

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ
Tecnologia em Eletrônica Automotiva**

**BRUNO SILVA PEREIRA
HENRIQUE MENDES CORDEIRO**

**SISTEMA DE DIAGNOSE VEICULAR
ON-BOARD EM UMA PLATAFORMA DIDÁTICA
DE GERENCIAMENTO ELETRÔNICO**

Santo André – São Paulo
2012

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ
Tecnologia em Eletrônica Automotiva**

**BRUNO SILVA PEREIRA
HENRIQUE MENDES CORDEIRO**

**SISTEMA DE DIAGNOSE VEICULAR
ON-BOARD EM UMA PLATAFORMA DIDÁTICA
DE GERENCIAMENTO ELETRÔNICO**

Monografia apresentada ao Curso de Tecnologia em Eletrônica Automotiva da FATEC Santo André, como requisito parcial para conclusão do curso em Tecnologia em Eletrônica Automotiva

*Orientador: Prof. Orlando Salvo Junior
Co-Orientador: Prof. Dr. Armando Antonio Maria Laganá*

Santo André – São Paulo
2012

Pereira, Bruno Silva; Cordeiro, Henrique Mendes

Sistema de Diagnose Veicular *On-Board* em uma Plataforma Didática de Gerenciamento Eletrônico / Bruno Silva Pereira, Henrique Mendes Cordeiro – 2012

nº 67: IL

Orientador: Orlando Salvo Junior

Co-Orientador: Armando Antonio Maria Laganá

Trabalho de Conclusão de Curso – Faculdade de Tecnologia de Santo André, Santo André, 2012.

1. Diagnose veicular. 2. Detecção de falhas. 3. Protocolo de Comunicação.

Faculdade de Tecnologia de Santo André "Dr. Newton da Costa Brandão"

LISTA DE PRESENÇA

SANTO ANDRÉ, 18 de dezembro de 2012.

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO TRABALHO
DE CONCLUSÃO DE CURSO COM O TEMA "SISTEMA DE DIAGNOSE
VEICULAR ON – BOARD EM UMA PLATAFORMA DIDÁTICA."
DOS ALUNOS DO 6º SEMESTRE DESTA U.E.

BANCA

PRESIDENTE:

PROF. ORLANDO DE SALVO JR.



MEMBROS:

PROF. DR. ARMANDO ANTONIO MARIA LAGANÁ



CONVIDADO ENG.º PAULO HENRIQUE RODRIGUES CORREA



AGRADECIMENTOS

Gostaríamos de agradecer aos nossos familiares que nos deram força e apoio para que este projeto se tornasse realidade.

Agradecemos ao professor Laganá ser o incentivador do desenvolvimento deste projeto e ao professor Orlando por nos orientar em sua construção.

Agradecemos ao professor Wagner por nos incentivar a concluir o projeto dentro do prazo.

Agradecemos todos os funcionários da Fatec e nossos companheiros que nos acompanharam nesta jornada para a conclusão deste curso.

*“O único lugar onde o sucesso vem
antes do trabalho é no dicionário”*

Albert Einstein

RESUMO

Desde a concepção dos primeiros automóveis, notou-se que era preciso a inclusão de sistemas que auxiliassem a interação do homem com o veículo, como por exemplo alarmando-o sobre possíveis falhas no motor. Atualmente esses sistemas gerenciam parâmetros do veículo, com o objetivo de propiciar ao condutor a verificação dos resultados das falhas de campo detectadas pelo sistema de diagnose, levando em consideração a agilidade e precisão das informações adquiridas. Este trabalho tem como objetivo o desenvolvimento de um sistema de diagnose veicular *on-board* numa plataforma didática de gerenciamento eletrônico desenvolvido pela FA-TEC. Este sistema analisará possíveis falhas no gerenciamento do motor e permitirá a comunicação com um equipamento de diagnóstico veicular, via protocolo de comunicação de padrão internacional OBD-2 (*On-Board Diagnostics Second Generation*) ISO15765, para a visualização de parâmetros, como por exemplo, rotação, temperaturas, lambda, entre outros. Nos últimos tempos, os sistemas de gerenciamento do motor tornaram-se muito complexos, dificultando a detecção de problemas e falhas. A implementação de um sistema de diagnose se faz necessária para o diagnóstico rápido de falhas e com maior precisão do que feito manualmente, principalmente as falhas relacionadas às emissões de poluentes, reduzindo parâmetros como tempo e custo de manutenção.

Palavras chave: diagnose veicular, detecção de falhas, protocolo de comunicação.

ABSTRACT

Since the conception of the first automobiles, it was noted that the inclusion of systems that would assist the human-machine interface, such as alarming about possible engine failures. Currently, these systems manage vehicle parameters, with the goal of providing the driver to check the results of the field faults detected by the diagnostic system, taking into account the quickness and accuracy of the acquired information. This study aims to develop an on-board vehicle diagnostic system in a didactic electronic management platform developed by FATEC. This system will consider possible faults in the engine management and provide communication with vehicle diagnostic equipment via international standard communication protocol OBD-2 (On-Board Diagnostics Second Generation) ISO15765, for viewing parameters, such as, engine speed, temperature, lambda, among others. In recent times, the engine management became very complex, making more difficult the problems and faults detection. The implementation of a diagnostic system is necessary for the fast diagnostic fault and with higher accuracy than manually done, especially faults related to pollutants emissions, reducing parameters such as time and maintenance costs.

Key words: vehicle diagnosis, fault detection, communication protocol.

LISTA DE ILUSTRAÇÕES

Figura 1.1 <i>Interface</i> Homem-Máquina (Extraído de (GUIMARÃES, 2007, p. 235)....	17
Figura 2.1 Poluição em Los Angeles, Califórnia (Extraído de (The Atlantic, 2012)) ..	19
Figura 2.2 Lâmpada Indicadora de Mau-Funcionamento (Extraído de (MACHADO e OLIVEIRA, 2007))	23
Figura 2.3 Monitoramento do Catalisador (Adaptado de (Inga Escapamentos, 2012))	24
Figura 2.4 Detecção de <i>misfire</i> (Extraído de (JUNIOR, 2012b))	24
Figura 2.5 Conceito da plausibilidade (Extraído de (JUNIOR, 2012b))	26
Figura 2.6 Barramento de Diagnóstico ISO 9141 (Extraído de (GUIMARÃES, 2007, p. 243))	27
Figura 2.7 Localização do conector de diagnóstico (Extraído de (JUNIOR, 2012b))	29
Figura 2.8 Pinagem do conector OBD-2 (Extraído de (JUNIOR, 2012b))	29
Figura 2.9 Exemplo de um código de falha decodificado (Adaptado de (ISO, 2005))	32
Figura 2.10 Equipamento de Diagnóstico (Extraído de (JUNIOR, 2012b))	33
Figura 2.11 <i>Frame</i> CAN 2.0A (Extraído de (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003)).....	36
Figura 2.12 <i>Frame</i> CAN 2.0B (Extraído de (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003)).....	36
Figura 2.13 As 7 Camadas do modelo ISO/OSI (Extraído de (HODEL, 2011)).....	37
Figura 2.14 Par de fio trançados	38
Figura 2.15 Sinal diferencial da rede CAN (Extraído de (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003)).....	38
Figura 3.1 Diagrama de Blocos da ECU (Extraído de (DIAS, 2011))	44
Figura 3.2 Dispositivos utilizados neste projeto	45
Figura 3.3 Fluxograma da interrupção de chegada de dados na porta serial	46
Figura 3.4 Faixa de operação do sinal do sensor de posição da borboleta	48
Figura 3.5 Fluxograma da tarefa de diagnóstico	48
Figura 3.6 Fluxograma da tarefa de conversão de parâmetros.....	49
Figura 3.7 Fluxograma da tarefa de comunicação de diagnóstico	53

Figura 3.8 Fluxograma da função do serviço \$01	54
Figura 4.1 Projeto conectado ao equipamento de diagnóstico pelo conector OBD...	58
Figura 4.2 Equipamento de Diagnóstico VAS5052 (Extraído de (AUDI VOLKSWAGEN, 2005))	59
Figura 4.3 Led da lâmpada indicadora de mau-funcionamento iluminada	60
Figura 4.4 Led da lâmpada indicadora de mau-funcionamento apagada.....	60
Figura 4.5 Menu de serviços e identificação do protocolo de comunicação.....	61
Figura 4.6 Mensagem indicadora de comunicação no <i>display</i>	61
Figura 4.7 Primeira tela dos parâmetros de leitura no serviço \$01	62
Figura 4.8 Segunda tela dos parâmetros de leitura no serviço \$01	62
Figura 4.9 Leitura dos códigos de falha no serviço \$03	63
Figura 4.10 Tela de confirmação da eliminação dos códigos de falha	63

LISTA DE TABELAS

Tabela 2.1 Primeiro Caractere do Código de Falha (Adaptado de (BELO, 2003))	31
Tabela 2.2 Segundo Caractere do Código de Falha (Adaptado de (BELO, 2003)) ...	31
Tabela 2.3 Terceiro Caractere do Código de Falha (Adaptado de (BELO, 2003))	31
Tabela 2.4 Exemplos de DTCs (Extraído de (GUIMARÃES, 2007))	32
Tabela 2.5 Exemplo de uma mensagem de solicitação ISO 15765	39
Tabela 2.6 Exemplo de uma mensagem de resposta ISO 15765	40
Tabela 3.1 Conversão da temperatura (Adaptado de (ISO, 2006))	50
Tabela 3.2 Conversão da rotação (Adaptado de (ISO, 2006))	50
Tabela 3.3 Conversão do valor da abertura da válvula borboleta (Adaptado de (ISO, 2006))	51
Tabela 3.4 Conversão do valor de pressão absoluta no coletor de admissão (Adaptado de (ISO, 2006))	51
Tabela 3.5 Conversão do valor de tensão da sonda lambda (Adaptado de (ISO, 2006))	52
Tabela 3.6 Resposta da solicitação do PID 0x00	53
Tabela 3.7 Mensagem com o valor da temperatura do líquido de arrefecimento	55
Tabela 3.8 Mensagem com o valor da pressão absoluta no coletor de admissão	55
Tabela 3.9 Mensagem com o valor da rotação do motor	55
Tabela 3.10 Mensagem com o valor da temperatura do ar de admissão	55
Tabela 3.11 Mensagem com o valor da posição da válvula borboleta	56
Tabela 3.12 Mensagem com a informação de sondas <i>lambdas</i> presentes	56
Tabela 3.13 Mensagem com o valor da tensão da sonda <i>lambda</i>	56
Tabela 3.14 Mensagem com o valor do número de falhas presentes	57
Tabela 3.15 Mensagem de resposta com os códigos de falha	57
Tabela 3.16 Mensagem de confirmação do serviço \$04	57

LISTA DE ABREVIATURAS

CAN	- Protocolo de Comunicação (<i>Controller Area Network</i>)
CARB	- <i>California Air Resources Board</i>
CO	- Monóxido de Carbono
CO ₂	- Dióxido de Carbono
CONAMA	- Conselho Nacional do Meio Ambiente
DLC	- Código do tamanho dos dados (<i>Data Length Code</i>)
DTC	- Códigos de Diagnóstico de Falha (<i>Diagnostic Trouble Codes</i>)
ECU	- Unidade de controle eletrônico (<i>Electronic Control Unit</i>)
EGR	- Sistema de recirculação dos gases de escape (<i>Exhaust Gas Recirculation</i>)
GND	- Terra (<i>Ground</i>)
HC	- Hidrocarboneto
HMI	- Interface Homem-máquina (<i>Human-machine interface</i>)
ID	- Identificador da mensagem (<i>Identifier</i>)
ISO	- Organização Internacional para Padronização (<i>International Organization for Standardization</i>)
KWP 2000	- Protocolo de Comunicação (<i>Keyword Protocol 2000</i>)
LIM	- Lâmpada indicadora de mau-funcionamento
NO _x	- Óxido de Nitrogênio
O ₂	- Oxigênio
OBD	- Sistema de diagnóstico embarcado (<i>On-Board Diagnostics</i>)
OBD-2	- Sistema de diagnóstico embarcado segunda geração (<i>On-Board Diagnostics Second Generation</i>)
PID	- Identificador de parâmetro (<i>Parameter Identifier</i>)

PWM	- Modulação por largura de pulso (<i>Pulse Width Modulation</i>)
RTOS	- Sistema Operacional de Tempo Real (<i>Real Time Operating system</i>)
SAE	- Sociedade de Engenheiros Automotivos (<i>Society of Automotive Engineers</i>)
SPI	- Interface Serial Periférica (<i>Serial Peripheral Interface</i>)
TPS	- Sensor de posição da válvula borboleta (<i>Throttle Position Sensor</i>)
USART	- Transmissor e Receptor Universal Síncrono e Assíncrono (<i>Universal Synchronous Asynchronous Receiver Transmitter</i>)
VPW	- Modulação de pulso variável (<i>Variable Pulse Width</i>)

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivos e Motivação.....	18
1.2	Conteúdo	18
2	SISTEMAS DE DIAGNOSE VEICULAR	19
2.1	Histórico	19
2.1.1	Padrão OBD no Brasil (OBDBr).....	20
2.2	Sistemas de Diagnóstico <i>On-Board</i>	22
2.3	Monitoramento do Gerenciamento do Veículo.....	23
2.3.1	Monitoramento da Eficiência do Catalisador	23
2.3.2	Monitoramento de Falhas de Ignição	24
2.3.3	Monitoramento do Sistema de Combustível.....	25
2.3.4	Monitoramento da Sonda <i>Lambda</i>	25
2.3.5	Monitoramento dos Demais Sistemas do Veículo	25
2.4	Protocolos de Comunicação de Diagnóstico	26
2.4.1	Protocolos mais Utilizados	27
2.4.1.1	ISO 9141	27
2.4.1.2	KWP2000 (ISO 14230)	28
2.4.1.3	ISO 15765 CAN Bus	28
2.4.2	Conector OBD-2	28
2.4.3	Serviços de Diagnóstico	29
2.4.4	Códigos de Diagnóstico de Falha.....	30
2.5	Sistemas de diagnóstico <i>Off-Board (Scanner)</i>	32
2.5.1	Funções do <i>scanner</i>	33
2.6	Rede CAN.....	34
2.6.1	Mensagens do CAN	35
2.6.1.1	Formato das mensagens CAN	35
2.6.2	Modelo ISO/OSI	36
2.6.3	Características Físicas	37
2.6.4	Diagnóstico sobre a rede CAN (ISO 15765).....	38

2.7	USART (Transmissor e Receptor universal síncrono e assíncrono).....	40
2.8	SPI (Interface Serial Periférica)	40
2.9	Sistemas Operacionais de Tempo Real (RTOS)	41
3	METODOLOGIA	43
3.1	<i>Hardware</i> Utilizado	43
3.2	Desenvolvimento do <i>Software</i>	46
3.2.1	Tarefa de Diagnóstico de Falhas.....	47
3.2.2	Tarefa de Conversão de Parâmetros	49
3.2.2.1	Conversão dos valores de temperatura	50
3.2.2.2	Conversão do valor da rotação do motor	50
3.2.2.3	Conversão do valor de posição da válvula borboleta.....	51
3.2.2.4	Conversão do valor de pressão no coletor de admissão	51
3.2.2.5	Conversão do valor de tensão da sonda <i>lambda</i>	52
3.2.3	Tarefa de Comunicação	52
3.2.3.1	Estabelecimento da Comunicação.....	53
3.2.3.2	Serviço \$01	54
3.2.3.2.1	PID 0x05 – Temperatura do líquido de arrefecimento	54
3.2.3.2.2	PID 0x0B – Pressão absoluta no coletor de admissão	55
3.2.3.2.3	PID 0x0C – Rotação do motor	55
3.2.3.2.4	PID 0x0F – Temperatura do ar de admissão	55
3.2.3.2.5	PID 0x11 – Posição da válvula borboleta	55
3.2.3.2.6	PID 0x14 – Tensão da sonda <i>lambda</i>	56
3.2.3.3	Serviço \$03	56
3.2.3.4	Serviço \$04	57
4	RESULTADOS OBTIDOS.....	58
4.1	Testes Realizados	59
4.1.1	Diagnóstico de falhas do sensor TPS.....	59
4.1.2	Interação com o equipamento de diagnóstico	61
4.1.3	Utilização dos serviços OBD-2	62
5	CONCLUSÃO	64
5.1	Propostas Futuras	65

6 REFERÊNCIAS.....	66
--------------------	----

1 INTRODUÇÃO

Com o crescimento da eletrônica no cenário mundial e a utilização de circuitos integrados cada vez menores e confiáveis, sua introdução na indústria automobilística foi inevitável, principalmente devido à procura pela redução de custos de fabricação. Isso possibilitou o surgimento de novas tecnologias, substituindo elementos mecânicos por eletrônicos, aumentando desempenho e confiabilidade do veículo.

