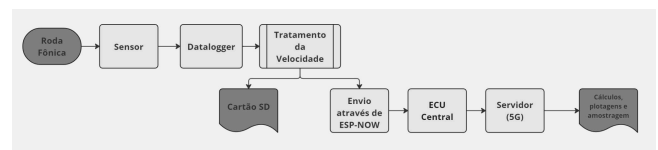


Figura 1: Fluxo do sistema de coleta de dados. Fonte: Autoral.

Desenvolvimento do hardware

Após a definição dos requisitos do projeto, o desenvolvimento do hardware foi realizado no *software Altium Designer*, visto que além de ser amplamente utilizado na indústria, possui a possibilidade de gerenciamento de biblioteca e componentes. Priorizou-se dar continuidade à filosofia de modularidade do projeto eletrônico atual, otimizar e compactar a placa de circuito com a implementação de componentes SMD (*Surface-Mount Device*), bem como considerar a disponibilidade de recursos dos patrocinadores da equipe. Desta maneira, com base na literatura de John Catsoulis em *Designing Embedded Hardware*, realizou-se a escolha dos componentes eletrônicos.

Como é possível observar no diagrama elétrico do sistema (Figura D1), a alimentação, protegida por um fusível de 1A, é realizada por uma fonte *buck*, a qual converte a tensão de 12V, relativo à bateria, para 3.3V. Para isolamento elétrico, essencial para proteção em casos de pico de tensão, e

transmissão de sinais, utilizou-se um optoacoplador, cujo sinal de entrada é estabilizado por um resistor *pull-up*. Para tanto, dois conectores foram posicionados, sendo um para o *buck* e o outro para leitura do sinal do MOSFET do sensor instalado na roda fônica. Em caso de alta impedância, o mesmo resistor *pull-up* excita o sinal para 12V, o qual possibilita o chaveamento do sinal para verdadeiro (3,3 V) ao microcontrolador via optoacoplador. Em contrapartida, quando em GND, esse componente garante um estado lógico falso (GND) no pino de leitura, como exhibe a Figura 2.

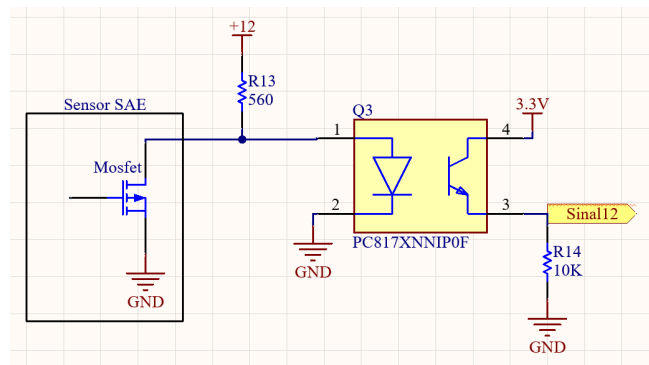


Figura 2: Utilização do optoacoplador para leitura do sensor. Fonte: Autoral.

Além do sistema citado anteriormente, implementou-se um segundo sistema de monitoramento que pode ser utilizado para a leitura caso ocorra alguma falha ou dano com o circuito anterior. Possui uma trilha que conecta-se diretamente a um outro pino de leitura do microcontrolador. Sendo um sistema de redundância, optou-se em utilizar valores nominais divulgados no Informativo 03, no qual o sistema aplica uma tensão de 3,3 V via resistor de *pull-up* no pino do sinal do sensor fornecido para o teste, e a partir dessa tensão é possível diferenciar um sinal *HIGH* e *LOW*.

O *ESP32-WROOM-32*, pelo seu custo-benefício, foi definido como processador. Esse microcontrolador é especialmente adequado devido à presença de dois núcleos que possibilita a execução simultânea do processamento dos dados e da comunicação *ESP-NOW* e *Wi-Fi*, o qual já é utilizado no sistema embarcado do veículo devido à confiabilidade e ao alcance de envio. Somado a isso, a fim de fornecer uma interface USB para comunicação serial, empregou-se o CP2102, amplamente utilizado em módulos de desenvolvimento da *ESPRESSIF*. É pertinente observar que um cartão SD foi incorporado ao sistema para viabilizar o armazenamento e a análise subsequente de dados por meio do protocolo SPI (*Serial Peripheral Interface*). Essa redundância tem como objetivo assegurar a continuidade da coleta e a preservação de

dados em situações onde possam ocorrer interrupções na comunicação com o servidor da equipe.

