

Solução de Iluminação Automóvel Baseada na Rede CAN

JOÃO PEDRO ESTÊVÃO FRANCISCO

(Licenciado em Engenharia Eletrotécnica)

Dissertação para a obtenção do grau de Mestre em
Engenharia Eletrotécnica – Ramo de Automação e Eletrónica Industrial

Orientadores:

Prof. Doutor Armando José Leitão Cordeiro
Prof^ª. Doutora Maria da Graça Vieira de Brito Almeida

Júri:

Presidente: Prof. Doutor Luís Manuel dos Santos Redondo
Vogais: Prof^ª. Doutora Carla Solange Pires Correia Viveiros
Prof. Doutor Armando José Leitão Cordeiro

Setembro de 2016

Dissertação realizada em regime de coorientação por:

Doutor Armando José Leitão Cordeiro

Professor Adjunto a Área Departamental de
Engenharia Eletrotécnica Energia e Automação

Instituto Superior de Engenharia de Lisboa

e

Doutora Maria da Graça Vieira de Brito Almeida

Professora Adjunta da Área Departamental de
Engenharia Eletrotécnica Energia e Automação

Instituto Superior de Engenharia de Lisboa

Resumo

Esta dissertação trata da integração de uma solução de iluminação automóvel baseada na rede CAN, cobrindo as várias etapas do desenvolvimento da aplicação, desde a conceção da solução à descrição do *software* desenvolvido, passando pela explicação da seleção e da implementação do *hardware*. Inclui-se também um breve estudo sobre os sistemas de iluminação nos automóveis.

Para fundamentar esta dissertação, para além da rede CAN são apresentadas várias redes indispensáveis nos automóveis atuais, tais como a rede LIN, a rede FlexRay e a rede MOST, uma das razões para a escolha destas redes foi cada uma representar uma classe SAE distinta, classes essas que também são apresentadas.

Os resultados experimentais obtidos com a aplicação desenvolvida permitem concluir sobre o desempenho da rede CAN e dos periféricos associados, bem como da comunicação SPI.

Palavras-chave

Iluminação automóvel, rede de comunicação CAN, circuitos embebidos, comunicação SPI, ecrã tátil, fio eletroluminescente.

Abstract

This dissertation treats the integration of an automotive lighting solution based on the CAN network, covering the various stages of application development, from solution conception to software developed description, through the explanation of the selection and implementation of hardware. It also includes a brief study of the lighting systems in automobiles.

In support of this dissertation, shows in addition to the CAN network a variety of indispensables networks in today's automobiles, such as the LIN network, FlexRay network and MOST network, one of the reasons for choosing these networks was each represent a distinct SAE class, those classes are also presented.

The experimental results obtained in the application developed allow to conclude on the performance of the CAN network and associated peripherals, and SPI communication.

Keywords

Automotive lighting, CAN network, embedded circuits, SPI communication, touch screen, electroluminescent wire.

Agradecimentos

Ao Doutor Armando José Leitão Cordeiro, Professor Adjunto da Área Departamental de Engenharia Eletrotécnica Energia e Automação do Instituto Superior de Engenharia de Lisboa, responsável como orientador científico, pelo apoio, orientação, conhecimentos transmitidos e pelo ânimo que me deu durante todo o tempo de desenvolvimento desta dissertação de mestrado.

À Doutora Maria da Graça Vieira de Brito Almeida, Professora Adjunta da Área Departamental de Engenharia Eletrotécnica Energia e Automação do Instituto Superior de Engenharia de Lisboa, responsável como orientadora científica, pelo apoio, pelos conselhos, por todo o tempo dedicado à leitura e análise desta dissertação e pela sua disponibilidade que demonstrou durante o período de orientação desta dissertação de mestrado.

Aos meus pais por todo o apoio que me deram, sem eles nunca poderia ter chegado aqui e à minha restante família, por todo apoio que foram dando ao longo de todo o meu percurso académico.

Aos meus colegas do ISEL, com quem convivi e passei bons e maus momentos, desde a licenciatura ao mestrado, pela colaboração e ajuda a ultrapassar esta etapa da minha vida. Espero que nos voltemos a encontrar por aí.

A todos aqueles que, apesar de não terem sido citados anteriormente, contribuíram direta ou indiretamente durante o meu percurso e durante a realização desta dissertação de mestrado.

Índice

Resumo	i
Abstract	ii
Agradecimentos	iii
Lista de Figuras	v
Lista de Tabelas	vii
Lista de Siglas	viii
1. Introdução	1
1.1 Enquadramento e motivação	2
1.2 Objetivos	3
1.3 Estrutura da dissertação e convenções	4
2. Estado da Arte	6
2.1 Introdução	6
2.2 Conceitos gerais	8
2.3 Classes das redes dos automóveis	11
2.4 Rede CAN	14
2.5 Rede LIN	22
2.6 Rede FlexRay	29
2.7 Rede MOST	36
2.8 Sistemas de iluminação nos automóveis	44
3. Solução Desenvolvida	46
3.1 Descrição geral	46
3.2 Escolha do material	48
3.3 Implementação do <i>hardware</i>	54
3.4 Programação dos dispositivos	56
3.4.1 Microcontroladores dsPIC33EP256GP502	56
3.4.2 Ecrã táctil eDIP160B-7LWTP	64
4. Resultados Experimentais	68
5. Conclusões	72
5.1 Perspetivas de desenvolvimento futuro	73
Bibliografia	75
Anexo – Projeto das placas de circuito impresso	78

Lista de Figuras

Figura 2.1 – Correspondência entre o modelo de referência normalizado e a rede CAN de acordo com ISO 11898.	14
Figura 2.2 – Rede CAN e a sua interface.....	15
Figura 2.3 – Exemplo de uma fase de arbitragem entre 3 nós CAN.	17
Figura 2.4 – Estrutura da trama de dados.	18
Figura 2.5 – Diagrama das regras dos contadores de erros.....	20
Figura 2.6 – Diferenças entre os protocolos CAN e CAN FD.	21
Figura 2.7 – Arquitetura mestre-escravo da rede LIN.	24
Figura 2.8 – Distribuição de tramas num ciclo de comunicação LIN.	25
Figura 2.9 – Estrutura de espaços de uma trama LIN.....	26
Figura 2.10 – Modelo do estado de comunicação de um escravo LIN.....	29
Figura 2.11 – Combinação de topologias ativa e passiva.	32
Figura 2.12 – Princípio da comunicação FlexRay.....	33
Figura 2.13 – Fluxo de comunicação no segmento dinâmico.....	34
Figura 2.14 – Estrutura de uma mensagem FlexRay.	35
Figura 2.15 – Hierarquia do sistema MOST.....	38
Figura 2.16 – Diferença entre as arquiteturas NIC e INIC.	40
Figura 2.17 – Composição de uma trama MOST.	41
Figura 2.18 – Canais MOST à frequência de 48 KHz.	42
Figura 2.19 – Bloco MOST com a mensagem de controlo.....	43
Figura 3.1 – Diagrama geral simplificado da solução proposta para o controlo da iluminação baseado na rede CAN sem referência dos componentes.	46
Figura 3.2 – Diagrama geral simplificado da solução proposta para o controlo da iluminação baseado na rede CAN com referência dos componentes.....	48
Figura 3.3 – Aparência do dsPIC33EP256GP502.	49
Figura 3.4 – Aparência do MCP2562.	50
Figura 3.5 – Aparência do CL7à esquerda e do CL330 em placa breakout à direita.	51
Figura 3.6 – Diagrama de blocos funcional do HV852.	52
Figura 3.7 – Aparência do HV852 em placa breakout.	52
Figura 3.8 – Aparência do MCP1755S à esquerda e do MIC29150 à direita.....	53
Figura 3.9 – Aparência do eDIP160B-7LWTP.	54
Figura 3.10 – Projeto da montagem no Fritzing.	55
Figura 3.11 – Fotografia da aplicação desenvolvida.	55
Figura 3.12 – Fluxogramas geral dos microcontroladores, à esquerda do módulo central e à direita dos restantes módulos.	57
Figura 3.13 – Fluxograma da configuração do módulo CAN.	58
Figura 3.14 – Código ilustrativo da seleção de pinos.	59
Figura 3.15 – Código ilustrativo da inicialização da rede CAN.	59
Figura 3.16 – Tempo de bit da comunicação CAN.	60
Figura 3.17 – Código ilustrativo da definição de mascaras.	60
Figura 3.18 – Código ilustrativo da definição de filtros.	61
Figura 3.19 – Código ilustrativo da passagem para modo de funciona normal.....	63
Figura 3.20 – Código ilustrativo do início do programa.....	64

Figura 3.21 – Código ilustrativo da macro MnAutoStart.	65
Figura 3.22 – Aparência da página inicial do ecrã.....	65
Figura 3.23 – Código ilustrativo da macro de toque 2.....	66
Figura 3.24 – Aparência da 1ª página de botões do ecrã.	67
Figura 3.25 – Aparência da 2ª página de botões do ecrã.	67
Figura 4.1 – Sinais à entrada (em baixo) e à saída (em cima) do transceptor.	68
Figura 4.2 – Mensagem do módulo central e respostas.....	69
Figura 4.3 – Ciclo completo de mensagens SPI.	69
Figura 4.4 – Pormenor das mensagens SPI com a frequência do clock.	70
Figura 4.5 – Transição de médios para máximos.	71
Figura 4.6 – Intervalo de tempo entre ação e atuação.....	71

Lista de Tabelas

Tabela 1 – Comparação das características das redes.....	7
Tabela 2 – Características das classes SAE.	12
Tabela 3 – Atribuição dos identificadores das mensagens.	62

Lista de Siglas

ACK	Reconhecimento
ARQ	<i>Automatic Repeat Request</i>
AUTOSAR	<i>Automotive Open System Architecture</i>
BCC	<i>Block Checksum</i>
CAN	<i>Controller Area Network</i>
CM	<i>Connection Master</i>
CRC	<i>Cyclic Redundancy Check</i>
CSMA/ BA	<i>Carrier Sense Multiple Access with Bitwise Arbitration</i>
D2B	<i>Digital Data Bus</i>
DLC	Código de comprimento de dados
DSP	<i>Digital Signal Processor</i>
ECU	Unidades de controlo eletrónico
EHC	<i>External Host Controller</i>
EMC	<i>Electromagnetic Compatibility</i>
EOF	Fim da trama
ESD	Electrostatic Discharge
ESP	<i>Electronic Stability Program</i>
FIFO	<i>First In First Out</i>
HMI	<i>Human Machine Interface</i>
ID	Identificador
IDE	Identificador de extensão
INIC	<i>Intelligent Network Interface Controller</i>
ISO	<i>International Standardization Organization</i>
LDF	<i>LIN Description File</i>

LED	<i>Light Emitting Diode</i>
LIN	<i>Local Interconnected Network</i>
LSB	<i>Least Significant Bit</i>
MOST	<i>Media Oriented Transport System</i>
MSOP	<i>Plastic Micro Small Outline</i>
NCF	<i>Node Capability File</i>
NIC	<i>Network Interface Controller</i>
NIT	<i>Network Idle Time</i>
OLED	<i>Organic LED</i>
PPS	<i>Peripheral Pin Select</i>
PWM	<i>Pulse Width Modulation</i>
RTR	Pedido de transmissão remota
SAE	<i>Society of Automotive Engineers</i>
SFR	<i>Special Function Registers</i>
SOF	Início da trama
SOIC	<i>Plastic Small Outline</i>
SOT-223	<i>Plastic Small Outline Transistor</i>
SPDIP	<i>Skinny Plastic Dual In-Line</i>
SPI	Serial Peripheral Interface
TDMA	<i>Time Division Multiple Access</i>
TO-220	<i>Plastic Transistor Outline</i>
TQ	<i>Time Quanta</i>
TTCAN	<i>Time Triggered CAN</i>
TTP	<i>Time Triggered Protocol</i>

1. Introdução

A história recente do automóvel é caracterizada por uma forte aposta nos sistemas eletrônicos. Com o surgimento de novas tecnologias de semicondutores e a proliferação de sistemas elétricos/eletrônicos programáveis esta tendência no sector automóvel era praticamente inevitável tal como aconteceu em muitos outros sectores, desde o mais simples aparelho doméstico até ao mais avançado e complexo sistema aeronáutico. Apesar de diversas questões sobre segurança se colocarem com frequência em resultado de alguns acidentes em veículos automóveis e sistemas de transporte em geral, comprovadamente provocados por falhas em sensores e controladores eletrónicos existentes a bordo, é do consenso geral que estes sistemas apresentam mais vantagens e comodidades do que problemas de segurança acrescidos.

Esta crescente massificação de componentes eletrónicos no sector automóvel resulta também de estratégias competitivas entre as diferentes marcas com o objetivo de aumentar as suas vendas e por outro lado da necessidade de satisfazer as exigências cada vez maiores dos clientes em termos de acesso a novas tecnologias. Acrescem ainda as exigências dos legisladores sobre a redução das emissões, dos consumos e aumento da sua eficiência energética. Mesmo os veículos automóveis considerados de gama mais baixa possuem atualmente entre 30 a 50 unidades de controlo eletrónicas (ECU) integrados no motor, chassis, portas, bancos, sistemas de iluminação, sistemas de som, sistemas de segurança e retenção, entre muitos outros. Isto significa que a maioria dos veículos atualmente executam dezenas de milhões de linhas de código de *software*, diagnosticando ou controlando quase tudo de forma direta ou indireta. Estima-se que atualmente o custo médio do *software* e da eletrónica esteja entre 35% a 40% do custo total de produção do automóvel. Estima-se também que nos próximos 10 anos estes valores possam subir para 50% no caso dos veículos convencionais e 80% para os híbridos [1].

Além de diagnosticar e controlar a grande maioria dos sistemas usados a bordo, os veículos estão a começar a entrar numa nova era, analisando também uma parte do mundo que os rodeia. Utilizam, por exemplo, a tecnologia aplicada aos radares para detetar a presença de objetos externos, lasers para medir a distância entre veículos, sistemas de visão para reconhecer pessoas, animais ou sinalização de trânsito, ultrassons para detetar objetos durante o estacionamento, etc. Alguns fabricantes de automóveis afirmam que a tendência

será para partilhar informações externas ao veículo e, em seguida, explorar essa informação para melhorar a segurança. Por exemplo, os veículos poderão comunicar entre si indicando se há lençóis de água, gelo na estrada ou mesmo a existência de um acidente [1].

A integração da maioria dos componentes eletrónicos nos veículos automóveis só é possível devido à existência de diversas soluções de redes de comunicação digital. Estas possibilitam a coexistência de forma organizada de todos estes elementos utilizando um número bastante reduzido de cablagens, permitindo ao condutor ter acesso a um diverso leque de possibilidades de configuração de modos de condução, navegação, iluminação, entretenimento e falhas no seu veículo num ecrã ou ao alcance de um simples toque num botão. É neste área que se pretende conduzir esta dissertação de mestrado, mostrando as potencialidades de uma das redes de comunicação digital mais utilizadas nos automóveis, a rede CAN.

1.1 Enquadramento e motivação

A rede CAN é uma rede de comunicação digital utilizada em alguns modelos de automóveis há já bastantes anos. No entanto, apesar de ter sido concebida originalmente para este efeito migrou também rapidamente para a área da Automação Industrial. Esta rede de campo em Automação é uma das redes que faz parte do conteúdo da disciplina de Redes de Automação e Supervisão, integrada no plano de estudos do Mestrado em Engenharia Eletrotécnica – Ramo de Automação e Eletrónica Industrial. Deste modo, justifica-se plenamente a realização desta dissertação de mestrado onde se efetua um estudo mais aprofundado do seu funcionamento, juntamente com o desenvolvimento de uma aplicação concreta que se considera ter valor técnico-científico relevante para a obtenção do grau de Mestre em Engenharia Eletrotécnica.

Nos últimos anos tem havido também uma preocupação crescente com soluções de iluminação mais eficientes em termos luminotécnicos e energéticos com a utilização de tecnologias Xénon, Néon, LED e eletroluminescente. Muitos fabricantes de automóveis têm apostado também em algumas destas tecnologias como forma de convencer por um lado os clientes e a sociedade das preocupações das marcas em relação às questões ambientais e por outro como forma de renovar esteticamente os seus novos modelos dando-lhe um ar substancialmente mais moderno. Optou-se então por juntar na mesma dissertação

dois temas que são bastantes atuais e que despertam a atenção de muitos engenheiros e amantes do mundo motorizado.

A opção por este tema para a realização da dissertação deve-se também em grande parte a uma preferência pessoal por esta área da engenharia eletrotécnica, aliada ao gosto pelo desenvolvimento de *hardware* e *software* para aplicações de automação que envolvam redes de campo.

1.2 Objetivos

O principal objetivo desta dissertação consiste em desenvolver uma solução de iluminação automóvel baseada na rede CAN, que possibilite o controlo das diversas luzes dos quatro faróis de um veículo convencional.

Para realizar o controlo será necessário implementar uma rede CAN com recurso a transdutores e microcontroladores em cada um dos 5 nós da rede, correspondendo aos 4 módulos descentralizados, um em cada canto do veículo para a iluminação, e ao módulo central para o comando. A atuação será realizada através de botões mecânicos ou num ecrã táctil. Esse ecrã táctil transmitirá informação ao microcontrolador através de uma comunicação SPI (*Serial Peripheral Interface*).

Pesquisar sobre as várias redes presentes nos automóveis bem como a sua normalização, o funcionamento dos vários circuitos embebidos serão tarefas a empreender.

Para cumprir o objetivo principal desta dissertação foi necessário explorar e desenvolver alguns objetivos parciais, tais como:

- Estudar as várias redes existentes nos automóveis, de forma a perceber as suas vantagens e desvantagens, e as aplicações a que se destinam;
- Compreender todos os elementos necessários para a implementação da aplicação, desde *hardware* a *software*;
- Desenvolver um protótipo capaz de controlar a iluminação automóvel com precisão e em tempo real;
- Analisar o desempenho do protótipo desenvolvido de modo a descobrir as suas várias limitações;
- Comparar os resultados obtidos com as expectativas para assegurar o cumprimento dos objetivos propostos.

1.3 Estrutura da dissertação e convenções

Esta dissertação encontra-se organizada em cinco capítulos, todos eles divididos em diversos subcapítulos. À exceção do presente capítulo e do capítulo 5, reservado às conclusões, todos os outros apresentam uma estrutura semelhante, com início numa breve introdução e, de um modo geral, finalizando com conclusões. Os capítulos que constituem esta dissertação são os seguintes: Capítulo 1 – Introdução; Capítulo 2 – Estado da Arte; Capítulo 3 – Solução Desenvolvida; Capítulo 4 – Resultados Experimentais; Capítulo 5 – Conclusões.

O capítulo 1 inicia-se com uma introdução ao tema da dissertação, realçando a sua importância e enquadramento na área da engenharia em que se insere. Definem-se de seguida, os principais objetivos a alcançar, terminando com a apresentação da organização geral e convenções adotadas.

Apresenta-se no capítulo 2 uma síntese do estado da arte relativa às principais redes de comunicação digital série utilizadas nos veículos automóveis mostrando um pouco da história e evolução até à atualidade. Apresenta-se também uma síntese sobre sistemas de iluminação nos automóveis evidenciando-se as tendências atuais.

No capítulo 3 apresenta-se a solução desenvolvida para iluminação automóvel baseada na rede CAN. A primeira parte deste capítulo é dedicada a descrição geral da solução, seguindo-se uma síntese da escolha do material e a sua implementação. Na segunda parte do capítulo apresenta-se a explicação da programação dos dispositivos utilizados na solução desenvolvida.

Apresenta-se no capítulo 4 a análise de alguns resultados experimentais, por exemplo uma mensagem transmitida na rede CAN, obtidos com recurso a um osciloscópio digital.

O capítulo 5 apresenta conclusões de natureza geral e particular, incluindo os aspetos da presente dissertação que são considerados sucessos e outros que poderiam ser melhorados, e ainda as perspetivas de trabalho futuro.

Com o objetivo de facilitar a leitura e compreensão deste documento foram utilizadas convenções que se descrevem de seguida:

- Sempre que exista correspondência entre as grandezas presentes nas equações utilizam-se unidades do Sistema Internacional (S.I.) de unidades de medida. Nos múltiplos e submúltiplos dessas unidades as respetivas abreviaturas;
- Recorreu-se, sempre que possível, a conceitos presentes na Língua Portuguesa. Em determinados casos os conceitos surgem acompanhados pela designação na Língua Inglesa, colocada entre parêntesis e em itálico, para que a compreensão do seu significado seja mais fácil e menos ambígua.

2. Estado da Arte

2.1 Introdução

Este capítulo apresenta uma síntese do estado da arte relativa às principais redes de comunicação digital série utilizadas nos veículos automóveis mostrando um pouco da história e evolução até à atualidade. Apresenta-se também uma síntese sobre sistemas de iluminação nos automóveis evidenciando-se por fim as tendências atuais.

Atualmente a forte dependência dos sistemas eletrônicos e do *software* nos veículos automóveis é bastante surpreendente. O surgimento das primeiras unidades de controlo eletrónicas (ECU) para o sector automóvel, com a única finalidade de efetuar o controlo do ponto de ignição, surgiram em 1977 pela *General Motors (GM)*, no modelo *Oldsmobile Toronado*. Em 1978, a *GM* ofereceu como opção para os clientes que adquirissem o modelo *Cadillac Seville*, o primeiro computador de bordo com recurso a um microprocessador da Motorola, Modelo 6802, mostrando a velocidade, a quantidade de combustível, o tempo de viagem e algumas informações sobre o motor. Este microprocessador acumulava também as funções de controlo da injeção eletrónica de combustível e velocidade de cruzeiro (*cruise control*). Em 1981, a *GM* já usava no total cerca de 50 mil linhas de código em todas as unidades eletrónicas programáveis na sua gama de veículos ligeiros de passageiros. Rapidamente outras marcas seguiram pelo mesmo caminho [1].

Durante os primeiros anos, as unidades de controlo eletrónicas ainda realizam as suas tarefas de forma completamente autónoma e isolada. No entanto, muito cedo se percebeu que através da coordenação de aplicações colocadas em diferentes unidades seria possível aumentar consideravelmente as funcionalidades dos veículos. Esta foi a motivação para a integração de sistemas de comunicação digital série no automóvel. É necessário não esquecer que esta integração só foi possível como consequência dos avanços registados na indústria onde as redes de comunicação digital série se começaram a desenvolver sensivelmente a partir de 1975. O primeiro caso relevante destas redes de comunicação série terá sido lançado pela firma americana Cutler-Hammer com o nome Directrol. As redes Genius I/O da General Electric, Interbus-S da Phoenix e Sensoplex da Turck surgiram entre 1985 e 1987 e ainda hoje são usadas, o mesmo acontecendo com a rede Arcnet que viria a migrar das aplicações de escritório para o ambiente industrial.

Desde a introdução de sistemas de comunicação série, os complexos e muitas vezes divergentes sistemas cablados no automóvel tornaram-se uma coisa do passado. Os sistemas de comunicação vieram não só simplificar o projeto e instalação das cablagens mas também reduzir o peso e espaço necessário para essas mesmas cablagens. No entanto, a integração de cada vez mais funções de segurança e sistemas de conforto a bordo, ao longo dos anos, tem exigido um crescente aumento de volume na troca de dados. De modo a gerir este crescimento de forma sustentada novas soluções têm surgido a vários níveis para lá da rede CAN, tais como as redes de comunicação LIN, FlexRay e MOST. Na tabela 1 apresenta-se um quadro com um resumo das características das redes mais utilizadas nesta área.

Tabela 1 – Comparação das características das redes.

	LIN	CAN	FlexRay	MOST
Categoria	Classe A	Classe B ou classe C	Classe C ou classe D	Classe D
Topologia	Barramento, estrela	Barramento	Barramento, estrela	Ponto a ponto
Transmissão	Um condutor simples	Par torcido	Par torcido ou fibra ótica	Fibra ótica
Codificação	NRZ	NRZ-5	NRZ	BiPhase
Método de acesso	<i>Master/ slave</i>	CSMA/ BA	TDMA	<i>Master/ slave</i>
Deteção de erros	BCC	CRC	CRC	CRC
Comprimento do cabeçalho	2 bits/byte	11 ou 29 bits	40 bits	Não especificado
Comprimentos dos dados	8 bytes	0 a 8 bytes	0 a 246 bytes	Não especificado
Ritmo de transmissão	20 kbit/s	10 kbit/s a 1 Mbit/s	10 Mbit/s	25 Mbit/s
Custo	Baixo	Médio	Médio	Alto

Outras redes também utilizadas nos veículos automóveis, como a VAN da Renault, Peugeot e Citroen, a MICS da Mitsubishi e a Palnet da Mazda não foram abordadas nesta dissertação uma vez que são redes com proprietários específicos, não existindo por isso tanta informação.

