



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

**Relatório de Atividade de Laboratório 1: Setup Experimental de Redes Intra
veiculares Baseadas no Protocolo CAN**

Danilo Augusto Barbosa Nogueira (dabn)

Paulo Sérgio Galdino de Souza (psgs)

IF747 – Redes Automotivas

Recife,

2025

1. Introdução e Objetivos

Este relatório descreve o processo de desenvolvimento de um setup experimental para comunicação em redes intraveiculares, conforme especificado na "Atividade de Laboratório 1". O objetivo geral do projeto foi desenvolver um ambiente funcional composto por um barramento CAN físico e três Unidades de Controle Eletrônico (ECUs) simuladas, capazes de trocar mensagens entre si.

Os objetivos específicos alcançados foram:

- Montagem de um barramento CAN físico interligando três dispositivos;
- Projeto das mensagens CAN a serem trocadas entre as ECUs;
- Implementação das lógicas de transmissão e recepção de mensagens para cada ECU utilizando o software Vehicle Spy 3;
- Criação de dashboards (Painéis Gráficos) para a visualização dos sinais e do estado de cada ECU;

2. Materiais e Métodos

Para a execução da atividade, foram utilizados os seguintes materiais e ferramentas:

- **Hardware:** Três dispositivos com interface CAN (ValueCAN4-2) atuando como ECUs.
- **Software:** Vehicle Spy 3.

A metodologia consistiu em configurar três projetos distintos no Vehicle Spy, cada um representando uma ECU (Chassis, Body e PowerTrain), conforme a matriz de comunicação de referência. Cada instância do software foi associada a um dispositivo ValueCAN4-2 exclusivo, permitindo a simulação e monitoramento em um barramento CAN real.

- **ECU Chassis:** Associada ao hardware ValueCAN4-2 V2C29, como visto na fig. 1.
- **ECU Body:** Associada ao hardware ValueCAN4-2 V2C301, como visto na fig. 2.
- **ECU PowerTrain:** Associada ao hardware ValueCAN4-2 V2C305, , como visto na fig. 3.

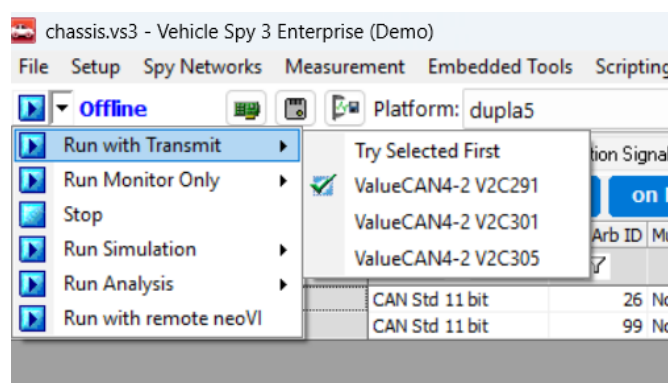


Fig. 1: ECU relacionado ao Chassi

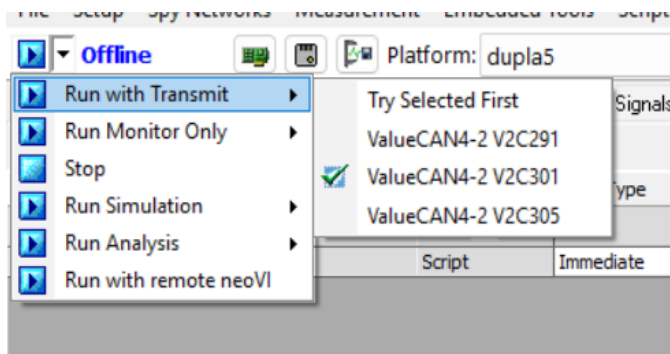


Fig. 2: ECU relacionado ao Body

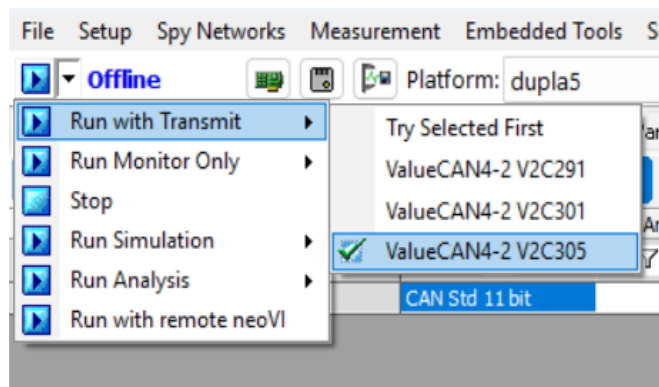


Fig. 3: ECU relacionado ao PowerTrain

3. Desenvolvimento e Implementação

3.1 Projeto da rede e mensagens

A implementação foi dividida em três partes principais: o projeto das mensagens, a implementação da lógica funcional e a criação dos painéis de visualização.

