



Projeto e desenvolvimento de uma unidade de controlo eletrónico para aplicação em veículos elétricos

Mestrado em Engenharia Automóvel

Alexandre Filipe Brás Francisco

Leiria, Abril de 2020



Projeto e desenvolvimento de uma unidade de controlo eletrónico para aplicação em veículos elétricos

Mestrado em Engenharia Automóvel

Alexandre Filipe Brás Francisco

Trabalho de Projeto realizado sob a orientação do Professor Doutor Carlos Daniel Henriques Ferreira, do Professor Doutor Luís Manuel Conde Bento e do Professor Doutor Nuno Manuel Lucas Vieira Lopes

Leiria, Abril de 2020

Originalidade e Direitos de Autor

O presente relatório de projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Curso de Mestrado em Engenharia Automóvel, no ano letivo 2019/2020, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

Agradecimentos

Aos meus orientadores Professor Doutor Carlos Daniel Henriques Ferreira, Professor Doutor Luís Manuel Conde Bento e Professor Doutor Nuno Manuel Lucas Vieira Lopes, que me acompanham desde o início do meu percurso académico, pelo tempo e disponibilidade despendida, pelas sugestões efetuadas e por todo o encorajamento que sempre me transmitiram.

Aos meus pais, António Francisco e Filomena Francisco, pelo incondicional apoio prestado em toda a minha vida e em especial nesta fase que exigiu grande esforço da nossa parte. Também aos meus irmãos e família pela motivação e apoio que sempre me deram.

À minha namorada, Mariana Vaz, pela compreensão em momentos em que o tempo era limitado, bem como pelas palavras de ânimo e incentivo transmitidas nesta fase.

Aos meus colegas que iniciaram esta etapa comigo, pelo apoio e partilha de conhecimentos que me deram ao longo do meu percurso académico.

Aos técnicos responsáveis pelo Laboratório de Engenharia Automóvel, Engenheiro Diogo Costa e Engenheiro Nuno Pires, pela ajuda e disponibilidade que sempre providenciaram.

Ao Sr. Trevor Lees da empresa Lynch, pelo apoio e informação técnica que me facultou.

A todos os funcionários da Tec-Atlântica de Torres Vedras, empresa onde exerci funções durante uma fase desta etapa, por toda a compreensão e ajuda que me deram para que fosse possível alcançar este objetivo.

À Escola Superior de Tecnologia e Gestão do Politécnico de Leiria, pelos meios que sempre me disponibilizou e que fizeram com que fosse possível a realização deste projeto.

Resumo

No presente documento é apresentado o projeto e desenvolvimento de uma unidade de controlo eletrónico de baixo custo, conhecida como ECU (*Electronic Control Unit*), orientada para aplicação e controlo de veículos elétricos utilitários.

Através da análise do estado da arte foi possível entender a arquitetura deste tipo de veículos e observar as soluções de ECU's comercialmente disponíveis no mercado. Com a caracterização destas soluções foi possível definir os requisitos e funcionalidades necessárias para este tipo de ECU's. Foram também identificados alguns pontos de melhoria passíveis de serem realizados a um custo mais reduzido, comparativamente com as soluções existentes.

O estudo das tecnologias de suporte permitiu identificar os protocolos de comunicação relevantes, inerentes ao funcionamento de uma ECU. A análise de arquiteturas de código permitiu decidir o tipo de estratégia de código a adotar para a implementação do algoritmo de funcionamento da ECU desenvolvida.

No âmbito deste estudo foi realizado o projeto e execução do *hardware* necessário para o desenvolvimento da ECU aqui apresentada. É demonstrada a escolha de componentes e as funcionalidades que são disponibilizadas pela ECU proposta, tais como a existência de 3 barramentos de comunicação CAN, essenciais num veículo desta natureza.

Realizaram-se testes de validação de forma a comprovar o correto funcionamento da ECU desenvolvida, bem como para verificar todas as funcionalidades que constituem este módulo eletrónico.

Foi desenvolvido um simulador de um veículo elétrico que permitiu avaliar o desempenho da ECU quando aplicada num ambiente real. Também se verificou que, quando exposta a ruídos, interferências e vibrações, a ECU demonstrou ser robusta.

Palavras-chave: Unidade de controlo eletrónico, Veículo elétrico utilitário, CAN, Arduino, Código aberto

Esta página foi intencionalmente deixada em branco.

Abstract

The present study presents the design and development of a low cost electronic control unit, known as an ECU (Electronic Control Unit), aimed at the application and control of electric utility vehicles.

Through the analysis of the state of the art it was possible to understand the architecture of this type of vehicles and observe the solutions of commercially available ECU's in the market. With the characterization of these solutions it was possible to define the requirements and functionalities necessary for this type of ECU's. Some points of improvement that could be achieved at a lower cost were also identified, compared to existing solutions.

The study of support technologies allowed the identification of relevant communication protocols, inherent to the functioning of an ECU. The analysis of code architectures allowed to decide the type of code strategy to be adopted for the implementation of the developed ECU's working algorithm.

Within the scope of this study, the design and execution of the necessary hardware for the development of the ECU projected here was carried out. The choice of components and features that are offered by the proposed ECU is demonstrated, such as the existence of 3 CAN communication buses, essential in a vehicle of this nature.

Validation tests were conducted in order to ensure the correct functioning of the developed ECU, as well as to verify all the functionalities that constitute this electronic module.

An electric vehicle simulator was developed to assess the performance of the ECU when applied in a real environment. It was also found that, when exposed to noise, interference and vibrations, the ECU proved to be robust.

Keywords: Electronic control unit, Electric utility vehicle, CAN, Arduino, Open source

Índice

Originalidade e Direitos de Autor	iii
Agradecimentos	iv
Resumo	v
Abstract	vii
Lista de Figuras	x
Lista de Tabelas	xiii
Lista de Siglas	xiv
1. Introdução	1
1.1. Contexto e motivação	1
1.2. Objetivos	3
1.3. Estrutura do projeto	3
2. Estado da Arte	5
2.1. História do veículo elétrico	5
2.2. Arquitetura veículo elétrico	7
2.3. Unidades disponíveis para controlo de veículos elétricos	16
3. Tecnologias de Suporte	25
3.1. Protocolos de comunicação	25
3.1.1. Comunicação CAN	25
3.1.2. Comunicação I2C.....	34
3.1.3. Comunicação SPI.....	36
3.1.4. Comunicação Série.....	39
3.2. Arquitetura de código	41
3.2.1. Superloop.....	41

3.2.2.	Foreground/Background Systems	42
4.	MeaTec32.....	43
4.1.	Requisitos e especificações de uma ECU	43
4.2.	Módulos da MeaTec32	48
4.3.	Testes de componentes	62
4.4.	Desenvolvimento da PCB	64
4.5.	Produção e montagem da PCB	68
5.	Validação da MeaTec32	73
5.1.	Equipamentos e metodologia de testes.....	73
5.2.	Reguladores de tensão	74
5.3.	Portos I/O.....	75
5.4.	Comunicações	80
5.5.	Inspeção por termografia da PCB	84
5.6.	Simulador de veículo elétrico	88
5.7.	Procedimentos de teste	93
5.7.1.	Aquisição de sinais externos	94
5.7.2.	Comunicação com o controlador.....	94
5.7.3.	Comunicação com o painel de instrumentos.....	96
5.7.4.	Desenvolvimento de código	96
5.8.	Discussão de Resultados	99
6.	Conclusões e Trabalhos Futuros.....	101
	Referências Bibliográficas	103
	Anexos	106

Lista de Figuras

Figura 2.1 – Motor elétrico de corrente contínua por Thomas Davenport [1].	5
Figura 2.2 – Veículo elétrico “Electrobat” de 1894 [2].	6
Figura 2.3 – Arquitetura geral do veículo elétrico, adaptado de [1].	8
Figura 2.4 – Diagrama de motores elétricos utilizados em veículos elétricos, adaptado de [3].	9
Figura 2.5 – Função simplificada do controlador [5].	11
Figura 2.6 – Influência do arrefecimento no desempenho de um motor elétrico [6].	13
Figura 2.7 – Exemplo de sistema de monitorização de bateria, adaptado de [7].	14
Figura 2.8 – Diagrama de blocos de uma unidade de controlo eletrónico, adaptado de [10].	15
Figura 2.9 – ECU Vector VC121-12 [12].	16
Figura 2.10 – ECU Ecotrons EV2206B03 [13].	18
Figura 2.11 – ECU TTCControl HY-TTC-30H [14].	19
Figura 2.12 – ECU EVT Motor Verks GEVCU [15].	20
Figura 2.13 – Análise gráfica de unidades de controlo eletrónico.	23
Figura 3.1 – Topologia de uma rede CAN, adaptado de [17].	26
Figura 3.2 – Níveis de tensão num barramento CAN, adaptado de [18].	27
Figura 3.3 – Método de <i>bit stuffing</i> , adaptado de [20].	28
Figura 3.4 – Estrutura de uma <i>frame</i> de dados no protocolo CAN <i>Standard</i> [17].	29
Figura 3.5 – Estrutura de uma <i>frame</i> de dados no protocolo CAN <i>Extended</i> [17].	31
Figura 3.6 – Estrutura de uma <i>frame</i> de dados no protocolo CAN FD <i>Standard</i> [23].	32
Figura 3.7 – Rede física do protocolo de comunicação I2C, adaptado de [25].	35
Figura 3.8 – Transferência de dados no barramento I2C, adaptado de [25].	36
Figura 3.9 – Condições de <i>start</i> e <i>stop</i> no barramento I2C, adaptado de [25].	36
Figura 3.10 – Comunicação SPI com múltiplos escravos, adaptado de [27].	37
Figura 3.11 – Configuração SPI do tipo <i>daisy-chain</i> , adaptado de [27].	38
Figura 3.12 – Diagrama de comunicação SPI mestre/escravo, adaptado de [27].	38
Figura 3.13 – Transmissão de dados de comunicação série, adaptado de [28].	40

Figura 3.14 – Exemplo de uma arquitetura do tipo <i>Superloop</i>	41
Figura 3.15 – Arquitetura <i>Foreground/Background</i> , adaptado de [29].	42
Figura 4.1 – Diagrama de blocos da MeaTec32.	46
Figura 4.2 – Avaliação da MeaTec32.	48
Figura 4.3 – Bloco de alimentação de uma unidade de controlo eletrónico automóvel.	49
Figura 4.4 – Regulador de tensão de 5V.	50
Figura 4.5 – Regulador de tensão de 3.3V.	51
Figura 4.6 – Entradas digitais.	52
Figura 4.7 – Tensão de saída em função da tensão de entrada [35].	53
Figura 4.8 – Entrada analógica isolada.	55
Figura 4.9 – Leitura de resistência variável.	55
Figura 4.10 – Aquitetura CAN MeaTec32.	56
Figura 4.11 – Armazenamento de dados.	58
Figura 4.12 – Isolador digital para comunicação série.	58
Figura 4.13 – (a) Saída digital <i>low-side</i> ; (b) Saída digital <i>high-side</i>	59
Figura 4.14 – Saída digital genérica <i>high-side</i> e <i>low-side</i>	60
Figura 4.15 – Saída digital de potência.	61
Figura 4.16 – Determinação da resistência de shutdown [36].	62
Figura 4.17 – Curva característica da tensão de <i>feedback</i> do AU1R3314S.	63
Figura 4.18 – Aplicação do limite de corrente AU1R3314S.	64
Figura 4.19 – (a) Vista 2D superior da PCB; (b) Vista 2D inferior da PCB; (c) Vista 3D superior da PCB; (d) Vista 3D inferior da PCB.	66
Figura 4.20 – Vista final 3D superior da PCB.	67
Figura 4.21 – Vista final 3D inferior da PCB.	68
Figura 4.22 – (a) PCB em fase de projeto; (b) PCB final.	70
Figura 4.23 – MeaTec32 finalizada.	70
Figura 4.24 – <i>Pinout</i> da MeaTec32.	71
Figura 5.1 – Tensão de saída dos reguladores de tensão.	75
Figura 5.2 – Sinal PWM à entrada e saída do isolador digital ISO7740DWR.	76
Figura 5.3 – Leitura de sinal analógico isolado.	77

Figura 5.4 – Saída digital em controlo de PWM.....	78
Figura 5.5 – Tensão de <i>feedback</i> e corrente de carga.....	79
Figura 5.6 – Desativação de carga por excesso de consumo de corrente.	80
Figura 5.7 – Mensagem CAN convencional.....	82
Figura 5.8 – Mensagem CAN FD.....	83
Figura 5.9 – Leitura da linha de transmissão de dados de comunicação série.	84
Figura 5.10 – Disposição de equipamentos para inspeção por termografia: (a) módulo desenvolvido, (b) multímetro térmico, (c) cargas genéricas, (d) fonte de alimentação.....	85
Figura 5.11 – Temperatura de funcionamento dos MOSFET's de potência.....	86
Figura 5.12 – Evolução da temperatura de funcionamento dos MOSFET 's de potência.	87
Figura 5.13 – Temperatura de funcionamento do microcontrolador.....	88
Figura 5.14 – Motor Lynch LEM-200.....	90
Figura 5.15 – Sistema de contactor e <i>pre-charge</i>	91
Figura 5.16 – Painel de instrumentos e painel de comando.	91
Figura 5.17 – Simulador elétrico: (a) contactor e <i>pre-charge</i> , (b) controlador, (c) MeaTec32, (d) motor elétrico, (e) painel de comando, (f) painel de instrumentos.	92
Figura 5.18 – Esquema de ligações do simulador.....	93
Figura 5.19 – Ambiente de programação do software Eclipse IDE.	97
Figura 5.20 – Fluxograma do programa de testes.	98

Lista de Tabelas

Tabela 1 – Caracterização das unidades comercialmente disponíveis	21
Tabela 2 – Campos de uma <i>frame</i> de dados numa mensagem CAN <i>Standard</i> [17].	29
Tabela 3 – Configuração do campo DLC no CAN FD, adaptado de [24].	33
Tabela 4 – Caracterização das unidades comercialmente disponíveis	47
Tabela 5 – Especificações da PCB produzida [37].	69
Tabela 6 – Características técnicas do osciloscópio Picoscope® 2205 [38].	73
Tabela 7 – Características técnicas do multímetro térmico TRMS Fluke 279 FC[39].	74
Tabela 8 – Dados disponibilizados pelo controlador [42].	95
Tabela 9 – Mensagem CAN de dados do controlador	95

Lista de Siglas

ABS	<i>Anti-Lock Braking System</i>
ADC	<i>Analog to Digital Converter</i>
ARM	<i>Advanced RISC Machine</i>
ASCII	<i>American Standard Code for Information Interchange</i>
BLDC	<i>Brushless DC</i>
BMS	<i>Battery Management System</i>
CAN	<i>Controller Area Network</i>
CAN FD	<i>CAN with Flexible Data-Rate</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
DC	<i>Direct Current</i>
ECU	<i>Electronic Control Unit</i>
EEPROM	<i>Electrically Erasable Programmable Read Only Memory</i>
EMC	<i>Electromagnetic Compatibility</i>
I2C	<i>Inter-Integrated Circuit</i>
IPO	<i>Input-Process-Output</i>
LIN	<i>Local Interconnect Network</i>
MISO	<i>Master Input/Slave Output</i>
MOSFET	<i>Metal-Oxide Semiconductor Field-Effect Transistor</i>
MOSI	<i>Master Output/Slave Input</i>
NTC	<i>Negative Temperature Coefficient</i>
OTA	<i>Over-The-Air</i>
PCB	<i>Printed Circuit Board</i>
PTC	<i>Positive Temperature Coefficient</i>
PWM	<i>Pulse Width Modulation</i>
RTOS	<i>Real Time Operating System</i>
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SMD	<i>Surface Mount Device</i>
SPI	<i>Serial Peripheral Interface</i>

USB	<i>Universal Serial Bus</i>
VE	Veículo Elétrico
VMCI	Veículo com Motor de Combustão Interna

1. Introdução

O presente capítulo introduz o projeto desenvolvido, sendo mencionado o contexto e motivação, descrevem-se também os objetivos pretendidos e por fim é apresentada a estrutura do presente projeto.

1.1. Contexto e motivação

No passado, a indústria automóvel era movida essencialmente à base de combustíveis de origem fóssil. No entanto, com as alterações climáticas verificadas mais recentemente, uma nova mentalidade gera uma procura de soluções inovadoras que fazem uso de energias renováveis.

O interesse pela eletrificação dos veículos tem vindo a aumentar, com a maior parte dos grandes fabricantes de automóveis a apostar na mobilidade elétrica. Conscientes de que a mobilidade elétrica é o futuro do setor automóvel, está a ser feita a passagem gradual para este tipo de tecnologia. Mas a mobilidade elétrica é muito mais, permite desenvolver meios de transporte alternativos, tais como trotinetes elétricas, motociclos elétricos, *karts's* elétricos, entre tantos outros.

Nesse sentido, várias empresas começam a introduzir no mercado viaturas elétricas de custo reduzido. A par de tudo isto, existem ainda entusiastas que por interesse próprio, desenvolvem e adaptam veículos utilitários (motociclos, *buggys*, entre outros) em veículos elétricos. Isto permite a utilização de um veículo que é silencioso, não polui e ainda é económico. Devido ao facto de a tecnologia elétrica ser bastante flexível e prática, facilmente se pode efetuar diferentes configurações dentro do mesmo veículo, como por exemplo o uso de tração dianteira ou tração traseira, consoante a posição do motor elétrico.

Dos principais componentes que constituem o sistema do veículo elétrico, é o controlador que assume a principal tarefa de gestão desta arquitetura. No entanto, em veículos elétricos utilitários, os controladores são de baixo custo e normalmente apenas assumem funções de gestão da energia elétrica. Para a gestão de todo o restante sistema, onde o subsistema de

entretenimento, de arrefecimento e de comunicações são alguns exemplos, existe a necessidade de um módulo que consiga realizar estas tarefas de forma eficaz e segura, mantendo um custo reduzido.

Desta forma, o presente projeto dedica-se ao estudo, caracterização e desenvolvimento de uma ECU de baixo custo e *open source* (código aberto). Destina-se a ser aplicada em veículos elétricos de pequenas dimensões, de custo reduzido, conhecidos como veículos elétricos utilitários.

1.2. Objetivos

O trabalho realizado neste projeto tem como objetivo projetar e desenvolver uma unidade de controlo eletrónico simples e de baixo custo para aplicação em veículos elétricos de baixo custo. É fundamental que tarefas como aquisição de sinais, processamento interno, comunicações, controlo de subsistemas e monitorização interna sejam garantidas, não comprometendo deste modo quer o desempenho quer o custo. Pretende-se que esta unidade seja robusta, *user-friendly* e com a particularidade de ser *open source*. Deve ser possível utilizar esta unidade com *softwares* de programação grátis, dispensando deste modo a compra de licenças de utilização. Também a sua programação deve ser acessível a utilizadores com níveis de conhecimento distintos.

1.3. Estrutura do projeto

O presente documento encontra-se organizado em 6 capítulos. No presente capítulo é apresentada a contextualização deste projeto, a motivação para o trabalho realizado e os objetivos gerais.

No capítulo 2 é exposto o estado da arte, onde é analisada a arquitetura de um veículo elétrico e são abordados alguns dos constituintes deste tipo de veículos, entre os quais, as unidades de controlo eletrónico.

No capítulo 3 apresentam-se as tecnologias de suporte ao presente projeto. É feita a introdução a alguns protocolos de comunicação e tipos de arquitetura de código utilizados em sistemas embebidos.

No capítulo 4 descreve-se o desenvolvimento do *hardware* relativo à MeaTec32 que foi desenvolvida, detalhando o circuito elétrico, a escolha de componentes e a montagem da placa de circuito impresso.

No capítulo 5 descrevem-se os procedimentos de testes de validação que comprovam o correto funcionamento da MeaTec32. Também é abordada a montagem experimental para validação da MeaTec32 num ambiente similar ao do veículo elétrico e é feita uma análise e discussão dos resultados obtidos na execução deste projeto.

Por fim, no capítulo 6, discutem-se as conclusões do trabalho realizado e apresentam-se algumas propostas para melhoramento a serem realizadas em trabalhos futuros.

Esta página foi intencionalmente deixada em branco

2. Estado da Arte

Neste capítulo, de modo a entender o evoluir da tecnologia na área de veículos eléctricos, é feito o enquadramento do trabalho realizado. Começa-se por abordar a história do veículo eléctrico, analisando posteriormente a arquitetura do veículo eléctrico e os principais constituintes relacionados com o presente projeto.

2.1. História do veículo eléctrico

Um veículo eléctrico é um veículo em que o sistema de propulsão é baseado no motor eléctrico. A fonte de energia neste tipo de veículo é eléctrica e encontra-se armazenada nas baterias do veículo. Assim, como o veículo com motor de combustão interna (VMCI) necessita de reabastecer o seu depósito de combustível, também o veículo eléctrico necessita de recarregar as suas baterias com energia eléctrica. Numa arquitetura mais simples esta energia é proveniente de uma fonte externa, como por exemplo a rede eléctrica doméstica. O veículo eléctrico não é algo recente, remonta aliás ao início do século XIX, quando, em 1834, Thomas Davenport desenvolveu um motor eléctrico de corrente contínua que acabou por ser patenteado em 1837 como o primeiro motor deste género. Ainda em 1834 Davenport instalou o motor por si desenvolvido num veículo de três rodas, juntamente com uma bateria não recarregável e nasce assim aquele que é conhecido como o primeiro veículo eléctrico [1]. Na Figura 2.1 pode-se observar um esboço do motor eléctrico de corrente contínua desenvolvido por Thomas Davenport.

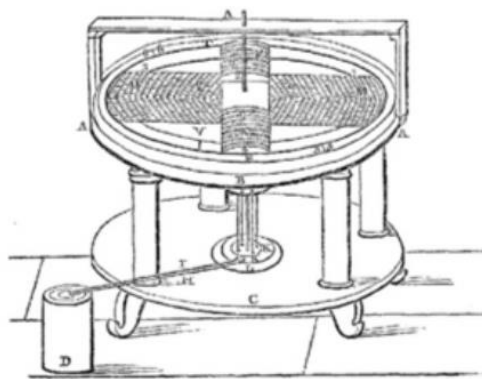


Figura 2.1 – Motor eléctrico de corrente contínua por Thomas Davenport [1].

Nesta época, o interesse pelo veículo elétrico era cada vez maior, uma vez que era um veículo silencioso, prático e de baixa manutenção, ao contrário do que acontecia com os veículos a vapor da época. A procura por este tipo de veículos levou David Salomons, em 1874, a desenvolver o primeiro veículo elétrico dotado de baterias recarregáveis [1]. Posteriormente inúmeras empresas desenvolveram veículos elétricos, como é o exemplo do veículo elétrico que se observa na Figura 2.2, contruído pela empresa *Electric Carriage and Wagon Company*.



Figura 2.2 – Veículo elétrico “Electrobat” de 1894 [2].

No ano de 1900, dos 4200 veículos vendidos nos Estados Unidos da América, 38 % eram veículos elétricos, o que demonstra a grande popularidade dos mesmos naquela época [1]. De facto, a maioria dos táxis utilizados eram veículos elétricos.

No entanto a partir do ano de 1930 o uso do veículo elétrico entrou em declínio, devido em parte à invenção do motor de arranque elétrico que eliminava a necessidade da manivela manual para o VMCI, à maior autonomia deste último tipo de veículo e à produção em massa do Ford Model T. A aquisição de um veículo elétrico era mais dispendiosa face à aquisição de um VMCI e o último novo modelo de um veículo elétrico vendido nos Estados Unidos da América foi o veículo *Automatic*, fabricado pela *Automatic Transmission Company of Buffalo*. O custo deste veículo era de 1200\$ enquanto que o custo de um Ford Model T rondava os 300\$. A junção destes fatores fez o veículo elétrico entrar em “hibernação” [1].

2.2. Arquitetura veículo elétrico

Os veículos elétricos que foram colocados no mercado, em meados dos anos 80, eram muitas vezes adaptações efetuadas a veículos de motor de combustão interna. O VMCI era alterado de forma a que fosse possível receber o motor elétrico, baterias e outros sistemas associados ao veículo elétrico. Resumidamente os veículos elétricos não eram desenvolvidos de raiz com o seu principal objetivo em mente. Um veículo adaptado dispunha assim do peso, dos materiais e da aerodinâmica de um VMCI e desta forma não era explorada a total potencialidade do veículo elétrico. Mais tarde, este tipo de veículo começou a ser desenvolvido de raiz, onde fatores como velocidade máxima, autonomia, aceleração e aerodinâmica eram considerados desde o início do projeto. O uso de materiais compósitos de baixo peso aliado a *designs* que permitiam coeficientes de arrasto reduzidos, bem como a redução das perdas por atrito, derivado de sistemas mecânicos otimizados, permitiram ao veículo elétrico colmatar a sua principal desvantagem até ao momento, a baixa eficiência energética.

Face ao VMCI, a configuração do veículo elétrico é extremamente flexível, as baterias podem facilmente ser distribuídas pelo chassis de forma a manter o centro de gravidade do veículo próximo do solo. O uso de motores elétricos colocados diretamente nas rodas permite flexibilidade das configurações de tração do veículo, bem como o controlo independente da velocidade de cada roda. Também o carregamento do veículo é flexível, podendo ser carregado com sistemas de energia solar existentes no próprio veículo, fazendo uso da energia que é recuperada na utilização do veículo ou o mais habitual uso de energia da rede doméstica.

A arquitetura do veículo elétrico pode ser dividida em três subsistemas principais, propulsão elétrica, fonte de energia e sistemas auxiliares [1]. A propulsão elétrica é composta pelo controlador eletrónico, inversor, motor elétrico e mecanismo de transmissão de movimento mecânico. O subsistema de fonte de energia é formado pela unidade de carregamento, o gestor de energia e a bateria. Os sistemas auxiliares são compostos pela unidade de direção assistida, a unidade de gestão térmica e a bateria de sistemas auxiliares. Na Figura 2.3 observa-se a arquitetura do veículo elétrico, onde os sinais provenientes do acelerador e do travão do veículo são adquiridos pelo controlador eletrónico. Este regula o fluxo de corrente, a ser aplicado ao motor elétrico. O conversor de potência converte a corrente contínua da bateria em corrente alternada, que possa ser utilizada pelo motor elétrico. As setas

bidirecionais representadas na Figura 2.3 indicam que o fluxo de energia eléctrica ocorre nos dois sentidos, onde é recuperada a energia gerada na travagem para efetuar o carregamento da bateria. A unidade de gestão de energia comunica diretamente com o controlador eletrónico, a unidade de carregamento e a fonte de energia. O conversor de tensão dos sistemas auxiliares converte a tensão da fonte de energia principal, a bateria, para uma tensão de trabalho dos componentes auxiliares que tipicamente é de 12 V.

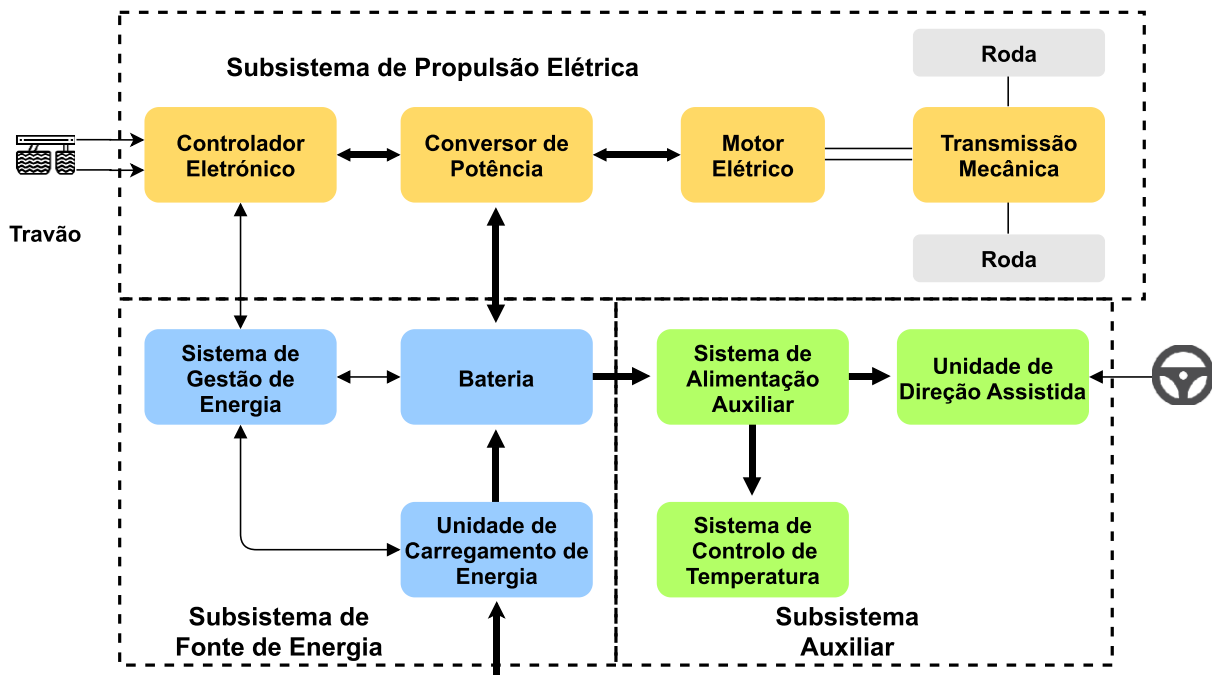


Figura 2.3 – Arquitetura geral do veículo eléctrico, adaptado de [1].

Da arquitetura apresentada na Figura 2.3 realça-se o controlador eletrónico, pois o objetivo do presente projeto está centrado no desenvolvimento de um módulo auxiliar para funcionamento em conjunto com o controlador, podendo partilhar ou assumir por completo as funções por ele desempenhadas. De seguida detalham-se os principais módulos integrados na arquitetura geral do veículo eléctrico, que funcionam em conjunto com o sistema de controlo, tais como os tipos de motores eléctricos utilizados, o sistema de gestão de bateria, o controlador do motor eléctrico e as unidades de controlo existentes para veículos eléctricos.

Motores eléctricos

Os motores eléctricos são utilizados em diversos contextos domésticos e industriais. Desde cedo foram aplicados em meios de transporte como comboios, autocarros e outros tipos de veículos utilitários. A função do motor eléctrico num veículo eléctrico é converter a energia eléctrica, proveniente da bateria, em energia mecânica e estes motores dividem-se essencialmente em dois grupos distintos: motores eléctricos de corrente contínua e motores eléctricos de corrente alternada.

A Figura 2.4 ilustra um diagrama com alguns destes motores, como por exemplo o motor DC (*Direct Current*), motor de indução, motor síncrono de ímanes permanentes, motor de relutância comutado e motor de corrente contínua sem escovas.

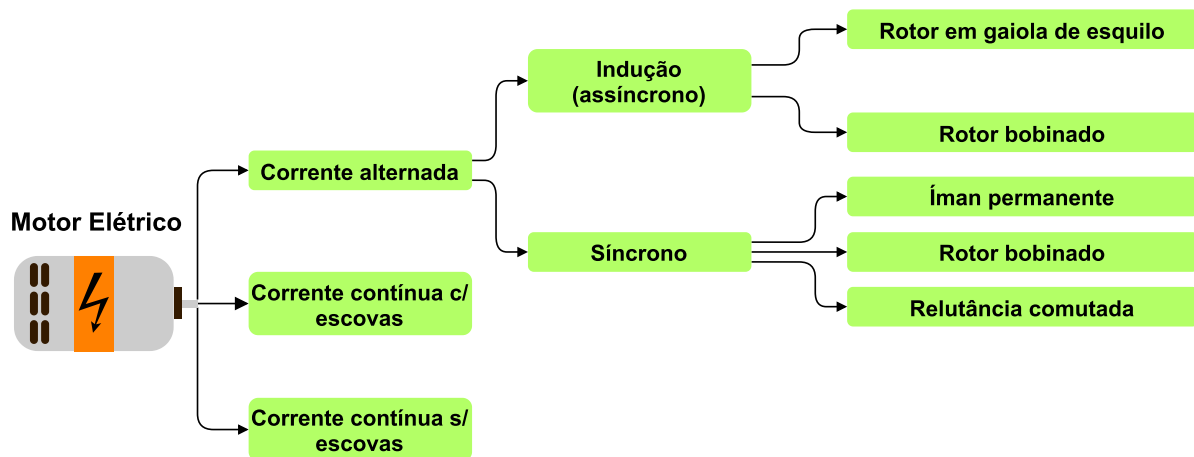


Figura 2.4 – Diagrama de motores eléctricos utilizados em veículos eléctricos, adaptado de [3].

Os motores de corrente contínua sem escovas, também conhecidos como BLDC (*Brushless DC*) são caracterizados pela sua fiabilidade e durabilidade tendo sido utilizados nos primeiros veículos eléctricos. Atualmente ainda são utilizados em veículos híbridos e veículos eléctricos de pequenas dimensões, como bicicletas, trotinetes, entre outros [4]. Por outro lado, ainda que não muito usual, os motores de corrente contínua com escovas oferecem um binário elevado e, à semelhança dos motores BLDC, são uma hipótese para a conversão de veículos eléctricos utilitários devido à simplicidade do processo de conversão. Atualmente, alguns dos fabricantes de veículos eléctricos de alta performance como a Tesla®, equipam os seus veículos com motores de indução assíncronos. Este tipo de motor, à semelhança dos motores BLDC, oferece a vantagem de não possuir desgaste mecânico pois não existe contacto entre as partes

móveis interiores. No entanto, a sua utilização em veículos automóveis poderá não ser longa, uma vez que começam a existir novas formas de controlo de outros tipos de motores que podem oferecer vantagens face a este motor. A Tesla[®] por exemplo, no seu *Model 3* já faz uso de uma variação do motor síncrono de ímanes permanentes.

Controladores

Num veículo elétrico, o controlador encontra-se situado entre a bateria, que é responsável por armazenar e fornecer energia, e o motor elétrico. O controlador tem como função principal controlar a energia que é entregue ao motor elétrico, daí o seu nome técnico. Através da aquisição dos sinais do sensor de pedal do acelerador, do sensor de travão e outros sensores afetos ao modo de condução, o controlador efetua a gestão da energia entregue ao motor elétrico, de modo a controlar a velocidade e binário que o motor elétrico produz [5]. É comum a utilização de dois sensores de pedal do acelerador tal como num veículo de combustão interna, o controlador efetua a leitura dos dois sinais e, através da comparação dos valores reais destes sensores, verifica se existe plausibilidade daquilo que é a intenção do condutor. Caso exista uma discrepância no sinal, o controlador pode entrar em modo de segurança protegendo assim o sistema e ocupantes.

O controlador pode também gerir a modelação dos sinais de controlo para cada fase do motor elétrico, controlar o sentido de rotação do motor elétrico, gerir recuperação de energia em travagens regenerativas e controlar a temperatura do motor elétrico.

Consoante a intenção do condutor, efetua o controlo do motor elétrico da forma mais eficiente e segura possível. No início, a variação da velocidade de um motor DC era realizada através de um potenciómetro de potência, o que não era eficiente pois as perdas de energia neste componente contribuíam para a redução da autonomia do veículo elétrico. Atualmente, a técnica utilizada é o uso do controlador que, através da aplicação de um sinal PWM (*Pulse Width Modulation*), regula de forma precisa a energia entregue ao motor elétrico. Na Figura 2.5 observa-se esta técnica simplificada, mais especificamente uma situação em que o condutor pretende aplicar 50 % de potência ao motor DC. O controlador, através da aquisição deste sinal, aplica um *duty-cycle* de 50 % à tensão da bateria de 96 V para obter uma tensão média aplicada ao motor DC de 48 V.

