



Projeto

Mestrado em Engenharia Automóvel

***Desenvolvimento de uma interface de comunicação  
para motores de veículos pesados***

**Marcelo de Pinho Soares**

Leiria, *Setembro* de 2016





Projeto

Mestrado em Engenharia Automóvel

***Desenvolvimento de uma interface de comunicação  
para motores de veículos pesados.***

**Marcelo de Pinho Soares**

Dissertação de Mestrado realizada sob a orientação do Doutor Carlos Daniel Henriques Ferreira, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria e coorientação do Doutor Luis Manuel Ventura Serrano, Professor da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, *Setembro de 2016*

*Esta página foi intencionalmente deixada em branco*

***“I have no idols. I admire work, dedication and competence.”***

Ayrton Senna

*Esta página foi intencionalmente deixada em branco*

**À minha Família, Namorada e Amigos**

*Esta página foi intencionalmente deixada em branco*

# Agradecimentos

---

Durante o meu percurso académico tive o prazer de conhecer várias pessoas, todas elas me ajudaram a crescer como pessoa e como profissional, assim agradeço a todas elas.

Os meus primeiros agradecimentos vão para os meus dois orientadores: Professor Doutor Carlos Ferreira e Professor Doutor Luis Serrano, que me acompanharam ao longo de todo o projeto. Agradeço todo o apoio, disponibilidade e tempo que me foi despendido, assim como todo o conhecimento que me transmitiram.

Agradeço aos amigos e técnicos responsáveis do Laboratório de Engenharia Automóvel, Engenheiro Nuno Pires e Engenheiro Diogo Costa, pela valiosa ajuda na parte prática do projeto e pelo conhecimento transmitido ao longo do meu percurso académico.

Agradeço a minha família pelo constante apoio e compreensão dedicado em toda a minha vida. Agradeço também a todo o incentivo que me transmitiram necessário para atingir os objetivos traçados.

Agradeço à minha namorada por todo apoio que me facultou durante este projeto, relembrando os meus objetivos nos momentos mais difíceis. Agradeço também todo o ânimo que me transmitiu.

Agradeço a todos os meus amigos que me acompanhar ao longo deste projeto, pela confiança que me transmitiram e pela sua amizade nos momentos críticos. A todos eles um muito obrigado.

Agradeço por fim, mas sem menos importância, ao Instituto Politécnico de Leiria, pela maneira como me acolheu e por todos os meios que disponibilizou para a realização deste projeto.

*Esta página foi intencionalmente deixada em branco*

# Resumo

---

O objetivo do presente trabalho é desenvolver uma interface que comunique com um veículo pesado adquirindo informação de sensores e que controle o regime do motor. Com este interface é possível analisar a condição de motores após o seu acondicionamento de modo a garantir a sua fiabilidade antes de ser reintroduzido no seu ciclo de utilização.

Para tal, neste projeto, descodificam-se os protocolos de comunicação existentes nos veículos pesados, através da aplicação de métodos de engenharia inversa aos barramentos CAN. Estudam-se vários métodos de controlo do regime do motor e analisam-se quais os métodos mais eficazes. Desenvolve-se *hardware* que promova a ligação entre a informação presente nos barramentos CAN e um computador (porta série). Define-se um protocolo de comunicação dedicado e implementa-se este na comunicação entre o *hardware* e o computador.

Recorre-se ao *software* LabVIEW® para o desenvolvimento da aplicação, executada num computador, que possibilita visualizar os dados adquiridos pelo *hardware*. Com esta aplicação, o utilizador controla o regime do motor através de um ambiente gráfico “*user friendly*”. No fim de cada ensaio gera-se um ficheiro de dados que contem toda a informação adquirida permitindo uma interpretação e análise *à posteriori*.

Realizam-se vários testes ao sistema desenvolvido, tanto ao nível de *hardware* como de *software*, com o objetivo de validar as funções implementadas. Após a análise dos resultados obtidos verifica-se correto funcionamento da interface e da aplicação desenvolvida durante o projeto.

*Palavras-chave:*

*Comunicação em Veículos; Veículo Pesado; Aquisição de dados; Controlo; Diagnóstico; Manutenção.*

*Esta página foi intencionalmente deixada em branco*

# Abstract

---

The main goal of the project is developing a device to communicate with a heavy-duty vehicle, acquire information from sensors and control the engine speed. With this device it is possible to analyze the state of the engine after it has been rebuilt, in order to guarantee the reliability before being allocated to a new vehicle.

The existent communication protocols in heavy-duty vehicles, are decoded by applying reverse engineering methods to the CAN buses. Various engine speed control methods are studied and compared, in order to determine which one is the most effective. A piece of hardware is developed to allow the connection between the information in the buses and a computer using a custom communication protocol.

LabVIEW® is used to develop the data viewing and recording application, which runs on a PC. This application also allows the user to control the engine speed through a “user friendly” GUI. At the end of each session a file that contains all the information acquired through hardware is created, allowing the user to interpret and analyze it afterwards.

Hardware and software validation tests are performed, to ensure all of the features are working properly. Test results are then presented and discussed.

*Keywords:*

*Vehicle Communication; Heavy-Duty Vehicle; Data Acquisition; Control; Diagnostic; Maintenance.*

*Esta página foi intencionalmente deixada em branco*

# Lista de figuras

---

Figura 1.1 – Ciclo de vida dos veículos pesados.....	2
Figura 2.1- Esquema representativo dos componentes do caso de estudo. A) Motor Scania®, B) Sistema elétrico de controlo e seus componentes, C) Controlador R480 e D) Conector OBD-II <i>Type B</i> .....	5
Figura 2.2 - Motor DT12 17 480 Euro 4[3]. .....	6
Figura 2.3 - Gráficos de potência e de binário do motor D12 17 480 Euro 4[4]. .....	7
Figura 2.4 – Barramentos CAN presentes num Scania® R480[5].....	8
Figura 2.5 - Barramentos CAN presentes no caso de estudo[5]. .....	9
Figura 2.6 - Esquema das unidades EMS e COO[6]–[8]. .....	12
Figura 2.7 – Controlador R480.....	13
Figura 2.8 - Conector OBD-II <i>Type B</i> . 4) GND, 6) CAN H, 14) CAN L e 16) +24 V. ....	14
Figura 2.9 - OSI <i>Seven Model</i> [9]. .....	16
Figura 2.10 - Esquema das duas camadas inferiores do modelo OSI. ....	17
Figura 2.11 – Exemplo de uma sequência de mensagens <i>Broadcast</i> .....	21
Figura 2.12 – Exemplo de mensagens Pergunta-Resposta. ....	22
Figura 2.13 – Comparação do consumo de combustível entre o segundo método proposto e a medição digital do consumo.....	25
Figura 2.14 – Percentagem de erro do cálculo do combustível[16].....	26
Figura 3.1 - PEAK Systems CAN Analyser[20]. .....	30
Figura 3.2 - Osciloscópio da PicoScope®[21]. .....	30
Figura 3.3 - Equipamento diagnóstico VCI 2 da Scania®[22]. .....	31
Figura 3.4 - Fotografia do painel de instrumentos durante a aquisição dos dados. ....	32
Figura 3.5 - Esquema de ligação do VCI2 e o interface CAN.....	37
Figura 3.6 - Esquema de ligação do interface CAN ao barramento vermelho. ....	39
Figura 3.7 - Gráfico valores dos parâmetros adquiridos durante o envio da mensagem a 100 Hz. 44	
Figura 3.8 - Esquemático da ligação do Arduino® UNO e o interface CAN ao barramento vermelho.....	45
Figura 3.9 - Gráfico valores dos parâmetros adquiridos durante a replicação da mensagem pelo Arduino®.....	46
Figura 3.10 - Gráfico valores dos parâmetros adquiridos durante o envio das mensagens de <i>Cruise Control</i> . .....	47
Figura 3.11 - Diagrama de ligação do potenciômetro da posição do acelerador[6]. .....	48
Figura 3.12 - Gráfico da curva de transferência do potenciômetro original. ....	49
Figura 3.13 - Esquema de ligação do potenciômetro externo ao COO. ....	49
Figura 4.1 - Esquema de ligação do interface EDAS.....	52
Figura 4.2 - Diagrama de blocos dos componentes presentes no EDAS. ....	52
Figura 4.3 – Diagrama de blocos dos componentes da PCB. ....	59
Figura 4.4 – Esquema de ligação do <i>transceiver</i> CAN. ....	60

Figura 4.5 – Esquema de ligação do potenciômetro digital. ....	61
Figura 4.6 – Camada superior da PCB. ....	61
Figura 4.7 – Camada inferior da PCB. ....	62
Figura 4.8 – Protótipo funcional da PCB. ....	62
Figura 4.9 – Pinout da ficha OBD-II modificada. ....	64
Figura 4.10 – Esquema do cabo de ligação entre o EDAS e a ficha OBD-II modificada. ....	64
Figura 4.11 – Exemplo de uma sequência de mensagens do Handshake. ....	66
Figura 4.12 - Exemplo de uma sequência de mensagens durante a configuração (Config). ....	67
Figura 4.13 – Exemplo de uma sequência de mensagens durante a aquisição e controlo (Request). ....	69
Figura 4.14 – Fluxograma da fase inicial do código. ....	72
Figura 4.15 - Fluxograma do Programa Principal. ....	75
Figura 4.16 - Fluxograma da interrupção CAN. ....	75
Figura 5.1 - Exemplo de um diagrama de blocos do LabVIEW®. ....	80
Figura 5.2 - Exemplo de um painel frontal do LabVIEW®. ....	80
Figura 5.3 - Diagrama de blocos simplificado da aplicação desenvolvida. ....	83
Figura 5.4 – Diagrama de blocos da função <i>Handshake</i> . ....	84
Figura 5.5 - Diagrama de blocos da função Configuração. ....	84
Figura 5.6 - Diagrama de blocos da função Pedido de parâmetros de alta prioridade. ....	85
Figura 5.7 - Diagrama de blocos da função Controlo da posição do acelerador. ....	86
Figura 5.8 - Diagrama de blocos da função Guardar os dados. ....	87
Figura 5.9 – <i>Layout Menu</i> de seleção. ....	88
Figura 5.10 - <i>Layout Menu Admin</i> . ....	89
Figura 5.11 - <i>Layout</i> Visualização instantânea com controlo do acelerador. ....	90
Figura 5.12 - <i>Layout</i> Visualização gráfica com controlo <i>cruise control</i> . ....	91
Figura 6.1 – Sistema implementado em modo funcionamento. ....	93
Figura 6.2 – Gráfico resultante do teste ao potenciômetro digital. ....	95
Figura 6.3 – Gráfico resultante do teste à frequência. ....	97
Figura 6.4 – Esquema de ligação do teste à aquisição. ....	98
Figura 6.5 – Gráficos resultantes do ensaio da aquisição de dados. ....	99
Figura 6.6 – Gráfico da rotação do motor durante a fase de arranque. ....	100
Figura 6.7 – Gráfico do teste ao controlo posição do acelerador. ....	101
Figura 6.8 – Gráfico do teste do <i>cruise control</i> . ....	102
Figura 6.9 – Gráfico resultante do teste ao parâmetro de emergência Rotação do motor. ....	103
Figura 6.10 - Gráfico resultante do teste ao parâmetro de emergência Temperatura do motor. ..	103
Figura 6.11 – Gráfico no tempo da rotação e temperatura do motor. ....	104
Figura 6.12 - Gráfico no tempo da Massa de Ar e Consumo atual de combustível. ....	105
Figura 7.1 – Gráfico de dispersão relacionando a Pressão de Combustível e a Rotação do Motor. .....	109

# Lista de tabelas

---

Tabela 2.1 - Resumo das unidades de comando presentes no caso de estudo.....	11
Tabela 2.2 - Identificação dos componentes do CAN-ID.....	18
Tabela 2.3 - Exemplo de PGN inserido numa mensagem CAN.....	18
Tabela 2.4 - Exemplo de um PGN descrito na norma[11]. .....	20
Tabela 2.5 – Formato do <i>extended</i> CAN-ID para diagnóstico OBD-II genérico.....	23
Tabela 2.6 – Definição de alguns parâmetros no modo 01[14]. .....	24
Tabela 2.7 – Mensagem de pedido da temperatura do motor.....	24
Tabela 3.1 – Informação da norma SAE J1939 relativa aos parâmetros apresentados. ....	33
Tabela 3.2 - Mensagens adquiridas no barramento verde. ....	33
Tabela 3.3 - Descodificação da rotação do motor. ....	34
Tabela 3.4 - Descodificação do nível de combustível.....	34
Tabela 3.5 - Descodificação da temperatura ambiente. ....	35
Tabela 3.6 - Descodificação da pressão de óleo. ....	35
Tabela 3.7 - Comparação dos parâmetros escolhidos para validar a norma SAE J1939.....	35
Tabela 3.8 - Descodificação de duas mensagens de diagnóstico SDP3. ....	37
Tabela 3.9 - Mensagem do modo OBD-II genérico a solicitar a rotação do motor.....	38
Tabela 3.10 - Mensagem de resposta da EMS a solicitação OBD-II genérico.....	38
Tabela 3.11 - Descrição do PGN associado ao controlo de binário e rotação do motor[11]......	40
Tabela 3.12 - Mensagem TSC1 enviada para o barramento CAN vermelho. ....	40
Tabela 3.13 - Mensagem com o PGN 0xFF80 presente no barramento vermelho.....	42
Tabela 3.14 – Mensagem de teste com o PGN 0xFF80.....	42
Tabela 3.15 - Mensagens de controlo do <i>Cruise Control</i> . ....	47
Tabela 3.16 - Valores, em tensão, dos sinais de saída do potenciómetro. ....	48
Tabela 4.1 – Soluções encontradas para aquisição de dados. ....	53
Tabela 4.2 – Parâmetros a adquirir durante um ensaio.....	54
Tabela 4.3 - Soluções encontradas para o controlo da rotação do motor.....	55
Tabela 4.4 – Lista de componentes do EDAS. ....	56
Tabela 4.5 – Protocolo de comunicação na fase de Handshake.....	66
Tabela 4.6 - Protocolo de comunicação na fase de Config. ....	67
Tabela 4.7 – Protocolo de comunicação na fase de Request. ....	68
Tabela 4.8 – Frequência de aquisição dos parâmetros.....	77
Tabela 6.1 – Excerto das mensagens presentes no barramento CAN durante o teste à comunicação CAN.....	96
Tabela 10.1 - Descrição da descodificação dos parâmetros presentes na no PGN 0x0000[11]. .	115
Tabela 10.2 - Campo de dados da mensagem de request através do SDP3. ....	116
Tabela 10.3 – Parâmetros disponíveis através dos vários métodos. ....	118

*Esta página foi intencionalmente deixada em branco*

# Lista de siglas

---

APS	Air Processing System
APPS	Accelerator Pedal Position Sensor
ASCII	American Standard Code for Information Interchange
CAN	Controller Area Network
CCV	Cruise Control Vehicle
COO	Coordinator
EDAS	Engine Data Acquisition System
EMS	Engine Management System
GUI	Graphical User Interface
ICL	Instrument Cluster
ISO	International Organization for Standardization
OBD	On-Board Diagnostics
OSI	Open Systems Interconnection
PCB	Printed Circuit Board
PGN	Parameter Group Number
RN	Random Number
RNC	Random Number Calculated
SA	Source Address
SAE	Society of Automotive Engineers
SDP3	Scania Diagnos & Programmer 3
SPI	Serial Peripheral Interface
SPN	Suspect Parameter Number
VCI2	Vehicle Communication Interface 2
VIS	Visibility System

*Esta página foi intencionalmente deixada em branco*

# Índice

---

<b>AGRADECIMENTOS</b>	<b>VII</b>
<b>RESUMO</b>	<b>IX</b>
<b>ABSTRACT</b>	<b>XI</b>
<b>LISTA DE FIGURAS</b>	<b>XIII</b>
<b>LISTA DE TABELAS</b>	<b>XV</b>
<b>LISTA DE SIGLAS</b>	<b>XVII</b>
<b>ÍNDICE</b>	<b>XIX</b>
<b>1. INTRODUÇÃO</b>	<b>1</b>
1.1. Contexto e Motivação	1
1.2. Objetivos	2
1.3. Estrutura do relatório	3
<b>2. REVISÃO BIBLIOGRÁFICA</b>	<b>5</b>
2.1. Veículo Pesado	5
2.2. Comunicação em veículos	14
2.3. Outros trabalhos na área	24
<b>3. DESCODIFICAÇÃO DOS PROTOCOLOS E MÉTODOS DE CONTROLO</b>	<b>29</b>
3.1. Equipamentos utilizados	29
3.2. Norma SAE J1939	32
3.3. Modo Diagnóstico (SDP3)	36
3.4. Modo OBD-II Genérico	38
3.5. Métodos de controlo do motor	39
<b>4. DESENVOLVIMENTO DO <i>HARDWARE</i></b>	<b>51</b>
4.1. Introdução, características e funcionalidades	51
4.2. Seleção dos componentes	55
4.3. Projeto e Desenvolvimento da placa de circuito impresso (PCB)	59
4.4. Cabos de ligação	63
4.5. Protocolo de comunicação (EDAS-PC)	65
4.6. Desenvolvimento do código do microcontrolador	69
<b>5. <i>SOFTWARE</i> DE VISUALIZAÇÃO E CONTROLO</b>	<b>79</b>
5.1. Ferramenta de desenvolvimento	79
5.2. Metodologia	80
5.3. Funções Implementadas	83
5.4. Apresentação do Ambiente Gráfico	87
<b>6. TESTES DE VALIDAÇÃO</b>	<b>93</b>
6.1. Implementação do sistema	93
6.2. Validação do <i>Hardware</i>	94
6.3. Validação das funcionalidades do sistema	96
6.4. Exemplo de um ensaio	104

<b>7.</b>	<b>DISCUSSÃO DE RESULTADOS</b>	<b>107</b>
<b>8.</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>111</b>
8.1.	Trabalhos Futuros	112
<b>9.</b>	<b>BIBLIOGRAFIA</b>	<b>113</b>
<b>10.</b>	<b>ANEXOS</b>	<b>115</b>
10.1.	Mensagem de Torque/Speed Control	115
10.2.	Parâmetros disponíveis	116
10.3.	Decodificação do Protocolo SAE J1939	121

# 1. Introdução

---

## 1.1. Contexto e Motivação

---

Atualmente o transporte terrestre de mercadorias é dominado pelos veículos pesados que corresponde a 75,1% das mercadorias transportadas na Europa. Em Portugal este valor é superior à média europeia correspondendo a 93,2% das mercadorias transportadas a nível nacional [1]. Muitos destes veículos pesados que circulam nas estradas utilizam como meio de propulsão motores reconicionados que são, tipicamente, provenientes de veículos em fim de vida. O fim de vida destes veículos deve-se, principalmente a duas causas, a acidentes e ao número de horas de utilização. Quando a causa do fim de vida é o elevado número de horas de operação, o motor possui um elevado desgaste sendo necessário submeter o mesmo a um reconicionamento.

Apresenta-se na Figura 1.1 um esquema simplificado do ciclo de vida dos veículos pesados que é iniciado pela produção em série e respetiva colocação em seu processo de utilização que culmina num fim de vida de um veículo pesado. Após o fim de vida, alguns dos motores são reconicionados e posteriormente reintegrados no ciclo de utilização. Porem após o reconicionamento deve-se analisar o estado do motor de modo a garantir a sua fiabilidade durante a sua nova utilização. Atualmente, a análise efetuada é superficial sendo apenas verificado o valor de alguns parâmetros de funcionamento.

O presente trabalho centra-se no desenvolvimento de uma interface capaz de analisar, com um maior rigor, a condição de um motor de veículo pesado. Potencialmente, com este dispositivo melhora-se o processo de manutenção de um veículo pesado através da redução do tempo de imobilização do veículo. Através da redução do tempo de imobilização de um veículo melhora a produtividade das frotas de transporte terrestre de mercadorias.

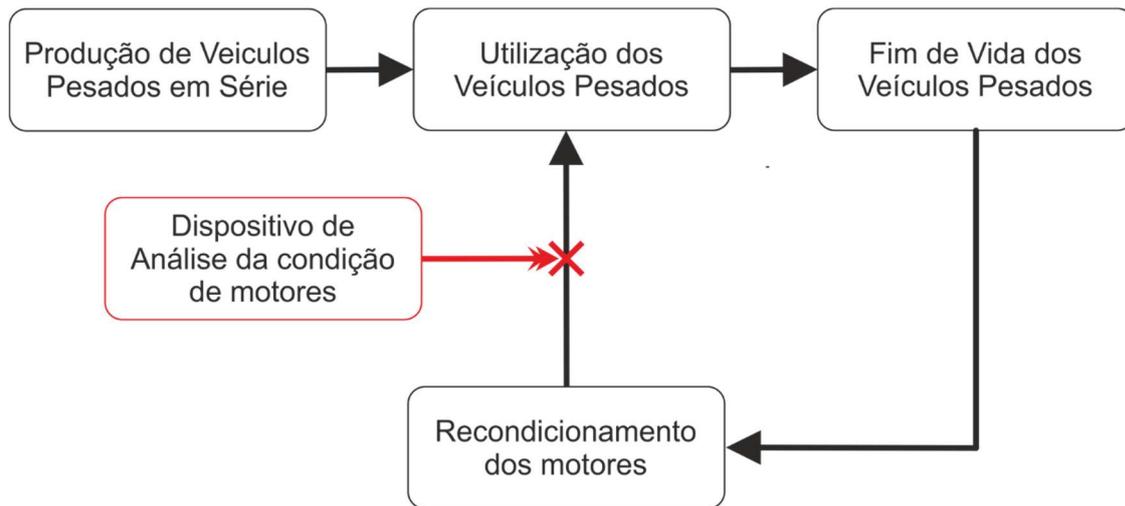


Figura 1.1 – Ciclo de vida dos veículos pesados

## 1.2. Objetivos

---

Este projeto tem como objetivo projetar e desenvolver uma interface capaz de promover uma análise mais rigorosa ao motor. Desenvolveu-se um interface com a capacidade de comunicar com um veículo pesado adquirindo informação dos sensores do motor e também controlar o próprio motor. Esta informação deve ser visualizada e o controlo do motor através de um *software* dedicado ao *hardware* desenvolvido.

Este interface deve possibilitar uma futura integração num banco ensaios de motores e assim adquirir informação em todas as velocidades de rotação e carga do motor. Também se pode efetuar, com este interface, uma análise de condição de motores a veículos que já se encontram no seu ciclo de utilização de modo a avaliar o seu estado.

## 1.3. Estrutura do relatório

---

O presente trabalho encontra-se organizado em oito capítulos. O primeiro capítulo consiste na introdução do tema do trabalho, sua contextualização e objetivos. No segundo capítulo efetua-se uma revisão bibliográfica expondo os componentes integrantes de um veículo pesado com interesse para o trabalho, o protocolo de comunicação utilizado em veículos pesados e por fim alguns trabalhos realizados na área. No terceiro capítulo apresenta-se a descodificação dos vários protocolos presentes no veículo em estudo e os métodos testados de controlo do motor. Posteriormente, no capítulo quatro descreve-se todo o desenvolvimento do *hardware*, desde a seleção dos componentes até ao desenvolvimento do código do microprocessador. No quinto expõe-se o desenvolvimento do *software* de visualização e controlo do motor, composto pela metodologia, as funções implementadas e por fim o ambiente gráfico para o utilizador. No sexto capítulo apresenta-se a descrição dos testes efetuados ao interface desenvolvido e resultados obtidos. No sétimo capítulo discute-se os resultados obtidos na fase de testes e apresenta-se as respetivas conclusões. Por fim, no oitavo capítulo retira-se as conclusões de todo o trabalho desenvolvido e apresenta-se as sugestões para trabalhos futuros.



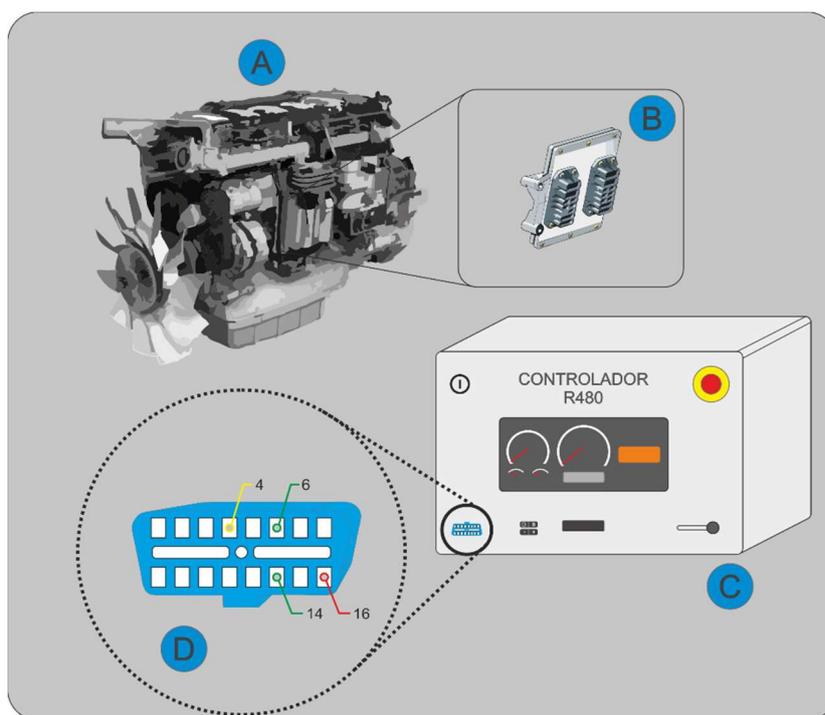
## 2. Revisão Bibliográfica

---

### 2.1. Veículo Pesado

---

Nesta secção apresenta-se informações relativas ao veículo pesado do fabricante Scania® e os componentes integrantes. De uma forma resumida, como se exemplifica na Figura 2.1, o presente trabalho incide sobre quatro elementos; motor de um veículo pesado, a respetiva unidade de comando, um controlador R480 e o conector do tipo OBD-II (que se encontra no controlador). De notar que o controlador R480 é constituído pelas unidades de controlo, componentes necessários para efetuar o controlo do motor e apresentar o feedback sobre o seu funcionamento ao utilizador através de um painel de instrumentos dos veículos Scania®.

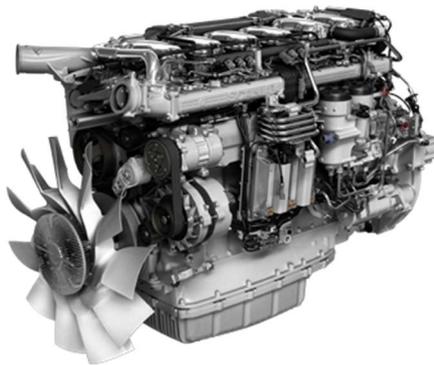


**Figura 2.1- Esquema representativo dos componentes do caso de estudo.**  
**A) Motor Scania®, B) Sistema elétrico de controlo e seus componentes,**  
**C) Controlador R480 e D) Conector OBD-II Type B.**

### 2.1.1. A - Motor Scania®

---

O motor utilizado no case em estudo é proveniente de um veículo pesado, Scania® R480 de 2007. De acordo com o *software* de diagnóstico Scania Diagnos & Programmer 3® [2], a referência do motor é DT12 17 480 Euro 4. Como principais características o motor DT12 17 480 (apresentado na Figura 2.2), possui um sistema de admissão baseado numa sobrealimentação através de um turbocompressor. Posteriormente, o ar admitido é arrefecido no *intercooler* e é direcionado para os cilindros. Este motor também possui um sistema de *turbocompund* de modo a reaproveitar a energia dos gases de escapes e adicionar essa energia à cambota. Quanto à sua cilindrada, tem como capacidade 12,7 litros de volume distribuídos igualmente por 6 cilindros em linha. Por cada cilindro são usadas 4 válvulas, onde duas são de admissão e as outras duas de escape. O sistema de injeção de combustível é do tipo injetor-bomba, onde a pressão de combustível é controlada em duas condutas independentes, cada conduta serve um grupo de três injetores, por um sistema de electroválvulas.



**Figura 2.2 - Motor DT12 17 480 Euro 4[3].**

Como se mostra na Figura 2.3, o motor Scania® DT12 17 480 apresenta uma potência máxima de 353 kW às 1900 rpm/min e um binário máximo de 2250 Nm, constante entre as 1100 e 1450 rpm/min. Tal como é perceptível pela referência do motor, este motor respeita a norma Euro 4 relativamente aos níveis de emissão de poluentes.

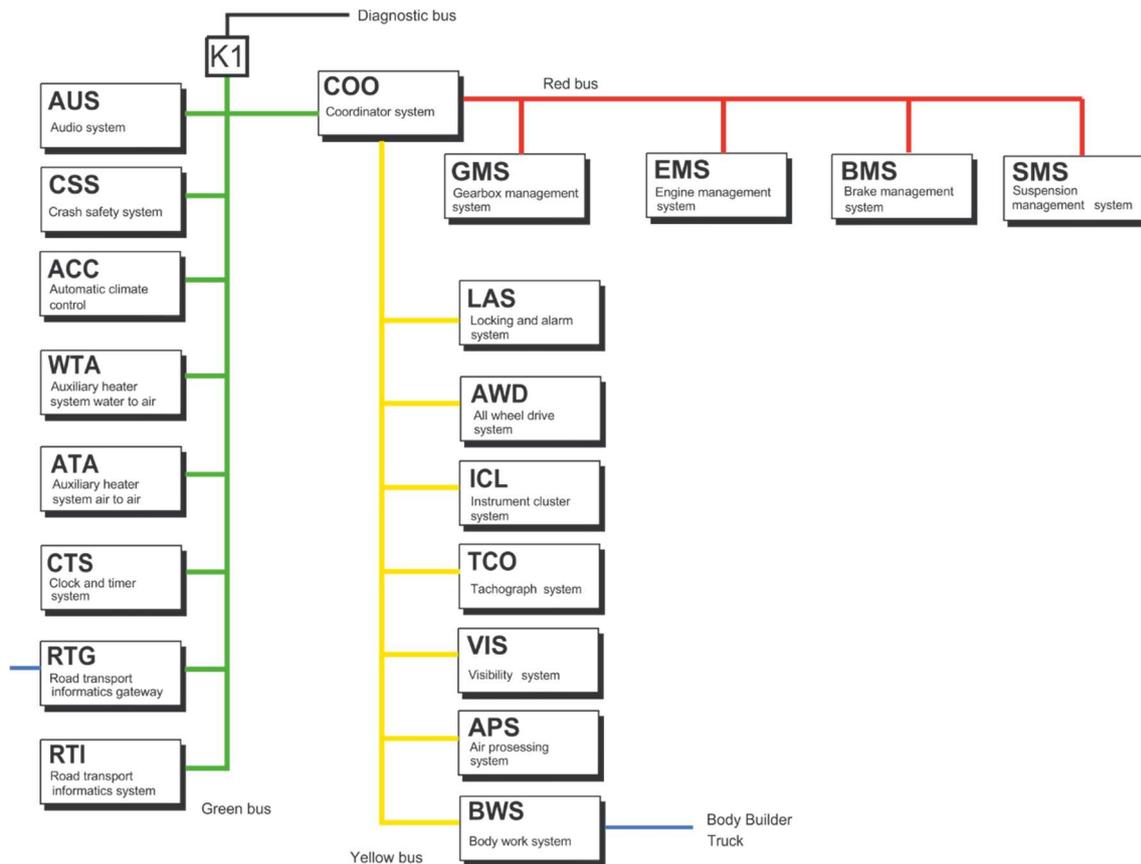


Figura 2.3 - Gráficos de potência e de binário do motor D12 17 480 Euro 4[4].

## 2.1.2. B - Sistema elétrico e seus componentes

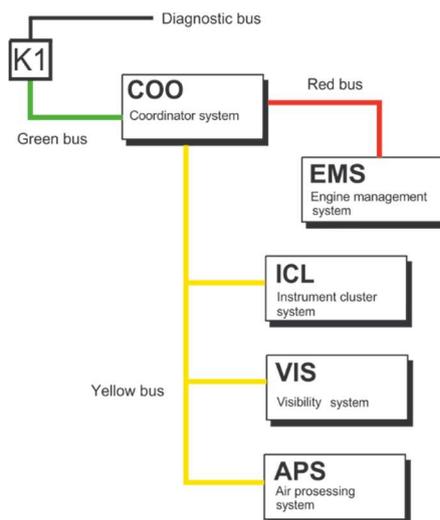
---

O veículo pesado, Scania® R480, de onde foi retirado os componentes para o sistema em estudo pode conter várias unidades de comando. As unidades de comando estão agrupadas de acordo com as suas funcionalidades e comunicam entre si recorrendo a três barramentos CAN (Figura 2.4). O barramento vermelho “Red bus” é utilizado pelas unidades de controlo do motor, sistema de travagem, caixa de velocidades e suspensão. Depois utiliza o barramento amarelo “Yellow bus”, para os sistemas de instrumentação, tacógrafo, processamento de ar, controlo de acesso e alarme, entre outros. Por fim, o barramento verde “Green bus” é utilizado pelos sistemas auxiliares. A cor dos barramentos aparenta indicar o nível de importância e prioridade no funcionamento do veículo. De notar que o coordenador (COO) interliga os três barramentos e a ficha de diagnóstico OBD-II, “K1” esta conectada ao barramento verde.



**Figura 2.4 – Barramentos CAN presentes num Scania® R480[5].**

De notar que o caso de estudo (Figura 2.1), só dispõe das unidades essenciais para colocar em funcionamento e controlar o motor. As unidades presentes são: EMS (unidade de controlo do motor), COO (coordenador), ICL (painel de instrumentos), VIS (sistema de iluminação e visibilidade) e APS (sistema de ar comprimido). A Figura 2.5 resume as unidades disponíveis no case de estudo e os respetivos barramentos. De notar que o barramento verde é utilizado exclusivamente para a ligação da ficha de diagnóstico ao coordenador e que o barramento vermelho é utilizado por dois dispositivos: unidade do controlo do motor (EMS) e coordenador (COO).



**Figura 2.5 - Barramentos CAN presentes no caso de estudo[5].**

### **Descrição da função das várias unidades**

Neste ponto explica-se de uma forma simplificada qual a função das várias unidades, a sua localização e qual o barramento CAN utilizado.

#### **COO – Coordenador**

O coordenador tem como função coordenar o funcionamento dos sistemas do veículo. É um componente essencial ao funcionamento do veículo pois recebe informações, sob a forma de mensagens CAN (provenientes das outras unidades de comando) e de sensores diretamente ligados às suas entradas. Do processamento da informação recebida resultam comandos (mensagens individualmente dirigidas aos sistemas do veículo) e o reenvio de dados generalizados entre os diferentes barramentos. O coordenador utilizado no nosso caso de estudo encontra-se dentro do controlador R480, onde ligam os três barramentos CAN previamente descritos.

#### **EMS – Sistema de controlo do motor**

O objetivo principal deste sistema é controlar o funcionamento do motor, através de um conjunto de sensores e atuadores. Os parâmetros de entrada, utilizados para determinar as variáveis utilizadas no controlo do motor, são providenciadas através de dois métodos, podem ser obtidos através de sensores que se encontram ligados diretamente a unidade de comando ou então encontram se disponíveis no barramento CAN vermelho. Os parâmetros presentes no barramento CAN são enviados por outras unidades de comando

do veículo. Quanto à sua localização, o sistema de controlo do motor encontra-se fixo ao motor na sua lateral esquerda.

### **ICL – Sistema do painel de instrumentos**

Esta unidade de controlo recebe, maioritariamente, mensagens do barramento CAN provenientes de outras unidades de comando. Estas mensagens podem conter valores dos vários sinais e/ou informações que serão posteriormente apresentadas ao condutor no painel de instrumentos. No entanto, esta unidade de controlo também envia mensagens para o barramento, relacionadas com hora, data e iluminação da cabine. Permitindo assim ao utilizador enviar alguns comandos através de botões que se encontram ligados ao painel de instrumentos. Esta unidade é parte integrante do painel de instrumentos, que no caso de estudo, está acoplada ao controlador R480. O barramento CAN utilizado por esta unidade de comando é o barramento amarelo.

### **VIS – Sistema de Visibilidade**

A finalidade do sistema de visibilidade é controlar a iluminação externa e as funções do limpa-vidros com o objetivo de melhorar a visibilidade do motorista e assegurar que o veículo seja visível no trânsito. Para isso é requerida informação presente no barramento CAN amarelo, como exemplo, saber qual a carga (em quilogramas) em cada eixo de modo a ajustar a altura dos faróis. No caso de estudo esta unidade situa-se dentro do controlador R480.

### **APS – Sistema de Ar Comprimido**

A unidade do APS regula a alimentação do ar do compressor. Este módulo também tem a função de secar e limpar o ar comprimido. Em caso de queda de pressão de um dos circuitos, a unidade protege o sistema contra a perda de pressão nos outros circuitos. Sabendo que o ar comprimido é um sistema crítico para a travagem dos veículos pesados, caso seja identificado uma avaria no sistema é logo transmitida essa informação através do barramento CAN amarelo. A unidade de APS também se encontra dentro do controlador R480.

De modo a facilitar trabalhos futuros, apresenta-se de seguida uma tabela (Tabela 1) que contém a designação das unidades, a sua localização e qual o barramento CAN que utilizam.

**Tabela 2.1 - Resumo das unidades de comando presentes no caso de estudo.**

<b>Sigla</b>	<b>Designação</b>	<b>Localização</b>	<b>Barramento</b>
<b>COO</b>	Coordenador	Controlador R480	Todos
<b>EMS</b>	Sistema de controlo do motor	Motor DT12 17	Vermelho
<b>ICL</b>	Sistema do painel de instrumentos	Controlador R480	Amarelo
<b>VIS</b>	Sistema de visibilidade	Controlador R480	Amarelo
<b>APS</b>	Sistema de Ar Comprimido	Controlador R480	Amarelo

Para o presente projeto, desenvolvimento de um interface de comunicação para motores de veículos pesados, as unidades de controlo que apresentam uma maior importância são a unidade de controlo do motor (EMS) e o coordenador (COO). De modo a facilitar o estudo destas unidades de comando, apresenta-se na Figura 2.6 um esquema com o conjunto de sensores e atuadores associados a cada uma delas. Este esquema é uma versão simplificada dos esquemas elétricos originais provenientes do *software* Scania Multi@[6]–[8], com o objetivo de facilitar a interpretação/análise das unidades e dos sensores/atuadores associados. No que diz respeito ao COO, o esquema apenas apresenta os sensores e ligações relevantes para este projeto, os restantes componentes/ligações foram ocultados.

Como se pode ver na Figura 2.6 os sensores do pedal do acelerador, pedal de travão, pedal da embraiagem e botões do *cruise control* estão diretamente conectados ao coordenador. No coordenador é feito o processamento da informação dos sensores e esta é, posteriormente, enviada para a unidade de controlo do motor através do barramento vermelho. Neste barramento também se pode encontrar outros parâmetros, tais como temperatura ambiente, pressão barométrica, entre outras informações úteis ao funcionamento do motor. Tal como o coordenador, a unidade de controlo do motor também adquire informações úteis provenientes dos sensores.

De acordo com a pesquisa efetuada, o barramento CAN verde utiliza o protocolo de comunicação definido pela norma **SAE J1939**.

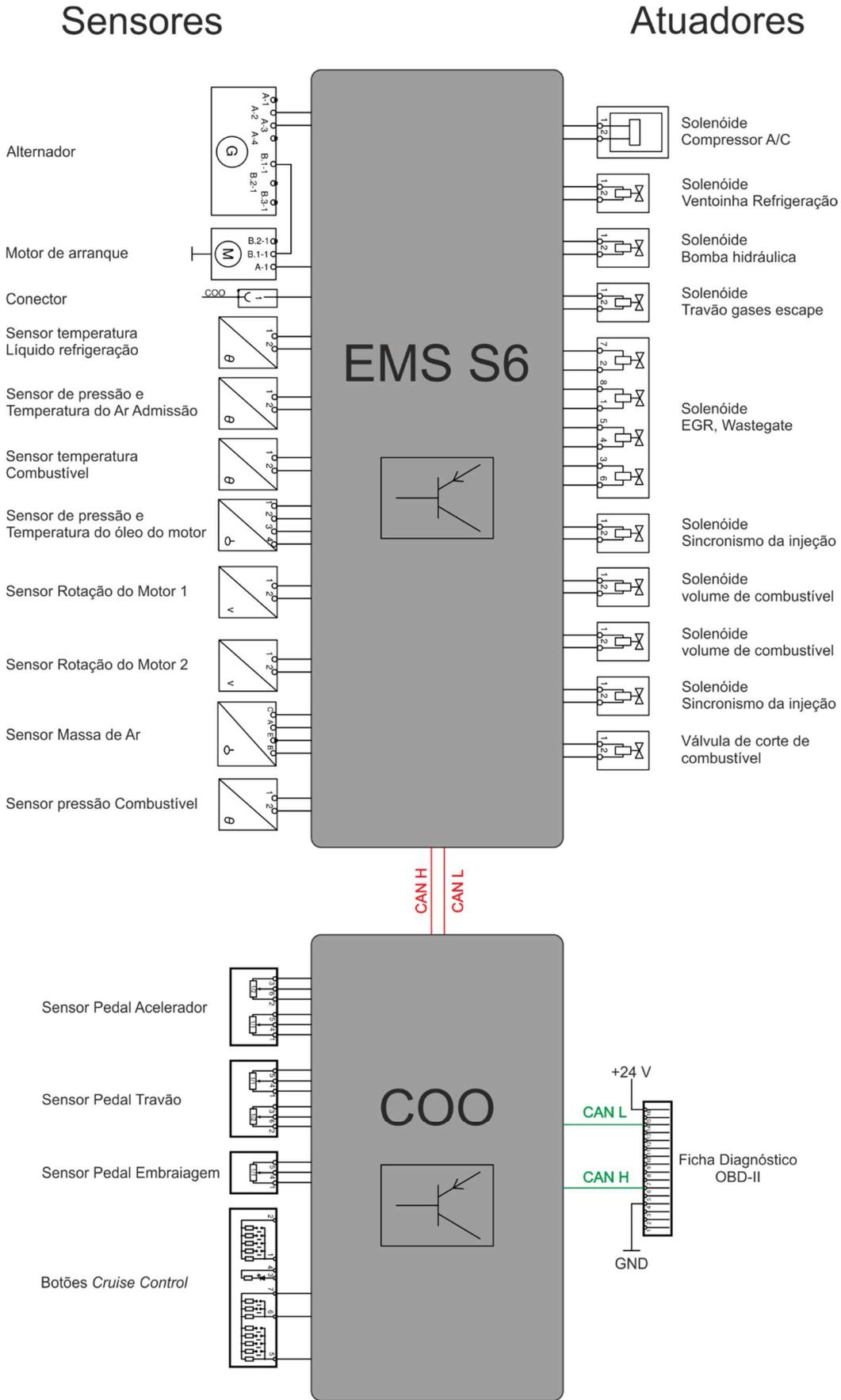


Figura 2.6 - Esquema das unidades EMS e COO[6]–[8].