Foram definidas duas mensagens principais na rede "DW CAN 01" para coordenar as funções do sistema:

- **Brake_Operation_Indicator:**
 - ID de Arbitragem: 26.
 - DLC: 2 bytes.
 - Fonte: ECU Chassis.
 - Descrição: Carrega um sinal analógico que indica a intensidade de acionamento do freio.
 - Visto na fig. 4;

- **Belt_sensor:**
 - ID de Arbitragem: 99.
 - DLC: 1 byte.
 - Fonte: ECU Chassis.
 - Descrição: Contém sinais digitais que informam o estado do cinto de segurança e a ocupação do assento.
 - Visto na fig.5;

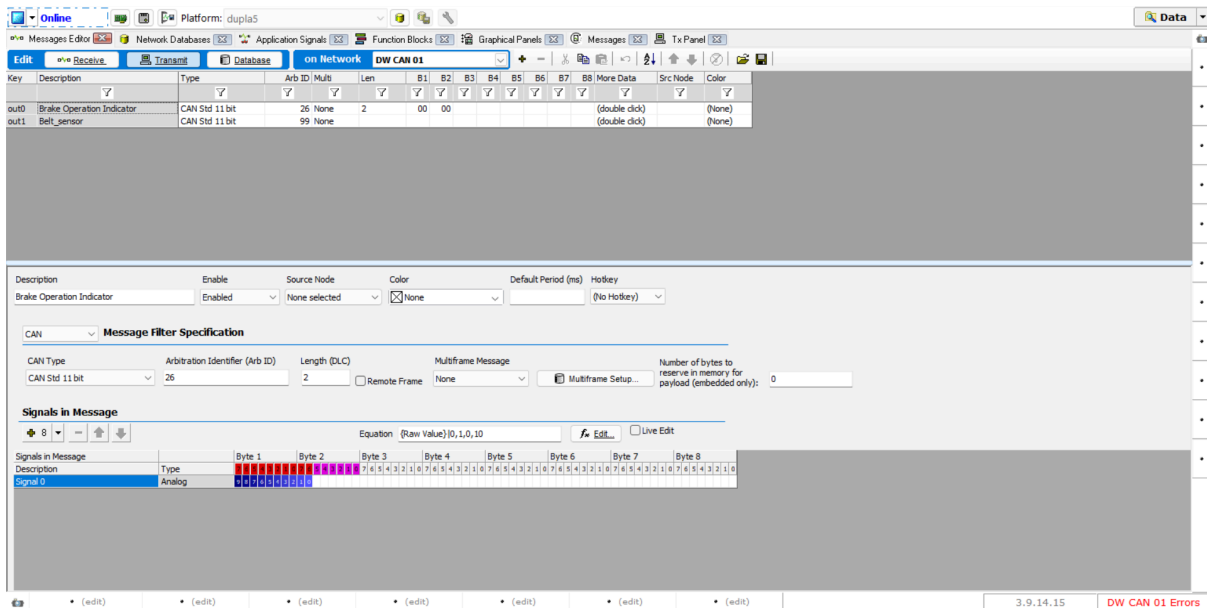


Fig.4: Mensagens transmitidas pela ECU Chassi

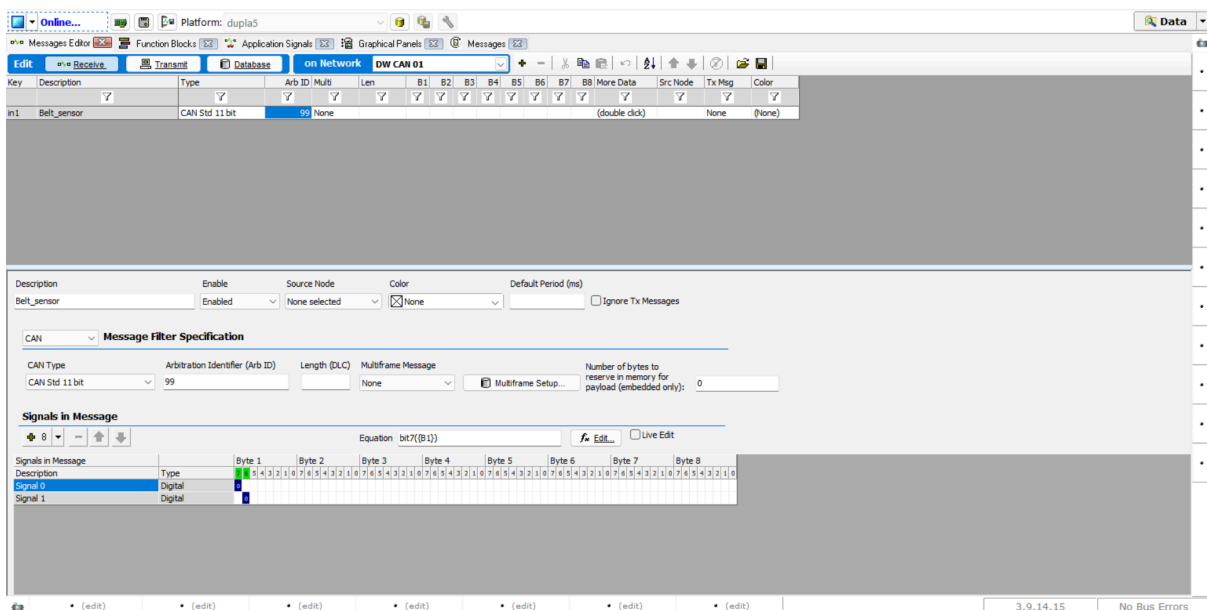


Fig.5: Mensagem recebida pela ECU Body

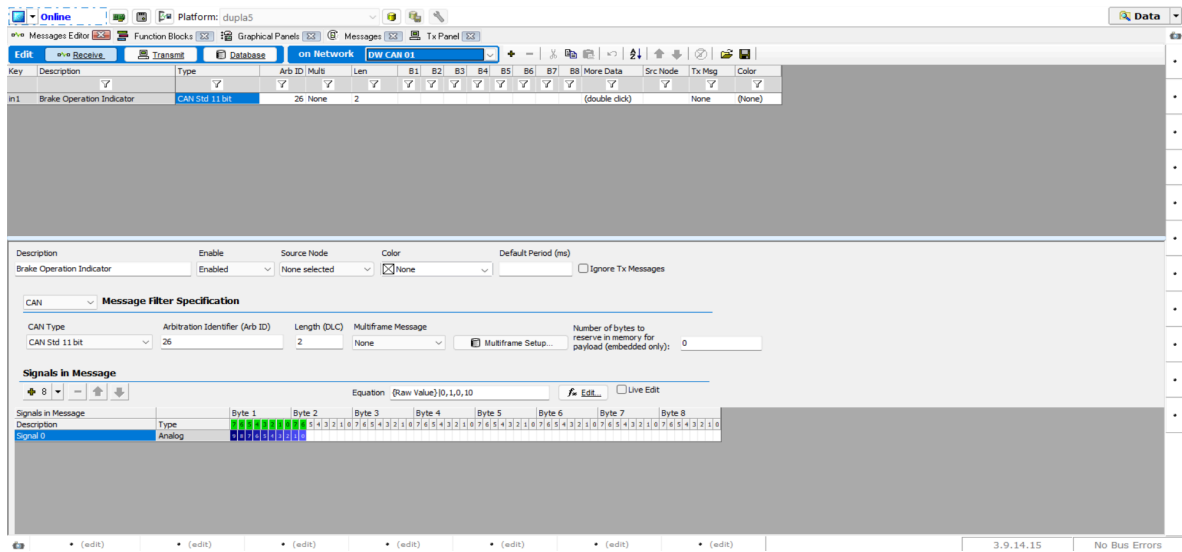


Fig.6: Mensagem recebida pela ECU PowerTrain

A ECU Body foi configurada para receber a mensagem **Belt_sensor**, partindo do Chassi e a ECU PowerTrain, vista na fig.6, foi configurada para receber a mensagem **Brake_Operation_Indicator**, também sendo transmitida pela ECU Chassi.

3.2 Lógica de funcionamento

A lógica principal do sistema foi implementada na ECU Chassis através de Function Blocks, visto na fig. 7.

- Definição de Sinais de Aplicação:** Foram criados sinais internos na ECU Chassis para gerenciar o estado do sistema, como **AssentoOcupado**, **CintoAfivelado** e **AVISO_LED**, vistos na figura 8.
- Lógica do Alarme de Cinto:** Foi criado um script (**Logica_Alarme_cinto**) que executa a seguinte verificação:
 - Condição: **SE** o sinal **ASSENTOOCUPADO** estiver ativo **E** o sinal **CINTOAFIVELADO** não estiver ativo.
 - Ação: O sinal **AVISO_LED** é definido como **1** (ligado).
 - Caso contrário: O sinal **AVISO_LED** é definido como **0** (desligado).
- Transmissão de Mensagens:** Após a execução da lógica, o script popula os sinais da mensagem de saída **Belt_sensor** e a transmite na rede CAN a cada 500 ms.