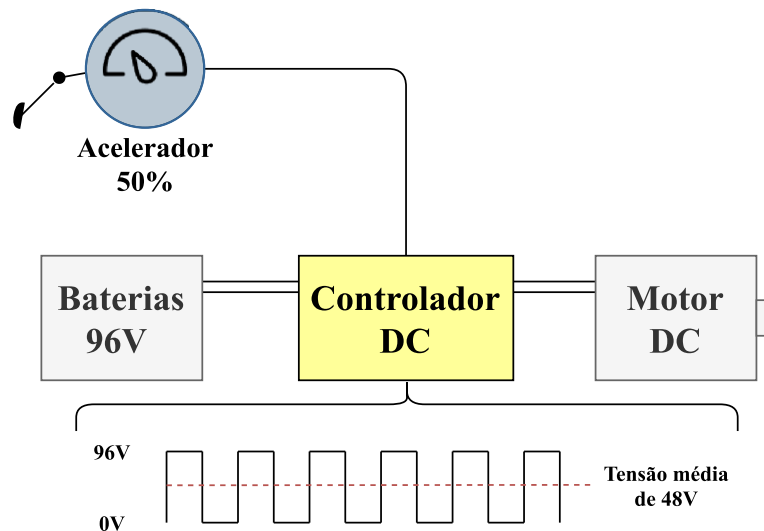


Figura 2.5 – Função simplificada do controlador [5].

Em arquiteturas onde o motor eléctrico utilizado seja de corrente alternada, o controlador também pode integrar a função de inversão da corrente contínua para corrente alternada que possa ser aplicada ao motor eléctrico.

Como se percebe, um controlador não é uma ECU do veículo, ainda que possa fazer a gestão de outras tarefas, a maioria dos veículos eléctricos possui uma ECU genérica para o controlo de outros sistemas, como por exemplo a *interface* visual com o utilizador ou dos sistemas auxiliares do veículo.

Gestão térmica do motor eléctrico

O funcionamento teórico do motor eléctrico é simples: converte energia eléctrica em energia mecânica. No entanto, parte da energia que entra no motor acaba por ser convertida em calor. A gestão térmica do motor eléctrico é um aspeto fundamental no controlo de um veículo eléctrico pois, para que o motor eléctrico seja o mais eficiente possível, é essencial que este opere em gamas de temperatura para as quais foi desenvolvido. Num motor eléctrico existem essencialmente três formas de arrefecimento, são elas o arrefecimento por convecção, arrefecimento forçado a ar e arrefecimento a líquido. A dissipação de calor no primeiro tipo de arrefecimento é conseguida através de convecção, ou seja, o calor é retirado do motor eléctrico através do corpo ou das aberturas da carcaça do mesmo. No arrefecimento forçado a ar, o motor dispõe normalmente de uma ventoinha que força a entrada do ar à temperatura

ambiente no motor, reduzindo assim a sua temperatura interior. No arrefecimento a líquido, é utilizado um circuito específico no interior no motor, junto ao estator, no qual circula um líquido de refrigeração, permitindo assim que os componentes do motor permaneçam isolados do circuito de refrigeração.

Na Figura 2.6 observa-se as curvas de binário de um motor elétrico em função da sua rotação, onde se demonstra a influência que o tipo de arrefecimento exerce sobre o desempenho do motor elétrico [6]. O gráfico representa o binário desenvolvido por um determinado motor elétrico em função da sua rotação. No eixo das ordenadas o valor “x” representa o binário contínuo, enquanto o valor “3x” representa o binário de pico, que tipicamente é o triplo do binário contínuo.

A azul escuro observa-se o binário contínuo para um motor refrigerado por ar forçado. O aumento de rotação do veio do motor provoca a subida da temperatura e, de forma a manter a temperatura controlada, a corrente consumida pelo motor é atenuada, ocorrendo uma diminuição do binário. Caso contrário, se o binário permanecesse constante, o motor iria sobreaquecer.

Analisando as curvas a vermelho (que representa o binário contínuo para um motor refrigerado a óleo) e a azul claro (que identifica o binário de pico para o mesmo tipo de refrigeração) observa-se que o binário contínuo é superior, face ao tipo de refrigeração a ar forçado, em cerca de 75 %.

Caso o líquido de arrefecimento seja uma mistura de água e etilenoglicol, a curva de binário contínuo representada a verde e a curva de binário de pico, a cor laranja, sofrem um aumento face ao tipo de arrefecimento anterior. Pela Figura 2.6 entende-se que o tipo de arrefecimento afeta seriamente o binário contínuo que pode ser desenvolvido por um determinado motor e influencia apenas ligeiramente o binário de pico [6].

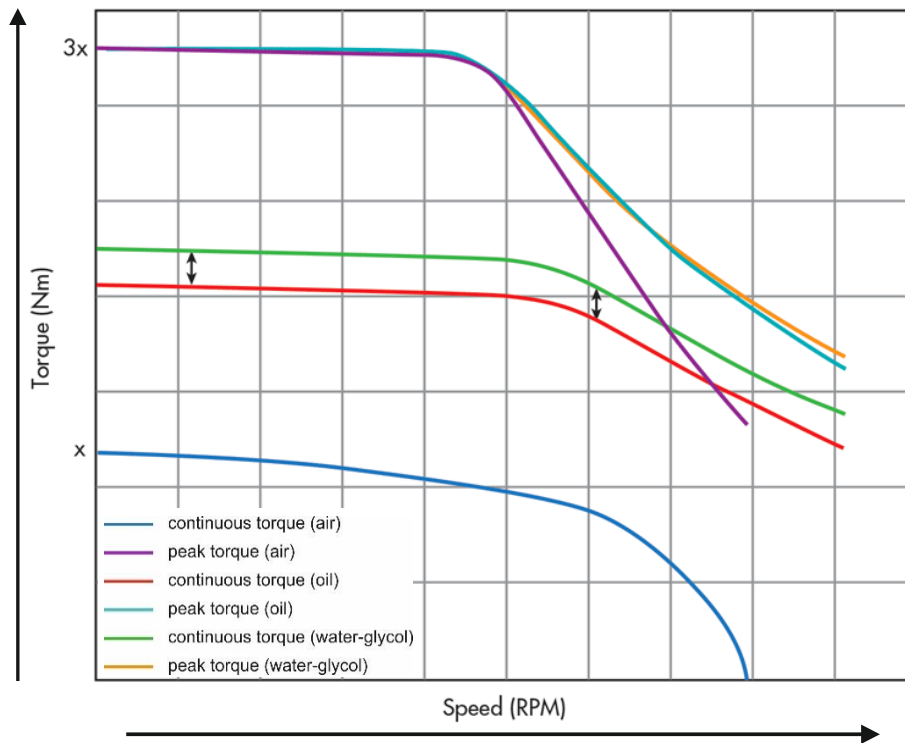


Figura 2.6 – Influência do arrefecimento no desempenho de um motor eléctrico [6].

Sistema de gestão de bateria

O sistema de gestão de bateria, conhecido como BMS (*Battery Management System*) é um sistema fundamental na arquitetura do veículo eléctrico. Com a ajuda deste sistema é possível garantir que as baterias do VE operem em condições ótimas, assegurando assim a proteção das células e permitindo prolongar a vida útil das mesmas. De modo a garantir estas condições o BMS efetua a monitorização das principais grandezas de funcionamento das baterias como a tensão, corrente e temperatura das células que a compõem. O BMS permite a comunicação com os outros sistemas do VE através de uma *interface* digital, do tipo CAN (*Controller Area Network*), em especial com a unidade de controlo eletrónico. Assim, através da monitorização realizada, os outros sistemas do VE podem ajustar em tempo real as condições de funcionamento do VE consoante os dados que o BMS lhe transmite. Em algumas arquiteturas de VE, o BMS pode efetuar o controlo em malha fechada da temperatura de funcionamento das células da bateria ou então apenas transmitir o estado da temperatura ao sistema específico que gere o arrefecimento das baterias. Na Figura 2.7 é possível observar a estrutura de um BMS proposto por [7].

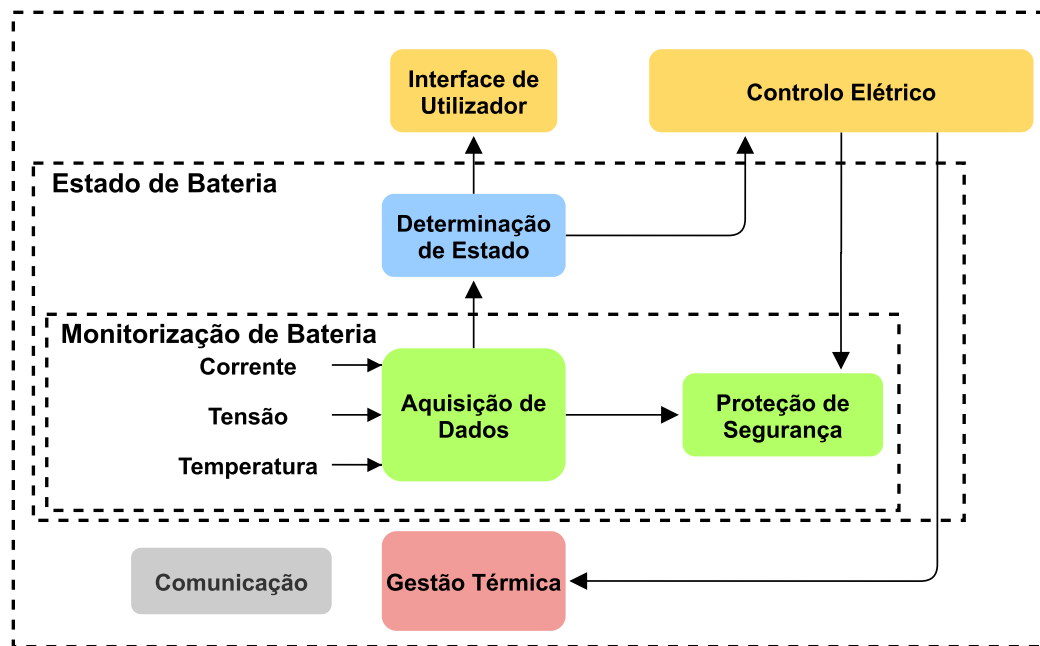


Figura 2.7 – Exemplo de sistema de monitorização de bateria, adaptado de [7].

Unidades de controlo eletrónico

Na década de 70 surgiram as primeiras unidades de controlo eletrónico [8], sendo que a capacidade computacional disponível nos automóveis não parou de crescer desde então. O programa de controlo é responsável pela gestão do sistema eletrónico do automóvel e encontra-se alojado na unidade de controlo eletrónico. Esta ECU é constituída por uma placa de circuito impresso onde se encontra, entre outros componentes, um ou mais microcontroladores responsáveis pela gestão do sistema no qual a ECU se encontra aplicada. Podem ser descritas como sistemas embebidos, que operam segundo o modelo IPO (*Input-Process-Output*) [9]. Através da aquisição de sinais de sensores do automóvel, é processada esta informação e posteriormente realizado o controlo dos atuadores afetos ao funcionamento do sistema. Na Figura 2.8 observa-se o diagrama de blocos simplificado de uma ECU genérica. A informação adquirida pelos sensores pode ser de diferentes grandezas físicas, tal como os atuadores podem ser de diferentes tecnologias de atuação, sendo deste modo necessário que a ECU se consiga adaptar a diferentes sistemas.

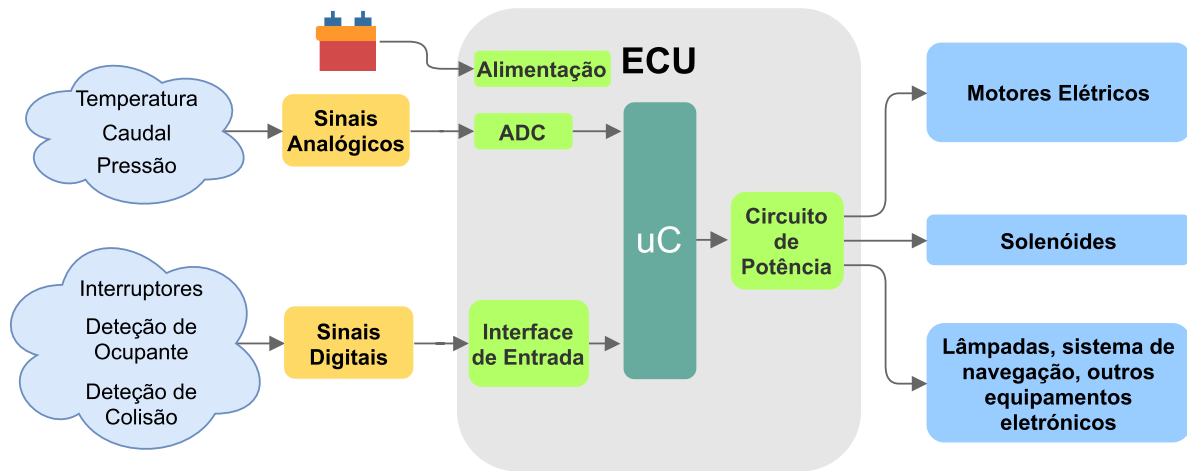


Figura 2.8 – Diagrama de blocos de uma unidade de controlo eletrónico, adaptado de [10].

Atualmente os fabricantes automóveis adotam uma estrutura distribuída, como a que se observa mais à frente na Figura 3.1, onde todas as ECU's se encontram conectadas entre si através de um barramento de comunicação, mas funcionam de forma independente, assumindo tarefas como a gestão do motor ou como o ABS (*Anti-Lock Braking System*) do automóvel [10].

Analisando a Figura 2.8 verifica-se que, no seu estado mais simples, existem quatro sistemas integrantes de uma ECU genérica. É constituída pelo sistema de regulação de alimentação que adapta a tensão proveniente da bateria, para o nível de tensão de trabalho do microcontrolador. Também é parte integrante o módulo de conversão ADC (*Analog to Digital Converter*), responsável pela conversão de sinais analógicos para sinais digitais, capazes de ser interpretados pelo processador da ECU. O processamento da ECU é realizado pelo microcontrolador que consiste num circuito integrado, composto pela memória de programa, unidade de processamento, portos de entrada e saída, memória de dados, relógio interno, entre outros [11]. O circuito de saída realiza a adaptação do nível de tensão do microcontrolador para o nível de tensão de trabalho dos atuadores existentes.

Uma ECU tem de ser capaz de resistir às vibrações causadas pelo funcionamento do automóvel, funcionar em diferentes zonas de temperatura e resistir a condições atmosféricas adversas. Como tal, o encapsulamento que contém todo o *hardware*, necessita de garantir isolamento de poeiras e humidade, ser resistente a vibrações e suportar a temperatura de funcionamento da ECU. É possível afirmar que desempenha um papel fundamental em manter a ECU em boas condições de funcionamento [11].

2.3. Unidades disponíveis para controlo de veículos elétricos

Existem no mercado diferentes tipos de ECU para aplicação em veículos elétricos, desde aplicações complexas, realizadas por fabricantes, até aplicações simples que são realizadas por entusiastas da área. De seguida apresentam-se alguns modelos de ECU para aplicação neste tipo de veículos.

Vector® VC121-12

A ECU Vector® VC121-12, desenvolvida pela empresa Vector®, é uma unidade de controlo genérica que se encontra focada para o setor automóvel, destina-se a ser utilizada no desenvolvimento e teste de protótipos deste setor, permitindo uma grande flexibilidade. Na Figura 2.9 observa-se o aspeto exterior desta ECU. A sua programação é feita através do *software* AUTOSAR™, disponibilizado pela NXP Semiconductors®. O custo de aquisição deste equipamento, que engloba as licenças de utilização, é cerca de 5870 €, à data da consulta (13/01/2020). Utiliza um microcontrolador de 32 bit, possui uma forte componente de comunicações com 6 *interface* CAN 1 *interface* Ethernet, 2 *interface* LIN (*Local Interconnect Network*) e 1 *interface* FlexRay, tornando-a flexível para adaptação em diferentes veículos [12]. Dispõe também de diversas entradas analógicas e digitais, com 40 saídas digitais para utilização em diferentes modos de corrente e com possibilidade de utilização de PWM. A tensão de alimentação é de 8 a 16 V DC e possui duas alimentações de 5 V para sensores.



Figura 2.9 – ECU Vector VC121-12 [12].

Especificações da ECU Vector® VC121-12:

- Microcontrolador de 32 bit e 80/120 MHz STMicroelectronics®;
- 6 *interface* CAN;
- 1 *interface* Ethernet;
- 2 *interface* LIN;
- 1 *interface* FlexRay;
- 14 entradas analógicas de 0 a 5 V;
- 8 entradas analógicas de 0 a 16 V;
- 20 entradas digitais até 18 V;
- 8 entradas de leitura de frequência até 20 kHz;
- 40 saídas digitais, com diferentes correntes máximas e capacidade de PWM;
- 2 alimentações de 5 V para sensores;
- Tensão de alimentação de 8 a 16 V DC.

Ecotrons® EV2206B03

A ECU Ecotrons® EV2206B03 que se observa na Figura 2.10 é desenvolvida pela empresa Ecotrons®. Este tipo de ECU é caracterizada como sendo uma unidade de supervisão para veículos elétricos e híbridos, funcionando em conjunto com a arquitetura elétrica já existente no veículo. A programação desta ECU é feita em Matlab/Simulink e permite programação OTA (*Over-The-Air*). O orçamento obtido para a aquisição desta ECU, incluindo as licenças de utilização, é de aproximadamente 3500 €, na consulta (13/01/2020). Baseia-se no microcontrolador NXP MPC5606 de 32 bit que funciona a 64 MHz. O módulo de comunicações é composto por 3 *interface* CAN e 1 *interface* LIN. Nesta ECU existem 13 entradas analógicas e 18 entradas digitais, 3 entradas para leitura de sinais de frequência e 28 saídas digitais com capacidade de monitorização de corrente [13]. Funciona a uma tensão de alimentação de 9 a 32 V DC e oferece 4 saídas para alimentações para sensores a 5 V.



Figura 2.10 – ECU Ecotrons EV2206B03 [13].

Especificações da ECU Ecotrons® EV2206B03:

- Microcontrolador de 32 bit e 64 MHz NXP Semiconductor® MPC5606;
- 5 *interface* CAN;
- 1 *interface* LIN;
- 12 entradas analógicas de 0 a 5 V;
- 1 entrada analógica de 0 a 36 V;
- 18 entradas digitais até 24 V;
- 3 entradas de leitura de frequência até 1.5 kHz;
- 28 saídas digitais, com diferentes correntes máximas e capacidade de PWM;
- 4 alimentações de 5 V para sensores;
- Tensão de alimentação de 9 a 32 V DC.

TTControl® HY-TTC-30H

A empresa austríaca TTControl® Hydac International desenvolveu esta ECU genérica especificamente para aplicações em veículos utilitários de pequeno porte onde o fator preço é fundamental, mas mantendo uma boa capacidade de processamento. Esta ECU, que se observa na Figura 2.11, tem um custo de aquisição, incluindo as licenças de utilização, de aproximadamente 1800 €, na consulta (15/01/2020). É utilizado um microcontrolador de 32 bit com uma frequência de 80 MHz e a comunicação é realizada através de apenas uma *interface* CAN. Possui 10 entradas analógicas para diferentes tipos de sensores, 4 entradas

digitais com possibilidade de leitura de frequência e 16 saídas de potência. Existe a função de monitorização em 6 destas saídas, capaz de determinar a corrente consumida pelas cargas. Disponibiliza apenas uma alimentação para sensores a 5 V e funciona com uma tensão de alimentação de 8 a 32 V DC. Para a sua programação desta ECU utiliza-se a linguagem C/C++. Verifica-se também que, para um módulo tão compacto, as especificações oferecidas são notórias.



Figura 2.11 – ECU TTControl HY-TTC-30H [14].

Especificações da ECU TTControl[®] HY-TTC-30H:

- Microcontrolador de 32 bit e 80 MHz Infineon[®] XC2200;
- 1 interface CAN;
- 8 entradas analógicas de 0 a 5 V/10 V;
- 2 entradas analógicas de 0 a 32 V;
- 4 entradas digitais até 32 V, com leitura de frequência até 10 kHz;
- 16 saídas com diferentes correntes máximas e capacidade de PWM;
- Monitorização da corrente de carga de 6 saídas digitais
- 1 alimentação de 5 V para sensores;
- Tensão de alimentação de 8 a 32 V DC;
- Monitorização de temperatura interna.

GEVCU 6.22

A empresa EVTV Motor Werks LLC disponibiliza esta ECU, que se observa na Figura 2.12, como um módulo genérico para controlo de veículos elétricos. Esta unidade foi desenvolvida com o intuito de ser *open source*. Integra um microcontrolador de 32 bit a uma frequência de 84 MHz, a comunicação é realizada através de duas *interface CAN* e uma *interface USB* (*Universal Serial Bus*) para programação e ajuste de parâmetros pelo utilizador. Possui 8 entradas analógicas e 4 entradas digitais isoladas, as 8 saídas digitais disponíveis permitem utilização de PWM para controlo das cargas. Disponibiliza diferentes alimentações para sensores de 5 e 12 V. A tensão de funcionamento é de 6 a 16 V DC e o custo de aquisição, não sendo necessárias licenças, é de aproximadamente 535€ à data da consulta (13/01/2020) [15].

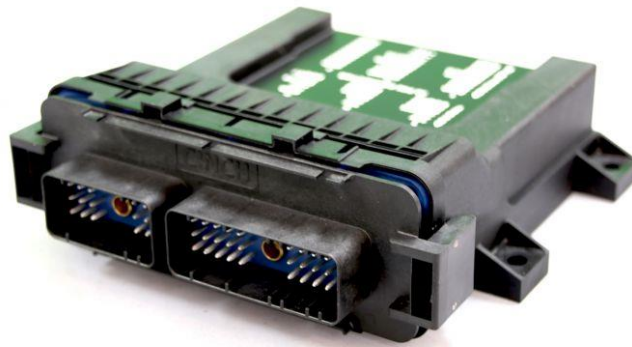


Figura 2.12 – ECU EVTV Motor Werks GEVCU [15].

Especificações da ECU EVTV Motor Werks LLC GEVCU

- Microcontrolador de 32 bit e 84 MHz Microchip® SAM3X8E;
- 2 *interface CAN*;
- 1 *interface USB*
- 8 entradas analógicas de 0 a 12 V;
- 4 entradas digitais de 0 a 15 V;
- 8 saídas digitais com capacidade de PWM;
- 6 alimentações de 5 V para sensores;

- 3 alimentações de 12 V para sensores;
- Tensão de alimentação de 6 a 16 V DC.

De forma a avaliar as unidades apresentadas, que representam o estado da arte em ECU's de baixo custo disponíveis comercialmente, foi efetuado um estudo comparativo (Tabela 1) que pretende salientar as principais características de cada ECU. Foram definidas 5 categorias que permitem avaliar cada ECU: processamento, comunicações, entradas e saídas, diagnóstico, simplicidade de utilização e preço.

Tabela 1 – Caracterização das unidades comercialmente disponíveis.

Funções	VECTOR VC121-12	ECOTRONS EV2206B03	TTCNTROL HY-TTC 30	GEVCU 6.22
Processamento	STM 32bit	NXP 32bit	Infineon 32bit	Atmel 32bit
	80~120MHz	64~150MHz	80MHz	84MHz
	192kB RAM	80~192kB SRAM	82kB RAM	96kB SRAM
	3MB Flash	3MB Flash	768kB Flash	512kB Flash
Comunicações	6 CAN	5 CAN	1 CAN	2 CAN
	1 Ethernet	1 LIN		
	2 LIN			
	1 FlexRay			
I/O	14 An 0-5V	12 An 0-5V	8 An 0-5V/10V	8 An 0-12V
	8 An 0-16V	1 An 0-36V	2 An 0-32V	4 Dig Input 0-15V
	20 Dig Input 0-16V	18 Dig Input 0-24V	4 Dig/Freq Input 0-32V	8 Dig Output 0-12V
	8 Freq Input 0-16V	3 Freq Input 0-24V	16 Dig Output	
	40 Dig Output	28 Dig Output		
Diagnóstico	N/D	N/D	6 Dig Output	N/D
Simplicidade	AUTOSAR	Matlab/Simulink	C Language	C Language
Preço	5 870 €	3 500 €	1 800 €	535 €

Com base nos dados da Tabela 1 foi atribuída uma pontuação dentro de cada categoria para cada ECU, com o processamento a representar a frequência de processamento e memória de cada processador. As comunicações caracterizam a diversidade e quantidade de canais de protocolos de comunicação, as entradas e saídas permitem distinguir a versatilidade de cada ECU. O diagnóstico pretende caracterizar a capacidade de ECU em diagnosticar o estado das suas saídas digitais, a simplicidade define a facilidade de utilização e programação de cada ECU, englobando a curva de aprendizagem necessária para o utilizador. O preço incorpora o valor da ECU em si, bem como todos os custos relativos à aquisição de *software* e licenças. A

cada categoria foi atribuída uma avaliação de 1 a 5 valores, com base nas especificações de cada ECU, sendo que 5 valores significa o melhor desempenho na categoria correspondente.

Na Figura 2.13 observa-se a análise gráfica que permite diferenciar as unidades abordadas com base na pontuação que lhes foi atribuída. A ECU Vector® VC121-12 demonstrou ser a ECU mais completa a nível do seu desempenho, no entanto é também a que apresenta um maior custo. A aplicação desta ECU e da ECU ECOTRONS num veículo elétrico utilitário não é justificável, devido ao grande custo que apresentam. A ECU TTCControl® HY-TTC-30H apresenta um custo moderado, no entanto a existência de apenas uma *interface* CAN pode ser um entrave à utilização desta ECU num VE utilitário. A ECU GEVCU 6.22 apresenta um custo relativamente baixo e tem a vantagem acrescida de ser flexível devido ao facto de ser *open source*, no entanto a falta de diagnóstico de saídas digitais também pode ser um entrave à escolha desta ECU.

Em resumo, constata-se a falta de uma ECU de baixo custo que consiga reunir as principais características aqui demonstradas e possivelmente melhorar alguns aspetos funcionais. Possuir características como programação em *softwares* gratuitos, como o Arduino IDE ou o Eclipse IDE, a existência de barramentos CAN convencional e CAN FD (*CAN with Flexible Data-Rate*) com isolamento nestes barramentos que permitam a utilização da ECU em sistemas onde é necessário isolamento elétrico entre dois módulos de comando, são algumas das funcionalidades que se podem melhorar. Também a proteção de saídas de potência por corrente e temperatura contribui para a capacidade da ECU em realizar a monitorização do sistema em que se integra. A ECU que se pretende desenvolver tem como alvo de aplicação, veículos utilitários em que o seu custo é da ordem dos 2000€, podendo ir até 15000€ para veículos utilitários top de gama. É, portanto, necessário desenvolver uma ECU que permita tudo isto, mas a um custo reduzido.

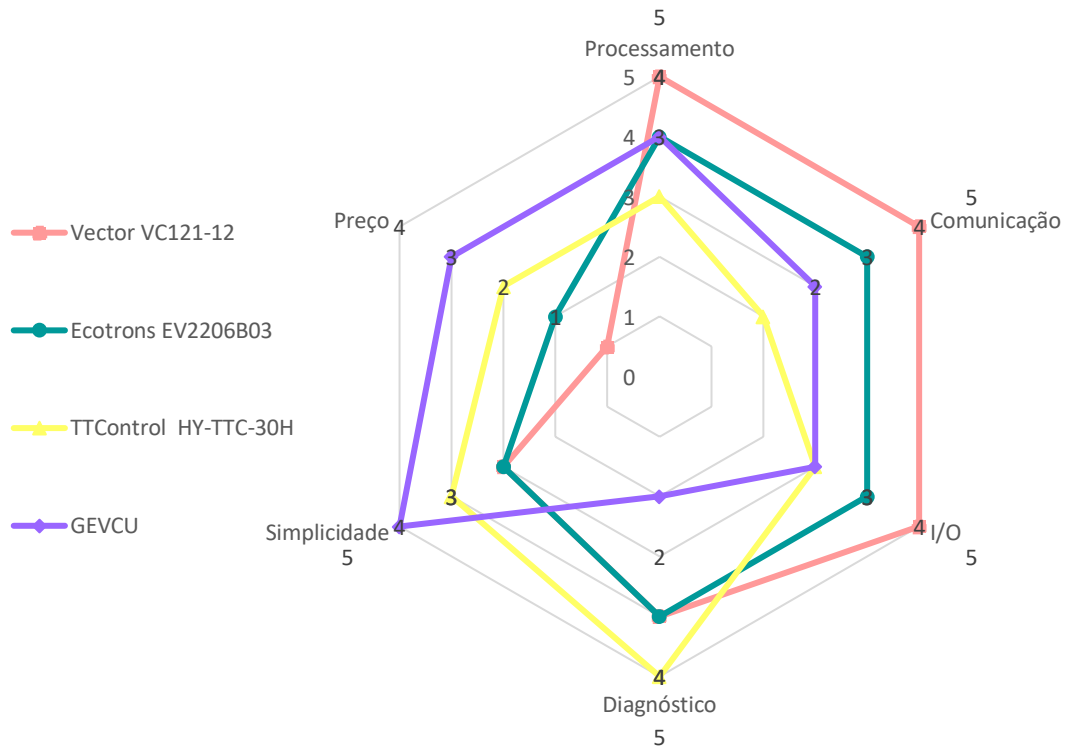


Figura 2.13 – Análise gráfica de unidades de controlo eletrónico.

Esta página foi intencionalmente deixada em branco

3. Tecnologias de Suporte

Neste capítulo é feito o enquadramento das tecnologias de suporte utilizadas no desenvolvimento do presente projeto. Inicia-se o capítulo com a apresentação de alguns dos protocolos de comunicação presentes no automóvel, abordando em detalhe os protocolos que se utilizam no desenvolvimento deste projeto. Por fim introduzem-se algumas arquiteturas de código utilizadas em programação de sistemas embebidos.

3.1. Protocolos de comunicação

Um veículo moderno possui diferentes tipos de comunicação que fazem a gestão e monitorização de todos os sistemas do veículo. Nas secções seguintes detalham-se os protocolos de comunicação utilizados.

3.1.1. Comunicação CAN

O protocolo de comunicação CAN é tipo de comunicação principal da ECU desenvolvida, que permite conectar múltiplas unidades de controlo em um barramento único. Num veículo podem existir distintas configurações de barramento CAN conforme as necessidades de cada unidade e normalmente utilizam-se dois ou três barramentos distintos para as diferentes redes presentes (instrumentação, gestão de energia ou segurança). O CAN foi inicialmente desenvolvido para a indústria automóvel com o intuito de melhorar a comunicação nos automóveis e reduzir o número de cablagens elétricas, através da utilização de um barramento que utiliza apenas dois fios. A escolha deste protocolo recaiu sobre o tipo de vantagens que oferece, tais como a fiabilidade, robustez em ambientes ruidosos e resistência a interferências eletromagnéticas. Estas características tornam-no ideal para a utilização num veículo elétrico. No futuro, caso seja necessário adicionar mais dispositivos de controlo e *interface* ao veículo, é facilmente configurado um novo dispositivo no barramento CAN, uma vez que necessita apenas de se conectar à rede já existente, eliminando a necessidade de acrescentar ligações.

A norma ISO-11898: 2003 define o barramento CAN, a comunicação CAN suporta vários mestres, onde todos têm permissões e funcionalidades idênticas entre si. O CAN funciona

como uma comunicação *broadcast* onde as mensagens são transmitidas ao barramento inteiro e não apenas a um nó [17]. As mensagens neste barramento podem ser muitas, mas o seu tamanho é curto, cada mensagem possui um identificador que distingue a origem dos dados transmitidos, desta forma os nós, com base neste identificador, podem determinar se processam a mensagem ou se a ignoram. Uma vez que todos os nós servem de mestre, a transmissão das mensagens dá-se pela prioridade que as mesmas possuem onde um endereço menor significa uma maior prioridade. Este sistema é regido pelo protocolo CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) que define a prioridade das mensagens a transmitir de modo a evitar a colisão no barramento.

Configuração de uma rede CAN

Uma rede CAN é composta no mínimo por dois nós e, em barramentos com uma distância inferior a cerca de 40 metros, a taxa de transmissão de dados pode ir até 1 Mbit/s. Cada nó é composto por um processador que efetua o tratamento das mensagens, um controlador CAN que faz a gestão dos dados transmitidos e recebidos para o barramento e por fim um *transceiver* que adapta os níveis de sinal do barramento para um sinal digital interpretável pelo processador. Um exemplo de uma rede CAN pode ser visto na Figura 3.1, onde se verifica a existência de duas resistências de terminação do barramento que têm como função evitar reflexões [17].

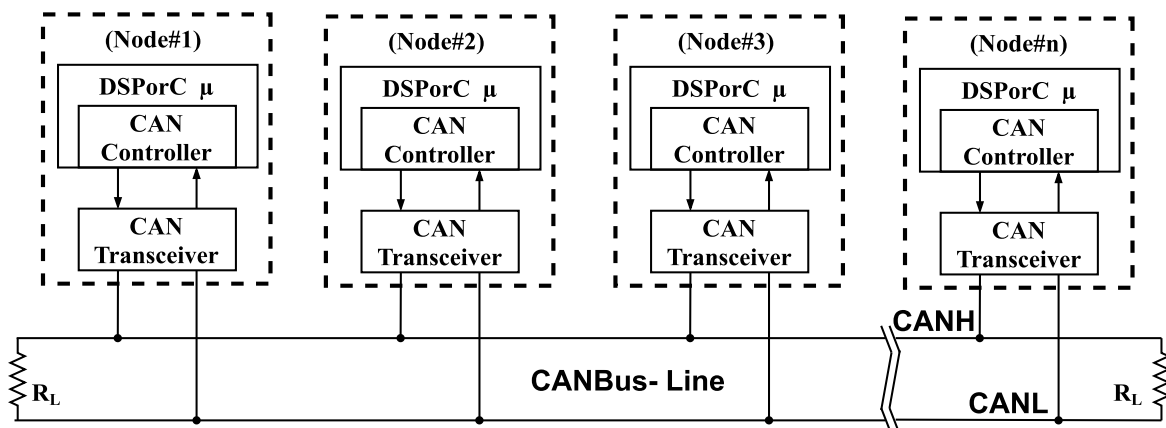


Figura 3.1 – Topologia de uma rede CAN, adaptado de [17].

Níveis de tensão no barramento CAN

A comunicação CAN pode traduzir-se pelos níveis de tensão no barramento. A transmissão de dados é feita de forma diferencial, onde os dados transmitidos dependem da diferença de tensão entre as duas linhas, que fisicamente constituem o barramento, denominadas de CAN High (CANH) e CAN Low (CANL). Na Figura 3.2 é possível observar os estados lógicos que existem num barramento CAN: o estado dominante (*Dominant Logic L*) e o estado recessivo (*Recessive Logic H*). Estes estados são definidos pela diferença de tensão entre a linha CAN High e CAN Low, se $CANH - CANL$ for inferior a 0.5 V, o estado diz-se recessivo (“1”). Caso $CANH - CANL$ for superior a 1.5 V, então o estado é dominante (“0”).

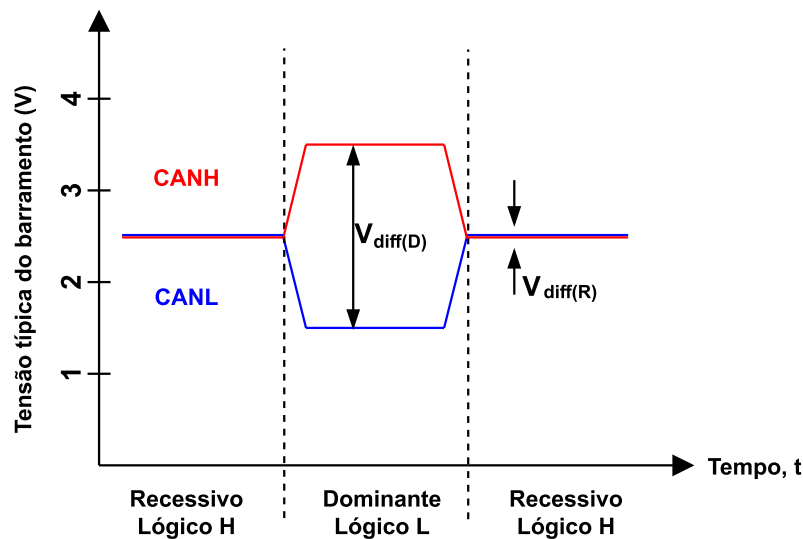


Figura 3.2 – Níveis de tensão num barramento CAN, adaptado de [18].