Aguarda-se que parcerias como a AUTOSAR (*Automotive Open System Architecture*) fundada em 2003 (ou outras do género) venham ajudar a criar e estabelecer arquiteturas abertas e normalizadas de *hardware* e *software* para unidades de controlo eletrónicas no mundo automóvel.

2.2 Conceitos gerais

Nos automóveis a estrutura de ligação física dos vários elementos da rede utiliza geralmente a topologia em barramento (*bus*), que é muito fácil de projetar e implementar devido à interface de barramento passivo. Para além desta, existem as topologias em anel e em estrela, podendo também existir combinações de várias topologias como veremos mais adiante.

Um dos requisitos para a troca de dados na comunicação série é a necessidade de atribuição exclusiva dos dados a serem enviados aos nós de barramento. As mensagens, a enviar e a receber, têm no início da mensagem um endereço único no barramento que pode identificar o dispositivo a que se destina a mensagem ou pode identificar a mensagem que está a ser enviada. Isto significa que, em princípio, todos os dados estão disponíveis para qualquer nó receber, mas cada nó do barramento tem a tarefa de filtrar os dados que são relevantes para ele. Isto é realizado com a ajuda dos endereços, referidos aqui como identificadores. A fim de que o recetor adquira o identificador e os dados como uma unidade, o remetente envia ambos em conjunto como uma trama. Uma trama típica abrange o endereço e os dados com um reconhecimento de início e de fim, que são usados principalmente para sincronizar os emissores e recetores. Uma trama é também referida como uma "mensagem".

Dois dos aspetos fundamentais de qualquer rede de comunicação são a velocidade de transmissão (fundamental para a troca de dados em tempo real) e integridade dos dados. Um sistema distribuído em rede só pode atingir o seu objetivo se todos os dados chegarem ao nó de destino no tempo esperado e sem erros. O desempenho de uma rede de comunicação série depende, substancialmente, da capacidade de envio dos seus dados em tempo útil e da capacidade com o qual ele pode detetar e corrigir erros que nela surjam. Quantitativamente a integridade dos dados pode ser descrita como a probabilidade residual de erro. Probabilidade residual de erro é entendida como o produto da probabilidade A, que os dados transmitidos são corrompidos, pela probabilidade B que os dados corrompidos

podem permanecer sem ser detetados [2]. A integridade de dados de um sistema de barramento série depende então por um lado das medidas adotadas para evitar a corrupção das mensagens de dados, e por outro das medidas adotadas para detetar (e eventualmente corrigir) os dados corrompidos.

No caso dos veículos automóveis são diversas as potenciais fontes emissoras de ruído (por condução ou por radiação), tal como os transitórios de alta frequência dos contactos dos relés, os motores dos ventiladores, conversores comutados a alta frequência, incluindo alguns circuitos osciladores usados em aplicações digitais. Quanto maior for o sucesso a eliminar estas causas, maior será a imunidade ao ruído e a probabilidade de transmissão de dados. Algumas medidas como o afastamento de circuitos de potência e circuitos de sinal (o que nem sempre é fácil dadas as dimensões do automóvel), a utilização de cabos com pares torçados e blindagens nos circuitos e cabos podem contribuir para mitigar este efeito. Além disso, outros aspetos como a velocidade de transmissão, a codificação de sinais (denominada de código de linha) e a impedância característica do meio físico de transmissão são também importantes para este fim.

Entres os métodos de deteção de erros mais comuns estão a técnica da soma de verificação designada pela sigla BCC (*Block Checksum*) e a técnica do teste de redundância cíclica conhecida pela sigla CRC (*Cyclic Redundancy Check*) em que o remetente efetua uma soma de verificação a partir da trama de dados com base nos bits de paridade ou através de um polinómio gerador de 8, 16 ou 32 bits. Em seguida, envia esta soma de verificação no fim da trama de dados. Usando este procedimento o recetor é capaz de verificar se a trama de dados recebida está correta. A eficiência de um método de deteção de erros depende essencialmente do ganho que se obtém em termos de probabilidade de erro antes e depois da aplicação desse mesmo método. Um método que garanta uma menor probabilidade de erro com um algoritmo mais reduzido em termos de dimensão e de necessidade de processamento é normalmente o preferido. Como regra geral, depois de detetar um erro de comunicação é necessário efetuar alguma forma de correção ou simplesmente ignorar a mensagem e desencadear um procedimento para reenvio da mensagem com a sigla genérica ARQ (*Automatic Repeat Request*). Alguns métodos de correção de erros mais conhecidos são os códigos de *Hamming*, códigos multidimensionais de verificação de paridade ou códigos *Reed-Solomon*. Por razões de eficiência os métodos de correção de erros não são implementadas em redes de comunicação dos automóveis. [3]

Em termos de acesso ao barramento de comunicação série por parte de cada nó, existem essencialmente quatro métodos de acesso: *master-slave*, espontâneo, acesso múltiplo por divisão de tempo, e por passagem de testemunho; em certos casos são usados métodos híbridos com um árbitro de rede. No caso das redes de comunicação série conhecidas nos automóveis só os três primeiros métodos de acesso são utilizados.

Em sistemas de barramento de comunicação série com acesso ao barramento controlado, como nos métodos de acesso *master-slave* e por passagem de testemunho, os direitos de acesso ao barramento já estão claramente definidos. Tais sistemas oferecem o tráfego de mensagens determinista como um importante pré-requisito para alcançar sistemas de barramento de série em tempo real. No entanto, uma vez que a sequência de comunicação inteira é executada de acordo com um cronograma e não pode ser influenciada, os sistemas de barramento de série, com acesso ao barramento controlado, são caracterizados por um comportamento pouco dinâmico.

Isto não se aplica a sistemas de barramento com acesso ao barramento espontâneo. Neste método cada nó do barramento tem o direito de aceder ao barramento de forma espontânea, por exemplo, em resposta a um evento que ocorreu e dado que o barramento não está disponível numa janela temporal limitada, o acesso torna-se bastante mais rápido. No entanto, com este método aumenta o risco de colisão entre os dados que são enviados para a rede, dependendo da densidade de eventos, do tamanho das mensagens e do ritmo de transmissão disponível.

A monitorização do barramento pelos nós que desejam aceder à rede reduz significativamente o risco de colisão, esta classe de métodos de acesso tem uma designação genérica com a sigla CSMA (*Carrier Sense Multiple Access*). O risco de colisão pode ser prevenida totalmente por introdução de prioridades de mensagem. No entanto, estes métodos de acesso ao barramento com base na monitorização do barramento e prioridades de mensagens não podem garantir pontualidade. É possível que as mensagens de baixa prioridade sejam adiadas por um tempo excessivamente longo.

Com base no modelo de referência para a comunicação de dados especificado pela ISO (*International Standardization Organization*), a interface de comunicação série de um nó de rede no automóvel é normalmente subdividida em duas camadas: uma camada inferior ou camada física e uma camada superior ou camada de ligação de dados [4].

Algumas das tarefas tratadas pela camada de ligação de dados são o endereçamento, a integração da trama, o método de acesso ao barramento, a sincronização e a detecção e correção de erros. Estas funções são definidas por um protocolo de comunicação. A especificação da camada física, por outro lado, abrange todos os aspetos a partir da interface do barramento físico à transmissão de sinal através de determinado suporte físico.

Geralmente a interface do barramento físico é implementada com a ajuda de um transceptor (*transceiver*). Um controlador de comunicação abrange a camada de ligação de dados. Se todos os nós do barramento dentro do sistema seguem o mesmo protocolo de comunicação e a mesma especificação de camada física, estão satisfeitas as condições prévias fundamentais para a troca de dados sem problemas entre os nós do barramento. Em comunicação série, a aplicação do remetente passa os dados a serem enviados para o controlador de comunicação, sendo este o responsável por adicionar o endereço e informações de controlo e de sincronização à mensagem de dados, criando assim uma trama. O transceptor agora transmite a trama através do barramento. No lado do recetor, o transceptor aceita a trama e passa para o controlador de comunicação, que avalia a informação que lhe é transmitida e no caso da receção de dados corretos encaminha a trama de dados para a aplicação do recetor. Isto resulta num fluxo de comunicação hierárquico e portanto transparente. Isso é garantido pela conclusão das tarefas de comunicação atribuídas às camadas, pelo protocolo de comunicação e definição da camada física.

Para algumas tarefas como a gestão do barramento ou diagnóstico e configuração dos nós do barramento, a funcionalidade de comunicação proporcionada pela camada de ligação de dados é insuficiente. Definindo camadas superiores e respetivamente protocolos de comunicação superiores, as funcionalidades de comunicação podem ser expandidas. Sendo que estão definidas classes para as redes dos automóveis como veremos a seguir.

2.3 Classes das redes dos automóveis

A SAE (*Society of Automotive Engineers*) categorizou as redes dos automóveis nas classes A, B e C com a ordem crescente de criticidade em tempo real e restrições de confiabilidade. Existe uma relação hierárquica entre as classes das redes. Por definição, a classe A está contida na classe B, bem como a classe B está contida na classe C. A tabela 2 resume e compara características das três classes das redes.

Tabela 2 – Características das classes SAE.

	Classe A	Classe B	Classe C
Propósito	Controlo de sensores/atuadores	Partilha de informação	Controlo em tempo real
Informação	Ocasional	Frequente	Tempo real
Velocidade	Menor de 10 kbit/s	Entre 10 kbit/s e 125 kbit/s	Maior de 125 kbit/s
Tempo de resposta	Largo (<150ms)	Variado	Estreito
Prioridade	Baixa	Média	Alta
Informação perdida ou corrupta	Transtorna	Transtorna ou falha	Falha
Custo por nó	<1\$	<2\$	<4\$

A classe A é a primeira categoria SAE e mantém o menor ritmo de transmissão, um ritmo que atinge o valor máximo de 10 kbit/s e deve suportar a transmissão de mensagens orientada por eventos. É usada na baixa gama de comunicação de propósito geral. A implementação da classe A tem reduzido significativamente o volume de cablagens nos automóveis. O custo é geralmente de cerca de "x" por nó. De acordo com os dados de 2010, uma estimativa muito grosseira de 0,50\$ a 1\$ pode ser usada para o valor de "x" [5]. Este custo inclui os circuitos embebidos (microcontrolador, transceptor, etc.), o *software*, os conectores, o serviço, etc.

Dispositivos da classe A suportam as operações de sistemas de conforto como sensores e atuadores. As aplicações típicas envolvendo essas redes incluem controlo de luzes, de limpa pára-brisas, de portas (por exemplo, fechaduras, abertura / fecho de janelas) ou de assentos (por exemplo, motores de posição do assento, controlo de ocupação). O requisito de latência para aplicações de classe A é largo, indo de 50ms a 150ms [6].

A classe B admite ritmos de transmissão entre 10 kbit/s e 125 kbit/s e normalmente suporta a grande maioria das comunicações não críticas. A utilização da classe B pode eliminar sensores redundantes e outros elementos do sistema, proporcionando um meio para transferir dados entre os nós e deve suportar a transmissão orientada por eventos e algumas mensagens periódicas, deve também permitir o modo de suspensão. O custo é de cerca de 2x por nó.

A informação partilhada numa rede de classe B não é crítica para o funcionamento de todos os sistemas a que se encontra ligada. O atraso de um pedaço específico de informação não irá causar uma falha crítica em qualquer um dos sistemas. Portanto o tempo de resposta na rede de classe B não é tão estreito como na classe C. De facto, o tempo de resposta pode ser variável, dependendo da aplicação. Outra característica da rede de classe B é a interligação de sistemas diferentes.

A classe C suporta ritmos de transmissão entre 125 kbit/s e 1 Mbit/s. Devido a este nível de desempenho, a classe C facilita o controlo distribuído através de sinais com altos ritmos de transmissão tipicamente associados a sistemas de controlo em tempo real e crítico, com tempo de resposta de poucos milissegundos (por exemplo, o controlo do motor e da suspensão). O par trançado é o meio físico de transmissão escolhido em vez de par trançado blindado ou fibra ótica. O custo é de cerca de 3x a 4x por nó. No entanto, a gama alta de utilizações da classe C baseia-se em meios físicos de transmissão caros, como fibras óticas, que podem aumentar o custo por nó para um número muito superior ao previsto.

Categorias mais elevadas para comunicação com desempenho entre 1 Mbit/s e 10 Mbit/s são esperadas no futuro. Categorias dedicadas a dados multimédia podem ser esperadas para impulsionar as necessidades de largura de banda e desempenho, sendo já chamada classe D.

2.4 Rede CAN



A rede CAN (*Controller Area Network*) foi desenvolvida pela empresa alemã Robert Bosch para automação de veículos automóveis do grupo Daimler-Benz em substituição das cablagens convencionais. Foi desenvolvida de forma a fornecer uma solução de comunicação digital série fiável e bastante robusta. Os fabricantes de microeletrónica licenciados pela Bosch vendem anualmente dezenas de milhões de chips de interface CAN, em todo o mundo, para o sector automóvel.

A rede CAN tem conformidade com as normas ISO 11898 desde 1993 e está organizada em várias partes. A entidade que suporta a rede CAN como sistema aberto tem o nome CAN in Automation. A primeira parte contém o protocolo CAN e abrange toda a camada de ligação de dados e uma parte da camada física do modelo normalizado de referência OSI para a comunicação de dados da ISO 7498. Note-se que não há uma correspondência direta entre estas normas. Todas as regras estão implementadas na eletrónica de cada interface, baseada em circuitos integrados de vários fabricantes. Por exemplo citam-se o chip PCA82C200 da Philips, 82527 da Intel e MCP2562 da Microchip.

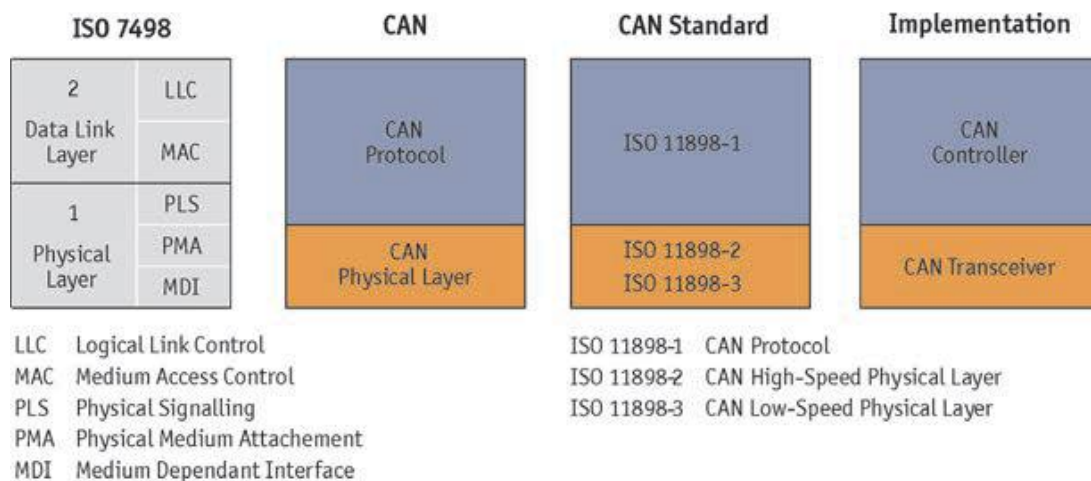


Figura 2.1 – Correspondência entre o modelo de referência normalizado e a rede CAN de acordo com ISO 11898.

A segunda parte (ISO 11898-2) descreve a camada física da rede CAN de alta velocidade, e a terceira parte (ISO 11898-3) a camada física de baixa velocidade. Estas duas partes cobrem a camada física da ISO 7498 incluindo ainda a interface de acesso ao meio físico, os ritmos de transmissão e os níveis de tensão. No caso dos automóveis, a

camada física da CAN de alta velocidade é usada principalmente em aplicações de chassis e de transmissão. É implementada pelo transceptor CAN de alta velocidade, que suporta um ritmo de transmissão máximo de 1 Mbit/s. Na variante de baixa velocidade o transceptor CAN suporta um ritmo de transmissão máximo de 125 kbit/s e é principalmente utilizado nas áreas de utilidade geral e de sistemas de conforto [7].

Cada interface de rede CAN é composta por um controlador e um transceptor. Enquanto o controlador CAN lida com o protocolo da rede, o transceptor CAN assume a tarefa de interligar fisicamente o controlador ao barramento da rede operando com sinais de tensão em modo diferencial, prevendo-se uma alimentação entre 1,8V a 5,5V. A transmissão dos dados com sinais de tensão em modo diferencial é caracterizada por aumentar significativamente a imunidade ao ruído quando comparada com outras soluções. Para isso são necessárias duas linhas de comunicação denominadas de *CAN High* e *CAN Low*, que são terminadas em ambas as extremidades com uma impedância característica muito semelhante à das linhas de modo a atenuar as reflexões na transmissão na variante de alta velocidade.

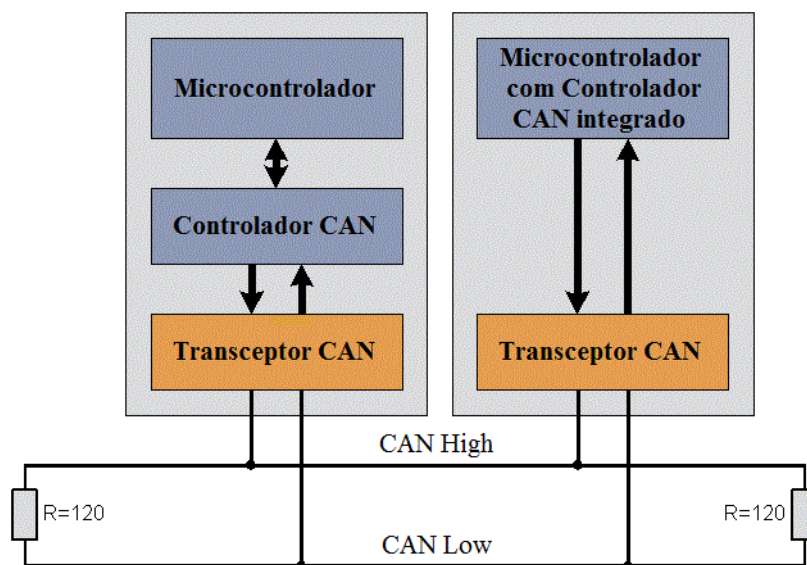


Figura 2.2 – Rede CAN e a sua interface.

A rede CAN utiliza mecanismos de endereçamento e filtragem para estruturar as mensagens a enviar e a receber dentro de cada nó da rede. Os endereços das mensagens, normalmente referidos como identificadores (IDs), não identificam os nós CAN de destino, em vez disso eles identificam as próprias mensagens de modo que, em princípio, todas as mensagens estão disponíveis para serem recebidas por todos os nós CAN. Os filtros na rede

CAN são utilizados para que cada nó filtre apenas a informação relevante que pretenda obter da rede. Os 11 bits de largura do identificador permitem a especificação de até 2048 mensagens numa rede.

A difusão de mensagens oferece as seguintes vantagens: economia de custos mediante a utilização partilhada de sensores, fácil implementação e sincronização dos processos distribuídos e acima de tudo uma elevada flexibilidade no que diz respeito à configuração. Omitindo os endereços dos nós faz com que seja possível integrar outros nós no barramento sem ter de modificar o *hardware* ou o *software* de nós do barramento existentes. No entanto, isto só é verdade se o nó de barramento acrescentado for exclusivamente um recetor.

As mensagens que são transmitidas numa rede CAN e a sua sequência não dependem de uma progressão temporal, em vez disso, dependem da ocorrência de eventos especiais. Cada nó CAN é, em princípio, autorizado a aceder ao barramento CAN imediatamente após ocorrer um evento. Dado o curto comprimento de mensagem, de no máximo 130 bits no formato padrão, e o seu alto ritmo de transmissão, de dados de até 1 Mbit/s, este método permite reações rápidas a eventos assíncronos. Esta é uma condição importante para a transmissão de dados em tempo real, na faixa de milissegundos, que é principalmente uma exigência das aplicações de chassi e de transmissão.

Dado que a comunicação CAN não se baseia em qualquer cronograma, o tráfego de mensagens não é determinado e isso implica o risco inerente de colisões. Este risco aumenta com o aumento da carga do barramento e interfere na capacidade de resposta em tempo real do sistema. É utilizado o método de acesso ao barramento com monitorização chamado CSMA/ BA (*Carrier Sense Multiple Access with Bitwise Arbitration*) de modo a assegurar a transmissão de dados em tempo real.

O nó CAN que pretende enviar mensagens começa por escutar o barramento. Se este está disponível o nó CAN pode começar a transmitir a sua mensagem imediatamente. Por outro lado, se ele deteta atividade no barramento deve adiar a sua solicitação de envio até que o barramento CAN esteja disponível e a transmissão de mensagem atualmente em execução tenha sido concluída. Aguardando para além disso um intervalo de três tempos de bit. Assim a transmissão da mensagem em curso não é interrompida e o acesso ao barramento é não destrutivo.

surgir uma situação em que as mensagens CAN chegam demasiado tarde aos recetores ou são suprimidas por completo. Por outro lado, este método de acesso ao barramento produz uma relação recíproca entre a extensão da rede e o ritmo de transmissão máxima. Durante a arbitragem bit a bit um nó CAN enviando um nível recessivo deve ser capaz de detetar com segurança um nível dominante, sendo que o nível recessivo corresponde ao nível lógico "1" e o nível dominante ao nível lógico "0". O intervalo de tempo de bit deve ser dimensionado de modo que os tempos de propagação do sinal no barramento CAN sejam totalmente compensados, o que por sua vez define um ritmo de transmissão máximo utilizável.

São principalmente as tramas de dados que são responsáveis pela transmissão de dados nos automóveis. Na verdade também existem tramas remotas para solicitar dados mas estas quase nunca são usadas uma vez que a transmissão de dados no automóvel tipicamente não é solicitada, pelo contrário, é fornecida principalmente por iniciativa do gerador de informações. Os dois tipos de tramas têm esquemas idênticos, a única diferença é que o campo de dados é omitido na trama remota.

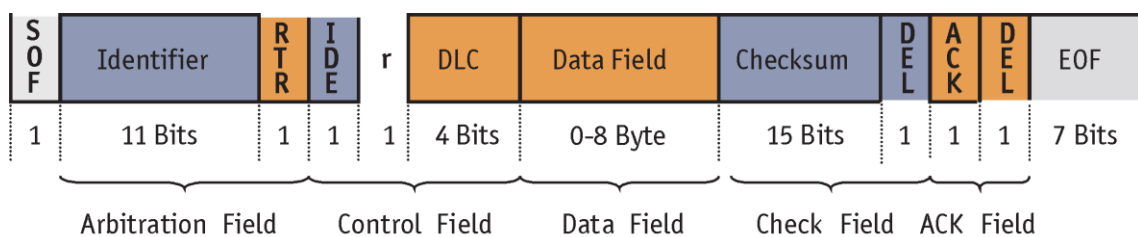


Figura 2.4 – Estrutura da trama de dados.

Um pré-requisito básico para a transmissão de tramas de dados e de tramas remotas é o sincronismo entre o emissor e o recetor. Dado que uma linha de relógio foi omitida por razões de custo e de esforço, o sincronismo é conseguido pelos flancos do sinal e um mecanismo de ressincronização bem definido. Cada transmissão de mensagem começa com a transmissão do bit dominante de sincronização do início da trama (SOF) e isso gera o primeiro flanco do sinal. O recetor assegura a sincronização ao longo de toda a transmissão através da avaliação de cada flanco do sinal que chega e adaptando o seu próprio tempo de bit se for necessário. A técnica de encrustamento de bits assegura que um bit complementar aparece o mais tardar depois de cinco bits homogêneos, proporcionando assim um flanco de sinal.

A seguir ao SOF, vai o identificador, que pode ser tanto o identificador padrão de 11 bits como o identificador estendido de 29 bits de comprimento. O formato padrão é mais usual na área automóvel. O formato expandido normalmente é utilizado em conjunto com protocolos de nível mais elevado, tais como SAE J1939. O formato do ID que está a ser usado é indicado pelo bit identificador de extensão (IDE). O bit de pedido de transmissão remota (RTR) indica se é uma trama de dados ou uma trama remota.

