

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

**CENTRO DE INFORMÁTICA**

**IF747 – REDES AUTOMOTIVAS**

**PROJETO – PARTE 1**

**PROFESSOR: Divanilson Campelo**

**MONITOR: Paulo Freitas**

**Rubem Moura**

**Thaysa Barros**

**Recife – 2016**

**PROPOSTA**

Como parte do projeto de Redes Automotivas foi lançada a proposta de implementar um sistema de captura de dados de um veículo em tempo real, cujos dados deveriam constituir um banco de dados. Dentre as solicitações, é necessária a tradução desses dados para que em outra etapa do projeto, as informações coletadas sejam usadas em simulações e aplicações.

**ETAPA 1 – TESTES EM CAMPO**

Baseado nas solicitações do projeto, o primeiro passo foi adquirir um Scanner Automotivo OBD2, o ELM 327, que envia dados aos aplicativos via Bluetooth. Com o ELM 327 foram realizadas três coletas de dados em um Ford Fiesta 2014, cada uma com um aplicativo diferente, o Torque Free, Torque Pro e OBD Doctor, sendo os dois primeiros usados em celular e o último em um notebook. No entanto, as duas versões do Torque não forneciam dados no formato de frames, que foi estabelecido como objetivo do projeto, e quanto ao OBD Doctor, a aquisição de dados não foi iniciada, apesar de estarem conectados.

Diante dos resultados negativos, outro Scanner foi utilizado, Vgate Bluetooth Wireless, e o OBD Doctor no notebook comunicando-se via Wifi. Nessa configuração, foram obtidos os frames de quatro sensores de um Ford Ka 2015, os quais informam velocidade, rotação, temperatura de resfriamento do motor e temperatura do ar que entra para participar da combustão, essa quantidade de sensores não confere com o solicitado, 5 sensores, mas dentre os muitos sensores disponíveis, esses foram os únicos que geraram informação através desse aplicativo. Os scanners podem ser visualizados na Figura 1 e as interfaces de comunicação, na Figura 2 do Apêndice.

**ETAPA 2 – INTERPRETAÇÃO DO DADOS**

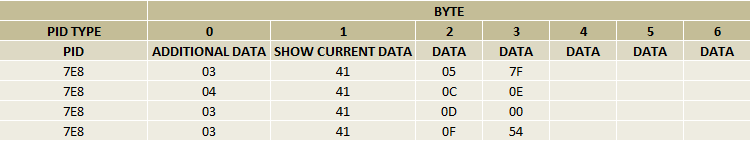
Assim, com um arquivo txt gerado pelo OBD Doctor, Figura 3, os dados foram transferidos para o Excel, Figura 4, onde as informações são facilmente separáveis por colunas, a fim de facilitar a visualização e manipulação do frame e a interpretação dele. Nessa figuras com trechos dos dados coletados, está indicada a coluna de direção, que significa o sentido de envio da mensagem: >> indica o envio de uma mensagem de requisição da ECU para o sensor e << indica a mensagem de resposta que parte do sensor e segue para a ECU.

Abaixo seguem exemplos de mensagem requisitada, Tabela 1, e mensagem de reposta, Tabela 2, separados de acordo com o significado de cada caractere.

W:\public_html\Redes Automotivas\Relatório\msgRequest.PNG

Tabela 1 – Mensagem-requisição

Tabela 2 – Frame de mensagem-resposta.



Acompanhando a tabela mensagem-requisição, o modo 01, o único que aparece neste teste, pede aos sensores que mostrem seus valores em tempo real, enquanto o PID revela o sensor ao qual a mensagem é destinada.

Seguindo para a tabela de mensagem-resposta, o PID TYPE é um identificador de uma ECU, o 7E8 é o identificador de resposta para a ECU #1 que corresponde ao módulo de controle do motor. As demais colunas formam o *payload*, são ao todo 8 bytes, mas os 6 da tabela serão suficientes para explicar os dados coletados. O byte 0 informa o byte máximo de dado que será enviado, nos exemplos tivemos dois casos, 03 e 04, observe que onde tem 03 o dado está preenchido até o byte 3 e o 04, até o byte 4. O byte 2 é um modo de consulta e 41 indica modo em tempo real, o byte 3 é a identificação do sensor e os seguintes são os que correspondem ao valor assumido pelos sensores. A identificação do sensor ou PID para os sensores utilizados no teste é: 05 para temperatura de resfriamento do motor, 0C para rotação, 0D para velocidade e 0F para temperatura do ar de admissão.

**ETAPA 3 – TRADUÇÃO DOS DADOS**

Para registrar a interpretação do *payload*, pois ficou acordado que os próximos passos do projeto seriam feitos com dado tratado, foi desenvolvido um programa em linguagem C que converte de hexadecimal para decimal os itens “DATA” do *payload*, Figura 5. Além disso, é necessário mais do que a simples conversão de base numérica, é preciso aplicar as fórmulas abaixo, segundo a referência [1]. As fórmulas 2 e 3 são, respectivamente, dos sensores 0C, 0D e a fórmula 1 é a mesma para os sensores 05, e 0F.

(1)

(2)

(3)

Observe que o valor *A* corresponde ao decimal do primeiro byte de dado e B é o decimal do segundo byte de dado.

Em seguida, no Excel os dados não tratados foram filtrados de maneira que só as mensagens-resposta permanecessem, já que elas serão necessárias na etapa seguinte, a simulação, para então submetê-las a um programa de tratamento de dados.

No código, os dados de mensagem-resposta foram importados de um arquivo txt, iniciando a sequência de eventos que varre todo o banco de dados até que a conversão completa seja feita. Os eventos do programa consistem principalmente em importar um frame – Figura 5,a partir da linha 163, *main* –, converter de hexadecimal para decimal – Figura 5, funções da linha 6 a 70 – e salvar em um novo arquivo – Figura 5, linha 72, dentro da função *processPayload* –, repetindo sequencialmente esses passos até o fim do banco de dados.

**APÊNDICE**

|  |  |  |
| --- | --- | --- |
|  | C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\interfaces.png | |
| Figura 1 – ELM 327 e Vgate. | Figura 2 – Interfaces para aquisição de dados. | |
| C:\Users\thays\Downloads\modeloBancoDados.PNG | |  |
| Figura 3 – Arquivo txt gerado pelo OBD Doctor. | | Figura 4 – Banco de dados no Excel. |
| C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\cod1.png | | |
| Figura 5 – Código de conversão do banco de dados. | | |
| C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\cod2.png | | |
| Figura 5 – Código de conversão do banco de dados. | | |
| C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\cod3.png | | |
| Figura 5 – Código de conversão do banco de dados. | | |
| C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\cod4.png | | | |
| Figura 5 – Código de conversão do banco de dados. | | | |
| C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\imagem01.png | | C:\Users\thays\Google Drive\Faculdade\Engenharia Eletrônica\9º periodo 2016.2\Redes Automotivas\Projeto\Relatório\image02.png |
| Figura 6 – Console com a conversão de dados. | | Figura 7 – Novo arquivo com dados convertidos. |

**REFERÊNCIAS**

[1] <https://en.wikipedia.org/wiki/OBD-II_PIDs#Mode_01>

[2] Material de Aula – IF747