Uma vez que no barramento CAN a transmissão é realizada de forma diferencial, a tolerância a ruído é elevada porque o ruído é induzido nas duas linhas do barramento, mantendo as diferenças de tensão idênticas entre as duas linhas.

Sincronização do barramento CAN

Sendo que cada nó presente no barramento possui o seu próprio relógio e que no barramento não existe um sinal de relógio, é necessário que exista um método de sincronização entre todos os dispositivos. A sincronização é feita através da transição de um estado recessivo para um estado dominante, desta forma se a transmissão de uma mensagem for apenas constituída

por bits a “0” (estado dominante) não existe a sincronização referida [19]. Para garantir que existem transições fixas e a sincronização permanente dos nós, é utilizado o método de *bit stuffing*. Este método estabelece que, a cada cinco bits iguais consecutivos, seja adicionado um bit com o valor oposto de modo a forçar uma transição. Este bit adicional é colocado pelo transmissor e retirado pelo recetor da mensagem, desta forma o utilizador não se apercebe deste método. Pode ver-se a aplicação deste método na Figura 3.3, em que no primeiro nível está a mensagem a transmitir, no segundo nível a mensagem transmitida com os bits adicionais e no último nível a mensagem recebida com a remoção dos bits adicionais.

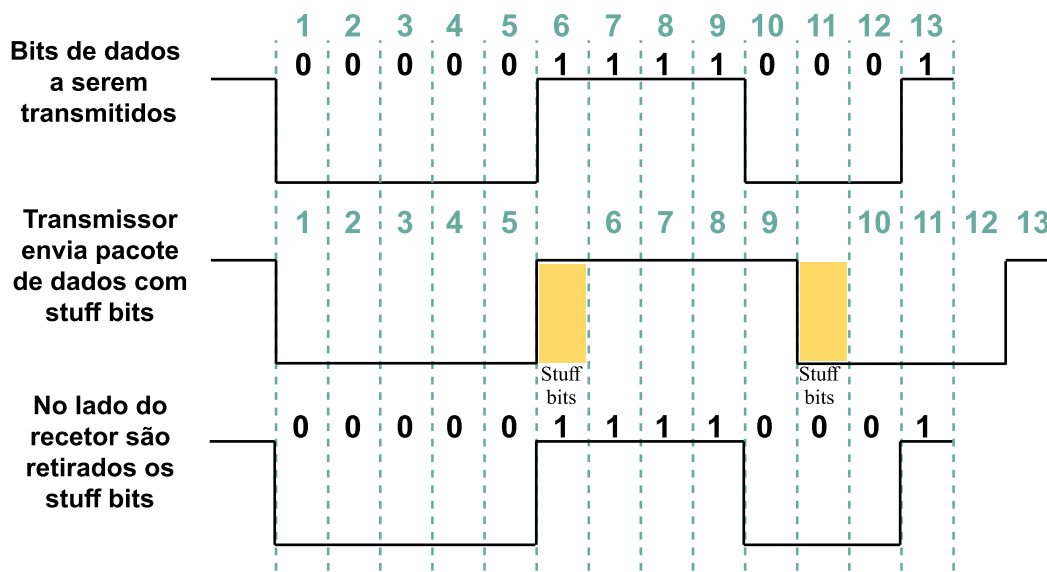


Figura 3.3 – Método de *bit stuffing*, adaptado de [20].

Tipos de mensagens CAN

Existem quatro tipos de mensagens CAN [19]:

- **Data frame:** Mensagem utilizada para enviar dados entre os vários nós no barramento;
- **Remote frame:** Utilizada quando um determinado nó solicita a outro nó que lhe transmita uma mensagem. Esta mensagem de pedido de transmissão não contém dados;
- **Error frame:** Mensagem transmitida sempre que um dos nós deteta que existe uma mensagem com erro. Esta mensagem com o erro é “destruída” levando o transmissor a repetir a transmissão da mensagem;

- **Overload frame:** Mensagem utilizada por um nó quando este necessita de mais tempo para processar a mensagem que está a receber. Deste modo todos os nós no barramento sabem que têm de esperar um pouco para a transmissão da próxima mensagem.

Estrutura de mensagens CAN

As mensagens CAN do tipo *data frame*, podem dividir-se em dois formatos, *standard*, que utiliza um identificador de mensagem de 11 bits e *extended*, em que o identificador de mensagem é composto por 29 bits. Na Figura 3.4 demonstra-se a estrutura da *frame* de dados (*data frame*) para CAN *standard*, também intitulado de CAN 2.0A [16].

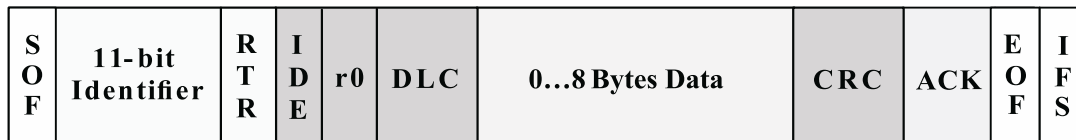


Figura 3.4 – Estrutura de uma *frame* de dados no protocolo CAN *Standard* [17].

De modo a entender cada um dos campos da mensagem CAN apresenta-se de seguida, na Tabela 2, a descrição detalhada da estrutura da *frame* de dados da Figura 3.4.

Tabela 2 – Campos de uma *frame* de dados numa mensagem CAN *Standard* [17].

Campo	Nº de bits	Descrição
<i>Start of Frame (SOF)</i>	1	Início da transmissão da <i>frame</i>
<i>Identifier</i>	11	Identificador dos dados transmitidos
<i>Remote Transmission Request (RTR)</i>	1	Pedido de transmissão de dados por um nó
<i>Identifier Extension (IDE)</i>	1	Define o formato da mensagem, <i>standard</i> ou <i>extended</i>
<i>Reserved bit (r0)</i>	1	Bit reservado
<i>Data Length Code (DLC)</i>	4	Código de 4 bits que define o nº de bytes de dados
<i>Data</i>	0-64 (0-8 bytes)	Dados a ser transmitidos

<i>Cyclic Redundancy Check (CRC) + Delimiter</i>	16	Método de deteção de erros
<i>Acknowledge (ACK) + Delimiter</i>	2	Processo de confirmação de recebimento
<i>End of Frame (EOF)</i>	7	Define o fim da <i>frame</i>
<i>Interframe Space (IFS)</i>	7	Tempo específico requerido pelo controlador

De seguida, de forma a compreender por completo a Tabela 2, detalha-se por completo a estrutura de uma *frame* de dados apresentada na Figura 3.4:

- ***Start of Frame (SOF)***: O SOF define o início da transmissão de uma mensagem com um bit dominante, também serve como forma de sincronização de todos os nós num barramento após ausência de comunicação durante algum tempo;
- ***Identifier***: Identificador de 11 bit que identifica a mensagem. Também serve para definir a prioridade da mesma, quanto mais baixo o seu valor, maior é a prioridade da mensagem em aceder ao barramento [21];
- ***Remote Transmission Request (RTR)***: Pedido de transmissão de dados pelo nó recetor em que o bit fica no estado dominante. Este pedido é enviado a todo o barramento, mas o identificador da mensagem determina qual é o outro nó que deve enviar os dados requeridos pelo primeiro nó;
- ***Identifier Extension (IDE)***: Bit que define o formato da *frame* CAN, se for dominante (“0”), o formato é *standard*;
- ***Reserved bit (r0)***: Bit reservado para futura expansão do barramento, necessita de ser dominante no envio de dados;
- ***Data Length Code (DLC)***: Este código de 4 bits, como o nome indica, pretende informar sobre o número de bytes de dados que fazem parte da mensagem a ser transmitida;
- ***Data***: Valores de variáveis a serem transmitidos, como por exemplo a rotação e temperatura de um motor;
- ***Cyclic Redundancy Check (CRC)+ Delimiter***: O CRC (15 bits) mais o delimitador (1 bit), contêm a soma de verificação do número de bits transmitidos da mensagem anterior para que seja possível detetar se ocorreu erros na transmissão dos dados;

- **Acknowledge (ACK) + Delimiter:** Campo composto por 2 bits recessivos, em que o delimitador é sempre recessivo. O ACK é recessivo no envio de uma mensagem nova por parte do nó emissor, se o nó recetor verificar que a mensagem está correta (através do CRC), coloca o bit do ACK a dominante para que o emissor tenha conhecimento que a mensagem foi recebida com sucesso. Caso o ACK continue a recessivo, o emissor torna a repetir a mensagem assim que possuir permissão [21];
- **End of Frame (EOF):** Campo de 7 bits recessivos que indicam o fim de uma *frame*;
- **Interframe Space (IFS):** Define o tempo necessário entre cada mensagem recebida por cada nó.

Na Figura 3.5 estão representados os campos que constituem uma *frame* de dados numa mensagem CAN para o modo *extended*, também conhecido por CAN 2.0B [16].

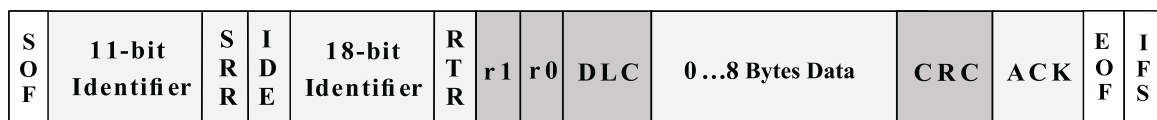


Figura 3.5 – Estrutura de uma *frame* de dados no protocolo CAN *Extended* [17].

Este tipo de formato de *frame* contém alguns campos adicionais, comparativamente ao formato *standard*. Estes campos são:

- **Identifier 2:** Identificador adicional composto por 18 bits;
- **Substitute Remote Request (SRR):** Bit recessivo que é transmitido em CAN 2.0B e substitui o bit RTR em mensagens CAN 2.0A [17];
- **Reserved bit (r1):** Bit reservado do protocolo CAN 2.0B.

CAN Flexible Data-Rate

Devido ao aumento constante da tecnologia e funcionalidades disponíveis no veículo moderno é necessário trocar mais informação e de forma mais rápida. No entanto, o CAN convencional até aqui descrito apenas permite velocidade de comunicação até 1 Mbit/s e transmissão de mensagens com um pacote de dados de 8 bytes. Para dar resposta a esta necessidade, iniciou-se em 2011, o desenvolvimento do protocolo CAN FD, iniciado pela

Bosch® associada a outros fabricantes automóveis, que pretendeu desenvolver um protocolo que consiga estar preparado para o futuro [22].

Pretende-se que a integração deste protocolo no automóvel seja feita de forma simples e progressiva, onde ao longo dos anos o objetivo dos fabricantes é realizar a transição de forma natural de CAN convencional para CAN FD. A implementação deste novo protocolo é relativamente simples, o barramento utilizado é o mesmo, são necessários controladores e *transceiver* CAN distintos e as alterações a nível de *software* são mínimas devido à semelhança do formato da mensagem entre os dois protocolos [23].

Também o formato de mensagens CAN FD divide-se entre CAN *standard*, com um identificador de mensagem de 11 bits e *extended*, com um identificador de 29 bits. A estrutura de uma mensagem *standard* pode ser observada na Figura 3.6.

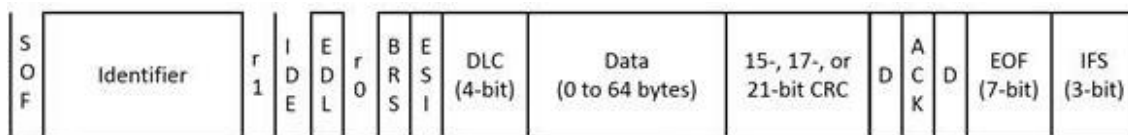


Figura 3.6 – Estrutura de uma *frame* de dados no protocolo CAN FD *Standard* [23].

As semelhanças são muitas entre os dois protocolos e, de modo entender as alterações feitas face ao protocolo de CAN convencional, enunciam-se de seguida os campos adicionais na estrutura de uma mensagem CAN FD:

- **Extended Data Length (EDL):** Bit que define o formato da mensagem CAN FD, *standard* ou *extended*. É recessivo (“1”) se a mensagem for *standard*;
- **r1, r0:** O bit r1 é reservado para futura expansão do barramento, o bit r0 é utilizado para resincronização da mensagem quando existe alteração da velocidade de comunicação. O estado comum é dominante (“0”) para ambos;
- **Bit Rate Switch (BRS):** Bit que define a velocidade dos dados da mensagem em relação à velocidade de configuração da mensagem e pode ser no máximo de 1 Mbit/s. Se for definido como recessivo (“1”) os dados da mensagem serão enviados a uma velocidade diferente, no máximo de 5 Mbit/s [24]. A configuração da mensagem é composta pelos campos à esquerda deste valor que também se encontra incluído;

- **Error State Indicator (ESI):** Bit que traduz a ocorrência de um erro ativo por parte do transmissor. Se for dominante (“0”) o erro é ativo, se estiver em modo de erro passivo é recessivo (“1”);
- **Data Length Code (DLC):** Indica o número de bytes de dados da mensagem. Pode variar entre “1001” e “1111” para indicar mensagens de 12, 16, 20, 24, 32, 48 e 64 bytes;
- **Cyclic Redundancy Check (CRC):** A sua função é igual à do CRC para o CAN convencional, mas o número de bits pode ser de 17 ou 21 no CAN FD.

Uma particularidade interessante é a forma de utilização do campo que define o número de bytes de dados da mensagem, que é composto tal como em CAN convencional, por 4 bits. Até mensagens de 8 bytes o campo DLC apresenta o mesmo comportamento face ao CAN convencional, para uma mensagem de 8 bytes o campo correspondente é de “1000”. Para que o campo DLC mantenha no máximo um tamanho de 4 bits, são utilizados os restantes 7 valores de 9 a 15 para identificação de mensagem de CAN FD [24]. Esta utilização do campo DLC para CAN FD é visível na Tabela 3, onde se faz a comparação entre as duas vertentes de comunicação CAN.

Tabela 3 – Configuração do campo DLC no CAN FD, adaptado de [24].

DLC (bin)	DLC (dec)	Classic CAN	CAN FD
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	8
1001	9	8	12
1010	10	8	16
1011	11	8	20

1100	12	8	24
1101	13	8	32
1110	14	8	48
1111	15	8	64

Em resumo, com a utilização de CAN FD é possível o envio de mais dados por cada mensagem, podendo ir até 64 bytes de dados, a uma velocidade até 5 Mbit/s, dependendo do *hardware* utilizado. O CAN FD funciona em modo convencional caso estejam presentes no barramento nós de CAN convencional, sendo que, sempre que possível, transita para CAN FD. Também os campos de arbitragem, como o ID e *acknowledge* mantêm as velocidades do CAN convencional. Espera-se que a transição definitiva para este novo protocolo seja realizada pelos fabricantes de automóveis a partir de 2020.

3.1.2. Comunicação I2C

Na comunicação de uma unidade de controlo com os seus periféricos é comum a utilização do protocolo de comunicação I2C (*Inter-Integrated Circuit*), este protocolo série é síncrono e *half duplex*. Os nós presentes no barramento podem ser mestres ou escravos, dependendo da funcionalidade que lhes foi atribuída pelo programador. Um nó que seja responsável pelo barramento é considerado como mestre, enquanto que um nó que apenas responda às ordens do mestre é considerado escravo. Um exemplo disto é o de um processador (mestre) e o de um sensor de temperatura (escravo), que apenas fornece o valor de temperatura ao mestre. A sua camada física é composta por dois fios (linhas) que efetuam a troca de informação entre os dispositivos do barramento, a linha SDA (*Serial Data*) que é utilizada para a transferência de dados e a linha SCL (*Serial Clock*), também necessita de existir uma massa comum em todo o barramento para que os dispositivos possam estar referenciados de forma idêntica. A linha SCL é comandada pelo dispositivo mestre e transporta o sinal de relógio de modo a sincronizar os dispositivos no barramento [25]. O destinatário da mensagem enviada é caracterizado por um identificador que está presente na própria mensagem, deste modo apenas o nó correto processa a mensagem que lhe foi endereçada. O identificador de cada nó pode ter um tamanho entre 7 bit e 10 bit dependendo do fabricante e do tipo de dispositivo. Este

identificador é único em cada barramento, mas pode ser alterado caso existam dois nós da mesma referência no barramento.

O protocolo I2C possui diferentes tipos de velocidades de transmissão de dados, sendo que no modo mais rápido (*ultra fast mode*) a frequência de relógio pode ir até aos 5 MHz. Na Figura 3.7 é possível observar a composição de uma rede I2C, para cada barramento, as linhas SDA e SCL necessitam de estar referenciadas à tensão de alimentação do sistema através de uma resistência de *pull-up*.

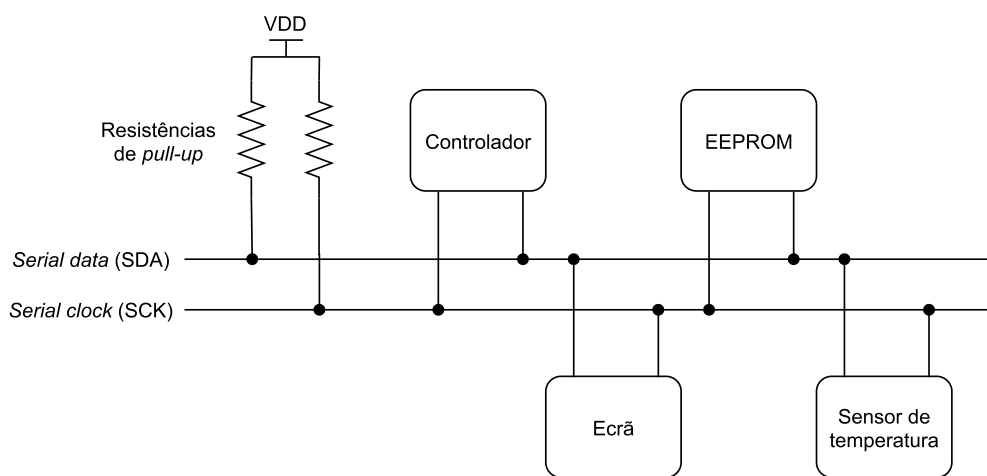


Figura 3.7 – Rede física do protocolo de comunicação I2C, adaptado de [25].

As linhas SDA e SCL no estado de repouso permanecem ambas com um nível lógico alto devido às resistências de *pull-up*. No momento em que o nível lógico do relógio é alto, a linha de dados deve permanecer estável, sendo permitida a comutação da linha de dados apenas quando o nível lógico do relógio é baixo, como se observa pela Figura 3.8.

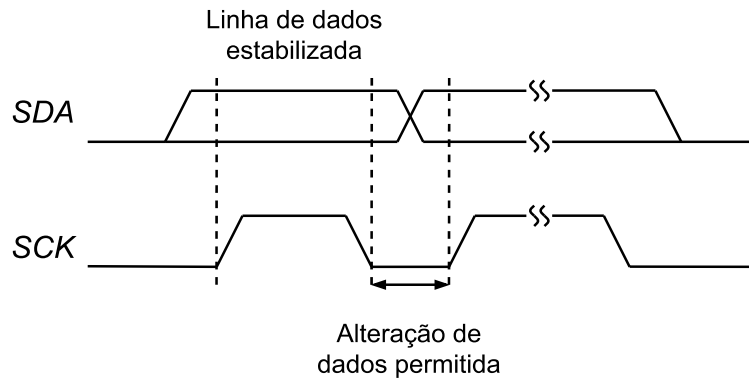


Figura 3.8 – Transferência de dados no barramento I2C, adaptado de [25].

Como a transferência de dados se dá com um nível de relógio alto e o nível lógico da linha SDA estável, o protocolo de comunicação I2C utiliza o nível lógico alto do relógio para estabelecer as condições de início e fim de transmissão de uma mensagem. Como se observa pela Figura 3.9, cada mensagem começa com uma condição de *start* e termina com uma condição de *stop*. Em ambas as condições, a linha de relógio encontra-se com um nível lógico alto. A condição de *start* dá-se com a transição da linha de dados de um nível lógico de alto para baixo e a condição de *stop* é o inverso da primeira condição.

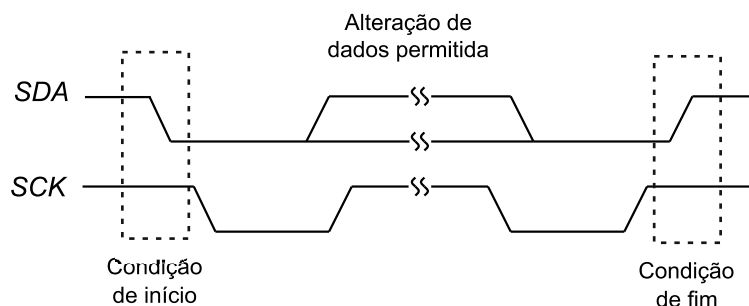


Figura 3.9 – Condições de *start* e *stop* no barramento I2C, adaptado de [25].

3.1.3. Comunicação SPI

A comunicação SPI (*Serial Peripheral Interface*) também é habitualmente utilizada para comunicação entre um microcontrolador e diferentes periféricos. É um protocolo síncrono série, *full duplex*, composto por um mestre que é responsável por gerar o sinal de relógio e por um ou vários escravos. Nesta comunicação são utilizadas duas linhas separadas, para envio e receção de dados, respetivamente, permitindo que os dados percorram o barramento nas duas

direções de forma simultânea. A linha utilizada pelo mestre para transmissão de dados é denominada de MOSI (*Master Output/Slave Input*) e para que o mestre receba dados dos escravos é utilizada a linha MISO (*Master Input/Slave Output*) [26]. Na Figura 3.10 demonstra-se uma configuração onde são utilizados múltiplos escravos, em que o mestre utiliza a linha de CS (*Chip Select*) para seleccionar com qual dos escravos pretende comunicar, esta linha é normalmente ativa a “0”.

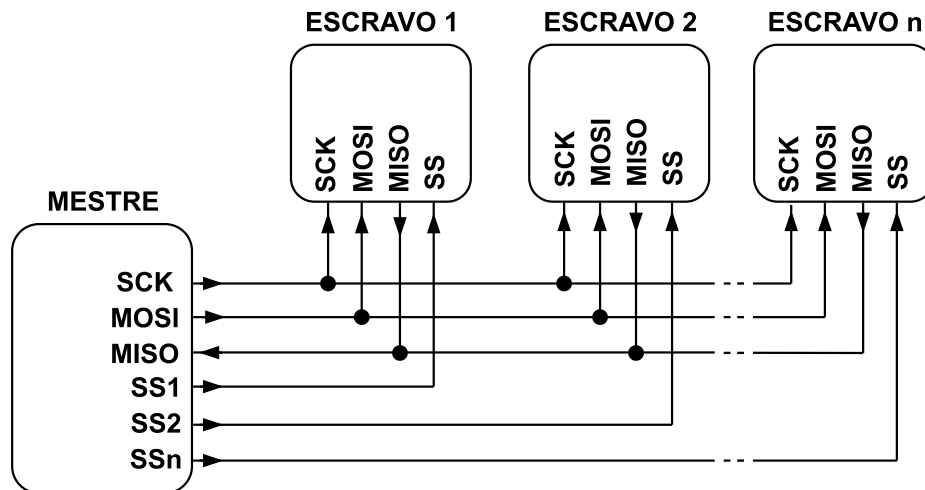


Figura 3.10 – Comunicação SPI com múltiplos escravos, adaptado de [27].

Este tipo de configuração é ideal para aplicações onde se utilizam poucos escravos pois, como facilmente se percebe, o número de terminais utilizados no microcontrolador cresce com o aumento de escravos utilizados. Em situações em que existem muitos escravos pode-se utilizar uma configuração do tipo *daisy-chain*, onde apenas se utiliza um pino de CS para múltiplos escravos que recebem simultaneamente o sinal de relógio. Na Figura 3.11 observa-se uma configuração *daisy-chain*, onde o mestre envia dados ao primeiro escravo que os transmite ao próximo escravo e assim sucessivamente. Apenas o último escravo envia os seus dados e os dos escravos anteriores para o mestre, para tal é necessário que o sinal de relógio esteja adequado ao número de escravos disponíveis e que exista deslocamento dos dados transmitidos.

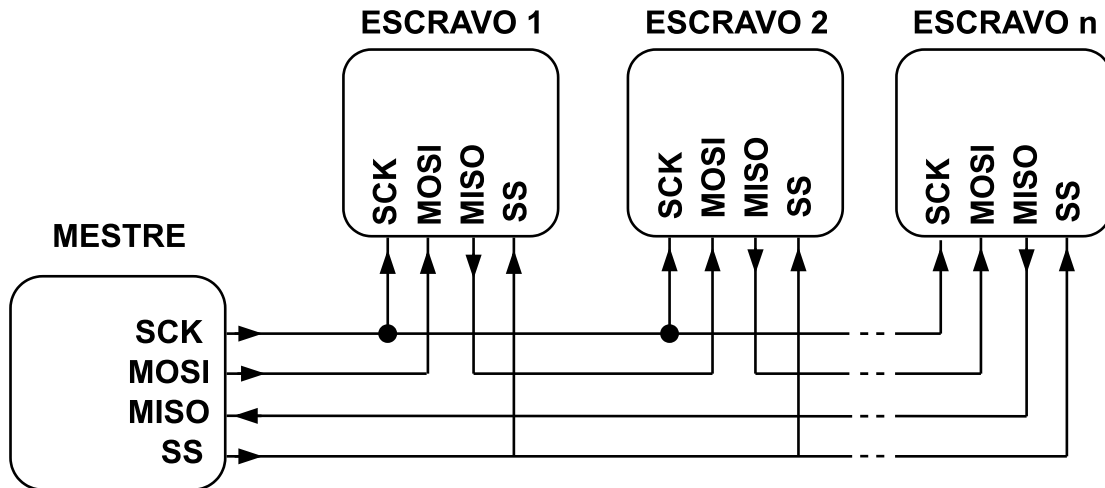


Figura 3.11 – Configuração SPI do tipo *daisy-chain*, adaptado de [27].

Na Figura 3.12 demonstra-se a comunicação entre um mestre e um escravo, onde após a colocação da linha de CS a nível lógico baixo, se inicia o sinal de relógio. De seguida o mestre inicia a transmissão de dados, solicitando informação ao escravo. O escravo após receber a mensagem do mestre, envia a informação solicitada ao mestre antes deste efetuar a desativação da linha CS. A leitura dos dados é feita na transição de sinal do relógio, podendo ser feita no bordo ascendente ou descendente dependendo da escolha do programador. A maioria dos escravos utilizados em SPI são periféricos com necessidades elevadas de taxas de comunicação, como por exemplo EEPROM's (*Electrically Erasable Programmable Read Only Memory*), ADC's e sensores.

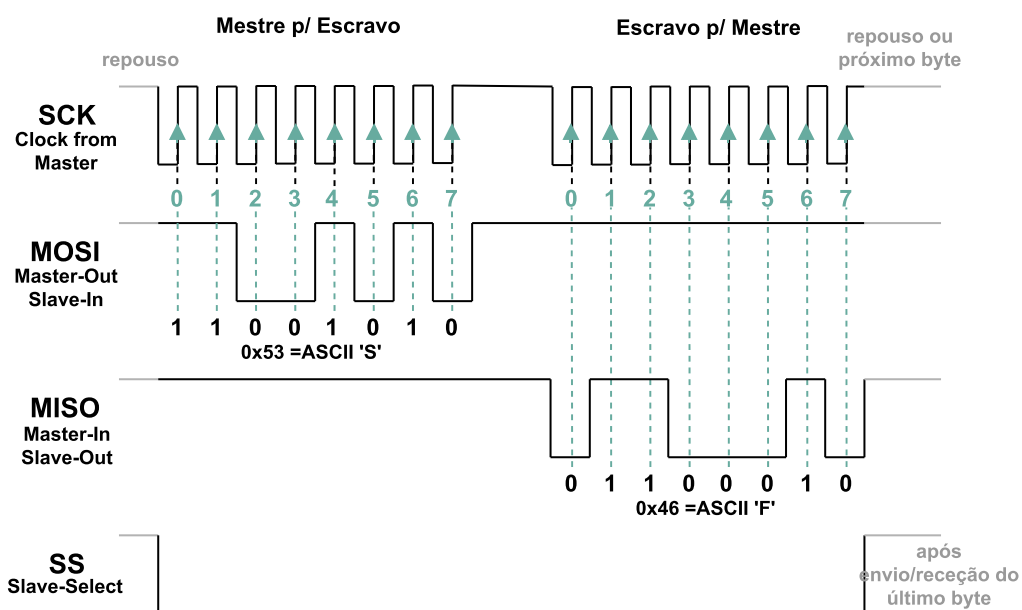


Figura 3.12 – Diagrama de comunicação SPI mestre/escravo, adaptado de [27].

3.1.4. Comunicação Série

A comunicação série pode ser utilizada para transmitir dados entre dispositivos, como um computador e uma unidade de controlo eletrónico. Ao contrário da comunicação paralela, neste tipo de comunicação é transmitido um bit de cada vez. A velocidade de comunicação é inferior à alcançada com a comunicação paralela, mas o uso de apenas uma linha de dados em vez de várias linhas de dados para a transmissão de uma mensagem pode ser vista como uma vantagem em certas aplicações.

A comunicação série pode ser síncrona, através da utilização de uma linha de relógio ou assíncrona, onde não se utiliza relógio. De seguida é feita uma análise apenas à comunicação série assíncrona por ser a de maior interesse para o presente projeto. A comunicação assíncrona permite assim simplificar o protocolo de comunicação, no entanto necessitam de existir outros métodos de sincronização da transmissão de dados para garantir que não ocorrem erros.

Estes métodos são específicos para a comunicação série assíncrona e estão presentes dentro da própria mensagem transmitida. São para isso incorporados na mensagem um bit de início e de fim de transmissão de dados, conhecidos como bits de sincronização, e um bit de paridade para deteção de erro de transmissão. Sendo que não existe sinal de relógio é fundamental que a velocidade de transmissão de dados, conhecida como *baudrate* e que expressa o número de bits por segundo (bps) transmitidos na linha de dados, seja igual nos dois dispositivos que estão a comunicar. De outra forma os dispositivos ficam fora de sincronismo e os dados transmitidos são corrompidos.

A transmissão de uma mensagem pode ser observada na Figura 3.13 onde o primeiro bit define o início de transmissão da mensagem, que ocorre na transição da linha de dados de um estado de repouso de nível lógico alto para um nível lógico baixo. De seguida são enviados os 8 bits de dados da mensagem (01010101), o primeiro bit a ser enviado é o bit menos significativo (LSB - *Least Significant Bit*) e o último bit é o mais significativo (MSB - *Most Significant Bit*). A condição de fim da mensagem é definida pelo último bit ou em certas aplicações através dos dois últimos bits, e caracteriza-se pela transição da linha de dados de um nível lógico baixo para o estado de repouso de nível lógico alto [28].

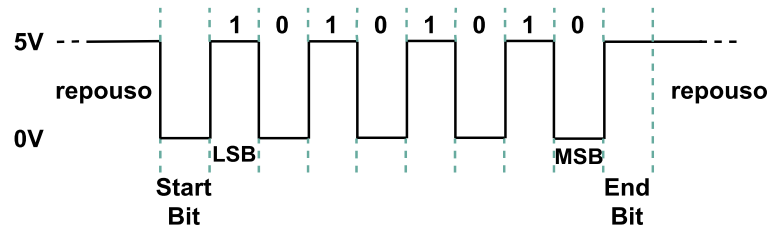


Figura 3.13 – Transmissão de dados de comunicação série, adaptado de [28].

Para comunicação entre dois dispositivos, onde ambos transmitem e recebem dados, são necessárias apenas 3 linhas de comunicação. A linha de transmissão (TX), a linha de receção (RX) e a massa (GND). De referir que numa ligação série só existe troca de dados entre dois dispositivos. A ligação TX de um dispositivo necessita de ser conectada à ligação RX do outro dispositivo e o inverso também se aplica. As massas destinam-se a definir a tensão de referência de 0 V entre os dois dispositivos.

3.2. Arquitetura de código

À semelhança de outros módulos eletrónicos, a ECU, desempenha diversas tarefas, é necessário adquirir informação de sensores, efetuar o processamento dos sinais adquiridos e proceder à atuação de componentes. A metodologia adotada no desenvolvimento dos algoritmos implementados na ECU depende da arquitetura de *software* definida. Deste modo é feita de seguida uma análise a dois tipos comuns de arquitetura de código utilizadas em sistemas embebidos.

3.2.1. Superloop

Este tipo de arquitetura é normalmente utilizada em aplicações onde o funcionamento de todo o sistema seja previsível. Pode ser descrito como um ciclo sem fim, onde neste são executadas uma série contínua de tarefas e no final do ciclo a primeira tarefa volta novamente a ser executada. Um exemplo de código para este tipo de arquitetura pode ser observado na Figura 3.14. As aplicações desta arquitetura destinam-se a sistemas onde as tarefas a desempenhar sejam poucas, os algoritmos desenvolvidos sejam de baixa complexidade e onde possam existir recursos de memória limitados. No entanto, não é possível prever o tempo decorrido em cada tarefa, o que o torna utilizável apenas em sistemas em que o tempo não seja uma variável crítica. O processador, uma vez que está sempre a desempenhar tarefas, é utilizado por completo, o que não permite a utilização de um modo eficiente de poupança de energia. Existe ainda a dificuldade na deteção de eventos externos.

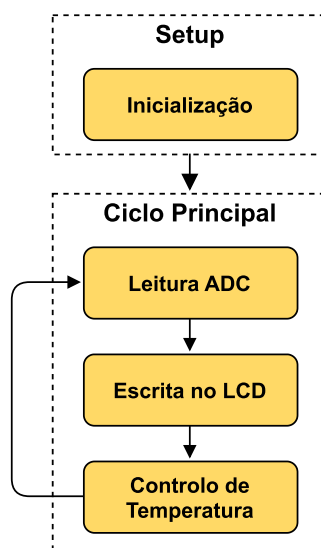


Figura 3.14 – Exemplo de uma arquitetura do tipo *Superloop*.

3.2.2. Foreground/Background Systems

A maioria das aplicações que fazem uso de um microcontrolador para executar tarefas de forma autónoma utiliza a arquitetura *Foreground/Background*. Esta arquitetura consiste num ciclo infinito principal do tipo *Superloop*, como detalhado na secção anterior 3.2.1, que se designa por *Background* e onde as tarefas de menor prioridade são executadas [29]. As interrupções são tratadas nas rotinas de serviço correspondente ao tipo de prioridade das mesmas, estas rotinas são chamadas de *Foreground*, onde se encontram as tarefas consideradas críticas. Como se observa pela Figura 3.15 é possível verificar que todas as tarefas inseridas no *Background* têm a mesma prioridade, se ocorrer um atraso numa destas funções, também o resto do ciclo vai ser prejudicado. Uma vez que o tempo total de execução destas tarefas não é constante, é necessário que existam estratégias que permitam monitorizar o tempo de execução da tarefa de forma a garantir o correto processamento de todo o código.

