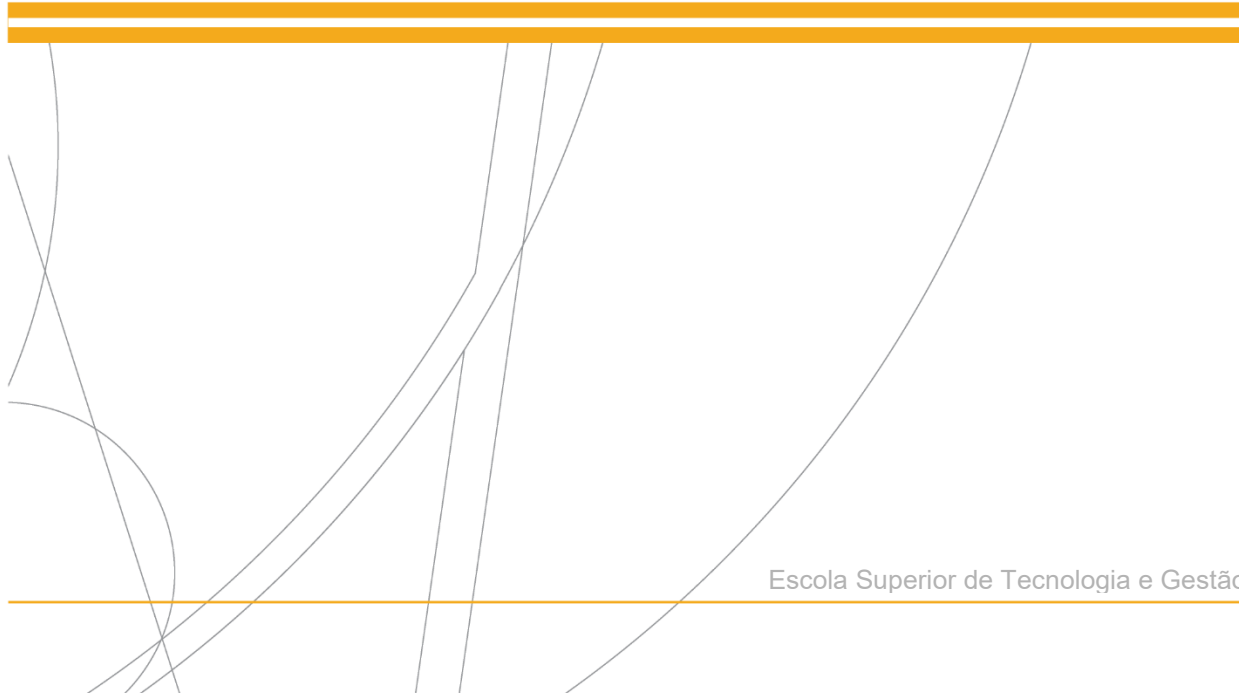
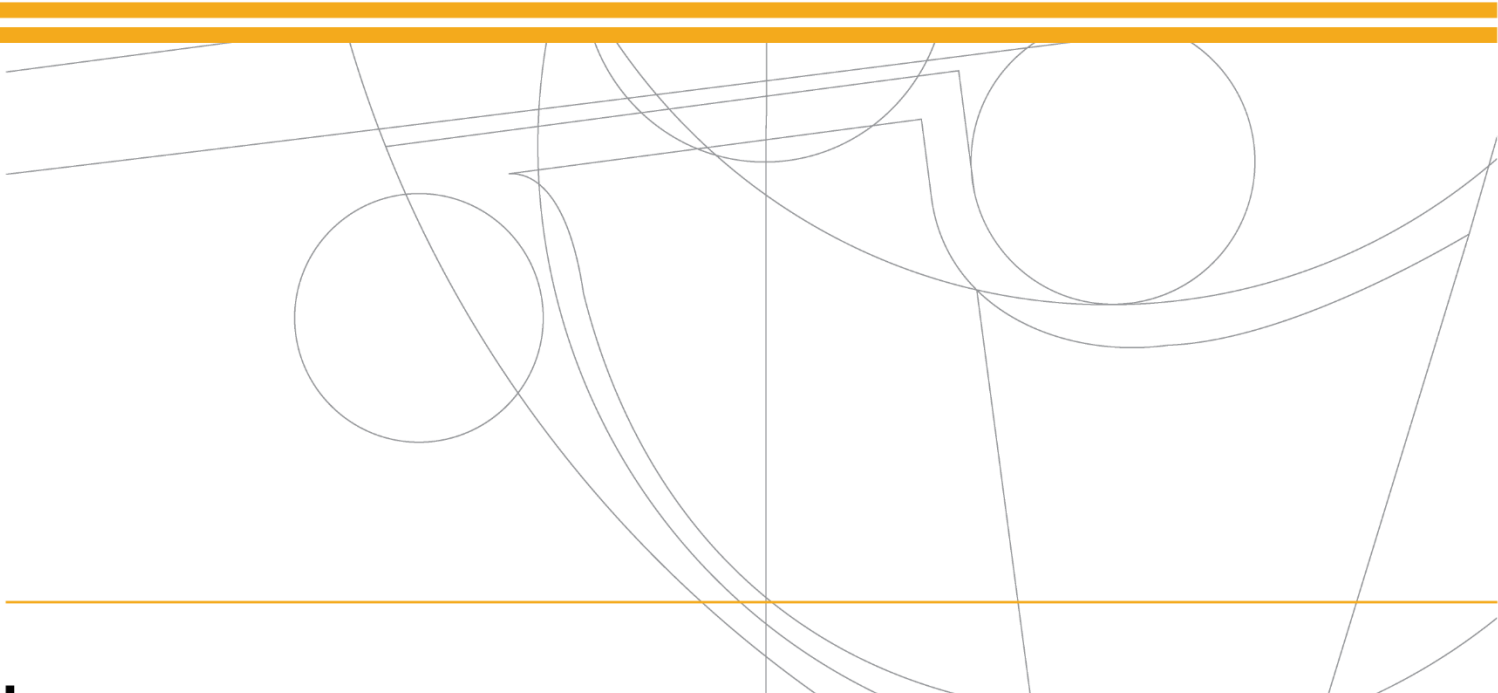




ETESTBOX

Sistema portátil de diagnóstico e programação



Nelson Alberto da Cunha Pereira



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Nelson Alberto da Cunha Pereira

ETESTBOX

Sistema portátil de diagnóstico e programação

Nome do Curso de Mestrado
Eletrónica e Eletrificação Automóvel

Trabalho efetuado sob a orientação de
Professor David Macário (ESTG)
Eng^a Diana Palma (Borgwarner)

Julho de 2026

MEMBROS DO JÚRI

Presidente:

Doutor Sérgio Ivan Fernandes Lopes

Professor Coordenador do Instituto Politécnico de Viana do Castelo

Vogal:

Doutor Ricardo Augusto Rodrigues da Silva Severino

Professor Adjunto do Instituto Politécnico do Porto

(Arguente)

Vogal:

Mestre David Paulo Torres Macário

Professor Adjunto Convidado do Instituto Politécnico de Viana do Castelo

(Orientador)

AGRADECIMENTOS

A realização deste trabalho contou com o contributo e apoio de várias pessoas e entidades, às quais expresso o meu mais profundo agradecimento.

Em primeiro lugar, agradeço ao meu orientador académico, **Professor David Macário**, pela orientação, pela disponibilidade e pelas valiosas sugestões que foram determinantes para a qualidade deste trabalho.

À **BorgWarner**, deixo um agradecimento especial pela oportunidade de desenvolver este projeto em ambiente industrial, pela confiança demonstrada e pela disponibilização de todo o material e recursos necessários. Um reconhecimento particular à **Diana Palma**, que me acompanhou como orientadora industrial, pelo apoio técnico, pela orientação prática e pela forma sempre colaborativa com que guiou o desenvolvimento deste projeto.

À minha esposa, **Diana Silva**, agradeço pelo apoio incondicional, paciência e encorajamento constante, fundamentais para a conclusão desta etapa. À minha mãe, **Eugénia**, e ao meu pai, **Alberto**, agradeço profundamente por todo o apoio, dedicação e exemplo que sempre me transmitiram. Estendo este agradecimento a toda a minha família, pelo suporte contínuo e por acreditarem sempre no meu percurso.

A todos os que, de forma direta ou indireta, contribuíram para a realização deste trabalho, deixo o meu sincero reconhecimento e gratidão.

RESUMO

A crescente complexidade dos sistemas eletrônicos automóveis exige ferramentas de teste flexíveis, portáteis e capazes de garantir a validação rigorosa de módulos antes da sua integração em veículos. Neste contexto, o presente trabalho descreve o desenvolvimento da eTestBox, uma plataforma modular e portátil destinada ao teste funcional de unidades eletrônicas automóveis (UUTs). A solução integra num único dispositivo as capacidades avançadas de comunicação LIN e CAN, aquisição analógica de alta precisão, entradas e saídas digitais isoladas, alimentação programável e monitorização de corrente, permitindo a execução de testes completos de fim de linha e validação funcional.

A arquitetura do sistema assenta num Raspberry Pi como controlador central, complementado por uma DAQ MCC128 para leitura analógica, um módulo PCAN-PRO FD para comunicação automóvel e um conjunto de circuitos dedicados projetados em PCB para garantir isolamento galvânico, redução de ruído e compatibilidade com requisitos industriais. Uma fonte de alimentação interna gerida por bateria, aliada a conversores CC-CC dedicados, fornece todas as tensões necessárias ao sistema, incluindo um barramento de 30 V que alimenta uma fonte programável M5Stack PPS, permitindo alimentar o UUT com tensão e corrente ajustáveis durante o ensaio.

O dispositivo apresenta ainda uma interface UUT baseada num conector DB15 que agrega todas as funcionalidades necessárias: medições analógicas em modo *single-ended* ou diferencial, saídas digitais por relé, entradas digitais optoisoladas, comunicação CAN e LIN, alimentação fixa de 12 V e alimentação programável KL30 com descarga automática. Todo o sistema é controlado por uma interface gráfica desenvolvida em PyQt e suporta execução automática de sequências de teste através de um sequenciador integrado.

Os resultados obtidos demonstram que a eTestBox constitui uma solução robusta, versátil e de baixo custo para ambientes industriais, oferecendo portabilidade, autonomia e capacidade de adaptação a diferentes cenários de teste automóvel. A combinação entre hardware dedicado e software modular permite ainda a expansão futura do sistema para novos protocolos ou

funcionalidades, reforçando a sua utilidade em contextos de mobilidade elétrica e manufatura inteligente.

PALAVRAS CHAVE: eTestBox; Testes de Fim de Linha (EOL); Comunicação Automóvel; CAN; LIN; Aquisição de Dados (DAQ); Raspberry Pi; Sequenciador de Testes; Sistemas Embebidos; Eletrónica Automóvel; Plataforma de Teste Modular.

ABSTRACT

The increasing complexity of automotive electronic systems demands flexible, portable and reliable testing tools capable of ensuring rigorous validation of electronic modules before their integration into vehicles. This work presents the development of the eTestBox, a modular and portable platform designed for functional testing of automotive electronic units (UUTs). The solution integrates, in a single device, advanced capabilities for LIN and CAN communication, high-precision analog acquisition, isolated digital I/Os, programmable power supply control and real-time current monitoring, enabling comprehensive end-of-line testing and functional validation.

The system architecture is built around a Raspberry Pi as the central controller, supported by a MCC128 DAQ for analog measurement, a PCAN-PRO FD module for automotive communication, and dedicated PCB circuitry designed to ensure galvanic isolation, noise reduction and full compatibility with industrial requirements. An internal battery-powered supply system, combined with multiple DC-DC converters, generates all the required voltages, including a 30 V auxiliary bus that feeds a programmable M5Stack PPS power module used to supply the UUT with configurable voltage and current profiles.

The UUT interface is implemented through a DB15 connector that aggregates all functional channels, including differential or single-ended analog inputs, relay-based digital outputs, opto-isolated digital inputs, CAN and LIN communication lines, a fixed 12 V output and a programmable KL30 supply with automatic discharge capability. All system functions are controlled through a PyQt-based graphical interface and an integrated test sequencer enabling full automation of test procedures.

The results demonstrate that the eTestBox provides a robust, versatile and cost-effective solution for industrial environments, offering portability, autonomy and adaptability to different automotive testing scenarios. The combination of dedicated hardware and modular software also enables future expansion to support new protocols and functionalities, reinforcing its relevance in the context of electric mobility and smart manufacturing.

KEYWORDS: eTestBox; End-of-Line Testing (EOL); Automotive Communication; CAN Bus; LIN Bus; Data Acquisition (DAQ); Test Sequencer; Embedded Systems; Raspberry Pi; Automotive electronics; Modular Test Platform.

LISTA DE ABREVIATURAS/SIGLAS/SÍMBOLOS

ACK	<i>“Acknowledgment”</i> Reconhecimento
CAN	<i>“Controller Area Network”</i> Rede de Área do Controlador
CRC	<i>“Cyclic Redundancy Check”</i> Verificação Cíclica de Redundância
DAQ	<i>“Data Acquisition System”</i> Sistema de Aquisição de Dados
ECU	<i>“Electronic Control Unit”</i> Unidade de Controlo Eletrónico
EOL	<i>“End-of-Line”</i> Fim de Linha
EMI	<i>“Electromagnetic interference”</i> Interferência eletromagnética
GPIO	<i>“General-Purpose Input/Output”</i> Entrada/Saída de Uso Geral
GUI	<i>“Graphical User Interface”</i> Interface Gráfica de Utilizador
HMI	<i>“Human-Machine Interface”</i> Interface Homem-Máquina
I²C	<i>“Inter-Integrated Circuit”</i> Circuito Inter-Integrado
LIN	<i>“Local Interconnect Network”</i> Rede de Interconexão Local
MES	<i>“Manufacturing Execution System”</i> Sistema de Execução de Manufatura
MQTT	<i>“Message Queuing Telemetry Transport”</i> Transporte de telemetria de filas de mensagens
OEM	<i>“Original Equipment Manufacturer”</i> Fabricante Original de Equipamento
OPC UA	<i>“Open Platform Communications Unified Architecture”</i> Arquitetura Unificada de Comunicações de Plataforma Aberta

PCB	<i>“Printed Circuit Board”</i>
	Placa de circuito impresso
PLC	<i>“Programmable Logic Controller”</i>
	Controlador Lógico Programável
PPS	<i>“Programmable Power Supply”</i>
	Fonte de Alimentação Programável
RPi	<i>“Raspberry Pi”</i>
	Raspberry Pi
SCADA	<i>“Supervisory Control and Data Acquisition”</i>
	Supervisão, Controlo e Aquisição de Dados
SPI	<i>“Serial Peripheral Interface”</i>
	Interface periférica série
UART	<i>“Universal Asynchronous Receiver-Transmitter”</i>
	Recetor-Transmissor Assíncrono Universal
UUT	<i>“Unit Under Test”</i>
	Unidade Sob Teste

ÍNDICE

AGRADECIMENTOS.....	v
RESUMO	vii
ABSTRACT	ix
LISTA DE ABREVIATURAS/SIGLAS/SÍMBOLOS.....	xi
ÍNDICE	xiii
1. Introdução	1
1.1. Enquadramento	1
1.2. Objetivos	1
1.3. Motivações.....	2
1.4. Estrutura.....	2
2. Estado da Arte e Fundamentação Teórica	5
2.1. Sistemas de Teste Automóvel.....	5
2.1.1. Plataformas Modulares de Teste	6
2.2. Protocolos de Comunicação Automóvel.....	6
2.2.1. Controller Area Network (CAN).....	7
2.2.2. Local Interconnect Network (LIN)	9
2.2.3. Outros Protocolos Relevantes.....	11
2.3. Sistemas de Aquisição de Dados (DAQ)	12
2.3.1. Conceitos Fundamentais.....	12
2.3.2. Arquitetura e Parâmetros de Desempenho.....	13
2.3.3. Importância dos sistemas DAQ em Ensaaios Automóveis.....	14
2.4. Arquiteturas de Software para Sistemas de Teste.....	15
3. Requisitos do sistema	17
3.1. Requisitos funcionais.....	17
3.2. Requisitos não funcionais.....	19
3.3. Requisitos de validação	21
4. Arquitetura da Solução.....	25
4.1. Visão Global do Sistema.....	25
4.2. Arquitetura Funcional	26
4.2.1. Interface de Utilizador	27
4.2.2. Controlador Central	28
4.2.3. Sistema de Aquisição Analógica.....	31
4.2.4. Comunicação LIN/CAN	33
4.2.5. Entradas e Saídas Digitais.....	35

4.2.6.	Sistema de Alimentação.....	38
4.2.7.	Interface com a UUT	43
5.	Implementação Técnica.....	47
5.1.	Implementação de Hardware	47
5.1.1.	Integração do Display e Controlador	47
5.1.2.	Integração da Fonte Programável.....	49
5.1.3.	Integração da DAQ	52
5.1.4.	Desenvolvimento do PCB Multicamada.....	53
5.1.5.	Desenvolvimento do Chassis Integrado.....	57
5.1.6.	Montagem do Protótipo	59
5.2.	Implementação do Software	62
5.2.1.	Arquitetura da Aplicação	62
5.2.2.	Módulo de Controlo da Alimentação.....	66
5.2.3.	Módulos de Comunicação (LIN e CAN)	67
5.2.4.	Módulos de Monitorização	69
5.2.5.	Módulo de Conversão Analógico-Digital.....	71
5.2.6.	Módulo de Entradas e Saídas Digitais	72
5.2.7.	Sequenciador e Motor de Execução	73
5.2.8.	Modo Administrador.....	77
6.	Validação Experimental e Avaliação de Desempenho	79
6.1.	Validação da Comunicação LIN e CAN	79
6.2.	Validação da DAQ	81
6.3.	Validação das Entradas e Saídas Digitais.....	84
6.4.	Validação do Sequenciador	86
6.5.	Ensaios de Autonomia	87
7.	Análise Crítica	89
7.1.	Discussão dos Resultados Obtidos	89
7.2.	Limitações e Propostas de Melhorias Futuras	89
	Conclusões	93
	Referências Bibliográficas.....	95
	Apêndices.....	99
	Apêndice A – Diagrama Elétrico da eTestbox	99
	Apêndice B – Lista De Componentes do PCB.....	105
	Apêndice C – Lista de Componentes do Chassis	106
	Apêndice D – Especificação das Funções do Sequencer Engine	107
	Apêndice E – Layout Do PCB.....	108

ÍNDICE DE FIGURAS

Figura 1 - Máquina teste de Fim de Linha na Borgwarner	5
Figura 2 - Sistema de teste modular Vector VT System	6
Figura 3 - Arquitetura de redes de comunicação automóvel (Leen & Heffernan, 2002).....	7
Figura 4 - Topologia típica de uma rede CAN (Elettronica Open Source, 2018)	8
Figura 5 - Formato de uma frame CAN (Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008).....	8
Figura 6 - Exemplo de um LIN Description File (LDF)	10
Figura 7 - Topologia típica de uma rede LIN (CSS Electronics , 2025)	10
Figura 8 - Formato de uma frame LIN (Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008).....	11
Figura 9 - Diagrama genérico de sistema de aquisição de dados (DAQ).	13
Figura 10 - Arquitetura MVVM.....	16
Figura 11 - Exemplo de sistema de diagnostico portátil (Ultima EDI Engine Diagnostic Instrument)	25
Figura 12 - Diagrama de blocos da eTestBox	26
Figura 13 - Vista frontal e lateral do HMI.....	27
Figura 14 - Visão geral do Raspberry Pi 5 (Donnison, 2023)	29
Figura 15 - DAQ MCC128 (Measurement Computing Corporation, 2023)	31
Figura 16 - Diagrama blocos da DAQ (Measurement Computing Corporation, 2023)	33
Figura 17 - Peak PCAN-Pro FD (PEAK-System, 2025)	34
Figura 18 - Circuito das entradas digitais da eTestBox.....	37
Figura 19 - Circuito das saídas digitais da eTestBox.....	38
Figura 20 - Circuito de gestão de alimentação da eTestBox	40
Figura 21 - Conversor DC-DC principal U1 (JTD2024S05).....	41
Figura 22 - Conversor DC-DC 30V U4 (CCG30-24-15D)	41
Figura 23- Fonte de alimentação programável M5Stack PPS	43
Figura 24 - Circuito de comutação e medição de corrente da alimentação KL30	44
Figura 25 - Gravação da imagem base do Raspberry Pi OS para o cartão SD utilizando o Raspberry Pi Imager	48
Figura 26 - Comando para obter e instalar o software package.....	48
Figura 27 - Arranque do Raspberry Pi através do HMI.....	49
Figura 28 - Configuração do pino de interrupção do painel tátil	50
Figura 29 - Esquema ligação do M5Stack PPS ao Raspberry Pi.....	50
Figura 30 - Teste experimental do módulo M5Stack PPS	51
Figura 31 - Mapa de endereços do protocolo I2C do M5Stack PPS.....	52
Figura 32 - MCC DAQ HAT Manager e Control Panel	53
Figura 33 - Editor do esquemático no Kicad	54
Figura 34 - Editor do layout do PCB	55
Figura 35 - Lista de ficheiros Gerber gerados.....	55
Figura 36 - Vista superior do PCB com identificação dos componentes.....	56
Figura 37 - Vista inferior do PCB	56
Figura 38 - Mala rígida SERPAC SE230.....	57
Figura 39 - Vista exterior e interior do Chassis Integrado da eTestBox	58
Figura 40 - Vista explodida e identificação dos elementos.....	58
Figura 41 - Processo soldadura dos componentes no PCB	59
Figura 42 - Passos da montagem do chassis	60

Figura 43 - Vista interna da eTestBox.....	61
Figura 44 - Vista exterior da eTestBox	61
Figura 45 - Arquitetura do software.....	63
Figura 46 - Interface gráfica da eTestBox.....	64
Figura 47 - Definição de arranque automático da aplicação eTestBox.....	66
Figura 48 - Separador da fonte de alimentação.....	67
Figura 49 - Separador da comunicação LIN.....	68
Figura 50 - Separador da comunicação CAN	69
Figura 51 - Separador de registo	70
Figura 52 - Separador de visualização de gráficos	71
Figura 53 - Separador do conversor analógico-digital	72
Figura 54 - Separador de entradas e saídas	73
Figura 55 - Separador do sequenciador	74
Figura 56 - Diagrama blocos do sequenciador	76
Figura 57 - Passos de acesso ao separador Admin.....	77
Figura 58 - Separador de administração	78
Figura 59 - Validação com Vector CANoe.....	80
Figura 60 - Validação das tensões e correntes com Agilent 34970A	82
Figura 61 - Ensaio de validação das entradas e saídas digitais	85
Figura 62 - Curva descarga bateria sem UUT	87
Figura 63 - Curva descarga bateria com UUT	88

ÍNDICE DE TABELAS

Tabela 1 - Tabela comparação de protocolos	12
Tabela 2 - Requisitos de validação experimental.....	23
Tabela 3 - Atribuição de pinos do PCAN-Pro FD.....	35
Tabela 4 - Especificações técnicas do conversor DC-DC JTD2024S05.....	41
Tabela 5 - Especificações técnicas do conversor DC-DC CCG30-24-15D.....	42
Tabela 6 - Especificações técnicas da fonte M5Stack PPS	43
Tabela 7 - Atribuição de pinos da ligação ao UUT.....	45
Tabela 8 - Exemplo de uma tabela de importação no sequenciador.....	75
Tabela 9 - Erros absoluto e relativo dos sinais KL30	83
Tabela 10 - Erros absoluto e relativo das entradas analógicas de uso geral	84
Tabela 11 - Limiar de deteção das entradas digitais.....	84
Tabela 12 - Estado lógico e corrente de entrada em função da tensão aplicada	85

1. Introdução

1.1. Enquadramento

A crescente eletrificação da indústria automóvel, aliada ao aumento da complexidade dos sistemas eletrónicos embebidos, tem vindo a transformar profundamente os processos de desenvolvimento, validação e produção de componentes automóveis. Os veículos modernos integram múltiplas unidades de controlo eletrónico (ECUs), responsáveis por funções críticas como gestão de energia, controlo de atuadores e comunicação em rede, recorrendo a protocolos como CAN e LIN.

Neste contexto, os processos de teste assumem um papel fundamental, em particular os ensaios de fim de linha (End-of-Line, EOL), onde cada unidade produzida deve ser validada funcionalmente antes da sua expedição. Estes testes devem garantir fiabilidade, repetibilidade e rastreabilidade, sendo simultaneamente rápidos e adaptáveis à elevada variabilidade de produtos.

O presente trabalho foi desenvolvido em colaboração com a **BorgWarner**, uma empresa multinacional de referência no setor automóvel, especializada no desenvolvimento de tecnologias para mobilidade eficiente e eletrificada. No contexto da BorgWarner, a validação funcional de módulos eletrónicos é realizada através de sistemas de teste dedicados, frequentemente baseados em plataformas modulares de elevado custo e complexidade.

1.2. Objetivos

O objetivo principal deste trabalho consiste no desenvolvimento de uma plataforma portátil de teste funcional para módulos eletrónicos automóveis, designada **eTestBox**, capaz de suportar ensaios de validação em ambiente industrial.

De forma mais detalhada, os objetivos específicos deste trabalho passam pelo desenvolvimento de uma arquitetura modular capaz de integrar múltiplas funcionalidades essenciais ao contexto de teste automóvel, nomeadamente comunicação através dos protocolos CAN e LIN, aquisição de sinais analógicos, controlo de entradas e saídas digitais e alimentação programável de dispositivos sob teste (UUTs). Paralelamente, pretende-se implementar um sistema de controlo baseado num controlador embebido que permita a execução

automatizada de sequências de teste, a monitorização em tempo real de parâmetros elétricos e a interação com o utilizador através de uma interface gráfica intuitiva. Adicionalmente, o sistema deverá apresentar características alinhadas com os requisitos industriais, incluindo portabilidade, robustez elétrica, flexibilidade de configuração e capacidade de integração com sistemas externos. Por fim, é também objetivo validar experimentalmente o desempenho da solução desenvolvida, demonstrando a sua aplicabilidade em cenários reais de teste automóvel.

1.3. Motivações

A motivação central deste trabalho resulta da identificação de limitações nas soluções de teste atualmente disponíveis. Embora plataformas comerciais como sistemas PXI ou Vector VT ofereçam elevada precisão e funcionalidade, apresentam desvantagens significativas quando aplicadas a determinados contextos industriais, nomeadamente:

- Elevado custo de aquisição e manutenção
- Dependência de infraestruturas dedicadas
- Baixa portabilidade
- Complexidade de configuração e adaptação

Estas limitações tornam-se particularmente relevantes em ambientes onde é necessário realizar testes rápidos, distribuídos ou fora de bancada fixa.

Assim, emerge a necessidade de desenvolver uma solução intermédia: uma plataforma de teste que combine funcionalidade e robustez técnica com simplicidade, portabilidade e custo reduzido, mantendo simultaneamente capacidade de integração com sistemas industriais e suporte a protocolos automóveis.

1.4. Estrutura

O presente relatório encontra-se organizado em **seis capítulos** principais, complementados por uma secção final de conclusões, sendo estruturado de forma progressiva desde o enquadramento conceptual até à validação experimental da solução desenvolvida. O **Capítulo 1**, correspondente à introdução, apresenta o enquadramento do trabalho, os objetivos definidos, as

motivações subjacentes ao desenvolvimento da solução proposta e a organização global do documento.

No **Capítulo 2** é apresentado o estado da arte e a fundamentação teórica, abordando os conceitos e tecnologias relevantes para o desenvolvimento da eTestBox, nomeadamente os protocolos de comunicação automóvel, os sistemas de aquisição de dados, os sistemas de teste automóvel e as arquiteturas de software aplicadas a sistemas de teste. Este capítulo estabelece a base técnica necessária para compreender as decisões de engenharia adotadas ao longo do trabalho.

O **Capítulo 3** apresenta os requisitos do sistema que serviram de base ao desenvolvimento da solução. São definidos os requisitos funcionais e não funcionais da plataforma, bem como o plano de validação experimental utilizado para verificar o cumprimento dos requisitos estabelecidos e avaliar o desempenho da solução desenvolvida.

No **Capítulo 4** é descrita a implementação técnica da solução, abrangendo tanto a componente de hardware como de software. São apresentados o processo de integração dos diferentes módulos, o desenvolvimento do PCB e do chassis, a montagem do protótipo e a arquitetura da aplicação, bem como os diversos módulos funcionais e o sequenciador responsável pela execução dos testes automatizados.

O **Capítulo 5** apresenta a validação experimental e a avaliação de desempenho do sistema desenvolvido, incluindo os ensaios realizados para validar a comunicação LIN e CAN, o desempenho do sistema de aquisição de dados, o funcionamento das entradas e saídas digitais, o comportamento do sequenciador e a autonomia do sistema em condições reais de operação.

Por fim, o **Capítulo 6** corresponde à análise crítica dos resultados obtidos, onde são discutidas as principais limitações identificadas e apresentadas propostas de melhoria e evolução futura. O documento termina com a secção de conclusões, onde são sintetizados os principais contributos do trabalho e avaliado o grau de cumprimento dos objetivos definidos.

2. Estado da Arte e Fundamentação Teórica

2.1. Sistemas de Teste Automóvel

Os ensaios de EOL constituem a última etapa do processo produtivo de módulos eletrónicos automóveis, tendo como objetivo validar funcionalmente cada unidade antes da sua expedição. Ao contrário dos ensaios de desenvolvimento, que se focam na caracterização detalhada do desempenho, os testes EOL privilegiam rapidez, repetibilidade e rastreabilidade.

Num contexto industrial, os sistemas EOL devem garantir:

- Execução automatizada de sequências de teste;
- Verificação de limites elétricos e funcionais;
- Comunicação com a unidade sob teste (UUT) através de protocolos normalizados;
- Registo estruturado de resultados para integração com sistemas MES.

Com a crescente eletrificação automóvel, os ensaios EOL passaram a incluir não apenas medições elétricas convencionais, mas também validação de comunicação em rede (CAN, LIN, Ethernet), verificação de software embarcado e testes de conformidade energética. Esta evolução aumentou significativamente a complexidade das bancadas de teste, exigindo maior flexibilidade arquitetural e capacidade de reconfiguração (Praprotnik, Gartner, Zauner, & Horauer, 2009).



Figura 1 - Máquina teste de Fim de Linha na Borgwarner

2.1.1. Plataformas Modulares de Teste

No domínio das plataformas comerciais de teste automóvel destacam-se arquiteturas modulares como **NI PXI** (National Instruments), **Vector VT System** ou **dSPACE** e **ETAS**.

Estas soluções modulares de teste são amplamente utilizadas na indústria automóvel, como sistemas baseados em arquiteturas PXI ou bancadas dedicadas de validação e *Hardware-in-the-Loop*, caracterizam-se por elevada precisão metrológica, escalabilidade através da adição de módulos especializados e integração com ambientes avançados de automação e análise. Estas soluções suportam múltiplos protocolos de comunicação e oferecem elevada capacidade de sincronização e processamento em tempo real, sendo particularmente adequadas a ambientes laboratoriais e fases de desenvolvimento onde a flexibilidade funcional e o rigor técnico são prioritários. No entanto, a sua adoção implica frequentemente investimento inicial significativo, dependência de ecossistemas proprietários e requisitos de infraestrutura física dedicada, fatores que podem limitar a sua aplicabilidade em cenários onde se privilegia portabilidade, simplicidade operacional ou implementação descentralizada.



Figura 2 - Sistema de teste modular Vector VT System

2.2. Protocolos de Comunicação Automóvel

Esta complexidade dos sistemas eletrónicos nos veículos modernos tem conduzido à necessidade de arquiteturas de comunicação cada vez mais robustas e eficientes. Em vez de ligações ponto-a-ponto, os fabricantes de automóveis adotaram redes de comunicação digitais, capazes de interligar

unidades de controlo eletrónico (*Electronic Control Units, ECUs*), sensores e atuadores.