O campo de dados com largura de 64 bits está disponível para a transmissão de informação útil, em que o número exato de bytes úteis é indicado por um código de comprimento de dados (DLC). Seguindo o campo de dados está a chamada verificação de redundância cíclica (CRC). O remetente gera a sequência de CRC com base em todos os bits a serem transmitidos, um gerador polinomial e um algoritmo bem definido. Independente da filtragem de mensagens, o mesmo processo ocorre na extremidade de receção com os bits que chegam. As duas sequências são comparadas, e a confirmação é feita no campo de reconhecimento (ACK) após o bit recessivo delimitador de CRC. No fim de uma trama de dados, após o bit recessivo delimitador do ACK, vem o fecho da trama (EOF) composta por sete bits recessivos.

A probabilidade de que as mensagens corrompidas possam permanecer sem ser detetadas é extraordinariamente baixa. Estima-se a ser $4,7 \times 10^{-11}$ [8]. Responsáveis por isso são os mecanismos de deteção de erro definidos no protocolo CAN. No lado do recetor, além do CRC, independente da filtragem de mensagens, que é capaz de detetar até cinco erros dentro de uma mensagem CAN, também são feitas verificações do formato e da técnica de enrustamento de bits. O remetente realiza o monitoramento dos bits enviados e avalia o campo ACK.

Se assumirmos uma taxa de erro de 10^{-3} numa rede CAN, então dado um tempo de funcionamento anual de 1000 horas, um ritmo de transmissão de 500 kbit/s, uma carga do barramento média de 25% e um comprimento médio de mensagem de 80 bits, estatisticamente falando uma mensagem CAN corrompida vai permanecer sem ser detetada pelo protocolo CAN apenas uma vez a cada 4000 anos [9]. O que se entende como sendo a taxa de erro é a relação de mensagens CAN corrompidas para o número total de mensagens CAN transmitidas.

Assim que um mecanismo de deteção de erro sinaliza um erro de transmissão, o nó CAN que detetar o erro termina a transmissão da mensagem, colocando uma sinalização de erro composta por seis bits dominantes no barramento. A sinalização de erro viola intencionalmente a técnica de encrustamento de bits para que, em toda a rede, cada nó CAN perceba o que até então era um erro local e responda terminando a transmissão da mensagem, ou seja, anexando uma sinalização de erro. Este método assegura a consistência dos dados em toda a rede que é tão importante em aplicações distribuídas.

A correção de erros consiste na repetição da mensagem CAN abortada pelo mesmo remetente, assim que o barramento CAN está livre de novo, após o delimitador de erro e os 3 bits de separação de mensagens. No projeto do sistema deve-se considerar que o método de acesso ao barramento CSMA/ BA não garante a repetição imediata. O tempo de recuperação do erro depende da prioridade da mensagem e da carga do barramento.

Sinalização de erro dá a cada nó CAN a capacidade de terminar as transmissões de mensagens em curso. Uma vez que isto também se aplica a nós CAN com defeito, tais nós são capazes de trazer toda a comunicação CAN a uma paralisação. Para evitar isso, cada nó CAN possui monitoramento da rede, um nó encontrado defeituoso pode desconectar-se com base em contadores de erros e regras para controlar os contadores de erros.

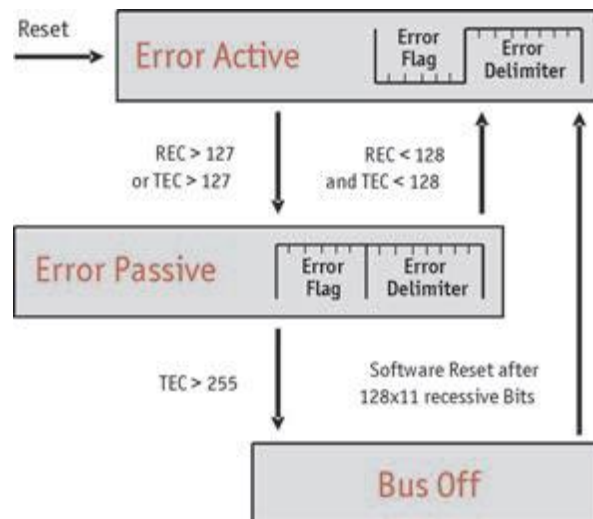


Figura 2.5 – Diagrama das regras dos contadores de erros.

Numa rede CAN cada transmissão de mensagem é reconhecida simultaneamente por todos os recetores no bit ACK, independentemente da filtragem de mensagens. Um nível dominante significa um reconhecimento positivo e um nível recessivo significa um

reconhecimento negativo. O remetente coloca um nível recessivo no bit ACK, apenas um reconhecimento positivo é suficiente para confirmar uma transmissão de mensagem correta. Devido a este reconhecimento positivo, os nós CAN reconhecendo negativamente são substituídos e permanecem desconhecidos. Por isso eles enviam uma sinalização de erro após o delimitador ACK.

Até há poucos anos atrás a CAN foi a tecnologia de barramento série mais procurada na indústria automóvel. A forte aposta nos sistemas eletrónicos nos veículos levou a rede CAN a encontrar os seus limites. Os fabricantes de veículos questionam a adequação do barramento CAN especialmente para aplicações em tempo real, com uso intensivo da largura de banda e de segurança altamente crítica para veículos motorizados, tais como o sistema de assistência ao condutor para manter a faixa de rodagem, mas também em aplicações de sistemas de conforto sensíveis ao custo.

Posto isto, a Robert Bosch introduziu o novo protocolo CAN FD, similar ao protocolo CAN mas com o ritmo de transmissão flexível, em março de 2012. Os novos recursos principais deste protocolo são que ele estende o comprimento dos dados úteis de oito para 64 bytes e oferece ritmos de transmissão de dados significativamente mais elevados. Com base nos seus dados de desempenho, a CAN FD está posicionada entre a CAN de alta velocidade (1 Mbit/s) e a FlexRay (10 Mbit/s). É ideal para preencher a lacuna de desempenho entre estes sistemas de barramento de baixo custo, ver figura 2.6 [10].

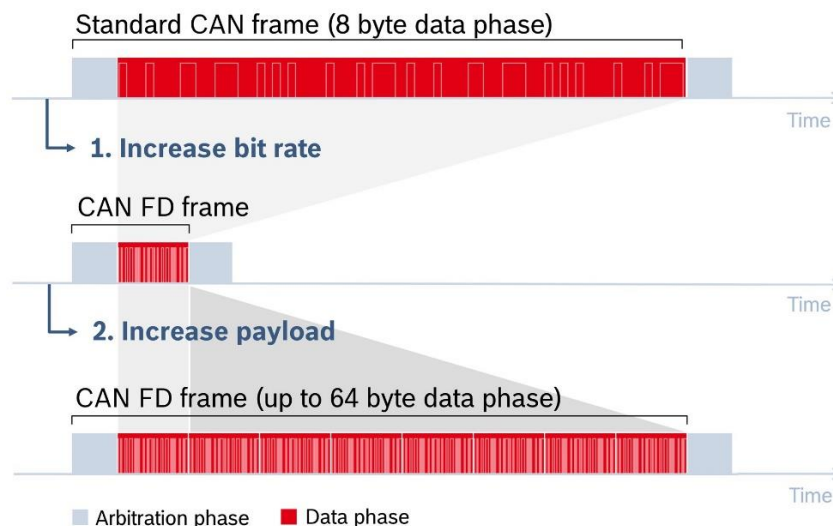


Figura 2.6 – Diferenças entre os protocolos CAN e CAN FD.

2.5 Rede LIN



A rede LIN (*Local Interconnect Network*) estabeleceu-se rapidamente como a tecnologia de escolha para a troca de dados simples e de baixo custo nos automóveis. Hoje em dia, muitos fabricantes contam com a rede LIN para transmitir sinais pouco críticos na área de chassi e de sistemas de conforto.

A procura crescente de sistemas de conforto de condução pelos utilizadores de carros levaram a uma forte aposta nos sistemas eletrónicos nesta área da tecnologia dos veículos. Isto reflete-se na migração dos vários componentes eletrónicos para a área de sistemas de conforto.

Foi prática habitual interligar o continuamente crescente número de sensores, atuadores, motores de passo e motores DC diretamente a um módulo de controlo central. Esta tendência reuniu-se gradualmente com o criticismo. Isso levou a um rápido aumento nos custos de cablagem, juntamente com maior requisito de espaço, aumentando o peso e significativamente maior suscetibilidade a falhas. Além disso, a crescente individualização requeria muitas variantes diferentes dos tipos de condutores e conectores, que por sua vez tornaram a produção, a instalação e a manutenção consideravelmente mais difíceis.

Os fabricantes reconheceram rapidamente que uma rede de componentes ao longo de um sistema de barramento seria uma solução ideal neste domínio de aplicação automóvel. No entanto, uma vez que o barramento CAN não era um candidato para uso na área de sensores/atuadores, muitos fabricantes e fornecedores começaram a desenvolver os seus próprios sistemas de barramento de sensores/atuadores em meados dos anos 1990 por forma a diminuir os custos.

Isto resultou gradualmente na criação de inúmeros barramentos rentáveis e simples para sensores/atuadores. No ano de 2000, a LIN chegou ao mercado das redes como um outro sistema de barramento série para a área de atuador/sensor. Esta tecnologia tem prevalecido numa frente ampla, e a LIN pode agora ser encontrado em quase todos os veículos, normalmente em aplicações de sistemas de conforto, tais como o controlo de temperatura, o ajuste do banco, os controlos de portas e os ajustes de espelhos.

Uma razão importante para a rápida integração da LIN foi a fundação do consórcio LIN, a que fabricantes e fornecedores de automóveis, bem como fabricantes de semicondutores e de ferramentas se tinham juntado. O seu objetivo era criar um padrão de comunicação comum entre fabricantes para a área dos atuadores/ sensores. Com a definição de uma camada física simples e de baixo custo com base na norma ISO 9141, bem como um protocolo de comunicação simples e fácil. O consórcio LIN lançou as bases para o sucesso, preparando o terreno para a implementação de nós de barramento simples e bastante rentáveis.

O consórcio LIN não se focou apenas na comunicação LIN em si mas também forneceu uma metodologia de desenvolvimento e essa promoveu a aceitação do sistema de barramento LIN na indústria automóvel. O fluxo de trabalho LIN faz com que seja possível automatizar o desenvolvimento de uma rede LIN, com a economia de tempo e de custos. Os pilares da metodologia de desenvolvimento são dois formatos de troca de dados usados para descrever toda a rede LIN e os nós LIN individuais. A rede LIN segue um cronograma temporal bem definido para centralizar o estado da rede.

Servindo para descrever uma rede LIN inteira estão a sintaxe uniforme da linguagem de configuração LIN e o normalizado *LIN Descrição File* (LDF). Definidas no LDF estão todas as propriedades de uma rede LIN, em particular as relações de comunicação. Ferramentas geradoras utilizam o LDF para gerar os componentes de *software* necessários para a comunicação LIN. Além disso, o LDF fornece as informações necessárias para análise e ferramentas de medição e de ensaio.

Da mesma forma, a descrição de nós LIN individuais é dada pela sintaxe uniforme da linguagem de capacidade de nó e pelo normalizado *Node Capability File* (NCF). O NCF descreve as características de desempenho de um nó escravo LIN, incluindo as definições de trama e de sinal, o ritmo de transmissão e o diagnóstico, e no âmbito da conceção do sistema ele representa a base para a geração automática do LDF.

Os dois formatos de troca de dados e o processo de configuração definido na especificação LIN permitem que os usuários implementem um tipo de escravo LIN, por exemplo um motor de passo, várias vezes numa rede LIN ou reutilizem um tipo de escravo LIN em diferentes redes LIN.

A documentação detalhada da especificação é um contributo para o sucesso da LIN. A especificação LIN 2.2A, em vigor desde Dezembro de 2010, define a camada física, o protocolo de comunicação, o fluxo de trabalho LIN, bem como diagnósticos e configuração dos nós LIN [11].

O objetivo de criar um protocolo de comunicação de baixo custo para a troca de dados em série na área de sensores/atuadores não críticos de segurança afeta principalmente o projeto da camada física. A transmissão do sinal físico numa rede LIN não envolve o uso de uma transmissão de sinal diferencial familiarizado pela rede CAN, em vez disso é utilizada uma linha de um único fio convencional. Para garantir a imunidade ao ruído, a tensão de alimentação e a massa da eletrónica da ECU são usadas como tensões de referência para o nível do barramento. Um transceptor LIN serve como interface do barramento físico. Um nível de pelo menos 40% abaixo da tensão de alimentação é interpretada pelos recetores como um lógico "0". Os recetores interpretam um nível de pelo menos 60% acima da tensão de alimentação como um lógico "1".

O ritmo de transmissão máximo é limitado a 20 kbit/s para manter as emissões de ruído dentro dos limites. Para comprimentos de linha até 40 metros, o número máximo recomendado de nós é 16. Isso tem em conta as capacitâncias do nó e da linha, bem como a constante de tempo máximo permitida na rede LIN fixada na especificação LIN.

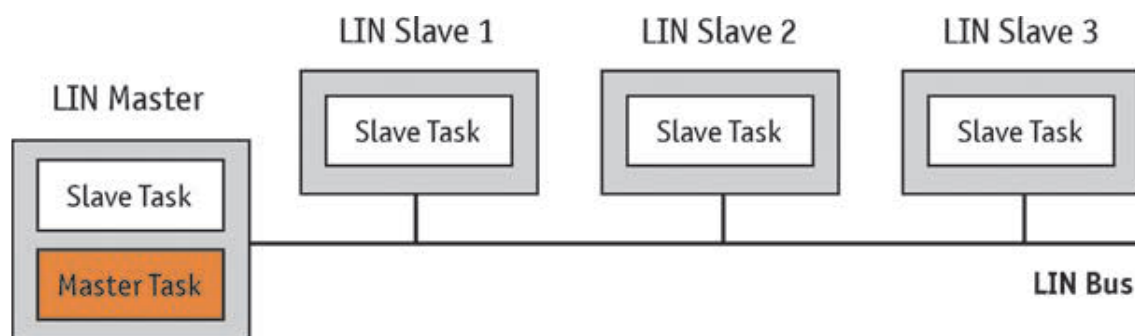


Figura 2.7 – Arquitetura mestre-escravo da rede LIN.

Em termos de tecnologia de circuito, uma rede LIN é equivalente a um circuito de coletor aberto. Uma resistência de *pull-up* assegura que o nível do barramento permanece quase ao nível da tensão de alimentação enquanto os transístores Tx de todos os nós LIN estão inibidos. O nível do barramento é puxado quase ao nível da massa, tão rapidamente

como um dos transístores Tx é ativado. Por conseguinte, o estado lógico "0" é referido como o nível dominante e o estado lógico "1" o nível recessivo.

A comunicação numa rede LIN é baseada na arquitetura mestre-escravo. Uma rede é composta por um nó mestre LIN e por pelo menos um nó escravo LIN. Por razões de custo, controladores de comunicação explícitos não são utilizados. Em vez disso a comunicação LIN é implementada por tarefas de *software* em cada nó, as chamadas tarefas de escravo. O mestre LIN também tem uma tarefa de mestre que é usada para coordenar a comunicação da rede.

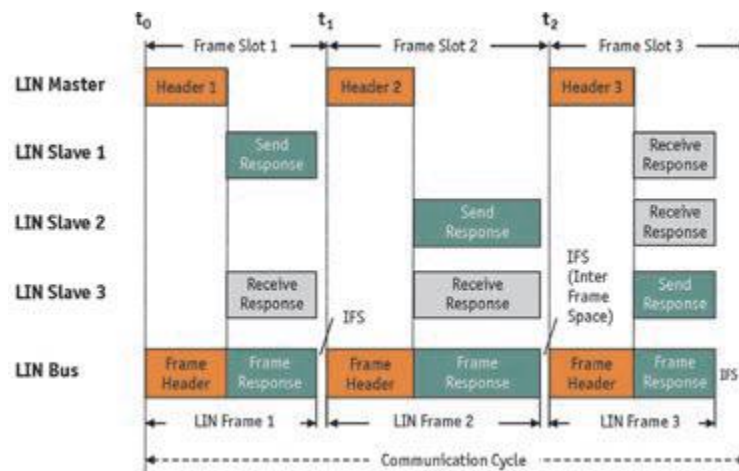


Figura 2.8 – Distribuição de tramas num ciclo de comunicação LIN.

A coordenação é conseguida por meio da execução periódica do cronograma LIN que está organizado numa estrutura de espaços. No início de cada espaço da estrutura a tarefa do mestre transmite um cabeçalho da trama com um identificador que todos os escravos LIN avaliam na sua tarefa de escravo. Imediatamente após o cabeçalho da trama o escravo LIN transmite a trama de resposta associada com o identificador. A trama LIN que consiste no cabeçalho da trama e trama de resposta, está disponível para ser recebida por cada nó LIN devido ao endereçamento de mensagem baseado no identificador.

Devido à falta de um controlador de comunicação, a transmissão de dados em série numa rede LIN é tratada ao longo da interface de comunicação série do microcontrolador e é realizada byte a byte. A interface transmite cada byte com o LSB (*Least Significant Bit*) em primeiro lugar, e o byte é precedido por um bit de início e sucedido um bit de paragem. Ou seja, uma trama LIN é composta por uma combinação de um número de tramas transmitidas pela interface distribuídas entre o cabeçalho da trama e trama de resposta.

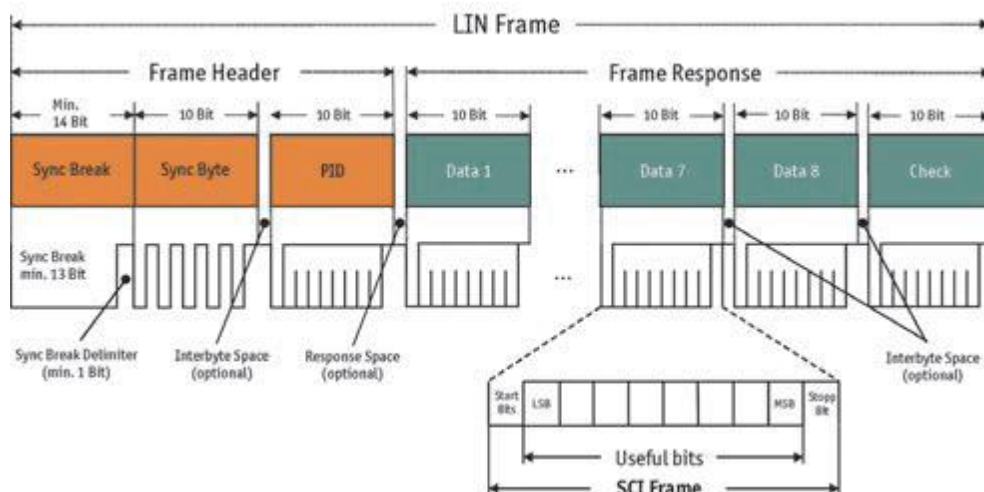


Figura 2.9 – Estrutura de espaços de uma trama LIN.

Ao transmitir o cabeçalho da trama o mestre LIN executa duas tarefas de comunicação principais. Sincroniza os escravos LIN e delega a comunicação atribuindo um emissor e um ou mais recetores para a trama de resposta.

Devido a questões de sensibilidade de custos, os escravos LIN podem usar osciladores internos com uma tolerância de até 14% da frequência. Portanto o mestre LIN transmite uma rutura de sincronização primeiro para todos os escravos LIN saberem que a transmissão de uma trama LIN vai começar. A rutura de sincronização é composta por pelo menos 13 bits dominantes consecutivos e isso provoca um erro na interface de todos os escravos LIN. Ela é encerrada pelo delimitador da rutura de sincronização, com pelo menos um bit recessivo. O mestre LIN transmite o tempo de pulso da comunicação com o subseqüente byte de sincronização tendo o valor 0x55.

Um identificador serve para delegar a comunicação e tem seis bits de comprimento, é protegido por dois bits de paridade com par ou ímpar. O identificador e os dois bits de paridade juntos são referidos como o identificador protegido. Os primeiros 60 identificadores estão disponíveis para comunicação de dados úteis. Os últimos quatro identificadores são reservados, dos quais os identificadores 60 e 61 são utilizados para fins de diagnóstico.

A trama de resposta é composta por até oito bytes de dados e uma soma de verificação para a deteção de erros. É feita uma distinção entre a soma de verificação clássica e a estendida. A soma de verificação clássica é a soma módulo 256 de todos os

bytes de dados. Há um erro de transmissão se o resultado de adicionar a soma módulo 256 para os bytes de dados que chegam não é igual a 0xFF. A soma de verificação estendida também considera o identificador protegido na formação da soma módulo 256.

Uma vez que os escravos LIN são geralmente equipados com microcontroladores muito simples e de baixo custo, eles são autorizados a atrasar um pouco a transmissão da trama de resposta bem como podem inserir pausas de envio entre as tramas transmitidas pela interface. No geral isto pode prolongar a trama de resposta em 40%. O mesmo aplica-se à trama de cabeçalho, principalmente porque existem diferentes métodos para gerar a rutura da sincronização. É essencial considerar a reserva de 40% no projeto do cronograma LIN.

A especificação LIN contém normas para tornar o ciclo de comunicação que é rigidamente definido pelo cronograma LIN mais flexível. Os dois tipos de trama, "trama esporádica" e "trama provocada por evento", são fornecidas para esta finalidade. Ao introduzir estes tipos de tramas suplementares, tornou-se prática comum para referir-se à trama LIN convencional como uma trama incondicional.

A trama esporádica é entendida como uma trama incondicional que compartilha o mesmo espaço da estrutura com outras tramas incondicionais. As tramas esporádicas são transmitidas integralmente pelo mestre LIN conforme necessário pelo que colisões são impossíveis. Se o mestre LIN não tem nenhuma necessidade para qualquer uma das tramas, o espaço da estrutura associado simplesmente permanece vazio.

A trama provocada por evento foi introduzida para comunicar alterações ou eventos esporádicos por parte dos escravos LIN. Essencialmente corresponde a uma trama incondicional mas com a diferença de que múltiplas tramas de resposta de diferentes escravos LIN são atribuídas ao cabeçalho da trama. A trama de resposta que é utilizada para completar o cabeçalho da trama provocada por evento depende das necessidades dos escravos LIN relacionados. A trama de resposta da trama provocada por evento é identificada pelo identificador protegido da trama incondicional associada.

Em contraste com a trama esporádica, as colisões não podem ser excluídas na trama provocada por evento. Em caso de uma colisão o mestre LIN é responsável pela transmissão de todas as tramas incondicionais atribuídos à trama provocada por evento, através da ativação e processamento de um "Cronograma de Resolução de Colisão".

As convencionais tramas incondicionais e as especiais tramas de diagnóstico são ambas adequadas para diagnosticar os escravos LIN. As tramas incondicionais são utilizados para o diagnóstico baseado em sinais simples, enquanto as tramas de diagnóstico são utilizadas tanto para diagnósticos definidos pelo usuário como para diagnósticos baseados num protocolo de transporte normalizado e serviços de diagnóstico uniformes.

A especificação LIN define duas tramas de diagnóstico, a "trama de pedido do mestre" e a "trama de resposta do escravo". A trama de pedido do mestre (ID = 0x60) representa o pedido de diagnóstico. Neste caso, o mestre LIN transmite tanto o cabeçalho da trama como e a trama de resposta. Por exemplo, uma trama de pedido do mestre é transmitida se houver um pedido de diagnóstico. A trama de resposta do escravo (ID = 0x61) corresponde à resposta de diagnóstico. Neste caso, o mestre LIN transmite o cabeçalho, e o escravo LIN específico transmite a resposta.

A especificação LIN define a gestão de estado e a gestão de rede. A gestão de estado especifica que os escravos LIN devem informar o mestre LIN de erros de transmissão que são detetados, tais como erros de paridade ou de soma de verificação. Isto é feito por um "sinal de resposta de erro" numa trama incondicional, no entanto esta trama não contém qualquer informação adicional sobre o tipo de erro. A especificação LIN não define o tratamento de erros, em vez disso deixa essa tarefa para o usuário.