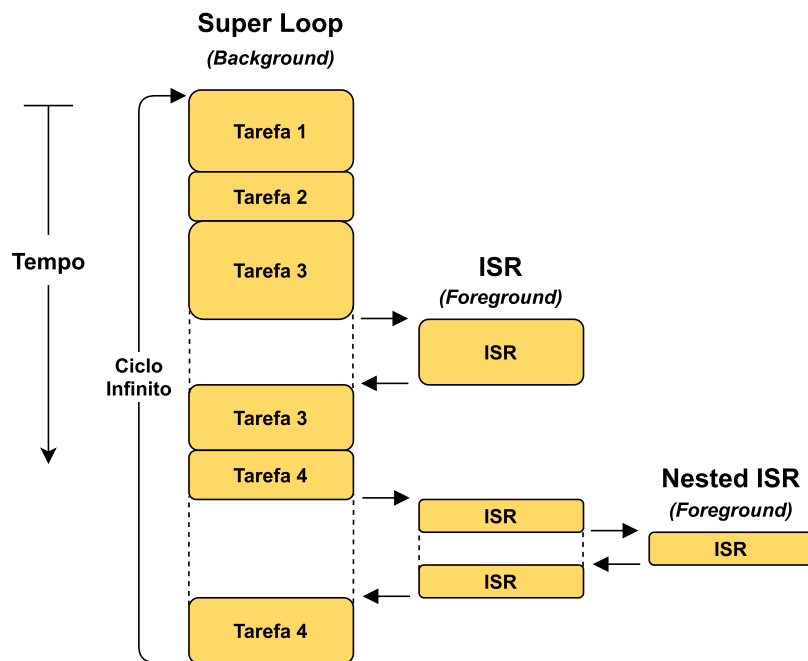


Figura 3.15 – Arquitetura Foreground/Background, adaptado de [29].

4. MeaTec32

Neste capítulo é detalhado de forma sucinta, o projeto e execução da ECU proposta, batizada de MeaTec32. São apresentadas as especificações que caracterizam a MeaTec32 e abordados os módulos que constituem este projeto. Para cada módulo da MeaTec32 são apresentados os circuitos elétricos desenvolvidos e as características dos componentes utilizados. Por fim referem-se os testes realizados a alguns componentes, bem como todo o desenvolvimento e montagem da placa de circuito impresso.

4.1. Requisitos e especificações de uma ECU

Após a análise da oferta existente de ECU's para veículos elétricos utilitários, realizada na secção 2.3, foi possível identificar quais os requisitos necessários que uma ECU deve preencher para este tipo de veículos. Primeiramente é importante realçar a necessidade de baixo custo, pois num veículo que é de custo reduzido, a ECU não pode contribuir significativamente para o investimento total.

A gestão de múltiplas tarefas que é realizada pela ECU de um veículo elétrico utilitário, obriga à utilização de um microcontrolador com um desempenho médio/alto que consiga acompanhar a elevada taxa de aquisição e tratamento de dados inerentes ao funcionamento deste tipo de veículos. A utilização de um microcontrolador que permita uma programação simples para o utilizador, mantendo a hipótese de escalar a complexidade da programação desenvolvida, deve ser disponibilizada. Deste modo não se restringe a utilização da ECU a sistemas simples, garantindo a flexibilidade em adaptar a ECU a sistemas de maior complexidade.

Como se verificou na Figura 2.3, a arquitetura do veículo elétrico pode ser dividida em três subsistemas distintos, nomeadamente a propulsão elétrica, a fonte de energia e os sistemas auxiliares. Como tal, é necessário que exista no mínimo uma *interface* de comunicação para troca de informação dentro de cada subsistema, pois cada um apresenta uma prioridade diferente e deve ser tratado como tal. O tipo de comunicação deve ser de fácil implementação, de elevada velocidade, robustez e de baixo custo, sendo a comunicação CAN o protocolo que preenche estes requisitos.

A existência de entradas e saídas é considerado um requisito básico, no entanto, tem de existir uma relação de compromisso face ao preço que esta funcionalidade pode acarretar. O veículo elétrico utilitário necessita destas entradas e saídas para aquisição e controlo de sistemas auxiliares, mas a verdade é que por vezes a maioria das ofertas disponibilizadas não justifica o investimento que é feito. Na possibilidade de falta de entradas e se for justificável, existe sempre a hipótese de aquisição de módulos adicionais de CAN que permitem expandir este requisito.

A capacidade de diagnóstico de cargas de potência deve ser disponibilizada não só para segurança do próprio sistema, mas também para auxílio do programador na implementação de uma ECU deste tipo num veículo elétrico utilitário. A utilização de funcionalidades que garantam a segurança do sistema, tais como o isolamento elétrico da ECU em relação aos sistemas de alta tensão, a proteção do sistema de alimentação, entre outros, também deve ser tomada como um requisito essencial.

A implementação desta ECU em veículos elétricos utilitários, muitas das vezes em veículos ainda em fase de protótipo, requer que a utilização desta seja simples. Deve ser compatibilizada a programação e implementação desta ECU com plataformas *open source*. A simplicidade da ECU pode atrair e beneficiar utilizadores menos experientes que numa fase inicial não se sintam confortáveis com a utilização de sistemas mais complexos.

Especificações da MeaTec32

Na secção 2.3 verificou-se que qualquer uma das ECU's apresentadas pode ser aplicada a um veículo elétrico utilitário, sendo que como se demonstrou, a ECU TTCControl[®] HY-TTC-30H e a ECU GEVCU 6.22 podem ser consideradas as escolhas mais acertadas para a gestão de um veículo deste tipo. A primeira ECU referida apenas possui uma *interface* CAN e não é *open source*, enquanto que a segunda ECU referida também só dispõe de duas *interface* CAN e não possibilita diagnóstico nas suas saídas. Existe assim espaço para melhorias, que possam ser providenciadas a um preço semelhante ou inferior. O desenvolvimento de uma ECU para este tipo de veículos, que preenche a lacuna aqui apresentada, é uma possibilidade que é respondida através da realização do presente projeto.

Através da análise do conjunto de requisitos de uma ECU para um veículo elétrico utilitário, foi possível estabelecer um conjunto de especificações para a ECU desenvolvida no presente projeto. De seguida apresentam-se as especificações definidas para a MeaTec32:

Microcontrolador:

- Microchip® SAM3X8E 32 bit de 84 MHz;

Entradas:

- 4 entradas digitais isoladas, com medição de frequência de 5 Hz a 20 kHz;
- 2 entradas analógicas isoladas de 0 a 5 V;
- 4 entradas analógicas não isoladas de 0 a 3.3 V, com possibilidade de utilização de 2 entradas para leitura de sensores resistivos de temperatura;

Saídas:

- 4 saídas digitais isoladas de 7 A do tipo *high side* com monitorização de corrente de carga e configurável como saída PWM de frequência máxima de 200 Hz;
- 1 saída digital isolada de 6 A do tipo *high side*, configurável como saída PWM de frequência máxima de 6 kHz;
- 1 saída digital isolada de 6 A do tipo *low side*, configurável como saída PWM de frequência máxima de 6 kHz;

Interface:

- Porta USB para programação;

Comunicação:

- Comunicação série com linhas de dados isoladas;
- 2 canais nativos de CAN isolados;
- 1 canal de CAN FD não isolado;

Software:

- Arduino IDE/Eclipse IDE com programação em C;

Outros:

- Monitorização da temperatura interna do microcontrolador;
- EEPROM de 512 Kb;
- Alimentação de sensores de 3.3 V até 300 mA e de 5 V até 1000mA.

Para enquadramento das especificações aqui descritas, encontra-se na Figura 4.1 o diagrama de blocos da MeaTec32 desenvolvida.

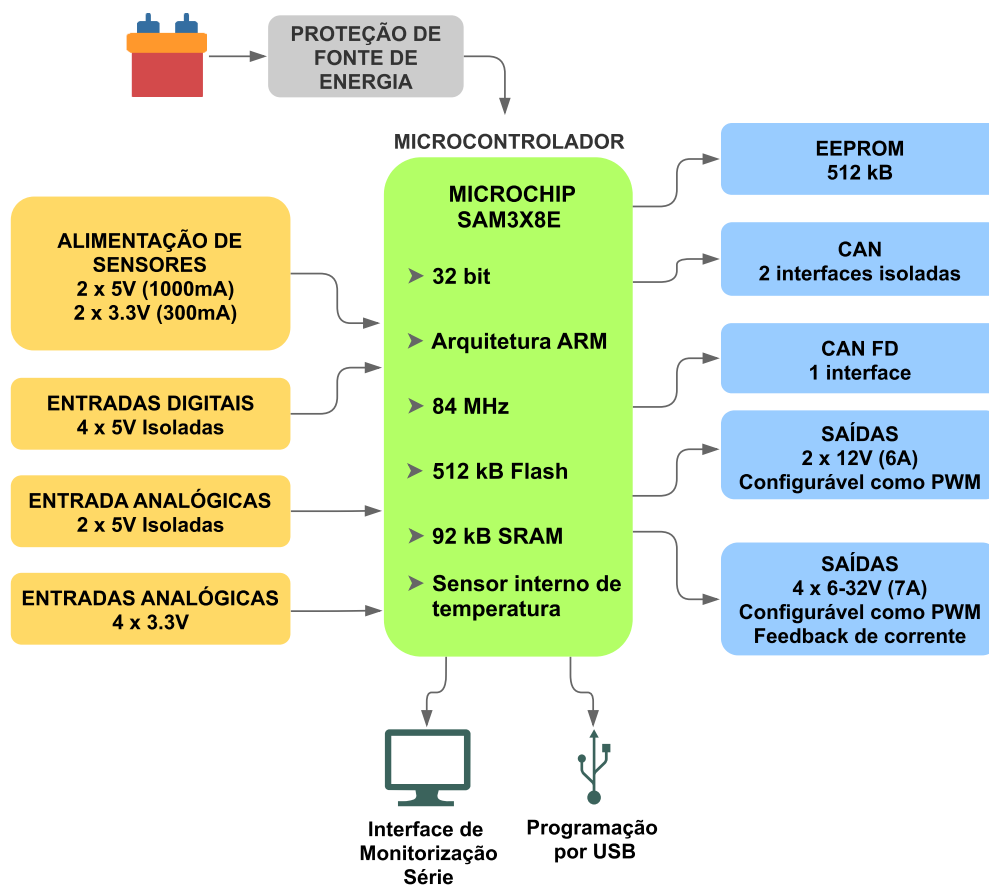


Figura 4.1 – Diagrama de blocos da MeaTec32.

Para a análise da MeaTec32 face às ECU's aqui já apresentadas, observa-se na Tabela 4 uma análise comparativa semelhante à já demonstrada na Tabela 1. A comparação é realizada entre a MeaTec32 e as duas ECU's de menor custo anteriormente analisadas, a TTControl[®] HY-TTC-30H e a ECU GEVCU 6.22. A classificação segue as mesmas diretivas anteriormente apresentadas.

Tabela 4 – Caracterização das unidades comercialmente disponíveis.

Funções	TTCONTROL HY-TTC 30	GEVCU 6.22	MeaTec32 1.0
Processamento	Infineon 32bit	Atmel 32bit	Atmel 32bit
	80MHz	84MHz	84MHz
	82kB RAM	96kB SRAM	96kB SRAM
	768kB Flash	512kB Flash	512kB Flash
Comunicações	1 CAN	2 CAN	2 CAN 1 CAN FD
	8 An 0-5V/10V 2 An 0-32V 4 Dig/Freq Input 0-32V 16 Dig Output	8 An 0-12V 4 Dig Input 0-15V 8 Dig Output 0-12V	2 An 0-5V 4 An 0-3.3V 4 Dig Input 0-5V 2 Dig Output 0-12V 4 Dig Output 6-32V
Diagnóstico	6 Dig Output	N/D	4 Dig Output
Simplicidade	C Language	C Language	C Language
Preço	1 800 €	535 €	160 €

Com base nestes dados, pode-se observar na Figura 4.2 a comparação da MeaTec32 com as outras duas unidades aqui abordadas. Verifica-se que na categoria de entradas/saídas, a MeaTec32 possui uma classificação inferior às restantes ECU's, devido ao compromisso tomado entre o custo e a existência de entradas e saídas, salientado na abordagem deste requisito na secção 4.1. Em todas as restantes categorias, verifica-se que a MeaTec32 possui um desempenho global igual e na maioria das vezes superior face às restantes ECU's. Analisando o processamento verifica-se que, para a mesma arquitetura de 32 bit, o processador da MeaTec32 trabalha uma frequência superior à frequência do processador da TTControl®. Através da análise da quantidade de instruções realizada por segundo, também conhecida por MIPS (*Millions of Instructions Per Second*), é possível comparar o desempenho destes dois processadores. Verificou-se que o processador da TTControl® é classificado como 80 MIPS [30], enquanto que o processador da MeaTec32 possui uma classificação de 105 MIPS [31], sendo assim superior, conseguindo efetuar até 105 milhões de instruções por segundo.

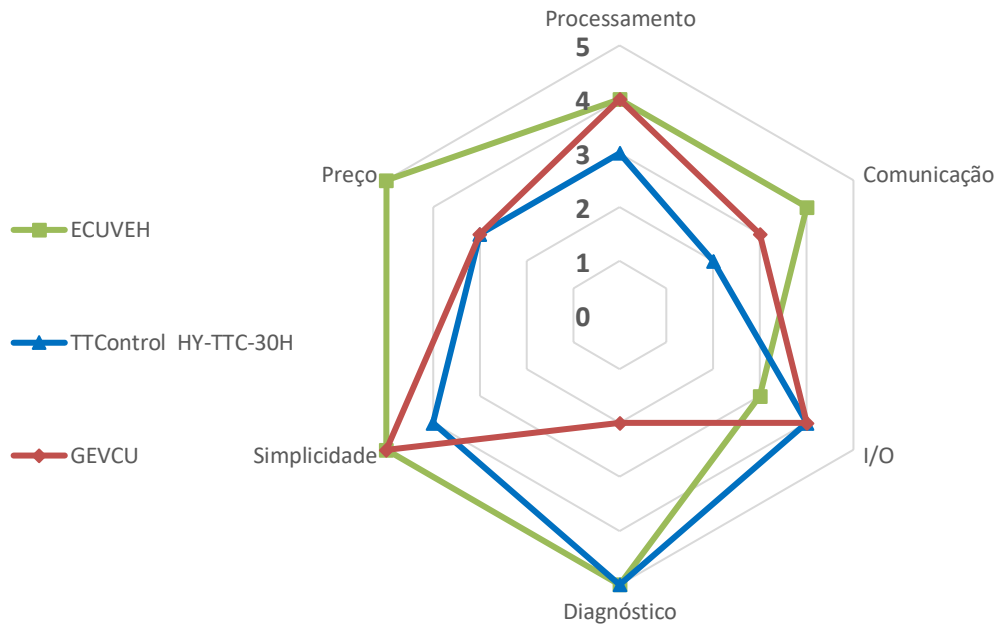


Figura 4.2 – Avaliação da MeaTec32.

4.2. Módulos da MeaTec32

Nesta secção descrevem-se as funcionalidades dos módulos que constituem a MeaTec32, a escolha de componentes, passando por detalhar o circuito elétrico da MeaTec32 e a função dos seus componentes.

Microcontrolador

Para cumprir os requisitos das unidades atuais presentes no mercado de veículos utilitários, a escolha do processador reincidiu sobre o microcontrolador SAM3X8E desenvolvido pela Microchip®. Este microcontrolador baseia-se numa arquitetura ARM (*Advanced RISC Machine*) de 32 bit e com uma frequência de trabalho de 84 MHz, o que permite processar elevadas quantidades de informação sem perda de desempenho. Possui também uma memória de programa *flash* de 512 Kbytes e uma memória estática de 100 Kbytes, a par da maioria das unidades semelhantes no mercado. Integra periféricos tais como: *Ethernet*, 2 canais CAN

nativos, USB de alta velocidade (480 Mbps), 16 canais de conversão analógico-digital com resolução de 12 bit, entre outros. A escolha deste microcontrolador permite atingir um desempenho elevado, mas mantendo a unidade acessível ao utilizador mediano. Sendo este microcontrolador utilizado idêntico ao utilizado na conhecida plataforma Arduino® Due, acrescenta à MeaTec32 a capacidade de programação e utilização de bibliotecas *open source* já existentes. Deste modo, facilmente se consegue reproduzir as aplicações desenvolvidas pelo utilizador, pois a comunidade *open source* já se encontra familiarizada com este microcontrolador.

Regulação de tensão

O módulo de regulação de tensão da MeaTec32 efetua o controlo de alimentação de todos os componentes e está localizado na própria placa de circuito impresso, também chamada de PCB (*Printed Circuit Board*). Como se verifica na Figura 4.3, numa unidade de controlo eletrónico automóvel, a tensão de alimentação da bateria de 12 V necessita de ser adaptada a diversos blocos de alimentação, normalmente alcançado através do uso de vários reguladores de tensão. As cargas genéricas como lâmpadas, bombas de água, entre outros, são alimentadas pelo bloco de 12 V que é destinado a cargas de potência. Através da alimentação de 12 V da bateria, é regulada a tensão para a alimentação de sensores a 5 V, como sensores de posição do pedal acelerador, sensores de temperatura, entre outros. A alimentação de 5 V é também utilizada para alimentar outros componentes como isoladores digitais ou controladores CAN. Para a parte de controlo da MeaTec32 é realizada a adaptação de tensão para 3.3 V que se destina a componentes como o microcontrolador, armazenamento de dados, entre outros [32].

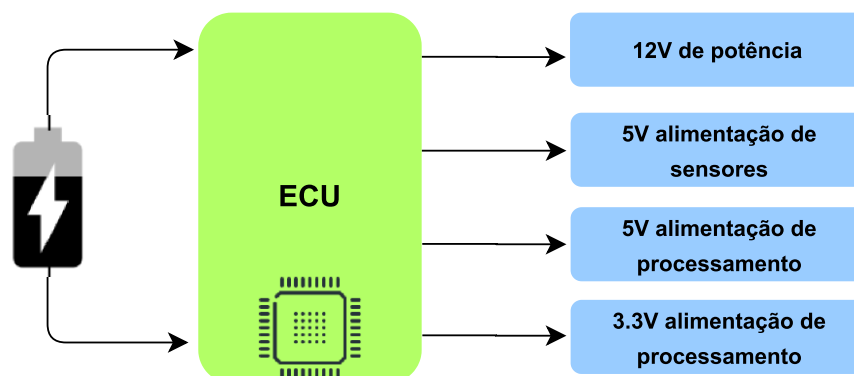


Figura 4.3 – Bloco de alimentação de uma unidade de controlo eletrónico automóvel.

A seleção de componentes disponíveis no mercado foi feita com base nos fatores e requisitos adequados à MeaTec32. O custo é dos fatores que maior peso tem sobre a escolha dos reguladores de tensão, pois um dos requisitos da MeaTec32 é que seja de baixo custo. Uma vez que irá estar integrada num veículo automóvel, a ocorrência de curto-circuitos e alimentações invertidas são uma possibilidade. Como tal, os reguladores de tensão devem possuir funções de deteção de curto-circuito, desativação de alimentação ou recuperação automática se for possível. A eficiência dos componentes também é dos fatores cruciais para a escolha realizada. A utilização de reguladores de tensão comutados permite uma maior eficiência, de forma a contribuir para o controlo da temperatura de funcionamento da MeaTec32.

O microcontrolador SAM3X8E trabalha a um nível de tensão de 3.3 V [33] e uma vez que a maioria dos veículos possuem um sistema de 12 V, é necessário regular o nível de tensão fornecida à parte de controlo. Como tal são utilizados dois reguladores de tensão comutados, numa primeira fase a tensão é regulada de 12 V para 5 V com recurso ao regulador de tensão com referência P78E05-1000 da CUI Inc[®]. Na Figura 4.4 encontra-se representado o circuito elétrico responsável pela regulação de tensão, onde a saída do regulador (5 V) não se encontra isolada da entrada (12 V) pois não existe essa necessidade.

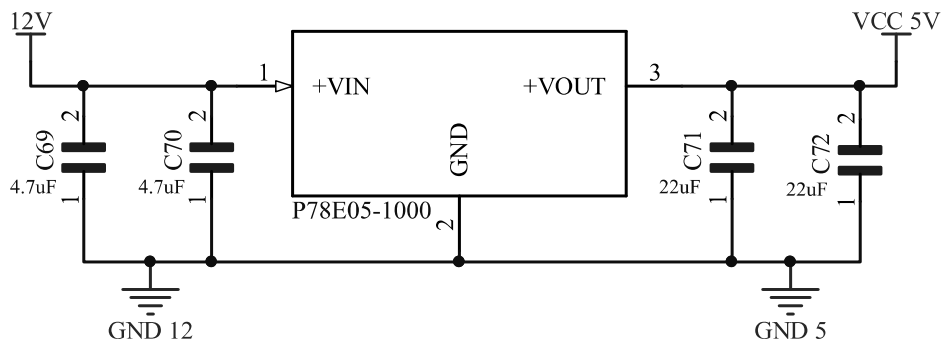


Figura 4.4 – Regulador de tensão de 5V.

De seguida a tensão é regulada para 3.3 V com a utilização do regulador de tensão de referência PDS1-S5-S3-S da CUI Inc[®], como se pode observar na Figura 4.5. Este regulador possui isolamento entre a entrada e a saída, para que toda a parte de controlo do módulo que opera a 3.3 V fique isolada da parte de potência que trabalha a 5 V e 12 V. A utilização de

isolamento permite manter a tensão de trabalho do microcontrolador imune ao ruído, que é crítico em unidades de controlo de veículos.

Para a existência de compatibilidade eletromagnética, conhecida como EMC (*Electromagnetic Compatibility*) [34], o *datasheet* do regulador de tensão de 3.3V recomenda a utilização do circuito que se observa na Figura 4.5.

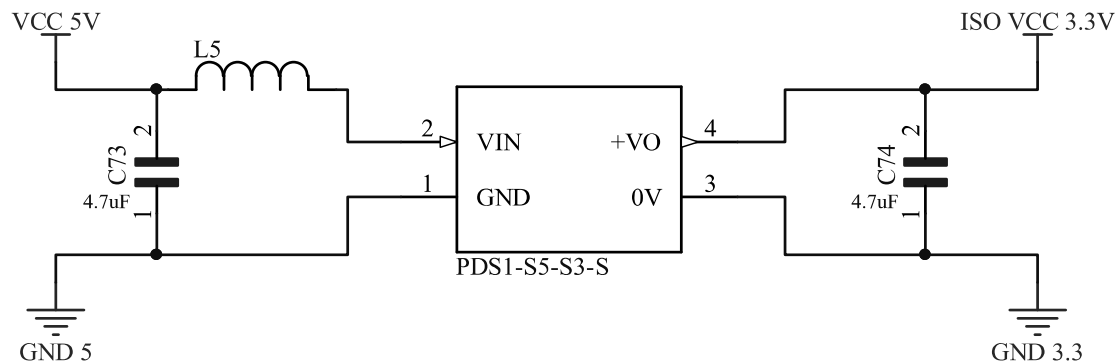


Figura 4.5 – Regulador de tensão de 3.3V.

Entradas digitais

As entradas digitais destinam-se à leitura de sinais digitais que podem assumir dois valores, valor lógico baixo (“0”) e valor lógico alto (“1”). Necessitam na maioria das vezes de ser referenciadas à tensão de alimentação do microcontrolador da unidade, utilizando resistências de *pull-up*. Em aplicações onde o microcontrolador da unidade de controlo trabalha a 3.3 V, a leitura de entradas digitais não é realizada por este componente devido às diferenças na tensão de trabalho. Para leitura de entradas digitais de por exemplo 5 V, é necessário utilizar um componente intermédio, de preferência isolado entre a entrada e a saída, que realiza a adaptação de sinais de 5 V para a tensão de trabalho do microcontrolador.

Para a realização desta tarefa na MeaTec32, selecionou-se o isolador digital de referência ISO7740DWR da Texas Instruments®, na Figura 4.6 observam-se as ligações deste componente. A escolha recaiu sobre este isolador digital devido ao relativo baixo custo quando comparado com outras opções do mercado, conseguindo manter as funções base requeridas para a leitura de sinais digitais. Integra 4 canais digitais unidireccionais, o que permite integrar este módulo num único componente e assim poupar espaço. Possui também isolamento galvânico entre as entradas e saídas, garantido que a adaptação de sinais é feita de

forma segura. A tensão máxima admissível por este componente nas suas entradas é de 6 V, permitindo assim a leitura contínua de forma segura para sinais de 5 V, valor de tensão disponibilizado pelo regulador de tensão de 5 V.

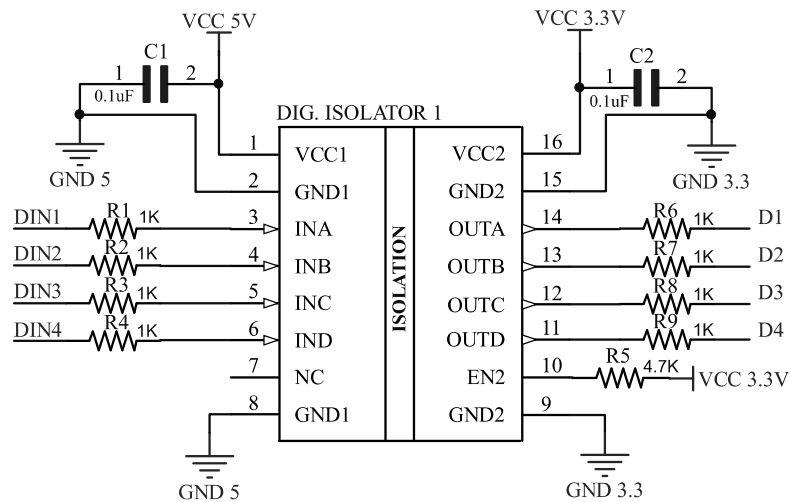


Figura 4.6 – Entradas digitais.

Entradas analógicas

As entradas analógicas permitem a leitura de sinais analógicos para monitorização de sensores tais como sensores de temperatura, sensores de pressão, entre outros. Num veículo, a ocorrência de ruídos é comum, bem como a possibilidade da existência de interferências eletromagnéticas que podem comprometer a leitura de sinais analógicos. De modo a isolar eletricamente e também combater ruído e interferências eletromagnéticas que possam ocorrer, utilizou-se em duas das entradas analógicas, amplificadores isolados. A escolha deste componente foi feita com base no baixo preço de custo, relativamente à funcionalidade que apresenta. O componente escolhido foi o amplificador isolado de referência AMC1211AQDWVRQ1 da Texas Instruments®, que possui uma barreira de isolamento entre a sua entrada e saída, permitindo ser resistente ao ruído e interferências eletromagnéticas.

A MeaTec32 possui 6 entradas analógicas, 4 das quais não isoladas, onde a tensão máxima permitida é de 3.3V, e as 2 entradas restantes são isoladas dos 5 V. De seguida detalham-se a componente de entradas analógicas isoladas e de entradas analógicas não isoladas.

Entradas Analógicas Isoladas

De acordo com o *datasheet* do amplificador isolado AMC1211AQDWVRQ1, a tensão máxima permitida à entrada do terminal VIN é de 2 V [35]. No entanto, o módulo foi desenvolvido para suportar a leitura de sinais analógicos até 5 V em duas entradas (A1 e A2). Desta forma realizou-se, numa primeira fase, a adaptação do sinal máximo de 5 V (A1) para uma tensão máxima de 1.9 V (VIN) através de um divisor de tensão, onde a expressão matemática é dada pela Equação 4.1:

$$V_{in} [V] = \frac{R_{11}}{R_{11} + R_{10}} \times A1 \quad (4.1)$$

A tensão de saída do amplificador isolado é apresentada de forma diferencial e na Figura 4.7 analisa-se a relação entre a tensão de entrada e a tensão de saída diferencial deste componente. A tensão de saída encontra-se disponível nos terminais VOUTN e VOUTP, como se verifica pela Figura 4.8. Analisando a Figura 4.7 verifica-se, com a ajuda das linhas traçadas a azul que, para uma tensão máxima de 1.9 V à entrada do amplificador, VOUTP assume o valor de 2.4 V e VOUTN assume o valor de 0.5 V.

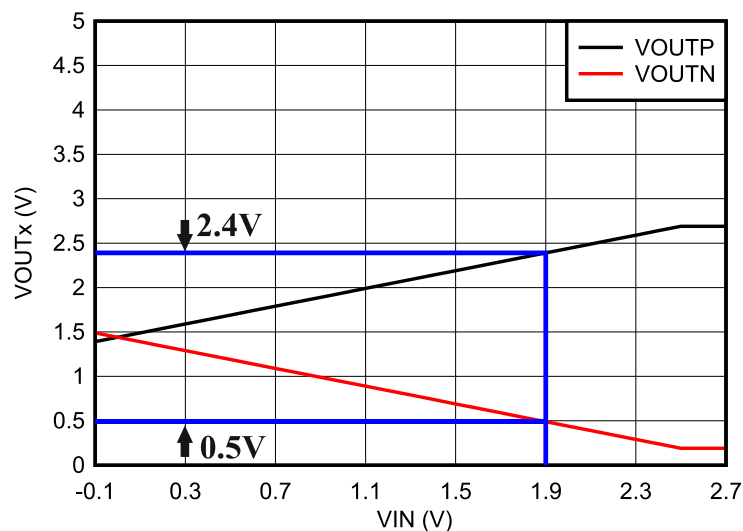


Figura 4.7 – Tensão de saída em função da tensão de entrada [35].

De modo a converter a saída diferencial do amplificador isolado num único valor de tensão que o microcontrolador consiga receber num só pino analógico, recorreu-se ao amplificador

de instrumentação INA826AID da Texas Instruments®. Como se verifica pela Figura 4.8, converte os valores da saída do amplificador isolado VOUTN1 e VOUTP1, na entrada analógica AN1 através da subtração destes dois valores. Este amplificador de instrumentação realiza ainda a adaptação do valor de tensão máximo de 1.9 V para perto de 3.3 V, que é o valor máximo permitido à entrada do pino analógico do microcontrolador. O ganho (G) deste amplificador de instrumentação pode ser determinado pela Equação 4.2. Com uma tensão máxima permitida no microcontrolador ($V_{max_{uc}}$) de 3.3 V e uma tensão máxima à saída deste amplificador ($V_{max_{in}}$) de 1.9 V, obtém-se um ganho de 1.74.

$$G = \frac{V_{max_{uc}}}{V_{max_{in}}} \quad (4.2)$$

Por fim, o valor analógico (AN1), correspondente à saída do amplificador de instrumentação e que é adquirido pelo microcontrolador, pode ser traduzido pela seguinte equação 4.3. Com um ganho de 1.74 e com o valor máximo de VOUTP1 de 2.4 V e VOUTN de 0.5 V, como observado anteriormente na Figura 4.7, AN1 fica com um valor máximo de 3.3 V.

$$AN1 [V] = G \times (VOUTP1 - VOUTN1) \quad (4.3)$$

Na Figura 4.8 encontra-se o circuito elétrico utilizado neste tipo de entradas analógicas, onde se observa a integração do amplificador isolado AMC1211AQDWVRQ1 e do amplificador de instrumentação INA826AID. A escolha deste último componente deve-se ao facto de ser da mesma família que o amplificador isolado e ser de baixo custo, conseguindo manter características como baixo consumo de corrente a utilização de uma única resistência externa para definição do ganho.

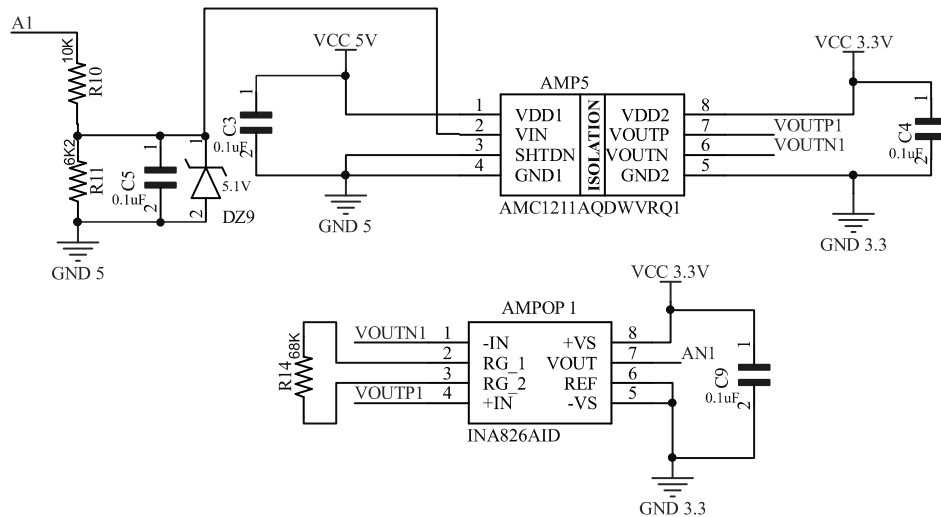


Figura 4.8 – Entrada analógica isolada.

Entradas Analógicas Não Isoladas

Como referido na secção 4.1 existem 4 entradas não isoladas para sensores alimentados a 3.3 V e que não necessitam de isolamento relativamente à zona de processamento da MeaTec32. Apesar das entradas do microcontrolador se encontrarem protegidas contra sobretensões através de díodos de Zener, é necessário ter precaução de modo a não afetar o microcontrolador. De forma a aumentar a versatilidade deste módulo, é possível utilizar 2 das entradas analógicas não isoladas para leitura de sensores de temperatura do tipo termístores, normalmente utilizados na medição de temperatura em veículos automóveis. Observa-se na Figura 4.9 a representação de uma leitura deste tipo, mais especificamente de um sensor de temperatura NTC (*Negative Temperature Coefficient*). Com a colocação física de um *jumper* interno é possível estabelecer um divisor de tensão através da resistência de *pull-up* representada e deste modo medir a queda de tensão no sensor de temperatura NTC.

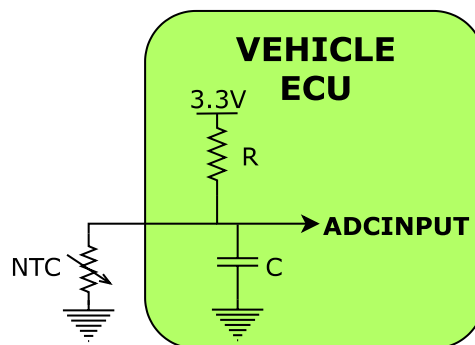


Figura 4.9 – Leitura de resistência variável.

Comunicação CAN

Um dos pontos fundamentais da MeaTec32 é a comunicação CAN, nomeadamente o facto de existirem 3 barramentos CAN. A estruturação destes canais foi pensada de modo a que um canal seja de alta prioridade, para a comunicação com o controlador do motor elétrico, outro dos canais é de baixa prioridade, para a comunicação com o carregador e sistema de gestão de energia e um canal para a interface com o condutor/utilizador.

O microcontrolador SAM3X8E possui dois controladores de CAN integrados na sua arquitetura, sendo necessário para completar o barramento, adicionar externamente dois *transceivers*. Para o terceiro barramento CAN é necessário a utilização externa de um controlador CAN e *transceiver* CAN, este controlador comunica com o microcontrolador através do protocolo SPI. Na Figura 4.10 encontra-se representada a arquitetura adotada para a comunicação CAN, onde se verifica a utilização interna dos controladores nativos do microcontrolador e *transceivers* externos associados ao canal CAN 0 e CAN1. Observa-se também o controlador e *transceiver* externo associado ao canal CAN 2, do tipo CAN FD como analisado na secção 3.1.1.

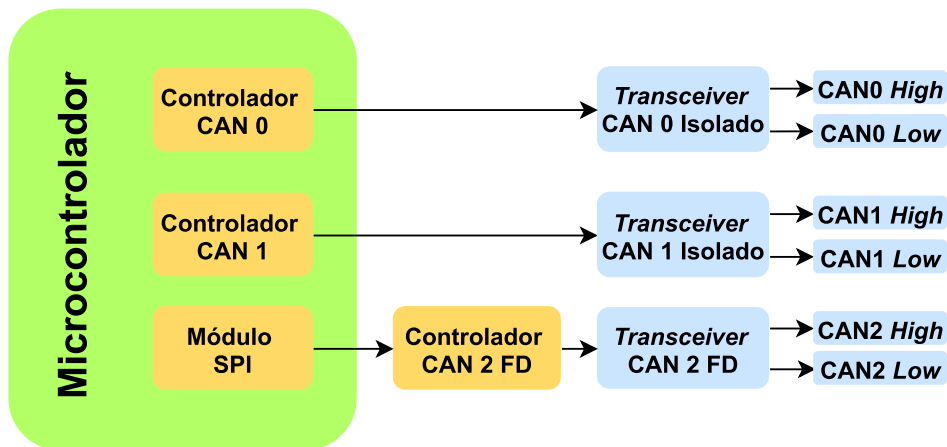


Figura 4.10 – Arquitetura CAN MeaTec32.