Na ECU Body, um Function Block foi utilizado para mapear os sinais recebidos na mensagem **Belt_sensor** para os sinais de aplicação locais, que controlam o dashboard.

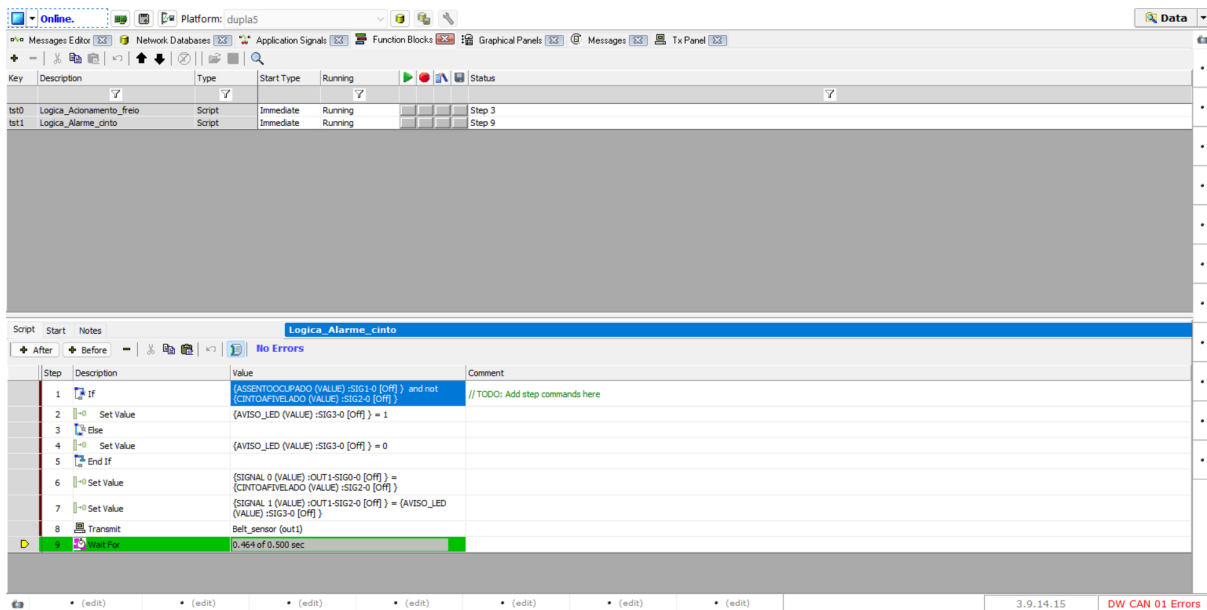


Fig.7: Function Blocks relacionadas ao Chassi



Fig. 8: Sinais criados na ECU Chassi

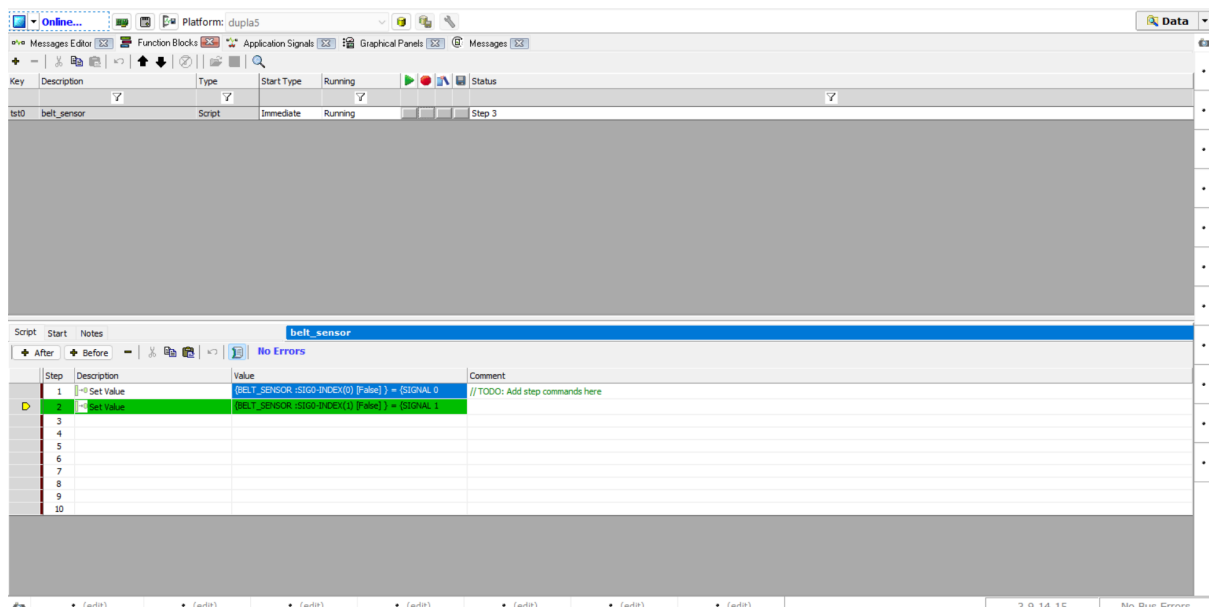


Fig. 9: Function Blocks relacionadas ao Body

3.3 Dashboards (Painéis Gráficos)

Para cada ECU, foi desenvolvido um painel gráfico customizado para interação e visualização dos dados da rede, conforme o objetivo da atividade.

- **Painel da ECU Chassis:** Visto na figura 10, contém botões para simular os inputs "Assento ocupado" e "Sem cinto", um medidor analógico para setar a intensidade do freio e um indicador visual (LED vermelho) que representa o **AVISO_LED** do alarme.
- **Painel da ECU Body:** Visto na figura 11, exibe o estado recebido do sensor de cinto e do assento através de dois indicadores visuais (LED): "Cinto" e "Assento ocupado".
- **Painel da ECU PowerTrain:** Visto na figura 12, apresenta um indicador de "Acionamento do freio" (LED) e um medidor de intensidade que exibe o valor analógico do sinal de freio, variando de 0 a 1023.