A tarefa principal da gestão de rede LIN é regular a transição de todos os escravos numa rede LIN do estado de comunicação normal ao estado de suspensão e na direção oposta. Se os escravos LIN não detetarem qualquer atividade no barramento por quatro segundos, eles mudam do estado de comunicação normal para o estado de suspensão. A mesma condição provoca um comando de suspensão do mestre LIN, que é na verdade uma trama de pedido do mestre especial.

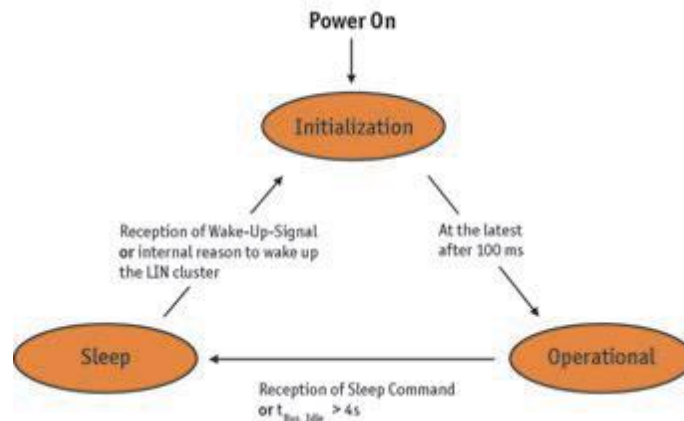


Figura 2.10 – Modelo do estado de comunicação de um escravo LIN.

Por outro lado, quando um escravo LIN deteta um sinal de *wake up* seguido por um cabeçalho válido, ele muda de estado de suspensão através do estado de inicialização para o estado comunicação normal. O sinal de *wake up* consiste de um pulso dominante de 250 microssegundos no mínimo e de 5 milissegundos no máximo e pode ser enviado por qualquer nó LIN. A especificação LIN fixa uma fase de inicialização máxima de 100 milissegundos, ou seja, o mestre LIN deve começar a executar o cronograma LIN o mais tardar após este período de tempo. Se ele permanecer passivo, o escravo LIN envia outro sinal de *wake up*. O número de repetições e o intervalo entre as repetições, bem como os tempos de espera são definidos na especificação.

2.6 Rede FlexRay



De acordo com vários estudos, nunca foi tão seguro conduzir automóveis como nos dias de hoje. Decisivos na redução do número de acidentes e em reduzir a gravidade dos acidentes são os sistemas de segurança ativa e os sistemas de assistência que apoiam os motoristas na sua tarefa de conduzir o veículo. Um estudo realizado por um certo número de fabricantes de automóveis conhecidos mostra, por exemplo, que o ESP (*Electronic Stability Program*) reduziu o número de acidentes de derrapagem em até 80%. Fazendo uma contribuição tão importante para reduzir a gravidade dos resultados de acidentes são as habitáculos de passageiros cada vez mais seguros e sistemas de retenção otimizados [12].

À luz do objetivo de reduzir as mortes na estrada, a indústria automóvel está concentrada em desenvolver os sistemas de segurança ativa existentes, os sistemas de assistência ao condutor e desenvolvimento de sistemas inovadores. Uma vez que estes sistemas não só fornecem informações e instruções mas muitas vezes também fazem intervenções corretivas e assumem as tarefas de condução, isto não seria possível sem as interfaces eletrônicas entre o chassi e a transmissão. A combinação de sistemas de assistência na travagem e assistência na direção está pensado para ter um grande potencial na segurança.

A implementação de funções de segurança e de assistência ao condutor cada vez mais desafiadoras andam de mãos dadas com a integração cada vez mais intensa de ECUs eletrônicas no automóvel. Essas implementações exigem ritmos de transmissão muito altos para transmitir o número crescente de sinais de controlo e de estado, sinais esses que não só necessitam de ser transmitidos de forma extremamente rápida como a sua transmissão também precisa de ser absolutamente determinista.

Essa é a razão para a crescente importância dos sistemas de comunicação que garantam a transmissão rápida e determinista de dados no automóvel. A potencial utilização de sistemas de barramento série ainda prevê a conceção de estruturas e mecanismos de tolerância a falhas. Embora sistemas de barramento série possam oferecer recursos abrangentes e os benefícios de uma maior liberdade de projeto, montagem simplificada, personalização do veículo, etc., os requisitos de transmissão de dados no automóvel são consideravelmente elevados porque estes sistemas pertencem à classe de sistemas tolerantes a falhas. Eles devem continuar a funcionar de forma aceitável mesmo quando ocorre um erro.

A rede CAN não consegue satisfazer estes requisitos devido à sua orientação por eventos e o acesso ao barramento conduzido com prioridade, a sua largura de banda limitada de 1000 kbit/s com base nas restrições físicas no automóvel e falta de estruturas e mecanismos de tolerância a falhas.

A certeza de que a rede CAN dificilmente poderia ser esperada para satisfazer os crescentes requisitos de transmissão de dados no automóvel sobre o médio prazo, levou ao desenvolvimento de uma série de sistemas de barramento série determinísticos e tolerantes a falhas com ritmos de transmissão muito maiores que a rede CAN. Exemplos disso

incluem: TTP (*Time Triggered Protocol*), Byteflight e TTCAN (*Time Triggered CAN*). Apesar de uma parceria de desenvolvimento criada em 2001 entre a Audi e a TTTech empresa promotora TTP, e embora Byteflight tenha sido aplicado com sucesso nos carros BMW da série 7 em 2001, foi a FlexRay que prevaleceu na indústria automóvel.

Uma razão importante para o sucesso da FlexRay foi a fundação do consórcio FlexRay no ano 2000, com dois fabricantes de veículos, a DaimlerChrysler e a BMW, e dois produtores de circuitos integrados, a Motorola e a Philips. Com base na tecnologia do barramento Byteflight originalmente desenvolvido pela BMW, o consórcio FlexRay criou o padrão de comunicação FlexRay, determinista e tolerante a falhas com um ritmo de transmissão de 10 Mbit/s para aplicações de segurança e de tempo extremamente crítico no automóvel.

Hoje em dia, o consórcio FlexRay é composto de sete parceiros essenciais: a BMW, a Bosch, a DaimlerChrysler, a Freescale, a General Motors, a Philips e a Volkswagen. Gradualmente uma série de membros associados juntou-se à organização.

Fazendo uma contribuição significativa para o sucesso da FlexRay foi a documentação detalhada da especificação FlexRay. As duas especificações mais importantes, o protocolo de comunicação e a camada física, estão atualmente na versão 3.0.1 [13].

Tal como no caso da comunicação de dados de uma rede CAN, a comunicação de dados de uma rede FlexRay baseia-se também numa estrutura de comunicação multi mestre. No entanto, aos nós FlexRay não é permitido o acesso ao barramento descontrolado em resposta a eventos relacionados com a aplicação, como é o caso da CAN. Pelo contrário, eles devem obedecer a um ciclo de comunicação definida com precisão que atribui um intervalo de tempo específico para cada mensagem FlexRay, o chamado TDMA (*Time Division Multiple Access*) que assim estabelece os tempos de envio de todas as mensagens FlexRay.

A comunicação desencadeada por tempo não só garante uma comunicação de dados determinista como também assegura que todos os nós de uma rede FlexRay podem ser desenvolvidos e testados independentemente uns dos outros. Além disso, a remoção ou adição de nós FlexRay numa rede existente não deve ter impacto no processo de

comunicação, isso é consistente com o objetivo que é muitas vezes necessário no desenvolvimento automóvel.

Seguindo os paradigmas de arquiteturas de comunicação desencadeada por tempo, a lógica subjacente da comunicação FlexRay consiste em desencadear todas as atividades do sistema quando pontos específicos são atingidos no ciclo de tempo. O sincronismo de todos os nós FlexRay é assegurado por um mecanismo distribuído de sincronização do relógio tolerante a falhas. Todos os nós FlexRay não só corrigem continuamente os tempos de início das mensagens de sincronização regularmente transmitidas, como também corrigem a duração dos ciclos de comunicação. Isso aumenta a eficiência da largura de banda e a robustez da sincronização.

A comunicação FlexRay pode basear-se numa camada física elétrica ou ótica. A transmissão do sinal elétrico é simples e vantajosa do ponto de vista económico. A transmissão do sinal ótico é comparativamente dispendiosa e caracteriza-se por uma substancial melhor compatibilidade eletromagnética em comparação com a transmissão do sinal elétrico.

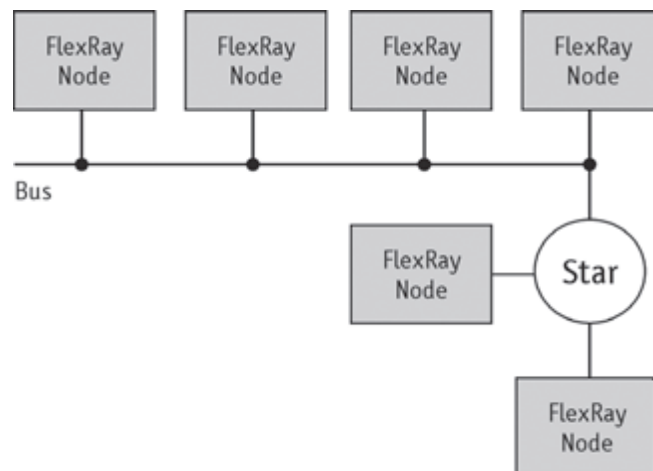


Figura 2.11 – Combinação de topologias ativa e passiva.

A comunicação FlexRay não está vinculada a uma topologia específica. Uma estrutura simples de barramento passivo é tão viável como uma topologia em estrela ativa ou uma combinação das duas. As principais vantagens da topologia em estrela ativa são a possibilidade de desligar ramos de comunicação defeituosos e a capacidade de terminar o barramento com terminações ativas ideais quando a transmissão do sinal físico é elétrico.

Para minimizar o risco de falha, a FlexRay oferece um esquema redundante do canal de comunicação. Este canal de comunicação redundante poderia, por outro lado, ser utilizado para aumentar o ritmo de transmissão para 20 Mbit/s. A escolha entre a tolerância a falhas e a largura de banda adicional pode ser feita individualmente para cada mensagem FlexRay.

Finalmente, um mecanismo de controlo independente denominado "Bus Guardian", garante que um nó FlexRay só recebe acesso ao barramento durante a sua vez no ciclo de comunicação. Isso evita a monopolização do barramento por um nó FlexRay com defeito.

Cada ciclo de comunicação é igual em comprimento e está essencialmente organizado num segmento de tempo estático e num segmento de tempo dinâmico. De importância central aqui é o segmento de tempo estático que começa cada ciclo de comunicação. Este subdivide-se num número de ranhuras estáticas igualmente longas definidas pelo utilizador com o máximo de 1023 ranhuras.

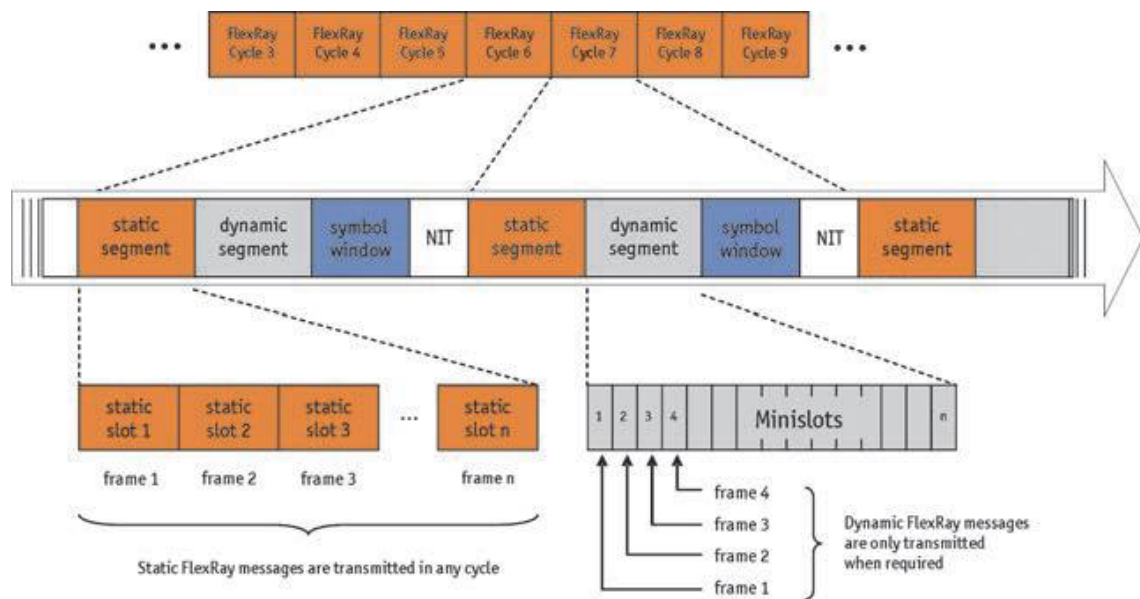


Figura 2.12 – Princípio da comunicação FlexRay.

A cada ranhura estática é atribuída uma mensagem FlexRay para ser enviada por um nó FlexRay. Atribuições de ranhuras estáticas a mensagens e nós FlexRay são feitas por número de ranhura, identificador de mensagem e o valor do contador de ranhura implementado em cada nó FlexRay. Para assegurar que todas as mensagens FlexRay são transmitidas no momento certo e na sequência correta em cada ciclo, os contadores de ranhuras em todos os nós FlexRay são incrementados sincronizadamente no início de cada

ranhura estática. Devido à sua transmissão de dados equidistante e portanto determinista, o segmento de tempo estático está predestinado para a transmissão de mensagens relevantes em tempo real.

Seguindo o segmento estático vai um segmento dinâmico opcional que tem o mesmo comprimento em cada ciclo de comunicação. Este segmento também está organizado em ranhuras mas em vez de ranhuras estáticas são chamadas mini ranhuras. A comunicação no segmento dinâmico também é baseada em alocações e incremento sincronizado dos contadores de ranhuras nos nós FlexRay.

No entanto, não é obrigatório transmitir as mensagens FlexRay associadas às mini ranhuras em cada ciclo de comunicação, em vez disso, só são enviadas quando for necessário. Se as mensagens não são necessárias, o contador de mini ranhuras é incrementado após o período de tempo definido. Enquanto uma mensagem FlexRay dinâmica está a ser transmitida, o incremento do contador de mini ranhuras é retardado pelo tempo de transmissão da mensagem.

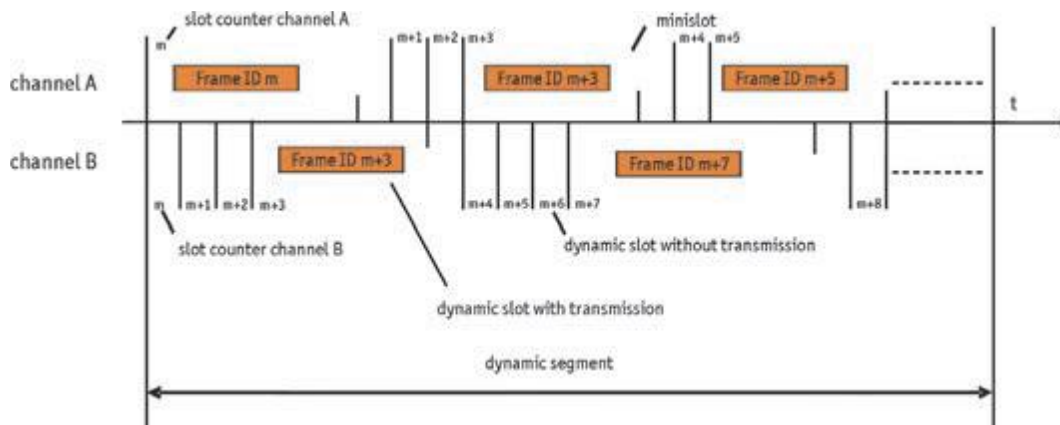


Figura 2.13 – Fluxo de comunicação no segmento dinâmico.

A atribuição de uma mensagem FlexRay dinâmica a uma mini ranhura define implicitamente a prioridade da mensagem FlexRay. Quanto mais baixo for o número da mini ranhura, maior será a prioridade da mensagem FlexRay dinâmica e mais cedo vai ser transmitida, tendo maior probabilidade de transmissão num segmento de tempo dinâmico limitado. A mensagem FlexRay dinâmica atribuída à primeira mini ranhura é sempre transmitida conforme necessário, desde que haja um segmento de tempo dinâmico suficientemente longo.

No projeto da comunicação deve ser assegurado que a mensagem FlexRay dinâmica com menor prioridade também pode ser transmitida, pelo menos desde que não existam outras necessidades de maior prioridade. O projetista de uma rede FlexRay deve também assegurar que a transmissão das mensagens FlexRay dinâmicas mais longas é mesmo possível. Caso contrário, o projeto de comunicação não faria qualquer sentido.

O ciclo de comunicação é completado por dois segmentos de tempo adicionais. O segmento "Symbol Window" serve para verificar a funcionalidade do "Bus Guardian" e o segmento NIT (*Network Idle Time*) fecha o ciclo de comunicação. Durante o NIT os nós FlexRay calculam os fatores de correção necessários para sincronizar os seus relógios locais. No final do NIT, uma correção dos tempos de início das mensagens é feita se necessário sendo que a correção da duração é sempre distribuída ao longo de todo o ciclo de comunicação. Não há nenhuma transmissão de dados durante o NIT.

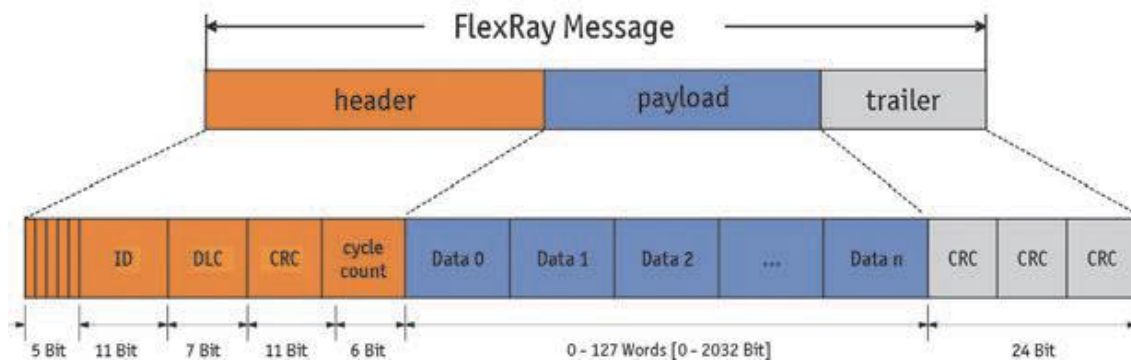


Figura 2.14 – Estrutura de uma mensagem FlexRay.

Os sinais numa rede FlexRay são transmitidos pela mensagem FlexRay bem definida, em que não existe essencialmente qualquer diferença entre os formatos das mensagens FlexRay transmitidas no segmento estático e as transmitidas no segmento dinâmico. Elas são cada uma composta por um cabeçalho, carga útil e reboque.

O cabeçalho compreende o campo de estado com cinco bits, o ID, o tamanho da carga útil e o contador de ciclo. O CRC do cabeçalho com 11 bits protege as partes do campo de estado, do ID e do tamanho da carga útil com uma distância de Hamming de 6. O ID identifica a mensagem FlexRay e representa uma ranhura no segmento estático ou dinâmico. No segmento dinâmico o ID corresponde à prioridade da mensagem FlexRay. Os bits individuais do campo de estado especificam a mensagem FlexRay mais

precisamente. Por exemplo, o bit indicador de trama sincronizada indica se a mensagem FlexRay pode ser usada para sincronização do relógio.

Depois do cabeçalho vem a chamada carga útil. Pode ser transportado um total de até 254 bytes úteis por uma mensagem FlexRay. O reboque inclui o CRC com 24 bits protetor da carga útil. Dada uma carga de até 248 bytes úteis, o CRC garante uma distância de Hamming de 6. Para uma carga útil maior a distância de Hamming é de 4.

2.7 Rede MOST



Os carros de classe privilegiada estão a crescer para se assemelharem a escritórios móveis. Em resposta à procura dos clientes, um número crescente de meios de informação e de entretenimento estão a fazer o seu caminho para os automóveis. Os desafios mais significativos nesta área são, em primeiro lugar, manter o custo da cablagem tão baixo quanto possível, em segundo lugar, satisfazer plenamente os requisitos funcionais elevados de um sistema de informação e de entretenimento no carro. Como resultado, o sistema de barramento MOST (*Media Oriented Transport System*) é agora usado para transmitir sinais de áudio e de vídeo em muitos modelos automóveis.

A eletrónica é responsável por um grande número de funções de segurança e de sistemas de conforto inovadoras em tecnologia automóvel. Os especialistas preveem que em poucos anos a eletrónica representará uma participação de até 30% do valor do veículo e o mercado mundial de eletrónica para carros crescerá aproximadamente 6% ao ano. Está previsto a indústria automóvel exibir taxas de crescimento rápidas, sobretudo na área de informação e entretenimento, dado o contínuo crescimento de veículos nas estradas. Um cidadão médio gasta cerca de 270 horas por ano num carro, sendo estas a caminho do trabalho ou do centro comercial ou das férias [14].

Ao longo do tempo, o rádio do carro foi completado pelo leitor de CD e de MP3. Este passou a incluir dispositivos de navegação e finalmente ecrãs de visualização também fizeram o seu caminho para os carros para reproduzir filmes vídeo e DVD. Além disso, unidades mãos-livres Bluetooth com microfones integrados estão a transformar gradualmente o cockpit do carro num centro de multimédia, em que todas as listas de

reprodução e diretórios de um leitor de MP3 podem ser exibidos diretamente no visor no veículo.

O custo e o esforço de cablagem já extensos estão a aumentar devido ao crescimento contínuo em soluções para redes de dispositivos de informação e de entretenimento de dimensões que não podem ser geridas por mais tempo. Felizmente, desde o início alguns fabricantes de automóveis reconheceram as vantagens que uma rede de barramento também iria oferecer nesta área. Em meados da década de 1990, a BMW e a Daimler começaram a desenvolver uma tecnologia de comunicação uniforme para a transmissão em série de sinais de áudio e de vídeo no veículo baseado no barramento D2B (*Digital Data Bus*), desenvolvido pela Matsushita e Philips.

Em 1998, a BMW, a Daimler, a Harman/Becker e a SMCSC fundaram a Cooperação MOST. Desde aquela época, a MOST estabeleceu-se como um padrão para a transmissão de dados multimédia em veículos. A Cooperação MOST é composta por 15 fabricantes de automóveis internacionais e mais de 70 produtores de dispositivos. A organização dos utilizadores lançou as bases para o sucesso da tecnologia através da definição de uma especificação extensa. A versão 3.0 da especificação MOST já existe desde julho de 2010. Está organizada nas secções de Aplicação, de Rede e de *Hardware* [15].

A secção de Aplicação descreve um modelo de dispositivo lógico baseado principalmente em métodos orientados a objetos, com o propósito de modelagem transparente e controlo distribuído de sistemas de informação e de entretenimento. Além disso, define um modelo de comunicação hierárquica, bem como serviços para a gestão de sistemas de informação e de entretenimento. A secção de Rede descreve o controlador de interface da rede MOST e os seus serviços, o gerenciamento da rede e a manipulação do transporte de dados num sistema MOST. A secção de *Hardware* lida com os aspetos da estrutura do *hardware* de um dispositivo MOST.

Um dispositivo MOST é subdividido num nível funcional e num nível de rede. No nível funcional, funcionalidades de informação e entretenimento são incorporadas nos chamados blocos de função. Cada bloco de função, por exemplo o "*Audio Disk Player*", fornece a rede MOST com um conjunto específico de funções, por exemplo a "*Track Position*" que pode ser acedida por tipos de operação tais como "*Set*" para definir uma faixa ou "*SetGet*" para definir e ler uma faixa.

Endereços funcionais são atribuídos a ambos, os blocos de função e as funções fornecidas por um bloco de função. Eles podem ser tomados a partir do chamado catálogo de funções, tal como os identificadores dos tipos de operação. Por exemplo, o "Audio Disk Player" tem o FBlockID = 0x31 e a função "Track Position " tem o FktID = 0x202.

A separação da função e da rede e a modelagem funcional torna possível a implementação de um modelo de comunicação funcional que é totalmente independente de componentes físicos. Portanto, não importa qual dos dispositivos MOST é usado para incluir uma função específica.



Figura 2.15 – Hierarquia do sistema MOST.