Aliado com a pressão pela diminuição das emissões de poluentes de combustíveis fósseis, a crise do petróleo e com a expansão de tecnologias embarcadas nos veículos, os sistemas que o compõem passaram a ser cada vez mais complexos, dificultando a detecção e manutenção de problemas. Desta forma os sistemas de diagnose veicular surgiram na década de 70 pela necessidade de diagnosticar falhas que não pudessem ser detectadas pelo homem, mas que pudessem ser detectadas por equipamentos eletrônicos, por meio de uma *interface* homem-máquina, com eficiência, agilidade e precisão, e ainda realizar o controle de emissões de poluentes (GUIMARÃES, 2007).

O sistema de diagnóstico veicular é composto basicamente por dois *softwares* conectados por uma linha de comunicação. O primeiro é realizado pelo módulo eletrônico do veículo, a ECU (*Electronic Control Unit*), e o segundo por um equipamento de diagnose que realiza a *interface* do módulo eletrônico com o operador, chamado de HMI (*Human-machine interface*) ou *interface* homem-máquina (GUIMARÃES, 2007). Este sistema pode ser visualizado na figura 1.1 a seguir.

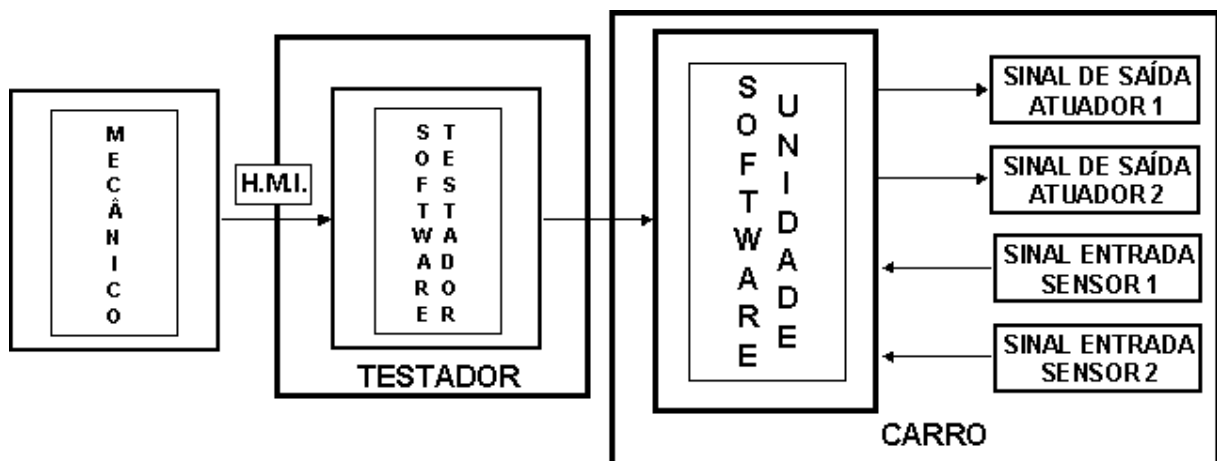


Figura 1.1 *Interface* Homem-Máquina (Extraído de (GUIMARÃES, 2007, p. 235).

1.1 Objetivos e Motivação

Este trabalho tem o objetivo de desenvolver um sistema de diagnose numa plataforma didática de gerenciamento eletrônico, possibilitando uma *interface* com um equipamento de diagnóstico para a visualização dos parâmetros de funcionamento do veículo em tempo real tais como a rotação do motor, temperatura do ar de admissão, posição da válvula borboleta e lista de códigos de falhas, identificando problemas eventuais.

A motivação para a criação deste projeto é o desafio de criar um sistema existente em uma ECU profissional, visto que as estratégias de funcionamento do sistema de gerenciamento eletrônico do motor são restritas fora do ambiente de desenvolvimento das indústrias automobilísticas devido à grande competição entre elas.

1.2 Conteúdo

Este trabalho estará dividido desta maneira; o capítulo 2 discorrerá sobre uma revisão bibliográfica com o objetivo de apresentar o assunto deste trabalho; o capítulo 3 apresentará o sistema completo na qual será possível analisar toda a arquitetura eletrônica e de *software*, no capítulo 4 descreverá os ensaios e os resultados obtidos, estratificando-os para um estudo comparativo entre o desejado e o realizado. Verificaremos se a meta foi atingida e quais foram as dificuldades encontradas neste projeto. Finalmente, no capítulo 5 discorreremos sobre as conclusões obtidas dos resultados e a proposição de novos estudos e desafios.

2 SISTEMAS DE DIAGNOSE VEICULAR

2.1 Histórico

Desde a incorporação da eletrônica embarcada nos veículos, principalmente entre os anos 70 e 80, começaram a surgir sistemas de diagnóstico nos veículos. Estes sistemas foram projetados para controlar os níveis de emissão de poluentes dos veículos nos padrões desejados pelo fabricante em cumprimento às leis ambientais (BELO, 2003).

O governo da Califórnia foi o primeiro a criar medidas para frear o aumento da poluição, dado que o alto índice destes compostos químicos estava afetando significativamente a sociedade em termos ambientais com a degradação do meio ambiente e afetando a saúde da população, conforme ilustrado na figura 2.1. Em 1966 a *California Air Resources Board* (CARB), obrigou a indústria automobilística a ter um controle da emissão de poluentes de seus veículos. Após dois anos, esta medida foi implementada em todo o território Norte-Americano (MACHADO e OLIVEIRA, 2007).



Figura 2.1 Poluição em Los Angeles, Califórnia (Extraído de (The Atlantic, 2012))

As leis impostas eram muito rígidas e obrigaram as corporações a tomarem medidas para reduzir as fontes poluidoras. A partir disso passou a ser obrigatório manutenções periódicas e padronizações das medições que seriam importantes de

ser realizadas para o controle das emissões. Além disso, foram adotadas novas tecnologias, por exemplo, os sensores e atuadores que auxiliavam no controle da emissão de poluentes, verificando periodicamente o ciclo de funcionamento do motor, e por consequência aumentavam o desempenho do veículo.

Neste período surgiu a idéia da criação de um sistema de diagnóstico de bordo com o objetivo de realizar a verificação do sistema de controle de emissões. Esse sistema foi denominado de OBD (*On-board Diagnostics*), este dispositivo realizava a verificação de parâmetros tais como: sensor de oxigênio, sistema de recirculação dos gases de escape, sistema de combustível, componentes eletrônicos, informação de diagnóstico e código de falhas. Em contrapartida, o sistema OBD não era muito eficiente, pois cada companhia criava seus próprios sistemas por completo, desde protocolos de comunicação a *hardware* e conexões (MACHADO e OLIVEIRA, 2007). Outro grande problema dos primeiros sistemas de diagnose era que eles apenas detectavam as falhas quando algum dos componentes monitorados não estava mais funcionando, permitindo que o motorista circulasse com o veículo em condições precárias causando um aumento nas emissões de poluentes.

Segundo Machado e Oliveira, no ano de 1988, a Sociedade dos Engenheiros Automotivos (SAE), determinou que todas as companhias adotassem um sistema de diagnóstico *on-board*. Isso levou à criação de um sistema chamado de OBD-1 (*On-board Diagnostics Generation 1*). Em 1996, devido à falta de padronização e as precárias condições de funcionamento do sistema anterior, surgiu o OBD-2 que trouxe melhorias na verificação dos componentes relacionados com o controle de emissões, trazendo informações confiáveis e precisas, aumentando a confiabilidade e facilitando a manutenção dos veículos.

A adoção dos sistemas de diagnóstico veicular trouxe inúmeros benefícios, ajudando na redução da emissão de poluentes, resultando na melhoria da qualidade de vida da sociedade.

2.1.1 Padrão OBD no Brasil (OBDBr)

O sistema de diagnóstico brasileiro denominado OBDBr surgiu em 13 de dezembro de 2004 com a resolução 354 do CONAMA (Conselho Nacional do Meio Ambiente), com o objetivo de identificar o mau funcionamento dos sistemas de controle de emissões. Para que as montadoras se adequassem a essa norma, foi ne-

cessário um período de adaptação, devido à necessidade de se desenvolver tecnologias que atendessem a esta resolução. Em 2007 iniciou-se a implantação do OBDBr-1 incorporando apenas 40% total de veículos produzidos, vale ressaltar que neste período a incorporação deste sistema gerou confusão nas montadoras que não sabiam qual norma atender entre o PROCONVE fase IV e o OBDBr-1 que vigoravam no mesmo ano. No ano seguinte foram incorporados 60%, e 100% da frota em 2009 (NETO, 2009). O OBDBr-1 deve monitorar os seguintes parâmetros:

- Sensor de Pressão Absoluta ou Fluxo de Ar;
- Sensor da Posição da Borboleta;
- Sensor de Temperatura de Arrefecimento;
- Sensor de Temperatura de Ar;
- Sonda *Lambda* (somente o sensor antes do catalisador);
- Sensor de Velocidade do Veículo;
- Sensor de Posição do Eixo Comando de Válvulas;
- Sensor de Posição do Virabrequim;
- Sistemas de Recirculação dos Gases de Escape (EGR);
- Sensor para Detecção de Detonação;
- Válvulas Injetoras;
- Sistema de Ignição;
- Módulo Controle Eletrônico do motor;
- Lâmpada Indicadora de Mau Funcionamento (LIM);
- Outros componentes que o fabricante julgue relevantes para a correta avaliação do funcionamento do veículo e controle de emissões de poluentes.

O OBDBr-2 iniciou em 2010 com o objetivo melhorar a detecção de falhas, identificar a deterioração de sensores e diminuir a emissão de poluentes, comparado ao OBDBr-1, devido a incorporação de uma segunda sonda *lambda* pós catalisador, instalada apenas para monitorar a eficiência do conversor catalítico (NETO, 2009).

Em 2010 apenas 60% dos veículos produzidos foram obrigados a seguir o padrão OBDBr-2, e a partir de 2011 100% da frota produzida incorporou o OBDBr-2.

2.2 Sistemas de Diagnóstico *On-Board*

Para Guimarães (2007), um sistema de diagnóstico veicular é o conjunto de ferramentas que permite a verificação do funcionamento dos módulos eletrônicos existentes em um veículo e também sua programação. Outra definição para o sistema de diagnose, é, basicamente a medição de parâmetros do veículo ou a leitura de falhas através do painel de instrumentos e também a medição destes parâmetros (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003).

A função dos sistemas de diagnóstico *On-Board* é monitorar todos os componentes que podem alterar o controle de emissões. Cada componente é monitorado por uma rotina de diagnose que verifica o seu correto funcionamento. Quando uma falha é detectada, o sistema gera um código de falha que é armazenado na memória não volátil da ECU. Assim este código de falha pode ser lido por uma ferramenta de diagnóstico *Off-Board* através de um protocolo de comunicação padrão (BELO, 2003).

O sistema de diagnóstico *On-Board* disponibiliza também os parâmetros de funcionamento do veículo em tempo real, tais como a rotação do motor, temperatura do ar de admissão, temperatura do motor, tempo de injeção de combustível, tempo de ignição, *lambda*, entre outros. Permite ainda que sejam feitas intervenções nos parâmetros de atuação do funcionamento do motor se for constatado que a integridade do motor pode ser afetada, como por exemplo a situação de *misfire*, além de indicar ao condutor do veículo através da iluminação da lâmpada indicadora de falhas no painel de instrumentos. Esta lâmpada (Figura 2.2) tem como objetivo alertar o condutor sobre alguma possível falha no funcionamento de algum componente ou sistema relacionado ao controle de emissões, como por exemplo na sonda *lambda*.

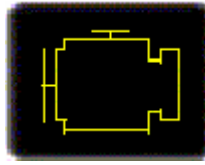


Figura 2.2 Lâmpada Indicadora de Mau-Funcionamento (Extraído de (MACHADO e OLIVEIRA, 2007))

2.3 Monitoramento do Gerenciamento do Veículo

Os principais sistemas que a ECU do motor monitora para diagnóstico são sistemas ligados ao controle da emissão de poluentes e serão apresentados a seguir.

2.3.1 Monitoramento da Eficiência do Catalisador

Segundo Milhor (2002), o sistema catalítico mais utilizado na indústria automobilística é o conversor catalítico de três estágios. O sistema opera recebendo os gases provenientes da combustão. Para que estes gases nocivos não afetem a saúde humana e o meio ambiente se faz necessário o pós-tratamento da combustão pelo conversor catalítico que transforma três gases tóxicos CO , HC e NO_x em três gases não tóxicos CO_2 , H_2O e N_2 , removendo mais de 90% dos gases poluentes.

O monitoramento deste sistema é baseado na comparação do nível de O_2 que entra no conversor medido pela primeira sonda *lambda*, que realiza o controle da mistura ar/combustível conforme a condição de trabalho do motor, com o nível de O_2 que sai do conversor pelo escapamento medido por uma segunda sonda *lambda* cujo objetivo é realizar o monitoramento da eficiência do catalisador (Figura 2.3). Essas moléculas de O_2 são usadas no processo catalítico do catalisador, logo, se os valores de saída medidos são maiores dos que os pré-estabelecidos para o veículo, o sistema de diagnóstico entende que há algum problema e toma as providências de armazenamento do diagnóstico de falha e ilumina a lâmpada de falhas no painel de instrumentos (BELO, 2003).

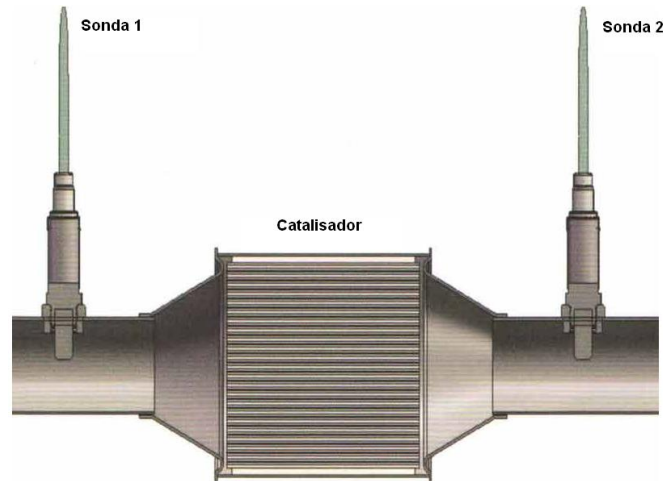


Figura 2.3 Monitoramento do Catalisador (Adaptado de (Inga Escapamentos, 2012))

2.3.2 Monitoramento de Falhas de Ignição

As falhas de ignição, também conhecidas como *misfire*, podem ocorrer pela falta de combustível em algum cilindro do motor, por desgastes de componentes, misturas de combustível alteradas e por falhas no sistema elétrico. O sistema de diagnóstico detecta essas falhas pela variação da velocidade do eixo de manivelas do motor no momento da combustão de um dado cilindro (figura 2.4) reconhecido pelo sensor de posição do comando de válvulas. Para confirmar esta informação, o sistema de diagnóstico leva em consideração a velocidade do veículo provida pelo sensor de velocidade do veículo. Se uma falha de ignição for detectada em algum cilindro, o sistema de diagnose atua no sistema de gerenciamento desativando o bico injetor do cilindro com falha, pois o catalisador seria danificado pelo combustível não queimado (BELO, 2003).

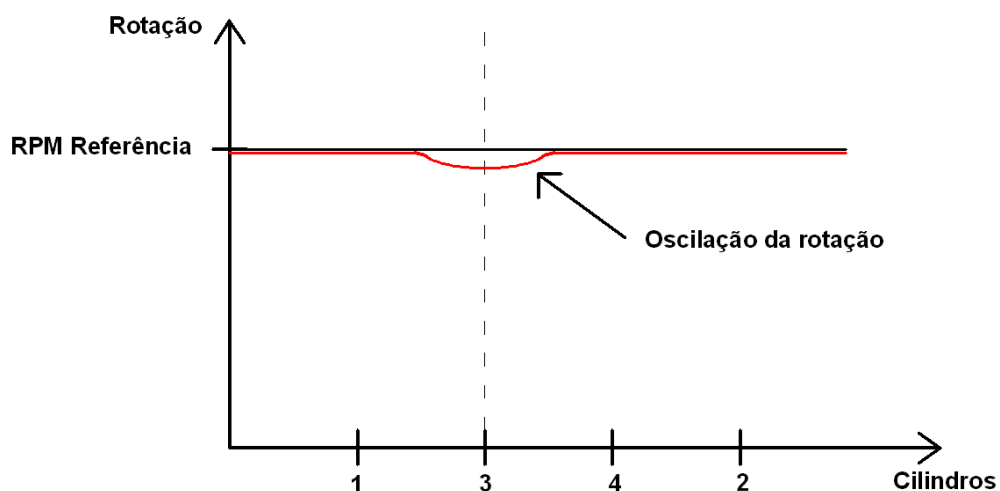


Figura 2.4 Detecção de *misfire* (Extraído de (JUNIOR, 2012b))

2.3.3 Monitoramento do Sistema de Combustível

Esse monitoramento tem o objetivo de prover a relação correta de ar-combustível em todas as condições de funcionamento do veículo. O diagnóstico deste sistema é feito comparando a qualidade da mistura através da sonda *lambda* com a quantidade ideal estipulada para cada carga e rotação do motor (BELO, 2003).