Com fim de estabelecer a precisão da leitura dos dados, tendo em vista que a frequência de aquisição é suficiente (o *ESP32* oferece 30MHz em portas seriais, conforme *datasheet* do microcontrolador), o limitante inicial da medição é a roda fônica. Assim, através da equação 7 foi calculada a resolução, que partindo do diâmetro eficaz dos pneus (D_{ef}) e quantidade de dentes da roda fônica (N), apresenta a resolução de 0,0484 m, resultado utilizado posteriormente para estabelecimento da resolução em termos de velocidade.

$$R_{roda\ fônica} = D_{ef} * \pi / N \quad (7)$$

Desenvolvimento do firmware

O *firmware* foi desenvolvido para atender às necessidades de aquisição de dados visando a análise de desempenho do veículo em tempo real. Como ferramenta de desenvolvimento, optou-se pelo uso do Arduino IDE, devido à sua ampla compatibilidade com diversos microcontroladores e facilidade de uso, o que acelera tanto a prototipagem quanto a depuração (ARDUINO, 2018). A fim de garantir a manutenibilidade e legibilidade do código, foi adotado a programação orientada a objetos (Figura B1) em linguagem C++ junto a padrões de programação modular, organizando o *firmware* em classes e arquivos distintos que encapsulam funcionalidades específicas, conforme as práticas recomendadas para código limpo e sustentável (MARTIN, 2012). Esse modelo de desenvolvimento, além de facilitar na documentação do projeto, permite que métodos de comunicação sejam facilmente intercambiáveis através da definição de interfaces e uso de polimorfismo entre elas. Com o objetivo de aproveitar os dois núcleos da *ESP32* isolou-se a leitura e tratamento dos dados de seu envio ao servidor, através da utilização do *FreeRTOS* como sistema operacional para gerenciamento das *tasks*, conforme o fluxograma apresentado na Figura 3.

O funcionamento do *firmware* é centralizado na classe “*Manager*”, que orquestra as operações principais, como a inicialização do *hardware*, processamento de dados do sensor de velocidade e comunicação dos dados filtrados. A classe “*Comunicação*” serve como interface base para os métodos de comunicação via *ESP-NOW* e *Wi-Fi*, com cada método implementado em suas respectivas subclasses, “*EspNow*” e “*WiFiCom*”. Os dados são armazenados utilizando a classe “*CartaoSD*”, responsável pela criação e

manipulação de arquivos CSV em um cartão SD, o que facilita a análise posterior. A filtragem dos dados é realizada pela técnica da média móvel, com uma janela de análise de apenas 5 valores, cuja classe suaviza as leituras de velocidade do sensor, eliminando variações abruptas e proporcionando uma leitura mais estável e precisa ao sistema sem impacto significativo no atraso em relação ao valor real. O gerenciamento de interrupções e a comunicação com o *hardware* são tratados através do uso de forma eficiente com tarefas separadas para envio de dados e processamento contínuo, otimizando o desempenho e permitindo uma operação multitarefa eficaz no ESP32 (ESPRESSIF SYSTEMS, 2024). Essa estrutura modular e clara melhora a facilidade de manutenção e a escalabilidade do sistema, o qual possibilita a adição ou substituição de funcionalidades com impacto mínimo no restante do código.

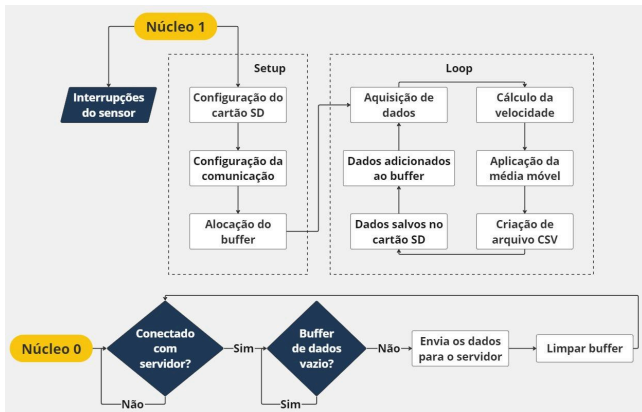


Figura 3: Fluxograma do firmware. Fonte: Autoral.

Para realizar a coleta do sinal do sensor, optou-se em utilizar uma rotina de interrupção por borda de subida. Nela calcula-se a diferença de tempo da interrupção passada em microssegundos. A partir dessa informação, há uma função na *task* no núcleo 1 responsável em calcular a velocidade a partir da diferença de tempo entre as duas últimas interrupções como mostrado na fórmula abaixo:

$$v = de * \pi / (n * \Delta t) \quad (8)$$

Onde os valores acima correspondem a:

- de : Diâmetro efetivo;
- n : Número de dentes da roda fônica;
- Δt : Diferença de tempo entre as interrupções.

O valor resultante da velocidade, junto com o tempo em microssegundos no qual essa função é executada é salvo em um arquivo CSV presente no cartão SD. Esse sistema está presente como uma forma de redundância, garantindo

a integridade dos dados em caso de instabilidade na conexão. Após realizar o *backup* desses dados, eles são inseridos em um *buffer* para envio ao servidor.

