



Desenvolvimento de Interface HMI para Veículo Autocarro Eléctrico

RÚBEN SEARA BRANDÃO

julho de 2018

Rúben Seara Brandão

Desenvolvimento de Interface HMI para Veículo Autocarro Elétrico

Departamento de Engenharia Eletrotécnica

Mestrado em Engenharia Eletrotécnica – Sistemas Elétricos de Energia

2018

Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de DSEE – Dissertação do Mestrado em Engenharia Eletrotécnica – Sistemas Elétricos de Energia

Candidato: Rúben Seara Brandão, Nº 1110703, 1110703@isep.ipp.pt

Orientação científica: Custódio Dias, cpd@isep.ipp.pt

Empresa: CAETANOBUS, Grupo Salvador Caetano

Supervisão: Tiago Rocha, tiago.rocha@caetano.bus.pt



Departamento de Engenharia Eletrotécnica

Mestrado em Engenharia Eletrotécnica – Sistemas Elétricos de Energia

2018

Agradecimentos

Aos meus pais pelo amor, apoio e incentivo depositados em mim.

Aos amigos e família por todos momentos passados.

Ao Professor Doutor Custódio por ter aceite o cargo de orientador, pela ajuda e por todo trabalho prestado.

A toda a CAETANOBUS, nomeadamente à equipa na qual trabalhei, pela ajuda, pela forma como me acolheram e pela cordialidade mostrada.

À Gdańsk University of Technology - Politechnika Gdańska.

Ao Instituto Superior de Engenharia do Porto.

Resumo

A segurança é um dos pilares em que assenta o sucesso dos mercados automóveis. O setor automóvel encontra-se em constante desenvolvimento, sobretudo na inclusão de veículos 100% elétricos nas estradas. Estes veículos incorporam sistemas eletrónicos cada vez mais complexos e tecnologias inovadoras com o intuito de melhorar a eficiência energética, a segurança e a redução de custos. Num ambiente intraurbano, sobretudo nas grandes cidades, a elevada afluência de trânsito requer cada vez mais sistemas de transporte inteligentes (*Intelligent Transport Systems – ITS*) que ajudem a auxiliar os motoristas de veículos acerca da sua condução.

É neste âmbito que surge este trabalho, onde é pretendido desenvolver um *software* a aplicar numa *human-machine interface* (HMI) instalada num autocarro elétrico com o objetivo de prevenir o motorista de causar ou ser envolvido em acidentes rodoviários. Através da instalação de unidades de transmissão de dados em pontos estratégicos nas estradas e da instalação de unidades de leitura de dados no autocarro vai ser possível estabelecer uma comunicação, que terá como objetivo emitir alertas de perigo que se adivinhem no percurso que o autocarro se encontra a efetuar. A esta comunicação dá-se o nome de *vehicle-to-everything* (V2X), que consiste em passar informações entre um veículo e uma entidade e vice-versa. Este inovador tipo de comunicação, levará a benefícios em termos de segurança nas estradas, eficiência energética obtida na forma de condução dos motoristas e na fluidez de trânsito nas grandes cidades, levando consequentemente a uma redução de custos das várias identidades a que podem estar relacionadas um acidente.

Na primeira fase deste trabalho foi efetuada uma análise referente à forma como a informação deve ser transmitida ao motorista sem o privar ou dificultar a sua condução. Foi também realizado um estudo sobre a comunicação *Controller Area Network* (CAN), que é uma rede de comunicação entre as unidades de controlo eletrónico de um veículo e os outros dispositivos, de forma a compreender como é realizada a comunicação *in-vehicle* e sobre o *software* a utilizar para a elaboração deste projeto. Por fim, realizou-se um estudo acerca da forma como a comunicação entre veículos e infraestrutura rodoviária pode ser realizada.

O *software* elaborado possui funções que permitem desempenhar um papel de auxílio na condução do motorista, cumprindo com os objetivos estabelecidos neste projeto. Os testes elaborados ao *software* desenvolvido permitiram verificar e aperfeiçoar o comportamento deste conforme o pretendido.

Por fim, foram realizadas as conclusões necessárias e descritas perspectivas de trabalhos futuros neste âmbito.

Palavras-Chave

HMI, Comunicação CAN, Codesys, V2V, V2I, ITS, Road Safety, Segurança Ativa

Abstract

Safety is one of the pillars on which the success of automotive markets is highlighted. The automotive sector is constantly in development, especially in the inclusion of 100% electric vehicles on the roads. These vehicles feature increasingly complex electronic systems and innovative technologies with the aim of improving energy efficiency, safety and cost reduction. In an intra-urban environment, especially in large cities, the high flow of traffic requires more and more *Intelligent Transport Systems* (ITS) that help to assist motorists in their driving.

It is within this scope that this work appears, where it is intended to develop a software to be applied in a *human-machine interface* (HMI) installed in an electric bus with the purpose of preventing the driver from causing or being involved in road accidents. Through the installation of data transmission units at strategic points on the roads and the installation of data reading units on the bus, it will be possible to establish a communication that will have the objective of issuing warnings of danger that can be discerned in the route that the bus is doing. This communication is called vehicle-to-everything (V2X), which consists of passing information between a vehicle and an entity and vice versa. This innovative type of communication will lead to benefits in terms of road safety, energy efficiency achieved in the form of driver driving and traffic flow in large cities, leading to a reduction in costs of various identities that may be related to an accident.

In the first phase of this work, an analysis was made regarding the way in which the information should be transmitted to the driver without depriving him or hindering his driving. A study was also carried out on the Controller Area Network (CAN) communication, which is a communication network between the electronic control units of a vehicle and the other devices, in order to understand how in-vehicle communication it is done and on the software development to be used in this project. Finally, a study was carried out on how the communication between vehicles and road infrastructure can exist.

The elaborated software has functions that allow to play a helping role in the driving way of the driver, fulfilling the objectives established for this project. The tests

developed to the software allowed to verify and to improve the behavior of this one according to what it was intended.

Finally, the necessary conclusions and described future prospective work in this area were carried out.

Keywords

HMI, CAN Communication, Codesys, V2V, V2I, ITS, Road Safety, Segurança Ativa

Índice

Agradecimentos	iii
Resumo	v
Abstract.....	vii
Índice	ix
Índice de Figuras	xiii
Índice de Tabelas	xv
Acrónimos	xvii
1. Introdução.....	1
1.1. Enquadramento.....	1
1.2. CaetanoBus.....	2
1.3. Objetivos	3
1.4. Identificação do problema.....	4
1.4.1. Requisitos do sistema e caderno de encargos.....	6
1.4.2. Arquitetura do sistema	7
1.5. Cronograma.....	8
1.6. Estrutura	8
2. HMI – <i>Human Machine Interface</i>	11
2.1. HMI e o seu desenvolvimento.....	11
2.2. <i>User-centered design</i>	12
2.2.1. A importância dos processos UCD	14
2.2.2. Como medir o sucesso de um processo UCD	15
2.2.3. Vantagens e desvantagens de um processo UCD.....	15
2.3. A importância do design da interface de utilizador.....	16
2.3.1. Tomada de decisão humana	16
2.3.2. Vigilância	16
2.3.3. Carga de trabalho	17
2.3.4. Consciência Situacional	17
2.3.5. Erro humano.....	18
2.3.6. Engenharia cognitiva.....	18
3. Comunicação CAN.....	21
3.1. Características do CAN.....	21
3.2. Método de endereçamento	22

3.3.	Processo de arbitragem.....	23
3.4.	Formato das <i>frames</i>	24
3.5.	Tipos de <i>frames</i>	26
3.5.1.	<i>Frame</i> de dados	26
3.5.2.	<i>Frame</i> remota	27
3.5.3.	<i>Frame</i> de erro	27
3.5.4.	<i>Frame</i> de sobrecarga	28
3.6.	Intervalo entre <i>frames</i>	29
3.7.	Codificação de <i>frames</i>	30
3.8.	Tratamento de erros.....	31
3.8.1.	Sinalização de erro	31
3.8.2.	Deteção de erro.....	32
3.9.	Normalização	32
3.10.	Aplicações e conclusão	32
4.	Software Codesys	35
4.1.	Programação.....	35
4.2.	<i>Runtime System</i>	36
4.3.	<i>Codesys Visualization</i>	36
4.4.	<i>CodesysFieldbus</i>	37
4.5.	<i>Codesys Motion + CNC</i>	37
5.	Comunicação <i>Vehicle-to-X</i>	39
5.1.	<i>Vehicle-to-Vehicle (V2V)</i>	39
5.2.	<i>Vehicle-to-Infrastructure (V2I)</i>	43
5.3.	<i>Vehicle-to-Pedestrian (V2P)</i>	45
5.4.	<i>V2X Standards</i>	45
6.	Descrição da aplicação desenvolvida	47
6.1.	Definição de mensagens a serem aplicadas como comunicação V2I.....	47
6.2.	Desenvolvimento do conjunto de máscaras	60
6.3.	Codificação das mensagens em comunicação CAN	61
6.4.	Desenvolvimento do software	68
6.5.	Definição de tempo de ciclo.....	76
6.6.	Testes ao software	77
7.	Conclusões e perspectivas de trabalhos futuros	81
7.1.	Conclusão	81
7.2.	Perspetivas de trabalhos futuros	82

Referências Bibliográficas.....	85
Anexo A.....	87
General description.....	87
System / Project name	87
Goals.....	87
Environment of the system.....	87
Anexo B - Imagem de mensagem de alerta de trânsito	89
Anexo C - Imagem de mensagem de alerta de vento	89
Anexo D - Imagem de mensagem de alerta de gelo.....	89
Anexo E - Imagem de mensagem de alerta de visibilidade.....	89
Anexo F - Imagem de mensagem de alerta de precipitação	90
Anexo G - Imagem de mensagem de alerta de obras móveis.....	90
Anexo H - Imagem de mensagem de alerta de via fechada.....	90
Anexo I - Imagem de mensagem de alerta de estrada fechada.....	91
Anexo J - Imagem de mensagem de alerta de veículo prioritário	91
Anexo L - Imagem de mensagem de alerta de velocidade otimizada	91

Índice de Figuras

Figura 1: Autocarro e.Cobus	2
Figura 2: Autocarro e.City Gold.....	3
Figura 3: HMI Display (modelo HY-eVision2 7.0) a ser utilizada. (Foto captada pelo mestrando)	5
Figura 4: A vermelho encontra-se localizado o display em questão. (Foto captada pelo mestrando)	5
Figura 5: Arquitetura do sistema referente a este projeto (comunicação V2I).....	7
Figura 6: Calendarização do estágio de dissertação	8
Figura 7: Ilustração de situação de maior valor num processo UCD	14
Figura 8: Os "golfos" da Execução e da Avaliação[5]	19
Figura 9: Etapas de ligação entre "golfos"[5].....	19
Figura 10: Estágios de atividade do usuário na travessia dos "golfos" - Exemplo adaptado [6].....	20
Figura 11: Método de endereçamento em CAN [7]	23
Figura 12: Processo de arbitragem do CAN [8] [9]	24
Figura 13: Diferentes tipos de formatos de frames do CAN [11]	25
Figura 14: Frame de erro do CAN [13]	28
Figura 15: Frame de sobrecarga do CAN [13]	29
Figura 16: Intervalo entre frames do CAN [13]	30
Figura 17: Formato de uma frame de erro em CAN [8].....	31
Figura 18: Ilustração de V2V Communication [23].....	40
Figura 19: Ilustração de comunicação V2V (Aviso de obstáculo) [23]	41
Figura 20: Exemplo de comunicação V2V [24] Y – Veículo Y ; X – Veículo X.....	42
Figura 21: Exemplos de segurança rodoviária através de comunicação V2I [28]	44
Figura 22: Máscara referente à mensagem de alerta de aproximação de veículo prioritário.....	60
Figura 23: Ilustração de codificação das mensagens de alerta em PCAN.....	67
Figura 24: Legenda dos elementos constituintes dos fluxogramas	68
Figura 25: Fluxograma Função de prioridade	69
Figura 26: Ilustração do menu principal do software na HMI desenvolvida	70
Figura 27: Ilustração da mensagem de alerta de aproximação de veículo prioritário (idioma em inglês).....	71
Figura 28: Ilustração da mensagem de alerta de aproximação de obras móveis na estrada	71
Figura 29: Ilustração do menu principal do software na HMI desenvolvida	72
Figura 30: Fluxograma Função de distância	73
Figura 31: Fluxograma Função Marcha-Atrás	75
Figura 32: Imagem exemplo de câmara ON com comando de marcha-atrás ativa num autocarro	76
Figura 33: Ilustração de método de teste de software	77

Índice de Tabelas

Tabela 1: Exemplo de um possível envolvimento UCD	13
Tabela 2: Vantagens e desvantagens dum processo UCD.....	15
Tabela 3: Frame de dados do CAN	26
Tabela 4: Exemplo de codificação bit-stuffing em CAN r – bit recessivo; d – bit dominante.	31
Tabela 5 : Topologia de Caracterização de Mensagem Limite de velocidade otimizada	49
Tabela 6: Topologia de Caracterização de Mensagem Alerta de Congestionamento ...	50
Tabela 7: Topologia de Caracterização de Mensagem Aviso de Condições Meteorológicas Adversas	52
Tabela 8: Topologia de Caracterização de Mensagem Alerta de Obras: via de Trânsito Fechada.....	54
Tabela 9: Topologia de Caracterização de Mensagem Alerta de Obras: Rodovia Fechada	55
Tabela 10: Topologia de Caracterização de Mensagem Alerta de Obras: Trabalhos Móveis na Via.....	57
Tabela 11: Topologia de Caracterização de Mensagem Alerta de Aproximação de Veículo Prioritário	58
Tabela 12: Mensagem de trânsito em CAN.....	62
Tabela 13: Mensagem de via de trânsito fechada em CAN	62
Tabela 14: Mensagem de estrada fechada em CAN.....	63
Tabela 15: Mensagem de trabalhos móveis em CAN	63
Tabela 16: Mensagem de condições meteorológicas adversas em CAN	64
Tabela 17: Mensagem de limite de velocidade otimizada em CAN	64
Tabela 18: Mensagem de alerta de aproximação de veículo prioritário em CAN	65
Tabela 19: Mensagem de velocidade em CAN	65
Tabela 20: Mensagens de distâncias em CAN	66
Tabela 21: Definição de critérios de concretização de teste.....	78
Tabela 22: Definição de casos de teste	79

Acrónimos

ACK – Acknowledge

CAN – Controller Area Network

CFC – Continuous Function Chart

CNC – Computer Numerical Control

CoDeSys – Controller Development System

CPU – Central Processing Unit

CRC – Cyclic Redundancy Checksum

DLC – Data Length Code

DSRC - Dedicated Short-Range Communications

EoF – End of Frame

FBD – Function Block Diagram

GPS - Global Positioning System

HMI – Human Machine Interface

IDE – Identifier Extension

IEC – International Electrotechnical Commission

IEEE - Institute of Electric and Electronic Engineers

IL – Instruction List

ISO – International Standard Organization

I2V – Infrastructure to Vehicle

LD – Ladder Diagram

OBU or OBE – Vehicle On-Board Unit or Equipment

PLC – Controladores Lógicos Programáveis

RES – Reservados

RPM – Rotações por Minuto

RSU or RSE – Roadside Unit or Equipment

RTR – Remote Transmit Request

SAE - Society of Automotive Engineers

SFC – Sequential Function Chart

SoF – Start of Frame

SRR – Substitute Remote Request

ST – Structured Text

UCD – User-Centered Design

V2I – Vehicle to Infrastructure

V2P – Vehicle to Pedestrian

V2V – Vehicle to Vehicle

V2X - Vehicle to Everything

WiFi – Wireless Fidelity

CAPÍTULO 1

1. INTRODUÇÃO

O setor automóvel está em constante evolução. Constantemente são apresentadas novas evoluções tecnológicas, que representam melhorias significativas a nível de eficiência energética, segurança rodoviária, redução de custos, etc. A digitalização da sociedade em que se vive acompanha esta evolução. Hoje, no universo de todo o setor automóvel analisam-se fatores como a conectividade, sistemas de ajuda à condução e novas formas de mobilidade. Juntamente com estes fatores começam a surgir tecnologias que possibilitam a comunicação entre veículos e a infraestrutura que os rodeia. A previsão desta possível comunicação implica o aumento de nível de segurança nas estradas e, conseqüentemente, uma redução na frequência em que ocorrem acidentes, mas para tal é necessário que um grande volume de veículos admita este sistema.

Várias instituições e empresas estão a apostar na conectividade e comunicação entre veículos de forma a estar um passo à frente aquando do possível surgimento desta tecnologia em larga escala. É neste âmbito que surge o trabalho proposto que originou esta dissertação, no qual se pretendeu desenvolver um sistema que permita um apoio à condução de veículos de transporte pesado de passageiros. Para tal, desenvolveu-se um programa de *software*, que permite uma comunicação entre um veículo e a infraestrutura rodoviária. Este projeto foi realizado em ambiente empresarial na empresa CaetanoBus, do Grupo Salvador Caetano, com a orientação dos Engenheiros Tiago Rocha e Ricardo Vasconcelos e do Professor Doutor Custódio Dias, do Instituto Superior de Engenharia do Porto.

1.1. ENQUADRAMENTO

O documento à disposição foi realizado no âmbito da disciplina de dissertação de curso de Mestrado em Engenharia Eletrotécnica – Sistemas Elétricos de Energia, do Instituto Superior de Engenharia do Porto. Tem como tema o Desenvolvimento de uma *Human Machine Interface* (HMI) para Veículo Autocarro Elétrico, escolhido por acordo entre o candidato, o orientador científico e os supervisores na empresa em questão.

O trabalho efetuado consistiu no apoio à empresa na recolha, análise e organização de informação relativa à temática de sistemas de aquisição de dados entre um veículo elétrico e a rede viária e elaboração de *software* para uma interface entre humano e máquina, permitindo o aperfeiçoamento de tecnologias de informação de suporte à decisão do condutor em tempo-real.

1.2. CAETANOBUS

A CaetanoBus, empresa do Grupo Salvador Caetano, é a maior fabricante de carroçarias e autocarros em Portugal. Tem como responsabilidade a produção de autocarros e chassis de várias marcas e com diferentes tipos de especificações, e.g. para transporte urbano, turismo, aeroportos e miniautocarros. O seu desenvolvimento e inovação são parte notória no ramo, permitindo um grande reconhecimento no mercado justificado com a sua constante expansão e atribuição de prémios como *Green Project & DME Awards 2011* pela criação do autocarro de tração 100% elétrica “e.Cobus”. De momento a empresa produz vários tipos de autocarros nos quais se destacam:



Figura 1: Autocarro e.Cobus

O *e.Cobus*, o primeiro autocarro de tração 100% elétrica exclusivo para aeroportos, compreende uma lotação máxima de 112 pessoas e está equipado com baterias com capacidade de 85 kWh ou 116 kWh (Li-FePo4) e motor síncrono de ímãs permanentes, que permitem uma autonomia estimada de entre 60 a 70 km e uma velocidade máxima de 50 km/h. O seu carregamento pode ser realizado através de um carregador a bordo - AC trifásico (tempo de carga 6 horas) ou carregador externo - DC 700V (tempo de carga 3 horas).



Figura 2: Autocarro e.City Gold

O autocarro *e.City Gold* vem corresponder às crescentes imposições políticas de circulação de veículos Diesel nas grandes cidades e às evoluções tecnológicas emergentes. O *e.City Gold* é um veículo que aposta na não emissão de gases poluentes, ruído diminuto e redução de custos de mobilidade. Devido às grandes dinâmicas de uma cidade, como os constantes congestionamentos até ao elevado número de sinalizações, que obriga às constantes paragens, este tipo de veículo apresenta a grande vantagem no seu baixo consumo energético por quilómetro, capacidade de regeneração de energia e consumo energético praticamente nulo nos períodos de tempo em que está efetivamente parado. Equipado com baterias de capacidade até 250 kWh e motor síncrono com potência nominal de 160 kW @ 1500 rpm, é-lhe possível alcançar uma autonomia de até 200 km e uma velocidade máxima de 70 km/h. Este tipo de autocarro encontra-se de momento em circulação nas cidades do Porto e Guimarães.

1.3. OBJETIVOS

O objetivo final é atingir o desenvolvimento de *software* para uma *human-machine interface* (HMI) entre os sistemas de aquisição de dados (veículo e rede viária) e o condutor de um autocarro elétrico. Para tal, será necessário:

- Definir qual a informação relevante e o respetivo nível de prioridade.
- Definir qual a forma mais adequada para apresentar a informação ao condutor.
- Criar um conjunto de máscaras num ecrã, que apresentem a informação desejada de uma forma coerente.

- Elaborar o *software* necessário ao desenvolvimento deste sistema.
- Definição de testes em banca – Depois de realizada a programação, definir métodos de teste e verificação de cumprimento dos requisitos inicialmente propostos.

1.4. IDENTIFICAÇÃO DO PROBLEMA

De modo a desenvolver um programa automático para aplicar numa HMI com o objetivo de transmitir mensagens de comunicação V2I (*Vehicle-to-Infrastructure*) para aplicação num autocarro elétrico, foi planeada a seguinte estrutura de desenvolvimento de projeto:

- Pesquisa/definição de mensagens a serem desenvolvidas e aplicadas como comunicação V2I de auxílio a condutor.
- Desenho de máscaras e estabelecimento de prioridades entre mensagens que concorram ao acesso do barramento CAN (*Controller Area Network*) durante o mesmo período de tempo.
- Codificação das mensagens em comunicação CAN.
- Desenvolvimento de *software* e respetiva ligação à HMI ao dispor.
- Realização de testes/simulações e possíveis correções.
- Aceitação final.

O desenvolvimento deste *software* terá como objetivo estabelecer uma comunicação real entre uma OBU (*Vehicle On-Board Unit*) e a interface HMI, instalada no autocarro, através de CAN. A interface em questão contém um *display* e está colocada na frente/topo ao lugar do condutor no autocarro, podendo este ter uma rápida e simples interpretação dos dados que estão a ser recebidos.



Figura 3: HMI Display (modelo HY-eVision2 7.0) a ser utilizada. (Foto captada pelo mestrando)

A Figura 3 faz referência ao *display* utilizado no âmbito desta dissertação. O modelo *HY-eVision2 7.0* foi adquirido pela entidade empresarial e permite a sua programação na ferramenta CoDeSys. O *menu* exibido corresponde ao *menu* principal da HMI, desenvolvido pela CaetanoBus. Este exibe informações de autonomia em km, consumos instantâneos e consumos médios em kWh/100 km. Por fim, exibe dados de energia regenerada e consumida em kW.



