

**INTEGRAÇÃO DE PAINEL DE INSTRUMENTOS AUTOMOTIVO REAL, EM  
SIMULADORES VEICULARES, VIA PROTOCOLO CAN-BUS**

**REAL AUTOMOTIVE INSTRUMENT CLUSTER INTEGRATION INTO VEHICLE  
SIMULATORS USING CAN-BUS PROTOCOL**

**INTEGRACIÓN DE PANEL DE INSTRUMENTOS AUTOMOTRIZ REAL EN  
SIMULADORES VEHICULARES A TRAVÉS DEL PROTOCOLO CAN-BUS**

 10.56238/ramv20n15-003

**Clebson Queiroz dos Santos**

Graduando em Engenharia da Computação  
Instituição: Centro Universitário Mário pontes Jucá (UMJ)  
E-mail: clebsonqs@hotmail.com

**André Queiroz Rocha**

Graduando em Engenharia da Computação  
Instituição: Centro Universitário Mário pontes Jucá (UMJ)  
E-mail: deco.qrocha@gmail.com

**Francisco Abud Nascimento**

Orientador do curso de Engenharia da Computação  
Instituição: Centro Universitário Mário pontes Jucá (UMJ)  
E-mail: francisco.abud@umj.edu.br

---

**RESUMO**

A crescente busca por realismo nos simuladores automobilísticos tem motivado a integração de componentes físicos aos sistemas virtuais, visando proporcionar uma experiência mais imersiva ao usuário. Neste contexto, o presente trabalho propõe a utilização de painéis de instrumentos reais conectados a simuladores digitais, como solução para ampliar a fidelidade visual e funcional da simulação. Para isso, emprega-se o software open source SimHub, responsável por coletar os dados de telemetria gerados pelos jogos automobilísticos, convertê-los e transmiti-los a um microcontrolador. Este dispositivo interpreta os dados e os envia ao painel físico por meio do protocolo CAN-Bus (Controller Area Network bus), amplamente utilizado na comunicação entre módulos eletrônicos automotivos. Como resultado, o painel responde dinamicamente às informações recebidas, exibindo dados como rotação do motor (RPM), velocidade, temperatura, nível de combustível, indicadores de marcha, setas e controle de tração, por meio de ponteiros e displays. Conclui-se que a integração proposta é tecnicamente viável e contribui significativamente para o aumento da imersão e do realismo em ambientes de simulação automotiva.

**Palavras-chave:** Simuladores. Painel de Instrumentos. SimHub. CAN-Bus. Telemetria. Microcontrolador.



## ABSTRACT

The growing pursuit of realism in automotive simulators has motivated the integration of physical components into virtual systems, aiming to provide a more immersive user experience. In this context, the present work proposes the use of real instrument clusters connected to digital simulators as a solution to enhance the visual and functional fidelity of the simulation. To achieve this, the open-source software SimHub is employed. SimHub is responsible for collecting the telemetry data generated by the racing games, converting it, and transmitting it to a microcontroller. This device interprets the data and sends it to the physical dashboard via the CAN-Bus (Controller Area Network bus) protocol, which is widely used in communication between automotive electronic modules. As a result, the dashboard responds dynamically to the received information, displaying data such as engine revolutions per minute (RPM), speed, temperature, fuel level, gear indicators, turn signals, and traction control, through pointers and displays. It is concluded that the proposed integration is technically viable and contributes significantly to increasing immersion and realism in automotive simulation environments.

**Keywords:** Simulators. Instrument Cluster. SimHub. CAN-Bus. Telemetry. Microcontroller.

## RESUMEN

La creciente búsqueda de realismo en los simuladores automovilísticos ha motivado la integración de componentes físicos a los sistemas virtuales, con el objetivo de proporcionar una experiencia de usuario más inmersiva. En este contexto, el presente trabajo propone la utilización de paneles de instrumentos reales conectados a simuladores digitales como solución para ampliar la fidelidad visual y funcional de la simulación. Para ello, se emplea el software de código abierto SimHub, responsable de recolectar los datos de telemetría generados por los juegos de carreras, convertirlos y transmitirlos a un microcontrolador. Este dispositivo interpreta los datos y los envía al panel físico mediante el protocolo CAN-Bus (Controller Area Network bus), ampliamente utilizado en la comunicación entre módulos electrónicos automotrices. Como resultado, el panel responde dinámicamente a la información recibida, mostrando datos como revoluciones por minuto del motor (RPM), velocidad, temperatura, nivel de combustible, indicadores de marcha, intermitentes y control de tracción, a través de punteros y pantallas. Se concluye que la integración propuesta es técnicamente viable y contribuye significativamente al aumento de la inmersión y el realismo en entornos de simulación automotriz.