Os principais protocolos de comunicação utilizados na indústria automóvel diferem em termos de taxa de transmissão, robustez, custo e finalidade de aplicação. Entre estes destacam-se o **LIN** (*Local Interconnect Network*) e o **CAN** (*Controller Area Network*, bem como soluções de maior desempenho como FlexRay e Ethernet Automóvel, utilizadas em cenários mais exigentes. (Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008)

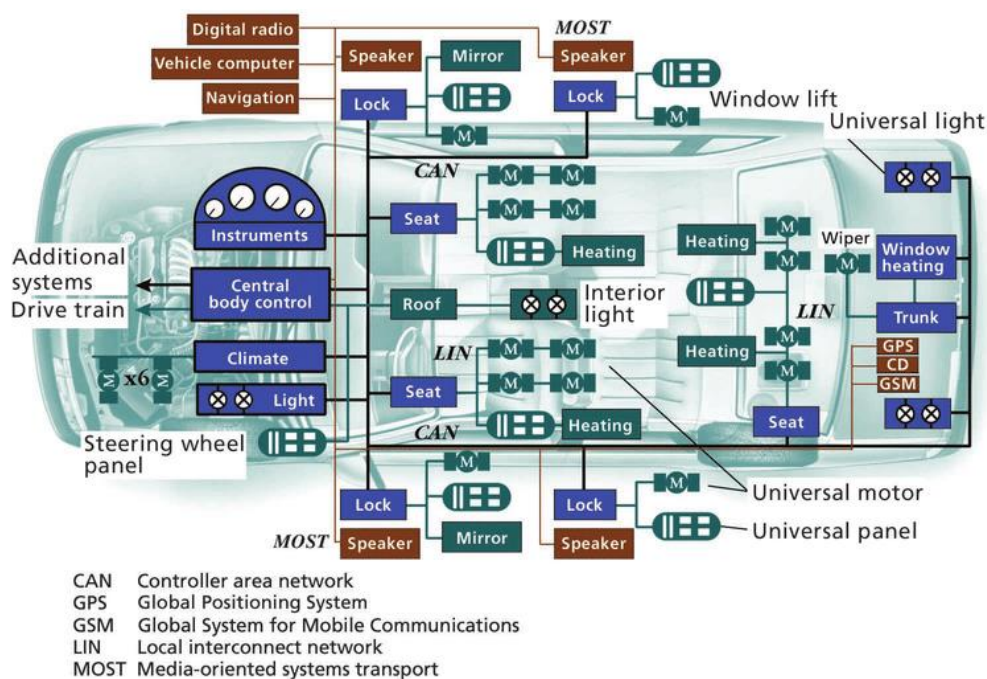


Figura 3 - Arquitetura de redes de comunicação automóvel (Leen & Heffernan, 2002)

2.2.1. Controller Area Network (CAN)

O **CAN** é o protocolo de comunicação automóvel mais amplamente utilizado, desenvolvido pela Bosch na década de 1980 com o objetivo de reduzir significativamente a complexidade e o comprimento da cablagem nos veículos, substituindo ligações ponto-a-ponto por uma rede multiplexada. Tornou-se norma ISO em 1994 (ISO 11898) e consolidou-se como padrão de facto na indústria automóvel devido ao seu baixo custo, robustez e capacidade de garantir atrasos de comunicação limitados.

Fisicamente, o CAN utiliza um par entrançado de cobre e suporta diferentes velocidades conforme o domínio de aplicação: tipicamente 250 ou 500 kbit/s para controlo em tempo real (*powertrain* e *chassis* – SAE classe C) e 125 kbit/s para eletrónica de carroçaria (SAE classe B). A taxa máxima pode atingir 1 Mbit/s em barramentos curtos.

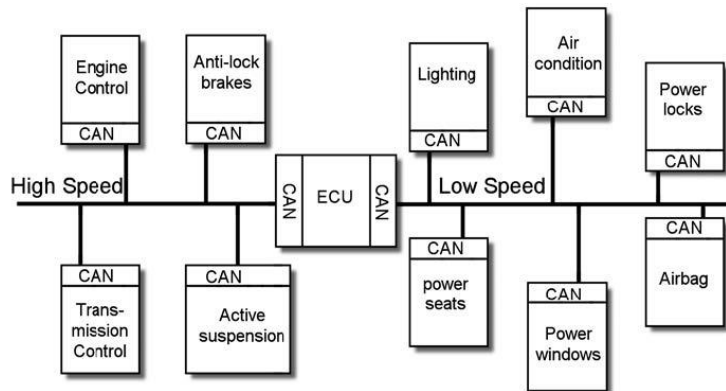


Figura 4 - Topologia típica de uma rede CAN (Elettronica Open Source, 2018)

A comunicação baseia-se em mensagens identificadas por um identificador (ID) transmitido na própria *frame*, cujo valor determina a prioridade de arbitragem no acesso ao barramento. O mecanismo de arbitragem é não destrutivo e permite que múltiplos nós transmitam simultaneamente, prevalecendo automaticamente a mensagem de maior prioridade. A codificação utiliza representação NRZ com *bit stuffing* (inserção de bit após cinco bits consecutivos iguais), garantindo sincronização adequada.

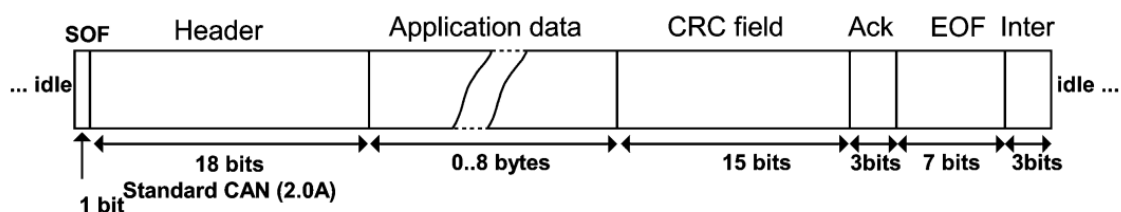


Figura 5 - Formato de uma frame CAN (Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008)

A *frame* CAN padrão (CAN 2.0A) pode transportar até 8 bytes de dados. O protocolo inclui vários mecanismos de deteção de erro, nomeadamente verificação de CRC, monitorização da estrutura da *frame*, deteção de erros de *bit stuffing* e emissão de error *flags* por qualquer nó que identifique uma falha. Adicionalmente, possui mecanismos de confinamento de falhas baseados em contadores de erro, que permitem isolar nós com comportamento anómalo

(Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008).

A interpretação semântica das mensagens transmitidas no barramento é frequentemente definida através de um ficheiro **DBC (Database CAN)** que constitui uma base de dados estruturada onde são descritas as mensagens CAN e os respetivos sinais, incluindo identificadores, posição dos *bits*, fatores de escala, *offsets*, unidades físicas e limites de validade. Enquanto o protocolo CAN especifica como os dados são transmitidos, o DBC define o que esses dados representam (CSS Electronics, 2025).

Desta forma, o ficheiro DBC funciona como um modelo formal da rede, permitindo que ferramentas de diagnóstico, monitorização e teste descodifiquem automaticamente as *frames* CAN e apresentem os valores físicos dos sinais de forma inteligível. A sua utilização é particularmente relevante em ambientes de validação e EOL, onde é necessário interpretar grandes volumes de dados de forma estruturada e consistente.

2.2.2. Local Interconnect Network (LIN)

O **LIN** é um protocolo de comunicação série de baixo custo, classificado como rede SAE classe A, destinado a aplicações automóveis que não exigem elevada largura de banda, ao contrário do CAN. Foi desenvolvido por um consórcio de fabricantes automóveis (como DaimlerChrysler, Volkswagen, BMW e Volvo) e é amplamente utilizado em veículos.

A especificação LIN (versão 2.1) define não apenas o protocolo de transmissão (camadas física e de ligação de dados), mas também um protocolo de diagnóstico e uma linguagem de descrição da rede (LDF – *LIN Description File*), que permite especificar capacidades dos nós, tabelas de agendamento (*schedule tables*) e configuração global do *cluster*. Esta abordagem facilita a geração automática de configuração por ferramentas de *software*.

```

HELLA:IBS-200X-P-24V.ldf
10
11 LIN_description_file;
12 LIN_protocol_version = "2.0";
13 LIN_language_version = "2.0";
14 LIN_speed = 19.2 kbps;
15
16 Nodes {
17     Master: SG, 10 ms, 0 ms ;
18     Slaves: IBS ;
19 }
20
21 Signals {
22     Batt_CNominal: 8, 255, SG, IBS ;
23     Batt_U0Min: 8, 255, SG, IBS ;
24     Batt_U0Max: 8, 255, SG, IBS ;
25     Batt_CCANominal: 8, 255, SG, IBS ;
26     Temperature_Upred: 8, 255, SG, IBS ;
27     Batt_Technology: 3, 7, SG, IBS ;
28     Batt_ChangeNotification: 1, 0, SG, IBS ;
29     BatteryCurrent: 16, 65535, IBS, SG ;
30     BatteryVoltage: 16, 65535, IBS, SG ;

```

Figura 6 - Exemplo de um LIN Description File (LDF)

A arquitetura LIN baseia-se num modelo **mestre–escravo**, onde um único nó mestre controla a comunicação e vários nós escravos respondem aos pedidos. A camada física utiliza um único fio, com débito máximo de 20 kbit/s, limitado por restrições de EMI.

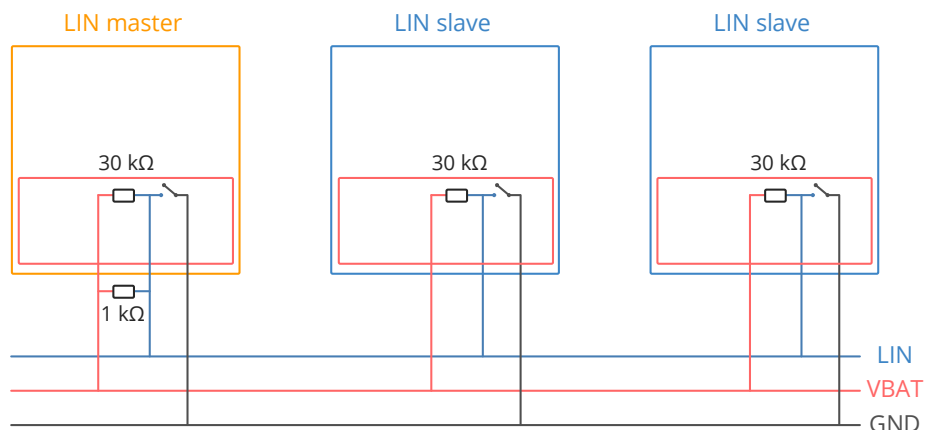


Figura 7 - Topologia típica de uma rede LIN (CSS Electronics , 2025)

A comunicação é determinística e controlada através de uma *schedule table*, que define a ordem e o instante de transmissão das *frames*.

Cada transmissão ocorre dentro de um “*frame slot*”. O mestre inicia a comunicação enviando um cabeçalho (*header*), que inclui:

- *Break* (sinalização de início de *frame*),
- *Sync* (sincronização temporal),
- *Protected Identifier* (6 bits de ID + 2 bits de paridade).

O escravo correspondente responde com os dados (até 8 bytes) seguidos de *checksum*, permitindo deteção de erros. Existem 64 identificadores possíveis.

A detecção de erros é assegurada por bits de paridade e *checksum*. O protocolo inclui ainda mecanismos para **modo sleep e wake-up**, importantes para otimização energética em veículos. (Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008)

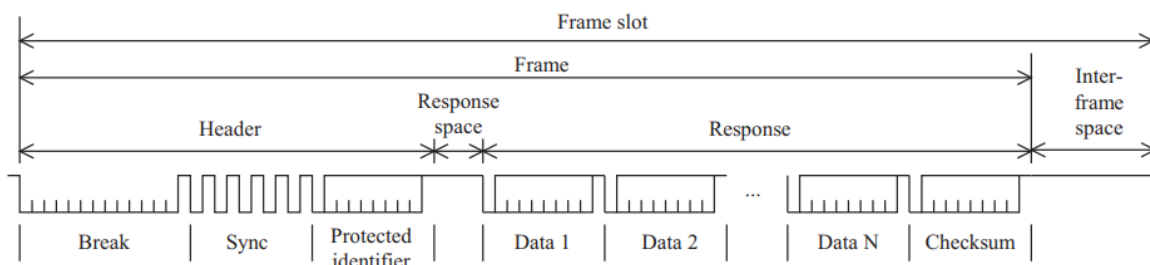


Figura 8 - Formato de uma frame LIN (Navet, Song, Simonot-Lion, & Wilwert, A Review of Embedded Automotive Protocols, 2008)

2.2.3. Outros Protocolos Relevantes

Para além dos protocolos CAN e LIN, amplamente utilizados na comunicação entre unidades de controlo eletrónico, os veículos modernos recorrem também a outras tecnologias de comunicação especializadas.

- **FlexRay** – protocolo determinístico, com velocidades até 10 Mbps, utilizado em sistemas críticos de segurança, como *steer-by-wire* e *brake-by-wire* (ISO 17458:2013).
- **MOST (Media Oriented Systems Transport)** – destinado a aplicações de *infotainment* e multimédia, com elevada largura de banda.
- **Ethernet Automóvel (100BASE-T1, 1000BASE-T1)** – adotada em sistemas avançados de assistência ao condutor (*Advanced Driver Assistance Systems*, ADAS) e condução autónoma, devido à necessidade de transmitir vídeo e grandes volumes de dados em tempo real.

A coexistência destes diferentes protocolos reflete a crescente complexidade das arquiteturas eletrónicas dos veículos modernos e constitui um desafio adicional para os sistemas de teste automóvel. Embora os protocolos FlexRay, MOST e Ethernet sejam relevantes em aplicações específicas, os protocolos CAN e LIN continuam a representar a maior parte das necessidades de comunicação encontradas em ensaios de validação e testes de EOL, especialmente em módulos eletrónicos de produção em larga escala. Esta predominância resulta da sua ampla adoção na indústria, da maturidade

tecnológica e do equilíbrio entre desempenho, custo e complexidade de implementação.

Protocolo	Velocidade Máxima.	Topologia	Aplicações típicas	Custo	Determinismo
LIN	20 kbps	Mestre–escravo	Vidros, bancos, iluminação	Muito baixo	Alto
CAN	1 Mbps (Clássico) / 8 Mbps (FD)	Barramento	Motor, ABS, transmissão	Médio	Médio
FlexRay	10 Mbps	Dual channel	Segurança crítica	Alto	Muito alto
Ethernet	≥100 Mbps	Ponto-a-ponto / estrela	ADAS, infotainment, condução autónoma	Médio-alto	Baixo

Tabela 1 - Tabela comparação de protocolos

2.3. Sistemas de Aquisição de Dados (DAQ)

Os **sistemas de aquisição de dados (DAQ, *Data Acquisition Systems*)** desempenham um papel central em contextos industriais e laboratoriais, permitindo medir, converter, armazenar e processar sinais provenientes de sensores ou dispositivos eletrónicos. A sua utilização em ambientes de teste e validação automóvel é essencial para a caracterização do comportamento elétrico e funcional de módulos eletrónicos, assegurando a conformidade com requisitos técnicos e normativos.

2.3.1. Conceitos Fundamentais

Um sistema DAQ tem como objetivo fundamental converter **grandezas físicas** (como tensão, corrente, temperatura, pressão, aceleração) em **sinais digitais processáveis por computador**.

As funções principais incluem:

- **Medição** – captação de sinais provenientes de sensores ou circuitos eletrónicos.
- **Conversão** – transformação de sinais analógicos em digitais por meio de **Conversores Analógico–Digitais (ADCs)**.

- **Processamento** – filtragem, armazenamento e análise dos dados adquiridos.
- **Atuação** – em alguns casos, também é possível enviar sinais de saída através de **Conversores Digital–Analógico (DACs)**, permitindo gerar estímulos de teste.

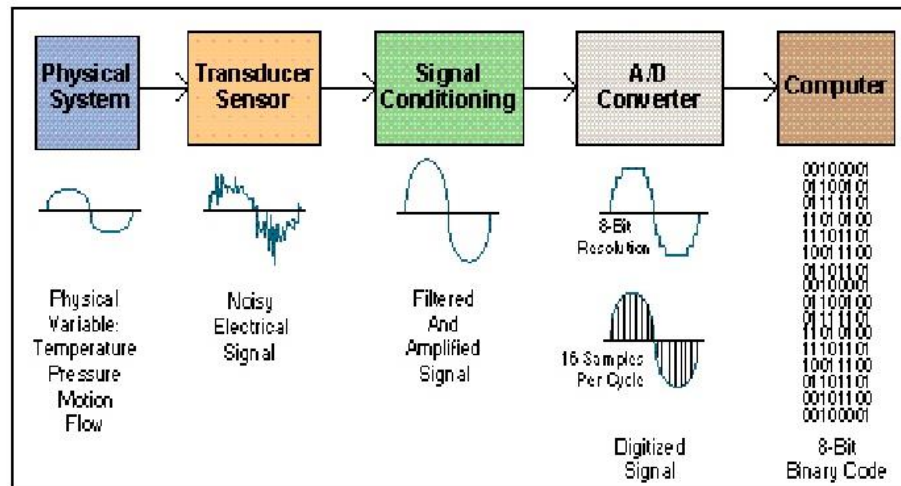


Figura 9 - Diagrama genérico de sistema de aquisição de dados (DAQ).

Adaptado a partir de (Ajithkumar, Ajith, Sabu, Kurian, & Beena, 2024)

2.3.2. Arquitetura e Parâmetros de Desempenho

Um sistema DAQ é composto, tipicamente, pelos seguintes blocos:

1. **Sensores/Transdutores** – convertem grandezas físicas em sinais elétricos (ex.: termopares, *shunts* de corrente).
2. **Circuitos de Condicionamento de Sinal** – incluem amplificação, filtragem e isolamento, de modo a adaptar o sinal às características do ADC.
3. **Conversores Analógico–Digitais (ADC)** – definem a **resolução** (bits) e a **taxa de amostragem** do sistema.
4. **Interface de Comunicação** – garante a ligação ao sistema de processamento central (USB, SPI, Ethernet, PCIe).
5. **Software de Aquisição e Análise** – fornece meios de visualização e processamento dos dados.

Estes blocos são organizados em arquiteturas compactas ou distribuídas, dependendo do nível de desempenho requerido. (Digilent, 2025)

Ao selecionar um sistema DAQ para integração em ferramentas de teste como a **eTestBox**, devem ser considerados parâmetros fundamentais:

- **Resolução (bits)** – número de níveis de quantização. Resoluções mais elevadas (12–24 bits) permitem medições mais precisas.
- **Taxa de Amostragem (*Samples/s*)** – define a rapidez com que os sinais são captados. Deve cumprir o teorema de Nyquist ($\geq 2 \times$ frequência máxima do sinal).
- **Número de Canais** – determina a capacidade de monitorizar múltiplos sinais em simultâneo.
- **Configuração de Entradas** – *single-ended* (referenciadas a um comum) ou diferencial (medição entre dois pontos, maior imunidade a ruído).
- **Faixa Dinâmica** – intervalo de tensões suportadas pelo sistema (ex.: ± 10 V).
- **Ruído e Precisão** – parâmetros como **INL (*Integral Non-Linearity*)**, **offset** e **drift térmico** influenciam a qualidade da aquisição.
- **Capacidade de *Trigger* e *Buffering*** – fundamentais para sincronizar medições com eventos específicos.

2.3.3. Importância dos sistemas DAQ em Ensaios Automóveis

Na indústria automóvel, os sistemas DAQ desempenham um papel fundamental na realização de ensaios funcionais e de validação em componentes eletrónicos e eletromecânicos. Estes sistemas são amplamente utilizados na monitorização de tensões e correntes em ECUs durante testes funcionais, bem como na validação de protocolos de carga e gestão de energia, particularmente relevantes no contexto da mobilidade elétrica. Adicionalmente, são aplicados em ensaios de resposta dinâmica de sensores, como acelerómetros e sensores de pressão, onde a precisão temporal e a integridade do sinal são fatores críticos.

Os sistemas DAQ são igualmente integrados em bancadas de EOL, permitindo verificar automaticamente a conformidade funcional de cada unidade produzida

antes da sua expedição. Nestes contextos industriais, os requisitos associados aos sistemas de teste são particularmente exigentes e incluem, em primeiro lugar, a portabilidade, de forma a possibilitar a sua utilização tanto em ambiente de fábrica como em testes de campo ou em diferentes linhas de produção. A robustez é igualmente essencial, assegurando imunidade a ruído eletromagnético (EMI) e resistência a variações ambientais típicas do ambiente industrial.

Outro requisito fundamental é a capacidade de integração com plataformas de controlo e automação, como PLCs, sistemas embebidos ou controladores baseados em Raspberry Pi, permitindo a incorporação dos sistemas de teste em arquiteturas industriais mais amplas. Por fim, o equilíbrio entre custo e desempenho assume particular relevância, sendo necessário garantir níveis de precisão e fiabilidade adequados sem comprometer a viabilidade económica da solução, especialmente em ambientes de produção em série.

2.4. Arquiteturas de Software para Sistemas de Teste

O desenvolvimento de sistemas de teste modernos exige arquiteturas de software que garantam modularidade, escalabilidade e facilidade de manutenção. A separação clara entre camadas - nomeadamente aquisição de dados, lógica de controlo, interface gráfica e comunicação externa - reduz o acoplamento entre componentes e facilita a evolução incremental do sistema.

Arquiteturas monolíticas tendem a dificultar a integração de novos módulos ou protocolos, aumentando o risco de regressões funcionais. Em contraste, abordagens baseadas em camadas e serviços promovem maior robustez e reutilização de código.

Entre os padrões arquiteturais amplamente utilizados em aplicações com interface gráfica destacam-se o *Model–View–Controller* (MVC) e o *Model–View–ViewModel* (MVVM). Ambos promovem a separação entre dados, lógica e apresentação, embora com diferenças na forma como a interação é estruturada.

O padrão MVVM, em particular, tem sido adotado em aplicações modernas devido à sua capacidade de desacoplar a lógica de negócio da interface, facilitando testes unitários e extensibilidade. Em sistemas de teste industrial, onde novos módulos de hardware e funcionalidades podem ser adicionados ao

longo do tempo, esta separação torna-se especialmente relevante (Fowler, 2004).

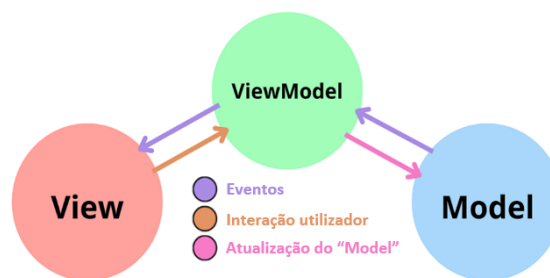


Figura 10 - Arquitetura MVVM

3. Requisitos do sistema

3.1. Requisitos funcionais

Com base nas necessidades identificadas no contexto dos testes de módulos eletrónicos automóveis e nas limitações observadas nas soluções existentes, foram definidos os requisitos funcionais do sistema a desenvolver. Estes requisitos descrevem as funcionalidades que a solução deverá disponibilizar para assegurar a execução de testes de validação de forma fiável, flexível e compatível com os processos industriais.

Os requisitos funcionais apresentados a seguir constituem a base para o desenvolvimento da arquitetura do sistema e para a posterior validação da solução implementada.

- **LIN:** comunicação bidirecional através de dois canais LIN independentes, operando a velocidades até **20 kb/s**. Cada canal deverá permitir a comunicação entre um **nó mestre** e até **15 nós escravos**, de acordo com a arquitetura definida pela especificação LIN, possibilitando a integração e validação de múltiplos módulos eletrónicos numa mesma rede;
- **CAN:** comunicação bidirecional através de dois canais independentes, operando a velocidades até **1 Mb/s**. Cada canal deverá permitir a comunicação com, pelo menos, **30 nós CAN**, de acordo com os requisitos mínimos estabelecidos para redes compatíveis com a norma ISO 11898;
- **Fonte de alimentação do UUT:** o sistema deverá disponibilizar uma fonte de alimentação programável com ajuste independente de tensão e corrente, permitindo configurar valores de saída compreendidos entre **0V e 24V** e entre **0A e 1A**. A fonte deverá possibilitar a alimentação controlada das unidades em teste, bem como a limitação de corrente para proteção dos dispositivos e execução de ensaios em diferentes condições de funcionamento;
- **DAQ:** módulo de aquisição de dados com **4 canais** de entrada analógica independentes, compatíveis com sinais de tensão na gama de **0-10V**, destinados à ligação de sensores e dispositivos externos. O módulo deverá suportar uma taxa de amostragem até **100 kS/s** e uma resolução

mínima de **16 bits**, garantindo a aquisição precisa e fiável de grandezas analógicas durante a execução dos testes;

- **Entradas digitais:** incluir **4 entradas** digitais isoladas galvanicamente, capazes de detetar sinais com tensões de entrada **até 28VDC**. O isolamento galvânico deverá garantir a proteção dos circuitos internos contra diferenças de potencial, ruído elétrico e falhas provenientes dos dispositivos externos ligados ao sistema;
- **Saídas digitais:** **4 saídas** digitais baseadas em relés com **contacto seco**, permitindo a comutação de circuitos externos sem imposição de tensão própria. As saídas deverão possibilitar a simulação de contactos elétricos e o acionamento de dispositivos externos utilizados durante a execução dos testes;
- **Execução automatizada** de sequências de teste configuráveis, possibilitando a definição, edição e reutilização de procedimentos de teste adaptados às características de diferentes módulos eletrónicos.
- **Interface gráfica** intuitiva e adequada ao ambiente industrial, permitindo a configuração dos ensaios, a monitorização do estado da unidade sob teste e a visualização dos resultados de forma clara e acessível para operadores e técnicos.
- **Arquitetura modular e escalável**, permitindo a integração futura de novos protocolos de comunicação, módulos de hardware e funcionalidades adicionais. Para este efeito, a plataforma deverá incluir interfaces de expansão, nomeadamente portas Ethernet e USB, assegurando a sua adaptabilidade a futuras necessidades industriais e tecnológicas.

Adicionalmente, a solução deverá garantir robustez operacional, escalabilidade e compatibilidade com padrões de comunicação industrial, nomeadamente MQTT e OPC UA, possibilitando a sua integração em plataformas SCADA e MES e assegurando rastreabilidade estruturada dos resultados de teste.

Ao conjugar estes requisitos, a solução proposta pretende posicionar-se como uma plataforma intermédia entre instrumentos laboratoriais de elevada

complexidade e soluções simplificadas de teste dedicado. A sua arquitetura deverá ser modular e extensível, permitindo acomodar a evolução contínua dos requisitos técnicos associados à eletrificação automóvel, contribuindo para a implementação de metodologias de ensaio mais ágeis, distribuídas e adaptáveis às necessidades emergentes do setor.

3.2. Requisitos não funcionais

Para além das funcionalidades disponibilizadas, o sistema deverá cumprir um conjunto de requisitos não funcionais relacionados com desempenho, fiabilidade, flexibilidade e adequação ao contexto industrial. Estes requisitos são fundamentais para garantir que a solução possa ser utilizada de forma eficaz em ambientes de produção, validação e suporte técnico, mantendo níveis adequados de robustez, repetibilidade e facilidade de utilização.

Portabilidade: A solução deverá apresentar dimensões e peso reduzidos que permitam o seu transporte e utilização em diferentes ambientes de trabalho. Esta característica assume particular relevância em atividades de suporte à produção, validação de protótipos e campanhas de triagem no cliente, onde nem sempre é possível recorrer a bancadas de teste fixas ou sistemas laboratoriais de grande dimensão. Adicionalmente, o sistema deverá ser concebido para operar de forma autónoma através de alimentação por bateria, garantindo idealmente uma autonomia mínima de 8 horas, correspondente a um turno completo de trabalho, permitindo a sua utilização contínua sem dependência de fontes de alimentação externas.

Robustez Elétrica: O sistema deverá ser concebido para operar em ambientes industriais sujeitos a interferências eletromagnéticas, ruído elétrico e variações das condições de alimentação. A arquitetura deverá incluir mecanismos adequados de proteção e isolamento, minimizando o impacto de perturbações externas na fiabilidade das medições e na integridade dos equipamentos ligados ao sistema.

Segurança: A solução deverá incorporar mecanismos de proteção destinados a salvaguardar tanto os equipamentos sob teste como os próprios utilizadores. Estes mecanismos incluem a limitação de corrente da fonte programável,

proteção contra ligações incorretas, monitorização de condições anómalas de funcionamento e estratégias de recuperação segura em caso de falha.

Modularidade: A arquitetura deverá seguir uma abordagem modular, permitindo que diferentes subsistemas possam ser desenvolvidos, substituídos ou atualizados de forma independente. Esta característica facilita a manutenção, reduz o impacto de futuras alterações e aumenta a flexibilidade da solução para diferentes cenários de aplicação.

Facilidade de Configuração: O sistema deverá disponibilizar mecanismos simples e intuitivos para a configuração de testes, parâmetros de comunicação e interfaces de aquisição. A redução da complexidade de configuração contribui para diminuir o tempo necessário para colocar novos ensaios em funcionamento e reduz a probabilidade de erros operacionais.

Repetibilidade: Os resultados obtidos deverão apresentar um elevado grau de consistência entre execuções sucessivas realizadas nas mesmas condições. Esta característica é particularmente importante em aplicações de validação e ensaios de fim de linha, onde a repetibilidade dos resultados influencia diretamente a confiança no processo de teste.

Rastreabilidade: O sistema deverá permitir o registo e armazenamento da informação relevante associada aos ensaios realizados, incluindo parâmetros de configuração, resultados obtidos, identificação do operador, data e hora de execução. Esta capacidade é essencial para processos de qualidade, análise de falhas e auditoria industrial.

Disponibilidade e Custo dos Componentes: De forma a garantir a viabilidade de replicação da solução nas diferentes unidades do grupo empresarial, a arquitetura deverá privilegiar a utilização de componentes comerciais (off-the-shelf) amplamente disponíveis no mercado. A seleção dos componentes deverá ter em consideração não apenas o desempenho técnico, mas também a facilidade de aquisição, os prazos de entrega e a disponibilidade global dos fornecedores, permitindo a rápida construção ou substituição de sistemas equivalentes quando necessário.

Manutenibilidade e Substituição de Componentes: Os módulos de hardware deverão privilegiar interfaces normalizadas, protocolos de comunicação

amplamente suportados e conectores de utilização corrente na indústria. Sempre que possível, deverão ser utilizados componentes passíveis de substituição por alternativas equivalentes sem necessidade de alterações significativas na arquitetura do sistema. Esta abordagem reduz dependências de fornecedores específicos, facilita a manutenção e aumenta a longevidade da solução.

3.3. Requisitos de validação

A validação experimental tem como objetivo verificar o cumprimento dos requisitos funcionais e não funcionais definidos para o sistema desenvolvido. Para esse efeito, serão realizados ensaios dedicados aos principais subsistemas da plataforma, avaliando o seu desempenho em condições representativas de utilização industrial. Na Tabela 2 são definidos os objetivos, as condições de teste, os equipamentos utilizados e os critérios de aceitação.

Dispositivo	Condições de teste	CrITÉrios de aceitação
CAN	• Velocidade de comunicação: 1 Mb/s; • Dois canais CAN em funcionamento; • Troca contínua de mensagens; • Rede composta por dispositivos CAN simulados ou reais.	• Receção de 100% das mensagens transmitidas; • Ausência de erros de barramento; • Atualização correta dos sinais recebidos.
LIN	• Velocidade de comunicação: 19,2 kb/s; • Um mestre e múltiplos escravos; • Leitura e escrita de sinais definidos em ficheiro LDF.	• Resposta correta de todos os escravos; • Ausência de timeouts; • Atualização correta dos sinais.
Fonte Programável	• Configuração das tensões de saída em 5, 12 e 24 V; • Configuração dos limites de corrente em 0.2, 0.5 e 1A; • Medição da tensão e corrente de saída através de equipamento externo calibrado (multímetro).	• Erro inferior a $\pm 2\%$; • Funcionamento correto da limitação de corrente; • Estabilidade da saída durante o ensaio; • Concordância entre os valores indicados pelo sistema e os valores medidos pelo equipamento externo.