A *task* de envio dos dados, localizada no núcleo 0, é responsável exclusivamente pela comunicação entre o sistema e o servidor. Quando há algum dado no buffer, é realizado o envio via protocolo *Wi-Fi*, conectando-se a uma rede local e enviando os dados mediante protocolo *MQTT* diretamente ao servidor. Alternativamente, o envio pode ocorrer via *ESP-NOW*, após o emparelhamento com outro dispositivo *ESP32* conectado à rede via modem *LTE*, como no protótipo desenvolvido pela equipe. Esse acesso a uma rede com ampla cobertura em quase todo o território nacional garante uma alta taxa de transferência de dados de forma confiável a um custo acessível. O pacote de dados que será enviado ao servidor conterá os parâmetros: número do teste, timestamp do microcontrolador (μs) e velocidade.

Precisão

Com fim de determinar a resolução da leitura final de velocidade, verificou-se o valor em microssegundos equivalente a 40km/h - valor limite estabelecido para o ensaio, com uso da equação 9.

$$v = R_{roda\ fônica} * 1000000 / \Delta t \quad (9)$$

O valor obtido foi de 4359 μs . A partir desse, com variação de um microssegundo, foi estabelecida a medição seguinte a 40km/h, de 4260 μs . Através dos dois valores, foram calculados v_1 e v_2 , obtendo-se a resolução total de, aproximadamente, 0,002 m/s, ou 0,007 km/h.

$$R = v_1 - v_2 \quad (10)$$

Dado que a situação crítica de menor precisão da medição é com a maior velocidade, isso demonstra que o maior passo de precisão do *datalogger* está conforme a meta estabelecida no escopo através da NBR 10312. Pelos mesmos cálculos é possível verificar a escolha de medição em microssegundos ao invés de milissegundos, já que a resolução seria mil vezes maior, 7 km/h, o que seria insuficiente para o ensaio. Também é possível verificar através do mesmo tipo de cálculo, que quanto maior a velocidade do veículo, menor a precisão. Isso pode ser verificado no gráfico da figura 4, que no eixo vertical esquerdo exibe a escala de precisão em microsegundos e no eixo direito em milissegundos.

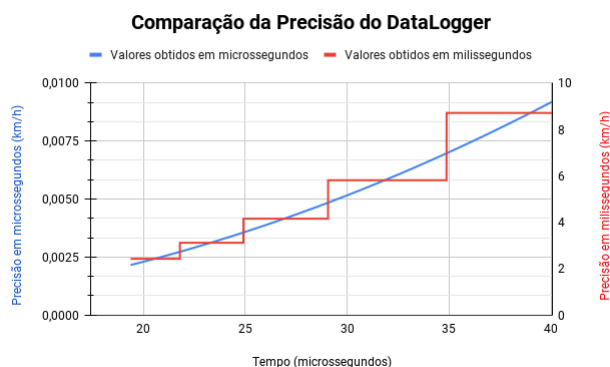


Figura: 4. Comparação de precisão. Fonte: Autoral.

Case de proteção

Para facilitar a fixação do sistema no veículo, uma case foi projetada no software SolidWorks (figuras A2 e A3), levando em consideração, principalmente, vedação e empacotamento. O invólucro conta com espaços para entrada do chicote adequados a conectores IP67 do modelo SP2110, abertura para comunicação USB do microcontrolador e utilização de O'rings para melhor estanqueidade às condições externas. Posto isso, com uma largura de 100x51x80mm e espessura de 4mm, torna-se viável a fixação desse componente na proteção de derramamento com parafusos, devido à proximidade a partir de onde os dados serão aquisitados. Para garantir boa resistência, baixo custo e facilidade de impressão, foi escolhido o material PLA, com massa final de 90g.

Software

Para determinar o *Road Load* de forma precisa, são utilizados métodos integrativos e derivativos para obter as informações de deslocamento e aceleração a partir dos dados de velocidade adquiridos pelo sensor na roda fônica. Esses dados são então ajustados por uma regressão quadrática, resultando na equação:

$$RL(v) = A * v^2 + B * v + C \quad (11)$$

Esse processo permite a identificação dos coeficientes que descrevem as forças resistivas ao movimento, em especial o coeficiente quadrático A, pois representa a resistência aerodinâmica. A partir desse, pode-se calcular o coeficiente de resistência aerodinâmica C_d utilizando a fórmula:

$$C_d = \frac{2 * A}{\rho * A_f} \quad (12)$$

Para facilitar o acompanhamento dos dados do veículo em tempo real, um servidor web foi desenvolvido, com auxílio das ferramentas Node.js e Express, os quais promovem a construção de um servidor rápido e facilmente escalonável. O Node.js é uma plataforma que permite a execução do código do lado do servidor, destacando-se pelo seu modelo assíncrono e pela capacidade de conexões múltiplas. O Express, por sua vez, é um framework facilitador para a criação de rotas e APIs para manipulação de dados. A linguagem de programação escolhida para o servidor foi o JavaScript, devido à sua flexibilidade e ao vasto ecossistema de bibliotecas, além da alta compatibilidade com o tipo de dados ao qual o servidor terá acesso.