**Palabras clave:** Simuladores. Panel de Instrumentos. SimHub. CAN-Bus. Telemetría. Microcontrolador.



## 1 INTRODUÇÃO

A simulação veicular é uma área de pesquisa e desenvolvimento em constante expansão, de grande importância para diversas aplicações, desde o entretenimento até o treinamento profissional de motoristas. A eficiência desses sistemas depende, da capacidade de replicar o comportamento dinâmico e a sensação de estar em um veículo real.

Geralmente, os simuladores digitais enfatizam o aprimoramento da física e do feedback tátil (como volantes e pedais), mas negligenciam a fidelidade visual do cockpit. Muitos sistemas utilizam interfaces virtuais na própria tela, causando uma desconexão entre a ação física do usuário e a resposta visual do veículo. O que reduz a imersão e o realismo, podendo dificultar a interpretação dos dados do simulador em comparação a um veículo Real.

Sendo assim, surgiu uma questão norteadora para o presente artigo: É tecnicamente viável, utilizando um protocolo de comunicação padrão da indústria automotiva, para integrar um painel de instrumentos automotivo real em simuladores veiculares? Qual o impacto na imersão e no realismo da simulação?

Este estudo é justificado pela necessidade de eliminar essa disparidade entre os sistemas virtuais e os componentes físicos.

A utilização de um painel de instrumentos real proporciona um ganho significativo de realismo, uma vez que o usuário interage com os mesmos elementos que encontraria em um veículo de verdade. A proposta de utilizar o protocolo *CAN-Bus* (*Controller Area Network bus*) é uma solução robusta e amplamente utilizada na comunicação automotiva.

### 1.1 OBJETIVOS

Os objetivos do presente estudo são:

1. Oferecer uma solução de *hardware* e *software* que possibilite integrar um painel de instrumentos automotivo real com simuladores veiculares digitais;
2. Apresentar uma interface de comunicação com o uso do *software* de código aberto SimHub e um microcontrolador para converter e transmitir os dados de telemetria do simulador para o painel de instrumentos;
3. Possibilitar a comunicação de dados entre o microcontrolador com módulo de interface *CAN-Bus* MCP2515, e o painel de instrumentos físico;
4. Demonstrar a solução proposta, exibindo dinamicamente no painel físico, informações do veículo simulado (RPM, velocidade, nível de combustível, setas, etc.);
5. Observar como a solução apresentada auxilia para o aumento da imersão e do realismo sentido em simuladores automotivos.



## 2 REFERENCIAL TEÓRICO

Por meio de pesquisas para o presente trabalho, foi possível perceber, a ausência de trabalhos que abordem o tema em questão, sendo evidenciada a necessidade de mais estudos nessa área. Então, a pesquisa foi focada em tecnologias correlacionadas, evidenciando o protocolo de comunicação *CAN-Bus*.

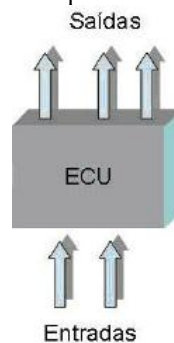
Conforme Bogoni (2009), os desenvolvedores de simuladores veiculares, seguem dois caminhos, ambos focados no realismo, simuladores com propósitos comerciais e simuladores para pesquisa e desenvolvimento de funções específicas. Os simuladores de propósito comercial normalmente são focados em entretenimento, educação e terapia. Enquanto que os simuladores destinados à pesquisa e desenvolvimento, focam na coleta de dados e controle das ações do condutor.

Esta pesquisa concentrou-se em simuladores comerciais por sua maior acessibilidade e por dispensarem o uso de *hardwares* específicos. No entanto, o modelo proposto pode ser alterado e adaptado aos requisitos de simuladores voltados à pesquisa e desenvolvimento (P&D).

Conforme exposto por Guimarães e Saraiva (2003), os dois principais conceitos de Arquiteturas Eletroeletrônicas para a utilização do protocolo CAN bus são as arquiteturas **centralizada** e **distribuída**.