DAQ	● Aplicação de sinais de referência de 0, 2.5, 5, 7.5 e 10V; ● Ensaios realizados nos 4 canais analógicos; ● Comparação dos valores medidos com equipamento de referência externo calibrado; ● Verificação do funcionamento à taxa máxima de aquisição de 100 kS/s; ● Realização de medições repetidas para avaliação da repetibilidade.	● Erro de medição dentro da precisão especificada; ● Concordância entre os valores medidos pelo sistema e pelo equipamento de referência externo; ● Funcionamento simultâneo dos 4 canais; ● Ausência de perda de amostras a 100 kS/s; ● Repetibilidade das medições dentro das tolerâncias definidas.
Entradas Digitais	● Aplicação de sinais de 0 V, 2 V, 12 V, 24 V e 28 V; ● Verificação do comportamento em todos os canais de entrada; ● Medição da corrente de entrada para os diferentes níveis de tensão aplicados; ● Verificação do funcionamento contínuo durante múltiplas comutações de estado.	● Identificação correta dos estados ON/OFF em todos os canais; ● Detecção do estado lógico HIGH para tensões iguais ou superiores a 2 V, garantindo compatibilidade com dispositivos baseados em lógica TTL; ● Ausência de ativações indevidas ou transições espúrias; ● Funcionamento simultâneo de todos os canais; ● Corrente de entrada inferior a 60 mA quando aplicada a tensão máxima de 28 V, garantindo a proteção e integridade dos optoacopladores utilizados.
Saídas Digitais	● Acionamento individual dos quatro relés; ● Ensaio de 1000 ciclos consecutivos de comutação para cada relé; ● Medição do tempo de resposta entre o comando de	● Comutação correta dos contactos em todos os ciclos; ● Ausência de falhas de acionamento durante o ensaio; ● Estado lógico consistente após cada comutação;

	acionamento e a alteração do estado do contacto.	• Tempo de comutação inferior a 3 ms, de acordo com as especificações do fabricante do relé.
Bateria	Ensaio 1 – Funcionamento em vazio: • Sistema alimentado exclusivamente por bateria; • Sem carga aplicada à saída programável; • Interface gráfica ativa; • Monitorização contínua da tensão da bateria até ao desligamento do sistema. Ensaio 2 – Funcionamento sob carga: • Sistema alimentado exclusivamente por bateria; • Fonte programável configurada para 12 V; • Carga constante de 70 mA aplicada à saída; • Comunicação LIN ativa durante todo o ensaio; • Interface gráfica ativa; • Monitorização contínua da tensão da bateria até ao desligamento do sistema.	• Funcionamento contínuo e estável durante todo o ensaio; • Ausência de reinicializações ou falhas de comunicação; • Manutenção da comunicação LIN durante toda a descarga da bateria; • Tensão da bateria dentro dos limites de funcionamento até ao final do ensaio; • Autonomia mínima de 8 horas no cenário representativo de utilização.

Tabela 2 - Requisitos de validação experimental

4. Arquitetura da Solução

4.1. Visão Global do Sistema

A **eTestBox** foi concebida como um sistema modular composto por subsistemas interligados, permitindo flexibilidade na configuração e facilidade de manutenção. Esta abordagem modular garante que cada componente (aquisição de dados, comunicação veicular, entradas/saídas digitais, fonte de alimentação e interface de utilizador) possa ser desenvolvido, testado e atualizado de forma independente, sem comprometer o funcionamento global da solução.

A arquitetura segue o princípio de **camadas funcionais**, em que o **Raspberry Pi** atua como controlador central, interligando os diferentes módulos através de interfaces padrão (USB, SPI, I²C, GPIO). Este modelo facilita a escalabilidade do sistema, permitindo a integração futura de novos periféricos ou protocolos, sem necessidade de alterações estruturais significativas.

Adicionalmente, a utilização de componentes de baixo custo e com ampla documentação (como o Raspberry Pi e o módulo DAQ MCC128) assegura não apenas a viabilidade económica do projeto, mas também a sua **sustentabilidade técnica**, uma vez que a substituição ou atualização de módulos pode ser feita de forma simples. Esta arquitetura foi também pensada para **suportar operações em diferentes contextos industriais**, desde bancadas de laboratório até linhas de produção, destacando-se pela portabilidade e pela possibilidade de alimentação autónoma através de bateria.



Figura 11 - Exemplo de sistema de diagnostico portátil (Ultima EDI Engine Diagnostic Instrument)

4.2. Arquitetura Funcional

A arquitetura da **eTestBox** é dividida em:

- **Camada de Interface de Utilizador** - Implementada em PyQt seguindo o padrão MVVM, permite a interação com todos os módulos da ferramenta.
- **Camada de Controlo Central** - Raspberry Pi como controlador principal, responsável pela orquestração das operações e comunicação entre módulos.
- **Camada de Aquisição de Dados** - Módulo DAQ MCC128 para leitura de sinais analógicos.
- **Camada de Comunicação Veicular** - Interfaces para LIN e CAN, com suporte a leitura e escrita de sinais com base em ficheiros LDF e DBC.
- **Camada de I/O Digital** - GPIOs controlados via biblioteca gpiod para acionamento de relés e leitura de entradas digitais.
- **Camada de Alimentação** - Fonte programável com medição e regulação de tensão e corrente.

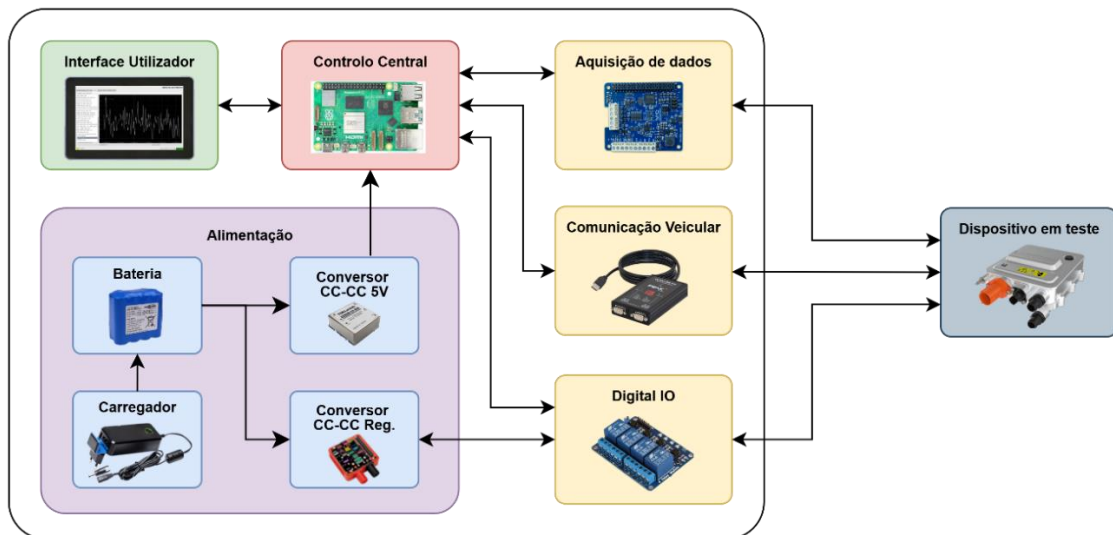


Figura 12 - Diagrama de blocos da eTestBox

4.2.1. Interface de Utilizador

Como interface de utilizador foi escolhido o **EDATEC HMI3010-101C**, um módulo HMI de 10,1" concebido para integração direta com o Raspberry Pi e adequado a ambientes industriais. Esta solução oferece um ecrã tátil capacitivo robusto, de elevada sensibilidade e resistente ao uso intensivo, permitindo uma interação intuitiva mesmo quando operado com luvas finas - uma necessidade frequente em linhas de produção automóvel. A sua compatibilidade elétrica e mecânica com o Raspberry Pi elimina a necessidade de adaptadores ou estruturas adicionais, reduzindo significativamente a complexidade do sistema e aumentando a fiabilidade global. A estrutura HMI industrial assegura ainda proteção mecânica, rigidez estrutural e capacidade de funcionamento contínuo, características essenciais para um equipamento como a eTestBox, destinado a uso prolongado em ambiente industrial.



Figura 13 - Vista frontal e lateral do HMI

A seleção deste dispositivo foi orientada por critérios como **tamanho**, **luminosidade**, **durabilidade** e **compatibilidade**. O tamanho de 10,1" revelou-se ideal para a interface a desenvolver, permitindo apresentar simultaneamente gráficos, medições, controlos de teste e menus sem comprometer a legibilidade. A luminosidade elevada e o painel IPS garantem excelente visibilidade sob diferentes condições de iluminação. A durabilidade do módulo, incluindo a resistência a vibrações e a operação contínua 24/7, assegura estabilidade a longo prazo. Já a compatibilidade com o Raspberry Pi, tanto a nível mecânico

como de alimentação e interfaces, foi determinante para permitir uma integração limpa, sólida e de baixo risco.

Durante a fase de estudo, foram consideradas alternativas como monitores HDMI com painel tátil USB, tablets Android e HMIs industriais convencionais. Os monitores HDMI apresentavam maior volumetria, necessidade de cabos adicionais e ausência de integração mecânica, aumentando o risco de falhas e a complexidade de montagem. Os tablets, embora portáteis, não oferecem integração nativa com GPIOs, não permitem controlo direto de hardware industrial e apresentam restrições importantes no software. Por outro lado, HMIs industriais tradicionais, apesar da sua robustez, são significativamente mais caros, menos flexíveis e pouco compatíveis com o ecossistema Python utilizado neste projeto. Assim, o HMI3010-101C destacou-se como a opção com melhor relação entre robustez, integração, custo e experiência de utilização, justificando plenamente a sua adoção como interface principal da eTestBox.

A interface gráfica da eTestBox será desenvolvida em **PyQt**, tirando partido da sua flexibilidade e da integração nativa com o ecossistema Python previsto para todo o sistema. Para garantir uma arquitetura software modular, escalável e fácil de manter, será adotado o padrão **MVVM (Model–View–ViewModel)**. Esta abordagem permitirá que o sistema evolua de forma organizada, facilitando a integração de novos módulos de hardware ou funcionalidades futuras, sem comprometer a estabilidade da interface. A estrutura da aplicação será organizada em **separadores funcionais**, correspondentes às principais áreas operacionais da eTestBox: controlo da fonte programável, aquisição analógica (DAQ), gestão das entradas e saídas digitais (GPIOs), comunicação LIN/CAN e sequenciador de testes. Esta organização permitirá uma navegação intuitiva e uma utilização eficiente por parte do operador, mantendo uma visão clara e segmentada das diferentes funcionalidades do sistema.

4.2.2. Controlador Central

O **controlador central** da eTestBox é baseado num **Raspberry Pi 5**, escolhido pela sua versatilidade, baixo custo e elevada integração de interfaces. Esta unidade coordena a execução das tarefas de aquisição, comunicação e atuação,

funcionando como núcleo do sistema, assegurando a comunicação entre os diferentes módulos e a interface com o operador.

O Raspberry Pi foi selecionado em detrimento de PLCs industriais ou controladores dedicados devido a:

- **Custo acessível**, permitindo uma solução economicamente competitiva;
- **Flexibilidade** de software, com suporte nativo a Python e bibliotecas de comunicação (CAN, LIN, GPIO, DAQ);
- **Dimensões compactas e baixo peso**, adequados a um sistema portátil;
- **Consumo energético reduzido**, tipicamente entre 3 W e 7 W, o que possibilita a sua alimentação direta através de uma **bateria portátil** (powerbank ou bateria dedicada).
- Comunicações integradas, incluindo **Ethernet, Wi-Fi e Bluetooth**, eliminando a necessidade de módulos externos adicionais.

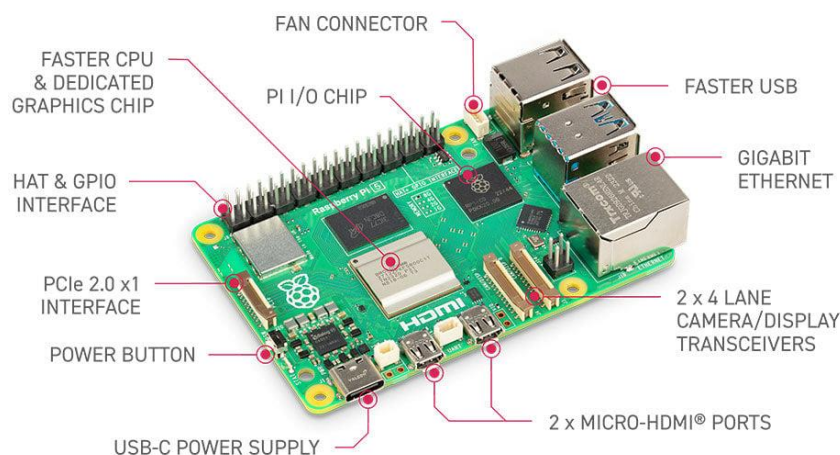


Figura 14 - Visão geral do Raspberry Pi 5 (Donnison, 2023)

A utilização de baterias garante que a **eTestBox pode operar em ambientes sem acesso imediato à rede elétrica**, fator crítico em testes de campo ou em linhas de produção onde a mobilidade é um requisito essencial.

Apesar das vantagens funcionais e económicas, a adoção do Raspberry Pi num contexto industrial introduz algumas restrições técnicas que devem ser consideradas no dimensionamento do sistema:

- **Armazenamento baseado em cartão microSD**, cuja tecnologia NAND flash apresenta desgaste progressivo associado a ciclos de escrita, podendo comprometer a fiabilidade a longo prazo em aplicações com elevado volume de *logging*. Para mitigar o desgaste associado à memória flash dos cartões microSD, foram instalados cartões de uso industrial e o sistema foi concebido para minimizar operações de escrita contínua. Os dados relevantes são registados apenas quando necessário e podem ser exportados ou sincronizados com sistemas externos, reduzindo o número de ciclos de escrita no armazenamento local.
- **Ausência de determinismo temporal nativo**, uma vez que o sistema operativo Linux não fornece, por defeito, características de tempo real. No entanto, os requisitos temporais da aplicação desenvolvida não exigem determinismo estrito. As tarefas executadas, como a aquisição de dados, a comunicação com equipamentos e a execução de sequências de teste, apresentam tolerâncias temporais da ordem de milissegundos, sendo plenamente compatíveis com o modelo de escalonamento do Linux convencional.
- **Robustez elétrica inferior a equipamentos industriais** dedicados, exigindo proteção adicional contra sobretensões, ruído eletromagnético, descargas eletrostáticas (ESD) e variações abruptas de alimentação. Para aumentar a robustez elétrica do sistema, foram consideradas medidas de proteção ao nível da alimentação e das interfaces de comunicação, incluindo utilização de fontes estabilizadas, filtragem adequada e isolamento dos sinais críticos sempre que necessário.
- **Limitações térmicas** em operação contínua sob carga elevada, podendo requerer dissipação ativa ou passiva adicional para garantir estabilidade. No sistema desenvolvido, esta limitação foi mitigada através da instalação de ventilação ativa, recorrendo a um ventilador dedicado para promover a circulação de ar e melhorar a dissipação térmica do Raspberry Pi, garantindo assim temperaturas de funcionamento estáveis durante operação prolongada.

4.2.3. Sistema de Aquisição Analógica

A **aquisição analógica** na eTestBox é assegurada pelo módulo **MCC128**, desenvolvido pela Measurement Computing Corporation e concebido para funcionar em conjunto com o **Raspberry Pi** através da interface SPI. Este módulo desempenha um papel fundamental na monitorização de parâmetros elétricos dos dispositivos sob teste (UUTs), permitindo medir com precisão tensões e correntes durante os ensaios funcionais.

O módulo MCC128 integra as seguintes especificações técnicas:

- **8 canais analógicos de entrada**, configuráveis individualmente como *single-ended* (referenciados a um comum) ou em 4 pares diferenciais, garantindo maior imunidade a ruído em ambientes industriais.
- Conversor analógico–digital (ADC) de **16 bits**, proporcionando elevada resolução nas medições, com capacidade de detetar variações muito pequenas nos sinais elétricos.
- Taxa de amostragem máxima de **100 kS/s**, ajustável conforme o tipo de teste, permitindo desde medições de baixa frequência (tensão contínua) até monitorização de sinais com maior dinâmica.
- Faixa de entrada programável de **±10 V**, compatível com a maioria dos sinais elétricos utilizados em módulos automóveis.



Figura 15 - DAQ MCC128 (Measurement Computing Corporation, 2023)

Na arquitetura da eTestBox, o módulo **MCC128** está diretamente interligado ao Raspberry Pi através do barramento **SPI**, assegurando comunicação síncrona de baixa latência e elevada integridade de dados.

A configuração dos canais analógicos é efetuada via software, a partir da interface gráfica do sistema, permitindo ao operador selecionar individualmente o modo de aquisição em função das características elétricas da UUT.

No contexto operacional da eTestBox, a aquisição analógica é utilizada essencialmente para:

- Monitorização da **tensão de alimentação** fornecida à UUT;
- Medição da **corrente consumida** durante diferentes fases do teste;
- **Validação de limites** definidos para **parâmetros elétricos**, com comparação em tempo real face a valores de referência definidos no sequenciador de testes, desencadeando alarmes ou reprovação da UUT em caso de não conformidade;
- Análise de **comportamento dinâmico de sinais**, incluindo deteção de oscilações, *ripple*, picos transitórios ou variações abruptas associadas a mudanças de estado funcional.

A adoção do MCC128 apresenta as seguintes vantagens técnicas no contexto da eTestBox:

- **Custo reduzido** comparativamente a soluções de laboratório.
- **Integração nativa com o Raspberry Pi**, sem necessidade de controladores adicionais.
- **Precisão adequada** para testes funcionais em ambiente industrial.

Contudo, devem ser consideradas as seguintes limitações:

- **Número restrito de canais** (8 entradas, face a sistemas mais avançados que oferecem dezenas ou centenas).
- **Taxa de amostragem inferior** a soluções de alta performance (ex.: sistemas NI PXI).
- Ausência de **saídas analógicas**, restringindo a geração de sinais de estímulo.

Apesar destas limitações, o módulo MCC128 cumpre plenamente os requisitos definidos para a eTestBox, permitindo equilibrar **desempenho, portabilidade e custo**, o que o torna a escolha mais adequada para esta solução.

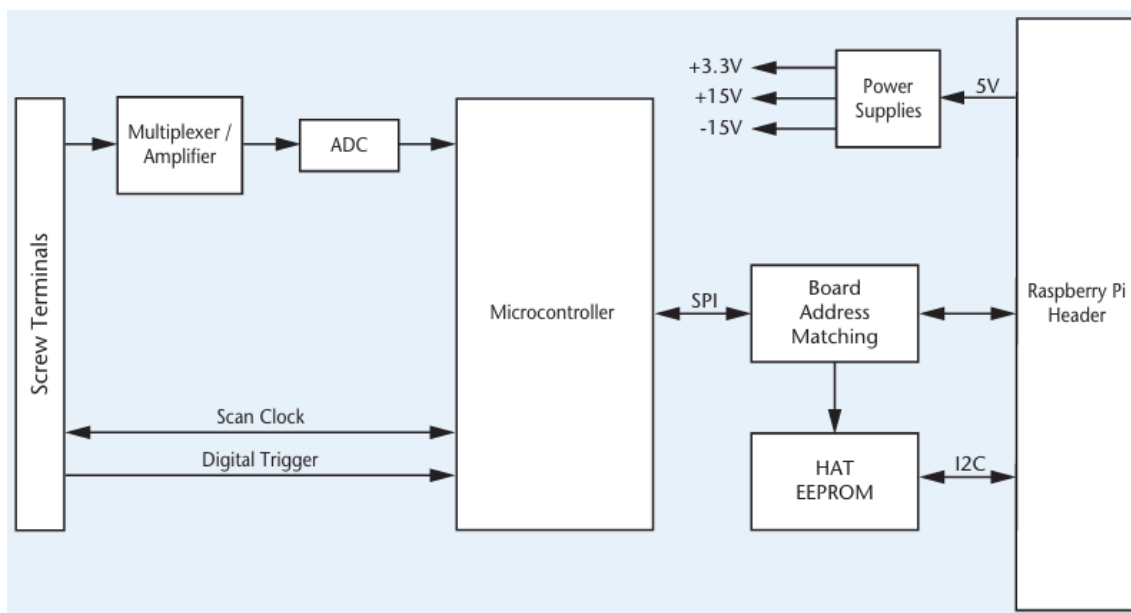


Figura 16 - Diagrama blocos da DAQ (Measurement Computing Corporation, 2023)

4.2.4. Comunicação LIN/CAN

A comunicação veicular na eTestBox é assegurada pelo módulo **PCAN-Pro FD**, desenvolvido pela **PEAK-System Technik GmbH**, que oferece **dois canais CAN FD** e **dois canais LIN** num único dispositivo de interface USB.

A interface suporta a operação concorrente de todos os canais, possibilitando a execução de ensaios paralelos em duas UUTs distintas, mantendo isolamento lógico e físico entre redes. Esta capacidade é particularmente relevante em cenários de validação funcional e EOL com múltiplos módulos, onde é necessário garantir independência entre barramentos e ausência de interferência cruzada.

O PCAN-Pro FD apresenta as seguintes especificações:

- **2 canais CAN FD**, suportando taxas de transmissão até 8 Mbit/s, compatíveis com arquiteturas automóveis modernas que utilizam *bit rate switching* (BRS) para aumento de velocidade de comunicação;
- **2 canais LIN** independentes, com velocidades até 20 kbit/s, operando em modo *Master* ou *Slave*, permitindo simulação de nós de rede ou monitorização passiva conforme o cenário de teste.
- **Interface USB 2.0**, garantindo integração direta com o Raspberry Pi.

- **Isolamento galvânico** nos canais de comunicação, assegurando separação elétrica entre a interface e a UUT, aumentando a imunidade a ruído eletromagnético e protegendo contra sobretensões ou diferenças de potencial entre massas.



Figura 17 - Peak PCAN-Pro FD (PEAK-System, 2025)

As interfaces **CAN** do PCAN-Pro FD são suportados nativamente em Linux através do *framework* **SocketCAN**, uma camada de abstração que integra dispositivos CAN como interfaces de rede (netdev).

Desta forma, cada canal CAN é identificado no sistema como um dispositivo de rede independente (por exemplo, **can0** e **can1**), permitindo ao software aceder ao barramento CAN com as mesmas chamadas utilizadas para comunicação TCP/IP.

Para o **LIN**, o suporte é assegurado pelo **PLIN driver** para Linux, também fornecido pela PEAK-System. Este driver permite a utilização dos canais LIN do PCAN-Pro FD através da interface USB, sendo compilado e instalado localmente para o *kernel* ativo. O módulo principal do driver, que é automaticamente carregado pelo sistema quando o dispositivo é conectado (PEAK-System, 2025). Os canais de comunicação do PCAN-Pro FD são disponibilizados através de duas fichas DB9, cada uma correspondendo a um conjunto independente de canais CAN e LIN, permitindo a ligação direta de até duas UUTs em simultâneo.

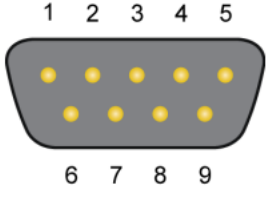
Pin	Assignment	D-Sub socket
1	CAN_V+ (external, optional)	
2	CAN_Low	
3	CAN_GND	
4	LIN	
5	LIN_GND	
6	LIN_GND	
7	CAN_High	
8	Not assigned	
9	LIN_VBat	

Tabela 3 - Atribuição de pinos do PCAN-Pro FD

4.2.5. Entradas e Saídas Digitais

A camada de **entradas e saídas digitais (I/O)** da eTestBox foi desenvolvida para permitir a interação direta com as **UUTs**, possibilitando tanto a leitura de sinais discretos provenientes do sistema em teste como o controlo de atuadores e cargas externas.

Esta funcionalidade é essencial para a realização de **ensaios funcionais**, nos quais é necessário verificar o estado lógico de sensores, botões ou indicadores, bem como acionar saídas digitais que simulam condições de operação reais.

O controlo das entradas e saídas digitais é realizado através da biblioteca **gpiod**, que permite o acesso direto aos **GPIOs** do Raspberry Pi em ambiente Linux. Cada UUT dispõe de **duas entradas digitais optoisoladas e duas saídas digitais de contactos secos**, o que permite a realização de testes independentes e paralelos.

As **entradas digitais** da eTestBox foram concebidas para garantir **isolamento elétrico completo, compatibilidade com diferentes tensões de sinal e imunidade a ruído**, assegurando a integridade do sistema de controlo mesmo em ambientes industriais adversos.

Cada entrada é baseada num **optocoplador SFH620A**, que converte o sinal proveniente do UUT em um nível lógico compatível com o Raspberry Pi, mantendo separação galvânica entre ambos.

Na Figura 18, os sinais de entrada aplicados pelo UUT1 chegam ao conetor **J7** e são encaminhados às entradas dos optocopladores **U5** e **U7** através das

resistências **R12** e **R13** de **1 kΩ**, que limitam a corrente que atravessa o LED interno de cada optocoplador.

Quando um sinal de entrada é aplicado, o LED é polarizado e emite luz, ativando o fototransistor correspondente no lado secundário. Esse transistor conduz e altera o estado lógico dos pinos **GPIO5** e **GPIO6** do Raspberry Pi, que passam a nível lógico baixo, indicando o estado ativo da entrada.

As resistências **R20** e **R21**, de **10 kΩ**, configuradas como *pull-up* a **+3,3 V**, asseguram um nível lógico alto quando não há sinal aplicado, enquanto **R16** e **R17** (1 kΩ) limitam a corrente de base e estabilizam o circuito de leitura.

Esta configuração assegura que o Raspberry Pi nunca esteja diretamente exposto ao potencial elétrico do UUT, eliminando o risco de danos por diferença de massa ou sobretensão.

Do ponto de vista elétrico, o dimensionamento da resistência de entrada de 1 kΩ foi escolhido para permitir o funcionamento seguro do LED do optocoplador em toda a gama de tensões esperadas. Como a corrente máxima admissível do LED é 60 mA, pela Lei de Ohm ($I = V/R$) obtém-se que a tensão máxima segura de operação é aproximadamente 60 V ($I \times R = 60 \text{ mA} \times 1 \text{ k}\Omega$). Desta forma, a entrada pode receber sinais de até 60 V DC sem ultrapassar os limites do componente, garantindo ampla margem de segurança.

Um aspeto adicional particularmente vantajoso é o facto de o **LED** interno do SFH620A ser **bidirecional**, o que significa que o circuito é **insensível à polaridade** do sinal aplicado pelo UUT. Isto simplifica a cablagem, elimina a necessidade de controlo de polaridade durante a ligação e assegura o funcionamento correto mesmo em casos de inversão acidental dos condutores. Esta solução proporciona uma interface de entrada **robusta, tolerante a ruído e eletricamente isolada**, capaz de suportar sinais industriais de 12 a 24 V e picos transitórios ocasionais até 60 V. Combinada com o isolamento galvânico do optocoplador (até **5 kV**).

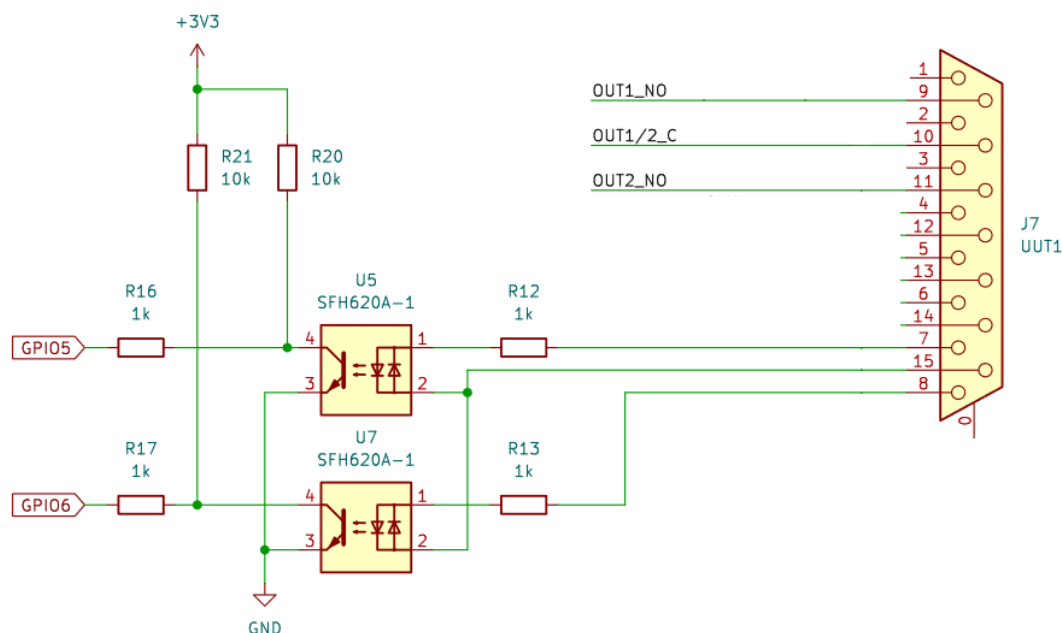


Figura 18 - Circuito das entradas digitais da eTestBox

As **saídas digitais** da eTestBox foram projetadas para permitir o **controle direto de sinais discretos** e o **acionamento de cargas externas** durante os ensaios funcionais dos **UUTs**, assegurando ao mesmo tempo isolamento elétrico e robustez contra picos de tensão ou ruído eletromagnético.

Cada UUT dispõe de **duas saídas digitais independentes**, implementadas através de **relés eletromecânicos Omron G6K-2**, comandados pelos **GPIOs 17 e 18** do Raspberry Pi. Estes relés disponibilizam **contactos secos** (sem potencial elétrico), o que permite uma ampla compatibilidade com diferentes tipos de circuitos, tanto em corrente contínua como alternada, até **1 A por canal**.

A Figura 19 apresenta o esquema elétrico das saídas digitais implementadas na eTestBox. O controle das bobinas dos relés é efetuado através de transístores **BC817** configurados como **interruptores NPN de saturação**. Cada GPIO ativa o seu respetivo transístor (**Q4** ou **Q5**) por meio de uma resistência de base de **1 k Ω** (**R8** e **R9**), permitindo o fluxo de corrente entre o emissor e o coletor. Quando o transístor entra em condução, estabelece o caminho de massa da bobina do relé, provocando a sua comutação.

A energia de retorno gerada pelo colapso do campo magnético da bobina, no momento de desligar, é absorvida por um **díodo de roda livre (freewheeling)** do tipo **SS14 (D5 e D6)**, colocado em antiparalelo com a bobina. Estes díodos de silício Schottky protegem os transístores e o GPIO contra sobretensões,

garantindo uma operação fiável e duradoura mesmo em ciclos de comutação repetidos.

O circuito é alimentado a **+5 V**, proveniente da fonte principal da eTestBox, o que assegura uma comutação rápida e eficiente das bobinas dos relés. Os contactos de saída (**OUT1_NO**, **OUT2_NO** e **OUT1/2_C**) são encaminhados para o conector dedicado ao UUT, disponibilizando **dois canais de contacto seco** que podem ser configurados em modo independente ou comum.

Esta topologia apresenta diversas vantagens: a separação galvânica entre o circuito de controlo e o circuito de carga, a proteção ativa contra sobretensões, e a compatibilidade com uma ampla gama de sinais de comando e tensão. Além disso, o uso de relés miniatura G6K-2 permite manter o **baixo consumo energético e dimensões reduzidas do PCB**, sem comprometer a durabilidade - tipicamente superior a **10⁶ ciclos mecânicos**.

Durante os testes automáticos, as saídas são controladas diretamente pelo software da eTestBox, podendo ser ativadas manualmente através da interface gráfica ou automaticamente pelo **sequenciador de testes**, em sincronização com medições analógicas, leituras digitais e comunicação LIN/CAN.

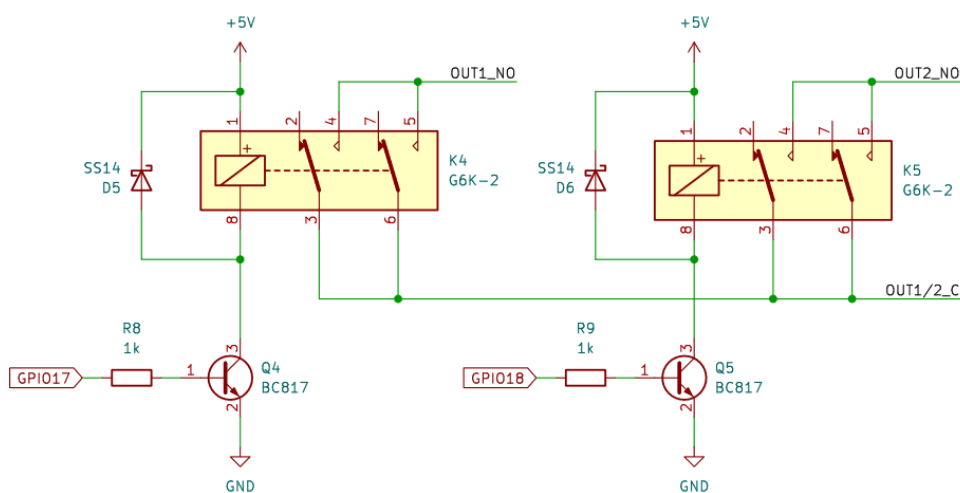


Figura 19 - Circuito das saídas digitais da eTestBox

4.2.6. Sistema de Alimentação

A fonte de alimentação da **eTestBox** foi projetada para fornecer energia estável, eficiente e isolada a todos os módulos do sistema, incluindo o Raspberry Pi, a DAQ MCC128, as interfaces LIN/CAN, os circuitos de I/O digital e os UUTs em

teste. O projeto da fonte combina **conversores DC-DC isolados**, reguladores lineares e proteções ativas, garantindo operação segura tanto em modo ligado à rede como em modo **portátil**, através da bateria interna de 14,4 V.