Figura 4: A vermelho encontra-se localizado o *display* em questão. (Foto captada pelo mestrando)

A Figura 4 realça a localização do *display* no autocarro.

1.4.1. REQUISITOS DO SISTEMA E CADERNO DE ENCARGOS

Como requisitos para o sistema funcionar corretamente é necessário:

- Codificar as mensagens que se pretende exibir na HMI em comunicação CAN (comunicação *in-vehicle*). Só após a codificação das mensagens é possível trabalhar o programa automático. Para tal, realizou-se um estudo sobre comunicação CAN (capítulo 3). A codificação de mensagens em CAN encontra-se desenvolvida no capítulo 6, ponto 6.3.
- Definir um tempo de ciclo que corra o programa automático. Este tempo de ciclo é definido pela empresa e quando implementado corretamente permite que o programa automático não gere bloqueios ou atrasos de execução, garantindo a segurança dos ocupantes do veículo. A sua especificação encontra-se no capítulo 6, ponto 6.5.

A entidade empresarial pediu como caderno de encargos:

- O desenvolvimento de uma biblioteca em que conste a codificação das mensagens em CAN (Anexo A e ponto 6.3 para a sua compreensão) .
- O desenvolvimento de um *software* que:
 - Após a receção de dados, permita ao *display* emitir um alerta sonoro para captar a atenção do condutor e que permita à mensagem em questão ser disponibilizada ao mesmo durante dez segundos, desaparecendo de seguida, voltando o *display* ao *menu* principal da HMI. Todo este processo deve ser realizado automaticamente através de *software*, nunca precisando o condutor em ocasião alguma tocar no *display* enquanto em condução.
 - Mostre a mensagem definida como mais prioritária (caso haja mais do que uma mensagem no mesmo período de tempo a pedir acesso de exibição na HMI) independentemente de em que *menu* ou *submenu* o *display* da HMI se encontrar.

Ao longo do capítulo 6 é possível constatar o desenvolvimento destes pontos.

1.4.2. ARQUITETURA DO SISTEMA

Para compreensão do leitor, a Figura 5 caracteriza um esboço genérico da arquitetura do sistema referente ao desenvolvimento deste projeto.

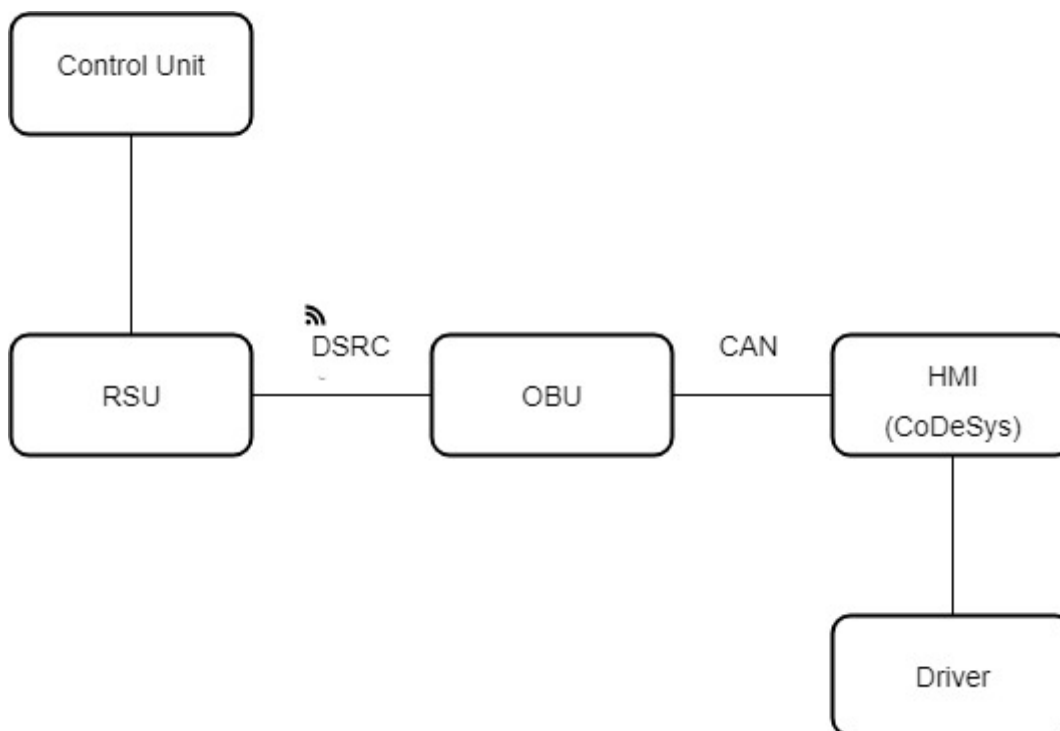


Figura 5: Arquitetura do sistema referente a este projeto (comunicação V2I)

A arquitetura do sistema desta dissertação compreende uma comunicação *vehicle-to-infrastructure* (V2I). A origem dos dados transmitidos parte de uma *Control Unit* que pode ser, por exemplo, um Centro de Controlo de Tráfego. Estes dados são transmitidos a RSUs que por sua vez, através de *Dedicated Short-Range Communications* (DSRC), partilham tais dados com a OBU instalada no autocarro. Através de uma comunicação *in-vehicle* (CAN) os dados são codificados em mensagens. Estas mensagens são trabalhadas no programa automático desenvolvido em CoDeSys – *Structured Text* e exibidas ao condutor no *display* da HMI em questão.

Existem alternativas tecnológicas para a transmissão *wireless* de dados, como a *cellular-V2X* referenciada no ponto 5.1, ou a possibilidade de programar a HMI em qualquer um dos estilos de programação CoDeSys (ver ponto 4.1). No entanto, visto o *menu* principal

(Figura 3) ter sido desenvolvido pela empresa em CoDeSys – *Structured Text*, foi pedido por esta a continuação da programação neste estilo.

A comunicação desde a *Control Unit* até à OBU é externa à empresa e, devido a tal, não será trabalhada.

1.5. CRONOGRAMA

Tendo por base as atividades desenvolvidas ao longo deste projeto, estas foram calendarizadas de acordo com a Figura 6. A coluna *ID* numera as atividades efetuadas. A coluna *Task Name* descreve as atividades realizadas e, por fim, as colunas *Início* e *Conclusão* fazem referência de datas. Como *ID1* tem-se o início do estágio na empresa e a definição do projeto desta dissertação. No *ID2*, foi realizado um estudo do problema em questão que permitiu realizar o enquadramento teórico desta dissertação. Os *ID3* e *ID4* correspondem ao desenvolvimento de *software* desde o seu ponto inicial até à sua aceitação final. Ao longo destas atividades foi realizada, paralelamente, a elaboração deste documento (*ID5*) e o *ID6* marca a conclusão do estágio na empresa.

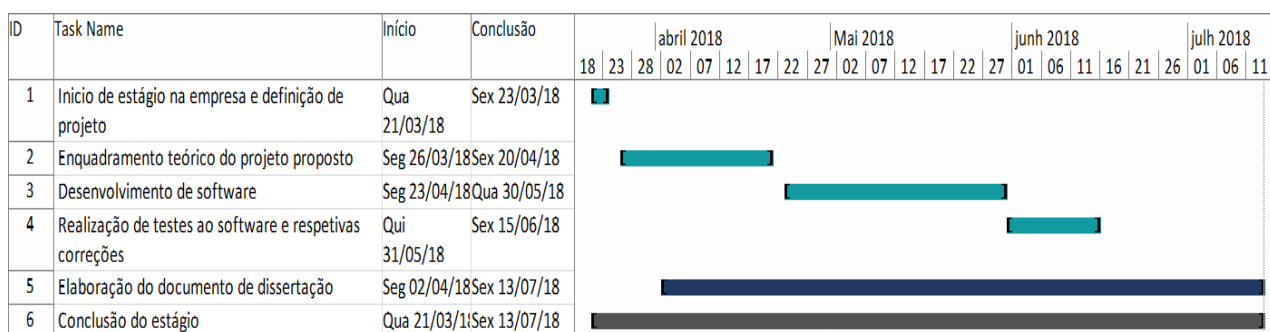


Figura 6: Calendarização do estágio de dissertação

1.6. ESTRUTURA

Este documento está segmentado em sete capítulos.

Neste primeiro capítulo é realizada a introdução da temática e uma breve apresentação da entidade empresarial e são identificadas algumas características constituintes do tema. É, também, identificado o problema proposto pela empresa, assim como o caderno de

encargos e os requisitos para o sistema funcionar, incluindo um esboço da arquitetura do sistema. Por fim, são apresentados os objetivos a atingir com o desenvolvimento do trabalho.

No capítulo número dois são analisadas questões ergonómicas entre o ser humano e os sistemas interativos que o rodeiam.

O terceiro capítulo aborda a forma como a troca de informação entre as unidades de controlo eletrónico e outros dispositivos dum veículo automóvel é realizada.

O capítulo número quatro faz uma pequena introdução ao *software* utilizado como ferramenta de trabalho no âmbito deste projeto.

O quinto capítulo analisa os diferentes tipos de comunicação, entre eles comunicações entre veículos, comunicações entre veículos e infraestruturas rodoviárias, comunicações entre veículos e pedestres. Realça ainda uns exemplos práticos já efetuados nos Estados Unidos da América.

O sexto capítulo diz respeito ao corpo do trabalho. É aqui explicada a codificação de mensagens, a atribuição de prioridades, o desenho de máscaras e, por fim, a criação do programa automático e os casos de teste.

Para finalizar, o capítulo número sete concluí este relatório e aborda perspectivas de trabalhos futuros.

CAPÍTULO 2

2. HMI – *HUMAN MACHINE INTERFACE*

No presente capítulo será abordado o conceito de HMI, como deve ser desenvolvida uma HMI eficiente, bem como a engenharia cognitiva inerente a este processo.

2.1. HMI E O SEU DESENVOLVIMENTO

Segundo a norma ISO 9241-110:2006 – *Ergonomics of human-system interaction* [1], o termo interface com o utilizador é tido como “todas as partes de um sistema interativo que fornecem informações e controlos necessários para que o utilizador realize uma determinada tarefa com o sistema interativo”. Pode-se, então, afirmar que *Human Machine Interface* (HMI) é um campo de estudo que se concentra na interação entre humanos e computadores/máquinas, englobando várias ciências como engenharias, ciências cognitivas, ergonomia, etc. É o ponto de ação em que o humano está em contacto com a máquina (ex: caixa de multibanco).

O termo HMI surgiu na década de 1980 com o objetivo de tornar a interação entre o humano e um computador numa espécie de diálogo aberto entre dois humanos e, desde então, evoluiu até aos dias de hoje (ex: *voice-user interface*).

Alguns aspetos que se devem ter em consideração aquando duma criação HMI são [2]:

- Elaborada tendo em conta as tarefas dos futuros utilizadores.
- Adaptada ao entendimento do utilizador.
- Revelar informação útil.
- Considerar primeiro a sua correta funcionalidade e só no fim o seu design.
- Autoexplicativa, sem exigir estudo prévio.

O objetivo dos programadores de uma HMI é criar uma interface “amiga do utilizador”, tendo, para tal, de entender o funcionamento futuro da máquina, bem como as necessidades do utilizador. Para entender tais necessidades, ao invés de exigir que o utilizador adapte as suas necessidades para aprender a utilizar uma ferramenta, são criados sistemas que relacionam as necessidades do utilizador com o seu meio envolvente → *User-Centered Design*.

2.2. USER-CENTERED DESIGN

O UCD – *User-Centered Design* (ou Design Centrado no Utilizador) é um design que se concentra nas necessidades do utilizador, tendo como objetivo produzir sistemas acessíveis e adaptados ao entendimento do mesmo, evitando assim aspetos negativos relativos à sua saúde e desempenho. É um processo que abrange fatores cognitivos como a perceção, memória, aprendizagem, etc., em relação à interação entre o utilizador e a máquina.

Eis alguns aspetos em que o programador se deve focar ao criar esta interface [2]:

- Quem serão os utilizadores desta interface?
- Quais os seus objetivos?
- Quais as suas tarefas?
- Qual será o nível de experiência do utilizador?
- Como pensará o utilizador do modo de funcionamento do sistema?
- Como poderá o design desta interface facilitar os processos cognitivos do utilizador?

De modo a desenvolver uma melhor compreensão das necessidades do utilizador, o UCD aplica uma mistura de métodos investigativos (ex: pesquisas e entrevistas), métodos criativos (ex: *brainstorming*) e ferramentas. O envolvimento dos utilizadores em todas as fases de design e avaliação é crucial.

Tabela 1: Exemplo de um possível envolvimento UCD [2]

Técnica	Objetivo	Estágio do ciclo de Design
Entrevistas e questionários	Recolher dados acerca das necessidades dos usuários; avaliações de alternativas de design	Início do projeto de design
Focus Groups	Incluir uma ampla gama de partes interessadas para discutir questões e requisitos	Começo/Desenvolvimento do projeto de design
Teste de usabilidade	Recolher dados relacionados com o critério de usabilidade mensurável	Final do projeto de design

Existem algumas considerações simples que as equipas de desenvolvimento podem observar sem trabalhar com os usuários finais. Uma dessas considerações são restrições ambientais e como o meio ambiente restringirá o design da interface. Tais restrições podem ser, por exemplo, a localização, o nível de ruído, a temperatura ou a acessibilidade. Tem-se como exemplos uma interface de controlo de voz impossível de usar num ambiente muito ruidoso, e/ou, ainda, uma interface de um interruptor de luz alojado no teto (que não teria usabilidade devido à sua localização).

Pode-se então subdividir uma abordagem UCD em quatro fases distintas [3]:

- Especificar o contexto de uso: identificar quem utilizará o produto, para o que o utilizará e sob que condições de utilização.
- Especificar requisitos: identificar requisitos de negócio ou metas dos utilizadores que devem ser atendidas para o negócio ser bem-sucedido.
- Criar soluções de design.
- Avaliação: Usualmente feita com testes de usabilidade com utilizadores reais.

O desenvolvimento de protótipos UCD deve ser feito com antecedência durante a fase do projeto, de modo que os testes de validação possam ser feitos para verificar o design da

interface. A interface só deve ser implementada após ter a certeza que o projeto corresponde aos requisitos.

2.2.1. A IMPORTÂNCIA DOS PROCESSOS UCD

Do ponto de vista do utilizador, a usabilidade é essencial pois pode marcar a diferença entre utilizar uma ferramenta de forma simples e precisa, aproveitando o processo, ou de não o conseguir e, conseqüentemente, desistir e/ou ficar frustrado. Do ponto de vista da equipa de desenvolvimento, a usabilidade é crucial, porque realça a diferença entre o sucesso ou o fracasso de um sistema. Obviamente que os aspetos referidos têm peso no sucesso ou insucesso de um negócio. Um sistema com pouca usabilidade pode reduzir a produtividade de uma equipa de trabalho para um nível inferior ao que se tinha sem usar tal sistema. Tal aspeto pode causar perda de tempo e prejuízo. De um modo geral, as pessoas tendem a adquirir/usufruir de sistemas de maior simplicidade, mesmo que não estejam cientes do que significa usabilidade.

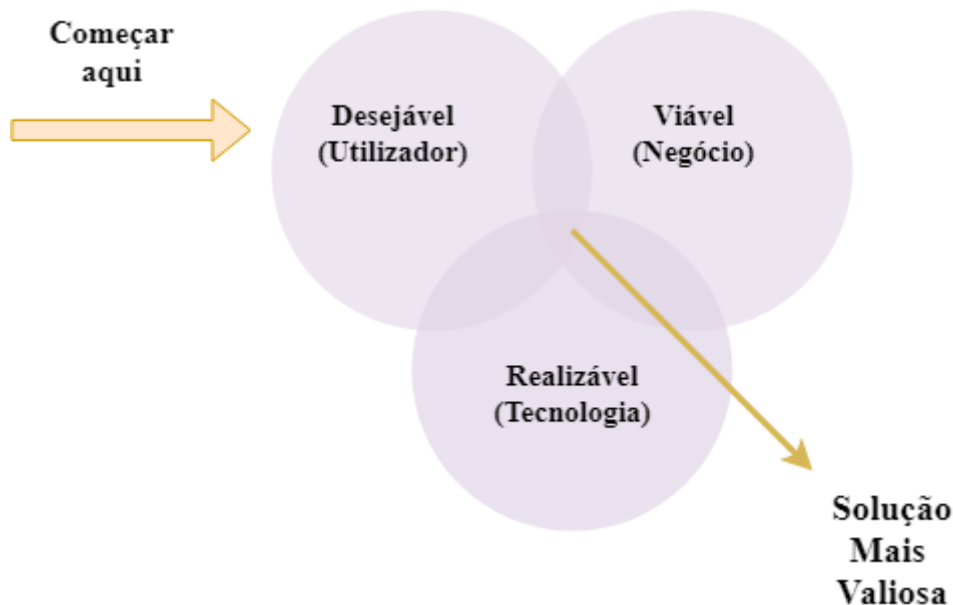


Figura 7: Ilustração de situação de maior valor num processo UCD

2.2.2. COMO MEDIR O SUCESSO DE UM PROCESSO UCD

O sistema interativo é uma forma relevante de aumentar a usabilidade de um sistema, pois avalia os estágios do projeto constantemente, contribuindo para um progressivo melhoramento dos processos. Desta forma, a equipa de desenvolvimento estará sempre incorporando o *feedback* dos utilizadores garantindo um melhoramento do sistema. É, para tal, importante testar os sistemas com os seus verdadeiros utilizadores. A concentração de esforços nos processos UCD é um dos melhores caminhos para atingir um elevado nível de usabilidade.

2.2.3. VANTAGENS E DESVANTAGENS DE UM PROCESSO UCD

Como em tudo, tais implementações admitem um conjunto de vantagens e desvantagens ao aplicar um processo UCD. Se, por um lado, dá à equipa de desenvolvimento um conhecimento aprofundado de aspetos psicológicos, organizacionais, fatores sociais e ergonómicos, por outro lado, também aumentará o custo de implementação e custo de tempo. Deve ser feita uma abordagem que realce os prós e contras dum processo deste género tendo em conta o tempo, custo e benefícios [4].

Tabela 2: Vantagens e desvantagens dum processo UCD

Vantagens	Desvantagens
Produtos mais eficientes e seguros	Aumento de custo
Assiste a expectativa de manuseamento dos utilizadores	Maior tempo despendido
Produtos requerem menos <i>redesign</i> e surgirão com maior facilidade no mercado	Pode ser difícil traduzir alguns tipos de dados num <i>redesign</i>
A colaboração entre equipas e utilizadores levará a um aumento de soluções, visto que são analisadas questões de equipa e de utilizadores simultaneamente	O produto pode ser demasiado específico para um uso mais geral, portanto, não é facilmente transferível para outros clientes; mais caro

2.3. A IMPORTÂNCIA DO DESIGN DA INTERFACE DE UTILIZADOR

O design de uma HMI pode afetar a capacidade de trabalho de um utilizador. O design é igualmente responsável pelo afetamento do utilizador quanto à tomada de decisões, compreensão da situação atual, além de supervisionar e fornecer comandos para o sistema computacional. Existe ainda uma variedade de fatores humanos que podem afetar todos os projetos HMI. Tais fatores estão relacionados com a tomada de decisões do utilizador, consciência da situação atual, vigilância, níveis de carga de trabalho e erro humano. Todos os aspetos referidos neste ponto devem ser tomados em consideração na criação de um sistema HMI. Nos pontos seguintes serão abordados tais aspetos e um meio de incorporar a identificação destes fatores na implementação de uma interface *human-machine*.

2.3.1. TOMADA DE DECISÃO HUMANA

Todos os dias o ser humano toma milhares de decisões. Tais decisões são tomadas em ambientes bastante diversificados e de uma maneira dinâmica. Dependendo de um conjunto de situações e da tarefa do humano, uma má tomada de decisão pode ter consequências drásticas. O piloto de um avião que tome uma má decisão ao descolar pode colocar muitas vidas em perigo. A compreensão dos processos de tomada de decisão de um ser humano deve, então, ser tomada em consideração na implementação de um projeto HMI para apoiar no processo de uso do utilizador. Existem estudos da tomada de decisão de um humano (ex: [*Sources of Power, Gary Klein 1998*]), que podem e devem ser aplicados ao desenvolvimento de técnicas de tomada de decisão e cooperação de equipas robóticas. Quanto mais complexo for o projeto HMI mais informação ao utilizador sobre tomada de decisões deve ser fornecida. De um modo geral, as equipas de desenvolvimento de HMI devem beneficiar da riqueza de informações disponíveis à tomada de decisões humana [2].

2.3.2. VIGILÂNCIA

Segundo Wickens e Hollands, [*Wickens & Hollands, Engineering Psychology and Human Performance, 2000*] a vigilância é tida como um conjunto de situações em que um operador tem de detetar sinais durante um longo período de tempo, sendo os sinais intermitentes, imprevisíveis e infrequentes. A manutenção da vigilância pode ser

interrompida por inúmeros fatores. Se uma informação de tarefa do operador for intermitente, este pode ficar aborrecido devido à falta de estimulação. Esta situação pode levar ao desgaste mental e/ou sonolência. Este é um exemplo de uma ocorrência, que pode afetar o processo de vigilância do ser humano. A excessiva divulgação de informação é um outro exemplo que pode impedir o ser humano de manter uma vigilância adequada. Existem vários casos em que projetos HMI requerem que o operador seja mantido em constante vigilância. Por outro lado, há também casos em que o operador simplesmente supervisiona o funcionamento de um sistema em geral. Ambos os casos requerem que uma equipa de desenvolvimento HMI forneça dados ricos acerca deste tema, aplique metodologias para evitar o risco e que o nível de vigilância dos operadores seja sempre tido em consideração.

2.3.3. CARGA DE TRABALHO

A carga de trabalho excessiva ou demasiado baixa pode levar a que o ser humano entre em reduções de vigilância ou sob estimulação, respetivamente. Cada ser humano tem um nível diferente de capacidade de processamento mental, que pode ser afetado pela falta de tarefas a completar, stress, falta de sono, condições ambientais, etc. A carga de trabalho deve ser equilibrada, de modo a permitir ao ser humano tomar decisões acertadas e estar em plena capacidade de funções. No caso de equipas HMI, a alocação de tarefas entre operador e sistema HMI é um aspeto importante a ter em consideração. As medidas de carga de trabalho mental devem ser empregadas não só para a alocação automática de tarefas, mas, também, no design e desenvolvimento inicial da HMI, para garantir que o operador não esteja rotineiramente sob ou sobre carregado [2].