2.3.4 Monitoramento da Sonda *Lambda*

A sonda *lambda* (ou sensor de oxigênio) tem um papel importante no sistema de funcionamento do motor e no sistema de diagnóstico. Portanto se faz necessário o monitoramento do correto funcionamento deste sensor, realizado através de testes de continuidade, testes dos aquecedores do sensor, e testes das tensões geradas pelo sensor (BELO, 2003).

2.3.5 Monitoramento dos Demais Sistemas do Veículo

Como dito anteriormente, os principais sistemas que o sistema de diagnóstico monitora são relacionados ao controle de emissão de poluentes. Entretanto, os demais sistemas do veículo são monitorados, principalmente os ligados aos sistemas de segurança. Um dos métodos utilizados para monitorar estes sistemas é através da medição dos valores que devem estar dentro de uma faixa de operação conhecida.

Entretanto, existem casos em que o valor de um sensor pode estar dentro da faixa de operação e mesmo assim pode estar com uma falha. Para diagnosticar esse tipo de falha é utilizado o conceito de plausibilidade, que consiste em cruzar informações de um sensor com outros sensores ou com outras informações do motor, ou até mesmo com um mapa de funcionamento de um sensor. Por exemplo se o valor do sensor de posição da borboleta estiver com um valor alto, porém dentro de sua faixa de operação, e a rotação do motor estiver em marcha lenta (figura 2.5), significa que a válvula borboleta estaria totalmente aberta e a rotação baixa em marcha lenta, esta condição não é possível de acontecer no motor pois ele provavelmente morreria por excesso de ar ao invés de continuar funcionando, logo, isto é sinal de que pode haver algum problema com a válvula borboleta e então é armazenado um DTC de valor não plausível referente à válvula borboleta (JUNIOR, 2012b).

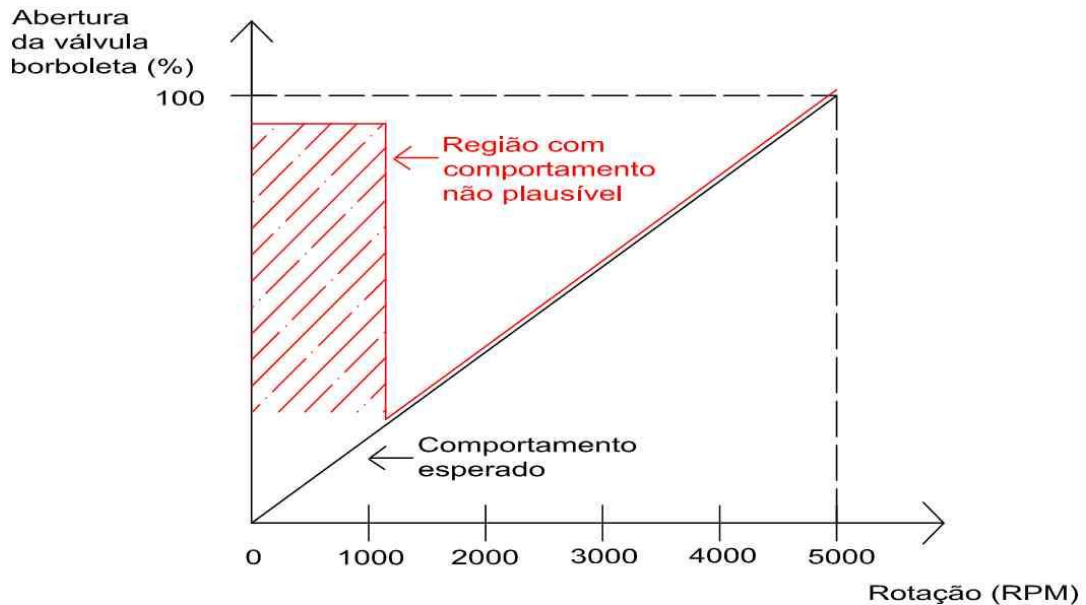


Figura 2.5 Conceito da plausibilidade (Extraído de (JUNIOR, 2012b))

Atualmente, a monitoração desses sistemas é bastante complexa, sendo baseada em modelos matemáticos probabilísticos tais como o explicado por Belo (2003), ou o proposto por Gomes (2008).

2.4 Protocolos de Comunicação de Diagnóstico

Segundo Guimarães (2007), protocolos de comunicação são meios de transmissão e recepção de dados com o objetivo de comunicar os módulos e sensores e atuadores eletrônicos através de microcontroladores e *transceivers*. Guimarães ainda explica que existem variados tipos de protocolos de comunicação, cada um com suas próprias características, trazendo consigo aplicações mais adequadas para tal tipo de protocolo.

Os protocolos de comunicação utilizados nos sistemas de diagnóstico são padronizados por duas principais organizações, a ISO (*International Organization for Standardization*) e a SAE (*Society of Automotive Engineers*) (CERQUEIRA, BEZERRA, *et al.*, 2009).

Cada protocolo necessita de um tratamento para se conectar entre o sistema de diagnóstico do veículo e o equipamento externo de diagnóstico. Existem equipamentos externos com a capacidade de identificar qual o protocolo utilizado pelo veículo e se adequar a ele.

Existem basicamente cinco protocolos usados na *interface* OBD-2:

- SAE J1850 PWM;
- SAE J1850 VPW;
- ISO 9141-2;
- ISO 14230 KWP 2000;
- ISO 15765 CAN Bus.

2.4.1 Protocolos mais Utilizados

2.4.1.1 ISO 9141

O protocolo ISO 9141 descreve os requisitos para estabelecer a troca de dados entre o módulo e o equipamento de diagnóstico (GUIMARÃES, 2007). As principais especificações desta norma são:

- Os módulos devem possuir uma ou duas linhas de comunicação, denominadas de linha K e linha L;
- A linha K é uma linha de comunicação bidirecional, pois envia e recebe dados de forma digital serial;
- A linha L é responsável somente por enviar dados do equipamento de diagnóstico para a ECU sendo uma linha de comunicação unidirecional.

Essas duas linhas, K e L, conectadas formam um barramento, conforme apresentado na figura 2.6 a seguir:

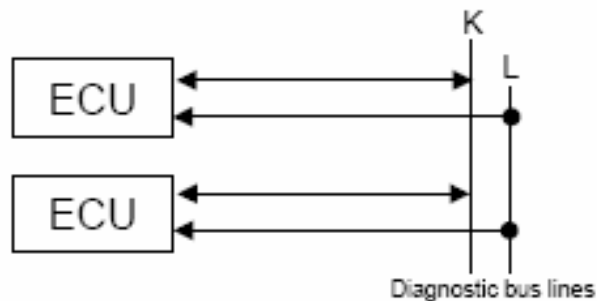


Figura 2.6 Barramento de Diagnóstico ISO 9141 (Extraído de (GUIMARÃES, 2007, p. 243))

2.4.1.2 KWP2000 (ISO 14230)

O protocolo KWP 2000 (*Keyword Protocol 2000*) é o nome dado à norma ISO 14230. Este protocolo foi desenvolvido baseado no ISO 9141-2 com o objetivo de ser flexível e obter maior desempenho no interfaceamento entre o módulo e o equipamento de diagnóstico (GUIMARÃES, 2007).

2.4.1.3 ISO 15765 CAN Bus

Atualmente o protocolo ISO 15765 está substituindo o protocolo KWP2000 pelo motivo de utilizar como protocolo de transmissão física o protocolo CAN (*Controller Area Network*), conferindo maior velocidade de transmissão e confiabilidade do sistema devido à robustez deste. Em 2008 nos Estados Unidos, o protocolo ISO 15765 CAN Bus foi implementado como a única *interface* permitida para diagnóstico (BOSCH, 2005). Este protocolo será detalhado no tópico 2.6.4 desta monografia.

2.4.2 Conector OBD-2

O conector padrão do sistema de diagnóstico é definido pela ISO 15031-3, tendo que atender algumas especificações (JUNIOR, 2012b), tais como:

- O conector deve estar localizado no compartimento do motorista ou do passageiro, conforme apresentado na figura 2.7;
- Delimitado pela extremidade do painel de instrumentos a 300mm além do eixo do veículo;
- De fácil acesso a partir do assento do motorista;
- Localizado preferencialmente entre a coluna de direção e o eixo do veículo;
- Deve ser montado para facilitar a conexão e desconexão do conector.



Figura 2.7 Localização do conector de diagnóstico (Extraído de (JUNIOR, 2012b))

A pinagem do conector também é definida pela ISO 15031-3, ilustrado na figura 2.8:

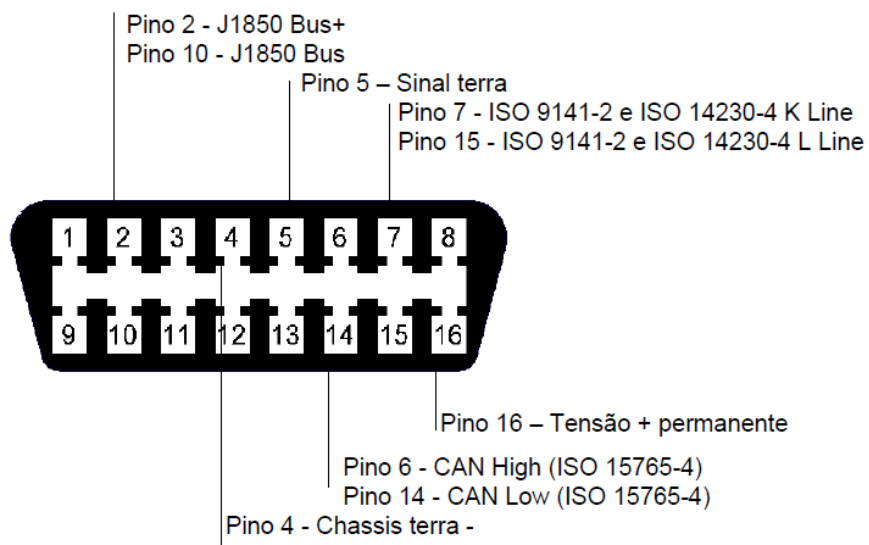


Figura 2.8 Pinagem do conector OBD-2 (Extraído de (JUNIOR, 2012b))

2.4.3 Serviços de Diagnóstico

Os serviços de diagnóstico são modos de operação entre um equipamento de diagnóstico e uma unidade de controle. Estes serviços separam quais tipos de informação o operador do equipamento de diagnóstico quer executar. Segundo Bastos (2012) existem atualmente dez destes serviços e são definidos pela ISO 15031-5, é denominado a cada um destes serviços também um número de identificação:

- Serviço \$01 - Solicita os parâmetros de leitura;
- Serviço \$02 - Solicita *freeze-frames*;
- Serviço \$03 - Solicita os códigos de falha armazenados;
- Serviço \$04 - Solicita o apagamento da memória de falhas;
- Serviço \$05 - Solicita os resultados de testes de monitoramento da sonda *lambda*;
- Serviço \$06 - Solicita os resultados de testes de monitoramento de outros componentes ligados ao sistema de emissões;
- Serviço \$07 - Solicita os códigos de falhas armazenados durante o último ciclo de condução do veículo;
- Serviço \$08 - Solicita o controle para testes de componentes;
- Serviço \$09 - Solicita informações do veículo;
- Serviço \$0A – Solicita os códigos de falhas com *status* permanentes.

Dentro de cada serviço existem ainda outros identificadores, chamados de identificadores de parâmetros (PIDs) (BASTOS, 2012). O PID é um identificador que indica uma informação específica de um sistema do veículo. Por exemplo, se o equipamento de diagnóstico necessita requisitar o valor de rotação do motor, ele precisará requisitar o serviço \$01, que requisita os parâmetros de leitura, e também precisará requisitar o PID 0x0C, que é o identificador do parâmetro rotação do motor.

A lista completa dos PIDs pode ser consultada no apêndice B da norma ISO 15031-5 (BASTOS, 2012).

2.4.4 Códigos de Diagnóstico de Falha

Os códigos de diagnóstico de falha são gerados quando uma falha é detectada pelo sistema de diagnóstico do veículo. Esses códigos têm o objetivo de facilitar a detecção de falhas no veículo pelos mecânicos, possibilitando um diagnóstico rápido e preciso.

Segundo Gomes (2008) o DTC (*Diagnostic Trouble Code*) ou Código de Diagnóstico de Falha é uma lista que relaciona uma falha do veículo a um código es-

pecífico para o tipo de falha, a leitura dessa lista é feita por um equipamento de diagnóstico acessando a memória de falhas do sistema de diagnóstico da ECU.

Os códigos de diagnóstico de falhas são definidos da seguinte forma, o primeiro caractere determina a qual sistema se refere a falha (Tabela 2.1).

Valor	Valor Binário	Significado
P	00	Motor (<i>Powertrain</i>)
C	01	Chassi (<i>Chassis</i>)
B	10	Corpo (<i>Body</i>)
U	11	Rede (<i>Network</i>)

Tabela 2.1 Primeiro Caractere do Código de Falha (Adaptado de (BELO, 2003))

O segundo caractere indica a entidade responsável pela criação do código (Tabela 2.2).

Valor	Significado
0	ISO/SAE
1	Fabricante do veículo
2	ISO/SAE
3	ISO/SAE

Tabela 2.2 Segundo Caractere do Código de Falha (Adaptado de (BELO, 2003))

O terceiro dígito se refere a um subgrupo do primeiro código (Tabela 2.3).

Valor	Significado
0	Sistema de controle de ar / combustível e emissões
1	Sistema de controle de ar / combustível
2	Sistema de controle de ar / combustível
3	Sistema de ignição
4	Controles de emissão auxiliares
5	Sistema de controle de velocidade
6	Central eletrônica e entrada e saída
7	Transmissão
8	Transmissão
9	Transmissão

Tabela 2.3 Terceiro Caractere do Código de Falha (Adaptado de (BELO, 2003))

Segundo Gomes (2008), o quarto e o quinto caracteres informam problemas relacionados a componentes específicos de cada fabricante.

A figura 2.9 a seguir apresenta um exemplo de um código de falha decodificado (código B1234):

DTC byte 1								DTC byte 0							
\$9				\$2				\$3				\$4			
1	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0
B		1		2				3				4			

Figura 2.9 Exemplo de um código de falha decodificado (Adaptado de (ISO, 2005))

A tabela 2.4 a seguir apresenta exemplos de DTCs.

Número do DTC	Descrição da falha
B3806	Sistema de comutação do farol – curto para terra (GND)
B3793	Sistema de comutação do limpador de pára-brisa – curto para terra (GND)
B2610	Entrada do reostato do painel - resistência muito alta
B3045	Imobilizador do motor – curto para terra (GND)
B2555	Iluminação interna – curto para terra (GND)
B3135	Relé de travamento da porta – contato “colado”
U2103	Erro em módulo eletrônico conectado ao barramento de baixa velocidade
U2105	Perda de comunicação com módulo de controle do motor (ECU)

Tabela 2.4 Exemplos de DTCs (Extraído de (GUIMARÃES, 2007))

2.5 Sistemas de diagnóstico *Off-Board* (Scanner)

Os sistemas de diagnóstico *Off-Board*, conhecidos no mercado automotivo como *scanner* ou *tester*, fazem a *interface* homem-máquina, realizando a leitura de parâmetros da ECU do veículo em tempo real, e a leitura dos códigos de falhas, os chamados DTCs, e disponibilizando essas informações de maneira organizada conforme a solicitação do operador, além de permitir a interação com o sistema para realizar testes, como por exemplo, realizando a abertura de portas, acionamento de vidros, teste de bicos injetores, entre outros (JUNIOR, 2012b).

O equipamento realiza a comunicação com o módulo de diagnóstico realizando testes para encontrar o protocolo de comunicação utilizado pelo veículo. Caso o dispositivo não encontre o protocolo, esse informará ao usuário que não foi possível realizar a comunicação.

Estes dispositivos, como apresentado na figura 2.10 são ferramentas muito importantes para a solução de problemas pelos mecânicos, visto que os veículos hoje em dia possuem uma grande complexidade devido ao aumento da eletrônica embarcada.



Figura 2.10 Equipamento de Diagnóstico (Extraído de (JUNIOR, 2012b))

2.5.1 Funções do *scanner*

Para que o equipamento de diagnóstico *off-board* atenda aos padrões da indústria automotiva, deve ter as seguintes funcionalidades:

- Determinação automática da interface de comunicação para fornecer serviços de diagnóstico;
- Realizar a leitura dos códigos de falha (DTC's) e apresentar ao usuário a descrição deste código de falha. A descrição dos códigos de falha é exclusivamente função do *scanner*, que possui uma biblioteca com todos os códigos de falhas e suas descrições, sendo que a ECU do veículo tem apenas a função de informar o código de falha em si (JUNIOR, 2012b);
- Leitura dos dados de funcionamento;

- *Freeze-frames* – parâmetros de funcionamento do motor no momento em que foi gerado um código de falha, esta funcionalidade é importante para auxiliar na reparação do veículo, pois provê mais informações sobre a falha do que apenas o DTC, estreitando a detecção da possível causa da falha;
- Apagar a memória dos códigos de falha.