A **arquitetura geral** do sistema de alimentação, representada na Figura 20, assegura **autonomia energética, segurança elétrica e controlo do processo de inicialização e encerramento** do dispositivo, preservando a integridade do sistema operativo do Raspberry Pi e prolongando a vida útil da bateria. A **alimentação principal** é proveniente de uma **bateria de iões de lítio de 14,4 V / 7000 mAh**, composta por múltiplas células conectadas em série e configurada para fornecer energia aos barramentos internos de **+5 V, +12 V e +30 V**. Esta bateria está ligada ao circuito através dos pinos 1 e 2 do conector J1, constituindo a **principal fonte de energia** do sistema.

Nos pinos 3 e 4 do mesmo conector encontra-se ligado um **interruptor geral**, que permite a desconexão física completa do equipamento, interrompendo toda a alimentação da eTestBox e eliminando qualquer consumo residual.

O processo de **carregamento da bateria** é efetuado através de um **carregador externo**, conectado ao **conector J2**, o qual fornece a tensão e corrente adequada para recarregar as células. O caminho de carga é protegido pelo **fusível F1**, que previne danos em caso de avaria do carregador ou sobrecorrente durante o processo de recarga.

De forma complementar, o **fusível F2** protege a bateria contra **curtos-circuitos ou sobrecargas** originadas na própria eTestBox, evitando degradação das células e riscos térmicos.

Um **relé biestável K1** desempenha um papel central no sistema, atuando como **interruptor principal de energia (power latch)**. Este relé possui **duas bobinas independentes**, permitindo **comutação em ambas as direções (ligar e desligar)**, sem consumo contínuo de corrente após o acionamento. Quando o utilizador pressiona o **botão de ligar**, ligado aos **pinos 1 e 2 do conector J3**, o **transístor Q9** é ativado, energizando a bobina de ligação do relé. O fecho dos contactos de K1 conecta a bateria ao barramento principal de alimentação, fornecendo energia a todos os módulos da eTestBox e iniciando automaticamente o **Raspberry Pi**.

O **processo de encerramento** foi concebido para ocorrer de forma **controlada e segura**, evitando a interrupção abrupta da alimentação do sistema operativo. Para isso, o **GPIO23 do Raspberry Pi** é configurado a nível de BIOS para assumir o **estado lógico alto** após o encerramento completo do sistema operativo. Este sinal comanda, através de um **transistor de controlo (Q1)**, a bobina de desligamento do relé K1, que abre os contactos e interrompe totalmente a ligação da bateria, garantindo que não exista descarga quando o dispositivo está desligado.

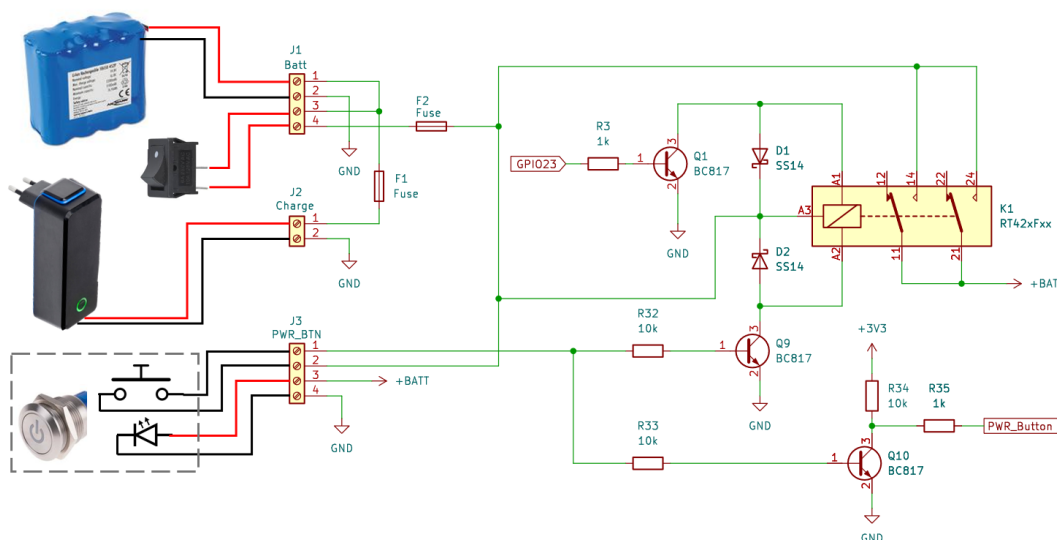


Figura 20 - Circuito de gestão de alimentação da eTestBox

A arquitetura inclui **conversores CC–CC** para a alimentação principal, circuitos analógicos e de instrumentação, complementados por um controlo inteligente que permite reduzir o consumo global e prolongar a autonomia da bateria.

O **conversor U1**, ilustrado na Figura 21, é responsável pela conversão da tensão da bateria (14,4 V) para **+5 V**, constituindo a **fonte principal de alimentação** de todo o sistema. Este estágio fornece energia ao Raspberry Pi, ao display, à DAQ MCC128, às interfaces LIN/CAN, e aos circuitos digitais auxiliares.

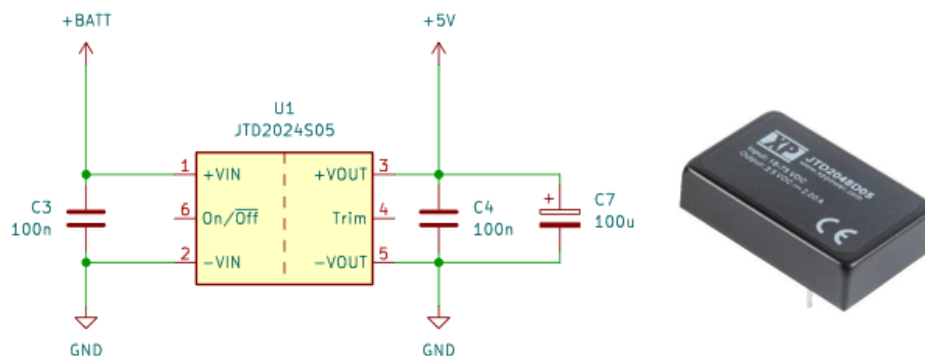


Figura 21 - Conversor DC-DC principal U1 (JTD2024S05)

Faixa de tensão de entrada [DC]	9 to 36V
Eficiência	89%
Tensão de saída nominal [DC]	+5V
Corrente de saída máxima	4 ^a
Ruído máximo de ondulação	75mV
Potência máxima de saída	20W

Tabela 4 - Especificações técnicas do conversor DC-DC JTD2024S05

O **conversor U4**, representado na Figura 22, é um módulo **isolado e simétrico** responsável pela geração das tensões de **+15 V e -15 V**. Neste projeto, a linha negativa (**Vout-**) é referenciada ao **GND**, criando assim um **barramento de 30 V** utilizado como **fonte de alimentação de alta tensão** para módulos auxiliares. Essa configuração permite alimentar circuitos que requerem tensões mais elevadas sem a necessidade de uma segunda bateria ou de um conversor boost adicional.

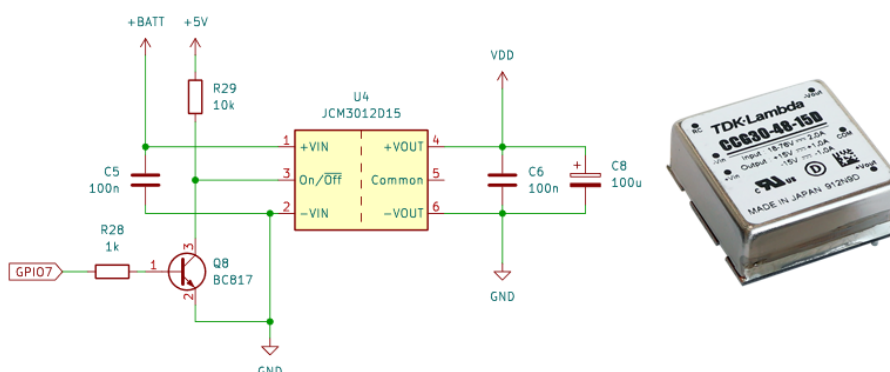


Figura 22 - Conversor DC-DC 30V U4 (CCG30-24-15D)

Faixa de tensão de entrada [DC]	9 to 36V
Eficiência	91%
Tensão de saída nominal [DC]	±15V
Corrente de saída máxima	1A
Ruído máximo de ondulação	95mV
Potência máxima de saída	30W

Tabela 5 - Especificações técnicas do conversor DC-DC CCG30-24-15D

O **barramento de 30 V** alimenta diretamente dois subsistemas:

- o **conversor U6**, que gera uma **saída fixa de 12 V** para alimentar **cargas externas** ou acessórios de teste do UUT;
- e a **fonte programável**, controlada pelo software da eTestBox, que permite ajustar dinamicamente a tensão e corrente de saída fornecida ao UUT, simulando diferentes condições de operação.

De forma a otimizar o desempenho energético, o **barramento de 30 V pode ser desligado através do GPIO7** do Raspberry Pi, que comanda a entrada ligar/desligar do conversor U4. Esta funcionalidade reduz o consumo residual dos conversores quando estes não estão em utilização, particularmente em ensaios onde o UUT é alimentado externamente, contribuindo assim para aumentar significativamente a autonomia da eTestBox.

A eTestBox integra também um **módulo de alimentação programável** baseado no M5Stack PPS, uma fonte DC controlada digitalmente, concebida para fornecer tensão e corrente ajustáveis ao UUT durante os ensaios funcionais. Este módulo foi escolhido devido à sua elevada flexibilidade, precisão e capacidade de controlo direto por software, permitindo simular diferentes condições elétricas aplicadas ao dispositivo em teste.

Para além disso, permite a monitorização contínua da tensão e corrente fornecidas ao UUT, permitindo ao sistema detetar desvios ou anomalias em tempo real, bem como registar estes parâmetros para análise posterior. A segurança operacional é reforçada graças aos mecanismos de proteção

integrados no módulo - incluindo proteção contra **sobrecorrente**, **sobretensão** e **sobretensão** - evitando danos tanto no UUT como no restante hardware da eTestBox.

No que diz respeito à sua integração elétrica, ilustrada na Figura 23, a alimentação é fornecida através do conector J10, ligado diretamente ao barramento interno de 30 V gerado pelo conversor simétrico U4. Após regulação interna, a tensão programada é disponibilizada ao UUT através do conector J11, garantindo um caminho de potência dedicado. Desta forma, o módulo atua como uma fonte programável totalmente integrada, capaz de alimentar o UUT com condições elétricas ajustáveis e monitorizadas.

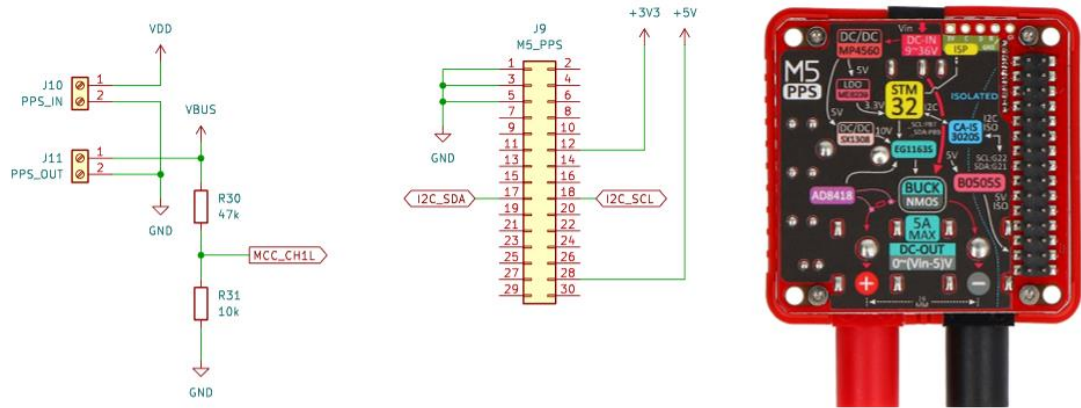


Figura 23- Fonte de alimentação programável M5Stack PPS

Faixa de tensão de entrada [DC]	9 to 36V
Faixa de tensão de saída [DC]	0.5-30V
Faixa de corrente de saída [DC]	0-5A
Precisão da tensão na saída	30mV
Precisão da corrente na saída	5mA
Potência máxima de saída	100W

Tabela 6 - Especificações técnicas da fonte M5Stack PPS

4.2.7. Interface com a UUT

A Figura 24 apresenta a interface entre a eTestBox e o UUT é realizada através de um conector **DB15** que agrega todas as ligações necessárias aos ensaios funcionais, incluindo entradas e saídas digitais, medições analógicas, comunicação automível e alimentação. Cada UUT dispõe de duas entradas

digitais isoladas, duas saídas digitais com contactos secos, uma entrada analógica configurável como diferencial ou como duas single-ended. A alimentação programável fornecida ao UUT é disponibilizada através do pino KL30, comandado pelos relés K2/K3, permitindo ligar ou cortar a energia proveniente do módulo M5Stack PPS; quando o relé está desligado, a saída é automaticamente descarregada por uma resistência de 100 Ω , garantindo a **descarga controlada da energia residual** no UUT, prevenindo que permaneça qualquer tensão após o término do ensaio.

A corrente consumida no KL30 é monitorizada pelo sensor de efeito Hall ACS723, cuja saída é lida pela DAQ.

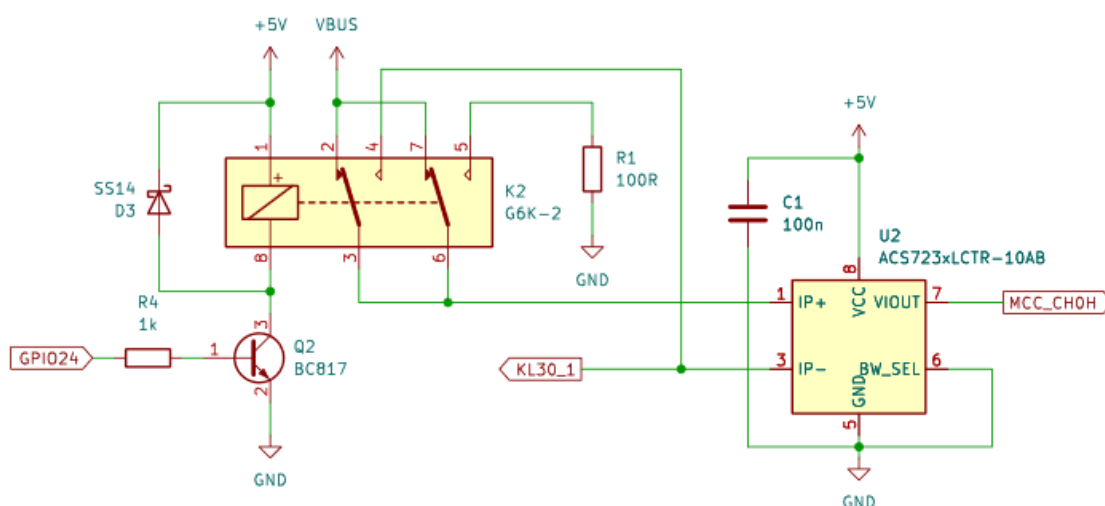


Figura 24 - Circuito de comutação e medição de corrente da alimentação KL30

Adicionalmente, o DB15 fornece uma alimentação fixa de 12 V no pino 14 para módulos que requeiram tensão estabilizada, e integra comunicação CAN através dos pinos 3 e 4, LIN através do pino 5. O pino 6 correspondente à alimentação do transceiver LIN do UUT. Esta interface unificada permite que a eTestBox execute testes completos, seguros e totalmente automatizados sobre uma vasta gama de módulos eletrónicos automóveis.

Pino	Nome / Função	Descrição Técnica
1	KL30	Alimentação programável para o UUT
2	GND	Massa comum para alimentação e comunicação
3	CAN_H	Linha CAN High para comunicação CAN/CAN FD.
4	CAN_L	Linha CAN Low para comunicação CAN/CAN FD.
5	LIN	Linha de comunicação LIN.
6	VBatt_LIN	Alimentação do transceiver LIN do UUT.
7	IN1	Entrada digital 1 (Opto-isolador bidirecional. Max 60V).
8	IN2	Entrada digital 2 (Opto-isolador bidirecional. Max 60V).
9	OUT1	Saída digital 1 (contacto seco normalmente aberto. Max 1A).
10	OUT_COM	Comum das saídas digitais (relés 1 e 2).
11	OUT2	Saída digital 2 (contacto seco normalmente aberto. Max 1A).
12	ADC_H	Entrada analógica – canal high (modo diferencial) ou canal single-ended 1.
13	ADC_L	Entrada analógica – canal low (modo diferencial) ou canal single-ended 2.
14	+12V	Alimentação fixa de 12V (Max 1A).
15	IN_COM	Comum das entradas digitais (opto-isoladores 1 e 2).

Tabela 7 - Atribuição de pinos da ligação ao UUT

5. Implementação Técnica

Este capítulo apresenta a materialização técnica da solução concebida, descrevendo de forma estruturada a implementação dos diferentes módulos que compõem a eTestBox. Após a definição conceptual da arquitetura e dos requisitos funcionais, procede-se aqui à concretização prática do sistema, abrangendo as dimensões eletrónica, mecânica e de software.

São detalhados os processos de desenvolvimento do PCB principal, a integração dos subsistemas de alimentação, aquisição de dados e comunicação automóvel, bem como a conceção do chassis dedicado e a implementação da aplicação de controlo. A descrição privilegia não apenas os aspetos construtivos, mas também as decisões de engenharia adotadas, os desafios técnicos identificados e as soluções implementadas para garantir robustez, estabilidade e escalabilidade da plataforma.

Este capítulo constitui, assim, a transição entre o domínio conceptual e a concretização física e lógica da solução proposta, evidenciando o rigor técnico subjacente ao desenvolvimento da eTestBox.

5.1. Implementação de Hardware

5.1.1. Integração do Display e Controlador

A integração HMI EDATEC HMI3010-101C constituiu o primeiro passo do processo de desenvolvimento da solução. Embora este equipamento seja desenvolvido para utilização com Raspberry Pi, o seu ecrã não funciona em modo *plug-and-play*, ao contrário do que seria expectável num sistema baseado nesta plataforma. Após a geração da imagem inicial do sistema operativo através do Raspberry Pi Imager, seguindo o procedimento padrão, revelou-se necessário realizar uma configuração adicional específica do fabricante.

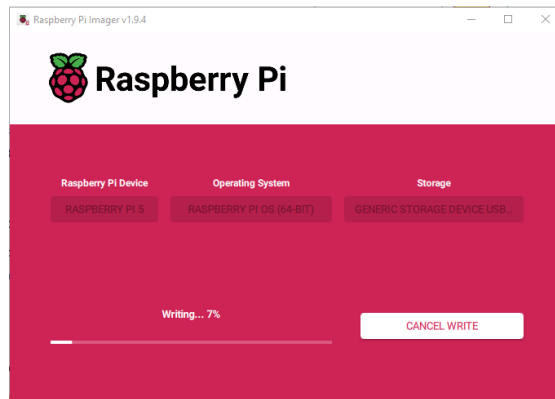


Figura 25 - Gravação da imagem base do Raspberry Pi OS para o cartão SD utilizando o Raspberry Pi Imager

Conforme documentado no guia oficial da EDATEC, o funcionamento correto do *display* e do painel tátil requer a instalação de um *firmware package* próprio (EDATEC, 2025). Para executar este processo foi indispensável ligar temporariamente um monitor externo via HDMI e utilizar um teclado USB, permitindo aceder à consola do Raspberry Pi e introduzir manualmente os comandos necessários à instalação do pacote de *firmware*.

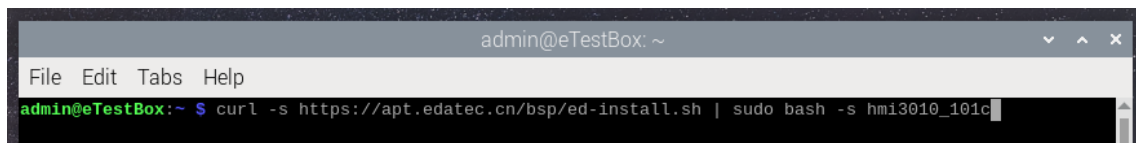


Figura 26 - Comando para obter e instalar o software package

Apenas após esta etapa o ecrã integrado do HMI ficou totalmente funcional, garantindo compatibilidade tanto ao nível do vídeo como do toque capacitivo.

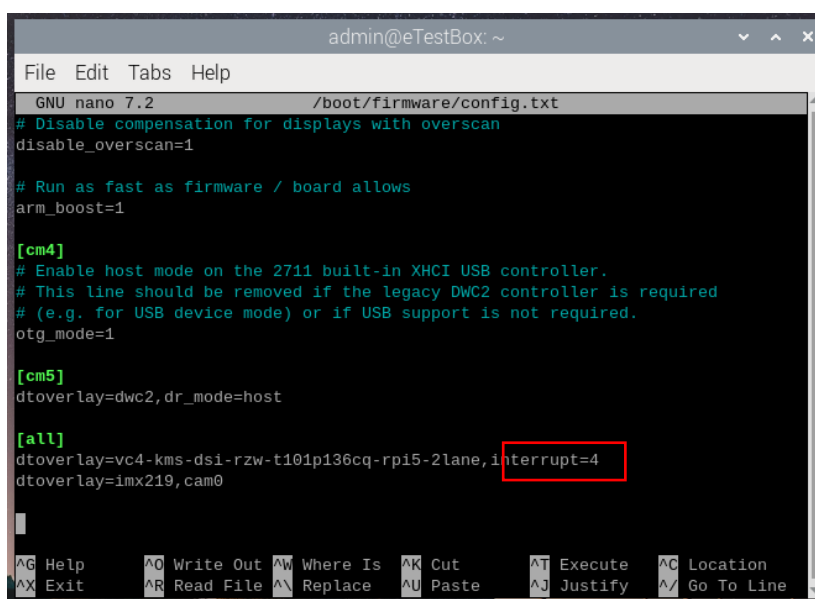


5.1.2. Integração da Fonte Programável

Após a configuração inicial do HMI, o passo seguinte consistiu na integração da fonte de alimentação programável **M5Stack PPS**, responsável por fornecer ao UUT uma tensão e corrente ajustáveis via comunicação **I²C**. Este módulo é ligado diretamente ao header de 40 pinos do Raspberry Pi, utilizando as linhas SDA/SCL para comunicação digital.

Contudo, logo após realizar as primeiras ligações, surgiu um problema de compatibilidade relacionado com o sistema de toque do HMI. O painel tátil utiliza, por defeito, o **GPIO2** como pino de interrupção do painel tátil, exatamente o mesmo pino utilizado pelo **SDA** do barramento **I²C**. Esta sobreposição impossibilitava o funcionamento simultâneo do painel tátil e da fonte programável.

A solução passou pela **alteração do pino de interrupção do painel tátil para o GPIO4**, libertando o GPIO2 para utilização exclusiva pelo **I²C**. Após alguma pesquisa, foi possível identificar o local onde esta configuração é definida: o ficheiro **/boot/firmware/config.txt**, onde se encontram as diretivas de inicialização e mapeamento de pinos do firmware do Raspberry Pi. A partir daí, a modificação do pino e posterior reinício do sistema permitiram que ambos os dispositivos, painel tátil e PPS, funcionassem corretamente sem conflitos de hardware.



```
admin@eTestBox: ~  
File Edit Tabs Help  
GNU nano 7.2 /boot/firmware/config.txt  
# Disable compensation for displays with overscan  
disable_overscan=1  
  
# Run as fast as firmware / board allows  
arm_boost=1  
  
[cm4]  
# Enable host mode on the 2711 built-in XHCI USB controller.  
# This line should be removed if the legacy DWC2 controller is required  
# (e.g. for USB device mode) or if USB support is not required.  
otg_mode=1  
  
[cm5]  
dtoverlay=dwc2,dr_mode=host  
  
[all]  
dtoverlay=vc4-kms-dsi-rzw-t101p136cq-rpi5-2lane,interrupt=4  
dtoverlay=imx219,cam0  
  
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^_/ Go To Line
```

Figura 28 - Configuração do pino de interrupção do painel tátil

Com o conflito do GPIO2 resolvido e a interface tátil plenamente funcional, foi possível avançar para a fase seguinte: o desenvolvimento de um programa de testes básicos em Python para controlo da fonte programável **M5Stack PPS**. O objetivo desta aplicação preliminar foi validar a comunicação I²C e garantir o correto funcionamento das operações essenciais do módulo, nomeadamente o **ajuste da tensão de saída, o limite de corrente, o comando de ativação/desativação da saída (output enable) e a leitura contínua dos valores de tensão e corrente medidos internamente** pelo dispositivo.

Em termos de integração física, foram realizadas as ligações necessárias entre o PPS e o Raspberry Pi através do header de 40 pinos. O pino **17 do M5Stack (SDA)** foi ligado ao **GPIO2** do Raspberry Pi e de forma análoga, o pino **18 (SCL)** foi conectado ao **GPIO3** do Raspberry Pi. Para garantir uma referência comum, foi estabelecida a ligação de massa entre o sistema, unindo o **pino 1 (GND) do M5Stack** ao **pino 6 (GND)** do Raspberry Pi. As alimentações necessárias para o funcionamento do módulo foram igualmente asseguradas: os **3.3 V do pino 12 do PPS** foram ligados ao **pino 1 (3V3)** do Raspberry Pi, enquanto os **5 V do pino 28** foram conectados ao **pino 2 (5V)** do Raspberry Pi. Estas ligações permitiram alimentar o módulo corretamente e estabelecer um canal de comunicação para os testes.

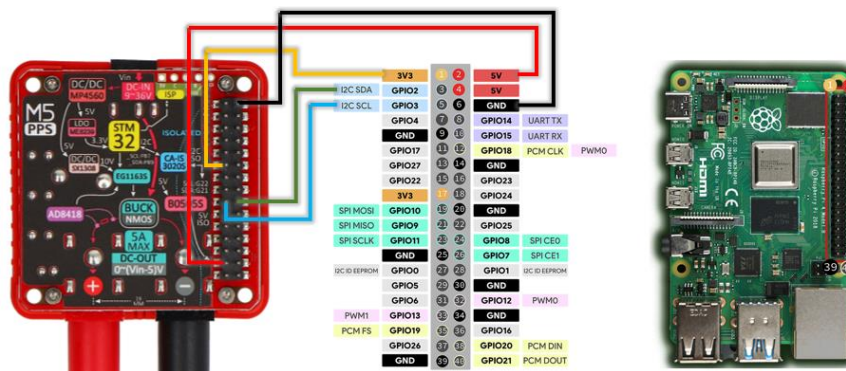


Figura 29 - Esquema ligação do M5Stack PPS ao Raspberry Pi

Com esta configuração, foi possível implementar e executar um conjunto de rotinas em Python destinadas a validar o controlo funcional da fonte, monitorizar o comportamento dinâmico da tensão e corrente e assegurar que o módulo PPS respondia de forma consistente aos comandos enviados pelo Raspberry Pi.

Na **Figura 30** pode observar-se um exemplo do teste funcional realizado ao módulo M5Stack PPS, no qual foi programada uma tensão de **5.40 V** através do software de validação desenvolvido em Python. Para confirmar a correta operação do módulo e a precisão da tensão fornecida, a saída foi simultaneamente medida com um **multímetro externo**, permitindo validar que o valor aplicado correspondia ao valor configurado via I²C. Este ensaio demonstrou não apenas o funcionamento adequado da fonte programável, mas também a fiabilidade das leituras disponibilizadas pelo módulo, constituindo um passo essencial antes da integração completa na eTestBox.



Figura 30 - Teste experimental do módulo M5Stack PPS

O controlo da fonte programável M5Stack PPS assenta na comunicação digital através do barramento **I²C**, onde todas as operações, tanto de escrita como de leitura, são realizadas por meio de **registos internos (register map)** definidos pelo fabricante. A Figura 31 apresenta o mapa de registos disponibilizado pelo módulo, no qual cada endereço corresponde a uma funcionalidade específica, como configuração da tensão de saída, limite de corrente, ativação da fonte, leitura de tensão real, leitura de corrente ou leitura da temperatura interna.

A escrita de valores, como a tensão ou a corrente desejada, é efetuada enviando um comando I²C seguido dos bytes correspondentes ao registo de destino (por exemplo, PSU_VOUT_SET para tensão ou PSU_IOUT_SET para corrente). De forma semelhante, a monitorização da fonte é realizada através da leitura sequencial de registos de *feedback*, tais como PSU_VOUT_READBACK e PSU_IOUT_READBACK, que devolvem a tensão e corrente realmente fornecidas pela saída. Estes registos atualizam-se periodicamente, permitindo ao Raspberry Pi obter, em tempo real, o estado da alimentação do UUT.

Este mecanismo baseado em registos torna o PPS altamente determinístico e fácil de integrar, uma vez que todo o comportamento da fonte é descrito de modo explícito no mapa de registos, garantindo uma comunicação fiável, estruturada e facilmente expansível. O uso deste protocolo simplifica também a implementação do software de controlo, permitindo que as funções de escrita e leitura sejam encapsuladas em rotinas Python usando a biblioteca **smbus2** que interage diretamente com os endereços I²C do módulo.

REG MAP (Addr:0x35)		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Power Control Setting	MODULE ENABLE	0x00				MODULE ENABLE											
	PSU VOUT SET ⁽¹⁾	0x10								PSU_VOUT_SET_1	PSU_VOUT_SET_2	PSU_VOUT_SET_3	PSU_VOUT_SET_4	PSU_IOUT_SET_1	PSU_IOUT_SET_2	PSU_IOUT_SET_3	PSU_IOUT_SET_4
Power Control Reading	PSU RUNNING MODE ⁽¹⁾	0x00				PSU RUNNING MODE											
	PSU DATA FLAG	0x00								PSU DATA FLAG							
	PSU_VOUT_READBACK PSU_IOUT_READBACK ⁽¹⁾	0x00								PSU_VOUT_READBACK_K_1	PSU_VOUT_READBACK_K_2	PSU_VOUT_READBACK_K_3	PSU_VOUT_READBACK_K_4	PSU_IOUT_READBACK_K_1	PSU_IOUT_READBACK_K_2	PSU_IOUT_READBACK_K_3	PSU_IOUT_READBACK_K_4
	PSU_TEMP_READBACK ⁽¹⁾	0x10	PSU_TEMP_READBACK_K_1	PSU_TEMP_READBACK_K_2	PSU_TEMP_READBACK_K_3	PSU_TEMP_READBACK_K_4	PSU_VIN_READBACK_1	PSU_VIN_READBACK_2	PSU_VIN_READBACK_3	PSU_VIN_READBACK_4							

Figura 31 - Mapa de endereços do protocolo I²C do M5Stack PPS

5.1.3. Integração da DAQ

Após a integração da fonte de alimentação foi realizada uma validação inicial do módulo de aquisição de dados MCC 128. Este teste preliminar teve como objetivo confirmar a correta comunicação com o Raspberry Pi e verificar a fiabilidade das medições analógicas, isolando eventuais problemas de hardware antes da integração final no sistema. Para esta validação foi utilizada a aplicação oficial disponibilizada pelo fabricante – “MCC DAQ HAT Manager” e “MCC 128 Control Panel” - garantindo que a verificação era efetuada com ferramentas certificadas e independentes do software desenvolvido no âmbito do projeto.