2.3.4. CONSCIÊNCIA SITUACIONAL

Outro aspeto importante relativo aos fatores humanos é a consciência situacional – capacidade do operador de entender corretamente as atividades da equipa robótica no seu ambiente atual. Como exemplo, temos os estudos de conscientização situacional relacionados com incidentes de aeronaves [2]. No caso de projeto HMI, o operador permanece informado da sua perceção de situação e dos dados recebidos pelo sistema robótico. Se os dados recebidos não descrevem completamente a operação do sistema

robótico e o operador não estiver presente no espaço de trabalho da máquina, a consciência situacional é prejudicada. A equipa HMI deve tentar fornecer a informação apropriada a determinadas situações, distribuir tarefas entre diferentes operadores e estar ciente das capacidades de trabalho dos operadores.

2.3.5. ERRO HUMANO

O erro humano é comum em várias causas de acidentes. Este pode estar associado a fatores externos como o ambiente em redor (clima), ou com fatores internos como o cansaço mental. O erro humano é tido como uma ação inadequada passível de causar um erro. Pesquisas inerentes ao uso de HMI discutem o benefício ou malefício do uso de máquinas em trabalho de cooperação com humanos. Em muitos casos, máquinas ou robôs estão a trabalhar com a vida de seres humanos (ex: situação de resgate). O erro humano deve ser uma das principais preocupações nestes casos. Deve ser tido em consideração o perigo que tais máquinas/robôs podem apresentar relativamente à vida do ser humano, mas, também, os casos de falhas causadas em sistemas robotizados por erro humano.

2.3.6. ENGENHARIA COGNITIVA

Devido à dificuldade que as equipas de desenvolvimento de sistemas computacionais enfrentaram, quando precisavam de considerar a relação entre os sistemas que concebiam e os futuros utilizadores, a área de fatores humanos tem vindo a desenvolver teorias quanto ao comportamento humano adequado a sistemas interativos. De modo a entender as habilidades humanas de aprendizagem, memória e resolução de problemas nos sistemas de interação, têm sido consideradas abordagens cognitivas da natureza do utilizador.

Foi então que Donald Norman [5] desenvolveu a “teoria da ação”, que pretende entender como o ser humano se relaciona com sistemas computacionais. O ser humano tem metas definidas em termos relevantes ao próprio ser humano (psicológico) e metas (físicas) relevantes aos mecanismos dos sistemas computacionais. Tal teoria, dá o nome à discrepância entre as variáveis psicológicas e físicas de “golfos”. A equipa de desenvolvimento de sistemas computacionais tem como objetivo encurtar tais “golfos”, para que sirvam de aproximação entre o utilizador e a interface, adaptando dessa forma a interface às necessidades do utilizador.

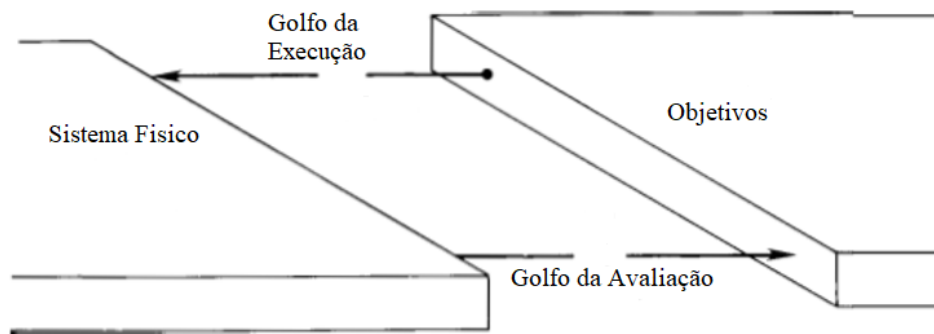


Figura 8: Os "golfos" da Execução e da Avaliação[5]

De maneira a encurtar tais “golfos”, Norman atribuiu seis etapas:

- “Golfo” da Execução – intenção; especificação da ação; execução.
- “Golfo da Avaliação – percepção; interpretação; avaliação.

O “golfo” da execução compreende todo processo mental para planejar as ações diante do sistema.

O “golfo” da avaliação compreende a avaliação do utilizador face aos objetivos pretendidos.

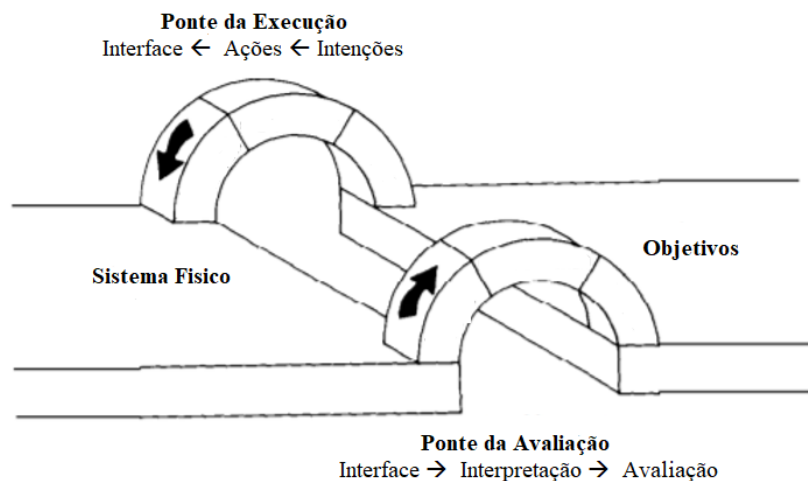


Figura 9: Etapas de ligação entre "golfos"[5]

Cabe à equipa de desenvolvimento de sistemas interativos estudar e compreender o modelo mental dos utilizadores de uma interface de um modo geral. A engenharia cognitiva é um processo de entendimento de interação utilizador-sistema e o estudo da teoria em questão possibilita à equipa de desenvolvimento criar um modelo mental adequado ao sistema pretendido mais próximo das expectativas do utilizador. A Figura 10 revela um exemplo prático da Teoria da Ação.

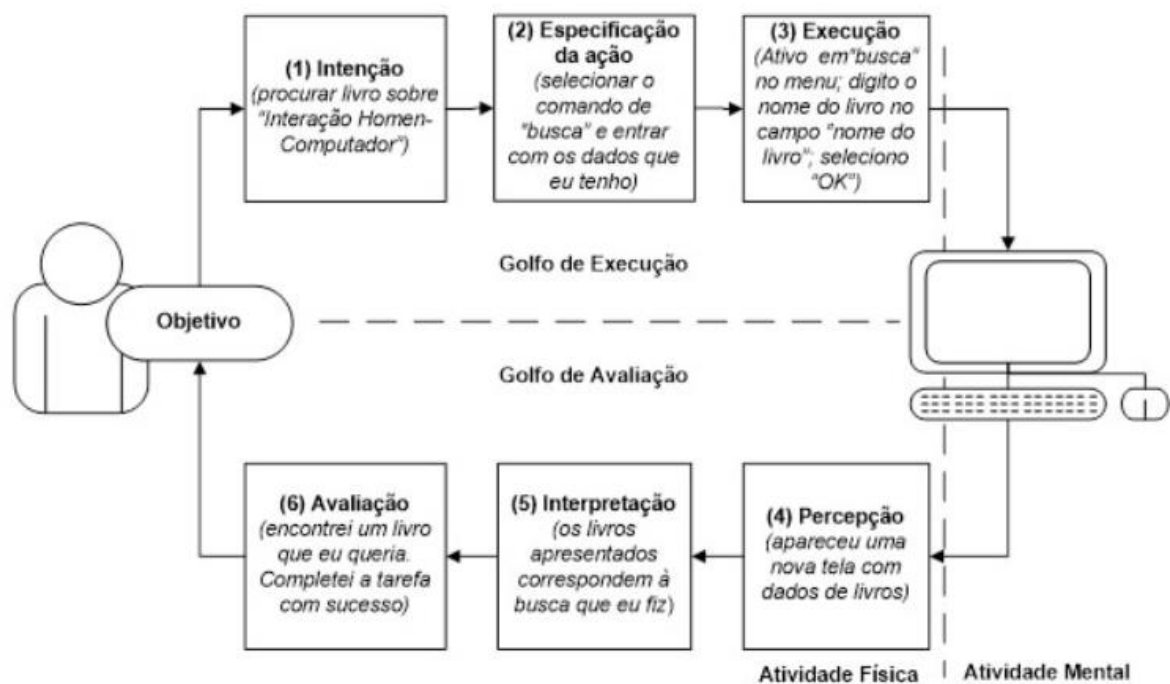


Figura 10: Estágios de atividade do usuário na travessia dos "golfos" - Exemplo adaptado [6]

CAPÍTULO 3

3. COMUNICAÇÃO CAN

Desenvolvido pela empresa Robert Bosch GmbH, em 1986, o protocolo de comunicações CAN (Controller Area Network) permite a troca de informação entre as unidades de controlo eletrónico e entre outros dispositivos. Por exemplo, considerando um automóvel, a unidade de controlo eletrónico que realiza o controlo do motor de combustão interna é responsável pelo cálculo das quantidades necessárias de injeção de combustível, os tempos de ignição e controla, igualmente, a temperatura do fluido refrigerante, informação que é enviada para o indicador de temperatura do painel de instrumentos.

A aplicação desta tecnologia para o controlo em tempo-real e a partilha de dados é utilizada em grande escala, sendo considerada uma boa solução para comunicar em rede de uma forma simples, barata e com características imunes ao ruído eletromagnético. Na comunicação CAN, mensagens como a temperatura ou, rotações por minuto (RPM), dum motor são transmitidas pela rede inteira, permitindo dados consistentes em todos Nós do sistema. O protocolo CAN define de que modo a informação é transferida entre dispositivos e é aprovado pela *International Standard Organization* (ISO). A ISO aprova-o como barramento *Standard* para redes de elevada velocidade (≥ 125 kbit/s) na partilha de informação em tempo-real em veículos automóveis. Após a aprovação da ISO, o CAN sofreu uma enorme aceitação por parte da indústria automóvel.

3.1. CARACTERÍSTICAS DO CAN

Hoje em dia, os protocolos de sistemas de rede em veículos automóveis são de extrema importância devido ao aumento de periféricos inteligentes, que permitem ajuda e controlo nos veículos automóveis. Os sistemas de rede apresentam a enorme vantagem da partilha de dados entre várias unidades, com elevado nível de segurança.

É um sistema em barramento, em que vários Nós podem pedir acesso ao meio de transmissão em simultâneo (*multi-mestre*). Contém também o conceito de *multicast*, isto é, permite que uma mensagem seja transmitida a um conjunto de recetores simultaneamente.

Nas redes CAN as mensagens possuem um identificador e são enviadas através de um emissor a todos os Nós CAN que "filtram", com base no identificador recebido, se a mensagem deve ou não ser processada. Para além disso, o identificador determina também a prioridade da mensagem no acesso a um barramento (as prioridades estão pré-definidas).

Uma mensagem CAN pode conter um máximo de 8 *bytes* de informação útil. A taxa máxima de transmissão especificada pode ser:

- 1 *Mbit/s* para distâncias até 40 m.
- 500 *Kbit/s* para distâncias até 100 m.
- 250 *Kbit/s* para distâncias até 250 m.
- 125 *Kbit/s* para distâncias até 500 m.
- 50 *Kbit/s* para distâncias até 1 km.

Nos próximos pontos será realizada uma análise ao protocolo CAN que justificam a aceitação deste tipo de comunicação *in-vehicle*.

3.2. MÉTODO DE ENDEREÇAMENTO

Para transmitir dados em comunicação CAN, o método tradicional de "remetente" e "destino" não se aplica. Por sua vez, os dados que se pretendem enviar têm um identificador único, sendo cada Nó da rede responsável por aceitar ou declinar a mensagem, realizando para tal um teste de aceitação ao identificador da mesma.

A tal teste chama-se de "*frame acceptance filtering*", usualmente tratado por "filtragem". Para além do identificador levar consigo o conteúdo da mensagem que pretende passar estabelece, também, a prioridade da mesma. Esta prioridade terá a capacidade de decisão quando várias estações competem pelo acesso a um barramento.

O método para transmitir e receber mensagens CAN consiste no seguinte: Se o *Central Processing Unit* (CPU) de um dos Nós pretender enviar uma mensagem, este prepara os dados da mesma mensagem e o respetivo identificador para o CAN ("Preparar"). A mensagem é, então, enviada pelo controlador CAN ("Enviar mensagem"). Todos os restantes Nós na rede CAN tornam-se recetores ("Receber mensagem"), fazendo então uma triagem para determinar se os dados recebidos são ou não relevantes para essa estação

("Seleção"). Se os dados se revelarem necessários serão processados ("Aceite"), caso contrário serão rejeitados ("Ignorar").

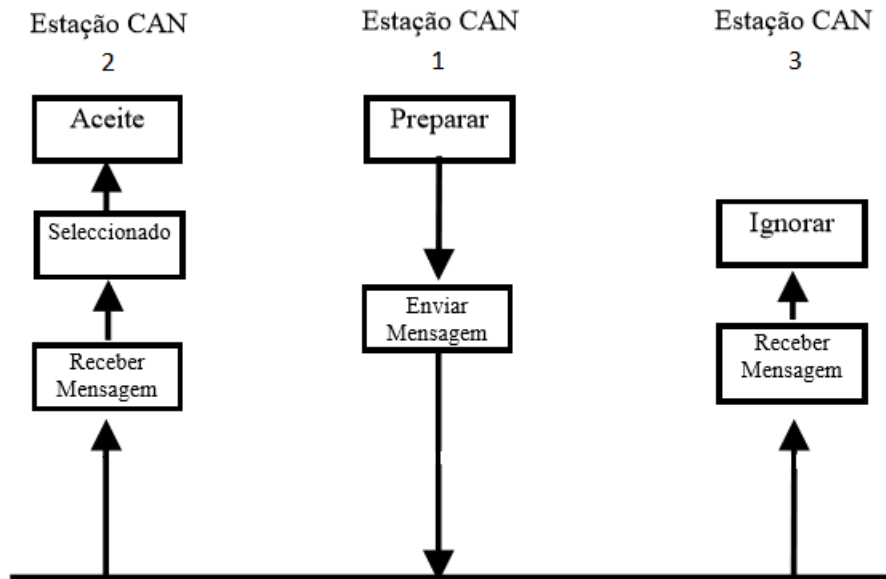


Figura 11: Método de endereçamento em CAN [7]

3.3. PROCESSO DE ARBITRAGEM

Para que os dados sejam processados em tempo real, estes devem ser transmitidos rapidamente. Isso exige que os dados tenham uma alocação rápida de barramento quando várias estações desejam enviar mensagens simultaneamente. A prioridade das mensagens transmitidas é especificada pelo identificador das respectivas mensagens.

Em CAN a atribuição de maior prioridade na transmissão de mensagens é definida pelo identificador de menor valor numérico (binário). Os conflitos de acesso a barramentos entre mensagens são resolvidos por arbitragem *bit a bit* nos identificadores. Este processo de arbitragem é resolvido da seguinte maneira:

- Enquanto é transmitido o identificador, cada Nó monitora o barramento série.
- Se o *bit* for recessivo, (nível lógico 1), e for diagnosticado um *bit* dominante, (nível lógico 0), o Nó desiste da transmissão e inicia a recepção dos dados que chegam.
- O Nó que transmite o objeto com menor identificador ganha acesso ao barramento e continua a transmissão.

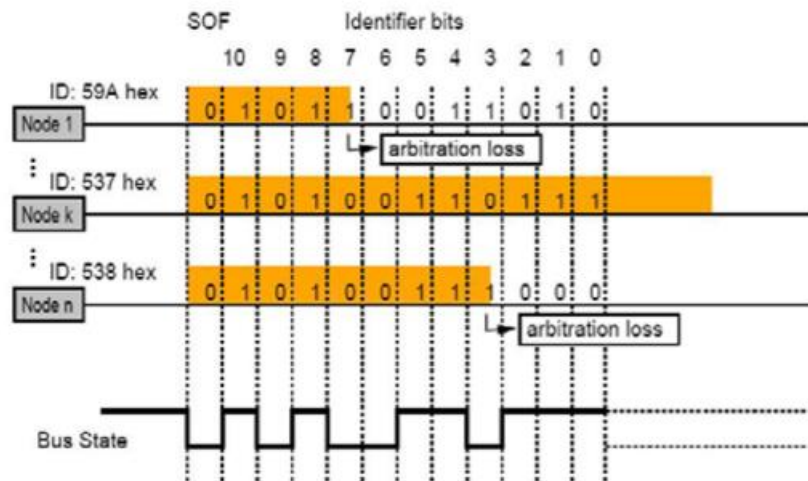


Figura 12: Processo de arbitragem do CAN [8] [9]

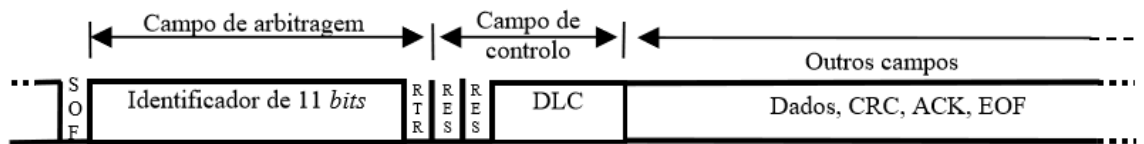
“O caso descrito mostra três Nós (l, k, n) a tentar enviar uma mensagem a dada altura. No *bit 7*, o Nó l percebe que enviou um *bit* recessivo, mas leu um *bit* dominante e desiste de enviar a sua mensagem – fica em modo de escuta apenas. No *bit 3*, o Nó n também entende que enviou um *bit* recessivo, mas leu um *bit* dominante. Também este Nó desiste de transmitir a mensagem. Desta forma, o Nó k prevalece e transmite a sua mensagem.” [9]

Desta forma, o acesso a barramentos é estabelecido através de prioridades, permitindo que a informação prioritária seja recebida em primeiro lugar. (Ex: carga do motor) tem que ser transmitida com mais frequência e, portanto, com menos atrasos que outras dimensões (ex: temperatura do motor) que mudam de forma relativamente lenta. [10]

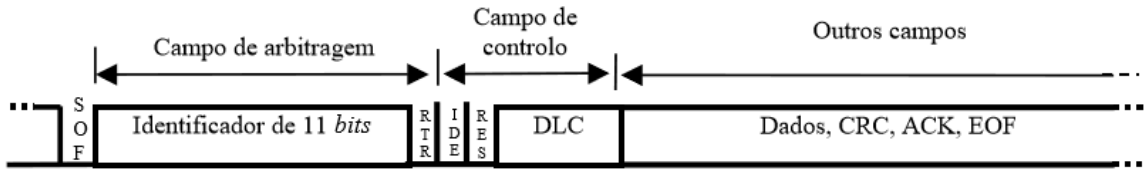
3.4. FORMATO DAS FRAMES

As *frames* (também conhecidas como mensagens) na comunicação CAN podem possuir um identificador de:

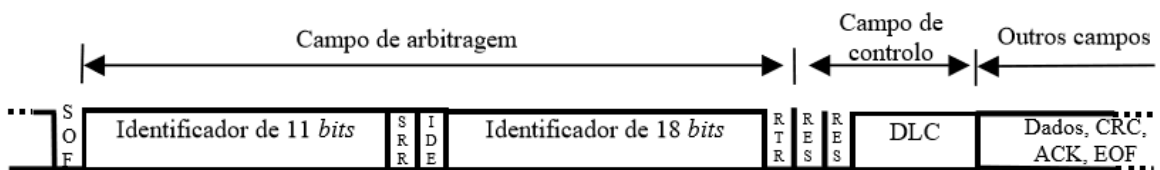
- 11 bits (versões 1.0; 1.1; 1.2; 2.0A).
- 11 *bits* (versão 2.0B) Formato *standard*, que se diferencia da versão 2.0A por possuir um bit de extensão.
- 29 *bits* (versão 2.0B) Formato *extended*.



(a) Formato duma *frame* de dados segundo as versões 1.0, 1.1, 1.2 e 2.0A.



(b) Formato duma *frame* de dados segundo a versão 2.0B (Formato *standard*).



(c) Formato duma *frame* de dados segundo a versão 2.0B (Formato *estendido*).

Figura 13: Diferentes tipos de formatos de frames do CAN [11]

Analisando a Figura 13, verifica-se que todos os formatos de *frame* disponíveis em CAN começam com o *bit Start of Frame* (SoF) seguindo-se 11 *bits* de identificação. Existem, também, os *bits* de controlo do campo de arbitragem que são:

- *Remote Transmit Request* (RTR) - distingue uma *frame* remota (*remote frame*) de uma *frame* de dados (*data frame*). O *bit* RTR dominante (nível lógico 0) corresponde a uma *frame* de dados. Um *bit* RTR recessivo (nível lógico 1) indica uma *frame* remota.
- *Substitute Remote Request* (SRR) - *bit* recessivo. Este *bit* substitui a posição do *bit* RTR numa *frame* de formato *extended*.
- *Identifier Extension* (IDE) - distingue *frames standard* (*bit* recessivo) de *frames extended* (*bit* dominante).

Assim, numa rede em que vários Nós iniciam uma transmissão de mensagens simultânea com o mesmo identificador aplicam-se as seguintes regras:

- *Frames* de dados são prioritárias em relação a *frames* remotas.
- *Frames standard* são prioritárias em relação a *frames extended*.

De modo a transmitir *frames* de ambos formatos, os controladores CAN requerem algumas regras em termos de implementação:

- Controladores CAN 2.0B ativos, recebem e transmitem ambos formatos de *frames*.
- Controladores CAN 2.0B passivos, transmitem e recebem *frames standard* e ignoram *frames extended*.
- Controladores CAN 1, trabalham com formatos *standard* e geram erro com formatos *extended*.

3.5. TIPOS DE *FRAMES*

As transferências de mensagens numa rede CAN são controladas através de quatro diferentes tipos de *frames*:

- *Frame* de dados.
- *Frame* remota.
- *Frame* de erro.
- *Frame* de sobrecarga.

3.5.1. *FRAME DE DADOS*

A Tabela 3 corresponde a uma *frame* de dados, i.e., unidades de comunicação entre as camadas de ligação de dados dos vários Nós.

Tabela 3: *Frame de dados do CAN*

Início de frame	Campo de arbitragem	Campo de controlo	Campo de dados	Campo CRC	Campo ACK	Fim de frame
-----------------	---------------------	-------------------	----------------	-----------	-----------	--------------

- Início de *frame* - A *frame* de dados começa no *bit Start of Frame (SoF)* através da transmissão de um *bit* dominante.
- Campo de arbitragem – é um campo de identificação (onze *bits* no formato *standard*; vinte e nove *bits* no formato *extended*), que identifica a prioridade e conteúdo das mensagens através do *bit* RTR, já mencionado em 3.4.