Enquanto a Arquitetura Distribuída utiliza diversas ECUs (*Electronic Control Units*) em um mesmo sistema para o tráfego de sinais, este experimento simula apenas parte de um sistema automotivo. Assim, fará uso da **Arquitetura Centralizada**, que consiste em uma única ECU responsável por receber os dados de telemetria do veículo virtual, processá-los e transmiti-los ao painel de instrumentos.

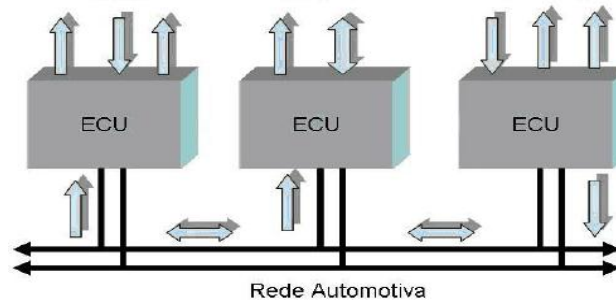
Figura 1 - Arquitetura Centralizada



Fonte: Pereira e Zeferino (2008).



Figura 2 - Arquitetura Distribuída



Fonte: Pereira e Zeferino (2008).

Neste trabalho, o papel da ECU é desempenhado pela integração entre o *software* SimHub e o Arduino. O SimHub coleta os dados dos sensores do veículo virtual e os transmite ao Arduino que, equipado com o módulo MCP2515, processa e converte essas informações para o barramento CAN no formato interpretado pelo painel de instrumentos.

### 3 METODOLOGIA

A metodologia deste trabalho seguiu uma abordagem experimental e tecnológica, dividida em quatro etapas fundamentais: identificação física (*pinout*), engenharia reversa do protocolo CAN, montagem do *hardware* e integração de *software*.

#### 3.1 IDENTIFICAÇÃO DE PINAGEM E INTERFACE FÍSICA

Inicialmente, realizou-se o levantamento técnico do *pinout* (diagrama de pinagem) dos painéis de instrumentos dos veículos Peugeot modelos 408 e 3008. Esta etapa foi crucial para a alimentação elétrica segura dos componentes e a correta identificação dos terminais de comunicação CAN High (CAN-H) e CAN Low (CAN-L). Para alguns modelos de painéis, esta pinagem pode ser encontrada através de pesquisas na internet em sites de manutenção, e em manuais de reparos automotivos.

#### 3.2 METODOLOGIA DE MAPEAMENTO FUNCIONAL E VALIDAÇÃO DE IDS

O mapeamento dos Identificadores (IDs) de interesse foi realizado em duas fases distintas para garantir o rigor do processo de engenharia reversa.

##### 3.2.1 Descoberta Inicial de IDs (Varredura Programada)

Dada a natureza proprietária dos pacotes de dados automotivos, aplicou-se a técnica de engenharia reversa por **Injeção de Sinais**. Utilizou-se um algoritmo desenvolvido em ambiente Arduino IDE para o envio de comandos hexadecimais variáveis ao barramento.



- Implementação: O microcontrolador, acoplado ao módulo MCP2515, realizou a varredura de um intervalo pré-determinado de IDs, transmitindo quadros com 8 octetos de dados preenchidos com valores conhecidos (ex: todos zeros ou FF FF FF FF FF FF FF FF);
- Procedimento: Observou-se o comportamento físico do painel simultaneamente à análise do Monitor Serial do Arduino;
- Aferição: A cada injeção de ID, a reação do painel (e.g., acendimento de uma luz indicadora, movimento de um ponteiro) era monitorada visualmente. O ID responsável pela reação observada era registrado via monitor serial do Arduino.

### 3.2.2 Validação de Ids (CanHacker)

Após a obtenção da lista de IDs, a ferramenta CanHacker foi empregada como a principal plataforma de validação, oferecendo maior controle e precisão na manipulação dos dados. O uso da ferramenta CanHacker nesta fase mostrou-se fundamental para a conversão do ID ativo em um ID funcional mapeado.