Os sistemas MOST são modelados numa filosofia de controlo hierárquico de três níveis, baseada no princípio mestre-escravo. Colocada no nível hierárquico mais alto está a HMI (*Human Machine Interface*), um controlador exposto que fornece ao utilizador funcionalidades no geral. No nível hierárquico médio estão os controladores do sistema, eles cobrem parte da funcionalidade do sistema e compartilham o seu conhecimento parcial do sistema com a HMI como mestre.

O nível hierárquico mais baixo é constituído pelos escravos do sistema, cujas funções são usadas por um ou mais controladores. Eles não estão equipados com qualquer conhecimento do sistema e isso aumenta substancialmente a sua flexibilidade no que diz respeito à configuração. É fácil adicionar escravos do sistema ou removê-los de um sistema MOST. Os comandos MOST são usados para controlo da comunicação. Os seus principais componentes são o FBlockID, o FktID, o tipo de operação e até 65535 bytes úteis.

A secção de Aplicação define blocos de função de nível superior e funções de gestão do sistema. As funções do sistema incluem por exemplo a função FktIDs com o FktID = 0x000, que é usada para consultar as funções suportadas por um bloco de função. A função do sistema "Notificação" com o FktID = 0x001, por outro lado, permite a criação da matriz de notificação para um bloco de função. Na matriz de notificação está a informação sobre qual o dispositivo MOST que deve ser notificado se uma determinada propriedade de um bloco de função mudou. Este mecanismo impede um aumento desnecessário da carga do barramento no sistema MOST.

Para consultar os seus blocos de função e os endereços, cada dispositivo MOST tem o bloco de função "*Net Block*" com o FBlockID = 0x01. Os blocos de função podem aprender sobre os blocos de funções implementados num dispositivo MOST utilizando a função FBlockIDs com o FBlockID = 0x000. Os FktIDs 0x002, 0x003 e 0x004 são usados respetivamente para encontrar o endereço físico, endereço lógico e endereço de grupo de um dispositivo MOST.

O mestre da rede desempenha um papel importante na gestão de um sistema MOST. Ele é responsável pela inicialização do sistema e gestão do registo central. Este registo contém os endereços lógicos dos dispositivos MOST implementados num sistema MOST e os endereços dos blocos de função contidos nesses dispositivos MOST.

A interface da rede MOST assegura que os blocos de função alojados nos vários dispositivos MOST são capazes de comunicação real entre si. Os serviços de sistema MOST, serviços de sistema de baixo nível e serviços de rede MOST, fornecem as funcionalidades de comunicação necessárias para o transporte de todos os dados multimédia relevantes, fluxos de bits contínuos, pacotes de dados e dados de controlo. Serviços de sistema de baixo nível são implementados no NIC (*Network Interface Controller*) e são colocados sobre a camada física.

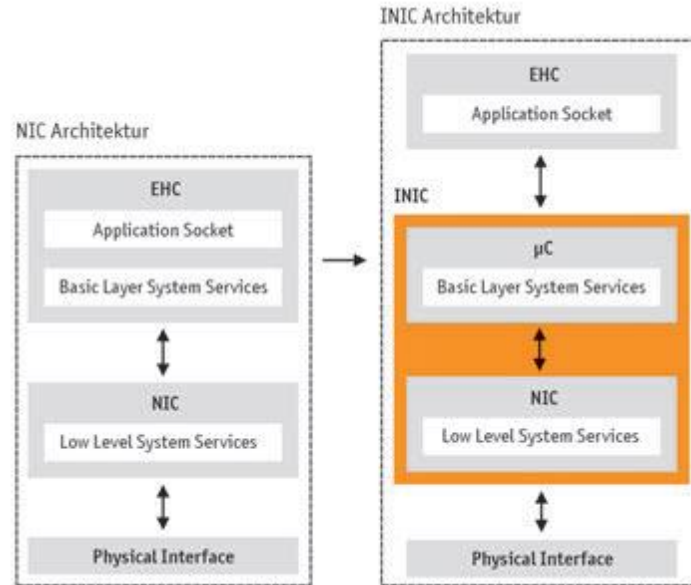


Figura 2.16 – Diferença entre as arquiteturas NIC e INIC.

Os serviços de rede MOST que abrangem a camada de transporte sob a forma de serviços de sistema de camada básica, e maior gestão sob a forma de suporte de aplicação, estão alojados num EHC (*External Host Controller*) e controlam o NIC. Deve-se assegurar que o EHC pode servir as partes da interface de rede com tempo crítico. Ao longo do tempo, com o desenvolvimento progressivo da tecnologia MOST de MOST 25 a MOST 50 e MOST 150, esta arquitetura depara-se agora com os seus limites.

Nos novos desenvolvimentos o INIC (*Intelligent Network Interface Controller*) substitui o NIC. Enquanto o INIC assume o controlo da execução de partes com tempo crítico do controlador de rede do EHC, apenas uma parte relativamente pequena do controlador de rede ainda é executado no EHC, o que representa essencialmente um suporte para a aplicação. Assim a arquitetura INIC alivia a carga do EHC. Para o controlo, o INIC fornece o EHC com uma interface funcional, a chamada INIC-API. As funções do INIC são encapsulados num bloco de função chamado FBlockINIC.

A tecnologia MOST permite a transmissão de fluxos de bits contínuos sem *buffer*. Isto envolve ter um dispositivo MOST especialmente designado a abastecer a trama MOST a uma frequência fixa (44,1 kHz ou 48 kHz) para o meio de transmissão, o qual é tipicamente ótico.

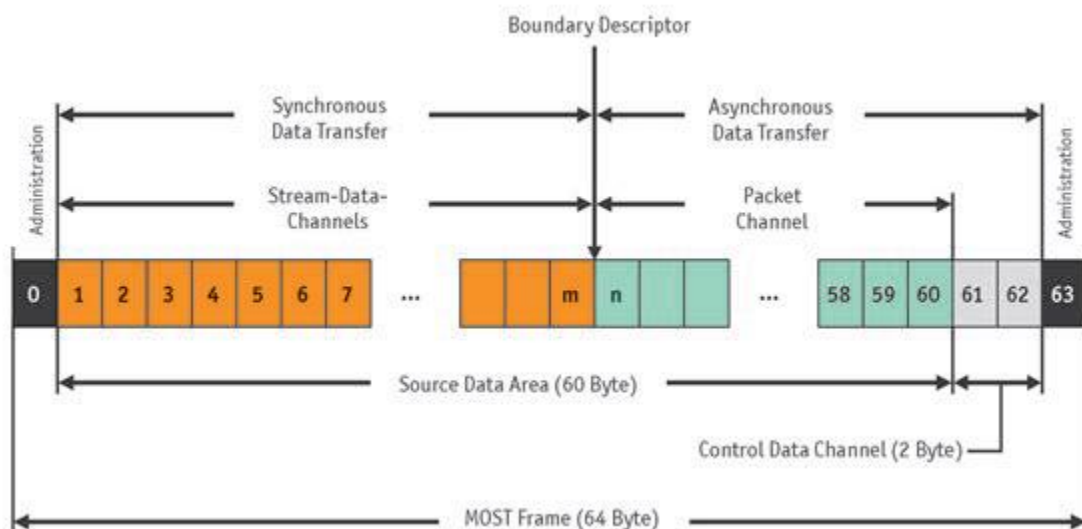


Figura 2.17 – Composição de uma trama MOST.

Num sistema MOST 25, a trama MOST fornece 60 canais de transmissão com 8 bits ou 15 words de 4 bytes cada, para a transmissão de fluxos de bits contínuos. A largura de banda de um canal de transmissão é 352,8 kbit/s a 44,1 kHz ou 384 kbit/s a 48 kHz.

Uma vez que os dispositivos MOST estão fisicamente interligados num anel, cada trama MOST deve passar por todos os dispositivos MOST com a frequência fixada pelo sincronismo do metre. Assim que os parceiros de comunicação relevantes estiverem conectados ao mesmo canal começa a transmissão de fluxos de bits contínuos.

A conexão ou a desconexão é geralmente feita por uma consulta do bloco de função CM (*Connection Master*) com o FblockID = 0x03. Para este efeito, o CM proporciona duas funções, a "*BuildSyncConnection*" e a "*RemoveSyncConnection*".

Na estrutura da construção de uma conexão, o CM solicita que a fonte de dados relevante, por exemplo um sintonizador de TV, tenha o número adequado de canais de transmissão alocados pelo sincronismo do mestre. Isso porque o sincronismo do mestre é responsável pela gestão da tabela de alocação de recursos de canal. O CM passa os endereços dos canais de transmissão alocados para o recetor de dados, por exemplo para um ecrã de visualização, para que ele possa conectar-se aos canais de transmissão. Finalmente, o CM atualiza a tabela de conexão sincronizada, que ele usa para gerenciar todas as conexões síncronas. A desconexão é realizada de acordo com o mesmo esquema.

Para permitir a transmissão de pacotes de dados, o utilizador tem a opção de reduzir o número de canais para o fluxo de bits contínuo até 24 usando o descritor de limite. Todos os canais de transmissão que não estão reservados para o fluxo de bits contínuo são combinados para formar o canal de pacotes de dados. Neste canal é possível um ritmo de transmissão máximo de até 12,7 Mbit/s com uma frequência de 44,1 kHz, um ritmo máximo de até 13,8 Mbit/s é atingida a 48 KHz. O descritor de limite é gerido pelo bloco de função mestre da rede com o FBlockID = 0x02. Ele pode ser definido através da função "Limite" com o FktId = 0xA03.

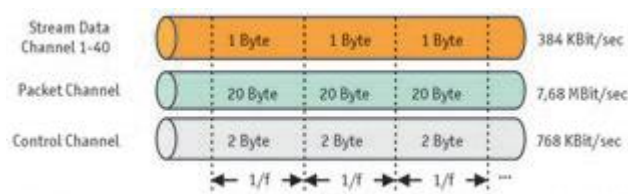


Figura 2.18 – Canais MOST à frequência de 48 KHz.

Um protocolo de serviços de sistema de baixo nível é utilizado para transmitir pacotes de dados. A estrutura compreende o campo de arbitragem, endereço de origem e de destino, código de comprimento de dados, campo de dados e proteção de dados. Um sinal que circula no anel regula o acesso ao barramento. O dispositivo MOST que tem o sinal do anel pode aceder ao canal de pacotes.

Finalmente, o sistema MOST deve transmitir os comandos MOST necessários para a gestão e controlo. As mensagens de controlo usadas são transmitidas no canal de controlo com 2 bytes. São necessárias 16 tramas MOST para transmitir uma mensagem de controlo. A largura de banda em 44,1 KHz é 705,6 kbit/s e em 48 KHz é 768 kbit/s. A transmissão de mensagens de controlo também se baseia num protocolo de serviços de sistema de baixo nível.

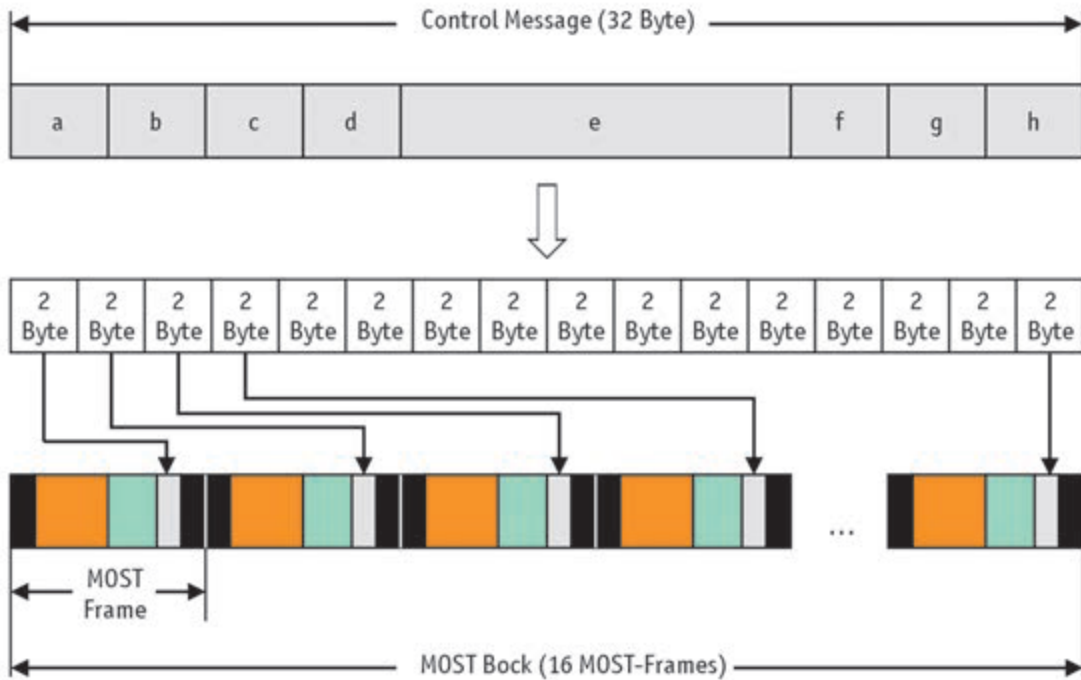


Figura 2.19 – Bloco MOST com a mensagem de controlo.

Atualmente, os condutores óticos de fibras de polímeros são a tecnologia de ponta para a transmissão de sinais de áudio e de vídeo no sistema MOST. Em geral, as propriedades técnicas das fibras de polímeros são muito superiores às dos meios de transmissão elétrica. Destacam-se a sua excelente imunidade eletromagnética e o relativamente alto ritmo de transmissão de dados de até 500 Mbit/s. Além disso, a combinação de fibra ótica, um díodo emissor de luz vermelho como fonte de luz e um fotodíodo PIN de silício como recetor de luz representa uma forma muito económica e comparativamente simples e manejável de transmissão de sinal ótico.

O MOST 150, que segue o MOST 50, é um sistema MOST já largamente utilizado nos automóveis. É baseado nesta tecnologia de emissor e de recetor e oferece ao utilizador um ritmo de transmissão de 150 Mbit/s. Pode lidar com os caminhos relativamente curtos do carro, até 20 metros de comprimento sem quaisquer problemas.

2.8 Sistemas de iluminação nos automóveis

A visão é com certeza o sentido mais importante para a orientação das pessoas que se propõem a conduzir um automóvel. Entretanto esta tarefa torna-se impossível de ser realizada durante a noite sem a ajuda dos faróis dianteiros dos veículos, cuja função principal é justamente iluminar a via em que o mesmo se encontra, permitindo ao seu condutor visualizar e avaliar as condições do caminho a percorrer e, sempre que necessário, desviar de obstáculos em tempo útil, nesta situação os mesmos podem ser considerados como um fator de segurança. Além disso, os faróis também têm como função tornar o automóvel visível a outros condutores e iluminar as laterais das vias em que o automóvel circula, permitindo a identificação e a leitura das sinalizações de trânsito pelo condutor.

Logo após os primeiros automóveis serem apresentados ao mundo, no final do século XIX, surgiu a necessidade de utilização dos mesmos durante a noite. Foram utilizados inicialmente os mesmos dispositivos usados nos comboios que eram lanternas a óleo, que em seguida ganharam um espelho ou anteparo de metal polido, atrás da chama, para melhorar a eficiência, concentrando a luz na direção desejada e, para evitar que a chama se apagasse usava-se um vidro transparente como proteção. Isto bastou para permitir passeios noturnos com os primeiros veículos automóveis. Mas foi com a invenção da lâmpada incandescente que os faróis automóveis deram um grande salto no seu desempenho e facilidade de uso, ainda no início do século XX. Os próximos saltos significativos em termos tecnológicos vieram a acontecer com o aparecimento do farol selado na década de 20, e depois já nos anos 90, as lâmpadas de descarga de gás ganharam destaque.

Neste momento é a tecnologia de iluminação LED que domina no processo de produção em escala. Esta tecnologia está baseada na emissão de luz a partir de uma pastilha semicondutora de alta eficiência que combina a emissão de luzes de várias cores mas principalmente a azul que permite, após uma combinação adequada, uma luz totalmente branca aos olhos humanos, perfeita para qualquer aplicação cujo objetivo final seja iluminar. De fato, os LEDs já eram utilizados anteriormente em faróis traseiros, a grande novidade no mundo da iluminação automóvel no século XXI é o surgimento dos primeiros faróis dianteiros a LED.

Esta tecnologia trouxe algumas vantagens significativas em relação às anteriores. De entre elas podem-se citar: menor potência para uma mesma iluminação, menor volume ocupado, maior durabilidade, luz sem a componente infravermelho e uma maior flexibilidade no *design* proporcionada aos projetistas de veículos. Logicamente tudo isso esbarra em algumas limitações técnicas como a dissipação de calor e as grandes variações do tom de branco que os fabricantes dos LEDs e de faróis tiveram de solucionar e por isso os LEDs de alta potência ainda apresentam um custo elevado [16].

O próximo grande passo para o futuro da iluminação automóvel é a tecnologia OLED (*Organic LED*). Após os LEDs se terem tornado o padrão para todos os fabricantes de automóveis, já começou a utilização da tecnologia OLED na produção dos primeiros veículos.

Os OLED proporcionam possibilidades completamente novas no *design* de lâmpadas e luminárias. Como painéis irradiadores, os OLED são adequados para uso como luz de sinalização, farolins ou mesmo para a iluminação interior do carro. O resultado é superfícies de luz homogénea em todos os formatos, muitas cores, longa vida e energeticamente eficientes. Graças a uma investigação extensiva, engenheiros conseguiram recentemente ultrapassar as últimas dificuldades técnicas, aumentar significativamente a resistência à temperatura para assegurar um uso longo sem problemas [17].

3. Solução Desenvolvida

3.1 Descrição geral

Como já foi referido nos objetivos desta dissertação, pretendeu-se estudar, projetar e desenvolver uma solução de comunicação digital série que permitisse interligar vários nós de uma rede de modo a comandar um sistema de iluminação. Propôs-se assim neste trabalho simular um sistema de iluminação de um automóvel com comando por rede CAN, numa tentativa de o aproximar a um veículo automóvel real. Para este efeito considerou-se a existência de cinco nós com módulos de rede CAN com dois pares de condutores entre cada nó (alimentação e comunicação). Na figura 3.1 está representado um diagrama geral simplificado da solução proposta para o controlo da iluminação baseado na rede CAN, onde se evidenciam os cinco módulos distintos, dentro dos retângulos a tracejado, e o par de condutores que estabelece a comunicação entre os mesmos. Por questões de simplificação o outro par não está representado na figura. De notar que cada bloco representa pelo menos um circuito integrado e as setas nos condutores representam o sentido da informação.

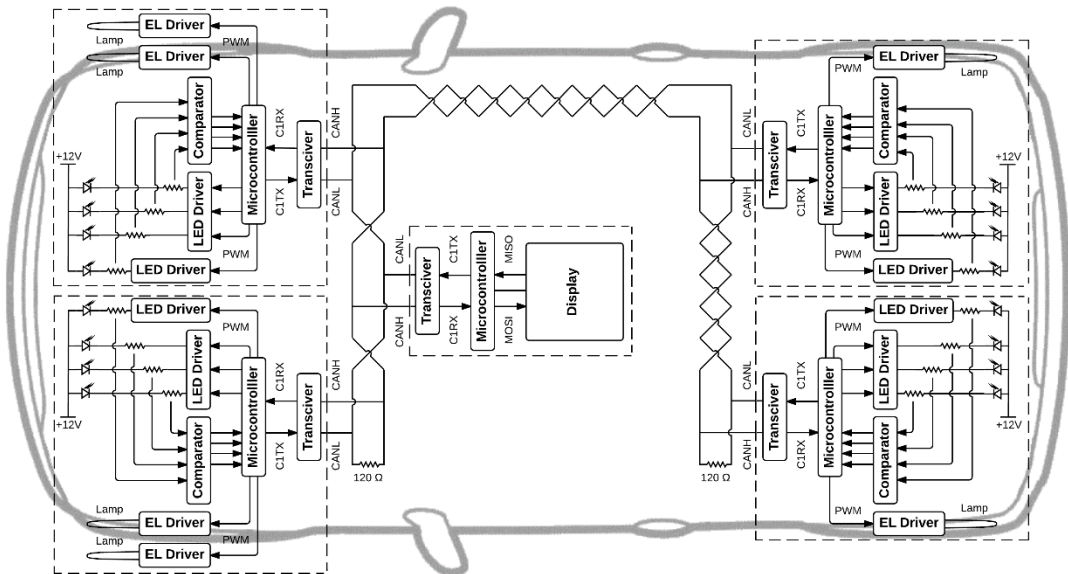


Figura 3.1 – Diagrama geral simplificado da solução proposta para o controlo da iluminação baseado na rede CAN sem referência dos componentes.

Todos os módulos realizam tarefas distintas apesar de quatro deles serem bastante semelhantes em termos de funcionalidade. Quatro desses módulos simulam o sistema de iluminação do automóvel. Estes foram denominados de módulos frontal direito e esquerdo e módulos traseiro direito e esquerdo. O quinto módulo foi denominado de módulo central e é responsável por gerir a informação de um ecrã táctil bem como de alguns botões de pressão colocados para simular comandos do condutor.

Os dois pares de condutores têm finalidades distintas. Um dos pares fornece a necessária alimentação aos módulos da rede CAN. Apesar dos módulos funcionarem com tensões mais reduzidas optou-se por efetuar a alimentação a 12V simulando a tensão da alimentação do automóvel. Em todos os módulos foram adicionados reguladores de tensão de 12V para 3,3V e de 12V para 5V uma vez que os diversos componentes que integram os módulos requerem diferentes tensões de alimentação. Além disso, com esta, opção evitam-se cablagens adicionais e alimentações centralizadas. O outro par de condutores é responsável pela comunicação entre módulos, distribuindo as linhas de comunicação CAN *High* e CAN *Low*. Para a solução desenvolvida optou-se por um par torçado sem blindagem dadas as curtas distâncias entre módulos e dada a baixa probabilidade de existência de interferências eletromagnéticas. Numa aplicação real este par deveria possuir blindagem de acordo com as recomendações da ISO 11898-2.

De modo a simular o sistema de iluminação de um automóvel optou-se por duas tecnologias diferentes. Utilizou-se fio eletroluminescente, que é uma tecnologia pouco divulgada, para simular as luzes diurnas e os indicadores de mudança de direção do automóvel. Utilizaram-se luzes LED (*Light Emitting Diode*) de alto brilho para simular a restante iluminação do automóvel.

É de salientar que numa fase inicial de estudo se incluiu um circuito integrado comparador com a função de averiguar o bom funcionamento de cada LED, contudo este circuito não foi incluído na aplicação final desenvolvida uma vez que se considerou que iria trazer um aumento desnecessário de complexidade ao sistema.

3.2 Escolha do material

São diversos os fabricantes de microeletrónica que dispõem de componentes para aplicações com a rede CAN, entre os quais se destacam a *Maxim Integrated*, a *Microchip Technology*, a *Texas Instruments* e a *Atmel*. Em resultado desta análise optou-se pelos componentes da *Microchip* para a aquisição dos microcontroladores e de transsetores necessários bem como da maioria dos restantes componentes periféricos. Esta escolha deveu-se às características dos produtos corresponderem ao pretendido para a aplicação em causa, ao conhecimento já adquirido no decurso da formação académica acerca do *software* de programação de microcontroladores deste fabricante e ainda devido à forte política de fornecimento de amostras gratuitas por parte destes. Outro aspeto que foi levado em consideração é o facto de este fabricante disponibilizar muitos dos seus componentes com encapsulamentos que permitem a sua utilização diretamente em *breadboard* (PDIP, SPDIP, TO-220, etc.).

A figura 3.2 ilustra, à semelhança da figura 3.1, um diagrama geral simplificado da solução proposta para o controlo da iluminação baseado na rede CAN mas agora com a referência de cada circuito integrado escolhido para o respetivo bloco. De assinalar que neste diagrama não estão representadas as alimentações dos circuitos integrados nem os reguladores de tensão de cada módulos por uma questão de simplicidade.