Os transceivers selecionados para o canal CAN 0 e CAN 1 são de referência ISO1050DUBR da Texas Instruments®, e possuem isolamento galvânico de forma a isolar fisicamente o barramento CAN dos sinais digitais. A escolha deste componente deve-se à sua simplicidade e às vantagens acrescidas que apresenta, quando utilizado em conjunto com uma fonte de alimentação isolada é possível garantir a proteção do barramento de comunicação.

O terceiro canal CAN que está associado à comunicação CAN FD, não se encontra isolado, por uma questão de simplicidade e de custo, pois os controladores de CAN FD isolados são dispendiosos. Pretende-se com este realizar a comunicação com sistemas que sejam de baixa tensão, como por exemplo uma *interface* com o condutor do veículo. Para tal utilizou-se um controlador CAN de referência MCP2517FD da Microchip® e um *transceiver* de referência TCAN332GD da Texas Instruments®, sendo que estes dois componentes são específicos para CAN FD. A escolha destes dois componentes deve-se à compatibilidade entre estes e o microcontrolador utilizado, além da simplicidade de funcionamento.

Armazenamento de dados

Uma funcionalidade de qualquer unidade de controlo nos veículos é a capacidade de armazenamento de dados em memória não volátil, ou seja, que seja capaz de reter a informação armazenada em períodos que não se encontra alimentada. Tipicamente, e apesar da maioria dos microcontroladores possuírem EEPROM interna, é utilizada uma EEPROM externa ao microcontrolador que permite o aumento de capacidade de armazenamento de dados. Neste tipo de situações utiliza-se uma comunicação SPI ou uma comunicação I2C para a troca de dados entre os dois dispositivos.

O microcontrolador SAM3X8E utilizado não possui EEPROM interna e como tal é necessário efetuar o armazenamento de dados, relativos ao funcionamento da MeaTec32, numa EEPROM externa. Optou-se pelo componente de referência AT24C512C da Microchip®, que pertence à família do microcontrolador e também é bastante conhecido na comunidade *open source*. Através de comunicação I2C é estabelecida a ligação entre o microcontrolador e a EEPROM, que possui uma capacidade de 512 kbits e funciona a uma frequência até 1 MHz, limitada apenas pela frequência máxima para I2C de 400 kHz disponibilizada pelo microcontrolador SAM3X8E. Na Figura 4.11 apresenta-se o circuito elétrico utilizado, onde se verifica a existência de duas resistências de *pull-up* na linha de dados e de relógio, necessárias para o funcionamento deste tipo de comunicação.

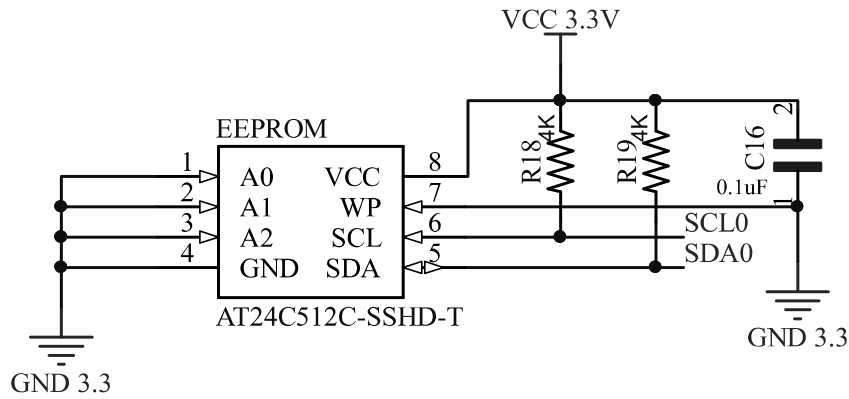


Figura 4.11 – Armazenamento de dados.

Comunicação série

Para comunicação entre o programador/utilizador e a unidade de controlo de um veículo é indispensável que exista um protocolo específico para esta tarefa. A comunicação é normalmente efetuada entre um computador e a unidade de controlo, podendo servir como forma de programação da unidade e/ou troca de dados com o utilizador em tempo real. A troca de dados é particularmente útil numa situação de ensaios, onde seja necessário visualizar em tempo real, parâmetros relativos ao funcionamento do veículo. A programação da presente MeaTec32 é realizada através do porto USB nativo do microcontrolador. Para comunicação com outro tipo de dispositivos série é utilizado o conector principal que possui 2 pinos dedicados à transmissão de dados (RX/TX). Para tal utilizou-se o isolador digital com a referência MAX12931 da Maxim Integrated®, em que se observa, na Figura 4.12, o circuito elétrico do mesmo. Pela análise deste circuito facilmente se verificam as duas alimentações distintas, necessárias para um componente isolado entre a sua entrada e a sua saída. Este componente possui dois canais para troca de dados de forma bidirecional e isolamento galvânico das linhas de transmissão (TX) e receção (RX), sendo assim possível o isolamento físico dos sinais digitais que percorrem estas linhas de dados.

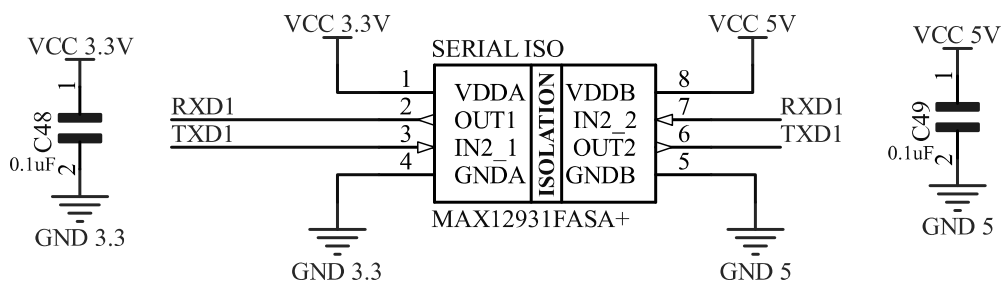


Figura 4.12 – Isolador digital para comunicação série.

Saídas digitais

A variedade de atuadores existentes leva a que a maioria das unidades de controlo possuam dois tipos de saídas digitais. Na saída digital do tipo *low-side* a carga encontra-se permanentemente alimentada à tensão da bateria, o controlo desta carga é feito através do corte à massa da mesma. Por motivos de eficiência e de velocidade de resposta, optou-se pela utilização de transístores do tipo MOSFET, este componente, quando comparado com outros tipos de transístor, permite uma grande eficiência, aliado a uma precisão de atuação superior. O MOSFET (*Metal-Oxide Semiconductor Field-Effect Transistor*) ao ser atuado internamente pela unidade fecha o circuito da carga, o que permite a passagem de corrente e consequentemente o funcionamento desta. Na saída digital do tipo *high-side* a carga encontra-se permanentemente conectada à massa e o controlo é feito através do MOSFET interno da unidade de controlo que efetua o corte à alimentação da carga. Na Figura 4.13 estão representados os esquemas de ligações para estes tipos de saídas.

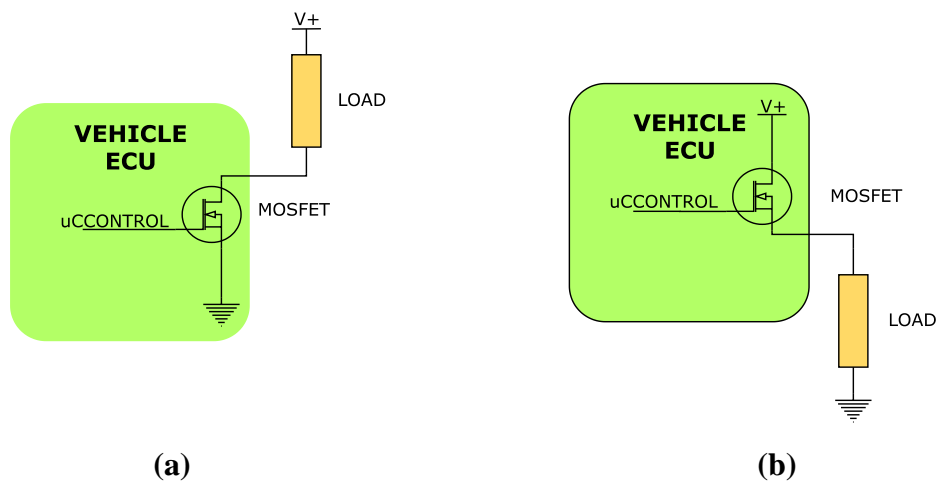


Figura 4.13 – (a) Saída digital *low-side*; (b) Saída digital *high-side*.

Atualmente as saídas digitais na unidade de controlo do veículo moderno já não são apenas sistemas do tipo *on/off*. Devido ao aumento da complexidade de atuadores disponíveis nos veículos, a necessidade de existência de diagnóstico nas saídas digitais é praticamente inevitável. De forma a efetuar um controlo rigoroso das cargas atuadas, podem existir saídas digitais com a funcionalidade de *feedback* de corrente. Neste tipo de saídas, é possível monitorizar a corrente consumida pela carga que está a ser controlada, de modo a perceber se a carga está a comportar-se corretamente ou se o consumo de corrente é excessivo e encontra-

se com uma possível avaria. Este tipo de funcionalidade é bastante útil, pois rapidamente se consegue desabilitar uma carga que se encontra com um comportamento inesperado e que de outra forma iria causar danos permanentes na unidade de controlo do veículo e seus periféricos.

A utilização de saídas digitais configuráveis como saídas PWM para a atuação de cargas que necessitem deste tipo de controlo é bastante comum em unidades de veículos, apresenta uma mais valia, pois, a unidade consegue-se adaptar a diversos tipos de veículos. Como referido na secção 4.1, este módulo dispõe de 6 saídas digitais. Estas saídas permitem controlar diversas cargas num veículo, como por exemplo uma bomba de líquido de arrefecimento ou um relé.

Saídas Digitais Genéricas

Duas das saídas são para controlo exclusivo de cargas que funcionam a 12 V, sendo que uma saída é destinada para cargas do tipo *high-side* e a outra saída destina-se a cargas do tipo *low-side*. Estas saídas encontram-se isoladas do microcontrolador através de um *optocoupler* de referência VOD205T da Vishay®. Nestas duas saídas existe a possibilidade de utilização de PWM para controlo de cargas, mas não existe funcionalidade de diagnóstico e de *feedback* de corrente. Na Figura 4.14 observa-se as ligações deste *optocoupler*, em que através de um sinal de 3.3 V é possível atuar de forma isolada uma carga de 12 V. Também se observam as restantes ligações dos MOSFET's para os dois tipos de saídas aqui disponíveis.

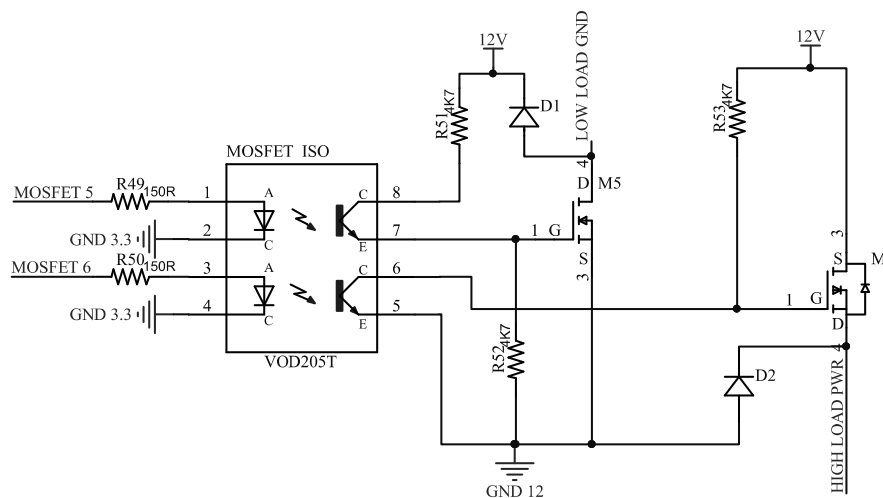


Figura 4.14 – Saída digital genérica *high-side* e *low-side*.

Saídas Digitais “Inteligentes”

Nas restantes saídas digitais do tipo *high-side*, optou-se por realizar um controlo de cargas que operam a uma tensão que pode variar entre 6 V e 32 V. Para tal, utilizaram-se MOSFET’s de potência “inteligentes” de referência AUIR3314S da Infineon®, denominados deste modo devido às características que apresentam, tais como proteção contra curto-circuito, corte por temperatura e *feedback* de corrente consumida [36]. A escolha deste componente baseou-se no baixo custo e nas funcionalidades que apresenta. Também a elevada corrente de saída de 18 A é um fator determinante, ainda que estas saídas se encontram estipuladas para correntes máximas de 7 A. Esta limitação deve-se ao conector principal utilizado apenas permitir 10 A de corrente em cada pino. Na Figura 4.15 observa-se o esquema de ligações deste MOSFET que é controlado por um MOSFET de sinal e possui alimentações independentes da MeaTec32, podendo ser alimentado até 32 V.

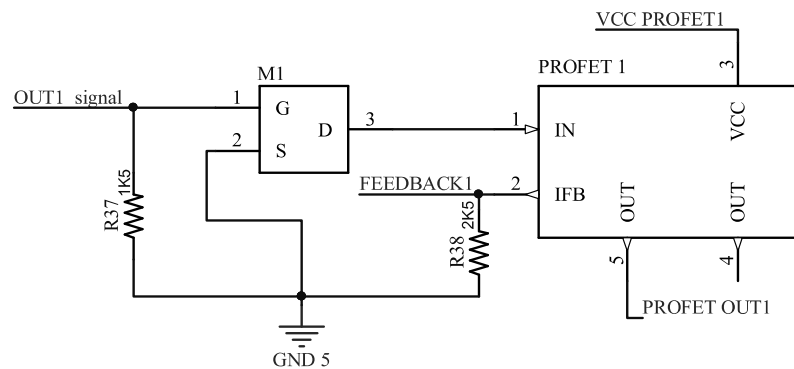


Figura 4.15 – Saída digital de potência.

A ficha técnica deste componente [36] salienta a capacidade de se efetuar a desativação da carga caso a corrente consumida por esta exceda um valor pré-definido. Este controlo é definido por uma resistência, R38 na Figura 4.15, que é determinada em função da corrente máxima desejada, como se pode observar no gráfico da Figura 4.16. Verifica-se que, por exemplo, para uma resistência de 3000 Ohm, a desativação da saída por corte de corrente dá-se a 8 A.

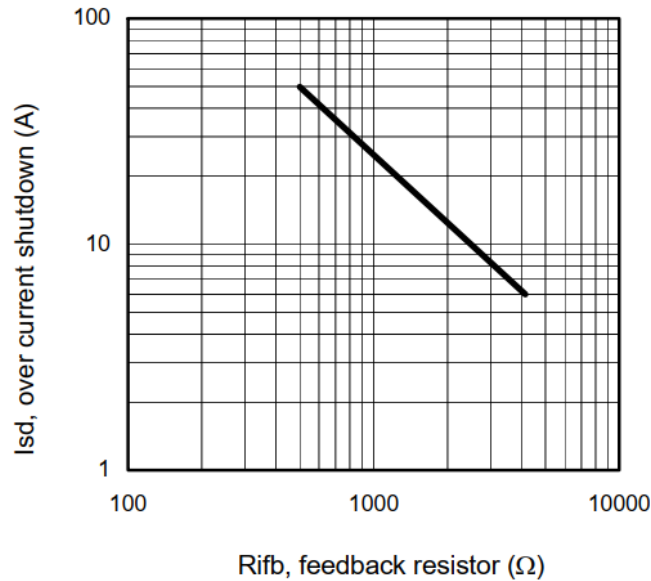


Figura 4.16 – Determinação da resistência de shutdown [36].

4.3. Testes de componentes

De modo a finalizar a escolha de componentes e também para compreender por completo o funcionamento destes, foi realizada uma avaliação do seu desempenho em placa de testes, também conhecida como *breadboard*. Os testes incidiram sobre os componentes principais da PCB realizada, tais como os MOSFET “inteligentes” AUIR3314S, o isolador digital ISO7740DWR, o amplificador isolado AMC1211AQDWVRQ1 e o microcontrolador SAM3X8E.

Uma das principais características do MOSFET AUIR3314S, como referido na secção 4.1, é a existência de um pino de *feedback* que apresenta na sua saída uma tensão proporcional à corrente consumida pela carga. Deste modo, é possível que o microcontrolador realize a monitorização da corrente da carga através da leitura da tensão de *feedback*, realizada através do isolador analógico AMC1211AQDWVRQ1.

De forma a determinar a gama de valores de *feedback* esperados em funcionamento normal, realizaram-se testes com diversas cargas semelhantes às encontradas no automóvel comum. Obteve-se assim a reta apresentada na Figura 4.17, que traduz a corrente consumida pela carga, em função da tensão de feedback fornecida pelo MOSFET “inteligente”.

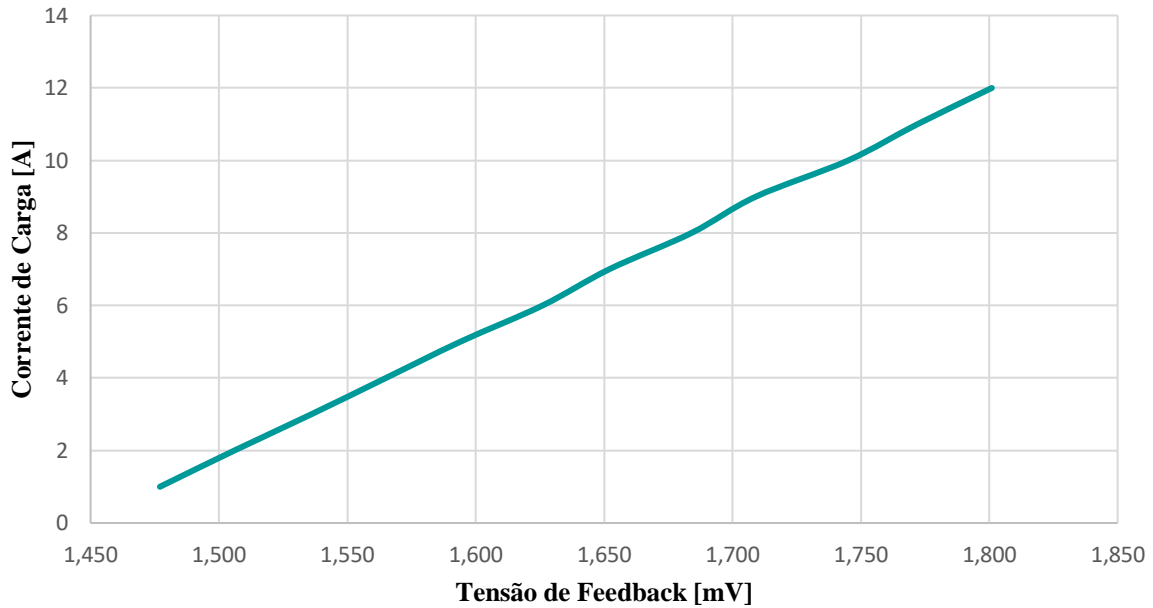


Figura 4.17 – Curva característica da tensão de *feedback* do AUIR3314S.

Verifica-se que a função é praticamente linear, sendo que pode ser aproximada pela seguinte equação 4.4:

$$I_{load} [A] = (33.89 \times V_{feedback}) - 49.043 \quad (4.4)$$

Também se efetuaram outros testes com o objetivo de verificar a atuação do MOSFET “inteligente”. Na Figura 4.18 observa-se as curvas de tensão retiradas do osciloscópio PicoScope[®], onde o sinal a vermelho representa a ordem de atuação por parte do microcontrolador e o sinal a azul traduz a tensão de saída do MOSFET. Na primeira onda observa-se a atuação do microcontrolador com um valor de tensão de 5 V e a saída do MOSFET a ficar ativa a um nível de 12 V. Na onda seguinte, provocou-se o aumento de corrente da carga de forma a exceder o limite de corrente definido e verifica-se que, apesar de existir ordem de atuação por parte do microcontrolador, a saída do MOSFET apenas fica ativa durante 900 μ s. Resumindo, uma vez que a situação é de excesso de corrente, a carga é instantaneamente desativada de modo a não provocar danos no sistema.

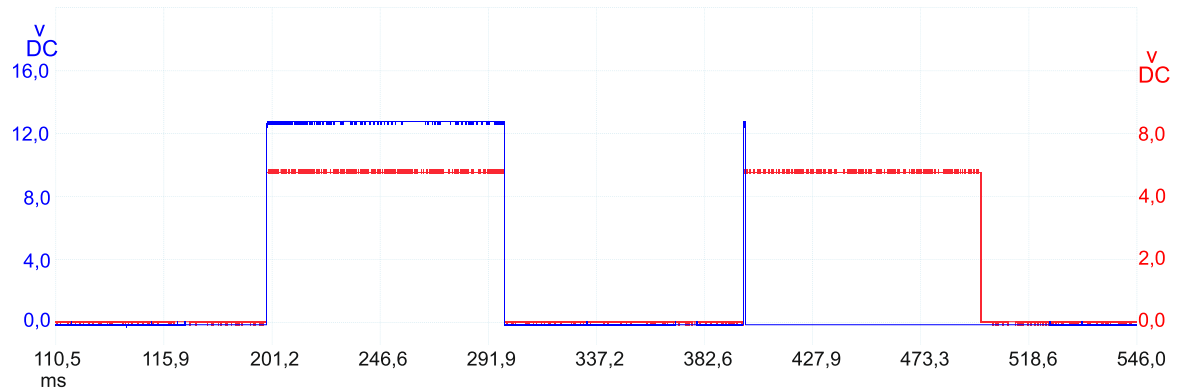


Figura 4.18 – Aplicação do limite de corrente AUIR3314S.

4.4. Desenvolvimento da PCB

Para o desenvolvimento da PCB foi necessário numa primeira fase efetuar o circuito esquemático na funcionalidade própria do *software* Altium® Designer. Os componentes encontram-se distribuídos pelo tipo de funcionalidade, como por exemplo: entradas, processador, saídas, entre outros. Após a validação do circuito esquemático foram definidas as especificações da PCB, referidas na Tabela 5, antes de se dar início ao processo de *routing*, ou seja, efetuar as ligações elétricas entre os componentes colocados na PCB. Estas especificações são feitas com base nas capacidades da empresa que efetuou o fabrico da PCB, referida na secção 4.5 e englobam a largura das pistas elétricas, as dimensões das vias utilizadas para a ligação do plano inferior (*bottom layer*) da PCB ao plano superior (*top layer*), entre outras. Tendo em conta o encapsulamento e conector utilizado, fabricados pela empresa Cinch® Connectivity Solutions, as dimensões da PCB foram restringidas a 100 mm por 85.1 mm. Estes dois componentes foram escolhidos de modo a que o módulo final seja robusto, à prova de poeiras e água, de acordo com as classes de proteção IP67 e IP69K. Optou-se, sempre que possível, pela utilização de componentes de montagem em superfície, conhecidos como componentes SMD (*Surface Mount Device*), devido à rentabilização de espaço que se consegue alcançar.

Os componentes estão colocados na PCB em duas zonas distintas que se encontram isoladas eletricamente entre si, em cada uma das zonas a disposição de componentes foi pensada de forma a rentabilizar o espaço disponível e a otimizar as ligações entre estes. Para uma melhor perceção da localização dos componentes deve-se observar a Figura 4.20 e a Figura 4.21, bem como o Anexo B.

A zona junto ao conector destina-se a componentes de potência e a componentes que trabalham a uma tensão de alimentação igual ou superior a 5 V, a zona de processamento, comunicações e acondicionamento de sinal encontra-se oposta ao conector, onde apenas se encontram componentes que trabalham a 3.3 V. Deste modo reduz-se a influência da interferência de ruídos sobre a parte de processamento que possam existir na zona de potência e salvaguarda-se a zona processamento do calor gerado na parte de potência. Cada uma das zonas possui um plano de massa próprio que efetua a ligação de massa entre os componentes. Numa das zonas, a ligação de massa referente a 12 V/5 V e na outra zona a ligação de massa referente a 3.3 V. Devido às dimensões da PCB e ao número de componentes, estes encontram-se colocados nas duas faces da PCB, *top layer* e *bottom layer*.

A disposição de componentes como resistências, cristais e condensadores foi realizada de acordo com o *datasheet* dos mesmos. Os condensadores por exemplo, encontram-se o mais próximo possível dos pinos de alimentação dos componentes, de modo a que a corrente de alimentação percorra os condensadores antes de alcançar os componentes. Deste modo garante-se a correta filtragem e estabilização da tensão de alimentação dos componentes. As ligações elétricas entre os componentes, apesar da existência de ferramenta de ligação automática no *software* utilizado, foram realizadas de forma individual e manual para que toda a PCB seja otimizada a nível de ligações e colocação de componentes.

Na Figura 4.19 observa-se, em fase de projeto, a vista superior da PCB a vermelho e representado a azul a vista inferior. Com a utilização da funcionalidade do *software* Altium® de visualização 3D, observa-se na Figura 4.19 a PCB desenvolvida sem a colocação de componentes.

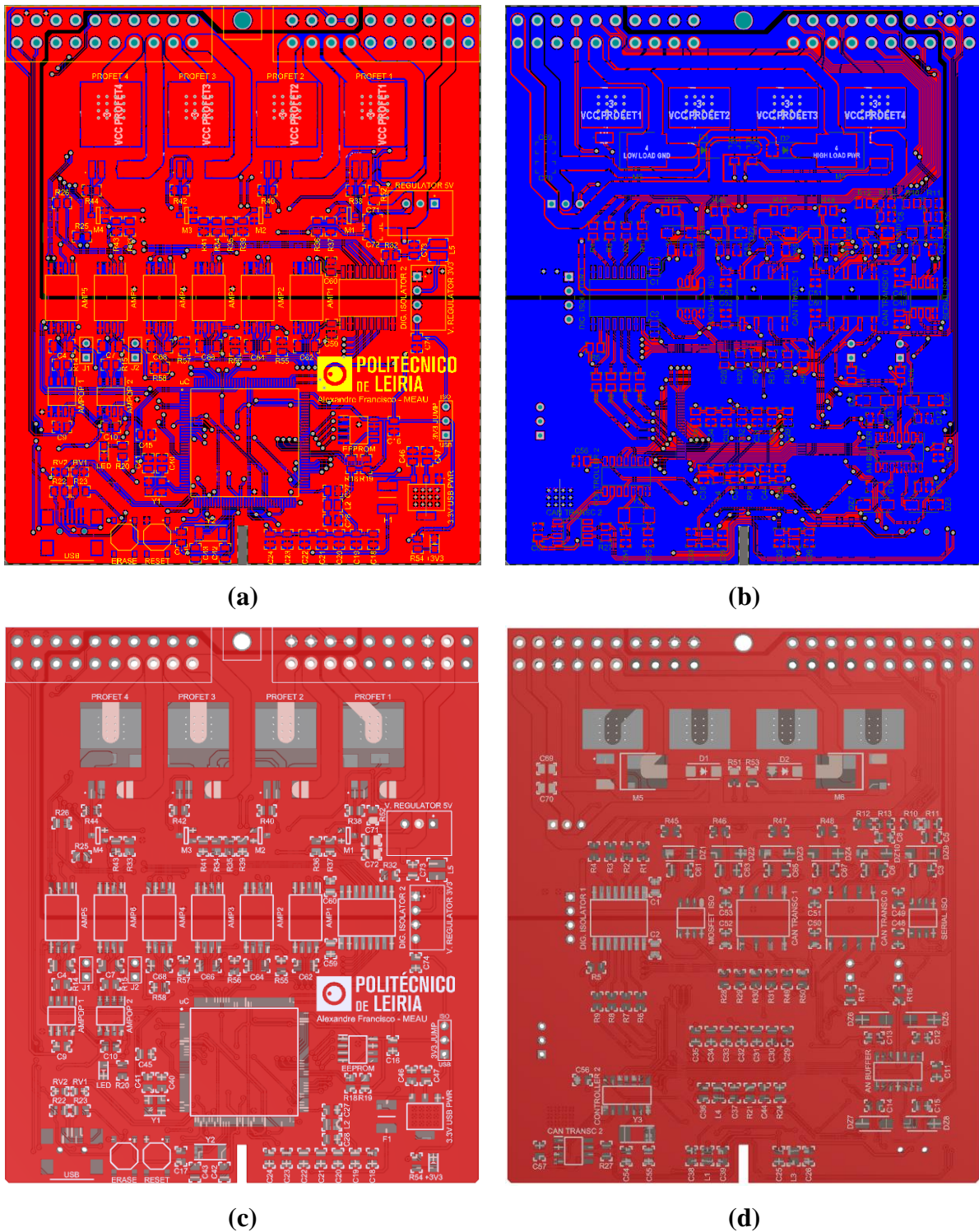


Figura 4.19 – (a) Vista 2D superior da PCB; (b) Vista 2D inferior da PCB; (c) Vista 3D superior da PCB; (d) Vista 3D inferior da PCB.

Após a colocação de componentes foi possível verificar a interação física entre estes dentro do espaço da placa. Na Figura 4.20 observa-se a vista final superior da PCB desenvolvida, onde se identifica claramente a zona de componentes isolados, tanto na zona superior como na

zona inferior. Esta zona, localizada no centro da PCB, separa a parte de potência da parte de processamento. Com esta vista 3D foi possível observar a integração dos componentes na PCB, perceber se existem erros de disposição de componentes que tenham de ser corrigidos e conhecer as dimensões finais do módulo desenvolvido.

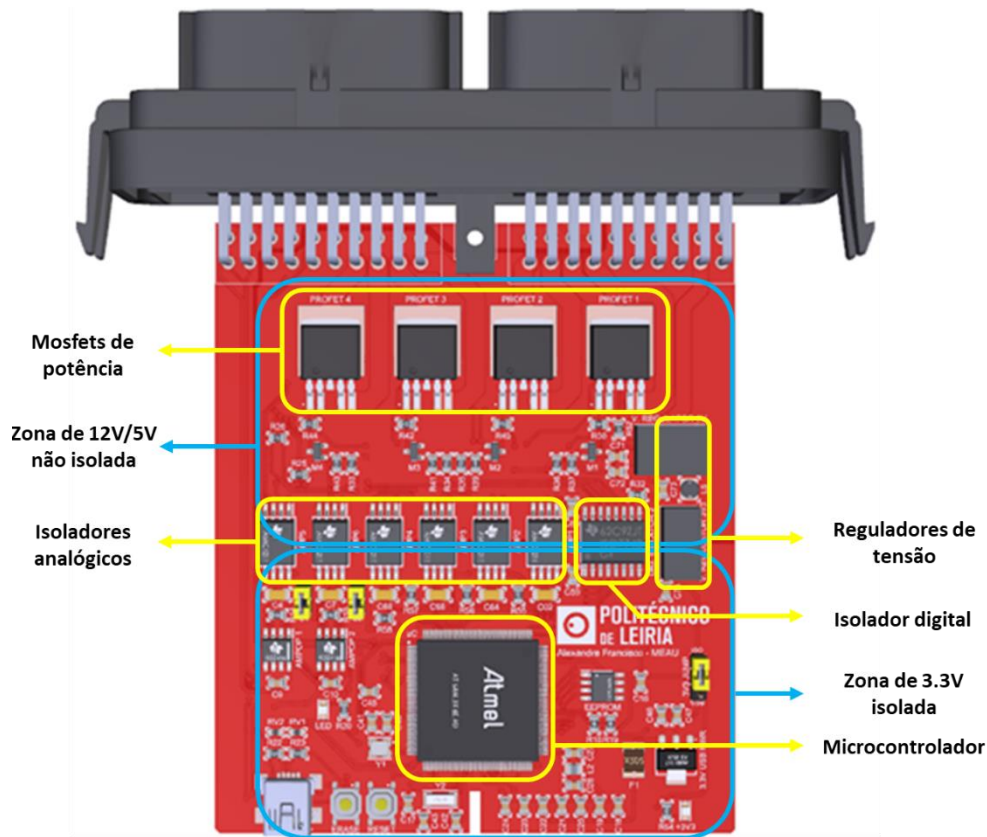


Figura 4.20 – Vista final 3D superior da PCB.

Na Figura 4.21 observa-se a vista final 3D da zona inferior da PCB, esta zona também possui dois planos de massa isolados. É possível por exemplo, verificar os MOSFET's genéricos, os componentes da comunicação CAN convencional e CAN FD e alguns isoladores digitais.

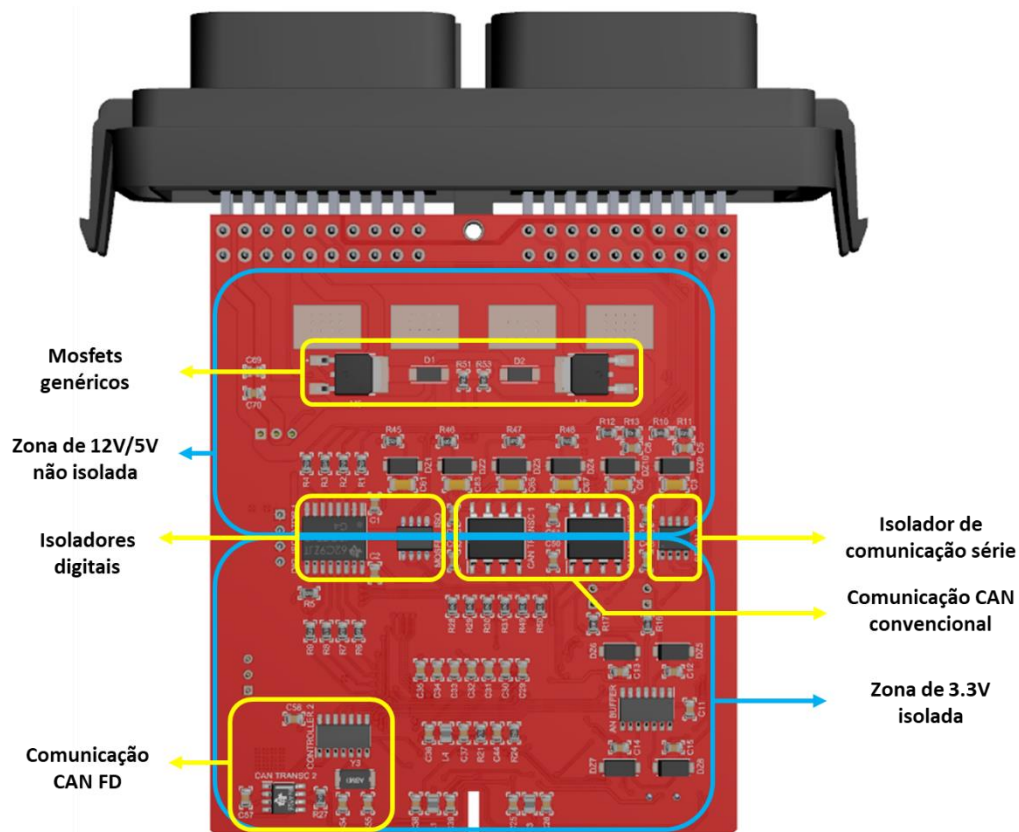


Figura 4.21 – Vista final 3D inferior da PCB.

4.5. Produção e montagem da PCB

Devido à relativa complexidade e tolerâncias da PCB apresentada, a produção da mesma foi realizada por uma empresa externa, nomeadamente a Seedstudio, que se encontra localizada na China e é especialista na produção de PCB de diferentes características. Na Tabela 5 encontram-se enumeradas as principais especificações da PCB produzida, a maioria destas especificações são *standard* do fabricante com o objetivo de reduzir o tempo de produção e o custo.