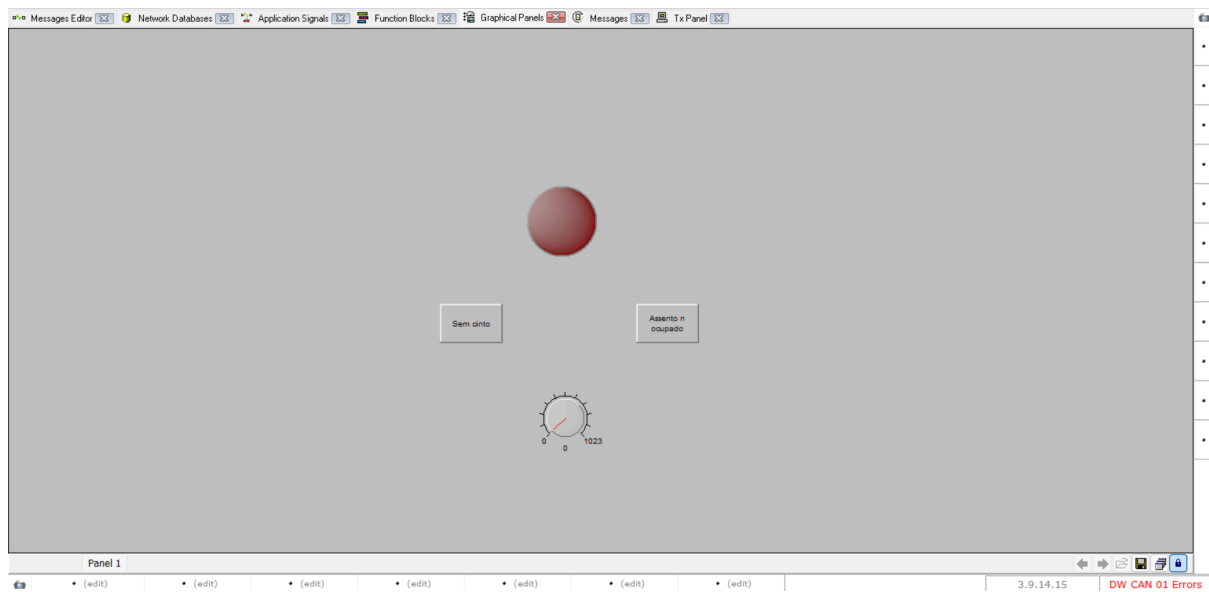


Fig. 10: Graphical pannel da ECU Chassi

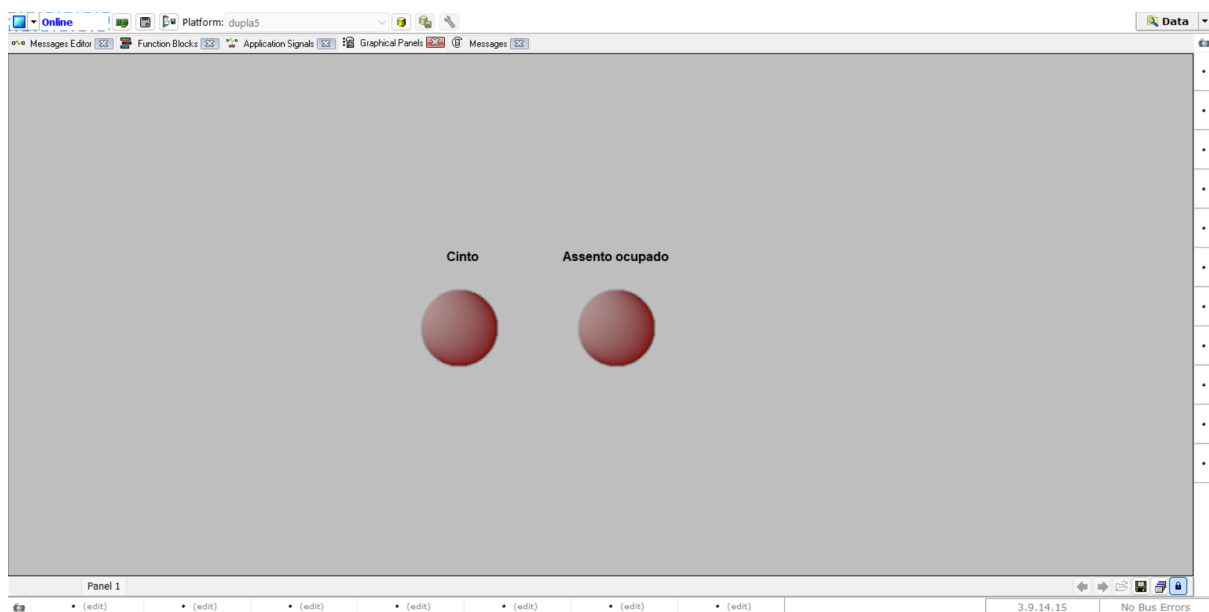


Fig. 11: Graphical pannel da ECU Body

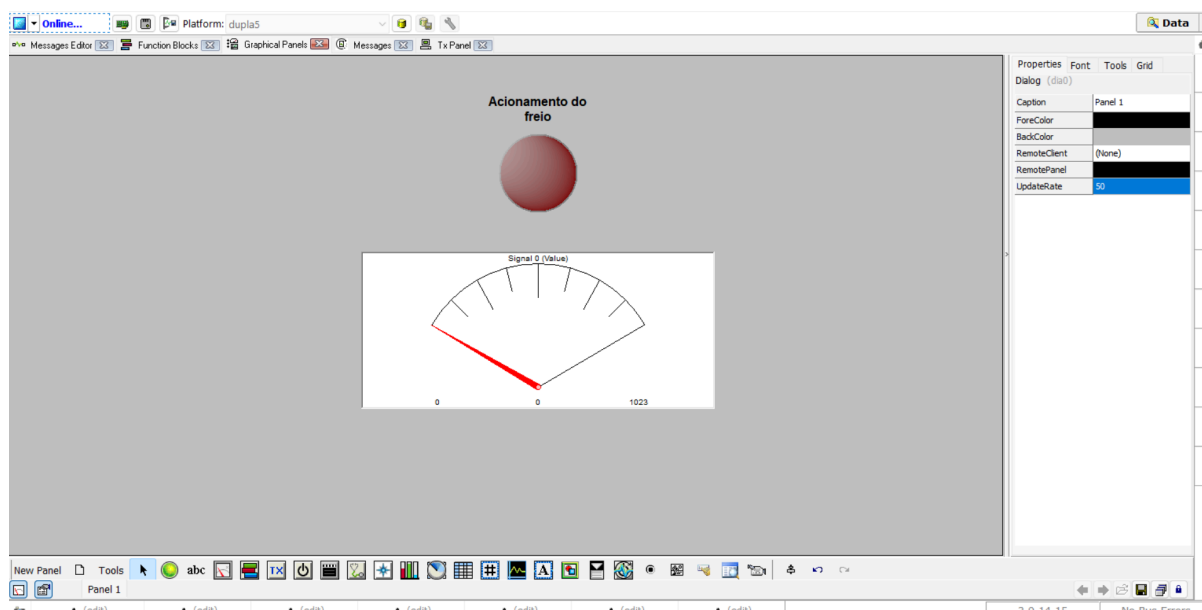


Fig. 12: Graphical pannel da ECU PowerTrain

4. Resultados

A comunicação entre as três ECUs foi estabelecida com sucesso. As janelas de monitoramento de mensagens (DataSpy) de cada ECU confirmam o tráfego das mensagens **Belt_sensor** (ID 99) e **Brake_Operation_Indicator** (ID 26) no barramento CAN, que se comprova pelas figuras 13, 14 e 15, sendo estas relacionadas ao Chassi, Body e Powertrain, respectivamente.