Numa primeira fase, através do DAQ HAT Manager, confirmou-se a deteção do dispositivo no barramento I²C, verificando o endereço atribuído e a correta identificação do módulo pelo sistema operativo. Esta etapa assegurou a integridade da comunicação ao nível físico e lógico. Posteriormente, com recurso ao “MCC 128 Control Panel”, foram testados os oito canais de entrada analógica. Os ensaios foram realizados em modo *Single-Ended*, com gama de entrada

configurada para 10 V. Foram aplicadas tensões conhecidas nos canais e os valores medidos comparados com leituras obtidas através de um multímetro externo de referência.

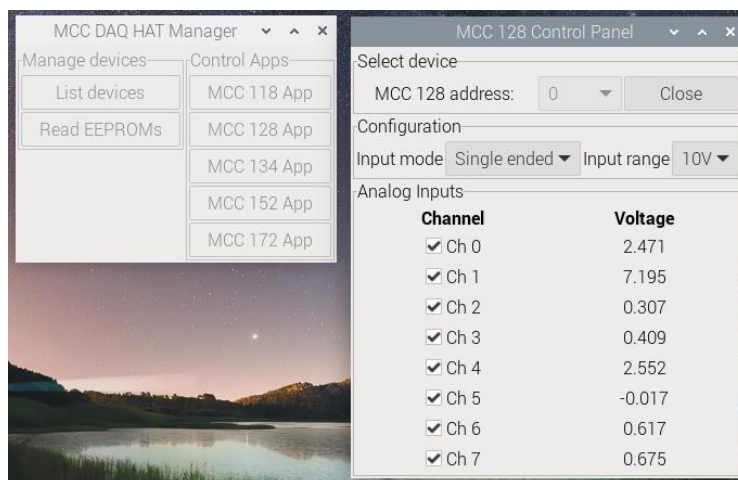


Figura 32 - MCC DAQ HAT Manager e Control Panel

5.1.4. Desenvolvimento do PCB Multicamada

Durante a fase inicial de validação do hardware, à medida que os diferentes módulos eram interligados para permitir os primeiros testes funcionais, tornou-se evidente que a abordagem baseada em cablagem direta, recorrendo a *jumpers*, *headers* e fios individuais, introduzia um conjunto de limitações significativas. Para além de resultar numa montagem visualmente complexa e de difícil manutenção, esta configuração revelava-se particularmente vulnerável a **interferências eletromagnéticas, ruído nos sinais analógicos e falhas intermitentes de ligação**, comprometendo a fiabilidade e a repetibilidade exigidas numa plataforma de teste automóvel.

Perante estes constrangimentos, tornou-se claro que a solução mais robusta e tecnicamente adequada passaria pelo desenvolvimento de uma **placa de circuito impresso (PCB) dedicada**, capaz de integrar de forma estruturada todos os módulos da eTestBox, incluindo a DAQ, o módulo PCAN, a fonte programável PPS, os conversores CC-CC e as interfaces com o UUT. Esta decisão permitiu não apenas uma organização significativamente mais limpa e profissional do sistema, mas também a redução do ruído elétrico, a minimização de percursos de massa, o controlo de impedâncias e a eliminação de variáveis indesejadas típicas de protótipos baseados em cablagem manual.

A primeira etapa do desenvolvimento do PCB correspondeu à **concepção do esquemático elétrico**, realizada com recurso ao software de desenho eletrónico **KiCad**. Esta ferramenta de código aberto permitiu a definição estruturada de todos os blocos funcionais do sistema, bem como a representação rigorosa das interligações elétricas entre os diferentes módulos.

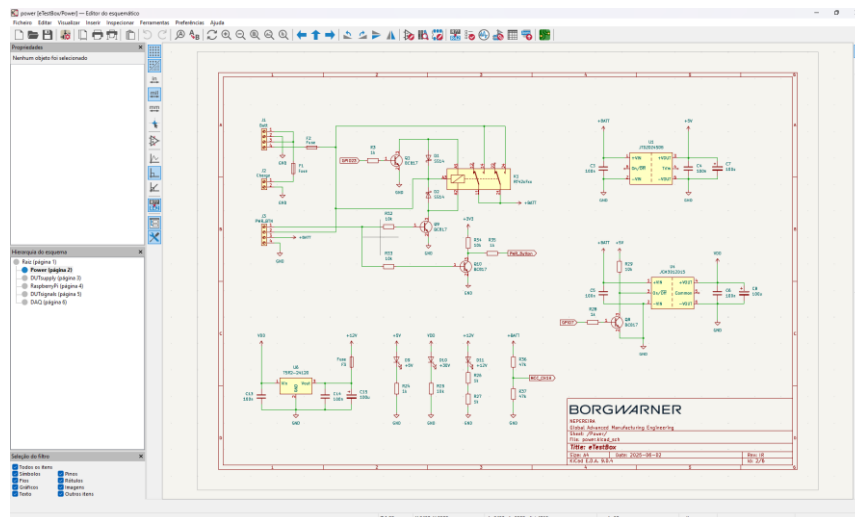


Figura 33 - Editor do esquemático no Kicad

Nesta fase, foi dada especial atenção à separação entre circuitos de potência, sinais digitais e circuitos analógicos sensíveis, bem como à correta implementação de proteções elétricas, isolamento galvânico, filtragem e referenciação de massas. A validação do esquemático constituiu um passo fundamental para garantir que o comportamento elétrico do sistema estivesse totalmente alinhado com os requisitos funcionais e de segurança definidos nos capítulos anteriores.

Concluída a fase de esquemático, procedeu-se ao *layout* e encaminhamento (*routing*) da placa, etapa particularmente crítica devido à elevada densidade de sinais provenientes do conector IDC de 40 pinos do Raspberry Pi, que inclui linhas de alimentação, sinais digitais de alta velocidade, interfaces I²C e SPI, entradas analógicas e múltiplos GPIOs. Por este motivo, foi adotada a concepção de um PCB multicamada de quatro camadas, uma vez que uma solução de duas camadas se revelou insuficiente para garantir um encaminhamento eficiente e tecnicamente adequado. A utilização de quatro camadas permitiu reservar planos dedicados para massa e alimentação, assegurando referências estáveis e de baixa impedância, enquanto as camadas restantes foram utilizadas para o

encaminhamento organizado dos sinais digitais e analógicos. Esta abordagem contribuiu de forma decisiva para a melhoria da integridade de sinal, redução de acoplamentos indesejados, mitigação de problemas de EMI e otimização do espaço disponível.

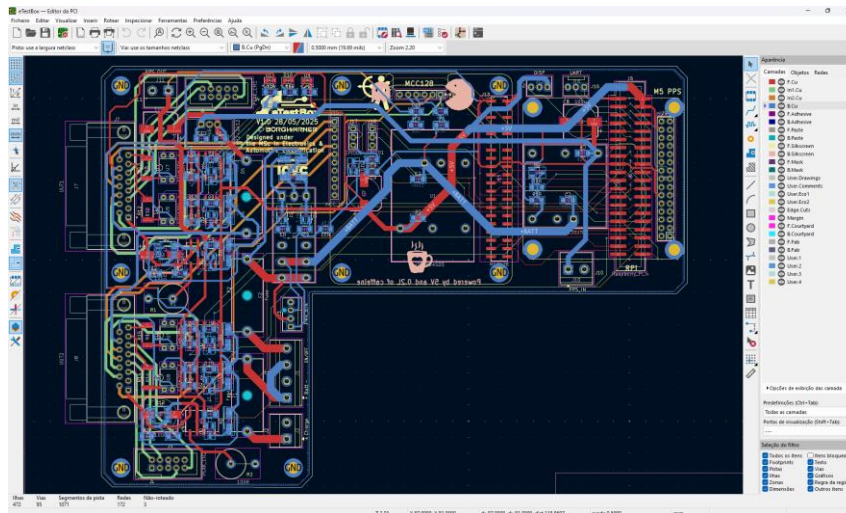


Figura 34 - Editor do layout do PCB

Após a conclusão do routing e a realização das verificações elétricas e de regras de projeto (Design Rule Check), foi efetuada a fase final de preparação para fabrico, que incluiu a geração dos ficheiros Gerber, ficheiros de furação (drill files) e documentação associada. Estes ficheiros representam a descrição completa do PCB necessária para o fabrico industrial da placa, assegurando que o layout concebido é reproduzido com exatidão pelo fabricante. Esta etapa marcou a transição do projeto do domínio do desenvolvimento para a sua materialização física, permitindo a produção de um PCB fiável, repetível e adequado ao contexto industrial.

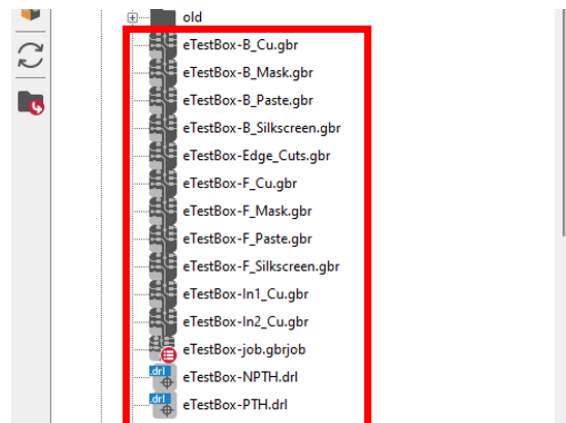


Figura 35 - Lista de ficheiros Gerber gerados

O resultado final é uma plataforma significativamente mais estável, robusta e elegante, onde cada componente se encontra devidamente alojado, com planos de alimentação dedicados, isolamento apropriado entre circuitos sensíveis e um encaminhamento otimizado dos sinais. A evolução para um PCB dedicado não representou apenas uma melhoria estética, mas sobretudo um passo decisivo para garantir o rigor técnico, a previsibilidade de comportamento e a robustez operacional exigidas a uma ferramenta industrial de teste como a eTestBox.

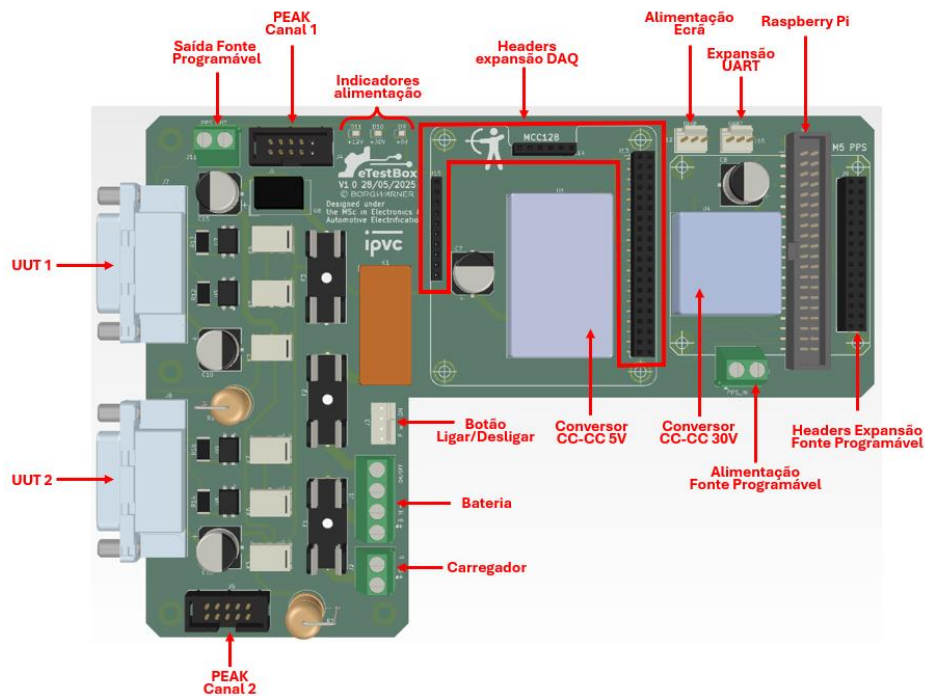


Figura 36 - Vista superior do PCB com identificação dos componentes

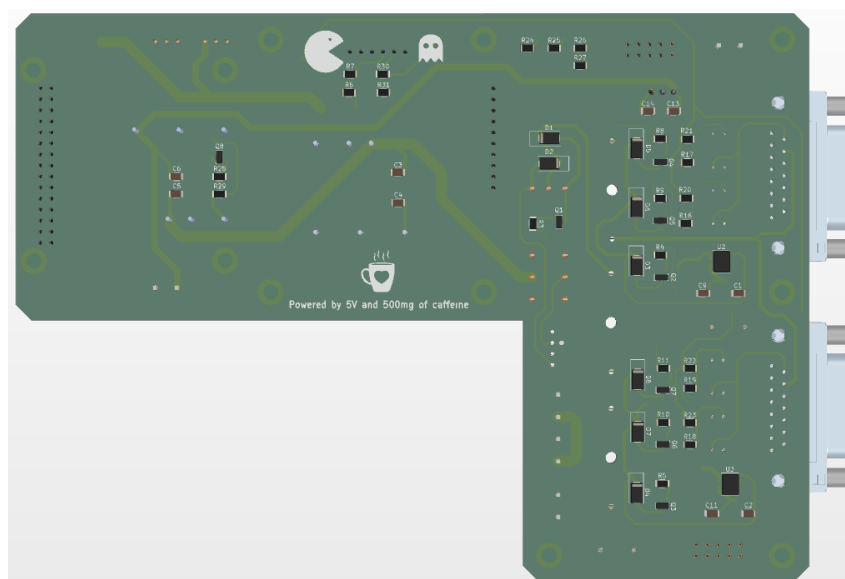


Figura 37 - Vista inferior do PCB

5.1.5. Desenvolvimento do *Chassis* Integrado

Numa fase inicial do projeto, estava previsto utilizar uma **mala de transporte standard** como base para alojar todo o hardware, incorporando suportes para o ecrã e para as fichas de ligação. No entanto, após uma análise extensa de catálogos de diversos fabricantes, rapidamente se verificou que nenhuma das soluções disponíveis era compatível com o **HMI escolhido**: ou a mala era demasiado pequena para integrar o ecrã, ou apresentava dimensões exageradamente grandes, tornando o conjunto volumoso e desproporcionado. Adicionalmente, a **altura interna** destas malas representava um claro desperdício de espaço, uma vez que o ecrã possui uma espessura reduzida e todo o hardware complementar ocupa relativamente pouco volume. Perante estas limitações, optou-se por abandonar a ideia de adaptar uma mala comercial e avançar para o desenvolvimento de um **chassis totalmente concebido de raiz**, especificamente desenhado para as necessidades da eTestBox e facilmente reproduzível através de **impressão 3D**.



Figura 38 - Mala rígida SERPAC SE230

O resultado deste processo foi a criação do **Chassis Integrado da eTestBox**, uma estrutura mecânica desenvolvida de forma dedicada para acomodar, de forma otimizada, todos os módulos do sistema, incluindo o HMI EDATEC, o Raspberry Pi, o PCB de quatro camadas, a bateria interna, o módulo M5Stack PPS e as interfaces automóveis. O desenho tridimensional deste chassis foi integralmente desenvolvido em **SolidWorks 2019**, permitindo uma modelação precisa das geometrias, tolerâncias e encaixes necessários para garantir a correta integração de todos os componentes. Este chassis integrado oferece não

apenas uma organização física limpa e eficiente, mas também vantagens estruturais significativas como: melhor controlo térmico, eliminação de cablagem desnecessária e uma apresentação final compacta e profissional, plenamente alinhada com as exigências de um ambiente industrial de teste automóvel.

A possibilidade de iterar rapidamente o design através de impressão 3D permitiu ainda otimizar geometrias, encaixes e pontos de fixação, garantindo um produto final funcional, estético e tecnicamente sólido.



Figura 39 - Vista exterior e interior do Chassis Integrado da eTestBox

A Figura 40 apresenta uma vista explodida do conjunto mecânico da eTestBox, evidenciando a organização interna dos principais subsistemas e a sua integração estrutural no chassis. É possível observar a disposição de todos os componentes, bem como das interfaces externas, botão de ligar/desligar e punho de transporte.

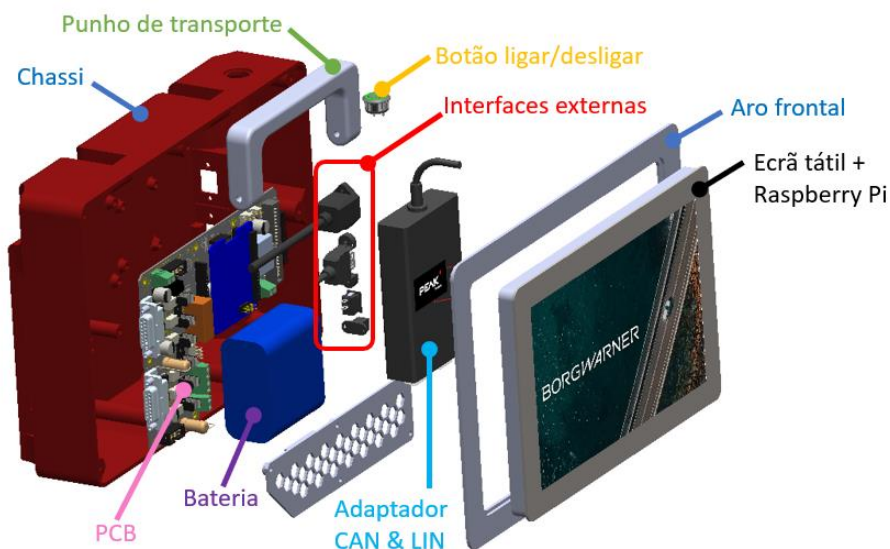


Figura 40 - Vista explodida e identificação dos elementos

5.1.6. Montagem do Protótipo

Após a conclusão do projeto eletrônico e geração dos ficheiros de fabrico, a placa de circuito impresso foi produzida por um fabricante externo, com base nos ficheiros **Gerber** previamente gerados no **KiCad**. Esta abordagem permitiu garantir **qualidade industrial** no fabrico da placa, nomeadamente ao nível de alinhamento das camadas, metalização das vias, máscara de solda e acabamento superficial.

Recebidas as placas, procedeu-se à montagem manual dos componentes eletrónicos, incluindo dispositivos SMD e componentes de potência. A soldadura foi realizada com recurso a ferro de soldar de ponta fina e controlo de temperatura adequado, assegurando precisão nas ligações e evitando pontes de solda ou aquecimento excessivo dos componentes sensíveis. Após a montagem de cada bloco funcional, foram realizados testes elétricos preliminares para verificar continuidade, ausência de curto-circuitos e correto funcionamento dos estágios de alimentação antes da integração completa no sistema.

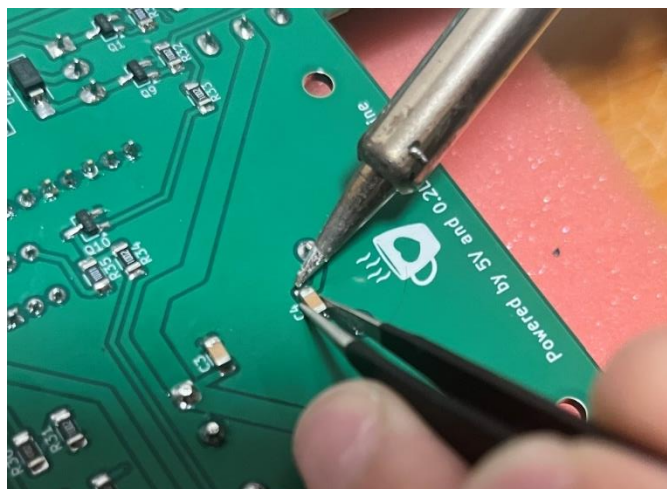


Figura 41 - Processo soldadura dos componentes no PCB

O chassis integrado foi produzido por **impressão 3D**, com base no modelo tridimensional previamente desenvolvido em SolidWorks. A utilização de manufatura aditiva permitiu iterar rapidamente o design, ajustar tolerâncias e validar o posicionamento físico dos componentes antes da montagem definitiva.

A Figura 42 ilustra as principais etapas do processo de fabrico e montagem do chassis da eTestBox. Inicialmente, é apresentada a impressão 3D das peças estruturais, evidenciando a geometria interna concebida para alojar os diferentes

módulos eletrônicos. Segue-se a inserção térmica das buchas metálicas roscadas nos pontos de fixação, realizada com ferro de soldar e controle de temperatura, garantindo alinhamento vertical e adequada resistência mecânica. Posteriormente, observa-se a fixação do PCB principal através de parafusos e espaçadores, assegurando estabilidade estrutural e isolamento mecânico. Por fim, é evidenciada a montagem dos conectores de painel (Ethernet, USB, alimentação e interruptor geral), que estabelecem a interface externa do equipamento. No seu conjunto, estas etapas demonstram a transição do modelo digital para um protótipo físico robusto e funcional.

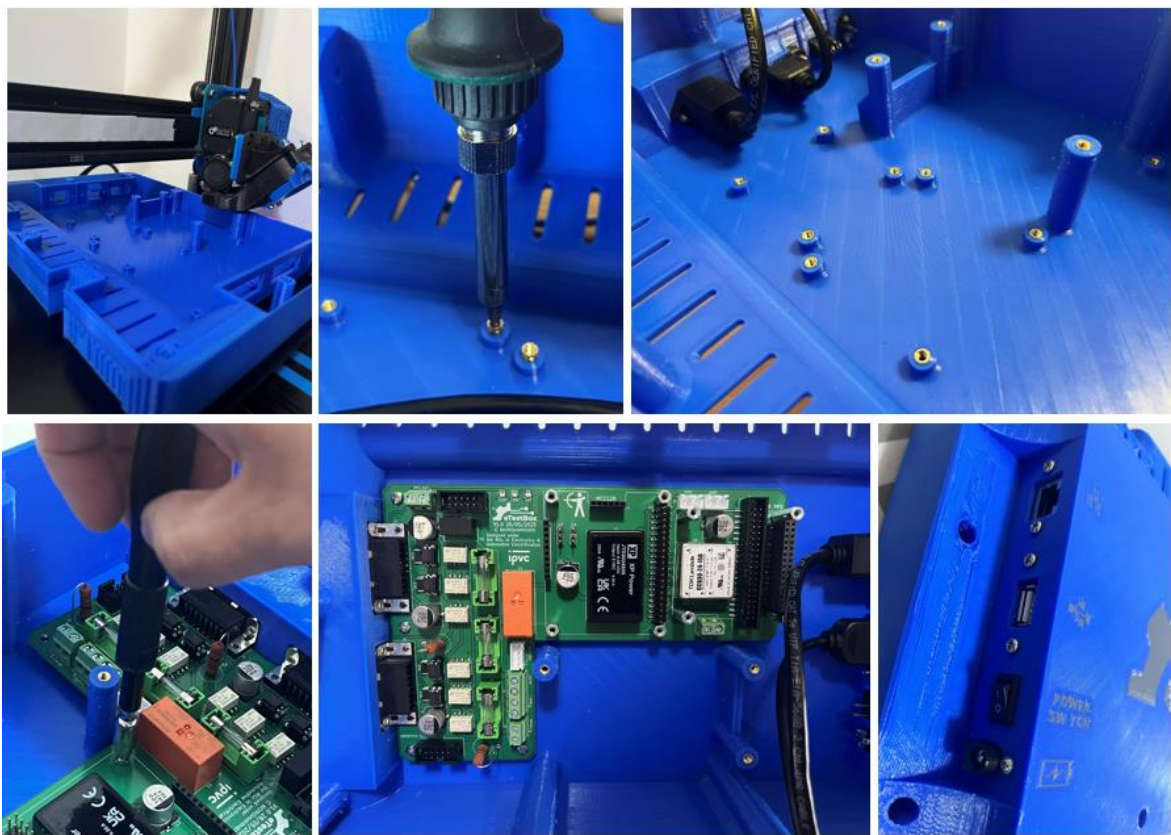


Figura 42 - Passos da montagem do chassis

A Figura 43 apresenta o interior da eTestBox numa fase avançada de integração mecânica e elétrica. No centro do conjunto encontra-se a placa principal (PCB verde), responsável pela interligação estrutural e elétrica de todos os módulos. Sobre esta estrutura estão montados os principais subsistemas: a DAQ MCC128, identificável pela PCB azul, dedicada à aquisição de sinais analógicos; e a fonte de alimentação programável M5Stack PPS, correspondente à PCB vermelha, responsável pelo fornecimento controlado de tensão e corrente ao UUT.

Na parte inferior observa-se a bateria interna, estrategicamente posicionada para estabilidade mecânica e distribuição de massa. À direita encontra-se o módulo PCAN-USB Pro, que assegura as comunicações CAN e LIN. O Raspberry Pi, por sua vez, não está montado diretamente nesta base, encontrando-se acoplado ao display frontal, sendo interligado ao PCB principal através de um cabo flat de 40 pinos (interface GPIO), visível na imagem. Esta disposição evidencia uma arquitetura modular, organizada e orientada para manutenção simplificada e redução de interferências entre domínios de potência, aquisição e comunicação.

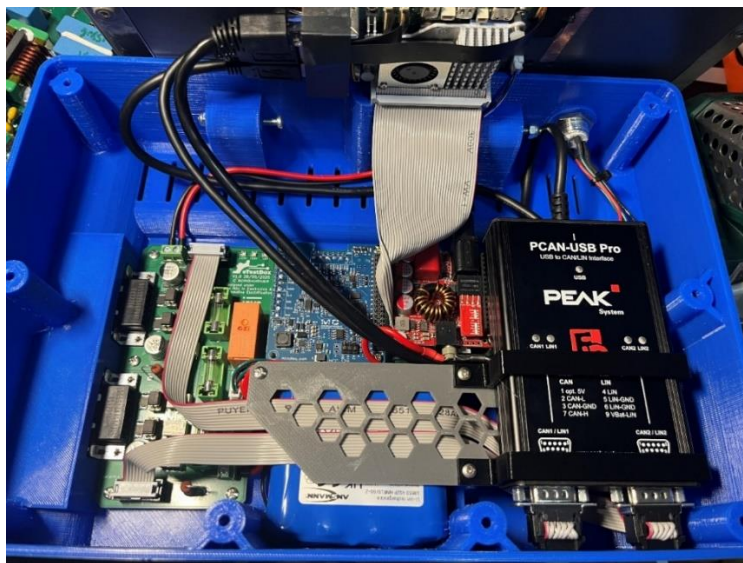


Figura 43 - Vista interna da eTestBox



Figura 44 - Vista exterior da eTestBox

5.2. Implementação do Software

5.2.1. Arquitetura da Aplicação

O desenvolvimento da camada de software da eTestBox foi estruturado segundo uma arquitetura modular, orientada à separação clara de responsabilidades entre interface gráfica, lógica de interação e comunicação com hardware. Esta abordagem permite reduzir o acoplamento entre componentes, facilitar a manutenção do código e garantir escalabilidade futura do sistema.

No núcleo da aplicação encontra-se o módulo `main_ui.py`, responsável pela definição da interface gráfica principal (QMainWindow), organização do layout e gestão dos diferentes separadores funcionais (*tabs*). Para além da componente visual, este módulo desempenha o papel de coordenação da aplicação, sendo responsável por:

- Inicializar o ambiente gráfico;
- Instanciar os diferentes módulos funcionais;
- Estabelecer a ligação entre a interface e as camadas de comunicação com hardware.

Cada funcionalidade da aplicação é implementada num módulo próprio, como “`power_tab.py`”, “`lin_tab.py`”, “`can_tab.py`”, entre outros. Estes ficheiros contêm:

- A definição do layout específico de cada separador;
- Os elementos de interação (botões, campos de entrada, indicadores);
- A lógica associada às ações do utilizador (eventos, sinais e *callbacks*).

Esta organização permite que cada funcionalidade evolua de forma independente, mantendo a coerência estrutural da aplicação.

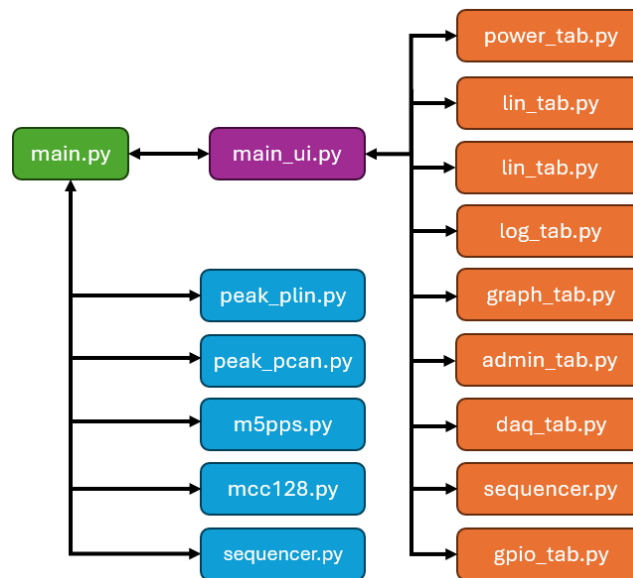


Figura 45 - Arquitetura do software

A camada de comunicação com hardware encontra-se isolada em módulos dedicados, nomeadamente:

- **peak_plin.py** - responsável pela comunicação LIN através da interface PLIN;
- **peak_pcan.py** - responsável pela comunicação CAN através da interface PCAN;
- **m5pps.py** - responsável pela comunicação I²C com a fonte de alimentação programável;
- **mcc128.py** - responsável pela aquisição de dados analógicos via DAQ MCC 128;
- **sequencer.py** - responsável pelo motor de execução do sequenciador.

Estes módulos constituem a camada de acesso a hardware, encapsulando a lógica de baixo nível, protocolos de comunicação e gestão de erros. Desta forma, a interface gráfica permanece desacoplada dos detalhes técnicos de implementação, interagindo com o hardware através de métodos bem definidos.

Na parte superior esquerda da interface encontram-se organizadas os separadores **de navegação**, que correspondem às principais funcionalidades disponíveis no sistema, tais como controlo da fonte de alimentação, comunicação LIN, registo de *logs*, visualização gráfica, administração, aquisição

MCC128 e gestão de GPIOs. Cada separador representa um módulo funcional independente. Ao selecionar um determinado separador, o conteúdo central da interface é automaticamente atualizado, apresentando exclusivamente os controlos, parâmetros e informações relevantes para essa funcionalidade específica, mantendo assim a organização e evitando sobrecarga visual.

A zona central constitui a área principal de interação, onde são exibidos os valores medidos em tempo real (por exemplo, tensão, corrente, modo de operação), bem como os comandos de controlo, como ajuste de tensão e corrente ou ativação/desativação da saída.

Na parte inferior da interface é apresentada informação de estado global do sistema. À esquerda é indicado o **nível de carga da bateria interna**, permitindo monitorizar a autonomia do equipamento. À direita é exibido o **estado da fonte de alimentação**, incluindo a tensão e corrente atualmente fornecidas ao UUT e a indicação do estado da saída (ON/OFF). Esta área funciona como uma barra de estado permanente, assegurando que o operador tem sempre visibilidade imediata das condições críticas de operação.



Figura 46 - Interface gráfica da eTestBox

A aplicação da eTestBox é iniciada automaticamente em **modo maximizado (full screen)**, ocupando a totalidade do ecrã e ocultando os elementos do sistema operativo subjacente. Esta abordagem foi adotada com o objetivo de transformar o conjunto Raspberry Pi + Linux numa plataforma dedicada, com comportamento

equivalente ao de um equipamento industrial autónomo, impedindo o acesso inadvertido do operador ao ambiente de trabalho do sistema operativo.

Desta forma, durante a operação normal, o utilizador interage exclusivamente com a interface da eTestBox, não sendo visíveis barras de tarefas, menus do sistema ou outras aplicações externas. O acesso ao sistema operativo apenas é possível mediante autenticação no **separador Admin**, onde a introdução da palavra-passe habilita funcionalidades avançadas. Esta estratégia aumenta a segurança operacional, reduz o risco de alterações acidentais ao sistema e garante maior robustez em ambiente industrial.

Para garantir um arranque automático da aplicação aquando da inicialização da eTestBox, foi configurado o sistema operativo do Raspberry Pi de modo a executar automaticamente o programa principal após o *boot*. Desta forma, a interface gráfica é carregada sem necessidade de intervenção do utilizador, assegurando um comportamento semelhante ao de um equipamento industrial dedicado.