### 2.1.3. C - Controlador R480

---

A função deste controlador é possibilitar uma interação com o motor quando este se encontra desmontado do veículo pesado. Com este componente é possível controlar a rotação do motor com dois métodos diferentes, mais precisamente, um acelerador manual ou através do sistema de *cruise control*. Na sua constituição encontra-se também o canhão de ignição, que permite dar o sinal de arranque do motor, o painel de instrumentos original do veículo que permite visualizar os parâmetros essenciais à utilização do motor. De modo a comunicar com as unidades de comando presentes, existe um conector OBD-II (ver 2.1.4) que permite ligar um interface de diagnóstico ao coordenador e veículo. Por último, possui um interruptor de corte de emergência da energia proveniente da bateria. Apresenta-se na Figura 2.7 uma imagem do controlador R480 utilizado neste caso de estudo.



Figura 2.7 – Controlador R480.

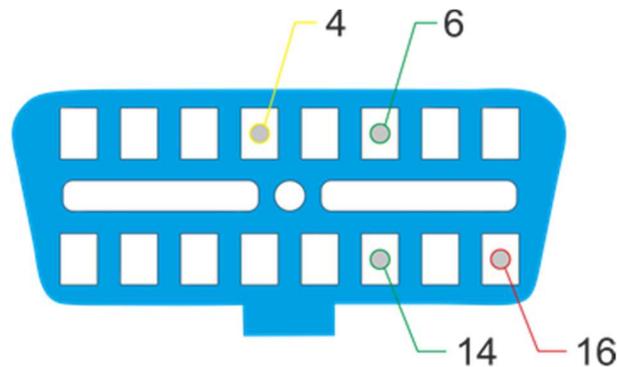
### 2.1.4. D - Conector OBD-II *Type*

---

Como é possível verificar pela Figura 2.8, o conector OBD-II disponível no controlador R480 apresenta ligação em apenas 4 pinos. Este conector tem a particularidade de ser, segundo a norma SAE J1962, do tipo “*Type B*” (utilizado em veículos pesados), isto quer dizer que no pino 16, o valor de tensão é de 24 volts. Os restantes pinos também apresentam funções de acordo com o definido na norma: pino 4 é a ligação de massa, pino 6 é o barramento CAN H e o pino 14 é o barramento CAN L.

Este conector é utilizado pelo interface de diagnóstico da Scania® para realizar o diagnóstico aos veículos Scania®, logo comunica com o coordenador que reúne a

informação de diagnóstico das unidades ligados nos barramentos amarelo e vermelho, ou permite enviar mensagens para comunicar com os sistemas ligados nos barramentos amarelo e vermelho, utilizando o coordenador como interface/repetidor de mensagens.



**Figura 2.8 - Conector OBD-II Type B.**  
4) GND, 6) CAN H, 14) CAN L e 16) +24 V.

## 2.2. Comunicação em veículos

---

Nesta secção aborda-se os vários tipos de comunicação utilizados em veículos, mais propriamente, em veículos pesados. Inicia-se pela norma SAE J1939, de seguida aborda-se o modo diagnóstico dedicado da Scania® e por fim o modo de diagnóstico OBD-II genérico.

### 2.2.1. Norma SAE J193

---

A norma SAE J1939 é definida pela *Society of Automotive Engineers* (SAE) e é principalmente usada em veículos pesados, máquinas agrícolas e máquinas industriais. Esta norma utiliza, como meio físico de transmissão de dados, um barramento CAN. Através deste barramento é possível enviar mensagens que contém um campo de dados constituído por 8 bytes. Utiliza um identificador de 29 bits, que identifica a mensagem, qual

a sua prioridade e a quem se destina. Segundo a norma, este barramento possui uma velocidade de transmissão de dados de 250 kbit/s. A norma J1939 é muito semelhante à antiga norma SAE J1708 e SAE J1587 que foram criadas em 1985. Estas normas mais antigas (SAE J1708 e J1587) foram também criadas com o intuito de regular a comunicação nos veículos pesados. Utilizam um meio de comunicação mais arcaico, uma adaptação do protocolo RS-485, que opera a velocidades de comunicação mais baixas que o barramento CAN. O valor do *baudrate* utilizado pela comunicação antiga é de 9600 bps, muito inferior a 250 000 bps, velocidade de comunicação verificada no barramento verde do sistema em estudo.

## 2.2.2. Barramento CAN

---

A SAE J1939 utiliza como meio físico de comunicação (*Physical layer*) um barramento CAN (*Controller Area Network*, ISO 11998). O barramento CAN é um protocolo de comunicação que tem como base o modelo OSI.

### Modelo OSI

O Modelo OSI *Seven* foi desenvolvido pela *International Organization for Standardization* (ISO). Tal como se representado na Figura 2.9, este modelo é composto por sete camadas com o intuito de normalizar os protocolos de comunicação. A norma SAE J1939 encontra-se dividida por vários documentos. Estes documentos definem cada uma das camadas do modelo OSI, sendo estas identificadas no nome documento, como por exemplo, o documento SAE J1939-71 define a sétima camada (*application layer document*). Será apenas descrito, neste relatório, as camadas mais relevantes para este projeto.

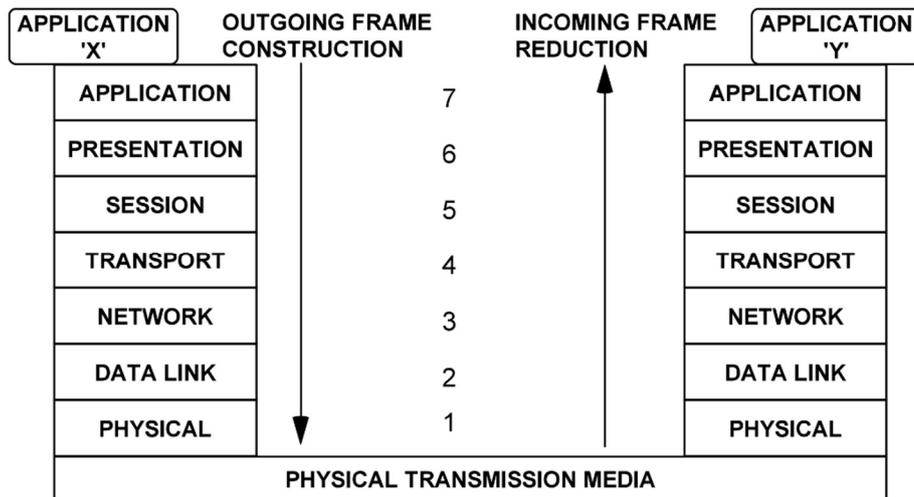


Figura 2.9 - OSI Seven Model[9].

### **Physical Layer e Data Link layer**

Neste ponto aborda-se as duas camadas inferiores do modelo OSI Seven. Com o intuito de facilitar a compreensão destas camadas, apresenta-se a Figura 2.10. Nesta figura apresenta-se um barramento CAN, um CAN *transceiver*, um controlador CAN e um microprocessador. A primeira camada (*Physical Layer*) define as características físicas do barramento, enunciando: meio físico de comunicação, níveis tensão, velocidade de transmissão de dados e as resistências nas extremidades do barramento. A norma define como meio físico de comunicação um par de fios entrelaçados (CAN H e CAN L), com uma frequência de entrelaçado de 40 voltas por metro. Quanto aos níveis de tensão, a norma define como valor recessivo 2,5 V, ou seja, ambos os fios (CAN H e CAN L) em modo de repouso possui um nível de tensão de 2,5 V. Os níveis de tensão associados ao modo dominante são: CAN H 3,5 V e CAN L 1,5 V. A velocidade de transmissão de dados definida pela norma é de 250 Kbits/seg. No fim do par de fios entrelaçado, em ambos os lados, a norma define a utilização de uma resistência de 120 Ω[10].

A segunda camada do modelo OSI (*Data Link Layer*) em como função providenciar uma transferência de dados fidedigna usando o barramento CAN. Consiste no envio dos dados com a necessária sincronização, controlo de sequência, controlo de erros e controlo de fluxo. Segundo este documento, SAE J1939-21, as mensagens presentes num barramento CAN utilizam um *extended CAN Identifier* (29 bit). Na Figura 2.10, verifica-se que após o processamento no controlador CAN a informação que se utiliza é o CAN-ID e o campo dos dados.

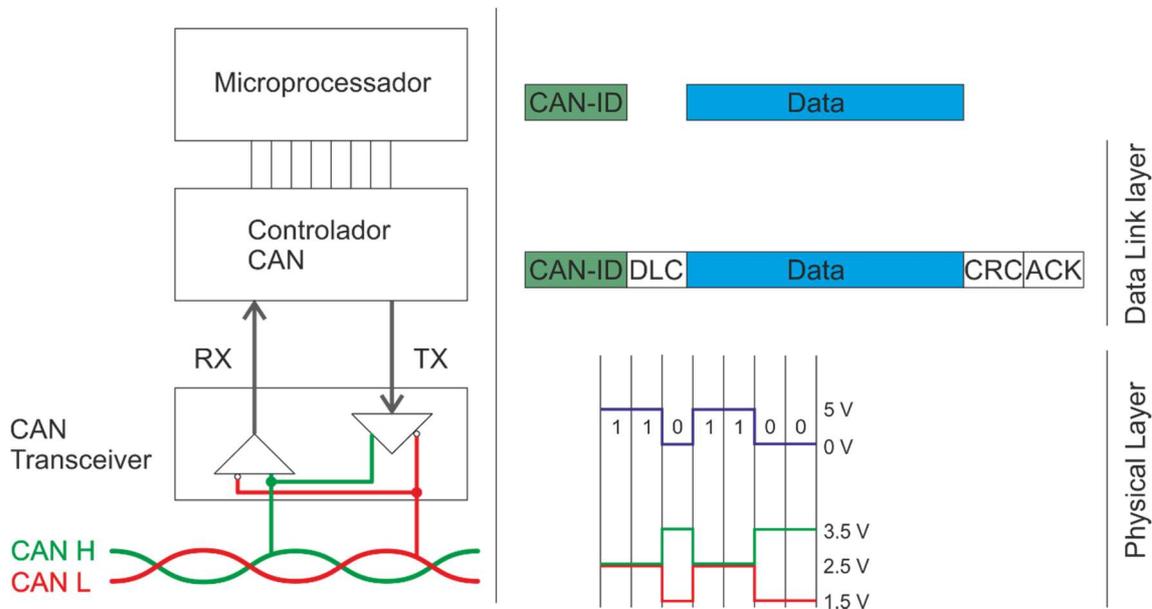


Figura 2.10 - Esquema das duas camadas inferiores do modelo OSI.

### 2.2.3. *Parameter Groups*

Os valores presentes no campo de dados das mensagens CAN correspondem a parâmetros, onde por sua vez correspondem a informação relativa a sensores do veículo ou a valores calculados pelas unidades de controlo. Estes parâmetros são associados em grupos que se encontram organizados pelo mesmo tópicos e a mesma velocidade de transmissão. A definição destes grupos de parâmetros encontra-se na *application layer document*, SAE J1939-71 [11]. O tamanho de um grupo de parâmetros não está limitado ao tamanho de uma mensagem CAN. Normalmente um grupo de parâmetros tem no mínimo 8 bytes e pode ser até 1785 bytes [12].

#### Interpretação do CAN Identifier (CAN-ID)

O *CAN Identifier* de uma mensagem definida pela norma SAE J1939 contém o *Parameter Group Number* (PGN), fonte da mensagem (SA), prioridade. A Tabela 2.2 apresenta a identificação das componentes de um identificador CAN, total de 29 bits, nos quais os 3 bits mais significativos são utilizados para definir a Prioridade da mensagem, os dois bits seguintes definem a página dos dados. Seguidamente, surge o *PDU Format* constituído por 8 bits, o *PDU Specific* através de 8 bits e por último a fonte da mensagem composta

por 8 bits. De forma a simplificar a compreensão da norma, o PGN é definido por 16 bits, os primeiros 8 bits corresponde ao PDU *Format* e os restantes 8 bit ao PDU *Specific*.

**Tabela 2.2 - Identificação dos componentes do CAN-ID.**

<b>Prioridade</b>	<b>Extensão pág. Dados</b>	<b>Pág. Dados</b>	<b>PDU Format</b>	<b>PDU Specific</b>	<b>Fonte mensagem</b>
3 bit	1 bit	1 bit	8 bit	8 bit	8 bit

### ***Parameter Group Number***

Cada grupo de parâmetros está associado um único número, ou seja, *Parameter Group Number* (PGN). Esta informação encontra-se no CAN-ID, apresentado na Tabela 2.2. Neste projeto utiliza apenas a comunicação nos veículos pesados definida pela norma J1939, ou seja, a extensão da página de dados e a página de dados é sempre 0. Na Tabela 2.3 apresenta-se uma mensagem CAN, onde é possível identificar-se o PDU *Format* e o PDU *Specific*.

**Tabela 2.3 - Exemplo de PGN inserido numa mensagem CAN.**

<b>CAN ID</b>				<b>CAN Data</b>							
<b>Priority</b>	<b>PDU Format</b>	<b>PDU Specific</b>	<b>SA</b>	<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>D5</b>	<b>D6</b>	<b>D7</b>
0C	F0	04	00	FF	FF	82	DF	1A	FF	FF	FF

Existem dois tipos de PGN definidos pela norma, os globais e os específicos.

- Os PGNs globais são usado nas mensagens gerais, ou seja são usadas para enviar informações para todos os dispositivos ligados no barramento. Nestas mensagens o PDU *Format* é sempre superior a 240 (0xF0), inclusive. O PDU *Specific* é a extensão do grupo. Considerando o exemplo:

**PGN** = 0xF004. Neste exemplo, o bit correspondente a extensão da página dos dados e a página dos dados são ambos "0". O PDU *Format* é 0xF0 e o PDU *Specific* é 0x04.

- Quanto aos PGNs específicos, estes são utilizados na comunicação ponto a ponto. O PDU *Format*, a extensão da página dos dados e a página dos dados são utilizados para identificar o *Parameter Group*. O PDU *Format* neste caso é sempre inferior a 239 (0xEF),

inclusive. O *PDU Specific* corresponde ao destinatário da mensagem. Considerando o exemplo:

PGN = 0xED00. Os bits relacionados com a página dos dados possuem o valor "0", o *PDU Format* possui o valor 0xED e quanto ao *PDU Specific* possui o valor 0x00.

A norma reserva alguns destes PGNs para os fabricantes de modo a que estes utilizem o mesmo barramento CAN sem entrar em conflito com a norma. Estes PGNs reservados são identificados através dos *PDU Format* que corresponde ao valor 0xFF. Portanto todas as mensagens presentes no barramento com o *PDU Format* correspondente a 0xFF, não se encontram descritas na norma SAE J1939. Para decodificar a informação contida nestas mensagens é necessário consultar bibliografia do fabricante do veículo pesado ou aplicar técnicas de engenharia inversa.

### **Exemplo de um PGN**

A norma J1939 descreve quais os parâmetros presentes em cada PGN. Na Tabela 2.4 apresenta-se um exemplo de um PGN com o valor em decimal 61444 (0xF004), este PGN segundo a norma é específico para o sistema eletrónico de controlo do motor e como se pode ver, os campos *Data Length*, *Extended Data Page*, *Data Page*, *PDU Format*, *PDU Specific* e *Default Priority*, apresentam, respetivamente, os valores de 8, 0, 0, 240 (0xF0), 4 (0x04) e 3.

Adicionalmente a norma define as variáveis sua colocação e número de bits utilizado para a restante mensagem. Por exemplo a variável *Engine Torque Mode* encontra-se nos primeiros 4 bits da mensagem e o valor do *Engine Speed* encontra-se nos bytes 4 e 5.

Tabela 2.4 - Exemplo de um PGN descrito na norma[11].

PGN 61444		Electronic Engine Controller 1	EEC1
<i>Engine related parameters</i>			
<i>Transmission Repetition</i>		<i>engine speed dependent</i>	
<i>Rate:</i>			
<i>Data Length:</i>	8		
<i>Extended Data Page:</i>	0		
<i>Data Page:</i>	0		
<i>PDU Format:</i>	240 (0xF0)		
<i>PDU Specific:</i>	4(0x04)		
<i>Default Priority:</i>	3		
<i>Parameter Group Number:</i>	61444 (0xF004)		
<i>Start Position</i>	<i>Length</i>	<i>Parameter NAME</i>	<i>SPN</i>
1.1	4 bits	Engine Torque Mode	899
2	1 byte	Driver's Demand Engine - Percent Torque	512
3	1 byte	Actual Engine - Percent Torque	513
4-5	2 bytes	Engine Speed	190
6	1 byte	Source Address of Controlling Device for Engine Control	1483
7.1	4 bits	Engine Starter Mode	1675
8	1 byte	Engine Demand – Percent Torque	2432

### Suspect Parameter Number (SPN)

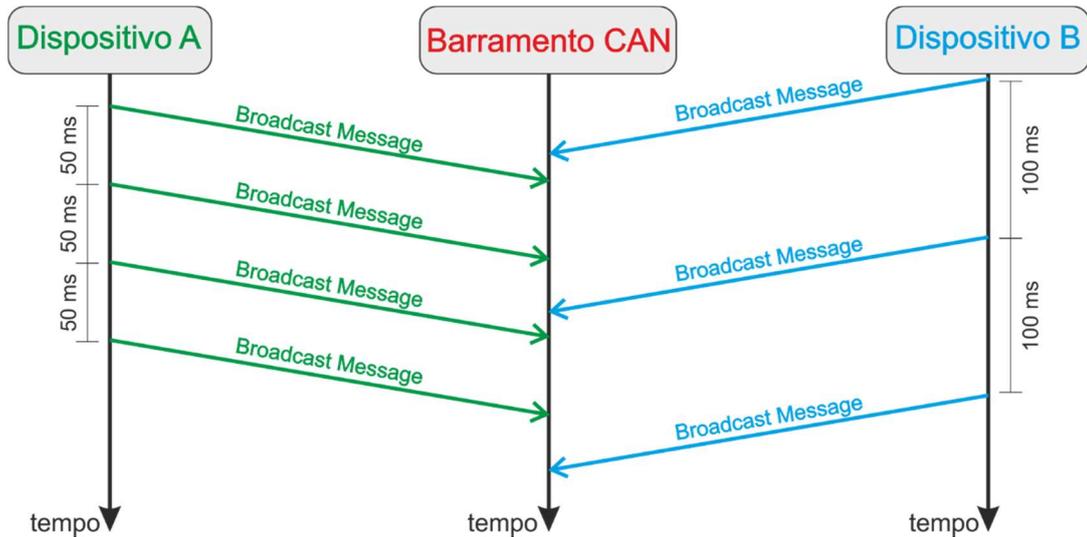
O *Suspect Parameter Number* é definido pela norma de modo a facilitar a decodificação do valor real e descreve a resolução e o *offset*. Assim é possível converter o valor em hexadecimal presente no barramento CAN em valor real.

## 2.2.4. Tipos de mensagem

A norma define dois tipos de mensagens, as mensagens *Broadcast* e as mensagens do tipo Pergunta-Resposta (*peer-to-peer*). Quando o seu campo dos dados não excede os 8 bytes utiliza uma comunicação comum através do barramento CAN, mas se o campo dos dados é superior a 8 bytes é necessário recorrer a um protocolo de transporte de mensagens, utilizando o TP.CM e o TP.DT descritos na norma SAE J1939-21.

No caso de uma comunicação *Broadcast* as mensagens são enviadas para o barramento ciclicamente não sendo necessário qualquer tipo de *acknowledge*. A Figura 2.11 apresenta um exemplo de uma sequência de mensagem do tipo *broadcast*. Como se pode verificar,

cada dispositivo envia a sua mensagem sempre com o mesmo intervalo de tempo, correspondendo a uma frequência constante ao longo do tempo. Assim a informação presente nas mensagens é enviado constantemente ao longo do tempo independentemente dos restantes dispositivos presentes no barramento.



**Figura 2.11 – Exemplo de uma sequência de mensagens *Broadcast*.**

Relativamente as mensagens do tipo pergunta-resposta (peer-to-peer) são tipicamente utilizadas na comunicação entre um equipamento de diagnóstico e uma unidade de comando. Apresenta-se na Figura 2.12 exemplos de mensagens pergunta-resposta e como se pode verificar um dos dispositivos envia a pergunta para o barramento e só depois o segundo dispositivo é que responde a pergunta. Este tipo de comunicação é mais lento pois necessita de duas mensagens para transmitir informação.

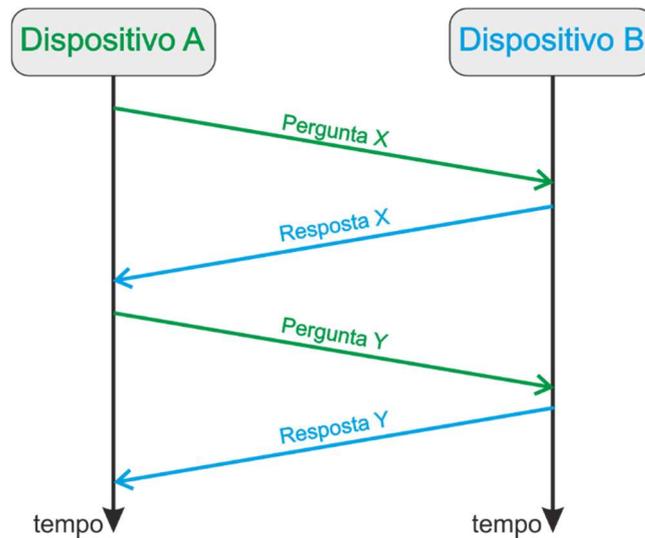


Figura 2.12 – Exemplo de mensagens Pergunta-Resposta.

## 2.2.5. Exemplo de uma mensagem *Broadcast*

---

Apresenta-se na Tabela 2.3 uma mensagem exemplo tem como intuito mostrar como interpretar uma mensagem definida pelo protocolo SAE J1939. A tabela composta pelo CAN-ID e o Data Field. Posteriormente descodifica-se e analisa-se a informação presente na mensagem.

Os primeiros dois bytes = 0x0C = 0000 1100 em formato binário. Os primeiros 3 bits não são usados porque consiste num identificador de 29 bits. Os 3 seguintes definem a prioridade da mensagem que neste caso é 3. De seguida é o bit reservado e a página dos dados (*data page*), que são usados para determinar o PGN completo.

O último byte do CAN-ID corresponde ao emissor da mensagem (identifica a unidade que enviou a mensagem) que neste caso é 0x00, que corresponde a EMS.

O PGN = 0xF004 segundo a norma SAE J1939/71 corresponde a *Electric Engine Controller #1* (EEC1). Este documento também descreve os parâmetros e as suas posições nos dados da mensagem.

Em relação aos dados da mensagem, quando se encontra o valor 0xFF significa que esse parâmetro não se encontra disponível. No exemplo os dados nos bytes 1, 2, 6, 7 e 8 não se encontram disponíveis[11].

Segundo a norma o data byte 3 corresponde a carga atual do motor em percentagem, que tem o valor em hexadecimal de 0x82. Este valor corresponde a 130 em decimal, convertendo segundo a norma corresponde a escala de 1% com um *offset* de -125. Assim o valor é 5% de carga atual do motor.

Na posição 4 e 5 corresponde a rotação do motor, sendo o byte 4 o LSB. O valor 0x1ADF = 6879 decimal. Usando o fator de escala disponibilizado pela norma que é 0,125 RPM/bit e sabendo que o *offset* é 0, após conversão corresponde a 859,875 RPM[13].

## 2.2.6. Diagnóstico OBD-II Genérico

---

A norma SAE J1939 reserva um espaço para implementar o diagnóstico OBD-II genérico que tem como objetivo uniformizar o diagnóstico *on-board* nos veículos. Através deste método de diagnóstico é possível recolher informações dos sensores do veículos, verificar a existência de erros e apagar esses erros, entre outras funções. Relativamente ao caso de estudo o diagnóstico OBD-II genérico é implementado através de duas normas, a ISO 15765-4 e a SAE J1979. A norma ISO 15765 define o formato do *extended* CAN-ID que se apresenta na Tabela 2.5. O CAN-ID é definido com prioridade 6, o PDU *Format* com o valor “DA” em hexadecimal, o PDU *Specific* define o destinatário da mensagem e o SA define o emissor da mensagem.

**Tabela 2.5 – Formato do *extended* CAN-ID para diagnóstico OBD-II genérico.**

CAN ID			
Priority	PDU Format	PDU Specific	SA
18	DA	“Target Adress”	“Source Adress”

Quanto ao campo de dados de uma mensagem CAN de diagnóstico OBD-II genérico é definido através da norma SAE J1979. A norma define vários serviços de diagnóstico, tais como solicitar os valores de sensores, ler e apagar códigos de erro, calibração de atuadores, entre outros. Apresenta-se apenas o serviço de pedido de dados atuais, que corresponde ao modo 01. De seguida é necessário identificar qual o parâmetro que se

pretende adquirir (PID) e para tal é necessário consultar a norma. Apresenta-se na Tabela 2.6 a definição de três parâmetros expondo o PID definido, a sua resolução e *offset*.

**Tabela 2.6 – Definição de alguns parâmetros no modo 01[14].**

PID	Nome do parâmetro	Resolução	Offset
04	Calculated engine load	0,4 %/bit	0
05	Engine Temperature	1 °C/bit	-40
0C	Engine RPM	0,25 RPM/bit	0

Em forma se resumo expõe-se uma mensagem exemplo através da Tabela 2.7, onde se pode visualizar qual o CAN-ID utilizado e o campo de dados de uma mensagem CAN em modo diagnóstico OBD-II genérico. Relativamente ao CAN-ID verifica-se que o destinatário da mensagem é o endereço 0x00 que corresponde a unidade de controlo do motor e que a fonte da mensagem é o endereço 0xFA que corresponde a um equipamento de diagnóstico. No campo de dados apenas são enviados três *bytes*, onde o primeiro indica o número de *bytes* constituintes no campo de dados, o segundo *byte* corresponde ao serviço de diagnóstico e o último *byte* corresponde à identificação do parâmetro desejado.

**Tabela 2.7 – Mensagem de pedido da temperatura do motor.**

CAN ID				CAN Data							
Priority	PDU Format	PDU Specific	SA	D0	D1	D2	D3	D4	D5	D6	D7
18	DA	00	FA	02	01	05	00	00	00	00	00

## 2.3. Outros trabalhos na área

---

Nesta secção apresenta-se alguns trabalhos realizados na área do projeto desenvolvido. Verifica-se que não se encontra muita informação relativa a testes de validação de motores após o seu acondicionamento e controlo de motor de veículos pesados. Contudo apresenta-se vários trabalhos realizados onde possuem objetivos diversos, desde validação de dados adquiridos através de mensagens CAN até aquisição de dados e controlo de um banco de ensaio de motores.

### 2.3.1. Medição de consumo atual de combustível

Os autores H. Fonseca, *et al* referem que a precisão da medição do consumo atual de combustível é uma característica muito importante, pois para além de informar o condutor com este parâmetro é possível desenvolver ferramentas de apoio à condução visando uma condução mais eficiente a nível de combustível. O trabalho realizado pelos autores apresentam dois novos métodos de medição do consumo de combustível de motores Otto e Diesel. O primeiro método utiliza o valor da massa de ar admitida em conjunto com a sonda lambda. Relativamente ao segundo utiliza um sensor de fluxo volumétrico baseado na equação de Bernoulli. Ambos os métodos foram ensaiados num banco de potência e ambos apresentaram uma redução no erro de 79% e 81%, respetivamente, quando comparados com o valor obtido através de diagnóstico OBD. Apresenta-se um dos resultados obtidos com este trabalho na Figura 2.14 onde se apresenta a comparação do consumo de combustível entre o segundo método proposto e a medição digital do consumo [15].

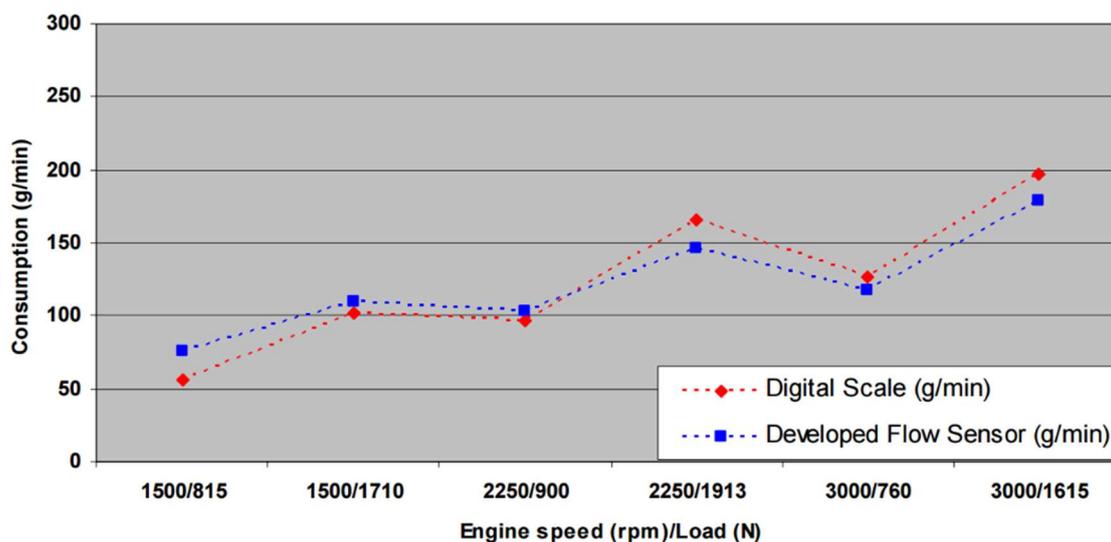


Figura 2.13 – Comparação do consumo de combustível entre o segundo método proposto e a medição digital do consumo.

### 2.3.2. Precisão dos dados presentes num barramento CAN

---

S. Marx analisou em simultâneo três diferentes métodos de aquisição de dados com o intuito de validar a veracidade dos valores adquiridos no barramento CAN. Adicionalmente determinou a precisão dos valores calculados pela unidade, ou seja não são diretamente medidos, pois estes valores possuem uma elevada importância durante uma investigação. Um dos parâmetros calculado pela unidade de comando que possui maior importância é o consumo de combustível atual. A comparação entre os valores calculados pela unidade e uma medição física indica a precisão dos dados presentes nas mensagens. Os resultados obtidos pelo S. Marx, expostos na Figura 2.14, mostram que as diferenças entre os valores são muito baixas ( $\pm 5\%$ ). Estes resultados indicam que o valor presente no barramento CAN pode ser usado para investigação [16].

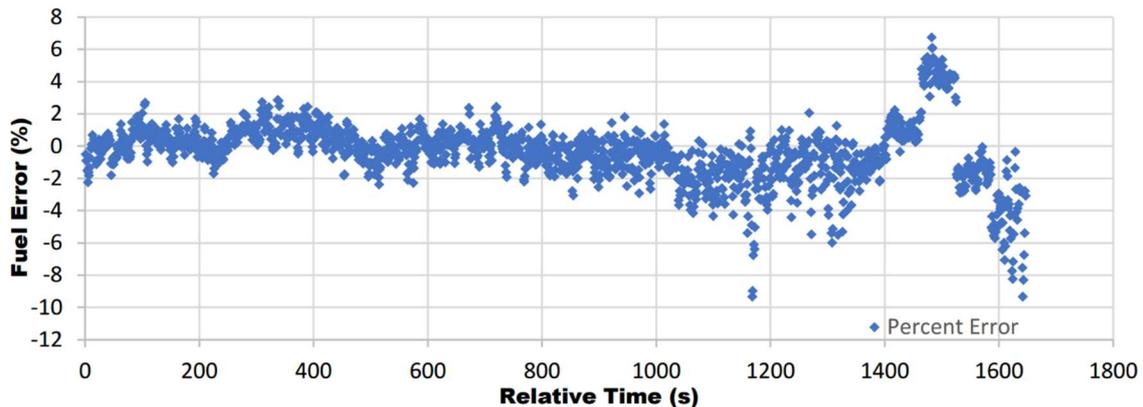


Figura 2.14 – Percentagem de erro do cálculo do combustível[16].

### 2.3.3. Resolução de avarias em motores

---

O âmbito da tese de mestrado de Oliver Kiffer é a resolução de avarias em motores do fabricante Scania® através de sensores externos. Kiffer afirma que o custo de reparação de um veículo é proporcional ao tempo de reparação. Apresenta também que os veículos atuais possuem um sistema de diagnóstico *on-board* (OBD-II) que avisa o condutor do mau funcionamento de algum parâmetro, mas este sistema não é capaz de identificar o problema isoladamente, onde por vezes é necessário recorrer a uma medição externa para isolar e resolver o problema[17].

Kiffer tinha como objetivo desenvolver um equipamento que permita analisar em simultâneo os valores do diagnóstico interno e valores adquiridos por sensores externos. O protótipo desenvolvido permite a resolução de uma avaria induzida num intervalo de tempo inferior quando comparado com o método de resolução de avarias tradicional[17].

### 2.3.4. Aquisição via OBD

---

A investigação de K. Smith e J. Miller tem como base o registo de vários parâmetros provenientes de vários automóveis. Para isso foi necessário desenvolver um dispositivo capaz de comunicar com um servidor remoto os parâmetros adquiridos pela comunicação OBD e a localização atual do automóvel. O dispositivo desenvolvido pelos investigadores é composto por um Arduino® UNO capaz de integrar um módulo de GPS e GSM que permite saber a localização e o envio de dados, respetivamente. Um módulo de OBD também foi integrado no dispositivo com a função de adquirir os parâmetros disponíveis. Os autores referem que as principais vantagens são o acesso por completo aos parâmetros disponíveis na comunicação OBD-II, a adaptabilidade do processamento de dados e a transmissão de dados para um servidor remoto por dados móveis. Os autores afirmam também que com o desenvolvimento deste dispositivo surgiram muitas oportunidades de trabalhos futuros em diversas áreas, tais como análise de emissões, notificação automática da condição da estrada e estratégias de manutenção adaptadas[18].

### **2.3.5. Sistema automatizado de ensaios de motores**

---

Segundo L. Serrano *et al*, num teste num motor de combustão interna é necessário impor diferentes regimes de operação definidos por duas variáveis, a velocidade de rotação do motor e o binário absorvido pelo dinamómetro. Os autores afirmam que a realização de testes é mais eficiente quando estes são efetuados através de um sistema automatizado que permita a aquisição de dados e controlo motor. Após o desenvolvimento do sistema automatizado, os autores referem que existe uma melhoria de precisão durante a execução do ciclo imposto. Afirmam também que a tendência e qualidade dos resultados obtidos mostram um sinal de sucesso pois existe uma menor possibilidade de se introduzir erros durante os ensaios[19].

## 3. Descodificação dos Protocolos e Métodos de controlo

---

Neste capítulo apresenta-se os equipamentos necessários para realizar a descodificação. Posteriormente aborda-se a descodificação da norma SAE J1939, mais concretamente, a descodificação das mensagens presentes no barramento CAN do tipo *Broadcast* emitidas pelo sistema em estudo. De seguida apresenta-se o protocolo dedicado do fabricante em modo de diagnóstico (SDP3), onde se exhibe a sua descodificação e caracterização. Por fim é apresentado o modo OBD-II genérico onde se exhibe os testes de validação efetuados e seus resultados.

### 3.1. Equipamentos utilizados

---

Antes de se apresentar o procedimento de descodificação, exhibe-se os equipamentos necessários para descodificar os vários protocolos. Para descodificar e analisar o sistema necessita-se de três equipamentos específicos, um *CAN Analyser*, um osciloscópio digital e um equipamento diagnóstico de veículo pesados. De seguida apresenta-se os equipamentos utilizados.

#### **CAN Analyser – PEAK Systems®**

O *CAN analyser* utilizado é do fabricante PEAK Systems® com a referência PCAN-USB. Este equipamento tem como função adquirir e exhibir as mensagens presentes num barramento CAN em tempo real. O *CAN analyser* permite também enviar mensagens para o barramento ciclicamente ou quando solicitado. Uma outra funcionalidade do equipamento é a capacidade de guardar as mensagens adquiridas possibilitando assim uma posterior análise das mensagens do barramento. Apresenta-se na Figura 3.1 uma imagem do equipamento PCAN-USB da PEAK Systems®.



**Figura 3.1 - PEAK Systems CAN Analyser[20].**

### **Osciloscópio – PicoScope®**

Relativamente ao osciloscópio utilizou-se um equipamento do fabricante Picotech® com a referência PicoScope® 2205-MSO, como se pode observar na Figura 3.2. Este osciloscópio tem a capacidade de adquirir dois canais analógicos e dezasseis canais digitais em simultâneo. Quanto à frequência de aquisição possui um valor máximo de 25MHz correspondendo a 200 *MegaSamples/s*. Durante a aquisição é possível decodificar mensagem de vários protocolos de comunicação.



**Figura 3.2 - Osciloscópio da PicoScope®[21].**

### Equipamento diagnóstico – VCI 2 e SDP3

Por fim apresenta-se o equipamento de diagnóstico utilizado para validar o conteúdo das mensagens CAN. O equipamento é composto pelo *hardware* (VCI 2) e pelo *software* (SDP3) que são ambos provenientes da Scania®.

Relativamente ao *hardware*, intitulado de VCI 2, tem como objetivo estabelecer a comunicação entre um veículo pesado e um computador portátil. Esta comunicação é efetuada através da conexão da ficha OBD ao veículo pesado e na porta USB de um computador. Exibe-se uma imagem deste *hardware* na Figura 3.3.



Figura 3.3 - Equipamento diagnóstico VCI 2 da Scania®[22].

O *software* SPD3 permite visualizar as variáveis, apagar erros, ajustar parâmetros entre outras funções. Dentro deste *software*, no modo de “Verificações e ajustes” é possível ler os códigos de erro memorizados nas unidades, e proceder a sua anulação. Ainda dentro deste modo é possível ajustar alguns parâmetros, tais como o ajuste da velocidade de marcha lenta e alterar o limitador de velocidade, entre outros. Permite também, calibrar alguns atuadores, codificar unidades de comando, entre outras opções.

Na função de “Registo de Sinal” é possível adquirir e guardar algumas variáveis. Tais como, rotação do motor, temperatura do líquido de refrigeração, entre muitas outras. Durante este registo é possível seleccionar qual a frequência de aquisição destas variáveis. Após a aquisição das variáveis seleccionadas guarda num ficheiro que possibilita a sua análise a *posteriori*.

## 3.2. Norma SAE J1939

Para visualização e aquisição das mensagens *Broadcast* no barramento verde foi utilizado um *CAN analyser*, fabricante PEAK® e o *software* PCAN-View®. Colocou-se o motor a trabalhar no seu regime de marcha lenta, conectou-se o analisador CAN na porta OBD-II disponível pelo controlador R480 e, mantendo o funcionamento do motor estável, foram recolhidas as mensagens existentes no barramento verde. As mensagens recolhidas são apenas as de *Broadcast*, ou seja, são as mensagens que o Coordenador envia ciclicamente para o barramento verde.

A aquisição durou, sensivelmente, 5 segundos e recolheu-se, ao todo, 1257 mensagens. Estas mensagens são cíclicas variando a taxa de atualização de cada CAN-ID. Quanto ao número de mensagens diferentes presentes no barramento, identifica-se 28 PGNs diferentes. Com o intuito de validar a informação obtida, processada segundo a norma J1939, fotografou-se o painel de instrumentos presente no controlador R480, como se apresentada na Figura 3.4.