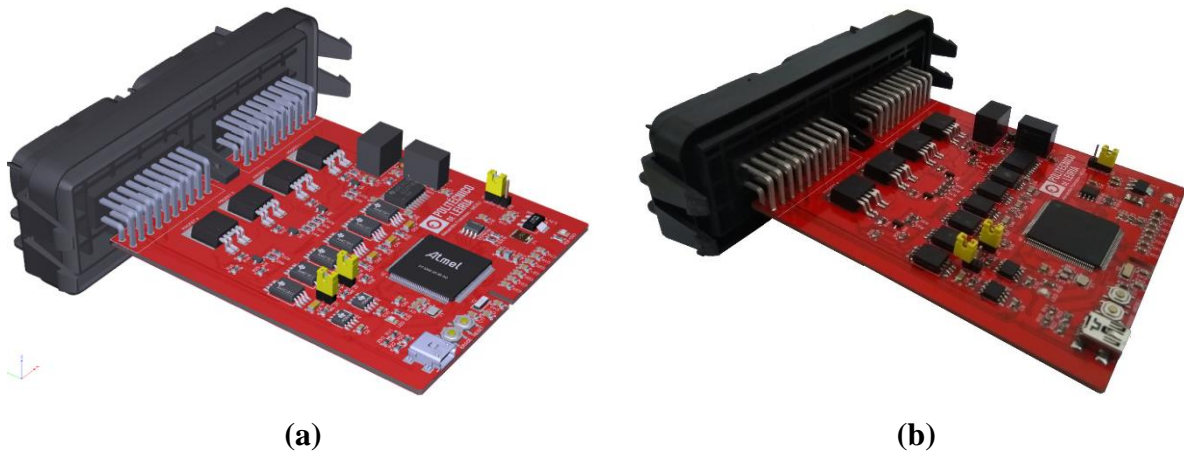
O material de base da PCB, FR-4 TG130, é constituído por resina de epóxi e fibras de vidro, onde a designação TG130 indica que o material suporta uma temperatura máxima de 130 °C, a partir da qual começa a deformar. Utilizaram-se duas camadas da PCB, *top layer* e *bottom layer*, sendo que o fabricante também realiza PCB's de quatro e seis camadas. As dimensões da PCB final são de 100 mm por 85.1 mm e com uma espessura de 1.6 mm, restringidas pelo encapsulamento escolhido. A máscara de solda, também designada por *soldermask*, é de cor vermelha e o acabamento superficial da PCB é feito através de um processo conhecido como

HASL (*Hot Air Solder Level*). Neste tipo de acabamento os *pads*, zona de contacto entre os componentes e a PCB, são revestidos com uma camada fina de estanho, de modo a resistirem à oxidação e para que o processo de soldadura de componentes seja facilitado. As pistas da PCB são constituídas por cobre com uma densidade superficial de 1 oz/ft², unidade típica utilizada na fabricação deste tipo de placas. As vias utilizadas têm um furo mínimo de 0.3 mm, um diâmetro mínimo de 0.7 mm e encontram-se cobertas com máscara de solda para que fiquem protegidas de oxidação. A largura mínima das pistas assim como o espaçamento mínimo entre pistas é de 0.1524 mm.

Tabela 5 – Especificações da PCB produzida [37].

Característica	Valor/Tipo
Material de base	FR-4 TG130
N.º de camadas	2
Dimensões	100 x 85.1 mm
Cor da máscara de solda	Vermelho
Espessura	1.6 mm
Acabamento	HASL
Densidade superficial	1 oz/ft ²
Diâmetro mínimo de furos	0.3 mm
Diâmetro mínimo de vias	0.7 mm
Largura mínima de pistas	0.1524 mm
Espaçamento mínimo entre pistas	0.1524 mm

Na Figura 4.22 (a) apresenta-se a vista 3D da PCB final retirada do *software* Altium® Designer e na Figura 4.22 (b) apresenta-se a PCB finalizada. Como se pode observar pelas duas figuras, o resultado foi o esperado, pois estas duas imagens assemelham-se por completo.



(a)

(b)

Figura 4.22 – (a) PCB em fase de projeto; (b) PCB final.

Como resultado encontra-se representada na Figura 4.23 a MeaTec32 finalizada, com a PCB dentro do encapsulamento.

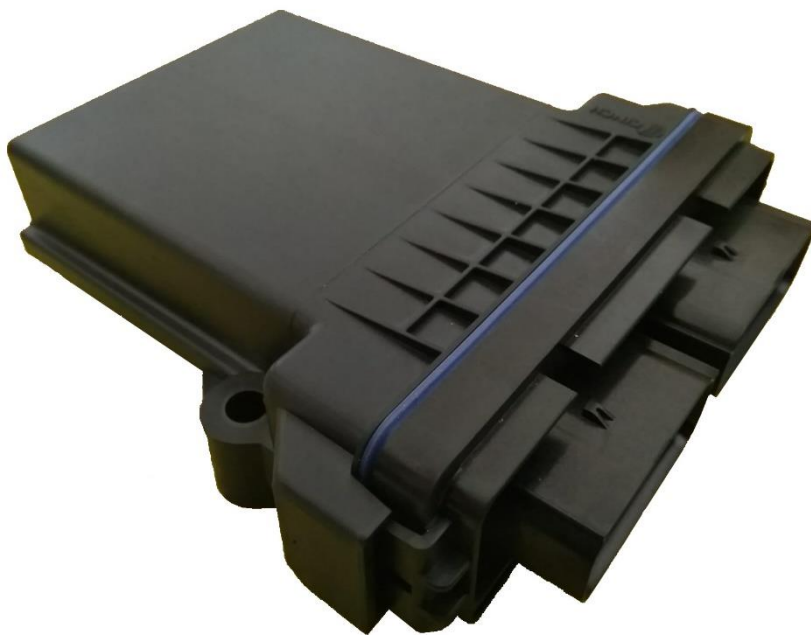
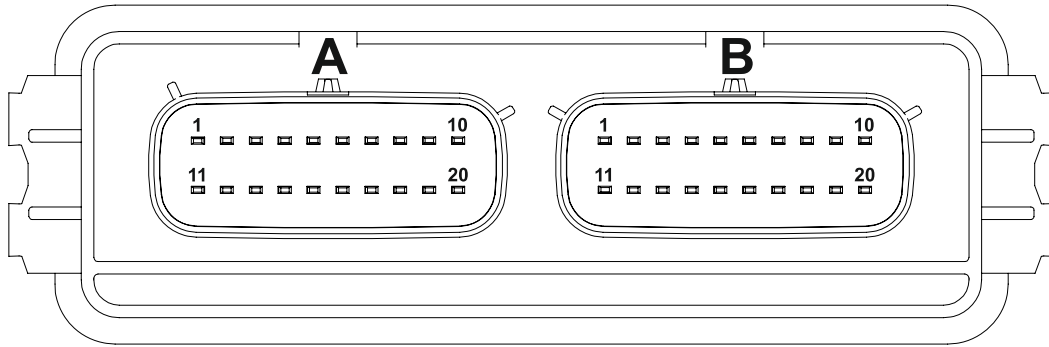


Figura 4.23 –MeaTec32 finalizada.

Na Figura 4.24 é possível visualizar a identificação de cada terminal da MeaTec32. Para uma descrição mais detalhada deve ser consultado o Anexo A que explica a função de cada pino e a sua localização no conector.



A1 - BAT_POS12V (R)	B1 - PFET_OUT3 (BL)
A2 - BAT_NEG0V (BR)	B2 - PFET_SUP3 (G)
A3 - SEN_0V_NISO (GY)	B3 - PFET_OUT4 (BL)
A4 - SEN_5V_NISO (Y)	B4 - PFET_SUP4 (G)
A5 - MFET_LS (BL)	B5 - CAN0_HI (BL/BR)
A6 - MFET_HS (G)	B6 - CAN0_LO (GY/R)
A7 - PFET_OUT1 (BL)	B7 - CAN1_HI (BL/BR)
A8 - PFET_SUP1 (G)	B8 - CAN1_LO (GY/R)
A9 - PFET_OUT2 (BL)	B9 - CAN2_HI (BL/BR)
A10 - MFET_SUP2 (G)	B10 - CAN2_LO (GY/R)
A11 - BAT_POS12V (R)	B11 - DIGI4 (W/GY)
A12 - BAT_NEG0V (BR)	B12 - DIGI3 (BL/GY)
A13 - SEN_0V_NISO (GY)	B13 - DIGI2 (BR/GY)
A14 - SEN_5V_NISO (Y)	B14 - DIGI1 (G/GY)
A15 - SERIAL_RX (BL/BK)	B15 - AVI1_ISO (W/17)
A16 - SERIAL_TX (GY/BK)	B16 - AVI2_ISO (W/18)
A17 - SEN_3V3_ISO (LG)	B17 - AVI3_NISO (W/19)
A18 - SEN_3V3_ISO (LG)	B18 - AVI4_NISO (W/20)
A19 - SEN_0V_ISO (PR)	B19 - AVI5_NISO (W/21)
A20 - SEN_0V_ISO (PR)	B20 - AVI6_NISO (W/22)

Figura 4.24 – Pinout da MeaTec32.

O custo de produção da MeaTec32 foi estimado como sendo de 158,87 €, para a contabilização do custo teve-se em consideração todos os componentes que constituem a MeaTec32, mas não foi realizada uma análise dos custos de matérias consumíveis e da mão de obra necessária para a montagem de componentes. No Anexo C pode ser consultada a lista de componentes e as suas referências, que inclui também o custo de produção.

Esta página foi intencionalmente deixada em branco

5. Validação da MeaTec32

No presente capítulo demonstram-se os testes que foram realizados à MeaTec32, os quais incidiram sobre o funcionamento geral dos periféricos desta unidade e das funcionalidades disponíveis. Também é abordado o simulador construído para este efeito bem como os resultados alcançados com esta montagem.

5.1. Equipamentos e metodologia de testes

A realização dos testes iniciais de validação foi feita em bancada, ou seja, fora do veículo. Deste modo é possível configurar praticamente todas as variáveis que influenciam o funcionamento da MeaTec32 e observar de perto o seu desempenho, naquele que pode ser considerado um ambiente controlado. Para a realização de testes, foi utilizado de uma forma geral, um osciloscópio que permite medir e interpretar os sinais provenientes da MeaTec32. O osciloscópio utilizado foi um Picoscope[®] 2205, na Tabela 6 observam-se algumas características deste equipamento, é possível efetuar a leitura de 2 canais em simultâneo, tanto em tensão como corrente. A gama de frequência medida é de 25 MHz, com uma taxa de amostragem máxima de 200 MS/s, mais do que suficiente para os testes aqui realizados. Outra característica fundamental na execução destes testes é a funcionalidade de descodificação série que este osciloscópio tem incorporado. Esta descodificação série permite ao utilizador do equipamento interpretar sinais provenientes de um barramento CAN, CAN FD, SPI, I2C, entre outros.

Tabela 6 – Características técnicas do osciloscópio Picoscope[®] 2205 [38].

Característica	Valor/Tipo
Canais de medição	2
Gama de frequência	25 MHz
Taxa de amostragem	200 MS/s
Descodificação	Série (CAN, SPI, I2C, UART)

Foi realizada uma análise termográfica que permite avaliar a temperatura de funcionamento de todos os componentes da MeaTec32. Com esta análise é possível diagnosticar eventuais problemas que não estão visíveis a olho nu, nomeadamente o calor gerado em excesso por certos componentes, devido à baixa eficiência energética que possam apresentar. O equipamento utilizado na inspeção por termografia da PCB, que se aborda na secção 5.5, foi o multímetro térmico TRMS Fluke 279 FC. Na Tabela 7 apresentam-se as características deste equipamento que permite a medição de temperaturas de $-10\text{ }^{\circ}\text{C}$ a $200\text{ }^{\circ}\text{C}$, com uma resolução de $0.1\text{ }^{\circ}\text{C}$. A emissividade de leitura é fixa e a temperatura medida é apresentada num ponto central do visor deste multímetro.

Tabela 7 – Características técnicas do multímetro térmico TRMS Fluke 279 FC[39].

Característica	Valor/Tipo
Gama de medição	$-10\text{ }^{\circ}\text{C}$ a $200\text{ }^{\circ}\text{C}$
Resolução de medição	$0.1\text{ }^{\circ}\text{C}$
Emissividade	0.95 (fixa)
Medição de temperatura	Central

5.2.Reguladores de tensão

Após o teste físico de confirmação de ligações elétricas, a MeaTec32 foi alimentada através de uma fonte de alimentação de 12 V. Foram medidos todos os pinos de alimentação dos componentes, na zona de 5 V e na zona de 3.3 V e o comportamento foi o esperado. Ficou comprovado que ambos os reguladores de 5 V e 3.3 V se encontravam a funcionar corretamente como se verifica na Figura 5.1. Representado a azul encontra-se o sinal de alimentação de 5 V e a vermelho encontra-se representado o sinal de alimentação de 3.3 V. Após a alimentação da MeaTec32 a 12 V, o regulador de tensão não isolado apresenta 5 V constantes e o regulador de tensão isolado apresenta na sua saída 3.3 V constantes.

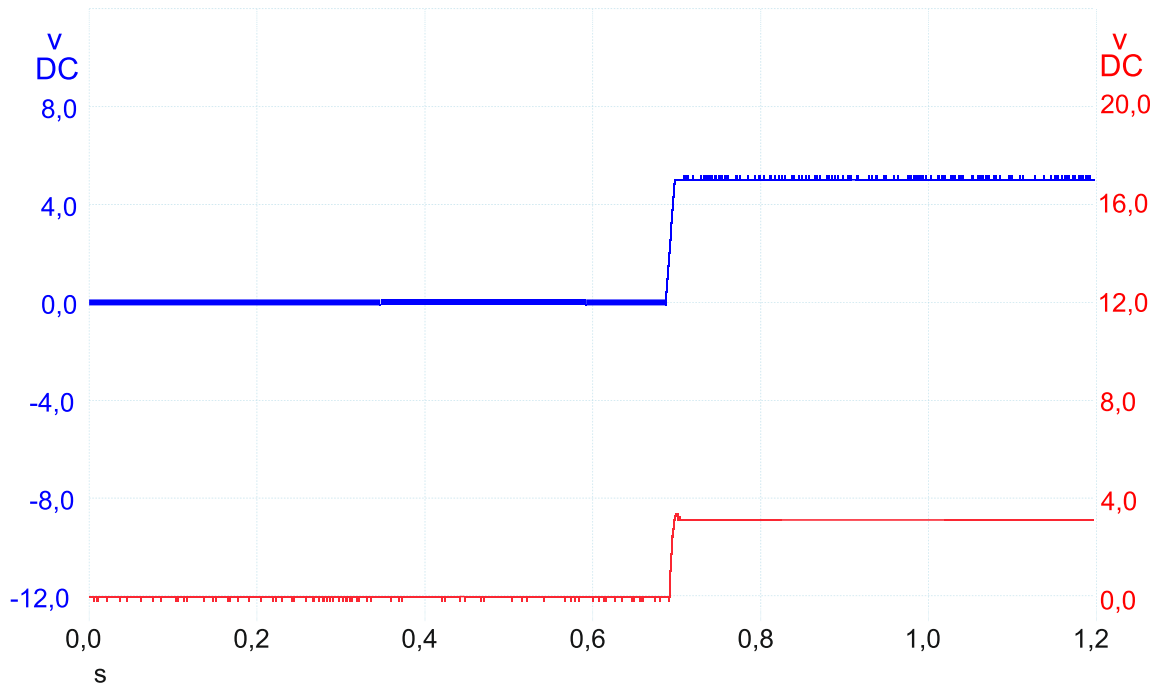


Figura 5.1 – Tensão de saída dos reguladores de tensão.

5.3. Portos I/O

Os portos de entradas e saídas da MeaTec32 encontram-se divididos em entradas digitais, entradas analógicas e saídas digitais. De seguida demonstra-se como se efetuaram os testes a cada periférico.

Entradas Digitais

As entradas digitais da MeaTec32 são constituídas pelo isolador digital de referência ISO7740DWR, que, tal como referido na secção 4.2, efetua a conversão de forma isolada de um sinal digital de 5 V, para um sinal digital de 3.3 V. Como tal, o teste desta funcionalidade incidiu sobre a medição de sinais à entrada da MeaTec32 e à saída do isolador digital, que corresponde também à entrada do microcontrolador. Uma vez que a leitura de sinais de frequência é realizada pelas entradas digitais, também foi necessário garantir que não existe distorção deste sinal entre a entrada e saída, comprovando assim que o isolador digital consegue responder às especificações da MeaTec32.

Como forma de comprovação desta funcionalidade, colocou-se um módulo genérico a gerar um sinal PWM com uma frequência de 980 Hz e um *duty-cycle* de 50 %. A medição deste

sinal gerado com o osciloscópio observa-se a azul na Figura 5.2 e a vermelho verifica-se o sinal que foi medido à saída do isolador digital. Como se observa pela Figura 5.2, o sinal de entrada é convertido de forma eficaz para o sinal de saída, que apresenta um nível de tensão alto de 3.3 V, capaz de ser adquirido pelo microcontrolador sem causar danos a este.

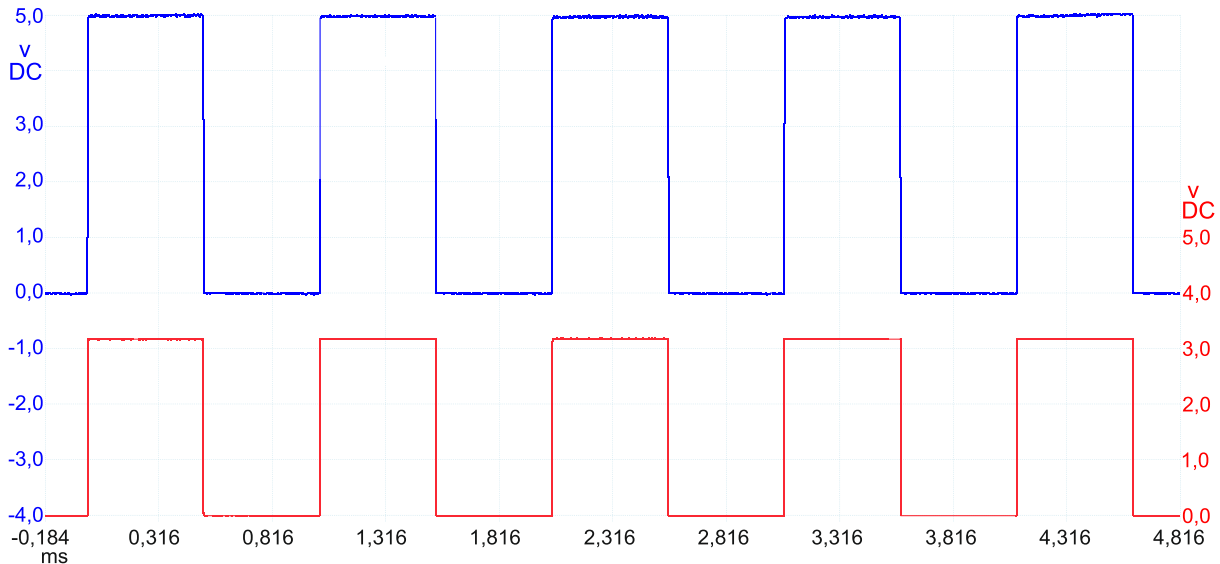


Figura 5.2 – Sinal PWM à entrada e saída do isolador digital ISO7740DWR.

A transmissão interna do sinal digital através da barreira de isolamento do isolador digital, pode desfasar o sinal de saída em relação ao sinal de entrada. Como internamente é realizado o acondicionamento deste sinal, o próprio *datasheet* refere que este fenómeno pode ocorrer. Nos testes realizados foi possível verificar um desfasamento de 600 nano segundos entre o sinal de saída e o de entrada. Com a realização destes testes foi possível verificar o correto funcionamento das entradas digitais, a conversão de sinais digitais e o pequeno desfasamento entre a entrada e saída deste componente, que pode ser desprezado na maioria das aplicações.

Entradas analógicas isoladas

De forma a comprovar o correto funcionamento das entradas analógicas isoladas, efetuou-se a medição de um sinal analógico à entrada do módulo, que varia de 0 V a 5 V, e após o acondicionamento de sinal, que varia de 0 V a 3.3 V. Na Figura 5.3 visualiza-se o sinal de entrada a azul e o sinal de saída para o microcontrolador a vermelho. É possível verificar que o sinal a vermelho segue o comportamento do sinal a azul, mas com um nível de tensão

inferior como era esperado. No entanto, para um sinal de entrada de 6 V, o sinal de saída é aproximadamente de 2.5 V e não de 3.3 V como seria ideal. Apesar do correto dimensionamento do circuito de acondicionamento de sinal, demonstrado na secção 4.2, a escolha do amplificador de instrumentação INA826AID não foi a mais correta. Após contacto com a Texas Instruments®, chegou-se à conclusão de que este componente a partir de uma tensão de 1.9 V deixa de ser linear, entrando em saturação, causada pela baixa tensão de alimentação (3.3 V).

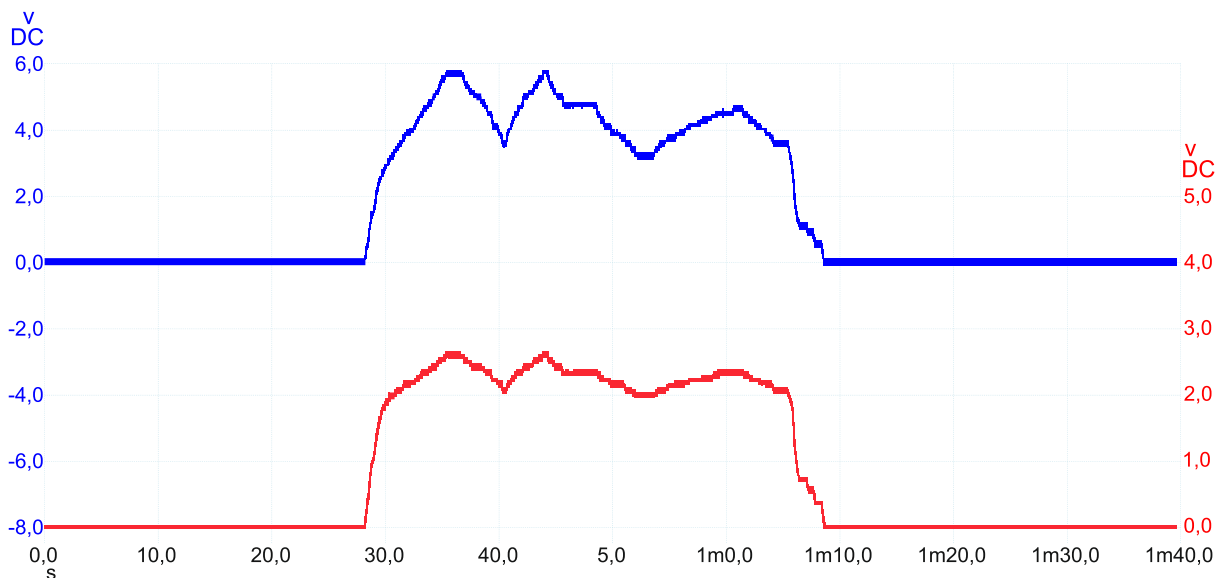


Figura 5.3 – Leitura de sinal analógico isolado.

Entradas analógicas não isoladas

As entradas analógicas não isoladas são destinadas para sensores que são alimentados a uma tensão de 3.3 V isolados, fornecidos pela MeaTec32. Estes sensores podem ser potenciómetros, sensores de temperatura, sensores de pressão, entre outros. No teste deste tipo de entradas pretendeu-se verificar a leitura de um sensor genérico do automóvel: um potenciómetro. Para este tipo de sensor, que é alimentado a 3.3 V, verificou-se externamente, através do osciloscópio, e internamente, através da leitura do ADC do microcontrolador, que o sinal do potenciómetro é adquirido e interpretado com sucesso.

Saídas Digitais

Para este teste foram utilizadas cargas resistivas encontradas no automóvel, nomeadamente 4 faróis provenientes de um simulador de iluminação. Com estas cargas pretendeu-se inicialmente comprovar o funcionamento normal de todos os MOSFET's apenas num situação de *on/off*. Também para se garantir o correto funcionamento das saídas, fez-se o teste de utilização das saídas digitais como PWM e da funcionalidade de *feedback* da corrente consumida pelas cargas.

Como referido na secção 4.1, todas as saídas digitais possuem capacidade de funcionar como saída de PWM. Para o teste desta funcionalidade efetuou-se o controlo de uma lâmpada com uma tensão de alimentação de 13.5 V. A saída para o controlo foi definida como um dos MOSFET “inteligentes”, onde a atuação da carga foi realizada com um *duty-cycle* de 50 % e uma frequência de 200 Hz. O sinal à saída do MOSFET pode-se observar na Figura 5.4. O *datasheet* destes MOSFET's da MeaTec32 refere que a frequência de trabalho máxima é de 200 Hz [36], sendo que em testes alcançou-se uma frequência de 400 Hz. No entanto, como especificação da MeaTec32, não é aconselhável exceder os 200 Hz, de modo a prolongar a vida útil da mesma.

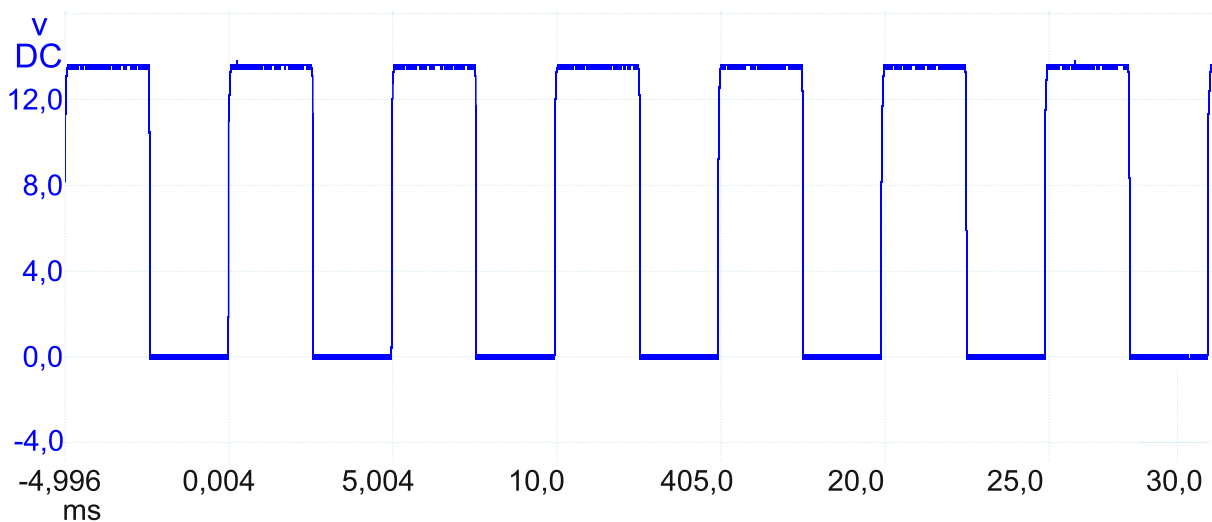


Figura 5.4 – Saída digital em controlo de PWM.

Também as saídas digitais genéricas, de *high-side* e *low-side*, foram testadas através de PWM, conseguindo-se alcançar uma frequência de trabalho máxima de 6 kHz. O principal teste das saídas digitais incidiu sobre a funcionalidade de monitorização de corrente consumida pela

carga. Para tal foi necessário efetuar a leitura da tensão de *feedback* que é monitorizada pelo ADC do microcontrolador, bem como realizar a medição externa deste sinal através do osciloscópio. Na Figura 5.5 observa-se o comportamento da tensão de *feedback*, representado azul, em função da corrente de carga, representado a vermelho. É possível verificar que a variação da corrente consumida pela carga provoca uma variação proporcional da tensão de *feedback* como era esperado. Esta medição comprova que a Equação 4.4, referida na secção 4.3, permite determinar de forma aproximada, a corrente consumida por cada carga da MeaTec32 a qualquer instante.

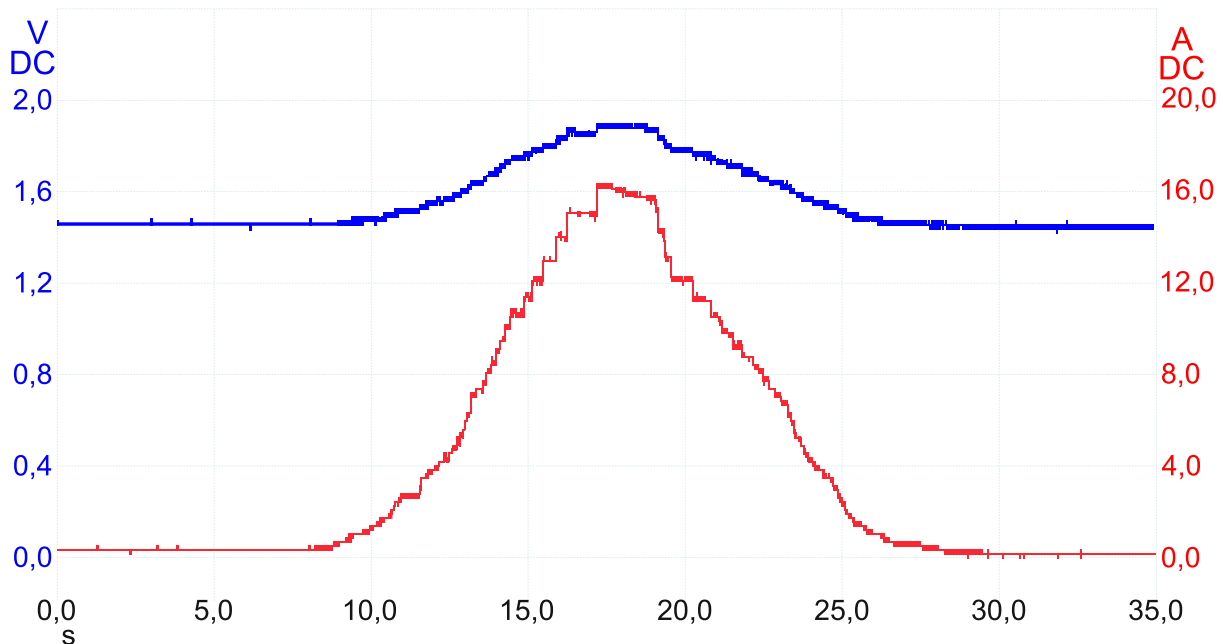


Figura 5.5 – Tensão de *feedback* e corrente de carga.

Através da monitorização desta tensão de *feedback*, o programador da MeaTec32 pode definir por *software* o limite máximo de corrente para cada uma das 4 saídas digitais de potência disponíveis. Na Figura 5.6 observa-se um exemplo deste procedimento, a corrente consumida pela carga está representada a vermelho e a tensão na saída digital está representada a azul. Neste caso em concreto, a corrente máxima permitida foi configurada por *software* como 8 A, verifica-se que à medida que o tempo do ensaio decorre e a corrente consumida pela carga aumenta, a tensão de saída permanece constante com um valor de 13.5 V devido ao facto de a corrente limite não ter sido atingida. Sendo que nesta função de teste, a leitura da tensão de *feedback* é feita de forma constante, assim que a corrente consumida pela carga excede o

valor máximo permitido, a MeaTec32 automaticamente faz o corte de alimentação da carga, colocando a sua saída a 0V de modo a proteger o sistema.

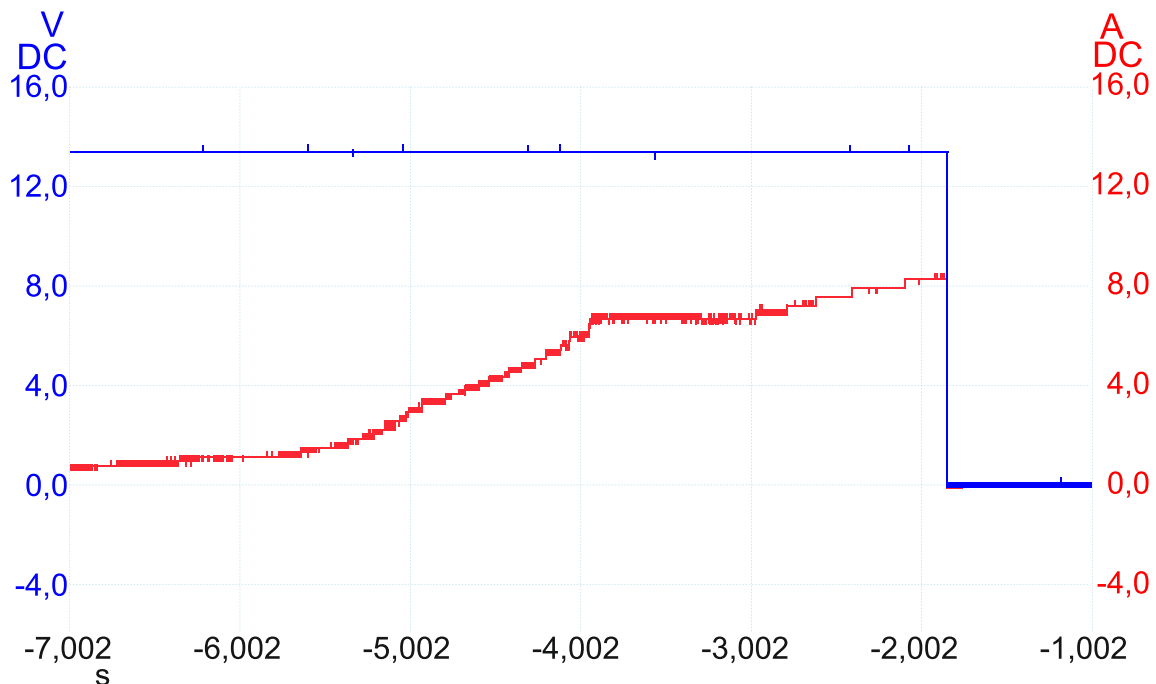


Figura 5.6 – Desativação de carga por excesso de consumo de corrente.

Certos sistemas podem sofrer de aquecimentos esporádicos ou outro tipo de fenómenos que provoquem o aumento momentâneo de corrente consumida pela carga sem que esta se encontre com defeito. Neste tipo de situações é possível estabelecer uma condição de leitura periódica da tensão de *feedback*, para que assim que a corrente consumida pela carga diminua, a MeaTec32 automaticamente ative novamente essa carga. Ao contrário de um fusível que quando é ultrapassado o valor de corrente para o qual foi projetado, interrompe o circuito definitivamente e é necessário ser substituído, este tipo de funcionalidade é largamente benéfica na utilização no veículo automóvel moderno. Não se efetuou o teste de desativação por temperatura, por ser um teste destrutivo. A temperatura de corte especificada para o MOSFET AUIR3314S é de 165 °C, o que iria acabar por danificar a PCB da MeaTec32.

5.4. Comunicações

Nesta secção referem-se os testes realizados às comunicações da MeaTec32, nomeadamente os testes de comunicação CAN e da comunicação série.

Comunicação CAN

Na comunicação CAN, o teste foi realizado sobre os três barramentos existentes na MeaTec32, de modo a verificar se não existem erros de ligação entre os componentes referentes à comunicação CAN, tais como controladores e *transceiver*, e se o funcionamento destes está de acordo com o esperado. Os primeiros barramentos a serem testados foram os do CAN convencional e só depois se avançou para o teste do CAN FD. Para tal recorreu-se à função de descodificação série do osciloscópio. A escolha deste equipamento para a realização de testes face a outro dispositivo que comunique através de CAN, como por exemplo outra unidade de controlo ou um visor, deve-se ao facto da possibilidade de monitorização dos sinais elétricos em tempo real do barramento CAN. Também as opções de visualização de dados dentro da descodificação série oferecem uma vantagem que uma comunicação simples entre dois dispositivos CAN não disponibiliza.