No sistema Linux utilizado (Raspberry Pi OS), o arranque automático de aplicações gráficas pode ser configurado através do mecanismo de *autostart* do ambiente gráfico (LXDE). Para esse efeito, foi criado um ficheiro do tipo *.desktop* no diretório:

/home/admin/.config/autostart/main.desktop

O conteúdo apresentado na Figura 47 define uma entrada de execução automática com os seguintes parâmetros:

[Desktop Entry] - identifica uma entrada válida de aplicação no ambiente gráfico.

Type=Application - especifica que se trata de uma aplicação executável.

Name=eTestBox - nome interno da aplicação.

Exec=python3 /home/admin/etestbox/main.py - *script* a ser executado automaticamente.

StartupNotify=false - desativa notificações visuais de arranque.

Terminal=false - impede a abertura de uma janela de terminal associada ao processo.

Desta forma, sempre que o sistema é iniciado e o utilizador efetua login no ambiente gráfico, o *script* main.py, correspondente à aplicação principal da eTestBox, é automaticamente executado, garantindo rapidez de operação, redução de erros humanos e maior robustez no contexto industrial.

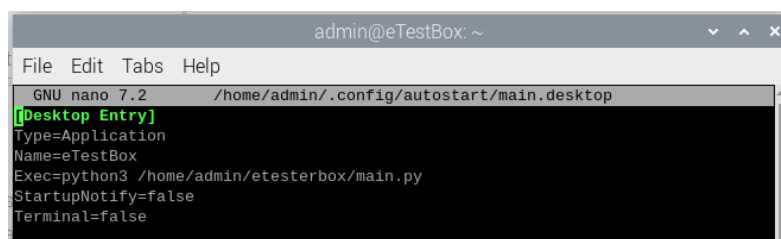


Figura 47 - Definição de arranque automático da aplicação eTestBox

5.2.2. Módulo de Controlo da Alimentação

O **separador “Power Supply”** constitui o módulo responsável pelo controlo e monitorização da fonte de alimentação programável integrada na eTestBox. Esta interface permite configurar os parâmetros elétricos fornecidos ao UUT e acompanhar, em tempo real, o comportamento da saída.

Na zona superior são apresentados os valores medidos instantaneamente de tensão (*Voltage*) e corrente (*Current*), bem como o estado da saída (*Output ON/OFF*) e o modo de operação da fonte, tipicamente CV (*Constant Voltage*), indicando que a fonte está a regular em tensão constante ou CC (*Constant Current*) caso esteja a regular corrente constante.

Na secção inferior encontram-se os valores de referência programados para tensão e corrente. Estes parâmetros podem ser ajustados através dos botões incrementais “+” e “-”, permitindo uma configuração precisa e controlada das condições de teste. O botão principal, localizado na parte inferior da janela, possibilita ativar ou desativar a saída da fonte, garantindo controlo direto sobre a energização do UUT.

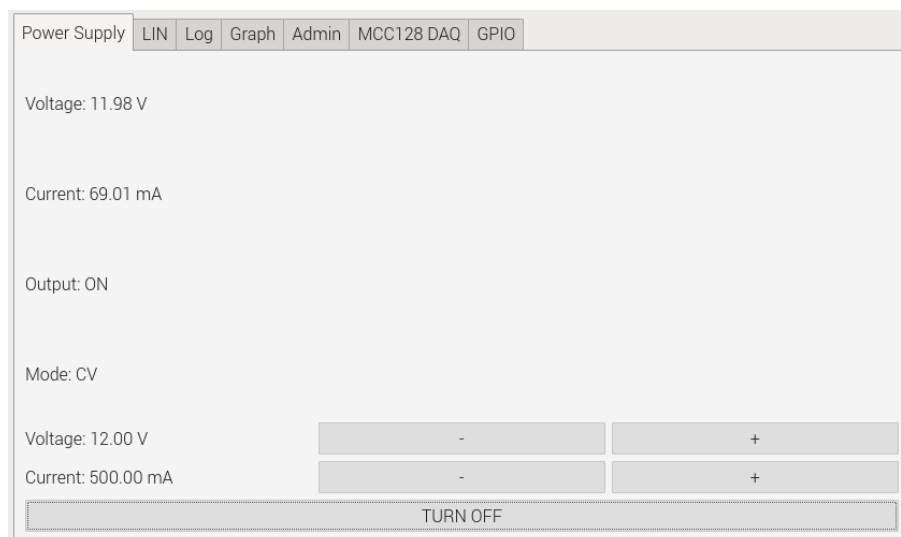


Figura 48 - Separador da fonte de alimentação

5.2.3. Módulos de Comunicação (LIN e CAN)

O **separador “LIN”** destina-se à configuração, execução e monitorização da comunicação LIN entre a eTestBox e o UUT. Esta interface permite carregar a descrição da rede através de um ficheiro LDF, seleccionar a interface física de comunicação e controlar a execução das tabelas de agendamento (*Schedule Tables*).

Na zona superior encontra-se o botão “Selecionar ficheiro LDF”, que permite importar o ficheiro de descrição da rede LIN. Após carregamento, o nome do ficheiro é apresentado, garantindo rastreabilidade do projeto ativo. De seguida, é possível seleccionar a porta LIN disponível no sistema (por exemplo, /dev/plin0), correspondente ao canal físico do adaptador PEAK configurado no Linux.

A secção “Schedule Table” permite escolher a tabela de agendamento definida no LDF (ex.: RUN_MAIN), que determina a sequência e periodicidade das *frames* transmitidas pelo *master*. Os botões “Start” e “Stop”, bem como o comando “Iniciar Schedule”, possibilitam controlar a execução da comunicação LIN em tempo real.

Na parte inferior encontra-se a área de monitorização das últimas *frames* LIN, onde são registadas as mensagens transmitidas (Tx) e recebidas (Rx), incluindo *timestamp*, identificador da *frame* e *payload* em formato hexadecimal. Está também disponível um campo de filtragem por ID ou conteúdo, facilitando a análise de tráfego. Por fim, os botões “Guardar Log” e “Limpar Log” permitem

respetivamente exportar os registos para ficheiro ou limpar a janela de monitorização.

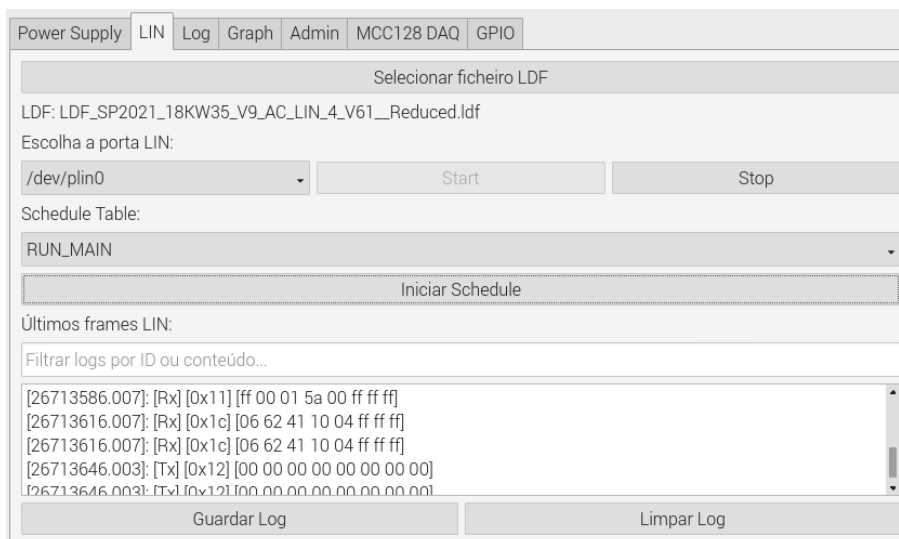


Figura 49 - Separador da comunicação LIN

O **separador “CAN”** destina-se à configuração, execução e monitorização da comunicação CAN entre a eTestBox e o UUT. Tal como no módulo LIN, esta interface permite carregar a descrição da rede, seleccionar a interface física de comunicação e acompanhar em tempo real o tráfego no barramento. Na zona superior encontra-se o botão “Selecionar ficheiro DBC”, que permite importar o ficheiro de descrição da rede CAN (DBC), de forma análoga ao comportamento do módulo LIN com **ficheiros LDF**.

Segue-se a seleção da interface CAN disponível no sistema (can0 ou can1), correspondente à interface SocketCAN configurada no Linux. É também possível definir o *bitrate* da comunicação (ex.: 500 kbit/s). Os botões “Start” e “Stop” permitem iniciar e interromper a monitorização do barramento, estabelecendo ou terminando a ligação ao canal físico.

Na parte inferior encontra-se a área de monitorização das últimas *frames* CAN, onde são registadas as mensagens transmitidas (Tx) e recebidas (Rx), incluindo *timestamp*, identificador (ID), comprimento da trama (DLC) e *payload* em formato hexadecimal. Quando um ficheiro DBC está carregado, os sinais são automaticamente decodificados e apresentados juntamente com os respetivos valores físicos, permitindo uma análise semântica das mensagens. Tal como no

módulo LIN, existe um campo de filtragem por ID ou conteúdo, facilitando a análise de tráfego em redes com elevado volume de mensagens. Os botões “Guardar Log” e “Limpar Log” permitem, respetivamente, exportar os registos para ficheiro ou limpar a janela de monitorização.

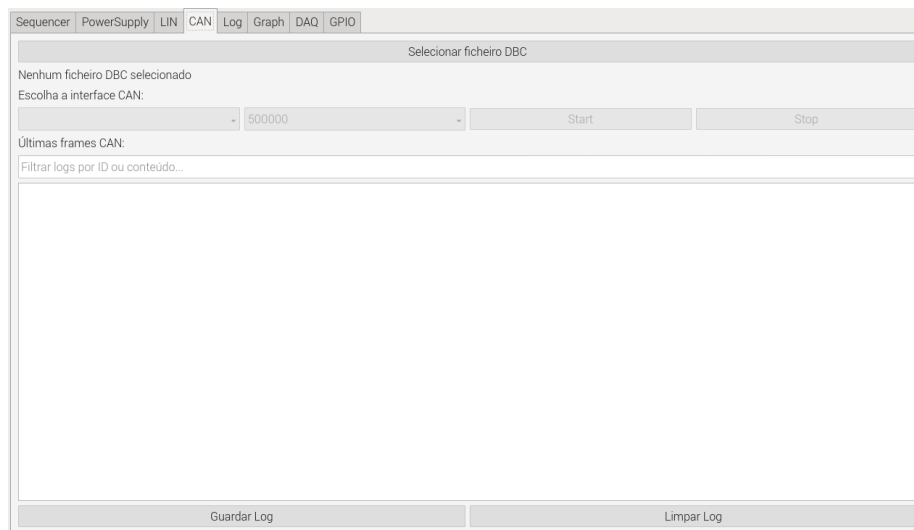


Figura 50 - Separador da comunicação CAN

5.2.4. Módulos de Monitorização

O **separador “Log”** destina-se à visualização detalhada dos registos internos do sistema, permitindo acompanhar o estado de inicialização, configuração e execução dos diferentes módulos da eTestBox. Nesta janela são apresentados, de forma cronológica e com carimbo temporal (*timestamp*), os eventos relevantes ocorridos durante a operação do sistema. Esta informação é particularmente útil para diagnóstico técnico, validação de configuração e deteção de eventuais falhas de comunicação ou erros de inicialização, em especial no contexto de desenvolvimento e comissionamento.

Na parte superior encontram-se os botões “Guardar Log” e “Limpar Log”, que permitem respetivamente exportar os registos para análise posterior ou limpar a janela para facilitar a monitorização de novos eventos.

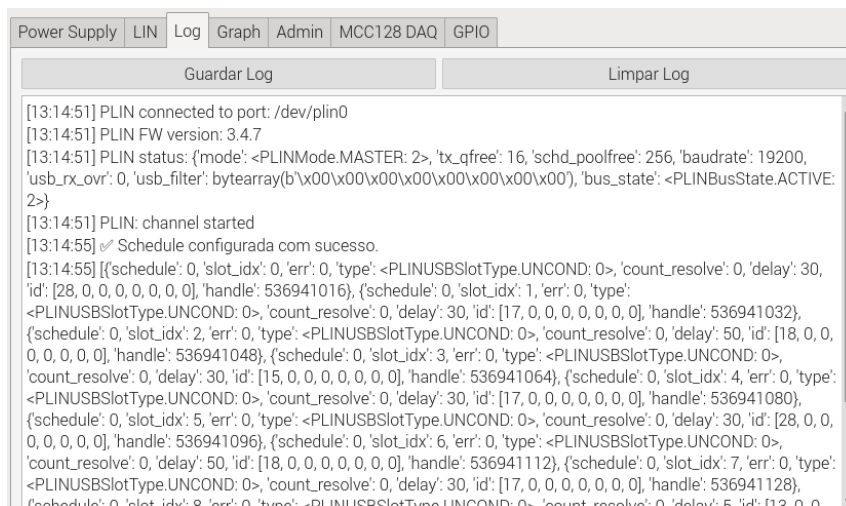


Figura 51 - Separador de registro

O separador “**Graph**” destina-se à **visualização gráfica em tempo real** dos sinais monitorizados pelo sistema, permitindo uma análise temporal detalhada do comportamento elétrico e funcional do UUT.

Na zona lateral esquerda encontra-se a lista de sinais disponíveis para representação gráfica. Estes sinais podem incluir variáveis provenientes da comunicação LIN (extraídas dinamicamente do ficheiro LDF), bem como medições analógicas adquiridas pela DAQ, como tensão de alimentação (KL30_Voltage) ou correntes dos UUTs. A seleção de um determinado sinal ativa a sua visualização na área principal do gráfico.

A zona central corresponde ao painel de representação temporal, onde o sinal selecionado é apresentado em função do tempo. O eixo horizontal representa a evolução temporal (*timestamp*) e o eixo vertical o valor medido da variável. Esta visualização permite identificar variações, oscilações, picos transitórios ou comportamentos anómalos durante o ensaio.

Na parte inferior encontram-se as opções “Exportar CSV”, que permite guardar os dados do gráfico para análise posterior, e “Limpar Gráfico”, que reinicia a visualização para uma nova aquisição.

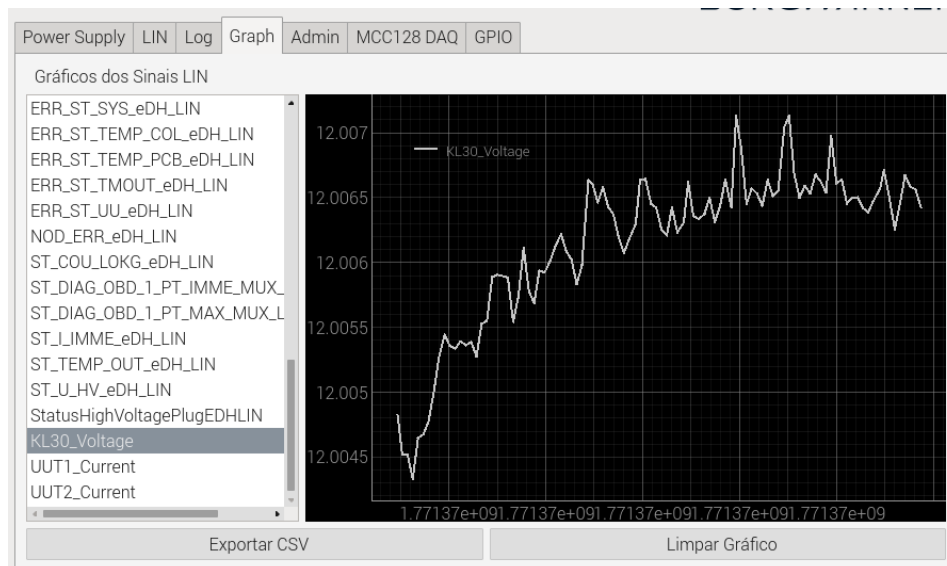


Figura 52 - Separador de visualização de gráficos

5.2.5. Módulo de Conversão Analógico-Digital

O **separador “DAQ”** destina-se à monitorização e configuração dos **canais analógicos** do módulo de aquisição de dados MCC128 integrado na eTestBox. Esta interface permite visualizar, em tempo real, os valores medidos nos diferentes canais e definir o modo de aquisição para os canais configuráveis.

Na parte superior são apresentados os canais dedicados à medição de grandezas críticas do sistema, nomeadamente as correntes associadas às linhas KL30_1 e KL30_2 (CH0H e CH0L), bem como as tensões da bateria interna (CH1H) e da fonte de alimentação programável (CH1L). Os valores são exibidos continuamente, permitindo acompanhar o comportamento elétrico do sistema durante os ensaios.

Os canais adicionais (CH2 e CH3) podem ser configurados através de um menu de seleção que permite escolher o modo de medição, como *Single-Ended* ou diferencial, consoante a natureza do sinal a adquirir. Esta flexibilidade possibilita adaptar a aquisição a diferentes cenários de teste, garantindo compatibilidade com sensores, sinais de baixa amplitude ou medições referenciadas a massa.

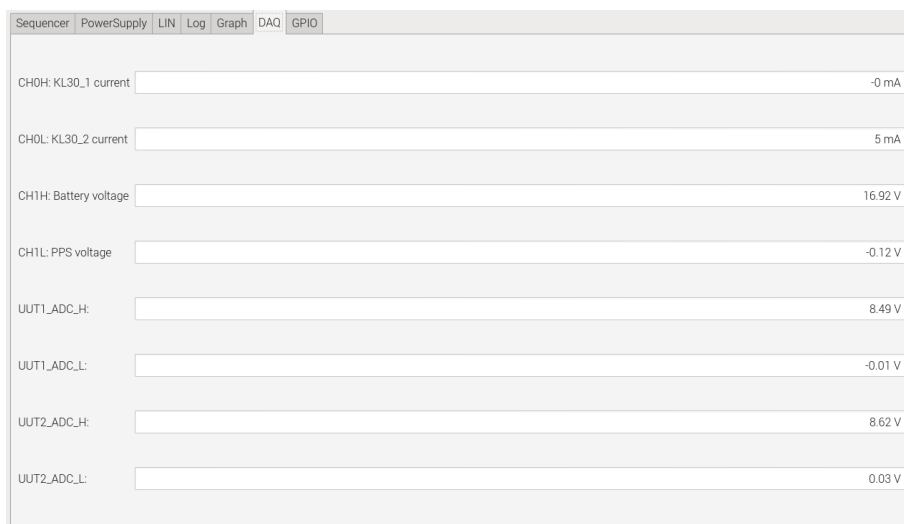


Figura 53 - Separador do conversor analógico-digital

5.2.6. Módulo de Entradas e Saídas Digitais

O **separador “GPIO”** permite a monitorização e controlo direto das **entradas e saídas digitais** do Raspberry Pi utilizadas na eTestBox. Este módulo funciona como uma interface de baixo nível para atuação de relés e leitura de estados digitais associados ao UUT e aos circuitos internos do sistema.

Na secção “Saídas”, encontram-se listados os relés e sinais de controlo comandados por GPIOs específicos (por exemplo, KL30_1 Relé, KL30_2 Relé, Enable CC e relés associados ao UUT1 e UUT2). A ativação ou desativação destes elementos é realizada através de caixas de seleção, permitindo energizar ou desligar cargas, comutar alimentação ou ativar funções auxiliares do sistema.

Na secção “Entradas”, são apresentados os estados lógicos das entradas digitais opto-isoladas associadas aos UUTs, bem como o estado do botão de “Power-On”. Cada entrada é indicada em tempo real com o respetivo nível lógico (“HIGH”/“LOW”), permitindo validar sinais externos, estados de sensores ou confirmações de atuação.

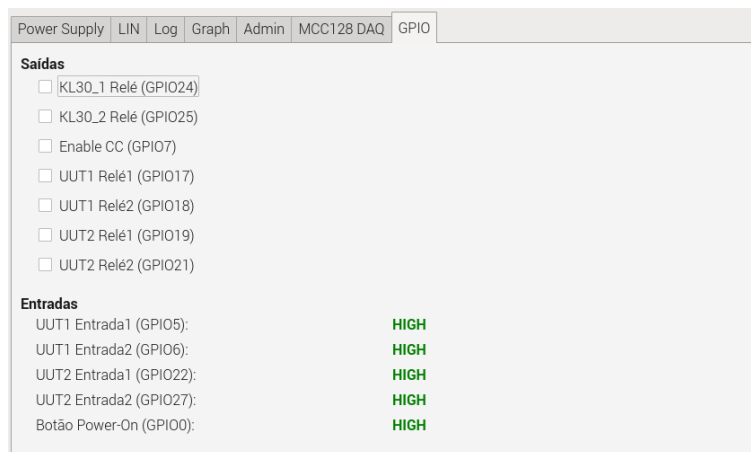


Figura 54 - Separador de entradas e saídas

5.2.7. Sequenciador e Motor de Execução

O **separador “Sequenciador”** constitui o núcleo de automação da eTestBox, permitindo a execução estruturada de planos de teste previamente definidos (por exemplo, a partir de ficheiros Excel com **test tables**).

Na parte superior encontram-se os controlos principais de execução, nomeadamente os botões “Executar sequência” e “Abortar”, bem como o estado corrente do sistema (ex.: “Pronto para executar”). Durante a execução, o sistema apresenta de forma destacada o resultado global do ensaio, recorrendo a um indicador visual de alto contraste (ex.: “RESULTADO FINAL: FAIL” a vermelho), permitindo ao operador identificar imediatamente o desfecho do teste.

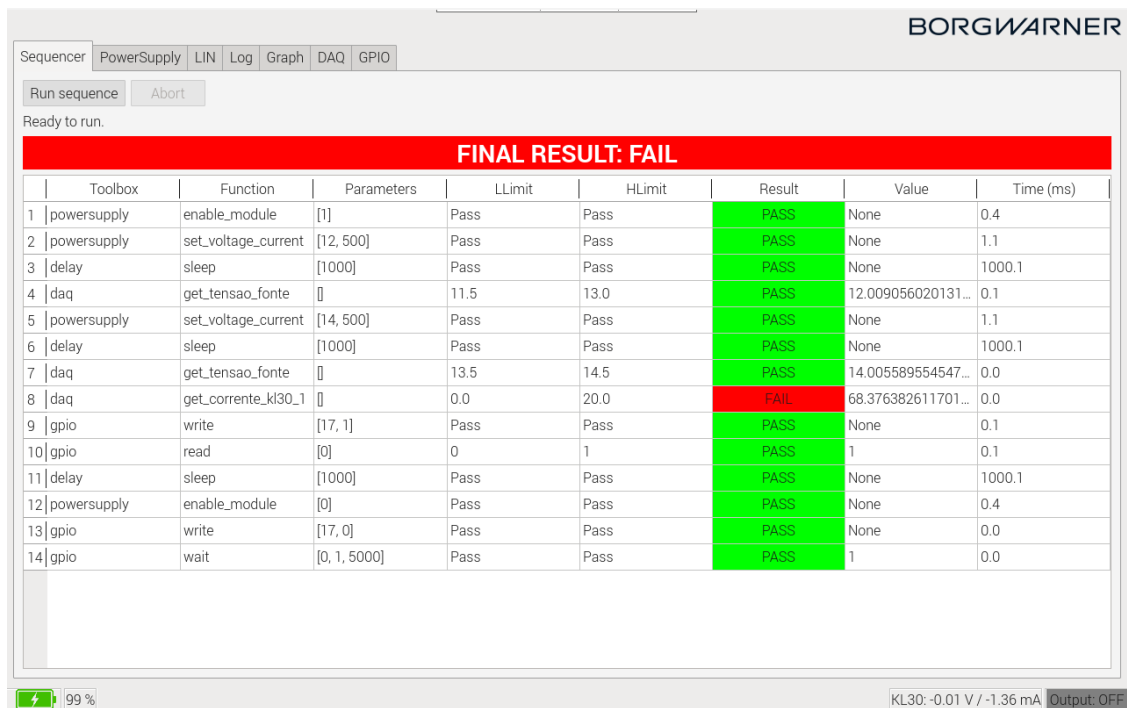
A zona central é composta por uma tabela estruturada de passos de teste, onde cada linha corresponde a uma instrução executada pelo sequenciador. Para cada passo são apresentados:

- **Toolbox** - módulo utilizado (ex.: powersupply, daq, gpio, delay);
- **Função** - ação executada (ex.: set_voltage_current, get_tensao_fonte, write, read, sleep);
- **Parâmetros** - argumentos associados à função;
- **LLimit / HLimit** - limites inferior e superior definidos para validação;
- **Resultado** - estado individual do passo (PASS/FAIL), com codificação cromática;
- **Retorno** - valor efetivamente medido ou retornado pela função;

- **Tempo (ms)** - duração de execução do passo.

Esta abordagem permite não apenas a execução automática de sequências complexas (ativação de fonte, carregamento de LDF, arranque de *schedule tables*, leitura de sinais, atuação de GPIOs, medições DAQ, etc.), mas também a rastreabilidade completa de cada etapa, incluindo validação por limites e análise temporal.

A aba do sequenciador transforma assim a eTestBox numa plataforma de teste verdadeiramente programável, capaz de executar ciclos automáticos reproduzíveis, com avaliação objetiva de conformidade e registo detalhado de resultados.



The screenshot shows the BORGWARNER Sequencer interface. At the top, there are tabs for Sequencer, PowerSupply, LIN, Log, Graph, DAQ, and GPIO. Below the tabs, there are buttons for 'Run sequence' and 'Abort'. The status 'Ready to run.' is displayed. A prominent red banner across the middle of the interface reads 'FINAL RESULT: FAIL'. Below this banner is a table with 14 rows of test steps. The table has columns for Step, Toolbox, Function, Parameters, LLimit, HLimit, Result, Value, and Time (ms). Step 8, which involves a DAQ function 'get_corrente_kl30_1', has a 'FAIL' result. The bottom status bar shows a battery icon at 99%, and text indicating 'KL30: -0.01 V / -1.36 mA' and 'Output: OFF'.

	Toolbox	Function	Parameters	LLimit	HLimit	Result	Value	Time (ms)
1	powersupply	enable_module	[1]	Pass	Pass	PASS	None	0.4
2	powersupply	set_voltage_current	[12, 500]	Pass	Pass	PASS	None	1.1
3	delay	sleep	[1000]	Pass	Pass	PASS	None	1000.1
4	daq	get_tensao_fonte	[]	11.5	13.0	PASS	12.009056020131...	0.1
5	powersupply	set_voltage_current	[14, 500]	Pass	Pass	PASS	None	1.1
6	delay	sleep	[1000]	Pass	Pass	PASS	None	1000.1
7	daq	get_tensao_fonte	[]	13.5	14.5	PASS	14.005589554547...	0.0
8	daq	get_corrente_kl30_1	[]	0.0	20.0	FAIL	68.376382611701...	0.0
9	gpio	write	[17, 1]	Pass	Pass	PASS	None	0.1
10	gpio	read	[0]	0	1	PASS	1	0.1
11	delay	sleep	[1000]	Pass	Pass	PASS	None	1000.1
12	powersupply	enable_module	[0]	Pass	Pass	PASS	None	0.4
13	gpio	write	[17, 0]	Pass	Pass	PASS	None	0.0
14	gpio	wait	[0, 1, 5000]	Pass	Pass	PASS	1	0.0

Figura 55 - Separador do sequenciador

O motor de execução (**Sequencer Engine**) constitui o núcleo lógico de automação da eTestBox, sendo responsável pela execução determinística e estruturada de planos de teste previamente definidos sob a forma de tabelas parametrizadas (*test tables*). Este mecanismo permite coordenar, de forma integrada, todos os subsistemas da plataforma - fonte de alimentação programável, aquisição de dados, comunicação *LIN/CAN* e controlo de *GPIOs* - assegurando coerência temporal, validação automática de critérios de aceitação e rastreabilidade integral dos resultados.

A arquitetura do motor de execução baseia-se na interpretação sequencial de uma tabela estruturada, tipicamente importada a partir de ficheiro Excel, onde cada linha representa um passo elementar de teste. Cada passo é descrito por um conjunto de atributos fundamentais: o módulo alvo (*toolbox*), a função a executar, os parâmetros associados, os limites inferior e superior de validação (quando aplicável) e, opcionalmente, um tempo máximo de espera (*timeout*). Esta abordagem promove a separação entre lógica de teste e implementação de baixo nível, permitindo que o comportamento do sistema seja reconfigurado sem alteração do código-fonte.

toolbox	function	parameters	LLimit	HLimit
powersupply	enable_module	[1]	Pass	Pass
powersupply	set_voltage_current	[12;500]	Pass	Pass
delay	sleep	[1000]	Pass	Pass
daq	get_tensao_fonte	[]	11.5	13.0
daq	get_corrente_kl30_1	[]	0.0	20.0
gpio	write	[17;1]	Pass	Pass
gpio	read	[0]	0	1
delay	sleep	[1000]	Pass	Pass
gpio	wait	[0;1;5000]	Pass	Pass

Tabela 8 - Exemplo de uma tabela de importação no sequenciador

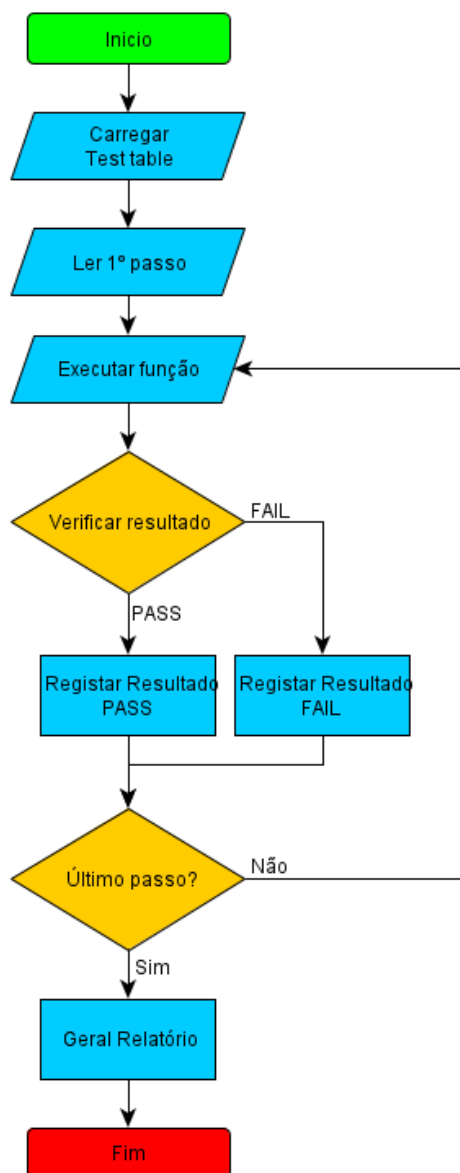
Durante a execução, o motor percorre sequencialmente cada linha da tabela, realizando um processo composto por quatro etapas principais: resolução dinâmica da função, execução da instrução, recolha do valor de retorno e avaliação dos critérios de aceitação. A resolução dinâmica consiste no mapeamento do identificador textual do módulo e da função para o método concreto implementado na respetiva camada de acesso ao hardware (por exemplo, m5pps.py, peak_plin.py, mcc128.py ou gestor de GPIOs). Desta forma, o sequenciador atua como camada intermédia de orquestração, não estabelecendo comunicação direta com o hardware, mas sim invocando métodos especializados encapsulados.

A especificação das funções suportadas pelo motor de execução, incluindo parâmetros, tipos de dados e valores de retorno, encontra-se detalhada no Apêndice D – Especificação das Funções do Sequencer Engine

Após a execução de cada função, o valor devolvido é comparado com os limites definidos na tabela de teste, quando aplicável. A condição de aceitação é formalmente expressa por:

$$LLimit \leq Valor_{retornado} \leq HLimit$$

Caso a condição seja satisfeita, o passo é marcado como **PASS**; caso contrário, como **FAIL**. O motor mantém um estado global acumulativo, sendo que a ocorrência de qualquer falha altera o estado final da sequência. Dependendo da configuração, a execução pode continuar até ao final da tabela ou ser interrompida imediatamente após a primeira falha.



Durante todo o processo são registados o resultado individual, o valor retornado, o tempo de execução e a ordem sequencial do passo, garantindo rastreabilidade completa do ensaio. Esta abordagem assegura determinismo, repetibilidade e transparência na avaliação.

Do ponto de vista arquitetural, o motor de execução materializa uma camada de abstração de alto nível que desacopla completamente a definição dos testes da implementação concreta dos dispositivos físicos. Tal característica confere à eTestBox elevada flexibilidade, permitindo a adaptação a novos projetos automóveis, a inclusão de futuras funcionalidades e a integração com sistemas externos.