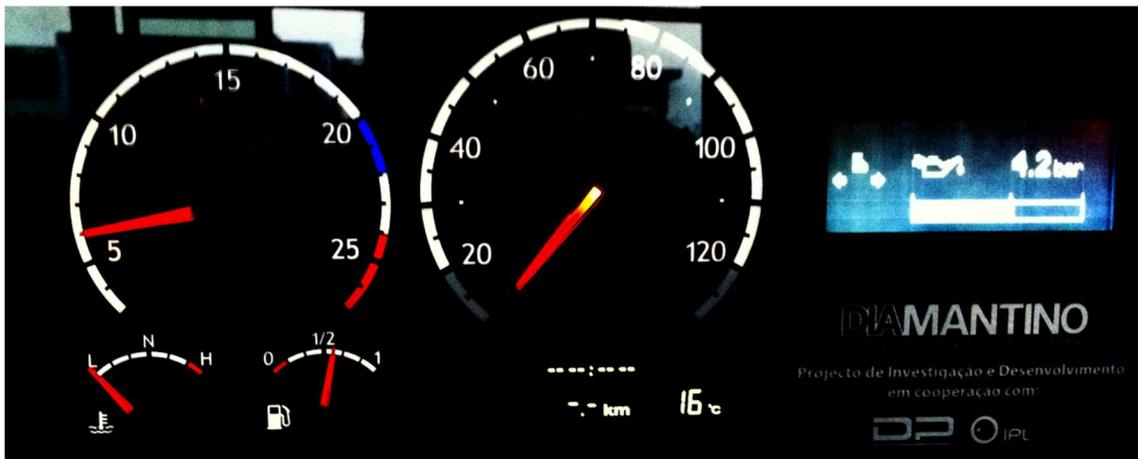


Figura 3.4 - Fotografia do painel de instrumentos durante a aquisição dos dados.

Através da análise da Figura 3.4 é possível recolher a seguinte informação: rotação do motor (sensivelmente 600 RPM), nível de combustível (próximo dos 50%), temperatura ambiente (16 °C) e o mostrador digital da pressão de óleo exibe o valor de 4.2 bar.

Através da consulta à norma SAE J1939 é possível descobrir a qual PGN se encontra os parâmetros apresentados no painel de instrumentos. A Tabela 3.1 indica qual é o PGN, a posição no campo dos dados, a resolução e o *offset* de que refere a cada um dos parâmetros.

**Tabela 3.1 – Informação da norma SAE J1939 relativa aos parâmetros apresentados.**

<b>Parâmetro</b>	<b>PGN</b>	<b>Data Byte</b>	<b>Resolução</b>	<b>Offset</b>
Rotação do motor	0xF004	4 (LSB) - 5	0,125 rpm/bit	0
Nível de combustível	0xFEFC	2	0,4 %/bit	0
Temperatura Ambiente	0xFE5F5	4 (LSB) - 5	0,0313 °C/bit	-273
Pressão de óleo	0xFEEF	4	4 kPa/bit	0

De modo a ser possível descodificar os valores presentes nas mensagens é necessário recorrer a equação (1). Esta é composta por três fatores:  $V_{Decimal}$ ,  $R_N$  e  $offset$ . O  $V_{Descodificado}$  adquire-se através da expressão descrita na equação.

$$V_{Decimal} \times R_N + offset = V_{Descodificado} \quad (1)$$

Apresenta-se seguidamente a Tabela 3.2, que contém as mensagens adquiridas no barramento verde. Seguidamente procede-se à descodificação das mensagens e validação dos valores descodificados através dos parâmetros apresentados no painel de instrumentos.

**Tabela 3.2 - Mensagens adquiridas no barramento verde.**

<b>CAN Identifier (CAN-ID)</b>			<b>Data Field (Bytes)</b>							
<b>Priority</b>	<b>PGN</b>	<b>SA</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
0C	F004	01	F0	7D	82	B9	12	FF	FF	FF
0C	F004	01	F0	7D	82	C2	12	FF	FF	FF
0C	F004	01	F0	7D	82	C6	12	FF	FF	FF
18	FEFC	27	FF	8F	FF	FF	FF	FF	FF	FF
18	FE5F5	27	C8	FF	FF	2A	24	FF	FF	FF
18	FEEF	27	A0	FF	FF	69	FF	FF	FF	0

### **Rotação do motor**

Como o valor da rotação oscila durante a aquisição tratam-se três mensagens que são apresentados na Tabela 3.3. Pela norma J1939 os bytes 4 e 5 contém o valor numérico da rotação (4 o menos significativo), dos quais resulta o valor 0x12B9 que convertido para decimal dá o valor 4793. A resolução e o  $offset$  (valor a zero) são obtidos através do SPN 190 definido na norma, com os valores de 0.125 rpm/bit e 0, respetivamente. Pela equação (1), é então possível obter a rotação em rotações por minuto. Como o valor médio da rotação do motor é muito próximo do valor indicado no painel de instrumentos, valida-se assim a descodificação da mensagem relativa à rotação do motor.

**Tabela 3.3 - Descodificação da rotação do motor.**

Hex	Dec	Resolução	Offset	Valor Descodificado	Média
12B9	4793			599,125 rpm	
12C2	4802	0,125 rpm/bit	0	600,25 rpm	600,042 rpm
12C6	4806			600,75 rpm	

### **Nível de combustível**

Os valores adquiridos relativamente ao PGN 0xFEFC são constantes ao longo da aquisição, descodifica-se então apenas uma das mensagens presente no barramento com este PGN. Na Tabela 3.4, descodifica-se a informação presente no PGN e apresenta o valor real. O valor presente na mensagem CAN é 0x8F que após conversão para decimal obtém-se o valor 143. Consultando o SPN 96, associado ao nível de combustível, adquirir-se o valor de resolução e *offset*, respetivamente, 0,4 %/bit e 0. O valor é descodificado através da equação (1), tendo como resultado o valor presente na tabela. O valor descodificado encontra-se acima de 50%, tal como indica o painel de instrumento. Assume-se então que o valor descodificado corresponde ao valor real de percentagem de nível de combustível.

**Tabela 3.4 - Descodificação do nível de combustível.**

Hex	Dec	Resolução	Offset	Valor Descodificado
8F	143	0,4 %/bit	0	57,2 %

### **Temperatura Ambiente**

Em relação a informação da temperatura ambiente, tal como o nível de combustível, não variou o seu valor durante a aquisição. O SPN associado a este parâmetro é o 171, onde se retira o valor da resolução e *offset*, respetivamente, 0.0313 °C/bit e -273. A informação deste parâmetro encontra-se nos bytes 4 e 5, que corresponde ao valor 0x242A, conseqüentemente, o valor decimal correspondente é 9458. Através da equação (1), é calculado o valor descodificado apresentado na Tabela 3.5. Quanto à informação da temperatura ambiente, o valor descodificado corresponde ao valor apresentado no visor do painel de instrumentos, é validado assim este parâmetro.

**Tabela 3.5 - Descodificação da temperatura ambiente.**

Hex	Dec	Resolução	Offset	Valor Descodificado
242A	9258	0,0313 °C/bit	-273	16,3 °C

### **Pressão de óleo**

Por fim, descodifica-se o valor da pressão de óleo presente no barramento CAN. As mensagens adquiridas com o PGN 0xFEEF assumem um valor constante ao longo da aquisição, no byte 4. O valor presente na mensagem corresponde a 0x69, que após conversão, obtém-se o valor decimal de 105. Quanto ao SPN, a norma define o 100 para o parâmetro pressão de óleo. Na Tabela 3.6 é descodificado o valor da pressão de óleo através da equação (1). O valor descodificado corresponde a 4,2 bar assumindo o mesmo valor apresentado no painel de instrumentos.

**Tabela 3.6 - Descodificação da pressão de óleo.**

Hex	Dec	Resolução	Offset	Valor Descodificado
69	105	4 kPa/bit	0	420 kPa

### **Validação**

Como conclusão dos resultados obtidos na presente seção, apresenta-se na Tabela 3.7 uma comparação entre os valores apresentados no painel de instrumentos e os valores descodificados das mensagens presentes no barramento CAN. Confirma-se que todos os parâmetros estão de acordo com o valor mostrado no painel de instrumentos, assume-se então que o protocolo utilizado no barramento CAN verde em modo *broadcast* é o SAE J1939.

**Tabela 3.7 - Comparação dos parâmetros escolhidos para validar a norma SAE J1939.**

Parâmetro	Valor indicado (ICL)	Valor descodificado (CAN)
Rotação do motor	Apróx. 600 RPM	600,042 RPM
Nível de combustível	Mais do que 50 %	57,2 %
Temperatura Ambiente	16 °C	16,3 °C
Pressão de óleo	4,2 bar	4,2 bar

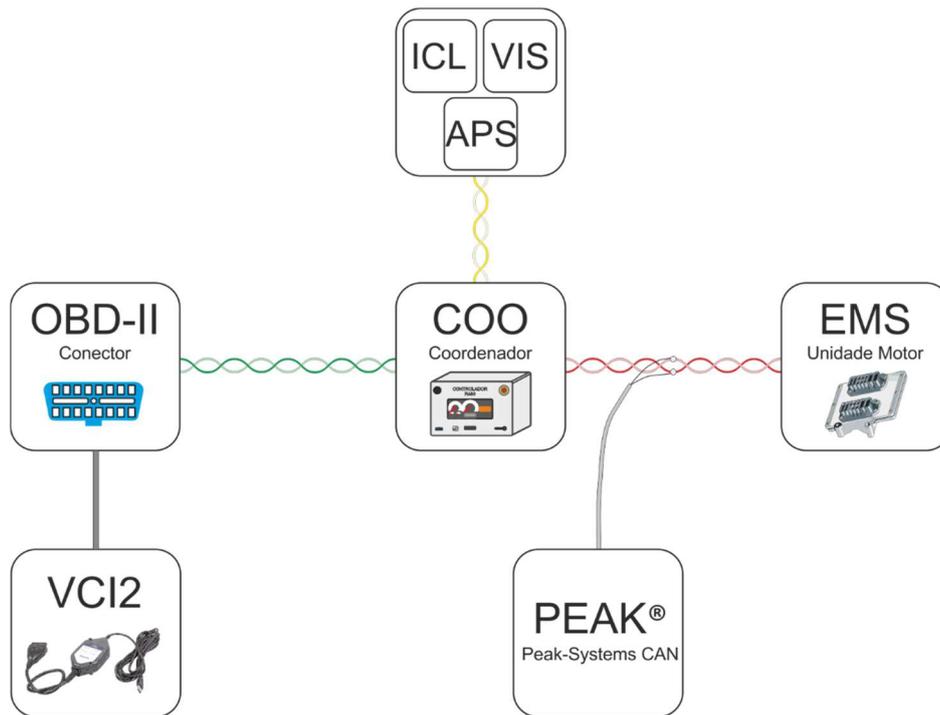
### 3.3. Modo Diagnóstico (SDP3)

---

Aborda-se nesta secção o modo de diagnóstico do *software* SDP3 e as mensagens CAN que este utiliza. O *software* SDP3 utiliza o interface de diagnóstico VCI2 que comunica com o barramento CAN verde. Este barramento encontra-se disponível no conector OBD-II nos pinos 6 e 14, que corresponde respetivamente, ao CAN H e CAN L.

Para realizar os testes necessários à caracterização do modo de diagnóstico, utilizou-se o equipamento de diagnóstico VCI2, conectado à ficha OBD-II, e o interface CAN conectado ao barramento vermelho. A Figura 3.5, ilustra o esquema de ligação dos componentes ao *hardware* disponibilizado. Após a montagem de todos os componentes, procedeu-se ao início de aquisição dos valores de parâmetros disponíveis pela *software* de diagnóstico SDP3. Utilizou-se a função registo de sinal para todos os parâmetros disponíveis, solicitando-os individualmente. Durante esta solicitação, o interface CAN adquiriu as mensagens presentes no barramento vermelho de modo a identificar a “pergunta” efetuada pelo equipamento de diagnóstico e a resposta da EMS.

Durante as várias solicitações efetuadas verificou-se que o CAN-ID da pergunta (request) é constante assumindo o valor de “18DA00FA”. Quanto a resposta da unidade o CAN-ID que inerente assume o valor “18DAFA00”. Ao longo das 75 variáveis disponíveis o CAN-ID é sempre constante variando apenas o campo dados associado a cada variável. A Tabela 10.2, em anexo, apresenta o campo de dados correspondentes a cada variável.



**Figura 3.5 - Esquema de ligação do VCI2 e o interface CAN.**

Após a identificação do campo de dados de todas as variáveis procedeu-se à descodificação de duas variáveis, rotação do motor e posição de acelerador. A Tabela 3.8 demonstra a descodificação do conteúdo da mensagem proveniente da EMS, validando assim mais um método de aquisição de parâmetros através do barramento CAN. Em termos de facilidade de acesso, este método só necessita de se conectar ao barramento verde que se encontra disponível na ficha OBD-II.

**Tabela 3.8 - Descodificação de duas mensagens de diagnóstico SDP3.**

		CAN-ID	DLC	Data Bytes													
				0	1	2	3	4	5	6	7	HEX	Decimal	Decoded	Real		
Engine Speed	Pergunta	18DA00FA	8	04	2F	FB	69	01	B1	00	00	0000	0	0	RPM	0	RPM
	Resposta	18DAFA00	8	06	6F	FB	69	01	00	00	00	1770	6000	600	RPM	599	RPM
	Pergunta	18DA00FA	8	04	2F	FB	69	01	B1	00	00	2210	8720	872	RPM	872	RPM
	Resposta	18DAFA00	8	06	6F	FB	69	01	17	70	00						
	Pergunta	18DA00FA	8	04	2F	FB	69	01	B1	00	00						
	Resposta	18DAFA00	8	06	6F	FB	69	01	22	10	00						
Accelerator Position	Pergunta	18DA00FA	8	04	2F	FB	3F	01	18	00	00	0000	0	0	%	0	%
	Resposta	18DAFA00	8	06	6F	FB	3F	01	00	00	00	0004	4	0,4	%	0,4	%
	Pergunta	18DA00FA	8	04	2F	FB	3F	01	18	00	00	03E8	1000	100	%	100	%
	Resposta	18DAFA00	8	06	6F	FB	3F	01	00	04	00						
	Pergunta	18DA00FA	8	04	2F	FB	3F	01	18	00	00						
	Resposta	18DAFA00	8	06	6F	FB	3F	01	03	E8	00						

### 3.4. Modo OBD-II Genérico

Com a descodificação das mensagens de diagnóstico identificou-se o CAN-ID utilizado para solicitar informação à unidade do motor. Posteriormente acedeu-se á norma SAE J1939, onde expõe outro método adquirir alguns parâmetros. Este modo utiliza o mesmo CAN-ID para efetuar a solicitação, com o valor “18DA00FA”. A EMS utiliza o CAN-ID “18DAFA00” para responder as solicitações. Quanto ao campo de dados, estes são definidos pela norma ISO 15765. Na Tabela 3.9 apresenta-se a solicitação de um parâmetro com o intuito de validar este método de aquisição. O parâmetro solicitado corresponde a rotação do motor que corresponde ao valor “0C” em hexadecimal.

Após o envio desta mensagem, a unidade enviou a mensagem descrita na Tabela 3.10. Segundo a norma ISO 15765, a informação solicitada encontra-se nos bytes D3 e D4. Obtém-se então o valor, em hexadecimal, 964 que convertido para decimal assume o valor 2404. Para descodificar este valor, segundo a norma, é necessário dividir por 4. Através desta operação obtém-se o valor descodificado de 601 rpm. Conclui-se então que é possível adquirir parâmetros através do modo OBD-II genérico.

Tabela 3.9 - Mensagem do modo OBD-II genérico a solicitar a rotação do motor.

CAN ID				CAN Data							
Priority	PDU Format	PDU Specific	SA	D0	D1	D2	D3	D4	D5	D6	D7
18	DA	00	FA	02	01	0C	00	00	00	00	00

Tabela 3.10 - Mensagem de resposta da EMS a solicitação OBD-II genérico.

CAN ID				CAN Data							
Priority	PDU Format	PDU Specific	SA	D0	D1	D2	D3	D4	D5	D6	D7
18	DA	FA	00	04	41	0C	09	64	00	00	00

## 3.5. Métodos de controlo do motor

---

Nesta secção apresenta-se os ensaios efetuados com o objetivo de controlar a rotação do motor. Nestes ensaios utilizou-se como meio de comunicação o barramento vermelho, pois verifica-se que é o meio de comunicação entre o COO e a EMS. Utilizou-se então, numa fase inicial, apenas o interface CAN conectado ao barramento vermelho. A Figura 3.6 ilustra a ligação do interface ao barramento vermelho, mantendo todo os restantes componentes ligados de fábrica.

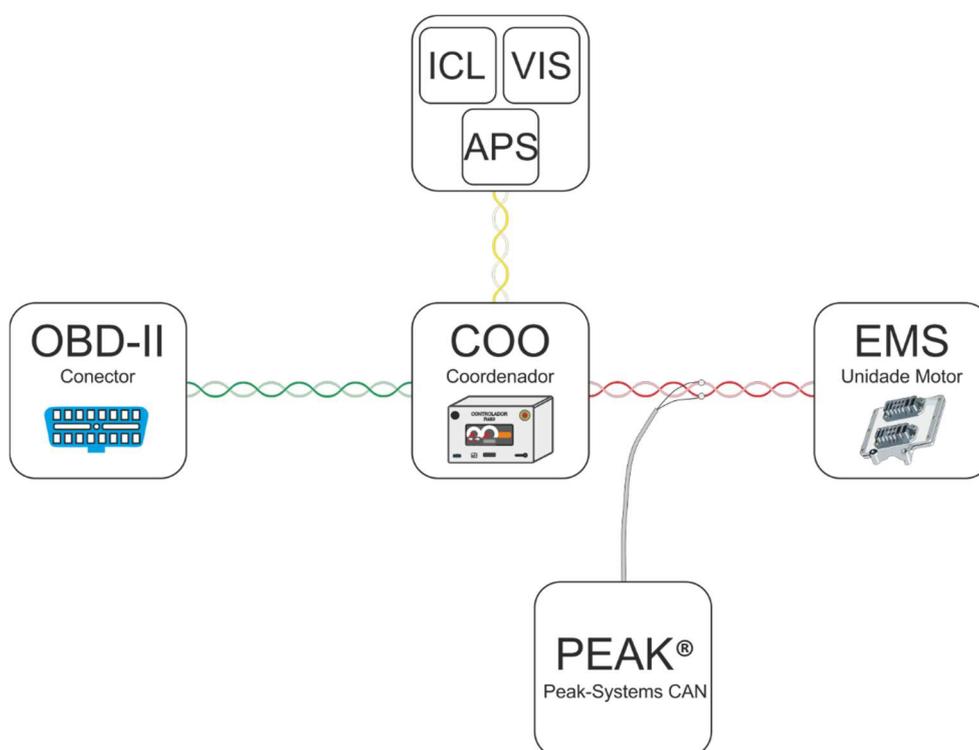


Figura 3.6 - Esquema de ligação do interface CAN ao barramento vermelho.

### 3.5.1. Mensagem de controlo de Torque/ Speed

---

Segundo a norma SAE J1939, definiu-se uma mensagem com o objetivo de controlar o binário disponibilizado pelo motor e a o valor da rotação. A Tabela 3.11 apresenta o PGN associado a esta mensagem e o campo dos dados relacionados com a informação relativa ao controlo do motor. Após a descrição dos parâmetros presentes nesta mensagem procede-se a sua descodificando tendo como base a própria norma. A Tabela 10.1 apresentada nos anexos, de forma resumida, a descodificação dos 7 parâmetros presentes

nesta mensagem possibilitando assim a construção de uma mensagem que será enviada posteriormente para o barramento. Através da consulta da Tabela 10.1, é possível compilar uma mensagem de controlo do binário e rotação do motor. Compilou-se então uma mensagem que se encontra descrita na Tabela 3.12, contendo o CAN-ID e o campo dos dados desta mensagem.

O CAN-ID desta mensagem define-se com os seguintes critérios: prioridade de 3, PGN “0x0000” e o SA é o COO, cujo valor associado é 0x27.

A mensagem define o método de controlo como rotação do motor com a linha de transmissão desacoplada, a prioridade é elevada, a rotação definida é de 700 RPM, a percentagem de torque de 0 %, a taxa de transmissão de 20 ms e com o intuito de controlar a rotação do motor temporariamente. Esta definição encontra-se no campo dos dados da mensagem CAN (Tabela 3.12).

**Tabela 3.11 - Descrição do PGN associado ao controlo de binário e rotação do motor[11].**

<b>PGN 0</b>		<b>Torque/Speed Control 1</b>		<b>TSC1</b>
<i>Transmission Rate:</i>	<i>Repetition</i>	<i>Control purpose dependent</i>		
<i>Data Length:</i>		8		
<i>Extended Data Page:</i>		0		
<i>Data Page:</i>		0		
<i>PDU Format:</i>		0 (0x00)		
<i>PDU Specific:</i>		DA		
<i>Default Priority:</i>		3		
<i>Parameter Group Number:</i>		0 (0x0000)		
<i>Start Position</i>	<i>Length</i>	<i>Parameter NAME</i>		<i>SPN</i>
1.1	2 bits	Engine Override Control Mode		695
1.3	2 bits	Engine Requested Speed Control Conditions		696
1.5	2 bits	Engine Requested Speed Control Conditions		897
2-3	2 bytes	Engine Requested Speed/Speed Limit		898
4	1 byte	Engine Requested Torque/Torque Limit		518
5.1	3 bits	TSC1 Transmission Rate		3349
5.4	5 bits	TSC1 Control Purpose		3350

**Tabela 3.12 - Mensagem TSC1 enviada para o barramento CAN vermelho.**

<b>CAN ID</b>				<b>CAN Data</b>							
<b>Priority</b>	<b>PDU Format</b>	<b>PDU Specific</b>	<b>SA</b>	<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>	<b>D4</b>	<b>D5</b>	<b>D6</b>	<b>D7</b>
0C	00	00	27	D1	BC	02	7D	FE	FF	FF	FF

Enviou-se a mensagem TSC1 para o barramento vermelho com o intuito de simular um comando gerado pelo COO. Após a compilação da mensagem e ligação do interface CAN ao barramento pretendido, iniciou-se o envio da mensagem (definida na Tabela 3.12). Durante o envio da mensagem, não se verificou qualquer alteração na velocidade de rotação do motor.

Posteriormente, após a descodificação da mensagem com o PGN 0xFEE3, verificou-se que o *hardware* em estudo não suporta este tipo de controlo. Concluindo assim que este método de controlo não é possível de implementar no *hardware* disponibilizado.

### 3.5.2. Mensagem com o valor da posição do acelerador

---

Segundo o esquema representado na Figura 2.6, o valor, em tensão (V), correspondente a posição do acelerador é obtido pelo COO. Este valor é um *input* desta unidade, ou seja, esta informação é adquirida e processada pelo COO. Após este processamento, esta informação, é enviada para a EMS através de uma mensagem CAN presente no barramento vermelho. Através da descodificação das mensagens *Broadcast* presentes no barramento vermelho foi possível compreender a mensagem que transmite esta informação.

A informação do pedal do acelerador surge em duas mensagens do barramento, com os respetivos PGN, 0xF003 e 0xFF80. Através da análise das duas mensagens identifica-se que a mensagem com o PGN 0xF003 é proveniente do motor e a mensagem com o PGN 0xFF80 é proveniente do COO. Assim considera-se que a mensagem proveniente do motor tem uma função de fornecer um feedback relativamente a posição do pedal do acelerador que este se encontra a operar. Por exclusão de partes, a mensagem com o PGN 0xFF80 contém a informação da posição do acelerador, proveniente da leitura do COO.

Seguidamente, procedeu-se à aquisição das mensagens presentes no barramento vermelho. Esta aquisição efetuou-se com o em modo de funcionamento em marcha lenta. Através da análise das mensagens recolhidas, verificou-se que o campo dos dados da mensagem CAN com o PGN 0xFF80 é constante ao longo do tempo. Apresenta-se na Tabela 3.13, o conteúdo da mensagem, o CAN-ID e o campo de dados. Com o auxílio da descodificação, presente em Anexo (Tabela 10.3), é possível reconhecer qual o byte que transporta a informação sobre o pedal do acelerador. Após a consulta ao Anexo, obtém-se

o valor da posição do acelerador, em hexadecimal, no byte D1. Através da equação (1), descodifica-se o valor, resultando assim no valor de percentagem do pedal de acelerador de 0%.

**Tabela 3.13 - Mensagem com o PGN 0xFF80 presente no barramento vermelho.**

CAN ID				CAN Data							
Priority	PDU Format	PDU Specific	SA	D0	D1	D2	D3	D4	D5	D6	D7
0C	FF	80	27	0C	00	0D	FF	FF	FC	3C	00

Posteriormente, à descodificação da mensagem, procedeu-se ao envio da mensagem de teste, através do interface CAN para o barramento vermelho. Apresenta-se a equação (2), que permite converter o valor parâmetro em valor decimal de modo a introduzir no campo dos dados das mensagens CAN. Esta equação é composta por  $V_{\text{Parâmetro}}$ ,  $R_N$  e  $offset$ . O  $V_{\text{Decimal}}$  adquire-se pela expressão descrita na formula.

$$(V_{\text{Parâmetro}} - \text{offset}) / R_N = V_{\text{Decimal}} \quad (2)$$

Assim é possível converter o valor de percentagem do acelerador para o valor correspondente na mensagem CAN. Através da consulta ao Anexo (Tabela 10.3) da descodificação das mensagens, define-se o valor de  $offset$  e resolução ( $R_N$ ) com os respetivos valores, 0 e 0,4 %/bit. Definiu-se então um valor de 9,6% para a posição do acelerador de teste e através da equação (2) converte-se para um valor em decimal de 24. Após esta etapa é necessário converter este valor em decimal para hexadecimal, resultando assim em 0x18. Seguidamente compilou-se a mensagem que se encontra descrita na Tabela 3.14. Esta tabela contém o CAN-ID e o campo de dados que são semelhantes a mensagem presente no barramento vermelho, apenas com a diferença no segundo byte (D1).

**Tabela 3.14 – Mensagem de teste com o PGN 0xFF80.**

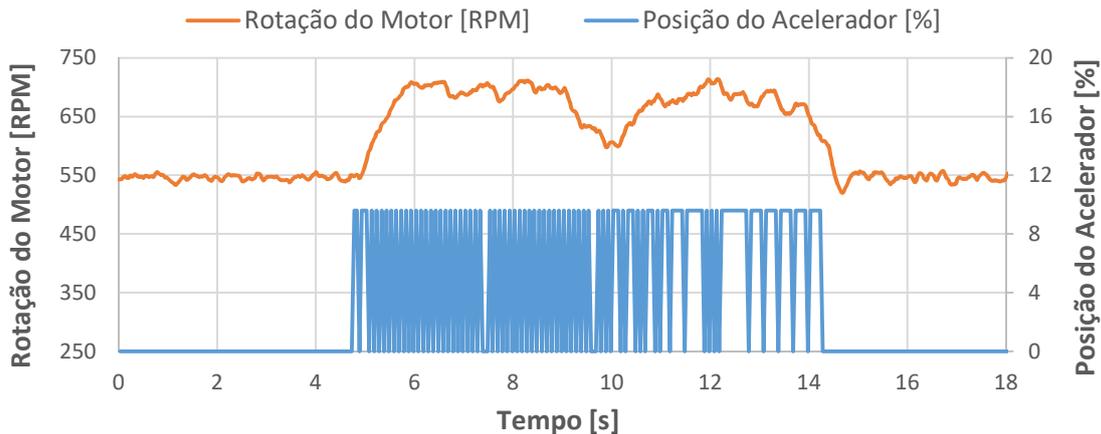
CAN ID				CAN Data							
Priority	PDU Format	PDU Specific	SA	D0	D1	D2	D3	D4	D5	D6	D7
0C	FF	80	27	0C	18	0D	FF	FF	FC	3C	00

Após a compilação da mensagem de teste, procedeu-se ao seu envio para o barramento vermelho. Quanto à frequência de envio foram testadas quatro, respetivamente, 25 Hz, 50 Hz, 100 Hz e 200 Hz.

Durante o envio da mensagem de teste foram recolhidos dois parâmetros de forma a entender o funcionamento da unidade do motor. Os parâmetros adquiridos foram a Rotação do Motor [RPM] e a Posição do Acelerador [%]. O parâmetro Rotação do Motor obteve-se através da descodificação da mensagem CAN com o PGN "0xF004", da posição 4 e 5 do campo de dados. Esta descodificação encontra-se disponível em Anexo (Tabela 10.3). Obteve-se o valor da percentagem do acelerador através da mensagem CAN com o PGN "0xF003" na segunda posição do campo de dados. Descodificou-se o valor da posição de acelerador com recurso a folha em Anexo (Tabela 10.3). Este valor da posição é obtido através de uma mensagem proveniente da unidade EMS, ou seja, possui uma taxa de atualização constante de 20 Hz e não se encontra sincronizada com a nossa frequência de envio.

Apresenta-se a Figura 3.7 que contém o gráfico que demonstra o valor dos parâmetros adquiridos durante o envio da mensagem de teste. Durante este teste utilizou-se uma frequência de envio de 100 Hz, que corresponde ao dobro da taxa de atualização da mensagem proveniente do COO. Tal como é possível observar no gráfico, o comportamento do motor é imprevisível. Assim não é possível controlar o motor através deste método.

Com a variação da frequência de envio da mensagem de teste, não se reconhece nenhum padrão no comportamento do motor. Conclui-se então que não é possível controlar a rotação do motor através deste método.



**Figura 3.7 - Gráfico valores dos parâmetros adquiridos durante o envio da mensagem a 100 Hz.**

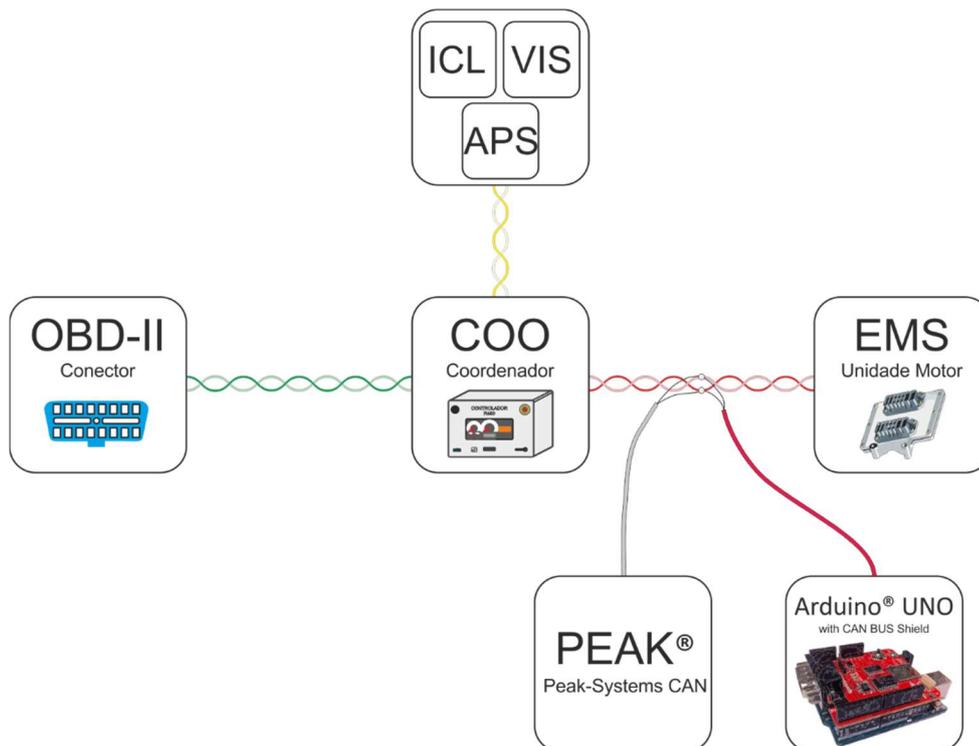
Posteriormente a esta conclusão procedeu-se a outro método de envio de mensagens para o barramento. Este segundo método consiste no envio da mensagem CAN imediatamente após a mensagem enviada pelo COO. Este método tem o intuito de perceber como é executado o controlo do motor por parte da EMS.

Para realizar este teste, recorreu-se a um Arduino® UNO equipado com um CAN BUS Shield. Através deste hardware é possível realizar um conjunto de ensaios específicos que o interface CAN não permite. Proporciona assim uma plataforma de teste mais alargada com mais potencial. A Figura 3.8 representa a ligação do Arduino® UNO, ao barramento vermelho em simultâneo com o interface CAN. O Arduino® UNO tem o objetivo de replicar a mensagem com o PGN “FF80” sempre que seja enviada pelo COO. A mensagem replicada pelo Arduino® altera apenas byte D1 do campo de dados, tal como é representado pela Tabela 3.14. Quanto ao interface CAN, este equipamento tem como objetivo adquirir todas as mensagens presentes no barramento vermelho. Através desta aquisição é possível elaborar o gráfico, presente na Figura 3.9. Este gráfico apresenta o valor dos parâmetros de rotação do motor e posição do acelerador durante a realização deste ensaio.

Através da análise dos resultados obtidos, expostos na Figura 3.9, constata-se que o valor da rotação do motor não se encontra constante ao longo do ensaio, tendo algumas quebras ao longo do ensaio. Quanto ao valor da posição de acelerador lido pela EMS também não foi constante durante o ensaio. Através da análise destas diferenças e como estas não ocorrem em simultâneo nada se pode concluir em relação a sua dependência. A única conclusão possível deste ensaio é a impossibilidade de controlar a rotação do motor

através deste método. Pois como se verifica no gráfico por vezes o valor da rotação oscila, ou seja, não é possível controlar o valor da rotação do motor ao longo do tempo de forma constante.

A única solução possível de se utilizar este método de controlo do motor é recorrer a uma *hardware* que possua duas linhas CAN, CAN A e CAN B. Com este hardware é possível dividir o barramento CAN vermelho em duas secções, onde uma delas conecta-se ao CAN A e a outra ao CAN B. Assim, com esta divisão, é possível replicar as mensagens presentes no barramento conectado ao CAN A para o barramento conectado ao CAN B e vice-versa. Todas as mensagens seriam replicadas de forma integral exceto a mensagem que contém o valor da posição do acelerador. Assim é possível controlar absolutamente o valor da posição do acelerador que se encontra presente no barramento ligado a EMS. Este tipo de solução implica a segmentação do barramento vermelho e a introdução de um elemento que pode originar uma falha no sistema.



**Figura 3.8 - Esquemático da ligação do Arduino® UNO e o interface CAN ao barramento vermelho.**

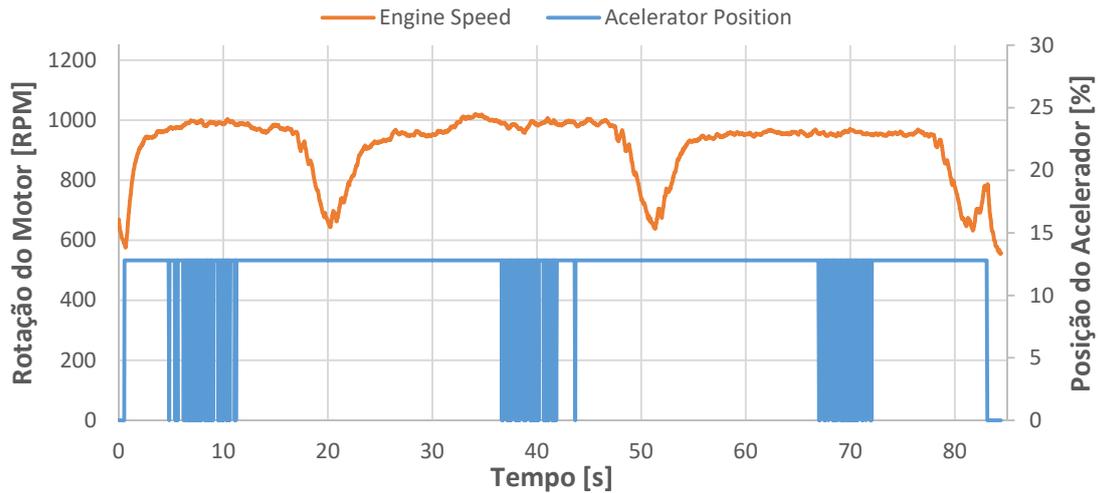


Figura 3.9 - Gráfico valores dos parâmetros adquiridos durante a replicação da mensagem pelo Arduino®.

### 3.5.3. Mensagem Cruise Control

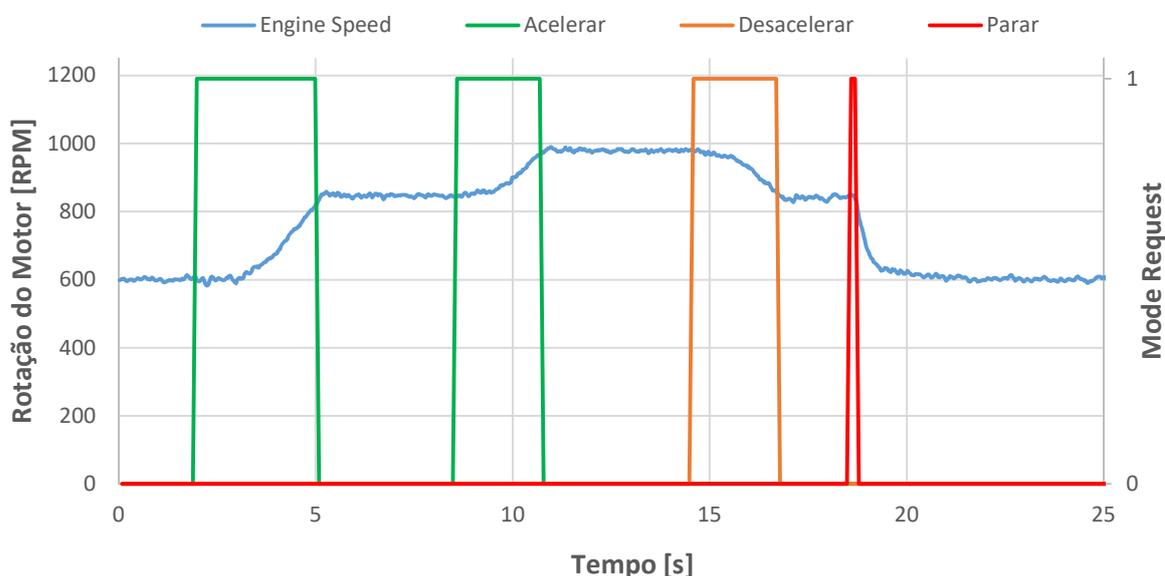
Através da análise da Figura 2.6, verifica-se que os botões de controlo da função de *Cruise Control*, encontram-se ligados ao COO. Então, tal como acontece no caso anterior, o COO envia uma mensagem CAN para o barramento vermelho com a informação obtida através deste botões. Após a análise e descodificação das mensagens presentes no barramento vermelho, identifica-se o CAN-ID “0x18FEF127” como o portador da informação relativa ao *cruise control*. Na Tabela 3.15 é apresentado o CAN-ID e o campo dos dados das mensagens CAN das funções de acelerar, desacelerar e parar. Após a recolha desta informação procedeu-se para a fase seguinte, que corresponde ao envio destas mensagens para o barramento vermelho e verificar o resultado desta ação. Procedeu-se ao este envio através do interface CAN e adquiriu-se as mensagens presentes no barramento através do mesmo equipamento.

A Figura 3.10 representa o gráfico com o valor da rotação de motor adquirido durante o envio das 3 mensagens, acelerar, desacelerar e parar. Tal como é possível de se verificar, este tipo de controlo funciona corretamente não interferindo com as restantes mensagens

presentes no barramento. Este método possui apenas a pequena desvantagem de ser um pouco lento a atuar.

**Tabela 3.15 - Mensagens de controlo do Cruise Control.**

Mode	CAN ID				CAN Data							
	Priority	PDU Format	PDU Specific	SA	D0	D1	D2	D3	D4	D5	D6	D7
Acelerar	18	FE	F1	27	F7	FF	FF	07	43	FF	FF	FF
Desacelerar	18	FE	F1	27	F7	FF	FF	07	07	FF	FF	FF
Parar	18	FE	F1	27	F7	FF	FF	03	03	FF	FF	FF



**Figura 3.10 - Gráfico valores dos parâmetros adquiridos durante o envio das mensagens de Cruise Control.**

### 3.5.4. Mensagem diagnóstico definição do ralenti

Nesta secção apresenta-se outro método de controlo do valor da rotação do motor. Para realizar este teste recorreu-se ao interface diagnóstico Scania® VCI2, onde se conectou a ficha OBD-II. Após esta ligação procedeu-se a tentativa de programar com um novo valor de ralenti. Utilizou se o *software* SDP3 para programar a EMS, com o motor em funcionamento a operar em modo ralenti. Após a definição do novo valor de ralenti, quando é executada a sua programação surge um aviso referente ao impedimento da programação

enquanto o valor da rotação for diferente de 0. Ou seja, não é possível programar um novo valor de ralenti, através do *software* SDP3, enquanto este se encontra em funcionamento. Invalida-se assim este método de controlo do motor.