- Campo de controlo – começa com dois *bits* reservados (RES) seguido pelo *Data Length Code* (DLC). O DLC contém quatro *bits*, pode assumir qualquer valor num intervalo entre [0,8] e indica o número de *bits* do campo de dados.
- Campo de dados – contém a informação a ser transmitida. O seu tamanho pode variar entre zero e oito *bytes*.
- Campo CRC – *Cyclic Redundancy Checksum* (CRC), composto pela sequência CRC (quinze *bits* usados para a verificação de erros das *frames* recebidas) seguido do delimitador CRC (um único *bit* recessivo).
- Campo ACK – *Acknowledge* (ACK) é um campo de reconhecimento composto por dois elementos distintos:
 - *ACK Slot*, um único *bit* recessivo transformado num *bit* dominante aquando duma receção sem erros da sequência CRC.
 - *ACK Delimiter* composto sempre por um único *bit* recessivo.
- Fim de *frame* – a *frame* de dados termina com o *end of frame* (EoF), feita de uma sequência fixa de sete *bits* recessivos [8] [12].

3.5.2. **FRAME REMOTA**

Inicia a transmissão de informação pedida a um determinado Nó de origem através de um pedido de dados. Disponível apenas no CAN clássico, tem a mesma estrutura de campo que a *frame* dados, mas sem um campo de dados.

3.5.3. **FRAME DE ERRO**

Os erros detetados no sistema são indicados através de uma *error frame*. Esta consiste numa *error flag* constituída por seis *bits* do mesmo nível lógico seguidos de um *error delimiter* constituído por oito *bits* recessivos. Quando um Nó deteta um erro, este descarta imediatamente a *frame* transportada e envia uma *error flag*.

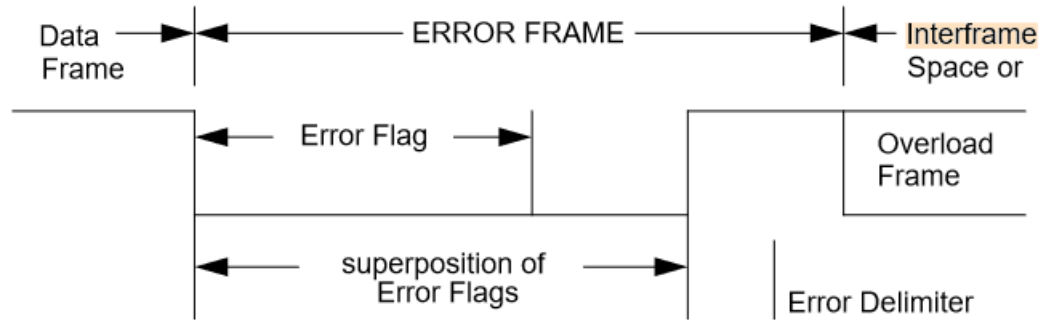


Figura 14: Frame de erro do CAN [13]

Existem dois tipos de *error flag*:

- *error flag* ativa: consiste em seis *bits* consecutivos dominantes.
- *error flag* passiva: consiste em seis *bits* consecutivos recessivos, a menos que sejam substituídos por *bits* dominantes de outros Nós.

Após a transmissão de *error flag* cada estação envia *bits* recessivos e monitoriza o barramento até ser detetado um *bit* recessivo, sendo depois transmitidos mais sete *bits* recessivos.

Os diferentes tipos de erros em comunicação CAN serão abordados no ponto 3.8.2 [13][12].

3.5.4. FRAME DE SOBRECARGA

A *frame* de sobrecarga (*overload frame*) tem a mesma estrutura da *frame* de erros, i.e., *flag* de sobrecarga e sobrecarga *delimiter*, residindo a única diferença na *flag* de sobrecarga que é constituída sempre por seis *bits* dominantes consecutivos.

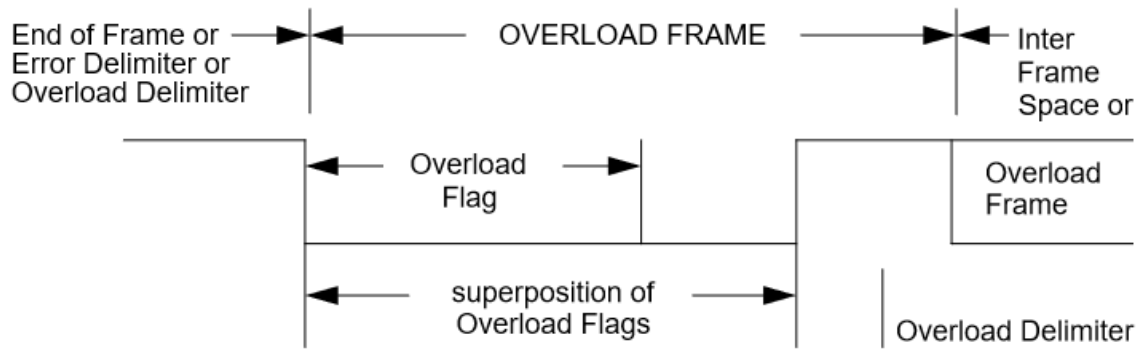


Figura 15: Frame de sobrecarga do CAN [13]

As *frames* de sobrecarga podem aparecer devido a razões externas: deteção de um *bit* dominante durante um intervalo de três *bits* recessivos; um recetor deteta um *bit* dominante no último *bit* do final do EoF. Pode também surgir por solicitação, devido à necessidade de um atraso extra por parte do circuito recetor interno (esta função foi a originalmente introduzida na criação deste tipo de *frame*). Uma diferença entre a *frame* de sobrecarga e a *frame* de erro é que a primeira é considerada como recebida corretamente. Após a transmissão de uma *flag* de sobrecarga, todos os Nós escutam o barramento até ser detetado um *bit* recessivo. Um sobrecarga *delimiter* funciona do mesmo modo que um *error delimiter*.

3.6. INTERVALO ENTRE *FRAMES*

Contém o tempo requerido pelo controlador para mover corretamente uma *frame* recebida para sua posição correta. A este espaço dado entre *frames* dá-se o nome de *interframe space*. Este intervalo apenas é precedente para *frames* de dados e *frames* remotas. O *interframe space* contém os campos intermissão (*intermission*) e barramento livre (*bus idle*).

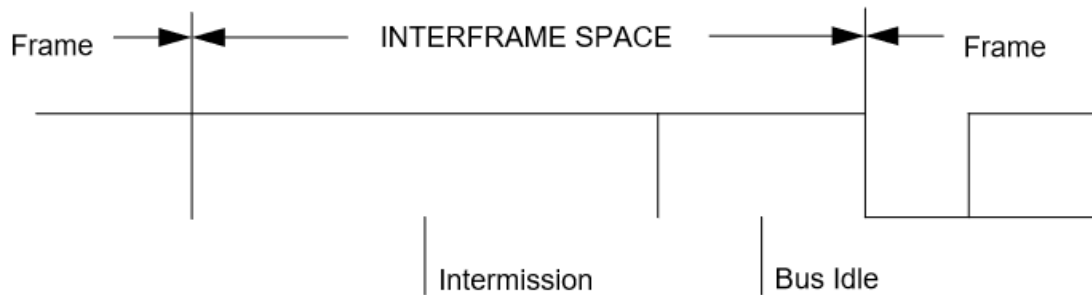


Figura 16: Intervalo entre frames do CAN [13]

- Campo de Intermissão (*intermission*) – consiste em três *bits* recessivos. Durante este campo não é permitido transmitir *frames* de dados nem remotas, sendo a única exceção a ocorrência de uma *frame* de sobrecarga.
- Campo de barramento livre (*bus idle*) – o barramento é estabelecido como livre e qualquer Nó com pretensão para transmitir pode ter acesso ao mesmo. Uma mensagem que esteja pendente é iniciada no primeiro *bit* após o campo de intermissão. A detecção de um *bit* dominante durante o campo de barramento livre é interpretada como início de *frame*, SoF.

3.7. CODIFICAÇÃO DE *FRAMES*

De maneira a evitar a transmissão de uma longa sequência de *bits* de idêntica polaridade, as *frames* de dados/remotas são sujeitas ao chamado *bit-stuffing*¹, que previne mais de cinco *bits* sequenciais de igual polaridade serem transmitidos, através de uma inserção automática de um *bit* extra.

O *bit-stuffing* [8] começa no início da *frame*, SoF, e termina no último dos quinze *bits* da *CRC sequence* (ver ponto 3.5.1). O *CRC delimiter*, campo ACK, e fim de *frame* (EoF) utilizam uma forma fixa de inserção de *bit*, dispensando o *bit-stuffing*. As *frames* de erro/sobrecarga também utilizam exclusivamente o método de forma fixa. Segue-se ilustrado um exemplo de codificação *bit-stuffing*.

¹ *Bit-stuffing*: De maneira a manter a sincronização, um *bit* de polaridade oposta é inserido após cinco bits consecutivos da mesma polaridade.

Tabela 4: Exemplo de codificação bit-stuffing em CAN
r – bit recessivo; *d* – bit dominante.

	Sequência a transmitir	Sequência transmitida
<i>Bit-stuffing</i>	r d d d d d x y	r d d d d d r x y
	d r r r r r x y	d r r r r r d x y
x, y ∈ {d, r}		

3.8. TRATAMENTO DE ERROS

Regra geral, os erros são sinalizados assim que são detetados. Um extenso conjunto de metodologias de deteção de erros, descritas de seguida, é usado no CAN para reforçar a confiabilidade da transferência de mensagens.

3.8.1. SINALIZAÇÃO DE ERRO

A sinalização de erro em CAN é executada da seguinte maneira:

- O Nó que deteta o erro inicia a sinalização através de uma *error flag*.
- Se existirem Nós que ainda não detetaram o erro, a *error flag* levará a uma *condition error* em todos Nós.
- Como consequência todos os Nós passarão a transmitir uma *error flag*.
- No máximo surgirão duas *error flags* na rede derivadas de todos os Nós, antes do *delimiter error* correspondente (Figura 17).

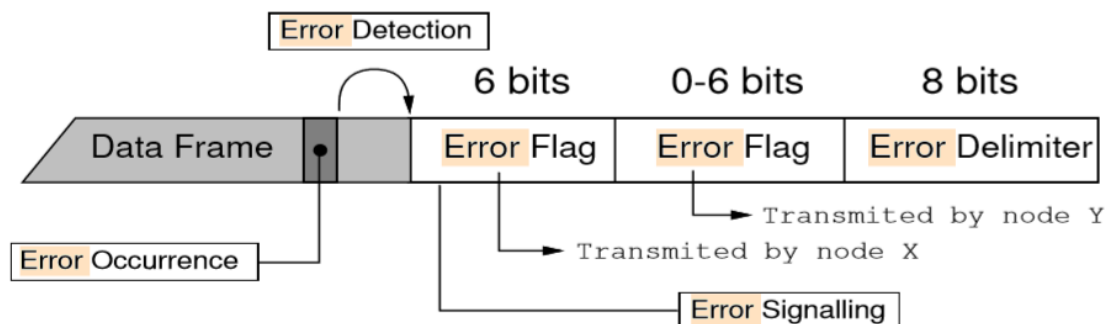


Figura 17: Formato de uma frame de erro em CAN [8]

3.8.2. DETEÇÃO DE ERRO

Um conjunto de cinco mecanismos são utilizados em CAN para a detecção de erro. Estes incluem [12]:

- *Bit error*: quando um Nó escreve uma sequência de *bits* no barramento, ele lê o estado do barramento ao mesmo tempo. Quando o Nó lê do barramento um estado diferente do estado do *bit* que ele tenta transmitir, deteta um erro de *bit*;
- *Bit-stuffing error*: quando um recetor deteta uma sequência de mais do que 5 *bits* da mesma polaridade nas zonas da mensagem protegidas pela regra de *bit-stuffing*;
- *CRC error*: o transmissor calcula o campo de CRC sobre as zonas da *frame* protegidas e envia essa informação na mensagem. Os recetores voltam a calcular o CRC e comparam o resultado com o campo que receberam. Nos Nós em que o resultado não corresponda ao CRC recebido na mensagem, esta é descartada, gerando uma *frame* de erro;
- *Frame check error*: Este mecanismo verifica a estrutura das *frames* dados/remota, verificando os campos de bits em relação ao formato fixo e ao tamanho da *frame*. Os erros detetados pelas verificações de estrutura são designados por “*format errors*”;
- *ACK error*: Detetado quando o emissor não deteta um *bit* dominante durante o campo ACK.

3.9. NORMALIZAÇÃO

A norma ISO 11898-1: 2015 especifica as características de configuração entre informação digital e os *layers* de dados CAN. Desenvolvido a pedido da comunidade industrial CAN, foi oficialmente apresentada na 13ª conferência CAN. Após a apresentação da ISO como norma internacional, o protocolo CAN foi melhorado em termos de confiabilidade e teve um incremento de adoção a nível industrial.

3.10. APLICAÇÕES E CONCLUSÃO

Apresentadas as características fundamentais da comunicação CAN, justifica-se a aplicação deste protocolo sobretudo no automóvel (*in-vehicle*) pelas suas capacidades

reais, funcionamento em tempo real e tolerância a falhas. Para além das características já discutidas, o baixo custo e o facto de ser um *standard* ISO revelam o interesse no CAN por parte das engenharias de automação.

CAPÍTULO 4

4. SOFTWARE CODESYS

O *Controller Development System* (CoDeSys) é um produto desenvolvido pela empresa alemã *3S-Smart Software Solutions*. Este foi projetado para o mercado dos Controladores Lógicos Programáveis (PLC). Está em conformidade com a norma IEC 61131-3 (*standard* sobre linguagem de programação PLC), suportando todas as cinco diferentes línguas de programação deste *standard*, combinando a utilidade das línguas de programação tipo C ou Pascal com as funções operacionais tipo PLC. Tem como vantagem a personalização rápida, a facilidade de uso e o alto desempenho. É utilizado por empresas como Mitsubishi, BECKHOFF, ABB, Schneider, SEW, Bosch, etc [14].

4.1. PROGRAMAÇÃO

O Codesys disponibiliza linguagens de programação personalizadas ou pré-definidas. Tem a particularidade de desenvolver um único programa através de subprogramas com linguagens de programação diferentes, sem qualquer tipo de problema. De seguida serão apresentados os diferentes estilos de programação Codesys, todos eles definidos na norma IEC 61131-3 [15]:

- *Structured Text* (ST) – estilo de programação estruturado em sintaxe semelhante à linguagem C+ ou Pascal. Existe um número variado de comandos pré-definidos, que permitem a criação de código de alta qualidade e arquiteturas de *software* complexas. Será o tipo de estilo utilizado no âmbito desta dissertação.
- *Function Block Diagram* (FBD) – estilo de programação altamente visual e gráfico. Contém todas as funções definidas na norma IEC 61131-3. A Codesys inclui uma grande biblioteca de blocos de funções predefinidas e permite também ao utilizador criar os seus próprios blocos caso seja esse o interesse.
- *Ladder Diagram* (LD) – Estilo de programação gráfica bastante utilizada em automação industrial.
- *Instruction List* (IL) – Lista de comandos. É uma subcategoria de linguagens de programação altamente simbólica. É a forma mais ilustrativa de números de código binário compreendidos por computadores.

- *Sequential Function Chart* (SFC) – Desenvolvido para gerar facilmente uma estrutura de programa. As funções estão listadas como etapas e as funções são executadas quando essa etapa está ativa.

O Codesys dispõe ainda de um editor gráfico adicional disponível, não definido na norma IEC 61131-3: *Continuous Function Chart* (CFC).

4.2. RUNTIME SYSTEM

De modo a programar e projetar um dispositivo em Codesys de acordo com a IEC 61131-3 é necessário utilizar o Control SoftPLC (Codesys), capaz de converter um computador num controlador industrial compatível com a IEC 61131-3. Para além disso, este sistema inclui funcionalidades adicionais para que o controlador possa comunicar com outros componentes de automação. O Control SoftPLC já vem implementado em todos os controladores listados na plataforma Codesys, para que o utilizador final possa programar imediatamente com Codesys [16].

4.3. CODESYS VISUALIZATION

Desenvolve uma interface HMI e a correspondente aplicação em PLC num único sistema de desenvolvimento IEC 61131-3. Contém as seguintes características [17]:

- Ecrãs de visualização de diferentes complexidades com base nos elementos disponíveis;
- Desenvolvimento da visualização da interface em paralelo com o PLC;
- Acesso direto a todas as variáveis de aplicação, devido à integração total no Sistema de Desenvolvimento IEC 61131-3;
- Os ecrãs de visualização criados podem ser reutilizados sempre que desejado;
- Vasta funcionalidade devido à integração no software de desenvolvimento IEC 61131-3;

4.4. *CODESYSFIELDBUS*

Para além da funcionalidade de um sistema de desenvolvimento IEC 61131-3 para programação de controladores industriais, a Codesys também oferece um amplo suporte de configuradores de redes de campo, entre eles: PROFIBUS, PROFINET, EtherCAT, CANopen, J1939, EtherNet/IP, Sercos, Modbus, IO-Link e BACnet. Deste modo, apresenta o benefício de ser uma ferramenta única para a programação de diferentes tipos de configuradores de redes de campo, reduzindo claramente o risco de erros, ser uma configuração simples e a possibilidade de implementar engenharia rápida [18].

4.5. *CODESYS MOTION + CNC*

Em contraste com as soluções clássicas de movimento, o Codesys SoftMotion fornece a funcionalidade de movimento necessária na forma de um *toolkit* de ferramentas diretamente integrado no sistema de desenvolvimento do PLC, permitindo desenvolver as tarefas de movimento/CNC/robô sem sair do ambiente familiar de desenvolvimento de um controlador lógico. As funcionalidades passam por: Câmera eletrônica; Controlo Multidimensional CNC e Controlador de robô de eixo múltiplo. [19].

CAPÍTULO 5

5. COMUNICAÇÃO *VEHICLE-TO-X*

Graças aos avanços tecnológicos ocorridos durante a última década no ramo da indústria automóvel, houve uma revolução na tecnologia dos veículos rodoviários. Hoje em dia, um veículo moderno pode conter cerca de cem microcomputadores e mais de um milhão de linhas de código, que permitem auxiliar o condutor por meio de operação, conforto, segurança e assistência do veículo. Decorre neste momento a tentativa de harmonizar a chamada *Vehicle-to-X*² *communication*, (V2X), que consiste em passar informações entre um veículo e uma entidade e vice-versa. Já é possível a condução de veículos de forma totalmente autónoma [20] e, num futuro não muito distante, os veículos nas estradas públicas poderão circular sem condutor. As melhorias tecnológicas permitem tornar viável este cenário, enquanto questões políticas, éticas e legais estão a ser discutidas.

O V2X é um sistema de comunicação veicular e engloba tipos específicos de comunicação como:

- *Vehicle-to-Vehicle* (V2V);
- *Vehicle-to-Infrastructure* (V2I);
- *Vehicle-to-Pedestrian* (V2P);

Todos os pontos referidos anteriormente serão analisados de seguida.

5.1. *VEHICLE-TO-VEHICLE* (V2V)

As comunicações V2V consistem numa rede *wireless*, onde veículos automóveis trocam informações entre si sobre o que estão a fazer em tempo real. Tais dados trocados podem conter velocidade, localização, direção de deslocamento, travagem, etc. De momento as tecnologias V2V mais avançadas são a *Cellular-V2X*³ e *Dedicated Short-Range Communications*⁴ (DSRC) [21]. O *cellular-V2X* está ainda em processo de normalização e espera-se que seja regulamentado entre 2018 e 2019. A DSRC é um padrão estabelecido

² *Vehicle-to-X* também é usualmente referido como *Vehicle-to-Everything*

³ *Cellular-V2X* é um recurso de comunicação *wireless* que utiliza *cellular standards* como o *Long Term Evolution* LTE.

⁴ DSRC é um recurso de comunicação *wireless* de curto e médio alcance bidirecional que permite uma transmissão de dados muito alta.

por entidades como a ISO [ver ISO 15628:2013 – *Intelligent Transports Systems*]. Utiliza uma frequência de 5.9 GHz, a mesma do WiFi. O alcance deste tipo de comunicação é de até 300 m ou de até 10 s se for considerada uma velocidade de autoestrada.

Pensa-se que V2V utilizará uma rede em malha (*mesh network*), o que significa que cada Nó (veículo) poderia receber, enviar e retransmitir sinais. Estima-se que entre cinco a dez retransmissões na rede reuniam condições de tráfego até uma milha ($\approx 1,6$ km) de distância [22].

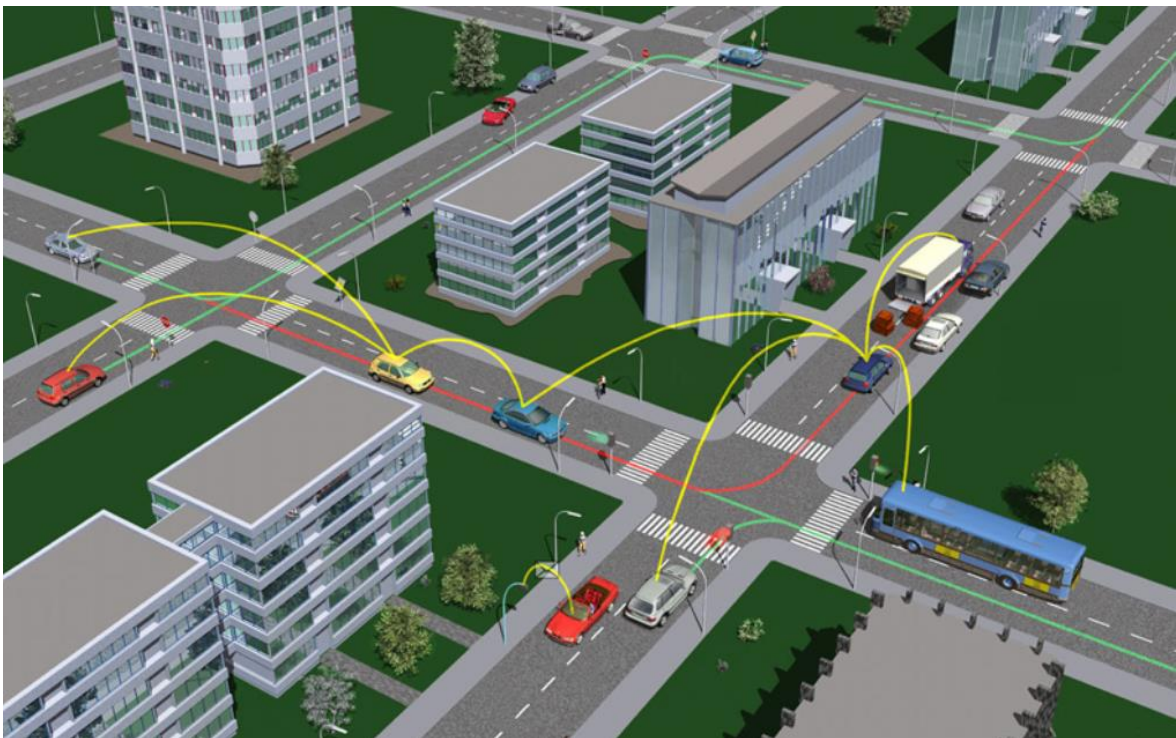


Figura 18: Ilustração de V2V Communication [23]

A comunicação V2V é focada essencialmente na segurança, eficiência de tráfego e poupança de energia. Em termos de segurança existe um benefício quando um veículo entra num cenário aplicável aos casos de uso:

- Aviso ao entrar em cruzamentos.
- Aviso de localização perigosa (ex: descoberta de obstáculos).
- Advertências de frenagem repentina (aviso colisão frontal).
- Mudanças de faixa; Sinais temporários.
- Aviso de passagem de veículos prioritários.