Assim, a fim de armazenar os dados da telemetria, o *MongoDB* foi escolhido como sistema de gerenciamento do banco de dados (SGBD) da equipe, devido à robustez e escalabilidade. A comunicação entre o servidor e datalogger é realizada através de requisições MQTT, que estabelece uma via bidirecional e uma conexão persistente, garantindo que o sistema embarcado, ao se conectar à internet, possa enviar dados continuamente e receber comandos sem a necessidade de renegociar a conexão a cada envio. A escolha desse protocolo de mensagens, capaz de lidar com múltiplos dispositivos, é justificada pela sua leveza, baixo consumo de banda e energia, além de oferecer suporte a criptografia via TLS/SSL, o qual torna comunicação mais segura ao proteger os dados contra interceptações e acessos não autorizados. Ademais, a tecnologia Server-Sent Events (SSE) possibilita ao servidor enviar atualizações automáticas para o cliente via HTTP, de modo a criar uma interação fluida e sem interrupções.

A possibilidade de download dos gráficos gerados pelos cálculos aqui apresentados permite um acompanhamento do desempenho do veículo ao longo do tempo. Destaca-se que alta responsividade e criação de um sistema “*user friendly*”, como apresentado no anexo C1, foram prioridades durante a elaboração do software, ambos para proporcionar uma boa experiência ao usuário. A aplicação inclui funcionalidades para simular o desempenho do veículo com base nos dados de telemetria através do cálculo das acelerações e deslocamentos, gerando gráficos detalhados conforme o exemplo presente no anexo C2. Esses são então disponibilizados em formato PDF para download, permitindo o acompanhamento e registro do histórico de desempenho.

CONCLUSÃO

Através do apresentado no relatório, é possível confirmar que as metas estabelecidas para o datalogger foram atingidas com sucesso. O dispositivo atende a todas as funcionalidades exigidas pelo ensaio de Coast Down, possui um hardware com precisão conforme as normas vigentes, e inclui soluções que garantem maior confiabilidade na leitura e armazenamento dos dados. Além disso, o firmware oferece funcionalidade RTOS e modularidade, e o invólucro foi projetado com soluções adequadas ao ambiente de Baja, validadas pela equipe em competições.

O software inclui tratamento, cálculo e plotagem de gráficos conforme os requisitos estabelecidos. O custo total do datalogger foi de R\$ 87,81, inferior ao orçamento previsto de R\$ 150,00 (conforme tabela A1), e todos os requisitos do projeto foram atendidos conforme descrito nos tópicos relevantes do desenvolvimento, com enfoque na resolução de aquisição de 0,007 km/h que é excelente dado o requisitado pela norma que regula esse tipo de ensaio.

Com resultados satisfatórios em todas as frentes de desenvolvimento, se torna viável iniciar a etapa de fabricação do datalogger. A próxima fase será a validação em campo com o teste de desaceleração em pista livre, o que permitirá realizar otimizações adicionais no sistema de transmissão do veículo.

REFERÊNCIAS

1. ARDUINO. What is Arduino?. [S. l.], 5 fev. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 9 set. 2024.
2. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (2019). NBR 10312: Veículos rodoviários automotores leves - Determinação da Resistência ao descolamento por desaceleração livre em pista de rolamento e simulação em dinamômetro.
3. CATSOULIS, John. Designing Embedded Hardware. 2. ed. [S. l.]: O'Reilly Media, 2005a. 396 p. ISBN 9780596007553.
4. ESPRESSIF SYSTEMS. FreeRTOS Overview. [S. l.], 2024. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html>. Acesso em: 9 set. 2024.

5. GILLESPIE, Thomas D. 1992. Fundamentals of Vehicle Dynamics. Premiere Series Books. Warrendale: SAE International.
6. LEAL, L. DA C. M.; ROSA, E. DA; NICOLAZZI, L. C. Uma introdução à modelagem quase-estática de veículos automotores de rodas. Brasil: Publicação interna do GRANTE – Departamento de Engenharia Mecânica, Universidade Federal de Santa Catarina. Florianópolis, p.247. 2001
7. MARTIN, Robert Cecil. Clean Code: A Handbook of Agile Software Craftsmanship. [S. l.: s. n.], 2012.
8. PROJECT MANAGEMENT INSTITUTE. Project Management Body of Knowledge 7ed, 2019. 196p.
9. SAE BRASIL. Informativo 03: Desafio Técnico Baja SAE BRASIL 2024 – Etapa Sul. Disponível em: <https://saebrasil.org.br/programas-estudantis/baja-sul/mensagens-e-informativos/>. Acesso em: 3 set. 2024.
10. ESPRESSIF SYSTEMS. Datasheet ESP32-WROOM-32-N4. Disponível em: https://www.lcsc.com/datasheet/lcsc_datasheet_2108131730_Esspressif-Systems-ESP32-WROOM-32-N4_C82899.pdf. Acesso em: 2 set. 2024.