Em síntese, o sequenciador não representa apenas um mecanismo de execução sequencial, mas sim um sistema formal de coordenação de testes industriais, dotado de capacidade de

Figura 56 - Diagrama blocos do sequenciador

validação automática, rastreabilidade integral e extensibilidade arquitetural.

5.2.8. Modo Administrador

O **separador “Admin”** corresponde à área de configuração avançada da eTestBox e destina-se exclusivamente a utilizadores com privilégios administrativos. Por motivos de segurança e integridade operacional, esta aba não se encontra visível por defeito, sendo desbloqueada apenas após autenticação válida. O acesso ao modo administrador é realizado através da interação com o logótipo da BorgWarner presente na interface principal; ao clicar sobre o mesmo, é apresentado um diálogo de autenticação onde deve ser introduzida a palavra-passe de administrador. Após validação bem-sucedida, a aba “Admin” passa a integrar o conjunto de separadores disponíveis na aplicação.

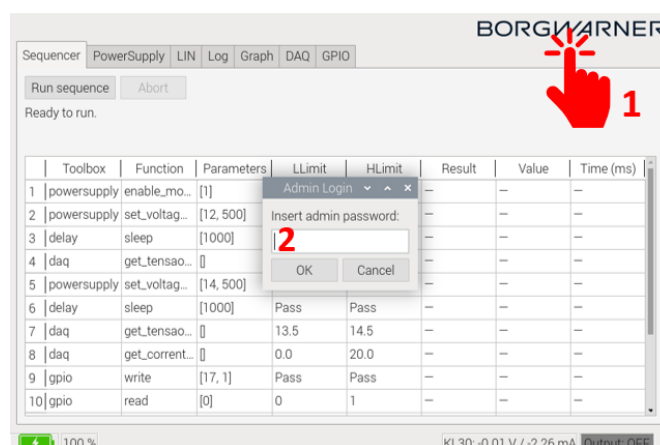


Figura 57 - Passos de acesso ao separador Admin

Esta área permite executar operações de carácter estrutural e de configuração, nomeadamente a saída do modo de ecrã completo, o carregamento de ficheiros de sequência de testes e o encerramento da sessão administrativa. A funcionalidade de saída do modo maximizado é particularmente relevante, pois a aplicação é iniciada intencionalmente em modo fullscreen, impedindo o acesso direto ao sistema operativo por parte do operador. Apenas no modo administrador é possível minimizar ou sair deste estado, garantindo assim um ambiente controlado em contexto industrial. Adicionalmente, a partir desta aba é possível seleccionar e carregar a sequência de testes a utilizar pelo sequenciador, sendo o caminho do ficheiro apresentado na interface para efeitos de

rastreabilidade. A sessão administrativa pode ser terminada através da opção “Log off”, restaurando o ambiente restrito de operação.



Figura 58 - Separador de administração

6. Validação Experimental e Avaliação de Desempenho

A etapa de validação constitui um dos momentos mais críticos no ciclo de desenvolvimento de sistemas, uma vez que é nesta fase que se comprova, de forma objetiva e mensurável, a conformidade da solução implementada com os requisitos funcionais, elétricos e operacionais previamente definidos. No contexto da eTestBox, esta fase assume particular relevância, dado que a plataforma se destina a operar em ambiente industrial automóvel, onde são exigidos elevados níveis de fiabilidade, precisão metrológica, robustez elétrica e estabilidade de comunicação.

A validação realizada não se limitou à verificação do correto funcionamento individual dos módulos, mas procurou avaliar o comportamento integrado do sistema como um todo, considerando interações entre subsistemas, resposta dinâmica, estabilidade sob carga e capacidade de execução repetitiva de ciclos de teste automatizados. Foram igualmente considerados critérios de robustez operacional, tais como imunidade a ruído elétrico, consistência de medições, determinismo na execução de sequências e integridade da comunicação com unidades de controlo eletrónicas reais (ECUs).

Os ensaios conduzidos abrangeram múltiplas dimensões da plataforma, nomeadamente a validação da comunicação LIN e CAN com dispositivos reais, a verificação metrológica da aquisição de dados, a atuação controlada de saídas digitais e relés, a leitura fiável de entradas, a execução de sequências automatizadas através do sequenciador e a avaliação do sistema de alimentação interna, incluindo testes de carga e descarga da bateria.

Os ensaios apresentados neste capítulo foram realizados de acordo com o plano de validação definido na Secção 3.3, onde se encontram especificadas as condições de teste e os critérios de aceitação para cada subsistema da plataforma. O objetivo desta fase consiste em demonstrar, através de evidência experimental, que a solução desenvolvida cumpre os requisitos funcionais e não funcionais previamente estabelecidos.

6.1. Validação da Comunicação LIN e CAN

A validação do subsistema de comunicação teve como objetivo verificar a capacidade da plataforma em estabelecer e manter comunicações fiáveis

através dos protocolos **LIN** e **CAN**, assegurando a correta troca de informação com dispositivos externos e a atualização consistente dos sinais recebidos.

Para a validação da comunicação CAN foram utilizados os dois canais CAN disponibilizados pela plataforma, configurados para uma velocidade de comunicação de até **1 Mb/s**. Os ensaios consistiram na troca contínua de mensagens com dispositivos reais e simulados, permitindo avaliar simultaneamente a transmissão, recepção e interpretação dos sinais presentes na rede. No caso da comunicação LIN, foi estabelecida uma rede constituída por um nó mestre e múltiplos escravos, recorrendo ao carregamento de ficheiros LDF para definição da estrutura da rede e dos sinais a monitorizar.

Durante os ensaios foram executadas operações de leitura e escrita de sinais, transmissão de mensagens de controlo, monitorização de variáveis de diagnóstico e atualização contínua de informação proveniente das ECUs ligadas ao sistema. Adicionalmente, foi utilizado o software **Vector CANoe** como ferramenta independente de monitorização, permitindo observar o tráfego presente nos barramentos LIN e CAN em paralelo com a eTestBox. Esta abordagem possibilitou verificar a correta temporização das mensagens, a integridade dos dados transmitidos, a ausência de erros físicos na rede e a coerência dos sinais apresentados pela aplicação desenvolvida.

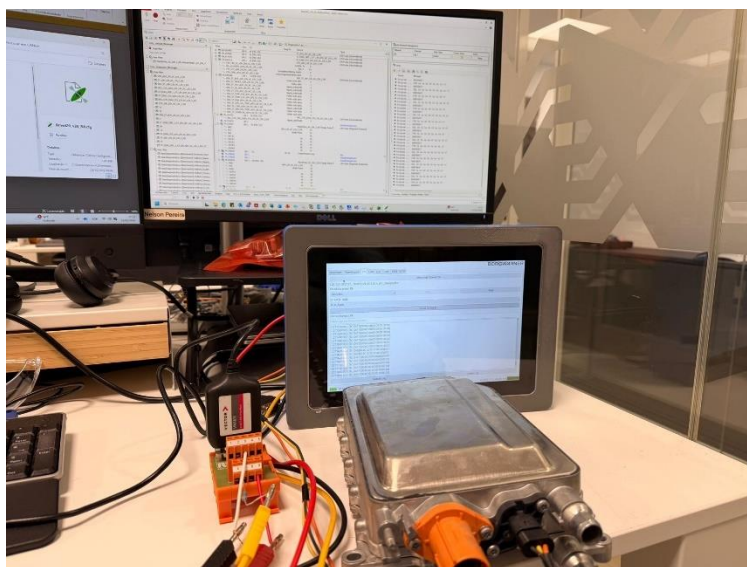


Figura 59 - Validação com Vector CANoe

Durante a execução dos ensaios não foram observadas perdas de mensagens, falhas de comunicação ou estados de erro associados aos barramentos. Na rede

LIN, operando a **19.2 kb/s**, todos os escravos responderam corretamente aos pedidos efetuados pelo mestre, não tendo sido registados **timeouts** ou inconsistências na atualização dos sinais definidos no ficheiro LDF. Na rede CAN, os dois canais operaram corretamente durante os ensaios de comunicação contínua, verificando-se a receção integral das mensagens transmitidas e a atualização correta dos sinais monitorizados.

A análise realizada através do Vector CANoe confirmou ainda a inexistência de erros de protocolo, colisões indevidas ou estados de erro do barramento, como **error passive** ou **bus-off**. Verificou-se igualmente que os sinais decodificados pela aplicação correspondiam aos valores efetivamente transmitidos na rede, demonstrando a correta interpretação dos ficheiros de descrição LDF e DBC.

Os resultados obtidos confirmam o correto funcionamento dos módulos de comunicação LIN e CAN, demonstrando a capacidade da plataforma para comunicar de forma estável e consistente com dispositivos automóveis reais, satisfazendo os critérios de validação definidos para este subsistema.

6.2. Validação da DAQ

A validação metrológica do sistema de aquisição de dados teve como objetivo aferir a **precisão**, **linearidade** e **repetibilidade** das medições de tensão e corrente realizadas pela placa MCC128 integrada na eTestBox. Para esse efeito, os valores medidos pelo sistema foram comparados com medições obtidas através de um **instrumento de referência** Keysight (Agilent) 34970A, amplamente reconhecido pela sua precisão e estabilidade em aplicações laboratoriais e industriais. Foram calculados os **erros absolutos** e **relativos** (percentuais) em diferentes pontos de operação distribuídos ao longo da gama de medição, permitindo avaliar o comportamento do sistema quanto a **offset**, erro de ganho e linearidade global.

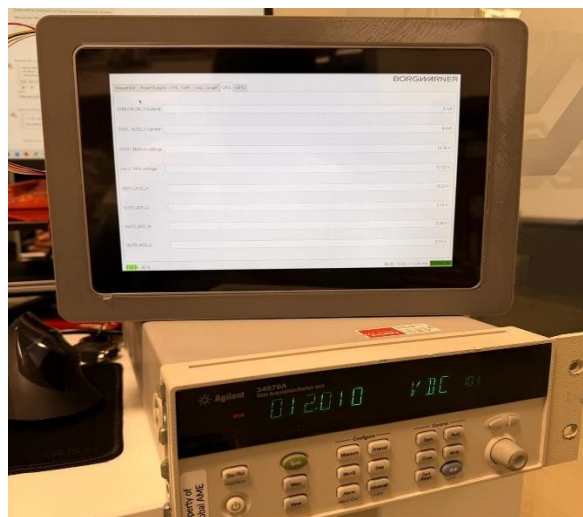


Figura 60 - Validação das tensões e correntes com Agilent 34970A

Para análise da tensão de alimentação KL30 foram aplicadas tensões de referência compreendidas entre 1 V e 24 V, comparando-se os valores medidos pela DAQ com os valores obtidos por instrumento de referência.

O erro absoluto máximo observado foi da ordem dos 10 mV em toda a gama de medição. Em termos relativos, o maior erro percentual ocorreu no extremo inferior da gama (1 V), fenómeno esperado em sistemas ADC devido à influência do erro de quantização e *offset* relativo.

A análise da distribuição dos pontos evidencia excelente linearidade ao longo da gama completa (1–24 V), não se observando tendência de ganho não linear ou deriva significativa. O comportamento praticamente simétrico do erro (ligeiramente positivo nos valores baixos e próximo de zero nos valores elevados) sugere presença de pequeno *offset* sistemático, mas sem impacto relevante na aplicação pretendida.

Considerando que, em ambiente EOL automóvel, **tolerâncias típicas** de validação de tensão se situam frequentemente entre $\pm 1\%$ e $\pm 2\%$, conclui-se que o sistema de medição de tensão apresenta desempenho plenamente adequado e tecnicamente robusto.

Na análise da corrente fornecida ao UUT1 e UUT2 foram utilizados vários pontos de carga entre 20mA e 1000mA.

Neste caso verificou-se uma tendência consistente de erro positivo proporcional à corrente aplicada. Este comportamento caracteriza um erro

predominantemente de ganho. A linearidade mantém-se elevada, indicando que o desvio é sistemático e previsível. Para aplicações industriais EOL, **tolerâncias de $\pm 5\%$ na medição de corrente são frequentemente aceitáveis**, sobretudo quando o objetivo é validação funcional e não caracterização metrológica de alta precisão. Ainda assim, a natureza sistemática do erro permite aplicar facilmente uma calibração por software.

	Programado	Medido (DAQ)	Referência (multímetro)	Erro absoluto	Erro relativo
KL30 (V)	1,00	1,01	1,00	0,01	1,4%
	5,00	5,01	4,99	0,02	0,4%
	12,00	12,00	11,98	0,02	0,2%
	24,00	23,99	23,90	0,09	0,4%
KL30_1 (mA)	20	20	20	0	0,0%
	35	36	35	1	2,9%
	100	102	99	3	3,0%
	200	206	199	7	3,5%
	500	512	500	12	2,4%
	1000	1025	990	35	3,5%
KL30_2 (mA)	20	21	20	1	5,0%
	35	36	35	1	2,9%
	100	103	99	4	4,0%
	200	204	199	5	2,5%
	500	510	500	10	2,0%
	1000	1018	990	28	2,8%

Tabela 9 - Erros absoluto e relativo dos sinais KL30

Foram igualmente avaliados os canais ADC associados aos UUT1 e UUT2, aplicando tensões de referência entre 0,5 V e 10 V. O erro típico manteve-se inferior a 10 mV em todos os pontos analisados. Observou-se excelente correspondência entre os canais ADC1 e ADC2 de cada UUT, bem como uniformidade entre UUT1 e UUT2, demonstrando elevada consistência inter-canal.

Não foi identificada instabilidade ou ruído significativo. A linearidade ao longo da gama analisada confirma o correto dimensionamento do circuito analógico.

	Programado (V)	0,5	1	2,5	5	7,5	10
	Referência (V)	0,49	0,99	2,48	4,97	7,46	9,94
ADC1 UTT 1	Medido (V)	0,50	1,00	2,50	5,00	7,50	10,00
	Erro Absoluto (V)	0,01	0,01	0,02	0,03	0,04	0,06
	Erro relativo (%)	2,0%	1,0%	0,8%	0,6%	0,5%	0,6%
ADC2 UTT 1	Medido (V)	0,50	1,00	2,50	5,00	7,50	9,99
	Erro Absoluto (V)	0,01	0,01	0,02	0,03	0,04	0,05
	Erro relativo (%)	2,0%	1,0%	0,8%	0,6%	0,5%	0,5%
ADC1 UTT 1	Medido (V)	0,50	1,00	2,50	5,00	7,50	10,00
	Erro Absoluto (V)	0,01	0,01	0,02	0,03	0,04	0,06
	Erro relativo (%)	2,0%	1,0%	0,8%	0,6%	0,5%	0,6%
ADC2 UTT 1	Medido (V)	0,50	1,00	2,50	5,00	7,49	9,99
	Erro Absoluto (V)	0,01	0,01	0,02	0,03	0,03	0,05
	Erro relativo (%)	2,0%	1,0%	0,8%	0,6%	0,4%	0,5%

6.3. Validação das Entradas e Saídas Digitais

A validação do subsistema de **entradas e saídas digitais** teve como objetivo verificar o correto funcionamento dos **optoacopladores** de entrada e dos **relés de contacto seco** integrados na plataforma. Os ensaios procuraram confirmar o cumprimento dos critérios de deteção dos **estados lógicos**, a integridade dos circuitos de isolamento, a capacidade de comutação dos relés e a repetibilidade do sistema durante ciclos prolongados de operação.

A validação das entradas digitais foi realizada através da aplicação de diferentes níveis de tensão, nomeadamente **0V, 2V, 5V, 12V, 24V e 28V**, medindo simultaneamente o estado lógico reconhecido pelo sistema e a **corrente absorvida por cada canal**. Foram igualmente determinados os **limiares de transição** entre os estados lógicos baixo e alto para cada entrada.

Os resultados obtidos, apresentados na Tabela 11, demonstram que todos os canais reconheceram corretamente o **estado lógico alto** para tensões iguais ou superiores a **2V**, garantindo compatibilidade com dispositivos baseados em lógica **TTL**. Os limiares de comutação situaram-se entre aproximadamente 1.49V e 1.92 V, verificando-se um comportamento consistente entre todos os canais.

	Limiar de nível lógico baixo	Limiar de nível lógico alto
UUT1 IN1	1.80V	1.85V
UUT1 IN2	1.83V	1.92V
UUT2 IN1	1.51V	1.62V
UUT2 IN2	1.49V	1.61V

Tabela 11 - Limiar de deteção das entradas digitais

Os valores de corrente medidos para os diferentes níveis de tensão encontram-se apresentados na Tabela 12. Verificou-se uma resposta uniforme entre todos os canais, sendo a corrente máxima observada de **26.6mA** para uma **tensão de entrada de 28 V**. Este valor é significativamente inferior ao limite máximo de 60mA definido para proteção dos optoacopladores, garantindo uma margem de segurança adequada para operação contínua.

	0V	2V	5V	12V	24V	28V
UUT1 IN1	OFF 0mA	ON 0.88mA	ON 3.80mA	ON 10.7mA	ON 22.6mA	ON 26.6mA
UUT1 IN2	OFF 0mA	ON 0.88mA	ON 3.80mA	ON 10.7mA	ON 22.6mA	ON 26.6mA
UUT2 IN1	OFF 0mA	ON 0.88mA	ON 3.79mA	ON 10.7mA	ON 22.6mA	ON 26.6mA
UUT2 IN2	OFF 0mA	ON 0.88mA	ON 3.79mA	ON 10.7mA	ON 22.6mA	ON 26.6mA

Tabela 12 - Estado lógico e corrente de entrada em função da tensão aplicada

A validação das saídas digitais foi efetuada através da execução de uma sequência automática desenvolvida especificamente para este ensaio. Para permitir a monitorização direta dos contactos dos relés, foi criada uma ligação física entre as saídas digitais e as respetivas entradas digitais através do conector DB9 disponibilizado na plataforma. Esta configuração permitiu utilizar as entradas digitais como mecanismo de verificação independente do estado dos relés, possibilitando a confirmação automática de cada comutação realizada.

Durante o ensaio, cada relé foi acionado individualmente através do sequenciador, sendo posteriormente verificada a alteração de estado na entrada digital correspondente. Foram executados 1000 ciclos consecutivos de comutação para cada canal, registando-se o tempo decorrido entre o comando de acionamento e a deteção da alteração de estado na entrada associada.

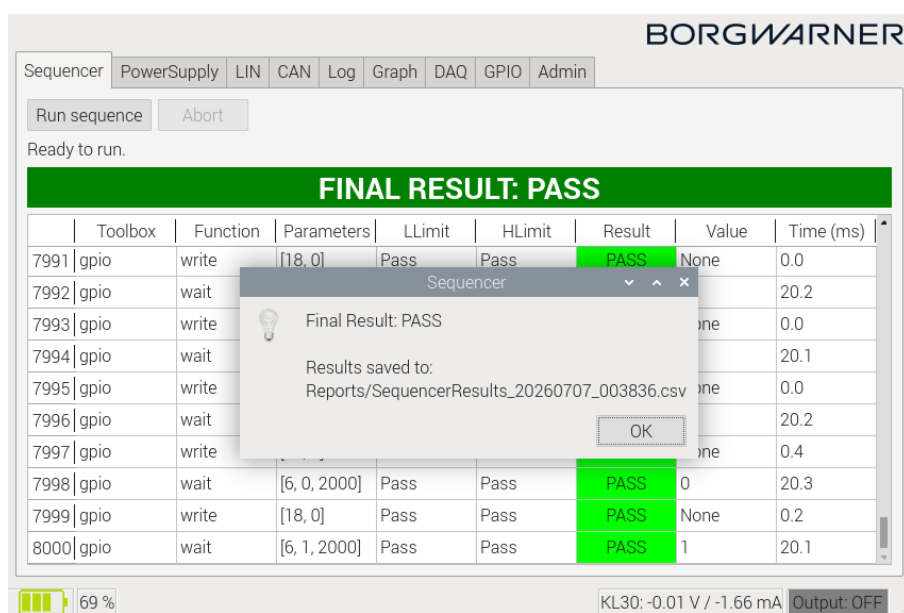


Figura 61 - Ensaio de validação das entradas e saídas digitais

Os resultados apresentados na Figura 61 demonstram a conclusão bem-sucedida da sequência de ensaio, sem ocorrência de falhas de acionamento ou inconsistências na leitura dos estados lógicos. Durante os 1000 ciclos executados, todos os relés apresentaram comportamento consistente e repetível, não tendo sido registradas falhas de comutação.

A análise estatística dos tempos de resposta demonstrou um comportamento consistente e repetível em todos os relés ensaiados. Os tempos médios de atuação obtidos foram de 1.19 ms para o Relé 1, 1.16 ms para o Relé 2, 1.21 ms para o Relé 3 e 1.17 ms para o Relé 4. O maior tempo de comutação observado durante todo o ensaio foi de 2.4 ms, permanecendo abaixo do limite máximo de 3 ms definido como critério de aceitação. Estes resultados evidenciam a elevada rapidez de resposta dos relés e confirmam a adequação do subsistema de saídas digitais aos requisitos de teste e automação definidos para a plataforma. Estes resultados evidenciam elevada rapidez de resposta e reduzida dispersão temporal entre ciclos consecutivos.

Os resultados obtidos demonstram que tanto as entradas digitais como as saídas digitais cumprem integralmente os critérios de aceitação definidos no plano de validação, verificando-se detecção correta dos estados lógicos, ausência de ativações indevidas, proteção adequada dos optoacopladores e funcionamento consistente dos relés durante ciclos repetitivos de operação.

6.4. Validação do Sequenciador

A validação do módulo de sequenciação automática incidiu sobre a capacidade do sistema em executar ciclos completos de teste de forma autônoma e determinística. Para este efeito, foram carregadas tabelas de teste estruturadas em formato Excel, contendo instruções para controlo da fonte de alimentação, leitura da DAQ, atuação de GPIOs e aplicação de temporizações.

Durante os ensaios, o motor de execução seguiu as instruções de forma sequencial, registrando tempos de execução, valores medidos e resultado de cada passo com base nos limites definidos. O sistema demonstrou comportamento determinístico, avaliação correta das condições PASS/FAIL e coerência entre os resultados apresentados e as medições reais observadas externamente.

6.5. Ensaio de Autonomia

Por fim, foram realizados ensaios de validação do sistema de alimentação interna, com particular incidência na **autonomia** e estabilidade da bateria integrada. O sistema foi operado em **regime contínuo** sob dois cenários distintos de carga, permitindo avaliar o seu desempenho energético em condições reais de utilização.

No primeiro regime de operação, sem UUT conectado (apenas consumo interno da eTestBox), a autonomia total medida foi de aproximadamente **9 horas e 30 minutos**. Neste cenário, a descarga da bateria apresentou comportamento progressivo e estável, com curva de tensão típica de descarga de bateria Li-ion, sem quedas abruptas ou instabilidade do sistema.

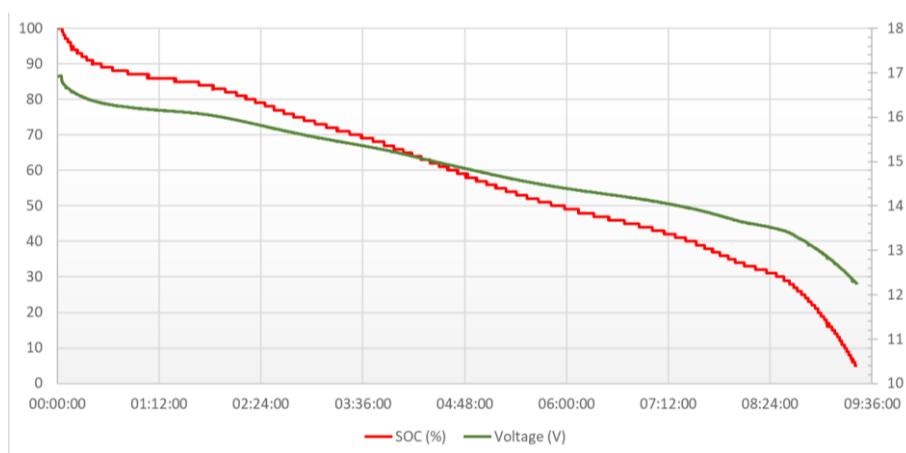


Figura 62 - Curva descarga bateria sem UUT

No segundo regime, considerado representativo de **utilização real em ambiente industrial**, a eTestBox alimentou um UUT a **12 V** com consumo médio de **70 mA**, valor compatível com os produtos tipicamente testados na BorgWarner. Nesta condição, a autonomia medida foi de aproximadamente **8 horas**, correspondendo a um turno completo de trabalho. A descarga manteve comportamento linear e previsível ao longo da maior parte do ciclo, verificando-se apenas a queda mais acentuada na fase final, característica da zona de descarga profunda da bateria.

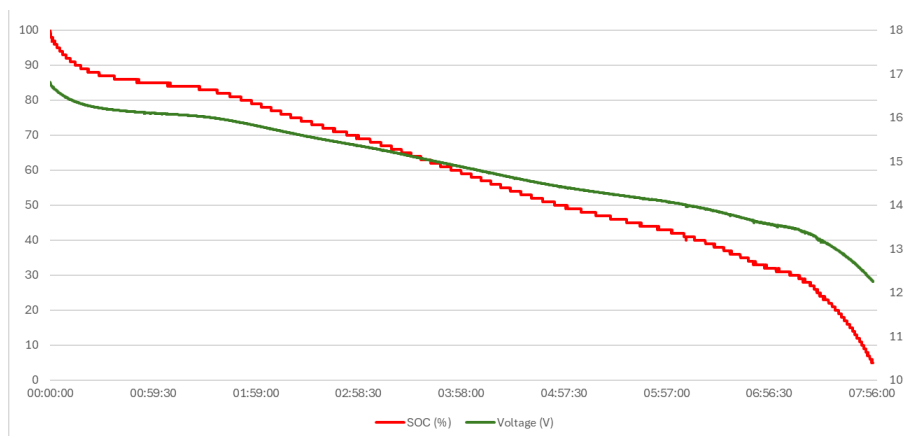


Figura 63 - Curva descarga bateria com UUT

Durante ambos os ensaios não se registaram reinícios inesperados, oscilações críticas de tensão ou degradação funcional dos módulos. O sistema de monitorização da bateria apresentou leituras coerentes com o estado real de carga (SOC), permitindo acompanhar a evolução energética da plataforma ao longo do tempo.

Os resultados obtidos demonstram que a eTestBox cumpre o requisito funcional definido de autonomia mínima para um dia de trabalho (8 horas) quando operando em regime típico com UUT alimentado.

7. Análise Crítica

7.1. Discussão dos Resultados Obtidos

A eTestBox demonstrou, no âmbito do presente trabalho, um desempenho **consistente e alinhado com os objetivos** inicialmente definidos. A arquitetura baseada em Raspberry Pi revelou-se **tecnicamente viável** para a execução de testes funcionais e de fim de linha (EOL), assegurando aquisição analógica, comunicação veicular e controlo automatizado num único sistema integrado.

A integração entre **hardware e software** mostrou-se **coerente e estável**, permitindo centralizar funções que, em contexto tradicional, exigiriam múltiplos instrumentos dedicados. O motor de sequenciação desenvolvido evidenciou robustez suficiente para suportar fluxos de teste estruturados, validação automática de limites elétricos e determinação do estado final da UUT. A sincronização entre aquisição de dados, comunicação CAN/LIN e lógica de teste confirmou a capacidade do sistema para operar de forma coordenada em ambiente industrial.

Durante o desenvolvimento, foi igualmente evidenciada a importância de práticas formais de gestão de configuração. A manutenção de cópias de segurança do sistema operativo e da aplicação revelou-se determinante para assegurar continuidade operacional após alterações experimentais. Esta experiência reforça a necessidade de incorporar metodologias estruturadas de controlo de versões e recuperação em sistemas embarcados aplicados a contexto industrial.

7.2. Limitações e Propostas de Melhorias Futuras

Apesar dos resultados positivos obtidos e da validação funcional da arquitetura proposta, a análise crítica do sistema permitiu identificar um conjunto de limitações técnicas que condicionam o seu grau de industrialização, escalabilidade e maturidade tecnológica. Estas limitações, contudo, configuram simultaneamente oportunidades claras de evolução futura.

Ao nível da interface gráfica, embora a estrutura modular implementada se revele funcional e coerente, subsistem fragilidades associadas à **usabilidade, organização hierárquica** da informação e **clareza dos indicadores** operacionais. Em contexto industrial, onde a rapidez de interpretação e a minimização de erro humano assumem carácter crítico, a ergonomia visual

constitui um fator determinante. Neste sentido, uma evolução natural do sistema passará pela reformulação da interface homem-máquina, privilegiando **dashboards** mais intuitivos, indicadores de estado de leitura imediata e fluxos de interação simplificados, orientados para o operador.

No domínio da integração com sistemas externos, a arquitetura atual permite comunicação com outras plataformas, mas carece de formalização estruturada através de **protocolos industriais** normalizados. A ausência de implementação consolidada de mecanismos como *OPC UA* ou *MQTT* limita a interoperabilidade direta com sistemas *MES* e *SCADA*, reduzindo o potencial de **rastreabilidade automática** e geração sistemática de **relatórios**. A adoção formal destes protocolos constituirá um passo decisivo para a plena integração da eTestBox em ecossistemas de Indústria 4.0, reforçando a conectividade e a integração vertical da informação de teste.

O motor de sequenciação desenvolvido demonstrou robustez e estabilidade na execução de fluxos lineares de teste; contudo, apresenta limitações ao nível da expressividade lógica. A inexistência de estruturas de controlo avançadas - como **ciclos condicionais dinâmicos**, ramificações estruturadas e **tratamento formal de exceções** - restringe a implementação de cenários de teste mais complexos. A sua evolução para um modelo mais flexível e declarativo permitirá aumentar a adaptabilidade da plataforma a diferentes tipologias de produto e requisitos de validação.

Relativamente à medição de corrente, verificou-se que os sensores atualmente utilizados apresentam **resolução limitada** em regimes de baixa corrente. Embora adequados para monitorização em funcionamento ativo, a caracterização de consumos em estados de *sleep* ou *standby* - particularmente relevantes em aplicações automóveis - poderá exigir maior sensibilidade. Futuras iterações de hardware poderão contemplar **sensores de maior resolução** ou estágios de amplificação dedicados a medições de baixa corrente, melhorando a capacidade de análise energética da *UUT*.

Adicionalmente, embora a eTestBox permita comunicação e controlo de ECUs através dos protocolos *LIN* e *CAN*, não foi implementado suporte completo a processos de **software flashing**. A programação de *firmware* automóvel envolve camadas adicionais de protocolo, como serviços diagnósticos estruturados,

gestão de sessões, autenticação, mecanismos de segurança e transferência segmentada de dados com controlo rigoroso de integridade. A inclusão desta funcionalidade representaria um avanço técnico significativo, posicionando a plataforma não apenas como sistema de validação funcional, mas também como ferramenta de programação e atualização de *firmware*.

Por fim, identificam-se ainda oportunidades de expansão funcional, nomeadamente a inclusão de **suporte a testes RF** (Rádio-Frequência), armazenamento estruturado local de resultados com base de dados dedicada, integração com serviços *cloud* para **análise remota** e implementação de mecanismos automáticos de calibração. Estas evoluções poderão reforçar a robustez, versatilidade e capacidade de escalabilidade da eTestBox, aproximando-a progressivamente de uma solução industrial plenamente consolidada.

Conclusões

O presente trabalho teve como objetivo o desenvolvimento de uma plataforma portátil, modular e industrialmente aplicável para testes de unidades eletrônicas automóveis, integrando comunicação LIN e CAN, aquisição de dados, controle de alimentação e automação de sequências de teste.

A eTestBox materializa uma solução tecnicamente sólida, concebida segundo princípios de engenharia eletrônica, integração de sistemas embarcados e desenvolvimento de software estruturado. O projeto evoluiu desde uma fase inicial de validação com cablagem experimental até à implementação de um PCB dedicado de quatro camadas, chassis próprio concebido em SolidWorks e arquitetura de software modular baseada em separação clara entre interface gráfica e camada de acesso a hardware.

Os resultados obtidos demonstram que a solução cumpre os requisitos funcionais e operacionais definidos, assegura autonomia adequada para operação em campo, mantém estabilidade elétrica e de comunicação, e suporta automação estruturada de testes industriais. Para além da componente técnica, o projeto evidencia capacidade de adaptação, escalabilidade e potencial de industrialização.