### 3.5.5. Potenciómetro externo para posição do acelerador

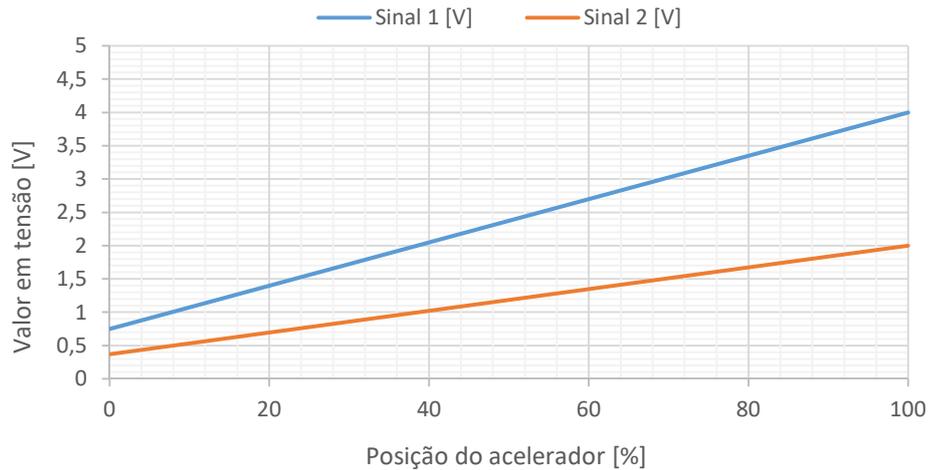
O potenciómetro original que o motor utiliza é do tipo *hall*, ou seja, não possui contacto mecânico e possui duas saídas, o Sinal 1 e o Sinal 2. A Figura 3.11 representa o diagrama de ligação dos pinos do potenciómetro. Quanto aos sinais de saída, apresenta-se a Tabela 3.16 que contém os valores em tensão das duas saídas, correspondentes à posição do pedal, em percentagem. A Figura 3.12 representa o gráfico da curva de transferência deste potenciómetro, onde se verifica que ambos os sinais de saída são lineares.



Figura 3.11 - Diagrama de ligação do potenciómetro da posição do acelerador[6].

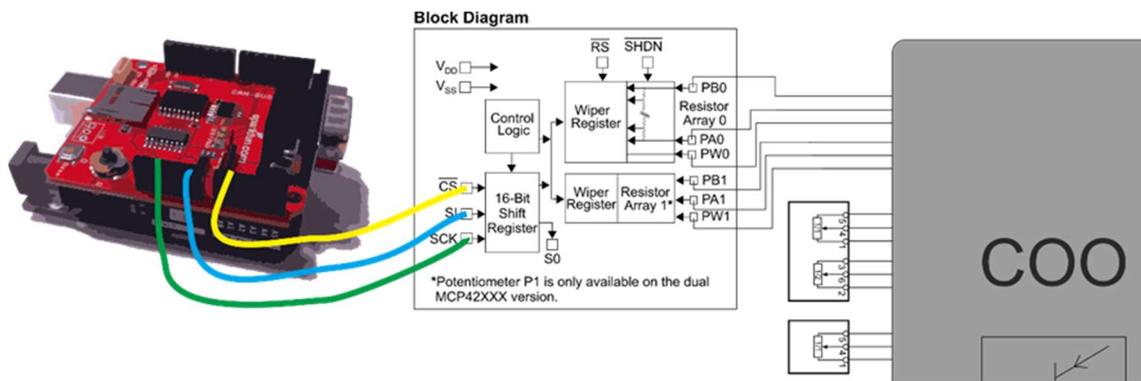
Tabela 3.16 - Valores, em tensão, dos sinais de saída do potenciómetro.

Posição [%]	Sinal 1 [V]	Sinal 2 [V]
0	0,75	0,37
100	4	2



**Figura 3.12 - Gráfico da curva de transferência do potenciômetro original.**

O esquema de ligação do pedal de acelerador ao COO encontra-se disponível na Figura 2.6. Após a análise deste esquema, verifica-se quais as ligações necessárias para efetuar a substituição por um potenciômetro externo que é controlado por SPI. A Figura 3.13 apresenta uma possível configuração de componentes necessários para proceder a definição do valor do potenciômetro.



**Figura 3.13 - Esquema de ligação do potenciômetro externo ao COO.**



## 4. Desenvolvimento do *Hardware*

---

Neste capítulo apresenta-se as soluções, encontradas durante a descodificação do protocolo, com o intuito de concretizar os objetivos do projeto. Todas as soluções apresentadas foram devidamente fundamentadas no capítulo anterior. Posteriormente à apresentação da solução, será apresentada as características do interface desenvolvido ao longo do projeto, onde se inclui a análise detalhada de todos os componentes constituintes e as suas funcionalidades. Apresenta-se também o projeto e desenvolvimento da placa de circuito impresso, os cabos de ligação, o protocolo de comunicação entre o *hardware* e o computador. Por fim apresenta-se o desenvolvimento do código do microprocessador.

### 4.1. Introdução, características e funcionalidades

---

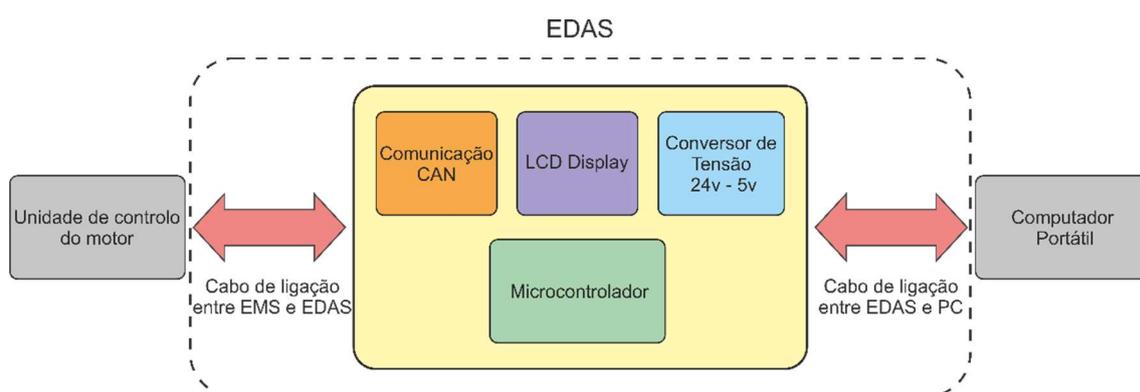
Ao longo desta secção apresenta-se as características e funcionalidade do *hardware*. De modo a facilitar a posição do *hardware* relativamente aos componentes existentes, apresenta-se na Figura 4.1 o esquema dos componentes envolvidos no projeto e as suas ligações. Na ilustração identifica-se um motor e a sua respetiva unidade de comando, um computador portátil e o *hardware* desenvolvido ao longo do projeto. A este *hardware* atribuiu-se o nome de Engine Data Acquisition System (EDAS) de modo a facilitar futuras referências.

Através da análise da ilustração, verifica-se que o EDAS necessita de algumas características, tais como, um meio físico de comunicação com a unidade do motor; um meio físico de comunicação com o computador portátil; capacidade de receber, processar e enviar informação (microcontrolador); possuir dois métodos possíveis de alimentação de tensão e por fim isolar todos os níveis de tensão da unidade do motor dos níveis de alimentação do computador portátil.



**Figura 4.1 - Esquema de ligação do interface EDAS.**

Com o intuito de facilitar a identificação das características necessárias, expõem-se a Figura 4.2. Esta figura contém o diagrama de blocos dos componentes presentes nos EDAS. Tal como se verifica, o sistema necessita de possuir um microcontrolador, um LCD Display, um conversor de tensão de 24V para 5V, comunicação CAN (2 linhas de comunicação), um cabo de ligação entre a unidade do motor e o sistema e também um cabo de ligação entre o nosso sistema e o computador portátil. Na próxima secção será apresentada a seleção de todos os componentes e as suas características.



**Figura 4.2 - Diagrama de blocos dos componentes presentes no EDAS.**

### 4.1.1. Aquisição de dados

---

Nesta subsecção são apresentados os quatro métodos possíveis de adquirir informação do barramento. A Tabela 4.1 expõem um resumo das vantagens e desvantagens de cada um dos métodos. Após a análise e discussão dos vários métodos conclui-se que o interface tem como requisitos aceder a informação presente nos dois barramentos CAN. Quanto ao método que será utilizado para adquirir os parâmetros será escolhido de acordo com a quantidade de parâmetros requisitados. Apresenta-se ainda neste capítulo uma lista dos parâmetros possíveis de adquirir durante um ensaio. Em anexo apresenta-se a Tabela 10.3 que contém todas as variáveis disponíveis no barramento CAN do veículo pesado.

Tabela 4.1 – Soluções encontradas para aquisição de dados.

Descrição	Vantagens	Desvantagens	Acessibilidade
<b>Broadcast</b> do barramento verde	- Comunicação espontânea	- Pode não ser universal	Elevada
<b>Broadcast</b> do barramento vermelho	- Comunicação espontânea - Mais parâmetros disponíveis	- Dificuldade de acesso	Baixa
<b>Diagnóstico</b> SDP3 (Peer-to-Peer)	- Accede a parâmetros calculados pela EMS	- Necessita de uma mensagem de <i>request</i> - Dedicado Scania®	Elevada
<b>OBD-II Genérico</b> (Peer-to-Peer)	- Compatível com vários veículos	- Necessita de uma mensagem de <i>request</i>	Elevada

Após a recolha de todos os parâmetros disponíveis no barramento CAN, procedeu-se a escolha dos parâmetros necessários ao projeto. A Tabela 4.2 apresenta todos os parâmetros selecionados e algumas informações relativas aos mesmos, tais como, o nome do parâmetro, a unidade de medida, a prioridade e o qual o método de aquisição. Em relação a prioridade dividiu-se em dois grupos, alta e baixa, de modo a facilitar o desenvolvimento do protocolo de comunicação. Dos dezanove parâmetros, durante o ensaio adquire-se continuamente apenas dezasseis, oito de alta prioridade e oito de baixa prioridade, os restantes três são apenas adquiridos no início de cada ensaio. Quanto ao método de aquisição utilizou-se apenas dois dos quatro métodos apresentados, o **Broadcast RED Bus** e o **Diagnostico SDP3**.

**Tabela 4.2 – Parâmetros a adquirir durante um ensaio.**

Nome do parâmetro	Unidades	Prioridade	Método de aquisição
Temperatura de Admissão	°C	Baixa	Broadcast RED Bus
Temperatura Ambiente	°C	Baixa	Broadcast RED Bus
Temperatura de combustível	°C	Baixa	Broadcast RED Bus
Temperatura do líquido refrigeração	°C	Baixa	Broadcast RED Bus
Pressão de admissão (MAP)	bar	Alta	Diagnóstico SDP3
Pressão Ambiente (barométrica)	bar	Baixa	Broadcast RED Bus
Pressão de alimentação de combustível	bar	Alta	Diagnóstico SDP3
Pressão de óleo	bar	Alta	Diagnóstico SDP3
Carga do pedal de acelerador	%	Alta	Broadcast RED Bus
Carga do motor ( <i>Engine Load</i> )	%	Alta	Broadcast RED Bus
Rotação do motor	rpm	Alta	Broadcast RED Bus
Consumo de combustível atual	l/h	Alta	Broadcast RED Bus
Massa de ar admitida (MAF)	kg/min	Alta	Diagnóstico SDP3
Tensão da bateria	V	Baixa	Diagnóstico SDP3
Nível do Tanque	%	Baixa	Broadcast RED Bus
Velocidade rotação ventoinha	rpm	Baixa	Diagnóstico SDP3
Binário de referência do motor	Nm	Baixa*	Broadcast RED Bus
Horas de funcionamento do motor	h	Baixa*	Broadcast RED Bus
Número de chassis	VIN	Baixa*	Broadcast RED Bus

(\*) – Adquire apenas uma vez no início de cada ensaio.

## 4.1.2. Controlo do motor

---

Relativamente ao método de controlo de rotação do motor, após a realização de vários ensaios, compilou-se a informação exposta na Tabela 4.3. Esta tabela expõe qual o desempenho de que cada método e as suas vantagens e desvantagens. Através da análise desta informação, verifica-se que apenas dois métodos são possíveis usar com fiabilidade. Estes dois métodos são, respetivamente, mensagem CAN de *Cruise Control* e o potenciómetro externo.

Para implementar este dois tipos de controlo, o interface necessita de comunicar com barramento vermelho e com um potenciómetro digital conectado a ficha do pedal do acelerador original.

Tabela 4.3 - Soluções encontradas para o controlo da rotação do motor.

	Descrição	Desempenho	Vantagens	Desvantagens	Acessibilidade
Mensagem CAN	<b>Torque/Speed Control</b>	Não é suportado pelo motor	Desconhecida	Desconhecida	Média
	<b>Valor da posição do acelerador</b>	Não controla o motor com estabilidade	Rapidez de atuação	Aumenta a probabilidade de falha	Muito Baixa
	<b>Cruise Control</b>	Controla o motor com estabilidade	Estabilidade da Rotação do Motor	Atua lentamente	Média
	<b>Definir ralenti</b>	Não é suportado pelo <i>software</i>	Desconhecida	Desconhecida	Elevada
	<b>Potenciómetro</b>	Controla o motor com estabilidade e rapidez	Compatível com qualquer motor com EMS	Necessidade de mais ligações	Elevada

## 4.2. Seleção dos componentes

---

Após a definição das características do EDAS, procedeu-se a seleção dos componentes do mesmo. Na Tabela 4.4 apresenta-se a lista de componentes principais selecionados, a tabela contém o nome do componente, a referência do fabricante, a quantidade e a sua função. Não se incorporou nesta lista os componentes genéricos, tais como, resistências, condensadores, entre outros.

**Tabela 4.4 – Lista de componentes do EDAS.**

<b>Qtd</b>	<b>Nome do componente</b>	<b>Referência</b>	<b>Função</b>
1	Microcontrolador	ChipKIT Max32®	Processar informação
2	<i>Transceiver</i> CAN	Microchip® MCP2562 E/SN	Converter tensões do barramento
1	Potenciômetro digital (SPI)	Microchip® MCP4251 E/SN	Controlar o valor da posição do acelerador
1	Regulador de tensão	Traco® TSR0.5-2450	Converter 24V para 5V
1	LCD <i>Display</i>	LCD0805	Indicar informações ao utilizador
1	Conector	Hirose® 3560-16S	Ligação da EMS ao EDAS
1	Isolador ótico genérico	Fairchild® MOCD207R2M	Isolar os níveis de tensão
1	Isolador ótico alta velocidade	Fairchild® HCPL0638	Isolar os níveis de tensão
1	Cabo OBDII <i>Type B</i>	J1962M <i>Type B</i>	Ligação da EMS ao EDAS
1	Cabo USB	USB A to mini B	Ligação do EDAS ao PC

Seguidamente, expõem-se as fundamentações das escolhas destes componentes, ou seja, as suas principais características. Relativamente a cada componente, apresenta-se também as suas principais funções no EDAS.

### **Microcontrolador**

Selecionou-se o ChipKIT Max32® como o microcontrolador do EDAS devido as suas características. Este microcontrolador possui todos os requisitos do *hardware* projetado, tais como, dois barramentos CAN, comunicação SPI, conversor porta série para USB e 85 pinos de entrada e saída. Quanto a velocidade do processamento pode tomar o valor máximo de 80 MHz. A tensão a que o microcontrolador ópera é de 3,3V. As principais vantagens de utilizar um microcontrolador com todos os subsistemas inerentes, integrados numa só placa, são: maior fiabilidade; facilidade de troca em caso de avaria ou mau funcionamento; facilidade aquisição. Relativamente ao número de pinos disponíveis, através de número elevado de pinos permite uma maior possibilidade de expansão mantendo o mesmo *hardware* de base.

### ***Transceiver* CAN**

O microcontrolador selecionado permite a leitura/escrita em dois barramentos CAN em simultâneo, mas para que isso seja possível necessita de um *transceiver* CAN para cada barramento. A função de um *transceiver* é converter os níveis de tensão do barramento para os níveis tensão do microcontrolador que neste caso é de 3,3V. O *transceiver*

selecionado é um Microchip® MCP2562 pois este permite operar a 3,3V em conformidade com o microcontrolador. Este componente beneficia de mais duas características, permite trabalhar a várias velocidades do barramento CAN (até a um máximo de 1Mbit/s) e respeita os requisitos da indústria automóvel.

### **Potenciómetro digital (SPI)**

Após a realização dos ensaios iniciais concluiu-se que é necessário utilizar um potenciómetro externo para controlar o valor da posição do pedal de acelerador de modo a controlar o motor. Selecionou-se então um potenciómetro digital controlado por uma comunicação SPI com a referência MCP4251 do fabricante da Microchip®. Este componente possui dois canais de saída que corresponde as duas entradas da unidade do motor para o sensor da posição do pedal de acelerador. Possui um resistência de 10kΩ entre os extremos com 256 estados possuindo assim uma resolução, aproximadamente, 39 Ω/estado. A alimentação deste componente será proveniente da unidade do motor de modo a que o sinal de saída do potenciómetro tenha o mesmo referencial que a unidade do motor, prevenindo assim algum mau funcionamento do sistema. Este potenciómetro possui também um pino de *shutdown* que nos permite, em caso de falha, desligar o sistema. A comunicação SPI entre o microcontrolador e este componente é isolada através de dois isoladores óticos, um isolador genérico e um de alta velocidade.

### **Regulador de tensão**

Para que o EDAS seja autónomo a nível de alimentação é necessário recorrer a um regulador de 24 V – 5 V. O valor de tensão de 5V será novamente convertido para 3.3V por um regulador interno no ChipKIT® Max32. Selecionou-se o regulador do fabricante TRACO® com a referência TSR0.5-2450. Este regulador possui uma eficiência energética de 86% com um baixo consumo de corrente quando se encontra em repouso. Estas características devem se a tecnologia utilizada, Step-Down converter, permitindo assim que a temperatura do componente seja inferior durante o seu funcionamento. Este componente é bastante compacto e possui uma proteção interna de curto-circuito.

### **LCD Display**

Durante o desenvolvimento do projeto recorreu-se a um LCD display com o intuito de facilitar a construção de algumas funções. Com o LCD é possível verificar qual o estado do dispositivo permitindo assim desenvolver as funções e as suas correlações com maior eficácia. O componente necessita de ter um tamanho reduzido de modo a ser possível a implementar na placa de circuito impresso. Tendo em conta alguns requisitos

selecionou-se um LCD com duas linhas com 8 caracteres cada uma. Assim permite uma maior facilidade de implementação do componente na PCB.

### **Conector**

O conector para este dispositivo necessita de ser compacto e com uma longa duração de ciclos de uso. Também necessita de ser distinto de modo a não ser facilmente replicado, adquirindo assim alguma segurança adicional. O conector selecionado é do fabricante Hirose® com a referência 3560-16S. Este conector contém 16 pinos, ou seja, possui quatro pinos livres de modo a possibilitar uma expansão do dispositivo. Quanto ao número de ciclos, este componente permite 20 000 ciclos de utilização. Este componente respeita as premissas apresentadas no início, sendo bastante compacto e não ser vulgarmente utilizado na indústria automóvel.

### **Isolador ótico**

Tal como foi referido no tópico sobre o potenciómetro digital é necessário isolar os níveis de tensão da comunicação e controlo entre o microcontrolador e o potenciómetro digital. Recorreu-se então a um isolador ótico genérico da Fairchild® com a referência MOCD207R2M. Este componente permite isolar duas linhas, o shutdown e o enable do potenciómetro digital. Em caso de falha o modo shutdown é ativado prevenindo assim uma gestão errada por parte da unidade do motor. Este isolador é compacto e de fácil implementação.

### **Isolador ótico de alta velocidade**

Este isolador ótico de alta velocidade tem como função separar os níveis de tensão das linhas de comunicação, designadamente, a linha de relógio (SCK) e de dados (SDO). O componente selecionado é do mesmo fabricante do isolador apresentado anteriormente, Fairchild®, com a referência HCPL0638. Este isolador tem a particularidade de permitir maior velocidade de comutação, que resulta numa maior velocidade de transmissão de dados. O valor máximo de transmissão de dados com este componente é de 10Mbit/s.

### **Cabo OBDII Type-B**

Este hardware necessita de uma conexão aos barramentos CAN presentes no veículo pesado, para isso é necessário um cabo OBDII. O conector dos veículos pesados tem uma particularidade, é um OBDII Type-B, ou seja, é utilizado em conectores OBDII de 24V. A referência do cabo selecionado é J1962M Type-B Open End, que corresponde ao cabo definido pela norma SAE. Num dos extremos, deste cabo, encontra-se conector OBDII Type-B e no outro extremo os 16 fios disponíveis para soldar/cravar num conector. Assim

é possível soldar o conector Hirose® no outro extremo criando assim a ligação física entre a unidade do motor e o EDAS.

### Cabo USB

Por fim é necessário um meio físico entre o EDAS e o computador portátil, utiliza-se então um cabo genérico USB de ligação. Este cabo possui numa extremidades um conector USB Type A e na outra extremidade um conector USB Type micro-B.

## 4.3. Projeto e Desenvolvimento da placa de circuito impresso (PCB)

---

Nesta secção apresenta-se todo o projeto e desenvolvimento relativo à placa de circuito impresso (PCB). Numa primeira etapa expõem-se o diagrama de blocos dos componentes presentes na PCB. Este diagrama apresenta-se na Figura 4.3, onde é possível verificar as principais ligações entre os vários componentes. Posteriormente será apresentado com maior detalhe os esquema de ligação dos componentes mais importantes para o projeto, enumerando, os *transceiver* CAN e o potenciómetro digital.

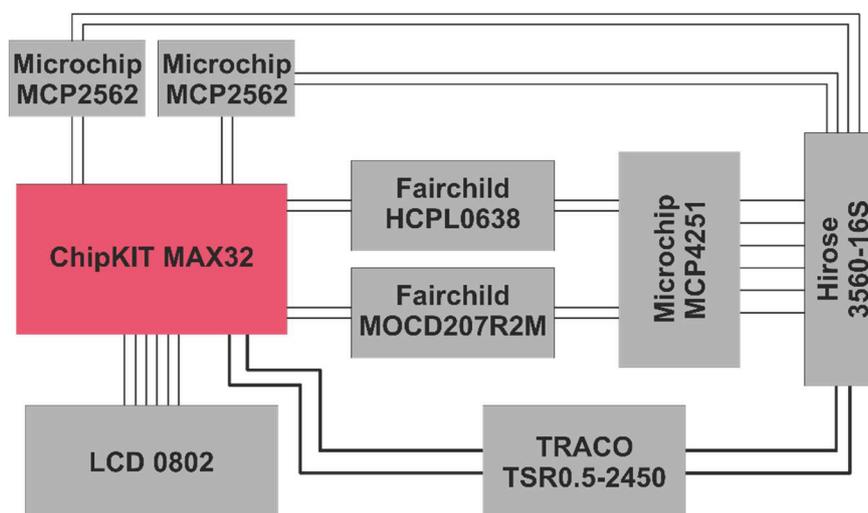


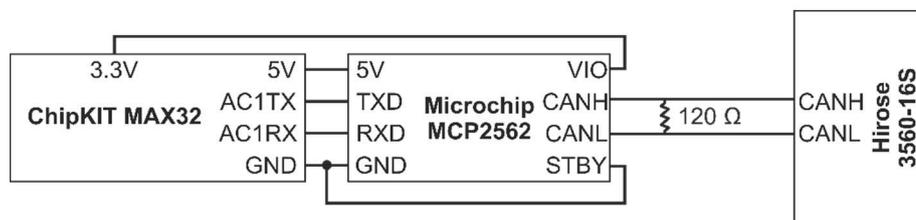
Figura 4.3 – Diagrama de blocos dos componentes da PCB.

No diagrama de blocos apresentado em cima não se encontra representado todas as ligações nem todos os componentes presentes na PCB, pois de modo a simplificar, apresentou-se os componentes mais relevantes ao projeto.

As ligações dos componentes respeitam as indicações dos fabricantes de modo a garantir o bom funcionamento dos mesmos. Apresenta-se agora dois esquemas mais detalhados da ligação dos *transceiver* CAN e do potenciamento digital. Estes esquemas encontram-se representados nas Figura 4.4 e Figura 4.5, respetivamente.

### **Transceiver CAN**

Na Figura 4.4 apresenta-se o esquema de ligação de um *transceiver* CAN ao microcontrolador. Como é possível observar, é bastante simples efetuar todas as ligações necessárias para o bom funcionamento deste componente. Este componente opera a uma tensão de 5V, no entanto o microcontrolador opera a 3.3V. Procedeu-se a ligação de 3.3V no pino VIO de modo a que as linhas de comunicação entre o ChipKIT® e o *transceiver* operem entre os 0V e os 3.3V. Relativamente ao barramento CAN, é necessário uma resistência de fim do barramento com o valor de 120  $\Omega$ .



**Figura 4.4 – Esquema de ligação do *transceiver* CAN.**

### **Potenciómetro digital (SPI)**

Posteriormente à apresentação do esquema de ligação do *transceiver*, segue-se o esquema de ligação do potenciómetro digital. Este esquema encontra-se representado na Figura 4.5, onde é possível visualizar todas as ligações necessárias durante o desenvolvimento da PCB.

A característica mais importante deste esquema de ligação é a utilização de isoladores óticos de modo a separar a alimentação dos sinais da EMS da alimentação dos sinais do EDAS. Através desta implementação previne-se curto-circuitos, leituras erradas por parte das unidades, entre outros aspetos. O potenciómetro digital é alimentado pelo COO, recebendo apenas a comunicação SPI por parte do microcontrolador.

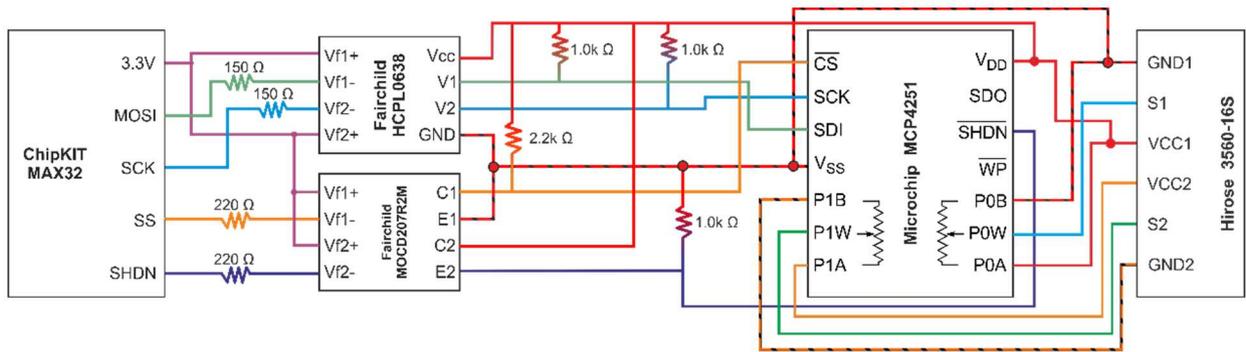


Figura 4.5 – Esquema de ligação do potenciómetro digital.

### Placa de circuito impresso (PCB)

Após a exposição dos esquemas de ligação mais relevantes do projeto, prossegue-se com a apresentação da placa de circuito impresso. Utilizou-se o software DesignSpark®, da RS Components®, para o desenvolvimento da PCB. A Figura 4.6 e Figura 4.7 representam, respetivamente, a camada superior e inferior da PCB. O intuito desta PCB é integrar no microcontrolador ChipKIT Max32®, utilizando apenas os pinos necessários ao projeto.

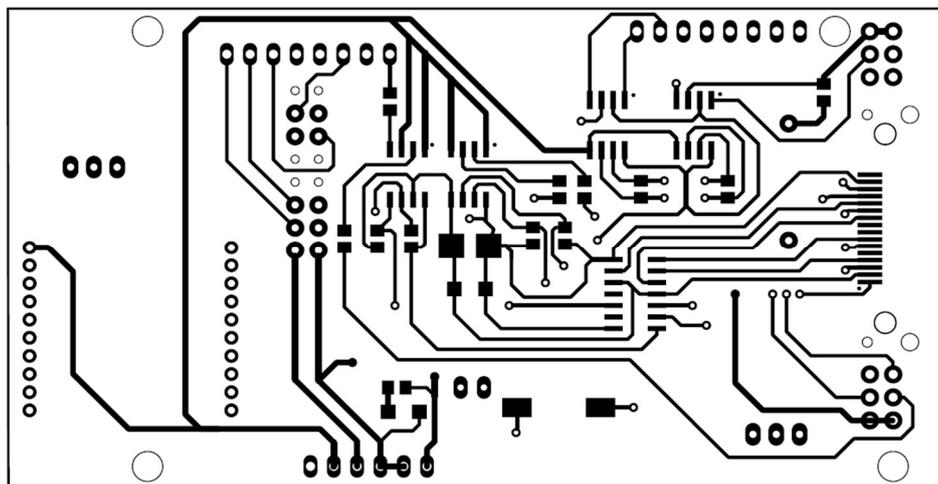
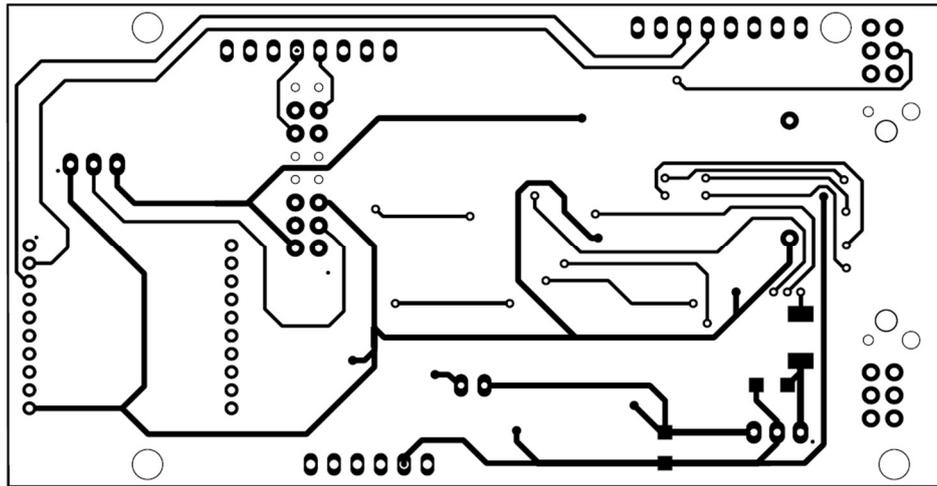
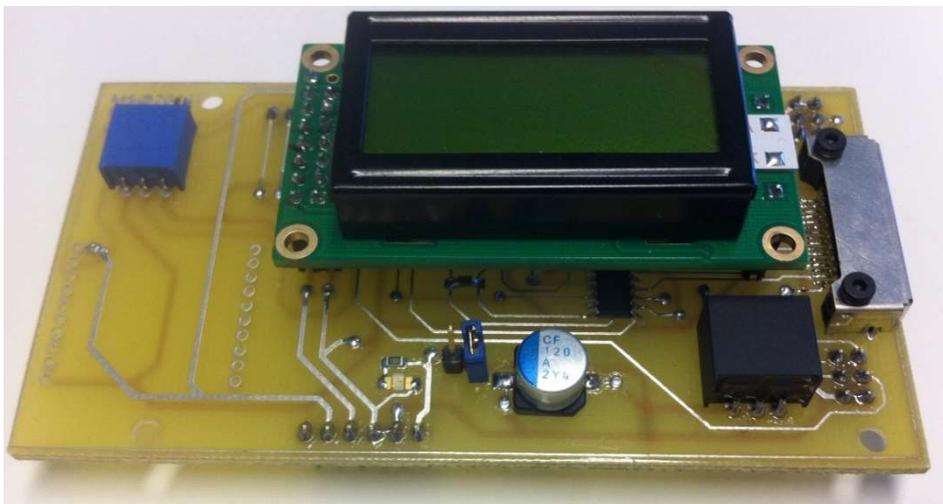


Figura 4.6 – Camada superior da PCB.



**Figura 4.7 – Camada inferior da PCB.**

Após a conclusão do processo de soldadura de todos os componentes obteve-se um protótipo funcional. A Figura 4.8 apresenta o protótipo já com todos os componentes soldados, finalizando assim a fase de projeto e desenvolvimento da placa de circuito impresso.



**Figura 4.8 – Protótipo funcional da PCB.**

## 4.4. Cabos de ligação

---

De seguida apresenta-se os dois cabos de ligação, o cabo de ligação entre o EDAS e a ficha OBD-II e ligação entre a ficha OBD-II e os seus componentes. A apresentação é composta por esquemas de ligação dos conectores e o cabo. Relativamente a ligação entre o EDAS e o PC utiliza-se um cabo USB genérico onde uma das extremidades é USB *Type A* (PC) e na outra extremidade é USB *Type Mini-A*.

### **Pinout da ficha OBD-II**

Durante o desenvolvimento do projeto sentiu-se a necessidade de alterar o *pinout* da ficha OBD II sendo necessário adicionar apenas mais alguns pontos de ligação. Manteve-se os pinos já existentes na posição original, adicionando-se apenas os pontos de ligação nos pinos que não interferem com o equipamento diagnóstico original do veículo.

Apresenta-se de seguida a Figura 4.9 que exhibe o *pinout* da ficha OBD-II modificada e as suas conexões internas. Identifica-se também o *pinout* do conector do sensor da posição do acelerador que nos permite controlar o seu valor através de um potenciómetro digital.

A Figura 4.10 apresenta o esquema de ligação da conexão entre o *hardware* desenvolvido e a ficha OBD II modificada.

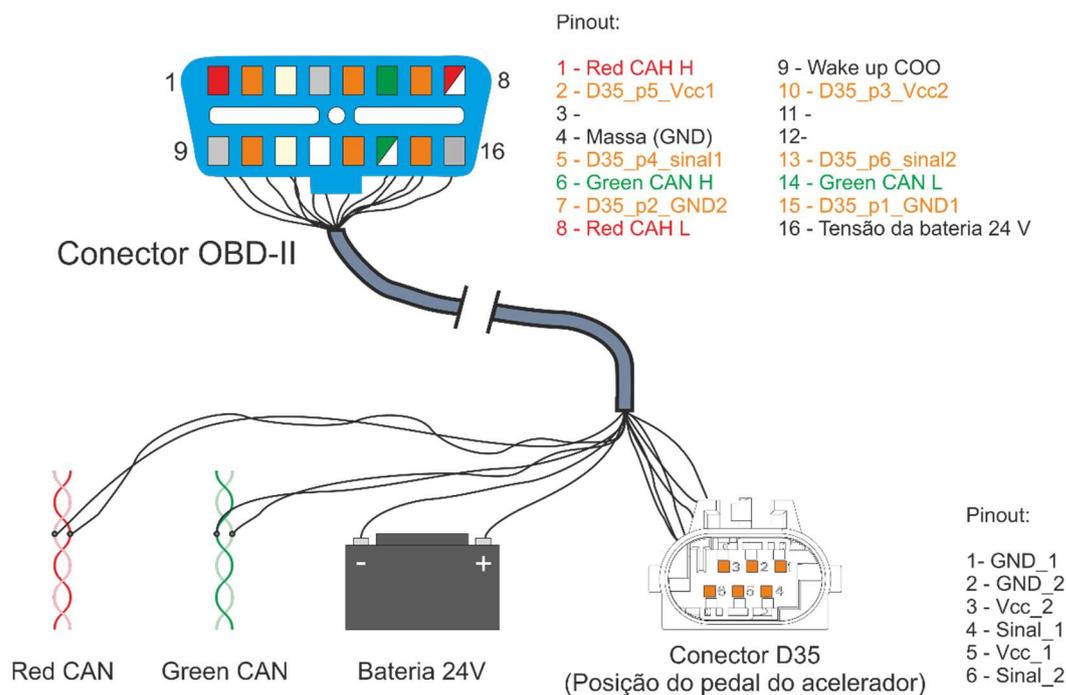


Figura 4.9 – Pinout da ficha OBD-II modificada.

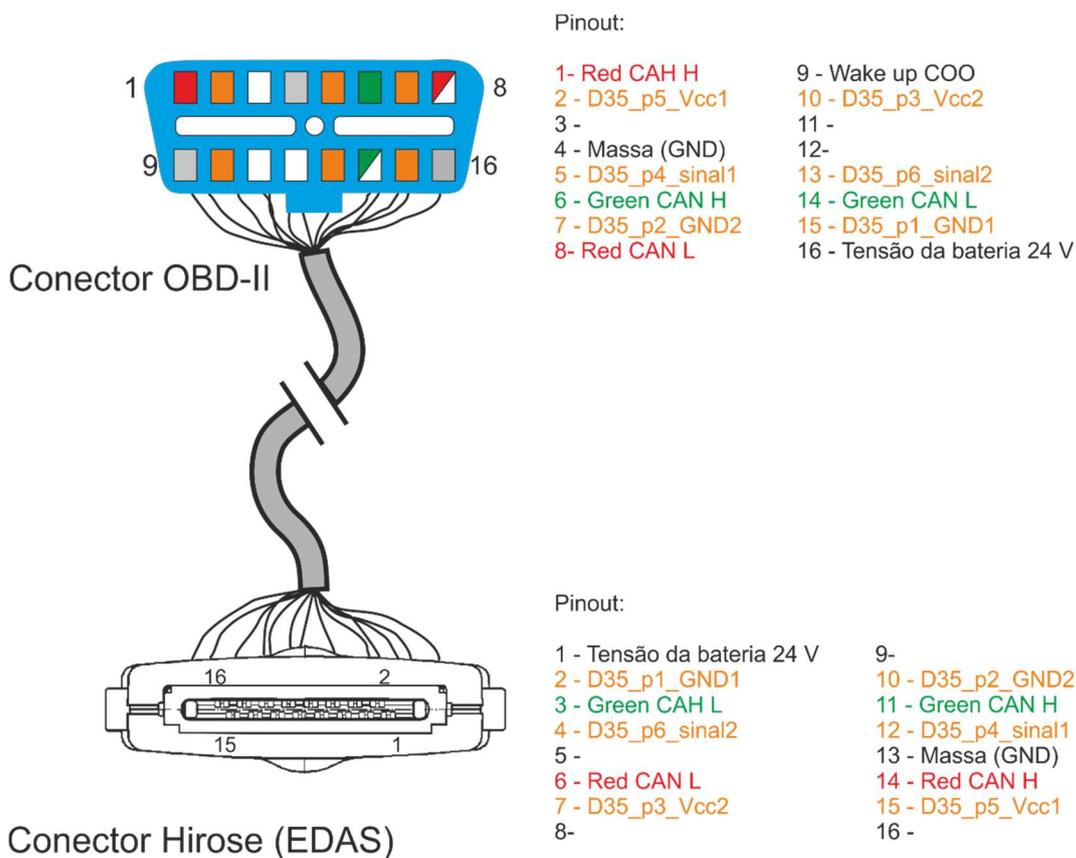


Figura 4.10 – Esquema do cabo de ligação entre o EDAS e a ficha OBD-II modificada.

## 4.5. Protocolo de comunicação (EDAS-PC)

---

Durante o desenvolvimento do projeto surgiu a necessidade de definir um protocolo de comunicação dedicado entre o EDAS e o PC, ou seja, um protocolo de comunicação exclusivo. Este protocolo divide-se em três fases, a fase de início de comunicação (**Handshake**), a fase de Configuração (**Config**) e a fase de Aquisição e controlo (**Request**).

### Protocolo de comunicação na fase Handshake

Esta primeira etapa tem como objetivo validar a comunicação entre o EDAS e o PC. O método utilizado encontra-se Tabela 4.5. Esta tabela demonstra os passos necessários à inicialização da comunicação. De modo a identificar o fim da transmissão de dados definiu-se um carácter de terminação, o *End of Transmission* (EOT) que corresponde a 0x04 em hexadecimal na tabela ASCII. Portanto, no final de cada *string* contém sempre o carácter de terminação.

Como é possível verificar, quem inicia a comunicação é o PC que envia um conjunto de caracteres, "START\04", formando constituído por 5 caracteres alfanuméricos e um carácter de terminação. Após o EDAS validar se o conteúdo recebido corresponde ao esperado, envia a *string* "CODE\04" com o intuito de solicitar um código de acesso. Posteriormente, o PC envia uma *string* composta por dois conjuntos de caracteres, o *Random Number* (RN) e o *Random Number Calculated* (RNC). O valor do RN é gerado aleatoriamente pelo PC e o seu valor pode variar entre 2048 e 9999. O valor do RNC é o dobro do RN, onde posteriormente é convertido em hexadecimal e de seguida efetua-se a troca dos dois bytes mais significativos pelos dois bytes menos significativos. Esta encriptação é, posteriormente, descodificada pelo EDAS e após a validação envia uma mensagem com uma *string* com o valor "CHECK\04". Com o intuito de facilitar a compreensão deste processo de gerar o código de acesso, apresenta-se seguidamente um exemplo.