APÊNDICE A

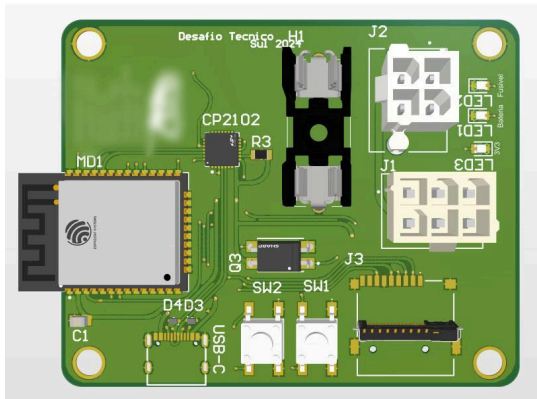


Figura A1: Placa do circuito. Fonte: Autoral.

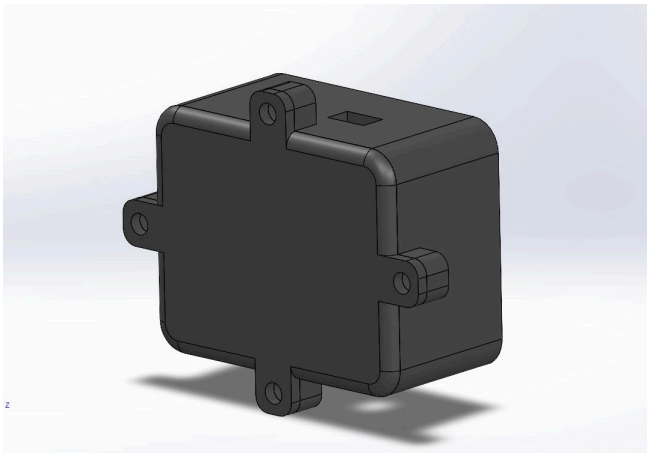


Figura A3: Modelagem da case (Externa). Fonte: Autoral.

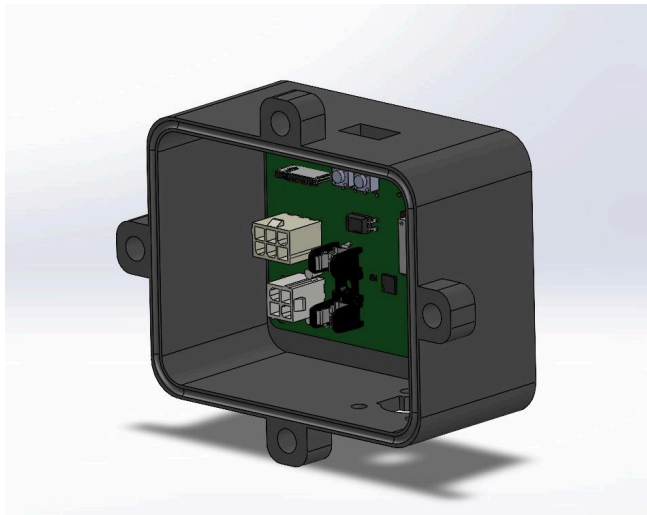


Figura A2: Modelagem da case. Fonte: Autoral.

Item	Empacotamento	Qtd	Custo (R\$)
Capacitor 10uF	0805	2	0,08
Capacitor 1uF	0805	1	0,05
Capacitor 100nF	0805	3	0,05
CP2102N-AD2-GQFN28R	QFN-28	1	11,25
Diodo ESD5Z5.0T1G	SOD-523	3	0,21
Diodo BAT54JFILM	SOD-523	2	0,66
Fuse Holder 5x20mm	-	1	0,87
Mini Fit 3x2	-	1	4,01
Mini Fit 2x2	-	1	4,00
Conector MICRO-SD-CARD	-	1	0,97
Led Vermelho 1,8V	0805	3	0,51
ESP32-WROOM-32	-	1	24,45
Transistor MMBT2222ALT1G	SOT-23	2	0,22
Foto Transistor PC817XNNIP0F	SOP-4	1	0,13
Resistor 5k1	0805	2	0,01
Resistor 1k	0805	1	0,01
Resistor 10k	0805	10	0,01
Resistor 22k1	0805	1	0,01
Resistor 47k5	0805	1	0,01
Resistor 560	0805	3	0,02
Resistor 68	0805	1	0,01
Push Button	15715638	2	0,20
Conector USB-C	-	1	0,34
Filamento PLA 0,5mm	NA	1	9,60
Cartão SD micro	NA	1	20,00
Conector IP67 SP2110	-	1	10,13
Total:	-	-	87,81

Tabela A1: BOM do datalogger. Fonte: Autoral.