Quando o veículo realiza o procedimento de diagnóstico, o operador do equipamento pode alterar alguns parâmetros de funcionamento de certos módulos do veículo - este procedimento é chamado de codificação -, sendo que os ajustes indevidos destes parâmetros podem gerar desconforto ao condutor do veículo, porém o equipamento não pode danificar o funcionamento do veículo (JUNIOR, 2012b).

2.6 Rede CAN

No início dos anos 80 não existia nenhum protocolo de comunicação serial que atendesse requisitos básicos de confiabilidade e segurança na aplicação automotiva. Com o advento da expansão da eletrônica embarcada nos veículos tornou-se necessário um protocolo de comunicação que gerenciasse o funcionamento destes dispositivos inseridos no veículo. Pensando nisso, em 1983 um grupo de engenheiros da Bosch iniciou o desenvolvimento de um protocolo de comunicação com a parceria de duas corporações: a Intel responsável por desenvolver o semicondutor e a Mercedes Benz responsável por realizar a especificação desta nova tecnologia que foi denominada de Rede CAN (BROEDEL e SANTOS, 2011). Protocolo este apresentado pela primeira vez para o mundo automobilístico, no congresso da SAE de Detroit em 1986 (HODEL, 2008).

A principal motivação da criação da Rede CAN era reduzir parâmetros como a quantidade de cabos, peso, custos e principalmente permitir alta velocidade no gerenciamento de módulos de controle, diagnosticando falhas no barramento e realizar o controle prévio destes erros, que classifica esta rede de comunicação como alto grau de confiabilidade em aplicações que envolvem tempo real (HODEL, 2008).

Este protocolo de comunicação é um dos mais utilizados na indústria automobilística devido ser extremamente robusto e com alto nível de confiabilidade na transmissão de informações disponibilizadas na rede. Projetado para um ambiente com

alto nível de ruído induzido e grandes oscilações de alimentação (HODEL, 2008). A rede CAN detém as seguintes características:

- Multimestre;
- Taxa de transmissão de até 1Mbps;
- Integridade dos dados (possui varios métodos de detecção de falha);
- Priorização de mensagem (protocolo orientado à mensagem);
- Detecção e desativação de nós com falhas;
- Flexibilidade de configuração;
- Distinção entre erros temporários e falhas permanentes em nós.

2.6.1 Mensagens do CAN

Segundo Hodel (2008) a rede CAN é um protocolo de comunicação serial síncrono. O sincronismo entre os módulos conectados a rede é realizado ao início de cada mensagem, que é disponibilizada no barramento (evento que ocorre em intervalos de tempo conhecidos e regulares).

Toda informação que trafega pelo barramento CAN é definida por prioridade, para que a mensagem ganhe o barramento, somente a mensagem que tiver maior prioridade terá acesso à rede. A prioridade, conforme a norma é estabelecida pelo campo de identificação da mensagem. Este campo pode ter 11 bits (padrão) ou 29 bits (estendido). O protocolo CAN apresenta diversos padrões para suas mensagens de acordo com a suas aplicações. As mensagens são enviadas em *frames* que são responsáveis por disponibilizar informações, desde quem as enviou até o que está sendo enviado ou recebido.

2.6.1.1 Formato das mensagens CAN

O Protocolo de comunicação CAN utiliza pequenas mensagens, com tamanho máximo de dados de 8 *bytes*. Esta tecnologia não dispõe um endereço explícito na transmissão de dados, sendo através da composição do *frame* que o receptor reconhece a mensagem. Este tipo mecanismo denomina-se de endereçamento orientado ao conteúdo. Ele identifica a variável a ser transmitida e não o endereço onde está a variável (HODEL, 2008).

Atualmente na indústria automobilística existem dois tipos de protocolo CAN: o CAN 2.0A (padrão) que se tornou o padrão ISO 11898-1 possui 11 bits para o ID das mensagens (figura 2.11), consequentemente é possível ter 2048 mensagens e o protocolo CAN 2.0B (estendido) possui 29 bits para o ID (figura 2.12), consequentemente é possível ter aproximadamente 537 milhões de mensagens (HODEL, 2008).

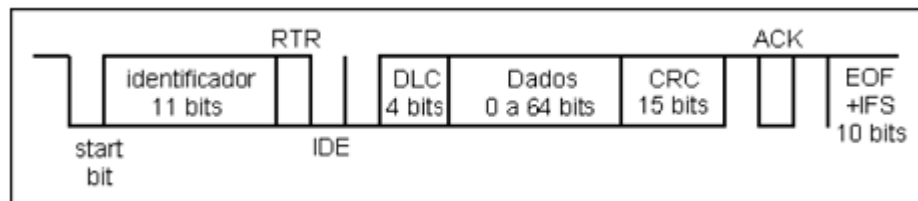


Figura 2.11 *Frame CAN 2.0A* (Extraído de (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003))

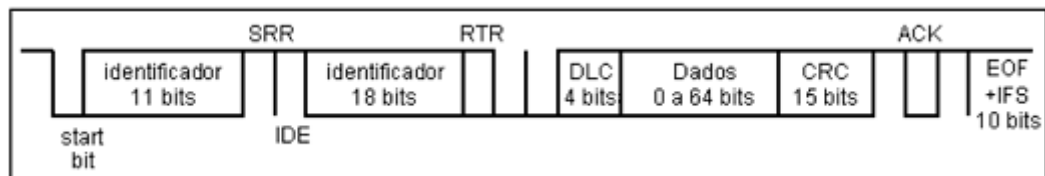


Figura 2.12 *Frame CAN 2.0B* (Extraído de (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003))

2.6.2 Modelo ISO/OSI

Desenvolvido pela ISO em 1983 este o modelo é utilizando como referência para definir as 7 camadas de rede de dados que compõe o modelo ISO/OSI. Sistema este composto pelas camadas física (facilita abstração), enlace (divisão de tarefas), rede (interação entre camadas feitas por serviços), transporte (uma camada apenas toma conhecimento da camada inferior), sessão (comunica-se com camadas de mesmo nível), apresentação (utiliza serviços da camada inferior) e aplicação (presta serviços para a camada superior) conforme apresentados na figura 2.13, o protocolo da rede CAN utiliza apenas os níveis físico e de enlace (HODEL, 2011).

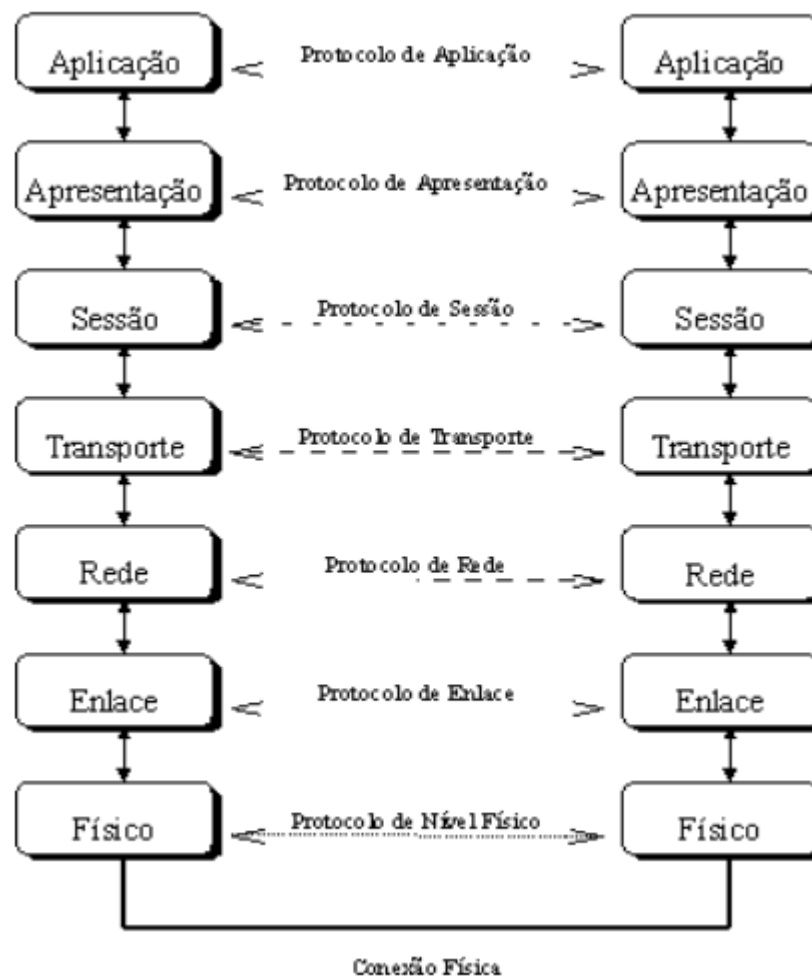


Figura 2.13 As 7 Camadas do modelo ISO/OSI (Extraído de (HODEL, 2011))

2.6.3 Características Físicas

A camada física, dentre as demais camadas do modelo ISO/OSI possui função fundamental na transmissão de sequências de *bits* sobre meio físico isto, porque ela define as interfaces elétrica, mecânica e de sincronização para a rede.

O objetivo da camada física é transmitir um fluxo dados de um meio para outro utilizando um par trançado para a transmissão de dados. Os fios são enrolados de forma helicoidal para evitar interferência eletromagnética conforme figura 2.14, já que um par de fios dispostos paralelamente forma uma antena. Quando os fios são trançados, as ondas de diferentes partes dos fios se cancelam, produzindo um sinal com menor interferência. Devido ao custo e ao desempenho obtidos os pares de trançados são amplamente utilizados (HODEL, 2011).

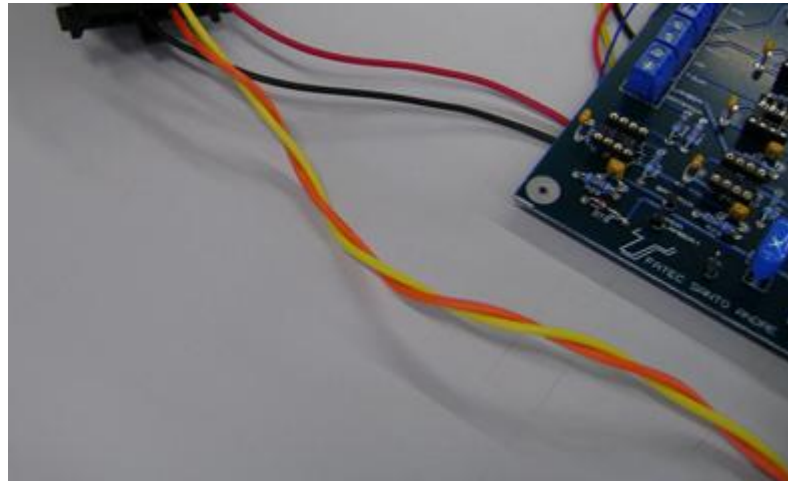


Figura 2.14 Par de fio trançados

O protocolo CAN utiliza o meio de transmissão diferencial, onde é avaliada a diferença de tensão entre dois fios, um fio possui o sinal de *CAN-high* e outro *CAN-low* conforme figura 2.15. Cada extremidade dos fios possui resistores de *pull-up* (120 Ohms), responsáveis por garantir a impedância do barramento necessário para a transmissão de dados.

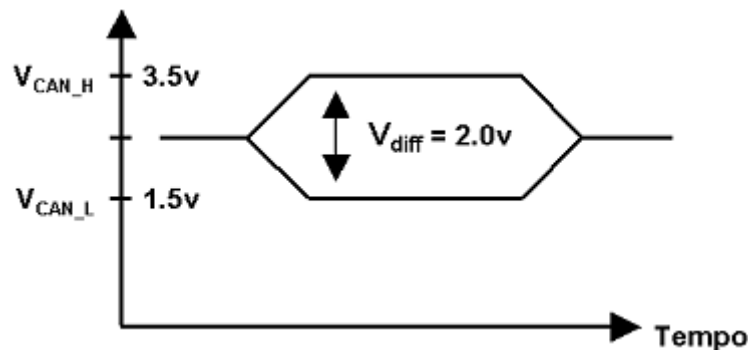


Figura 2.15 Sinal diferencial da rede CAN (Extraído de (AEA – ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA AUTOMOTIVA, 2003))

2.6.4 Diagnóstico sobre a rede CAN (ISO 15765)

A comunicação de diagnóstico ISO 15765 tem como base de transmissão o protocolo CAN. Podem ser utilizadas duas velocidades de transmissão, 250 kbps ou 500 kbps. Pode-se também utilizar o *frame* CAN padrão (ID de 11 *bits*) ou o *frame* CAN estendido (ID de 29 *bits*). O DLC das mensagens deve ser sempre oito *bytes*, mesmo que não sejam utilizados todos os oito *bytes* da mensagem (ISO, 2003).

No caso da utilização do ID de 11 *bits*, para que a comunicação seja efetuada, o equipamento de diagnóstico deve usar o ID 0x7DF em todas as suas mensagens. A ECU do veículo deve usar um ID na faixa de 0x7E8 a 0x7EF (ISO, 2003).

A forma mais simples dessa comunicação é através de *frames* únicos (*single frames*), que consiste na transmissão de dados em apenas um *frame* CAN de no máximo 8 *bytes* de dados. Neste caso, o primeiro *byte* do campo de dados indica quantos *bytes* de dados a seguir dos sete restantes serão utilizados (o valor deste *byte* varia de 1 a 6) (ISO, 2002).

O segundo *byte* indica o número do serviço de diagnóstico solicitado, que varia do serviço \$01 ao serviço \$0A. Essa solicitação é feita pelo equipamento externo de diagnóstico, e a ECU deve responder neste mesmo *byte* o número do serviço requisitado somado do valor 0x40. Por exemplo, se o equipamento solicita o serviço de parâmetro de leitura ele deve enviar o número 0x01 e a ECU deve responder neste *byte* com o número 0x41, ou seja 0x01 (referente ao serviço \$01) mais 0x40 (ISO, 2006).

O terceiro *byte* depende do serviço solicitado. Se o serviço solicitado for o serviço 0x01, o terceiro *byte* corresponde a algum PID que é relacionado a algum parâmetro como rotação, temperatura, entre outros. A ECU deve responder este *byte* com o mesmo número PID solicitado (ISO, 2006).

Os *bytes* seguintes são utilizados para a ECU enviar a resposta com os dados solicitados. Os *bytes* que não são utilizados podem ser zerados.

Na tabela 2.5 é apresentado um exemplo de uma mensagem de solicitação do serviço \$01 PID 0x05 (referente à temperatura do líquido de arrefecimento):

ID	DLC	Campo de Dados (<i>bytes</i>)							
		0	1	2	3	4	5	6	7
0x7DF	8	0x02	0x01	0x05	0x00	0x00	0x00	0x00	0x00

Tabela 2.5 Exemplo de uma mensagem de solicitação ISO 15765

Onde o *byte* 0 é o número de *bytes* utilizados na mensagem, o *byte* 1 é o número do serviço e o *byte* 2 é o número do PID.

Na tabela 2.6 é apresentada a mensagem de resposta à solicitação do exemplo acima:

ID	DLC	Campo de Dados (<i>bytes</i>)							
		0	1	2	3	4	5	6	7
0x7E8	8	0x03	0x41	0x05	A	0x00	0x00	0x00	0x00

Tabela 2.6 Exemplo de uma mensagem de resposta ISO 15765

Onde A no quarto *byte* significa o valor da temperatura do líquido de arrefecimento.

2.7 USART (Transmissor e Receptor universal síncrono e assíncrono)

O dispositivo de comunicação denominado USART é responsável por realizar a comunicação serial onde um *bit* é enviado de cada vez, esta comunicação ocorre de duas formas de operação: síncrona ou assíncrona sendo a última a comunicação a mais aplicada no desenvolvimento de projetos na plataforma de microcontroladores para comunicação com o mundo externo, onde cada *byte* serial é precedido por um *start-bit* de nível lógico baixo e encerrado por um *stop-bit* de nível lógico alto.

Esta interface é aplicada quando se deseja um barramento de comunicação com baixo custo e de baixa velocidade de comunicação, pois seu *hardware* é relativamente simples, necessitando apenas de um único fio (JUCÁ, 2010).

2.8 SPI (Interface Serial Periférica)

O Padrão de comunicação SPI foi desenvolvido pela parceria de duas corporações, Motorola e Freescale. Consiste em um barramento serial que utiliza 4 fios para a transmissão de dados serial síncrona entre diferentes circuitos integrados.

No modo "escravo", o microcontrolador comporta-se como um componente da rede, recebendo o pulso de *clock*. Em modo "mestre", o microcontrolador gera um pulso de *clock*. Não existe um protocolo fixo para o barramento SPI. No entanto, muitos microcontroladores possuem entrada com esta tecnologia. Assim, com a programação do microcontrolador torna-se fácil o gerenciamento de sensores ligados a entrada/saída SPI (JUNIOR, 2012a).