Exemplo:

**RN** (*Random number*) = **3766** (varia entre 2048 e 9999)

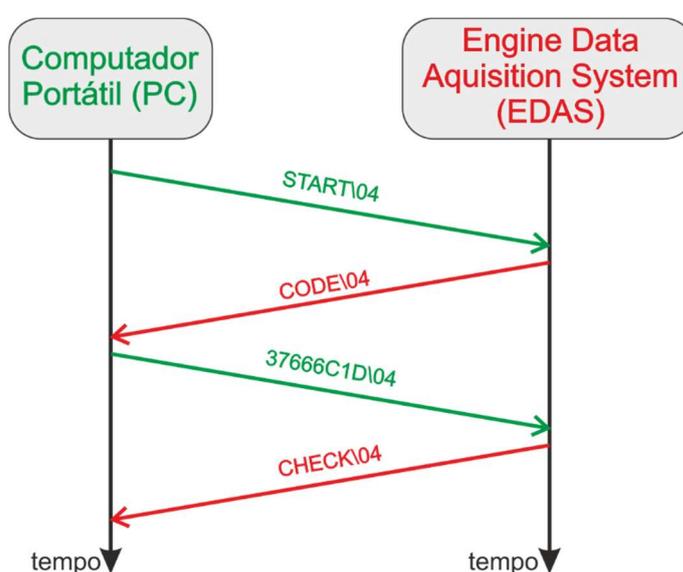
**RNC** (*Random number calculated*) = **RN** x 2 ⇔ (2 bytes msb trocar por 2 bytes lsb)

**RNC** (*Random number calculated*) ⇔ **3766** x 2 = 7532 (0x1D6C) ⇔ 27677 (0x6C1D)

**Tabela 4.5 – Protocolo de comunicação na fase de Handshake.**

Enviar para o EDAS	Recebe do EDAS
START\04	CODE\04
RNRNC\04	CHECK\04

Seguidamente apresenta-se a Figura 4.11 que ilustra a sequência de mensagens, ao longo do tempo, durante a fase de **Handshake**.



**Figura 4.11 – Exemplo de uma sequência de mensagens do Handshake.**

### **Protocolo de comunicação na fase Config**

Esta etapa tem como objetivo configurar qual o modo de aquisição e controlo do EDAS. De modo a simplificar, utiliza-se apenas o envio de uma *string* através da porta série que contém informação relativa ao modo de aquisição e controlo. O EDAS, posteriormente, envia uma *string* de feedback para o PC. Esta primeira *string* é sempre constituído por dez caracteres, os primeiros quatro correspondem a uma *string* contante com o valor de "MODE", o quinto e sexto caracter corresponde ao modo de aquisição, o sétimo, oitavo e nono caracter corresponde ao modo de controlo do motor e por fim o último caracter corresponde ao caracter de terminação.

O modo de aquisição é definido através da versão da unidade do motor, permitindo assim uma maior flexibilidade de expansão do sistema. As versões da unidade podem tomar o valor de S6, S7, S8 e S0, onde as primeiras três adquirem mensagens através do método *Broadcast* do barramento vermelho e ultima versão utiliza o método genérico de aquisição (Barramento verde).

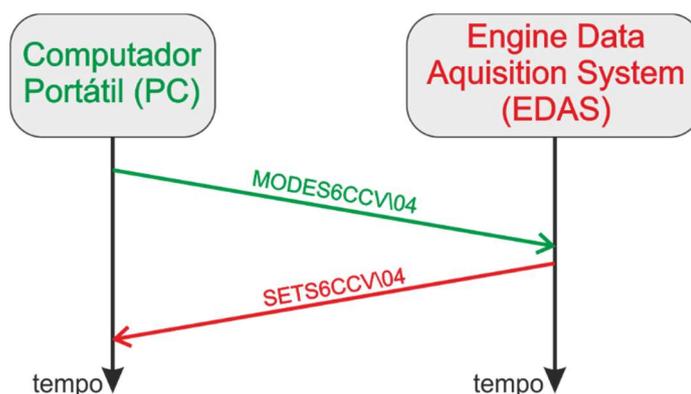
Quanto ao modo de controlo do motor, divide-se em três opções, posição do acelerador (APS), *Cruise Control* (CCV) e desativado (OFF). A opção de controlo posição do acelerador (APS) utiliza o potenciômetro digital presente no EDAS. Relativamente ao método de controlo *Cruise Control* utiliza apenas o barramento vermelho transmitindo assim mensagens CAN de controlo. A terceira e última opção é desativar o modo de controlo permitindo assim só aquisição de dados durante o uso do EDAS.

A Tabela 4.6 contém informação relativa as mensagens de configuração, sendo utilizado apenas duas mensagens a primeira contem a informação relativa ao modo de operação e a segunda é o feedback da configuração efetuada por parte do EDAS. Apresenta-se a Figura 4.12 com o intuito de exibir a sequência de mensagens de configuração ao longo do tempo.

**Tabela 4.6 - Protocolo de comunicação na fase de Config.**

**XX – Modo de aquisição; YYY – Modo de controlo.**

<b>Enviar para o EDAS</b>	<b>Recebe do EDAS</b>
MODEXXYYY\04	SETXXYYY\04



**Figura 4.12 - Exemplo de uma sequência de mensagens durante a configuração (Config).**

## Protocolo de comunicação na fase Request

Relativamente a última etapa do protocolo de comunicação tem como objetivo enviar pedidos de conjuntos de parâmetros, envio de mensagens de controlo do motor e ordens de início e fim da aquisição. Este grupo de mensagens divide-se em três subgrupos, início e fim da aquisição, aquisição de parâmetros e controlo do motor. Apresenta-se a Tabela 4.7 que contem os três grupos de mensagens.

A ordem de início e fim da aquisição utiliza as *strings*, “INIT AC” e “STOP AC”, respetivamente. Estas mensagens de ordem são enviadas pelo PC para o EDAS e não necessitam de feedback. O subgrupo de aquisição de parâmetros contém três mensagens diferentes, “RQSTH”, “RQSTL” e “RQST1”. Estes três pedidos correspondem as diferentes prioridades dos parâmetros definidos na Tabela 4.2. Para cada um dos pedidos, o EDAS envia um conjunto de informações, tal como é demonstrado na Tabela 4.7. Por fim, o subgrupo relativo ao controlo do motor é composto por quatro mensagens, enumerando, “SET TP<sub>yyy</sub>”, “SET CC<sub>w</sub>”, “STOP EN” e “START EN”. A mensagem “SET TP<sub>yyy</sub>” envia a informação relativa a posição do acelerador em percentagem onde o *yyy* pode variar entre 0 e 100. A mensagem “SET CC<sub>w</sub>” corresponde ao método de controlo do *Cruise Control*. Este controlo define-se através de três estados, UP, DOWN e STOP onde o carater *w* corresponde aos estados possíveis *u*, *d* ou *s*.

Com o intuito de facilitar a compreensão desta troca de mensagens durante a aquisição, apresenta-se a Figura 4.13. Esta figura ilustra a sequência de mensagens ao longo do tempo e identifica o emissor das mesmas.

**Tabela 4.7 – Protocolo de comunicação na fase de Request.**

Enviar para ChipKIT	Recebe do ChipKIT
INIT AC	
STOP AC	
RQSTH	RHMAPxxxxFPxxxxOPxxxxDDTxxELTxxRPMxxxxAFCxxxxMAFxxxxRE
RQSTL	RLIATxxAATxxxxFTxxECTxxAAPxxxxBTxxxxFTLxxCFSxxxxRE
RQST1	R1ERTxxxxEHxxxxxxxxVINxxxxxxxxRE
SET TP <sub>yy</sub>	
SET CC <sub>w</sub>	
STOP EN	
START EN	

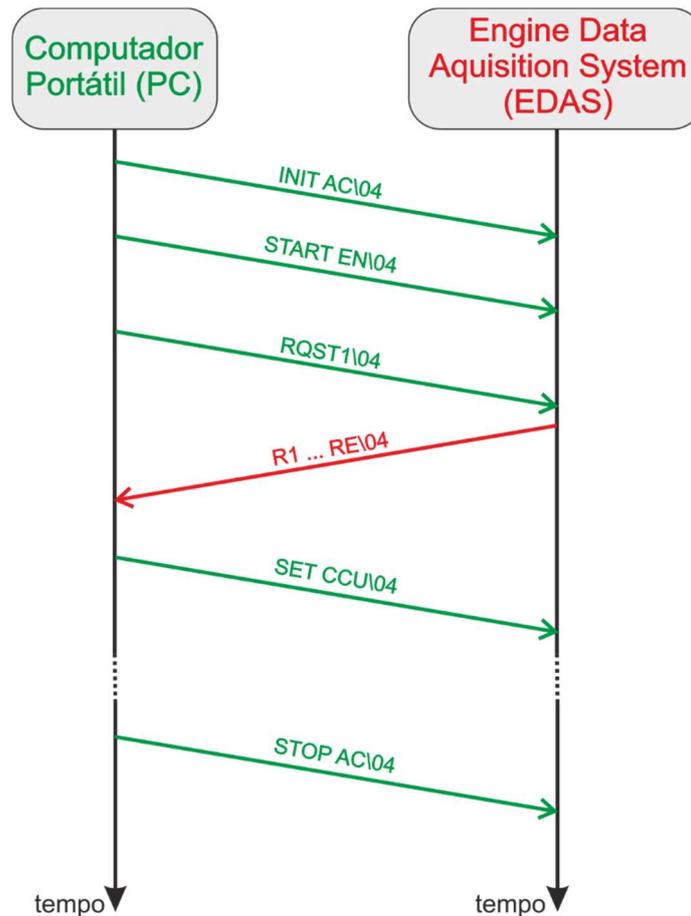


Figura 4.13 – Exemplo de uma sequência de mensagens durante a aquisição e controlo (Request).

## 4.6. Desenvolvimento do código do microcontrolador

---

Nesta secção aborda-se o desenvolvimento do código do microcontrolador onde se apresenta qual é o fluxo do programa e quais as funções mais importantes. Para isso dividiu-se o código em duas etapas, a primeira etapa é relativa a inicialização e configuração dos periféricos. A segunda parte expõe o código do programa principal onde contém as funções principais e as suas dependências.

Seguidamente, apresenta-se duas subsecções, a Fase inicial e o Programa Principal. É apresentado inicialmente as principais características das funções integradas e posteriormente o respetivo fluxograma.

## 4.6.1. Fase inicial

---

Relativamente à fase inicial, o fluxograma presente na Figura 4.14 expõe todas as etapas necessárias efetuar até executar o programa principal. Estas etapas tem o objetivo de aumentar a fiabilidade e a segurança do sistema e a possibilidade de expansão.

Interpretando o fluxograma verifica-se que numa primeira fase configura-se os periféricos, ou seja, todos os componentes presentes nos EDAS. É nesta fase que se inicia os vários tipos de comunicação presentes no sistema, tais como, SPI (Potenciómetro Digital), Série (USB PC), paralela (LCD). Após este início de comunicação é necessário configurar certos componentes, tais como o potenciómetro digital.

Após concluída as configurações dos periféricos, o programa inicia o seu ciclo de funcionamento. Este ciclo é composto por quatro subciclos, enumerando-os, **FirstSetup**, **Handshake**, **Config** e o **MainProgram**. Todos os subciclos estão sujeitos a certas condições iniciais de modo a garantir o correto funcionamento dos mesmos. De modo a facilitar a compreensão, prossegue se com a apresentação individual de cada função.

### **FirstSetup**

Esta função tem como objetivo iniciar os filtros e as máscaras para aquisição de mensagens do barramento CAN. Estes filtros e máscaras definem-se independentemente do barramento CAN, ou seja, o barramento CAN1 pode utilizar um conjunto de filtros e o CAN2 utilizar outro conjunto distinto de filtros. Após a configuração e inicialização dos filtros, procede-se com o início da comunicação CAN nos dois barramentos presentes no chipKIT®. Esta função encontra se integrada no ciclo de modo a permitir uma alteração posterior dos filtros e das máscaras sem ser necessário parar a execução do programa ou reiniciar.

### **Handshake**

A função **Handshake** permite inicializar e validar a comunicação entre o EDAS e o PC. Para tal, esta função utiliza o protocolo de comunicação dedicado anteriormente apresentado (4.5). Tal como o nome indica esta função refere-se a fase de **Handshake** do protocolo de comunicação. Numa primeira fase a função aguarda por instruções de inicialização através de mensagens na porta série. Quando ocorre essa instrução, o EDAS solicita um código e aguarda pela resposta do PC. Posteriormente o PC envia o código onde será analisado pelo EDAS e verifica se o código se encontra em conformidade com

o protocolo de comunicação. Após a validação deste código o EDAS envia a confirmação da validação para o PC finalizando assim a etapa de **Handshake** e fechando assim o ciclo avançando para a próxima função do microcontrolador.

### **Config**

Após a validação da comunicação, efetuada durante a fase de **Handshake**, o microcontrolador aguarda por informações relativas ao modo de aquisição e controlo do motor. Portanto o objetivo desta função **Config** é receber estas informações por parte do utilizador (PC) e programar assim internamente os seus registos. Esta função utiliza o protocolo de comunicação dedicado na fase **Config**, ou seja, aguarda uma mensagem na porta serie relativa a configuração do modo de operação. Após a receção desta mensagem é verificada a sua veracidade da informação e posteriormente programado os registos internos. Após a definição do modo de operação, o EDAS envia uma mensagem de feedback para o PC contendo os valores programados.

### **MainProgram**

Posteriormente à conclusão dos subciclos apresentados anteriormente o programa principal é executado, repetidamente, enquanto as condições de **Handshake** e **Config** se verificarem positivas. O programa principal é executado continuamente até que alguma destas condições não se verifiquem, o que aumenta a fiabilidade dos resultados obtidos durante os ensaios. Com esta arquitetura é possível sair deste ciclo e voltar a reconfigurar sem ser necessário reiniciar o *hardware*.

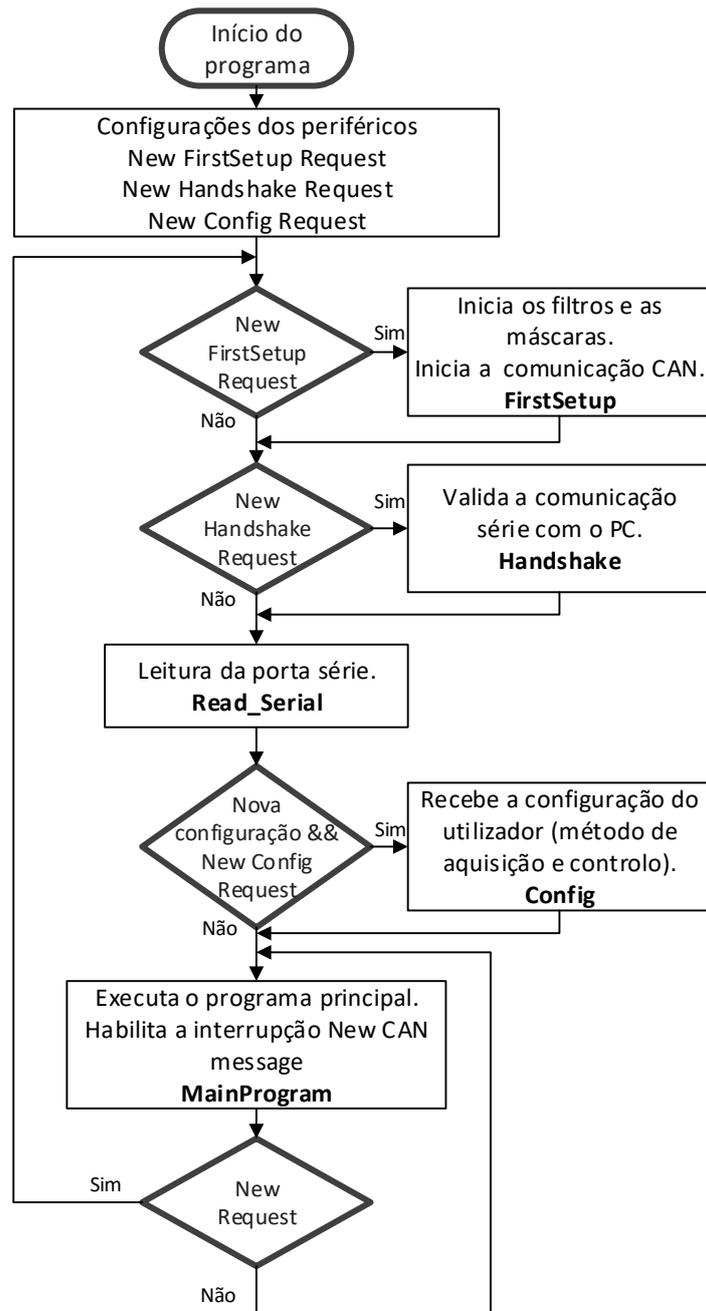


Figura 4.14 – Fluxograma da fase inicial do código.

## 4.6.2. Programa Principal (Main Program)

---

A segunda fase do código do microcontrolador corresponde ao programa principal (MainProgram). Nesta fase decorre toda a aquisição e controlo das variáveis definidas anteriormente. O programa principal possui um fluxo principal que é composto por três operações, leitura e execução de mensagens da porta série, exportar informação através da porta série e por fim procede-se ao controlo do motor através do valor do potenciómetro digital.

Durante este ciclo principal pode ocorrer interrupções geradas por mensagens no barramento CAN. Quando o EDAS recebe uma mensagem CAN, procede a sua descodificação e guarda o valor do parâmetro recolhido da mensagem. De seguida, quando solicitado pelo utilizador, é enviada uma mensagem CAN com o pedido de parâmetros (método de diagnóstico SDP3).

De seguida apresenta-se o fluxo principal e a execução da interrupção gerada por uma mensagem CAN. Descreve-se o processo de aquisição e controlo do sistema de forma pormenorizada e posteriormente exhibe-se o fluxograma relativo ao programa principal, ou seja, o ciclo principal e a interrupção CAN.

### **Fluxo principal do programa**

O fluxo principal do programa inicia-se com a leitura de comandos recebidos pela porta série, onde de seguida interpreta-se esses comandos. Estes comandos dividem-se em três tipos aquisição de parâmetros, controlo do motor e início/fim da aquisição. Estes comandos encontram-se disponíveis na Tabela 4.7 na fase de **Request**.

Posteriormente, se o modo de aquisição se encontrar ativo, envia-se os valores dos parâmetros adquiridos através da comunicação série. Tal como se encontra descrito na subsecção 2.1.1, dividiu-se os parâmetros a adquirir em três grupos, o RQSTH, RQSTL e RQST1. O RQSTH corresponde ao grupo de parâmetros com uma taxa de atualização elevada, o grupo RQSTL é constituído pelos parâmetros com uma taxa de atualização baixa e o RQST1 corresponde ao conjunto de parâmetros que só é necessário adquirir no início de cada ensaio, ou seja, só são adquiridos uma vez.

Por fim procedesse-se a atualização do valor da posição do potenciómetro digital. Este valor é definido pelo utilizador e recebe-se através dos comandos da porta serie de controlo do motor.

Apresenta-se o fluxograma das operações descritas anteriormente na Figura 4.15. Este fluxograma contém uma breve descrição das várias funções presentes e a ordem de ação das mesmas.

### **Interrupção mensagem CAN**

Durante a execução do programa pode ocorrer interrupções. Estas interrupções são originadas pela receção de novas mensagens CAN no barramento, ou seja, sempre que o EDAS recebe uma nova mensagem CAN o programa é suspenso e executa a interrupção. A esta interrupção associa-se um conjunto de operações relativas ao processamento e descodificação da mensagem recebida. Após a descodificação da mensagem, verifica-se se a mensagem contém algum valor dos parâmetros presentes na Tabela 4.2. Se isso se verificar, atualiza-se o valor do parâmetro e posteriormente envia-se as mensagens CAN de pedido de parâmetros.

O método de controlo do motor através do *Cruise Control* requer o envio de mensagens para o barramento contendo as instruções provenientes pelo utilizador. Sempre que o COO envie para o barramento a mensagem relativa ao estado do *Cruise Control*, o EDAS envia o novo estado. Após a finalização de todas as operações relativas a interrupção o programa volta ao seu ciclo principal e continua no ponto onde se encontrava no momento em que surgiu a interrupção.

Quanto ao fluxograma relativo as operações da interrupção apresenta-se na Figura 4.16. Neste fluxograma é apresentado tanto as operações descritas anteriormente como a ordem de execução.

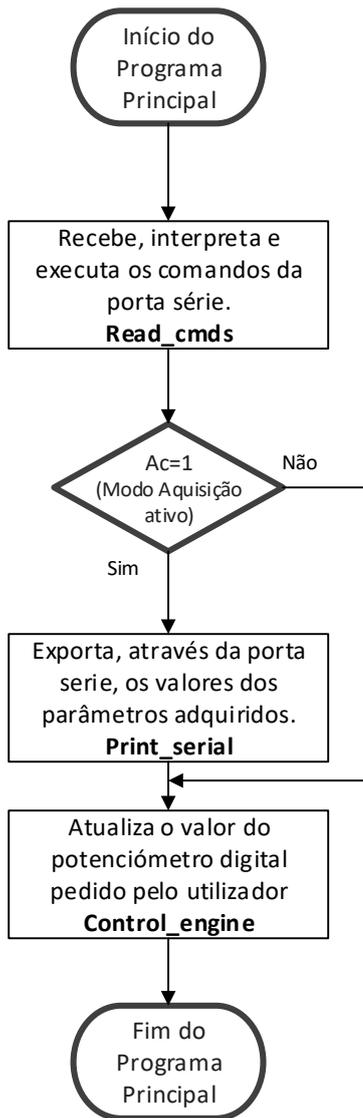


Figura 4.15 - Fluxograma do Programa Principal.

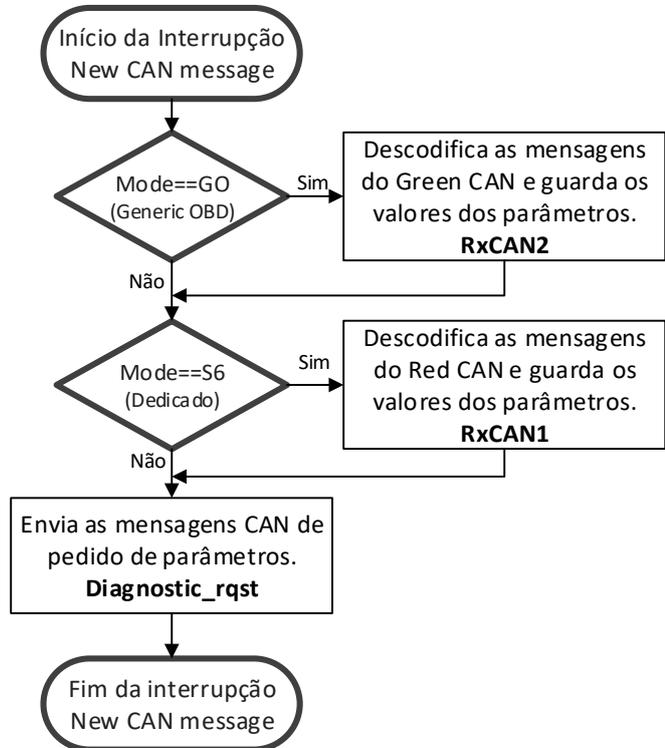


Figura 4.16 - Fluxograma da interrupção CAN.

### 4.6.3. Otimização da frequência de aquisição dos parâmetros (CAN)

---

Nesta secção aborda-se o tema da frequência da aquisição dos parâmetros durante um ensaio. Procedeu-se a recolha da frequência de atualização dos vários parâmetros selecionados utilizando o método de aquisição *Broadcast* RED Bus. Recolheu-se também o valor máximo de aquisição utilizando o método de aquisição diagnóstico SDP3. Após a recolha destas informações compilou-se a Tabela 4.8, onde se encontra a lista dos vários parâmetros a adquirir, a frequência de atualização dos vários métodos e por fim qual o método escolhido para cada parâmetro.

Segundo a tabela, a frequência de atualização do método *Broadcast* varia entre 0,2 Hz e 50 Hz. Porém dois parâmetros não são possíveis de adquirir através deste método sendo então necessário recorrer ao método de diagnóstico SDP3. Quanto ao método de diagnóstico, após efetuados alguns testes verificou-se que a frequência máxima de aquisição de cada parâmetro é de 20 Hz se adquirirmos até 6 parâmetros neste método.

Posteriormente verificou-se que a frequência de atualização de alguns parâmetros, através do método *Broadcast*, era demasiada baixa. Portanto recorreu-se assim o método diagnóstico para adquirir estes parâmetros. Resumindo, durante um ensaio recolhe-se 19 parâmetros diferentes, onde apenas seis se adquire pelo método de diagnóstico SDP3 e os restantes adquirem-se pelo método *Broadcast* RED Bus.

**Tabela 4.8 – Frequência de aquisição dos parâmetros.**

Nome do parâmetro	Frequência atualização [Hz]		Método de aquisição utilizado
	Broadcast	SDP3	
Temperatura de Admissão	2	20*	Broadcast RED Bus
Temperatura Ambiente	1	20*	Broadcast RED Bus
Temperatura de combustível	1	20*	Broadcast RED Bus
Temperatura do líquido refrigeração	1	20*	Broadcast RED Bus
Pressão de admissão (MAP)	2	20*	Diagnóstico SDP3
Pressão Ambiente (barométrica)	1	20*	Broadcast RED Bus
Pressão de alimentação de combustível	2	20*	Diagnóstico SDP3
Pressão de óleo	2	20*	Diagnóstico SDP3
Carga do pedal de acelerador	50	20*	Broadcast RED Bus
Carga do motor ( <i>Engine Load</i> )	50	20*	Broadcast RED Bus
Rotação do motor	50	20*	Broadcast RED Bus
Consumo de combustível atual	10	20*	Broadcast RED Bus
Massa de ar admitida (MAF)	-	20*	Diagnóstico SDP3
Tensão da bateria	-	20*	Diagnóstico SDP3
Nível do Tanque	1	20*	Broadcast RED Bus
Velocidade rotação ventoinha	1	20*	Diagnóstico SDP3
Binário de referência do motor	0,2	20*	Broadcast RED Bus
Horas de funcionamento do motor	1	20*	Broadcast RED Bus
Número de chassis	0,2	20*	Broadcast RED Bus

(\*) – Valor máximo utilizando até seis parâmetros pelo método diagnóstico SDP3.



## 5. *Software* de visualização e controlo

---

Tal como o título sugere neste capítulo aborda-se todos os temas relativos ao *software* desenvolvido para visualização dos parâmetros e controlo do motor. Primeiramente apresenta-se a ferramenta de desenvolvimento, LabVIEW®, de seguida expõem-se as características pretendidas para o *software*. Posteriormente exhibe-se o fluxograma do *software* e as funcionalidades implementadas. Por fim procede-se com, a apresentação do ambiente gráfico e os vários modos de funcionamento.

### 5.1. Ferramenta de desenvolvimento

---

Nesta secção apresenta-se a ferramenta de desenvolvimento utilizada, o programa LabVIEW® da National Instruments®. Este programa permite integrar vários sistemas de aquisição em simultâneo possibilitando assim uma maior facilidade de expansão do projeto através da utilização de placas de aquisição. O programa possui vários módulos extra que se podem instalar posteriormente sem ser necessário refazer o software. Quanto à versão utilizada, durante o desenvolvimento do projeto, é a versão mais recente disponibilizada para estudantes pela National Instruments® 2015 com o módulo de comunicação série (VISA).

O programa é composto por duas partes, o diagrama de blocos (*Block Diagram*) e o painel frontal (*Front Panel*). É no diagrama de blocos que ocorre o processamento das funções, que posteriormente exhibe os resultados no painel frontal. Quanto a programação do diagrama de blocos é baseado no método de programação em blocos. Este método facilita a interligação e concatenação de funções, a ligação entre o diagrama de blocos e o painel frontal, a inclusão de novas funções. O método utilizado promove uma maior facilidade de aprendizagem e versatilidade.

Seguidamente, apresenta-se um exemplo de um diagrama de blocos e de um painel frontal, respetivamente na Figura 5.1 e Figura 5.2. Nestes exemplos verifica-se que existe uma ligação perceptível entre o diagrama de blocos e o painel frontal. O exemplo é composto por um knob, cuja posição é controlada pelo utilizador e dois indicadores, um manómetro e uma barra vertical. Os valores destes indicadores atualizam-se continuamente durante a execução do programa.

Posteriormente apresenta-se o fluxograma do *software* desenvolvido com esta ferramenta. O *software* contém varias funções descritas no diagrama de blocos e os seus indicadores expostos no painel frontal.

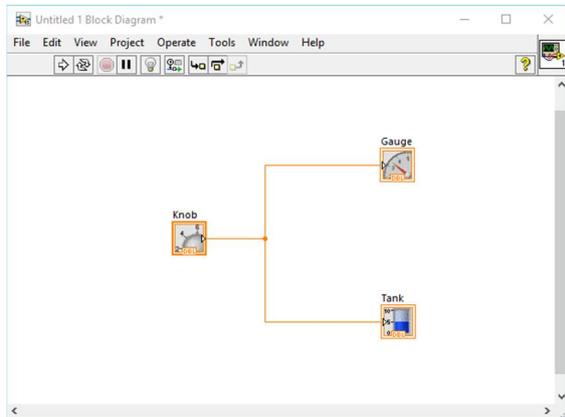


Figura 5.1 - Exemplo de um diagrama de blocos do LabVIEW®.

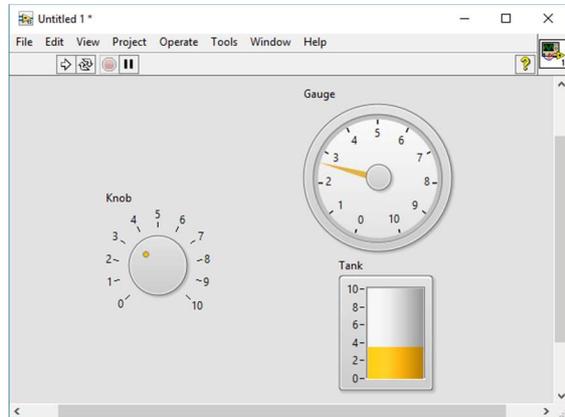


Figura 5.2 - Exemplo de um painel frontal do LabVIEW®.

## 5.2. Metodologia

---

Nesta secção apresenta-se o diagrama de blocos simplificado da aplicação desenvolvida com a ferramenta LabVIEW®. A aplicação é composta por várias funções desenvolvidas em conjunto com as funções do EDAS. A Figura 5.3 expõe o diagrama de blocos simplificado contendo todas as funções e sequências essenciais à aplicação. A aplicação é composta por um bloco sequencial com três etapas o Início do programa, o **Handshake** e o **Programa de Aquisição e Controlo**. O início do programa incorpora a função de inicialização da porta série e o reset das variáveis. Quanto à etapa de **Handshake** incorpora a função de **Handshake** que tem como objetivo validar a comunicação entre o EDAS e o PC segundo o protocolo de comunicação descrito na secção 4.5 deste documento.

Após a validação da comunicação, o programa inicia a etapa **Programa de Aquisição e Controlo**. Esta etapa é um ciclo que incorpora uma sequência de quatro fases o **Menu Principal**, o **Configuração**, a **Aquisição e controlo** e por último a etapa de **Guardar os dados** recolhidos. De seguida apresenta-se cada uma das fases onde se expõe os seus objetivos e as funções implementadas.

## **Menu Principal**

O objetivo desta fase é apresentar ao utilizador um menu de opções com uma fácil interpretação e leitura. Este menu contém a configuração do método de aquisição e controlo, a definição dos parâmetros de emergência, a definição do local de gravação do ficheiro de dados recolhidos durante o ensaio, o botão de execução em modo Admin e o botão de início da aquisição.

Após a inicialização, o painel frontal apresenta o menu e assim o utilizador seleciona os parâmetros que pretende e posteriormente pressiona o botão de início de aquisição e controlo. Ao pressionar o botão esta fase é finalizada avançando assim para a fase seguinte, configuração.

## **Configuração**

Antes de iniciar a aquisição e controlo do motor, é necessário configurar o EDAS. Esta fase tem como objetivo configurar o *hardware* e o *software*. Segundo o protocolo de comunicação apresentado na secção 4.5, a mensagem de configuração contém o modo de aquisição e o modo de controlo do motor. Portanto nesta fase envia-se uma mensagem para o EDAS com a informação dos métodos de aquisição e controlo selecionado pelo utilizador no **Menu principal**. O EDAS após receção da mensagem de configuração, este interpreta a mensagem e configura os parâmetros associados. Posteriormente envia uma mensagem de feedback, contém os parâmetros configurados, para o PC com o intuito de validar a configuração assim avançar para a próxima fase.

De seguida a aplicação desenvolvida valida os parâmetros configurados no EDAS e posteriormente envia a mensagem com a ordem de início da aquisição e controlo. Após a conclusão desta etapa, a aplicação inicia o ciclo de aquisição e controlo.

## **Aquisição e Controlo**

Antes de iniciar o ciclo de aquisição e controlo envia-se a mensagem de pedido de parâmetros inicial, "RQST1" e posteriormente inicia-se o ciclo associado a esta fase. O ciclo é constituído por cinco etapas, pedido de parâmetros de alta prioridade, pedido de

parâmetros de baixa prioridade, processamento e visualização dos parâmetros recolhidos, controlo do motor e por fim a etapa de estabelecer a frequência do ciclo.

Todas estas etapas decorrem sequencialmente de forma a não bloquear o programa e garantir assim a execução completa das mesmas. Relativamente às duas primeiras etapas de pedido de parâmetros executam-se de acordo o protocolo de comunicação apresentado na secção 4.5. Estas mensagens de pedido de parâmetros dividem-se em duas taxas de aquisição diferentes, onde os parâmetros com alta prioridade são adquiridos a uma frequência dez vezes superior aos parâmetros de baixa prioridade. Após a receção dos parâmetros procedesse a sua exibição no painel frontal através de dois métodos, manómetros (valores em tempo real) e gráficos (valores adquiridos durante o tempo de aquisição). Todos estes valores guardam-se num vetor de dados que posteriormente será gravado num ficheiro de dados.

Após a conclusão das etapas relativas à aquisição e visualização dos parâmetros prossegue-se com a etapa de controlo do motor. A etapa de controlo permite ao utilizador controlar o motor através de dois métodos, através da posição do pedal de acelerador e do cruise control. Para cada um dos métodos definiu-se um painel frontal de modo a facilitar a utilização. A frequência de controlo do motor é independente do método seleccionado pelo utilizador.

Por fim executa-se a etapa de estabelecer a frequência da aquisição e controlo. Para tal utiliza-se a função de metrónomo do LabVIEW® que permite definir um tempo de ciclo constante, ou seja, sempre que o tempo de execução do ciclo de aquisição e controlo for inferior ao tempo definido pelo utilizador, o programa aguarda o tempo necessário para recomençar o ciclo garantindo assim a uma taxa de aquisição constante. Enquanto o utilizador não premir o botão de parar aquisição o programa executa este ciclo de aquisição e controlo.

### **Guardar Dados**

Após a ordem de cessação da aquisição e controlo por parte do utilizador, a aplicação grava todos os dados recolhidos e de envia uma mensagem para o EDAS com a ordem de parar a aquisição e controlo do motor. Os dados recolhidos durante a aquisição guarda-se num ficheiro com a terminação “.dp”. Estes ficheiros podem ser explorados através de um programa de folha de cálculo. Definiu-se uma tipologia de ficheiro diferente do convencional de modo a promover uma maior exclusividade a aplicação desenvolvida.

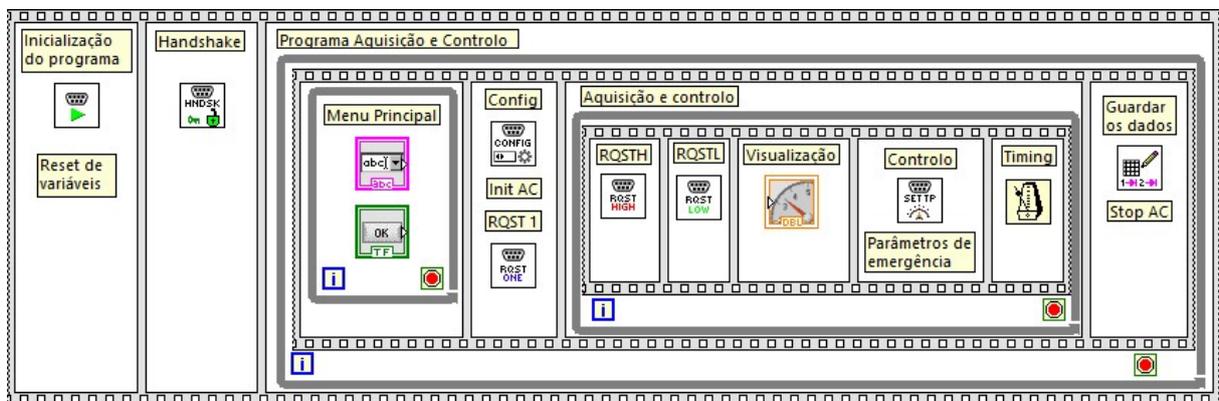


Figura 5.3 - Diagrama de blocos simplificado da aplicação desenvolvida.

## 5.3. Funções Implementadas

Esta secção tem como objetivo expor algumas das funções implementadas na aplicação desenvolvida. Tal como se pode verificar no digrama de blocos representado na Figura 5.3, a aplicação necessita de várias funções durante a execução do programa. Apresenta-se de seguida algumas dessas funções, tais como, handshake (HNDSK), configuração (CONFIG), pedido de parâmetros de alta prioridade (RQSTH), controlo da posição do acelerador (SET TP) e guardar os dados.

### Handshake (HNDSK)

A função *handshake* tem como objetivo validar a comunicação entre o PC e o EDAS utilizando o protocolo apresentado na secção 4.5. O protocolo define o envio de uma primeira mensagem do PC para o EDAS, de seguida o EDAS envia uma mensagem com o pedido de um código. Após o envio do código por parte do PC e a sua validação o EDAS envia a mensagem de validação de comunicação.

A Figura 5.4 apresenta o diagrama de blocos da função e a sequência de operações relacionadas com esta função. A função *Handshake* é constituída por 5 etapas, na primeira envia-se a mensagem “START\04”, de seguida aguarda-se pela receção da mensagem “CODE\04”. Após a receção da mensagem procede-se com a geração do código que posteriormente se envia para o EDAS. Este código sofre alterações descritas no protocolo de comunicação (4.5) de modo a dificultar o acesso por elementos “estranhos”.

Seguidamente aguarda-se pela mensagem de validação do código por parte do EDAS. Através da receção desta ultima mensagem valida-se a comunicação terminando assim a função com a alteração de um indicador booleano.

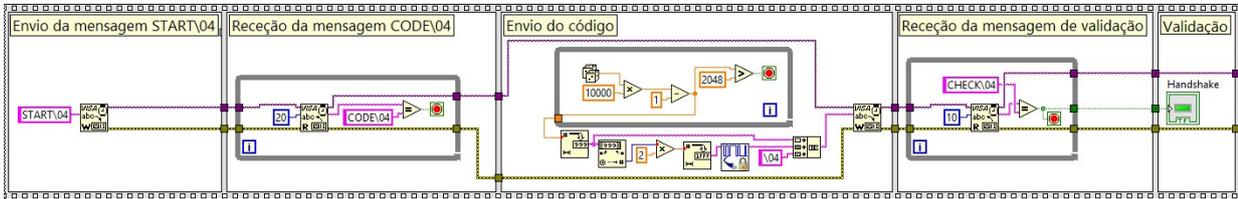


Figura 5.4 – Diagrama de blocos da função *Handshake*.

### Configuração (CONFIG)

O objetivo desta função é configurar o modo de aquisição e controlo do EDAS seleccionados pelo utilizador. Então a função de configuração só inicia após a seleção do modo de aquisição e de controlo por parte do utilizador.

Apresenta-se a Figura 5.5 que representa o diagrama de blocos da função **Configuração**. Verifica-se que a função é composta por três etapas, onde primeiramente envia-se o modo de aquisição e de controlo para o EDAS. Após o envio da mensagem com o modo aguarda-se pela mensagem de feedback do EDAS validando assim a configuração do sistema. Posteriormente guarda-se internamente o modo de controlo de modo alterar o painel frontal relativo ao modo de controlo escolhido.

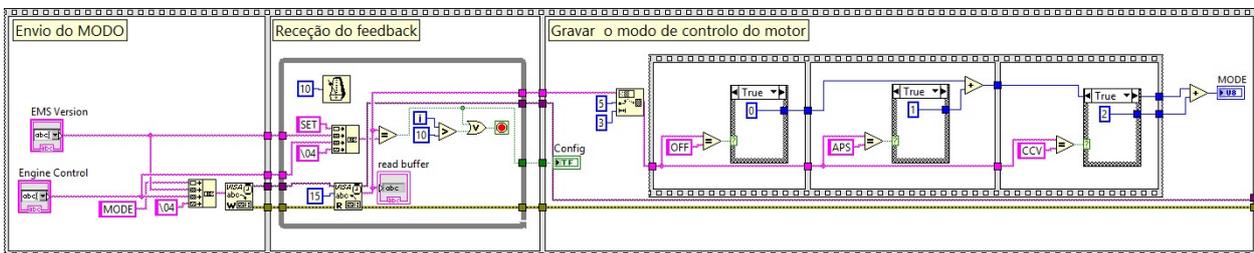


Figura 5.5 - Diagrama de blocos da função **Configuração**.

## Pedido de parâmetros de alta prioridade (RQSTH)

A função mais utilizada durante o ciclo de aquisição e controlo é a função pedido de parâmetros de alta prioridade. Esta função tem como objetivo solicitar ao EDAS os parâmetros de alta prioridade mais recentes e posteriormente descodifica-los e guardar nas variáveis correspondentes.