Segundo o Departamento de Transportes dos Estados Unidos da América, os V2V podem também evitar acidentes entre condutores que conduzam embriagados ou em estado de consumo de estupefacientes, visto que a mensagem de alerta pode ter uma implicação no tempo de reação destes condutores.

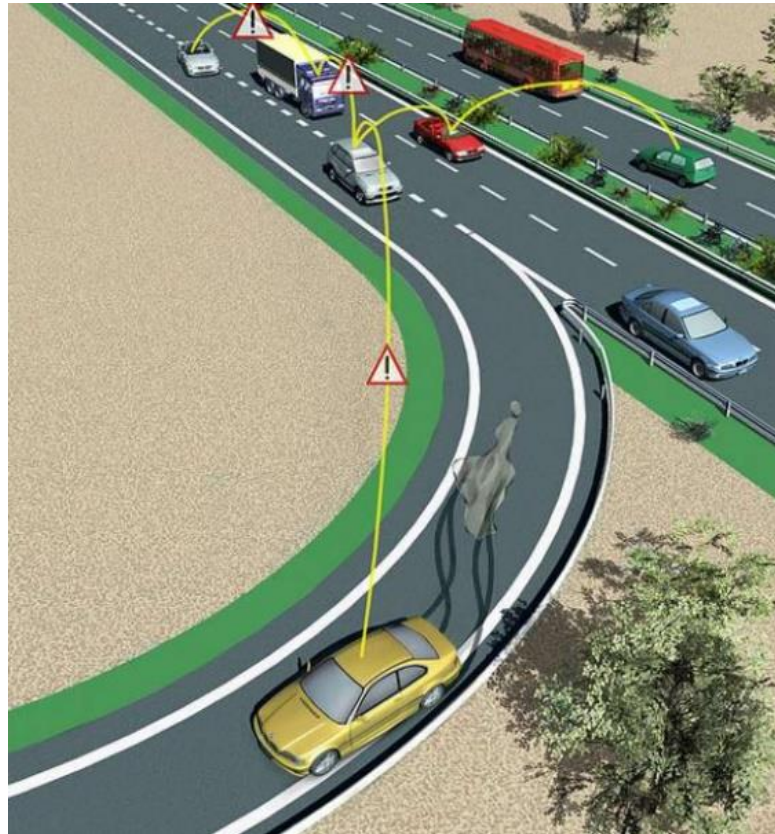


Figura 19: Ilustração de comunicação V2V (Aviso de obstáculo) [23]

Os casos de uso de eficiência de tráfego são aqueles destinados a melhorar a eficiência da rede de transporte, fornecendo informações aos proprietários da rede de transporte ou aos condutores da rede:

- Orientação de rota aprimorada e navegação.
- Semáforos adaptáveis ao *timing* de interseção dos veículos.
- Aviso de velocidade otimizada de semáforo verde.
- Limites de velocidade variável que permitem o acionamento de passagem ou paragem em semáforos.

Consequentemente, todos os pontos até aqui abordados levam a uma ligação direta de redução de consumos energéticos.

Os benefícios da comunicação V2V terão a mesma evolução de benefício que tiveram os telemóveis. Não será preciso que todos veículos tenham comunicação V2V, mas será preciso que muitos deles tenham: no início desta nova era V2V, se um determinado carro “X” tiver instalado comunicação V2V e um determinado carro “Y” (que segue à frente do carro “X”) não tiver comunicação V2V e travar de repente, os únicos métodos de auxílio ao condutor serão os métodos tradicionais que se conhecem hoje (aviso de colisão frontal, *cruise-control* adaptativo, etc.). Caso ambos veículos possuam V2V, será possível em tempo real ao condutor do veículo “X” receber informação de travagem repentina do veículo “Y”. Em caso de distração, este será um aviso crucial. Agora imagine-se uma autoestrada, onde uma dúzia de veículos seguem em curta distância entre si, onde quatro destes veículos possuem V2V, as probabilidades favorecem fortemente uma reação de frenagem antecipada num aviso de colisão evitando um embate múltiplo.

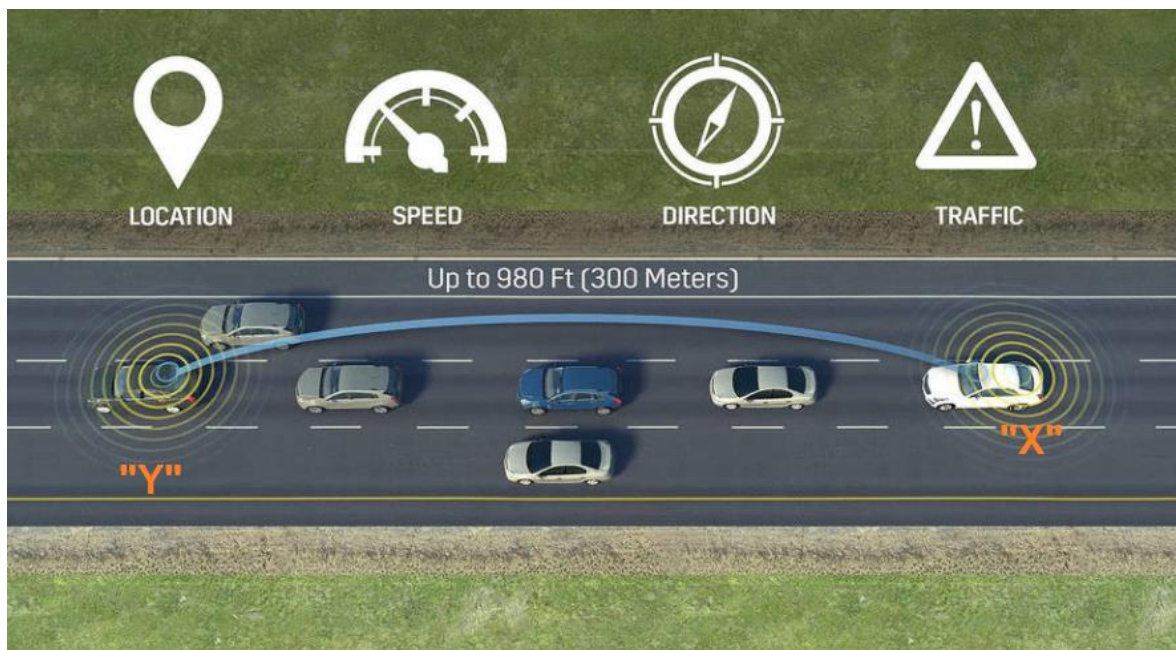


Figura 20: Exemplo de comunicação V2V [24]
Y – Veículo Y ; X – Veículo X

Outro exemplo, mais futurista, será quando todos carros tiverem equipados com comunicação V2V, podendo os condutores dirigir automaticamente por cruzamentos sem necessidade de paragem ou abrandamento.

As possibilidades num V2V são inúmeras, como a possibilidade de um veículo dirigível sem condutor, o abastecimento de energia sem ser necessário pagamento na hora, entre os anteriormente mencionados e muitos mais. Mas levanta algumas questões: poderá um V2V ser *hackeado* e virar uma ferramenta dum ataque terrorista? Poderão as entidades de

autoridade passar uma coima a um veículo V2V, porque receberam informação que esse veículo se dirigia a uma velocidade de 145 km/h? Inicialmente pensa-se que a comunicação V2V será voluntária, mas pode chegar a uma altura em que seja obrigatória (ex: veículo sem V2V não tem direito a seguro) [25] [26].

5.2. VEHICLE-TO-INFRASTRUCTURE (V2I)

Refere-se a um sistema de transporte inteligente onde os veículos e os sistemas de infraestrutura estão interconectados entre si. Esta interconecção proporcionará um conhecimento em tempo-real preciso da situação de tráfego atual em toda a rede rodoviária. Rege-se pelos mesmos objetivos de uma comunicação V2V, entre eles a otimização de tráfego, a segurança rodoviária, a poupança de energia e, conseqüentemente, a redução de emissões. Utiliza a mesma tecnologia *wireless* de uma comunicação V2V (5,9 GHz DSRC, ver ponto 5.1).

Diferentes arquiteturas V2I podem ser encontradas em diferentes trabalhos de pesquisa. Segundo o Departamento de Transportes dos Estados Unidos da América, através de um programa realizado em sistemas de transporte inteligentes [25], um sistema de arquitetura baseado em V2I deve conter os seguintes pontos:

- *Vehicle On-Board Unit or Equipment* (OBU or OBE) – Unidade de bordo do veículo ou equipamento.
- *Roadside Unit or Equipment* (RSU or RSE) – Unidade de equipamento à beira da estrada.
- *Safe Communication Channel*.

As OBUs são o lado do veículo propriamente dito numa comunicação V2I. São compostas por um rádio DSRC, um *Global Position System* (GPS), um processador de aplicações e interfaces, bem como própria HMI do veículo. As OBUs fornecem informações entre veículos e RSUs (V2I) e entre veículos (caso de comunicação V2V). Acomodam o armazenamento instantâneo de dados dependendo da sua capacidade de armazenamento e de comunicação. Após algum período de tempo os dados antigos são sobrescritos.

As RSUs podem ser montadas em qualquer ponto estratégico na rodovia, como interseções, postos de gasolina, postes de iluminação, etc. São compostas por um rádio DSRC, um processador de aplicações e interfaces, estando conectado à rede de

comunicações V2I. Estas podem gerar mensagens de prioridade para os veículos, embora os próprios veículos já tenham estabelecidas nas suas interfaces a priorização de mensagens. Aplicações de segurança locais e de veículo para veículo têm a mais alta prioridade; mensagens associadas a vários aplicativos de rede pública e privada têm menor prioridade. As mensagens de entretenimento terão a menor prioridade de todas [25].

As aplicações deste tipo de comunicação estão interligadas ao mesmo tipo de aplicação de uma comunicação V2V. No entanto, existem algumas que apenas se baseiam em sensores à beira da estrada que requerem apenas observação (como controlo do tráfego, medição de velocidade, etc.).

Em termos de segurança rodoviária, o V2I possibilita as seguintes hipóteses [27]:

- Advertência de perigo (ex: zona em obra a 500 m; Acidente a 500 m).
- Aviso de passagem de veículos prioritários.
- Advertência de velocidade (ex: O semáforo vai ficar vermelho. Por favor abrande).

Quanto à otimização de tráfego surgem hipóteses como:

- Informação cidadina (ex: Estacionamento disponível; dirija a 50km/h para apanhar semáforo verde).
- Aviso de passagem de veículos (ex: O semáforo vai ficar verde para autocarros em 8 s).
- Aviso de passagem geral (ex: O semáforo vai ficar verde em 8 s).

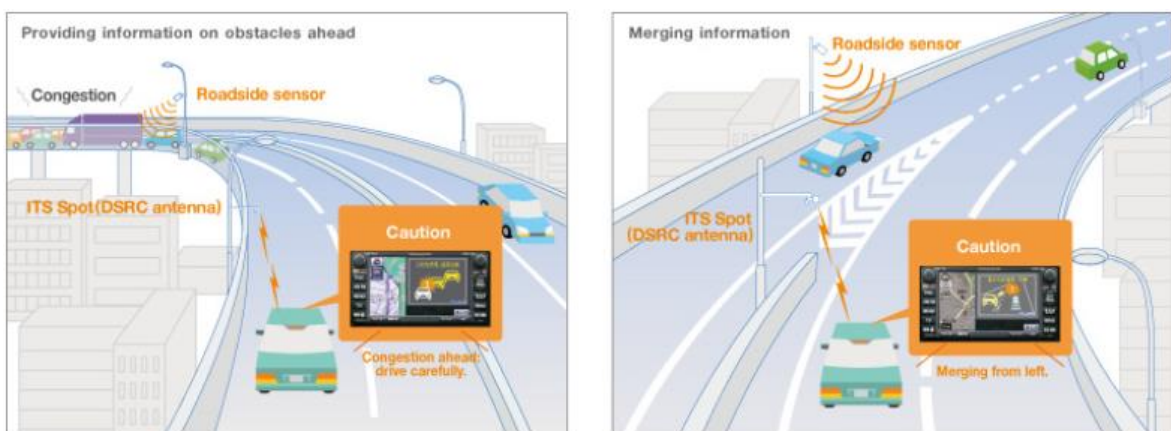


Figura 21: Exemplos de segurança rodoviária através de comunicação V2I [28]

Simulações reais de V2I já foram experimentadas, tendo como objetivo testar alguns dos tipos de auxílio através de interface acima mencionados [29]. Nessa simulação um condutor recebeu advertências de perigo vindas da comunicação V2I:

- Situação de encerramento de faixa – “Ao nos aproximarmos da zona de fecho de via de trânsito a uma velocidade de 40 mph [...] a RSU passou informação à interface do automóvel: [“Via encerrada à frente”]. Oito segundos depois a buzina tocou automaticamente e a interface apresentou a mensagem [“Via encerrada. Sair”]”.
- Aviso de velocidade de curva – “O carro aproxima-se de uma curva a uma velocidade de 30 mph. [...] A RSU envia informação à interface sobre o raio da curva, superfície seca da estrada, asfalto, ângulo de inclinação zero [...] uma mensagem emerge [“Abrandar”] e a buzina do automóvel soa”.

5.3. VEHICLE-TO-PEDESTRIAN (V2P)

A iniciativa de estudar uma possível comunicação entre veículos e peões e ciclistas foi tomada pela *General Motors* e tem como objetivo diminuir o número de mortos e feridos associados a fatalidades por veículos motorizados. A *General Motors* está a estudar a possibilidade de aceder aos *smartphones* dos peões e trocar de informação com veículos circundantes, num método similar às comunicações V2V e V2I. O ex-Secretário dos Transportes dos Estados Unidos da América, Anthony Fox, descreveu a possibilidade de comunicar *vehicle to everything* (V2X) como reduzindo potencialmente em 70% a 80% o número de condutores feridos em acidentes rodoviários [30].

5.4. V2X STANDARDS

Três *standards* de normalização trabalham com as redes veiculares. A *Institute of Electric and Electronic Engineers* (IEEE) IEEE 802.11p especifica extensões para comunicação *wireless* num ambiente veicular. A IEEE 1609 define a arquitetura e um conjunto complementar e padronizado de serviços e interfaces, que permitam coletivamente comunicações *wireless* V2V e V2I. O terceiro padrão é desenvolvido pela *Society of Automotive Engineers* (SAE). A SAE J2735 define conjuntos de mensagens, *frames* de dados e elementos que são usadas para trocas de segurança V2V e V2I.

CAPÍTULO 6

6. DESCRIÇÃO DA APLICAÇÃO DESENVOLVIDA

Este ponto do relatório aborda o desenvolvimento do projeto no seu todo, caracterizando a definição de mensagens do problema proposto, passando pela sua codificação e implementação do software, terminando na apresentação de resultados e respectivas conclusões.

6.1. DEFINIÇÃO DE MENSAGENS A SEREM APLICADAS COMO COMUNICAÇÃO V2I

No presente ponto será descrito de forma funcional os serviços e casos de uso implementados no âmbito desta dissertação. As mensagens foram estabelecidas pela entidade empresarial de acordo com os critérios por esta estabelecidos, correspondendo ao mestrando a respectiva pesquisa, compreensão, explicação e implementação das mesmas. Neste contexto, será utilizada a seguinte terminologia de caracterização de mensagem:

- Caso de uso – função do sistema, comportamento desejado (do sistema e atores), especificação dos limites do sistema e definição de pelo menos um cenário de uso.
- Situação – descreve a situação relevante, considerando metas e valores relacionados à função.
- Cenário – descreve o acontecimento temporal numa sequência de situações com base em eventos e ações.
- Atores – entidades externas (humanas) que interagem com o sistema. O sistema é afetado pelo comportamento dos atores, portanto essas relações são descritas nas descrições de casos de uso.

O conjunto de mensagens estabelecidas será descrito de seguida e corresponde a mensagens de:

- Limite de velocidade otimizada.
- Trânsito/congestionamento.
- Condições meteorológicas (aderência de piso, visibilidade, vento e precipitação).

- Obras (fechamento de via de trânsito, estrada encerrada, trabalhos móveis na via).
- Aproximação de veículos prioritários.

1) Mensagem de *limite de velocidade otimizada*

Tabela 5 :Topologia de Caracterização de Mensagem Limite de velocidade otimizada

Introdução de caso de uso	
Sumário	O assunto da mensagem corresponde a uma advertência em como limite de velocidade legal num determinado troço foi ultrapassado, podendo levar à ativação da luz vermelha do próximo semáforo a intercepar, levando à paragem do autocarro (aconselha-se diminuição de velocidade).
Contexto	Informa os utilizadores da estrada sobre a passagem do limite de velocidade real para que eles possam adaptar a sua velocidade de forma adequada.
Objetivo	O objetivo é informar os usuários da estrada sobre o limite de velocidade dinâmica, atualmente válido, e que não ultrapasse o limite máximo legal em vigor num determinado troço da rodovia. Evita-se, desta forma, a paragem do autocarro no semáforo mais próximo.
Comportamento Pretendido	Os utilizadores da estrada adaptam o seu comportamento de condução em conformidade com o limite de velocidade de condução aplicável.
Benefícios Esperados	Conveniência para os usuários, melhor conformidade com os limites de velocidade, maior segurança e potenciais benefícios ambientais.
Descrição de caso de uso	
Situação	Serão instaladas RSUs em partes sensíveis da rede de estradas, que permitam gerir e informar os usuários de estrada sobre condições de tráfego e velocidades otimizadas de modo a atingir o objetivo pretendido.
Lógica de Transmissão	Infrastructure to Vehicle (I2V).
Atores	e Operador de estrada: A fonte de qualquer informação dinâmica de

Relações	<p>limite de velocidade é o operador rodoviário através do Centro de Controlo de Tráfego.</p> <p>Usuário de estrada: O usuário da estrada pode usar as informações para melhor atender ao limite de velocidade atual.</p> <p>Provedor de serviços: O fabricante do veículo divulga informações de velocidade otimizada ao usuário da via.</p>
Cenário	A mensagem é recebida no veículo ao passar pela RSU e exibida ao motorista (em caso de excesso de velocidade). O usuário da estrada pode adaptar a sua velocidade.
Interface - HMI	As informações devem ser apresentadas ao utilizador da estrada e devem ser coerentes com os sinais de trânsito dinâmicos válidos atualmente. A informação deve ser exibida ao motorista com antecedência suficiente e no local certo.

2) Mensagem de *alerta de congestionamento*

Tabela 6: Topologia de Caracterização de Mensagem Alerta de Congestionamento

Introdução de caso de uso	
Sumário	Através do Centro de Controlo de Tráfego, o operador rodoviário deteta um engarrafamento e envia as informações ao usuário da estrada (mencionando as posições, o comprimento do engarrafamento e as seções/faixas em questão, se as informações estiverem disponíveis).
Contexto	Com os sistemas cooperativos de transporte inteligente, a disponibilidade e precisão de aviso de congestionamento são melhores do que os meios convencionais e, portanto, os motoristas são avisados com maior qualidade de informação.
Objetivo	O objetivo deste caso de uso é informar principalmente acerca de um possível fim de fila perigoso. O motorista pode mudar a sua abordagem de condução, adaptando a sua velocidade, e dirigindo a aproximação ao final da área de congestionamento cautelosamente.

Comportamento Pretendido	Motoristas corretamente informados adaptam o seu comportamento ao conduzir (por exemplo, reduzir a velocidade de aproximação, antes de chegar ao ponto final do engarrafamento e ao passá-lo).
Benefícios Esperados	Benefícios económicos: economia de recursos, dinheiro e tempo para todas as partes interessadas. Benefícios sociais: segurança no trânsito, número reduzido de incidentes. Benefícios pessoais: condução mais confortável. Benefícios ambientais: redução de consumo energético.
Descrição de caso de uso	
Situação	O congestionamento pode estar numa faixa específica (por exemplo, na saída de uma rodovia) ou em toda a secção rodoviária. A mensagem de aviso para a respetiva faixa ou secção da estrada é enviada aos usuários que se aproximam da área de tráfego.
Lógica de Transmissão	Infrastructure to Vehicle (I2V).
Atores e Relações	Operador de estrada: Descobre e envia quaisquer informações acerca de alerta de congestionamentos para o Centro de Controlo de Tráfego. Usuário de estrada: O usuário da estrada pode usar as informações para melhor atender a uma condução cautelosa. Provedor de serviços: Divulga informações relacionadas com o alerta de congestionamento para os motoristas. As fontes de informação podem ser: câmaras e/ou agentes de patrulha devidamente equipados.
Cenário	O operador no Centro de Controlo de Tráfego é informado sobre um engarrafamento na sua rede, confirma o seu comprimento e/ou faixa e a mensagem é então transmitida para os usuários da via. Os veículos próximos à área de engarrafamento recebem as informações e exibem-nas. O motorista adapta o seu comportamento.
Interface - HMI	O usuário recebe informações relacionadas com o objetivo

	pretendido, exibidas no painel, podendo adaptar a sua condução. Podem ser consideradas distâncias bastante próximas entre a RSU e o fim do congestionamento para não exibir nenhuma mensagem, caso contrário a informação deve ser exibida ao motorista com antecedência suficiente e no local certo.
--	---

- 3) Mensagem de *aviso de condições meteorológicas adversas* (aderência de piso/gelo, visibilidade dificultada, ventos fortes, precipitação).