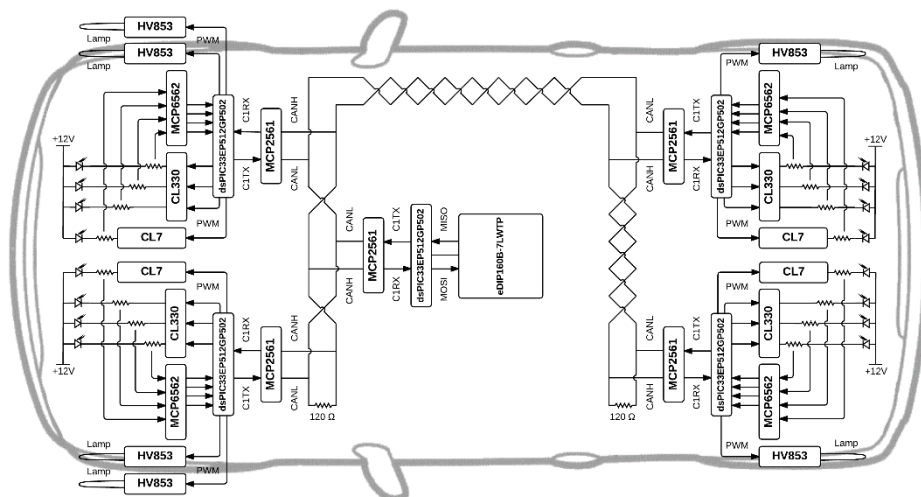


Figura 3.2 – Diagrama geral simplificado da solução proposta para o controlo da iluminação baseado na rede CAN com referenciação dos componentes.

Microcontrolador

O microcontrolador escolhido para a aplicação desenvolvida foi o dsPIC33EP256GP502, ver figura 3.3, que pertence a família dos DSP (*Digital Signal Processor*) da *Microchip* e possui uma arquitetura de 16 bits com uma velocidade máxima de 70 MIPS (*Millions of Instructions Per Second*). Este modelo possui uma memória de 256 Kbytes, considerada suficiente para esta aplicação. Esta família de processadores disponibiliza a função PPS (*Peripheral Pin Select*) que permite atribuir a alguns pinos uma função específica, entre várias, o que facilita as ligações e adaptação à aplicação em desenvolvimento. As principais características que motivaram a escolha deste microcontrolador foi o facto de este possuir um módulo integrado de comunicação CAN (*ECAN – Enhanced CAN*), bem como um módulo SPI (*Serial Peripheral Interface*) necessário para comunicar com o ecrã tátil. Para além do aspeto das comunicações foi este o escolhido também, por ter um oscilador interno dispensando assim vários componentes externos [18].



Figura 3.3 – Aparência do dsPIC33EP256GP502.

O ECAN é o mais recente aperfeiçoamento aos módulos CAN, este oferece muitas melhorias em relação aos módulos CAN, em termos da quantidade de *buffers* tanto de transmissão como receção, também da quantidade de filtros de aceitação de mensagens, e a operação FIFO por *hardware*. Ao mesmo tempo, os módulos ECAN são totalmente compatíveis com os módulos CAN. Sendo que o ECAN oferece três modos de funcionamento, o modo 0 é compatível com os módulos CAN, logo uma aplicação desenvolvida para um módulo CAN trabalha sem grandes alterações num módulo ECAN,

o modo 1 funciona de forma semelhante ao modo 0 mas utiliza as melhorias introduzidas pelo ECAN e o modo 2 tem os mesmos recursos do modo 1 mas recebe FIFO por *hardware*.

Tendo em conta as suas características e flexibilidade, o ECAN é útil para variadas aplicações. Em termos de performance, o ECAN destaca-se quando a carga do barramento é superior a 10% uma vez que praticamente não necessita de tempo de processamento mas em contrapartida necessita de um código maior, o CAN ainda é usado em aplicações onde a carga do barramento é inferior a 10% dado que requer um código mais pequeno [19].

Transcetor

O transcetor selecionado para a aplicação desenvolvida foi o MCP2562, ver figura 3.4. Este serve de interface entre o controlador CAN e o par de condutores do barramento da rede e tem implementado os requisitos para a camada física de acordo com a ISO-11898-2 e ISO-11898-3. Algumas características que levaram a seleção deste transcetor foi cumprir os requisitos para veículos automóveis (AEC-Q100 Rev. G) tanto ao nível da EMC (*Electromagnetic Compatibility*) como da ESD (*Electrostatic Discharge*) entre outras proteções e ainda suportar a variante da rede CAN de alta velocidade (1 Mbit/s).

Note-se que o microcontrolador escolhido funciona com uma tensão de 3,3V e os transdutores funcionam com uma tensão de 5V, sendo necessário informar o transcetor que a comunicação com o controlador CAN será realizada a 3,3V. Este transcetor, ao contrário de outros, permite ativar esta opção através do pino V_{IO} disponível para este efeito [20].



Figura 3.4 – Aparência do MCP2562.

Controladores LED

Foram escolhidos para esta aplicação dois controladores *LED*, o CL330 e o CL7 da *Microchip*, ver figura 3.5, sendo que o CL330 é capaz de controlar três *LEDs* separadamente com uma corrente de $30 \text{ mA} \pm 6 \%$ em cada um e o CL7 é capaz de controlar

um LED com uma corrente de $100 \text{ mA} \pm 5 \%$. Para estes valores de corrente, o fabricante aconselha a utilização de *LEDs* de “alto brilho” e “alto rendimento” ou *multichip* respetivamente uma vez que os *LEDs* convencionais apenas suportam 20 mA. Ambos os circuitos integrados permitem regular a intensidade dos *LEDs* através de sinal *PWM* na entrada lógica. Esta técnica foi usada para controlar e simular os médios e máximos. Ambos os integrados têm também proteção de temperatura excessiva em caso de falha de *LED* [21] [22].

De salientar ainda que estes circuitos integrados apenas são disponibilizados com o encapsulamento *SOIC (Plastic Small Outline)* com 8 pinos, por isso foi necessário utilizar placas de *breakout*, estas placas fazem a adaptação dos pinos permitindo a fácil integração de circuitos integrados com este encapsulamento em *breadboards*, como se observa na figura 3.5.

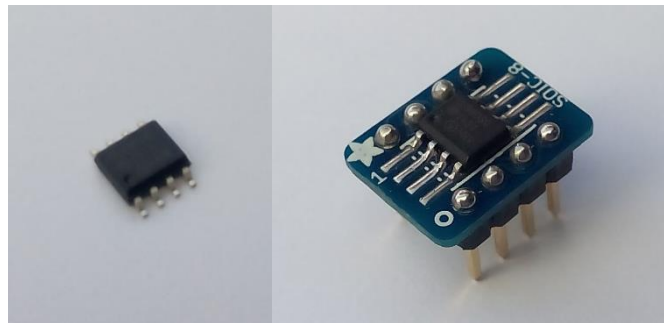


Figura 3.5 – Aparência do CL7 à esquerda e do CL330 em placa breakout à direita.

Controladores para fio eletroluminescente

Foi selecionado para esta aplicação o controlador de fio eletroluminescente HV852, ver figura 3.7, que alimenta o fio eletroluminescente com tensão alternada à frequência escolhida. É possível fazer uma analogia entre o fio eletroluminescente e um condensador, que neste caso emite luz ao carregar e descarregar devido a uma película de fósforo e essa luz será mais intensa se for aumentada quer a tensão quer a frequência, sendo que neste controlador a frequência pode ser definida por uma resistência ou por um *clock* externo; A intensidade da luz pode também ser controlada por um sinal *PWN* na entrada. Este circuito converte uma tensão de alimentação contínua de 5 V para aproximadamente 80 V em tensão alternada, através de um conversor interno comutado a alta frequência ligado com uma montagem em ponte completa (também interna) com MOSFETs, ver figura 3.6, dispensando o uso externo de qualquer componente passivo [23].

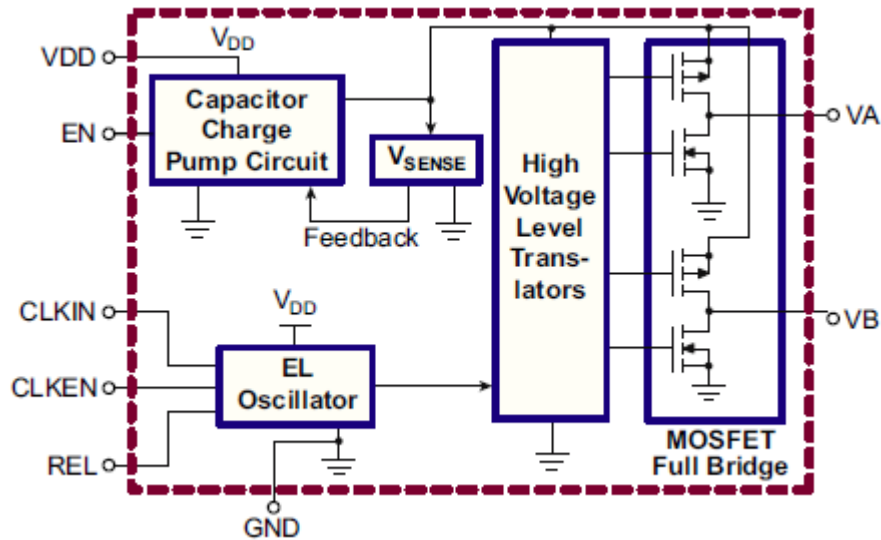


Figura 3.6 – Diagrama de blocos funcional do HV852.

Destacar ainda que este circuito integrado tem o encapsulamento MSOP (*Plastic Micro Small Outline*) por isso também foi necessário utilizar placas de *breakout* para possibilitar a ligação em *breadboards*.

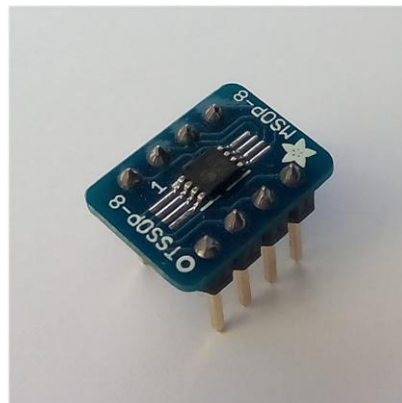


Figura 3.7 – Aparência do HV852 em placa breakout.

Reguladores de tensão

Os reguladores de tensão escolhidos para esta aplicação foram o MCP1755S, o MCP1703A e o MIC29150, ver figura 3.8, que regulam a tensão a partir da alimentação de 12 V para os 5 V e para os 3,3 V em cada módulo. Estes reguladores apresentam boas características, nomeadamente proteção contra sobrecarga e sobreaquecimento e também proteção contra sobretensões transitórias [24] [25] [26].

Para os quatro módulos periféricos de iluminação foi escolhido o MCP1755S tanto para os 5 V como para os 3,3 V. Este circuito integrado suporta uma corrente de 300 mA, sendo suficiente para cada módulo. O encapsulamento escolhido foi o SOT-223 (*Plastic Small Outline Transistor*) com 3 pinos e foram colocados em placas *breakout* para possibilitar a ligação em *breadboards*. Para o módulo central foi escolhido o MCP1703A uma vez que neste módulo não tem tanto consumo e este circuito integrado suporta uma corrente mais baixa de 250 mA. Foi escolhido o encapsulamento SOT-23A que suporta menos dissipação de energia mas suficiente para este módulo e foi também colocado em placas *breakout*. Para alimentar o ecrã tátil foi escolhido o MIC29150 com uma tensão de 3,3 V, este circuito integrado fornece uma corrente de 1,5A quando utilizado com dissipador. O encapsulamento é o TO-220 (*Plastic Transistor Outline*) que o permite ligar diretamente em *breadboards*.

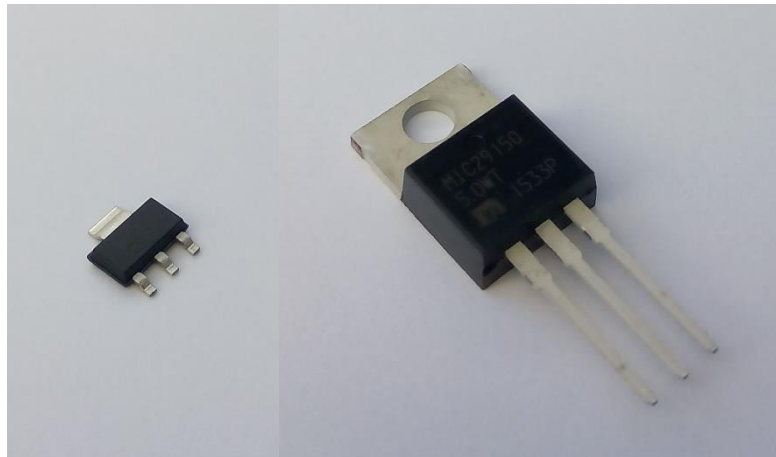


Figura 3.8 – Aparência do MCP1755S à esquerda e do MIC29150 à direita.

Ecrã Tátil

O ecrã tátil selecionado para a aplicação desenvolvida foi o eDIP160B-7LWTP da *Electronic Assembly*, ver figura 3.9, que é do tipo LCD e disponibiliza várias funções gráficas. As funções gráficas são disponibilizadas pela inteligência integrada no ecrã que facilita a programação do mesmo através de até 256 macros de comandos. Este ecrã é retro iluminado e apresenta a informação a azul e branco numa área de 160 por 104 pontos, as suas dimensões são 81,5 x 67,5 x 13,6 mm. Ao nível das comunicações possui RS-232, I²C e SPI sendo que a RS-232 vai ser utilizada para comunicar com um computador para programar o ecrã e a SPI vai ser utilizada para comunicar com o microcontrolador para troca de informação da aplicação. Suporta tensão de alimentação de 3,3 V ou 5 V e a

corrente será 190 mA ou 125 mA respetivamente, na aplicação desenvolvida será alimentado com a tensão 3,3 V [27].

A ligação ao ecrã táctil pode ser realizada pelo conector ZIP mas na aplicação desenvolvida será ligado às *breadboards* diretamente pelos dois conectores DIP de 20 pinos cada um, sendo necessário ter em atenção o afastamento pouco comum dos dois conectores DIP.



Figura 3.9 – Aparência do eDIP160B-7LWTP.

3.3 Implementação do *hardware*

A aplicação desenvolvida foi implementada em *breadboards* dispostas de forma a visualizar-se de forma intuitiva os vários módulos que integram o sistema de iluminação de um automóvel real, cada um colocado num dos seus cantos. Começou-se por fazer o projeto da montagem recorrendo ao *software Fritzing* da *Friends-of-Fritzing* [28] de forma a otimizar a disposição dos componentes nas *breadboards*. Vemos na figura 3.10 o projeto da montagem dos módulos central, dianteiro direito e traseiro direito, sendo que os módulos dianteiro esquerdo e traseiro esquerdo são idênticos aos respetivos módulos direitos existindo apenas espelho nos faróis e por isso não se encontram no projeto. Nesta figura as cores dos condutores representam a tensão que conduzem sendo que o condutor laranja conduz 12 V, o condutor verde conduz 5 V, o condutor azul conduz 3,3 V e o condutor castanho é usado para representar a massa, os condutores com faixas brancas conduzem sinais com a tensão respetiva às suas cores, à exceção do condutor castanho com faixas brancas que foi usado nas ligações ao ecrã táctil para facilitar a compreensão.

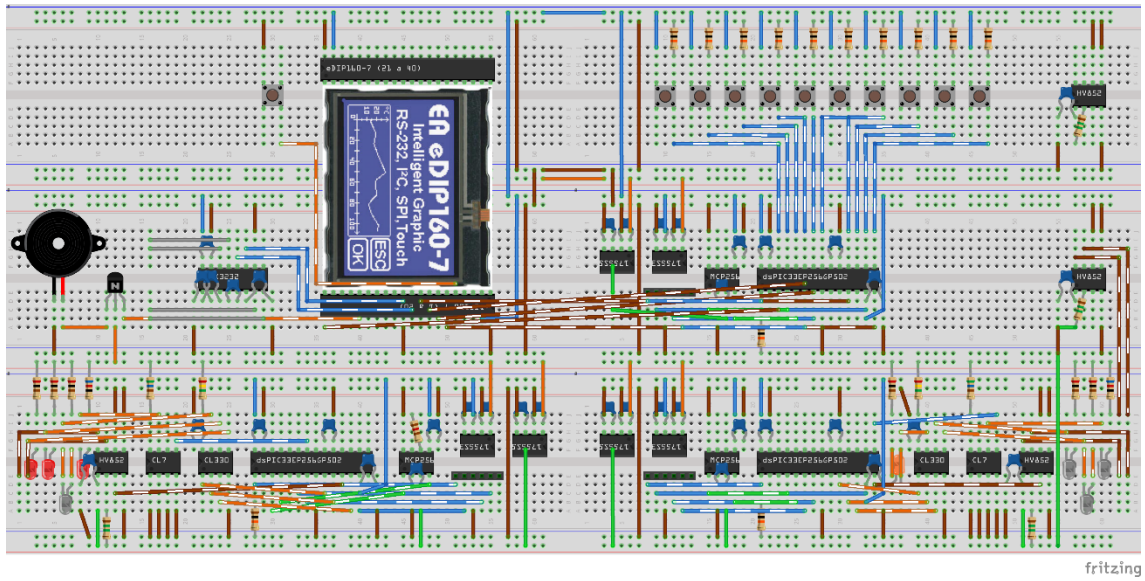


Figura 3.10 – Projeto da montagem no Fritzing.

A figura 3.11 é a vista geral da aplicação desenvolvida, onde se evidencia o módulo central montado nas quatro *breadboards* centrais e os outros quatro módulos distintos cada um na sua *breadboard*.

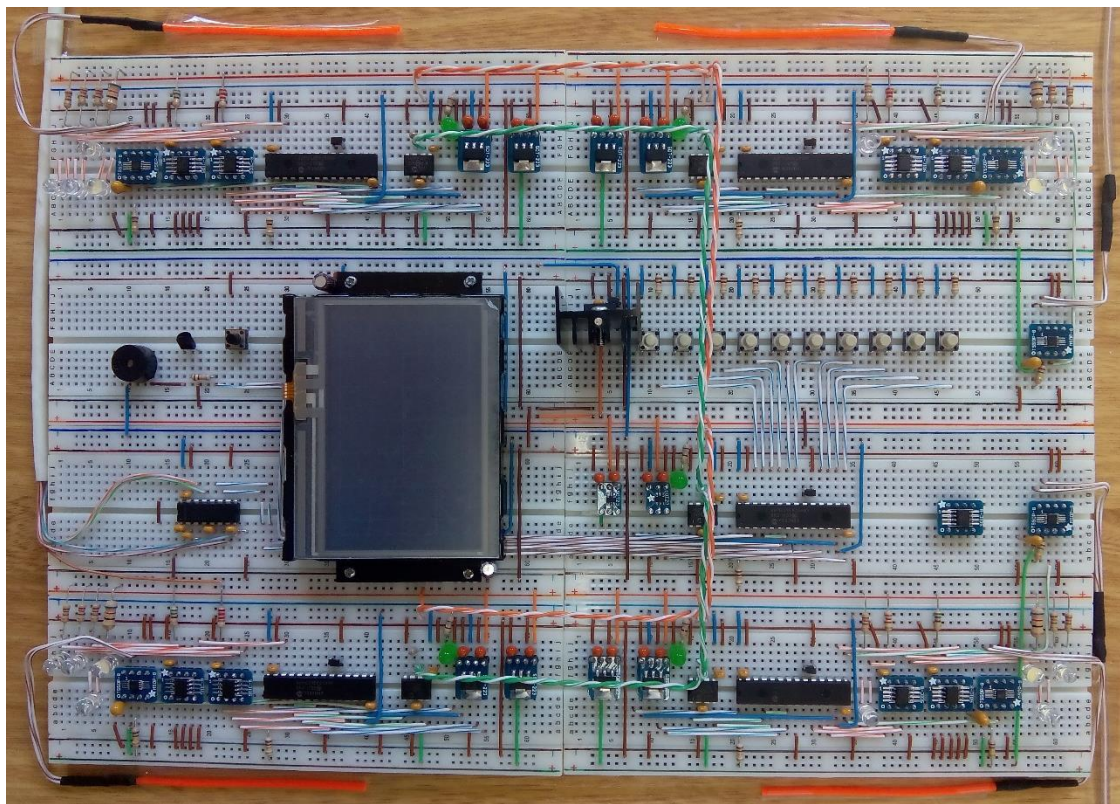


Figura 3.11 – Fotografia da aplicação desenvolvida.

Os dois pares de condutores entre módulos que foram referidos na secção 3.2 destacam-se na montagem estando estes mais elevados face aos restantes, pretende-se com isto representar que os dois pares de condutores que neste modelo do automóvel têm poucos centímetros poderiam ter vários metros num automóvel real, sendo o par laranja para a alimentação e o par verde para a comunicação entre módulos da rede CAN. Chama-se a atenção para o facto de cada módulo possuir um LED verde que sinaliza a presença dos 3,3 V no barramento do módulo.

3.4 Programação dos dispositivos

Para a aplicação desenvolvida foi necessário proceder à programação dos microcontroladores dsPIC33EP256GP502 e do ecrã táctil eDIP160B-7LWTP. Os *softwares* de programação utilizados foram os fornecidos pelos respetivos fabricantes, sendo que para os microcontroladores foi usado o *MPLAB X IDE* da *Microchip* e para o ecrã táctil foi utilizado o *KitEditor* da *Electronic Assembly*. As linhas de código completas dos programas encontram-se no suplemento digital da presente dissertação, passando-se seguidamente à explicação das partes mais relevantes.

3.4.1 Microcontroladores dsPIC33EP256GP502

Como já foi referido, os controladores dsPIC33EP256GP502 foram programados com o *software* MPLAB X IDE, mais especificamente em linguagem C graças a possibilidade de instalação do compilador XC16 para microcontroladores a 16 bits bem como um conjunto de bibliotecas para configuração dos vários periféricos. Houve também a preocupação de desenvolver as linhas de código com recurso a funções em alternativa ao endereçamento direto das variáveis a utilizar, com isto pretendeu-se dar uma maior portabilidade entre dispositivos podendo este código ser utilizado como referência para trabalhos futuros. As bibliotecas utilizadas foram as seguintes:

- Biblioteca XC.h: é uma biblioteca genérica do XC16 que vai incluir a biblioteca do microcontrolador definido no projeto, neste caso a “p33EP256GP502.h”;
- Biblioteca PPS.h: é uma biblioteca que permite efetuar a seleção de pinos associados aos periféricos (ex. rede CAN, protocolo SPI, etc.) através de um conjunto de funções;

- Biblioteca DMA.h: é uma biblioteca que possibilita a configuração dos parâmetros do módulo de acesso direto à memória usando funções;
- Biblioteca ECAN.h: é uma biblioteca que fornece funções para inicializar, configurar e controlar o módulo CAN;
- Biblioteca LIBPIC30.h: é uma biblioteca com funções de uso geral, tal como o uso de temporizadores, contadores, etc.

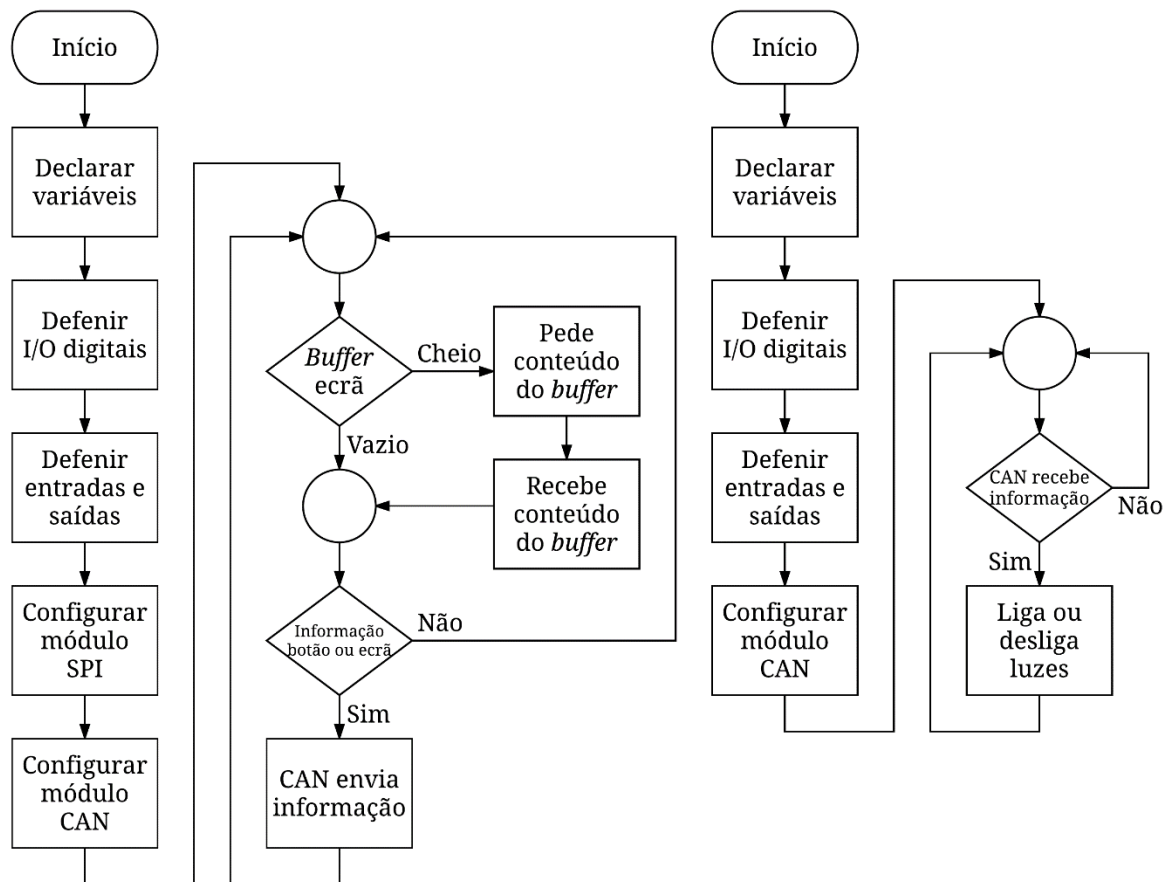


Figura 3.12 – Fluxogramas geral dos microcontroladores, à esquerda do módulo central e à direita dos restantes módulos.