CAN Convencional

Para a existência de comunicação CAN é necessário ocorrer troca de dados entre dois dispositivos, normalmente independentes um do outro. No entanto, para o teste de comunicação CAN convencional, colocou-se dois canais da MeaTec32 a comunicarem entre si, ou seja, no mesmo barramento os dispositivos existentes eram o CAN0 e o CAN1. Deste modo simula-se um ambiente real como seria de esperar num automóvel, onde vários dispositivos comunicam entre si. Na Figura 5.7 observa-se um pacote de dados capturado pela ferramenta de descodificação série do osciloscópio, a azul encontra-se representado a linha CAN *High* e a vermelho encontra-se a linha CAN *low*, que no fundo são um espelho uma da outra. A mensagem enviada, entre outros parâmetros, era caracterizada pelo identificador (ID) 100 e pela mensagem de 8 bytes que continha os dados 10 20 30 40 AA BB CC DD, a utilização destes parâmetros permitem validar o correto funcionamento da comunicação CAN convencional. Consegue-se verificar pela Figura 5.7 que a mensagem capturada possui o ID 100 e os dados da mensagem são os corretos, não existindo perda de informação durante o processo, garantido que a comunicação CAN convencional se encontra em correto funcionamento.

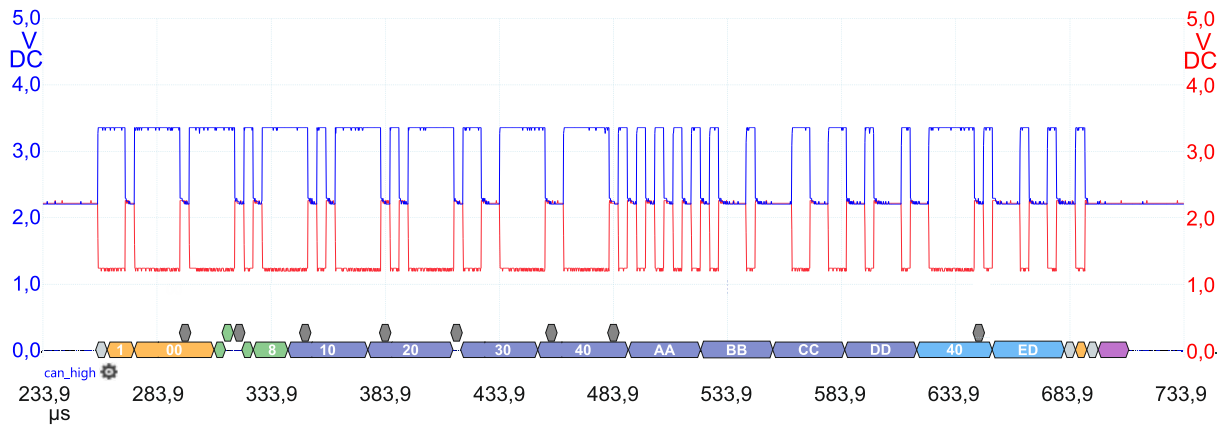


Figura 5.7 – Mensagem CAN convencional.

CAN FD

Para o teste funcional da comunicação CAN FD realizou-se a transmissão periódica de uma mensagem fixa, caracterizada pelo ID 123 e constituída por 64 bytes de dados, dos quais apenas os primeiros quatro se encontravam preenchidos com a mensagem “10 11 12 13” por opção própria. Através da ferramenta de descodificação série efetuou-se a monitorização dos pacotes de dados transmitidos para o barramento CAN FD. O objetivo foi verificar se a ferramenta de descodificação série conseguia automaticamente, depois de configurada, detetar a presença da mensagem completa sem que existisse perda de informação. Na Figura 5.8 encontra-se uma mensagem CAN capturada pelo osciloscópio, onde através da análise desta mensagem se verifica que o ID corresponde ao da mensagem transmitida, bem como os 64 bytes de dados transmitidos. Nomeadamente os dados “10 11 12 13” e os restantes preenchidos a “0”, foram capturados com sucesso. Dada a quantidade de dados da mensagem, não é possível obter uma boa resolução de imagem enquanto que se captura todos os dados da mensagem. Concluiu-se assim que todos os componentes referentes à comunicação CAN FD se encontram validados.

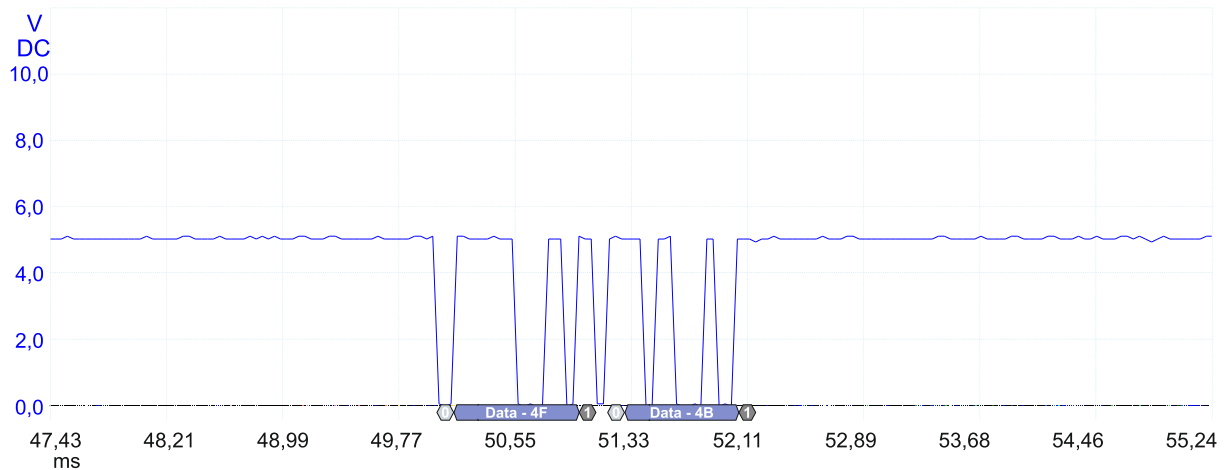


Figura 5.9 – Leitura da linha de transmissão de dados de comunicação série.

5.5. Inspeção por termografia da PCB

As condições de teste foram definidas como o controlo de 4 cargas resistivas em funcionamento constante durante cerca de 15 minutos. Estas cargas, alimentadas a 12 V, consomem individualmente cerca de 5 A, fazendo com que a corrente total percorrida na MeaTec32 seja cerca de 20 A. Para que o multímetro térmico consiga efetuar a leitura da temperatura superficial dos componentes, a medição foi realizada com a PCB removida do seu encapsulamento e exposta inicialmente a uma temperatura ambiente de 18 °C. Observa-se, na Figura 5.10 a disposição dos equipamentos utilizados para a realização deste teste, em que as medições foram realizadas de modo estático à medida que decorria o teste.

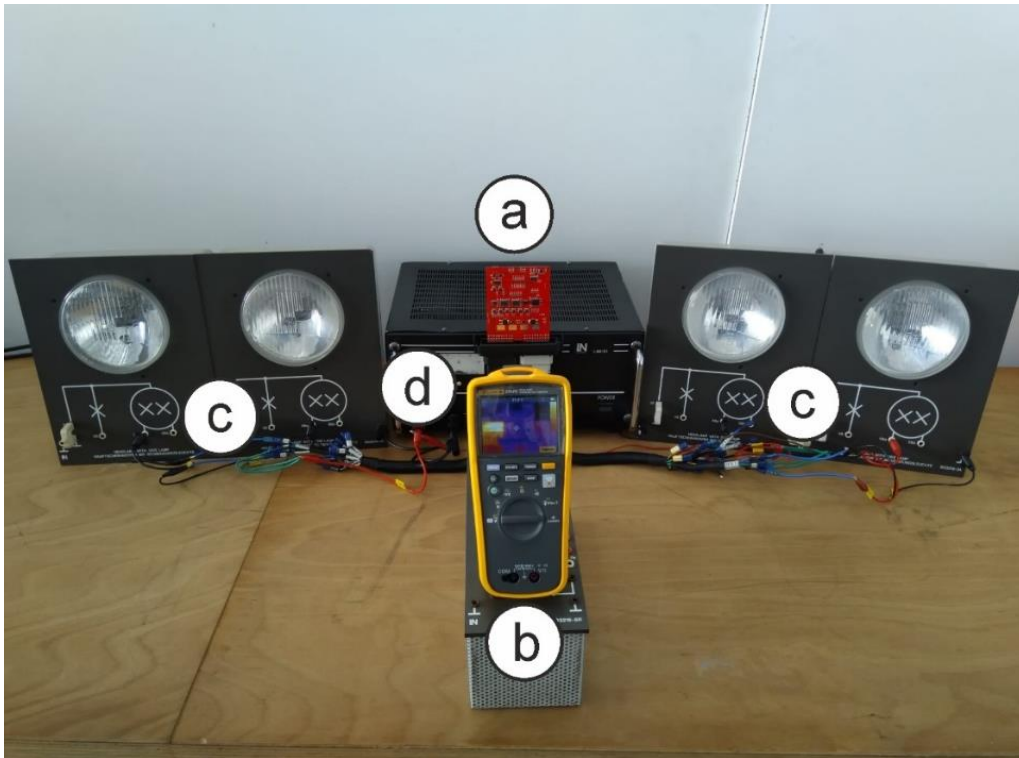


Figura 5.10 – Disposição de equipamentos para inspeção por termografia: (a) módulo desenvolvido, (b) multímetro térmico, (c) cargas genéricas, (d) fonte de alimentação.

Na Figura 5.11 observa-se a temperatura de funcionamento dos MOSFET's de potência para as condições do teste realizado. A temperatura alcançada ao fim de 15 minutos, que se mostrou constante foi de 56.3 °C, inferior à temperatura máxima de 165 °C, especificada pelo fabricante como a temperatura de *shutdown* [36]. De referir que nas mesmas condições de teste, com a MeaTec32 fechada no seu encapsulamento, seria de esperar que a temperatura atingida fosse mais elevada. No entanto, é pouco provável que, numa aplicação final, sejam atuados simultaneamente os 4 MOSFET's de potência à capacidade máxima. Na maioria das aplicações espera-se a atuação seletiva dos MOSFET's de potência, muitas das vezes com recurso a controlo PWM. Também é possível perceber a localização dos principais componentes e transferência de calor decorrida, para melhor enquadramento pode-se consultar novamente a Figura 4.19 (a), que demonstra a localização dos componentes no plano superior da MeaTec32.

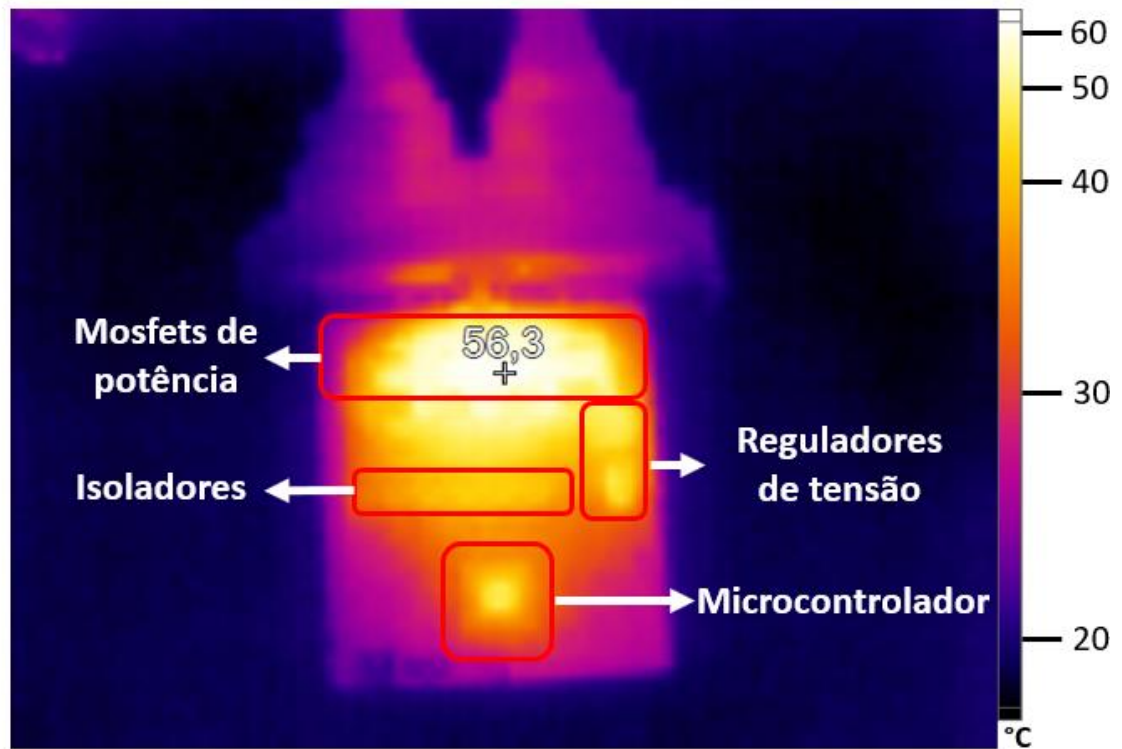


Figura 5.11 – Temperatura de funcionamento dos MOSFET's de potência.

De modo a analisar a evolução da temperatura em função do tempo decorrido, observa-se, na Figura 5.12, um ensaio retirado do multímetro térmico utilizado. A duração do teste, limitada pela ferramenta em si, foi de 10 minutos e atingiu-se uma temperatura de 56.2 °C, que em conjunto com o resultado da Figura 5.11, permite concluir que a temperatura fica estabilizada por volta dos 60 °C.

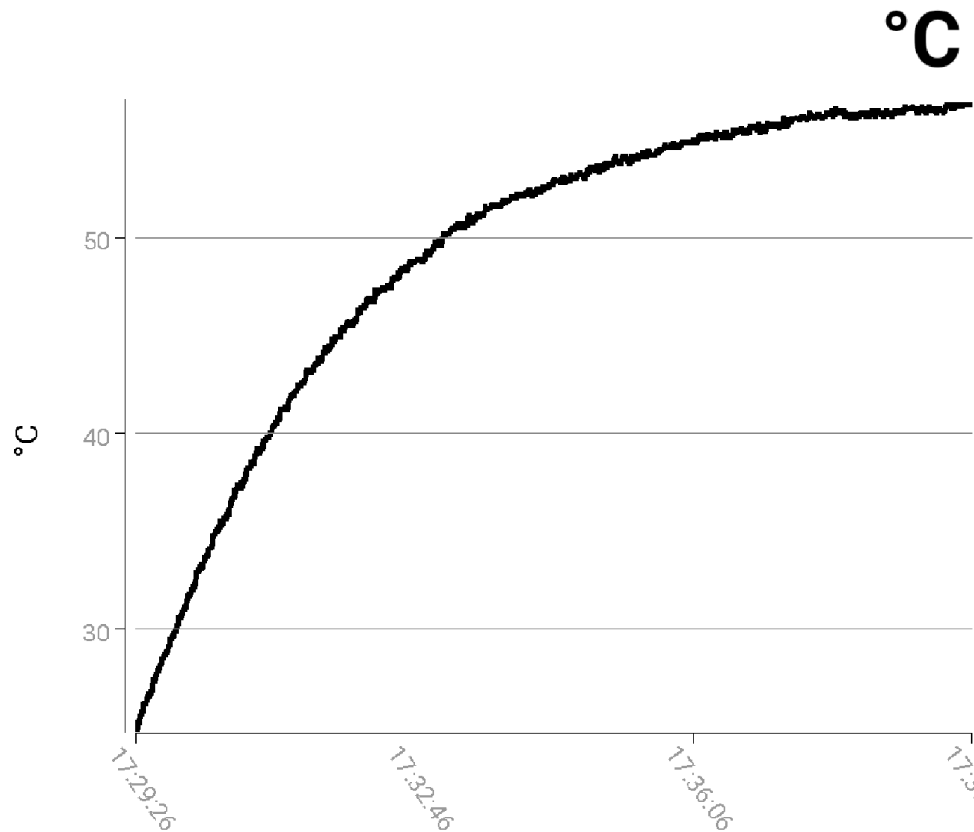


Figura 5.12 – Evolução da temperatura de funcionamento dos MOSFET's de potência.

Na Figura 5.13 observa-se, nas mesmas condições do teste anterior, o microcontrolador a uma temperatura estabilizada de 46 °C. Outro tipo de teste consistiu na comparação da temperatura do microcontrolador da MeaTec32 com o microcontrolador de um Arduino® Due, que são idênticos. O teste foi realizado com ambos os microcontroladores a desempenhar tarefas idênticas e após uma hora decorrida, a diferença de temperatura comparativamente ao Arduino® Due era de apenas mais 1 °C. A temperatura de funcionamento máxima especificada para este microcontrolador é de 85 °C [33], sendo assim possível afirmar que se encontra a funcionar dentro das especificações definidas pelo fabricante. De salientar novamente que seria expectável uma temperatura mais elevada para um teste realizado com a MeaTec32 dentro do encapsulamento.

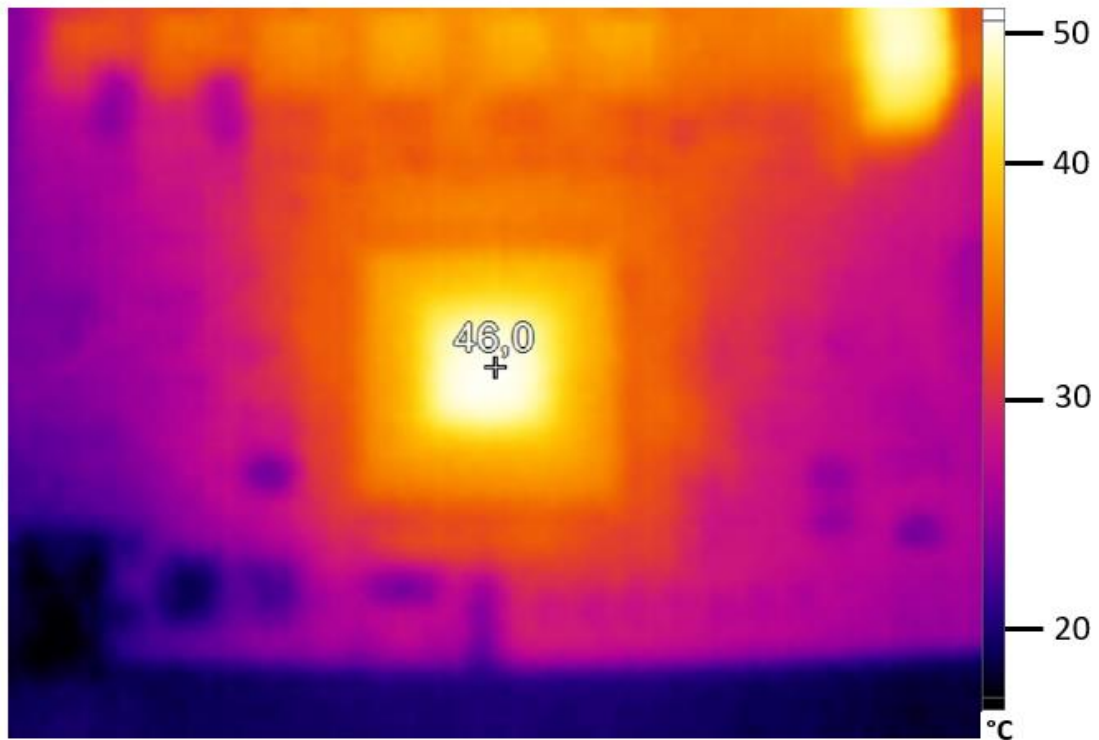


Figura 5.13 – Temperatura de funcionamento do microcontrolador.

Os reguladores de tensão, localizados no lado direito da PCB, atingem uma temperatura de 40 °C em funcionamento normal, abaixo da temperatura máxima de 85 °C definida pelo fabricante [40]. Também foi verificada a temperatura de funcionamento dos MOSFET's genéricos no controlo de cargas de 5 A, em que a temperatura alcançada foi de 46.4 °C, inferior à temperatura de funcionamento máxima de 175 °C referida pelo fabricante [41].

Com os resultados dos testes, ainda que sem o encapsulamento, é possível concluir que as temperaturas observadas se encontram dentro dos limites de segurança definidos pelos fabricantes de cada componente, garantido o correto funcionamento da MeaTec32.

5.6. Simulador de veículo eléctrico

Para a aplicação da MeaTec32 num ambiente semelhante ao do veículo eléctrico, realizou-se a adaptação de uma estrutura em perfil Bosch® para servir de simulador de um veículo eléctrico utilitário. Com esta montagem experimental é possível avaliar o desempenho da MeaTec32 quando exposta a ruídos, interferências e vibrações causadas pelos componentes que se

encontram num veículo elétrico utilitário. Para tal foram utilizados diversos componentes tais como um módulo de baterias, conversor DC-DC, controlador do motor, motor elétrico, contactor elétrico, painel de instrumentos, acelerador, sensores e alguns interruptores.

O módulo de baterias utilizado pertence a outro simulador, já existente, que faz parte de uma estação de carregamento elétrico presente no Laboratório de Engenharia Automóvel. A tensão fornecida pelas baterias é de 48 V DC, com uma capacidade nominal de 20 Ah a uma taxa de descarga de 3 C em contínuo (máximo de 15 minutos) e 10 C num máximo de 5 segundos.

Para alimentação dos componentes de baixa tensão (12 V), tais como a MeaTec32 e o painel de instrumentos, utilizou-se um conversor DC-DC isolado CQB150W-48S12 da Cincon[®] que regula a tensão de 48 V para 12 V, com uma corrente máxima de saída de 12.5 A.

O controlador do motor elétrico utilizado é fabricado pela empresa Curtiss-Wright Corporation, o modelo em questão é o Sigmadrive PMT445M. Este tipo de controlador é destinado a sistemas de 24 a 48 V, para o controlo de motores DC (monofásicos) e AC (com três fases) [42]. A configuração, realizada através de um programador específico, é feita através de comunicação CAN. Esta empresa fornece um visor próprio que recebe dados do controlador através de um barramento CAN específico e que apresenta esses dados ao utilizador. No entanto, este visor não foi adquirido, existindo apenas o controlador. Como tal, a MeaTec32 é responsável, entre outras tarefas, por adquirir os dados do controlador via CAN e mostrar no painel de instrumentos disponível, que foi retirado de um Volkswagen[®], as informações relativas ao funcionamento do simulador elétrico.

O motor elétrico disponibilizado, que é fabricado pela empresa britânica Lynch Motor Company Ltd, é um LEM-200. Este motor DC de ímanes permanentes é normalmente utilizado em barcos de pequeno porte, motociclos e *kartings*, sendo caracterizado por uma potência nominal de 8.5 kW, e por uma corrente máxima de 215 A. A rotação deste motor varia a uma taxa de 60 rpm por cada Volt aplicado aos terminais do mesmo e consegue atingir uma eficiência até 93 % [43]. Na Figura 5.14 observa-se o motor aplicado no simulador, onde foi necessário efetuar as fixações do mesmo, bem como as ligações elétricas entre este, o controlador e a MeaTec32.



Figura 5.14 – Motor Lynch LEM-200.

Uma vez que este motor eléctrico apenas possui um sensor de temperatura incorporado e não existe uma saída que indique a rotação do motor, foi necessário aplicar externamente um sensor de efeito Hall que consiga obter esta informação. O sensor utilizado, de referência SNG-QPCA-001 da Honeywell[®], possui duas saídas com sinais distintos para que seja possível identificar o sentido de rotação do motor.

O controlador efetua a gestão da alimentação através de uma resistência de *pre-charge* e um contactor que completam o circuito de alimentação assim que o controlador verifique que não existem erros atuais no sistema. O contactor disponibilizado pela empresa Tyco Electronics[®] permite ser utilizado em sistemas até 450 V DC e suporta uma corrente de 500 A a uma temperatura de 65 °C. Na Figura 5.15 observa-se o painel que se realizou, constituído pelo sistema de contactor e *pre-charge*, bem como alguns fusíveis de segurança.

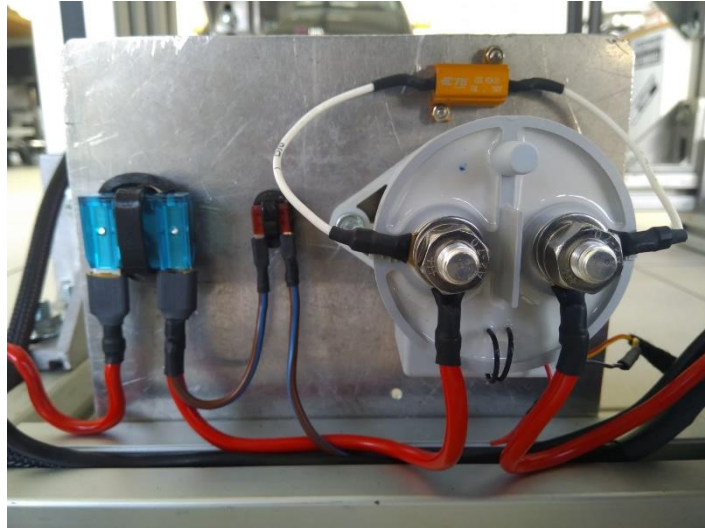


Figura 5.15 – Sistema de contactor e *pre-charge*.

O painel de instrumentos disponibilizado pertence ao grupo Volkswagen® e possui mostradores para a monitorização de velocidade, rotação, temperatura, entre outras variáveis. A *interface* entre este painel de instrumentos e a MeaTec32 é realizada através de comunicação CAN, para que o utilizador do simulador consiga assim visualizar dados relativos ao funcionamento deste sistema. Na Figura 5.16 observa-se o painel de instrumentos e o painel de comando fabricado.



Figura 5.16 – Painel de instrumentos e painel de comando.

A MeaTec32 desempenha o papel central neste simulador do VE, assumindo a gestão de todo o sistema. Através dos seus canais de comunicação CAN é estabelecida a ligação de dados

entre o controlador, a MeaTec32 e o painel de instrumentos. Além dos dados do controlador adquiridos através de um dos barramentos CAN, é realizada ainda a aquisição de sinais externos, analógicos e digitais. Após o processamento de toda esta informação externa e do controlador, é realizado o envio de todos os dados necessários para o painel de instrumentos. Através da conexão USB, com o computador, é ainda possível programar a MeaTec32 em tempo real e também obter dados do sistema.

Na Figura 5.17 observa-se o resultado do simulador eléctrico realizado, onde é possível verificar a disposição estratégica de todos os componentes. Na realização desta montagem foram seguidas as indicações fornecidas pelos fabricantes de todos os equipamentos, de modo a garantir a segurança e correto funcionamento de todo o sistema.

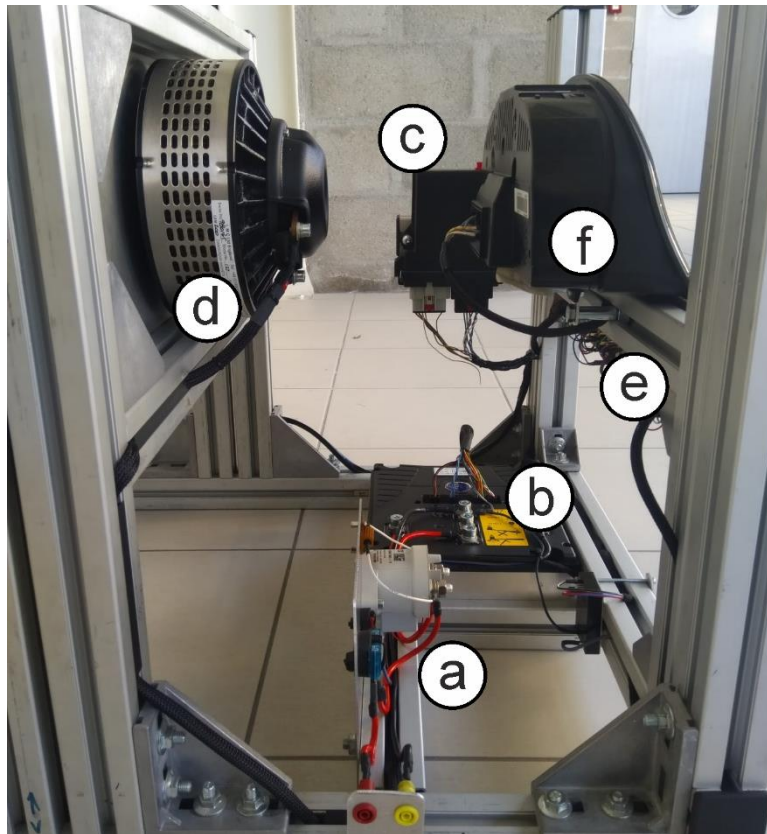


Figura 5.17 – Simulador eléctrico: (a) contactor e *pre-charge*, (b) controlador, (c) MeaTec32, (d) motor eléctrico, (e) painel de comando, (f) painel de instrumentos.

De modo a compreender por completo o funcionamento do simulador, observa-se na Figura 5.18 o esquema de ligações simplificado que foi utilizado nesta montagem experimental. Como se verifica, a tensão da bateria de 48V percorre o módulo de fusíveis de segurança e o

módulo de *pre-charge* e contactor até alimentar o controlador do motor. É ainda utilizado um conversor DC-DC para alimentação a 12V da MeaTec32 e do painel de instrumentos. O motor eléctrico encontra-se diretamente ligado ao controlador e implementado neste motor estão os sensores de temperatura e de rotação. Estes dois sinais são adquiridos pela MeaTec32 e posteriormente enviados ao painel de instrumentos. Também se consegue observar as ligações do barramento CAN0 e CAN1, que ligam a MeaTec32 ao painel de instrumentos e ao controlador.

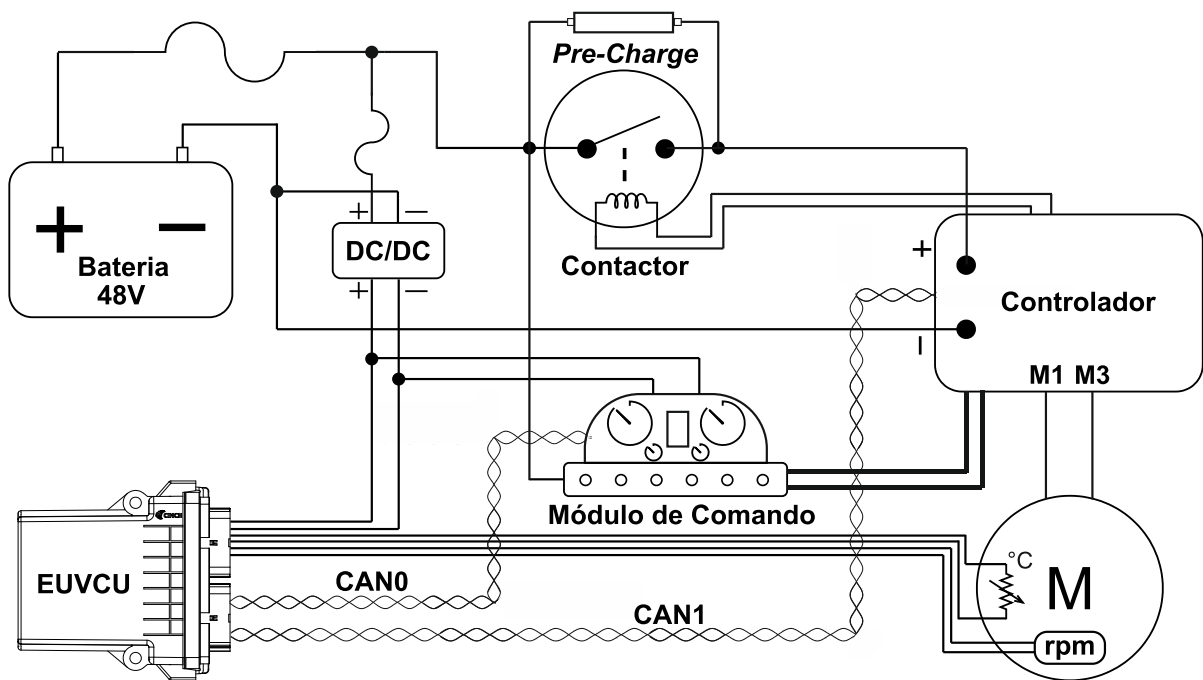


Figura 5.18 – Esquema de ligações do simulador.

5.7. Procedimentos de teste

Após realizada a montagem experimental, foram realizados os testes necessários para comprovar o funcionamento correto da MeaTec32 quando aplicada numa arquitetura de um veículo eléctrico. Nesta montagem, a MeaTec32, assume a responsabilidade de unidade central, sendo responsável pela interação entre o controlador e o painel de instrumentos, acumulando ainda algumas funções externas.

5.7.1. Aquisição de sinais externos

Sendo que o controlador não disponibiliza no seu protocolo CAN informações relativas à temperatura e rotação do motor elétrico, foi necessário instrumentar a montagem experimental de modo a que estas informações sejam disponibilizadas ao utilizador. Como referido anteriormente na secção 5.7, a rotação do motor elétrico é providenciada pelo sensor de Hall que foi aplicado no veio do motor. A MeaTec32, através das suas entradas digitais isoladas, é capaz de realizar a aquisição do sinal de rotação e efetuar o processamento deste para que consiga apresentar um valor de rotação ao utilizador.

A temperatura do motor já se encontra disponibilizada através de um sensor de temperatura do tipo PTC (*Positive Temperature Coefficient*) implementado pelo fabricante em questão. Através de uma das entradas analógicas da MeaTec32 que se encontra preparada para leitura de sensores do tipo termistor, é realizada a aquisição do sinal analógico do sensor de temperatura.

Após a aquisição e processamento interno destas informações, os dados são enviados através do barramento CAN para o painel de instrumentos que é detalhado na secção 5.7.3.

5.7.2. Comunicação com o controlador

A comunicação entre a MeaTec32 e o controlador é feita através do barramento CAN1 da MeaTec32, sendo que o barramento CAN0 é destinado à comunicação com o painel de instrumentos, como se detalha na secção seguinte. Através de dados técnicos e da monitorização do barramento CAN do controlador, foi possível descodificar o protocolo de comunicação que este utiliza no envio de dados para o visor próprio. Os principais dados que o controlador disponibiliza no barramento CAN, que são relativos ao estado de funcionamento do sistema elétrico, podem ser observados na Tabela 8. O visor utilizaria estes dados para constituir a informação a mostrar ao utilizador, no caso desta montagem experimental, a MeaTec32 solicita informação ao controlador, do mesmo modo que o visor iria solicitar.

A informação que constitui o campo de dados referido na Tabela 8, foi programada pelo fabricante como sendo a rotação do motor em %V, unidade específica do fabricante do controlador. Todos os restantes dados são relativos a esta informação, a sinalização indica se o valor de rotação do motor é positivo ou negativo e as casas decimais permitem saber em

concreto a ordem de grandeza da rotação. A unidade do campo de dados permite completar a informação obtida no primeiro ponto, que se encontra associada de forma gráfica a uma barra de dados. Por último, o controlador fornece informação sobre o estado da bateria, sob a forma de um ícone de uma bateria, que se encontra dividido em 10 segmentos.

Tabela 8 – Dados disponibilizados pelo controlador [42].

Dados	Designação
<i>Field Value</i>	Campo de dados
<i>Signed/Unsigned</i>	Sinalização do campo de dados
<i>Decimal Point</i>	Casas decimais do campo de dados
<i>Field Unit</i>	Unidade do campo de dados
<i>Bargraph</i>	Barra de dados gráfica do campo de dados
<i>BDI Graph</i>	Gráfico do estado da bateria

O controlador envia para o barramento, após o *handshake* que é detalhado na secção 5.7.4, uma só mensagem com o ID300, que engloba todas as variáveis associadas ao funcionamento do controlador. Na Tabela 9 observa-se a composição dos 8 bytes da mensagem CAN enviada pelo controlador, que é decodificada pela MeaTec32.

Tabela 9 – Mensagem CAN de dados do controlador.