2.9 Sistemas Operacionais de Tempo Real (RTOS)

Um sistema operacional de tempo real é um conjunto de códigos (normalmente chamado de *Kernel*) que controla a execução de tarefas em um microcontrolador quando este opera em uma aplicação multitarefa de tempo real. O RTOS gerencia quais tarefas serão executadas, a prioridade de cada tarefa e também a passagem de dados e mensagens entre as tarefas (TORRES, 2012).

As principais partes de um RTOS são o escalonador, os serviços do RTOS e ferramentas de sincronização e mensagens.

O escalonador é o sistema operacional em si, é o que controla e seleciona qual tarefa irá executar e também qual o momento certo para começar a execução de cada tarefa. Existem três tipos de escalonadores, o escalonador cooperativo, escalonador do tipo *Round Robin* e o escalonador *preemptive* (TORRES, 2012).

No escalonador cooperativo cada tarefa executa todo seu código e retorna o controle do processador para o sistema operacional. É um controle mais simples que os outros tipos de escalonadores e não tem o controle dos tempos de execução de cada tarefa (TORRES, 2012).

No escalonador do tipo *Round Robin* cada tarefa recebe uma porção de tempo para executar, sendo que as tarefas não podem ultrapassar seu limite de tempo, do contrário o sistema operacional retoma o controle do processador e passa para a próxima tarefa. A tarefa que teve seu tempo esgotado é colocada no final da fila de execução e só voltará a executar quando as outras tarefas terminarem de executar (TORRES, 2012).

O escalonador *preemptive* é o mais avançado, ele funciona como os sistemas anteriores, porém é baseado em prioridades de execução e concede a cada tarefa uma prioridade. As tarefas de maior prioridade ganham o controle do processador mesmo que outras tarefas estejam executando. Quando uma tarefa é interrompida por uma tarefa de maior prioridade, o sistema operacional salva o contexto da tarefa e parte para a de maior prioridade, assim logo que esta tarefa termina, o sistema operacional devolve o controle do processador para a tarefa que estava executando anteriormente para que possa terminar sua execução (TORRES, 2012).

Os serviços do RTOS são ferramentas para o gerenciamento das tarefas, que inclui a manipulação de interrupções, relógio do sistema, gerenciamento de disposi-

tivos, memória e entradas e saídas. As ferramentas de sincronização e mensagens são utilizadas para o controle entre tarefas, uma vez que é possível enviar e receber mensagens de uma tarefa para outra e compartilhar recursos como por exemplo a leitura de uma porta analógica (TORRES, 2012).

Atualmente a utilização de sistemas operacionais de tempo real na indústria automotiva tem se tornado cada vez maior, principalmente porque proporciona uma padronização dos sistemas, o que traz muitos benefícios como a redução nos custos de desenvolvimento, melhoras na qualidade de controle do *software* e tem eficiência e confiabilidade (AFONSO e RENESAS, 2012).

Um dos RTOS mais utilizados na indústria automotiva é o OSEK (traduzindo do alemão: Sistemas abertos e suas *interfaces* para eletrônica em veículos automotores), criado por empresas alemãs em 1983. Este sistema operacional foi padronizado pela ISO 17356 e hoje evoluiu para o consórcio AUTOSAR (AFONSO e RENESAS, 2012).

3 METODOLOGIA

Este projeto tem como objetivo desenvolver um sistema de diagnóstico *on-board* em uma plataforma didática de gerenciamento eletrônico. Para a confecção deste sistema de diagnose foi utilizado uma unidade controle eletrônico de motores desenvolvida na FATEC Santo André por Dias (2011).

O *software* do sistema de diagnóstico foi desenvolvido utilizando linguagem C com o compilador PICC da *Costumer Computer Services* (CCS).

Foi utilizado um sistema operacional de tempo real para fazer o gerenciamento das tarefas do sistema de diagnóstico, pois com a inserção de um sistema operacional, chegamos cada vez mais à realidade da indústria automotiva que utiliza sistemas operacionais de tempo real em suas aplicações embarcadas, devido à confiabilidade e robustez desses tipos de sistemas, facilitando o desenvolvimento, reduzindo o tempo de projeto e a complexidade pois aproveita melhor os recursos do *hardware* e o tempo de execução dos processos (TEMPO REAL EVENTOS, 2010). O RTOS utilizado neste projeto foi o CCS RTOS que consiste em um sistema operacional cooperativo multitarefa. Assim, podemos separar uma tarefa para cada função.

A comunicação de diagnóstico utilizada foi a ISO 15765, pois esse padrão é baseado no protocolo CAN que é atualmente um dos protocolos mais utilizados nos veículos por possuir confiabilidade na transmissão e recepção de dados, robustez na comunicação de dados (menos susceptível a interferências) e maior velocidade de operação se comparado com outros sistemas de comunicação automotivos.

3.1 *Hardware* Utilizado

A unidade de controle eletrônico utilizada tem uma concepção didática e apresenta condições mínimas para que se possa efetuar o gerenciamento de motores (DIAS, 2011).

A ECU tem 3 blocos básicos de processamento, sendo eles, respectivamente, o bloco de gerenciamento, o bloco de sincronismo e o bloco de comunicação e diag-

nose. Todos os blocos utilizam um microcontrolador PIC18f452 da Microchip, com *clock* de entrada de 20MHz.

O bloco de gerenciamento é o principal, ele recebe os sinais dos principais sensores existentes em um motor de combustão interna, assim, o gerenciamento define o tempo de injeção e o avanço de ignição com base nas leituras dos sensores. O gerenciamento envia os dados para o bloco de sincronismo através de um protocolo SPI, e envia para o bloco de comunicação através de um protocolo U-SART. O bloco de sincronismo é o responsável por atuar os injetores e as bobinas de ignição no tempo certo, sincronizado com a rotação do motor. O bloco de comunicação é o responsável por disponibilizar as informações através da rede CAN ou até mesmo em um *display* de cristal líquido (DIAS, 2011). A figura 3.1 a seguir apresenta o diagrama de blocos da ECU:

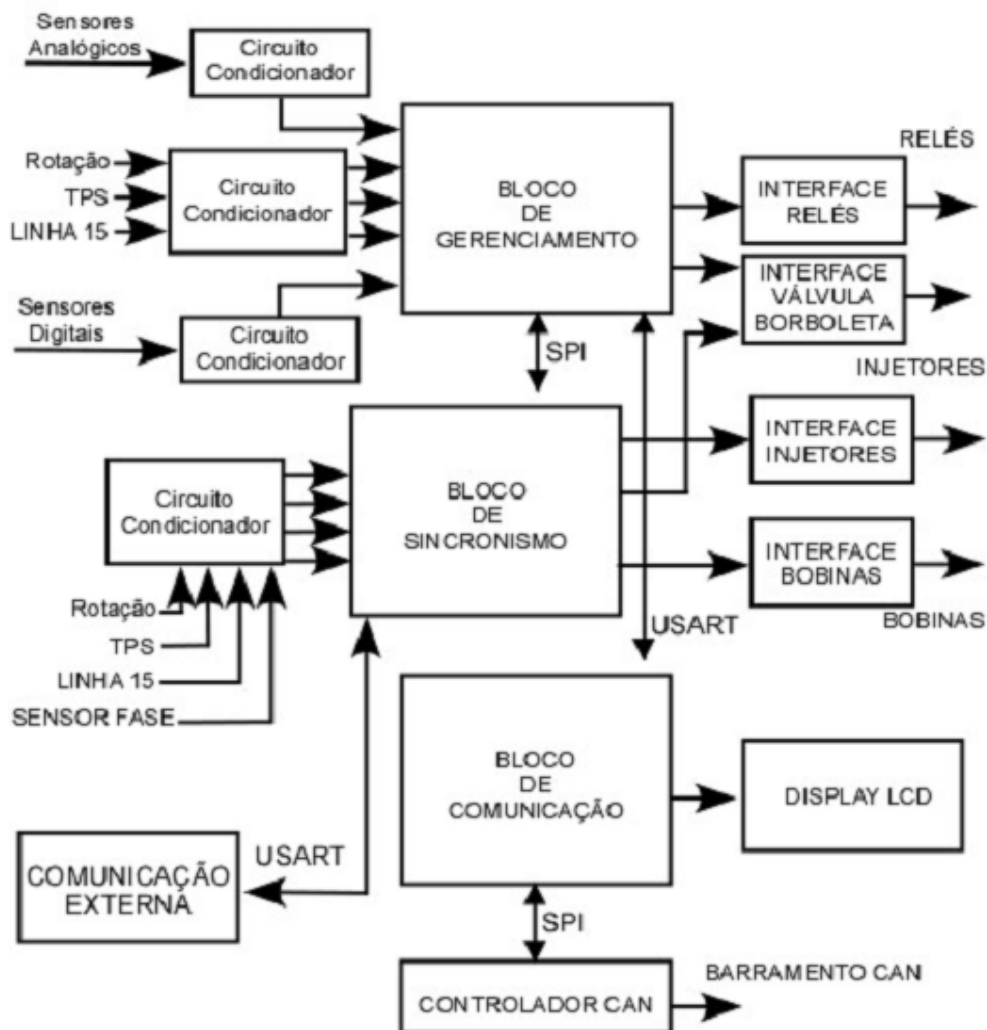


Figura 3.1 Diagrama de Blocos da ECU (Extraído de (DIAS, 2011))

Como o objetivo deste projeto é desenvolver um sistema de diagnóstico, foram utilizados os seguintes dispositivos conforme figura 3.2:

- O microcontrolador de Gerenciamento;
- O microcontrolador de Comunicação (também chamado de microcontrolador de diagnose);
- O controlador CAN e o *transceiver* CAN.

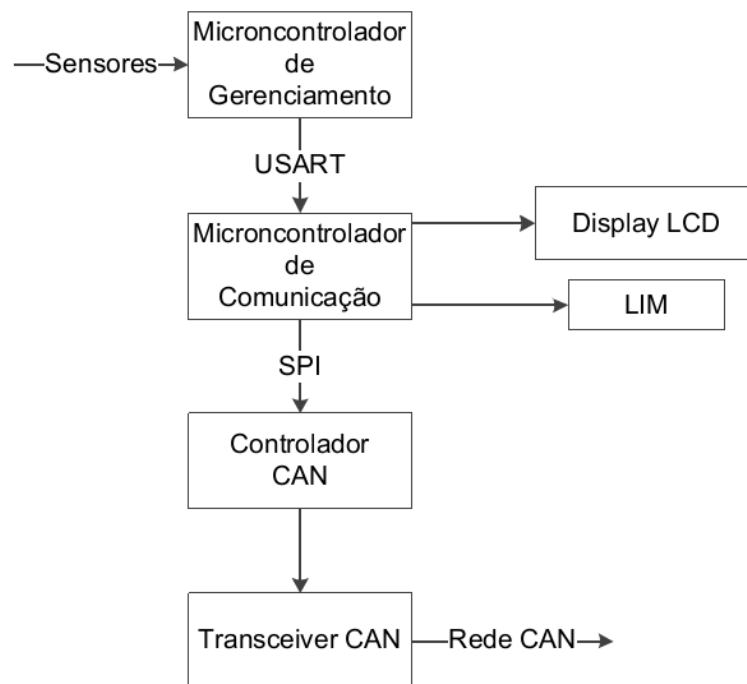


Figura 3.2 Dispositivos utilizados neste projeto

O microcontrolador de gerenciamento envia os parâmetros de funcionamento do motor lidos pelos sensores para o microcontrolador de comunicação via UART, obtendo esses dados, o bloco de comunicação fica responsável por realizar o diagnóstico desses parâmetros

O bloco de diagnose faz a comunicação com o controlador CAN MCP2515 via SPI, responsável por formar o *frame* de dados, assim, este controlador envia o *frame* de dados para o *transceiver* CAN PCA82C251 que realiza as conversões de tensões para as linhas CAN-*High* e CAN-*Low*, possibilitando o acesso à rede CAN.

Para mais detalhes sobre essa unidade de controle recomenda-se a consulta de Dias (2011).

3.2 Desenvolvimento do Software

Antes de desenvolver o *software* do sistema de diagnose foi necessário desenvolver um pequeno algoritmo para a simulação parâmetros do motor no microcontrolador de gerenciamento. Foram simulados seis parâmetros do motor:

- Temperatura do ar de admissão;
- Temperatura do líquido de arrefecimento;
- Rotação do motor;
- Pressão no coletor de admissão;
- Posição da válvula borboleta (TPS);
- Tensão da sonda *lambda*.

Com esses parâmetros foram simulados três condições de funcionamento de um motor, marcha lenta, carga parcial e plena carga. Todas essas condições de funcionamento são influenciadas de acordo com a variação do sinal do sensor TPS (implementado por um potenciômetro).

Esses parâmetros simulados são enviados ao microcontrolador de comunicação para a realização do diagnóstico via protocolo serial, que os recebe em uma interrupção de chegada de dados na entrada serial do microcontrolador e os guarda em uma união entre uma estrutura de dados com todos os parâmetros do motor e um vetor para o recebimento dos dados pela serial, conforme apresentado no fluxograma da figura 3.3 a seguir.

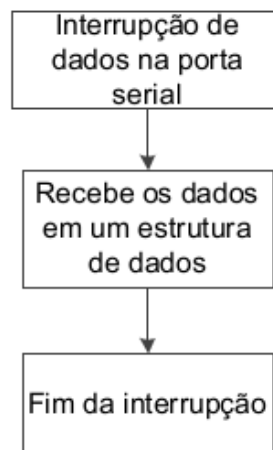


Figura 3.3 Fluxograma da interrupção de chegada de dados na porta serial

O programa inicia com as configurações de *hardware* e periféricos. Dentre os periféricos está a configuração do controlador CAN.

No controlador CAN é configurado os modos de sincronismo da rede CAN. Foi configurado uma taxa de transmissão de 500KHz, o DLC das mensagens com 8 *bytes* de tamanho, e utilizamos o padrão CAN 2A, que consiste em um ID de 11 *bits*. O ID foi configurado para 0x7E8 conforme o protocolo ISO 15765.

É também iniciada a memória de falhas para verificar se há algum código de falha armazenado anteriormente.

Após estas configurações, o controle do programa é entregue ao RTOS que gerencia as tarefas que serão executadas cada uma a seu tempo.

O programa é dividido em três tarefas principais:

- Tarefa de diagnóstico de falhas – Executa a cada 500ms para verificar o funcionamento de parâmetros;
- Tarefa de conversão de parâmetros – Executa a cada 250ms para fazer a conversão dos parâmetros;
- Tarefa de comunicação – Executa a cada 100ms para realizar a comunicação.

Essas tarefas serão detalhadas nos tópicos seguintes.

3.2.1 Tarefa de Diagnóstico de Falhas

Para este projeto foi implementado apenas dois códigos de falha por motivos de limitação na comunicação, que demandaria desenvolver uma estrutura de comunicação de envio de múltiplos *frames* conforme ISO 15765.

As falhas implementadas são referentes ao sensor TPS. Para este sensor, admitimos que este trabalha em uma faixa de operação que vai de 1 Volt a 4 Volts (figura 3.4).