A lógica implementada na ECU Chassis operou como esperado: ao simular um cenário de "assento ocupado" sem o cinto afivelado através do painel gráfico, a mensagem **Belt_sensor** era transmitida, e o LED de aviso no painel da própria ECU Chassis era acionado. Consequentemente, a ECU Body recebia a mensagem e atualizava seu dashboard para refletir o estado. A comunicação da ECU PowerTrain também foi validada pela visualização da mensagem de freio em seu painel.

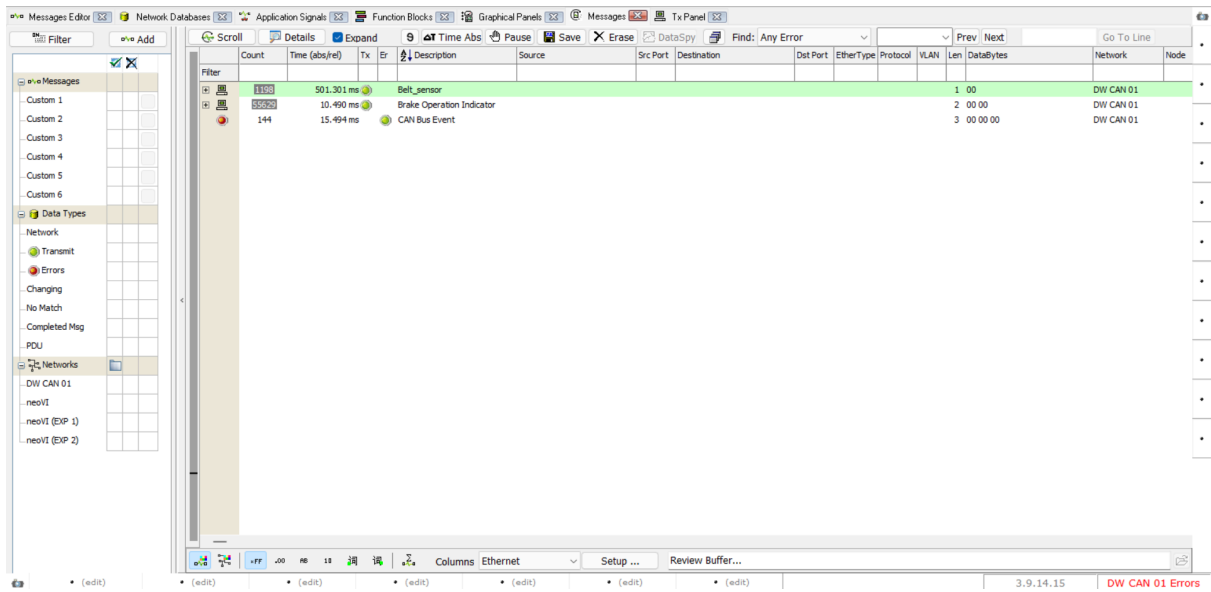


Fig. 13: Monitoramento de mensagens da ECU relacionada ao Chassi

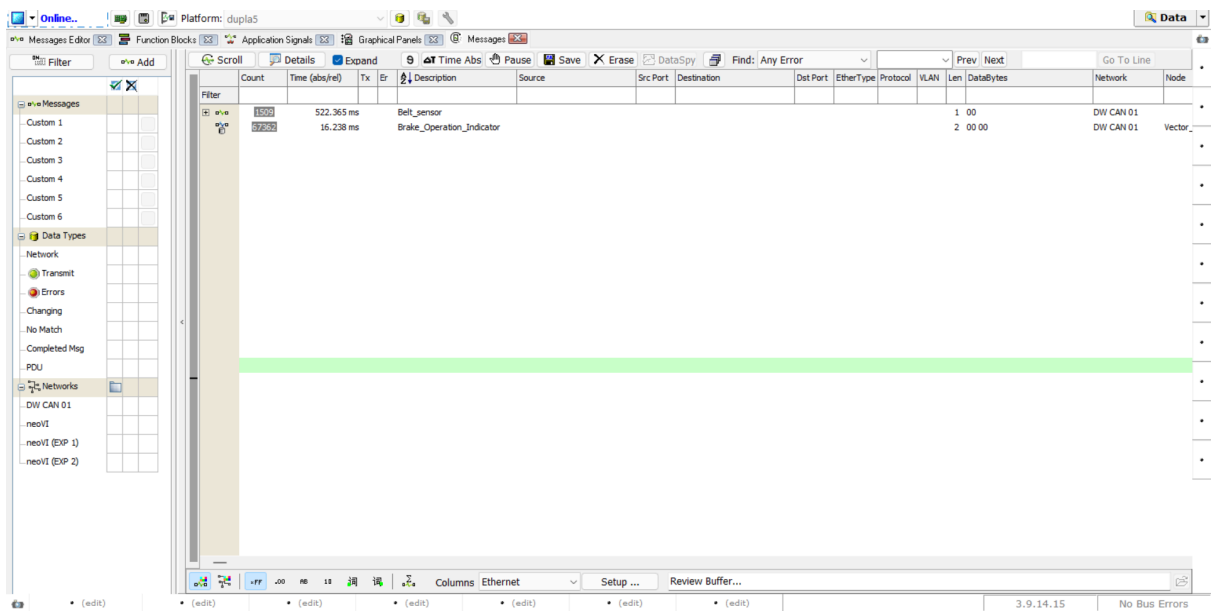


Fig. 14: Monitoramento de mensagens da ECU relacionada ao Body.

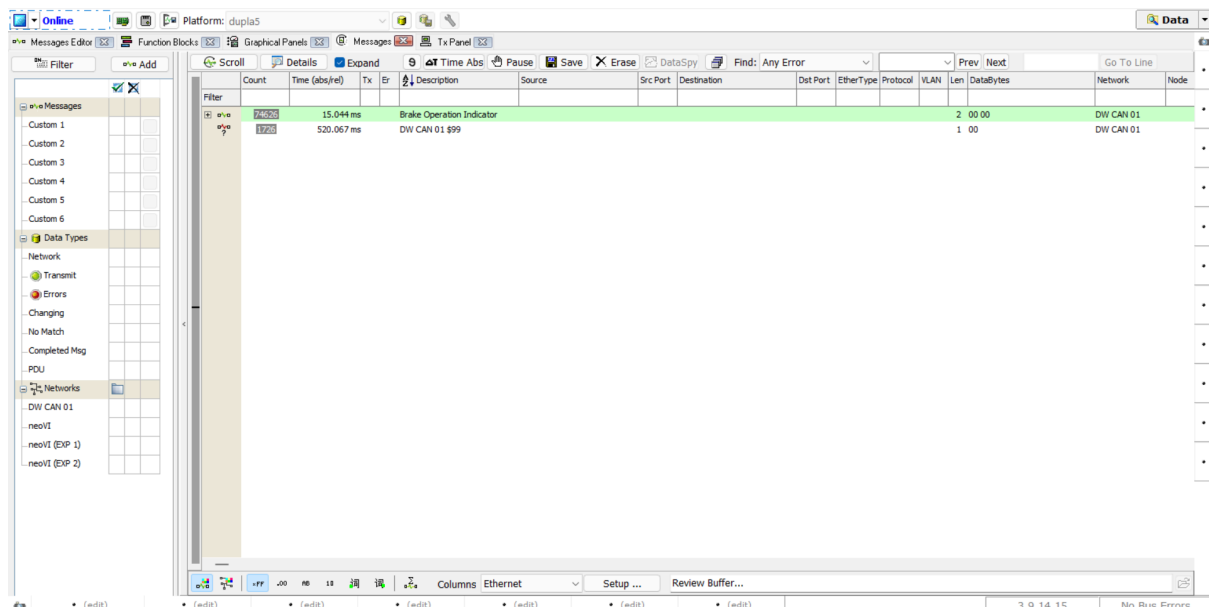


Fig. 15: Monitoramento de mensagens da ECU relacionada ao PowerTrain

5. Conclusão

A atividade de laboratório foi concluída com êxito, atingindo todos os objetivos propostos. Foi possível projetar e implementar um sistema de comunicação CAN funcional entre três ECUs simuladas, utilizando hardware e software padrão da indústria automotiva. O desenvolvimento abrangeu desde a definição de mensagens e sinais até a criação de lógicas de aplicação com Function Blocks e interfaces de visualização com Graphical Panels, proporcionando uma compreensão prática e aprofundada sobre a arquitetura de redes veiculares.

6. Referências bibliográficas

- [1] PASTA1.0 CAN-ID List v1.0E. Disponível em: <https://github.com/pasta-auto/PASTA1.0/blob/master/doc/PASTA1.0%20CAN-ID%20List%20v1.0E.pdf>. Acesso em: 02 de Julho de 2025.
- [2] Vspy 3 Documentation. Disponível em: <https://docs.intrepidcs.com/vspy-3-documentation>. Acesso em: 02 de Julho de 2025.