Tabela 7: Topologia de Caracterização de Mensagem Aviso de Condições Meteorológicas Adversas

Introdução de caso de uso	
Sumário	As condições meteorológicas mostram informações meteorológicas locais precisas e atualizadas. Os motoristas são informados sobre condições meteorológicas perigosas à frente, especialmente onde o perigo é difícil de perceber visualmente, como gelo ou fortes rajadas de vento.
Contexto	A condição climática faz parte do conjunto mínimo de serviços de informação sobre o tráfego relacionados com a segurança rodoviária gratuitos aos utilizadores nas estradas europeias. Condições climáticas adversas são condições climáticas incomuns, severas ou fora de estação que podem afetar uma condução segura.
Objetivo	Melhorar a segurança no trânsito através de meios adicionais de mensagens para informar os condutores de forma mais precisa sobre as condições meteorológicas e informações sobre o estado das estradas.
Comportamento Pretendido	O condutor do veículo adapta o seu comportamento de condução em conformidade com os regulamentos de condução aplicáveis e qualquer conselho ou orientação fornecidos.
Benefícios	O principal impacto esperado é uma condução mais atenta

Esperados	fornecendo informações reais e contínuas sobre as condições da estrada (aderência, visibilidade, vento, chuva) e melhorando a segurança no trânsito, reduzindo o número e a gravidade dos acidentes.
Descrição de caso de uso	
Situação	Espera-se que informe os condutores sobre informações atuais e/ou esperadas relacionadas à precipitação ou condições climáticas extremas.
Lógica de Transmissão	Infrastructure to Vehicle (I2V).
Atores e Relações	<p>Operador rodoviário: valida o aviso, emite informações de acionamento.</p> <p>Provedor de serviço: recebe e garante que as informações de acionamento estejam corretas, aciona o aviso I2V.</p> <p>Usuário final: recebe o aviso via unidade de bordo.</p> <p>Motorista de veículo: é informado sobre condições meteorológicas perigosas à frente a tempo de adaptar o seu comportamento de condução.</p>
Cenário	O operador no Centro de Controlo de Tráfego é informado sobre condições climáticas extremas na sua rede. Ele junta as informações, confirma-as e a mensagem é então distribuída e transmitida aos usuários da estrada. Os veículos recebem as informações e exibem-nas ao motorista. O motorista adapta o seu comportamento.
Interface - HMI	O usuário recebe as informações relacionadas, exibidas no painel. A mensagem de aviso é exibida com antecedência suficiente para que o motorista possa adaptar o comportamento de condução e, ao mesmo tempo, não esteja muito distante do segmento de estrada afetado.

4) Mensagem de *alerta de obras: via de trânsito fechada*

Tabela 8: Topologia de Caracterização de Mensagem Alerta de Obras: via de Trânsito Fechada

Introdução de caso de uso	
Sumário	O usuário da estrada recebe informações sobre o encerramento de parte de uma via de trânsito, via inteira ou várias vias, mas sem o encerramento da estrada. O encerramento de via é devido à realização de obras rodoviárias estáticas.
Contexto	Atualmente, muitos usuários de estradas entram nos locais das obras rodoviárias ou embatem nos equipamentos de proteção do local, às vezes causando vítimas. As informações com antecedência suficiente evitariam esse tipo de situação, adaptando o comportamento do usuário da via.
Objetivo	O objetivo é permitir que os usuários da estrada previnam o encerramento de vias e adaptem a sua velocidade e a faixa na estrada. O objetivo não é sinalizar o encerramento de uma estrada e, portanto, nenhuma rota alternativa será transmitida.
Comportamento Pretendido	Maior vigilância. Adaptação da velocidade. Mudança de faixas (se necessário).
Benefícios Esperados	Reduzir o risco e o número de acidentes e situações perigosas para usuários e trabalhadores da estrada. Informar o usuário da estrada sobre o risco de desconforto na estrada (desacelerar, manobrar) Gestão de tráfego aprimorado.
Descrição de caso de uso	
Situação	Obras rodoviárias equipadas com cones de aviso / sinais de trânsito temporários / setas luminosas iluminadas, numa estrada com faixas de rodagem separadas ou em faixa dupla. Bloqueio das faixas com RSU (obras rodoviárias de curta duração).
Lógica de Transmissão	Infrastructure to Vehicle (I2V).

Atores	e	O operador da estrada é a origem da informação da mensagem.
Relações		O usuário da estrada que se aproxima da área é o usuário final deste serviço (recebe a informação). Prestador de serviços: o planejador de obras rodoviárias do operador rodoviário ou o RSU instalado num reboque (no caso do “modo autónomo”).
Cenário		O operador rodoviário programa obras estáticas e planejadas. A informação contém todos os elementos que podem ser usados para descrever o local de trabalho (posição inicial/final do local de trabalho, duração) com precisão. A mensagem é, então, transmitida para os usuários da estrada. O usuário da estrada recebe as informações e processa-as.
Interface - HMI		Quando o usuário da estrada chega perto do local da zona de trabalho, ele recebe informações para permitir que ajuste a sua velocidade e posição na estrada para evitar situações perigosas. As informações precisam de ser exibidas na HMI com antecedência suficiente.

5) Mensagem de *alerta de obras: rodovia fechada*

Tabela 9: Topologia de Caracterização de Mensagem Alerta de Obras: Rodovia Fechada

Introdução de caso de uso	
Sumário	O usuário da estrada recebe informações sobre o encerramento de uma rodovia devido a um conjunto de obras estáticas. O encerramento é temporário.
Contexto	Quando os usuários da estrada ficam impedidos de passar sem serem informados sobre a situação, estes podem ficar ansiosos e fazer reviravoltas perigosas ou usar uma pista inadequada. Proporcionar este tipo de informação pode prevenir essas situações, trazendo mais segurança e conforto para todos.
Objetivo	Permitir que o motorista antecipe o encerramento de uma estrada para que possa escolher uma rota alternativa.

Comportamento Pretendido	Adaptação de rota alternativa.
Benefícios Esperados	Segurança. Melhor gestão de tráfego. Maior conforto para os usuários das estradas.
Descrição de caso de uso	
Situação	Numa via dupla: uma direção é fechada, sem cruzamento de faixa de rodagem. Numa faixa de rodagem de duas vias: toda a rodovia está fechada. Em ambos os casos: um desvio é indicado perto do encerramento.
Lógica de Transmissão	Infrastructure to Vehicle (I2V).
Atores e Relações	O operador da estrada é o remetente da mensagem. Pode estar em contato com os outros operadores rodoviários para implementar um itinerário de desvio inteligente. O usuário da rodovia é o usuário final do serviço. O prestador de serviços pode ser o operador rodoviário.
Cenário	O operador rodoviário programa obras rodoviárias estáticas e planejadas. Esta informação contém todos os elementos que podem ser usados para descrever com precisão o local de trabalho (posição inicial/final do encerramento, duração) e potenciais rotas alternativas. A mensagem é então transmitida aos usuários da estrada que se aproximam do encerramento da estrada, para que possam adaptar o seu itinerário. A informação é recebida no veículo e exibida ao motorista.
Interface - HMI	Quando o usuário da estrada chega perto do local da zona de trabalho, recebe um alerta intrusivo no caso de uma antecipação temporal significativa para que possa adaptar a sua rota.

6) Mensagem de *alerta de obras: trabalhos móveis*

Tabela 10: Topologia de Caracterização de Mensagem Alerta de Obras: Trabalhos Móveis na Via

Introdução de caso de uso	
Sumário	O utilizador da estrada recebe informações sobre uma zona na estrada que contém, num determinado momento, a neutralização de parte de uma faixa (mas sem encerramento da estrada) devido a um local de trabalho móvel planejado.
Contexto	Muitos usuários de estradas entram nos locais das obras rodoviárias ou embatem nos equipamentos de proteção do local. As informações com antecedência suficiente evitariam esse tipo de situação, principalmente num local de trabalho móvel que é “mais leve” em termos de proteção e sinalização.
Objetivo	O objetivo deste caso de uso é informar um usuário da estrada de uma zona de trabalho móvel onde encontrará agentes operacionais na zona.
Comportamento Pretendido	Maior vigilância. Adaptação da velocidade. Mudança de faixas (se necessário).
Benefícios Esperados	Reduzir o risco de acidentes (para usuários, agentes rodoviários). Informar o usuário da estrada sobre o risco de desconforto na estrada (desaceleração, manobras). Melhor gestão de tráfego.
Descrição de caso de uso	
Situação	Várias situações que impliquem trabalhos móveis, como limpeza de estradas, instalações de equipamentos, manutenções, etc.
Lógica de Transmissão	Infrastructure to Vehicle (I2V).
Atores e Relações	O operador da estrada é o remetente da mensagem. O usuário da rodovia é o usuário final do serviço. O prestador de serviços é o planejador de obras rodoviárias do operador rodoviário.
Cenário	O operador rodoviário programa obras rodoviárias móveis. A informação contém todos os elementos que podem ser utilizados

	para descrever com precisão a zona de trabalho (posição inicial/final da zona de trabalho, duração). A mensagem é transmitida para os usuários da via. O veículo recebe a informação, processa-a e exibe-a ao motorista.
Interface - HMI	Quando o usuário da estrada chega perto da zona de trabalho planejada recebe um alerta para permitir que ajuste a sua velocidade e posição na rodovia. O alerta precisa de ser exibido na HMI com antecedência suficiente.

7) Mensagem de *aproximação de veículo prioritário*.

Tabela 11: Topologia de Caracterização de Mensagem Alerta de Aproximação de Veículo Prioritário

Introdução de caso de uso	
Sumário	Através de comunicação V2I ou V2V, o operador rodoviário deteta a aproximação de um veículo prioritário e envia as informações ao usuário da estrada (mencionando as posições e as seções/faixas em questão, se as informações estiverem disponíveis).
Contexto	Com os sistemas cooperativos de transporte inteligente, a disponibilidade e precisão de aviso de aproximação de veículo prioritário são essenciais e, portanto, os motoristas são avisados com maior qualidade de informação.
Objetivo	O objetivo deste caso de uso é informar principalmente acerca de uma possível aproximação. O motorista pode mudar a sua abordagem de condução, adaptando a sua velocidade, mudando de via de trânsito e dirigir cautelosamente.
Comportamento Pretendido	Motoristas corretamente informados adaptam o seu comportamento ao conduzir (por exemplo, reduzir a velocidade de aproximação, mudar de via de trânsito ou simplesmente cedendo passagem).
Benefícios Esperados	Benefícios econômicos: economia de recursos, dinheiro e tempo para todas as partes interessadas. Benefícios sociais: segurança no

	trânsito, número reduzido de incidentes. Benefícios pessoais. Benefícios ambientais: redução de consumo energético.
Descrição de caso de uso	
Situação	A aproximação do veículo prioritário pode estar numa faixa específica (por exemplo, na saída de uma rodovia). A mensagem de aviso sobre a respetiva faixa ou seção da estrada é enviada aos usuários.
Lógica de Transmissão	Vehicle to Vehicle (V2V) ou Infrastructure to Vehicle (I2V).
Atores e Relações	Usuário de estrada: O usuário da estrada pode usar as informações para melhor atender a uma condução cautelosa. Provedor de serviços: Divulga informações relacionadas com o alerta de aproximação de veículo prioritário para os motoristas. As fontes de informação podem ser: os próprios veículos, RSUs, câmaras e/ou agentes de patrulha devidamente equipados.
Cenário	A mensagem é comunicada conforme a aproximação do veículo prioritário com os veículos que se mantenham no mesmo percurso. Os veículos próximos à área de aproximação recebem as informações e exibem-nas. O motorista adapta o seu comportamento.
Interface - HMI	O usuário recebe informações relacionadas com o objetivo pretendido, exibidas no painel, podendo adaptar a sua condução.

6.2. DESENVOLVIMENTO DO CONJUNTO DE MÁSCARAS

Este ponto teve como o objetivo a criação do conjunto de máscaras que servem como caracterização das mensagens que se pretendem mostrar na interface, bem como a atribuição das respectivas prioridades caso duas ou mais mensagens concorram no mesmo período de tempo ao acesso de barramento (i.e., pretende-se mostrar no *display* da interface a mensagem considerada como a mais prioritária). As máscaras devem apresentar a informação desejada de uma forma coerente para facilitar ao motorista a sua rápida compreensão. A ferramenta de auxílio para a elaboração destas máscaras foi a *draw.io*, uma ferramenta de livre acesso usada, sobretudo, para a implementação de diagramas de processo, fluxogramas e diagramas de rede.

Na Figura 22 encontra-se ilustrada uma das máscaras criadas neste projeto. Todas as restantes ilustrações encontram-se disponíveis em anexo.



Figura 22: Máscara referente à mensagem de alerta de aproximação de veículo prioritário

As respetivas prioridades de acesso relativas às diferentes mensagens foram atribuídas da seguinte forma (em ordem crescente, sendo a mensagem nº1 a menos prioritária e a mensagem nº10 a mais prioritária):

1. Mensagem de limite de velocidade otimizada.
2. Mensagem de condição meteorológica – vento fortes.
3. Mensagem de condição meteorológica – visibilidade dificultada.
4. Mensagem de condição meteorológica – aderência ao piso/gelo.
5. Mensagem de condição meteorológica – precipitação.
6. Mensagem de alerta de congestionamento.
7. Mensagem de alerta de obras: rodovia fechada.
8. Mensagem de alerta de obras: via de trânsito fechada.

9. Mensagem de alerta de obras: trabalhos móveis.
10. Mensagem de aproximação de veículo prioritário.

A atribuição da ordem de prioridade foi realizada em conjunto entre a entidade empresarial e o mestrando. Justifica-se que a mensagem nº1 não represente propriamente o maior perigo em relação às demais, mas antes uma necessidade de economizar energia/custos. As mensagens nº2 até nº5 são acerca de condições meteorológicas adversas, representando um perigo. No entanto, o motorista responsável pela condução tem ou deve ter percepção das condições meteorológicas que o rodeiam. Quanto às mensagens nº6 até nº10, estas representam situações que o motorista não consegue prever o perigo sem o auxílio a uma ferramenta qualquer. Desta forma, atribuiu-se à mensagem nº6 o alerta de congestionamento. As mensagens nº7 até nº9 justificam-se com uma maior prioridade por envolverem agentes externos em trabalho aquando da passagem do autocarro (ex: trabalhadores de obras) e, por último, a mensagem nº10, considerada como a maior prioritária por representar uma emergência.

6.3. CODIFICAÇÃO DAS MENSAGENS EM COMUNICAÇÃO CAN

De modo a iniciar o desenvolvimento do *software* é necessário estruturar as mensagens relevantes dos diferentes componentes da rede de comunicação CAN. Como analisado anteriormente (capítulo 3, nomeadamente no ponto 3.5.1), uma *frame* de dados contém várias informações, sendo uma delas o campo de dados, que pode conter até oito *bytes*. O identificador é outra informação importante, pois contém todos os sinais que se pretende transmitir. Na tabela 12, coluna “Endereço”, encontra-se o identificador em sistema de numeração hexadecimal. Na segunda coluna, “Sinal”, encontra-se o nome atribuído ao sinal de modo a ser de fácil identificação. A coluna referente ao “Comentário” refere em que posição do canal é enviado o sinal (*byte* e *bit*). As “Condições de Entrada” referem-se às condições admitidas para entrada do sinal. As mensagens de auxílio ao motorista foram codificadas no *software PCAN Explorer* (*software* de comunicação CAN) e serão analisadas de seguida:

Tabela 12: Mensagem de trânsito em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Mensagem de alerta de trânsito	Byte 0, bit 0-1	00 - Trânsito OFF
			01 - Trânsito ON
			10 - Erro
			11 - Não disponível

A mensagem CAN proveniente do alerta de trânsito é enviada com um tempo de ciclo de 250 ms, através do *bit* 0 e 1 do *byte* 0. As condições de entrada podem ser 00, 01, 10 e 11, que representam, respectivamente, a mensagem de alerta de trânsito não ativa, mensagem de alerta de trânsito ativa, o erro e o não disponível. As condições, erro de sinal (de aquisição de sinal) e não disponível (não aplicável) são essenciais para programar funções de segurança. Seguindo exatamente a mesma metodologia, no mesmo *byte* 0, no *bit* 2 e 3, *bit* 4 e 5, *bit* 6 e 7, tem-se as mensagens de alerta de via de trânsito fechada, estrada fechada e trabalhos móveis, respectivamente.

Na tabela 13, a condição de entrada 00 - representa a mensagem *de alerta de via de trânsito fechada* não ativa, 01 – representa a mensagem de alerta via de trânsito fechada ativa, 10 – erro e 11 – não disponível.

Tabela 13: Mensagem de via de trânsito fechada em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128(Hex) Tempo de ciclo: 250 ms	Mensagem de alerta de via de trânsito fechada	Byte 0, bit 2-3	00 – Via Fechada OFF
			01 – Via Fechada ON
			10 - Erro
			11 - Não disponível

Na tabela 14, a condição de entrada 00 - representa a mensagem *de alerta de estrada fechada* não ativa, 01 – representa a mensagem de alerta de estrada fechada ativa, 10 – erro e 11 – não disponível.

Tabela 14: Mensagem de estrada fechada em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128(Hex) Tempo de ciclo: 250 ms	Mensagem de alerta de estrada fechada	Byte 0, bit 4-5	00 – Estrada Fechada OFF
			01 – Estrada Fechada ON
			10 - Erro
			11 - Não disponível

Na tabela 15, a condição de entrada 00 - representa a mensagem *de alerta de trabalhos móveis na rodovia* não ativa, 01 – representa a mensagem de alerta de trabalhos móveis na rodovia ativa, 10 – erro e 11 – não disponível.

Tabela 15: Mensagem de trabalhos móveis em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Mensagem de alerta de trabalhos móveis	Byte 0, bit 6-7	00 – Trabalhos móveis OFF
			01 – Trabalhos móveis ON
			10 - Erro
			11 - Não disponível

Passando para as mensagens *de alerta de condições meteorológicas adversas* (precipitação, vento, aderência ao piso e visibilidade – tabela 16), no *byte 1*, é possível codificar a informação destas mensagens em apenas quatro *bits* poupando espaço ao identificador. Assim sendo, tem-se nos *bits* 0-3 as condições de entrada: 0000 – mensagem de alerta de ventos fortes OFF, 0001 – mensagem de alerta de ventos fortes ON, 0010 – erro , 0011 – não disponível, 0100 – mensagem de alerta de visibilidade dificultada OFF, 0101 – mensagem de alerta de visibilidade dificultada ON, 0110 – erro , 0111 – não disponível, 1000 – mensagem de alerta de aderência de piso/gelo OFF, 1001 – mensagem de alerta de aderência de piso/gelo ON, 1010 – erro , 1011 – não disponível, 1100 – mensagem de alerta de precipitação OFF, 1101 – mensagem de alerta de precipitação ON, 1110 – erro , 1111 – não disponível.

Tabela 16: Mensagem de condições meteorológicas adversas em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Mensagem de alerta de condições meteorológicas adversas	Byte 1, bit 0-3	0000 - Vento OFF
			0001 - Vento ON
			0010 - Erro
			0011 – Não disponível
			0100 - Visibilidade OFF
			0101 - Visibilidade ON
			0110 - Erro
			0111 – Não disponível
			1000 – Gelo OFF
			1001 – Gelo ON
			1010 - Erro
			1011 – Não disponível
			1100 – Precipitação OFF
			1101 – Precipitação ON
1110 - Erro			
1111 – Não disponível			

As mensagens de *limite de velocidade otimizada* e de *aproximação de veículo prioritário*, estão definidas no *byte 1, bit 4-5* e *bit 6-7* respectivamente.

Na tabela 17, a condição de entrada 00 - representa a mensagem de alerta de velocidade otimizada não ativa, 01 – representa a mensagem de alerta de velocidade otimizada ativa (excesso de velocidade), 10 – erro e 11 – não disponível.

Tabela 17: Mensagem de limite de velocidade otimizada em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Mensagem de limite de velocidade otimizada	Byte 1, bit 4-5	00 – Velocidade otimizada OFF
			01 – Velocidade otimizada ON
			10 - Erro
			11 - Não disponível

Na tabela 18, a condição de entrada 00 - representa a mensagem de alerta de aproximação de veículo prioritário não ativa, 01 – representa a mensagem de alerta de aproximação de veículo prioritário ativa, 10 – erro e 11 – não disponível.

Tabela 18: Mensagem de alerta de aproximação de veículo prioritário em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Mensagem de alerta de aproximação de veículo prioritário	Byte 1, bit 6-7	00 – Aproximação de veículo prioritário OFF
			01 – Aproximação de veículo prioritário ON
			10 - Erro
			11 - Não disponível

No *byte 2* (tabela 19), está codificada a velocidade instantânea do autocarro. Este sinal ocupa todos os *bits* do *byte 2*, de forma a conseguir devolver um valor em decimal, que corresponda à velocidade máxima passível de ser atingida pelo autocarro. Note-se que 8 *bits* em nível lógico 1 corresponde ao valor máximo de 255 em decimal, utilizado, neste caso, como o alcance máximo possível de uma certa velocidade correspondente em km/h.

Tabela 19: Mensagem de velocidade em CAN

Endereço	Sinal	Comentário	Condição de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Velocidade do autocarro	Byte 2, bit 0-7 Alcance (decimal) = 255 1 km/h por bit	Velocidade (km/h)

No *byte 3*, ocupando os seus oito *bits*, está codificada a distância entre autocarro e fim de fila de trânsito em metros. A particularidade deste *byte 3* prende-se no facto de, no *software* PCAN, ter-se multiplicado em cada um dos oito *bits* do *byte 3* pelo *factor* 10. O *factor* 10 em PCAN não é mais do que uma multiplicação dos *bits* pelo número decimal 10. Desta forma, não utilizando um outro identificador, pode-se obter a leitura de valores decimais desde 0 até 2550 metros (i.e., 0-255 * 10). O único senão em utilizar este método prende-se na comunicação CAN só ler e enviar dados numéricos num intervalo de 10 em 10 valores decimais (ex: 10metros, 20 metros, 30 metros até 2550 metros). Isto permite

aumentar o alcance da leitura da distância pretendida na comunicação CAN. Nos *bytes* 4, 5 e 6, é utilizada exatamente a mesma metodologia de multiplicação pelo *factor* 10, mas nestes casos para se obter a distância entre veículo e o ponto de alerta associado às mensagens de via fechada, estrada fechada e obras móveis, respetivamente (tabela 20).