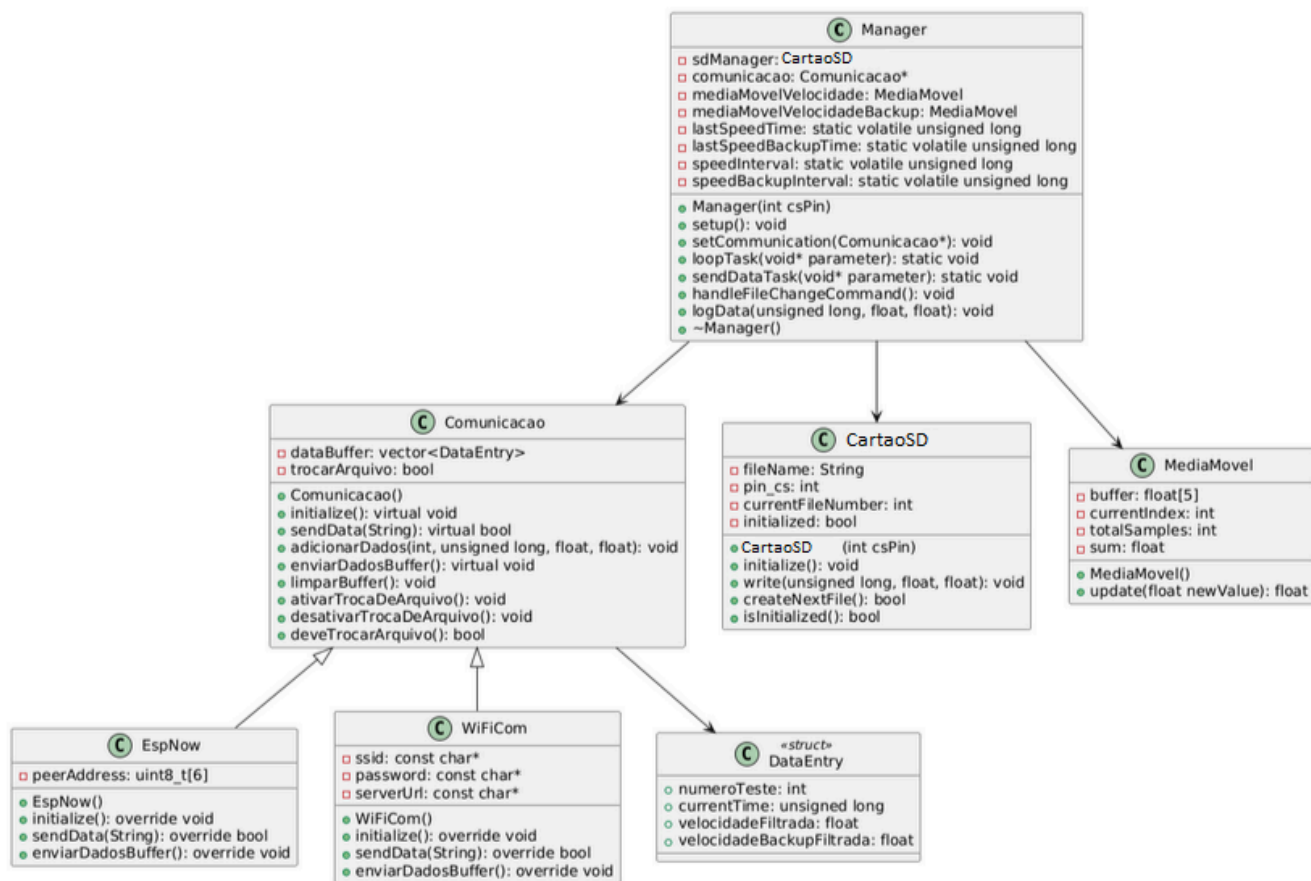


Figura B1: UML. Fonte: Autoral.