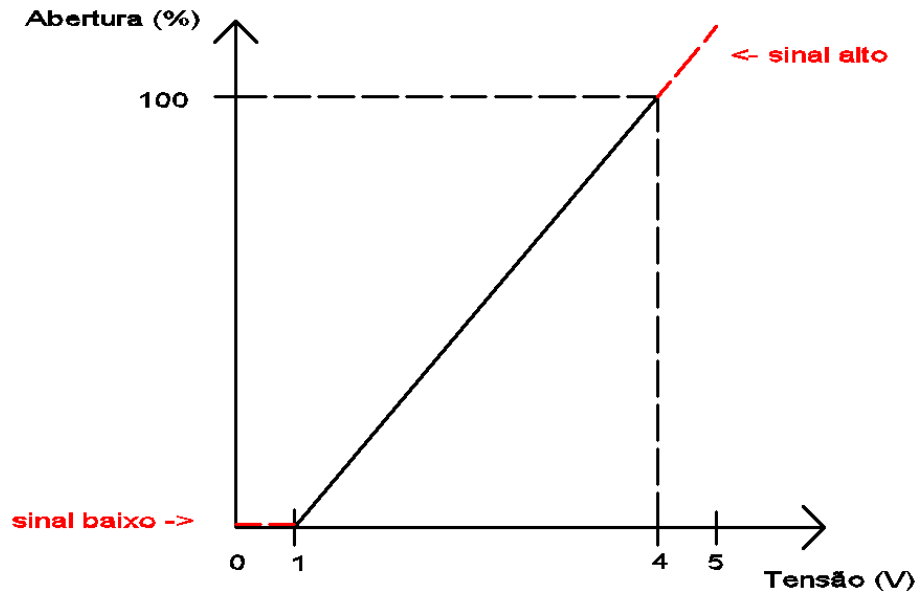


Figura 3.4 Faixa de operação do sinal do sensor de posição da borboleta

Assim, se for detectado uma tensão menor do que 1 Volt, o programa armazena o código de falha P0122 que corresponde a uma tensão baixa do sensor de posição da borboleta, conforme ISO 15031-6. Se, o sinal for maior que 4 Volts, o programa armazena o código de falha P0123 que corresponde a uma tensão alta deste mesmo sensor. Se detectado qualquer destas condições de falhas citadas anteriormente, o programa ilumina a lâmpada indicadora de mau-funcionamento (LIM). A figura 3.5 a seguir apresenta o fluxograma desta tarefa:

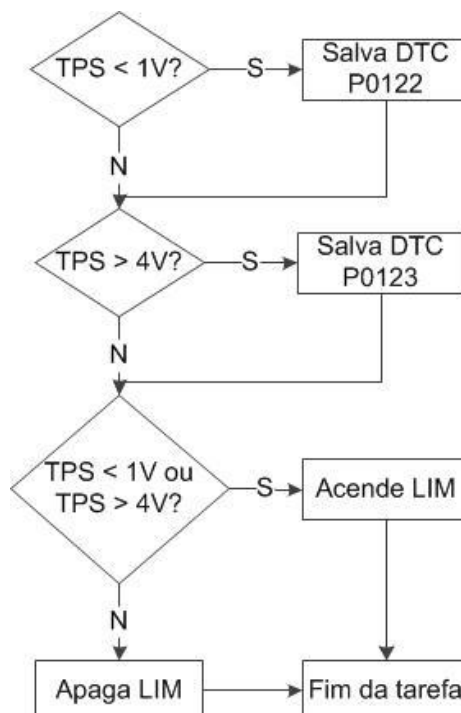


Figura 3.5 Fluxograma da tarefa de diagnóstico

As falhas são armazenadas na memória EEPROM do microcontrolador. Cada código de falha ocupa dois *bytes* da memória. A cada código detectado é incrementado uma variável com o número de falhas presentes que também é salvo na memória.

3.2.2 Tarefa de Conversão de Parâmetros

A maioria dos sinais são aquisitados na forma de tensão, mas, para que o equipamento de diagnose *off-board* interprete os parâmetros corretamente se faz necessário uma rotina de conversão de parâmetros. Os sinais de tensão são convertidos em grandezas tais como temperatura em graus Celsius, abertura de posição da borboleta em porcentagem, rotação em RPM, entre outros. Além disso os parâmetros devem corresponder à faixa de valores especificados na ISO 15031-5. A figura 3.6 a seguir apresenta o fluxograma desta tarefa.

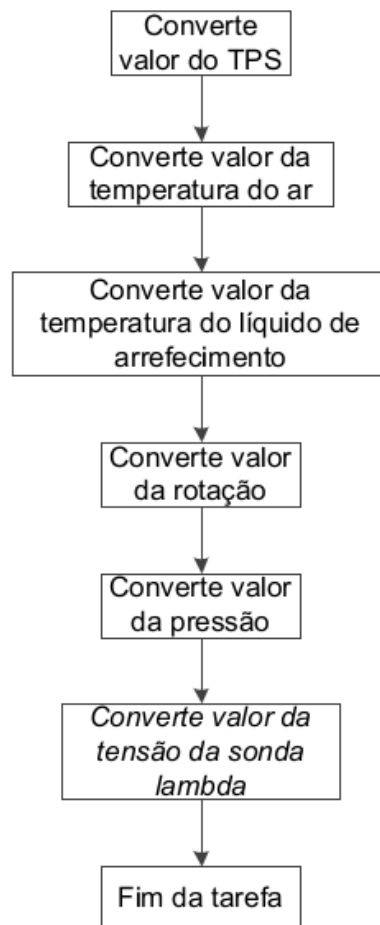


Figura 3.6 Fluxograma da tarefa de conversão de parâmetros

Cada conversão de parâmetros utilizado neste projeto será detalhado a seguir.

3.2.2.1 Conversão dos valores de temperatura

Os valores de temperatura do ar de admissão e temperatura do líquido de arrefecimento tem apenas 1 *byte* e varia de -40°C a 215°C, em uma escala de 1°C por *bit*, com um *offset* de -40°C. Assim, se o valor da variável convertida for de 0, corresponde a -40°C, e se for 255, corresponde a 215°C. Por exemplo, se o valor da temperatura for 18°C, o seu valor correspondente convertido será de:

$$temp.convertida = 18^{\circ} + 40^{\circ} = 58 \quad 1$$

O valor convertido é interpretado pelo *scanner* como 18° de acordo com a norma ISO 15031-5, apresentado na tabela 3.1 a seguir.

Descrição	Bytes	Valor min.	Valor máx.	Escala por <i>bit</i>
Temperaturas	1	-40°C	215°C	1°C por <i>bit</i> com <i>offset</i> de -40°C

Tabela 3.1 Conversão da temperatura (Adaptado de (ISO, 2006))

3.2.2.2 Conversão do valor da rotação do motor

O valor da rotação corresponde a dois *bytes* e varia de 0 RPM a 16383,75 RPM em uma escala de 0,25 RPM por *bit*. Logo se o valor da variável convertida for 0, o valor da rotação é de 0 RPM, e se for 65535 corresponde a 16383,75 RPM. Por exemplo, se o valor da rotação for de 800 RPM, o respectivo valor da variável convertida será de:

$$rotacao.convertida = 800 \text{ RPM} * 4 = 3200 \quad 2$$

O valor convertido é interpretado pelo *scanner* como 800 RPM de acordo com a norma ISO 15031-5, apresentado na tabela 3.2 a seguir.

Descrição	Bytes	Valor min.	Valor máx.	Escala por <i>bit</i>
Rotação	2	0 RPM	16383,75 RPM	0,25 RPM por <i>bit</i>

Tabela 3.2 Conversão da rotação (Adaptado de (ISO, 2006))

3.2.2.3 Conversão do valor de posição da válvula borboleta

O valor do sensor de posição da válvula borboleta é convertido em porcentagem de abertura da válvula. Este valor tem um *byte* e varia de 0% a 100% em uma escala de 100/255 por *bit*. Isto é, se o valor da variável convertida for de 0 corresponde a 0% e se for 255 corresponde a 100%. Por exemplo, se o valor de abertura da borboleta for de 50%, o valor convertido será de:

$$tps.convertido = 50/(100/255) \cong 127 \quad 3$$

O valor convertido é interpretado pelo *scanner* como aproximadamente 50% de abertura, conforme tabela 3.3.

Descrição	Bytes	Valor min.	Valor máx.	Escala por <i>bit</i>
Porcentagem de abertura da válvula borboleta	1	0%	100%	100/255 por <i>bit</i>

Tabela 3.3 Conversão do valor da abertura da válvula borboleta (Adaptado de (ISO, 2006))

3.2.2.4 Conversão do valor de pressão no coletor de admissão

O valor do sensor de pressão tem um *byte* e varia de 0KPa a 255KPa em uma escala de 1KPa por *bit*. Assim, se o valor da variável convertida for de 0 corresponde a 0KPa e se for de 255 corresponde a 255KPa. Por exemplo, se o valor da pressão for de 100KPa, o valor da variável convertida será de:

$$pressão.convertida = 100KPa = 100 \quad 4$$

O valor é interpretado pelo equipamento de diagnóstico como 100KPa de acordo com a tabela 3.4.

Descrição	Bytes	Valor min.	Valor máx.	Escala por <i>bit</i>
Pressão absoluta no coletor de admissão	1	0KPa	255KPa	1KPa por <i>bit</i>

Tabela 3.4 Conversão do valor de pressão absoluta no coletor de admissão (Adaptado de (ISO, 2006))

3.2.2.5 Conversão do valor de tensão da sonda *lambda*

O valor de tensão da sonda *lambda* tem um *byte* e varia de 0V a 1,275V em uma escala de 0,005V por *bit*. Assim, se o valor da variável convertida for de 0 corresponde a 0V e se for de 255 corresponde a 1,275V. Por exemplo, se o valor de tensão for de 1V, o valor da variável convertida será de:

$$tensãoLambda.convertido = 1V / 0,005 = 200 \quad 5$$

Este valor é interpretado pelo *scanner* como 1V de acordo com a tabela 3.5:

Descrição	Bytes	Valor min.	Valor máx.	Escala por <i>bit</i>
Pressão absoluta no coletor de admissão	1	0KPa	255KPa	1KPa por <i>bit</i>

Tabela 3.5 Conversão do valor de tensão da sonda *lambda* (Adaptado de (ISO, 2006))

Feito todas as conversões dos valores dos sinais de sensores, o programa pode se comunicar com um equipamento de diagnóstico para enviar os parâmetros que assim poderão ser apresentados na tela do equipamento.

3.2.3 Tarefa de Comunicação

Para se comunicar com um equipamento de diagnóstico, a ECU precisa seguir um protocolo de comunicação. O protocolo escolhido neste projeto foi o ISO 15765. Os serviços implementados neste projeto foram os serviços \$01 (solicita os parâmetros de leitura do motor), \$03 (solicita os códigos de falhas armazenados na memória) e \$04 (apaga a memória de falhas).

Para que o programa entre em cada um destes serviços, a tarefa fica aguardando o recebimento de uma mensagem no barramento CAN. Quando a mensagem chega, o programa lê o *byte* referente ao serviço solicitado e se caso for algum dos serviços implementados, o *software* executa as funções referentes a este serviço, conforme o fluxograma da figura 3.7 a seguir:

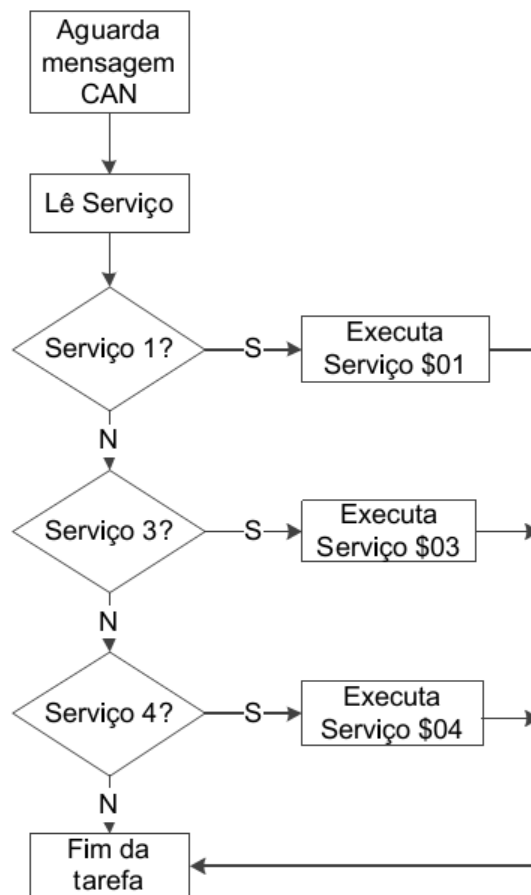


Figura 3.7 Fluxograma da tarefa de comunicação de diagnóstico

3.2.3.1 Estabelecimento da Comunicação

Para que a comunicação inicie, o equipamento de diagnóstico deve enviar uma mensagem com o serviço \$01 com o PID 0x00 (solicita a informação de quais PIDs na faixa de PIDs entre 0x01 e 0x20 estão disponíveis na ECU) (ISO, 2003). O programa responde a mensagem com os PIDs disponíveis neste projeto, que são os PIDS 0x00, 0x01, 0x05, 0x0B, 0x0C, 0x0F, 0x11, 0x13 e 0x14. A mensagem é codificada conforme tabela 3.6:

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x06	0x41	0x00	0b00001000	0b00110010	0b10110000	0b00000000	0x00

Tabela 3.6 Resposta da solicitação do PID 0x00

Cada *bit* em nível lógico 1 nos *bytes* 3, 4, 5 e 6 correspondem aos PIDs disponíveis na faixa de PIDs entre 0x01 e 0x20, a contagem começa no *bit* 7 do *byte* 3 com o PID 0x01 até o *bit* 0 do *byte* 6 com o PID 0x20.

Com a comunicação estabelecida o usuário do equipamento pode selecionar os serviços \$01, \$03, ou \$04.

3.2.3.2 Serviço \$01

Dentro do serviço \$01 o usuário do equipamento pode selecionar os PIDs que correspondem aos parâmetros de leitura que ele deseja observar. Quando o programa entra neste serviço, ele analisa, lendo o *byte* referente ao recebimento dos PIDs, qual foi o PID solicitado, e se caso for algum dos PIDs implementados, é executado a função deste PID, conforme fluxograma da figura 3.8 a seguir:

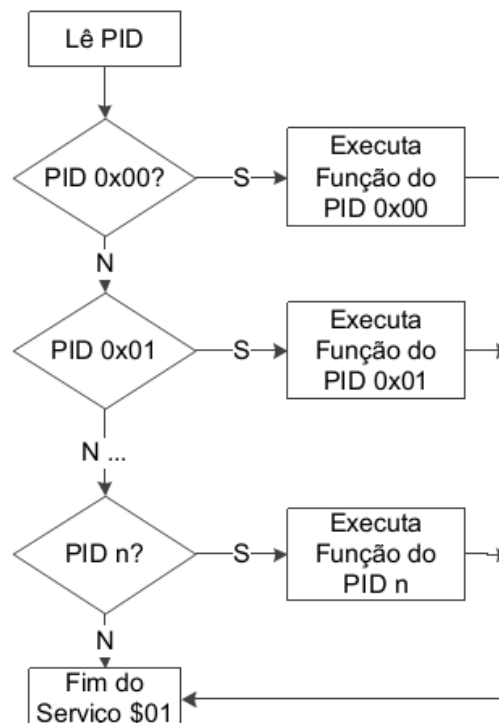


Figura 3.8 Fluxograma da função do serviço \$01

Como exposto anteriormente, foram implementados 6 parâmetros de leitura que serão detalhados nos tópicos a seguir. Vale ressaltar que todos os parâmetros enviados são os parâmetros devidamente convertidos nos padrões ISO (ver tópico 3.2.2).

3.2.3.2.1 PID 0x05 – Temperatura do líquido de arrefecimento

O parâmetro de leitura da temperatura do líquido de arrefecimento é requisitado pelo serviço \$01 com o PID 0x05. O programa responde à solicitação em uma mensagem (tabela 3.7) com o valor da temperatura.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x03	0x41	0x05	temperatura	0x00	0x00	0x00	0x00

Tabela 3.7 Mensagem com o valor da temperatura do líquido de arrefecimento

3.2.3.2.2 PID 0x0B – Pressão absoluta no coletor de admissão

Este parâmetro é solicitado pelo serviço \$01 com o PID 0x0B. O programa responde a essa solicitação em uma mensagem (tabela 3.8) com o valor da pressão.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x03	0x41	0x0B	pressão	0x00	0x00	0x00	0x00

Tabela 3.8 Mensagem com o valor da pressão absoluta no coletor de admissão

3.2.3.2.3 PID 0x0C – Rotação do motor

Este parâmetro é solicitado pelo serviço \$01 com o PID 0x0C. O programa responde a solicitação em uma mensagem (tabela 3.9) com o valor da rotação do motor. O valor de rotação do motor retorna dois *bytes* de dados.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x04	0x41	0x0B	rotação <i>byte</i> alto	rotação <i>byte</i> baixo	0x00	0x00	0x00

Tabela 3.9 Mensagem com o valor da rotação do motor

3.2.3.2.4 PID 0x0F – Temperatura do ar de admissão

O valor da temperatura do ar de admissão é solicitado pelo serviço \$01 com o PID 0x0F. O programa responde esta solicitação em uma mensagem (tabela 3.10) com o valor da temperatura do ar de admissão.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x03	0x41	0x0F	temperatura	0x00	0x00	0x00	0x00

Tabela 3.10 Mensagem com o valor da temperatura do ar de admissão

3.2.3.2.5 PID 0x11 – Posição da válvula borboleta

O valor da posição da válvula borboleta requisitado pelo serviço \$01 PID 0x11. O programa responde a solicitação em uma mensagem com o valor da posição da válvula borboleta conforme a tabela 3.11 a seguir:

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x03	0x41	0x0F	posição válvula borboleta	0x00	0x00	0x00	0x00

Tabela 3.11 Mensagem com o valor da posição da válvula borboleta

3.2.3.2.6 PID 0x14 – Tensão da sonda *lambda*

Para requisitar o valor da tensão da sonda *lambda*, o equipamento de diagnóstico primeiro requisita o PID 0x13. O PID 0x13 retorna um *byte* e indica quais e quantas sondas *lambda*s existem no veículo. Como existe apenas uma sonda *lambda* neste projeto foi colocado em nível lógico 1 apenas o *bit* 7 do *byte* 3 que representa a sonda *lambda* 1 banco 1, conforme ISO 15031-5. A tabela 3.12 a seguir apresenta a mensagem de resposta a este PID:

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x03	0x41	0x13	0b10000000	0x00	0x00	0x00	0x00

Tabela 3.12 Mensagem com a informação de sondas *lambda*s presentes

Com a informação de quais sondas estão presentes, o equipamento de diagnóstico requisita o PID 0x14 com o valor da tensão da sonda *lambda*. O programa responde a solicitação em uma mensagem (tabela 3.13) com o valor da tensão da sonda *lambda*.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x03	0x41	0x14	tensão sonda <i>lambda</i>	0x00	0x00	0x00	0x00

Tabela 3.13 Mensagem com o valor da tensão da sonda *lambda*

3.2.3.3 Serviço \$03

O serviço \$03 possibilita informar quantos e quais DTCs estão presentes na memória de falhas da ECU. O equipamento de diagnóstico requisita este serviço enviando primeiro uma mensagem (tabela 3.14) com o serviço \$01 PID 0x01, responsável por informar quantos códigos de falhas estão armazenados. Além disso, este PID informa se a lâmpada indicadora de mau-funcionamento está iluminada e também o estado dos monitores de emissões.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x06	0x41	0x01	nº de falhas + estado da LIM	0x00	0x00	0x00	0x00

Tabela 3.14 Mensagem com o valor do número de falhas presentes

Com essa informação, o equipamento envia uma mensagem com serviço \$03. O programa responde a mensagem do serviço \$03 enviando o número de DTCs e em seguida os códigos de falha em si. A tabela 3.15 a seguir apresenta o envio da mensagem de resposta do serviço \$03:

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x06	0x43	nº de falhas	DTC 1 <i>byte</i> 1	DTC 1 <i>byte</i> 0	DTC 2 <i>byte</i> 1	DTC 2 <i>byte</i> 0	0x00

Tabela 3.15 Mensagem de resposta com os códigos de falha

O *byte* 2 indica qual a quantidade de falhas presentes e os DTCs do *byte* 3 e 4 e do *byte* 5 e 6 são os DTCs que são lidos da memória de falhas.