Na figura 3.12 vemos as diferenças entre o módulo central que envia mensagens CAN e um dos restantes módulos que recebe mensagens CAN.

Criou-se um único programa uma vez que o código é semelhante para todos os módulos mas para definir de forma específica o código para cada módulo, foi utilizado o sistema de definições, em que se define no cabeçalho do programa o módulo que vai ser programado e assim habilita-se de forma automática o código específico para esse módulo.

Código específico é por exemplo o teste dos *buffers*, só é incluído o código correspondente aos *buffers* a testar por esse módulo.

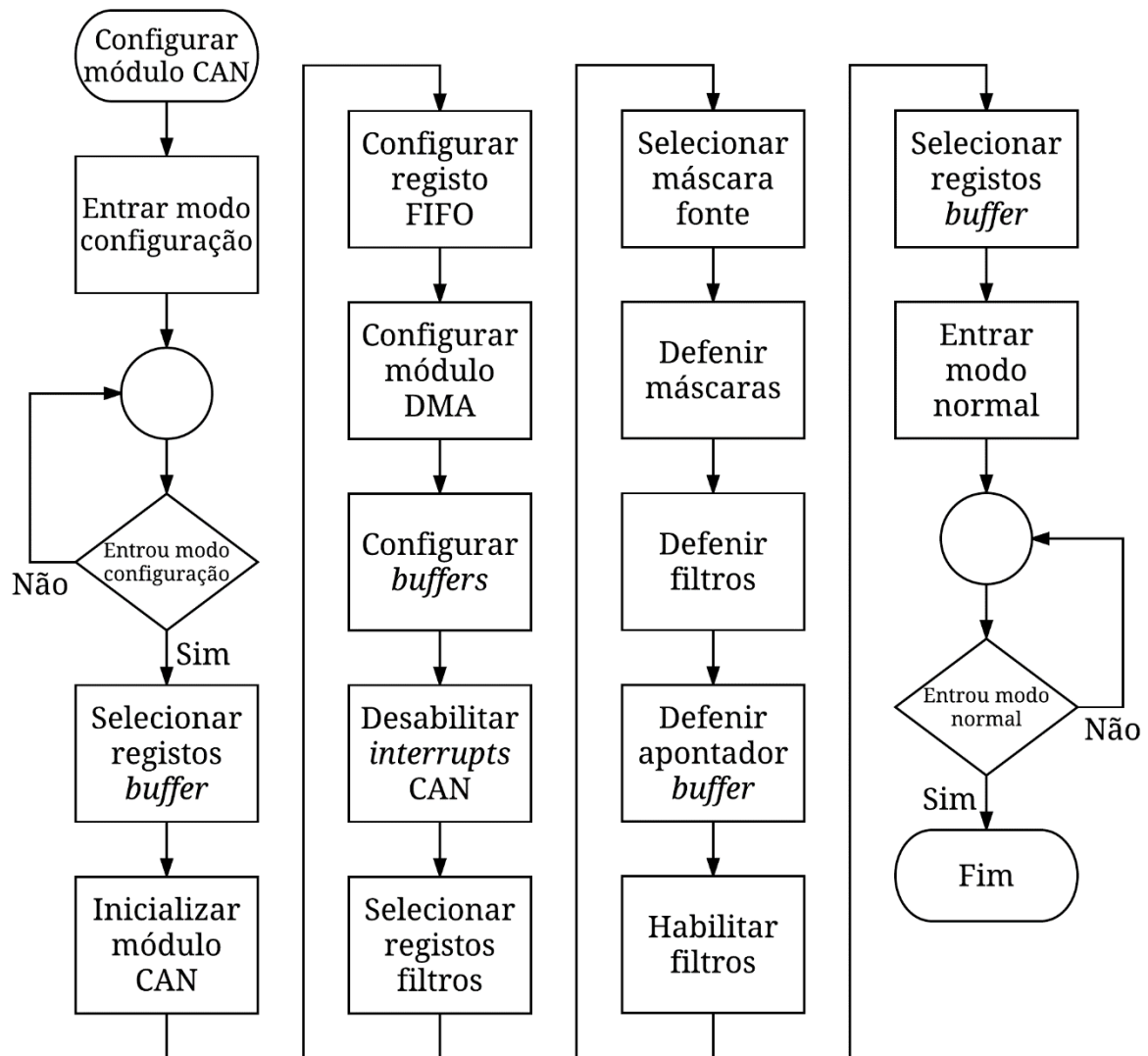


Figura 3.13 – Fluxograma da configuração do módulo CAN.

Atribuição dos pinos da rede CAN

O primeiro passo da programação consiste em selecionar os pinos que vão receber e transmitir as mensagens da rede CAN com recurso às funções da biblioteca PPS.h. Antes de selecionar os pinos é necessário primeiramente desbloquear essa funcionalidade que está protegida, de seguida atribuíram-se os pinos desejados, neste caso o pino RPI33 ao recetor e o pino RP35 ao transmissor do módulo da rede CAN, para concluir bloqueia-se a funcionalidade, tal como se ilustra na figura seguinte.

```

PPSUnlock;
PPSInput (IN_FN_PPS_C1RX, IN_PIN_PPS_RPI33);
PPSOutput (OUT_FN_PPS_C1TX, OUT_PIN_PPS_RP35);
PPSLock;

```

Figura 3.14 – Código ilustrativo da seleção de pinos.

Inicialização do módulo rede CAN

Conforme se mostrou no fluxograma da figura 3.13, o próximo passo consiste em colocar o módulo da rede CAN no modo de configuração e aguarda a confirmação que entrou no mesmo, depois é necessário garantir que os registos SFRs (*Special Function Registers*) de *buffer* estão seleccionados e finalmente inicializamos o módulo de rede CAN, atribuindo valores de TQ (*Time Quantas*) para cada fase da amostragem de forma a obter a velocidade de comunicação desejada. Uma unidade TQ equivale ao tempo do período do oscilador multiplicado por dois e pelo multiplicador de escala [29]. Com recurso ao *software* “CAN Bit Time Calculator” foram escolhidos 10 TQs para obter a velocidade de 184,250 kbit/s, não alcançamos velocidades superiores porque estamos limitados pela frequência do oscilador interno. Este procedimento encontra-se ilustrado na figura seguinte.

```

CAN1Initialize (CAN_SYNC_JUMP_WIDTH1 &
                CAN_BAUD_PRE_SCALE (1),
                CAN_WAKEUP_BY_FILTER_DIS &
                CAN_PHASE_SEG2_TQ (3) &
                CAN_PHASE_SEG1_TQ (5) &
                CAN_PROPAGATIONTIME_SEG_TQ (1) &
                CAN_SEG2_FREE_PROG &
                CAN_SAMPLE3TIMES);

```

Figura 3.15 – Código ilustrativo da inicialização da rede CAN.

Na figura 3.16 é possível ver a representação dos tempos de amostragem que perfazem um total de 5,427 μ s por bit. O tempo de sincronismo é sempre 1 TQ, foi definido 1 TQ para o tempo de propagação, para o tempo da fase 1 e o tempo da fase 2 foram definidos 5 TQs e 3 TQs respetivamente, totalizando os 10 TQs. Sendo que a amostragem é feita entre o tempo da fase 1 e o tempo da fase 2, servindo esses tempos para compensar erros que possam ocorrer devido a desfasamentos nos flancos, podendo esses tempos serem ajustados pelo mecanismo de ressincronização até ao máximo determinado no parâmetro *Synchronization Jump Width*.

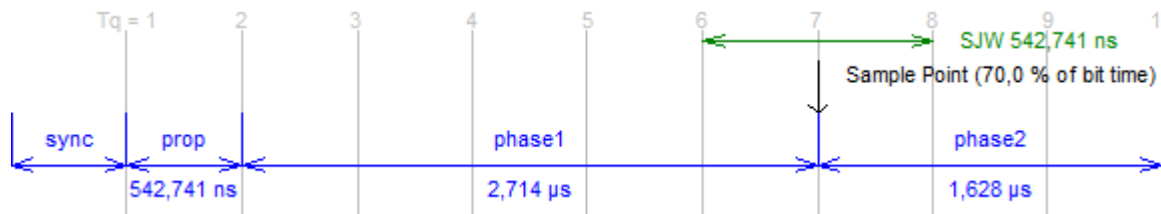


Figura 3.16 – Tempo de bit da comunicação CAN.

Configuração do módulo DMA

O passo seguinte consiste em configurar o módulo DMA (*Direct Memory Access*) que aloca espaço na RAM para os 16 *buffers* e define os canais de acesso, sendo configurado que o canal 0 é atribuído à transmissão e o canal 1 é atribuído à recepção, sendo que ambos vão encaminhar mensagens da rede CAN com 7 *words*. Nesta fase é necessário definir-se cada um dos sete *buffers* onde se vão guardar as mensagens recebidas ou as mensagens a transmitir e atribuir-lhe uma prioridade, nesta fase optamos por escolher que os *buffers* 1 e 2 vão guardar mensagens para transmitir e os restantes *buffers* vão guardar mensagens recebidas.

Desativar mascaras da rede CAN

A próxima etapa da programação consiste em desativar as máscaras da rede. As mascaras que dão a possibilidade de cada filtro aceitar mais do que um identificador de mensagem ignorando bits. Neste caso não foram utilizadas uma vez que os 16 filtros existentes são suficientes uma vez que serão atribuídos 13 identificadores apenas.

```
CAN1SetMask(0, CAN_MASK_SID(0x7FF) & CAN_MATCH_FILTER_TYPE, CAN_MASK_EID(0));
CAN1SetMask(1, CAN_MASK_SID(0x7FF) & CAN_MATCH_FILTER_TYPE, CAN_MASK_EID(0));
CAN1SetMask(2, CAN_MASK_SID(0x7FF) & CAN_MATCH_FILTER_TYPE, CAN_MASK_EID(0));
```

Figura 3.17 – Código ilustrativo da definição de mascaras.

Configuração dos filtros da rede CAN

Finalmente a última etapa consiste em configurar os filtros da rede CAN de modo a ativar os respetivos identificadores das mensagens e associa-los a um *buffer* apontando assim as diferentes mensagens para diferentes *buffers* de forma a ser mais fácil analisá-las. Limpando a variável C1FEN1 são desabilitados todos os filtros para de seguida se ativarem apenas os necessários. Na figura seguinte ilustra-se a parte das instruções relativas à configuração dos filtros.

```
C1FEN1 = 0;

CAN1SetFilter(0, CAN_FILTER_SID(0x000) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(1, CAN_FILTER_SID(0x010) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(2, CAN_FILTER_SID(0x020) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(3, CAN_FILTER_SID(0x030) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(4, CAN_FILTER_SID(0x040) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(5, CAN_FILTER_SID(0x050) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(6, CAN_FILTER_SID(0x060) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(7, CAN_FILTER_SID(0x070) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(8, CAN_FILTER_SID(0x080) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(9, CAN_FILTER_SID(0x090) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(10, CAN_FILTER_SID(0x0A0) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(11, CAN_FILTER_SID(0x0B0) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
CAN1SetFilter(12, CAN_FILTER_SID(0x0C0) & CAN_RX_EID_DIS, CAN_FILTER_EID(0));
```

Figura 3.18 – Código ilustrativo da definição de filtros.

Os identificadores foram atribuídos por ordem de importância do destinatário uma vez que terão prioridade de transmissão as mensagens com identificadores mais baixos. Recorde-se que esta é uma característica intrínseca desta rede. Foram deixados identificadores de reserva entre os vários identificadores atribuídos de forma a permitir a implementação de trabalhos futuros. A tabela 3 mostra o identificador de cada destinatário bem como a mensagem a enviar para o endereçar.

Tabela 3 – Atribuição dos identificadores das mensagens.

Destinatário	Identificador	Mensagem a enviar
Falha	0x000	0x0000
OK	0x010	0x0040
STOP	0x020	0x0080
4 Piscas	0x030	0x00C0
Pisca Esquerdo	0x040	0x0100
Pisca Direito	0x050	0x0140
Máximos	0x060	0x0180
Médios	0x070	0x01C0
Mínimos	0x080	0x0200
Marcha a Ré	0x090	0x0240
Nevoeiro T.	0x0A0	0x0280
Nevoeiro F.	0x0B0	0x02C0
Diurna	0x0C0	0x0300

Dependendo do módulo, este vai filtrar as mensagens recebidas e apenas aceita aquelas que tiverem um identificador correspondente a um filtro ativo. Por exemplo o módulo dianteiro direito filtra as mensagens e aceita as correspondentes aos 4 piscas, pisca direito, máximos, médios, mínimos, nevoeiro frente e diurna.

Após todos estes passos, o módulo da rede CAN está finalmente completamente configurado, é apenas necessário pedir para o módulo passar para o modo de

funcionamento normal e aguarda a confirmação que entrou no mesmo de forma a poder começar a comunicar. Para isso basta executar as seguintes instruções.

```
C1CTRL1bits.REQOP = 0;  
while (C1CTRL1bits.OPMODE != 0);
```

Figura 3.19 – Código ilustrativo da passagem para modo de funciona normal.

Transmissão de mensagens pelo módulo central

A cada ciclo, o módulo central começa por testar a entrada que sinaliza a presença de dados no *buffer* do ecrã tátil, caso a entrada esteja ativa este envia a mensagem SPI a pedir o conteúdo do buffer e guarda os dados uteis na variável Touch depois de verificar o bcc. Será transmitida uma mensagem com um pedido aos módulos descentralizados sempre que seja carregado um botão do ecrã tátil ou um botão de pressão, é testado qual o botão carregado e de seguida é testado o estado anterior da luz correspondente de forma a enviar o pedido que altera esse estado, para alguns botões são ainda testados outros estados para permitir o envio. Por exemplo, foi carregado no botão correspondente ao nevoeiro traseiro, o estado anterior do nevoeiro traseiro era desligado, é testado e o estado dos mínimos é ligado pelo que se envia o pedido para ligar o nevoeiro traseiro.

Receção de mensagens pelos módulos descentralizados

Os módulos descentralizados esperam que seja assinalada a receção de uma mensagem pela bandeira de *buffer* de receção cheio. Quando esta mensagem é sinalizada, vamos testar a variável FILHIT que guarda a informação do filtro que ativou a bandeira. Sabendo qual foi o filtro, sabemos em que *buffer* foi colocada a mensagem e passamos a testar a mensagem desse buffer para interpretar o pedido do módulo central. Depois é realizada a ação pedida e enviada a confirmação de ação sucedida para o módulo central. Por exemplo, chegou uma mensagem que sinalizou a bandeira, tinha o identificador 0x030 por isso a bandeira foi ativada pelo filtro 3 logo a mensagem foi guardada no *buffer* 3, foi testada a terceira *word* desse buffer que tinha o valor 1, isto correspondeu a ligar os 4 piscas.

3.4.2 Ecrã táctil eDIP160B-7LWTP

Para além do *software* de programação KitEditor, foram utilizados outros programas complementares da Electronic Assembly, tais como o eDIP Terminal e o BitmapEdit. O eDIP Terminal permite comunicar com o ecrã táctil através da porta de comunicação emulada pelo adaptador de RS-232 para USB, servindo para testar a comunicação e fazer configurações simples. O BitmapEdit serve para trabalhar as imagens a mostrar no ecrã, ajustando o tamanho das imagens e convertendo-as para preto e branco uma vez que o ecrã é monocromático.

O programa do ecrã táctil começa por definir o modelo utilizado e o nome do projeto, em seguida diz que vai comunicar através da porta de comunicação 3 com o ritmo de transmissão de dados 115200 e pede para verificar o programa depois de transferir.

```
eDIP160-7 "Display Rede CAN"  
AutoScan: 1  
COM3: 115200  
VERIFY ;
```

Figura 3.20 – Código ilustrativo do início do programa.

O ecrã táctil é programado com base em macros. A macro MnAutoStart é executada tanto quando o ecrã é ligado como quando lhe é feito *reset*, esta macro começa por desligar a funcionalidade “Terminal” e definir o brilho de 90%, depois roda a orientação 180° e de seguida configura os botões tácteis e cria um botão em todo o ecrã que manda executar a macro de toque 200, depois é colocado o logotipo do ISEL à esquerda e o nome à direita com as respetivas configurações.

Macro: MnAutoStart

```
#TA  
  
#YH 90  
  
#DO 2  
  
#AE 0  
#AR 0  
#AI 0  
  
#AT 0, 0, 160, 104, 200, 0, ""  
  
#UI 0, 0, 6  
  
#ZF 0  
#ZZ 2, 2  
#ZY 0  
#ZC 91, 6, "ISEL"  
  
#ZF 1  
#ZZ 1, 1  
#ZY 2  
#ZC 92, 24, "Instituto Superior|de Engenharia de|Lisboa"  
  
#ZF 1  
#ZZ 1, 1  
#ZY 1  
#ZC 90, 52, "(Toque para|continuar)"
```

Figura 3.21 – Código ilustrativo da macro MnAutoStart.

A figura 3.22 é a aparência do ecrã depois de ser executada a macro MnAutoStart, tocando no ecrã continuamos para a macro de toque 200 que mostra a informação do trabalho e aguarda outro toque no ecrã.



Figura 3.22 – Aparência da página inicial do ecrã.

Esse toque no ecrã chama a macro de toque 2 que começa por apagar o botão que a chamou e de seguida apaga o ecrã para meter os botões depois de configurar a sua margem, o tipo de letra e animação ao toque. Foram usados caracteres especiais definidos no tipo de letra presente na memória interna do ecrã para representar setas.

```
TouchMacro: 2

#AL 0, 0 ; Delete touch area

#DL ; Delete display

#AE 1 ; Touch frame
#AF 0 ; Label font
#AI 1 ; Touch key response

#AZ 2, 1 ; Label zoom factor
#AT 0, 0, 52, 33, 11, 0, $04 ; Define touch switch

#AZ 1, 1 ; Label zoom factor
#AT 54, 0, 105, 33, 12, 0, "" ; Define touch switch

#UI 69, 9, 7 ; Load internal image

#AZ 2, 1 ; Label zoom factor
#AT 107, 0, 159, 33, 13, 0, $03 ; Define touch switch

#AZ 1, 1 ; Label zoom factor
#AT 0, 35, 79, 68, 14, 0, "STOP" ; Define touch switch

#AZ 1, 1 ; Label zoom factor
#AT 81, 35, 159, 68, 15, 0, "CReverse" ; Define touch switch

#AZ 2, 2 ; Label zoom factor
#AT 54, 70, 105, 103, 4, 0, $05 ; Define touch key

#AZ 2, 2 ; Label zoom factor
#AT 107, 70, 159, 103, 6, 0, $1E ; Define touch key
```

Figura 3.23 – Código ilustrativo da macro de toque 2.

Na primeira tela de controlo da aplicação desenvolvida é possível ligar e desligar cada uma das luzes, na primeira linha controlam-se os piscas esquerdo, direito e quatro piscas, na segunda linha controlam-se as luzes de stop e de marcha a trás, na terceira linha está o botão de passagem para a segunda tela de controlo, tendo a aparência que vemos na figura 3.24.



Figura 3.24 – Aparência da 1ª página de botões do ecrã.

Na figura 3.25 vemos a segunda página de botões que apresenta em cima os botões dos mínimos, dos médios e dos máximos, respetivamente da esquerda para a direita, na linha do meio à esquerda está o botão das luzes de nevoeiro dianteiras e à direita está o botão luzes de nevoeiro traseiras, em baixo está o botão que permite voltar a primeira página de controlo das luzes.

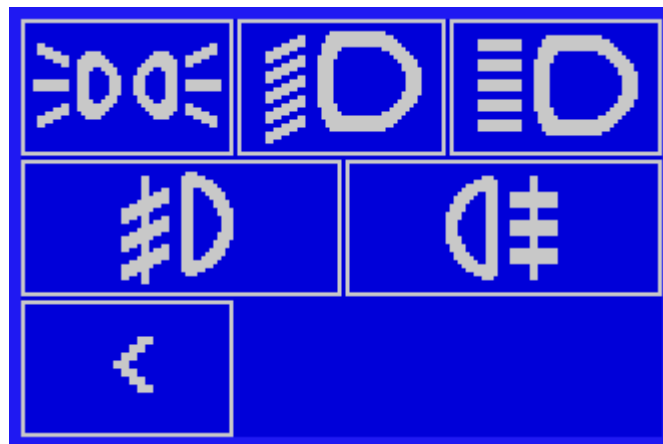


Figura 3.25 – Aparência da 2ª página de botões do ecrã.

4. Resultados Experimentais

Apresenta-se neste capítulo alguns resultados experimentais obtidos com o osciloscópio mas o resultado experimental fundamental a retirar da aplicação desenvolvida é o bom funcionamento de todo o sistema de iluminação automóvel, simulando o uso normal ou mesmo intensivo do sistema, permitindo demonstrar a viabilidade da solução proposta nesta dissertação. Sendo possível verificar o funcionamento da rede CAN e dos seus periféricos associados nas figuras seguintes.

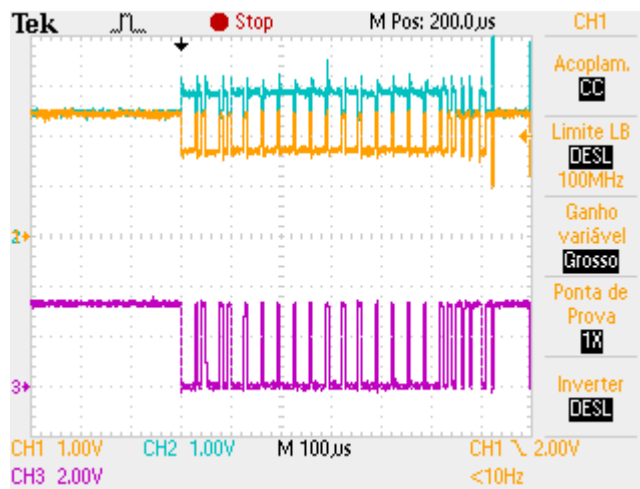


Figura 4.1 – Sinais à entrada (em baixo) e à saída (em cima) do transceptor.

Na figura 4.1 vemos uma trama completa transmitida na rede CAN. Em baixo observamos o sinal à entrada do transceptor, ou seja, à saída do controlador CAN integrado no microcontrolador, em que os 3,6 V correspondem ao nível lógico "1" e os 0 V correspondem ao nível lógico "0". Em cima observamos o sinal à saída do transceptor, ou seja, a tensão em modo diferencial no barramento CAN, em que o nível recessivo corresponde ao sinal não diferencial e o nível dominante corresponde ao sinal diferencial. Analisando a figura podemos garantir o correto funcionamento do transceptor, tendo em conta que faz a correta conversão dos níveis lógicos entre o barramento e o controlador CAN e vice-versa, i.e., quando o sinal à entrada tem um nível lógico "1", as tensões de saída têm o mesmo valor de tensão enquanto que, quando o sinal à entrada tem um nível lógico "0", as tensões de saída têm valores diferentes de tensão.