A Figura 5.6 representa o diagrama de blocos da função RQSTH que é composto apenas por duas etapas, o envio da mensagem a solicitar os parâmetros e a etapa de receção e descodificação dos parâmetros presentes na mensagem. Após a receção da mensagem identifica-se quais os parâmetros contidos, de seguida converte-se os valores segundo a norma SAE J1939. Por fim atribui-se estes valores convertidos aos parâmetros correspondentes.

Quanto as outras mensagens de pedido de parâmetros, RQSTL e RQST1, processam-se da mesma maneira.

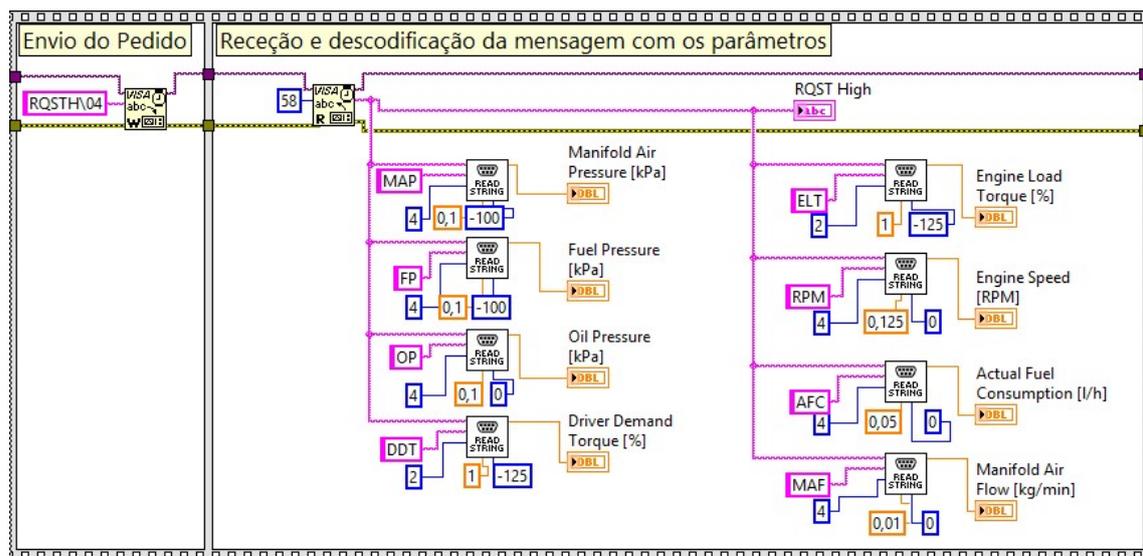
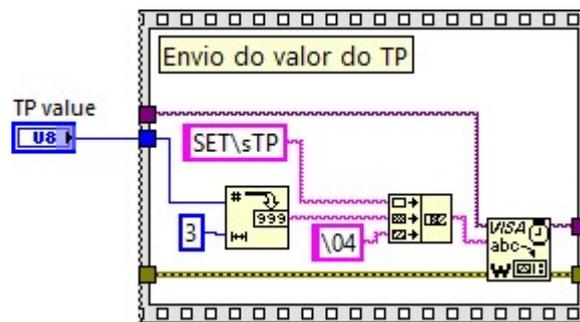


Figura 5.6 - Diagrama de blocos da função Pedido de parâmetros de alta prioridade.

## Controlo da posição do acelerador (SET TP)

O modo controlo do motor divide-se em três opções, controlo da posição do acelerador, controlo do *cruise control* ou controlo desativado. A função que se apresenta de seguida é relativa ao método de controlo da posição do pedal do acelerador. Esta função tem como objetivo enviar o valor da posição do acelerador (TP) seleccionado pelo utilizador para o EDAS.

Representa-se a na Figura 5.7 o diagrama de blocos da função controlo da posição do acelerador. A função apenas é composta por uma etapa, o envio do valor do TP para o EDAS. O valor da posição do acelerador é selecionado pelo utilizador durante o ciclo de aquisição e controlo. Quando se executa esta função adquire-se o ultimo valor da posição do acelerador, onde de seguida envia-se a mensagem para o EDAS.



**Figura 5.7 - Diagrama de blocos da função Controlo da posição do acelerador.**

### **Guardar os dados**

Por fim, apresenta-se a função Guardar os dados que tem como objetivo guardar todos os parâmetros adquiridos durante o ciclo de aquisição e controlo. A Figura 5.8 expõe o diagrama de blocos da função guardar os dados. Durante a execução do ciclo adquire-se os valores dos parâmetros, e no fim de cada iteração do ciclo guarda-se estes valores nas respetivas variáveis, tal como se demonstra no diagrama de blocos. Estes valores são armazenados de forma indexada ou seja a informação acumula-se durante o decorrer do ciclo. Quando o ciclo é finalizado esta informação é guardado num ficheiro de dados com a terminação “.dp”. Este ficheiro de dados contem também a data e hora do ensaio.

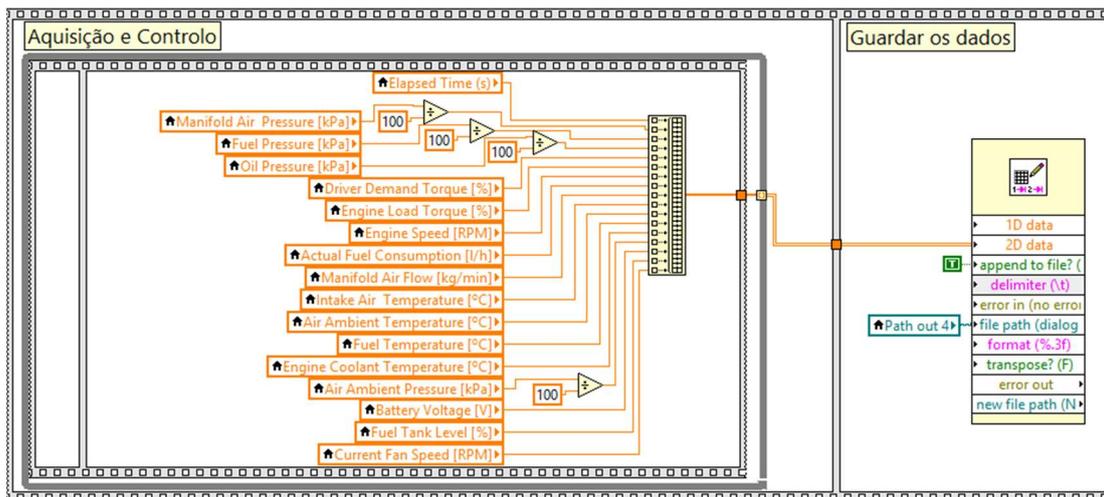


Figura 5.8 - Diagrama de blocos da função Guardar os dados.

## 5.4. Apresentação do Ambiente Gráfico

Por fim apresenta-se o ambiente gráfico da aplicação, ou seja, o painel frontal de interação com o utilizador. O objetivo do painel frontal é ser simples de utilizar durante o ensaio. Tendo em conta este objetivo dividiu-se em dois painéis, o painel principal e o painel de controlo. O painel principal ocupa a maior parte do ambiente gráfico enquanto o painel de controlo é uma coluna a direita do ambiente gráfico.

O painel principal é composto por quatro *layouts*, enumerando-os, o menu de seleção, o menu *admin*, a visualização instantânea e a visualização gráfica. Relativamente ao painel de controlo é constituído por quatro layouts, o controlo do acelerador, o controlo do *Cruise Control*, o controlo desativado e o layout menu. O *layout* menu do painel de controlo é apenas utilizado enquanto o menu de seleção do painel principal se encontrar ativo. De seguida apresenta-se as características e funcionalidades dos vários *layouts* presentes no painel frontal.

### Menu de Seleção

Quando se inicia o programa, o painel frontal apresenta o menu de seleção onde o utilizador seleciona os parâmetros de aquisição e controlo. Este *layout* permanece ativo enquanto o utilizador não pressionar o botão de início de aquisição. Quanto ao painel de controlo apresenta-se o *layout* menu corresponde a uma coluna à direita sem botões.

A Figura 5.9 apresenta o *layout* do menu de seleção que é composto por várias opções. Primeiramente seleciona-se qual o método de aquisição e controlo do ensaio que se irá

realizar. O método de aquisição identifica-se através da versão da unidade da EMS e o método de controlo é constituído por três opções, controlo OFF, *Cruise Control* e controlo do acelerador.

A etapa seguinte refere-se ao local de gravação do ficheiro de dados do ensaio. Seleciona-se o caminho de gravação e por fim um botão de ativar a gravação. De seguida define-se os valores dos parâmetros de emergência. Estes parâmetros de emergência podem solicitar a paragem do motor quando se encontra fora dos valores de segurança.

Por fim, define-se o valor da frequência de aquisição que pode variar entre 1 Hz e 20 Hz. Quando o utilizador premir o botão de início da Aquisição, o programa segue para outro *layout* dependo das opções selecionadas.

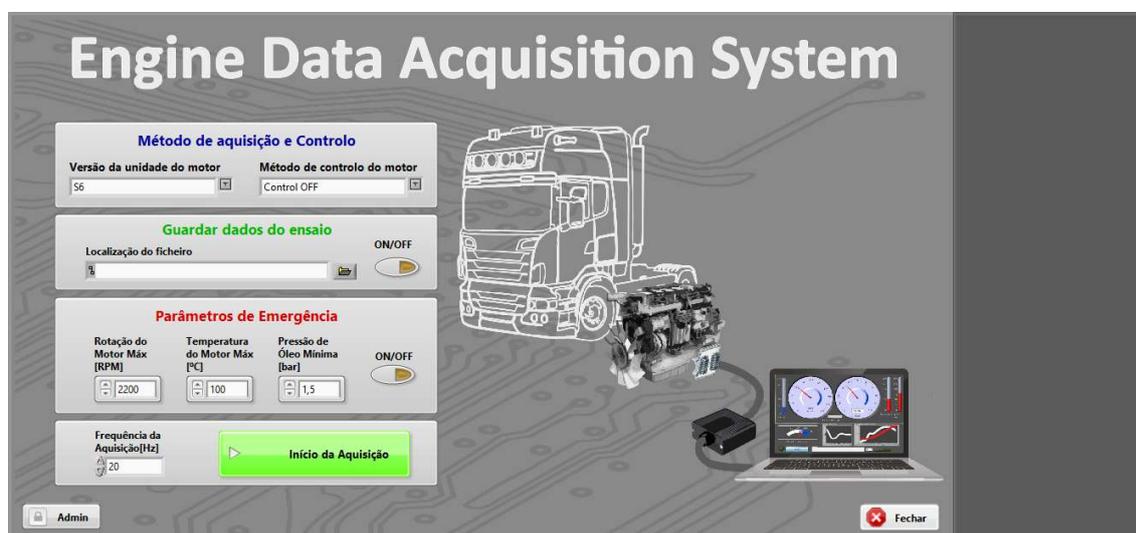


Figura 5.9 – *Layout* Menu de seleção.

### Menu Admin com controlo do motor desativado

Durante o menu de seleção o utilizador pode aceder ao modo *Admin* que lhe permite visualizar a receção das mensagens em tempo real. Para aceder a este modo é necessário introduzir uma palavra passe. Este modo *Admin* é composto pelo *layout Admin* em conjunto com o layout do controlo do motor desativado, tal como se apresenta na Figura 5.10.

O menu *Admin* é constituído pela visualização das três mensagens com os parâmetros pedidos em tempo real. Exibe-se também os valores descodificados das mensagens e dois indicadores das etapas concluídas de modo a permitir um modo de diagnóstico mais rápido. Neste modo *Admin* também é possível configurar a porta da comunicação série.

O painel de controlo, apesar da opção de controlo do motor se encontrar desativada é possível solicitar o arranque ou a paragem do motor através de dois botões. Por fim, para sair deste modo é necessário premir o botão voltar que ativa o menu de seleção novamente.

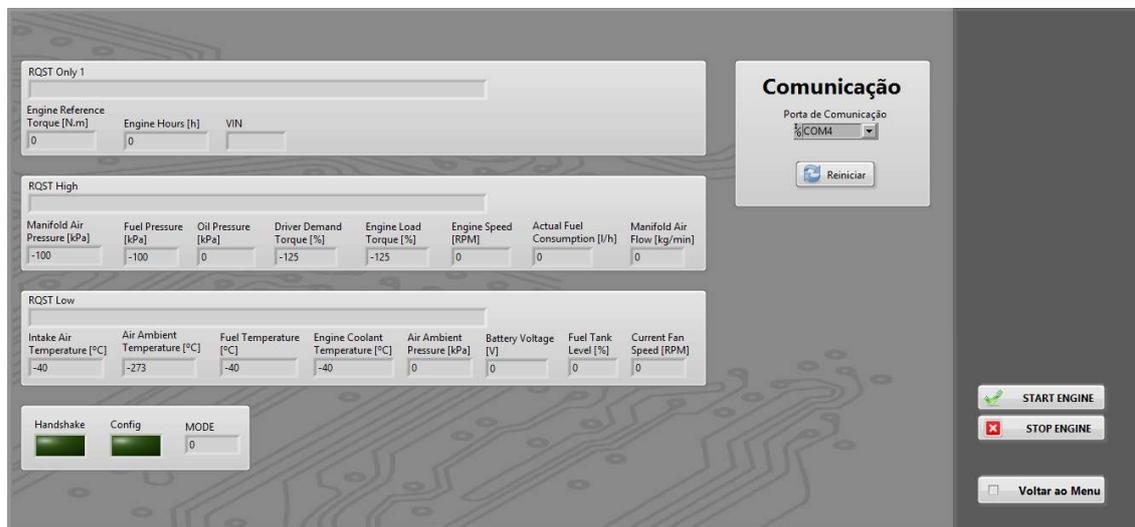


Figura 5.10 - *Layout Menu Admin.*

### Visualização instantânea com controlo do acelerador

Após a ordem de início do ciclo de aquisição e controlo, por defeito, o *layout* apresentado é o de visualização instantânea de parâmetros. O painel de controlo encontra-se dependente das opções selecionadas pelo utilizador durante o menu de seleção.

De seguida apresenta-se a Figura 5.11 onde se exhibe um exemplo de uma visualização instantânea com controlo do acelerador. No painel principal expõe-se todos os parâmetros em tempo real através de manómetros e indicadores. Os valores dos parâmetros exibem-se em indicadores numéricos junto dos seus manómetros. Agruparam-se os parâmetros de grandezas semelhantes forma a facilitar a sua análise. No painel de controlo a opção selecionada é controlo do acelerador que permite ao utilizador definir o valor acelerador em tempo real. Esta definição efetua-se através do deslocamento de uma barra vertical ou introdução do valor de percentagem do acelerador desejado.

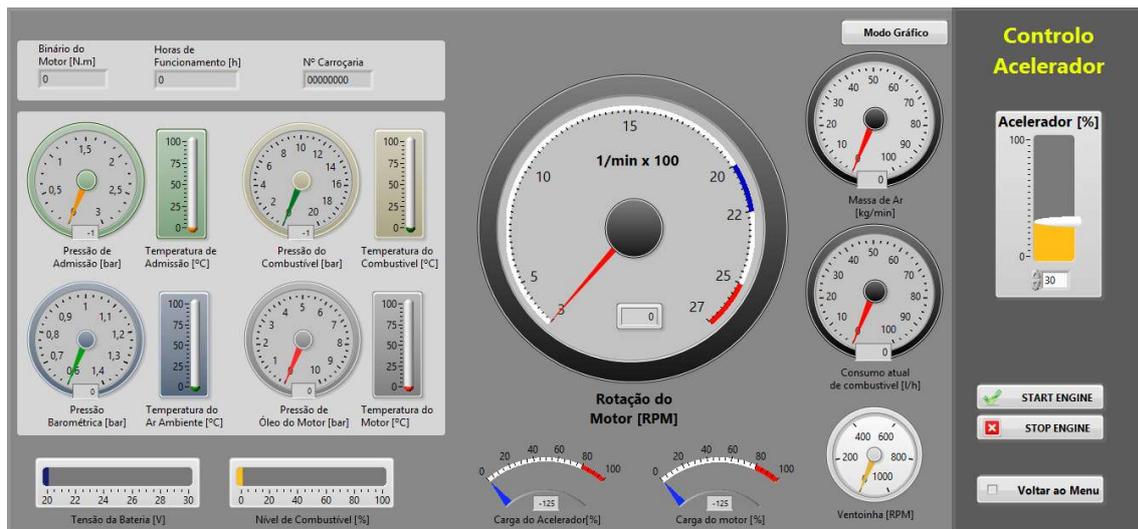


Figura 5.11 - *Layout* Visualização instantânea com controlo do acelerador.

### Visualização gráfica com controlo Cruise Control

Uma outra forma de visualizar os parâmetros recebidos durante o ciclo de aquisição e controlo é através de gráficos no tempo. Estes gráficos registam o valor dos parâmetros ao longo do tempo, chama-se a este *layout* visualização gráfica. Em conjunto com este tipo de visualização é possível utilizar-se qualquer um dos métodos de controlo apresentados.

A Figura 5.12 representa o *layout* visualização gráfica com controlo *cruise control* onde é composto por dois gráficos e um conjunto de botões de controlo do motor. Relativamente aos gráficos, o gráfico superior exibe a rotação do motor e quantidade de ar admitida ao longo do tempo. Quanto ao gráfico inferior apresenta-se a evolução de quatro temperaturas (admissão, ambiente, combustível e motor) ao longo do tempo. Através desta exibição gráfica facilita-se o diagnóstico do motor. O *cruise control* é efetuado através de três ações, acelerar, desacelerar e parar *cruise control*. A ação parar é composta por apenas um botão central "OFF", as restantes duas ações são constituídas, cada uma delas, por dois botões. Estes dois botões permitem dois tipos de controlo, um permite uma alteração do valor mais rápido e o outro uma alteração unitária possibilitando um controlo mais refinado.

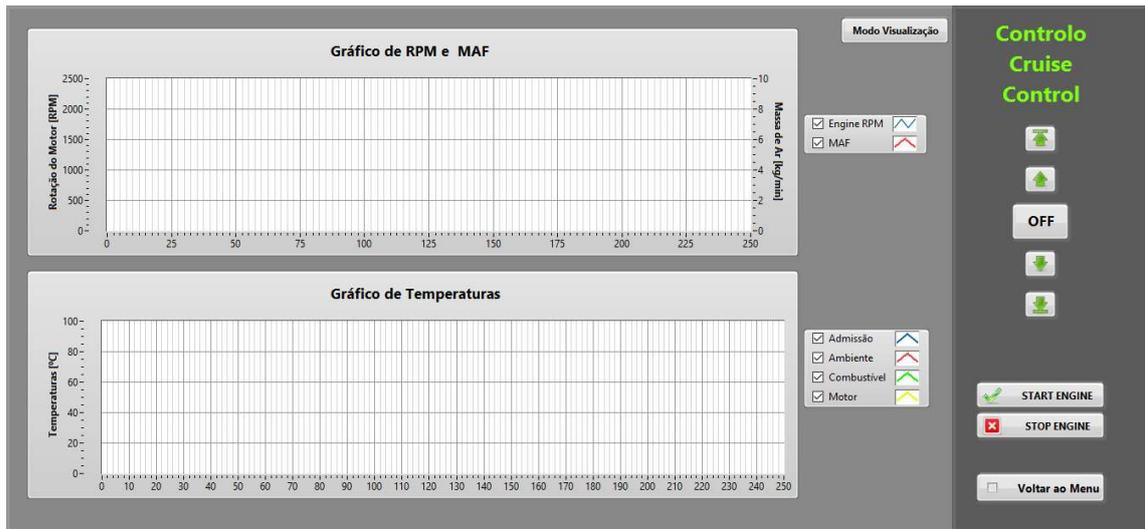


Figura 5.12 - *Layout* Visualização gráfica com controlo *cruise control*.



## 6. Testes de validação

---

Após a descrição do projeto de *hardware* e software apresenta-se os testes efetuados ao sistema com o objetivo de validar o seu funcionamento. Este capítulo tem como objetivo apresentar a descrição dos testes efetuados e os resultados obtidos.

### 6.1. Implementação do sistema

---

Nesta secção apresenta-se a implementação do EDAS num veículo pesado com o objetivo de se realizar os testes de validação. A Figura 6.1 consiste numa fotografia do sistema conectado ao veículo pesado e ao computador portátil. A conexão, ao veículo pesado, é efetuada através da ficha OBD-II, relativamente ao computador utiliza-se um conector USB. O veículo pesado utilizado é do fabricante Scania® e com o código do motor DT 12 17 480 Euro 4. Numa primeira fase necessita-se de calibrar a posição do acelerador com o intuito de não gerar erro de leitura por parte do COO.



Figura 6.1 – Sistema implementado em modo funcionamento.

## 6.2. Validação do *Hardware*

---

Dividiu-se a fase de validação em duas componentes, iniciando pela validação do *hardware* e posteriormente do *software* e suas funcionalidades. O presente subcapítulo aborda os testes efetuados ao *hardware* garantindo o correto funcionamento dos componentes selecionados durante o desenvolvimento do mesmo. De seguida descreve-se o procedimento do teste realizado ao potenciômetro digital e à comunicação CAN.

### 6.2.1. Potenciômetro Digital

---

Sendo o potenciômetro digital um componente com elevada importância para o controlo do motor é necessário garantir o correto funcionamento do mesmo. Realizou-se então o procedimento de teste que consiste na variação do valor da posição do acelerador e em simultâneo, adquire-se o valor de tensão nas duas saídas do potenciômetro. A aquisição do valor de tensão das duas saídas efetuou-se através do osciloscópio do fabricante Picotech®.

Apresenta-se na Figura 6.2 o gráfico resultante do teste efetuado ao potenciômetro digital. Este gráfico é composto pelo nível de tensão [V] dos dois sinais do potenciômetro ao longo do tempo, em segundos. Tal como se pode observar no gráfico, o nível de tensão do sinal 1 é o dobro do sinal 2 ao longo do tempo. Relativamente ao ruído no valor adquirido durante o teste, tem como origem na alimentação do potenciômetro por parte da unidade de controlo. Como a variação dos valores de tensão em ambas as saídas do potenciômetro encontra-se dentro dos limites definidos (0V a 4V) conclui-se que o potenciômetro funciona corretamente.

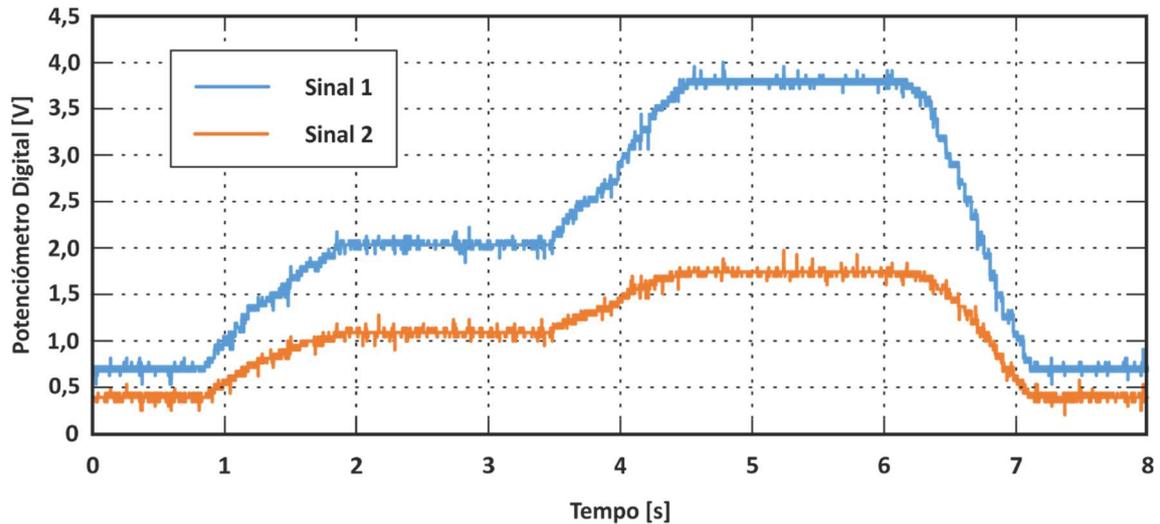


Figura 6.2 – Gráfico resultante do teste ao potenciômetro digital.

## 6.2.2. Comunicação CAN

A base do sistema desenvolvido é a comunicação CAN portanto o teste ao seu funcionamento possui uma elevada importância. Durante a realização do teste utiliza-se o CAN *analyser* que possui a capacidade de adquirir as mensagens presentes no barramento e guardar num ficheiro. O teste consiste no envio de mensagens para o barramento CAN solicitando parâmetros do motor. O emissor destas mensagens é o sistema desenvolvido, o EDAS, e o recetor é a EMS.

Utiliza-se o CAN *analyser* com o intuito de adquirir estas mensagens que posteriormente serão analisadas. Através desta análise é possível validar o correto funcionamento da comunicação CAN. A Tabela 6.1 apresenta um excerto das mensagens presentes no barramento CAN durante o teste. A tabela é composta pelo número de mensagem, o tempo em que surgiu, o seu CAN-ID e o seu campo de dados. Verifica-se que são enviadas mensagens pelo EDAS com a solicitação de parâmetros, estas mensagens encontram-se realçadas com um fundo verde na tabela. Após o envio da mensagem de solicitado, a EMS envia uma mensagem para o barramento com o valor do parâmetro solicitado, tal como se visualiza na tabela a mensagem numero 15 e 20 (fundo laranja). Após esta análise verifica-se que as mensagens estão a ser enviadas pelo EDAS e a EMS responde as solicitações efetuadas, o que valida assim a comunicação CAN.

**Tabela 6.1 – Excerto das mensagens presentes no barramento CAN durante o teste à comunicação CAN.**

Nr.	Tempo [ms]	CAN-ID	Data Field							
			D0	D1	D2	D3	D4	D5	D6	D7
10	39,3	18FFA103	C3	FF						
11	39,8	0CFF8027	0C	00	0D	FF	FF	FC	3C	00
12	41,9	18FEF200	00	00	FF	FA	FF	FF	FF	FF
13	42,5	18DA00FA	04	2F	FC	71	01	18	00	00
14	49,2	0CFE6CEE	FF	FF	FF	FF	FF	FF	00	00
15	51	18DAFA00	06	6F	FC	71	01	01	20	00
16	51,5	18DA00FA	04	2F	FB	2C	01	18	00	00
17	52,1	0CF00400	FF	A4	7D	00	00	FF	FF	FF
18	60,2	0CFF8027	0C	00	0D	FF	FF	FC	3C	00
19	60,8	18FFA027	FF	F3	00	FF	FF	FF	FF	FF
20	61,4	18DAFA00	06	6F	FB	2C	01	00	00	00
21	61,9	18DA00FA	04	2F	FB	2F	01	18	00	00
22	62,5	18F00029	F0	7D	FF	F3	FF	FF	FF	FF

## 6.3. Validação das funcionalidades do sistema

Nesta secção apresenta-se a descrição dos procedimentos de teste efetuados ao sistema com o intuito de validar o *software* e as funções implementadas. De seguida apresenta-se detalhadamente o processo de validação, os equipamentos utilizados e qual o seu esquema de ligação. As funções implementadas são as seguintes, aquisição de dados, controlo do motor, paragem de emergência.

### 6.3.1. Frequência de aquisição

Antes de se iniciar os testes às funções implementadas no *software*, é necessário validar o tempo de execução do ciclo do software. O sistema desenvolvido utiliza a comunicação série como meio de comunicação entre o LabView® e o EDAS. Definiu-se a comunicação série com os seguintes parâmetros, *baudrate* de 460800 bps, um stop bit e sem paridade. Através destas definições é possível calcular o tempo necessário para transmitir um carácter. Por cada carácter é necessário transmitir 10 bits, ou seja, com o *baudrate* definido é possível transmitir, teoricamente, 46080 caracteres por segundo.

Durante um ciclo de execução do *software*, na situação crítica, é necessário transmitir dois pedidos de parâmetros e receber dois conjuntos de parâmetros. Os pedidos de parâmetros são constituídos por 6 caracteres cada um, o que perfaz 12 caracteres. Quanto à receção,

um dos conjuntos é composto por 55 caracteres e o outro é constituído por 51. Somando todos os caracteres necessários transmitir, obtém-se o valor de 118 caracteres. Sabendo que a comunicação série implementada consegue, teoricamente, transmitir 46080 caracteres por segundo e que são necessários transmitir 118 caracteres num ciclo, é possível calcular o número de ciclos por segundo. Após este cálculo obteve-se um valor teórico de 390,5 ciclos de aquisição por segundo o que corresponde sensivelmente a uma frequência de aquisição máxima, teórica, de 390 Hz. Após o cálculo teórico da frequência máxima realizou-se teste a frequência de aquisição do *software*. O teste consistiu na variação da frequência de aquisição nos seguintes valores 10 Hz, 20 Hz, 25 Hz e 30 Hz. Durante os testes adquire-se o tempo de ciclo e guarda-se no ficheiro de dados. Com este tempo de ciclo é possível calcular a frequência real do ciclo. Após a realização dos 4 testes, analisou-se o tempo de ciclo e calculou-se a frequência de cada iteração. Com estes dados construiu-se o seguinte gráfico representado na Figura 6.3.

Como se pode verificar, para frequências definidas para valores abaixo dos 25 Hz (inclusive), o valor de cada ciclo de iteração é praticamente constante e corresponde a frequência definida. Quando valor de frequência definida é de 30 Hz o tempo de ciclo não é constante ao longo do tempo o que corresponde a uma variação na frequência do ciclo. Como se pode verificar no teste a 30 Hz, apesar do valor da frequência ser praticamente constante ao longo do tempo, valores mais baixos de frequência.

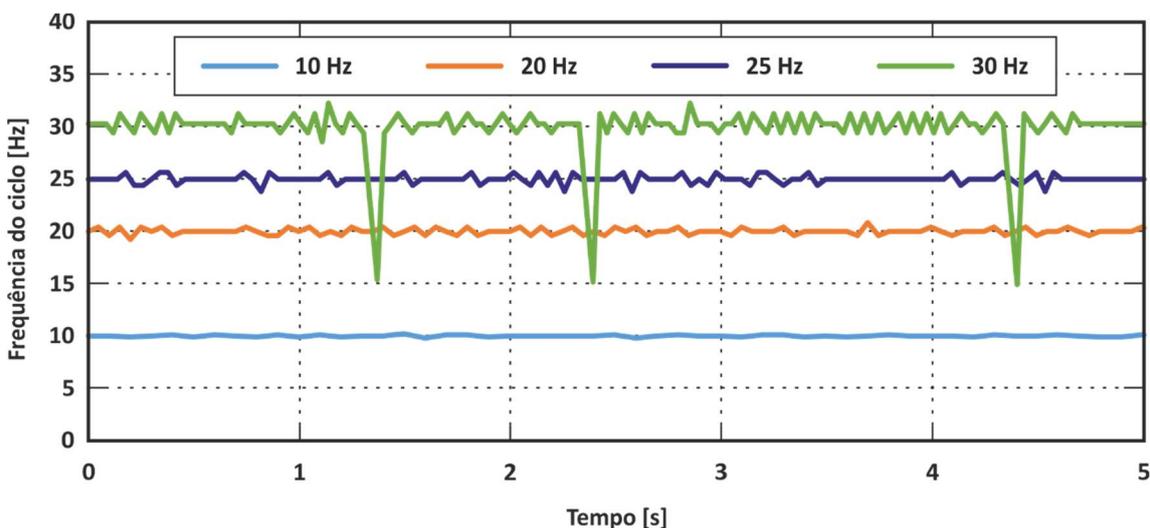


Figura 6.3 – Gráfico resultante do teste à frequência.

### 6.3.2. Aquisição de dados

Após a validação do correto funcionamento do *hardware* desenvolvido e do tempo de ciclo do software é necessário validar os parâmetros recolhidos durante um ensaio. Tendo como base de comparação o equipamento de diagnóstico da Scania®, adquire-se com os dois sistemas, em simultâneo, um conjunto de parâmetros. Posteriormente interpreta-se os dados adquiridos permitindo assim validar a aquisição de parâmetros pelo EDAS.

De seguida apresenta-se a Figura 6.4 que demonstra as ligações efetuadas e os vários equipamentos necessários ao teste à aquisição. Como se pode verificar é necessário dois computadores e um equipamento de diagnóstico. Um dos computadores conecta-se ao equipamento de diagnóstico e o outro ao EDAS.

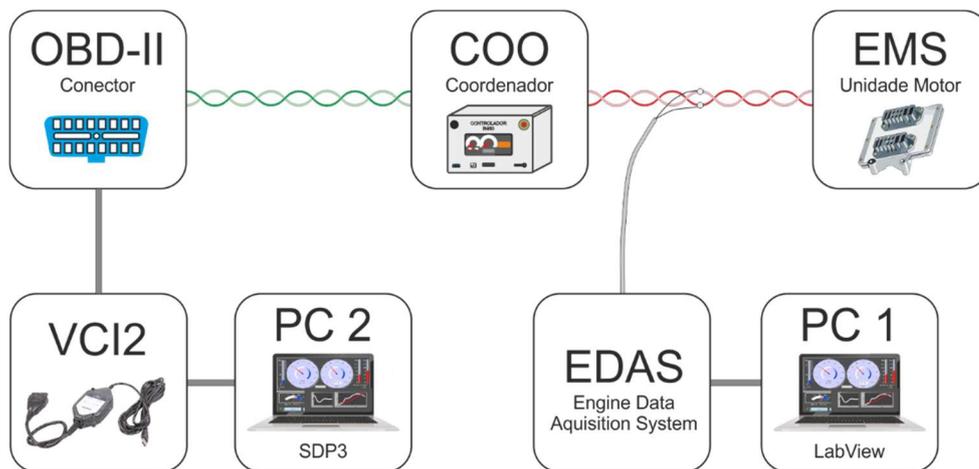


Figura 6.4 – Esquema de ligação do teste à aquisição.

Após a execução do teste é necessário analisar os dados recolhidos por ambos os equipamentos de modo a validar a aquisição do EDAS. Importa-se os dados recolhidos para uma folha de cálculo, de seguida sincroniza-se os dois ficheiros resultantes do ensaio e elabora-se os seguintes gráficos presentes na Figura 6.5. A figura contém três gráficos onde cada um deles apresenta a evolução de dois valores do mesmo parâmetro adquiridos pelos dois equipamentos ao longo do tempo. O primeiro gráfico exibe o valor da rotação do motor adquirido pelo EDAS (azul) e o SDP3 (laranja). Relativamente ao segundo gráfico exibe-se o parâmetro massa de ar admitida pelo motor adquirido também pelos dois equipamentos. Quanto ao último parâmetro, exibe-se a pressão de óleo adquirida pelos dois métodos.

Como se pode verificar os valores adquiridos por ambos os equipamentos são coerentes ao longo do tempo, validando assim a aquisição de dados do EDAS. Em certos pontos, o equipamento de diagnóstico SDP3 apresentou algumas falhas na aquisição provando assim que não se encontra otimizado para aquisição de dados. Quanto a frequência de aquisição, verifica-se que o EDAS possui uma taxa de aquisição superior ao equipamento de diagnóstico SDP3. É possível verificar a diferença da taxa de aquisição com maior detalhe na Figura 6.6 que apresenta a rotação do motor na fase de arranque. Analisando o gráfico na fase de arranque verifica-se que a aquisição efetuada pelo EDAS possui uma taxa muito superior a taxa de aquisição do equipamento de diagnóstico SDP3.

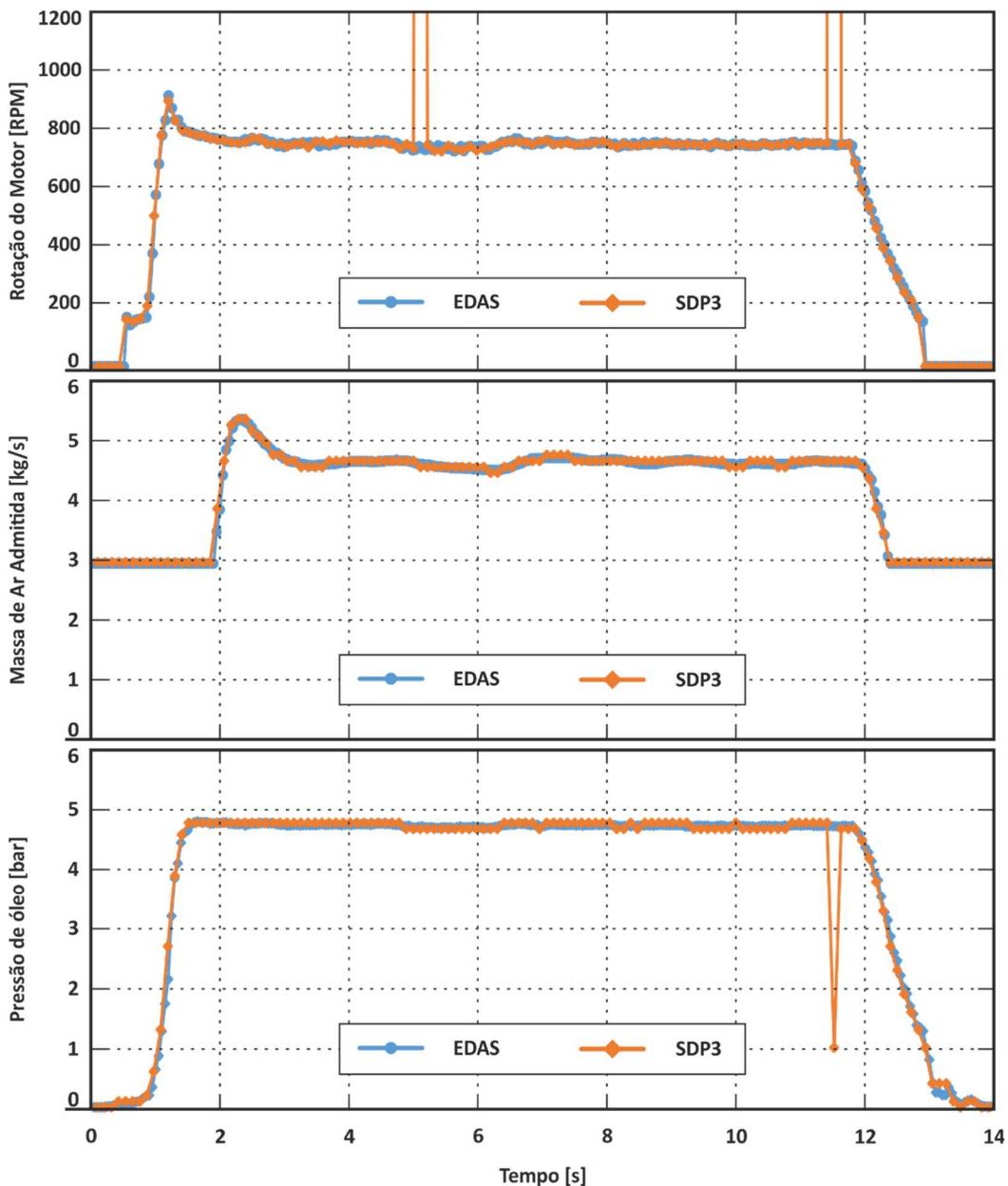


Figura 6.5 – Gráficos resultantes do ensaio da aquisição de dados.

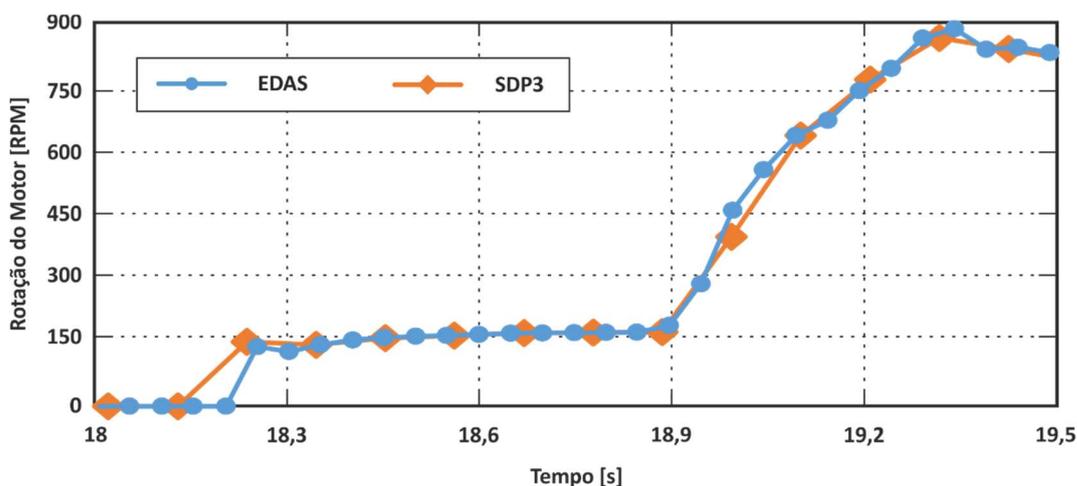


Figura 6.6 – Gráfico da rotação do motor durante a fase de arranque.

### 6.3.3. Controle do Motor

---

Após a validação do método de aquisição de dados do EDAS, procede-se com a validação do controlo do motor. O sistema permite dois tipos de controlo, posição do acelerador e *cruise control*. O método de validação do controlo consiste na aquisição de vários parâmetros e em simultâneo controlar através de cada um dos métodos. Posteriormente, elabora-se um gráfico onde se exibe a rotação ao longo do tempo em simultâneo com o valor de controlo desejado pelo utilizador.