- Configuração: O CanHacker foi configurado para o mesmo *bitrate* de comunicação do painel de instrumentos em uso. A interface de comunicação serial do microcontrolador Arduino foi conectada ao computador e a do controlador CAN ao barramento do painel.
- Teste de Injeção de Dados: Cada ID da lista inicial foi testado individualmente através da função de injeção manual (*Transmit*) do CanHacker. A validação funcional foi realizada através da manipulação dos octetos do campo de dados.

Exemplo (Velocidade): Para validar o ID de Velocidade, os 3º e 4º octetos foram manipulados com valores hexadecimais crescentes, desde 00 00 até valores altos (ex: FF FF). A leitura e o envio desses valores diretamente no formato hexadecimal (conforme discutido na seção 3.2.1) permitiu determinar o fator de escala utilizado pela montadora Peugeot Soci t  Anonyme (PSA) e, conseqentemente, o *payload* exato que corresponde a uma velocidade especifica.

Esta metodologia combinada (varredura r pida seguida por inje o controlada e leitura em hexadecimal) garantiu um mapeamento robusto e validado dos IDs, que foram ent o utilizados para a comunica o final com o SimHub.

Embora parte dos Identificadores (IDs) tenha sido obtida atrav s da t cnica de varredura e inje o de sinais, utilizou-se a documenta o t cnica do projeto de c digo aberto PSA CAN (AUTOWP, 2025) como base de compara o. A consulta a essa base de dados permitiu n o apenas confirmar a fun o de identificadores j  mapeados, mas tamb m corrigir a interpreta o de determinados *payloads* e identificar IDs de interesse que ainda n o haviam sido mapeados durante as varreduras iniciais.



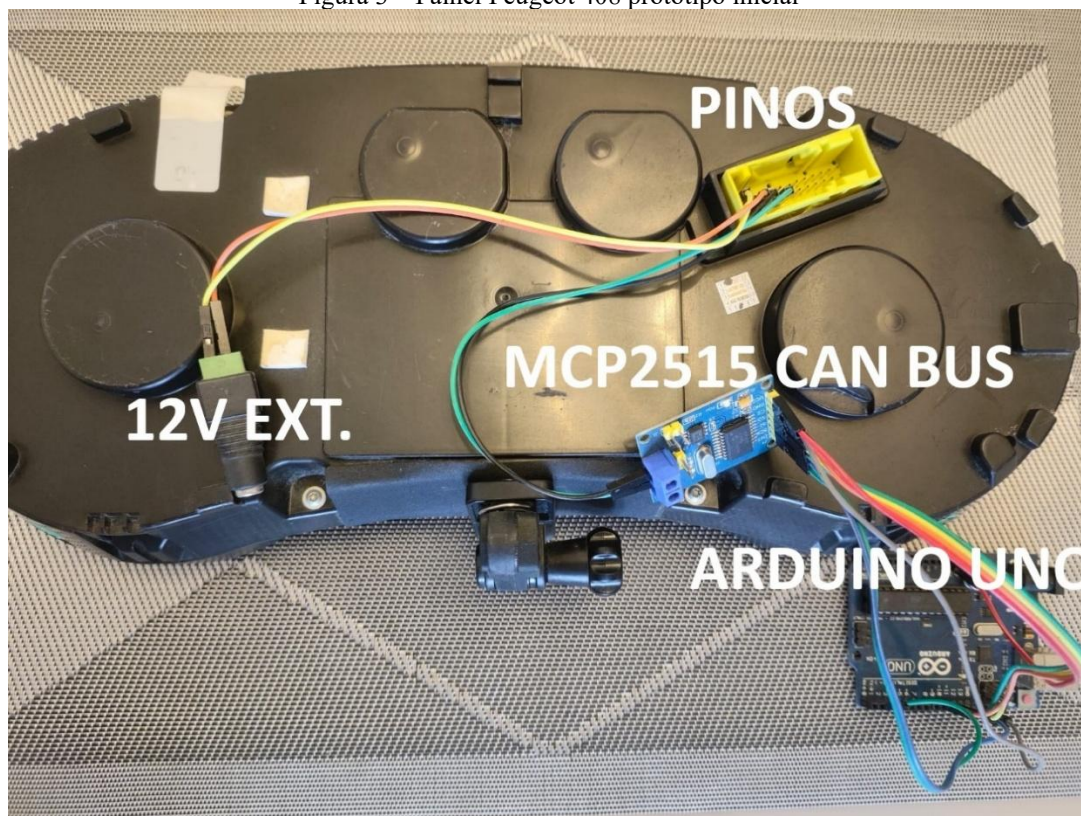
Entretanto, observou-se durante os testes que apenas uma parcela dos Identificadores listados na documentação era idêntica aos sinais operacionais dos painéis físicos utilizados neste estudo Peugeot 408 e 3008. Essas divergências evidenciaram que, mesmo dentro de uma mesma fabricante ou grupo (como a PSA), existem variações significativas de protocolos dependendo do modelo, ano de fabricação e versão do *firmware* da ECU (Unidade de Controle Eletrônico).

