

Implementação e Análise de Rede CAN, Parte 1

Gabriel Soares¹, Maria Eduarda Melo¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

{gss12, meom}@cin.ufpe.br

Resumo. Este relatório detalha a implementação e análise de um projeto de rede CAN, envolvendo a comunicação entre três Unidades de Controle Eletrônico (ECUs). Utilizando o software Vehicle Spy 3, exploramos as dinâmicas de comunicação nessa rede, destacando os desafios enfrentados, soluções aplicadas e os insights adquiridos durante o processo. Os resultados obtidos fornecem uma visão abrangente das interações entre as ECUs.

1. Introdução

A implementação e análise de redes Controller Area Network (CAN) desempenham um papel crucial nos sistemas automotivos modernos, permitindo a comunicação eficiente entre Unidades de Controle Eletrônico (ECUs) dentro dos veículos. Neste relatório, exploramos um projeto focado em estabelecer a comunicação entre três ECUs utilizando o software Vehicle Spy 3. Vale ressaltar que para as ECUs, utilizamos o ValueCAN4 e o NeoVi Fire 2. À medida que a indústria automotiva continua a evoluir, entender as complexidades das redes CAN torna-se cada vez mais vital para garantir a confiabilidade e o desempenho dos sistemas de veículos.

2. Estrutura do Barramento

A montagem da rede de comunicação CAN para simulação das trocas de mensagens entre componentes de uma rede automotiva deu-se através da conexão de jumpers diversos a uma protoboard para interligar os dispositivos ValueCAN4 ou NeoVi Fire 2 em série, junto com resistores de 120 ohms. As partes USB foram conectadas a um computador para fornecer alimentação e permitir a troca de dados para configuração das mensagens e lógica.

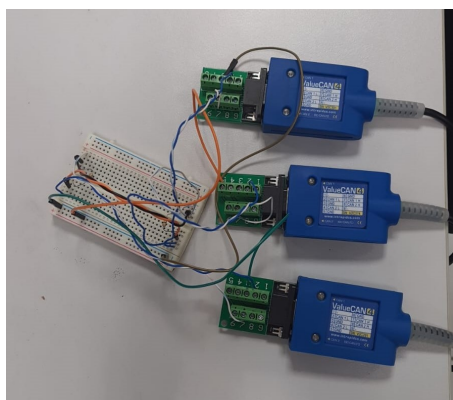


Figura 1. Esquemático do Barramento CAN

3. Projeto da Troca de Mensagens

Para simplificar a especificação das mensagens e a reutilização em diferentes arquivos do Vehicle Spy, foi criado um arquivo .dbc com auxílio das referências [4] e [5].

```
BO_12 velocidadeMotor: 16 Vector_XXX
SG_Velocidade_do_motor_signal : 31|16@1+ (0.25,0) [0|16384] "rpm" Vector_XXX

BO_13 temperaturaRefrigeradorMotor: 8 Vector_XXX
SG_Temperatura_refrigerador_Motor_s : 31|8@1+ (1,-40) [-40|215] "degC" Vector_XXX

BO_18 velocidadeCarro: 8 Vector_XXX
SG_Velocidade_Carro_signal : 31|8@1+ (1,0) [0|255] "km/h" Vector_XXX

BO_14 odometro: 32 Vector_XXX
SG_odometro : 31|32@1+ (0.1,0) [0|429496730] "km" Vector_XXX

BO_11 posicaoAcelerador: 8 Vector_XXX
SG_posicaoAcelerador : 31|8@1+ (0.392,0) [0|100] "%" Vector_XXX

BO_15 nivelTanque: 8 Vector_XXX
SG_nivelTanque : 31|8@1+ (0.392,0) [0|100] "%" Vector_XXX

BO_17 temperaturaAmbiente: 8 Vector_XXX
SG_temperaturaAmbiente : 31|8@1+ (1,-40) [-40|215] "degC" Vector_XXX

BO_18 janelaEsquerda: 16 Vector_XXX
SG_posicaoJanelaEsquerda : 31|8@1+ (0.5,0) [0|127.5] "%" Vector_XXX
SG_switchPressionadoCima : 22|1@1+ (1,0) [0|1] "on/off" Vector_XXX
SG_switchPressionadoBaixo : 20|1@1+ (1,0) [0|1] "on/off" Vector_XXX

BO_19 janelaDireita: 16 Vector_XXX
SG_posicaoJanelaDireita : 31|8@1+ (0.5,0) [0|127.5] "%" Vector_XXX
SG_switchPressionadoCima : 22|1@1+ (1,0) [0|1] "on/off" Vector_XXX
SG_switchPressionadoBaixo : 20|1@1+ (1,0) [0|1] "on/off" Vector_XXX

BO_20 statusJanelas: 8 Vector_XXX
SG_statusJanelaEsquerda : 31|1@1+ (1,0) [0|1] "true/false" Vector_XXX
SG_statusJanelaDireta : 28|1@1+ (1,0) [0|1] "true/false" Vector_XXX
```

Figura 2. Arquivo DBC com o projeto das mensagens

4. ECU 1 - Painel de Informações

A primeira ECU configurada foi denominada de "Painel de Informações", através do seu painel gráfico, podemos visualizar as variações da velocidade do carro decorrentes da aceleração e da frenagem; a velocidade do motor; a temperatura ambiente dentro do carro e odômetro dispondo a quantidade fictícia de quantos quilômetros totais o carro já percorreu.