3.2.3.4 Serviço \$04

O serviço \$04 é o responsável por solicitar a limpeza da memória de falhas. Para solicitar este serviço o equipamento de diagnóstico envia uma mensagem com apenas o número deste serviço e a ECU deve responder também apenas com uma mensagem (tabela 3.16) com o número do serviço como confirmação de que o serviço foi aceito.

Campo de Dados (<i>bytes</i>)							
0	1	2	3	4	5	6	7
0x01	0x44	0x00	0x00	0x00	0x00	0x00	0x00

Tabela 3.16 Mensagem de confirmação do serviço \$04

Quando este serviço é requisitado, a ECU responde com a confirmação e apaga toda a memória de falhas que fica na memória EEPROM do microcontrolador. O programa também apaga as variáveis de controle das falhas.

4 RESULTADOS OBTIDOS

Para a análise do sistema desenvolvido neste projeto, foi utilizado um equipamento de diagnóstico *off-board* para realizar a comunicação e fazer a *interface* do sistema *on-board*, sendo possível coletar informações de diagnóstico do módulo eletrônico do motor. Através da interface amigável do scanner é possível requisitar informações específicas de funcionamento do motor ,através dos menus de serviços tendo como objetivo facilitar o diagnostico rápido com confiabilidade nas informações aqisitadas, além de não precisar realizar a desmontagem de componentes, auxiliando o reparador na manutenção do veículo.

A figura 4.1 apresenta o projeto e sua ligação com o equipamento de diagnóstico pelo conector OBD-2.

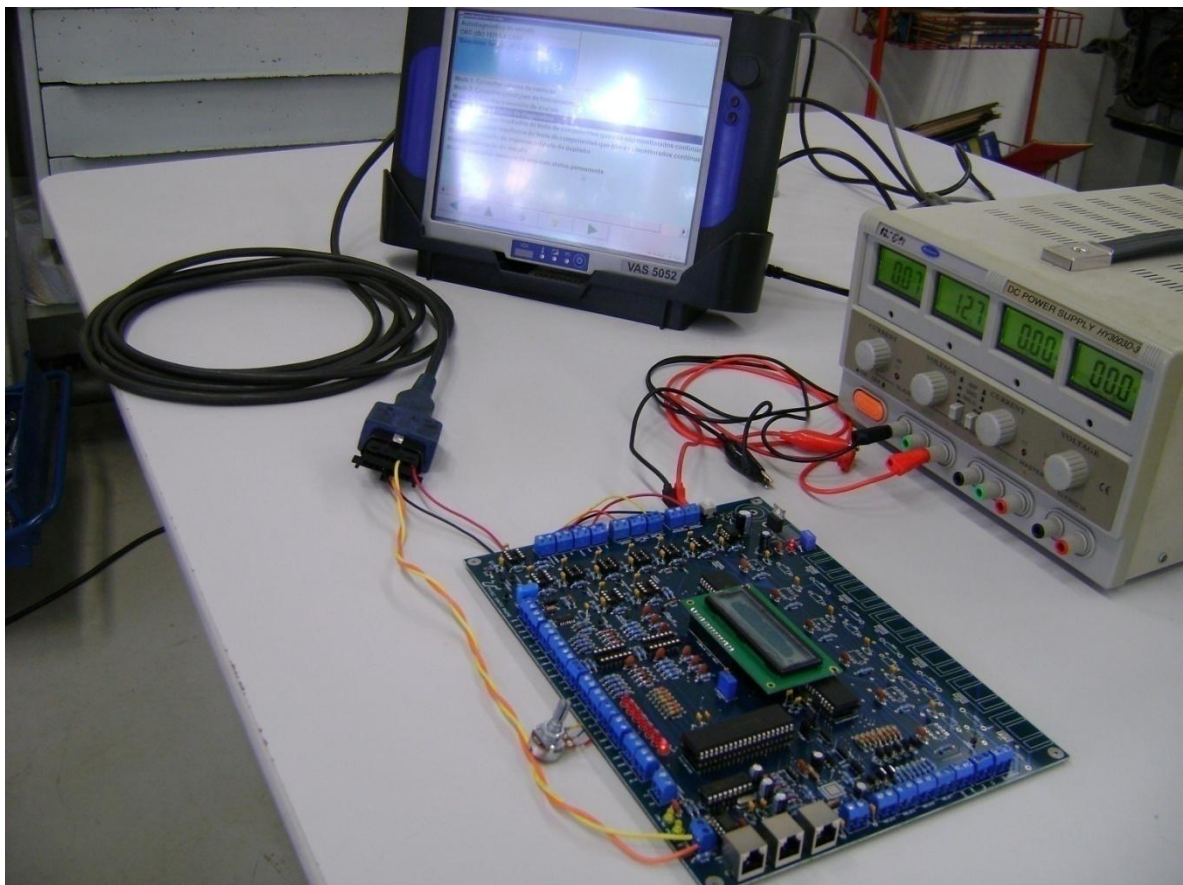


Figura 4.1 Projeto conectado ao equipamento de diagnóstico pelo conector OBD

O equipamento utilizado para os testes foi o VAS5052. O VAS5052 (figura 4.2) é um equipamento de diagnóstico *off-board* desenvolvido pela Audi Volkswagen.

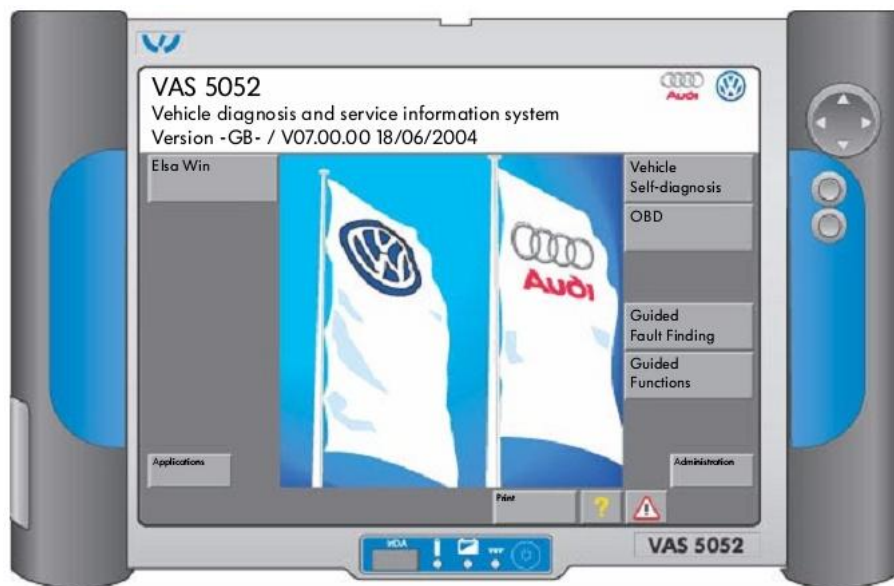


Figura 4.2 Equipamento de Diagnóstico VAS5052 (Extraído de (AUDI VOLKSWAGEN, 2005))

Este equipamento possui funções de auto-diagnóstico, funções OBD, e funções guiadas para detecção de falhas. Para este projeto utilizaremos as funções OBD que são padronizadas pela ISO.

4.1 Testes Realizados

Para comprovar a funcionalidade deste projeto foram realizados alguns testes:

- Diagnóstico de falhas do sensor TPS;
- Interação com o equipamento de diagnóstico (estabelecimento de comunicação);
- Utilização dos serviços OBD-2.

4.1.1 Diagnóstico de falhas do sensor TPS

Para simular uma condição de falha no funcionamento do motor foi adotado um potenciômetro como sensor TPS, que quando sai de sua faixa de operação (1V a 4V) gera uma condição de mau-funcionamento. Essa condição determina que a LIM acenda indicando que há falhas nos sistemas responsáveis pelo controle de e-

missões, que é o caso deste sensor. A figura 4.3 a seguir apresenta o sensor TPS funcionando fora da faixa de operação com o *led* da LIM iluminada.

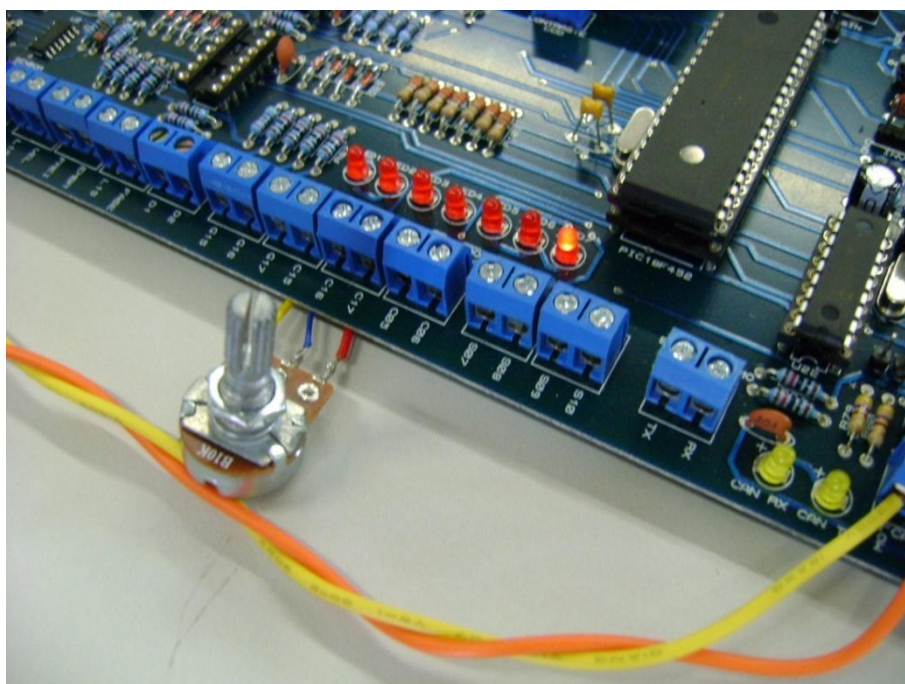


Figura 4.3 Led da lâmpada indicadora de mau-funcionamento iluminada

Quando o TPS está dentro de sua faixa de operação, a LIM permanece apagada, conforme figura 4.4.

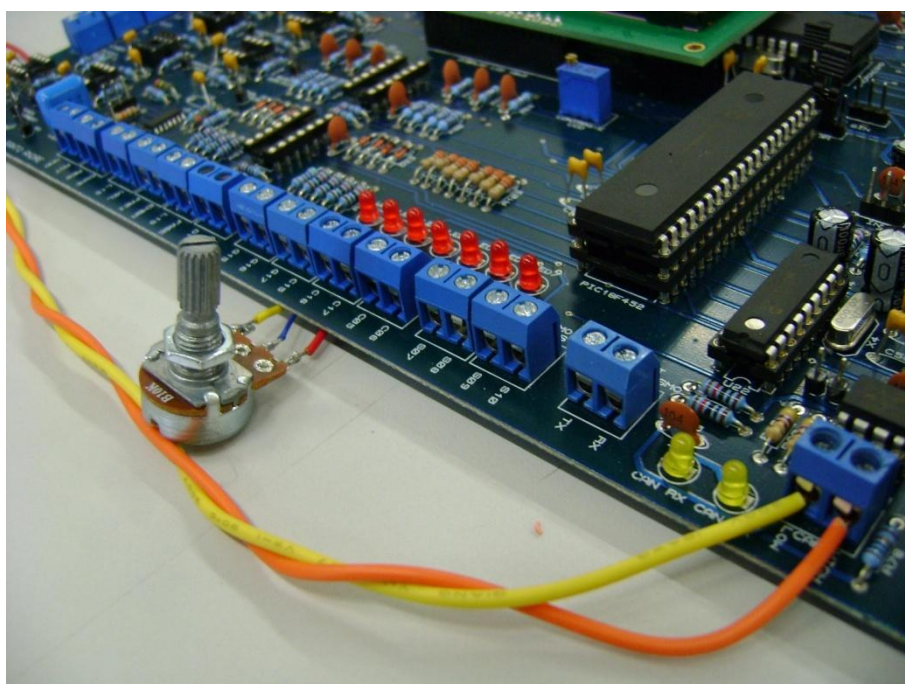


Figura 4.4 Led da lâmpada indicadora de mau-funcionamento apagada

4.1.2 Interação com o equipamento de diagnóstico

Para realizar a interação, é utilizada a função OBD-2 do equipamento de diagnóstico. Quando acessado esta função, o equipamento de diagnóstico requisita o estabelecimento de comunicação entre os módulos. Assim, quando identificado o protocolo de comunicação é disponibilizado os serviços de diagnóstico, conforme apresentado na figura 4.5.

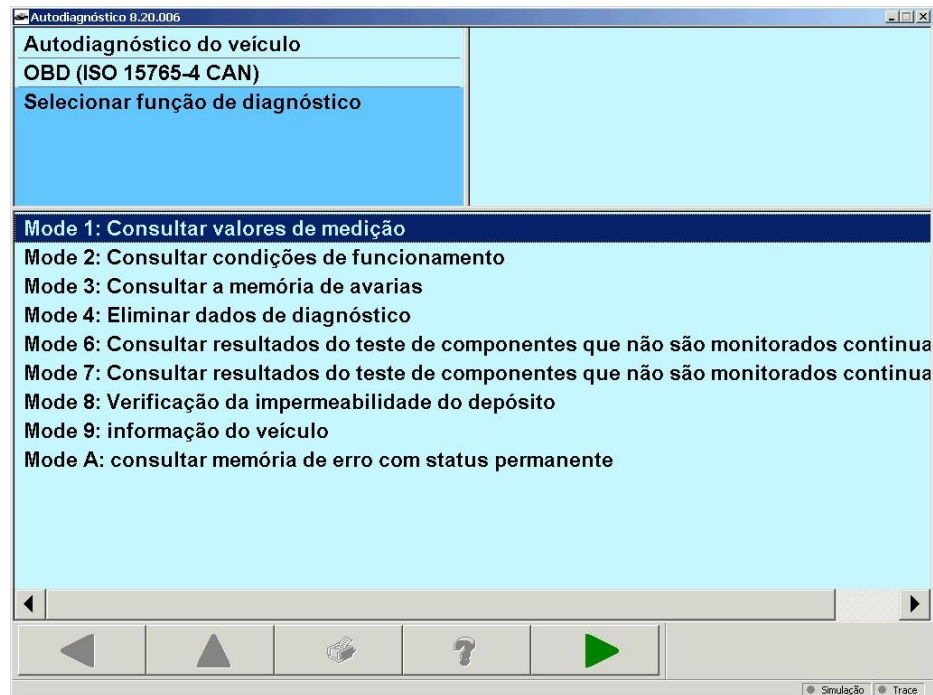


Figura 4.5 Menu de serviços e identificação do protocolo de comunicação

É também apresentado no display de LCD uma mensagem (figura 4.6) que indica que está ocorrendo uma comunicação.



Figura 4.6 Mensagem indicadora de comunicação no *display*

4.1.3 Utilização dos serviços OBD-2

Com os serviços disponíveis é possível acessar os três serviços implementados.