### 3.3 ARQUITETURA DO SISTEMA E *HARDWARE*

O sistema foi implementado utilizando uma arquitetura centralizada. A interface física entre o computador e o barramento de dados dos painéis foi composta por:

- **Microcontrolador Arduino:** Responsável pelo processamento lógico e execução do algoritmo de comunicação.
- **Módulo MCP2515:** Controlador CAN que permite ao Arduino interpretar e transmitir sinais físicos compatíveis com o padrão automotivo ISO 11898.

Figura 3 – Painel Peugeot 408 protótipo inicial



Fonte: Elaborado pelos autores (2025).

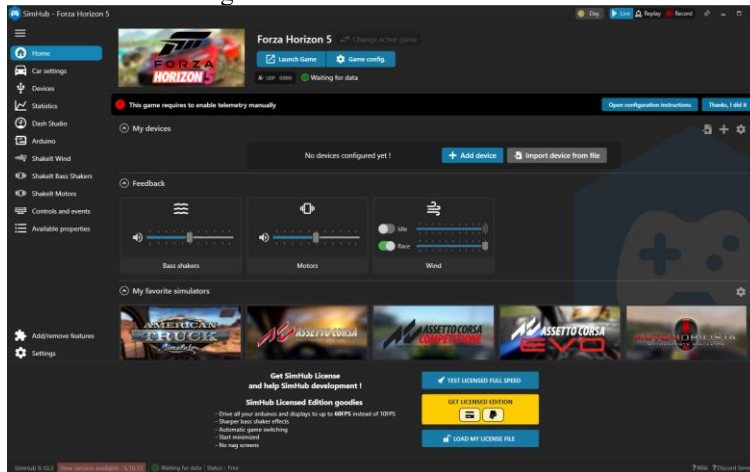
### 3.4 INTEGRAÇÃO VIA *SOFTWARE* (SIMHUB)

Para a extração dos dados telemétricos dos simuladores de corrida, utilizou-se o *software SimHub*. A integração seguiu o fluxo:

1. Captura dos dados brutos do jogo (SDK/Telemetria);

2. Conversão dos valores reais para o formato hexadecimal mapeado na etapa 3.2;
3. Envio via interface Serial para o Arduino, que injeta os pacotes no barramento CAN em tempo real.

Figura 4 - Dashboard do SIMHUB



Fonte: Elaborado pelos autores (2025).

## 4 RESULTADOS E DISCUSSÕES

### 4.1 MAPEAMENTO E DECODIFICAÇÃO DE MENSAGENS CAN

O sucesso do projeto residiu na engenharia reversa e no mapeamento preciso dos **Identificadores (IDs)** do barramento CAN do painel veicular utilizado.

- **Resultados de Mapeamento:** Identificadores específicos foram correlacionados com dados de telemetria do jogo (e.g., RPM, Velocidade, Nível de Combustível). Por exemplo, o valor hexadecimal 0B6 foi identificado como o ID responsável por atualizar o ponteiro de RPM e velocímetro;
- **Decodificação Hexadecimal:** A telemetria, gerada pelo SimHub e processada pelo Arduino, foi convertida para o **formato hexadecimal** de 8 octetos (*bytes*) antes de ser enviada. Este formato permite a comunicação eficiente e concisa com o painel.
  - **Exemplo Prático:** Para simular uma velocidade de **60 km/h**, o Arduino calculou o *payload* (campo de dados) de 8 *bytes*, que foi transmitido como uma sequência hexadecimal (ex: 00 00 00 3C 00 00 00 00). O painel decodifica o valor 3C como 60 km/h (onde 3C hex = 60 dec).