Byte	Dados
0	Valor do campo de dados (MSB)
1	Valor do campo de dados (LSB)
2	Valor da barra de dados gráfica
3	Valor de casas decimais e sinalização
4	Primeiro caractere da unidade do campo de dados
5	Segundo caractere da unidade do campo de dados
6	Terceiro caractere da unidade do campo de dados
7	Valor do estado da bateria

5.7.3. Comunicação com o painel de instrumentos

Após a aquisição e processamento das informações detalhadas nas secções 5.7.1 e 5.7.2, a MeaTec32 efetua o envio dos dados para o painel de instrumentos do utilizador através do barramento CAN0. Este painel, na sua aplicação original, recebia informação da ECU do veículo para a apresentação de rotação do motor, temperatura do motor, velocidade do veículo e alertas que permitem informar o condutor de que existe algum erro no sistema. Também recebia informação analógica e digital, por exemplo para apresentação do nível de combustível através de um sensor resistivo, que na aplicação atual não é utilizada. No barramento CAN da MeaTec32 são enviados os seguintes dados para o painel de instrumentos:

- Informação relativa ao sistema operacional do veículo, para que o painel de instrumentos possa desligar os avisos de erros de *airbag* e controlo de tração;
- Velocidade de rotação do motor elétrico, apresentada no mostrador de rotação do motor de combustão interna;
- Temperatura do motor elétrico, apresentada no mostrador de temperatura do motor de combustão interna;
- Barra de dados relativa à rotação do motor em %V. Esta barra de dados varia de 0 a 50, conforme a rotação do motor e é apresentada no mostrador de velocidade do veículo;
- Aviso de temperatura do motor elétrico e da MeaTec32.

5.7.4. Desenvolvimento de código

Tendo em consideração as funções que a MeaTec32 necessita de desempenhar, detalhadas nas secções anteriores, foi desenvolvido o código que permite controlar a montagem experimental. Para tal, foi utilizado o *software* programação Eclipse IDE, que se encontra disponível através da Eclipse FoundationTM, sem custos de aquisição ou de utilização. O ambiente de programação deste *software* pode-se observar na Figura 5.19, quando comparado por exemplo ao *software* Arduino IDE, é bastante superior. Dispõe funcionalidades que o Arduino IDE não possui, tais como a edição instantânea de bibliotecas, de funções, entre outras.

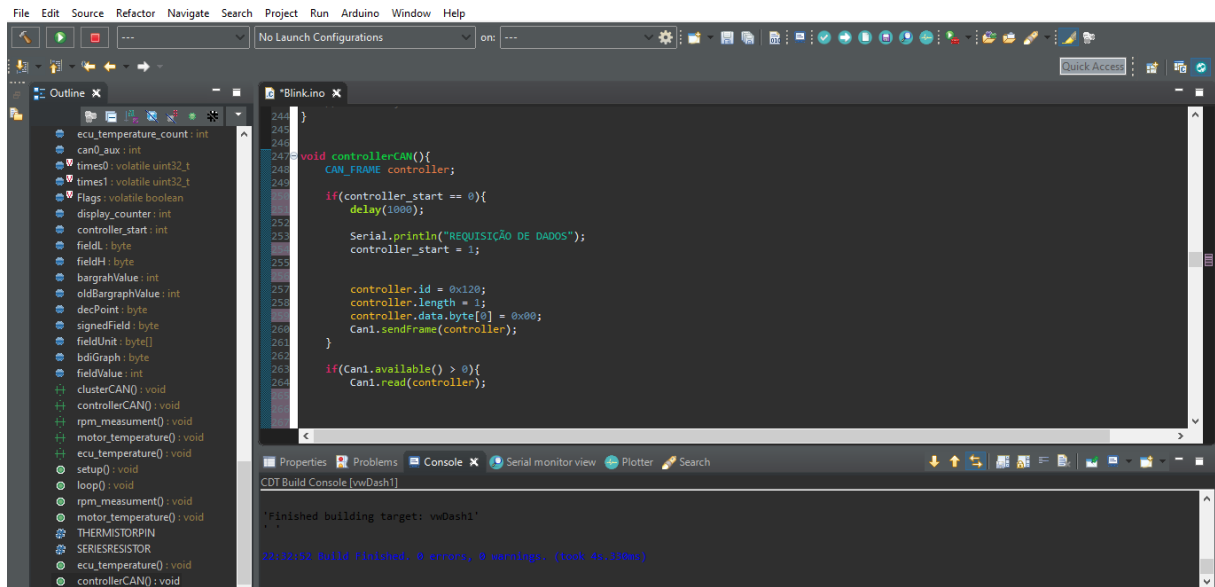


Figura 5.19 – Ambiente de programação do software Eclipse IDE.

A arquitetura de código utilizada foi do tipo *foreground/background*, onde existem um conjunto de tarefas que são executadas num ciclo sem fim, existindo uma interrupção específica para a leitura de rotação do motor eléctrico. O fluxograma que representa este código encontra-se na Figura 5.20.

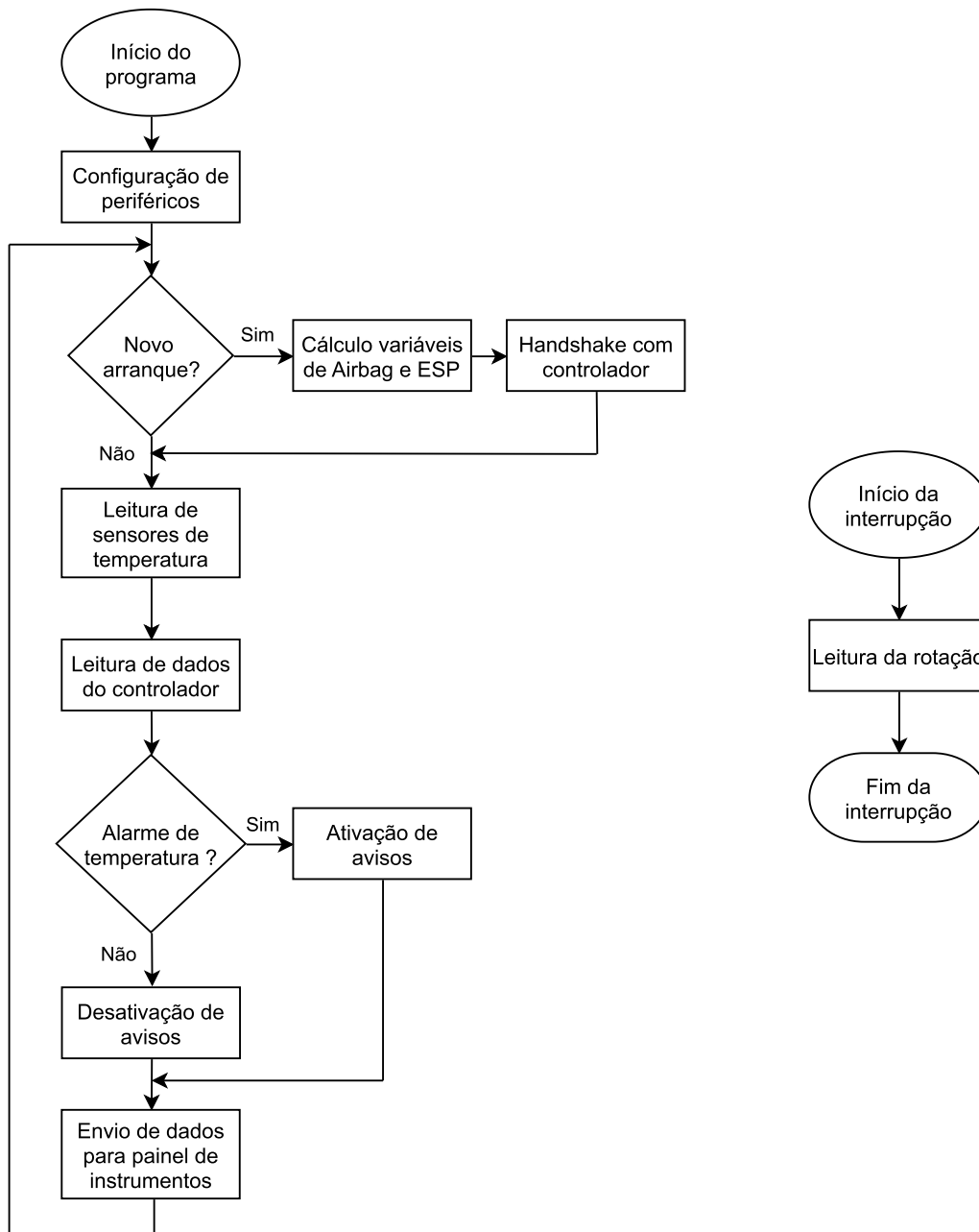


Figura 5.20 – Fluxograma do programa de testes.

No arranque de cada ciclo é realizado o cálculo das variáveis associadas ao sistema de *airbag* e de controlo de tração que permitem ao painel de instrumentos perceber que não existem erros. Também é realizado o *handshake* com o controlador Sigmadrive PMT445M, em que este envia uma mensagem com o ID110 para o barramento e espera que lhe seja enviada uma mensagem com o ID100, que indica a existência de outro nó presente no barramento. Só depois deste *handshake* é que o controlador começa a enviar dados relativos ao funcionamento do mesmo. Após realizada esta tarefa é adquirido e processado o sinal do

sensor de temperatura que se encontra incorporado no motor elétrico. De seguida são obtidos os dados que o controlador envia em *broadcast*, sendo estes dados posteriormente processados para envio futuro.

A leitura da rotação do motor elétrico é feita de forma constante e através de uma interrupção que deteta a mudança do estado do sinal digital do sensor de rotação externo.

Ainda antes do envio da informação recolhida, é verificado se existe algum alarme de temperatura ativo. Este alarme pode ser despoletado através do sensor de temperatura do motor e do sensor de temperatura integrado na MeaTec32, caso exista algum alarme ativo, essa informação é colocada nas mensagens CAN a enviar ao painel de instrumentos.

Após a aquisição e processamento de toda a informação anteriormente descrita é enviado um conjunto de mensagem CAN para o painel de instrumentos, após a execução de todas estas tarefas, o ciclo volta a ser inicializado. Estas mensagens enviadas possuem identificadores distintos, consoante o mostrador a atuar no painel de instrumentos. Para indicação visual de erros no sistema, são utilizadas duas lâmpadas de aviso já existentes no painel de instrumentos, que permitem informar o utilizador se existe algum alarme de temperatura e qual a sua origem.

Conseguiu-se assim desenvolver um código simples e prático que permite controlar com sucesso a montagem experimental. A memória do microcontrolador utilizada pelo código é cerca de 4% da sua memória de dados do programa, para executar todo o código. Isto demonstra que existe capacidade do microcontrolador em executar algoritmos mais complexos, sem sobrecarregar a memória do mesmo.

5.8. Discussão de Resultados

A escolha do microcontrolador mostrou-se acertada. Com a existência interna de dois controladores CAN foi possível economizar espaço e simultaneamente reduzir o custo associado ao protocolo CAN da MeaTec32.

O isolamento adicionado à MeaTec32 através da utilização de componentes isolados como o regulador de tensão, isoladores analógicos e digitais demonstrou ser funcional sem comprometer o funcionamento do sistema. Uma vez que a utilização destes componentes não

implicam configurações adicionais por parte do utilizador, existe efetivamente um aumento da robustez, mantendo a sua simplicidade. As funcionalidades associadas às saídas digitais, nomeadamente o *feedback* e *shutdown* disponibilizado pelos MOSFET's "inteligentes", demonstraram ser precisas e adicionam à MeaTec32 uma fiabilidade acrescida.

Os testes de monitorização de temperatura permitiram garantir que as temperaturas de funcionamento não excedem os limites definidos pelos fabricantes dos respetivos componentes elétricos. A escolha em utilizar reguladores de tensão comutados face a reguladores de tensão lineares como foi referido na secção 4.2, justificou-se, uma vez que a temperatura de funcionamento atingida nestes componentes é relativamente baixa comparativamente a reguladores de tensão lineares que podem facilmente ultrapassar os 80 °C. No entanto, uma vez que os testes de temperatura foram realizados sem o encapsulamento da MeaTec32, é expectável que a temperatura aumente quando em funcionamento normal.

A implementação prática realizada num simulador de um veículo elétrico utilitário foi determinante para testar por completo o funcionamento da MeaTec32, foi possível verificar que o barramento CAN isolado funciona corretamente com outros tipos de dispositivos. Verificou-se que, apesar de todas as vibrações e ruídos que a estrutura do simulador do VE causava, não foram detetadas evidências da influência destes fenómenos no desempenho da MeaTec32. Esta implementação permite concluir que a MeaTec32 se encontrará apta para ser aplicada em diferentes veículos elétricos utilitários, o que demonstra a sua versatilidade.

6. Conclusões e Trabalhos Futuros

Este projeto teve como objetivo o desenvolvimento de uma unidade de comando para veículos elétricos utilitários. A revisão bibliográfica permitiu compreender a arquitetura do veículo elétrico bem como as soluções existentes no mercado para unidades de comando. Através da análise destas soluções foi possível estudar os requisitos para este tipo de módulos e identificar pontos de melhoria que podem ser providenciados a um preço acessível. O estudo de tecnologias de suporte possibilitou a compreensão de diferentes protocolos de comunicação e a aplicação de cada um dentro da unidade de comando desenvolvida. Também com a análise de arquiteturas de código de sistemas embebidos foi possível identificar a melhor estratégia de código a ser implementada neste projeto.

Foi projetado o circuito elétrico que permitiu responder às necessidades identificadas. Para tal foi feita a identificação dos componentes que constituem cada módulo da MeaTec32, realizando-se o desenvolvimento e montagem da PCB final com todos estes componentes integrados. Como resultado apresenta-se a unidade MeaTec32 com um custo de produção final aproximado de 158,87 €. E que se destinge por apresentar 3 barramentos CAN, diversos periféricos com isolamento elétrico, microcontrolador de 32 bit, entre outras características. Foi assim desenvolvida uma ECU totalmente funcional que com a realização dos testes de validação permitiu comprovar de forma individual cada uma das funcionalidades da MeaTec32. Com a implementação prática da MeaTec32 num simulador de veículo elétrico foi possível comprovar por completo o funcionamento desta unidade. Ficou assim demonstrado a versatilidade e funcionalidade que esta unidade apresenta.

A realização deste projeto permitiu atingir o objetivo inicialmente proposto, foi possível desenvolver uma ECU de baixo custo para veículos elétricos utilitários que permite o controlo de funções genéricas deste tipo de veículos. Também as funcionalidades de monitorização implementadas, bem como a utilização de uma plataforma *open source* acrescentam valor a este projeto no sentido em que pode ser facilmente replicado por entusiastas da área. Foi assim possível contribuir para uma comunidade que se encontra em rápido crescimento, efetuando cada vez mais conversões elétricas de veículos de todo o tipo.

O trabalho aqui realizado constitui apenas a primeira versão feita da MeaTec32, sendo que como em qualquer projeto, é necessário dar continuidade ao melhoramento com a introdução e aperfeiçoamento de algumas funcionalidades. De seguida são apresentadas propostas para trabalhos futuros:

- Utilização de um encapsulamento que permita a dissipação de calor gerada pelos dispositivos de potência para o exterior;
- Escolha de MOSFET's mais eficientes que permitam menos perdas para a mesma corrente;
- Escolha de um amplificador de instrumentação que opere de forma linear tendo em consideração a tensão de alimentação da MeaTec32;
- Desenvolvimento de uma *interface* em Labview que permita ao utilizador efetuar configurações do sistema e simultaneamente visualizar dados em tempo real;
- Implementação de uma funcionalidade para leitura de corrente;
- Utilização de isoladores digitais para leitura de sinais digitais até 12V.

Referências Bibliográficas

- [1] C. C. Chan and K. T. Chau, *Modern Electric Vehicle Technology (Monographs in Electrical and Electronic Engineering)*. 2001.
- [2] A. C. Madrigal, “The Electric Taxi Company You Could Have Called in 1900 - The Atlantic,” 2011. [Online]. Available: <https://www.theatlantic.com/technology/archive/2011/03/the-electric-taxi-company-you-could-have-called-in-1900/72481/>. [Accessed: 01-Mar-2019].
- [3] S. R. Gomes, “Máquinas Elétricas I: Aula 13 - Funcionamento de Motor de Indução Trifásico,” 2015. [Online]. Available: <http://maquinaseletricas.blogspot.com/2016/03/manutencao-industrial-aula-11-inspecao.html>. [Accessed: 01-Mar-2019].
- [4] Infineon Technologies, “Brushless DC motor (BLDC).” [Online]. Available: <https://www.infineon.com/cms/en/applications/solutions/motor-control-and-drives/brushless-dc-motor/>. [Accessed: 09-Jan-2020].
- [5] M. Brain, “Inside an Electric Car.” [Online]. Available: <https://auto.howstuffworks.com/electric-car2.htm>. [Accessed: 09-Jan-2020].
- [6] E. Hendrickson, “Optimize Your Vehicle by Cooling Electric Motors and Generators,” 2017. [Online]. Available: <https://www.machinedesign.com/motion-control/optimize-your-vehicle-cooling-electric-motors-and-generators>. [Accessed: 05-Mar-2019].
- [7] Y. Xing, E. W. M. Ma, K. L. Tsui, and M. Pecht, “Battery management systems in electric and hybrid vehicles,” *Energies*, vol. 4, no. 11, pp. 1840–1857, 2011.
- [8] W. B. Ribbens, *Understanding Automotive Electronics*. 2013.
- [9] Aradex, “ECU Electronic Control Unit.” [Online]. Available: <https://www.aradex.de/en/products/ecu-electronic-control-unit/>. [Accessed: 07-Jan-2020].
- [10] M. M. Canteli, “Sensors,” *Handb. Automot. Power Electron. Mot. Drives*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [11] Robert Bosch GmbH (Ed.), *Bosch Automotive Electrics and Automotive Electronics : Systems and Components, Networking and Hybrid Drive*. 2014.
- [12] Vector, “Vector Controllers VC,” 2019.

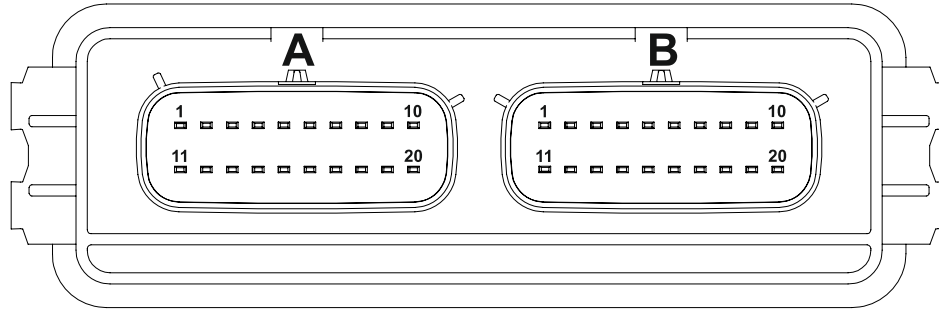
- [13] ECOTRONS LLC, “ECOTRONS EV2206B03.”
- [14] TTControl Hydac International, “HY-TTC 30H Control Unit.”
- [15] EVTV Motor Verks, “EVTV Motor Verks Store: Generalized Electric Vehicle Control Unit (GEVCU) Version 6.” [Online]. Available: <http://store.evtv.me/proddetail.php?prod=gevcu6>. [Accessed: 08-Jan-2020].
- [16] CAN in Automation (CiA), “CAN lower-and higher-layer protocols.” [Online]. Available: <https://www.can-cia.org/can-knowledge/>. [Accessed: 18-Mar-2019].
- [17] S. Corrigan, “Introduction to the Controller Area Network (CAN) Application Report Introduction to the Controller Area Network (CAN),” no. August 2002, pp. 1–17, 2002.
- [18] J. Griffith, “What do CAN bus signals look like?,” 2015. [Online]. Available: https://e2e.ti.com/blogs_/b/industrial_strength/archive/2015/06/04/what-do-can-bus-signals-look-like. [Accessed: 19-Mar-2019].
- [19] W. Stuart, “Introduction to Controller Area Network (CAN).” .
- [20] S. P. Annamalai, “Controller Area Network (CAN) - In Detail,” 2018. [Online]. Available: <http://www.embien.com/blog/controller-area-network-can-in-detail/>. [Accessed: 20-Mar-2019].
- [21] R. Bosch, “CAN Specification Version 2.0,” *Rober Bousch GmbH, Postfach*, vol. 300240, p. 72, 1991.
- [22] CAN in Automation (CiA), “CAN FD - The basic idea.” [Online]. Available: <https://www.can-cia.org/can-knowledge/can/can-fd/>. [Accessed: 07-Dec-2019].
- [23] National Instruments, “Understanding CAN with Flexible Data-Rate (CAN FD),” 2019. [Online]. Available: <https://www.ni.com/pt-pt/innovations/white-papers/14/understanding-can-with-flexible-data-rate--can-fd-.html>. [Accessed: 07-Dec-2019].
- [24] CSS Electronics, “CAN FD Explained - A Simple Intro (2019).” [Online]. Available: <https://www.csselectronics.com/screen/page/can-fd-flexible-data-rate-intro/language/en>. [Accessed: 09-Dec-2019].
- [25] Texas Instruments, “Reference Guide,” 2006.
- [26] P. Dhaker, “Introduction to SPI Interface | Analog Devices,” no. September, pp. 1–5, 2018.
- [27] Sparkfun, “Serial Peripheral Interface (SPI).” [Online]. Available: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>. [Accessed: 09-Dec-

2019].

- [28] Sparkfun, “Serial Communication.” [Online]. Available: <https://learn.sparkfun.com/tutorials/serial-communication>. [Accessed: 11-Dec-2019].
- [29] R. Oshana and M. Kraeling, *Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications*. 2013.
- [30] Infineon Technologies AG, “XC22xx Series - Infineon Technologies.” [Online]. Available: <https://www.infineon.com/cms/en/product/microcontroller/legacy-products-c500-c166-xc166-audio1-family/xc2200-family-body/xc22xx-series/>. [Accessed: 01-Apr-2020].
- [31] Arm Limited, “Cortex-M3 – Arm Developer.” [Online]. Available: <https://developer.arm.com/ip-products/processors/cortex-m/cortex-m3>. [Accessed: 01-Apr-2020].
- [32] V. Vaidya, “Low Cost Hardware Design Techniques for Robust and Reliable Power-Supply Circuits for Automotive ECUs,” 2012.
- [33] Atmel, “SAM3X / SAM3A Series datasheet,” *Atmel Corp.*, p. 1459, 2015.
- [34] “SERIES : PDS1-S | DESCRIPTION : DC-DC CONVERTER,” pp. 2–7, 2018.
- [35] A. Operating *et al.*, “AMC1311x-Q1 High-Impedance , 2-V Input , Reinforced Isolated Amplifiers,” 2018.
- [36] I. R. Rec, “PROGRAMMABLE CURRENT SENSE HIGH SIDE SWITCH,” vol. 3314, pp. 1–16, 2010.
- [37] L. Seeed Technology Co., “Seeed Studio, The IoT Hardware enabler.” [Online]. Available: <https://www.seeedstudio.com/>. [Accessed: 16-Nov-2019].
- [38] Pico Technology, “PicoScope 2000 Series (A API),” 2011.
- [39] F. Corporation, “Fluke 279 FC Thermal Multimeter.”
- [40] CUI Inc, “SERIES : P78E-1000 | DESCRIPTION : NON-ISOLATED DC SWITCHING REGULATOR P78E XX-1000,” pp. 1–8, 2018.
- [41] ON Semiconductor, “NVD5117PL,” 2017.
- [42] PG Drives Technology, “Sigmadrive PM Traction - Technical Manual,” 2008.
- [43] Lynch Motor Company Ltd, “MOTORS Overview LEM-200.”

Anexos

Anexo A – Pinout da MeaTec32



A1 - BAT_POS12V (R)
A2 - BAT_NEG0V (BR)
A3 - SEN_0V_NISO (GY)
A4 - SEN_5V_NISO (Y)
A5 - MFET_LS (BL)
A6 - MFET_HS (G)
A7 - PFET_OUT1 (BL)
A8 - PFET_SUP1 (G)
A9 - PFET_OUT2 (BL)
A10 - MFET_SUP2 (G)
A11 - BAT_POS12V (R)
A12 - BAT_NEG0V (BR)
A13 - SEN_0V_NISO (GY)
A14 - SEN_5V_NISO (Y)
A15 - SERIAL_RX (BL/BK)
A16 - SERIAL_TX (GY/BK)
A17 - SEN_3V3_ISO (LG)
A18 - SEN_3V3_ISO (LG)
A19 - SEN_0V_ISO (PR)
A20 - SEN_0V_ISO (PR)

B1 - PFET_OUT3 (BL)
B2 - PFET_SUP3 (G)
B3 - PFET_OUT4 (BL)
B4 - PFET_SUP4 (G)
B5 - CAN0_HI (BL/BR)
B6 - CAN0_LO (GY/R)
B7 - CAN1_HI (BL/BR)
B8 - CAN1_LO (GY/R)
B9 - CAN2_HI (BL/BR)
B10 - CAN2_LO (GY/R)
B11 - DIGI4 (W/GY)
B12 - DIGI3 (BL/GY)
B13 - DIGI2 (BR/GY)
B14 - DIGI1 (G/GY)
B15 - AVI1_ISO (W/17)
B16 - AVI2_ISO (W/18)
B17 - AVI3_NISO (W/19)
B18 - AVI4_NISO (W/20)
B19 - AVI5_NISO (W/21)
B20 - AVI6_NISO (W/22)

Color Code:

(R) - RED
(BR) - BROWN
(GY) - GRAY
(Y) - YELLOW
(BL) - BLUE
(G) - GREEN
(BL/BK) - BLUE/BLACK
(GY/BK) - GRAY/BLACK
(BL/BR) - BLUE/BROWN

(GY/R) - GRAY/RED
(W/GY) - WHITE/GRAY
(BL/GY) - BLUE/GRAY
(BR/GY) - BROWN/GRAY
(G/GY) - GREEN/GRAY
(W/19) - WHITE 17
(W/N) - WHITE N
(W/22) - WHITE 22

ECUVEH Pinout

Mating Connector A: Molex 20 position keying A #0334722006

Pin Number	Designation	FullName	Function
A1	BAT_POS12V	BATTERY POSITIVE 12V	Battery positive power supply for the ECU
A2	BAT_NEG0V	BATTERY NEGATIVE 0V	Battery negative power supply for the ECU
A3	SEN_0V_NISO	SENSOR 0V NON-ISOLATED	Sensors negative 0V supply non-isolated
A4	SEN_5V_NISO	SENSOR 5V NON-ISOLATED	Sensors positive 5V supply non-isolated
A5	MFET_LS	MOSFET LOW-SIDE	12V mosfet low-side
A6	MFET_HS	MOSFET HIGH-SIDE	12V mosfet high-side
A7	PFET_OUT1	POWER MOSFET OUTPUT 1	Power mosfet output 1
A8	PFET_SUP1	POWER MOSFET SUPPLY 1	Power mosfet supply voltage 1
A9	PFET_OUT2	POWER MOSFET OUTPUT 2	Power mosfet output 2
A10	MFET_SUP2	POWER MOSFET SUPPLY 2	Power mosfet supply voltage 2
A11	BAT_POS12V	BATTERY POSITIVE 12V	Battery positive power supply for the ECU
A12	BAT_NEG0V	BATTERY NEGATIVE 0V	Battery negative power supply for the ECU
A13	SEN_0V_NISO	SENSOR 0V NON-ISOLATED	Sensors negative 0V supply non-isolated
A14	SEN_5V_NISO	SENSOR 5V NON-ISOLATED	Sensors positive 5V supply non-isolated
A15	SERIAL_RX	SERIAL COMMUNICATION RX	Serial communication receiver
A16	SERIAL_TX	SERIAL COMMUNICATION TX	Serial communication transmitter
A17	SEN_3V3_ISO	SENSOR 3.3V ISOLATED	Sensors positive 3.3V supply isolated
A18	SEN_3V3_ISO	SENSOR 3.3V ISOLATED	Sensors positive 3.3V supply isolated
A19	SEN_0V_ISO	SENSOR 0V ISOLATED	Sensors negative 0V supply isolated
A20	SEN_0V_ISO	SENSOR 0V ISOLATED	Sensors negative 0V supply isolated

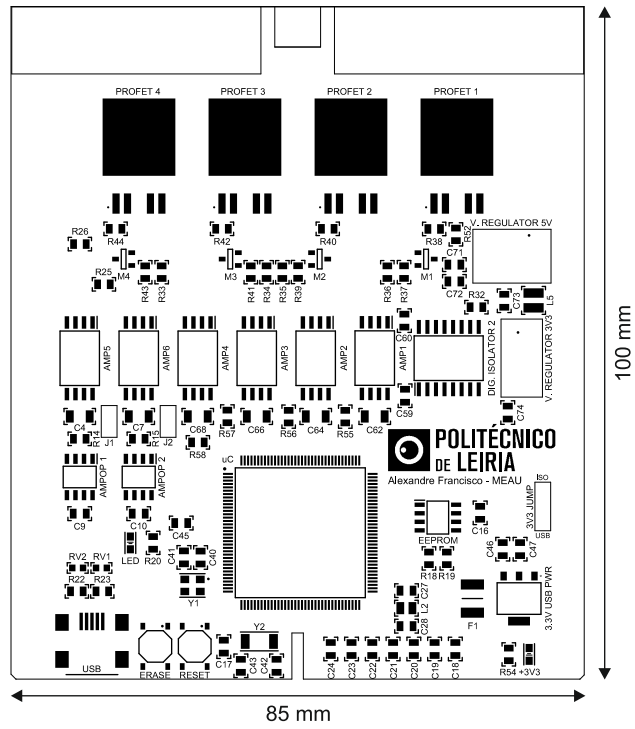
ECUVEH Pinout

Mating Connector B: Molex 20 position keying B #0334722007

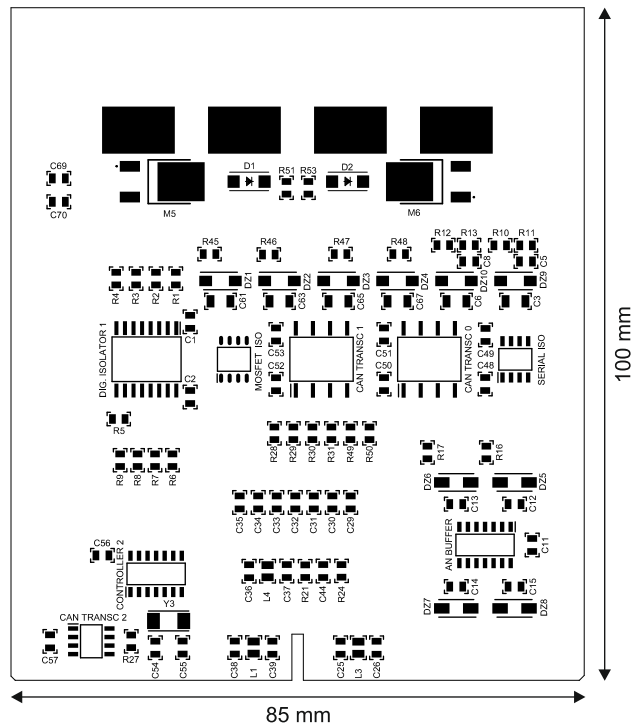
Pin Number	Designation	FullName	Function
B1	PFET_OUT3	POWER MOSFET OUTPUT 3	Power mosfet output 3
B2	PFET_SUP3	POWER MOSFET SUPPLY 3	Power mosfet supply voltage 3
B3	PFET_OUT4	POWER MOSFET OUTPUT 4	Power mosfet output 4
B4	PFET_SUP4	POWER MOSFET SUPPLY 4	Power mosfet supply voltage 4
B5	CAN0_HI	CAN BUS 0 HIGH	Can bus channel 0 high
B6	CAN0_LO	CAN BUS 0 LOW	Can bus channel 0 low
B7	CAN1_HI	CAN BUS 1 HIGH	Can bus channel 1 high
B8	CAN1_LO	CAN BUS 1 LOW	Can bus channel 1 low
B9	CAN2_HI	CAN BUS 2 HIGH	Can bus channel 2 high
B10	CAN2_LO	CAN BUS 2 LOW	Can bus channel 2 low
B11	DIGI4	DIGITAL INPUT 4	Digital input 4 with internal pull-up
B12	DIGI3	DIGITAL INPUT 3	Digital input 3 with internal pull-up
B13	DIGI2	DIGITAL INPUT 2	Digital input 2 with internal pull-up
B14	DIGI1	DIGITAL INPUT 1	Digital input 1 with internal pull-up
B15	AVI1_ISO	ANALOG VOLTAGE INPUT 1 ISOLATED	Analog voltage input 1 isolated
B16	AVI2_ISO	ANALOG VOLTAGE INPUT 1 ISOLATED	Analog voltage input 2 isolated
B17	AVI3_NISO	ANALOG VOLTAGE INPUT 3 NON ISOLATED	Analog voltage input 3 non-isolated
B18	AVI4_NISO	ANALOG VOLTAGE INPUT 4 NON ISOLATED	Analog voltage input 4 non-isolated
B19	AVI5_NISO	ANALOG VOLTAGE INPUT 5 NON ISOLATED	Analog voltage input 5 non-isolated
B20	AVI6_NISO	ANALOG VOLTAGE INPUT 6 NON ISOLATED	Analog voltage input 6 non-isolated

Anexo B – Vista 2D da MeaTec32

TOP VIEW



BOTTOM VIEW



Anexo C – Lista de componentes e custo da MeaTec32

QTD	Designação	Referência	Custo
1	Microcontrolador	ATSAM3X8EA-AU	8,71 €
1	Isolador Digital	VOD205T	0,87 €
1	Isolador Digital	MAX12931FASA+	2,02 €
6	Isolador Analógico	AMC1211AQDWVRQ1	32,76 €
2	Ampop Instrumentação	INA826AID	5,14 €
2	Transceiver CAN Isolado	ISO1050DUBR	8,54 €
1	Transceiver CAN FD	TCAN332GD	2,45 €
1	Controlador CAN FD	MCP2517FD-H/SL	2,02 €
4	Smart Mosfet	AUIR3314STRL	12,72 €
4	Mosfet de Sinal	BSS138	1,04 €
1	Mosfet N-Channel	NVD5C454NT4G	0,94 €
1	Mosfet P-Channel	NVD5117PLT4G-VF01	2,22 €
1	Regulador de Tensão 5V	P78E05-1000	4,29 €
1	Regulador de Tensão 3V3	PDS1-S5-S3-S	3,70 €
1	Bobine	SDR0302-6R8ML	0,36 €
1	Cristal 12 MHz	ECS-120-12-33Q-JES-TR	0,80 €
1	Cristal 32.768 kHz	ECS-.327-9-39-TR	0,73 €
1	Cristal 20 MHz	ECS-200-20-23A-EN-TR	0,88 €
1	Conectores Secundários	33472-2006/ 33472-2007	7,90 €
1	PCB		1€
1	Terminais p/ Conectores	33012-2001	6,20 €
6	Díodos Zener 5,1V	1SMA5918BT3G	1,69 €
4	Díodos Zener 3,3V	1SMA5913BT3G	1,57 €
2	Díodos	S2KDFQ-13	0,62 €
119	Condensadores/Resistências		3,57 €
4	Esferas de ferrite	MH2029-300Y	0,36 €
3	Headers	5-146285-3	1,11 €
3	Jumpers	151-8010-E	0,36 €
2	Supressores	CG0603MLC-05E	0,74 €
12	Condensadores 1206 0,1uF	1206YC104MAT2A	2,88 €
1	EEPROM	AT24C512C-SSHD-T	0,79 €
2	Micro switch	8-1437565-1	0,74 €
1	Conector USB	1734035-4	1,27 €
2	Led 0805	599-0191-007F	1,02 €
1	Regulador 3.3V	ZSR330GTA	1,01 €
2	Isolador Digital	ISO7740DWR	6,54 €
1	Conector Principal p/ PCB	5810140011	18,29 €
1	Encapsulamento p/ PCB	5810130065	11,02 €
	Custo Total		158,87 €