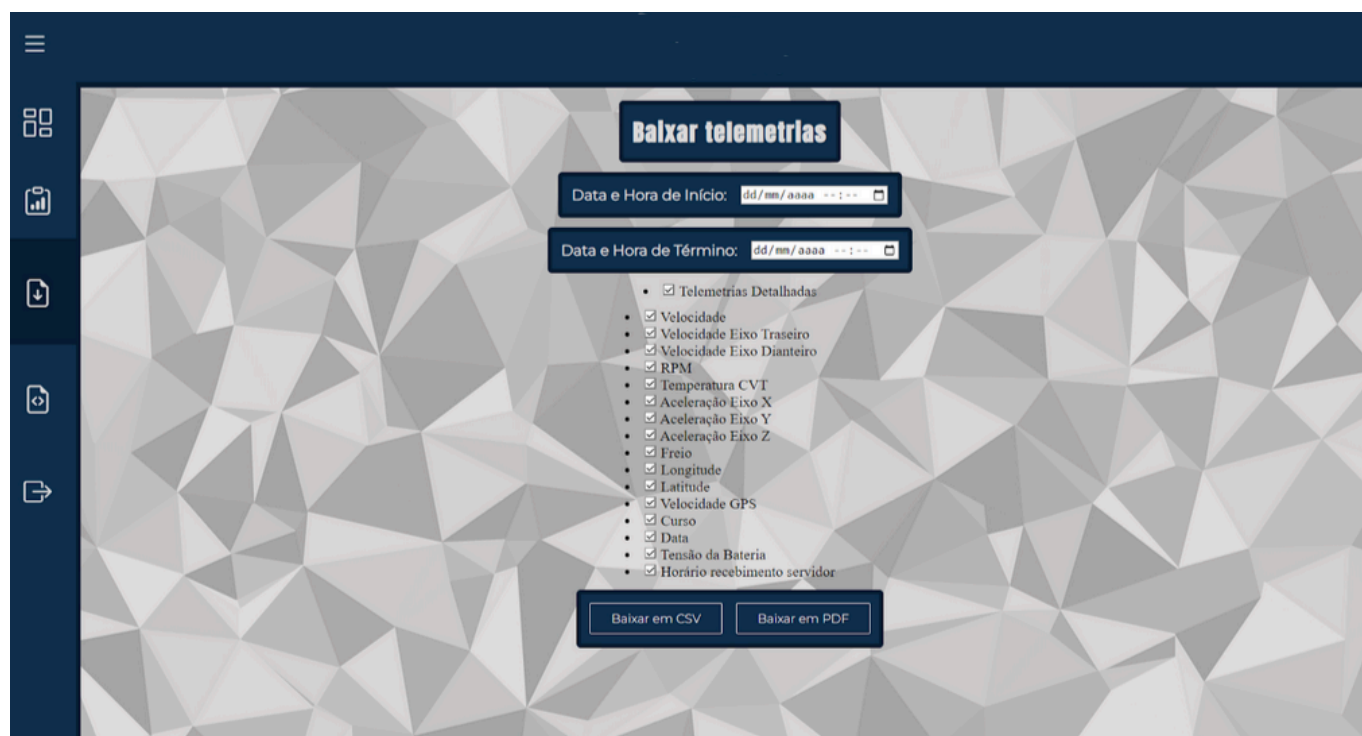


Figura C1: Interface página de downloads. Fonte: Autoral.