### 4.2 EFICIÊNCIA E SINCRONIZAÇÃO DA REDE CAN

O uso do protocolo CAN, implementado via módulo **MCP2515**, demonstrou sua eficácia em ambientes de simulação em tempo real, atingindo a latência necessária para uma experiência imersiva.

- **Comprovação da Prioridade (Arbitragem):** A rede CAN, mesmo com múltiplos dados (RPM, velocidade, luzes, etc.) sendo enviados simultaneamente, manteve a integridade e a prioridade das mensagens. Como o ID é o campo de prioridade, a mensagem de RPM e



velocidade (geralmente com o ID de menor valor e, portanto, **maior prioridade**) sempre dominou o barramento, garantindo que a atualização da rotação fosse precisa e sem *lag*;

- **Taxa de Transmissão:** O sistema operou com sucesso a uma taxa de *baud* (velocidade) de 125 kbps, permitindo que o painel atualizasse os dados em menos de 10 ms, mimetizando o comportamento de uma ECU veicular real.

#### 4.3 VIABILIDADE DO *HARDWARE* DE BAIXO CUSTO

O projeto validou a utilização de componentes de baixo custo para a simulação de sistemas embarcados complexos.

Quadro 1 – Relação de componentes e viabilidade técnica

Componente	Função no Projeto	Resultado da Viabilidade
Arduino Uno	Processamento da lógica SimHub -> CAN	Desempenho satisfatório na conversão de telemetria para payloads CAN.
MCP2515	Interface CAN	Sucesso na modulação e transmissão dos sinais.
Painel de instrumentos	Display	Confirmação de que o painel reconhece os payloads como se viessem de uma ECU real, provando a eficácia da engenharia reversa.

Fonte: Elaborado pelos autores (2025).

Figura 5 - Painel Peugeot 3008 protótipo final



Fonte: Elaborado pelos autores (2025).

#### 4.4 VALIDAÇÃO QUALITATIVA E VERSATILIDADE ENTRE TÍTULOS

A solução proposta foi submetida a testes práticos com um grupo de motoristas e entusiastas, utilizando uma ampla gama de *softwares* de simulação. Essa diversidade foi essencial para validar a versatilidade da interface Arduino-SimHub em diferentes motores de física e protocolos de telemetria.

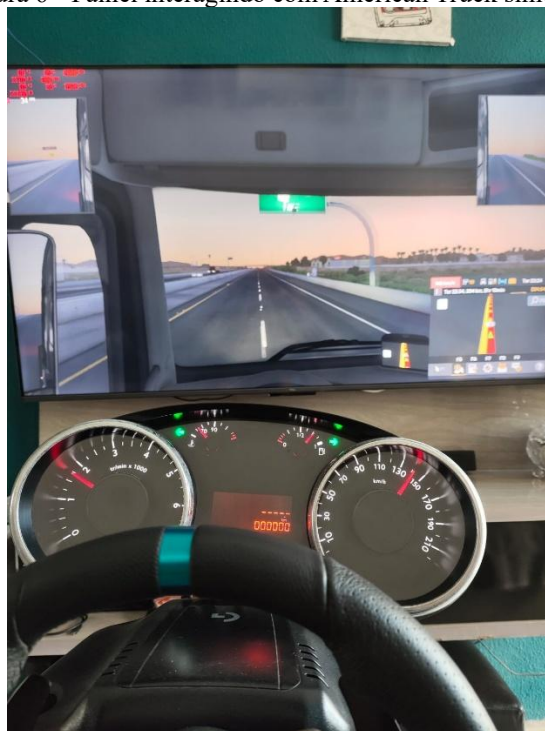
- **Simuladores de Alta Fidelidade (*Hard Sims*):** Em títulos como *Assetto Corsa Competizione (ACC)*, *Automobilista 2* e o novo *Assetto Corsa Evo*, o foco dos testes foi a precisão do RPM e velocímetro. A resposta imediata do painel físico foi considerada crucial para a performance do condutor;



- **Simuladores de Carreira e Velocidade:** Títulos como *F1 22* e a franquia *Forza (Motorsport e Horizon)* serviram para observar o comportamento do *hardware* em cenários de alta velocidade e constantes variações de telemetria, onde o sistema manteve a integridade dos dados sem travamentos ou atrasos perceptíveis;
- **Simulação de Transporte (ATS):** O *American Truck Simulator* foi o título com maior volume de testes, contando com a avaliação de motoristas reais. Neste cenário, a imersão foi potencializada pelo uso de indicadores de longo curso, como o nível de combustível, a temperatura do óleo e o funcionamento das luzes de direção (setas).