Selecionando o serviço \$01 é possível acessar os parâmetros de funcionamento do motor em tempo real, conforme figuras 4.7 e 4.8.

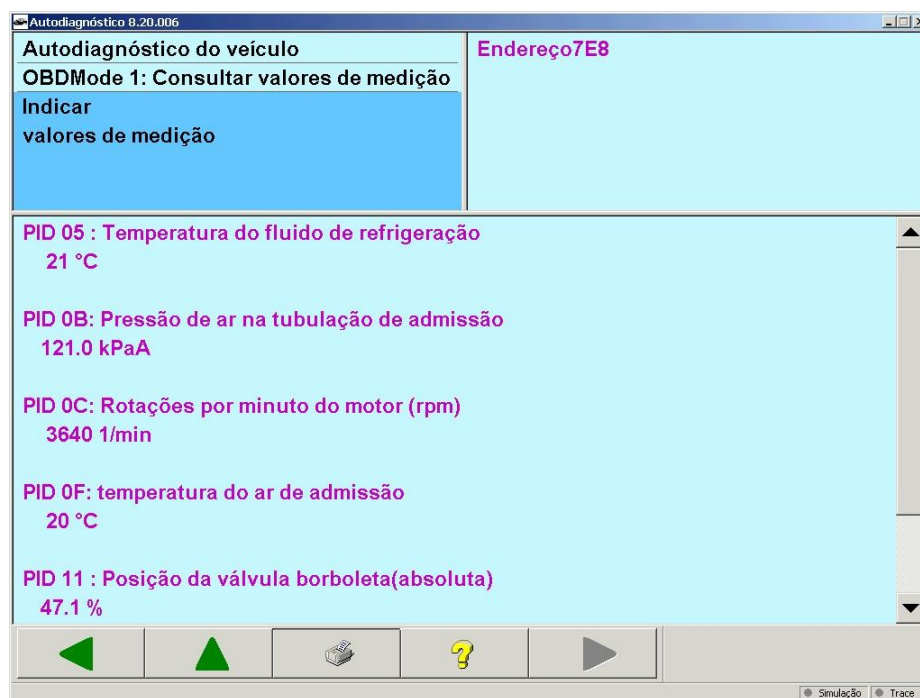


Figura 4.7 Primeira tela dos parâmetros de leitura no serviço \$01

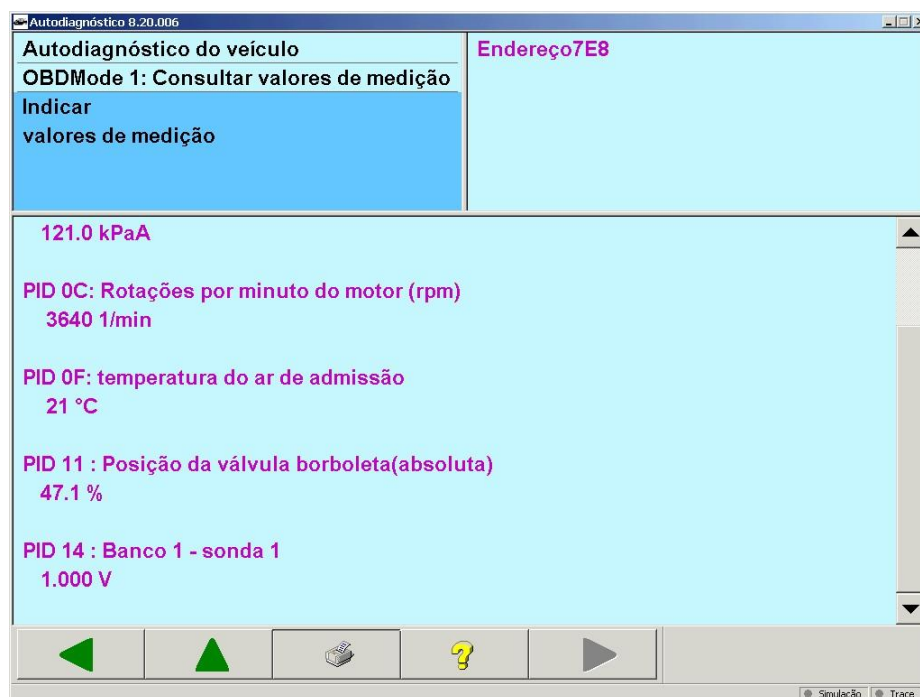


Figura 4.8 Segunda tela dos parâmetros de leitura no serviço \$01

No serviço \$03 é possível realizar a leitura dos códigos de falha armazenados na memória da ECU, conforme apresentado na figura 4.9.

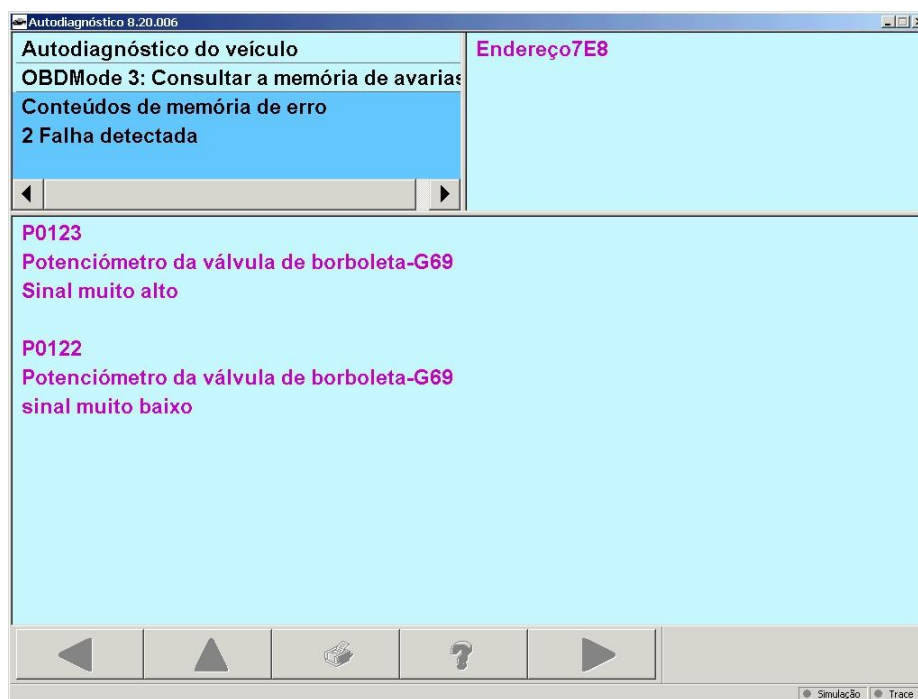


Figura 4.9 Leitura dos códigos de falha no serviço \$03

Após ter verificado os códigos de falha, é possível no serviço \$04 apagar a memória de falhas da ECU, conforme apresentado na figura 4.10.

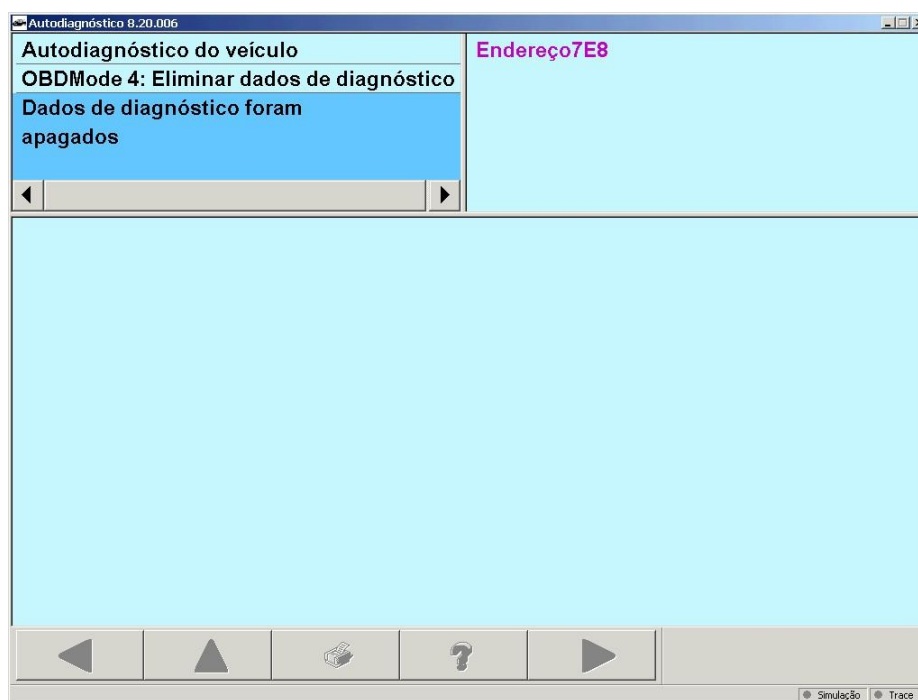


Figura 4.10 Tela de confirmação da eliminação dos códigos de falha

5 CONCLUSÃO

Com o aumento da frota de veículos em escala mundial e a introdução da eletrônica embarcada, os sistemas de diagnóstico veicular foram implementados, devido à pressão dos governos pela busca da diminuição da emissão de poluentes, motivados pela melhoria da qualidade de vida das pessoas, visto que os antigos sistemas veiculares não realizavam nenhum controle de emissões, gerando danos ao meio ambiente e à saúde da sociedade.

Desta forma, os sistemas de diagnóstico veicular surgiram com o objetivo de monitorar o correto funcionamento dos motores à combustão, por exemplo a relação estequiométrica para a queima completa da mistura ar/combustível admitida, com o intuito de reduzir a emissão de poluentes, sendo que um mau funcionamento detectado pelo sistema alerta o condutor do veículo para realizar o reparo. E ainda possibilita uma *interface* rápida e eficiente para a detecção de falhas pelos mecânicos de manutenção.

Este projeto apresentou o desenvolvimento de um sistema de diagnóstico veicular utilizando uma plataforma didática de gerenciamento eletrônico que monitora o funcionamento do motor através da leitura dos parâmetros, caso alguma irregularidade seja detectada, o sistema de diagnóstico gera um código de falha (DTC) e o armazena na memória de falhas. Assim como no ciclo de trabalho do motor, quando é detectado uma falha relacionada ao controle de emissões, é acesa uma lâmpada (LIM) indicando um mau-funcionamento do sistema. O dispositivo também disponibiliza alguns parâmetros de leitura do motor (PIDs e DTCs) em tempo real para um equipamento de diagnóstico *off-board*.

A realização deste trabalho é um passo importante para o desenvolvimento de uma ECU profissional, permitindo que os alunos compreendam melhor conteúdos teóricos a partir deste experimento, bem como pesquisadores dispostos a utilizar como referência este projeto para desenvolver novos sistemas de gerenciamento, visto que em contrapartida os sistemas atuais são restritos fora do ambiente industrial.

5.1 Propostas Futuras

Como propostas futuras, notamos que é possível a implementação de melhorias para que o projeto torne-se mais robusto e chegue mais próximo de uma unidade de gerenciamento eletrônico profissional, como:

- Implementação de mais serviços de diagnóstico;
- Implementação de mais parâmetros de leitura;
- Integração com um projeto de gerenciamento;
- Utilização de um sistema operacional de tempo real mais avançado, por exemplo OSEK, trazendo mais segurança e confiabilidade do sistema;
- Criação de uma estrutura que permita a comunicação por múltiplos *frames* do protocolo ISO 15765;
- Detecção de uma maior quantidade de número de falhas;
- Desenvolvimento de um *bootloader* via rede CAN.

6 REFERÊNCIAS

- AEA – Associação Brasileira de Engenharia Automotiva. **V Seminário sobre a Eletro-Eletrônica Aplicada à Mobilidade – Diagnose Veicular**, São Paulo, 2003.
- AFONSO, Leonardo; RENESAS. **Renesas Programa Universitário - Apresentação durante a semana de tecnologia da Fatec Santo Andre**. 2012.
- AUDI VOLKSWAGEN. **Diagnosis with VAS 5051, VAS 5052**. Manual de treinamento e serviço. 2005.
- BASTOS, Eduardo. **Estudo das Diferenças dos Requerimentos das Principais Legislações de On Board Diagnostics para Padronização de Testes de Desenvolvimento e Validação de Transmissão Automática de Automóveis**. Centro Universitário do Instituto Mauá de Tecnologia. São Caetano do Sul - SP. Monografia. 2012.
- BELO, Valdeci Pereira. **Sistema para Diagnóstico Automático de Falhas em Veículos Automotores OBD-2**. UFMG Belo Horizonte - MG. Dissertação de Mestrado. 2003.
- BOSCH, Robert. **Manual de Tecnologia Automotiva**. Tradução da 25ª ed. São Paulo - SP: Edgard Blücher, 2005.
- BROEDEL, Alex; SANTOS, Hebert Gomes dos. **Sistema de Auxílio de Estacionamento Modularizado com Protocolo CAN**. Santo André. Monografia de TCC. 2011.
- CERQUEIRA, Ademar Dultra et al. **Sistema de Diagnóstico para Veículos que Utilizam os Protocolos ISO 9141 e ISO 14230 Através de uma Plataforma em LabView®**. FATEC Santo André. Santo André. Monografia de TCC. 2009.
- DIAS, Bruno Martin de Alcântara. **Plataforma Didática de Injeção Eletrônica para Controle de Motores de Combustão Interna**. FATEC Santo André. Santo André. Monografia de TCC. 2011.
- GOMES, Cleber William. **Diagnose de Motores Diesel por meio de Redes Bayesianas**. Centro Universitário da FEI. São Bernardo - SP. Dissertação de Mestrado. 2008.
- GUIMARÃES, Alexandre de Almeida. **Eletrônica Embarcada Automotiva**. 1ª ed. São Paulo: Erica, 2007.
- HODEL, Kleber Nogueira. **Limites Do Protocolo Can (Controller Area Network) Para Aplicações Que Exigem Alto Grau De Confiabilidade Temporal**. Universidade de São Paulo. São Paulo. Dissertação de Mestrado. 2008.
- HODEL, Kleber Nogueira. **Notas de Aula da Disciplina de Redes de Comunicação Automotiva Ministrada ao Curso de Eletrônica Automotiva da Fatec Santo André**. Santo André. 2011.
- Inga Escapamentos, 2012. Disponível em:
<<http://www.ingaescapamentos.com.br/assets/img/blog/catalisador-2-sondas.jpg>>.
Acesso em: 6 Setembro 2012.

International Organization for Standardization. **ISO 15765-2 Road vehicles - Diagnostics on Controller Area Network (CAN) - Part 2: Network layer services.** ISO. 2002.

International Organization for Standardization. **ISO 15765-4 Road vehicles - Diagnostics on Controller Area Network (CAN) - Part 4: Requirements for emissions-related systems.** ISO. 2003.

International Organization for Standardization. **ISO 15031-6 Road Vehicles - Communication between vehicle and external equipment for emissions-related diagnostics - Part 6: Diagnostic trouble code definitions.** ISO. 2005.

International Organization for Standardization. **ISO 15031-5 Road Vehicles - Communication between vehicle and external equipment for emissions-related diagnostics - Part 5: Emissions-related diagnostic services.** ISO. 2006.

JUCÁ, Sandro. **Apostila de microcontroladores PIC e periféricos.** São Paulo. 2010.

JUNIOR, Hélio Taliani. **Estudo Dos Protocolos De Comunicação Das Arquiteturas Eletroeletrônicas Automotivas, com Foco nas suas Características e Respectivas Aplicações, Visando o Direcionamento para o Uso Adequado e Customizado em cada Categoria de Veículo.** São Caetano do Sul. Monografia. 2012a.

JUNIOR, Orlando Salvo. **Notas de Aula da Disciplina de Diagnose Veicular Ministrada ao Curso de Eletrônica Automotiva da Fatec Santo André.** Santo André. 2012b.

MACHADO, Antonio Sérgio Leite; OLIVEIRA, Bruno Rafael Resende. **O Sistema OBD (On-Board Diagnosis).** Instituto Superior de Engenharia do Porto. Porto - Portugal. Paper. 2007.

MILHOR, Carlos Eduardo. **Sistema de Desenvolvimento para Controle Eletrônico dos Motores de Combustão Interna Ciclo Otto.** Escola de Engenharia de São Carlos, USP. São Carlos - SP. Dissertação de Mestrado. 2002.

NETO, Alcides Carlos De Moraes. **Telemetria Automotiva via Internet Móvel.** Universidade de Brasília. Brasília. Monografia de TCC. 2009.

TEMPO REAL EVENTOS. Seminário RTOS. **Conceitos sobre RTOS,** São Paulo, 20 Novembro 2010.

The Atlantic, 2012. Disponível em:

<<http://www.theatlantic.com/infocus/2011/11/documerica-images-of-america-in-crisis-in-the-1970s/100190/>>. Acesso em: 25 Outubro 2012.

TORRES, Wesley Medeiros. **Notas de Aula da Disciplina de Tópicos Avançados de Programação de Microcontroladores Ministrada ao Curso de Eletrônica Automotiva da Fatec Santo André.** Santo André. 2012.