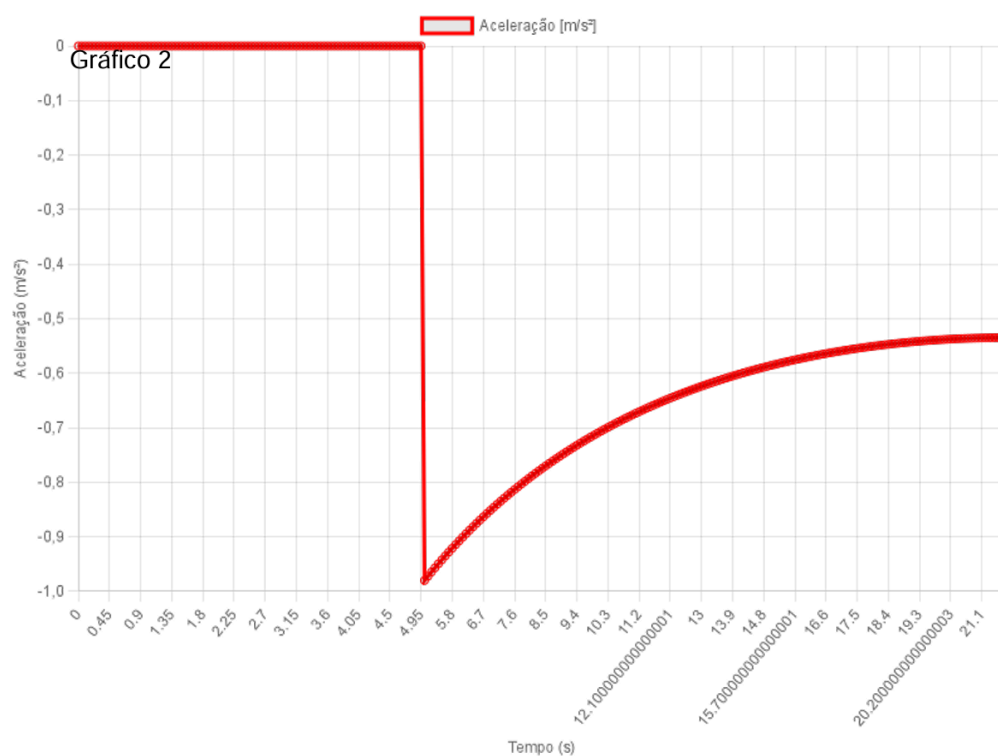


Figura C2: Exemplo de gráfico presente no PDF do desafio técnico. Fonte: Autoral.

APÊNDICE D

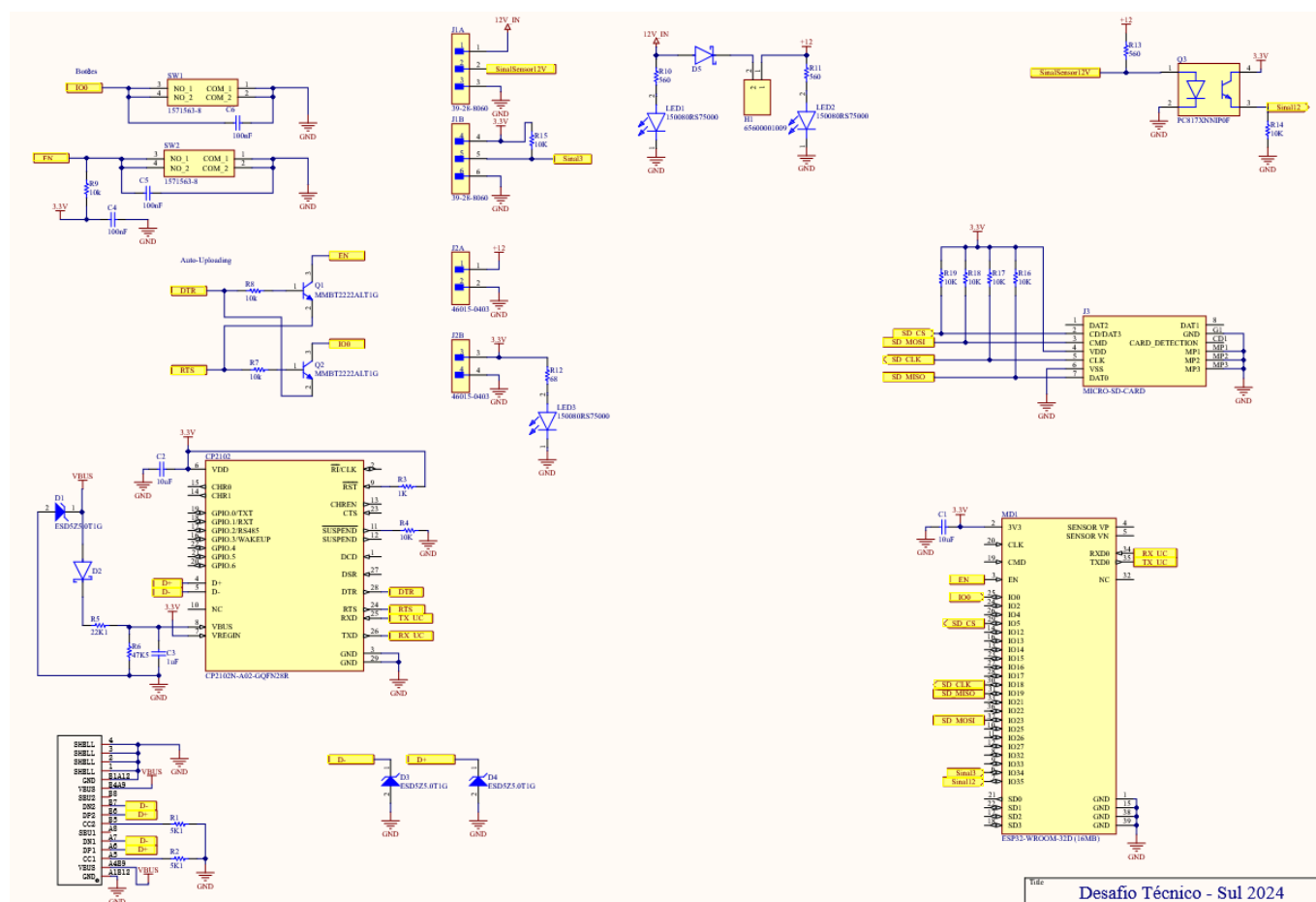


Figura D1: Diagrama Elétrico do sistema. Fonte: Autoral.