O feedback unânime dos testadores destacou que o uso do painel real elimina a necessidade de interfaces virtuais (*HUD*) na tela, permitindo uma condução mais instintiva. A transição da informação do ambiente virtual para um componente mecânico e tátil foi apontada como o principal fator de ganho no realismo, confirmando que a solução atende tanto ao público de simuladores profissionais quanto ao mercado de entretenimento de alta performance.

Figura 6 - Painel interagindo com American Truck simulator



Fonte: Elaborado pelos autores (2025).



Por fim, este artigo abre caminho para pesquisas futuras que explorem a integração de outros componentes do veículo, como também a implantação de uma arquitetura distribuída com outras unidades de controle (ECUs), bem como a aplicação deste modelo em simuladores voltados à reabilitação de motoristas ou ao desenvolvimento de interfaces homem-máquina (HMI) na indústria automotiva.

## AGRADECIMENTOS

Gostaríamos de expressar nossa gratidão a todos que, de forma direta ou indireta, contribuíram para a realização deste projeto.

Aos motoristas e entusiastas que dedicaram seu tempo para testar o protótipo e fornecer feedbacks valiosos. Suas impressões reais foram o que transformaram este projeto técnico em uma ferramenta de imersão comprovada.

**Individualmente, os autores gostariam de agradecer:**

**Clebson Queiroz:** À minha esposa, meu maior pilar de sustentação, pelo incentivo constante e pela paciência com as horas dedicadas a este estudo. À minha mãe e familiares, pelo apoio incondicional em toda a minha trajetória.

**André Queiroz:** Agradeço aos meus pais e amigos, pelo apoio fundamental. Ao meu irmão, agradeço por ser uma inspiração inicial que me guiou ao mundo da tecnologia. Sem o suporte de vocês, esta conquista não seria possível.

**Francisco Abud:** À Instituição e aos colegas de colegiado pelo suporte técnico e acadêmico, e à minha família pela compreensão e incentivo constante à pesquisa científica.



## REFERÊNCIAS

ARDUINO. Arduino IDE e microcontroladores. Disponível em: <https://www.arduino.cc/>. Acesso em: 25 jul. 2025.

AUTOWP. PSA CAN: PSA CAN bus documentation. Disponível em: <https://autowp.github.io/>. Acesso em: 07 ago. 2025.

BOGONI, Tales Nereu. Desenvolvimento de um simulador para monitoramento de técnicas de direção econômica em caminhões com o uso de ambientes virtuais. 2009. Dissertação (Mestrado) – PUC-RS, Porto Alegre, 2009. Disponível em: <https://repositorio.pucrs.br/dspace/bitstream/10923/1577/1/000417204-Texto%2BCompleto-0.pdf>. Acesso em: 25 jul. 2025.

GUIMARÃES, Alexandre de Almeida; SARAIVA, Antônio Mauro. Um roteiro de implementação de uma rede CAN (Controller Area Network). In: XI SIMPÓSIO INTERNACIONAL DE ENGENHARIA AUTOMOTIVA (SIMEA), 2003, São Paulo. Anais... São Paulo: Blucher, 2003. Disponível em: <https://www.alexag.com.br/Artigos/SIMEA2003.pdf>. Acesso em: 30 jul. 2025.

PEREIRA, Rodrigo V. M.; ZEFERINO, Cesar A. Aplicação do protocolo LIN como sub-rede CAN. In: VIII CONFERÊNCIA INTERNACIONAL DE APLICAÇÕES INDUSTRIAIS (INDUSCON), 2008, Poços de Caldas. Anais... Poços de Caldas: IEEE, 2008. Disponível em: [https://www.researchgate.net/publication/267372136\\_APLICACAO\\_DO\\_PROTOCOLO\\_LIN\\_CO\\_MO\\_SUB-REDE\\_CAN](https://www.researchgate.net/publication/267372136_APLICACAO_DO_PROTOCOLO_LIN_CO_MO_SUB-REDE_CAN). Acesso em: 10 jul. 2025.

SIMHUB. SimHub - DIY Device Driver for Sim Racers. Disponível em: <https://www.simhubdash.com/>. Acesso em: 25 jul. 2025.