Tabela 20: Mensagens de distâncias em CAN

Endereço	Sinal	Comentário	Condições de Entrada
18FF0128 (Hex) Tempo de ciclo: 250 ms	Distância Trânsito	Byte 3, bit 0-7 Factor = 10 Alcance (decimal) = 2550	Distância (m)
18FF0128 (Hex) Tempo de ciclo: 250 ms	Distância Via Fechada	Byte 4, bit 0-7 Factor = 10 Alcance (decimal) = 2550	Distância (m)
18FF0128 (Hex) Tempo de ciclo: 250 ms	Distância Rua Fechada	Byte 5, bit 0-7 Factor = 10 Alcance (decimal) = 2550	Distância (m)
18FF0128 (Hex) Tempo de ciclo: 250 ms	Distância Obras Móveis	Byte 6, bit 0-7 Factor = 10 Alcance (decimal) = 2550	Distância (m)

O *byte* 7 do identificador foi deixado em vazio, pois não surgiu necessidade de o utilizar. Tem-se de seguida, a Figura 23 que ilustra a codificação de todas as mensagens de alerta realizadas no *software PCAN Explorer*.

Variables	Layout							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	ObrasMoveis		EstradaFechada		ViaFechada		Transito	
Byte 1	VeiculoPrioritario		VelocidadeOtimizada		CondicoesMeteorologicas			
Byte 2	Velocidade							
Byte 3	DistanciaTransito							
Byte 4	DistanciaViaFechada							
Byte 5	DistanciaEstradaFechada							
Byte 6	DistanciaObrasMoveis							
Byte 7								

Figura 23: Ilustração de codificação das mensagens de alerta em PCAN

6.4. DESENVOLVIMENTO DO SOFTWARE

O programa automático implementado neste projeto foi desenvolvido no *software* CoDeSys através de um estilo de programação em *Structured Text* conforme abordado no ponto 4.1. As funções ilustradas nesta secção encontram-se suportadas pela interligação entre os softwares P-CAN e CoDeSys, garantindo a correta execução e programação das mesmas. Para melhor compreensão, todas as seguintes funções foram representadas em diagramas de fluxo. O fluxograma ilustrado na Figura 24 representa a legenda dos elementos constituintes dos seguintes fluxogramas. A Figura 25 representa a receção de mensagens pela OBU instalada no autocarro e consequente representação gráfica na interface.

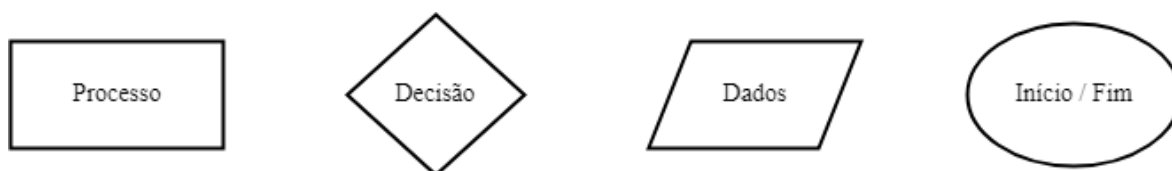


Figura 24: Legenda dos elementos constituintes dos fluxogramas

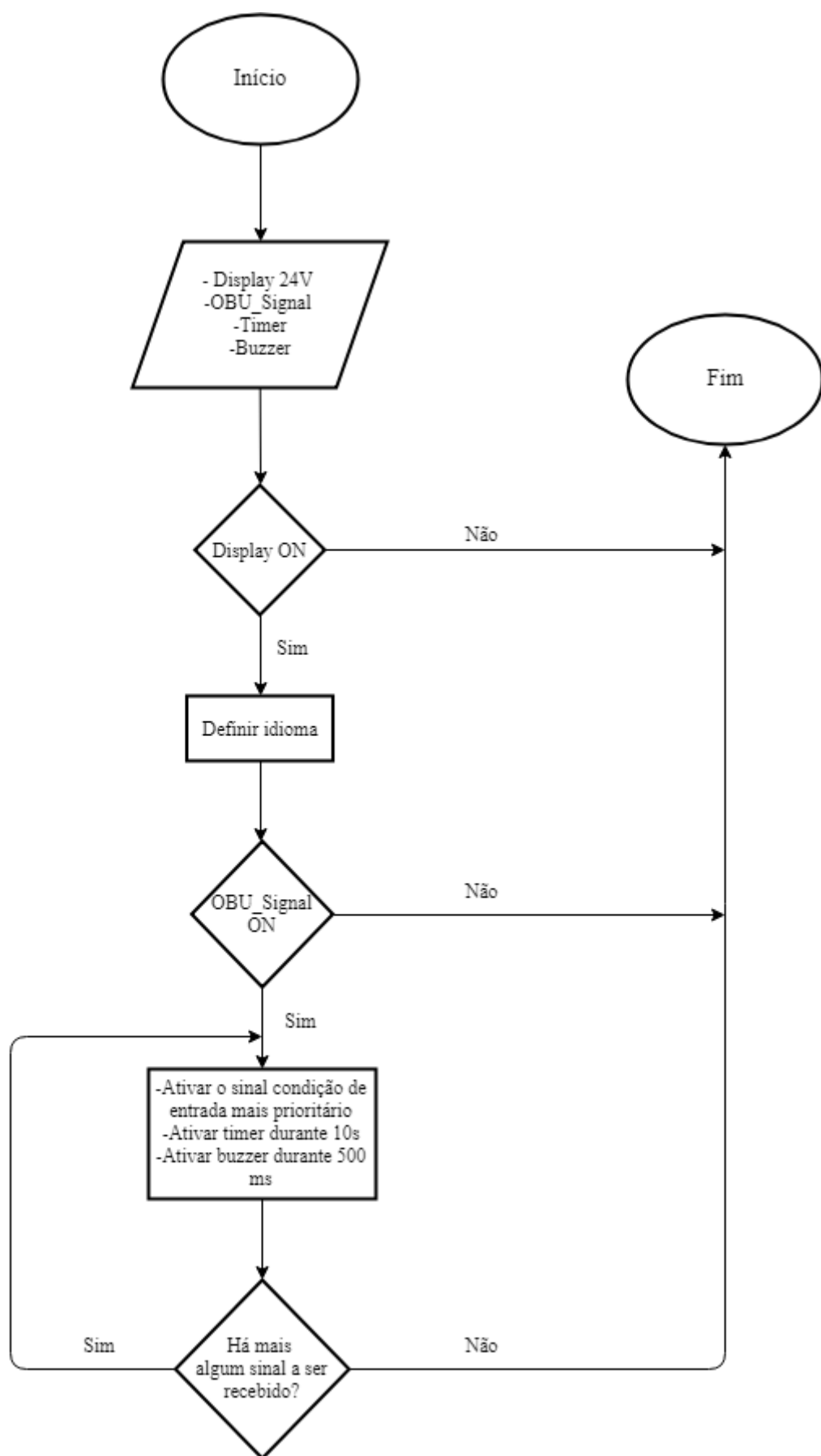


Figura 25: Fluxograma Função de prioridade

O fluxograma apresentado na Figura 25 representa a função de ativar e desativar as mensagens recebidas pela OBU através de DSRC. O autocarro ao passar num determinado ponto da estrada (onde esteja instalada uma RSU) recebe um sinal de entrada que contém a(s) mensagem(s) que se pretende mostrar ao motorista. Para iniciar este processo, o *display* da interface deve estar ligado (ON), obtendo o sinal de tensão de 24V das baterias auxiliares. É permitido ao motorista escolher se pretende ler a informação representada no *display* em língua portuguesa ou inglesa, funcionando o programa do mesmo modo independentemente da escolha de idioma. Caso seja recebida uma mensagem pela OBU (OBU_Signal ON), o programa deve ativar a condição de entrada definida como a mais prioritária (caso haja mais do que uma condição de entrada a pedir acesso no mesmo período de tempo), ativar um alerta sonoro (*buzzer#500 ms*) com duração de 500 ms de forma a captar a atenção do motorista e mostrar a respetiva mensagem de auxílio no *display*. Esta mensagem é exibida durante 10 s (*timer#10 s*), ao fim dos quais a mensagem deixa de ser exibida, podendo:

- O *display* voltar ao menu principal estabelecido caso não haja mais nenhuma mensagem em espera de acesso, terminando o ciclo.
- O *display* exibe a nova mensagem estabelecida como a mais prioritária, até não restarem mais mensagens para exibir, voltando assim ao menu principal estabelecido (funcionamento em ciclo).

A função representada não considera a receção de dados de distância do ponto atual do alerta em relação ao autocarro. Se alguma das condições anteriores não se verificar, a função começa novo ciclo.

Tem-se como exemplo prático a seguinte situação: O *display* da interface deve-se encontrar ligado, independentemente do menu ou tarefa que esteja a executar:



Figura 26: Ilustração do menu principal do software na HMI desenvolvida

A OBU recebe dois sinais de entrada no mesmo período de tempo, sendo eles a mensagem de alerta de aproximação de veículo prioritário (valor de prioridade 10) e a mensagem de alerta de trabalhos móveis (valor de prioridade 9). O programa ordena o *display* a mostrar a mensagem estabelecida como mais prioritária (durante 10 segundos, emitindo um alerta sonoro):



Figura 27: Ilustração da mensagem de alerta de aproximação de veículo prioritário (idioma em inglês)

Ao fim de 10 segundos a mensagem de valor de prioridade 10 desaparece e o *display* passa a mostrar a mensagem que se encontrava em espera (valor de prioridade 9) durante 10 segundos, emitindo um alerta sonoro:



Figura 28: Ilustração da mensagem de alerta de aproximação de obras móveis na estrada

Ao fim destes 10 segundos, como não existem mais mensagens em espera o *display* volta ao menu principal:



Figura 29: Ilustração do menu principal do software na HMI desenvolvida

Para além das condições de prioridade como critério de exibição de mensagens, para as mensagens de alerta de congestionamento, alerta de estrada fechada, alerta de via fechada e alerta de obras móveis, foi considerado como critério a distância do ponto atual destes alertas na rodovia até ao ponto de localização atual do autocarro (ver tabela 20). Os dados da distância correspondentes a cada mensagem alerta são recebidos através da RSU pela OBU instalada no autocarro, sendo a divulgação destes dados da responsabilidade de uma entidade como o Centro de Controlo de Tráfego. A Figura 30 representa o fluxograma utilizado para a caracterizar esta função.

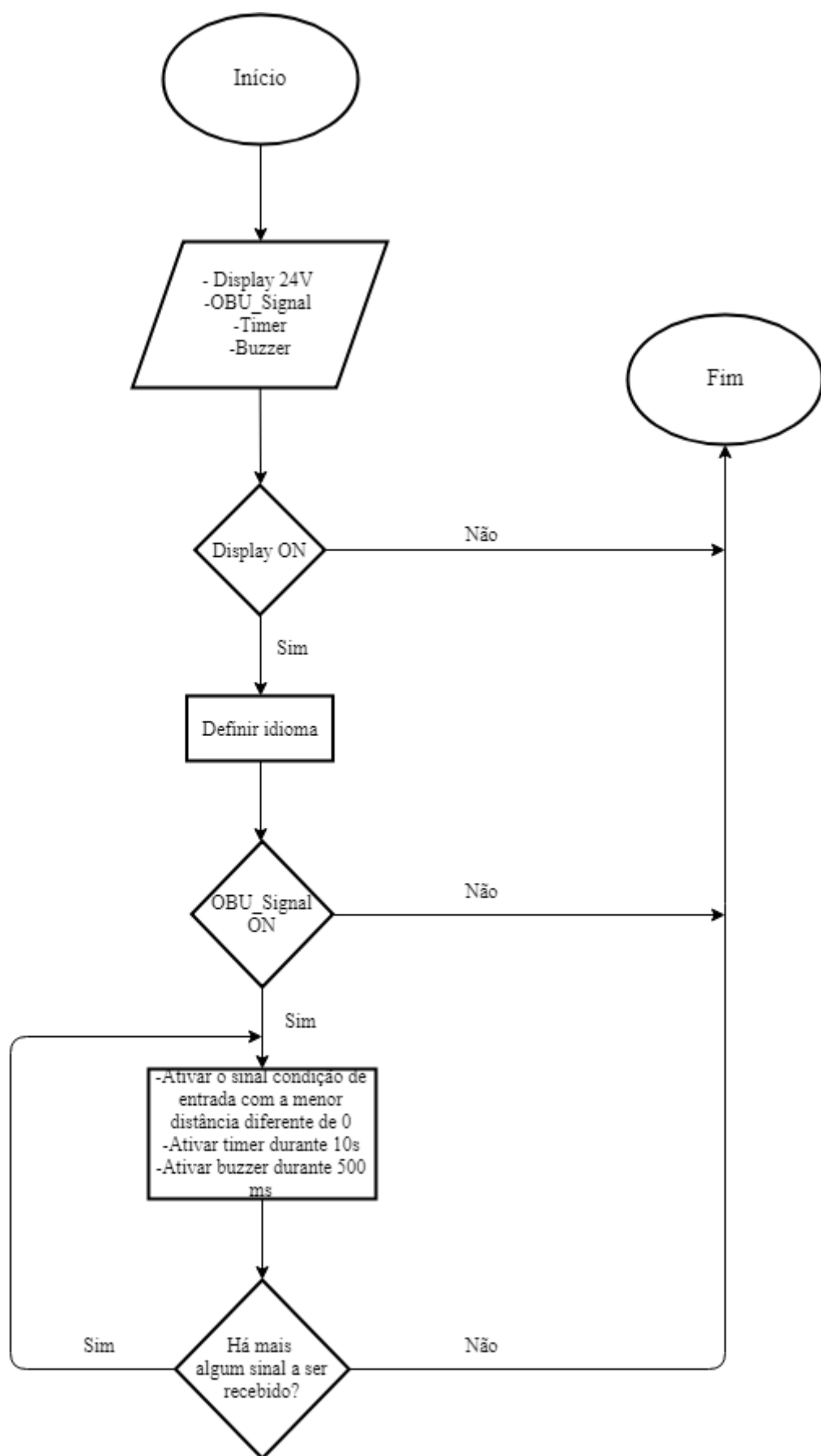


Figura 30: Fluxograma Função de distância

Identicamente ao fluxograma função de prioridade representado na Figura 25, este fluxograma função de distância contém a mesma estrutura de programa. Para iniciar este processo, o *display* da interface deve estar ligado (ON), obtendo o sinal de tensão de 24V das baterias auxiliares. É permitido ao motorista escolher se pretende ler a informação representada no *display* em língua portuguesa ou inglesa, funcionando o programa do mesmo modo independentemente da escolha de idioma. Caso seja recebido um sinal de entrada pela OBU (OBU_Signal ON), o programa deve ativar a condição de entrada definida como a que contém o menor valor numérico de dados de distância entre 1 até 2550 metros (caso haja mais do que uma condição de entrada a pedir acesso no mesmo período de tempo), ativar um alerta sonoro (*buzzer#500 ms*) com duração de 500 ms de forma a captar a atenção do motorista e mostrar a respetiva mensagem de auxílio no *display*. Esta mensagem é exibida durante 10 s (*timer#10 s*), ao fim dos quais a mensagem deixa de ser exibida, podendo:

- O *display* voltar ao menu principal estabelecido caso não haja mais nenhuma mensagem em espera de acesso, terminando o ciclo.
- O *display* exibir a nova mensagem estabelecida como a mais prioritária (relembrando que os critérios de prioridade neste caso funcionam em função da distância), até não restarem mais mensagens para exibir, voltando assim ao menu principal estabelecido (funcionamento em ciclo).

Paralelamente aos objetivos definidos nesta dissertação, foi implementada uma função extra onde o auxílio ao motorista continua a ser o âmbito. A função em questão consiste na ativação de uma câmara de filmar colocada na estrutura exterior traseira do autocarro que permite o auxílio de condução ao motorista sempre que este carrega no pedal de travão e no comando modo marcha-atrás (*reverse*). O fluxograma da Figura 31 ilustra esta função.

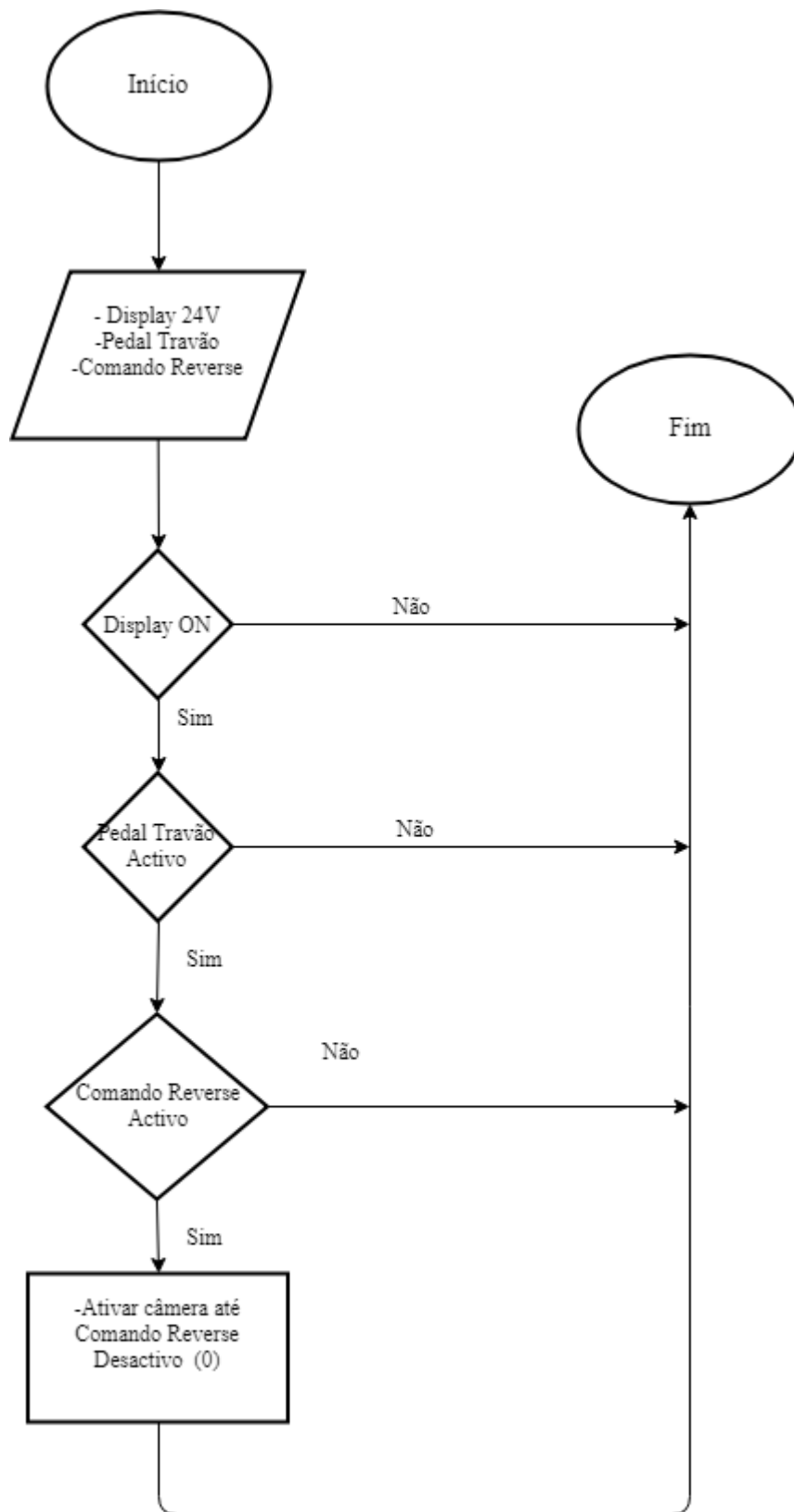


Figura 31: Fluxograma Função Marcha-Atrás

Para iniciar este processo, o *display* da interface deve estar ligado (ON), obtendo o sinal de tensão de 24V das baterias auxiliares. Sempre que o motorista ativar o pedal de travão (nível lógico 1) e ativar o botão de comando de marcha atrás (*Reverse*) a câmara de filmar é imediatamente ativada projetando a imagem que está a gravar para o *display* (Figura 32). Quando e sempre que o botão de comando de marcha-atrás (reverse) deixar de estar ativo (nível lógico 0) a projeção de imagem gravada desaparece do *display* e este retorna ao menu principal.



Figura 32: Imagem exemplo de câmara ON com comando de marcha-atrás ativa num autocarro

6.5. DEFINIÇÃO DE TEMPO DE CICLO

A definição de tempos de ciclo é de extrema importância para o correto funcionamento de *software*. Esta definição permite obter conhecimento de como se comportará o sistema aquando de uma ou mais alterações nas suas condições de entrada. Este comportamento compreende o tempo que demora a unidade de controlo eletrónica a ler entradas, executar o programa e atualizar as saídas (tempo de resposta). Esta característica encontra-se associada ao tempo de ciclo em que uma determinada tarefa pode ser executada. Para o correto funcionamento de uma função, o tempo de ciclo tem de ser superior ao tempo de demora de execução da própria função. Deste modo, evita-se a ocorrência de erros. O tempo de ciclo definido para o processamento do programa é de 250 ms. Este tempo consiste no tempo de ocorrência das funções até à execução da tarefa. Visto que com este tempo de ciclo, relativamente baixo, o programa corre normalmente e não bloqueia, garante-se a segurança dos ocupantes do veículo em simultâneo com a correta execução de programa.

6.6. TESTES AO SOFTWARE

A realização de testes ao *software* é de extrema importância de modo a verificar que os requisitos inicialmente propostos para o desenvolvimento de *software* foram cumpridos. Para tal, é proposto um método de testes ao *software*. O método escolhido consiste na definição de critérios nos quais o *software* deve obter aprovação de modo a ser permitida a sua futura instalação num autocarro. Na Figura 33 encontram-se ilustradas diferentes etapas de teste:

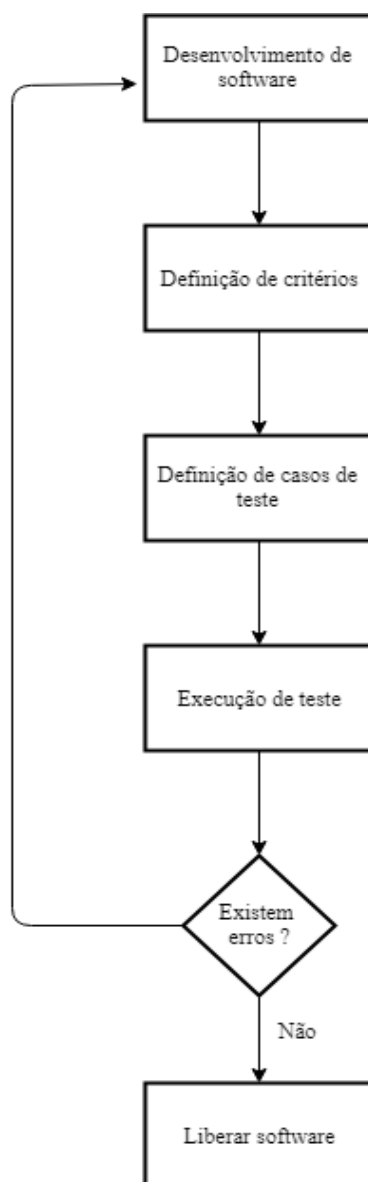


Figura 33: Ilustração de método de teste de software

Após o desenvolvimento de *software* é necessário definir critérios de concretização de teste para obter aprovação do teste em banca. De seguida, a definição de casos de teste serve

para executar o que se pretende avaliar, resultando na execução de teste. Caso não existam erros, pode-se documentar e liberar o *software*. Caso existam erros, estes devem ser documentados e todo o processo deve ser repetido até ser obtida aprovação final. Torna-se então necessário descrever detalhadamente todas as funções que se pretendem testar. Na Tabela 21 encontram-se definidos os critérios estabelecidos para avaliação de *software*.