Para o controlo da posição do acelerador o gráfico é composto pela rotação do motor e a posição do acelerador ao longo do tempo. Através da análise deste gráfico verifica-se o comportamento do motor relativamente a alteração do valor da posição do acelerador validando assim este tipo de controlo. Quanto ao método *cruise control* elabora-se um gráfico onde se exibe a rotação do motor e as ordens de controlo solicitadas pelo utilizador. Com a análise do gráfico é possível validar este método de controlo.

Relativamente ao equipamento utilizado para a validação do controlo do motor é apenas necessário recorrer ao sistema desenvolvido pois possui a capacidade de guardar os dados do ensaio, permitindo analisar os dados posteriormente.

Após a realização do teste ao controlo do motor através da posição do acelerador, elaborou-se o gráfico do valor da posição do acelerador, em percentagem, ao longo do tempo. Este gráfico apresenta-se na Figura 6.7, onde contém também a imagem dos vários

estados da posição do controlo do acelerador definido no *software*. Através da análise do gráfico e da posição definida no *software* verifica-se que nas posições de 0 % e 100% o valor lido pela unidade corresponde ao definido pelo utilizador, com isto podemos afirmar que o potenciómetro encontra-se calibrado. Quanto as posições intermédias, o valor lido pela unidade não corresponde exatamente ao definido, apresentando um pequeno desvio. Num contexto geral, o gráfico apresenta uma boa correlação entre a variação no *software* e o valor lido pela unidade validando assim este tipo de controlo do motor.

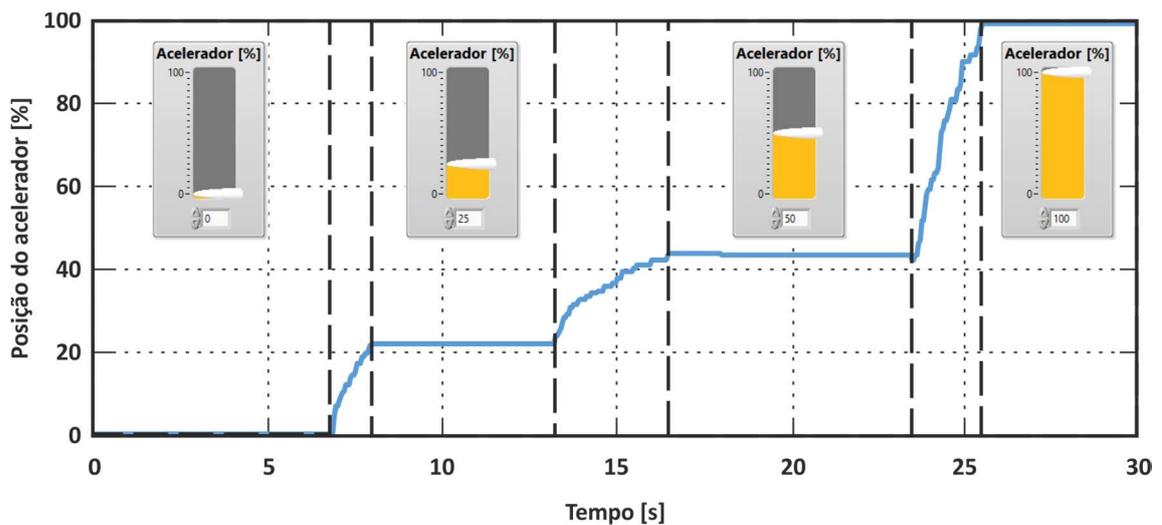


Figura 6.7 – Gráfico do teste ao controlo posição do acelerador.

Posteriormente, efetuou-se o teste ao segundo método de controlo implementado, o *cruise control*. Durante o ensaio solicitou-se dois pedidos de aceleração continuamente, cinco pedidos de desaceleração (um contínuo e quatro individuais) e por fim o pedido de parar o controlo. Após este ensaio, recorreu-se aos dados adquiridos durante o ensaio presente no ficheiro e produziu se o gráfico representado na Figura 6.8.

Através da análise deste gráfico verifica-se que o comportamento do motor corresponde aos pedidos do utilizador. Cumpriram-se todos os pedidos efetuados pelo utilizador durante o ensaio o que valida este tipo de controlo.

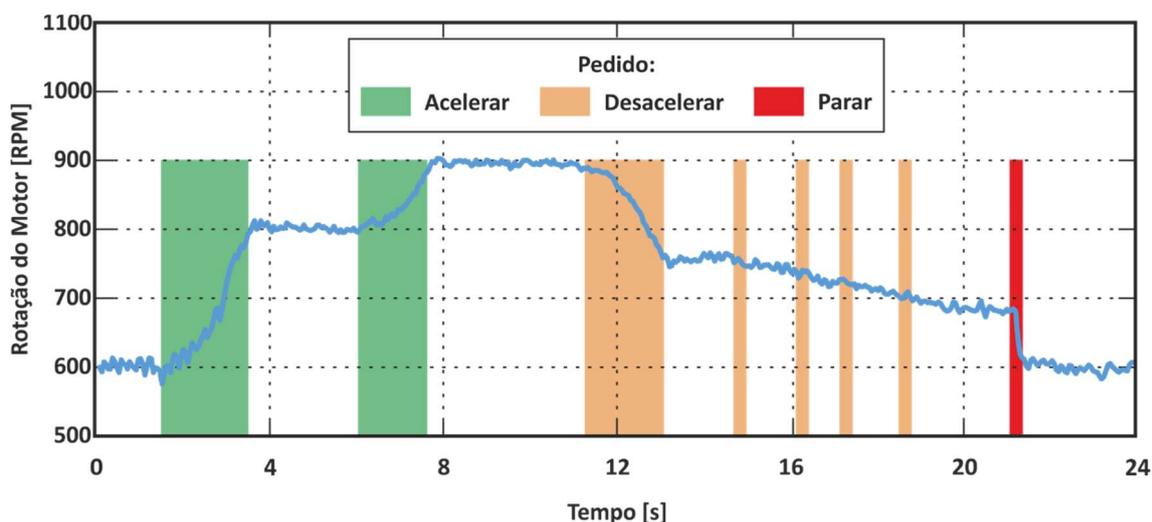


Figura 6.8 – Gráfico do teste do *cruise control*.

#### 6.3.4. Paragem de emergência

---

Uma outra funcionalidade implementada é a capacidade de parar o funcionamento do motor se algum dos parâmetros mais importantes se encontrar fora dos seus limites. Estes limites são definidos pelo utilizador permitindo assim uma maior flexibilidade. Recorrendo apenas ao sistema desenvolvido, adquire-se os dados relativamente a um ensaio onde ocorre uma paragem de emergência. Por fim elabora-se um gráfico onde se verifica qual o parâmetro que saiu dos limites definidos pelo utilizador e valida-se a função de paragem de emergência.

Efetuu-se dois testes com dois parâmetros de emergência diferentes, a Rotação do motor e a Temperatura do motor. No teste onde se limita a rotação do motor, definiu-se como limite 1000 RPM, ou seja, se o valor da rotação for superior ao limite, autonomamente o motor deve se desligar. Após este teste contruiu-se o gráfico presente na Figura 6.9 que valida assim o funcionamento desta função. A figura apenas contém a evolução da rotação do motor ao longo do tempo e o limite definido pelo utilizador.

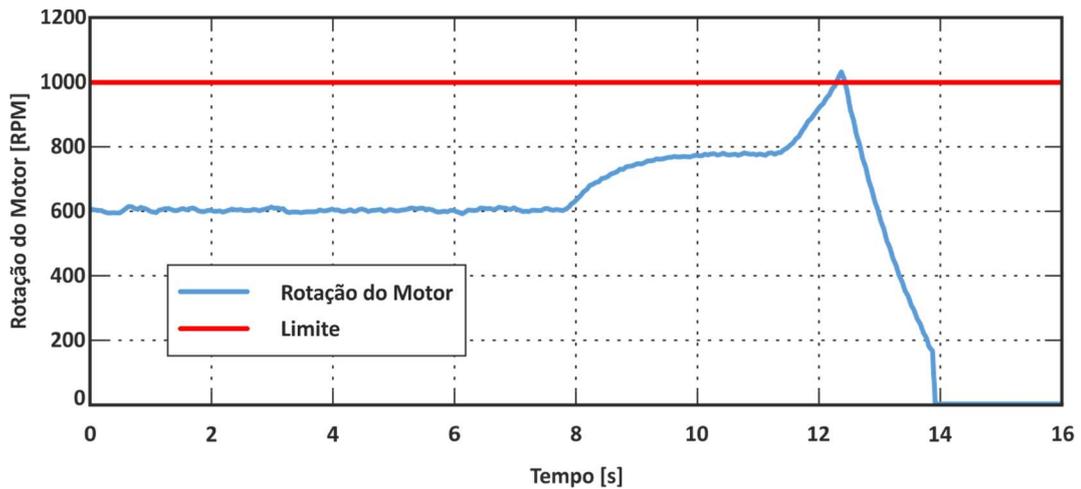


Figura 6.9 – Gráfico resultante do teste ao parâmetro de emergência Rotação do motor.

De seguida efetuou-se o teste ao parâmetro de emergência temperatura do motor onde se deixou o motor a funcionar durante alguns segundos até atingir uma temperatura superior a temperatura limite definida pelo utilizador. O valor do limite da temperatura definido é de 40° C e a temperatura do motor no início do teste possui o valor de 39° C. Posteriormente ao teste elaborou-se o gráfico apresentado na Figura 6.10 que apresenta a evolução da rotação e a temperatura do motor ao longo do tempo. Analisando o gráfico verifica-se que a partir do momento em que a temperatura do motor é superior ao limite definido pelo utilizador, o motor desliga-se autonomamente. Com isto validamos mais uma vez a função de paragem de emergência implementada no *software*.

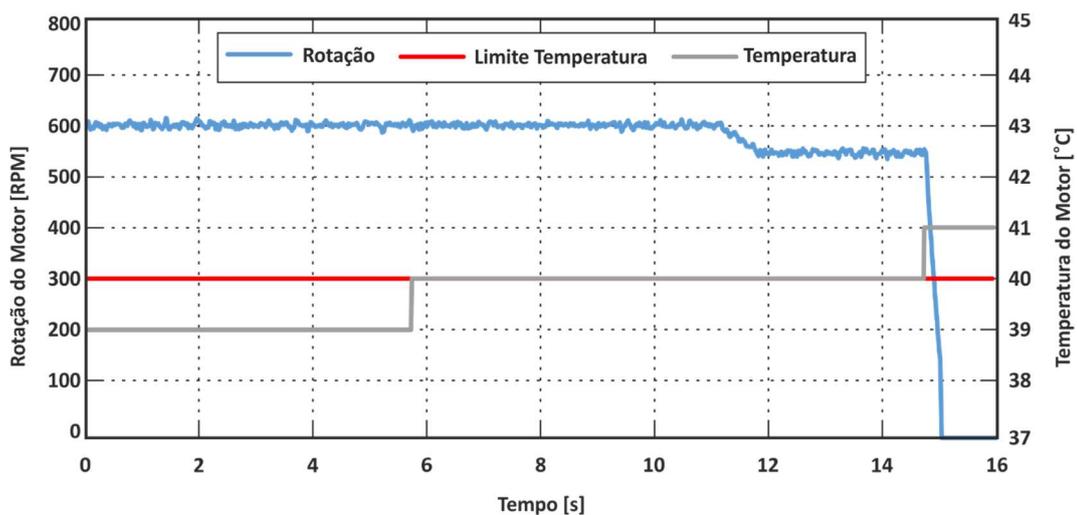


Figura 6.10 - Gráfico resultante do teste ao parâmetro de emergência Temperatura do motor.

## 6.4. Exemplo de um ensaio

---

Após a validação de todas as funcionalidades de *hardware* e *software* procedeu-se a realização de um ensaio final ao sistema. Este ensaio consistiu em adquirir durante dezoito minutos na fase de aquecimento do motor até atingir a temperatura de sensivelmente 60 °C. O objetivo do ensaio foi compreender certas correlações entre parâmetros do motor e assim posteriormente analisar estas dependências de modo a adquirir conhecimentos na área de análise de dados. Durante o ensaio utilizou-se apenas método da posição do acelerador como modo de controlo do motor.

Apresenta-se na Figura 6.11 um gráfico no tempo que exhibe a evolução da rotação e temperatura do motor ao longo ensaio. Através da análise ao gráfico verifica-se que o valor da temperatura do motor no início do ensaio era de 28 °C e, após dezoito minutos, o valor da temperatura era de 60 °C, aproximadamente. Quanto ao valor da rotação, no início do teste, com a temperatura do motor abaixo dos 40 °C, o valor de *ralenti* do motor era de 600 RPM, quando a temperatura atingiu valores superiores a 40 °C, a rotação do motor ao *ralenti* baixou para 550 RPM. Durante o ensaio acelerou-se instantaneamente em certos pontos, como por exemplo aos 4 min de ensaio, e acelerou-se de forma contínua em duas zonas temporais distintas, aos 8 minutos com a duração de um minuto e aos 13 minutos com um duração aproximadamente de quatro minutos.

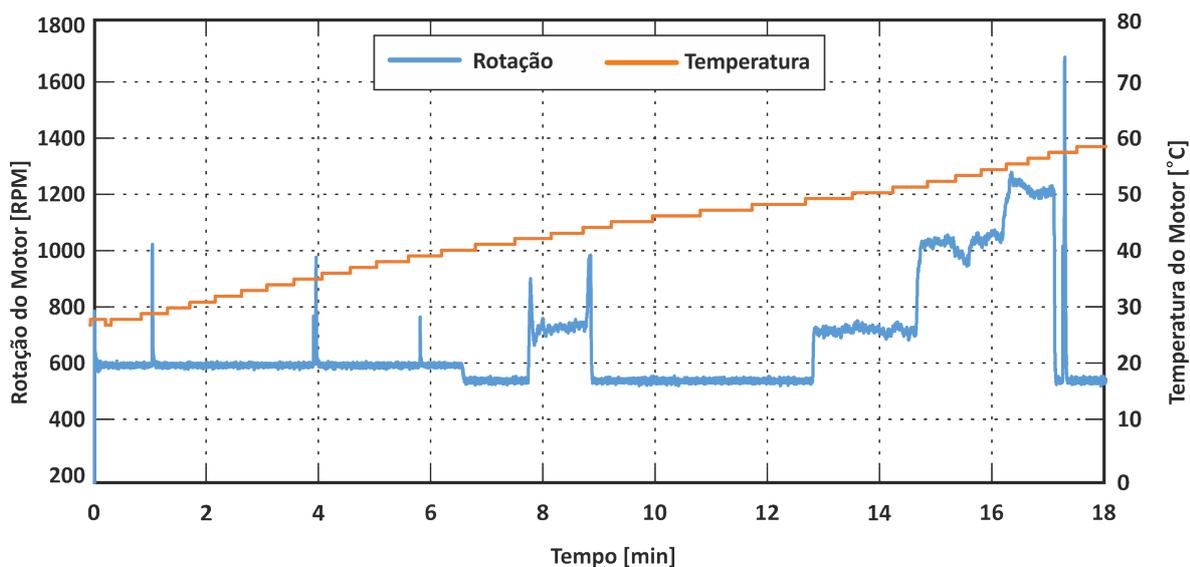


Figura 6.11 – Gráfico no tempo da rotação e temperatura do motor.

Com os dados adquiridos durante o ensaio elaborou-se um segundo gráfico no tempo com os valores da Massa de ar e do Consumo atual de combustível, exposto na Figura 6.12. Através da análise deste segundo gráfico é perceptível os momentos de aceleração instantânea do motor, pois tanto o valor da massa de ar como o consumo atual de combustível variam de forma instantânea sincronizado com as variações na rotação do motor, apresentado na Figura 6.11.

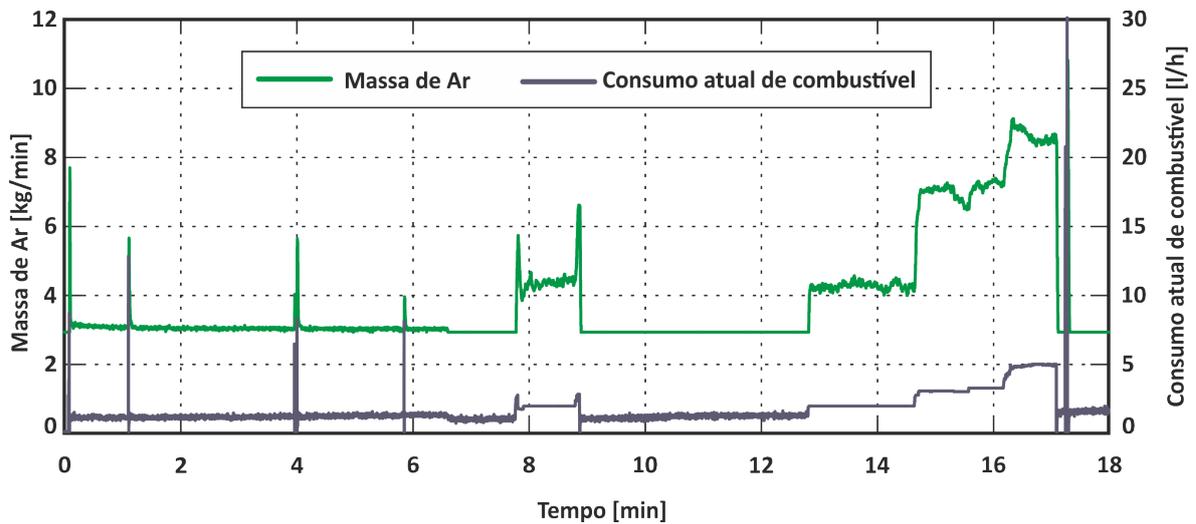


Figura 6.12 - Gráfico no tempo da Massa de Ar e Consumo atual de combustível.



## 7. Discussão de resultados

---

Neste capítulo discute-se os resultados obtidos ao longo do desenvolvimento da interface e expõe-se as conclusões retiradas após a análise dos resultados. Durante o capítulo aborda-se vários pontos do projeto, desde o *hardware* até aos métodos de controlo.

Após uma análise aos resultados obtidos durante os testes de validação verifica-se que de uma forma geral o interface desenvolvido cumpre as suas funções, sendo competente durante a aquisição de dados e controlando o motor de forma estável com qualquer um dos métodos.

### **Hardware**

Durante o desenvolvimento do *hardware* decidiu-se utilizar um microcontrolador com a capacidade de comunicação em dois barramentos CAN com o intuito de permitir vários tipos de utilização, uma aquisição de dados em modo dedicado que comunica através do barramento CAN vermelho e permite controlar o motor ou uma aquisição de dados genérica

Tal como se apresenta no capítulo do *hardware* o microcontrolador utilizado no projeto é um ChipKIT® Max32 que possui as características necessárias. Estas características traduziram-se em vantagens ao longo do desenvolvimento do interface, tais como ser uma plataforma de prototipagem que já incorpora diversos componentes necessários ao correto funcionamento. Assim o interface possui uma maior robustez a nível de *hardware* e uma maior facilidade de manutenção em caso de avaria.

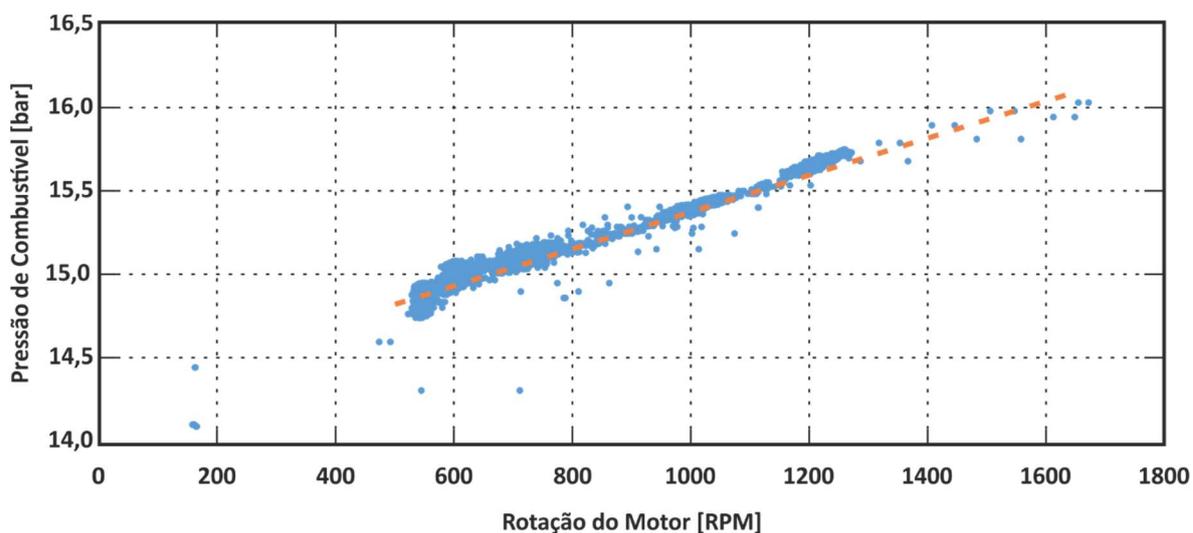
No projeto do circuito do potenciómetro digital utilizou-se dois isoladores óticos que possuem o objetivo de isolar os níveis de tensão do potenciómetro, alimentação proveniente do COO, dos níveis de tensão do microcontrolador. Com esta solução aumenta-se a fiabilidade e versatilidade deste tipo de controlo. Relativamente à fiabilidade implementou-se mais uma medida de segurança que em caso de falha de alimentação do microprocessador o potenciómetro é desativado através do pino *Shutdown*. Com esta implementação previne-se que se houver uma falha na alimentação, o valor da posição de acelerador lido pelo COO corresponde ao valor 0% e assim o valor da rotação do motor diminui até atingir o valor de *ralenti*.

## **Aquisição de dados**

Através da análise dos gráficos representados na Figura 6.5 verifica-se que os valores adquiridos pelo sistema desenvolvido são coerentes com os valores adquiridos pelo equipamento de diagnóstico, validando assim os métodos de aquisição escolhidos na fase de desenvolvimento do projeto. Verifica-se também que ao longo da aquisição ocorrem algumas falhas de aquisição de dados por parte do equipamento de diagnóstico contudo o EDAS não apresenta qualquer falha durante a aquisição. De notar que devido a otimização das frequências de aquisição dos parâmetros, através do uso maioritário do método de aquisição Broadcast e apenas seis parâmetros em modo diagnóstico SDP3, culmina numa aquisição com uma taxa elevada e sem erros. O método de aquisição do equipamento de diagnóstico é o diagnóstico SDP3 que comunica através de mensagens do tipo pergunta-resposta (peer-to-peer) tornando a aquisição mais lenta.

O gráfico apresentado na Figura 6.6 tem como objetivo exibir a diferença da frequência de aquisição de ambos os sistemas e conseqüentemente os resultados obtidos. Através da análise deste gráfico verifica-se o EDAS adquiriu 30 pontos e o equipamento de diagnóstico adquiriu apenas 15 pontos num intervalo de tempo de 1,5 segundos. Portanto é possível afirmar que o EDAS possui o dobro da frequência de aquisição do equipamento de diagnóstico. Relativamente aos valores obtidos na fase de arranque do motor, com o EDAS visualiza-se alguma oscilação do valor da rotação que não é visível com o equipamento de diagnóstico demonstrando assim a importância da frequência de aquisição.

Posteriormente à aquisição de dados durante um ensaio é possível analisar as correlações entre vários parâmetros do motor. Estas correlações possuem uma importância elevada na análise da condição do motor tais como a evolução da Pressão do Combustível com o aumento da Rotação do Motor. Apresenta-se na Figura 7.1 um gráfico que demonstra precisamente essa evolução, analisando-o verifica-se que os valores adquiridos num ensaio mostram que o valor da pressão de combustível é diretamente proporcional a rotação do motor. Adicionou-se uma linha de tendência (linha tracejada a laranja) e verifica-se que com a pressão de combustível aumenta 1 Bar por cada aumento de rotação do motor de 1000 RPM.



**Figura 7.1 – Gráfico de dispersão relacionando a Pressão de Combustível e a Rotação do Motor.**

### **Protocolo de Comunicação**

Com a implementação do protocolo dedicado entre o EDAS e o PC aumentou-se a fiabilidade de comunicação pois as mensagens enviadas são constituídas tanto pelo valor do parâmetro como a identificação a que parâmetro corresponde. Assim é possível identificar o parâmetro na mensagem e logo de seguida o valor correspondente reduzindo os erros de comunicação e consequentemente aumentando a fiabilidade da comunicação. Contudo esta identificação acrescenta um conjunto de caracteres na mensagem o que aumenta o tempo de envio e consequentemente diminui a frequência de comunicação. Futuramente pode-se simplificar ou até mesmo retirar a identificação dos parâmetros nas mensagens aumento a frequência da comunicação com o risco de aumentar o número de falhas na comunicação entre o EDAS e o PC.

### **Controlo do motor**

Durante o desenvolvimento do projeto estudaram-se vários métodos de controlo contudo no final utilizaram-se apenas dois métodos de controlo, um potenciômetro digital para a posição do acelerador e o método *cruise control*. Um dos métodos estudados consistia no envio de um novo valor da posição do acelerador sobrepondo ao valor enviado ciclicamente pelo COO. Apesar de este método apresentar algumas vantagens, tais como rapidez de

atuação e versatilidade, revelou-se ser pouco eficaz na sua função, não controlando inteiramente o valor da posição do acelerador lido pela unidade do motor (ver Figura 3.9).

O método *cruise control* tem como base a sobreposição de mensagem CAN relativas aos pedidos de aceleração, desaceleração e paragem do *cruise control*. Contudo contrariamente ao envio do valor da posição do acelerador, a sobreposição de pedidos relativos ao *cruise control*, não interfere com a estabilidade do sistema revelando assim ser um método de controlo do motor eficaz (ver Figura 6.8).

## 8. Conclusões e Trabalhos Futuros

---

O objetivo do presente trabalho foi desenvolver uma interface capaz de comunicar com um veículo pesado que permita adquirir informação dos sensores e controlar o motor. Esta informação deve ser visualizada e o motor controlado através de um *software* dedicado ao *hardware* desenvolvido. Com esse objetivo em mente efetuou-se uma revisão bibliográfica para recolher informações relativas aos componentes presentes num veículo pesados e perceber o protocolo de comunicação que o veículo pesado utiliza.

Em relação à comunicação nos veículos pesados, foi necessário proceder a descodificação das várias normas utilizadas e assim otimizar o processo de aquisição de dados. Posteriormente estudaram-se vários métodos de controlo do motor decidindo-se implementar dois métodos de controlo respetivamente o uso de um potenciómetro digital e da função *cruise control*.

Após a definição dos métodos de aquisição e controlo foi projetado e desenvolvido um *hardware* capaz de os integrar numa só placa. Deste modo procedeu-se à escolha dos componentes necessários, seguida pelo desenvolvimento da placa de circuito impresso e dos cabos de ligação. Durante o desenvolvimento do *hardware* sentiu-se a necessidade de definir um protocolo de comunicação dedicado entre o hardware e o PC. Este protocolo define várias fases necessárias que oferecem uma proteção extra na comunicação relativamente ao acesso a dados por via de terceiros.

Desenvolveu-se um *software* de visualização e controlo através da ferramenta LabVIEW® que incorpora várias funções de visualização e controlo. Todas estas funções são visualizadas num ambiente gráfico “*user friendly*” projetado com o objetivo de verificar o estado de um motor de um veículo pesado.

Posteriormente a todo o desenvolvimento, foram efetuados testes de validação tanto ao interface como ao *software* desenvolvido. Os resultados destes testes demonstram o correto funcionamento das funções de aquisição e um controlo estável do motor.

Com a realização deste projeto concebeu-se um equipamento no mercado capaz de analisar a condição de um motor de um veículo pesado, contribuindo assim para a área da manutenção de veículos pesados, otimizando o seu processo. Este equipamento poderá ser utilizado futuramente num banco de ensaio de motores de veículos pesados.

## 8.1. Trabalhos Futuros

---

Sugestões de trabalhos futuros:

- Integrar novos protocolos de comunicação utilizados em veículos;
- Otimizar o protocolo de comunicação dedicado tendo em vista o aumento da frequência de envio;
- Aumento da segurança em caso de falha de *hardware*;
- Otimizar o ciclo do *software* de visualização e controlo;
- Criar uma base de dados relativa a valores típicos dos parâmetros do motor;
- Desenvolver um *software* que permita a visualização e análise *á posteriori* de resultados obtidos em ensaios de motores guardados em ficheiros de dados.

## 9. Bibliografia

---

- [1] Eurostat, *Energy, transport and environment indicators*. 2014.
- [2] Scania, “Informações do produto - Relatório Scania Diagnos & Programmer 3.” 2015.
- [3] Scania, “Motor DT12 17 R480 Euro 4.” [Online]. Available: [http://www4.scania.com/templates/campaigns/jack/img/truck/standards\\_6cyl\\_egr.png](http://www4.scania.com/templates/campaigns/jack/img/truck/standards_6cyl_egr.png). [Accessed: 15-Jan-2016].
- [4] Scania, “Scania Web Page,” 2015. [Online]. Available: [www.scania.com](http://www.scania.com). [Accessed: 21-Dec-2015].
- [5] Scania, “Problemas de comunicação,” 2013.
- [6] Scania, “COO Schematic.” .
- [7] Scania, “EMS S6 Schematic.” .
- [8] Scania, “EMS S6 HPI Schematic.” .
- [9] SAE, “SAE J1939,” 2003.
- [10] SAE, “SAE J1939-11,” 1994.
- [11] SAE, “SAE J1939-71,” 2006.
- [12] M. Junger, “Introduction to J1939,” *Vector*, p. 11, 2010.
- [13] Kvaser, “J1939 Introduction.” [Online]. Available: <https://www.kvaser.com/about-can/higher-layer-protocols/j1939-introduction/>. [Accessed: 28-Dec-2015].
- [14] SAE, “SAE J1979,” 2006.
- [15] H. Fonseca, C. Ferreira, and T. Fernandes, “New Methodologies To Measure in Real Time Fuel Consumption of Internal Combustion Engines,” *15th Int. Conf. Exp. Mech.*, no. June, 2012.
- [16] S. E. Marx, “Controller Area Network (CAN) Bus J1939 Data Acquisition Methods and Parameter Accuracy Assessment Using Nebraska Tractor Test Laboratory Data,” University of Nebraska-Lincoln, 2015.
- [17] O. Kiffer, “Troubleshooting Scania Vehicles , Marine and Industrial Engines with External Sensors Industrial Engines with External Sensors,” Uppsala University, 2015.
- [18] K. Smith and J. Miller, “OBDII data logger design for large-scale deployments,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, 2013.
- [19] L. M. V. S. M. C. Gameiro da Silva, M. L. O. .S. Mateus, “Automatic Data Acquisition and Control of an Engine Test Bench.”
- [20] P. Systems, “PCAN-USB.” [Online]. Available: <http://www.peak-system.com/>. [Accessed: 30-Aug-2016].

- [21] PicoTech, "PicoScope 2205A." [Online]. Available: [www.picotech.com](http://www.picotech.com). [Accessed: 15-May-2016].
- [22] Efichip, "Scania Diagnos & Programmer." [Online]. Available: <http://efichip.com/>. [Accessed: 15-May-2016].

## 10. Anexos

### 10.1. Mensagem de Torque/Speed Control

Tabela 10.1 - Descrição da descodificação dos parâmetros presentes na no PGN 0x0000[11].

SPN	Parameter Name	Decoding		
		Resolution	Offset	Description
695	Engine Override Control Mode	4 states	0	00 – Override disable 01 – Speed Control 10 – Torque Control 11 – Speed/Torque limit control
696	Engine Requested Speed Control Conditions	4 states	0	00 – Transient optimized, driveline disengaged 01 – Stability optimized, driveline disengaged 10 – Stability optimized, driveline engaged, condition 1 11 – Stability optimized, driveline engaged, condition 2
897	Engine Requested Speed Control Conditions	4 states	0	00 Highest priority 01 High priority 10 Medium priority 11 Low priority
898	Engine Requested Speed/Speed Limit	0,125 Rpm/bit	0	RPM value = 0,125 x Data (dec)
518	Engine Requested Torque/Torque Limit	1 %/bit	-125 %	% value = 1 x Data (dec) - 125
3349	TSC1 Transmission Rate	8 states	0	000 – 1000 ms 001 – 750 ms 010 – 500 ms 011 – 250 ms 100 – 100 ms 101 – 50 ms 110 – 20 ms 111 – 10 ms
3350	TSC1 Control Purpose	32 states	0	00000 – Accelerator Pedal 00001 – Cruise Control 00010 – PTO Governor 00011 – Road Speed Governor 00100 – Engine Protection 00101/11110 – Reserved for SAE 11111 – Temporary Power Train Control

## 10.2. Parâmetros disponíveis

Tabela 10.2 - Campo de dados da mensagem de request através do SDP3.

Nome da variável	Unidades	Campo dados							
		D0	D1	D2	D3	D4	D5	D6	D7
Fluxo de ar	kg/min	04	2F	FC	71	01	18	00	00
Fluxo de ar, tensão	V	04	2F	FC	72	01	18	00	00
Sensor de fluxo de massa		04	2F	F4	06	01	18	00	00
Conteúdo da EGR, valor atual	%	03	22	FC	49	01	18	00	00
Conteúdo da EGR, valor esperado	%	03	22	FC	50	03	18	00	00
Válvula proporcional para válvula EGR	%	04	2F	F4	04	01	18	00	00
Válvula de ligação/desligamento para venturi		04	2F	F4	05	01	18	00	00
Válvula proporcional para válvula de comporta do sobrealimentador	%	04	2F	F4	09	01	18	00	00
Válvula de corte de combustível	%	04	2F	F4	0C	01	18	00	00
Válvula de corte de combustível		04	2F	FB	F3	01	18	00	00
Temperatura do combustível	°C	04	2F	FB	FA	01	18	00	00
Temperatura do combustível	V	04	2F	FC	3A	01	18	00	00
Pressão de alimentação do combustível	bar	04	2F	FB	2F	01	18	00	00
Pressão de alimentação do combustível	V	04	2F	FB	D9	01	18	00	00
Sensor de NOx	ppm	04	2F	FE	9F	01	18	00	00
Rotação do motor	rpm	04	2F	FB	69	01	B1	00	00
Velocidade do veículo	km/h	04	2F	FB	5E	01	18	00	00
Pressão de óleo	bar	04	2F	FB	2C	01	18	00	00
Sensor de pressão de óleo, valor em tensão	V	04	2F	FB	D6	01	18	00	00
Temperatura do líquido de arrefecimento	°C	04	2F	FB	32	01	18	00	00
sensor de temperatura do líquido de arrefecimento, valor da tensão	V	04	2F	FB	DC	01	18	00	00
Temperatura do ar de admissão	°C	04	2F	FB	30	01	18	00	00
Sensor de temperatura do ar de admissão, valor em tensão	V	04	2F	FB	DA	01	18	00	00
Pressão de ar de admissão	bar	04	2F	FB	2A	01	18	00	00
Sensor de pressão do radiador de ar, valor em tensão	V	04	2F	FB	D4	01	18	00	00
Motor de partida	%	04	2F	FB	03	01	18	00	00
Bobina de acoplamento para compressor do AC		04	2F	F4	02	01	18	00	00
Válvula proporcional para freio-motor	%	04	2F	F4	0A	01	18	00	00
Atuação do alternador		04	2F	F4	01	01	18	00	00
Controle do ventilador	%	04	2F	F4	07	01	18	00	00
Velocidade de rotação do ventilador	rpm	04	2F	FF	4C	01	18	00	00
Chave de partida		04	2F	FB	3C	01	18	00	00
Tensão da bateria	V	04	2F	FB	34	01	18	00	00
Pressão atmosférica	bar	03	22	FB	3E	01	18	00	00
Temperatura do ar	°C	03	22	FB	5B	01	18	00	00
Cilindro 1	mg/curso	03	22	FB	BA	01	18	00	00

Cilindro 2	mg/curso	03	22	FB	BB	01	18	00	00
Cilindro 3	mg/curso	03	22	FB	BC	01	18	00	00
Cilindro 4	mg/curso	03	22	FB	BD	01	18	00	00
Cilindro 5	mg/curso	03	22	FB	BE	01	18	00	00
Cilindro 6	mg/curso	03	22	FB	BF	01	18	00	00
Cilindro 7	mg/curso	03	22	FB	C0	01	18	00	00
Cilindro 8	mg/curso	03	22	FB	C1	01	18	00	00
Cilindro 1	ms	04	2F	FB	76	01	18	00	00
Cilindro 2	ms	04	2F	FB	77	01	18	00	00
Cilindro 3	ms	04	2F	FB	78	01	18	00	00
Cilindro 4	ms	04	2F	FB	79	01	18	00	00
Cilindro 5	ms	04	2F	FB	7A	01	18	00	00
Cilindro 6	ms	04	2F	FB	7B	01	18	00	00
Volume de combustível	mg/curso	03	22	FB	AC	03	18	00	00
Pedido de velocidade do ventilador pelo retarder	%	03	22	FB	40	03	18	00	00
Capacidade de arrefecimento solicitada	%	03	22	FD	AD	03	18	00	00
Controle do ventilador - o que controla		04	2F	FD	AE	03	18	00	00
Limitando o volume de combustível		03	22	FD	93	03	18	00	00
Comando do motor		03	22	FB	AE	03	18	00	00
Desvio da pressão de combustível	bar	03	22	FB	99	03	18	00	00
Modo de desvalidação		04	2F	F4	3C	01	18	00	00
Escolha de marchas		04	2F	FC	7F	01	18	00	00
Pressão de ar antes da restrição	bar	04	2F	FE	92	01	18	00	00
Temperatura no módulo da bomba do sistema SCR	°C	04	2F	FE	95	01	18	00	00
Nível do tanque	l	04	2F	FE	97	01	18	00	00
Status do sistema SCR		04	2F	FE	99	01	18	00	00
Status anterior no sistema SCR		04	2F	FE	9A	01	18	00	00
Temperatura dos gases de escape antes do catalisador	°C	04	2F	FE	9C	01	18	00	00
Redutor dosado	g/min	04	2F	FE	9E	01	18	00	00
Pressão do redutor	bar	04	2F	FE	93	01	18	00	00
Balanceamento do cilindro		04	2F	F4	12	01	18	00	00
Solicitação de redutor	g/min	04	2F	F4	2C	01	18	00	00
Posição do pedal do acelerador	%	04	2F	FB	3F	01	18	00	00
Volume máximo de combustível	mg/curso	03	22	FE	23	03	18	00	00
Pontos de interrupção	rpm	03	22	FE	24	03	18	00	00
Fator de anti cristalização		04	2F	FF	9C	01	18	00	00
Dosagem de redutor com fator de compensação		04	2F	FF	9D	01	18	00	00
Velocidade de rotação do ventilador	rpm	04	2F	FF	40	01	18	00	00
Horas totais de funcionamento do motor	h	03	22	FB	21	01	18	00	00

Tabela 10.3 – Parâmetros disponíveis nos vários métodos de aquisição.

Nome da variável	Unidades	SDP3	OBD-II	Red bus	Green bus
Carga do motor	%		X	X	X
Driver's demand engine - percent torque	%		X	X	X
Engine reference torque	Nm		X	X	X
Fluxo de ar	kg/min	X	X		
Fluxo de ar, tensão	V	X			
Sensor de fluxo de massa		X			
Conteúdo da EGR, valor atual	%	X	X		
Conteúdo da EGR, valor esperado	%	X	X		
Válvula proporcional para válvula EGR	%	X			
Válvula de ligação/desligamento para venturi		X			
Válvula proporcional para válvula de comporta do sobrealimentador	%	X			
Válvula de corte de combustível	%	X			
Válvula de corte de combustível		X			
Temperatura do combustível	°C	X		X	X
Temperatura do combustível	V	X			
Pressão de alimentação do combustível	bar	X	X	X	X
Pressão de alimentação do combustível	V	X			
Sensor de NOx	ppm	X			
Rotação do motor	rpm	X	X	X	X
Velocidade do veículo	km/h	X	X	X	X
Pressão de óleo	bar	X		X	X
Sensor de pressão de óleo, valor em tensão	V	X			
Temperatura do líquido de arrefecimento	°C	X	X	X	X
sensor de temperatura do líquido de arrefecimento, valor da tensão	V	X			
Temperatura do ar de admissão	°C	X	X	X	X
Sensor de temperatura do ar de admissão, valor em tensão	V	X			
Pressão de ar de admissão	bar	X	X	X	X
Sensor de pressão do radiador de ar, valor em tensão	V	X			
Motor de partida	%	X			
Bobina de acoplamento para compressor do AC		X			
Válvula proporcional para freio-motor	%	X			
Atuação do alternador		X			
Controle do ventilador	%	X			
Velocidade de rotação do ventilador	rpm	X			

Chave de partida		X			
Tensão da bateria	V	X			
Pressão atmosférica	bar	X	X	X	X
Temperatura do ar	°C	X	X	X	X
Cilindro 1	mg/curso	X			
Cilindro 2	mg/curso	X			
Cilindro 3	mg/curso	X			
Cilindro 4	mg/curso	X			
Cilindro 5	mg/curso	X			
Cilindro 6	mg/curso	X			
Cilindro 7	mg/curso	X			
Cilindro 8	mg/curso	X			
Cilindro 1	ms	X			
Cilindro 2	ms	X			
Cilindro 3	ms	X			
Cilindro 4	ms	X			
Cilindro 5	ms	X			
Cilindro 6	ms	X			
Volume de combustível	mg/curso	X			
Pedido de velocidade do ventilador pelo retarder	%	X			
Capacidade de arrefecimento solicitada	%	X			
Controle do ventilador - o que controla		X			
Limitando o volume de combustível		X			
Comando do motor		X			
Desvio da pressão de combustível	bar	X			
Modo de desvalidação		X			
Escolha de marchas		X			
Pressão de ar antes da restrição	bar	X			
Temperatura no módulo da bomba do sistema SCR	°C	X			
Nível do tanque	l	X	X	X	X
Status do sistema SCR		X			
Status anterior no sistema SCR		X			
Temperatura dos gases de escape antes do catalisador	°C	X			
Redutor dosado	g/min	X			
Pressão do redutor	bar	X			
Balanceamento do cilindro		X			
Solicitação de redutor	g/min	X			
Posição do pedal do acelerador	%	X	X	X	X
Volume máximo de combustível	mg/curso	X		X	
Pontos de interrupção	rpm	X		X	
Fator de anticristalização		X			
Dosagem de redutor com fator de compensação		X			

Velocidade de rotação do ventilador	rpm	X		X	X
Consumo combustível total do motor	l			X	X
Consumo combustível atual	l/h		X	X	X
Consumo combustível atual (km/h)	km/h			X	X
Número de chassi (VIN)				X	X
Horas totais de funcionamento do motor	h	X	X	X	X

### **10.3. Decodificação do Protocolo SAE J1939**

---

<b>PGN</b>	0xF003	<b>Name</b>	Electronic Engine Controller 2 (EEC2)
Identifies electronic engine control related parameters.			