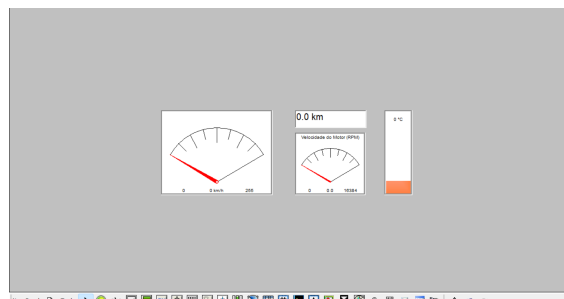


Figura 3. Print de tela do painel gráfico da ECU "Painel de Informações"

5. ECU 2 - Painel de Controle

A segunda ECU que apresentaremos é a ECU denominada "Painel de Controle", nela temos acesso aos botões representativos do freio e do acelerador, quatro botões que controlam as janelas das portas esquerda e direita, duas leds que indicam se as janelas estão abertas ou fechadas, um knob rotativo que controla a temperatura do ar condicionado e um gráfico de barra que revela a porcentagem na qual se encontra o nível do tanque do carro.

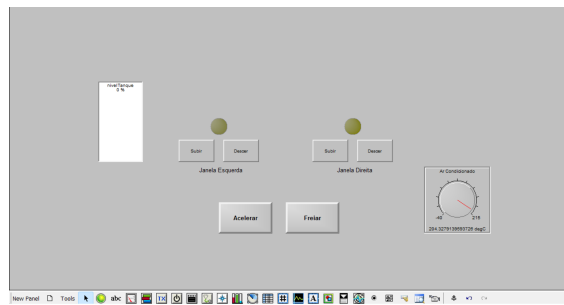


Figura 4. Print de tela do painel gráfico da ECU "Painel de Controle"

6. ECU 3 - Regulagem

A terceira ECU exibe, apenas, o quanto as janelas estão fechadas ou próximas disso, e retornam a informação quando as janelas chegam ao limite. Para a próxima parte do projeto pretendemos adicionar mais features.

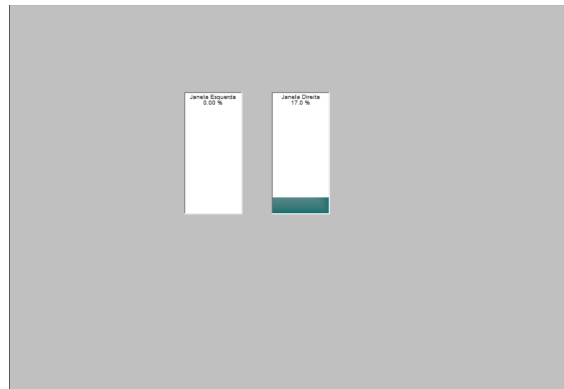


Figura 5. Print de tela do painel gráfico da ECU "Painel de Regulagem"

7. Comunicações

Realizamos transmissões entre três dispositivos ValueCAN4 ou NeoVi Fire 2, que atuam como ECUs, incluindo dados como nível do tanque, posição das janelas, velocidade do motor e do carro, pedais de aceleração e freio, odômetro e temperatura ambiente.

7.1. Transmissões ECU 1

A ECU 1, Painel de Informações, transmite uma única mensagem direcionada à ECU 2, Painel de Controle, referente ao sinal do nível do tanque de gasolina, que é calculado levando em conta a velocidade atual do carro, caso esteja em movimento ou não, recebida como mensagem vinda da ECU 2. Recebe uma mensagem de velocidade do carro, vinda da ECU 2 que está vinculada a um medidor no painel gráfico que exibe em km/h, também recebe uma mensagem de medidor do odômetro, temperatura ambiente e velocidade do motor, todas elas provenientes da ECU 2 e vinculadas a medidores no painel gráfico.

7.2. Trnasmissões ECU 2

A ECU 2, Painel de Controle, é a origem da maior parte das mensagens trocadas dentro da rede. Dela, partem as mensagens de: acionamento dos pedais, contendo dois sinais

8. Referências

[1] Intrepid Control Systems. Value CAN4-2. Disponível em: <https://cdn.intrepidcs.net/guides/ValueCAN4-2/>.

[2] Intrepid Control Systems. neoVI Fire 2. Disponível em: <https://cdn.intrepidcs.net/guides/neoVI-Fire2/>.

[3] PASTA1.0 CAN-ID List v1.0E. Disponível em: <https://github.com/pasta-auto/PASTA1.0/blob/master/doc/PASTA1.0List>

[4] CSS Electronics. DBC Editor for CAN Bus Database files. Disponível em: <https://www.csselectronics.com/pages/dbc-editor-can-bus-database>.

[5] CSS Electronics. OBD2 PID Overview [Lookup/Converter Tool, Table, CSV, DBC]. Disponível em: <https://www.csselectronics.com/pages/obd2-pid-table-on-board-diagnostics-j1979>

[6] Value Pasta Auto. Disponível em: <https://github.com/mintynet/value-pasta-auto>.

[7] Vehicle Spy 3 Graphical Panels - Advanced Features. Disponível em: <https://www.youtube.com/watch?v=poaBNNAes4>.

[8] Vehicle Spy 3 Basic Training. Disponível em: <https://www.youtube.com/playlist?list=PL77j9OugERjh9A0eLHHDCOiVsNZZ5Rm2w>.