Tabela 21: Definição de critérios de concretização de teste

Identificador	Critério de concretização de teste	Versão Software
ID1	Aprovação na verificação de função Prioridade e respetivo tempo de resposta	Codesys V3.5 SP9
ID2	Aprovação na verificação de função Distância e respetivo tempo de resposta	Codesys V3.5 SP9
ID3	Aprovação na verificação de função Marcha-Atrás e respetivo tempo de resposta	Codesys V3.5 SP9

Os critérios de concretização de teste ID1, ID2 e ID3 correspondem à aprovação na verificação de execução das funções Prioridade, Distância e Marcha-Atrás, conforme foi estabelecido no ponto 6.4. Na Tabela 22 encontram-se representados os diferentes casos de teste referentes ao critério de concretização de teste para cada um dos identificadores.

Tabela 22: Definição de casos de teste

Identificador	Tarefa	Descrição de procedimento	Variáveis	Entradas	Saídas	Resultado esperado	Resultado Obtido
ID1	Saída de sinal	Forçar a saída e entrada de sinal através de computador	OBU	Ligação PEAK CAN	Exibição de mensagens no display	Verificar se todas as condições de entrada/saída/ se reproduzem devidamente no display, bem como se a função de prioridades faz o cálculo automático de exibição de mensagens	✓
	Entrada de sinal						✓
	Erro	Forçar o erro de sinal através de computador					✓
	Não disponível						✓
ID2	Saída de sinal	Forçar a saída e entrada de sinal através de computador	OBU	Ligação PEAK CAN	Exibição de mensagens no display	Verificar se todas as condições de entrada/saída/ se reproduzem devidamente no display, bem como se a função de distância faz o cálculo automático de exibição de mensagens	✓
	Entrada de sinal						✓
	Erro	Forçar o erro de sinal através de computador					✓
	Não disponível						✓
ID3	Ativar câmara	Ao levar o comando reverse a nível lógico 1, ativar câmara	Reverse	Botão Reverse	Exibição de imagem no display	Após colocar autocarro em modo de marcha-atrás, projetar a imagem da câmara no display	✓

A Tabela 22 é compreendida pelos seguintes pontos:

- Identificador – identificadores dos critérios de concretização de testes.
- Tarefa – pontos específicos que se pretendem avaliar no programa.
- Descrição de procedimentos – Descrição das ações a se tomar para realizar a verificação de teste.
- Variáveis – Variáveis que devem ser “forçadas” para levar a cabo o teste.
- Entradas – Todas as entradas da unidade de controlo eletrónico.
- Saídas – Todas as saídas da unidade de controlo eletrónico.
- Resultado esperado – o que se espera obter como resultado do teste.
- Resultado obtido – o resultado que se obtém na realização de teste.

A análise à rede de comunicação CAN é realizada através de um adaptador, PEAK CAN (com porta USB), que faz a interligação entre um computador e o *display* da interface. O PEAK CAN representa a OBU na simulação de testes na bancada. Deste modo, é possível “forçar” a entrada e saída de todos os sinais (tarefas) para a realização de teste, através do envio das mensagens CAN no computador (entradas - ver ponto 6.3) que se reproduzem no *display* (saídas).

Em conjunto com os objetivos estabelecidos para esta dissertação (ponto 1.3), a identificação do problema proposto (ponto 1.4) e os testes realizados ao *software* (ponto 6.6) é possível analisar e afirmar que todos os pontos propostos foram cumpridos.

CAPÍTULO 7

7. CONCLUSÕES E PERSPETIVAS DE TRABALHOS FUTUROS

7.1. CONCLUSÃO

Tendo em vista o trabalho realizado e os factos que o sustentam é possível concluir que a evolução tecnológica, no que concerne a comunicações entre veículos e entre veículos e infraestruturas, é uma realidade atual que pode e deve auxiliar o modo de condução de um motorista. O projeto realizado apresenta-se ao motorista como uma prevenção de alertas/perigos, podendo levar à redução de sinistros que não seriam possíveis de evitar sem uma ferramenta de auxílio eficaz.

Em termos de ergonomia, o modo em como foi desenvolvido e implementado o *software* da HMI permite captar a atenção do motorista sem privar das funções que o mesmo esteja a desempenhar num dado momento. A informação revelada é útil e de fácil compreensão, permitindo com que seja simples a forma de abordagem à condução aquando de um alerta emitido pelo *display*.

O protocolo de comunicação CAN mostra-se bastante eficaz para o uso em âmbito automóvel. A partilha de informação em tempo real em veículos automóveis, as altas velocidades de transmissão de dados, a fiabilidade e a possibilidade de escrever milhares de mensagens num identificador são factos que o comprovam. Ao utilizar um único identificador foi possível codificar todas as mensagens pretendidas para este projeto, sobrando ainda 1 *byte* do identificador para possível posterior utilização. Também a metodologia utilizada para criação da biblioteca CAN na codificação das mensagens utilizadas pode ser expandida e aproveitada. Esta metodologia considerou todas as entradas e saídas de sinais, bem como erros e não disponibilidades, fazendo com que a comunicação de dados ocorra de uma forma segura e caracterize possíveis erros.

Como *software* de interligação com o CAN, o CoDeSys mostrou-se como uma ferramenta precisa e útil. A possibilidade de escrever código em estilo de texto estruturado revelou-se um sucesso, permitindo ao mestrando estruturar de uma forma objetiva o funcionamento do programa.

A formação académica obtida ao longo destes anos foi essencial como forma de ultrapassar as dificuldades notadas aquando das primeiras abordagens com este *software* e respetivo

estilo de programação. As funções desenvolvidas tiveram a aprovação da entidade empresarial e para fácil compreensão todas as linhas de código foram devidamente comentadas. Os testes de análise do *software* desenvolvido consideraram e verificaram todos os requisitos definidos. No fim, nota-se que foi possível concluir com sucesso todos os objetivos de programação para a interface se comportar conforme fora pretendido.

O estudo e análise ao tipo de comunicação entre veículos e entre veículos e infraestruturas mostrou-se muito interessante. Ao abordar uma tipologia de comunicação futurista a curto prazo, foi rapidamente compreendido que este tipo de comunicações traz inúmeras vantagens em termos de eficiência energética, mas sobretudo em termos de segurança. Conforme abordado ao longo desta dissertação, tudo aponta para que este método de comunicação entre veículos reduza de uma forma considerável o número de acidentes rodoviários, sendo, por isso, uma aposta de grandes marcas para implementação nos seus veículos no presente. As mensagens elaboradas neste projeto permitem dar o apoio necessário à tomada de decisão do motorista, podendo este antecipar cenários indesejáveis, que não seriam previstos sem este apoio.

Por todos os pontos evidenciados ao longo deste relatório pode concluir-se que o projeto realizado contém valor para a entidade empresarial e tornou-se, assim, numa mais valia para os projetos que possam a vir ser desenvolvidos e implementados pela CaetanoBus.

7.2. PERSPETIVAS DE TRABALHOS FUTUROS

No âmbito deste projeto podem considerar-se como perspetivas de trabalhos futuros a implementação em escala dos autocarros com instalação de OBU, que permita a comunicação com a infraestrutura. Esta comunicação pode ser ampliada em termos de mensagens que se pretendam comunicar e de diferentes tipos de dados que se pretendam revelar. Pode-se dar como exemplo, uma RSU que passe a informação à OBU, instalada num autocarro, em como o próximo cruzamento de estrada está desimpedido, podendo o autocarro avançar sem a total paragem do mesmo.

Para além de diferentes tipos de mensagens entre a infraestrutura e o autocarro, pode ser encarada uma possível comunicação entre autocarros (V2V). Ainda na comunicação DSRC, este tipo de comunicação é possível desde que em curtas médias distâncias entre veículos. Um exemplo de troca de dados entre V2V poderia ser uma frenagem brusca de

um dado autocarro que se encontrasse algumas dezenas de metros à frente de um outro autocarro. Por fim, existe ainda em estudo uma possível implementação de comunicação LTE (idêntica à usada na comunicação entre *smartphones*), que pode vir a ser utilizada em veículos automóveis. Este tipo de comunicação ainda se encontra numa fase precoce, não tendo ainda uma norma estabelecida que a defenda. As vantagens podem-se encontrar em alcances de distância mais elevados para efeitos de comunicação e na redução de equipamentos a instalar.

Referências Bibliográficas

- [1] S. C. Service, “International Standard - ISO 9241-110,” *Communication*, vol. 2006, pp. 1–8, 2003.
- [2] J. A. Adams, “Critical Considerations for Human-Robot Interface Development,” *AAAI Fall Symp. Hum. Robot Interact. Tech. Rep. FS-02-03*, pp. 1–8, 2002.
- [3] F. M. Guimarães, “O que é Design Centrado no Usuário?,” 2017. [Online]. Available: <https://medium.com/aela/o-que-é-design-centrado-no-usuário-11a9c13c3a2f>.
- [4] H. E. McLoone, M. Jacobson, C. Hegg, and P. W. Johnson, “User-centered design and evaluation of a next generation fixed-split ergonomic keyboard,” *Work*, vol. 37, no. 4, pp. 445–456, 2010.
- [5] D. a Norman, “Cognitive Engineering,” *User-centered Syst. Des.*, pp. 31–61, 1986.
- [6] M. Ramos, “IHC: Engenharia Cognitiva.” [Online]. Available: <http://www.marceloramos.com.br/publicacao/50/ihc-engenharia-cognitiva>.
- [7] E. D. T. Wünsche, “How the CAN network functions.” [Online]. Available: <http://www.ems-wuensche.com/about-can/can-functionality-info.html>.
- [8] José and Rufino, “An Overview of the Controller Area Network,” pp. 1–10, 1997.
- [9] P. Venda, “Controller Area Network - Fundamentos,” p. 10, 2003.
- [10] S. Corrigan, “Introduction to the Controller Area Network (CAN),” *Texas Instruments*, no. August 2002, pp. 1–17, 2016.
- [11] “Protocolo de comunicação CAN,” pp. 37–61.
- [12] CAN CiA Organization, “CAN data link layers in some detaille.” [Online]. Available: <https://www.can-cia.org/can-knowledge/can/can-data-link-layers/>.
- [13] R. Bosch, “CAN Specification Version 2.0,” *Rober Bousch GmbH, Postfach*, vol. 300240, p. 72, 1991.
- [14] “CoDeSys.” [Online]. Available: <https://www.codesys.com/>.
- [15] EATON, “Programação Básica CoDeSys,” 2012.
- [16] “CODESYS Runtime.” [Online]. Available: <https://www.codesys.com/products/codesys-runtime.html>.
- [17] “CODESYS Visualization.” [Online]. Available: <https://www.codesys.com/products/codesys-visualization.html>.
- [18] “CODESYS Fieldbus.” [Online]. Available: <https://www.codesys.com/products/codesys-fieldbus.html>.
- [19] “CODESYS Motion + CNC.” [Online]. Available: <https://www.codesys.com/products/codesys-motion-cnc.html>.
- [20] “Waymo.” [Online]. Available: <https://waymo.com/>.

- [21] S. Sill, “DSRC: The Future of Safer Driving.” [Online]. Available: https://www.its.dot.gov/factsheets/dsrc_factsheet.htm.
- [22] D. K. Vladimir Vishnevsky, “Distributed Computer and Communication Network,” 2015. [Online]. Available: <https://books.google.pt/books?id=xM-zCwAAQBAJ&pg=PA5&lpg=PA5&dq=Five+to+10+hops+on+the+network+would+gather+traffic+conditions+a+mile+ahead&source=bl&ots=lckgS9N4OM&sig=sI9-jDELFrpJ0rB-hSsjuo1APdg&hl=pt-PT&sa=X&ved=0ahUKEwjF5ePa67TaAhWF0RQKHe6zDd4Q6AEILzA>.
- [23] “Car2Car.org.” [Online]. Available: <https://www.car-2-car.org/index.php?id=5>.
- [24] “No Title.” [Online]. Available: <http://www.circulaseguro.pt/veiculos-e-tecnologia/car2car-quais-as-vantagens-de-estar-ligado>.
- [25] J. Harding *et al.*, “Vehicle-to-Vehicle Communications : Readiness of V2V Technology for Application,” no. August, p. 327, 2014.
- [26] C. Manifesto, “CAR 2 CAR Communication Consortium Manifesto,” *System*, p. 94, 2007.
- [27] S. Asthana and D. Ph, “Communication Technology,” no. September, p. 2015, 2013.
- [28] “Cooperative ITS.” [Online]. Available: http://www.toyota-global.com/innovation/intelligent_transport_systems/infrastructure/.
- [29] B. Howard, “V2X (vehicle-to-everything),” 2017. [Online]. Available: <https://www.extremetech.com/extreme/243542-v2x-conquers-bad-curves-construction-zones-red-lights>.
- [30] “U.S. DOT advances deployment of Connected Vehicle Technology to prevent hundreds of thousands of crashes.”

Anexo A

Biblioteca criada para a entidade empresarial referente à codificação de mensagens em CAN

“[...]”

General description

A communication system between the vehicles and the road network is in development, which allows the exchange of information between the two, that will allow the driver to make some decisions regarding his/her driving.

In view of this, an interface between the driver and the data acquisition systems of the vehicle and the road network has to be developed, presenting the information in a structured, synthetic, simple and easy to understand way, with different levels of priority.

System / Project name

OBU_VEH1 is the name of this project with a major focus in Vehicle to Infrastructure Communication.

Goals

The final goal is to develop an HMI interface between data acquisition systems (On Board Unit (OBU) and vehicle) and the driver of an electric bus. To do this, it will need to:

- Define the relevant information and its level of priority.
- Define the most appropriate way to present the information to the driver.
- Show a set of masks on a screen that present the desired information in a coherent way.

The major goal is to create an interface that will be useful in terms of road security and comfort for the driver, while at the same time it will reduce the cost of energy consumed.

Environment of the system

It refers to an intelligent transport system where vehicles and infrastructure systems are interconnected (Vehicle to Infrastructure Communication). This interconnection will provide accurate real-time knowledge of the current traffic situation across the road network.

An architecture system based on V2I should contain the following points:

- Vehicle On-Board Unit or Equipment (OBU or OBE).
- Roadside Unit or Equipment (RSU or RSE).
- Safe Communication Channel.

In terms of road safety, this project allows to show on the display the following hypotheses (and more):

- Hazard warning (eg: Accident at 500m);
- Notice of passing priority vehicles;
- Speed warning (ex: The traffic light will turn red. Please slow down);

Message / Addresses	Used Signal (SPN)	Comments	Conditions
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Traffic Jam Ahead Alert	Byte 1, bit 1-2 Resolution: 1 Offset: 0 Range: 0-3	00 = Message Alert OFF 01 = Message Alert ON 10 = Error 11 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Lane Closure Alert	Byte 1, bit 3-4 Resolution: 1 Offset: 0 Range: 0-3	00 = Message Alert OFF 01 = Message Alert ON 10 = Error 11 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Road Closure Alert	Byte 1, bit 5-6 Resolution: 1 Offset: 0 Range: 0-3	00 = Message Alert OFF 01 = Message Alert ON 10 = Error 11 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Mobile Works Alert	Byte 1, bit 7-8 Resolution: 1 Offset: 0 Range: 0-3	00 = Message Alert OFF 01 = Message Alert ON 10 = Error 11 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Weather Condition Warning Alert	Byte 2, bit 1-4 Resolution: 1 Offset: 0 Range: 0-15	0000 = Message Alert OFF 0001 = Rain Alert ON 0010 = Wind Alert ON 0011 = Grip Alert ON 0100 = Visibility Alert ON 0101 = Error 0111 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Dynamic Speed Alert	Byte 2, bit 5-6 Resolution: 1 Offset: 0 Range: 0-3	00 = Message Alert OFF 01 = Message Alert ON 10 = Error 11 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Priority Vehicle Alert	Byte 2, bit 7-8 Resolution: 1 Offset: 0 Range: 0-3	00 = Message Alert OFF 01 = Message Alert ON 10 = Error 11 = Not available
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Velocity	Byte 3, bit 1-8 Resolution: 1 Offset: 0 Range: 0-255	Speed (km/h)
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Traffic Jam Distance	Byte 4, bit 1-8 Resolution: 10 Offset: 0 Range: 0-2550	Distance (m)
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Lane Closure Distance	Byte 5, bit 1-8 Resolution: 10 Offset: 0 Range: 0-2550	Distance (m)
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Road Closure Distance	Byte 6, bit 1-8 Resolution: 10 Offset: 0 Range: 0-2550	Distance (m)
OBU_VEH1 Identifier: 18 FF 01 28h Rep.rate: 250 ms	Mobile Works Distance	Byte 7, bit 1-8 Resolution: 10 Offset: 0 Range: 0-2550	Distance (m)

[...]"

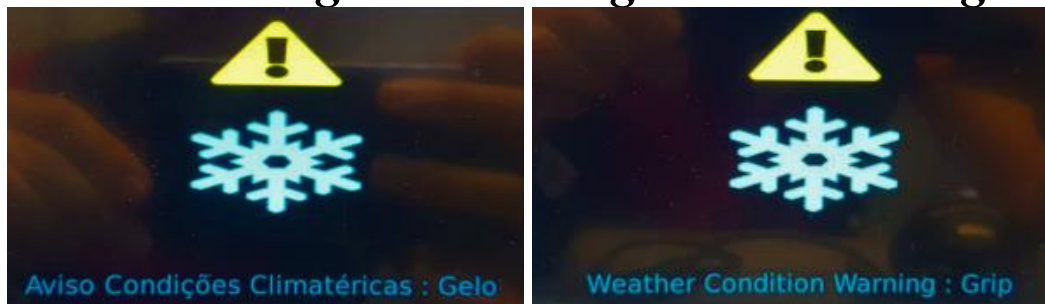
Anexo B - Imagem de mensagem de alerta de trânsito



Anexo C - Imagem de mensagem de alerta de vento



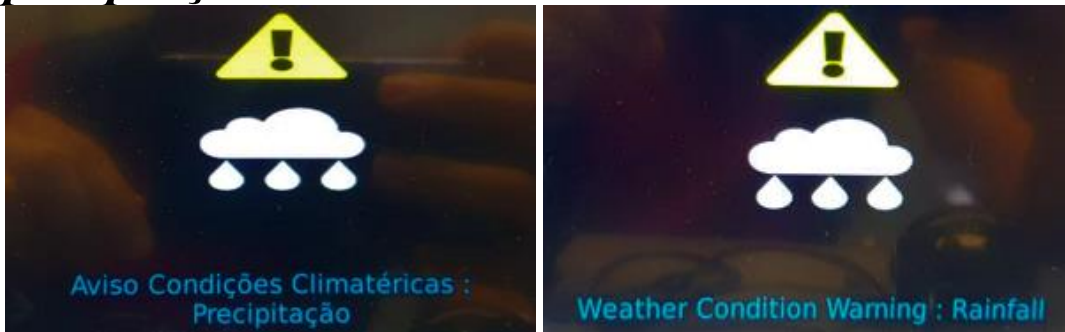
Anexo D - Imagem de mensagem de alerta de gelo



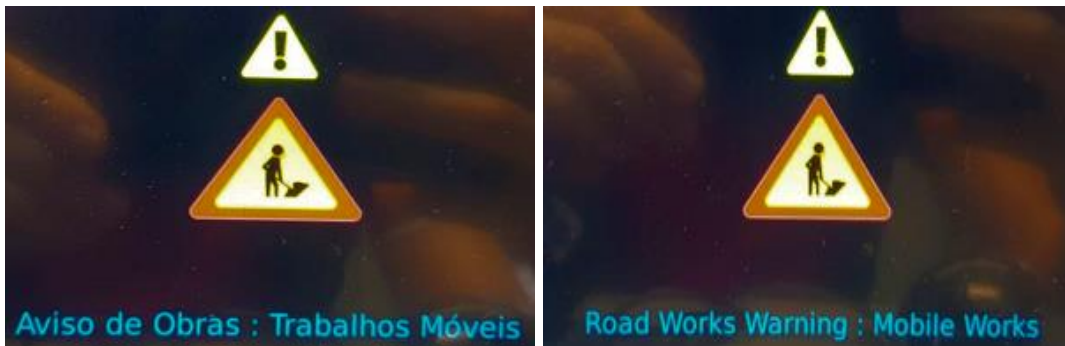
Anexo E - Imagem de mensagem de alerta de visibilidade



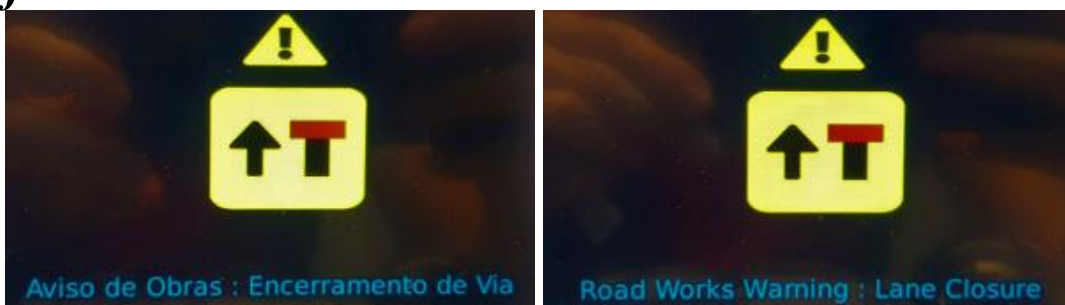
Anexo F - Imagem de mensagem de alerta de precipitação



Anexo G - Imagem de mensagem de alerta de obras móveis



Anexo H - Imagem de mensagem de alerta de via fechada



Anexo I - Imagem de mensagem de alerta de estrada fechada



Anexo J - Imagem de mensagem de alerta de veículo prioritário



Anexo L - Imagem de mensagem de alerta de velocidade otimizada