<b>Transmission Repetition Rate [ms]</b>	50
<b>Data length</b>	8 bytes
<b>Extended Data Page (R) :</b>	0
<b>Data page</b>	0
<b>PDU format</b>	0xF0
<b>PDU specific</b>	0x03
<b>Default priority</b>	3
<b>Parameter Group Number</b>	0xF003
<b>Source Address</b>	0x00 (Engine)

CAN-ID	Data				Decoding							
	Start Posit	Length	Parameter Name	SPN	Bin	Hex	Dec	Resolution	Offset	Real value	Units	
Data Field	1.1	2 bits	Accelerator Pedal 1 Low Idle Switch	558	00000000	00	0	4	states	0	Accelerator pedal 1 not in low idle condition	-
	1.3	2 bits	Accelerator Pedal Kickdown Switch	559	00000011	03	3	4	states	0	Not available	-
	1.5	2 bits	Road Speed Limit Status	1437	00000011	03	3	4	states	0	Not available	-
	1.7	2 bits	Accelerator Pedal 2 Low Idle Switch	2970	00000011	03	3	4	states	0	Not available	-
	2	1 byte	Accelerator Pedal Position 1	91	00101000	28	40	0,4	%/bit	0	16	%
	3	1 byte	Engine Percent Load At Current Speed	92	00000000	00	0	1	%/bit	0	0	%
	4	1 byte	Remote Accelerator Pedal Position	974	11111111	FF	255	0,4	%/bit	0	Not available	%
	5	1 byte	Accelerator Pedal Position 2	29	11111111	FF	255	0,4	%/bit	0	Not available	%
6.1	2 bits	Vehicle Acceleration Rate Limit Status	2979	00000000	00	0	4	states	0	Limit not active	-	
7	1 byte	Actual Maximum Available Engine - Percent Torque	3357	11111111	FF	255	0,4	%/bit	0	Not available	%	

<b>PGN</b>	0xF004	<b>Name</b>	Electronic Engine Controller 1 (EEC1)
Engine related parameters			

<b>Transmission Repetition Rate [ms]</b>	engine speed dependent
<b>Data length</b>	8 bytes
<b>Extended Data Page (R) :</b>	0
<b>Data page</b>	0
<b>PDU format</b>	0xF0
<b>PDU specific</b>	0x04
<b>Default priority</b>	3
<b>Parameter Group Number</b>	0xF004
<b>Source Address</b>	0x00 (Engine)

CAN-ID

Data Field

Start Posit	Length	Parameter Name	SPN	Data			Decoding				
				Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1.1	4 bits	Engine Torque Mode	899	00001111	0F	15	16	states	0	Not available	-
2	1 byte	Driver's Demand Engine - Percent Torque	512	10100100	A4	164	1	%/bit	-125	39	%
3	1 byte	Actual Engine - Percent Torque	513	01111101	7D	125	1	%/bit	-125	0	%
4-5	2 byte	Engine Speed	190	--	0000	0	0,125	rpm/bit	0	0	RPM
6	1 byte	Source Address of Controlling Device for Engine Control	1483	11111111	FF	255	Source Adress		0	Not available	--
7.1	4 bits	Engine Starter Mode	1675	00001111	0F	15	16	states	0	Not available	--
8	1 byte	Engine Demand – Percent Torque	2432	11111111	FF	255	1	%/bit	-125	Not available	%

PGN	0xFE6C	Name	Tachograph (TCO1)
-----	--------	------	-------------------

Transmission Repetition Rate [ms]	50
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0x6C
Default priority	3
Parameter Group Number	0xFE6C
Source Address	0xEE (Tachograph)

CAN-ID

Data Field

Start Posit	Length	Parameter Name	SPN	Data			Decoding				
				Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1.1	3 bits	Driver 1 working state	1612	00000111	07	7	8	states	0	Not available	-
1.4	3 bits	Driver 2 working state	1613	00000111	07	7	8	states	0	Not available	-
1.7	2 bits	Vehicle motion	1611	00000011	03	3	4	states	0	Not available	-
2.1	4 bits	Driver 1 Time Related States	1617	00001111	0F	15	16	states	0	Not available	-
2.5	2 bits	Driver card, driver 1	1615	00000011	03	3	4	states	0	Not available	-
2.7	2 bits	Vehicle Overspeed	1614	00000011	03	3	4	states	0	Not available	-
3.1	4 bits	Driver 2 Time Related States	1618	00001111	0F	15	16	states	0	Not available	-
3.5	2 bits	Driver card, driver 2	1616	00000011	03	3	4	states	0	Not available	-
4.1	2 bits	System event	1622	00000011	03	3	4	states	0	Not available	-
4.3	2 bits	Handling information	1621	00000011	03	3	4	states	0	Not available	-
4.5	2 bits	Tachograph performance	1620	00000011	03	3	4	states	0	Not available	-
4.7	2 bits	Direction indicator	1619	00000011	03	3	4	states	0	Not available	-
5-6	2 bytes	Tachograph output shaft speed	1623	--	FFFF	65535	0,125	rpm/bit	0	Not available	RPM
7-8	2 bytes	Tachograph vehicle speed	1624	--	0000	0	0,0039	km/h per bit	0	0	km/h

PGN	0xFF80	Name	DLN1 - Proprietary, all-speed engines
Scania PGN - Message sent to Engine (DA - 00)			

Transmission Repetition Rate [ms]	20
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0x80
Default priority	3
Parameter Group Number	0xFF80
Source Address	0x27 (Coordinator)

CAN-ID

Start Posit	Length	Parameter Name	SPN	Data			Decoding			
				Bin	Hex	Dec	Resolution	Offset	Real value	Units
1	1 byte	Version Of DLN1-Messages Configuration	--	00001100	0C	12	--	0	-	-
2	1 byte	Accelerator Pedal Position	--	00101000	28	40	0,4 %/bit	0	16	%
3.1	2 bits	AP Low Idle Switch Released	--	00000000	00	0	4 states	0	Throttle pedal not released	-
3.3	2 bits	AP kickdown switch	--	00000011	03	3	4 states	0	Not available	-
3.5	2 bits	Engine Start	--	00000000	00	0	4 states	0	OFF	-
3.7	2 bits	Engine Emergency Stop Demand	--	00000000	00	0	4 states	0	Engine Stop Not demanded	-
4.1	2 bits	Engine Stop	--	00000011	03	3	4 states	0	Not available	-
4.5	2 bits	Retarder Speed Control Offswitch	--	00000011	03	3	4 states	0	Not available	-
4.7	2 bits	Retarder Speed Control Setswitch	--	00000011	03	3	4 states	0	Not available	-
5.5	2 bits	Lamp test	--	00000011	03	3	4 states	0	Not available	-
5.7	2 bits	CC-Off	--	00000011	03	3	4 states	0	Not available	-
6.1	2 bits	Increased Idle Speed switch 1 (ISSW1)	--	00000000	00	0	4 states	0	ISSW1 not demanded	-
6.3	2 bits	Increased Idle Speed switch 2 (ISSW2)	--	00000011	0003	3	4 states	0	Not available	-
6.5	2 bits	Torque Limit 1	--	00000011	0003	3	4 states	0	Not available	-
6.7	2 bits	Torque Limit 2	--	00000011	0003	3	4 states	0	Not available	-
7.1	2 bits	Exhaust brake floor switch	--	00000000	0000	0	4 states	0	Switch not activated	-
7.3	2 bits	Exhaust brake Brake Assist Switch	--	00000011	0003	3	4 states	0	Not available	-
7.5	2 bits	Idle Command	--	00000011	0003	3	4 states	0	Not available	-
7.7	2 bits	White smoke limit request	--	00000000	0000	0	4 states	0	White smoke limit not demanded	-
8	1 byte	Retarder Selection	--	00000000	0000	0	0,4 %/bit	0	0	%

Data Field

PGN	0xFF89	Name	DLN6 - Proprietary
Scania PGN - Message sent to Engine (DA - 00)			

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0x89
Default priority	3
Parameter Group Number	0xFF89
Source Address	0x27 (Coordinator)

CAN-ID				Data			Decoding					
	Start Posit	Length	Parameter Name	SPN	Bin	Hex	Dec	Resolution	Offset	Real value	Units	
Data Field	4.1	2 bits	Road Speed Governor (Speed Limiter)	--	00000011	03	3	4	states	0	Not available	-
	4.3	4 bits	DPF Manual Activation	--	00001111	0F	15	4	states	0	Don't Care	-
	4.7	2 bits	DPF Manual Inhibit	--	00000011	03	3	4	states	0	Not available	-
	5-6	2 bytes	High Resolution Barometric Pressure	--	--	07BD	1981	0,05	kPa/bit	0	99,05	kPa
	8.1	2 bits	Battery Management Idle Speed Increase Request	--	00000001	01	1	4	states	0	Idle speed increase requested	-
	8.3	6 bits	Requested Generator PWM	--	00111111	3F	63	2	states	0	Not available	-

PGN	0xFFB0	Name	Coordinator General Information
Scania Message			

Transmission Repetition Rate [ms]	200
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0xB0
Default priority	3
Parameter Group Number	0xFFB0
Source Address	0x27 (Coordinator)

CAN-ID	Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			
						Bin	Hex	Dec	Resolution	Offset	Real value	Units
	4.3	2 bits	Engine stop switch	--	00000000	00	0	4	states	0	Engine stop switch not activated	-
	4.5	2 bits	0,8 bar parking brake	--	00000000	00	0	4	states	0	0,8 bar parking brake not set	-
	5.7	2 bits	Gearbox in reverse	--	00000000	00	0	4	states	0	Gearbox not in reverse	-

PGN	0xD0FF	Name	Cab Illumination Message
Message from Instrument Cluster (DA - FF)			

Transmission Repetition Rate [ms]	100
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xD0
PDU specific	0xFF
Default priority	6
Parameter Group Number	0xD0FF
Source Address	0x17 (ICL)

CAN-ID

Data	Start Posit	Length	Parameter Name	SPN	Data			Decoding				
					Bin	Hex	Dec	Resolution	Offset	Real value	Units	
	1	1 byte	Requested illumination percentage	--	11111010	FA	250	0,4	%/bit	0	100	%

PGN	0xEBFF	Name	Transport Protocol – Data Transfer
Message used to transport more than 8 bytes PGN (FEE3)			

Transmission Repetition Rate [ms]	As request
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xEB
PDU specific	0xFF
Default priority	6
Parameter Group Number	0xEBFF
Source Address	0x00 (Engine)

CAN-ID

Data Field

Start Posit	Length	Parameter Name	SPN	Data			Decoding			
				Bin	Hex	Dec	Resolution	Offset	Real value	Units
1	1 byte	Sequence number (1-255)	--	00000001	01	1	--	0	1	-
2	1 byte	Data	--	11000000	C0	192	--	0	--	-
3	1 byte	Data	--	00010010	12	18	--	0	--	-
4	1 byte	Data	--	10101001	A9	169	--	0	--	-
5	1 byte	Data	--	01100000	60	96	--	0	--	-
6	1 byte	Data	--	00111011	3B	59	--	0	--	-
7	1 byte	Data	--	11010101	D5	213	--	0	--	-
8	1 byte	Data	--	01000000	40	64	--	0	--	-

PGN	0xECFF	Name	Transport Protocol – Connection Managem
Message used to transport more than 8 bytes PGN (FEE3)			

Transmission Repetition Rate [ms]	As request
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xEC
PDU specific	0xFF
Default priority	6
Parameter Group Number	0xECFF
Source Address	0x00 (Engine)

CAN-ID

Data Field

Start Posit	Length	Parameter Name	SPN	Data			Decoding			
				Bin	Hex	Dec	Resolution	Offset	Real value	Units
1	1 byte	Control byte = 32 (0x20) for BAM	--	00100000	20	32	--	0	--	-
2	1 byte	Message size (low-byte)	--	00011100	1C	28	--	0	--	-
3	1 byte	Message size (high-byte)	--	00000000	00	0	--	0	--	-
4	1 byte	Total number of packages	--	00000100	04	4	--	0	--	-
5	1 byte	Reserved = 0xFF	--	11111111	FF	255	--	0	--	-
6	1 byte	Parameter Group Number (low-byte)	--	11100011	E3	227	--	0	--	-
7	1 byte	Parameter Group Number (mid-byte)	--	11111110	FE	254	--	0	--	-
8	1 byte	Parameter Group Number (high-byte)	--	00000000	00	0	--	0	--	-

PGN	0xFEE3	Name	Engine Configuration 1 (EC1)
Message via TP-BAM			

<b>Transmission Repetition Rate [ms]</b>	On change of torque/speed points of more than 10% since last transmission. Alternative: Every 5 s.
--	--

<b>Data length</b>	39 bytes
<b>Extended Data Page (R) :</b>	0
<b>Data page</b>	0
<b>PDU format</b>	0xFE
<b>PDU specific</b>	0xE3
<b>Default priority</b>	6
<b>Parameter Group Number</b>	0xFEE3
<b>Source Address</b>	0x00 (Engine)

CAN-ID

Data Field

Start Posit	Length	Parameter Name	SPN	Data			Decoding				
				Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1 - 2	2 byte	Engine speed at idle, point 1	188	--	12C0	4800	0,125	rpm/bit	0	600	rpm
3	1 byte	Percent torque at idle, point 1	539	00010010	12	18	1	%/bit	-125	-107	%
4 - 5	2 byte	Engine speed at point 2	528	--	3B60	15200	0,125	rpm/bit	0	1900	rpm
6	1 byte	Percent torque at point 2	540	11010101	D5	213	1	%/bit	-125	88	%
7 - 8	2 byte	Engine speed at point 3	529	--	1F40	8000	0,125	rpm/bit	0	1000	rpm
9	1 byte	Percent torque at point 3	541	11011011	DB	219	1	%/bit	-125	94	%
10 - 11	2 byte	Engine speed at point 4	530	--	2260	8800	0,125	rpm/bit	0	1100	rpm
12	1 byte	Percent torque at point 4	542	11011110	DE	222	1	%/bit	-125	97	%
13 - 14	2 byte	Engine speed at point 5	531	--	2AF0	10992	0,125	rpm/bit	0	1374	rpm
15	1 byte	Percent torque at point 5	543	11100001	E1	225	1	%/bit	-125	100	%
16 - 17	2 byte	Engine speed at high idle, point 6	532	--	1F40	8000	0,125	rpm/bit	0	1000	rpm
18 - 19	2 byte	Engine Gain (Kp) Of The Endspped Governor (Engine Configuration)	545	--	FFFF	65535	0,00078	%/rpm / b	0	Not available	%/rpm
20 - 21	2 byte	Reference engine torque	544	--	0949	2377	1	Nm/bit	0	2377	Nm
22 - 23	2 byte	Maximum momentary engine override speed, point 7	533	--	FEFE	65278	0,125	rpm/bit	0	Not available	rpm
24	1 byte	Maximum momentary over-ride time limit	534	11111110	FE	254	0,1	seg/bit	0	Not available	seg
25	1 byte	Requested speed control range lower limit	535	11111110	FE	254	10	rpm/bit	0	Not available	rpm
26	1 byte	Requested speed control range upper limit	536	11111110	FE	254	10	rpm/bit	0	Not available	rpm
27	1 byte	Requested torque control range lower limit	537	11111110	FE	254	1	%/bit	-125	Not available	%
28	1 byte	Requested torque control range upper limit	538	11111110	FE	254	1	%/bit	-125	Not available	%
29 - 30	2 byte	Engine Extended Range Requested Speed Control Range Upper Limit	1712	--	FEFE	65278	0,125	rpm/bit	0	Not available	rpm
31 - 32	2 byte	Engine Moment of Inertia	1794	--	FEFE	65278	0,004	kg-m^2/bi	0	Not available	kg-m^w2
33 - 34	2 byte	Engine Default Torque Limit	1846	--	FEFE	65278	1	Nm/bit	0	Not available	Nm
35	1 byte	Support Variable Rate TSC1 Message	3344	11111110	FE	254	256	states	0	Engine only supports standard temporary power train control	--
36	1 byte	Support TSC1 Control Purpose Group 1	3345	11111110	FE	254	256	states	0	Not available	--
37	1 byte	Support TSC1 Control Purpose Group 2	3346	11111110	FE	254	256	states	0	Not available	--
38	1 byte	Support TSC1 Control Purpose Group 3	3347	11111110	FE	254	256	states	0	Not available	--
39	1 byte	Support TSC1 Control Purpose Group 4	3348	11111110	FE	254	256	states	0	Not available	--

PGN	0xF000	Name	Electronic Retarder Controller 1 (ERC1)
-----	--------	------	---

Transmission Repetition Rate [ms]	100
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xF0
PDU specific	0x00
Default priority	6
Parameter Group Number	0xF000
Source Address	0x29 (Exhaust Retarder)

CAN-ID

Data Field

Start Posit	Length	Parameter Name	SPN	Data			Decoding				Units
				Bin	Hex	Dec	Resolution	Offset	Real value		
1.1	4 bits	Retarder Torque Mode	900	00000000	00	0	16	states	0	No request (default mode)	--
1.5	2 bits	Retarder Enable - Brake Assist Switch	571	00000011	03	3	4	states	0	Not available	--
1.7	2 bits	Retarder Enable - Shift Assist Switch	572	00000011	03	3	4	states	0	Not available	--
2	1 byte	Actual Retarder - Percent Torque	520	01111101	7D	125	1	%/bit	-125	0	%
3	1 byte	Intended Retarder Percent Torque	1085	11111111	FF	255	1	%/bit	-125	Not available	%
4.1	2 bits	Engine Coolant Load Increase	1082	00000011	03	3	4	states	0	Not available	--
4.3	2 bits	Retarder Requesting Brake Light	1667	00000000	00	0	4	states	0	--	--
5	1 byte	Source Address of Controlling Device for Retarder Control	1480	11111111	FF	255	Source Adress		0	Source Adress	--
6	1 byte	Drivers Demand Retarder - Percent Torque	1715	11111111	FF	255	1	%/bit	-125	Not available	%
7	1 byte	Retarder Selection, non-engine	1716	11111111	FF	255	0,4	%/bit	0	Not available	%
8	1 byte	Actual Maximum Available Retarder - Percent Torque	1717	11111111	FF	255	1	%/bit	-125	Not available	%

PGN	0xF001	Name	Electronic Brake Controller 1 (EBC1)
Used for brake control information			

Transmission Repetition Rate [ms]	100
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xF0
PDU specific	0x01
Default priority	6
Parameter Group Number	0xF001
Source Address	0x0B (Brakes)

CAN-ID

Start Posit	Length	Parameter Name	SPN	Data			Decoding				
				Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1.1	2 bits	ASR Engine Control Active	561	00000011	03	3	4	states	0	Not available	--
1.3	2 bits	ASR Brake Control Active	562	00000011	03	3	4	states	0	Not available	--
1.5	2 bits	Anti-Lock Braking (ABS) Active	563	00000011	03	3	4	states	0	Not available	--
1.7	2 bits	EBS Brake Switch	1121	00000011	03	3	4	states	0	Not available	--
2	1 byte	Brake Pedal Position	521	11111110	FE	254	0,4	%/bit	0	Error indication	%
3.1	2 bits	ABS Off-road Switch	575	00000011	03	3	4	states	0	Not available	--
3.3	2 bits	ASR Off-road Switch	576	00000011	03	3	4	states	0	Not available	--
3.5	2 bits	ASR "Hill Holder" Switch	577	00000011	03	3	4	states	0	Not available	--
3.7	2 bits	Traction Control Override Switch	1238	00000011	03	3	4	states	0	Not available	--
4.1	2 bits	Accelerator Interlock Switch	972	00000011	03	3	4	states	0	Not available	--
4.3	2 bits	Engine Derate Switch	971	00000011	03	3	4	states	0	Not available	--
4.5	2 bits	Engine Auxiliary Shutdown Switch	970	00000011	03	3	4	states	0	Not available	--
4.7	2 bits	Remote Accelerator Enable Switch	969	00000011	03	3	4	states	0	Not available	--
5	1 byte	Engine Retarder Selection	973	11111111	FF	255	0,4	%/bit	0	Not available	%
6.1	2 bits	ABS Fully Operational	1243	00000011	03	3	4	states	0	Not available	--
6.3	2 bits	EBS Red Warning Signal	1439	00000011	03	3	4	states	0	Not available	--
6.5	2 bits	ABS/EBS Amber Warning Signal (Powered Vehicle)	1438	00000011	03	3	4	states	0	Not available	--
6.7	2 bits	ATC/ASR Information Signal	1793	00000011	03	3	4	states	0	Not available	--
7	1 byte	Source Address of Controlling Device for Brake Control	1481	11111111	FF	255	Source Adress		0	Source Adress	--
8.3	2 bits	Halt brake switch	2911	00000011	03	3	4	states	0	Not available	--
8.5	2 bits	Trailer ABS Status	1836	00000011	03	3	4	states	0	Not available	--
8.7	2 bits	Tractor-Mounted Trailer ABS Warning Signal 1792	1792	00000011	03	3	4	states	0	Not available	--

Data Field

PGN	0xFDE1	Name	Cab A/C Climate System Information (CAC1)									
This message contains measurement and condition information from cab air conditioning components.												
Transmission Repetition Rate [ms]		1000										
Data length		8 bytes										
CAN-ID	Extended Data Page (R) :		0									
	Data page		0									
	PDU format		0xFD									
	PDU specific		0xE1									
	Default priority		6									
	Parameter Group Number		0xFDE1									
	Source Address		0x27 (Coordinator)									
Data				Data			Decoding					
	Start Posit	Length	Parameter Name	SPN	Bin	Hex	Dec	Resolution		Offset	Real value	Units
1	1 byte	Cab A/C Refrigerant Compressor Outlet Pressure	2609	11111111	FF	255	16	kPa/bit	0	Not available	kPa	

PGN	0xFEAE	Name	Air Supply Pressure (AIR1)
Air Supply Pressure			

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xAE
Default priority	6
Parameter Group Number	0xFEAE
Source Address	0x30 (Air Processing System)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			
					Bin	Hex	Dec	Resolution	Offset	Real value	Units
1	1 byte	Pneumatic Supply Pressure	46	11111111	FF	255	8	kPa/bit	0	Not available	kPa
2	1 byte	Parking and/or Trailer Air Pressure	1086	00000010	02	2	8	kPa/bit	0	16	kPa
3	1 byte	Service Brake Air Pressure Circuit #1	1087	00000000	00	0	8	kPa/bit	0	0	kPa
4	1 byte	Service Brake Air Pressure Circuit #2	1088	00000000	00	0	8	kPa/bit	0	0	kPa
5	1 byte	Auxiliary Equipment Supply Pressure	1089	11111111	FF	255	8	kPa/bit	0	Not available	kPa
6	1 byte	Air Suspension Supply Pressure	1090	11111111	FF	255	8	kPa/bit	0	Not available	kPa
7.1	2 bits	Air Compressor Status	1351	00000001	01	1	4	states	0	Compressor active	--

PGN	0xFEE5	Name	Engine Hours, Revolutions
-----	--------	------	---------------------------

<b>Transmission Repetition Rate [ms]</b>	On Request
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xE5
Default priority	6
Parameter Group Number	0xFEE5
Source Address	0x00 (Engine)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			Units
					Bin	Hex	Dec	Resolution	Offset	Real value	
1 - 4	4 byte	Engine Total Hours of Operation	247	--	000151F7	86519	0,05	h/bit	0	4325,95	hours
5 - 8	4 byte	Engine Total Revolutions	249	--	FFFFFFFF	4,29E+09	1000	r/bit	0	Not available	revolution

PGN	0xFEE6	Name	Time/Date (TD)

<b>Transmission Repetition Rate [ms]</b>	On request
<b>Data length</b>	8 bytes
<b>Extended Data Page (R) :</b>	0
<b>Data page</b>	0
<b>PDU format</b>	0xFE
<b>PDU specific</b>	0xE6
<b>Default priority</b>	6
<b>Parameter Group Number</b>	0xFEE6
<b>Source Address</b>	0x17 (Instrument Cluster)

CAN-ID				Data			Decoding					
	Start Posit	Length	Parameter Name	SPN	Bin	Hex	Dec	Resolution		Offset	Real value	Units
Data Field	1	1 byte	Seconds	959	11111110	FE	254	0,25	s/bit	0	Error indication	s
	2	1 byte	Minutes	960	11111110	FE	254	1	min/bit	0	Error indication	min
	3	1 byte	Hours	961	11111110	FE	254	1	hr/bit	0	Error indication	hour
	4	1 byte	Month	963	11111110	FE	254	1	month/bit	0	Error indication	month
	5	1 byte	Day	962	11111110	FE	254	0,25	day/bit	0	Error indication	day
	6	1 byte	Year	964	11111110	FE	254	1	year/bit	0	Error indication	years
	7	1 byte	Local minute offset	1601	11111110	FE	254	1	min/bit	-125	Error indication	min
	8	1 byte	Local hour offset	1602	11111110	FE	254	1	hr/bit	-125	Error indication	hour

PGN	0xFEE9	Name	Fuel Consumption (LFC)
-----	--------	------	------------------------

Transmission Repetition Rate [ms]	On request
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xE9
Default priority	6
Parameter Group Number	0xFEE9
Source Address	0x00 (Engine)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			
					Bin	Hex	Dec	Resolution	Offset	Real value	Units
1 - 4	4 byte	Engine Trip Fuel	182	--	FFFFFFFF	4,29E+09	0,5	L/bit	0	Not available	Liters
5 - 8	4 byte	Engine Total Fuel Used	250	--	00026170	156016	0,5	L/bit	0	78008	Liters

PGN	0xFEFE	Name	Engine Temperature 1 (ET1)
-----	--------	------	----------------------------

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xEE
Default priority	6
Parameter Group Number	0xFEFE
Source Address	0x00 (Engine)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding				
					Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1	1	1 byte	Engine Coolant Temperature	110	00111000	38	56	1	deg C/bit	-40	16	deg C
2	2	1 byte	Engine Fuel Temperature 1	174	00111000	38	56	1	deg C/bit	-40	16	deg C
3 - 4	3 - 4	2 bytes	Engine Oil Temperature 1	175	--	FFFF	65535	0,03125	deg C/bit	-273	Not available	deg C
5 - 6	5 - 6	2 bytes	Engine Turbocharger Oil Temperature	176	--	FFFF	65535	0,03125	deg C/bit	-273	Not available	deg C
7	7	1 byte	Engine Intercooler Temperature	52	11111111	FF	255	1	deg C/bit	-40	Not available	deg C
8	8	1 byte	Engine Intercooler Thermostat Opening	1134	11111111	FF	255	0,4	%/bit	0	Not available	%

PGN	0xFEFE	Name	Engine Fluid Level/Pressure 1 (EFL/P1)
-----	--------	------	--

Transmission Repetition Rate [ms]	500
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xEF
Default priority	6
Parameter Group Number	0xFEFE
Source Address	0x27 (Coordinator)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			
					Bin	Hex	Dec	Resolution	Offset	Real value	Units
1	1 byte	Engine Fuel Delivery Pressure	94	00001010	0A	10	4	kPa/bit	0	40	kPa
2	1 byte	Engine Extended Crankcase Blow-by Pressure	22	11111111	FF	255	0,05	kPa/bit	0	Not available	kPa
3	1 byte	Engine Oil Level	98	11111111	FF	255	0,4	%/bit	0	Not available	%
4	1 byte	Engine Oil Pressure	100	00000000	00	0	4	kPa/bit	0	0	kPa
5 - 6	2 bytes	Engine Crankcase Pressure	101	--	FFFF	65535	0,0078	kPa/bit	-250	Not available	kPa
7	1 byte	Engine Coolant Pressure	109	11111111	FF	255	2	kPa/bit	0	Not available	kPa
8	1 byte	Engine Coolant Level	111	00000000	00	0	0,4	%/bit	0	0	%

PGN	0xFE1	Name	Cruise Control/Vehicle Speed (CCVS)
-----	-------	------	-------------------------------------

<b>Transmission Repetition Rate [ms]</b>	100
<b>Data length</b>	8 bytes
<b>Extended Data Page (R) :</b>	0
<b>Data page</b>	0
<b>PDU format</b>	0xFE
<b>PDU specific</b>	0xF1
<b>Default priority</b>	6
<b>Parameter Group Number</b>	0xFE1
<b>Source Address</b>	0x00 (Engine)

CAN-ID	Data							Decoding				
	Start Posit	Length	Parameter Name	SPN	Bin	Hex	Dec	Resolution	Offset	Real value	Units	
Data Field	1.1	2 bits	Two Speed Axle Switch	69	00000011	03	3	4	states	0	Not available	--
	1.3	2 bits	Parking Brake Switch	70	00000001	01	1	4	states	0	Not available	--
	1.5	2 bits	Cruise Control Pause Switch	1633	00000011	03	3	4	states	0	Take No Action	--
	2 - 3	2 bytes	Wheel-Based Vehicle Speed	84	00000000	00	0	0,0039	km/h / bit	0	0	km/h
	4.1	2 bits	Cruise Control Active	595	00000000	00	0	4	states	0	Cruise control switched off	--
	4.3	2 bits	Cruise Control Enable Switch	596	00000001	01	1	4	states	0	Cruise control enabled	--
	4.5	2 bits	Brake Switch	597	00000010	02	2	4	states	0	Error	--
	4.7	2 bits	Clutch Switch	598	00000000	00	0	4	states	0	Clutch pedal released	--
	5.1	2 bits	Cruise Control Set Switch	599	00000011	03	3	4	states	0	Not available	--
	5.3	2 bits	Cruise Control Coast (Decelerate) Switch	600	00000000	00	0	4	states	0	Cruise control activator not in the position "coast"	--
	5.5	2 bits	Cruise Control Resume Switch	601	00000000	00	0	4	states	0	Cruise control activator not in the position "resume"	--
	5.7	2 bits	Cruise Control Accelerate Switch	602	00000000	00	0	4	states	0	Cruise control activator not in the position "accelerate"	--
	6	1 byte	Cruise Control Set Speed	86	11111111	FF	255	1	km/h / bit	0	Not available	km/h
	7.1	5 bits	PTO State	976	00011111	1F	31	32	states	0	Not available	--
	7.6	3 bits	Cruise Control States	527	00000000	00	0	8	states	0	Off/Disabled	--
	8.1	2 bits	Engine Idle Increment Switch	968	00000011	03	3	4	states	0	Not available	--
	8.3	2 bits	Engine Idle Decrement Switch	967	00000011	03	3	4	states	0	Not available	--
8.5	2 bits	Engine Test Mode Switch	966	00000011	03	3	4	states	0	Not available	--	
8.7	2 bits	Engine Shutdown Override Switch	1237	00000011	03	3	4	states	0	Not available	--	

<b>PGN</b>	0xFEFE2	<b>Name</b>	Fuel Economy (Liquid)
------------	---------	-------------	-----------------------

<b>Transmission Repetition Rate [ms]</b>	100
<b>Data length</b>	8 bytes
<b>Extended Data Page (R) :</b>	0
<b>Data page</b>	0
<b>PDU format</b>	0xFE
<b>PDU specific</b>	0xF2
<b>Default priority</b>	6
<b>Parameter Group Number</b>	0xFEFE2
<b>Source Address</b>	0x00 (Engine)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			
					Bin	Hex	Dec	Resolution	Offset	Real value	Units
1 - 2	2 bytes	Engine Fuel Rate	183	--	0000	0	0,05	L/h / bit	0	0	L/h
3 - 4	2 bytes	Engine Instantaneous Fuel Economy	184	--	FAFF	64255	0,0020	km/L / bit	0	125,4980469	km/L
5 - 6	2 bytes	Engine Average Fuel Economy	185	--	FFFF	65535	0,0020	km/L / bit	0	Not available	km/L
7	1 byte	Engine Throttle Position	51	11111111	FF	255	0,4	%/bit	0	Not available	%

PGN	0xFE5	Name	Ambient Conditions (AMB)
-----	-------	------	--------------------------

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xF5
Default priority	6
Parameter Group Number	0xFE5
Source Address	0x27 (Coordinator)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding				
					Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1	1	1 byte	Barometric Pressure	108	11000110	C6	198	0,5	kPa/bit	0	99	kPa
2 - 3	2	2 bytes	Cab Interior Temperature	170	--	FFFF	65535	0,0313	deg C/bit	-273	Not available	deg C
4 - 5	2	2 bytes	Ambient Air Temperature	171	--	242F	9263	0,0313	deg C/bit	-273	16,46875	deg C
6	1	1 byte	Engine Air Inlet Temperature	172	11111111	FF	255	1	deg C/bit	-40	Not available	deg C
7 - 8	2	2 bytes	Road Surface Temperature	79	--	FFFF	65535	0,03125	deg C/bit	-273	Not available	deg C

PGN	0xFE6	Name	Inlet/Exhaust Conditions 1 (IC1)
-----	-------	------	----------------------------------

Transmission Repetition Rate [ms]	500
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xF6
Default priority	6
Parameter Group Number	0xFE6
Source Address	0x00 (Engine)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding				
					Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1	1	1 byte	Engine Particulate Trap Inlet Pressure	81	11111111	FF	255	0,5	kPa/bit	0	Not available	kPa
2	2	1 byte	Engine Turbocharger Boost Pressure	102	00000010	02	2	2	kPa/bit	0	4	kPa
3	3	1 byte	Engine Intake Manifold 1 Temperature	105	00110111	37	55	1	deg C/bit	-40	15	deg C
4	4	1 byte	Engine Air Inlet Pressure	106	11111111	FF	255	2	kPa/bit	0	Not available	kPa
5	5	1 byte	Engine Air Filter 1 Differential Pressure	107	11111111	FF	255	0,05	kPa/bit	0	Not available	kPa
6 - 7	6 - 7	2 bytes	Engine Exhaust Gas Temperature	173	--	FFFF	65535	0,03125	deg C/bit	-273	Not available	deg C
8	8	1 byte	Engine Coolant Filter Differential Pressure	112	11111111	FF	255	0,5	kPa/bit	0	Not available	kPa

PGN	0xFEFC	Name	Dash Display (DD)
-----	--------	------	-------------------

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFE
PDU specific	0xFC
Default priority	6
Parameter Group Number	0xFEFC
Source Address	0x27 (Coordinator)

CAN-ID

Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding				
					Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1		1 byte	Washer Fluid Level	80	11111111	FF	255	0,4	%/bit	0	Not available	%
2		1 byte	Fuel Level	96	10011111	9F	159	0,4	%/bit	0	63,6	%
3		1 byte	Engine Fuel Filter Differential Pressure	95	11111111	FF	255	2	kPa/bit	0	Not available	kPa
4		1 byte	Engine Oil Filter Differential Pressure	99	11111111	FF	255	0,5	kPa/bit	0	Not available	kPa
5 - 6		2 bytes	Cargo Ambient Temperature	169	--	FFFF	65535	0,03125	deg C/bit	-273	Not available	dec C

PGN	0xFF60	Name	General Purpose Message 1
Scania Message sent from Engine			

Transmission Repetition Rate [ms]	5000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0x60
Default priority	6
Parameter Group Number	0xFF60
Source Address	0x00 (Engine)

CAN-ID

Data	Start Posit	Length	Parameter Name	SPN	Data			Decoding			Units
					Bin	Hex	Dec	Resolution	Offset	Real value	
1 - 4	4 bytes	Engine No.	80	--	09120615	1,52E+08	Hex number	0	09120615	--	

PGN	0xFF88	Name	DLN8
Scania Message sent from Engine			

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0x88
Default priority	6
Parameter Group Number	0xFF88
Source Address	0x00 (Engine)

CAN-ID

Start Posit	Length	Parameter Name	SPN	Data			Decoding				
				Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1	1 byte	Coolant Water Flow	--	11111111	FF	255	10	l/min / bit	0	Not available	l/min
2	1 byte	Cooling Fan Speed	--	00000000	00	0	40	rpm/bit	0	0	rpm
3 - 4	2 bytes	Applied Vehicle Speed Limit Proprietary	--	--	05A0	1440	0,0039	km/h / bit	0	5,625	km/h
5.1	6 bits	DPF Regeneration Countdown Timer	--	00111111	3F	63	3	states	0	Not available	--
5.7	2 bits	Emission-OBD reactivation mode	--	00000011	03	3	4	states	0	Not available	--
6.1	3 bits	StarterMotorTempStatus	--	00000000	00	0	8	states	0	Starter Motor Normal Temp	--
6.4	4 bits	Emission-OBD inducement state	--	00001111	0F	15	16	states	0	Not available	--
7	1 byte	Time to speed limiting	--	11111111	FF	255	1	h/bit	0	Not available	h
8.1	3 bits	Urea level inducement state	--	00000111	07	7	8	states	0	Not available	--
8.4	4 bits	Emission-OBD inducement failure reason	--	00001111	0F	15	16	states	0	Not available	--

Data Field

PGN	0xFFA0	Name	DLN5-Proprietary
Scania Message sent from Coordinator			

Transmission Repetition Rate [ms]	50
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0xA0
Default priority	6
Parameter Group Number	0xFFA0
Source Address	0x27 (Coordinator)

CAN-ID	Data Field	Start Posit	Length	Parameter Name	SPN	Data			Decoding			
						Bin	Hex	Dec	Resolution	Offset	Real value	Units
	2.1	2 bits	Split low switch	--	00000011	03	3	4	states	0	Not available	--
	2.3	2 bits	Range low switch	--	00000000	00	0	4	states	0	Not active	--
	3	1 byte	Clutch pedal position	--	00000010	02	2	0,4	%/bit	0	0,8	%

PGN	0xFFB1	Name	CUV information
Scania Message sent from Visibility System			

Transmission Repetition Rate [ms]	1000
Data length	8 bytes
Extended Data Page (R) :	0
Data page	0
PDU format	0xFF
PDU specific	0xB1
Default priority	6
Parameter Group Number	0xFFB1
Source Address	0x1E (Visibility System)

CAN-ID

Start Posit	Length	Parameter Name	SPN	Data			Decoding				
				Bin	Hex	Dec	Resolution	Offset	Real value	Units	
1.1	2 bits	Main beam intended	--	00000000	00	0	4	states	0	Off	--
1.3	2 bits	Dipped beam intended	--	00000001	01	1	4	states	0	On	--
2.1	2 bits	Front fog lamp intended	--	00000000	00	0	4	states	0	Off	--
2.3	2 bits	Rear fog lamp intended	--	00000000	00	0	4	states	0	Off	--
2.5	2 bits	Reverse lamp intended	--	00000000	00	0	4	states	0	Off	--
2.7	2 bits	Stop lamp intended	--	00000001	01	1	4	states	0	On	--
3.1	2 bits	Wiper intended	--	00000000	00	0	4	states	0	Off	--
3.5	2 bits	Direction indicator lamp status truck left	--	00000000	00	0	4	states	0	Off	--
3.7	2 bits	Direction indicator lamp status truck right	--	00000000	00	0	4	states	0	Off	--
4.1	2 bits	Direction indicator lamp status trailer left	--	00000000	00	0	4	states	0	Off	--
4.3	2 bits	Direction indicator lamp status trailer right	--	00000000	00	0	4	states	0	Off	--
4.5	2 bits	Direction indicator lever status	--	00000011	03	3	4	states	0	Don't care/take no action	--
4.7	2 bits	Horn Switch State	--	00000000	00	0	4	states	0	Off	--
7.1	2 bits	Work light toggle switch	--	00000000	00	0	4	states	0	Off	--
7.3	2 bits	Work light output status	--	00000000	00	0	4	states	0	Off	--
7.5	2 bits	58	--	00000001	01	1	4	states	0	On	--
8.5	2 bits	Parking lights	--	00000000	00	0	4	states	0	Switch not in parking lights position	--
8.7	2 bits	Driving lights	--	00000001	01	1	4	states	0	Switch in driving lights position	--

Data field