A eTestBox posiciona-se, assim, não apenas como uma ferramenta de teste funcional, mas como uma plataforma extensível, preparada para futuras evoluções tecnológicas e integração em ecossistemas industriais mais abrangentes, contribuindo para a eficiência e fiabilidade dos processos de validação no setor automóvel.

Referências Bibliográficas

- Adafruit Industries LLC. (2025). *Panel Mount Ethernet Extension Cable*. Obtido de Adafruit Industries LLC: <https://www.adafruit.com/product/909>
- Adafruit Industries LLC. (2025). *Panel Mount USB Cable - A Male to A Female*. Obtido de Adafruit Industries LLC: <https://www.adafruit.com/product/908>
- Ajithkumar, A., Ajith, A., Sabu, T., Kurian, T. T., & Beena, A. (2024). *Digital Aided Analog Data Acquisition for Telemetry*. Obtido de Research Gate: https://www.researchgate.net/publication/381679644_Digital_Aided_Analog_Data_Acquisition_for_Telemetry
- Allegro MicroSystems . (2025). *ACS724 Datasheet*. Obtido de Allegro MicroSystems : <https://www.allegromicro.com/-/media/files/datasheets/acs724-datasheet.ashx>
- Bishop, R. H. (2018). *The mechatronics handbook (2nd ed.)*. CRC Press.
- CAN BUS tools. (2025). Obtido de CAN BUS tools: <https://cantools.readthedocs.io>
- CiA. (2012). *CAN specification version 2.0*. Obtido de can-cia.org.
- CSS Electronics . (2025). *LIN Bus Explained - A Simple Intro*. Obtido de CSS Electronics : <https://www.csselectronics.com/pages/lin-bus-protocol-intro-basics>
- CSS Electronics. (2025). *CAN DBC File Explained - A Simple Intro*. Obtido de CSS Electronics: <https://www.csselectronics.com/pages/can-dbc-file-database-intro>
- Digilent. (2025). *Data Acquisition Handbook*. Obtido de Digilent Downloads: <https://files.digilent.com/reference%2Fdata-acquisition-handbook.pdf>
- Donnison, M. (2023). *The new Raspberry Pi 5 launches today*. Obtido de Kitronik: <https://kitronik.co.uk/blogs/resources/raspberry-pi-5-launch-today>
- EDATEC. (2025). *ED-HMI3010-101C 10.1-inch Industrial HMI Based on Raspberry Pi 5*. Obtido de EDATEC Documentation Center: <https://edatec.cn/docs/hmi3010-101c/ds/>
- Elettronica Open Source. (2018). *CAN bus*. Obtido de Elettronica Open Source: <https://it.emcelettronica.com/il-can-bus>
- Fowler, M. (2004). *Patterns of enterprise application architecture*. Addison-Wesley.
- Harrison, R., Vera, D., & Ahmad, B. (2016). *Engineering methods and tools for cyber-physical automation systems*. Obtido de <https://ieeexplore.ieee.org/document/7437382>
- International Organization for Standardization. (2015). *ISO 11898-1:2015 road vehicles — Controller area network (CAN)*. ISO.
- International Organization for Standardization. (2016). *ISO 17987-2:2016 road vehicles — Local interconnect network (LIN)*. ISO.
- International Organization for Standardization. (2020). *ISO 14229-1:2020 road vehicles — Unified diagnostic services (UDS)*. ISO.

Keysight Technologies. (2019). *34970A/34972A Data Acquisition / Switch Unit User's Guide*. Obtido de 34970A Data Acquisition / Data Logger Switch Unit: <https://www.keysight.com/hk/en/assets/9018-02644/user-manuals/9018-02644.pdf>

KiCad. (2025). *KiCad EDA documentation*. Obtido de <https://docs.kicad.org>

Leen, G., & Heffernan, D. (2002). *Expanding automotive electronic systems*. Obtido de IEEE Xplore: <https://ieeexplore.ieee.org/document/976923>

M5Stack Technology. (2025). *Module13.2 PPS Documentation*. Obtido de M5Stack: <https://docs.m5stack.com/en/module/Module13.2-PPS>

Measurement Computing Corporation. (2023). *MCC 128 DAQ HAT user's guide*. Obtido de <https://mccdaq.github.io/daqhats/overview.html#mcc-118>

Navet, N., Song, Y., Simonot-Lion, F., & Wilwert, C. (2005). *Trends in Automotive Communication Systems*. Obtido de Proceedings of the IEEE: <https://ieeexplore.ieee.org/document/1435746>

Navet, N., Song, Y., Simonot-Lion, F., & Wilwert, C. (2008). *A Review of Embedded Automotive Protocols*. Obtido de Proceedings of the IEEE: <https://ieeexplore.ieee.org/document/1435746>

PEAK-System. (2025). *PCAN-USB Pro FD user manual*. Obtido de https://www.peak-system.com/produktcd/Pdf/English/PCAN-USB-Pro-FD_UserMan_eng.pdf

PEAK-System. (2025). *PLIN-USB - User Manual*. Obtido de https://www.peak-system.com/produktcd/Pdf/English/PLIN-USB_UserMan_eng.pdf

Praprotnik, O., Gartner, M., Zauner, M., & Horauer, M. (2009). *A Test Suite for System Tests of Distributed Automotive Electronics*. Obtido de International Conference on Advances in Circuits, Electronics and Micro-electronics: <https://ieeexplore.ieee.org/document/5291499>

Python Software Foundation. (2025). *Python 3.14.3 documentation*. Obtido de <https://docs.python.org/3/>

Qt Company. (2025). *Qt documentation*. Obtido de <https://doc.qt.io>

Raspberry Pi Foundation. (2025). *Raspberry Pi documentation*. Obtido de <https://www.raspberrypi.org/documentation>

Same Sky (Formerly CUI Devices). (2024). *PJ-038BH Datasheet*. Obtido de Same Sky Technical Docs: <https://www.sameskydevices.com/product/resource/pj-038bh.pdf>

Smith, G. M. (2026). *End-of-Line Testing: Ensuring Quality and Performance in Modern Manufacturing*. Obtido de DEWESoft: <https://dewesoft.com/blog/end-of-line-testing>

TDK-Lambda. (2024). *CCG 15-30W Series Datasheet*. Obtido de TDK Product Center: https://product.tdk.com/system/files/dam/doc/product/power/switching-power/dc-dc-converter/catalog/ccg_e.pdf

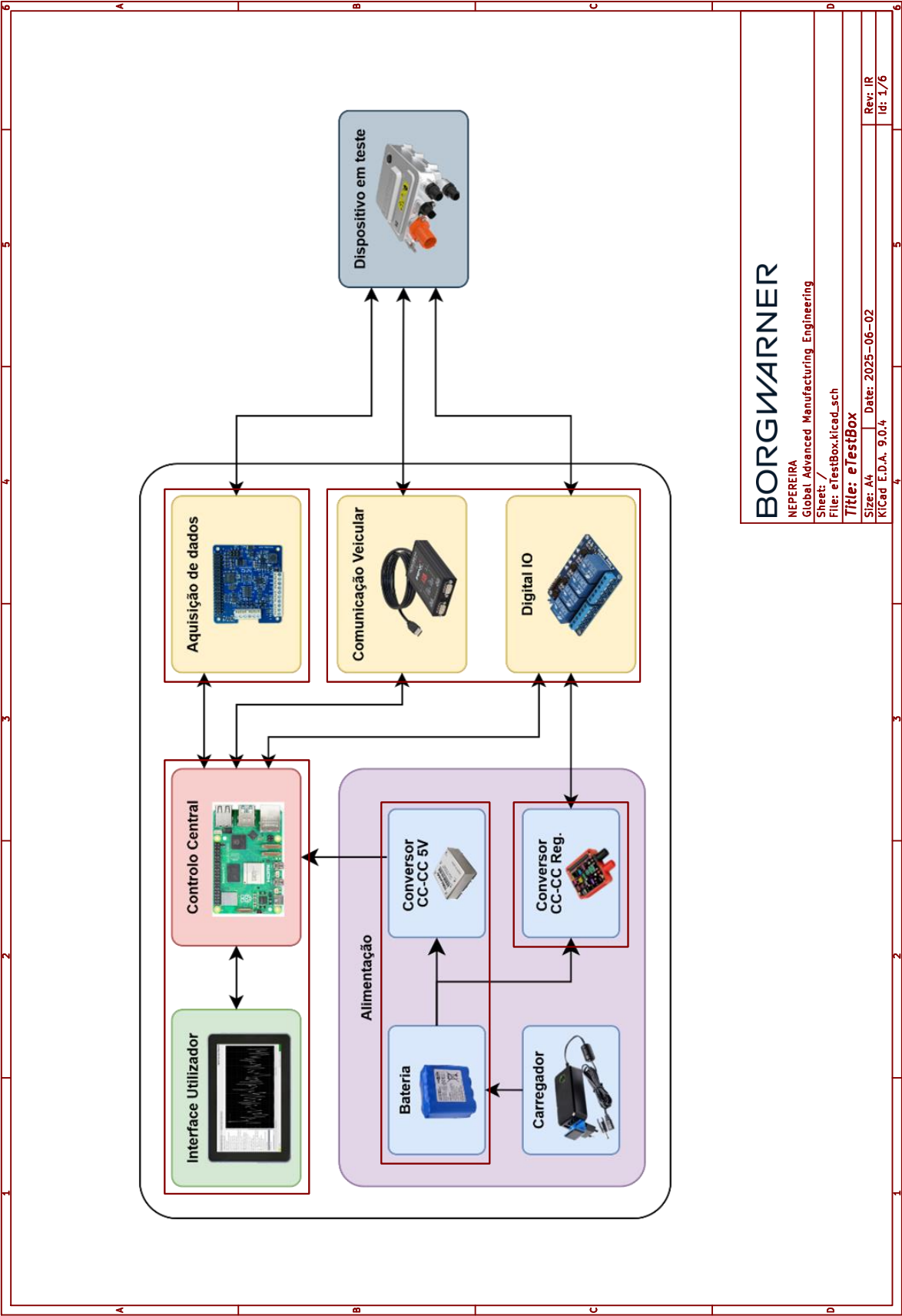
Traco Power. (2025). *TSR 2 Datasheet*. Obtido de Traco Power: https://www.tracopower.com/sites/default/files/products/datasheets/tsr2_datasheet.pdf?t=1772064001

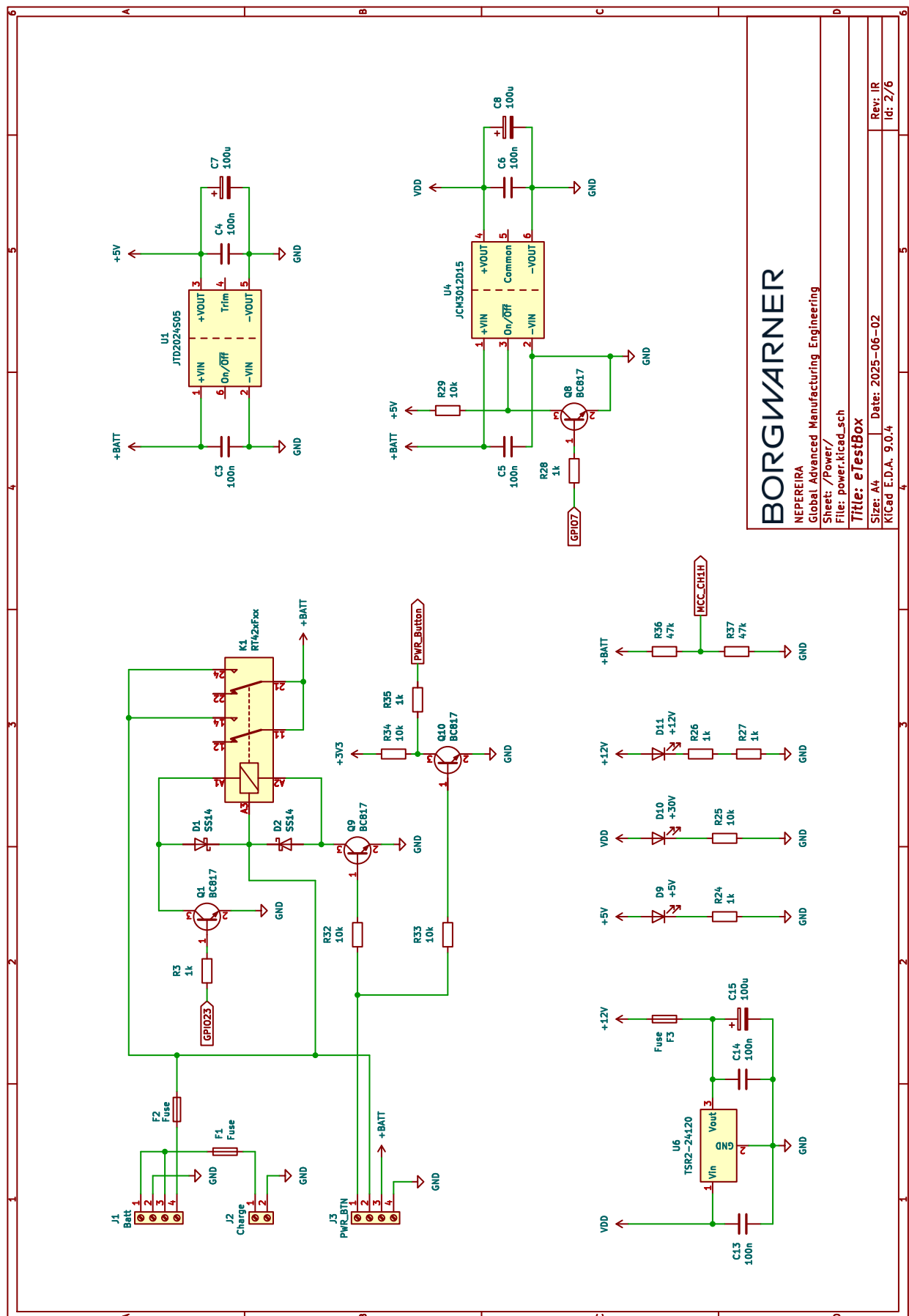
Willman, J. (2021). *Modern PyQt*. Hampton, VA, USA: Apress Media LLC.

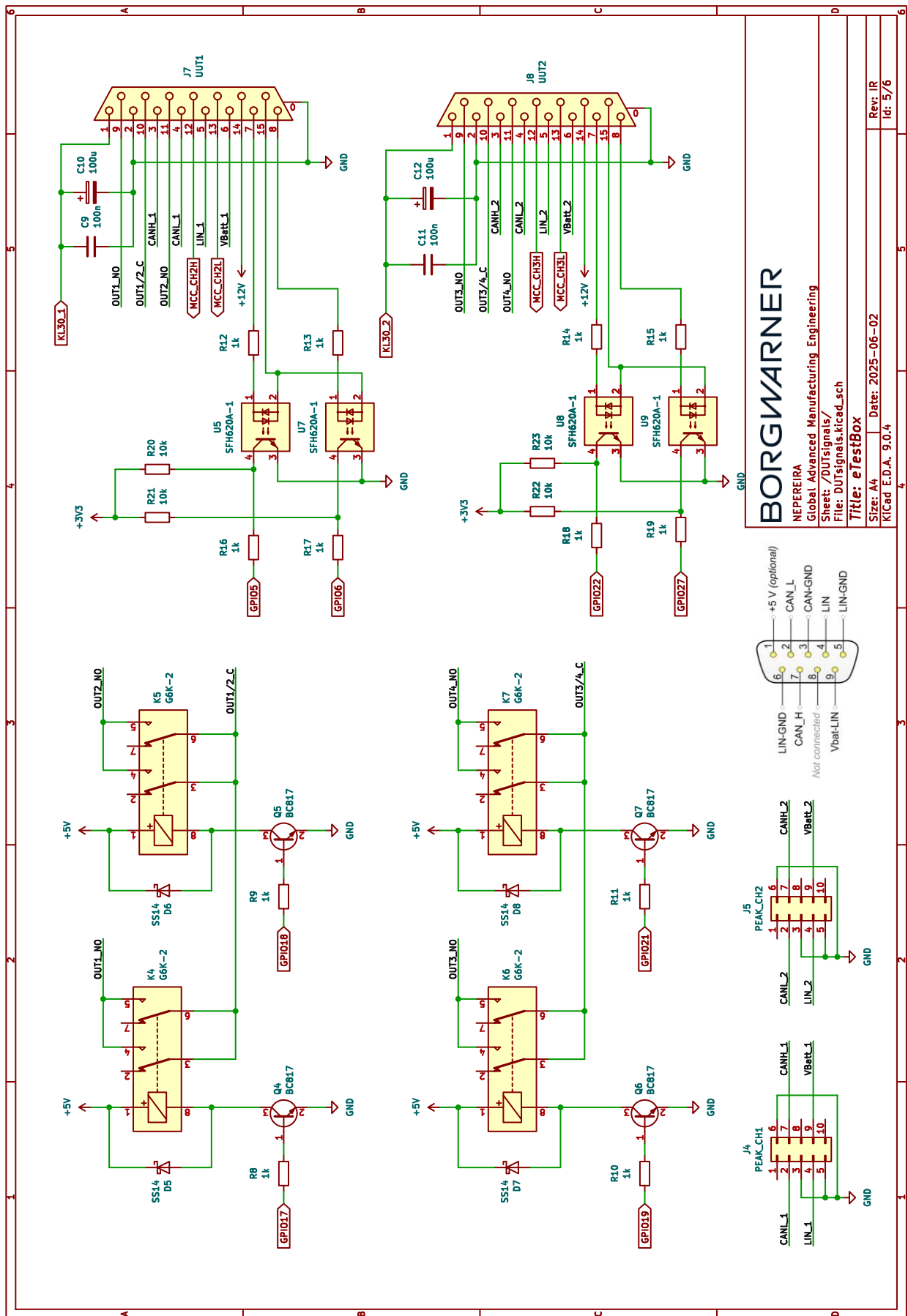
XP Power. (2025). *JTD20 series*. Obtido de XP Power datasheets:
https://www.xppower.com/portals/0/pdfs/SF_JTD20.pdf

Apêndices

Apêndice A – Diagrama Elétrico da eTestbox







BORGWARNER

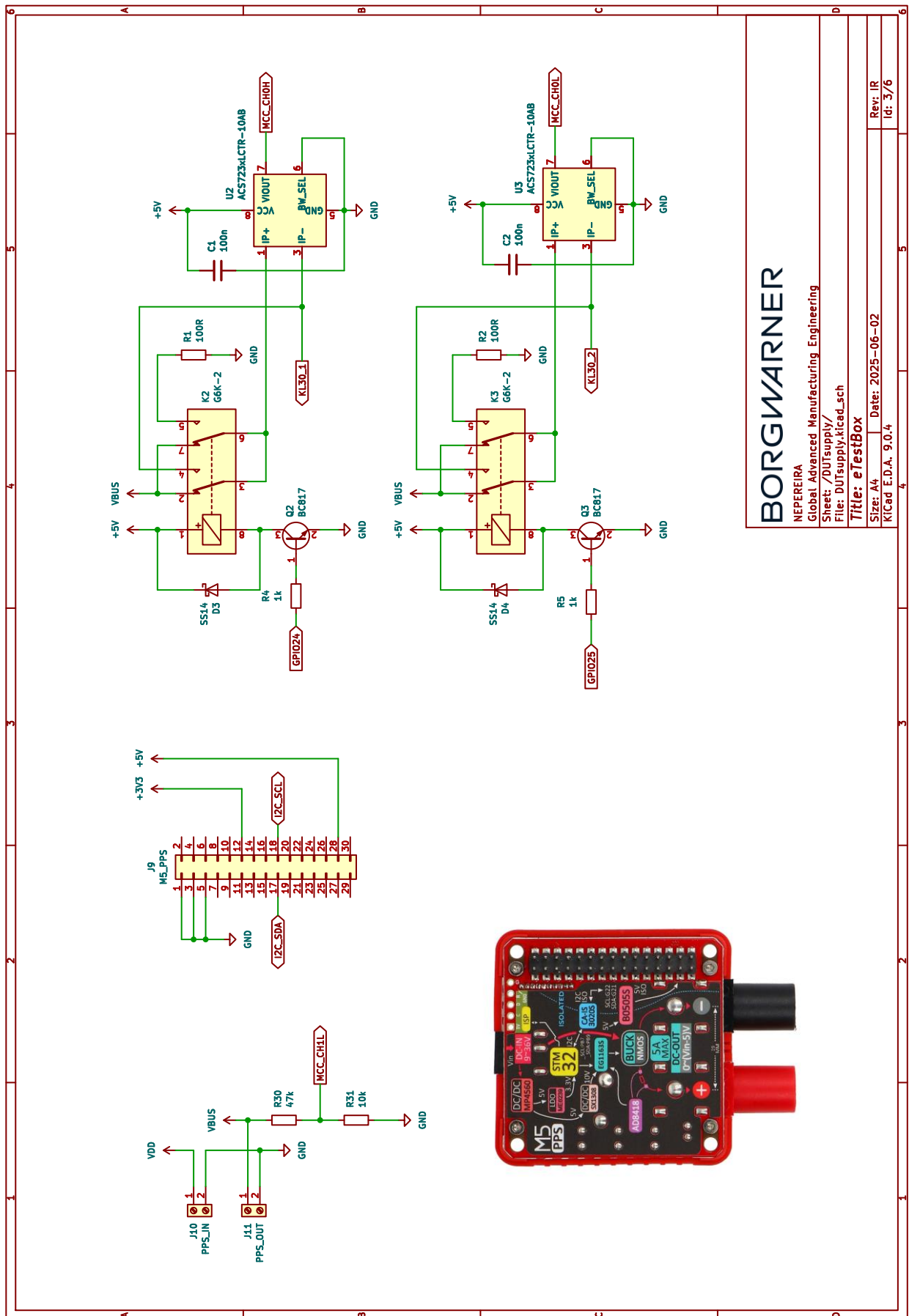
NEPEREIRA
Global Advanced Manufacturing Engineering

Sheet: /DUTsignals/
File: DUTsignals.kicad_sch

Title: eTestBox

Size: A4
KICad E.D.A. 9.0.4

Date: 2025-06-02
Rev. IP
Id: 5/6



BORGWARNER

NEPEREIRA
Global Advanced Manufacturing Engineering

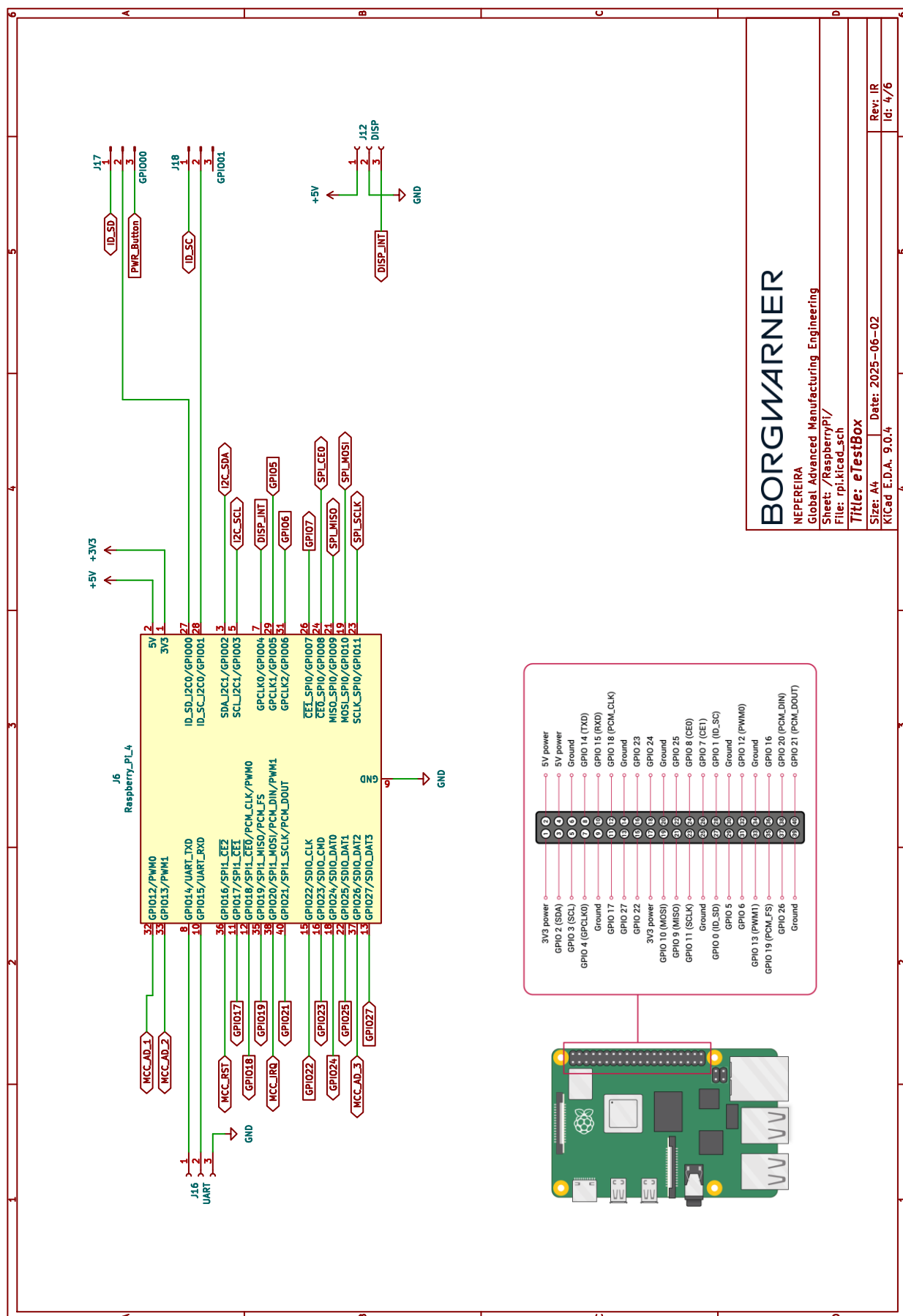
Sheet: /DUISupply/
File: DUISupply.kicad_sch

Title: eTestBox

Size: A4 Date: 2025-06-02

Kicad E.D.A. 9.0.4

Rev: IR
Id: 3/6



Apêndice B – Lista De Componentes do PCB

Designação	Marca	Modelo	Quantidade
C7,C15,C8,C10,C12	PANASONIC ELECTRONIC	EEH-ZA1H101P	5
C9,C3,C2,C14,C6,C11,C13,C1,C5,C4	Cal-Chip Electronics, Inc.	GMC31X7R104K100NT	10
D6,D1,D7,D5,D4,D8,D2,D3	OnSemi	SS14	8
D9, D10, D11	LITEON	LTST-C171KRKT	3
F3,F1,F2	Littelfuse Inc.	OPTF0078P	3
J1, J2, J10, J11	Dinkle Corporation	ELK508V-02P	4
J12, J16, J17, J18	JST Sales America Inc.	03JQ-BT	4
J13	Sullins Connector Solutions	NPTC202KFMS-RC	1
J14	Sullins Connector Solutions	PPTC061LFBN-RC	1
J15	Sullins Connector Solutions	PPTC101LFBN-RC	1
J3	JST Sales America Inc.	04JQ-BT	1
J4, J5	On Shore Technology Inc.	302-S101	2
J6	Sullins Connector Solutions	SBH11-PBPC-D20-ST-BK	1
J7, J7	Amphenol CS	LD15S13A4GV00LF	2
J9	Samtec Inc.	ESQ-115-44-L-D	1
K1	TE Connectivity	RT424F12	1
K2,K6,K3,K4,K7,K5	Omron	G6K-2F-Y DC4.5	6
Q8,Q7,Q5,Q2,Q6,Q9,Q3,Q4,Q1,Q10	OnSemi	BC817-16LT3G	10
R1,R2	TE Connectivity	RR03J470RTB	2
R15,R12,R14,R13	Vishay Dale	CRCW25121K00FKEG	4
R30,R6,R7	TE Connectivity	CRGCQ1206F47K	3
R31,R34,R22,R33,R20,R23,R21,R25, R32,R29	YAGEO	RC1206FR-0710KL	10
R35,R24,R26,R16,R9,R4,R27,R3,R17, R10,R5,R18,R19,R8,R11,R28	Stackpole Electronics Inc.	RNCP1206FTD1K00	16
U1	XP Power	JTD2024S05	1
U3,U2	Allegro MicroSystems	ACS724LLCTR-2P5AB-S	2
U4	TDK-Lambda	CCG30-24-15D	1
U6	Traco Power	TSR 2-24120	1
U8,U7,U9,U5	Vishay Semiconductor	SFH620A-1	4

Apêndice C – Lista de Componentes do Chassis

Designação	Marca	Modelo	Quantidade
Raspberry Pi 5 8GB RAM	Raspberry Pi	SC1112	1
HMI 10"	EDATEC	ED-HMI3010-101C-0000	1
Módulo CAN/LIN	PEAK	PCAN-USB Pro FD	1
Fonte alimentação programável	M5Stack Technology Co., Ltd	M137	1
DAQ MMC128	Digilent, Inc.	6069-410-001	1
Cartão memória 64GB	ATP Electronics, Inc.	AF64GUD4-BBBIM	1
Bateria Li-ion 14.8V / 7Ah	ANSMANN	2447-3035-20	1
Carregador bateria Li-ion	Mascot	2240000043	1
Flat cable 40 pinos	Assmann WSW Components	H3DDS-4006G	1
Flat cable 20 pinos	Assmann WSW Components	H3CCH-1018G	2
Ficha DB9 para flat cable	Amphenol CS	L17DEFRA09S	2
Insertos metálicos M3	Adafruit Industries	4255	25
Parafusos M3	Würth Elektronik	971110321	25
Conector DC Jack	Same Sky	PJ-038BH	1
Adaptador ethernet painel	Adafruit Industries	909	1
Adaptador USB painel	Adafruit Industries	908	1
Interruptor On/Off	Adam Tech	SW-R3-1A-A-1-0	1
Botão de pressão iluminado	TE Connectivity	AV1610P612R04	1

Apêndice D – Especificação das Funções do Sequencer Engine

ToolBox	Função	Parâmetros	Tipo / Domínio	Descrição	Valor de Retorno
Power Supply	enable_module(state)	state	{0,1}	Ativa (1) ou desativa (0) a saída da fonte	None
	set_voltage_current(voltage, current)	voltage, current	voltage: V (float)current: mA/A (float)	Configura tensão e limite de corrente	None
	read_voltage()	—	—	Lê a tensão atual da saída	Float (V)
	read_current()	—	—	Lê a corrente atual fornecida	Float (A ou mA)
DAQ	get_tensao_fonte()	—	—	Mede a tensão KL30	Float (V)
	get_corrente_kl30_1()	—	—	Mede a corrente no canal KL30_1	Float (mA)
	get_corrente_kl30_2()	—	—	Mede a corrente no canal KL30_2	Float (mA)
	read_adc(channel, mode)	channel, mode	channel: int mode ∈ {Single-Ended, Differential}	Leitura analógica configurável	Float (V)
LIN	load_idf(path)	path	String (caminho ficheiro)	Carrega ficheiro LDF	Boolean
	start_schedule(name)	name	String	Inicia schedule table	Boolean
	stop_schedule()	—	—	Interrompe schedule ativa	Boolean
	read_signal(signal_name)	signal_name	String	Lê valor atual do sinal	Float / Int
CAN	write_signal(signal_name, value)	signal_name, value	String, numérico	Escreve valor num sinal	Boolean
	load_dbc(path)	path	String (caminho ficheiro)	Carrega ficheiro DBC	Boolean
	start_schedule(name)	name	String	Inicia schedule table	Boolean
	stop_schedule()	—	—	Interrompe schedule ativa	Boolean
GPIO	read_signal(signal_name)	signal_name	String	Lê valor atual do sinal	Float / Int
	write_signal(signal_name, value)	signal_name, value	String, numérico	Escreve valor num sinal	Boolean
	write(pin, state)	pin, state	pin: int state ∈ {0,1}	Define estado lógico da saída	None
	read(pin)	pin	int	Lê estado lógico atual	{0,1}
Delay	wait(pin, expected_state, timeout_ms)	pin, expected_state, timeout_ms	pin: int expected_state ∈ {0,1} timeout_ms: int	Aguarda até atingir estado esperado ou timeout	{0,1} / Timeout
	sleep(delay_ms)	delay_ms	int (ms)	Introduz atraso controlado	None

Apêndice E – Layout Do PCB