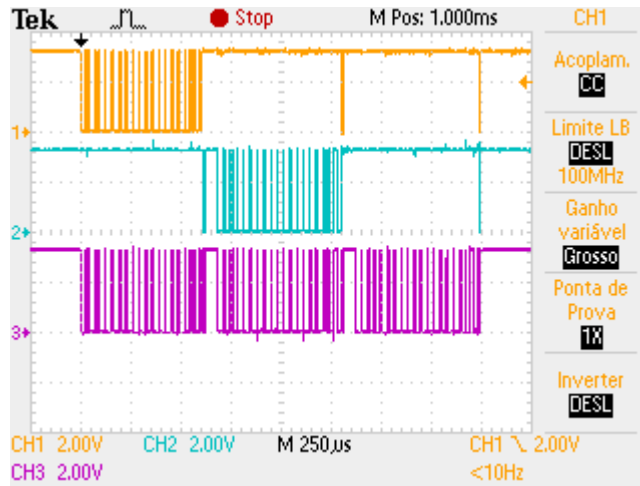


Figura 4.2 – Mensagem do módulo central e respostas

Na figura 4.2 observamos a roxo o sinal do barramento CAN à saída do transceptor, evidenciando-se três tramas distintas, sendo o sinal laranja a primeira trama que foi um pedido do módulo central para ligar os STOPS, a azul está a trama de resposta ao pedido por parte do módulo traseiro esquerdo com a confirmação que o STOP foi ligado, a terceira trama é a resposta por parte do módulo traseiro direito e depois o barramento fica livre. Tanto na laranja como na azul existe no final da mensagem o reconhecimento da recepção da mesma pelos outros módulos o que garante o bom funcionamento dos controladores CAN.

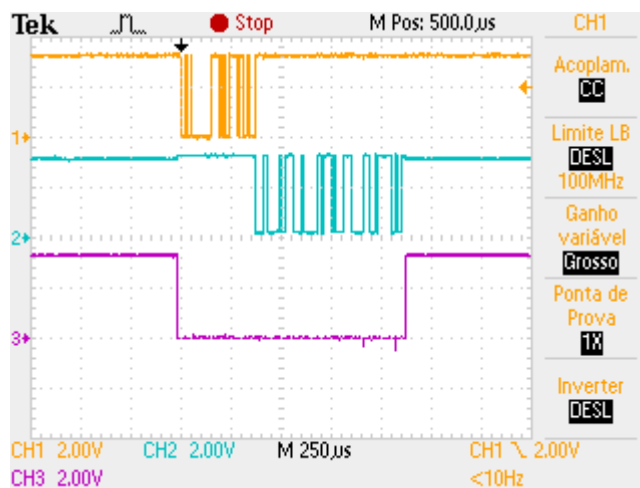


Figura 4.3 – Ciclo completo de mensagens SPI.

Na figura 4.3 vemos a roxo o sinal de *chip select* gerado pelo microcontrolador do módulo central que depois de ativado começa a enviar a mensagem representada a laranja que pede ao ecrã para enviar o conteúdo do *buffer* SPI. Este conteúdo é enviado de seguida como podemos ver no sinal azul e logo que recebe toda a mensagem o microcontrolador desativa o sinal de *chip select* completando assim o ciclo de mensagens SPI.

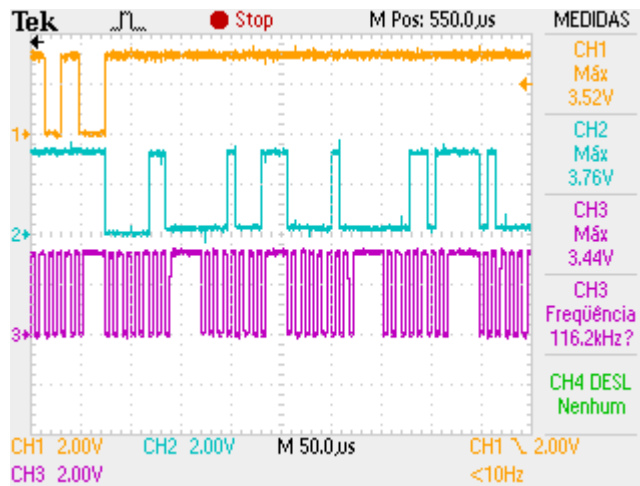


Figura 4.4 – Pormenor das mensagens SPI com a frequência do clock.

Na figura 4.4 observamos as mensagens SPI da figura 4.3 com pormenor no início da mensagem do ecrã com o conteúdo do *buffer* que respeita o *Smart Protocol* implementado no ecrã que especifica o formato das mensagens a receber e a enviar por este. Nas medidas podemos observar a frequência do *clock* de 116 kHz que se reflete num ritmo de transmissão de 116 kbit/s.

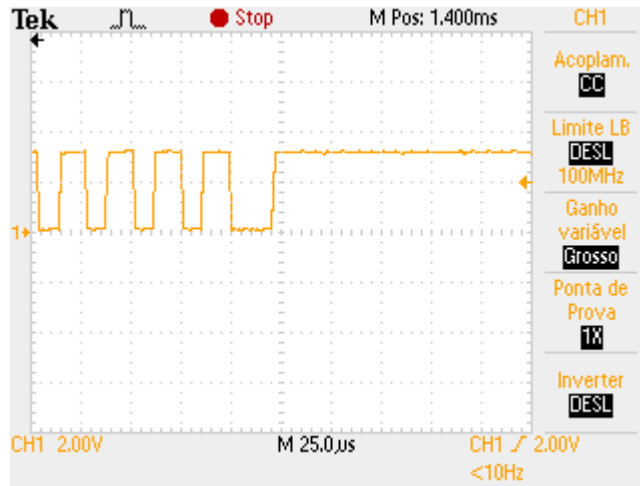


Figura 4.5 – Transição de médios para máximos.

Na figura 4.5 vemos o sinal gerado pelo microcontrolador que regula a intensidade do LED dos médios/ máximos para 60% quando são ligados os médios e quando são ligados os máximos oferece toda a sua intensidade.

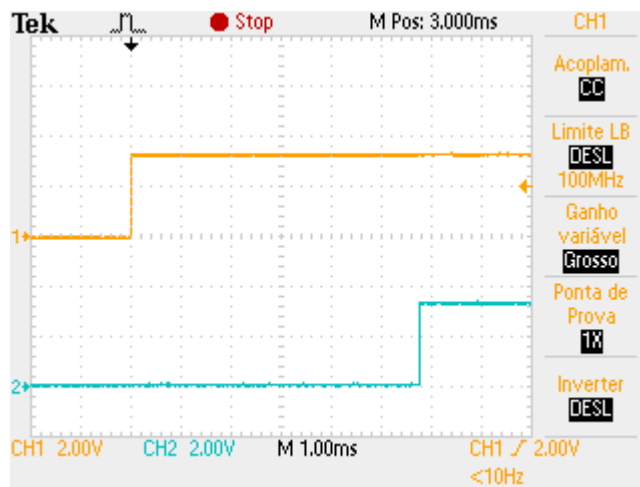


Figura 4.6 – Intervalo de tempo entre ação e atuação.

Na figura 4.6 observamos o intervalo de tempo entre o utilizador pressionar um botão de pressão e a sinalização correspondente ser ligada ou desligada. Estes 5.8 ms representam o tempo de aquisição, de processamento e de transmissão do sinal pela aplicação desenvolvida.

5. Conclusões

A percentagem de componentes eletrónicos nos automóveis está a crescer de ano para ano. A eletrónica desempenha um papel decisivo, não só no sentido de satisfazer os desejos dos clientes para uma melhor segurança e conforto na condução mas ao mesmo tempo para conseguir uma melhor economia de combustível e emissões de gases de escape reduzidas. Outro aspeto que não deve ser subestimado é a contribuição de sistemas de barramento série nos automóveis. Muitas funções nem sequer seriam possíveis sem a troca de dados entre os componentes eletrónicos.

Tendo em conta esta realidade, foram estudadas no estado da arte várias redes de comunicação utilizadas nos automóveis, começando pela rede CAN, passando pela rede LIN e pela rede FlexRay, e acabando na rede MOST. Nesse estudo foram apresentadas as suas principais características e modos de funcionamento a fim de cumprir o objetivo de perceber as vantagens e desvantagens das redes de comunicação utilizadas nos automóveis. Apresentou-se ainda no estado da arte as classes das redes dos automóveis que mostra de forma clara as aplicações a que se destinam cada tipo de rede. Para além das redes, fez-se uma abordagem histórica aos sistemas de iluminação nos automóveis com a evolução dos mesmos até aos dias de hoje, em que se evidenciam as novas tendências a este nível.

A solução de iluminação automóvel proposta nesta dissertação tem por base a rede CAN que se mostrou capaz de controlar com precisão e em tempo real todo o sistema. Contudo foram sentidas limitações ao nível da velocidade que ficou restrita a 184,250 kbit/s quando o objetivo seria atingir velocidade maiores uma vez que o *hardware* suporta até 1 Mbit/s, esta restrição deveu-se à baixa frequência do oscilador interno do microcontrolador. Todavia a velocidade revelou-se suficiente para a aplicação, não existindo atrasos na entrega das mensagens.

Uma vantagem foi o baixo custo de implementação desta rede quando comparado com outras, tanto ao nível do meio de transmissão como do transceptor e do controlador. Tendo em conta a realidade da indústria automóvel que tenta baixar ao máximo o custo de cada nó da rede, optou-se pela não colocação de oscilador externo uma vez que implicaria mais *hardware* e complexidade logo um maior custo por nó.

A utilização do fio eletroluminescente revelou ser um problema dado que não se encontrou fio com características compatíveis com o controlador, este problema deveu-se a falta de documentação técnica, inclusivamente por parte dos fornecedores, porque este produto é utilizado principalmente para fins recreativos. O resultado é a fraca intensidade luminosa do fio eletroluminescente apesar da correta implementação.

O tempo de desenvolvimento desta dissertação foi dividido praticamente de igual forma em três partes sendo elas, planeamento e projeto, implementação e programação da aplicação desenvolvida. A parte de planeamento e projeto foi muito desafiante dadas as muitas possibilidades, procurando sempre chegar ao melhor resultado. A implementação revelou-se a parte mais trabalhosa devido a minúcia e detalhe requeridos pela aplicação desenvolvida. A parte mais complexa foi a programação dos microcontroladores, principalmente os segmentos das comunicações. A programação do ecrã tátil foi intuitiva devido à facilidade dada pela inteligência implementada no mesmo.

Os resultados experimentais refletem o bom funcionamento da solução de iluminação automóvel baseada na rede CAN proposta nesta dissertação pelo que se cumpriram os objetivos propostos e se julga ter contribuído para mostrar como pode ser implementada uma rede CAN.

De salientar que apesar do bom funcionamento da aplicação desenvolvida, esta é protótipo e não deve ser utilizada em veículos reais uma vez que não está homologada para esse efeito, pelo que se rejeita a responsabilidade por qualquer dano que possa ocorrer devido à sua utilização.

5.1 Perspetivas de desenvolvimento futuro

Numa perspetiva de desenvolvimento futuro apresenta-se uma lista de trabalhos visando melhorar a solução proposta nesta dissertação ou abrir novos caminhos para o estudo de outras áreas relacionadas com esta temática, a saber:

- Expandir a aplicação desenvolvida com outros sensores que beneficiem com a descentralização dos módulos, por exemplo sensor de pressão dos pneus ou sensor de luminosidade;

- Montar as placas de circuito impresso (PCBs) que implementam a solução proposta nesta dissertação;
- Implementar a técnica de fio eletroluminescente com outro controlador ou outro fio para obter uma maior intensidade luminosa;
- Conceber uma interface entre a rede CAN, utilizada na aplicação desenvolvida, e outra rede utilizada em veículos automóveis.

Bibliografia

- [1] Charette, Robert N., This Car Runs on Code, IEEE Spectrum, 01/02/2009
- [2] Mayer, E., Serial Bus Systems in the Automobile – Architecture, tasks and advantages, Elektronik Automotive 07/2006, pp. 70
- [3] Palma, João, Introdução às Redes de Campo de Automação, 2004
- [4] Information technology – Open System Interconnection – Basic Reference Model: The Basic Model, International Standard ISO 7498-1, Issue 2, 1996-06-15
- [5] Lupini, Christopher A., In-Vehicle Networking Technology for 2010. SAE International, 2010
- [6] SAE International. Class A Application Definition (J2507-1)
- [7] CAN Specification Version 2.0, Robert Bosch GmbH, 09/1991
- [8] Unruh, J., Mathony, H.J., Kaiser, K.H.: Error Detection Analysis of Automotive Communication Protocols, SAE International Congress 1990
- [9] Mayer, E., Data communication in the automobile – Reliable data exchange with CAN in the automobile, Elektronik Automotive 08/2006, pp. 34-37
- [10] CAN FD Specification Version 1.0, Robert Bosch GmbH, 17/04/2012
- [11] LIN Specification Package Revision 2.2A, LIN Consortium, 31/12/2010
- [12] Make skidding a thing of the past, Robert Bosch GmbH, 09/2011
- [13] FlexRay Protocol Specification Version 3.0.1, FlexRay Consortium, 10/2010
- [14] Mayer, E., Serial Bus Systems in the Automobile – MOST for transmission of multimedia data, Elektronik Automotive 09/2007
- [15] MOST Specification Revision 3.0 E2, MOST Cooperation, 07/2010

- [16] Martins, Eduardo R., Comparação entre os desempenhos de faróis automotivos convencionais e aqueles que empregam díodos emissores de luz, Dissertação de Mestrado, Universidade de São Paulo, Brasil, 2005
- [17] Osram unveils product strategy towards serial production, OSRAM GmbH., Trade press 02/09/2013
- [18] dsPIC33EPXXXGP50X, dsPIC33EPXXXMC20X/50X and PIC24EPXXXGP/MC20X – 16-Bit Microcontrollers and Digital Signal Controllers with High-Speed PWM, Op Amps and Advanced Analog, Microchip Technology Inc., 29/11/2012
- [19] Gübel, Caio, AN916 – Comparing CAN and ECAN Modules, Microchip Technology Inc., 17/02/2004
- [20] MCP2561/2 – High-Speed CAN Transceiver, Microchip Technology Inc., 25/03/2014
- [21] CL330 – 3-Channel 30mA Linear LED Driver, Supertex Inc., 2008
- [22] CL7 – Linear, 100mA, Constant Current LED Driver with Enable Input, Supertex Inc., 2014
- [23] HV852 – High Voltage, Low Noise, Inductorless EL Lamp Driver, Supertex Inc., 2009
- [24] MCP1755/1755S – 300 mA, 16V, High-Performance LDO, Microchip Technology Inc., 29/11/2012
- [25] MCP1703A – 250 mA, 16V, Low Quiescent Current LDO Regulator, Microchip Technology Inc., 20/08/2013
- [26] MIC29150/29300/29500/29750 – High-Current Low-Dropout Regulators, Micrel Inc., 12/2012
- [27] EA eDIP160B-7LWTP – 160x104 with touch panel, ELECTRONIC ASSEMBLY, 01/2014
- [28] <http://fritzing.org/home/>

[29] Richards, Pat, AN754 – Understanding Microchip’s CAN Module Bit Timing, Microchip Technology Inc., 06/01/2001

[30] Road vehicles – Diagnostic systems – Part 1: Diagnostic services, International Standard ISO 14229-1.6, Issue 6, 22/02/2001

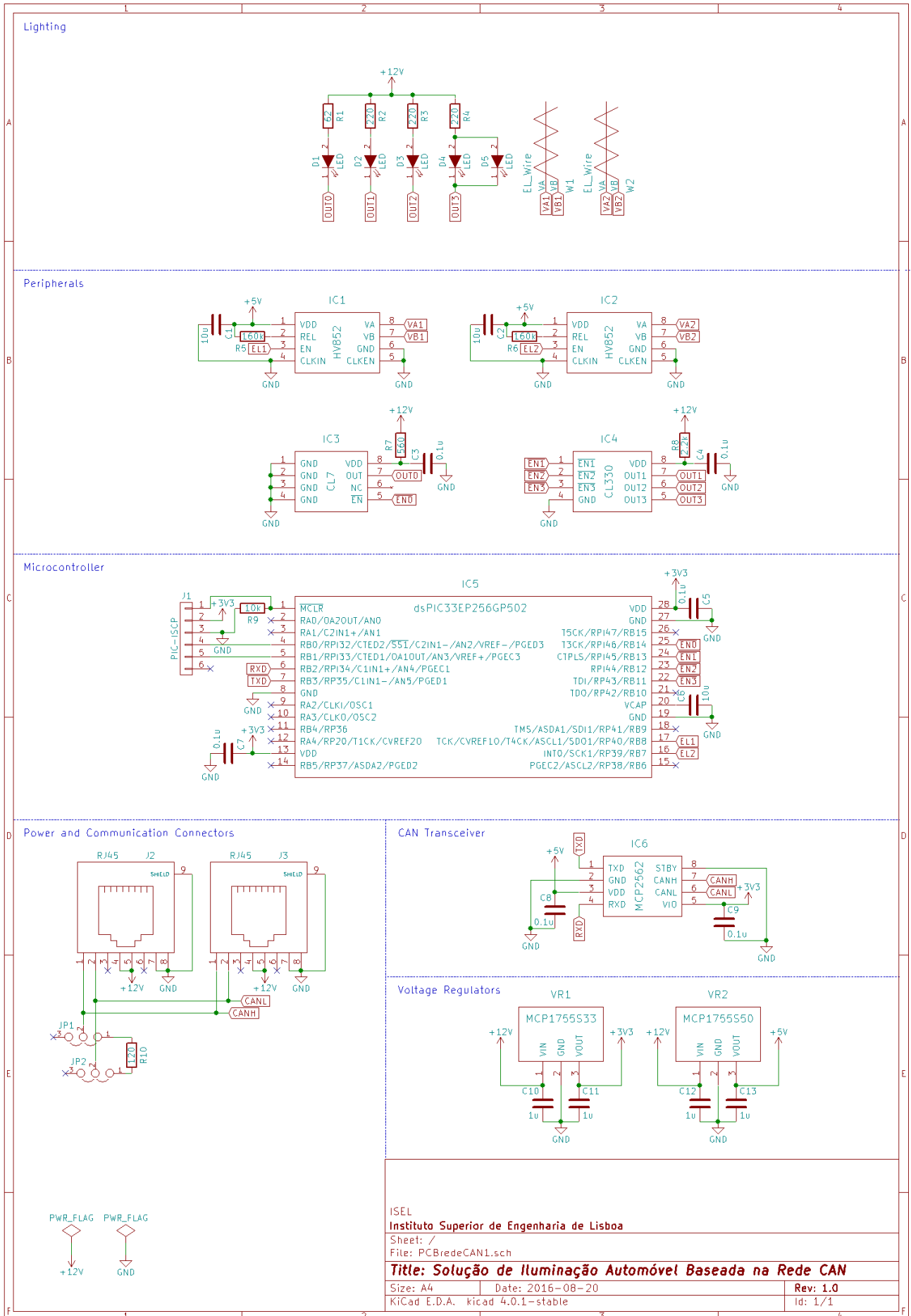
[31] Road vehicles – Diagnostics on Controller Area Network (CAN) – Part 2: Network layer services, International Standard ISO 15765-2.4, Issue 4, 21/06/2002

[32] Road vehicles – Diagnostics on Controller Area Network (CAN) – Part 3: Implementation of diagnostic services, International Standard ISO 15765-3.5, Issue 5, 12/12/2002

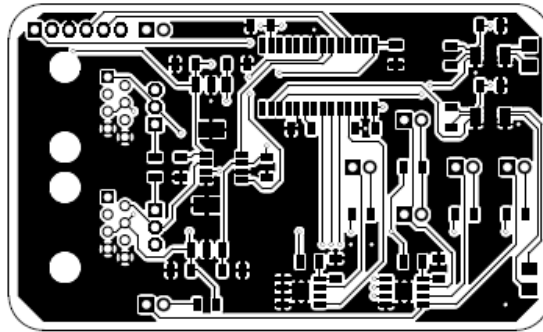
Anexo

Projeto das Placas de Circuito Impresso

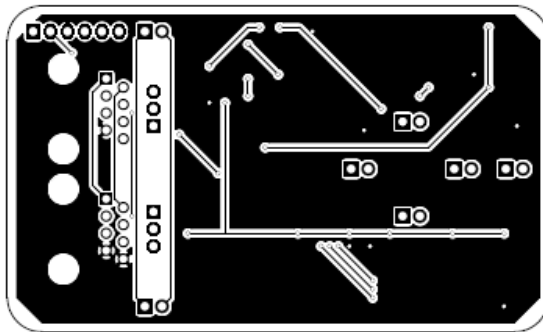
Solução de Iluminação Automóvel Baseada na Rede CAN



Front Copper Layer



Back Copper Layer



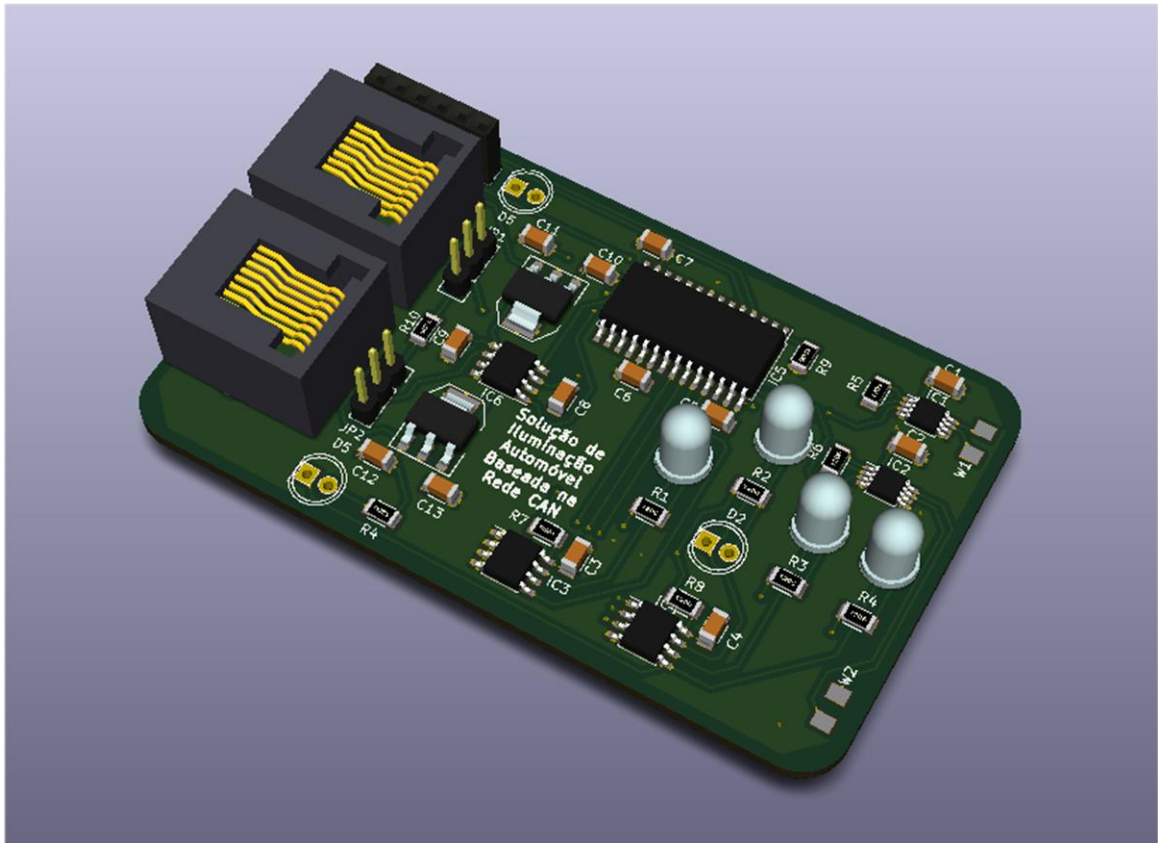
ISEL
Instituto Superior de Engenharia de Lisboa
Sheet:
File: PCBredeCAN1.kicad_pcb

Title: Solução de Iluminação Automóvel Baseada na Rede CAN

Size: A4 Date: 2016-08-20
KICad E.D.A. kicad 4.0.1-stable

Rev: 1.0
Id: 1/1

Vista frontal modelada em 3D:



Vista traseira modelada em 3D:

