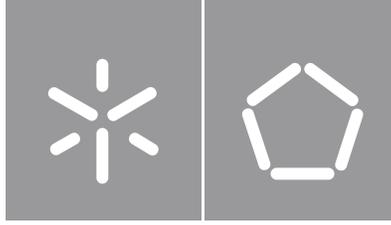




Universidade do Minho
Escola de Engenharia

Hugo Manuel da Cunha Ferreira

**Projeto dos principais subsistemas de um
pico-satélite para aplicações de ensino e de
investigação**



Universidade do Minho

Escola de Engenharia

Hugo Manuel da Cunha Ferreira

**Projeto dos principais subsistemas de um
pico-satélite para aplicações de ensino e de
investigação**

Dissertação de Mestrado

Mestrado Integrado em Engenharia Eletrónica Industrial e
Computadores

Trabalho efetuado sob a orientação do

Professor Doutor Alexandre Ferreira da Silva

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição-NãoComercial-Compartilhalgual

CC BY-NC-SA

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

AGRADECIMENTOS

Gostaria de expressar os meus sinceros agradecimentos a todos aqueles que fizeram parte deste trajeto e contribuíram de alguma forma para o término desta etapa final.

Primeiramente, gostaria de agradecer ao meu orientador, o Professor Doutor Alexandre Ferreira da Silva, pela sua disponibilidade, partilha de conhecimento e pelo constante auxílio durante a realização desta dissertação.

Aos meus pais, que sempre me apoiaram em todos os momentos, bons e menos bons, desta longa caminhada, principalmente à minha mãe que sempre me incentivou a dar o melhor de mim, mesmo quando eu duvidava das minhas próprias capacidades.

Ao meu irmão Hélder, pessoa que tanto admiro, por me ter apoiado em todas as minhas escolhas e me ter mostrado que com esforço e dedicação tudo é possível.

Aos meus queridos sobrinhos, que apareceram neste mundo durante esta caminhada, pelas alegrias e amor que tanto me dão.

Ao meu amigo e fiel parceiro de grupo, José Pires, que desde o meu segundo ano de curso, me apoiou e incentivou a não desistir, por todo o conhecimento partilhado comigo e por todas as memórias que agora levo para a vida.

Ao meu grande amigo e fisioterapeuta, Pedro Ferreira, por ter feito sempre parte da minha vida desde a infância, do meu percurso académico desde o ensino básico e agora durante o meu atual percurso de reabilitação após ter sido operado ao ligamento cruzado anterior.

Um agradecimento especial aos meus grandes amigos da “Favela” e do clube Grupo Desportivo Águas Santas, por terem sido peças fundamentais nesta longa caminhada.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Este trabalho foi realizado com apoio do projeto "PROMETHEUS - PocketQube Framework Designed for Research and Educational Access to Space" (ref. CMU/TIC/0017/2021), suportado pela Fundação para a Ciência e a Tecnologia no âmbito da Parceria Internacional CMU Portugal.

RESUMO

Nos últimos anos tem-se assistido a um fenómeno de democratização do setor espacial devido ao surgimento de novas empresas e universidades que desenvolvem projetos relacionados com esta indústria. Contudo, verifica-se que existem ainda poucas soluções de baixo-custo que permitam acesso facilitado ao Espaço e que viabilizem uma experiência aeroespacial, através da aprendizagem e montagem de um simples satélite.

Esta dissertação visa o desenvolvimento dos principais subsistemas de uma plataforma pico-satélite, também conhecida por *PocketQube*, idealizando um sistema modular e adaptável destinado a aplicações de ensino e investigação. Esta plataforma foi baseada no sistema PyCubed (uma plataforma aviônica, de tipologia *CubeSat*, de código aberto e testada contra radiação que integra sistemas de energia, computação, comunicação e determinação e controlo de atitude num único módulo, programável inteiramente em *Python*). Isto significa que todo o sistema estará amplamente disponível, com base em componentes COTS, e também *hardware* e *software* de código aberto.

Primeiro, foi efetuada uma revisão da literatura sobre pico e nano-satélites de modo a entender como estes funcionavam, as atuais tendências relativas a estes sistemas aeroespaciais, quais os seus subsistemas e como era estabelecida a conexão entre eles, que tipo de projetos didáticos e de código aberto têm sido desenvolvidos e quais as suas principais aplicações. Segundo, definiu-se os requisitos, restrições e a estrutura modular, constituída por cinco módulos: Suporte para as Baterias, Sistema Eletrónico de Potência, Computador de Bordo, Sistema das Comunicações e Payloads. Terceiro, desenvolveu-se a arquitetura do sistema com base no projeto PyCubed, dividindo-se os componentes desse projeto pelos vários módulos da plataforma. Quarto, para garantir modularidade ao sistema, seguiu-se uma abordagem idêntica à do projeto DynOSSAT-EDU, desenvolvido pela BHDynamics, utilizando o standard PQ60, alterando apenas o conector por um de 40 pinos. Por último, foram desenvolvidas as placas dos principais subsistemas respeitando as regras de *design* de diferentes fabricantes de modo que fosse possível fabricá-las.

Em suma, os resultados do projeto desta plataforma pico-satélite podem contribuir para futuras aplicações, quer de ensino quer de investigação.

PALAVRAS-CHAVE

Pico-satélite, PocketQube, Nano-satélite, CubeSat

ABSTRACT

In recent years, there has been a phenomenon of democratization of the space sector due to the emergence of new companies and universities that are developing projects related to this industry. However, it turns out that there are few low-cost solutions that allow easy access to Space and that make an aerospace experience possible, through learning and assembling of a simple satellite.

This dissertation aims the development of the main sub-systems of a picosatellite platform, also known as PocketQube, devising a modular and adaptable system for teaching and research applications. This platform was based on the PyCubed system (an open-source, radiation tested *CubeSat* avionics platform that integrates power, computing, communication, and attitude determination and control functionality into a single module programmable entirely in Python). This means that the entire system is widely available, based on commercial-off-the-shelf components and open-source hardware and software.

First, a review of the literature on pico and nanosatellites was carried out to understand how they worked, the current trends regarding these aerospace systems, what are its subsystems and how the connection between them was established, what kind of didactic and open-source projects have been developed and what are the main applications of the picosatellites. Second, the requirements, restrictions and modular structure were defined, consisting of five modules: Battery Holder, Electronic Power System, On-board Computer, Communications System and Payloads. Third, the system architecture was developed based on the PyCubed project, dividing the components of this project into the various platform modules. Fourth, to guarantee modularity to the system, an approach identical to that of the DynOSSAT-EDU project, developed by BHDynamics, was followed, using the PQ60 standard, changing only the connector for a 40-pin one. Finally, the boards of the main subsystems were developed respecting the design rules of different manufacturers so that they could be manufactured.

Summing up, the results of this picosatellite platform project can contribute to future applications, both research and teaching.

KEYWORDS

Picosatellite, PocketQube, Nanosatellite, CubeSat.

ÍNDICE

1. Introdução	1
1.1 Enquadramento e Motivação.....	1
1.2 Objetivos	6
1.3 Perguntas de Investigação.....	6
1.4 Contribuições.....	7
1.5 Estrutura da Dissertação.....	7
2. Revisão da Literatura.....	9
2.1 Abordagem da Revisão da Literatura	9
2.2 PocketQubes	11
2.3 Subsistemas.....	21
2.4 Projetos de código aberto.....	30
2.5 Aplicações	50
3. Projeto.....	55
3.1 Visão global	55
3.2 Definição de padrão de interoperabilidade	58
3.3 Arquitetura de sistema.....	60
3.4 Projeto dos sistemas.....	64
4. Projeto de Implementação	79
4.1 Desenvolvimento das PCBs.....	79
4.2 Simulações de circuitos	89
5. Conclusão	91
5.1 Trabalho Futuro	93
Referências Bibliográficas	95
Anexo A – Esquemático OBC	102
Anexo B – Esquemático EPS.....	103

Anexo C – Esquemático COMMS	104
Anexo D – Esquemático Battery Holder	105
Anexo E – Esquemático Payloads	106

ÍNDICE DE FIGURAS

Figura 1.1 - Especificações CubeSat no âmbito das classificações gerais de pequenos satélites [5].	3
Figura 1.2 - Número de CubeSats lançados por ano.	4
Figura 1.3 - CubeSats lançados por país.	5
Figura 1.4 - CubeSats em desenvolvimento por país.	5
Figura 2.1 - Diagrama PRISMA, adaptado de [13].	10
Figura 2.2 - Unidade padrão de um PocketQube 1p [17].	11
Figura 2.3 - Sliding backplane [19].	12
Figura 2.4 - Subsistemas de um CubeSat [20].	13
Figura 2.5 - Número de PocketQubes lançados por ano.	14
Figura 2.6 - Distribuição de PocketQubes lançados de acordo com a sua configuração.	15
Figura 2.7 - Distribuição de PocketQubes em desenvolvimento de acordo com a sua configuração.	16
Figura 2.8 - Lançadores de PocketQubes.	16
Figura 2.9 - Principais Deployers de PocketQubes.	17
Figura 2.10 - PocketQubes lançados por ano e por aplicação.	18
Figura 2.11 - Descrição das Missões efetuadas.	18
Figura 2.12 - Tipo de Organização responsável pelos PocketQubes lançados.	19
Figura 2.13 - PocketQubes lançados por país.	20
Figura 2.14 - PocketQubes em desenvolvimento por país.	20
Figura 2.15 - Massa de um satélite representada pela percentagem do projeto geral [32].	22
Figura 2.16 - Esquema funcional para o EPS de uma aeronave [32].	22
Figura 2.17 - Diagrama de blocos do ADCS [36][37].	27
Figura 2.18 - Visão geral simplificada do aquecimento de um veículo espacial em órbita, adaptada de [9].	28
Figura 2.19 - ArduSat, arquitetura geral [47].	33
Figura 2.20 - Diagrama de blocos da placa principal PyCubed [47].	34
Figura 2.21 - Visão geral da placa principal PyCubed [49].	35
Figura 2.22 - Pcb da placa da bateria PyCubed [49].	36
Figura 2.23 - Kit DynOSSAT-EDU [54].	37
Figura 2.24 - Subsistemas do UPSat [56].	38
Figura 2.25 - OzQube-1 vista cortada, adaptada de [58].	40

Figura 2.26 - Educative platform based on the PocketQube 60 Specification, adaptada de [60].....	41
Figura 2.27 -Estrutura mecânica do projeto Educative platform based on the PocketQube 60 Specification [59]–[61].	42
Figura 2.28 - PocketQube Kit v1.0 [62].	43
Figura 2.29 – ESAT [65].	44
Figura 2.30 – NessieSat [67].	45
Figura 2.31 - SanoSat-1 [69].	46
Figura 2.32 - UBO PocketQube Kit [70].	47
Figura 2.33 – CubeSatSim [73].	48
Figura 2.34 - Visão geral da Observação da Terra a partir do Espaço [6].	50
Figura 3.1 - Definição dos subsistemas.	57
Figura 3.2 - Descrição dos pinos do conector PQ60 [80].	59
Figura 3.3 - Diagrama de blocos geral idealizado, adaptado de [49].	61
Figura 3.4 - Divisão dos componentes por placa.	63
Figura 3.5 – Definição dos pinos do conector PQ40.	64
Figura 3.6 - Diagrama de blocos OBC.	66
Figura 3.7 – Circuito do carregador USB para 2-células de Li-Ion.....	68
Figura 3.8 - Registo REG01.	68
Figura 3.9 - Utilização do conector RBF Header, adaptado de [49].	69
Figura 3.10 - Diagrama de blocos EPS.	70
Figura 3.11 - Esquemático do carregador solar para 2 células Li-Ion em série.....	71
Figura 3.12 - Divisor Resistivo MPPT LTC4121 [82].	72
Figura 3.13 - Circuito para programar a tensão de carregamento completo do LTC4121 [82].	73
Figura 3.14 - Conector baterias placa EPS.	74
Figura 3.15 - Utilização do conector das baterias, adaptado de [49].	74
Figura 3.16 - Diagrama de blocos COMMS.	75
Figura 3.17 - Diagrama de blocos Battery Holder.	76
Figura 3.18 - Diagrama células de bateria em Battery Holder.	77
Figura 3.19 - Circuito de aplicação típica do IC de proteção de baterias R5460, adaptado de [83].	77
Figura 3.20 - Diagrama de blocos Payloads.....	78
Figura 4.1 - Desenho técnico do standard PQ60 [80].	80

Figura 4.2 – Design das 4 camadas de PCB do sistema PyCubed [48].	80
Figura 4.3 – Layout da placa OBC: (a) Camada F.Cu pcb OBC; (b) Camada In1.Cu pcb OBC; (c) Camada In2.Cu pcb OBC; (d)Camada B.Cu pcb OBC.	83
Figura 4.4 - Visão 3D da (a) parte frontal da placa OBC e (b) da parte traseira da placa OBC.	83
Figura 4.5 – Layout da placa EPS: (a) Camada F.Cu pcb EPS; (b) Camada In1.Cu pcb EPS; (c) Camada In2.Cu pcb EPS; (d) Camada B.Cu pcb EPS.	84
Figura 4.6 - Visão 3D da (a) parte frontal da placa EPS e (b) parte traseira da placa EPS.	85
Figura 4.7 – Layout da placa COMMS: (a) Camada F.Cu pcb COMMS; (b) Camada B.Cu pcb COMMS.	86
Figura 4.8 - Visão 3D da (a) parte frontal da placa COMMS e (b) parte traseira da placa COMMS.	86
Figura 4.9 – Layout da placa Battery Holder: (a) Camada F.Cu pcb Battery Holder; (b) Camada B.Cu pcb Battery Holder.	87
Figura 4.10 - Visão 3D da (a) parte frontal e (b) da parte traseira da placa Battery Holder.	88
Figura 4.11 – Layout da placa Payloads: (a) Camada F.Cu pcb Payloads; (b) Camada B.Cu pcb Payloads.	88
Figura 4.12 - Visão 3D da (a) parte frontal e (b) parte traseira da placa Payloads.	89
Figura 4.13 - Esquemático Regulador 3,3V para simulação.	90
Figura 4.14 - Simulação regulador 3,3V.	90

ÍNDICE DE TABELAS

Tabela 1 - Dimensões externas do PocketQube e dimensões da sliding backplate para diferentes números de unidades [19].	12
Tabela 2 - Tabela comparativa de projetos.	49
Tabela 3 - Regras de design dos fabricantes pcbway, jlpcb e eurocircuits.	81

LISTA DE ABREVIATURAS, SIGLAS E ACRÓNIMOS

AC	Alternating Current	Corrente Alternada
ACS	Attitude Control System	Sistema de Controlo de Atitude
ADC	Analog Digital Converter	Conversor Analógico para Digital
ADCS	Attitude Determination and Control System	Sistema de Controlo e Determinação de Atitude
ADS	Attitude Determination System	Sistema de Determinação de Atitude
ARM	Advanced RISC Machine	
ASPPM	ArduSat Payload Processor Module	Módulo Processador de Payload do ArduSat
CC	Constant Current	Corrente Constante
COMMS	Communications System	Sistema de Comunicações
COTS	Commercial Off-The-Shelf	Comercial pronto para uso
CV	Constant Voltage	Tensão Constante
CVCM	Collected Volatile Condensable Material	Material condensável volátil coletado
DAC	Digital Analog Converter	Conversor Digital para Analógico
DC	Direct current	Corrente Continua
DIY	Do It Yourself	Faz tu mesmo
DSN	Deep Space Network	Rede Espacial Profunda
ECSS	European Cooperation on Space Standardization	Cooperação Europeia em Padronização Espacial
EMI	Electromagnetic interference	Interferência Eletromagnética
EPS	Electronic Power System	Sistema Eletrónico de Potência
FPGA	Field Programmable Gate Array	
GAUSS	Group of Astrodynamics for the Use of Space Systems	
GPIO	General Purpose Input Output	Pinos de Entrada e Saída para uso geral

GPS	Global Positioning System	Sistema de Posicionamento Global
HW	Hardware	
IAC	Image Acquisition Component	Componente de Aquisição de Imagem
IARU	The International Amateur Radio Union	União Internacional de Radioamadorismo
IEEE	Institute of Electrical and Electronics Engineers	Instituto de Engenharia Elétrica e Eletrónica
IMU	Inertial Measurement Unit	Unidade de Medição Inercial
IoT	Internet of Things	Internet das Coisas
LEO	Low Earth Orbit	Órbita Terrestre Baixa
MEMS	Micro ElectroMechanical Systems	Sistemas Micro Eletromecânicos
MPPT	Maximum Power Point Tracking	Seguidor do ponto de potência máxima
MRAM	Magnetoresistive random-access memory	Memória não-volátil
MRFOD	Morehead Rome Femtosatellites Orbital Deployer	
MSP	Mixed Signal Processing	Processamento de sinais misturados
MSU	Morehead State University	
OBC	On Board Computer	Computador de Bordo
ODCS	Orbit Determination and Control System	Controlo e Determinação de Órbita
OSHW	Open Source Hardware	Hardware de código aberto
OSHWA	Open Source Hardware Association	Associação de hardware de código aberto
OSI	Open Source Initiative	Iniciativa de código aberto
OSS	Open Source Software	Software de código aberto
PCB	Printed Circuit Board	Placa de Circuito Impresso
PI	Research Questions	Perguntas de Investigação
PIC	Peripheral Interface Controller	
PQ60	PocketQube60	

PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses	Itens de relatório preferidos para revisões sistemáticas e meta-análises
SI	International System of Units	Sistema Internacional de Unidades
SPI	Serial Peripheral Interface	
STEM	Science Technology Engineering and Mathematics	Ciência Tecnologia Engenharia e Matemática
SU	Science Unit	Unidade de Ciência
SW	Software	
TCS	Thermal Control System	Subsistema de Controlo Térmico
TML	Total Mass Lost	Massa Total Perdida
UART	Universal Asynchronous Receiver-Transmitter	
UHF	Ultra High Frequency	
USB	Universal Serial Bus	
VHF	Very High Frequency	

1. INTRODUÇÃO

Esta dissertação de Mestrado consiste no projeto e no desenvolvimento dos principais subsistemas de um pico-satélite para aplicações de ensino e de investigação.

1.1 Enquadramento e Motivação

Atualmente o Espaço está ao alcance de todos, como resultado do que hoje se denomina democratização do setor espacial. Este fenómeno de democratização, denominado muitas vezes de *NewSpace*, refere-se ao surgimento da indústria dos voos espaciais privados e de outras iniciativas. Na última década, a entrada de empresas privadas veio revolucionar esta indústria e, por conseguinte, acelerar o desenvolvimento de um setor que, por sua vez, tem um elevado potencial. Aquilo que, durante muitos anos, esteve restrito apenas a instituições governamentais [1]. De acordo com *SpaceTech Partners* [1], *NewSpace* é o surgimento da indústria de voos espaciais privados. Novas iniciativas espaciais são criadas crescentemente, como empresas de lançamento privadas, pequenas constelações de satélites ou turismo suborbital, assim como esforços mais específicos para reinventar a cadeia de valor da indústria espacial tradicional [1].

Nos dias de hoje são inúmeras as iniciativas que permitem realizar experiências em ambientes de microgravidade. Contudo, o Espaço ainda é visto como um ambiente de elevados riscos latentes. No sector de investigação e de ensino, estes riscos representam custos elevados e nem sempre suportáveis, dificultando o desenvolvimento de projetos práticos na temática aeroespacial. Efetivamente, verifica-se que existem poucas soluções de baixo-custo que permitam acesso facilitado ao Espaço e que viabilizem uma experiência aeroespacial, através da aprendizagem e montagem de um simples satélite. Esta

barreira é crítica e limita as oportunidades de Investigação e Ensino, em virtude de falta de experiência neste ambiente.

No entanto, com os recentes avanços tecnológicos que se têm vindo a presenciar em todos os setores da ciência, um dos mais impulsionados foi o do domínio do Espaço [2], onde uma das particularidades tem sido o estudo e o crescente aumento e avanço da miniaturização de sistemas. Estes recentes avanços na tecnologia de miniaturização geraram um aumento no interesse da indústria espacial, especialmente em pequenas missões. Resultando assim no desenvolvimento de pequenas aeronaves prontas a utilizar, com baixo consumo de potência e baseadas em componentes *comercial-off-the-shelf* (COTS) [3]. Sendo uma das principais vantagens o custo de produção e utilização inferior à das aeronaves tradicionais. Continuando esta tendência de miniaturização de aeronaves, pequenos satélites foram utilizados para demonstrações de capacidades tecnológicas e também no Ensino [3]. Consequentemente, no setor da educação, várias universidades, por todo o mundo, começaram programas de nano-satélites, mais conhecidos por *CubeSats*, para ensinar aos seus estudantes os fundamentos de engenharia e de gestão de projetos num satélite, bem como proporcionar uma experiência prática [3]. A ideia foi proporcionar aos estudantes uma experiência em atividades espaciais, permitindo-lhes trabalhar em todo o ciclo de um projeto espacial, desde o conceito inicial e definição da missão até à sua operação no espaço [4].

O padrão *CubeSat* foi criado em 1999 pelo Professor Jordi Puig-Suari e pelo Professor Bob Twiggs, na *California State University* e na *Stanford University*, respetivamente [3] [5]. O termo *CubeSat* é utilizado para descrever um satélite miniaturizado, em que a sua Unidade padrão, denominada 1U é um cubo com 10 cm de lado, ou seja, um cubo $10 \times 10 \times 10 \text{ cm}^3$ com uma massa não superior a 1,33 Kg [3]. Estes poderiam ser utilizados singularmente, isto é, na sua unidade padrão 1U ou então de forma agregada com várias unidades para combinar num satélite de maiores dimensões e, porventura, com maior potencialidade para desempenhar mais capacidades, tais como 2U, 3U, 6U, e assim por diante (Figura 1.1)[5].

A motivação para o desenvolvimento inicial dos *CubeSats* foi a vertente educacional [4]. A ideia era fornecer uma experiência *hands-on* aos estudantes em atividades relacionadas com o Espaço, permitindo-lhes integrar num ciclo de trabalho real de um projeto espacial [4]. Como estas entidades

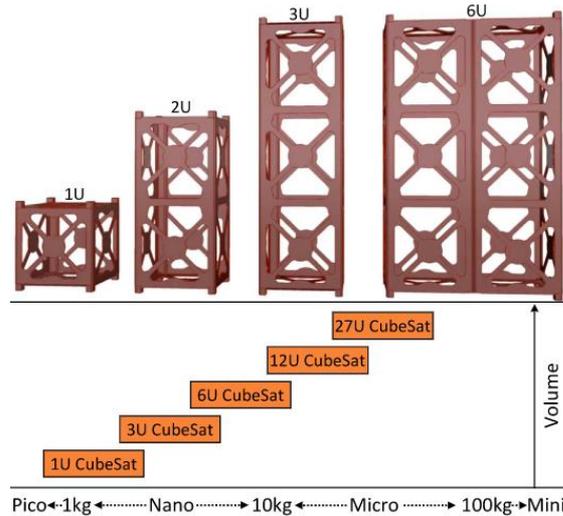


Figura 1.1 - Especificações CubeSat no âmbito das classificações gerais de pequenos satélites [5].

não dispunham de orçamentos tão alargados, era essencial que o desenvolvimento, lançamento e operação fossem mantidos ao preço mais baixo possível, o que levou também à padronização desta mesma estrutura [4]. A fim de alcançar a redução de custo do projeto pretendida, a utilização de componentes COTS foi também incentivada [4]. Uma das principais vantagens desta padronização foi permitir aos fornecedores de lançadores, veículos responsáveis por lançar os satélites, adotarem um sistema de desenvolvimento independente do fabricante do *CubeSat*, mas compatível com todos eles [6]. Estes pequenos satélites diminuíram consideravelmente o custo e a complexidade de desenvolvimento e lançamento em comparação com os satélites tradicionais, resultando então, na sua disseminação e num aumento do lançamento de pequenos satélites ao longo da última década [6]. Adicionalmente, possíveis falhas individuais durante o desenvolvimento do projeto de pequenos satélites irão ter um impacto bastante inferior que falhas ocorridas nos maiores e mais sofisticados, dado o seu baixo custo e curto tempo de desenvolvimento, visto que os satélites tradicionais têm orçamentos na ordem dos milhões de dólares e períodos de desenvolvimento entre 5 e 10 anos [6].

Os *CubeSats*, como foi referido anteriormente, foram inicialmente imaginados como plataformas educacionais ou de demonstrações de tecnologia que poderiam ser desenvolvidos e lançados num intervalo de 1 a 2 anos [7]. Contudo, missões avançadas recentes com *CubeSats* têm sido desenvolvidas e propostas, indicando que estes começaram claramente uma transição desde plataformas de ensino e de demonstração de tecnologias para missões reais de baixo-custo e com elevado valor potencial em termos de retorno científico e comercial [3].

Originalmente, os *CubeSats* foram desenvolvidos por instituições acadêmicas e apenas uma pequena porção dos satélites foram lançados pelo setor comercial [6]. Esta tendência tem vindo a mudar ao longo dos anos, mas mesmo assim, as instituições acadêmicas ainda fazem parte de uma grande fração de todos os *CubeSats* desenvolvidos e lançados [6].

De acordo com a plataforma *online Gunter's Space Page* [8], é possível analisar a evolução dos lançamentos de *CubeSats* (em diferentes combinações) desde o ano 2003 (Figura 1.2). Nesta altura, em meados dos anos 2000, eram apenas vistos como satélites para complementar a formação dos estudantes ou para corresponder a algumas exigências amadoras [4].

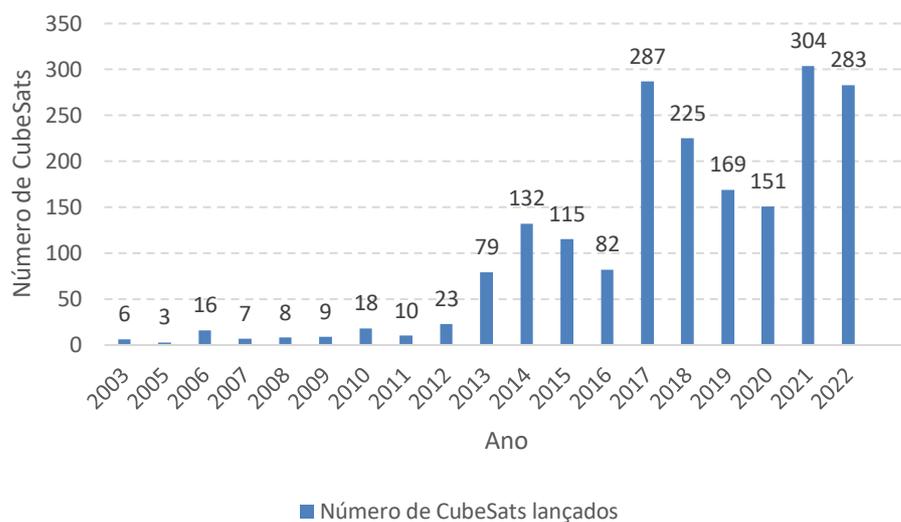


Figura 1.2 - Número de CubeSats lançados por ano.

Estes lançamentos incluem lançamentos falhados e satélites que por alguma razão caíram ou então desconhece-se a localização. Por outro lado, alguns ainda estão em órbita. É de salientar uma desaceleração nos lançamentos em 2019 e 2020, causada provavelmente pela pandemia COVID-19, que resultou num atrasado cumulativo do calendário de lançamentos [9]. Em 2023 espera-se o lançamento de 92 *CubeSats* [8].

Na Figura 1.3 e na Figura 1.4 são exibidas, através do mapa mundo, as taxas de *CubeSats* lançados e em desenvolvimento por país desde 2003 até a atualidade, respetivamente. Em ambas as figuras é possível entender que os Estados Unidos da América diferenciam-se com clara distinção, quer nos lançamentos efetuados quer nos projetos em desenvolvimento.

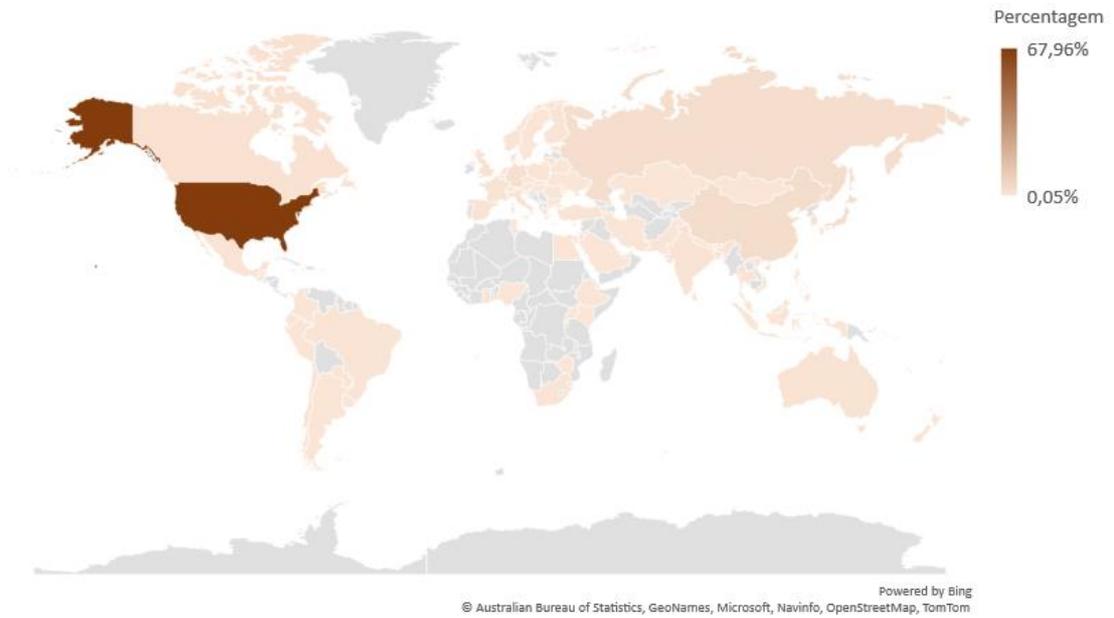


Figura 1.3 - CubeSats lançados por país.

De acordo com a base de dados [10], existem 4 *CubeSats* a serem desenvolvidos em Portugal.

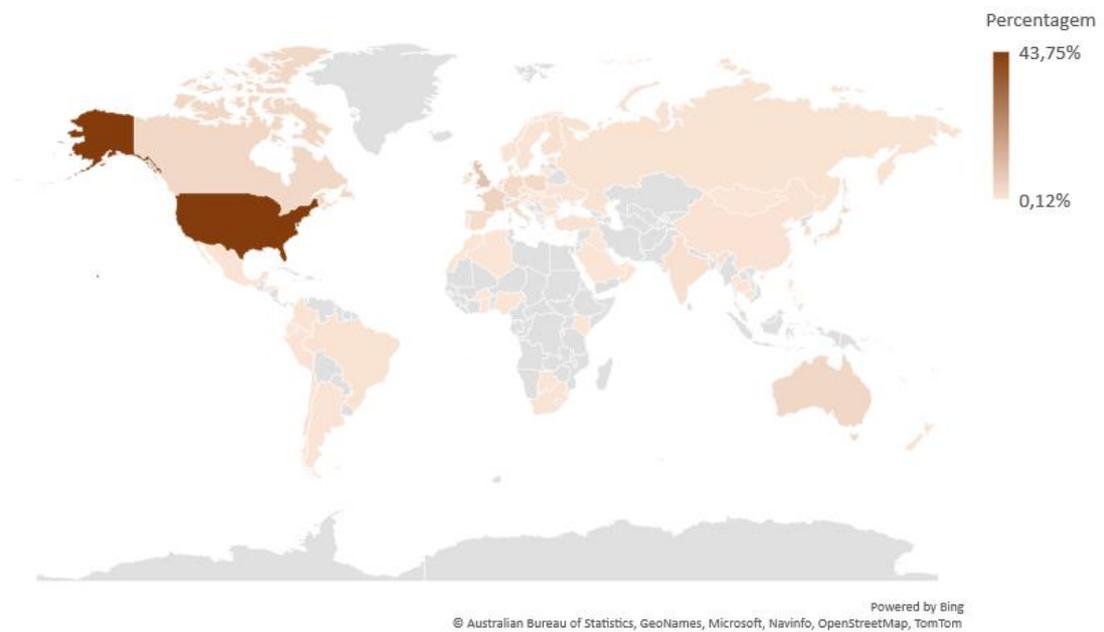


Figura 1.4 - CubeSats em desenvolvimento por país.

Na tentativa de democratizar ainda mais os *CubeSats*, os *PocketQubes*, também conhecidos por pico-satélites, foram propostos como uma variante evoluída deles, onde era pretendido reduzir ainda mais o tamanho e os custos do sistema [11]. Tanto os *CubeSats* como os *PocketQubes* operam

maioritariamente em *Low Earth Orbit* (LEO), uma vez que quando lançados são usados principalmente para missões de observação da Terra e demonstração de tecnologias [4][12]. Contudo, devido às restrições de dimensões que os *PocketQubes* possuem, estes não têm tanto espaço para integrar componentes ou *payloads* tal como os *CubeSats* têm. Desta forma, são utilizados em âmbitos diferentes, optando-se assim pelo uso de *PocketQubes* quem estiver interessado numa vertente mais educativa, tal como este projeto.

1.2 Objetivos

Esta dissertação tem como principal objetivo o desenvolvimento de uma plataforma *PocketQube*, que irá consistir numa ferramenta pedagógica-científica, ou seja, uma plataforma *hands-on* vocacionada para uma vertente educativa e que também poderá ser utilizada para investigação. Para tal, será necessário avaliar plataformas de código aberto para pequenos satélites, e tendo por base as mesmas idealizar uma solução para um *PocketQube* que seja suficientemente flexível e adaptável para poder ser usada posteriormente como base de aviónica para diferentes tipos de *payloads*.

Com base neste objetivo geral, é possível especificar alguns outros objetivos secundários que permitirão realizar o objetivo final. Primeiramente, deverá ser estudado o estado de arte de *CubeSats* e *PocketQubes* em código aberto, de forma a concluir qual o que mais se adequa às características do objetivo pretendido.

Uma análise dos requisitos, conceitos e processos para desenvolver um *PocketQube* é uma outra etapa que tem como finalidade definir qual a estrutura e componentes que se aplicam melhor às necessidades e capacidades de um pequeno satélite. Com a análise concluída, segue-se então para a idealização dos principais subsistemas de um *PocketQube*.

No final da execução do projeto, espera-se que este resulte num sistema modular, constituído pelos principais subsistemas de um pico-satélite, e adaptável de modo que qualquer pessoa o consiga utilizar e adaptar às suas necessidades, mais propriamente *payloads*.

1.3 Perguntas de Investigação

Ao longo desta dissertação foram abordadas várias perguntas de investigação (PI), tais como:

- **PI 1:** Quais as atuais tendências dos nano e pico-satélites?

- **PI 2:** Quais os sistemas essenciais num satélite?
- **PI 3:** Como garantir um desenho modular para um *PocketQube*?
- **PI 4:** Existe algum *standard* para plataformas *PocketQube*?
- **PI 5:** Como garantir adaptabilidade do *payload*?

1.4 Contribuições

Esta dissertação contribui com o projeto dos principais subsistemas de um pico-satélite, fornecendo uma plataforma modular e adaptável. Esta solução inclui o projeto e o desenvolvimento dos vários subsistemas.

As principais contribuições desta dissertação são as seguintes:

- Revisão da literatura sobre *CubeSats* e *PocketQubes*;
- Análise estatística sobre *CubeSats* e *PocketQubes*;
- Revisão descritiva dos vários subsistemas que constituem um satélite;
- Análise sobre os vários projetos de código aberto e *kits* didáticos existentes relacionados com pequenos satélites;
- Revisão das potenciais aplicações dos *PocketQubes*;
- Definição de um padrão de interoperabilidade para *PocketQubes*;
- Projeto e Desenvolvimento dos subsistemas para pico-satélites.

1.5 Estrutura da Dissertação

Esta dissertação está organizada em cinco capítulos.

O presente capítulo introduz o tema do trabalho e faz o enquadramento da sua pertinência e quais as principais questões de investigação que se procuram responder.

O Capítulo 2 apresenta uma revisão da literatura sobre os *CubeSats* e *PocketQubes*, onde é feita uma análise estatística sobre quantos já foram lançados. São também abordados os seus subsistemas

e potenciais aplicações, bem como é feita uma análise aos projetos de código aberto e *kits* didáticos de pequenos satélites.

O Capítulo 3 é dedicado ao projeto do sistema, onde é apresentada uma visão geral da arquitetura do sistema, uma definição de padrão de interoperabilidade e o projeto dos vários módulos que constituem esta plataforma *PocketQube*.

O Capítulo 4 apresenta o desenvolvimento das *PCBs* das várias placas do sistema e algumas simulações de circuitos de potência.

O Capítulo 5 diz respeito às conclusões desta dissertação e também ao trabalho futuro.

2. REVISÃO DA LITERATURA

Neste capítulo uma revisão da literatura é apresentada. A sua abordagem é discutida no primeiro tópico. Nos seguintes são apresentados conceitos úteis para o desenvolvimento deste projeto, assim como para a sua compreensão.

2.1 Abordagem da Revisão da Literatura

Inicialmente foi efetuada uma pesquisa acerca do tema da dissertação, visando conhecer os tópicos e escolher as palavras-chave relacionadas com o tema. Sendo que as eleitas foram “CubeSats” e “PocketQubes”.

Com as palavras-chave selecionadas, decidiu-se que a metodologia a ser utilizada para elaborar a revisão da literatura seria a abordagem PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*)[13]. Posto isto, a 18 de novembro de 2021 procurou-se na base de dados *Scopus* artigos de revisão com a palavra-chave “CubeSat”, ou então presente no título ou no resumo, cujo resultado foram 42 artigos. No mesmo dia e utilizando o mesmo método, alterou-se a pesquisa para “PocketQube”, resultando em 38 artigos [14].

Posteriormente, verificou-se com atenção os artigos e eliminou-se os que não correspondiam ao tópico inicialmente delineado ou então eram documentos duplicados. A partir daí, a leitura dos artigos tornou-se mais profunda enquanto se retirava informação deles. Esta informação foi sendo cuidadosamente separada em vários tópicos através da ferramenta *Mendeley*. Assim que toda a informação foi repartida, combinou-se tudo e fez-se um resumo que respondesse às perguntas dos tópicos.

Logo que se decidiu que a informação já respondia às perguntas, elaborou-se o documento onde se colocou o que foi feito até ao momento para solucionar o problema inicialmente proposto.

No dia 7 de dezembro de 2022 foi efetuada outra pesquisa na plataforma *scopus*. Procurou-se documentos com a palavra “cubesat” presente ou no título, ou no *abstract* ou nas *keywords*. Foram encontrados 5263 documentos. Limitando a pesquisa apenas a artigos de revisão e em inglês, restaram 47. Comparando com a pesquisa anterior, o número de novos documentos seria de cinco, no entanto foram realmente publicados mais sete artigos. Sendo que dois dos artigos encontrados na primeira pesquisa foram retirados do *scopus*. No mesmo dia, foi também feita uma pesquisa com a palavra “pocketqube”. Resultando em 6 novos artigos publicados desde a última pesquisa, ou seja, publicados em 2022. No dia 20 de dezembro de 2022 foi efetuada uma pesquisa no *scopus* por “Picosatellite”, resultando em 221 artigos.

Ao longo da realização do projeto também foram encontrados vários estudos via outros métodos, tais como *websites* e organizações. Na Figura 2.1 é possível observar o diagrama PRISMA, utilizado para identificar e selecionar os artigos para revisão e posteriormente inclusão na revisão da literatura da dissertação.

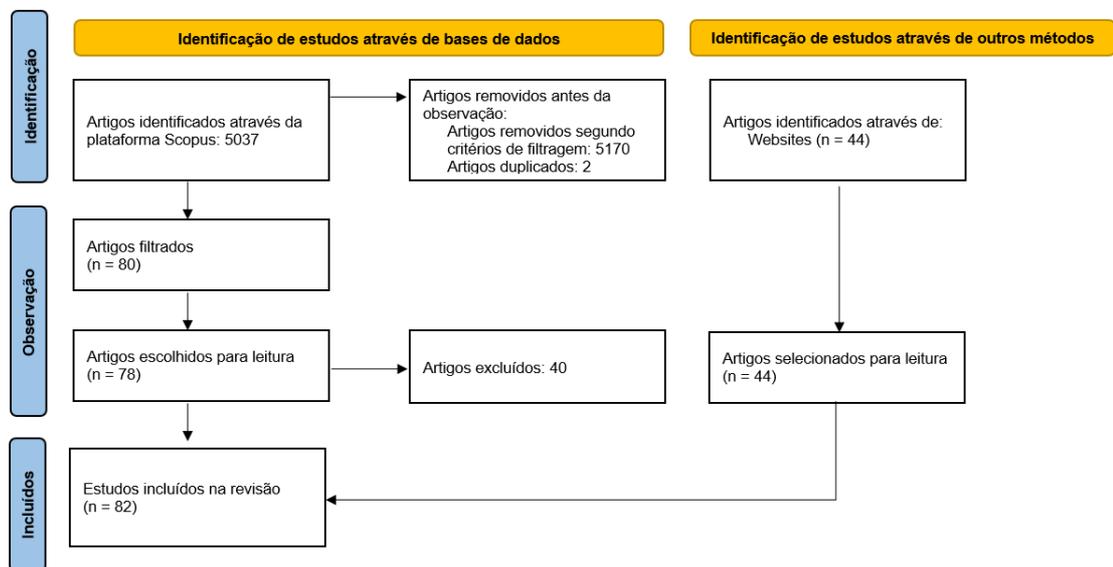


Figura 2.1 - Diagrama PRISMA, adaptado de [13].

2.2 PocketQubes

Os *PocketQubes* foram apresentados e propostos em 2009 pelo Professor Robert J. Twiggs numa colaboração entre a *Morehead State University* (MSU) e *Kentucky Space* [11]. Twiggs acreditava que os *PocketQubes* seriam uma ferramenta chave para reativar a educação em Ciência, Tecnologia, Engenharia e Matemática (STEM) em escolas do 2º e 3º ciclos do ensino básico nos Estados Unidos e por todo o mundo [11]. A sua denominação inicial era Pico-satélites, mas como eram de pequenas dimensões foi-lhes atribuído também o nome de *PocketQube* que, tal como o seu nome indica, este pequeno satélite era capaz de ser colocado no bolso das calças [11]. A sua unidade padrão é o 1p (Figura 2.2) e, tal como os *CubeSats*, assumem um formato cúbico. Porém, a plataforma dos *PocketQubes* tem 5 cm de lado, ou seja, um cubo $5 \times 5 \times 5 \text{ cm}^3$ com uma massa de aproximadamente 250 g [11][15]. Como são considerados pico-satélites podem variar no tamanho levando a uma alteração da sua massa entre 0,1 Kg e 1 Kg [16]. Em termos de comparação, cabem aproximadamente 8 *PocketQubes* dentro de um *CubeSat*, uma vez que o volume de um pico-satélite 1p é 125 cm^3 e o de um nano-satélite 1U é 1000 cm^3 [11].

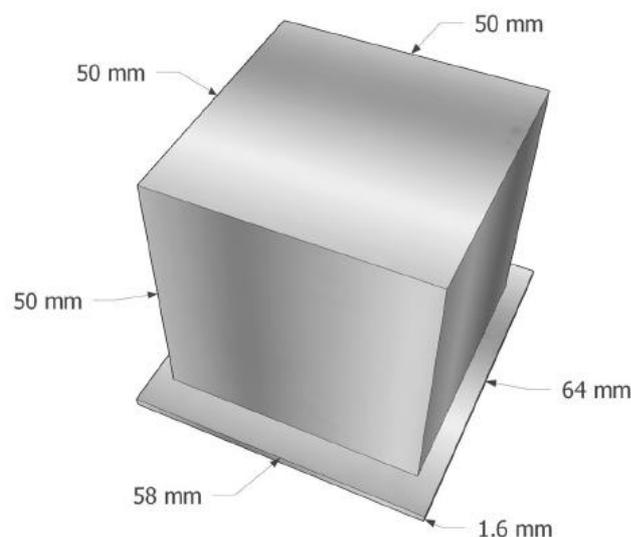


Figura 2.2 - Unidade padrão de um PocketQube 1p [17].

Uma colaboração entre a MSU e a GAUSS Srl. (*Group of Astrodynamics for the Use of Space Systems*) levou ao desenvolvimento de um sistema de lançamento de *PocketQubes* dedicado, denominado MRFOD (*Morehead Rome Femtosatellites Orbital Deployer*)[17][18]. Em vez de utilizar os cantos para deslizar no sistema de lançamento, como funciona nos *CubeSats*, é usada uma placa de

base que se estende mais uns milímetros da estrutura principal, que por sua vez desliza nos carris que o *deployer* possui [17], denominada *sliding backplate*, visível na Figura 2.3 [19].

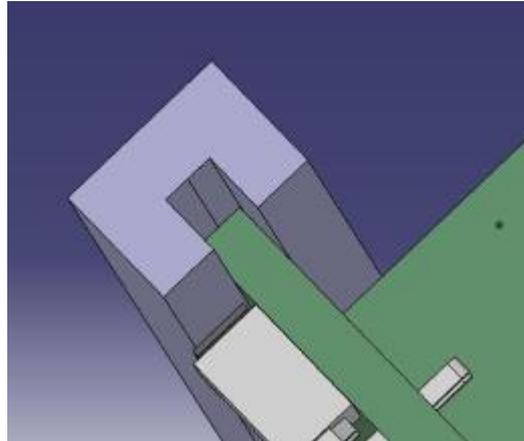


Figura 2.3 - *Sliding backplate* [19].

As dimensões externas dos *PocketQubes* e da *sliding backplate* são ligeiramente diferentes dependendo do número de unidades, como se pode observar na Tabela 1 [19].

Tabela 1 - Dimensões externas do *PocketQube* e dimensões da *sliding backplate* para diferentes números de unidades [19].

Número de Unidades (P)	Dimensões externas sem a <i>backplate</i> (mm)	Dimensões da <i>sliding backplate</i> (mm)
1P	50x50x50	58x64x1,6
2P	50x50x114	58x128x1,6
3P	50x50x178	58x192x1,6

De acordo com o *standard PocketQube* [19], estes devem utilizar no mínimo dois *kill switches* (muitas vezes montados com base em fins-de-curso) para manter o satélite desligado enquanto estiver no *deployer*. Estes devem estar colocados em contacto com os carris do lançador ou então com outro *PocketQube* para que, assim que for lançado, o *PocketQube* seja ligado [19]. Posto isto, depois do desenvolvimento do satélite, se a intenção for lançá-lo, é necessário recorrer a um intermediário, também conhecido por *broker*. Estes oferecem a oportunidade de colocar estes pequenos satélites em órbita pois possuem *deployers* que vão a bordo dos lançadores. Os *deployers* têm um limite de espaço e por isso, quem pretender lançar um satélite, basta fazer o aluguer de um lugar para ele, mediante os custos do

broker portador do *deployer*. No entanto, é necessário que os *PocketQubes* antes de ser cedidos ao *broker* tenham efetuado pelo menos os testes de vibração e de vácuo, para garantir que a missão decorra como planeado pelo menos durante a fase de lançamento, e com risco reduzido para os restantes *PocketQubes* com quem é partilhado o *deployer*.

Relativamente às estruturas base de um *PocketQube*, foram abordados vários projetos com o objetivo de entender quais são as fundamentais para o seu funcionamento. Verificou-se que eram idênticas às utilizadas no *CubeSats*, mas de menor tamanho.

Assim as estruturas base dividem-se em:

- EPS (*Electronic Power System*);
- OBC (*On Board Computer*);
- COMMS (*Communications System*);
- ADCS (*Attitude Determination and Control System*)
- Payloads.

Na Figura 2.4 é possível observar uma vista explodida de um *CubeSat*, mas que tem as mesmas estruturas base de um *PocketQube* [20].

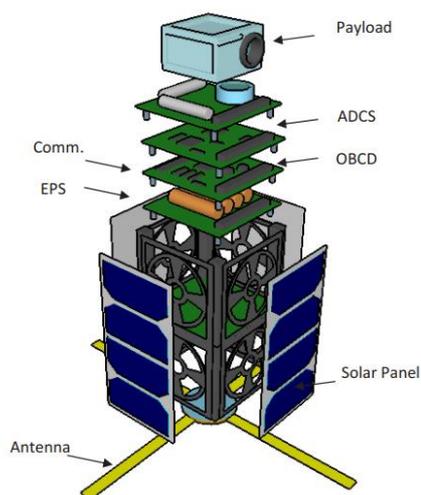


Figura 2.4 - Subsistemas de um CubeSat [20].

De acordo com as bases de dados [10], [21], [22], foi realizada uma análise estatística referente aos *PocketQubes* já lançados e aos que ainda estão em fase de desenvolvimento.

Primeiramente, analisou-se o número de *PocketQubes* que já foram lançados por ano resultando na Figura 2.5. Começando em 2013, ano em que foi lançado o *rocket Dnepr* contendo 32 satélites de

estudantes, organizações e agências governamentais [23]. Sendo que, destes 32, quatro deles eram *PocketQubes* e viriam a ser os primeiros da sua espécie a entrar em órbita [23]. Designados por \$50SAT, Wren, Beakersat-1 (depois do lançamento foi renomeado para T-LogoQube) e o QBScout-1 foram então os primeiros pico-satélites a serem lançados. A bordo do Dnepr foram ejetados pelo UniSat-5, satélite desenvolvido pela GAUSS Srl, utilizando um sistema de lançamento orbital chamado MRFOD, referido no início desta secção 2.2 [24].

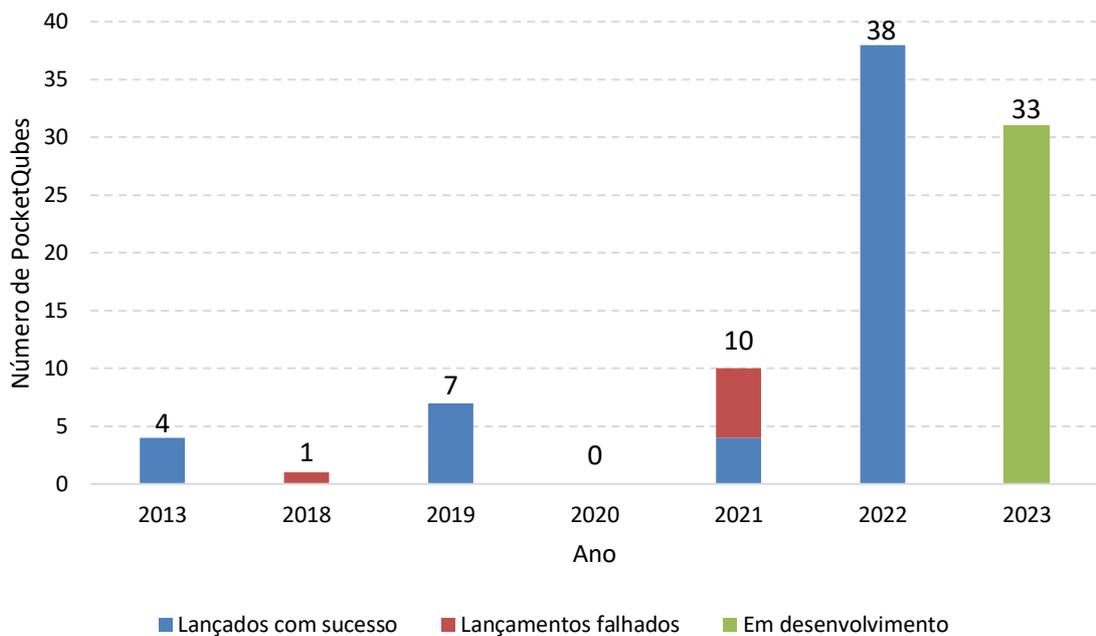


Figura 2.5 - Número de PocketQubes lançados por ano.

Da análise da Figura 2.5 é possível verificar que nem todos os lançamentos efetuados foram bem-sucedidos. Em 2018, um lançador desenvolvido pela *University of British Columbia*, denominado UBC Black Tusk, foi lançado. Segundo informação encontrada nas plataformas sociais da UBC Rocket, publicada no dia 19 de junho de 2018, um dos seus objetivos era transportar *PocketQubes* até 30.000 pés, ou seja, cerca de 9,14 km. No entanto, o lançamento foi uma tentativa falhada pois quando este atingiu uma altitude de aproximadamente 12.000 pés efetuou subitamente uma curva inesperada e acabou por se despenhar. Apesar de não existir muita informação sobre ele, segundo a base de dados [21], um pico-satélite denominado por Astria ia a bordo desse mesmo lançador, tendo-se igualmente despenhado e não conseguindo entrar em órbita. Já no ano de 2021, 6 dos 10 pico-satélites lançados não conseguiram entrar em órbita, uma vez que o seu veículo de lançamento (Firefly Alpha) explodiu alguns minutos após ter sido lançado [25]. Esta seria a primeira tentativa de lançamento deste *rocket* lançador, no entanto, a 1 de outubro de 2022 ocorreu o seu segundo teste de voo e alcançou a órbita com sucesso, conseguindo implantar 5 *PocketQubes* e 2 *CubeSats* [10]. Mesmo assim, alguns dos

satélites não conseguiram cumprir com o seu objetivo e acabaram por decair alguns dias depois, devido ao facto de terem sido libertados em órbitas mais baixas do que planeado originalmente [26]. No ano de 2023, estima-se que 33 *PocketQubes* estão em desenvolvimento, 31 previstos serem lançados nesse mesmo ano e 2 no ano de 2025 [10]. Salienta-se que como no início a taxa de insucesso era grande, começou-se a lançar mais *PocketQubes*, pois se algo corresse mal, o investimento também era menor.

A Figura 2.6 demonstra a distribuição de *PocketQubes* que já foram lançados de acordo com a sua configuração. As configurações mais utilizadas foram 1p (35%) e 2p (32%) [21]. Por outro lado, a Figura 2.7 exhibe uma possível distribuição dos pico-satélites ainda em desenvolvimento. Verifica-se uma tendência para desenvolver satélites 1p, com cerca de 61% dos projetos em curso.

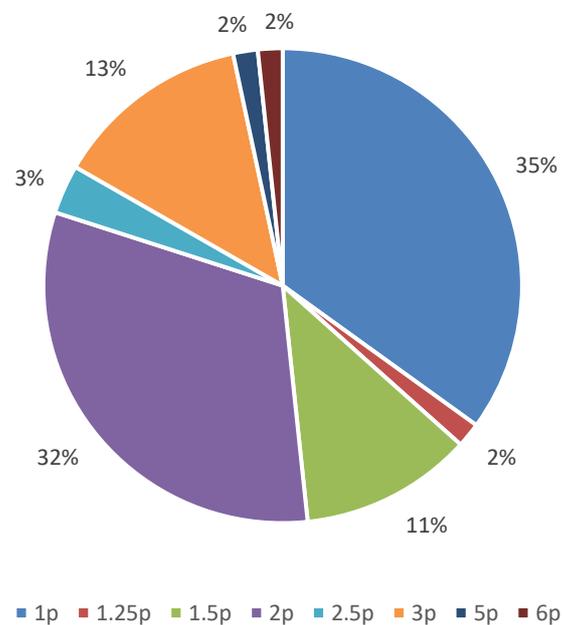


Figura 2.6 - Distribuição de *PocketQubes* lançados de acordo com a sua configuração.

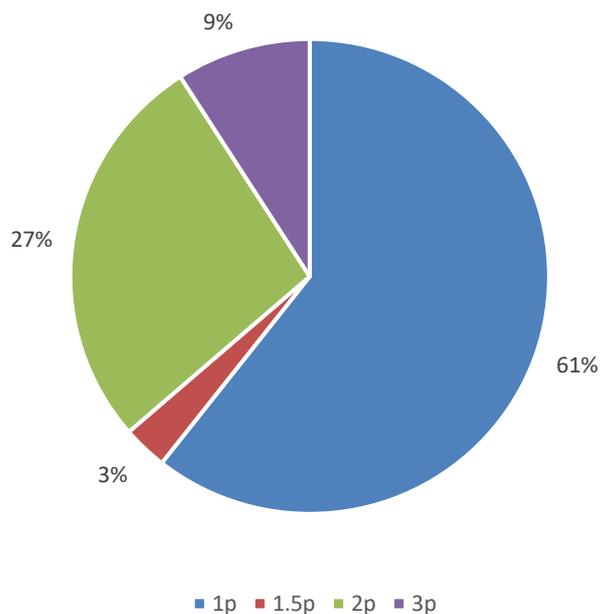


Figura 2.7 - Distribuição de PocketQubes em desenvolvimento de acordo com a sua configuração.

A Figura 2.8 e a Figura 2.9 mostram, respetivamente, as distribuições de lançadores de *PocketQubes* já utilizados em missões e os principais *deployers*. É possível verificar que o lançador Falcon 9, produzido pela Space X, é responsável pela maioria dos lançamentos, com cerca de 48% dos *PocketQubes* já lançados, ou seja, 29 satélites.

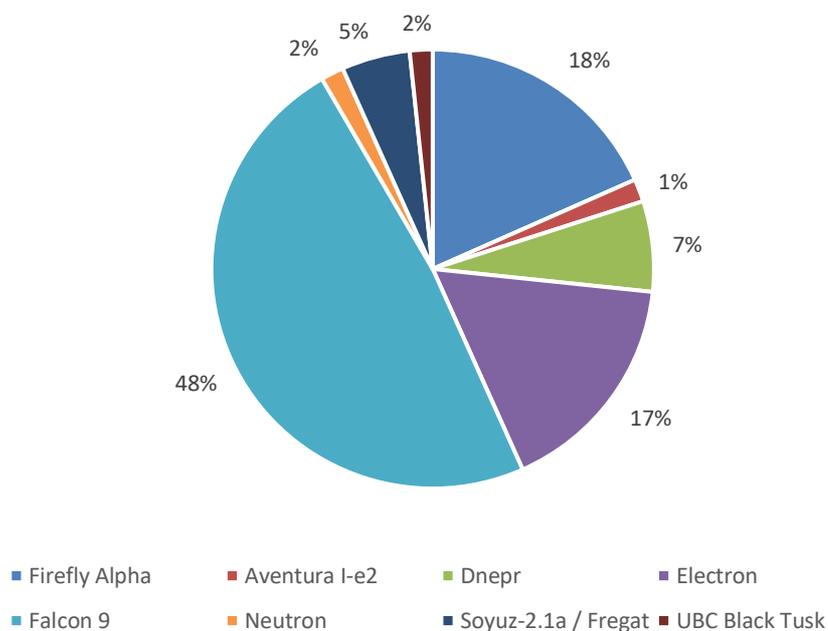


Figura 2.8 - Lançadores de PocketQubes.

Relativamente aos *deployers*, como se pode observar na Figura 2.9, os dois principais são o PocketPod, da intermediária FOSSA.Systems, e o Albapod da Alba Orbital.

O PocketPod é um *deployer* 8p projetado para colocar *PocketQubes* em órbitas LEO e, segundo consta da *webpage* da FOSSA Systems, é atualmente a solução mais competitiva economicamente para lançar satélites para o Espaço [27]. O custo para obter lugar para um pico-satélite com a configuração 1p é de 18.000 € [27].

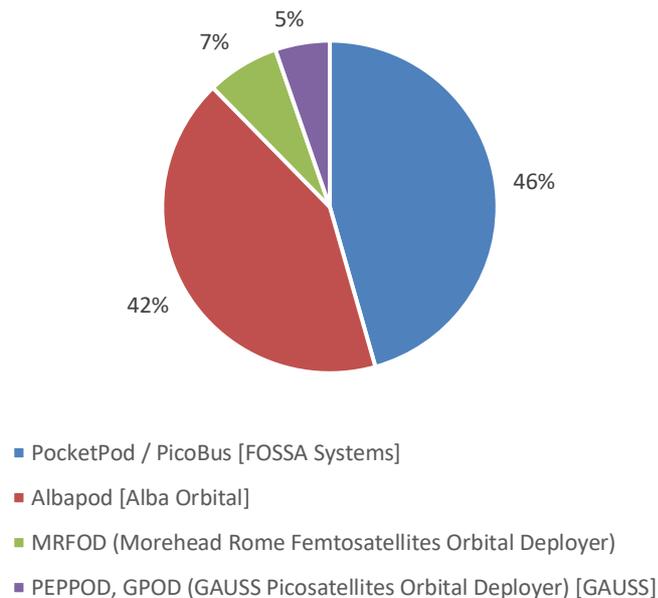


Figura 2.9 - Principais *Deployers* de *PocketQubes*.

O AlbaPod é um *deployer* desenvolvido pela Alba Orbital e o seu *standard* foi projetado para ter uma capacidade de suportar 6p de *PocketQubes*. No entanto, foram concebidos vários AlbaPod que são capazes de implantar até 96p [28]. O custo para obter lugar para um pico-satélite com a configuração 1p é de 25.000 € [28].

A respeito das aplicações às quais os *PocketQubes* lançados se empregaram, era espectável que uma das principais seria a vertente educacional, pois essa foi uma das causas pela qual eles foram desenvolvidos, como referido no capítulo 1.1. Através da análise da Figura 2.10 pode-se apurar que, efetivamente, uma das principais aplicações foi a educação, aumentando o número de projetos ano após ano. A demonstração de tecnologia também representa uma importante fração do número total de *PocketQubes* lançados. No capítulo 2.5 são abordadas as várias aplicações de forma detalhada.

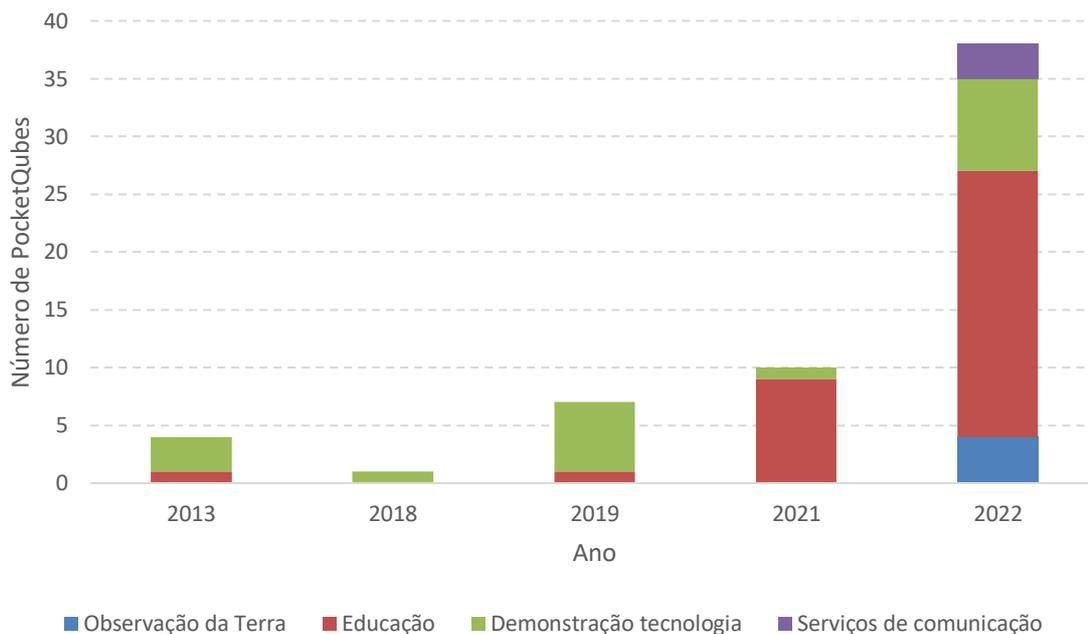


Figura 2.10 - PocketQubes lançados por ano e por aplicação.

A Figura 2.11 apresenta uma distribuição das várias missões já efetuadas, bem com a sua descrição. É possível observar que a maioria dos pico-satélites tinha como missão testar payloads. Seguidamente vem a missão de design e verificação do sistema, algo provável visto que o *PocketQube* ainda é considerado um satélite recente e como ainda não se definiu um *standard* relativamente à organização de sistema, por exemplo, serão cada vez mais as diferentes tentativas de *designs*.

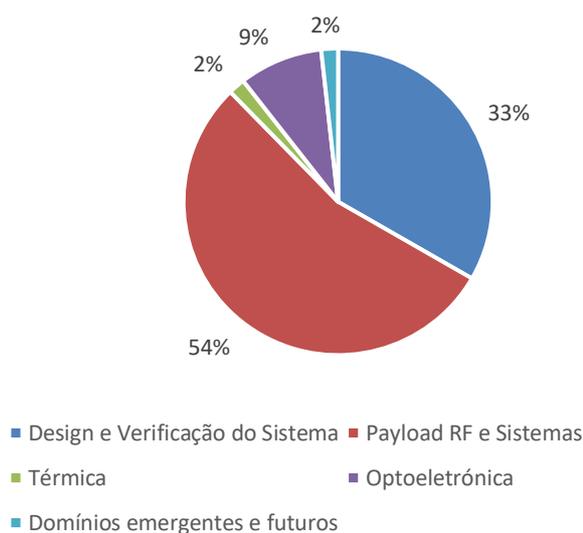


Figura 2.11 - Descrição das Missões efetuadas.

O gráfico da Figura 2.12 mostra uma distribuição do tipo de organização responsável pelos *PocketQubes* já lançados. Inicialmente foram as universidades que começaram por desenvolver os

primeiros projetos de pico-satélites, mas, entretanto, devido à sua elevada potencialidade, várias empresas têm apostado no desenvolvimento destes pequenos satélites. Seguem então uma abordagem semelhante à dos CubeSats, primeiro universidades e depois empresas. A porção de organizações sem fins lucrativos refere-se ao estado da organização na altura em que desenvolveu o satélite. Por exemplo, a FOSSA.Systems desenvolveu o primeiro satélite como sendo uma dessas organizações, mas atualmente é uma empresa que tem elevada importância nesta temática de pequenos satélites [29].

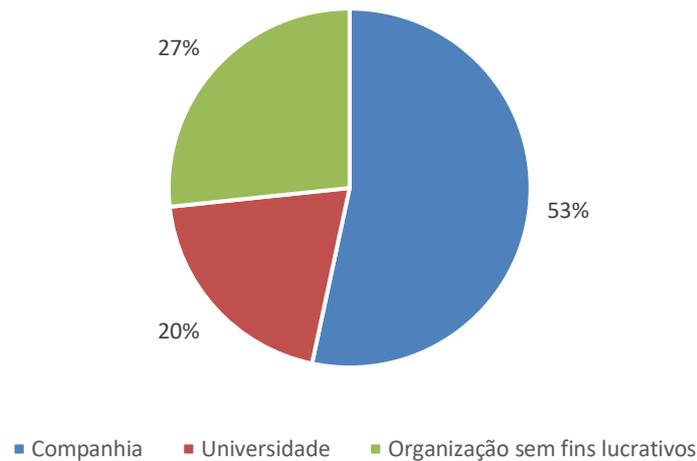


Figura 2.12 - Tipo de Organização responsável pelos PocketQubes lançados.

A Figura 2.13 e a Figura 2.14 mostram, através do mapa mundo, a distribuição de *PocketQubes* lançados e em desenvolvimento por país desde 2013 até a atualidade, respetivamente. Espanha destaca-se como sendo o país que mais pico-satélites lançou, 21. Seguem-se os Estados Unidos da América com 10, contrastando com a Figura 1.3 onde os EUA eram o país com mais lançamentos de *CubeSats*. Relativamente aos *PocketQubes* em desenvolvimento (Figura 2.14), o Reino Unido lidera com 7 e segue-se a Espanha com 6, continuando então o excelente trabalho no desenvolvimento destes pequenos satélites.

De acordo com a base de dados [10], existe apenas um *PocketQube* em desenvolvimento em Portugal, o PROMETHEUS. É também o único pico-satélite português presente na lista de satélites que solicitaram coordenação de frequência de satélite amador formalmente no IARU (The International Amateur Radio Union), reforçando a afirmação anterior [30].

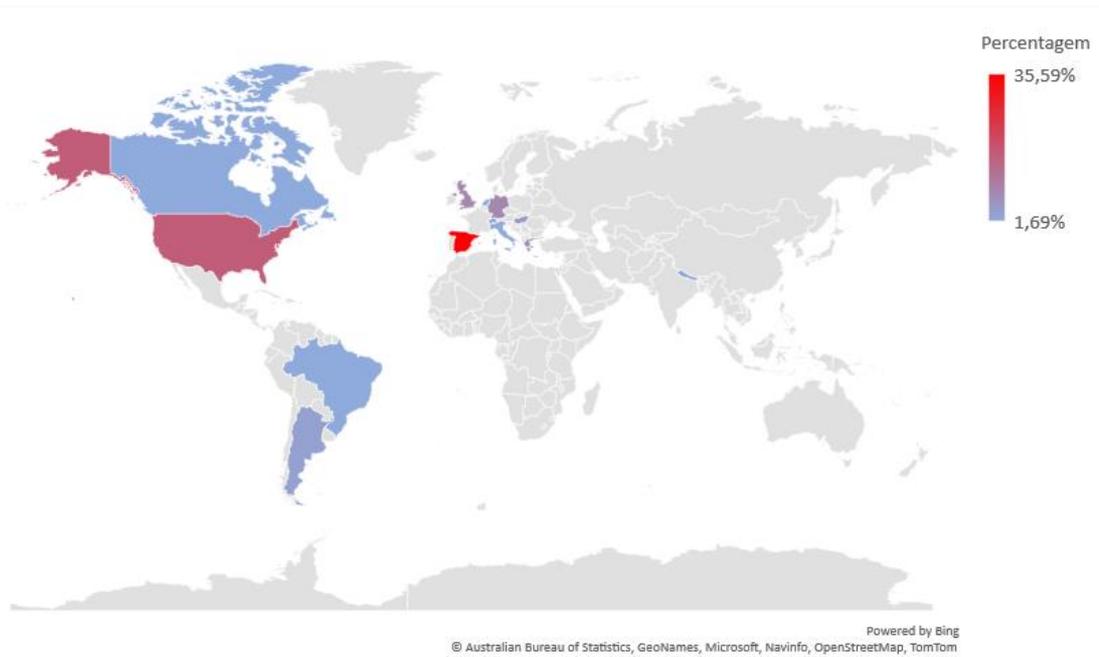


Figura 2.13 - PocketQubes lançados por país.

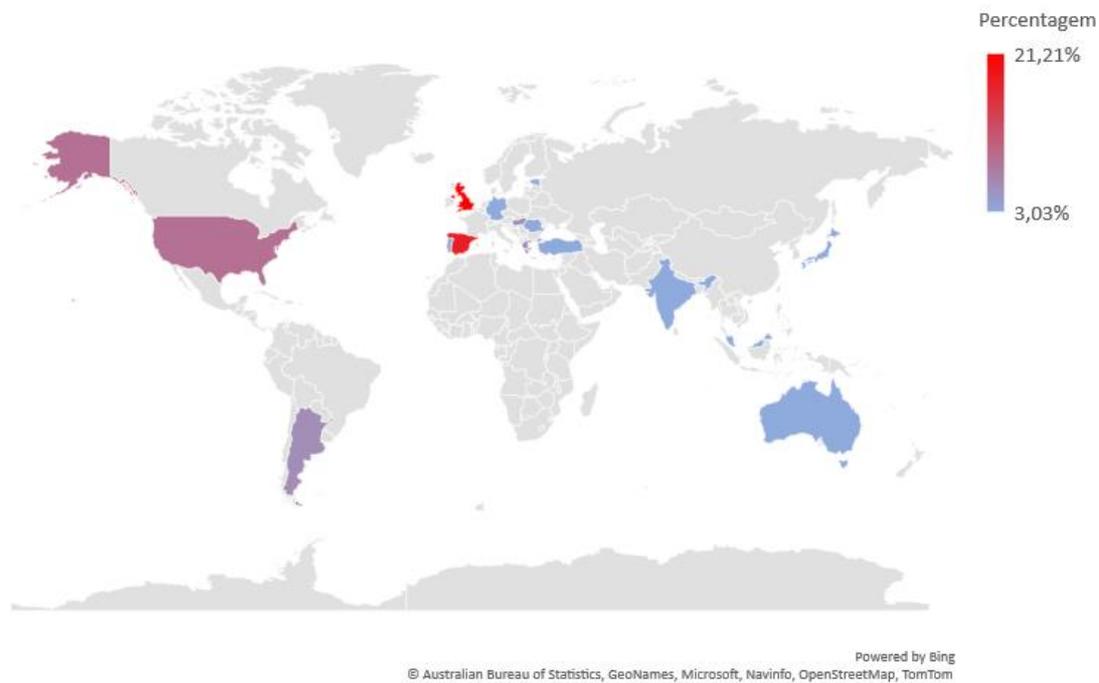


Figura 2.14 - PocketQubes em desenvolvimento por país.

2.3 Subsistemas

Um *PocketQube* ou um *CubeSat* são simplesmente um satélite só que de dimensões reduzidas e, como tal, são desenvolvidos, testados e projetados como um satélite convencional. Deste modo, nesta introdução do capítulo, aborda-se de uma forma geral um sistema de satélite operacional.

Assim que um satélite é lançado e colocado em órbita, há alguns componentes interativos necessários para operar uma missão de satélite [31]. Os cinco principais componentes do sistema são: [31]:

- Satélite;
- Estação terrestre;
- Centro de comando e controlo;
- Centro de armazenamento de dados;
- Centro de análise e distribuição de dados.

A estação terrestre é utilizada para receber dados provenientes do satélite (*downlink*) e também transmitir dados para ele (*uplink*) [31].

Todos os componentes necessários para operar uma missão de satélite podem ser contidos em apenas uma instalação, com exceção do próprio satélite [31].

Relativamente ao satélite, existem três elementos principais presentes em qualquer um, sendo eles: o *payload*, o barramento do satélite e o conjunto do adaptador do veículo de lançamento [31]. O *payload* é definido como o equipamento que desempenha a função de missão do satélite [31]. O barramento do satélite é definido como sendo a estrutura e os sistemas dentro do satélite, que fornecem funções que permitem ao *payload* executar a missão pretendida [31]. Isto inclui fornecimento de energia, proteção térmica, estabilidade e controlo orbital para que o *payload* funcione dentro dos limites do projeto [31].

O barramento do satélite é composto pelos subsistemas de Estruturas e Mecanismos, Potência, Comunicações e Telemetria, Controlo Térmico, Determinação e Controlo de Atitude, e Propulsão [31]. O peso total de um satélite pode ser representado pelas percentagens presentes na Figura 2.15.

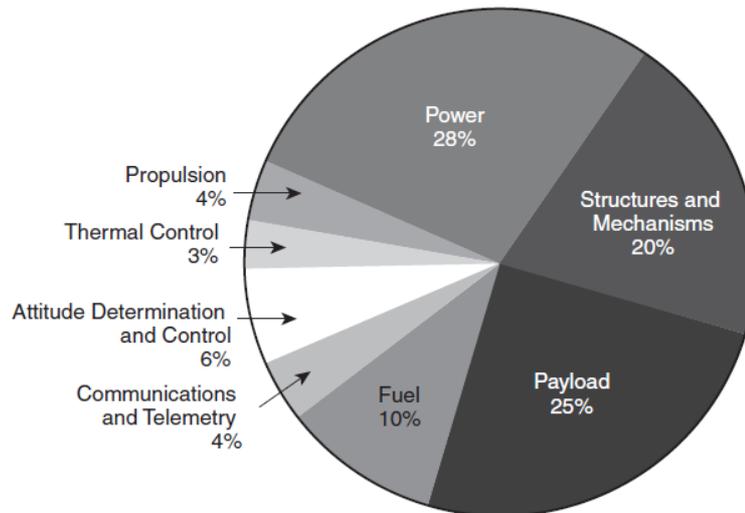


Figura 2.15 - Massa de um satélite representada pela percentagem do projeto geral [32].

2.3.1 Sistema Eletrônico de Potência

O Sistema Eletrônico de Potência, como se pode observar na Figura 2.16, fornece, armazena, distribui e controla a energia elétrica da aeronave [32]. O EPS é fundamental e é o maior subsistema, por isso é comum abranger uma grande porção de volume e massa em qualquer satélite [32].

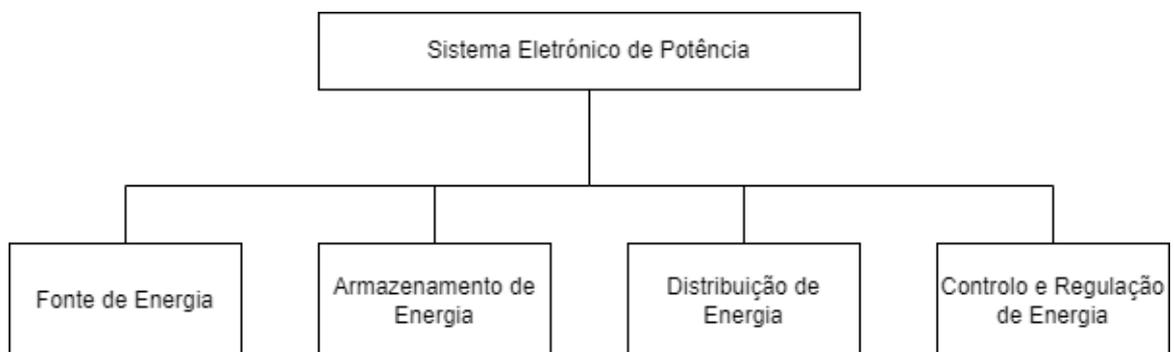


Figura 2.16 – Esquema funcional para o EPS de uma aeronave [32].

A função principal desempenhada pelo EPS é fornecer energia elétrica a todos os subsistemas e *payloads* [32]. Contudo, existem outras funções específicas desempenhadas pelo EPS. Tais como:

- Fornecer uma fonte contínua de energia elétrica para as cargas das aeronaves durante o tempo de vida da missão [32].
- Controlar e distribuir energia elétrica para a aeronave [32].

- Suportar requisitos energéticos para carga elétrica média e de pico [32].
- Fornecer conversores para AC (corrente alternada) e regular barramentos de potência DC (corrente contínua) [32].
- Fornecer capacidade de comando e telemetria sobre estado do EPS, assim como controlo por estação terrestre ou outro sistema autónomo [32].
- Proteger o *payload* da aeronave contra falhas no EPS [32].
- Suprimir tensões transientes de barramento e proteger contra falhas de barramento [32].

Os requisitos de dimensionamento mais importantes são as exigências de potência elétrica média e de pico e também o perfil orbital, isto é, inclinação e altitude [32].

As tecnologias das fontes de energias incluem células fotovoltaicas, painéis solares/fotovoltaicos, matrizes fotovoltaicas (módulos fotovoltaicos interligados) e também radioisótopos ou outra fonte de energia termonuclear [32].

Relativamente aos *CubeSats* e *PocketQubes*, a fonte de energia é normalmente governada por uma arquitetura de energia solar comum (células solares, painéis solares e matrizes solares) [9]. Dado os estrangimentos da área da superfície destes pequenos satélites, as células solares podem ser integradas logo na construção, dependendo dos requisitos energéticos da missão [9].

O armazenamento de energia é tipicamente aplicado através de baterias, com a finalidade de fornecer energia durante eclipses de órbita e também para satisfazer os requisitos de potência de pico dos instrumentos [9]. Estas baterias existem em grande variedade no mercado, tais como baterias de Li-ion (íões de lítio) de alta densidade de energia e baterias de Li-po (polímero de lítio) [9]. Sendo estas utilizadas como primárias de uso único ou então baterias secundárias recarregáveis [9].

Os sistemas de distribuição, regulação e controlo de energia são normalmente construídos pelos engenheiros de EPS dependendo dos requisitos do sistema e da missão [6]. Eles propõem uma potência específica alta ou uma taxa de potência-massa quando selecionam as tecnologias a utilizar na fonte e armazenamento de energia para minimizar o impacto da massa no sistema [9]. Desta forma, o volume do EPS é considerado uma restrição para os satélites destas tipologias [9].

Sendo assim, é considerado um desafio técnico desenvolver um EPS totalmente funcional que encaixe dentro de um cubo com 5 cm de lado. Para tal, de modo a melhorar o fornecimento de energia de um *PocketQube*, uma solução é desenvolver um sistema com um painel solar implantável, ou seja,

que abre após o lançamento e que pode acompanhar o movimento do sol omnidirecionalmente [33]. Através deste método, maior será a quantidade de células solares que se podem colocar num pequeno satélite [33].

2.3.2 Estrutura

A estrutura é o chassi principal da aeronave, que suporta mecanicamente todos os subsistemas, assim como desempenhar a função de escudo térmico e de radiação para componentes sensíveis [6].

A escolha do material é bastante importante quando se trata de estruturas de aeronaves pequenas [9]. Devem ser satisfeitos requisitos tanto para propriedades físicas (densidade, coeficientes de expansão térmica, tolerância à radiação) como para propriedades mecânicas (módulo de elasticidade, dureza) [9]. No geral, a escolha dos materiais da estrutura é imposta pelo ambiente de operação da aeronave [9].

Normalmente, as opções comuns para as estruturas primárias no mercado das pequenas aeronaves são o uso de estruturas COTS, construções personalizadas ou então uso de componentes impressos [9].

Tipicamente, as estruturas dos *CubeSat* são feitas de ligas de alumínio 6061 ou 7075 [9]. Contudo, estruturas para *CubeSats* e *PocketQubes* produzidas em impressoras 3D têm sido alvo de algum interesse por parte de projetistas [6].

2.3.3 Comando e Tratamento de dados (aviónica)

O subsistema de Comando e Tratamento de dados executa duas grandes funções [32]. É responsável por receber, validar, decodificar e distribuir comandos para outros subsistemas da aeronave [32], bem como reunir, processar e armazenar dados de manutenção, assim como da missão, para *downlink* ou para uso por um computador a bordo [32]. Este subsistema também inclui funções adicionais, tais como a medição do tempo da aeronave, monitorização do estado do computador e das interfaces de segurança [32].

À medida que os pequenos satélites estão a evoluir, desde missões consideradas de curto termo para missões potencialmente longas, a tolerância à radiação também entra na discussão no que diz respeito à seleção de materiais [5] [9]. Avanços recentes em tecnologias de microcontroladores COTS

permitem recursos de alto desempenho, embora com maior vulnerabilidade à radiação espacial. Sistemas comuns de tratamento de dados a bordo para *CubeSats* incluem microcontroladores FPGAs (*Field Programmable Gate Array*), MSP (*Mixed Signal Processing*) e PIC (*Peripheral Interface Controller*), assim como microcontroladores baseados em arquitetura ARM (*Advanced RISC Machine*) de alto desempenho e eficiência energética [6] [9]. Além do mais, existem várias plataformas de código aberto de desenvolvimento de *hardware* e *software* promissoras, tais como Arduino, BeagleBone e Raspberry Pi que estão a ganhar um elevado interesse por parte dos projetistas de *CubeSats*. Estas plataformas podem ter alto valor, fornecendo ferramentas simplificadas e económicas para desenvolvedores de pequenos satélites [6] [9].

Os satélites que usam um sistema de comando e tratamento de dados distribuído usam normalmente o protocolo de data I2C para a comunicação entre microcontroladores, elevando-se sobre o uso de USB e CAN, devido ao facto de o suporte do I2C estar já integrado na maioria dos microprocessadores e por isso consumir uma quantidade reduzida de potência [5] [34].

Em relação ao armazenamento de dados, tipicamente os *CubeSats* têm uma memória *on-board* variando entre alguns *kilobytes* ou *megabytes* [3]. Porém, alguns deles possuem uma memória *flash* adicional que pode aumentar a capacidade de armazenamento total para a ordem das centenas de *gigabytes*, dependendo dos requisitos da missão [3]. Esta capacidade de armazenamento de nada pode servir se as taxas de dados não forem aumentadas de acordo com o aumento de memória [3]. Com o auxílio da equação (1) [3]:

$$\text{Armazenamento (MB)} = \frac{15}{2} \Delta T \left(\frac{\text{min}}{\text{acesso}} \right) R_b (\text{Mbps}) \quad (1)$$

Onde Armazenamento (MB) é a capacidade de armazenamento necessária para esvaziar a memória durante um acesso à estação terrestre (*ground station*), em *megabytes* (MB); $\Delta T \left(\frac{\text{min}}{\text{acesso}} \right)$ é a duração média de um acesso à estação terrestre, em minutos; R_b (Mbps) é a taxa de dados *downlink*, em Mbps; o fator $\frac{15}{2}$ provém da conversão de *bits* em *bytes* e também da conversão de minutos em segundos [3]. É por isso possível entender que o verdadeiro fator limitante é a taxa de dados, e não o armazenamento de dados [3].

2.3.4 Comunicações

O subsistema de Comunicações é a interface entre o satélite e a *ground station*, que permite à aeronave fazer o *downlink* do seu *payload* e dos dados de manutenção para o centro de operação, transmitir comandos do operador para a nave (*uplink*), assim como estabelecer comunicações intersatélite, ou seja, entre satélites [6].

De acordo com um estudo efetuado no ano de 2010, a maioria dos pico e nano-satélites têm uma frequência de *downlink* na banda UHF (*Ultra High Frequency*) e transmitem os seus dados com uma forma digital de modulação, com taxas de dados típicas entre 1200 e 9600 bps [5]. Contudo, a banda VHF (*Very High Frequency*) e a banda S também se podem utilizar, sendo às vezes utilizadas como frequência secundária de *downlink* [5]. As taxas de dados VHF são inferiores a 9600 bps, enquanto as da banda S atingem os 256 kbps [5].

Bouwmeester e Guo, na sua investigação, concluíram que as capacidades de comunicação, ou seja, as taxas de dados entre os pico e nano-satélites e a *ground station* são limitadas principalmente pelo *link budget*, ao invés da existência da tecnologia de rádio [5]. Uma vez que existe uma grande variedade disponível de componentes eletrónicos com uma elevada taxa de dados [5].

2.3.5 Orientação, Navegação e Controlo

As aeronaves são constantemente sujeitas a vários tipos de perturbações quando estão em órbita, tais como perturbação magnética, aerodinâmica, gradiente de gravidade e efeitos da pressão da radiação solar/albedo [35]. E, para tal, é necessário algum tipo de sistema para medir e combater estas perturbações de forma a garantir que a missão seja cumprida [32].

O subsistema de Orientação, Navegação e Controlo é considerado como uma combinação do subsistema de Controlo e Determinação de Órbita (ODCS), na qual mede e mantém a posição do centro de massa do satélite em função do tempo, e o subsistema de Determinação e Controlo de Atitude (ADCS), que mede e mantém a orientação do satélite sobre o seu centro de massa [9]. Na órbita terrestre, a determinação da posição a bordo pode ser fornecida por um recetor GPS (*Global Positioning System*) [9]. Alternativamente, um sistema de rastreamento por radar baseado em terra também pode ser utilizado [9]. No espaço profundo, a determinação da posição é realizada pelo uso da *Deep Space Network* (DSN) e por um *radio transponder* a bordo da nave [9].

O ADCS é um dos subsistemas do satélite e desempenha um importante papel para estabilizar os sistemas do satélite no cumprimento da sua missão [31]. Este necessita de quatro elementos principais para desempenhar a sua função, nomeadamente a referência onde se realizam as medições, um sensor, um controlador e um atuador [31]. Porém, para uma mais fácil compreensão, este subsistema pode ser dividido em duas partes. O Sistema de Controlo de Atitude (ACS) que controla e estabiliza o sistema do satélite, e o Sistema de Determinação de Atitude (ADS) que mede o vetor de referência inercial e calcula a orientação do satélite em relação à Terra [36]. De seguida o valor desta medição será enviado para o ACS que o utilizará como referência para fazer uma correção durante a órbita do satélite [36]. A Figura 2.17 apresenta o diagrama de blocos respetivo ao ADCS, mas dividido em ACS e ADS [36][37].

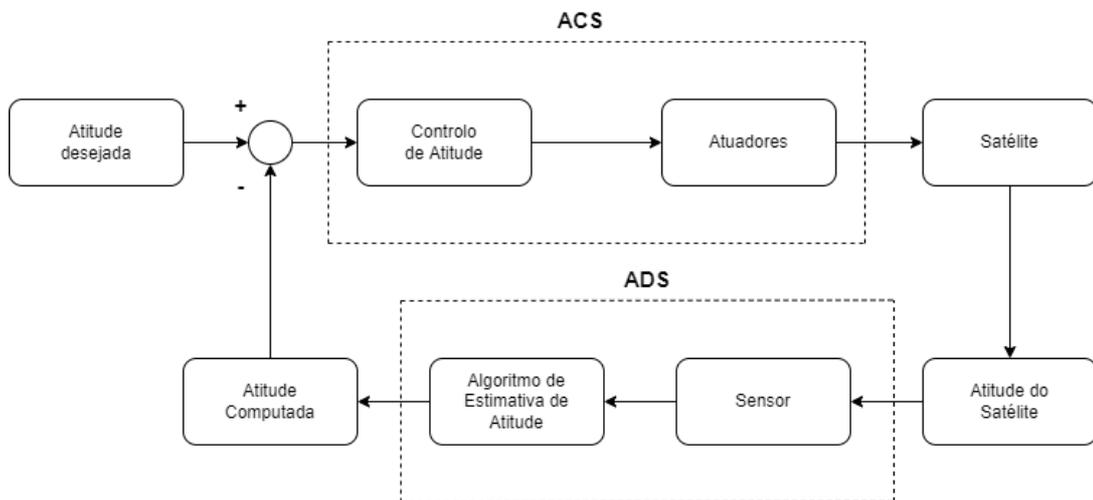


Figura 2.17 - Diagrama de blocos do ADCS [36][37].

A atitude do satélite é representada pela posição angular (Ângulos de Euler) e pela velocidade angular [36]. Contudo, a precisão da posição angular da atitude e a velocidade angular são constantemente afetadas por problemas de ruído no sensor, entre outros [36]. Por isso a estimativa de atitude é bastante importante [36].

O ADCS utiliza sensores tais como seguidores de estrelas, sensores de Sol, sensores de Terra (sensores de horizonte), *magnetometers* e giroscópios para determinar a atitude e a taxa de atitude da aeronave para depois ativar atuadores, nomeadamente rodas de reação, *magnetorquers* e propulsores com a finalidade de estabilizar e orientar a nave na direção pretendida [6], [9].

2.3.6 Controlo Térmico

A função do subsistema de Controlo Térmico (TCS) é manter toda a aeronave, os seus componentes e *payloads* a funcionar dentro dos limites de temperatura exigidos para cada fase da missão [32].

Normalmente são definidos dois limites [32]. Os limites operacionais, em que os componentes devem permanecer durante a operação [32]. E os limites de sobrevivência, onde os componentes devem permanecer sempre, ou seja, até mesmo quando o sistema não está com a fonte de energia ligada [32]. Deste modo, todos os componentes da aeronave têm diferentes gamas de temperaturas permitidas que, como tal, devem corresponder com os requisitos operacionais e de sobrevivência durante todas as fases da missão [9], [38]. A temperatura da aeronave é determinada pela quantidade de calor recebida, armazenada e emitida [9]. A Figura 2.18 demonstra de uma forma simplificada as trocas de calor efetuadas durante a órbita de um satélite [9].

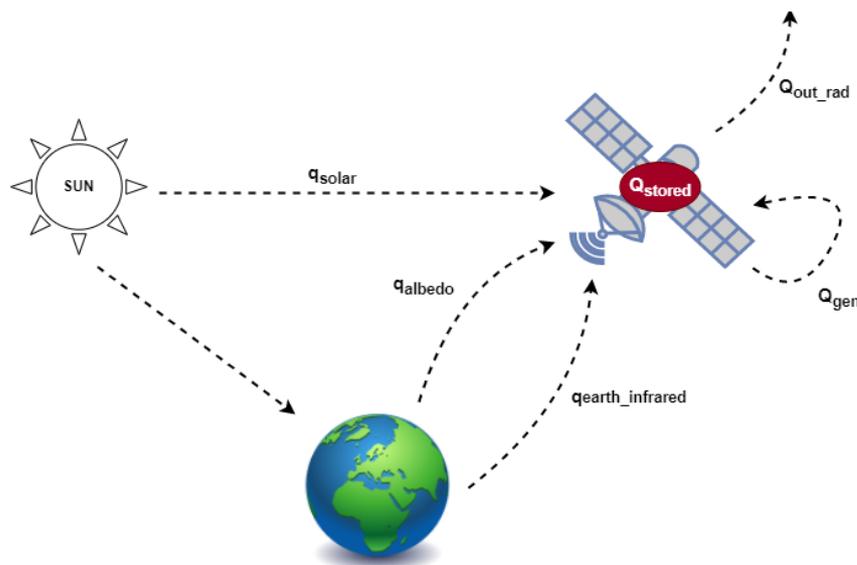


Figura 2.18 - Visão geral simplificada do aquecimento de um veículo espacial em órbita, adaptada de [9].

- Q_{gen} : calor gerado pela aeronave;
- Q_{out_rad} : calor emitido pela aeronave via radiação;
- Q_{stored} : calor armazenado pela aeronave;
- q_{solar} : aquecimento proveniente do sol;
- q_{albedo} : aquecimento solar refletido pela Terra;

- $q_{earth_infrared}$: calor infravermelho vindo da Terra.

Q_{gen} , Q_{out_rad} e Q_{stored} são representados como valores de calor, sendo Watts a unidade do Sistema Internacional de Unidades (SI) [9]. Enquanto q_{solar} , q_{albedo} e $q_{earth_infrared}$ são representados como fluxos de calor, Watts/m² no SI [9].

As técnicas de Controlo Térmico estão divididas em duas categorias, o Controlo Térmico Passivo e o Ativo [32]. O Controlo Térmico Passivo não necessita de uma entrada de energia, apenas utiliza materiais, revestimentos ou acabamentos de superfícies para manter os limites de temperatura [32]. Por outro lado, o Controlo Térmico Ativo mantém a temperatura através de meios ativos, tais como fontes de calor ou refrigeradores termoelétricos [32]. A abordagem passiva tem claramente várias vantagens significativas em relação à ativa, tais como um grau de confiança superior e também massa, volume e custo inferior [39]. O que faz deste procedimento o mais utilizado na vertente dos pequenos satélites [9].

2.3.7 Propulsão

De acordo com a pesquisa efetuada por J. Bouwmeester e J. Guo [5] em 2010, quase nenhuma das primeiras missões envolvendo nano-satélites utilizava sistemas de propulsão. Em agosto de 2011 é que, efetivamente, foi comprovado a utilização com sucesso de um sistema de propulsão num *CubeSat* [3]. O sistema utilizado foi um propulsor de gás frio a bordo do CanX-2, um *CubeSat* 3U lançado em 2008 [3].

Os sistemas de propulsão nos pequenos satélites são uma grande melhoria, alargando as suas capacidades ao permitir mudança de órbita e também o controlo preciso de atitude [6]. Verificando-se assim, nos últimos anos, uma investigação exaustiva e a realização de várias atividades de desenvolvimento [6].

Geralmente, as tecnologias de propulsão em pequenas aeronaves são categorizadas em sistemas de propulsão química, elétrica ou sem propulsor [6].

Os sistemas químicos são o método que tem funcionado desde o início da era espacial, demonstrando ser altamente capaz e de confiança [9]. São normalmente utilizados quando é necessário um impulso elevado ou então manobras rápidas [9]. Estes incluem sistemas baseados em hidrazina, mono ou bipropelente, híbridos, sistemas de gás frio/quente e propelentes sólidos [9]. Os sistemas de propulsão elétrica conseguem fornecer uma magnitude de impulso superior ao dos sistemas químicos,

contudo os custos de investigação e desenvolvimento são maioritariamente superiores [9]. Relativamente aos sistemas de propulsão sem propulsor, têm sido estudados diferentes modos operacionais tais como velas solares, amarras eletrodinâmicas (*Electrodynamic tethers*), entre outros [9]. Este último sistema procura gerar impulso através da interação com o meio envolvente, como, por exemplo, a pressão solar, campos magnéticos, entre outros [9]. Pelo contrário, os sistemas químicos e elétricos geram impulso através da expulsão de massa reativa, denominada de propelente [9].

Nos últimos anos, várias missões com *CubeSats* e *PocketQubes* utilizaram sistemas de propulsão tais como propulsor de gás frio, propulsor de plasma pulsado, propulsor de arco a vácuo, *electrospray* e também velas solares [6]. Também têm sido desenvolvidos sistemas de micropropulsão que são fabricados recorrendo à tecnologia MEMS (*Micro ElectroMechanical Systems*), uma vez que é a chave para a miniaturização [40] [41].

Em conclusão, os sistemas de propulsão são então necessários para a continuação do rápido crescimento e o aumento da capacidade de missão quer dos *CubeSats* quer dos *PocketQubes* [42]. No entanto, devido às limitações das próprias plataformas, o avanço destes sistemas desde a idealização até ao lançamento é um processo lento e demorado [42].

2.4 Projetos de código aberto

As plataformas código aberto são modelos de desenvolvimento e estão divididas em *software* de código aberto (OSS) e *hardware* de código aberto (OSHW) [43]. A autoridade que tornou o OSS reconhecido foi a *Open Source Initiative* (OSI), sendo que esta define o *software* de código aberto como sendo um *software* que pode ser usado, alterado e partilhado (modificado ou não) livremente por qualquer pessoa [43]. Este é realizado por várias pessoas e distribuído sob licenças que estão em conformidade com a definição de código aberto [43].

Por outro lado, a definição de *hardware* de código aberto é bem mais complexa, pois a utilização e desenvolvimento de *hardware* difere bastante com o uso de *software* e, como é uma área nova, pouca literatura está disponível sobre este assunto [43]. Além disso, o *software* é mais facilmente copiado e distribuído que o *hardware*, já que este último para ser modificado é necessário aceder aos desenhos, esquemáticos, diagramas, *layouts*, entre outros [43]. A *Open Source Hardware Association* (OSHWA) define *hardware* de código aberto como um *hardware* cujo *design* é disponibilizado publicamente para

que qualquer pessoa consiga estudar, modificar, distribuir, fabricar e vender o *design* ou o *hardware* baseado nesse *design* [43].

De forma a proteger a propriedade intelectual de indivíduos e entidades existem alguns direitos a ter em conta, sendo que os mais relevantes são o *copyright*, patentes e marcas registadas [43]. O *copyright* concede ao criador do trabalho original o direito de o utilizar e distribuir, bem como definir as condições para tal [43]. O objetivo principal é permitir ao autor beneficiar com a sua obra financeiramente, por exemplo [43].

A utilização de plataformas código aberto tem benefícios quer para o desenvolvedor, quer para o cliente, sendo que os principais são a confiabilidade, customização, inovação, colaboração e custos [43].

Existem algumas plataformas de *CubeSat* que se apresentam como sendo de código aberto, contudo a maioria desses projetos não se enquadram no contexto de código aberto definido anteriormente [43]. Alguns deles referem que se utilizou um dado componente no projeto, mas não especificam como é que ele foi integrado no satélite, como por exemplo o ArduSat [43].

Nos subcapítulos a seguir, são analisados vários projetos de *CubeSat* e *PocketQube* que se dizem ser de código aberto ou então são *kits* didáticos, onde nos seus artigos explicam não só as suas características, quer de *hardware* quer de *software*, mas também as suas capacidades e contribuições. No final do capítulo 2.4, é possível observar a Tabela 2 que apresenta uma comparação entre estes vários projetos.

2.4.1 OSSI-1

O OSSI-1, que significa *Open Source Satellite Initiative-1*, é um *CubeSat* 1U que foi lançado em 2013 [44]. Foi projetado por um operador de rádio amador chamado Hojun Song, que trabalhou neste projeto durante sete anos, desenhando e construindo o satélite apenas com componentes COTS em vez de utilizar *hardware* mais qualificado para uso no Espaço [45]. Utilizou um microcontrolador Arduino, uma bateria de Li-ion e um *transceiver* UHF/VHF [45].

Porém, as páginas onde foi documentado o desenvolvimento do satélite já não se encontram disponíveis, não sendo possível aceder aos seus esquemáticos nem outros detalhes.

2.4.2 LibreCube

A iniciativa de código aberto do design de um *CubeSat* denominada de LibreCube foi iniciada em 2014 [43]. Também teve a finalidade de servir como exemplo de como os projetos de código aberto de *CubeSats* podem ser realizados e publicados [43].

Os três tópicos característicos da iniciativa LibreCube são: (i) plataforma código aberto, (ii) modularidade e (iii) compatibilidade [43].

O modelo código aberto é implementado tendo cada produto (subsistemas, módulos, aplicações de *software*, etc.) ocupando um único repositório Git, hospedados na plataforma GitHub [43].

A modularidade é associada com a separação dos sistemas mais complexos do *CubeSat* em sistemas mais pequenos, cada um desempenhando uma função específica [43]. Porém, esta decomposição nem sempre é economicamente rentável. LibreCube tenta cumprir o modelo de sistema espacial ECSS (European Cooperation on Space Standardization) para a decomposição do sistema [43]. Eventualmente, os módulos individuais podem ser agregados para sistemas mais complexos e combinados numa variedade de configurações para corresponder às necessidades a missão [43].

A compatibilidade é o aspeto principal para assegurar longevidade à estrutura do LibreCube [43]. O objetivo aqui é que seja possível substituir módulos individuais com sistemas equivalentes ou versões posteriores [43].

A iniciativa LibreCube opera um website [46] que disponibiliza informações sobre a estrutura, *standards* aplicáveis, convenções de nomenclatura, acesso aos repositórios e outros recursos. Após replicar o repositório, os utilizadores são livres de estudar e modificar o *design* [43].

2.4.3 ArduSat

ArduSat é uma abreviatura para “Arduino Satellite” e é uma plataforma código aberto de um nano-satélite, baseado no padrão *CubeSat* [47]. A arquitetura geral pode ser observada na Figura 2.19.

O componente nuclear do ArduSat é o ArduSat Payload Processor Module (ASPPM) [47]. Para comunicar com a estação terrestre, o ArduSat está equipado com um recetor UHF *half-duplex*, a operar

na banda de satélite rádio amador 435-438 MHz [47]. A plataforma utilizada para comunicar com os sensores é o Arduino, uma plataforma código aberto de prototipagem de eletrônica [47].

Os sensores utilizados no ArduSat são um sensor de luminosidade, acelerômetro de três-eixos, giroscópio digital de três-eixos, sensor digital de temperatura, magnetômetro digital de três-eixos, sensor de temperatura infravermelho e um contador Geiger [47].

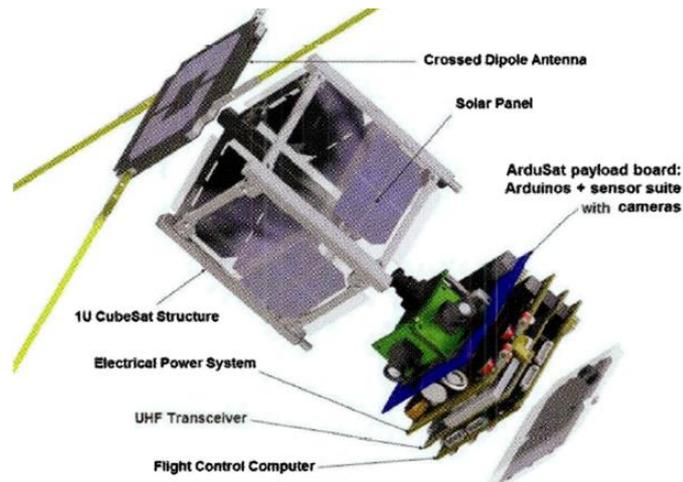


Figura 2.19 - ArduSat, arquitetura geral [47].

2.4.4 PyCubed

PyCubed é uma plataforma de código aberto de *hardware* e *software* para *CubeSats*, que também aborda muitos dos desafios comuns na construção de pequenos satélites [48]. Esta plataforma prioriza práticas de design de alta confiabilidade, componentes de baixo custo, majoritariamente COTS amplamente disponíveis e tolerantes a radiação, facilidade de utilização e documentação completa [48].

Com a adição de uma bateria, o PyCubed é capaz de executar todas as funções necessárias para uma missão de um *CubeSat* padrão, como foi o caso da missão bem-sucedida do KickSat-2 LEO *CubeSat* realizada de novembro de 2018 a março de 2019 [48][49].

PyCubed é uma *PCB* capaz de executar todas as operações que um *CubeSat* necessita, tais como o comando e controle do satélite, o subsistema de eletrônica de potência, telecomunicações, armazenamento e recolha de dados, interface com payloads e mecanismos de desenvolvimento à prova

de falha [49]. O seu *Hardware* está dividido pela placa principal (*mainboard*), e pela placa de bateria (*battery board*). O diagrama de blocos da placa principal pode ser observado na Figura 2.20, onde está dividido em subsistemas e é implementado numa única *PCB* do tamanho de uma placa PC104, aproximadamente 90 x 96 mm [50].

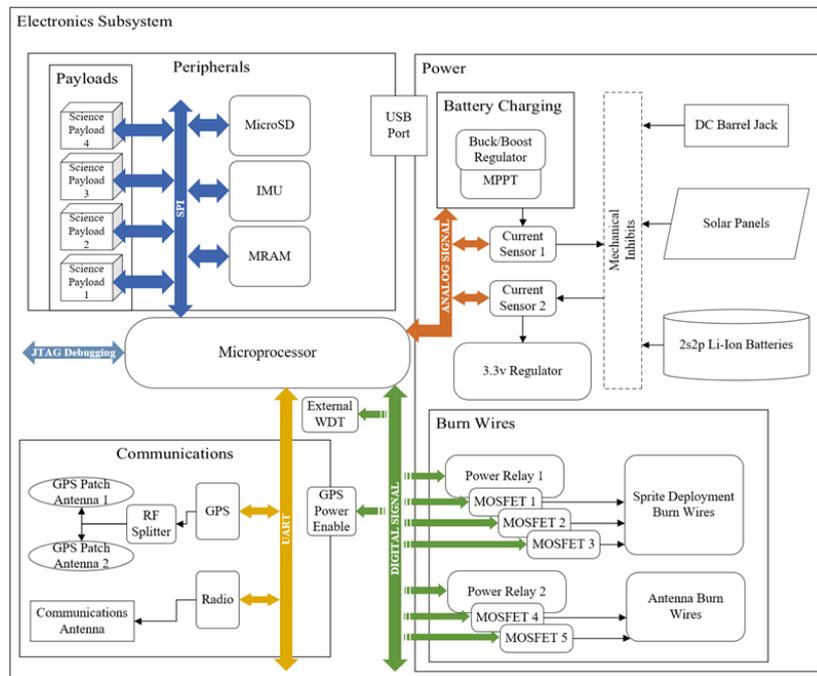


Figura 2.20 - Diagrama de blocos da placa principal PyCubed [47].

É importante salientar que o diagrama de blocos anterior corresponde à placa principal inicialmente utilizada na missão do *CubeSat kickSat-2* mas, ao longo dos anos, a equipa da PyCubed foi atualizando e modificando os esquemáticos da placa principal. Isto é, começaram com um primeiro esquemático designado por *mainboard_v1*, utilizado na missão, e posteriormente várias versões foram surgindo, sendo que a última é denominada de *mainboard_v5*. No entanto, não elaboraram outro diagrama de blocos mais atualizado de acordo com as mudanças efetuadas nos esquemáticos.

Relativamente à Figura 2.20, uma das principais características que se pode observar é que existe um microprocessador que recebe e envia dados para quase todos os componentes do sistema, ou seja, é uma espécie de 'cérebro' da operação, que constituirá um dos principais subsistemas da placa principal, o subsistema da Aviónica.

Quanto ao subsistema de Potência, é possível compreender que a placa PyCubed pode ser alimentada de duas formas: baterias Li-Ion ou por uma fonte de tensão DC. Contudo, tem elementos capazes de recarregar as baterias, quer através da fonte de tensão DC quer através da tensão proveniente de painéis solares.

Dentro do subsistema de Potência, existe outro subsistema, denominado *Burn Wires* que contém um circuito utilizado no mecanismo da abertura das antenas e também conectores para os painéis solares. A placa principal PyCubed possui dois circuitos *burn wires* independentes. O conceito de *burn wire*, utilizado como mecanismo de libertação, é simplesmente a utilização de um fio feito à base de uma liga metálica de níquel-crômio que, quando ativo, aquece por efeito de joule e derrete o fio de pesca preso no local de amarração, libertando os painéis solares ou as antenas [49][51].

Outro subsistema que pode ser observado é o de Comunicações, que possui um rádio, um GPS e as suas respetivas antenas.

Por último, o subsistema de Periféricos contém uma memória não volátil (MRAM), cartão MicroSD, um IMU (Unidade de Medição Inercial) e ainda outro subsistema *Payloads* que contém quatro blocos para payloads.

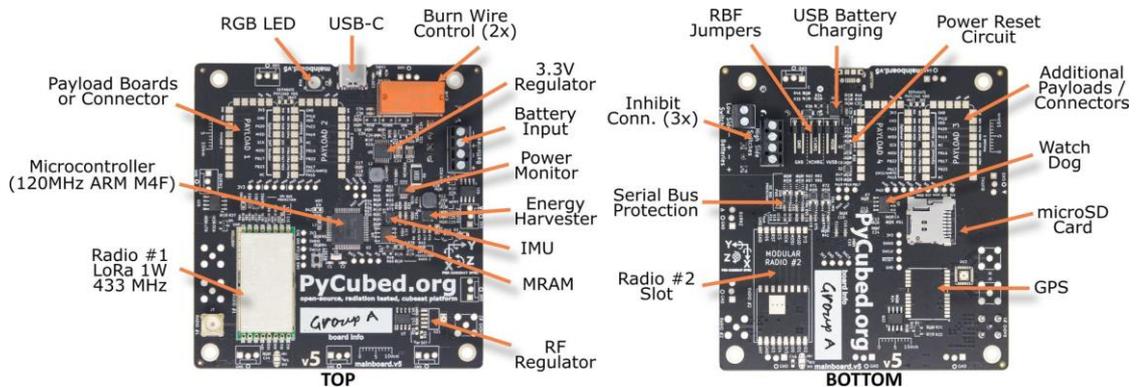


Figura 2.21 - Visão geral da placa principal PyCubed [49].

Comparando a Figura 2.21 (versão mais recente – v5) com o diagrama de blocos da Figura 2.20 verifica-se que alguns componentes foram removidos, tais como a fonte de alimentação DC que deixou de ser opção para alimentar o sistema e, no seu lugar, colocaram um conector USB também como alternativa para alimentar a placa e também recarregar as baterias. Também o conector JTAG e o conector Micro USB foram removidos da placa mais recente e foram substituídos pelo recetor USB2.0, que é o conector USB final. Outra das alterações foi a criação de um sistema de proteção do barramento das comunicações utilizadas. Realçando que na versão inicial eram utilizados os protocolos SPI e UART mas, na versão mais recente, foram utilizados o SPI e I2C.

Estes dois protocolos I2C e SPI estão vulneráveis a falhas em todo o barramento se pelo menos um dispositivo que faça parte do barramento apresentar algum tipo de mau funcionamento [52]. O

PyCubed tem uma solução inovadora, introduzida na versão mais recente da placa principal, para melhorar a confiabilidade do sistema através de circuitos automáticos de isolamento de barramento em série [49]. Uma vez que os circuitos de isolamento são autônomos, não há nada a ser alterado no *software* [49]. É de salientar que se for adicionado algum dispositivo com protocolo I2C ou SPI ao barramento das comunicações, é necessário incluir os circuitos de isolamento nessas placas também [49]. Nomearam este subsistema de *Bus Protection*.

Contudo, como foi referido anteriormente, para completar a análise do diagrama de blocos da placa principal (Figura 2.20) é fundamental conhecer e entender a placa responsável por suportar as baterias que alimentam a placa principal. Por isso, na Figura 2.22 é possível observar essa mesma placa.



Figura 2.22 - Pcb da placa da bateria PyCubed [49].

O circuito do conector para as baterias é destinado somente ao uso de células 18650 e atende aos requisitos de capacidade máxima de integradores *CubeSat* comuns [49]. Este circuito foi configurado no modo 2s3p, ou seja, duas células em série e três em paralelo.

O conector JST, auxiliado pelas inscrições “2S3P” e “7.2V”, é o responsável por estabelecer a ligação entre esta placa da bateria (Figura 2.22) e a placa principal (Figura 2.21), ligando a outro conector JST lá presente.

Na página da PyCubed [49], existe um guia prático para quem adquirir as placas, desde o download de *softwares*, testes e verificação de conectividade, entre vários exemplos para testar o seu correto funcionamento.

Todos os arquivos de projeto e documentação de confiabilidade estão disponíveis no repositório GitHub do projeto [53], encorajando os entusiastas a fazer um *fork* do design e construir as suas próprias

placas PyCubed [49], permitindo-lhes modificar, documentar e contribuir com o seu próprio conhecimento e experiência para o projeto de código aberto [48].

2.4.5 DynOSSAT-EDU

O DynOSSAT-EDU, desenvolvido pela BHDynamics, é um kit de desenvolvimento de pico-satélites de código aberto, compatível com CircuitPython e Arduino projetado para fabricantes, estudantes e amadores [54]. Tem como objetivo fornecer às comunidades DIY (*Do It Yourself*) e educacional uma plataforma *PocketQube* completa e realista, ensinando, praticando ou apenas despertando a curiosidade sobre a filosofia e tecnologia relacionadas ao *NewSpace* [54].

O sistema é composto por três placas independentes diferentes (Figura 2.23) usando o fator de forma PQ60 da *Libre Space Foundation*, mas substituindo o conector PQ60, que tinha 60 pinos, por um conector de 40 pinos de 1,27 mm, apelidado de PQBH40 [54]. As placas foram divididas em *OBC* (Computador de Bordo), *EPS* (Sistema Eletrônico de Potência) e *SatCom* (Comunicações).



Figura 2.23 – Kit DynOSSAT-EDU [54].

A placa *OBC* integra um microcontrolador ATSAM51, um IMU, um sensor de temperatura e outro de gases [54]. O subsistema *EPS* contém sistemas de gerenciamento da bateria e distribuição de energia, incluindo carregamento via painéis solares através da eletrônica MPPT comprovada em voo [54]. Contém também um microcontrolador ATSAM21 para controle geral da placa [54]. Por último, a placa

SatCom possui um módulo LoRa de longo alcance para simular comunicações LEO-estação terrestre e um módulo PAN 2.4 GHz (baseado no nRF52832 SoC) para simular comunicações LEO-LEO e permitir a comunicação com aplicativos móveis para projetos de ensino e simulação usando smartphones [54].

O custo do kit é de aproximadamente 229,85 euros, no entanto os ficheiros de *Hardware* e *Software* estão disponíveis num repositório GitHub, assim como outros detalhes [55].

2.4.6 UPSat

UPSat foi um *Cubesat* 2U construído e lançado pela Libre Space Foundation, iniciado pela Universidade de Patras [56]. A sua missão tinha como um dos objetivos ser o primeiro satélite totalmente de código aberto a ser lançado, fabricado na Grécia, minimizando o uso de componentes COTS [56]. Os desenvolvedores, nesta altura, referiram que até à data não havia nenhum projeto completamente de código aberto de satélite onde disponham todos os ficheiros, atualizações e progressos do sistema. Deste modo, todos os subsistemas do satélite foram projetados de raiz numa forma de *software* e *hardware* de código aberto [56]. O UPSat foi dividido em sete subsistemas, sendo eles o *Structural*, *EPS*, *IAC*, *SU*, *ADCS*, *OBS*, *COMMS*, como se pode observar na Figura 2.24 [56].

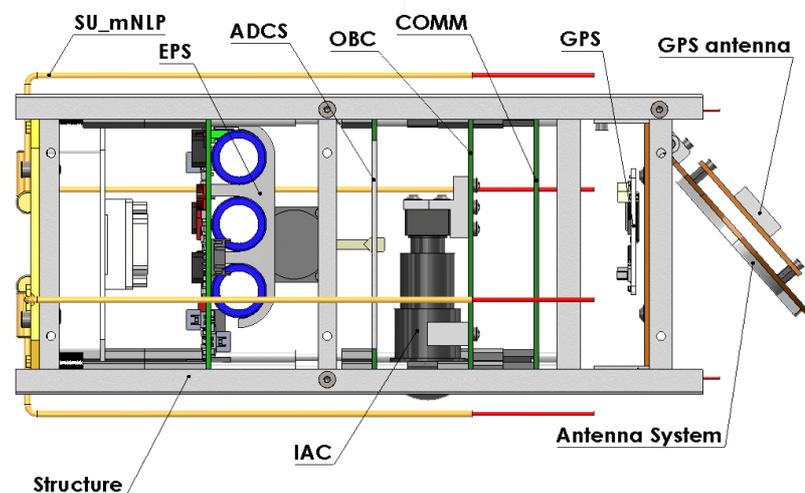


Figura 2.24 - Subsistemas do UPSat [56].

O subsistema Eletrónico de Potência (*EPS*) contém um microcontrolador STM32L1, aproveita energia de sete painéis solares através da tecnologia MPPT e possui um sistema de bateria de três células [56].

O subsistema *OBC*, contém um microcontrolador STM32F, com *software* construído em torno do sistema operacional FreeRTOS [56].

O *ADCS* possui também um microcontrolador STM32F4. Determina a atitude e posição do satélite através de uma fusão de sensores, tais como *GPS*, *magnetometer*, acelerómetro, giroscópio e um sensor solar, e controla o satélite através de *magnetorquers* e um *spin-torquers*, sendo estes os seus atuadores [56].

O subsistema *COMMS* foi desenvolvido utilizando um microcontrolador STM32F4 e *transceivers* CC1120. Possui um sistema de implantação de antenas com uma antena de GPS integrada [56].

O *IAC (Image Acquisition Component)* é uma placa *Linux* (DART-4460) a operando com o sistema *OpenWRT* que controla uma câmara *USB* (MU9PM-MH) [56].

O subsistema *Structural* foi desenvolvido tendo como base uma abordagem “híbrida” de componentes de alumínio, na estrutura, e CFRP (Polímero Reforçado com Fibra de Carbono) nas quatro faces [56].

Por último, o subsistema SU (Science Unit) utiliza um instrumento denominado de *Multi-Needle Langmuir Probe* que tem como função medir a corrente recolhida individualmente de quatro sondas de agulha, colocadas na frente do satélite. Esta corrente é convertida em tensão, filtrada, convertida para digital e depois enviada para o sistema central de telemetria [56].

O UPSat possui um repositório no GitLab, onde se pode encontrar todos os ficheiros relativos aos subsistemas [57].

2.4.7 OzQube-1

O OzQube-1, primeiro *PocketQube* Australiano, foi a primeira solução de pequeno satélite da *PicoSat Systems*, sendo uma plataforma pico-satélite, com o formato *PocketQube* como pode ser observado na Figura 2.25 [58].

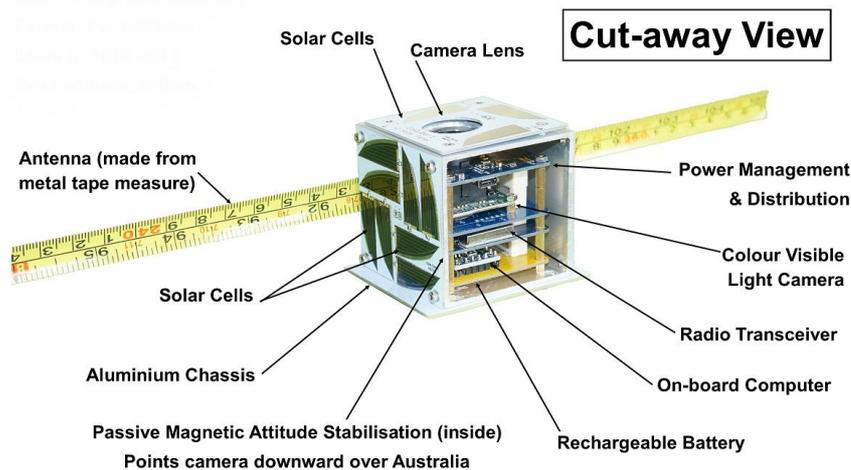


Figura 2.25 - OzQube-1 vista cortada, adaptada de [58].

Na Figura 2.25 observa-se uma vista cortada do pequeno satélite, onde é possível examinar como foi feita a divisão dos subsistemas. Segundo o artigo [58], este *PocketQube* foi idealizado para ser um satélite cuja aplicação seria Observação da Terra, tirando fotografias ao país australiano e transmiti-las para uma estação terrestre. Um facto interessante é que qualquer pessoa com uma antena adequada e um recetor de rádio poderá adquirir gratuitamente essas fotografias diretamente do OzQube-1 [58]. No entanto, o uso desta plataforma também revela interesse de potenciais áreas, tais como agricultura, sistemas de proteção, controlo ambiental e desastres naturais, recursos naturais e ainda investigações científicas [58].

2.4.8 Educative platform based on the PocketQube 60 Standard

Este específico projeto foi desenvolvido na Universidade de Zacatecas, no México, no qual foi idealizada e desenvolvida uma plataforma educativa com base no *standard PocketQube 60*, *standard* este que será abordado no capítulo 3.2. O objetivo principal deste projeto era promover o sistema de educação STEM no estado de Zacatecas [59]–[61].

A plataforma seria então dividida em módulos como EPS, OBC, COMMS e Payloads, como é possível observar na Figura 2.26.

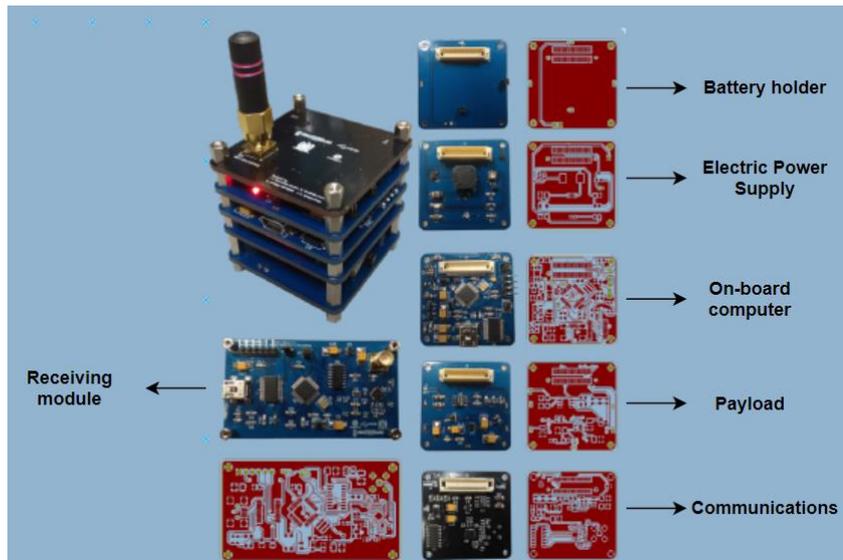


Figura 2.26 - Educative platform based on the PocketQube 60 Specification, adaptada de [60].

A placa *Battery holder* fornece suporte para uma bateria de lítio de 3,3 V, utilizada para alimentar o sistema [59]–[61].

O subsistema *EPS* tem como função regular a tensão fornecida pela bateria de Lítio, distribuindo-a em duas linhas de tensão, uma que mantém os 3,3 V e uma segunda que eleva a tensão através de um circuito *boost* para 5,7 V [59]–[61].

O subsistema *OBC* contém os componentes discretos utilizados para trabalhar com todas as funcionalidades da plataforma PQ60. É equipado com um microcontrolador ATmega328P, desempenhando a função de unidade de processamento principal. Integra também proteção contra curto-circuitos e sobretensões através de um disjuntor de reinicialização automática (LTC1153). Em relação às linhas de alimentação, o *OBC* fornece linhas de fonte de tensão reguladas, de 3,3 V (LP2985-3,3) e 5 V (LP2985-3,5), para alimentar os componentes discretos também presentes nesta placa [59]–[61].

Relativamente à placa dos *Payloads*, esta inclui um conjunto de sensores que permite a medição de variáveis físicas. Os sensores incluídos são: acelerómetro, magnetómetro, giroscópio, sensor de temperatura e um altímetro [59]–[61].

A placa que ocupa a posição do topo, a de Comunicações, inclui um *transceiver* NRF24L01 que opera na banda de 2,4 GHz e que transmite as informações medidas pelos sensores [59]–[61].

Uma vez que este projeto foi desenvolvido numa vertente mais educativa, a estrutura mecânica foi fabricada numa impressora 3D, apresentada na Figura 2.27.

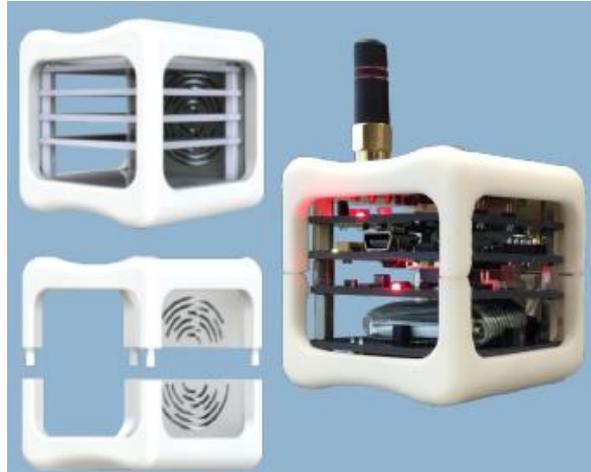


Figura 2.27 -Estrutura mecânica do projeto Educational platform based on the PocketQube 60 Specification [59]–[61].

Continuando a análise da Figura 2.26, existe ainda outro módulo, *Receiving module*, que não integra o *PocketQube* diretamente, pelo contrário, será um terminal recetor, ou então uma estação terrestre, que recebe os dados transmitidos pela plataforma [59]–[61]. Contém também um rádio NRF24L01 para demodular a informação, que é então processada e transmitida para um computador pessoal [59]–[61].

2.4.9 PocketQube Kit v1.0

A Alba Orbital também possui um kit de um *PocketQube* ideal para uma ampla variedade de áreas interessadas em construir um satélite de baixo custo, como por exemplo STEM, desde o ensino secundário ao ensino superior [62].

O kit custa cerca de 11769,91 euros e contém a placa *OBC*, uma *Protoboard*, *Labsat*, rádio *MinisatCom* (UHF 420-450 MHz) e a estrutura, como se pode observar na Figura 2.28 [62].

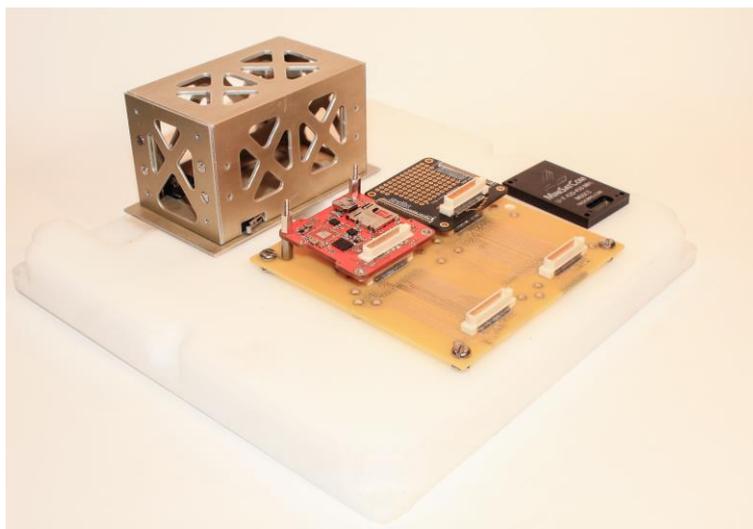


Figura 2.28 - PocketQube Kit v1.0 [62].

A placa OBC, contém o microcontrolador MSP430, suporte para cartões de memória e também foi desenvolvido tendo como base o *standard* PQ60 [62].

2.4.10 FOSSASAT-1

FOSSASAT-1 foi o primeiro satélite desenvolvido pela Fossa.Systems e também primeiro picosatélite espanhol, lançado a 6 de dezembro de 2019 [63]. Foi idealizado para fornecer comunicações IoT gratuitas e de código aberto para todo o mundo usando módulos LoRa baratos, onde os alunos poderiam comunicar com um satélite por menos de 20 euros [63]. Possui um repositório na plataforma GitHub, que contém documentos e ficheiros relativos ao *hardware* e *software* do sistema [63].

2.4.11 ESAT

O ESAT é um satélite educacional baseado no *standard CubeSat* desenvolvido para um ensino *hands-on*, ou seja, prático, para todos os níveis de educação: Educação STEM, estudos universitários e experiências profissionais [64], [65]. Apresenta um custo de 6,976 euros, mas está perfeitamente equipado para fazer experiências nas várias fases de desenvolvimento de um projeto para pequenos satélites, tais como design, fabrico, integração, validação e operação de satélites. Um aspeto bastante

interessante é que os utilizadores podem construir sobre ele pois o ESAT permite-lhes integrar e testar os seus próprios desenvolvimentos, tanto a nível de SW como de HW [64], [65]. Este satélite, presente na Figura 2.29, possui os seguintes subsistemas: sistema eletrónico de potência, comando e tratamento de dados, determinação e controlo de atitude e a estrutura [64], [65].



Figura 2.29 – ESAT [65].

2.4.12 ISTSat-1

O ISTSat-1 é um *CubeSat* 1U e será o primeiro *CubeSat* português a ser lançado no espaço, esperado ainda neste ano de 2023 [66]. Está a ser desenvolvido por estudantes e professores do Instituto Superior Técnico da Universidade de Lisboa [66]. O ISTSat-1 é um nano-satélite que está a ser desenvolvido sob o programa *Fly Your Satellite* da ESA [66]. O principal objetivo deste projeto é proporcionar aos alunos uma experiência interessante e desafiadora, a fim de promover o entusiasmo pelo espaço, ciência e tecnologia [66]. Foi criada uma página [66] onde é possível encontrar artigos publicados relativos a este nano-satélite e também dissertações dos alunos que fizeram parte deste projeto, cujo tema também era relativo a algum subsistema do *CubeSat*.

2.4.13 NessieSAT PocketQube

NessieSat (Figura 2.30) é um *PocketQube* fabricado também pela BHDynamics e possui um conjunto de módulos que permitem que o satélite desempenhe as funções exigidas por uma espaçonave para funcionar normalmente, coletar energia, realizar análises e comunicar com a Terra, isto é, funções básicas necessárias na construção de um *PocketQube* [67].

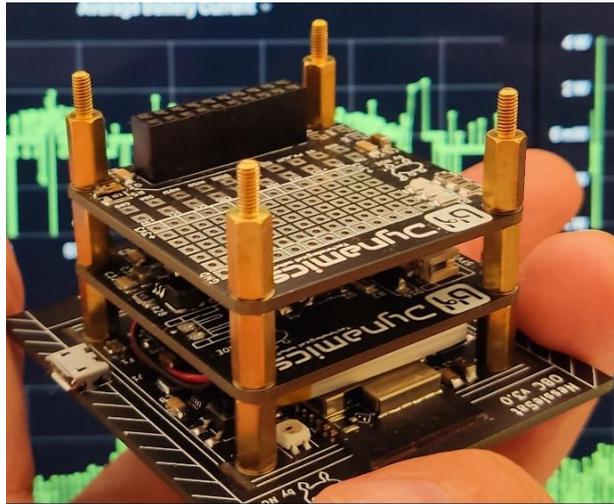


Figura 2.30 – NessieSat [67].

A placa que ocupa a posição inferior da Figura 2.30 é o módulo *OBC*. Desempenha a função de cérebro do pico-satélite e, por isso, é o módulo principal [67]. Contém um microprocessador 32-bit LX7 [67]. A meio, localiza-se o subsistema *EPS*, que suporta uma bateria de lítio polímero responsável por processar a energia proveniente dos painéis solares e a armazenar nas baterias para posteriormente distribuir pelo restante sistema [67]. A placa do topo, *Protoboard*, tem como funcionalidade o facto de o utilizador conseguir realizar os seus testes e experiências, tendo um acesso completo e fácil ao barramento do *PocketQube* [67]. Neste projeto, o barramento é denominado de PQ20N, por possuir 20 pinos [67].

2.4.14 Kitsat

O Kitsat é um kit totalmente funcional de um *CubeSat* 1U, especialmente desenvolvido para uso na educação, utilizando componentes normais que tornam o preço de compra do kit mais acessível [68].

O Kitsat é uma iteração do satélite Suomi 100, que está em órbita desde 2018 a recolher dados sobre o clima espacial [68]. O kit custa cerca de 1475 euros [68].

2.4.15 Sanosat-1 (Nepal PQ-1)

SanoSat-1 (Figura 2.31) foi o primeiro pico-satélite desenvolvido no Nepal. Um dos seus principais objetivos era idealizar, desenvolver e lançar um *PocketQube* acessível utilizando componentes COTS facilmente acessíveis [69].

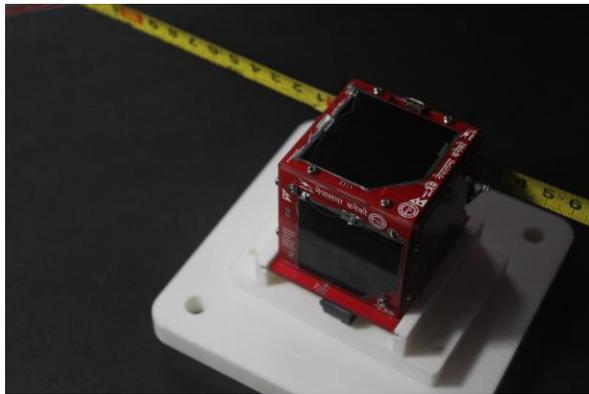


Figura 2.31 - SanoSat-1 [69].

Relativamente ao subsistema da estrutura do SanoSat-1, escolheram liga de alumínio Al-6061-T6 como material para a estrutura, devido ao seu baixo custo e também a força elevada [69]. O *EPS* contém uma bateria de Li-ion de 3,7 V, 1100 mAh, carregadores de bateria baseados na tecnologia MPPT, painéis solares, reguladores de tensão, dispositivos de monitoramento e limitadores de corrente [69]. O subsistema *OBC* é baseado num microcontrolador ATmega328, que desempenha as tarefas de controlo e tratamento de dados do *PocketQube* [69].

O SanoSat-1 foi lançado no dia 13 de janeiro de 2013 pelo lançador Falcon 9 da Space X [69].

2.4.16 UBO POCKETQUBE KIT

A *Satellite Applications Catapult* desenvolveu este projeto com o objetivo de produzir um kit de desenvolvimento de satélite baseado em *PocketQube* que permitiria ao público em geral aprender sobre os sistemas, as funcionalidades e as aplicações dos satélites, desenvolver também um conceito de

workshop que pudesse ser aplicado em todos os países e, finalmente, desenvolver uma comunidade *online* de utilizadores finais [70].

O Kit é composto pelos sistemas eletrónicos, *hardware* estrutural e uma biblioteca de *software*, todo idealizado internamente na Catapult [70]. Estes elementos já passaram por várias iterações de prototipagem, testes e revisão de *hardware* [70]. A atual versão é a segunda destes protótipos, sendo que uma terceira está atualmente em desenvolvimento [70].

Os sistemas eletrónicos do satélite foram divididos em três: *OBC*, que inclui *hardware* relativo a comunicações RF, *EPS* e uma placa de sistema de prototipagem e sensor denominada *PSB* [70]. Todos esses subsistemas foram conectados através de um barramento de sistema comum. Na placa OBC, os componentes principais eram o microcontrolador Atmel 32U4 e o rádio HopeRF RFM22B [70]. O subsistema *EPS* utilizou a tecnologia MPPT, um circuito acoplador ótico para permitir a medição da tensão dos painéis solares e regulação de carga para a bateria de lítio polímero, tanto do painel solar quanto da fonte USB [70]. Por último, a placa PSB fornece um conjunto de sensores e um *payload* [70]. Na Figura 2.32 observa-se uma vista explodida do kit completo UBO.

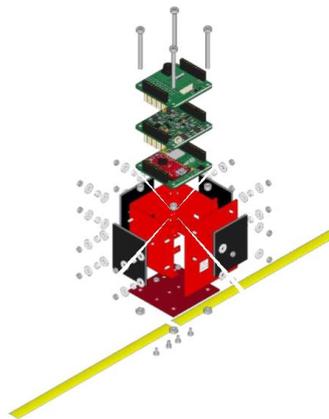


Figura 2.32 - UBO PocketQube Kit [70].

2.4.17 CubeSatSim

O Simulador *CubeSat*, também conhecido por CubeSatSim, desenvolvido pela AMSAT, foi projetado para se comportar, da forma mais razoável possível, como um satélite em órbita LEO para desmistificar o funcionamento destes [71]. Os planos de construção e *software* são totalmente de código aberto, e as informações sobre como construir um estão disponíveis em [72], incluindo publicações. A

sua primeira versão foi apresentada em 2018, mas, no entanto, novas versões e *designs* foram propostos até que mais tarde resultaram num novo *design*.

Na Figura 2.33 é possível observar como foi feito o empilhamento dos subsistemas no CubeSatSim. Começando pela parte inferior, encontra-se a placa principal, *Raspberry Pi Zero*, depois a placa da bateria e por fim a placa dos *payloads* STEM [73]. Tal como a versão anterior, utilizaram pinos GPIO de modo a empilhar as placas [73]. Relativamente ao subsistema *Payload*, na idealização do primeiro CubeSatSim, os sensores adicionais eram lidos por um *Arduino* e este depois comunicava com a *Raspberry Pi* [73]. No entanto, foi projetado, com a parceria de Bob Twiggs, uma placa *Payload* STEM que utilizava um microcontrolador STM32F103C8T6 para comunicar com a *Raspberry Pi* [73]. Possuía também um sensor de temperatura/pressão/humidade (BME280) e um módulo IMU (GY-521) [73].



Figura 2.33 – CubeSatSim [73].

Tabela 2 - Tabela comparativa de projetos.

Projeto	Tipo	uC	Radio	Payload	Custo	Ref.
OSSI-1	CubeSat	Arduino	UHF/VHF <i>transceiver</i>	-	-	[44] [45]
LibreCube	CubeSat	-		-	-	[43]
ArduSat	CubeSat	Arduino	UHF <i>transceiver</i> (435-438 MHz)	Sensores	-	[47]
PyCubed	CubeSat	ATSAMD51	RFM98PW (433 MHz)	-	-	[48][49] [51] [52]
DynOSSAT-EDU	PocketQube	ATSAMD51		-	229,85 €	[54]
UPSat	CubeSat	STM32F	CC1120 transceivers (402- 470 MHz)	m-NLP* e câmara	-	[56] [57]
OzQube-1	PocketQube	ATmega328P	RF4463F30	câmera	-	[58]
Edu.PQ60 standard	PocketQube	ATmega328P	NRF24L01	sensores	-	[59]–[61]
PocketQube kit v1.0	PocketQube	MSP430	MiniSatCom (420- 450MHz)	-	11.769,91 €	[62]
FOSSASAT-1	PocketQube	Atmega 328P AU	LoRa SX1278	-	-	[63]
ESAT	CubeSat			Térmico	6,976 €	[64], [65]
ISTSat-1	CubeSat	STM32L072CZ	LoRa SX1276	-	-	[66]
NessieSAT	PocketQube	32-bit LX7	-	-	-	[67]
KitSat	CubeSat	-	-	-	1475 €	[68]
SanoSat-1	PocketQube	ATmega328	Si4463	Sensor de radiação	-	[69]
UBO PocketQube kit	PocketQube	Atmel 32U4	RFM22B	-	-	[70]
CubeSatSim	CubeSat	Raspberry Pi Zero	-	-	-	[71]–[73]

* *multi-needle Langmuir probe*

2.5 Aplicações

Este capítulo tem como objetivo apresentar algumas das potenciais aplicações dos *PocketQubes*, bem como fornecer algumas dicas sobre as potencialidades destas plataformas e algumas considerações a seu respeito [17].

2.5.1 Observação da Terra

A observação espacial da Terra (*Earth Observation*) ajuda a expandir o nosso conhecimento sobre a Terra como um sistema integrado, bem como ajudar a melhor compreender a sua evolução natural e as mudanças ambientais induzidas pelo homem numa ampla gama de escalas temporais e espaciais [6].

Sensores ativos e passivos são os dois modelos principais de instrumentos de sensorização remota utilizados na observação da Terra, como é possível observar na Figura 2.34 [6].

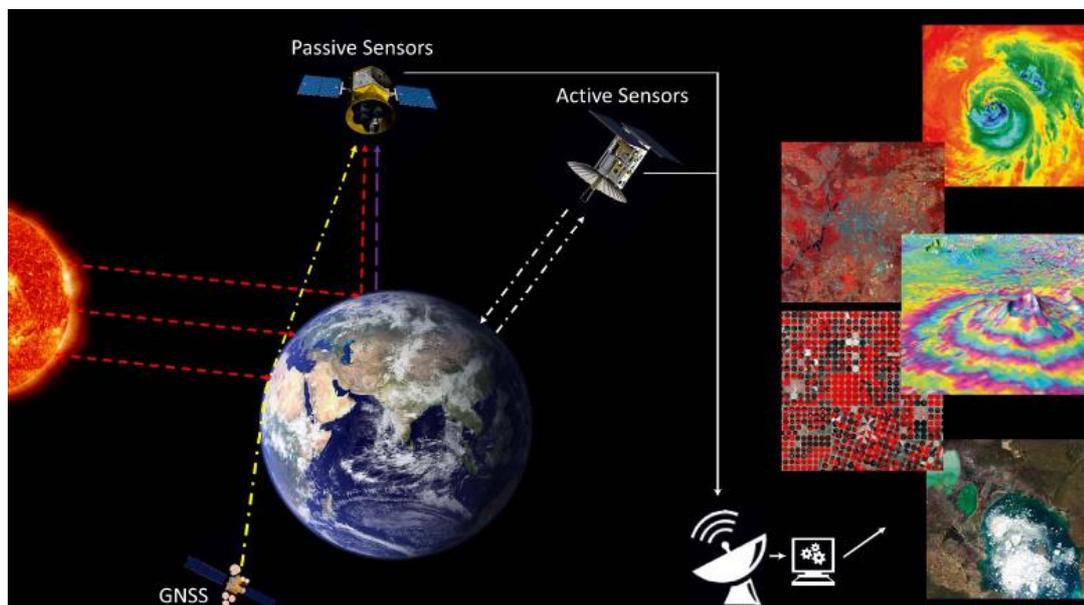


Figura 2.34 - Visão geral da Observação da Terra a partir do Espaço [6].

Os instrumentos ativos, tais como radares, *lidars*, altímetros de laser e radar, e *scatterometers* geram a sua própria radiação eletromagnética para iluminar o objeto alvo, para depois observar a porção refletida ou dispersada da radiação transmitida [6]. Permitindo, assim, ao sistema capacidades diurnas e noturnas, não dependendo da iluminação solar [6]. A limitação principal neste tipo de equipamentos é

que necessitam de alta potência, o que é limitado num *CubeSat* [3]. Foi então que, em 2011, foi determinado que radares, *lidars* e virtualmente qualquer dispositivo ativo não é compatível com plataformas de *CubeSats*, por causa da potência e restrições de tamanho [3].

Os sensores passivos, tais como radiômetros, espectrómetros, entre outros, não têm a sua própria fonte de radiação e detetam energia refletida ou emitida para observar o alvo de interesse, com o sol a ser a fonte mais comum de radiação secundária [6].

O *PocketQube* SMOG-1, ainda em órbita [21], tem como a sua principal missão a medição de *eletrosmog* (radiação não ionizada) emitida pelas estações de TV digital terrestre na banda 430-860 MHz [74].

2.5.2 Educação

Ensino e experiências *hands-on* numa plataforma de satélite real é mais inspirador e consegue fornecer uma maior aprendizagem do que apenas teoria e exercícios [17]. Nesta vertente, a relação custo-benefício de uma plataforma *PocketQube* é uma grande vantagem em comparação com os *CubeSats*, pois companhias pequenas e a maioria das instituições educacionais não têm poder económico para tal [17]. O aparecimento dos *CubeSats* abriu espaço a muitos interessados à volta do mundo, mas os *PocketQubes* têm potencial para abrir portas a ainda mais interessados e, como ainda é uma tecnologia recente, há muitos trabalhos de investigação e desenvolvimento em andamento nas universidades, por exemplo [17].

2.5.3 Demonstração de Tecnologia

Os *PocketQubes* são plataformas excelentes para demonstração de tecnologia de baixo custo, desde que o objeto ou o conceito demonstrado encaixe numa plataforma *PocketQube* [17]. A demonstração de tecnologia pode ser sobre subsistemas ou então componentes em si, mas também pode ser realizado para tecnologias tendo em consideração plataformas maiores [17].

2.5.4 Observação do clima Espacial

A órbita LEO prevista dos *PocketQubes* limita a diversidade de aplicações de clima espacial, mas a dinâmica do campo magnético terrestre em combinação com a radiação que penetra na termosfera consegue contornar essa restrição se forem necessárias medições simultâneas em vários pontos [17].

Os magnetômetros científicos encaixam num *PocketQube*, pois a física permite que eles tenham um tamanho reduzido e que as suas taxas de dados não sejam extremamente altas [17].

Embora o ambiente de radiação em LEO seja geralmente benigno, a aplicação científica potencial para instrumentos de radiação de partícula baseados em *PocketQubes* nessas órbitas é mais bem explorada por uma rede distribuída de sensores a bordo de uma constelação de satélites [17]. A análise destas medições recolhidas pelos sensores de radiação pode fornecer informações valiosas sobre a dinâmica do clima espacial, no tempo e na posição [17].

2.5.5 Serviços de comunicação

Os serviços de comunicação desempenhados pelos *PocketQubes* são extremamente limitados, devido à potência elétrica disponível [17]. No entanto, ainda é possível utilizá-los para este tipo de serviços, desde que estes requeiram *downlinks* e/ou largura de banda de dados baixo volume [17]. As principais áreas de interesse são o monitoramento de espectro (onde o *payload* será o próprio sistema de monitoramento de espectro, tal como no *PocketQube* Smog-1 [75]), Internet das coisas (IoT) e serviços de comunicação individual [17].

2.5.6 Exploração espacial

Os *PocketQubes* podem ir além da órbita da Terra usando a capacidade de partilha de viagens de missões de satélites de maiores dimensões [17]. O maior desafio técnico relativo a esta aplicação é a tolerância a radiação e o controlo térmico [17]. Uma vez que a maioria da tecnologia presente nos *PocketQubes* é desenvolvida recorrendo a componentes COTS para missões terrestres, será inadequado ir para além do ambiente LEO, onde existe uma maior radiação e altas doses de ionização [17].

O assunto sobre a qual questiona se os *PocketQubes* podem fornecer plataformas úteis e económicas para além da órbita terrestre requer um estudo complexo e dedicado, na qual será

necessário realizar comparações com plataformas maiores num determinado nível de missão, tendo em conta a utilidade e o custo geral [17].

3. PROJETO

Este capítulo descreve o desenvolvimento do projeto bem como as suas funcionalidades, considerando o conhecimento obtido no capítulo 2. Essas informações permitiram identificar algumas restrições e requisitos de modo a idealizar e projetar um sistema modular onde os seus subsistemas comunicassem entre si.

3.1 Visão global

Inicialmente, através da revisão da literatura, encontrou-se vários projetos de *PocketQubes* e *CubeSats*, tendo sido ou não lançados, onde era possível compreender como estes eram constituídos e quais os seus requisitos e restrições.

Os requisitos, isto é, algo indispensável para o correto funcionamento de um pequeno satélite, podem ser descritos com base nos seus subsistemas e nos componentes que lá serão inseridos. Mas, no entanto, se o objetivo da missão também incluir o lançamento, primeiramente existem outros requisitos que devem ser aplicados de acordo com o lançador, tais como requisitos mecânicos [19].

O *PocketQube* deverá garantir:

- que nenhum componente se separe do satélite durante toda a vida útil da missão, isto inclui, lançamento, ejeção e operação em órbita [19];
- a não utilização de materiais tóxicos, inflamáveis ou potencialmente perigosos [19];
- o requisito de degaseificação para evitar a contaminação de qualquer outra aeronave durante a integração, teste e lançamento, atendendo ao termos *total mass lost* (TML) e *collected volatile condensable material* (CVCM) [19][76];

- por último, e como já foi referido no capítulo 2.2, todos os *PocketQubes* devem possuir pelo menos dois *kill switches* para manter o satélite desligado enquanto este estiver no *deployer* [19].

Posteriormente, definiu-se então requisitos relacionados com subsistemas e componentes que são essenciais num projeto de um pequeno satélite com base noutros já desenvolvidos. Assim, para desempenhar uma missão básica, um *PocketQube* deverá ser capaz de alimentar e recarregar o sistema, armazenar energia, processar dados e comunicar com a estação terrestre. Para tal, podemos definir 4 subsistemas essenciais:

- i. Sistema Eletrónico de Potência;
- ii. Sistema de Aviónica também conhecido por Computador de bordo;
- iii. Sistema de Comunicações;
- iv. *Payload*, se aplicável.

Podendo juntar-se a estes requisitos, componentes comuns que são adicionados à estrutura, tais como a antena e os painéis solares.

Por outro lado, as restrições são limitações que existem no desenvolvimento de um pequeno satélite. No caso deste projeto, uma plataforma *PocketQube*, as principais restrições são o tamanho e a massa do sistema, como referido no capítulo 2.2. Relembrando que uma unidade (1p) de um pico-satélite é um cubo com 5 cm de lado e não deve ultrapassar as 250 g.

Assim sendo, definiu-se uma estrutura modular ou em pilha constituída por cinco placas (Figura 3.1): *Battery holder*, *EPS*, *OBC*, *COMMS* e *Payloads*.

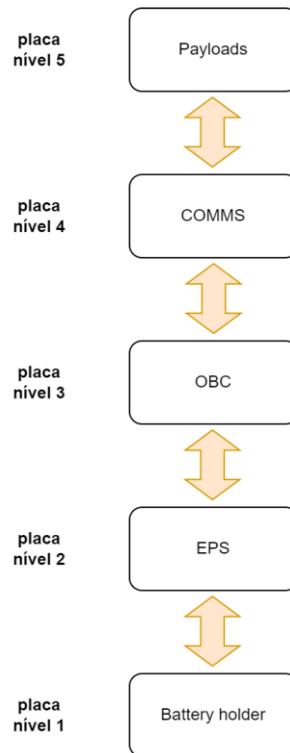


Figura 3.1 - Definição dos subsistemas.

Na Figura 3.1, observa-se então os subsistemas idealizados para este projeto de uma plataforma *PocketQube*. Definiu-se que cada placa teria de ser projetada para seguir as seguintes funções:

- *Battery holder*: suportar as baterias e fornecer energia ao sistema completo;
- *EPS*: regular a tensão fornecida pelas baterias e painéis solares;
- *OBC*: receber, validar, decodificar e distribuir comandos para outros subsistemas, isto é, lidar com todas as funcionalidades;
- *COMMS*: transmitir todas as informações recolhidas pelo OBC para um terminal recetor;
- *Payloads*: contém um conjunto de sensores e payloads.

Posteriormente foi investigado de que modo estes módulos poderiam comunicar entre si, documentado em detalhe no capítulo 3.2.

Ainda nesta fase, decidiu-se que o sistema deveria ser alimentado quer por uma bateria quer por USB (*Universal Serial Bus*), utilizando este último quando se estivesse a trabalhar em bancada. O USB teria não só como função alimentar o projeto, mas também recarregar as baterias.

3.2 Definição de padrão de interoperabilidade

Definidos os subsistemas no capítulo 3.1, investigou-se, posteriormente, como os vários projetos de *CubeSat* e *PocketQube* estabeleciam a comunicação entre os subsistemas e se já existia algum *standard* com esta finalidade. Pesquisou-se também o conceito de interoperabilidade e se este termo podia ser aplicado neste contexto, ou seja, na comunicação entre placas.

O Instituto de Engenharia Elétrica e Eletrônica (IEEE) define interoperabilidade como sendo a capacidade de dois ou mais sistemas ou componentes trocarem informação e utilizarem a informação que foi trocada [77].

Outra definição de interoperabilidade é proposta pela Organização Internacional de Normalização/Comissão Eletrotécnica Internacional (ISO/IEC) que diz ser a capacidade de comunicar, executar programas ou transferir dados entre várias unidades funcionais de uma maneira que exige que o utilizador tenha pouco ou nenhum conhecimento das características únicas dessas unidades [78].

Através destas definições é possível então entender que dois sistemas são interoperáveis se, basicamente, conseguirem trocar informação e utilizá-la [79]. Por isso, do ponto de vista deste projeto, cada placa será um sistema, e estas terão de ser interoperáveis.

No capítulo 2.4 verificou-se que uma grande parte dos projetos de *PocketQubes* utilizavam um *standard*, denominado PocketQube60 (PQ60).

O *standard* PQ60 foi criado para fornecer um ponto de partida nas missões e desenvolvimentos de *PocketQubes* [80]. O seu objetivo era também ajudar a estimular o crescimento e a progressão da comunidade relativa aos pico-satélites [80]. Uma vez que, com uma interface mecânica e elétrica *standard* para *PocketQubes*, os utilizadores finais poderiam focar-se nos elementos do *PocketQube* que mais lhe interessam [80].

O *standard* foi desenvolvido prevendo que uma pilha de sistemas seja conectada entre si para formar um *PocketQube*, onde cada sistema tem de cumprir com as restrições de uma estrutura *PocketQube* típica [80]. As conexões elétricas entre cada placa são efetuadas através de dois conectores empilhados, um na parte inferior da placa e outro na parte superior [80]. Por isso, cada sinal que faça parte do conector deve passar desde a placa do conector inferior até ao conector superior, permitindo que todos os sinais estejam presentes em todas as placas [80]. Os sinais presentes no conector do *standard* PQ60 foram selecionados para fornecerem as necessidades básicas de cada placa, mais

concretamente, de potência e comunicações [80]. O conector do *standard* PQ60 é composto por 60 pinos, e na Figura 3.2 é possível observar a descrição de cada um deles.

+3.3_Sw1	P1	P60	3V3 bus
+3.3_Sw1	P2	P59	3V3 bus
RTN_SW1_3V3	P3	P58	3V3 bus
RTN_SW1_3V3	P4	P57	3V3 bus
+3.3_Sw2	P5	P56	RTN_3V3
+3.3_Sw2	P6	P55	RTN_3V3
RTN_SW2_3V3	P7	P54	RTN_3V3
RTN_SW2_3V3	P8	P53	RTN_3V3
+3.3_Sw3	P9	P52	BatV bus
+3.3_Sw3	P10	P51	BatV bus
RTN_SW3_3V3	P11	P50	BatV bus
RTN_SW3_3V3	P12	P49	BatV bus
BatV_SW1	P13	P48	RTN_BatV
BatV_SW1	P14	P47	RTN_BatV
RTN_SW1_BatV	P15	P46	RTN_BatV
RTN_SW1_BatV	P16	P45	RTN_BatV
BatV_SW2	P17	P44	GPIO12
BatV_SW2	P18	P43	GPIO11
RTN_SW2_BatV	P19	P42	GPIO10
RTN_SW2_BatV	P20	P41	GPIO9
BatV_SW3	P21	P40	GPIO8
BatV_SW3	P22	P39	GPIO7
RTN_SW3_BatV	P23	P38	GPIO6
RTN_SW3_BatV	P24	P37	GPIO5
GPIO0	P25	P36	GPIO4
GPIO1	P26	P35	GPIO3
/RST	P27	P34	/SS/GPIO2
GND	P28	P33	SCK
SCL	P29	P32	MISO
SDA	P30	P31	MOSI

Figura 3.2 - Descrição dos pinos do conector PQ60 [80].

O *standard* fornece dois barramentos de tensão protegidos ($3.3V$ e *BatteryV*), seis linhas de energia comutadas (3x $3.3V$ e 3x *BatteryV*), dois barramentos de dados dedicados (I2C e SPI), 12 linhas *GPIO*, um pino que fornece uma linha \overline{SS} (Seleção do escravo) dedicada para o barramento SPI ou uma linha *GPIO* adicional e uma linha de *reset* [80].

Os barramentos de tensão protegidos foram projetados para sistemas de energia dentro do *PocketQube* [80]. Estas linhas serão fornecidas por um sistema de energia ou quadro de distribuição de energia [80].

As linhas de energia comutadas são projetadas para serem conectadas a sistemas de baixa corrente que são necessários, ou desejados, para serem ligados e desligados pelo utilizador [80].

No entanto, existem projetos de plataformas *PocketQube* em que começaram por utilizar o *standard* PQ60, seguindo as medidas da placa e dos furos de montagem, mas, na escolha do conector, trocaram o conector de 60 pinos por outro menor, talvez por não ser necessário uma quantidade tão

elevada de pinos. O projeto DynOSSAT-EDU da BHDynamics é um desses exemplos pois, através da sua análise (secção 2.4.5), verificou-se que nesse projeto seguiram o *standard* PQ60, mas, no que diz respeito ao conector de 60 pinos, trocaram-no por um de apenas 40 pinos.

3.3 Arquitetura de sistema

Como referido no capítulo 1.2, esta proposta de dissertação tem como um dos principais objetivos a avaliação de plataformas existentes de código aberto para pequenos satélites, e tendo por base as mesmas idealizar uma solução para um *PocketQube* que seja suficientemente flexível e adaptável para poder ser usada posteriormente como base de aviónica para diferentes tipos de payloads (instrumentação).

Através da análise dos vários projetos documentados no capítulo 2.4, devido ao facto de ser um projeto de código aberto e documentado detalhadamente, decidiu-se que a arquitetura do sistema PROMETHEUS seria baseada no sistema PyCubed (2.4.4), apesar de ser um sistema desenvolvido para satélites do tipo *CubeSat*.

Na Figura 3.3, surge uma versão modificada do diagrama de blocos da placa principal do PyCubed (Figura 2.20). Neste caso, é apresentado o diagrama de blocos do sistema idealizado, tendo como base a versão mais recente dos esquemáticos PyCubed e os requisitos definidos anteriormente para este projeto.

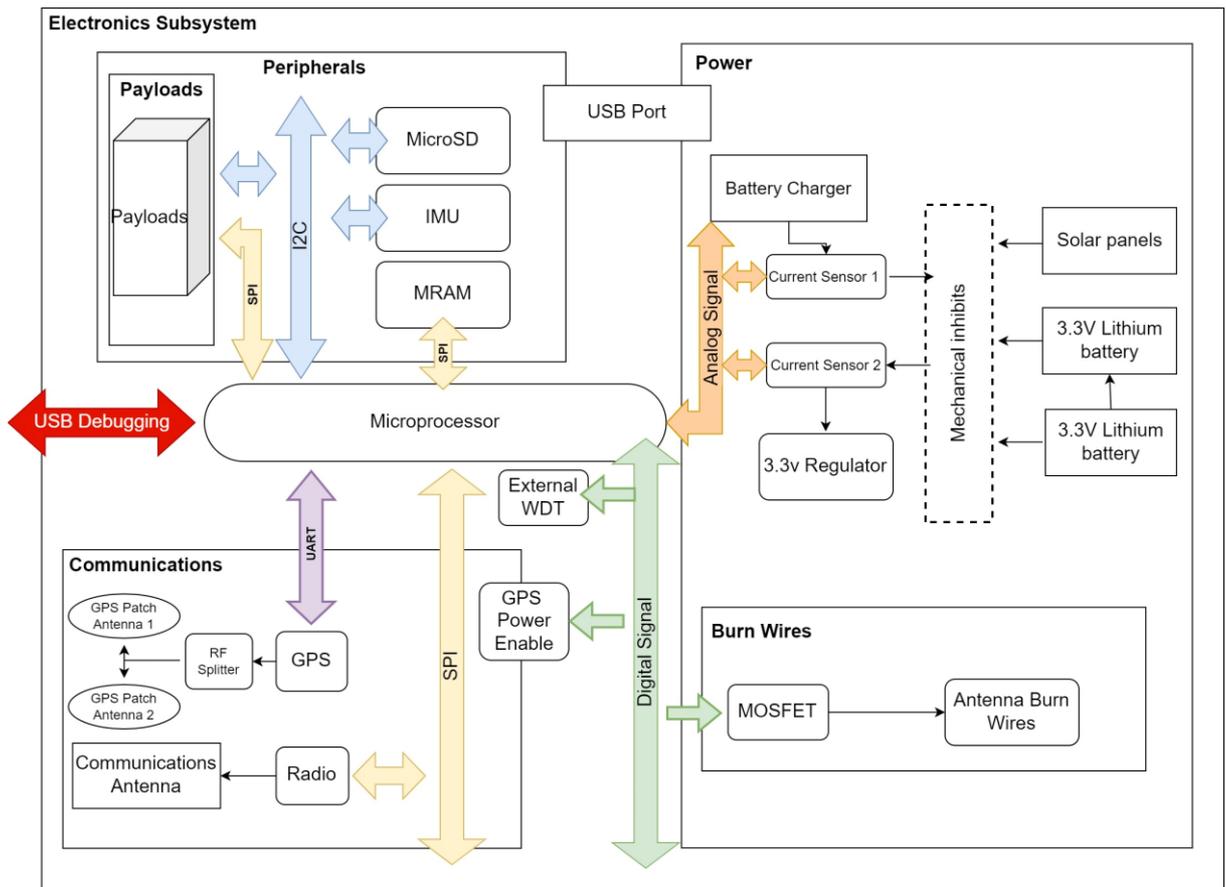


Figura 3.3 - Diagrama de blocos geral idealizado, adaptado de [49].

O diagrama de blocos da Figura 3.3 mostra que o sistema será composto por vários subsistemas. O mais importante será o que contém o microprocessador pois, como se pode observar, recebe e envia dados para todos os restantes subsistemas, desde sinais de protocolos de comunicação (I2C, SPI e UART), sinais analógicos, sinais digitais e ainda se conecta via USB para executar *debugging*. Este subsistema será o computador de bordo.

Relativamente ao sistema de Potência, percebe-se que são efetuadas duas ligações mecânicas: baterias e painéis solares. Este sistema possui também carregadores de bateria, através da energia dos painéis solares ou através da conexão USB. Contém também um regulador de tensão para 3,3 V e outro subsistema, denominado *Burn Wires*, responsável pela abertura das antenas após lançamento. O subsistema de Potência terá também uma grande tarefa de criar dois barramentos de tensão, um para a tensão da bateria e outro para 3,3 V.

O sistema das comunicações contém o rádio e o GPS, um comunica com o microprocessador através do protocolo SPI e o outro por UART, respetivamente. No entanto o GPS necessita de um sinal proveniente do microprocessador para ser ativo.

Por último existem os periféricos e os payloads, em que ambos comunicam com o microprocessador através do protocolo I2C, mas os payloads também podem utilizar o protocolo SPI.

O conector USB está presente no sistema de potência e periféricos pois, apesar de ser um periférico para fazer *debug* ao *software* do sistema, terá também como função alimentar o sistema e recarregar as baterias.

Após a análise da arquitetura, decidiu-se então quais os componentes a colocar em cada placa, atendendo a que, se possível, a placa OBC fosse independente para permitir a sua utilização de forma isolada em bancada. Para tal, a placa OBC teria de possuir sistemas de potência que transformassem a linha de tensão do USB nas duas linhas de tensão (VBATT e 3.3V) utilizadas pelos vários componentes, ter o microprocessador e seus periféricos e também um sistema de proteção do barramento das comunicações. Na Figura 3.4, observa-se como foi efetuada a divisão dos componentes pelas cinco placas da plataforma *PocketQube* e quais os sinais que são necessários transitar pelo conector de modo que estes sejam acessíveis em todas as placas.

Através da Figura 3.4, identificam-se 32 sinais que têm de passar pelo conector para poderem ser utilizados por outras placas. Estes sinais correspondem a sinais analógicos, digitais, GPIO ou então de protocolos de comunicação. Através da análise da Figura 3.4 é possível também concluir que as placas *Battery Holder* e a *Payloads* apenas necessitam de um conector, um necessita apenas na face superior, o outro na inferior, respetivamente.

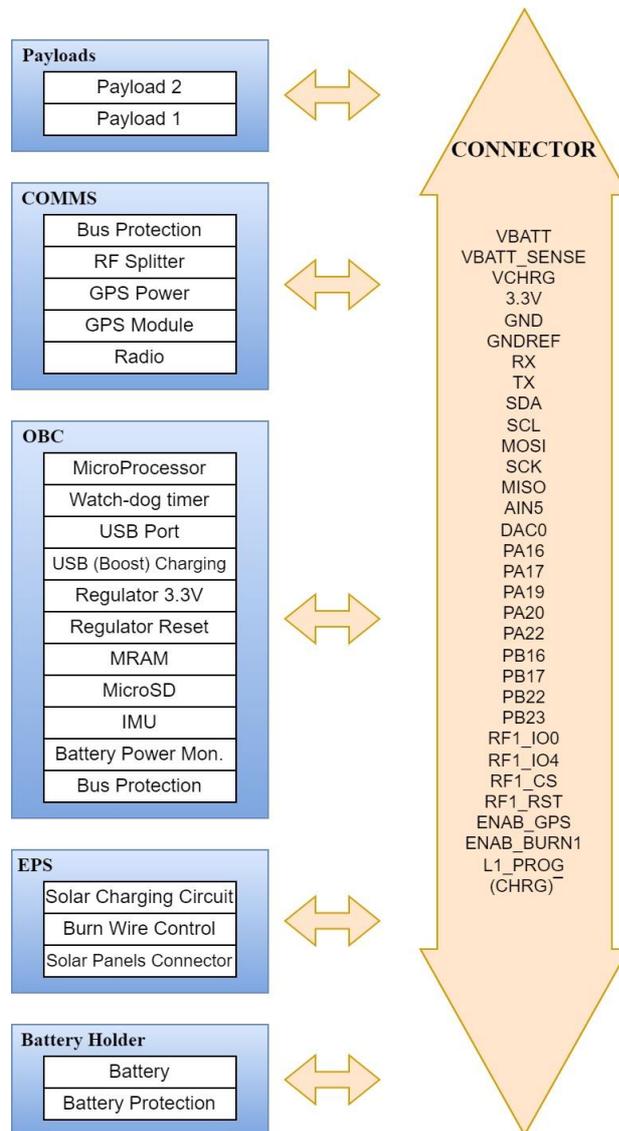


Figura 3.4 - Divisão dos componentes por placa.

No entanto, se se utilizasse o *standard* PQ60 e o seu conector de 60 pinos, iam ser pinos a mais para os 32 sinais. Deste modo, seguiu-se uma abordagem idêntica à do projeto DynOSSAT-EDU, desenvolvido pela BHDynamics, onde mantiveram as dimensões do *standard* PQ60, mas trocaram o conector por um de 40 pinos. No caso deste projeto, 40 pinos parecia um número ideal, mesmo que se tivesse de repetir alguns sinais pelos pinos do conector, era uma vantagem que se ia notar depois em fase de implementação das *PCBs* (capítulo 4).

Posto isto, criou-se um conector de 40 pinos, o PQ40, utilizado para garantir que as cinco placas eram interoperáveis. Os pinos do PQ40 podem ser observados na Figura 3.5.

L1_PROG	1	2	3.3V
VBATT	3	4	VBATT_SENSE
AIN5	5	6	SDA
GND	7	8	SCL
\overline{CHRG}	9	10	GND
RF1_IO0	11	12	MOSI
RF1_IO4	13	14	SCK
3.3V	15	16	MISO
DAC0	17	18	GND
RX	19	20	GND
TX	21	22	3.3V
ENAB_GPS	23	24	PA16
RF1_RST	25	26	PA17
ENAB_BURN1	27	28	PA19
RF1_CS	29	30	PB16
3.3V	31	32	PB17
PA20	33	34	GND
PB23	35	36	PA22
PB22	37	38	3.3V
GNDREF	39	40	VCHRG

Figura 3.5 – Definição dos pinos do conector PQ40.

3.4 Projeto dos sistemas

Neste subcapítulo é apresentado o projeto de cada subsistema ou placa da plataforma *PocketQube* tendo em conta a existência do conector PQ40 que possibilita a interoperabilidade dos sistemas. Não só será explicada a forma como o módulo funciona, mas também será apresentada uma explicação de cada componente que o constitui. Nos diagramas de blocos que serão apresentados nos capítulos a seguir, os sinais de GND e GNDREF não foram colocados, mas como estão presentes no conector PQ40, todas as placas têm acesso a eles.

Começou-se por projetar a placa OBC devido ao facto de ser a placa mais importante do sistema e, como tal, necessitava de mais atenção na análise dos sinais provenientes de outros subsistemas.

Também como se idealizou projetar esta placa independente das outras, era de esperar que fosse a placa com maior número de componentes. O desenvolvimento das placas foi realizado através do *software* de código aberto Kicad EDA, com o intuito de disponibilizar todo o projeto de forma livre, cumprindo as especificações de *open-access*.

3.4.1 OBC

Como referido anteriormente, na placa OBC, uma das particularidades que se queria implementar era que esta placa fosse independente. Isto é, que fosse possível utilizar esta placa em bancada sem que dependesse de outra(s) conectada(s). Os componentes que constituem esta placa foram todos baseados no sistema PyCubed, e são: o microprocessador, *watch-dog timer*, IMU, carregador USB para duas células de Li-ion, regulador 3.3V, monitor de bateria, conector USB-C, espaço para cartão microSD, MRAM, sistemas de proteção de barramento de dados e o conector PQ40.

Na Figura 3.6, é apresentado o diagrama de blocos da placa OBC, onde é possível identificar quais os sinais que cada componente utiliza, quais os sinais que passam pelo conector PQ40, qual a linha de alimentação que o componente necessita, protocolos de comunicação e ainda quais os sinais digitais enviados e recebidos pelo microprocessador. Percebe-se também que esta placa pode ser trabalhada em bancada de forma independente das outras, desde que ligada a USB, pois, consegue gerar as duas linhas de tensão e operar o sistema de *software*.

Relativamente ao esquemático da placa OBC, foi dividido em quatro páginas (página principal, circuitos de potência, conectores e circuitos de proteção) que podem ser consultadas no Anexo A – Esquemático OBC, para mais fácil compreensão e organização. A primeira página, ou seja, a principal, do esquemático da placa OBC contém o microprocessador, o *watch-dog timer*, o IMU e os símbolos para os furos de montagem da placa.

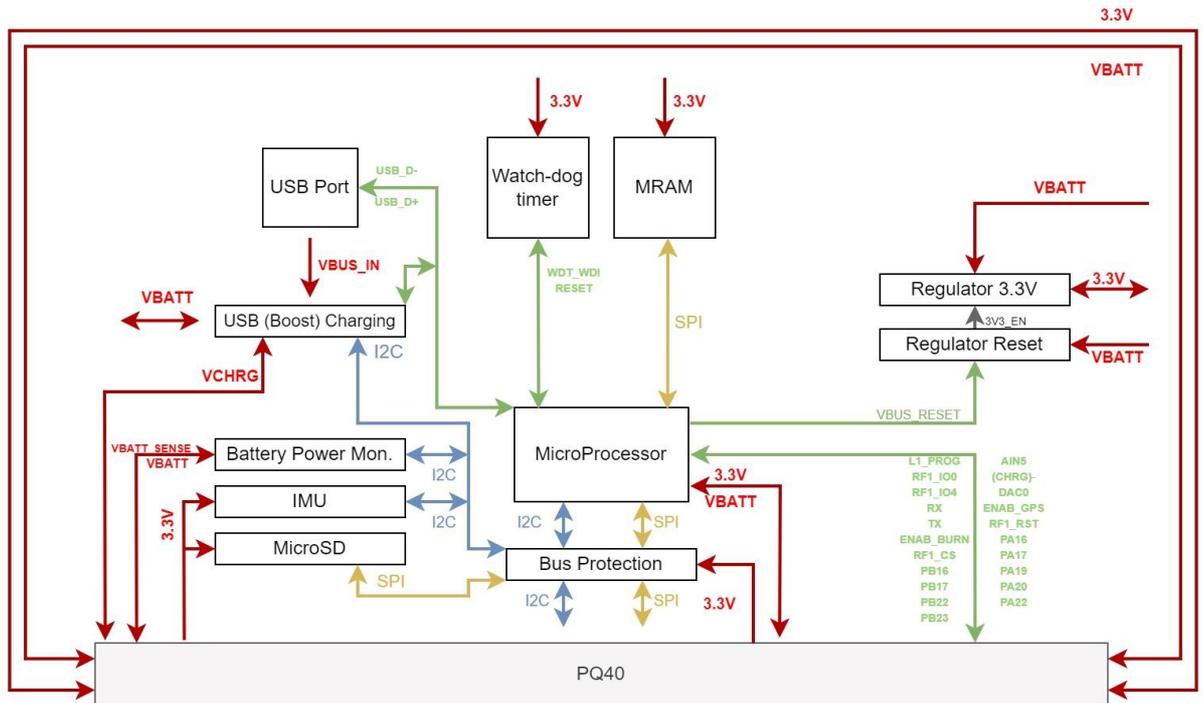


Figura 3.6 - Diagrama de blocos OBC.

O microcontrolador selecionado é da família ATSAMD5X, mais concretamente o ATSAMD51. Este microcontrolador é utilizado como “cérebro” também na versão mais recente da placa PyCubed. É controlado por USB e possui um processador de 32-bit ARM M4 com uma frequência de 120 MHz, capaz de configurar os seus pinos para executar uma vasta variedade de funções de *hardware*, tais como comunicação, ADC, DAC, entre outros [49]. Observa-se que no pino 42 do microprocessador está conectado um led RGB que tem como função indicar em que estado se encontra o interpretador Python em tempo real (CircuitPython).

O *external watch-dog timer MAX706RESA* é um circuito de supervisão de microprocessadores e, normalmente, este tipo de circuito fornece as seguintes quatro funções: (i) efetuar um *reset* quando o sistema é ligado, desligado ou quando há quebras de energia; (ii) uma saída *watch-dog* independente que fica num estado baixo se a entrada *watch-dog* não for alternada dentro de 1,6 s; (iii) um detetor de limite de 1,25 V para aviso de falha de energia, deteção de bateria fraca ou para monitorizar uma fonte de alimentação diferente da fonte principal; (iv) uma entrada de *reset* manual [81]. Neste sistema, o *external watch-dog timer* espera uma transição do estado da *flag WDT_WDI* pelo menos a cada 1,6 s, senão coloca a variável \overline{RESET} no estado baixo. Por sua vez, esta variável está conectada ao pino 52 (RESETN) do microprocessador e faz um *reset* de energia ao computador de bordo, neste caso, ao

microprocessador. Este *reset* também pode ser efetuado manualmente através do botão RESET, que por defeito está no estado alto.

Ainda na primeira página do esquemático da placa OBC, existe outro componente denominado IMU (BMX160), um integrado que contém um sensor de temperatura, acelerómetro, um magnetómetro e um giroscópio. Estes valores são normalmente utilizados nas aeronaves no sistema de determinação e controlo de atitude, mas, este projeto não contém atuadores. No entanto, se for pretendido, pode ser adicionada uma placa para determinação e controlo de atitude ao *PocketQube*, apenas sendo necessário projetar a placa com o mesmo conector (PQ40) que as outras e fazer uso dos pinos GPIO ainda lá disponíveis.

Os orifícios de montagem irão estar presentes no esquemático de todas as placas pois estes são os furos responsáveis por garantir que estas sejam empilhadas de forma robusta, utilizando-se um em cada canto da placa.

Na segunda página do esquemático da placa OBC, encontram-se os conectores: PQ60, conector USB, cartão microSD e MRAM.

O MR25H40 é um dispositivo de memória magnetorresistivo não volátil (MRAM) de 4 MB da Everspin Technologies [49]. A alta tolerância à radiação da memória magnética no PyCubed é usada para código de memória não volátil confiável e armazenamento de dados, bem como RAM para o microprocessador ATSAM51 durante a operação do interpretador Python em tempo real [49]. Outro componente para armazenamento de dados é colocado na placa OBC, na forma de entrada para microSD.

Um conector USB-C é também colocado na placa OBC para permitir ao utilizar operar o sistema e fazer *debugging*, fornecer uma linha de tensão que será utilizada para alimentar o sistema e também recarregar as baterias.

Para finalizar os conectores da placa OBC, vem o conector PQ40, responsável pelo empilhamento e pela interoperabilidade entre placas. Foram colocados dois conectores porque é um em cada face da *PCB*, pois esta placa está localizada a meio da pilha.

Na terceira página do esquemático da placa OBC, denominada *Power Management*, encontram-se os restantes circuitos que tornam a placa OBC operável, independente das restantes.

Primeiramente, temos o carregador USB de 2 A para duas células de Li-ion, BQ25883. Este dispositivo tem como função alimentar o sistema ou recarregar as baterias, ambas as funcionalidades através da conexão USB. Esta seleção é efetuada com uns *jumpers*, como é possível observar na Figura 3.7. Segundo o *datasheet* do componente, por defeito este carrega as baterias a 1500 mA, o que não convém quando se utiliza baterias com capacidade de 500mAh (abordadas na secção 3.4.4).

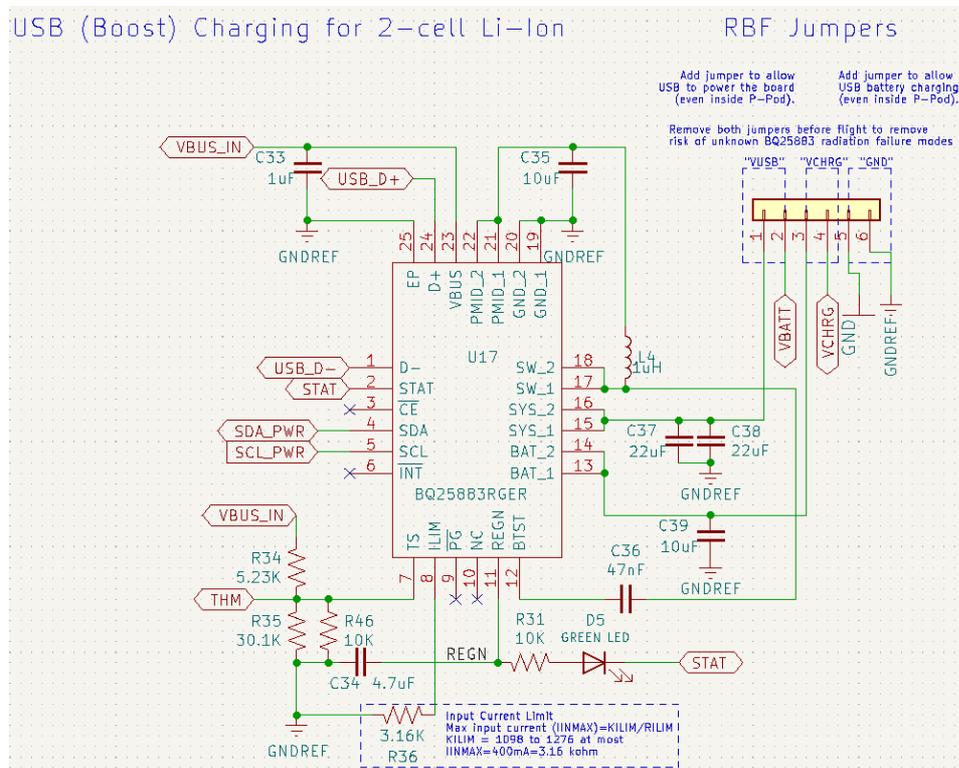


Figura 3.7 – Circuito do carregador USB para 2-células de Li-ion.

Para tal, é necessário alterar o registo REG01 (Registo de Limite de Corrente do Carregador), apresentado na Figura 3.8.

Bit	Field	Type	Reset by REG_RST	Reset by WATCHDOG	Description
7	EN_HIZ	R/W	Yes	Yes	Enable HIZ Mode: 0 – Disable (default) 1 – Enable
6	EN_ILIM	R/W	Yes	Yes	Enable ILIM Pin Function: 0 – Disable 1 – Enable (default)
5	ICHG[5]	R/W	Yes	Yes	1600 mA
4	ICHG[4]	R/W	Yes	Yes	800 mA
3	ICHG[3]	R/W	Yes	Yes	400 mA
2	ICHG[2]	R/W	Yes	Yes	200 mA
1	ICHG[1]	R/W	Yes	Yes	100 mA
0	ICHG[0]	R/W	Yes	Yes	50 mA

Fast Charge Current Limit
Offset: 100 mA
Range: 100mA – 2200mA
Default 1500 mA
Note: ICHG > 2.2A (2Ch) clamped to 2.2A. ICHG < 100mA (01h) clamped at 100mA

Figura 3.8 - Registo REG01.

Por norma não se deve carregar as baterias com uma corrente superior à sua capacidade, logo a opção mais adequada seria limitar a corrente a 400 mA, ou seja, ativar o bit 3 do registo REG01 do BQ25883, correspondendo a “8” em numeração decimal. Esta operação é efetuada na pasta *software* da PyCubed, em *software/mainboard-v05/lib/pycubed.py* através da linha de código 140 “*self.usb.charging_current=8*”, que por acaso a PyCubed já a tinha assim definido. Em *software/mainboard-v05/lib/bq25883.py* é também inicializado o carregador USB BQ25883 com uma corrente de entrada máxima de 500 mA, através da linha de código 100 “*self._iinlim=0*”.

O circuito de carregamento das baterias por USB é aliado de um conector denominado *RBF Header*. Este conector tem como função decidir se o USB alimenta a plataforma, se carrega as baterias ou ambos. As ligações para os diferentes modos de utilização podem ser observadas na Figura 3.9.

		GND	VCHRG	VUSB
Utilização em bancada	Sem bateria (alimentado por USB)	✓	✗	✓
	Com bateria	✗	✗	✗
Carregamentos das baterias através de USB	Em bancada	✓	✓	✗
Carregamentos das baterias através dos Painéis solares	Em bancada	✗	✗	✗

Figura 3.9 - Utilização do conector *RBF Header*, adaptado de [49].

O regulador TPS54226 é um regulador que através de uma tensão de entrada entre 4,5 V e 18 V, gera uma tensão de saída DC entre 0.76 V e 5.5 V. A alteração da variação da saída era alterada através da mudança dos valores dos componentes ligados a este dispositivo. Neste projeto, seguiu-se os valores utilizados na PyCubed, que foram seleccionados para uma tensão de saída de 3,3 V, através do *datasheet* do componente. A este componente foi adicionado outro sistema que permite fazer um *reset* ao barramento 3,3 V através de um sinal proveniente do microprocessador.

Por último, a presença do *battery power monitor* nesta placa deve-se ao facto deste componente utilizar o protocolo de comunicação I2C, assim o sistema de proteção deste barramento I2C fica todo na mesma placa. O *battery power monitor* tem como função medir a tensão de entrada e o consumo de corrente das baterias.

Na última página do esquemático OBC surgem os sistemas de proteção para os barramentos de comunicação I2C e SPI, documentados na secção 2.4.4.

3.4.2 EPS

A placa *EPS* é constituída por um circuito de carregamento solar das baterias, conector PQ40, conectores para os painéis solares, conectores para bateria e pelos *burn wires*. O seu diagrama de blocos é apresentado na Figura 3.10 e o seu esquemático encontra-se no Anexo B – Esquemático EPS.

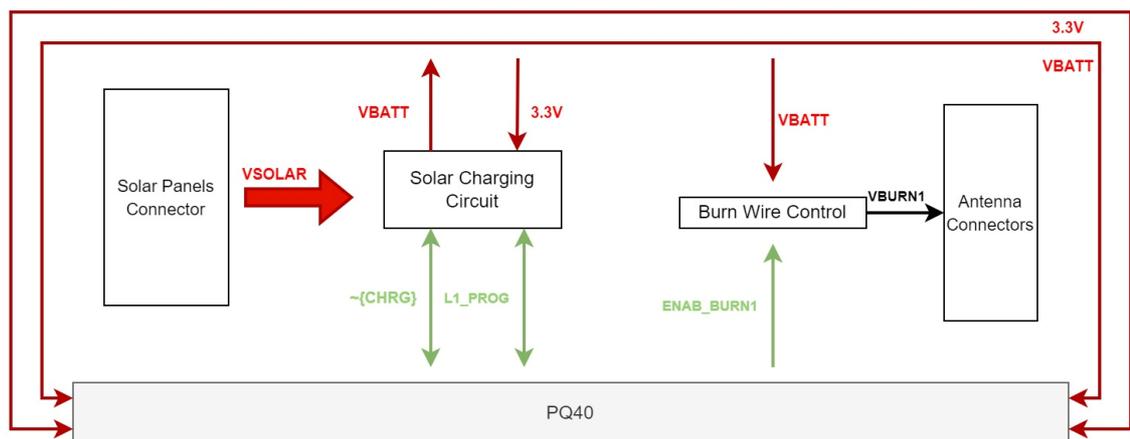


Figura 3.10 - Diagrama de blocos EPS.

Através da análise da Figura 3.10, entende-se as funcionalidades da placa EPS. Esta não só transforma energia solar numa linha de tensão VBATT para carregar as baterias e por sua vez alimentar o circuito, como também possui um circuito de *burn wires*, abordado na secção 2.4.4, responsável pela abertura das antenas depois do satélite ser lançado.

O sistema PyCubed possui um circuito de recolha de energia solar de alta eficiência configurado para carregar baterias Li-ion 2S, ou seja, duas células em série. Utilizaram o componente LTC4121 que é um carregador de bateria síncrono de 400 mA de corrente constante/tensão constante (CC/CV). Segundo o seu *datasheet* [82], o LTC4121 regula a sua tensão de entrada para uma percentagem

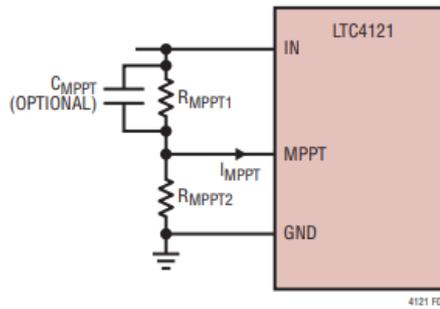


Figura 3.12 - Divisor Resistivo MPPT LTC4121 [82].

O divisor de tensão programável (K_r) tem como função definir a fração da tensão de entrada que aparece no pino MPPT, e é dado pela expressão (2):

$$K_r = \frac{R_{MPPT2}}{R_{MPPT1} + R_{MPPT2}} = \frac{V_{MPPT}}{V_{IN}} \quad (2)$$

Sabendo que existe também um divisor interno ($K_F = \frac{1}{10}$), calcula-se o K_r através de (3):

$$K_r = \frac{K_F}{75\%} = \frac{0,1}{0,75} = 0,1333 \quad (3)$$

Utilizando o esquemático da Figura 3.12, o rácio entre R_{MPPT1} e R_{MPPT2} é obtido pela expressão (4):

$$R_{MPPT1} = \frac{1 - (\frac{K_F}{75\%})}{(\frac{K_F}{75\%})} \times R_{MPPT2} = 6,5 \times R_{MPPT2} \quad (4)$$

Utilizando resistências *standard*, vem que $R_{MPPT1}=787 \text{ k}\Omega$ e $R_{MPPT2}=121 \text{ k}\Omega$. E que por sua vez equivalem às resistências R11 e R12 do circuito da Figura 3.11, respetivamente.

Com a utilização deste circuito de carregamento, é necessário também definir qual a tensão na qual o sistema cancela o carregamento, ou seja, quando a bateria está totalmente carregada. Esta tensão (V_{FLOAT}) é programada através do circuito da Figura 3.13, onde é colocado um divisor resistivo desde a bateria até ao pino FB e FBG. A tensão V_{FLOAT} da bateria é programável entre 3,5 V e 18 V, e é dada pela equação (5):

$$V_{FLOAT} = V_{FB(REG)} \times \frac{(R_{FB1} + R_{FB2})}{R_{FB2}} \quad (5)$$

Onde $V_{FB(REG)}$ é normalmente 2,4 V.

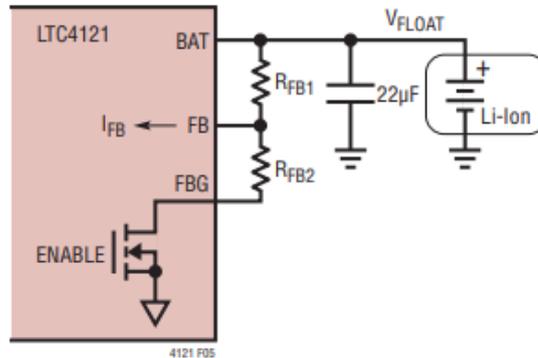


Figura 3.13 - Circuito para programar a tensão de carregamento completo do LTC4121 [82].

No *datasheet* do LTC4121 [82], existem duas tabelas que indicam os valores recomendados de R_{FB1} e R_{FB2} para cada tensão V_{FLOAT} . Tendo em conta que neste projeto seriam utilizadas duas baterias de 3,7 V em série, a tensão máxima quando estas estão totalmente carregadas é de 8,4 V, uma vez que cada célula pode atingir 4,2 V totalmente carregada. Logo, pela análise da tabela presente em [82], os valores de R_{FB1} e de R_{FB2} seriam de 2,1 M Ω e 845 k Ω , respetivamente. Voltando a utilizar a equação (5) com estes novos valores, resulta que $V_{FLOAT} = 8,36$ V. No esquemática da Figura 3.11, R19 corresponde a R_{FB1} e R16 a R_{FB2} .

O pino *PROG* é utilizado para definir a corrente máxima de carregamento, mas também pode ser utilizado para monitorizar ativamente a corrente de carregamento que entra nas baterias [49]. No entanto, este valor é apenas válido quando o sinal *CHRG* está baixo [49].

Esta placa possui também um conector capaz de desconectar a bateria, Figura 3.14, que possui três conexões, duas *high side* e uma *low side*.

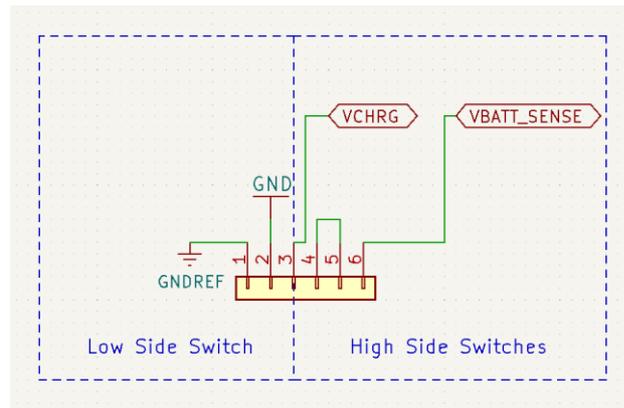


Figura 3.14 - Conector baterias placa EPS.

Estas conexões são efetuadas mediante o modo de utilização da plataforma, isto é, se está a ser utilizada em bancada com ou sem bateria e se as baterias estão a ser carregadas por USB ou pelos painéis solares. A configuração do conector para as várias formas de utilização da plataforma pode ser observada na Figura 3.15.

		High Side	Low Side
Utilização em bancada	Sem bateria (alimentado por USB)	✗	✗
	Com bateria	✓	✓
Carregamentos das baterias através de USB	Em bancada	✗	✗
	Em bancada	✓	✓

Figura 3.15 - Utilização do conector das baterias, adaptado de [49].

Ainda na placa EPS, surge na segunda página do esquemático, presente em Anexo B – Esquemático EPS, um sistema denominado *Burn Wires* que foi explicado na secção 2.4.4. No entanto, retirou-se o *relay* e o circuito é ativo apenas por um sinal proveniente do microprocessador.

3.4.3 COMMS

A placa COMMS é constituída por componentes capazes de receber e transmitir informação. É composta por um módulo de rádio, GPS, um divisor de frequência e as suas respetivas antenas.

O diagrama de blocos do módulo COMMS pode ser observado na Figura 3.16.

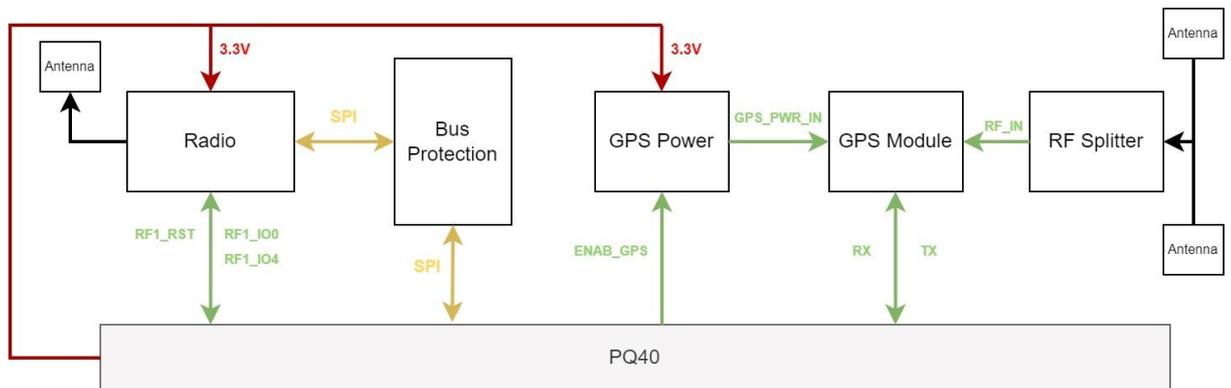


Figura 3.16 - Diagrama de blocos COMMS.

Esta placa permite que o sistema comunique com uma estação terrestre, através do rádio, e também saiba a sua latitude, longitude, a data e hora através do módulo GPS. O esquemático da placa COMMS está presente no Anexo C – Esquemático COMMS.

O rádio selecionado é um módulo LoRa RFM98W, frequência 433 MHz, fornece comunicação de longo alcance e alta imunidade a interferências, minimizando também o consumo de corrente.

3.4.4 Battery Holder

A placa *Battery Holder* tem como função suportar as duas baterias e alimentar o sistema. Na Figura 3.17 é apresentado o diagrama de blocos desse módulo e o seu esquemático pode ser consultado no Anexo D – Esquemático Battery Holder.

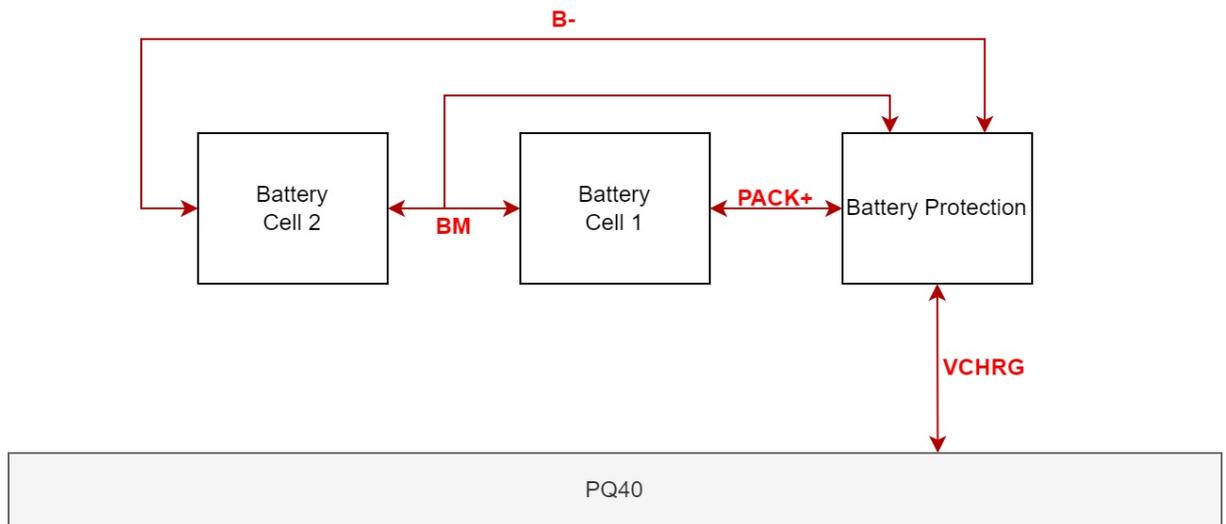


Figura 3.17 - Diagrama de blocos Battery Holder.

O sistema PyCubed utilizou um *pack* de 6 baterias de Li-ion, na configuração 2s3p. Quando estas estavam totalmente carregadas, poderiam rondar uma tensão de 8,4 V e, por isso, o sistema foi projetado para tal. No entanto, como este projeto se trata da idealização de uma plataforma *PocketQube*, as dimensões disponíveis são bastante menores que as de um *CubeSat*, pelo que não foi disponível utilizar o mesmo *pack* do PyCubed.

Portanto, a única solução para que fosse possível continuar a utilizar o sistema PyCubed como base era procurar células de bateria de 3,7 V que tivessem dimensões adequadas à plataforma *PocketQube* e colocá-las em série. A bateria utilizada pela BHDynamics no DynOSSAT-EDU é uma bateria Li-po 603030 3,7 V/500 mAh. Esta bateria tem dimensões de 30x30x6 mm (comprimento x largura x altura). Existe também a Li-po 802530 3,7 V/500 mAh que já possui as dimensões 30x25x8 mm, entre outras que encaixam no tamanho da placa *PocketQube*. No entanto, é preciso especial atenção sobre a capacidade das mesmas, pois este projeto foi desenvolvido para carregar as baterias com uma corrente máxima de 400 mA, logo se for utilizada uma bateria com uma capacidade menor que 400 mAh, podem acontecer anomalias e consequente avaria da bateria. Para tal, os registos abordados na secção 3.4.2 devem ser alterados.

Posto isto, o outro circuito que constitui a placa *Battery Holder* é um integrado que tem como função a proteção de células de Li-ion/Li-po colocadas em série. Como se referiu anteriormente, este projeto irá utilizar duas células de bateria em série para atingir uma tensão VBATT de 8,4 V. Mas o carregamento de duas células em série necessita de cuidados, e por isso foi utilizado o integrado R5460N233AF.

Na Figura 3.18 é possível observar um diagrama de blocos da ligação das células das baterias em série, e os respetivos sinais.

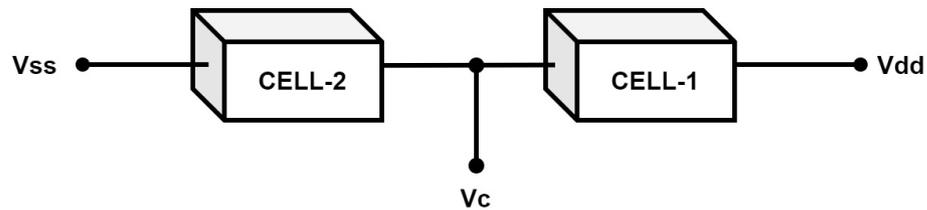


Figura 3.18 - Diagrama células de bateria em Battery Holder.

Sabe-se que a tensão entre Vdd e Vc é a tensão da célula 1. E a tensão entre Vc e Vss é a tensão da célula 2. Definidos estes sinais, implementou-se o esquemático utilizado no *datasheet* do integrado [83], como se pode observar na Figura 3.19.

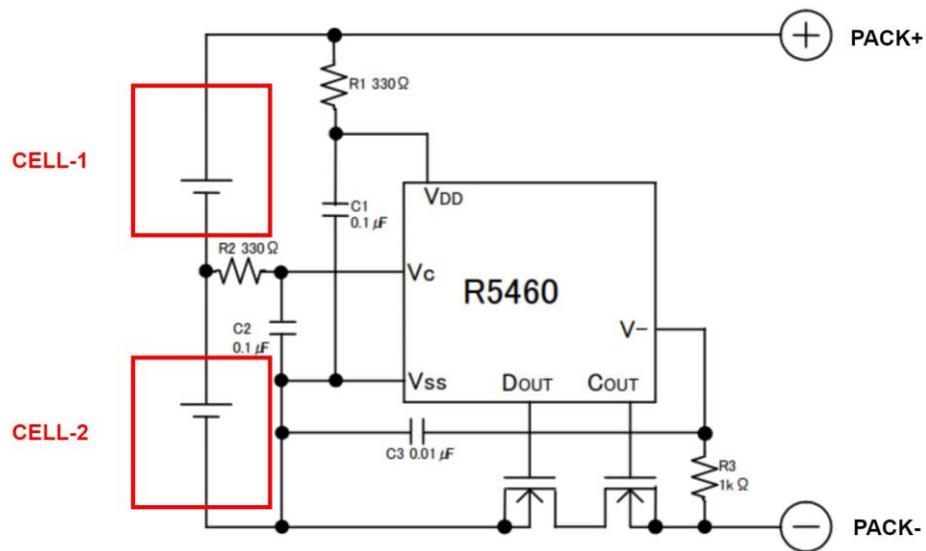


Figura 3.19 - Circuito de aplicação típica do IC de proteção de baterias R5460, adaptado de [83].

Através da monitorização da tensão das células das baterias, o integrado é capaz de desempenhar as seguintes funções: detetar sobrecargas, detetar descargas excessivas, proteção contra curto-circuitos, detetar uma corrente de carga excessiva e operar contra duas células não balanceadas [83].

3.4.5 Payloads

A placa *Payloads* tem, como o próprio nome indica, a função de suportar os módulos *payload* responsáveis por desempenhar a função da missão. Na Figura 3.20 é possível observar o diagrama de blocos da placa *Payloads* e o seu esquemático está presente em Anexo E – Esquemático *Payloads*.

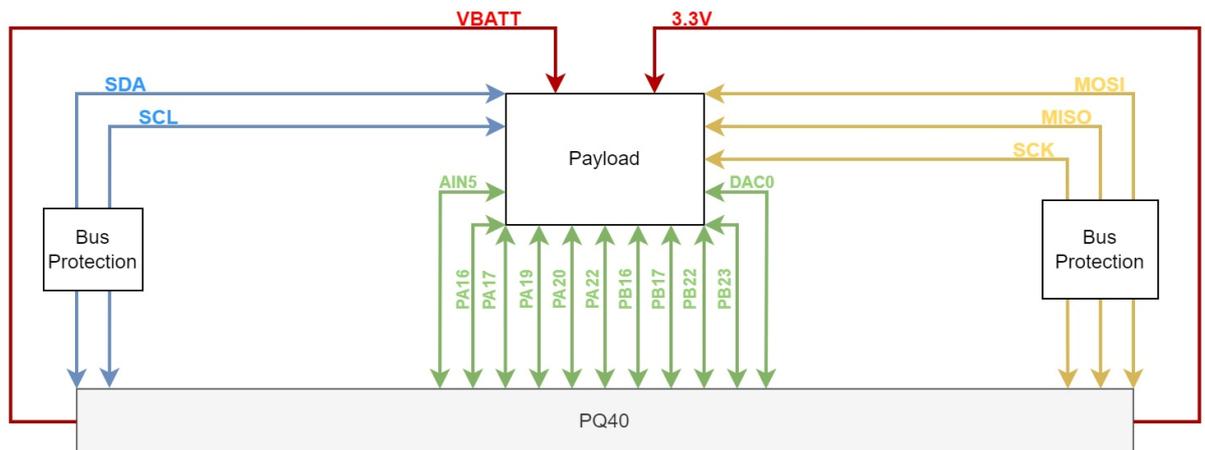


Figura 3.20 - Diagrama de blocos *Payloads*.

Através do conector *PQ40*, esta placa fornece acesso a 3,3 V, *VBATT*, *GND*, *GNDREF*, *I2C*, *SPI*, *UART* e pinos *GPIO* que trocam informações com o microprocessador presente na placa *OBC*. No diagrama da Figura 3.20, não estão representados os sinais do protocolo *UART*, no entanto, os pinos *PA16* e *PA17* desempenham funções diferentes dos outros pinos *GPIO*, sendo que também podem ser utilizados como sinais do protocolo *UART*.

Como referido na secção 2.4.4, sempre que se adicionar um *payload*, tem de se projetar o sistema de proteção para o barramento de comunicação que este utilizar, tal como acontece na placa *OBC*.

4. PROJETO DE IMPLEMENTAÇÃO

Este capítulo descreve o desenvolvimento das *PCBs* (circuitos impressos) respectivas a cada placa abordada no capítulo 3 e as simulações de alguns circuitos de potência utilizados para fornecer os dois barramentos de tensão.

4.1 Desenvolvimento das PCBs

No capítulo 3, decidiu-se que cada placa desta plataforma *PocketQube* ia seguir as dimensões utilizadas pelo *standard* PQ60. No entanto, também se referiu que devido ao facto de não haver necessidade de ter um conector com 60 pinos, trocou-se este por um conector de 40 pinos, idêntico ao do projeto DynOSSAT-EDU (2.4.5), que se denominou de PQ40 para efeitos de documentação do projeto.

A Figura 4.1 apresenta um desenho técnico que fornece as informações sobre as medidas máximas da placa e as localizações dos pontos de montagem e conector. Inicialmente, referiu-se que uma plataforma *PocketQube* possui um formato cúbico com apenas 5 cm de lado, mas, como se pode observar na Figura 4.1, cada placa terá as medidas de um quadrado com apenas 42 mm de lado, o que será um desafio técnico projetar todas as placas com estas medidas. Os 5 cm serão abrangidos com as placas de espessura 0,8 mm que irão conter os painéis solares.

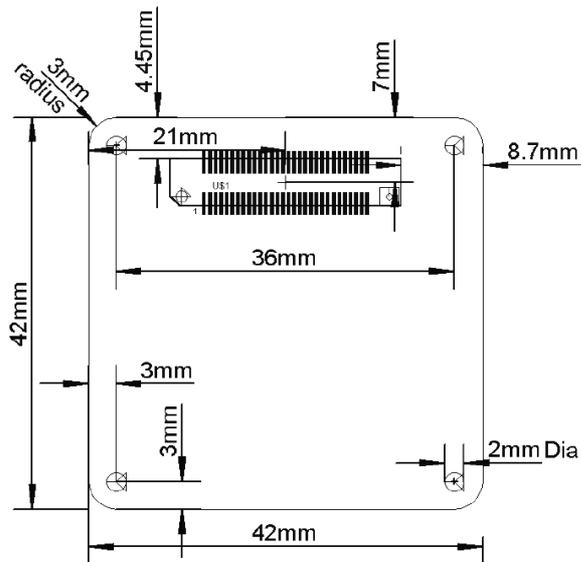


Figura 4.1 - Desenho técnico do standard PQ60 [80].

Relativamente ao sistema PyCubed, a *PCB* foi projetada para possuir uma impedância controlada de 50Ω , utilizando o empilhamento de 4-camadas de FR408HR, apresentado na Figura 4.2 [48]. Esta forma de empilhamento reduz a emissão e a absorção de EMI, mantendo a integridade do sinal e a robustez térmica [48]. As quatro camadas foram divididas em:

- Potência/Sinal
- *Ground*
- Sinal
- *Ground*

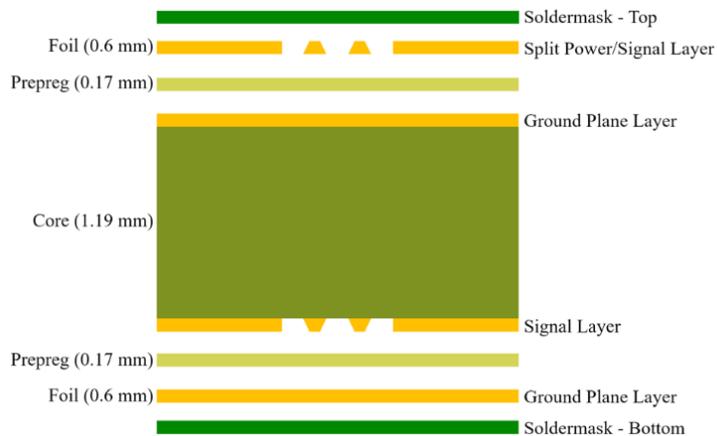


Figura 4.2 – Design das 4 camadas de PCB do sistema PyCubed [48].

Desta forma, antes de se começar a posicionar os componentes nas placas, analisou-se as capacidades e regras de *design* de alguns fabricantes de *PCBs* para garantir um certo grau de compatibilidade com o processo de fabrico. Os fabricantes que foram analisados são a *pcbway*, *eurocircuits* e *jlpcb*, e as suas capacidades e regras de *design* podem ser observadas na Tabela 3.

Tabela 3 - Regras de *design* dos fabricantes *pcbway*, *jlpcb* e *eurocircuits*.

	pcbway	jlpcb	eurocircuits
Número de camadas	1-10	1;2;4;6	2-16
Material	FR-4, Alumínio	FR-4	FR-4
Espessura da placa [mm]	0,2-2,4	0,4/0,6/0,8/1,0/1,2/1,6/2,0	0.2-3.2
Espaçamento mínimo [mm]	0,1	0,127 (1-2 camadas) 0,09 (4-6 camadas)	0,09
Largura mínima da pista [mm]	0.1	0,127 (1-2 camadas) 0,09 (4-6 camadas)	0,09
Largura anular mínima [mm]	0,15	0,13	0,1
Diâmetro mínimo da via [mm]	0,2	0,4	0,45
Espaço entre cobre e furo [mm]	-	0,2	0,2
Espaço entre cobre e berma [mm]	0,2	0,2	0,25
Diâmetro interior mínimo de furo [mm]	0,2	0,2	0,45
Espaçamento entre furos [mm]	0,4	0,5	-

Após analisar as características da tabela, analisou-se também que medidas eram utilizadas nas regras de *design* das *PCBs* da PyCubed e do DynOSSAT-EDU. Assim sendo, definiu-se que as placas da plataforma *PocketQube* iam ter uma espessura de 1,6 mm e as mais complexas iam ser compostas por 4 camadas, as mais simples tentar-se-ia apenas com 2. Já os valores das regras de *design*, escolheu-se os valores mínimos que satisfizessem qualquer um dos fabricantes abordados.

4.1.1 OBC

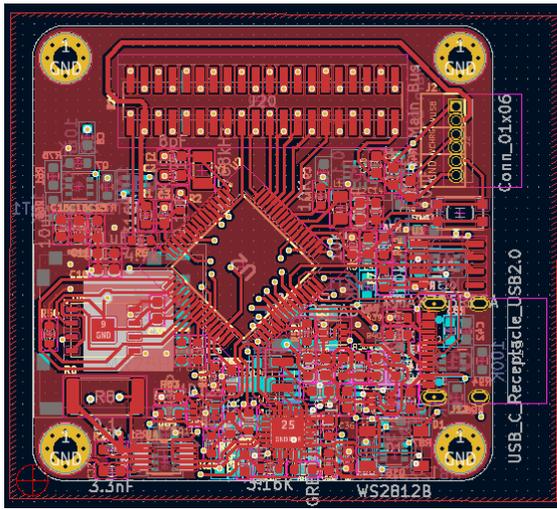
A primeira placa a ser projetada foi a OBC, pois os restantes módulos iam depender dos componentes que se conseguiam colocar nesta placa. Como foi referido na secção 3.4.1, esta placa foi idealizada para ser independente, mas, devido às restrições de dimensões, podia não ser possível colocar todos os componentes nem fazer as respetivas ligações. Considerando-a uma placa complexa, foi utilizada uma *PCB* de espessura 1,6 mm de 4 camadas, sendo que cada camada é atribuída a um tipo específico de sinal, numa divisão igual à utilizada no PyCubed: Potência/Sinal (F.Cu); *Ground* (In1.Cu); Sinal (In2.Cu); *Ground* (B.Cu).

Os componentes foram divididos pela parte frontal e traseira. Na parte frontal foram colocados o microprocessador, o *Watch-dog timer*, *Led RGB*, carregador e conector USB, monitor de bateria, MRAM e proteção do barramento I2C. Já na parte traseira foram colocados o IMU, regulador 3,3 V, MicroSD e proteção do barramento SPI.

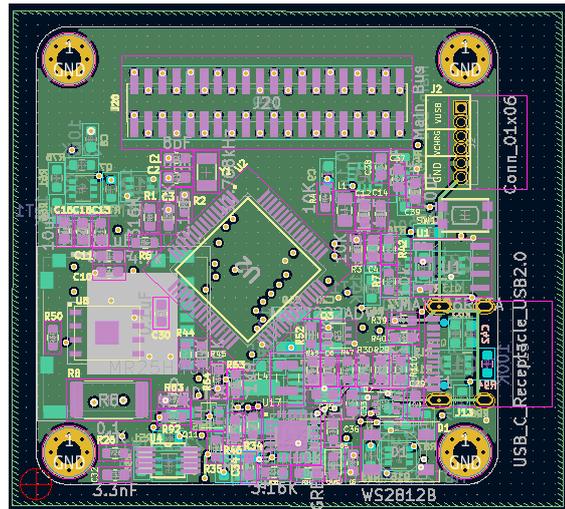
Na Figura 4.3 é possível observar o *layout* da placa OBC, bem como o layout das suas camadas. Observa-se que o microprocessador ocupa a posição central de forma estratégica.

Na camada F.Cu foram efetuadas as ligações de potência e sinal (Figura 4.3.a), na camada In1.Cu um plano de *Ground* (Figura 4.3.b), na camada In2.Cu ligações maioritariamente de sinal para o conector e algumas de potência (Figura 4.3.c), na camada B.Cu um plano de *Ground* e ligações de componentes colocados na parte traseira (Figura 4.3.d).

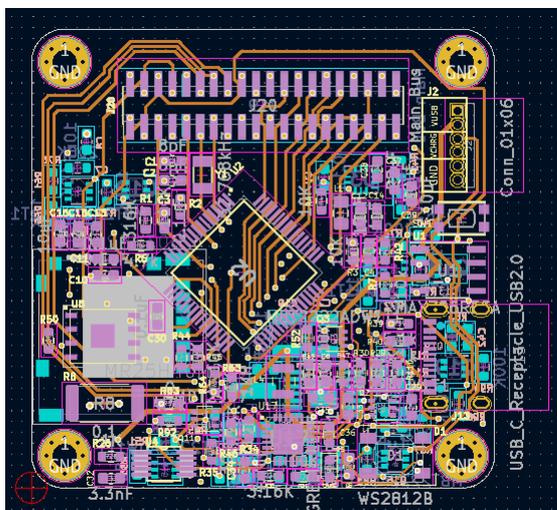
Na Figura 4.4 são apresentadas as duas faces de uma visão a 3 dimensões da placa OBC. Foram também estabelecidas algumas conexões para apresentar a localização dos principais componentes que constituem esta placa.



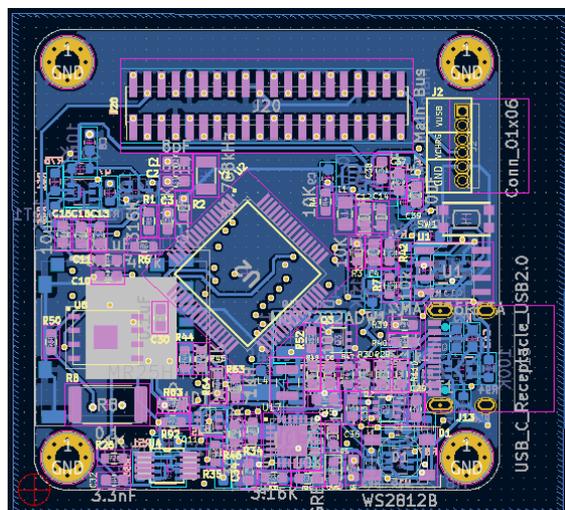
(a)



(b)

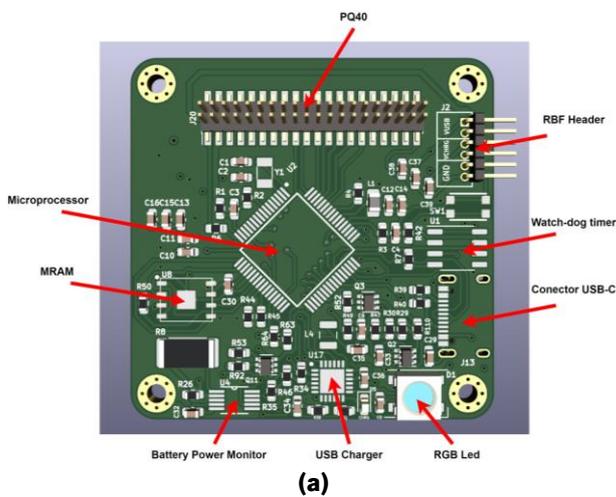


(c)

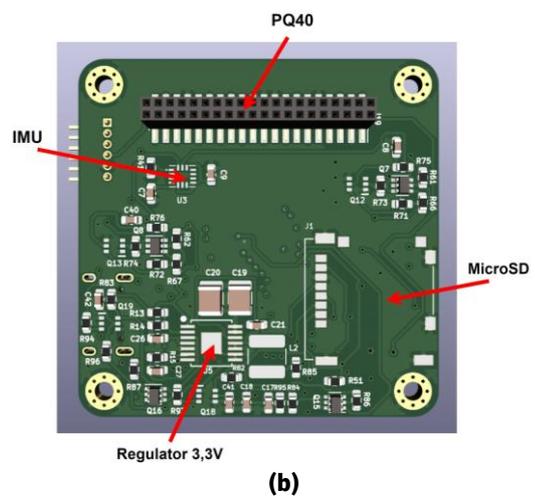


(d)

Figura 4.3 – Layout da placa OBC: (a) Camada F.Cu pcb OBC; (b) Camada In1.Cu pcb OBC; (c) Camada In2.Cu pcb OBC; (d) Camada B.Cu pcb OBC.



(a)



(b)

Figura 4.4 - Visão 3D da (a) parte frontal da placa OBC e (b) da parte traseira da placa OBC.

4.1.2 EPS

Na placa EPS, tal como a OBC, também foi necessária a utilização de 4 camadas, devido à sua complexidade. Na parte frontal da placa foram colocados os conectores da bateria, conectores dos painéis solares, e os *Burn Wires*. Na parte traseira, apenas o circuito de carregamento solar das baterias. Na Figura 4.5 é apresentado o *layout* da placa do sistema EPS, onde é possível observar que todos os conectores foram colocados nas periferias da placa, para fácil acesso.

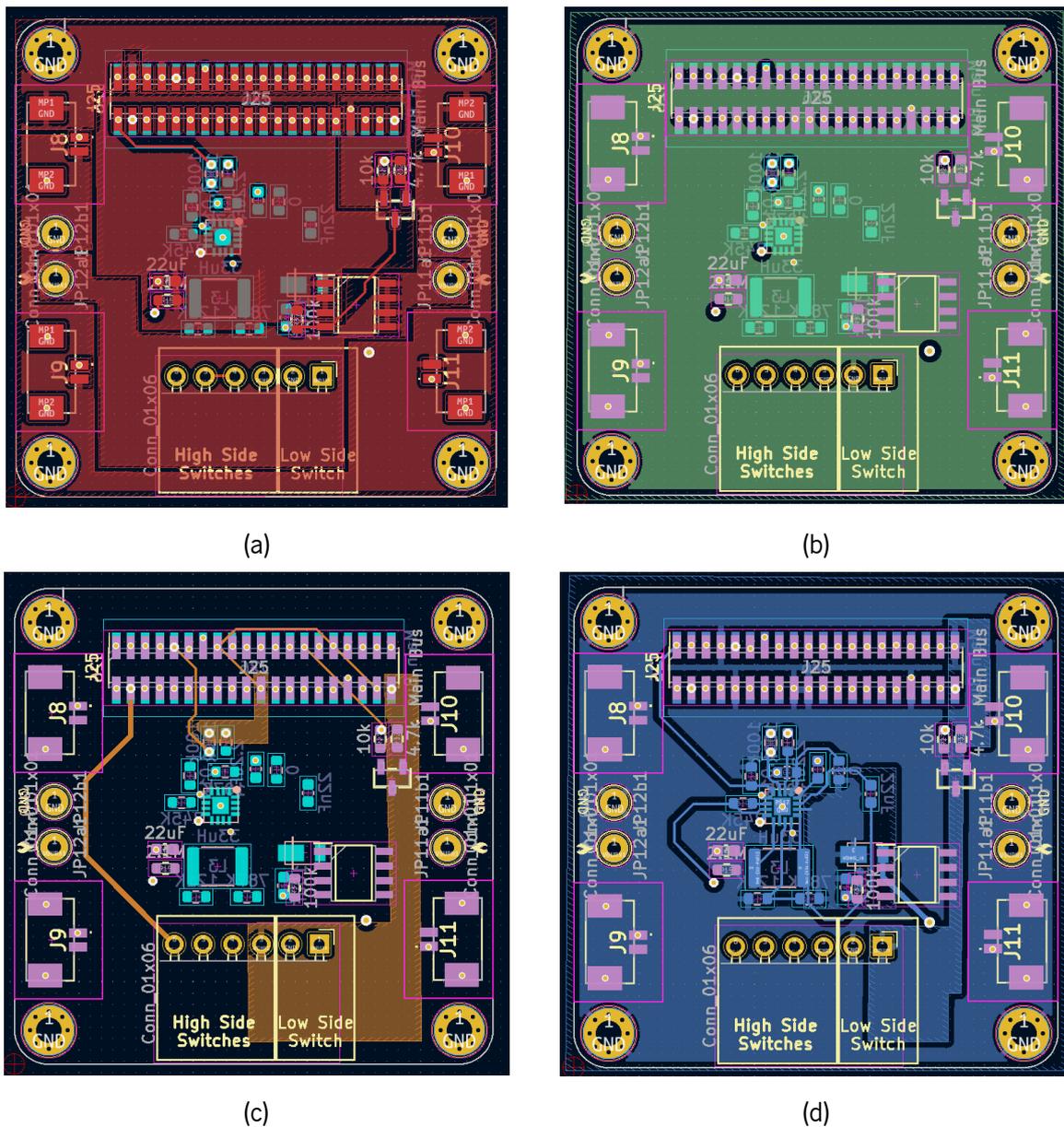


Figura 4.5 – Layout da placa EPS: (a) Camada F.Cu pcb EPS; (b) Camada In1.Cu pcb EPS; (c) Camada In2.Cu pcb EPS; (d) Camada B.Cu pcb EPS.

Na Figura 4.5.a é possível observar a camada F.Cu da placa EPS. As ligações foram estabelecidas utilizando preferencialmente zonas de cobre. Esta camada dedicou-se mais às ligações dos conectores dos painéis solares, dos *burn wires* e da bateria. Os conectores para os painéis solares são conectores Molex de 1,25 mm e ligar-se-iam a outros conectores presentes nas placas dos painéis solares localizadas na face do *PocketQube*.

A camada In1.Cu (Figura 4.5.b) é composta apenas por uma zona de plano *Ground*, com o objetivo de ajudar a reduzir interferências eletromagnéticas.

Na Figura 4.5.c, é possível observar a camada In2.Cu da placa EPS, onde se estabeleceu a ligação de alguns sinais ao conector PQ40.

Por último, na Figura 4.5.d, surgem não só as ligações dos componentes colocados na traseira da placa, mas também zonas de *Ground* e *GNDREF*.

Na Figura 4.6 são apresentadas as duas faces de uma visão a 3 dimensões da placa EPS. Foram também estabelecidas algumas conexões para apresentar a localização dos principais componentes que constituem esta placa.

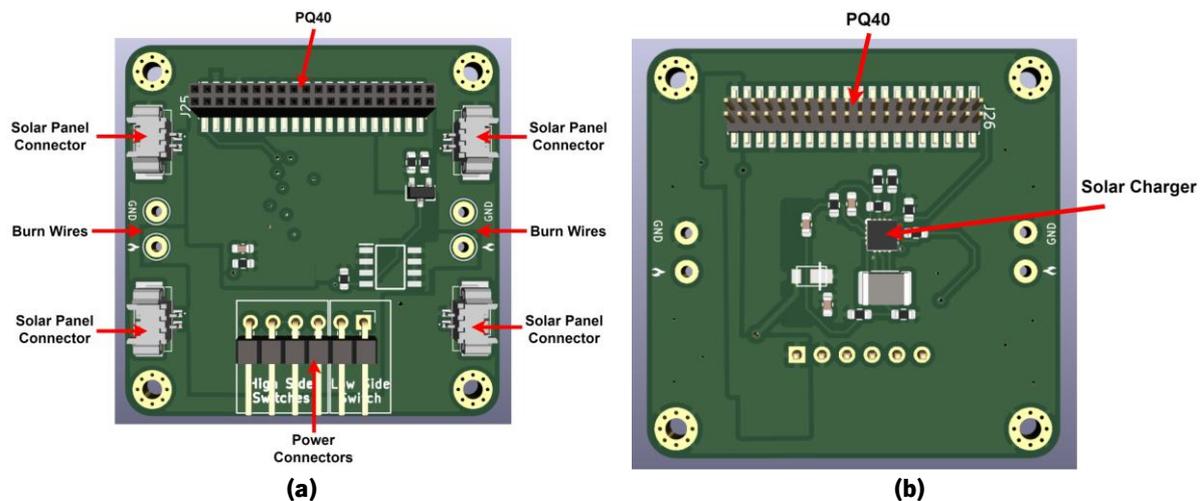


Figura 4.6 - Visão 3D da (a) parte frontal da placa EPS e (b) parte traseira da placa EPS.

4.1.3 COMMS

No desenvolvimento da placa COMMS, conseguiu-se colocar e estabelecer as ligações dos componentes em apenas 2 camadas. Os componentes colocados na parte frontal foram o rádio, o circuito de alimentação do GPS, o *RF Splitter* e a antena UFL do para o rádio. Já na parte traseira, colocou-se o módulo GPS e os circuitos de proteção do barramento SPI.

4.1.4 Battery Holder

A placa *Battery Holder* foi idealizada para suportar duas baterias, conter o circuito de proteção e estabelecer as ligações ao conector PQ40. Devido ao pequeno número de componentes, conseguiu-se desenvolver este módulo apenas com 2 camadas (F.Cu e B.Cu). Na Figura 4.9 é possível observar o *layout* da placa *Battery Holder*.

Na parte frontal da placa, surge um espaço para suportar a célula 2 da bateria, visto na secção 3.4.4 que esta deve ser a primeira a ser ligada. Já no outro lado, existe o espaço para suportar a célula 1. A ligação de ambas as células será efetuada através da solda dos seus fios às *pads* "+" e "-" correspondentes, polo positivo e negativo, respetivamente.

Esta placa possui também pontos de testagem (BM, B-, PACK-, PACK+) utilizados para medir tensões das células da bateria.

A camada F.Cu possui uma zona de 3,3 V e as ligações da célula 2 ao circuito de proteção, como se pode verificar na Figura 4.9.a.

A camada B.Cu, Figura 4.9.b, possui uma zona de *Ground* e as ligações da célula 1 ao circuito de proteção. Depois é efetuada a ligação entre o circuito de proteção e o conector PQ40 de modo a fornecer energia a todo o sistema.

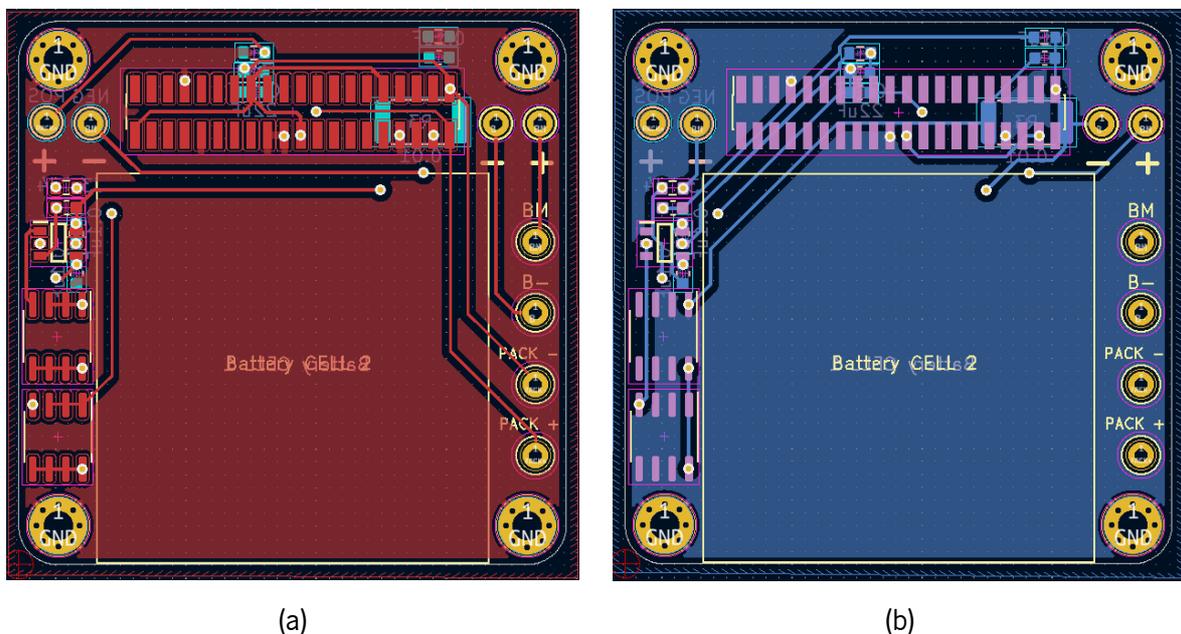


Figura 4.9 – Layout da placa Battery Holder: (a) Camada F.Cu pcb Battery Holder; (b) Camada B.Cu pcb Battery Holder.

No entanto, como referido na secção 2.4.4, sempre que se adicionar um *payload*, tem de se projetar o sistema de proteção para o barramento de comunicação que este utilizar. Por isso, o utilizador final terá de adicionar na *template Payloads* o sistema de proteção correspondente ao seu barramento de dados, do género dos sistemas de proteção presentes na placa OBC.

Na Figura 4.12 são apresentadas as duas faces de uma visão a 3 dimensões da placa *Payloads*. É possível observar que cada pino foi legendado para fácil utilização do utilizador final.

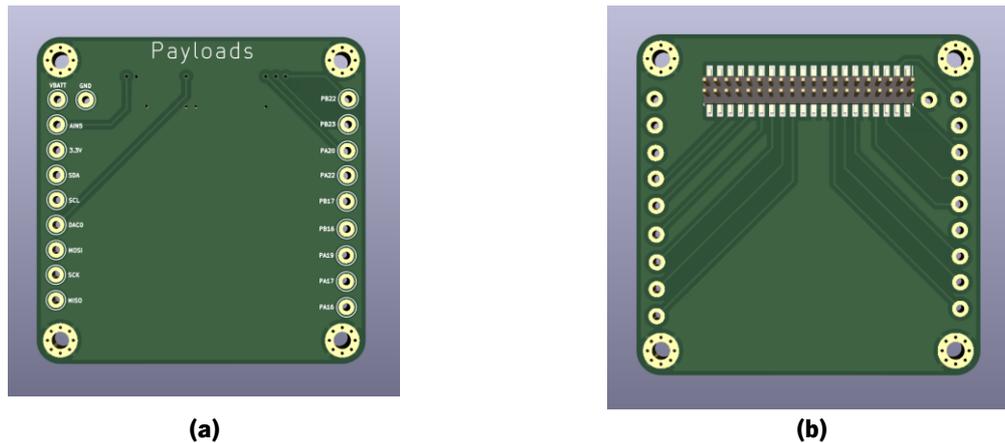


Figura 4.12 - Visão 3D da (a) parte frontal e (b) parte traseira da placa *Payloads*.

4.2 Simulações de circuitos

Neste capítulo será apresentada a simulação do circuito regulador de tensão DC-DC presente na placa OBC. Apenas foi possível simular este circuito pois não foram encontrados modelos SPICE dos outros componentes constituintes dos restantes circuitos de potência.

O regulador 3,3 V foi projetado para a partir de uma tensão de entrada DC entre 4,5 V e 18 V (VBATT) ter na saída uma tensão DC de 3,3 V. Foi referido no secção 3.4.1, que se manteve os valores dos componentes utilizados no sistema PyCubed, como é possível observar na Figura 4.13.

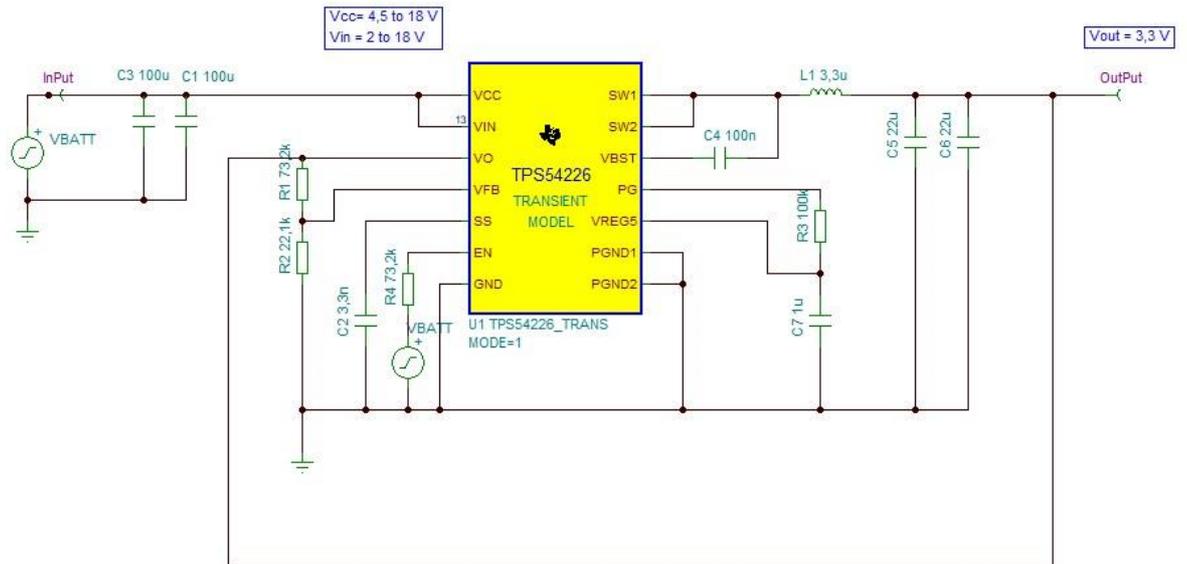


Figura 4.13 - Esquemático Regulador 3,3V para simulação.

Verificou-se que para um VBATT entre 4,5 V e 18 V o circuito gera uma saída cuja tensão são aproximadamente 3,3 V. Na Figura 4.14, simulou-se a entrada do circuito com um valor que corresponde à tensão VBATT quando as baterias estão totalmente carregadas, ou seja, 8,4 V. Tal como esperado, observa-se que a tensão atinge os 3,3 V e permanece constante.

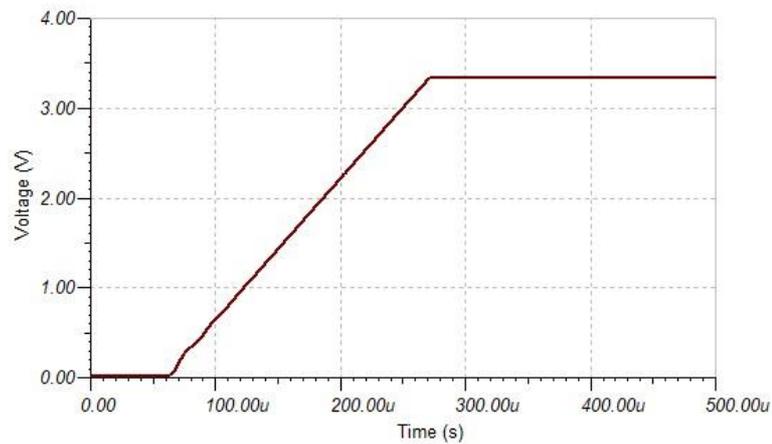


Figura 4.14 - Simulação regulador 3,3V.

5. CONCLUSÃO

O principal objetivo desta dissertação foi avaliar vários projetos de código aberto para pequenos satélites e idealizar os principais subsistemas de uma plataforma pico-satélite, de forma a ser utilizada numa vertente pedagógico-científica. Para esse efeito, projetou-se uma plataforma modular e adaptável a qualquer tipo de *payload*, desde que este fosse conectado ao subsistema *Payload* projetado para esta plataforma.

O trabalho desenvolvido durante a realização desta dissertação permitiu responder às Perguntas de Investigação (PI) delineadas no tópico 1.3.

PI 1: “Quais as atuais tendências dos nano e pico-satélites?”

Esta PI foi respondida no capítulo 2. Atualmente, verifica-se um aumento no desenvolvimento de plataformas de pequenos satélites, sendo que normalmente os nano-satélites destinam-se a missões científicas e os pico-satélites a uma vertente mais educativa, devido ao seu menor tamanho e custo de produção.

PI 2: “Quais os sistemas essenciais num satélite?”

Esta PI foi respondida no capítulo 2 e 3. Os sistemas essenciais para desempenhar com sucesso uma missão num pequeno satélite são o Sistema Eletrónico de Potência, o Sistema de Aviónica ou Computador de Bordo, o Sistema das Comunicações, Sistema de Determinação e Controlo de Atitude e a Estrutura. O EPS, será responsável por fornecer, armazenar, distribuir e controlar a energia elétrica da aeronave. O OBC é responsável por receber, validar, decodificar e distribuir comandos para outros subsistemas da aeronave, e também reunir, processar e armazenar dados de manutenção, bem como da missão para *downlink*. O COMMS é a interface entre o satélite e a *ground station*, permitindo à aeronave fazer o *downlink* dos dados do *payload* e de manutenção para o centro de operação, transmitir

comandos do operador para a aeronave (*uplink*), assim como estabelecer comunicações intersatélite, ou seja, entre satélites. O Sistema de Orientação, Navegação e Controlo é considerado como uma combinação do ODCS, na qual mede e mantém a posição do centro de massa do satélite em função do tempo, e o ADCS, que mede e mantém a orientação do satélite sobre o seu centro de massa. O ADCS desempenha um importante papel para estabilizar os sistemas do satélite no cumprimento da sua missão, necessitando de quatro elementos principais para desempenhar a sua função, nomeadamente a referência onde se realizam as medições, um sensor, um controlador e um atuador. O Sistema Estrutura suporta mecanicamente todos os subsistemas, desempenhando também a função de escudo térmico e de radiação para componentes sensíveis. Em adição a estes sistemas considerados essenciais, existem outros que podem ser integrados na plataforma de forma a melhorar o seu desempenho, tais como o Sistema de Propulsão e o Controlo Térmico.

PI 3: “Como garantir um desenho modular para um *PocketQube*?”

Esta PI foi respondida no capítulo 3. Quando foram definidos os requisitos e restrições do sistema tendo como base outros projetos já desenvolvidos, concluiu-se que a modularidade era obtida através do empilhamento das várias placas constituintes do sistema. Cada placa possuía então dois conectores, um em cada face, ligando-se às placas adjacentes. Este conector garantia que todas as placas possuísem os mesmos sinais.

PI 4: “Existe algum *standard* para plataformas *PocketQube*?”

Esta PI foi respondida no capítulo 3. Verificou-se que, efetivamente, existe um *standard* para plataformas *PocketQube* denominado PQ60. O *standard* foi desenvolvido prevendo que uma pilha de sistemas seja conectada entre si para formar um *PocketQube*, onde cada sistema tem de cumprir com as restrições de uma estrutura *PocketQube* típica. O PQ60 apresenta um desenho técnico da placa, com as dimensões e localizações dos componentes, e também um *pinout* do conector. No entanto, verificou-se que o documento relativo a esse *standard* não se encontra concluído e por isso alguns projetistas de *PocketQubes* modificam algumas das suas características de acordo com as suas necessidades.

PI 5: “Como garantir adaptabilidade do *payload*?”

No capítulo 3, projetou-se a placa dos *Payloads* para ter acesso a todos os pinos GPIO do microcontrolador, a todos os protocolos de comunicação e também a todos os sinais de potência. Desta forma, qualquer *payload* pode ser integrado desde que respeite as restrições da plataforma e a arquitetura do sistema.

5.1 Trabalho Futuro

Após a conclusão do que esta dissertação de mestrado se propôs no início, novos objetivos de investigação podem ser definidos como trabalhos futuros:

- Escrita de um artigo de revisão de literatura referente ao atual panorama desta tipologia de satélites, tal como analisado no capítulo 2.
- Produção das placas, com o objetivo de validar o desenho projetado.
- Desenvolvimento das placas alusivas aos painéis solares.
- Desenvolvimento da placa base da plataforma pico-satélite.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] “NewSpace - SpaceTec Partners.” <https://www.spacetec.partners/markets/newspace/> (accessed Dec. 28, 2021).
- [2] S. Radu, S. Uludag, S. Speretta, J. Bouwmeester, E. Gill, and N. Foteinakis, “Delfi-PQ: The first pocketcube of Delft University of Technology,” in *Proceedings of 69th International Astronautical Congress*, 2018.
- [3] D. Selva and D. Krejci, “A survey and assessment of the capabilities of Cubesats for Earth observation,” *Acta Astronautica*, vol. 74. pp. 50–68, May 2012. doi: 10.1016/j.actaastro.2011.12.014.
- [4] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, and R. Leonardi, “Towards the thousandth CubeSat: A statistical overview,” *International Journal of Aerospace Engineering*, vol. 2019. Hindawi Limited, 2019. doi: 10.1155/2019/5063145.
- [5] J. Bouwmeester and J. Guo, “Survey of worldwide pico- and nanosatellite missions, distributions and subsystem technology,” *Acta Astronaut*, vol. 67, no. 7–8, pp. 854–862, Oct. 2010, doi: 10.1016/j.actaastro.2010.06.004.
- [6] A. Poghosyan and A. Golkar, “CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions,” *Progress in Aerospace Sciences*, vol. 88. Elsevier Ltd, pp. 59–83, Jan. 01, 2017. doi: 10.1016/j.paerosci.2016.11.002.
- [7] J. Puig-Suari, C. Turner, and R. J. Twiggs, “SSC01-VIIIb-5 CubeSat: The Development and Launch Support Infrastructure for Eighteen Different Satellite Customers on One Launch,” in *15TH Annual/USU Conference on Small Satellites*, 2001.

- [8] "CubeSat - Gunter's Space Page." https://space.skyrocket.de/doc_sat/cubesat.htm (accessed Dec. 15, 2022).
- [9] NASA, "State of the Art report: Small Spacecraft Technology," California, Oct. 2021. [Online]. Available: <http://www.sti.nasa.gov>
- [10] "Nanosatellite & CubeSat Database | Nanosats Database." <https://www.nanosats.eu/database> (accessed Jan. 09, 2023).
- [11] R. A. Deepak and R. J. Twiggs, "Thinking Out of the Box: Space Science Beyond the CubeSat," *Journal Of Small Satellites*, vol. 1, pp. 3–7, 2012.
- [12] X. Chen, W. Yao, and P. Harkness, "Pocketqube deorbit times: Susceptibility to the solar cycle," in *IEEE Aerospace Conference Proceedings*, Jun. 2017. doi: 10.1109/AERO.2017.7943625.
- [13] "PRISMA." <https://www.prisma-statement.org/> (accessed Dec. 12, 2021).
- [14] "Scopus", Accessed: Nov. 18, 2021. [Online]. Available: <https://www.scopus.com>
- [15] J. ; Bouwmeester, E. ; Gill, S. ; Speretta, and S. Uludag, "A New Approach on the Physical Architecture of CubeSats & PocketQubes," in *Proceedings of the 15th Reinventing Space Conference, 2017*.
- [16] S. Speretta *et al.*, "CubeSats to PocketQubes: Opportunities and Challenges," in *International Astronautical Congress, 2016*, vol. 67, pp. 26–30.
- [17] J. Bouwmeester *et al.*, "Utility and constraints of PocketQubes," *CEAS Space Journal*, vol. 12, no. 4, pp. 573–586, Dec. 2020, doi: 10.1007/s12567-020-00300-0.
- [18] "PocketQubes." <https://www.gaussteam.com/services/satellite-design/pocketqubes/> (accessed Dec. 27, 2022).
- [19] S. Radu *et al.*, "PocketQube Standard: issue 1," Jun. 2018.
- [20] R. Darbali-Zamora, N. Cobo-Yepes, E. I. Ortiz-Rivera, E. E. Aponte-Bezarez, and A. A. Rincon-Charris, "Applying HOL/PBL to Prepare Undergraduate Students into Graduate Level Studies in the Field of Aerospace Engineering Using the Puerto Rico CubeSat Project Initiative," in *Proceedings - Frontiers in Education Conference, FIE*, Mar. 2019, vol. 2018-October. doi: 10.1109/FIE.2018.8659049.
- [21] "PocketQube - Wikipedia." <https://en.wikipedia.org/wiki/PocketQube> (accessed Jan. 04, 2022).

- [22] "PocketQube - Gunter's Space Page." https://space.skyrocket.de/doc_sat/pocketqub.htm (accessed Jan. 09, 2023).
- [23] Stephen Clark, "Silo-launched Dnepr rocket delivers 32 satellites to space," *SPACEFLIGHT NOW*, Nov. 21, 2013. [Online]. Available: https://spaceflightnow.com/news/n1311/21dnepr/#.Uo5_tcSrzko
- [24] "UniSat 5", Accessed: Jan. 09, 2023. [Online]. Available: https://space.skyrocket.de/doc_sdat/unisat-5.htm
- [25] Michael Sheetz and Andrew Evers, "Photos show space company Firefly's first rocket explode during launch above California," Sep. 2021, Accessed: Jan. 10, 2023. [Online]. Available: <https://www.cnbc.com/2021/09/03/photos-space-company-firefly-alpha-rocket-explodes-above-california.html>
- [26] Jeff Foust, "Firefly says Alpha launch a success despite payload reentries," Oct. 2022, Accessed: Jan. 10, 2023. [Online]. Available: <https://spacenews.com/firefly-says-alpha-launch-a-success-despite-payload-reentries/>
- [27] FOSSA Systems, "Launch Brokerage Service, FOSSA Systems." <https://fossa.systems/launch-brokerage-service/> (accessed Jan. 10, 2023).
- [28] Alba Orbital, "Albapod - PocketQube Deployer." <http://www.albaorbital.com/deployers> (accessed Jan. 11, 2023).
- [29] FOSSA Systems, "FOSSA Systems History." <https://fossa.systems/our-history/> (accessed Jan. 11, 2023).
- [30] "IARU." <http://www.amsatuk.me.uk/iaru/index.php> (accessed Jan. 13, 2023).
- [31] Dava Newman, *Interactive Aerospace Engineering and Design*, 1st ed. McGraw-Hill, 2002.
- [32] J. Richard. Wertz and W. J. Larson, *Space mission analysis and design*, 3rd ed. Microcosm Press and Kluwer Academic Publishers, 1999.
- [33] D. Li, P. Harkness, and T. Walkinshaw, "Design of a solar panel deployment and tracking system for PocketQube pico-satellite," in *IEEE Aerospace Conference Proceedings*, Jun. 2017. doi: 10.1109/AERO.2017.7943910.

- [34] J. Bouwmeester, S. P. van der Linden, A. Povalac, and E. K. A. Gill, "Towards an innovative electrical interface standard for PocketQubes and CubeSats," *Advances in Space Research*, vol. 62, no. 12, pp. 3423–3437, Dec. 2018, doi: 10.1016/j.asr.2018.03.040.
- [35] D. Baldacchino *et al.*, "Review and feasibility of active attitude control and detumbling for the UoMBSat-1 PocketQube pico-satellite," in *2017 25th Mediterranean Conference on Control and Automation, MED 2017*, Jul. 2017, pp. 1237–1243. doi: 10.1109/MED.2017.7984287.
- [36] M. Z. Hasan *et al.*, "REVIEW ON ATTITUDE ESTIMATION ALGORITHM OF ATTITUDE DETERMINATION SYSTEM," *ARPJ Journal of Engineering and Applied Sciences*, vol. 11, no. 7, 2016.
- [37] S. Maryam Sharun *et al.*, "Adaptive Neuro-Controller for Three-Axes Attitude Control of Innovative Satellite," *International Journal of Research and Reviews in Artificial Intelligence*, vol. 1, no. 1, Mar. 2011.
- [38] R. Kovács and V. Józsa, "Thermal analysis of the SMOG-1 PocketQube satellite," *Appl Therm Eng*, vol. 139, pp. 506–513, Jul. 2018, doi: 10.1016/j.applthermaleng.2018.05.020.
- [39] M. Bonnici, P. Mollicone, M. Fenech, and M. A. Azzopardi, "Analytical and numerical models for thermal related design of a new pico-satellite," *Appl Therm Eng*, vol. 159, Aug. 2019, doi: 10.1016/j.applthermaleng.2019.113908.
- [40] M. A. C. Silva, D. C. Guerrieri, A. Cervone, and E. Gill, "A review of MEMS micropropulsion technologies for CubeSats and PocketQubes," *Acta Astronautica*, vol. 143. Elsevier Ltd, pp. 234–243, Feb. 01, 2018. doi: 10.1016/j.actaastro.2017.11.049.
- [41] V. Pallichadath *et al.*, "In-orbit micro-propulsion demonstrator for PICO-satellite applications," *Acta Astronaut*, vol. 165, pp. 414–423, Dec. 2019, doi: 10.1016/j.actaastro.2019.09.004.
- [42] K. Lemmer, "Propulsion for CubeSats," *Acta Astronautica*, vol. 134. Elsevier Ltd, pp. 231–243, May 01, 2017. doi: 10.1016/j.actaastro.2017.01.048.
- [43] A. Scholz and J. N. Juang, "Toward open source CubeSat design," *Acta Astronaut*, vol. 115, pp. 384–392, Jun. 2015, doi: 10.1016/j.actaastro.2015.06.005.
- [44] Wikipedia, "OSSI-1." https://en.wikipedia.org/wiki/OSSI-1#cite_note-opensat-7 (accessed Jan. 20, 2023).

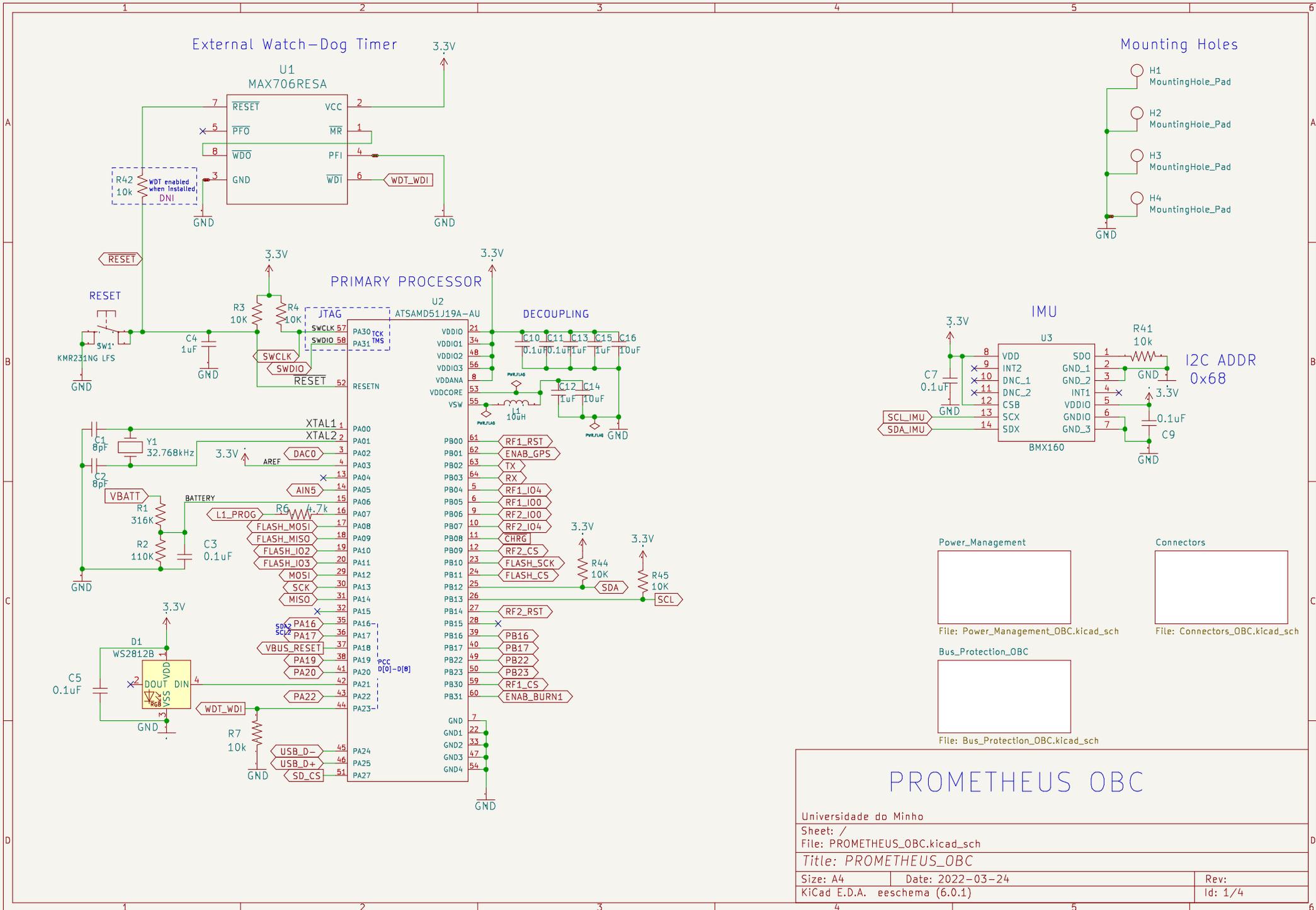
- [45] AMSAT-UK, “OSSI-1.” <https://amsat-uk.org/satellites/non-operational/ossi-1/> (accessed Jan. 20, 2023).
- [46] “LibreCube – open source space and earth exploration.” <https://librecube.org/> (accessed Dec. 27, 2021).
- [47] D. Geeroms, S. Bertho, M. De Roeve, R. Lempens, M. Ordies, and J. Prooth, “ArduSat, an arduino-based cubesat providing students with the opportunity to create their own satellite experiment and collect real-world space data,” in *Proceedings 22nd ESA Symposium on European Rocket and Balloon Programmes and Related Research*, 2015.
- [48] M. Holliday, A. Ramirez, C. Settle, T. Tatum, D. Senesky, and Z. Manchester, “SSC19-WKIII-04 PyCubed: An Open-Source, Radiation-Tested CubeSat Platform Programmable Entirely in Python,” in *33rd Annual AIAA/USU Conference on Small Satellites*, 2019.
- [49] “PyCubed.” <https://pycubed.org/> (accessed Dec. 28, 2021).
- [50] “Consórcio PC104 - Consórcio PC/104 Consórcio PC/104.” <https://pc104.org/> (accessed Jan. 17, 2023).
- [51] A. Thurn *et al.*, “A Nichrome Burn Wire Release Mechanism for CubeSats,” in *41st Aerospace Mechanisms Symposium*, May 2012.
- [52] M. Holliday, Z. Manchester, and D. G. Senesky, “On-Orbit Implementation of Discrete Isolation Schemes for Improved Reliability of Serial Communication Buses,” *IEEE Trans Aerosp Electron Syst*, vol. 58, no. 4, pp. 2973–2982, Aug. 2022, doi: 10.1109/TAES.2022.3142713.
- [53] “GitHub - pycubed/hardware: Pycubed PC104 layout design files.” <https://github.com/pycubed/hardware> (accessed Jan. 18, 2023).
- [54] BHDynamics, “DynOSSAT-EDU.” <https://sites.google.com/bhdyn.com/dynossat/home> (accessed Jan. 19, 2023).
- [55] BH Dynamics, “BH Dynamics GitHub.” <https://github.com/BHDynamics> (accessed Jan. 20, 2023).
- [56] “UPSat.” <https://upsat.gr/> (accessed Jan. 20, 2023).
- [57] “upsat · GitLab.” <https://gitlab.com/librespacefoundation/upsat> (accessed Jan. 20, 2023).

- [58] PicoSat Systems, "OzQube-1 Picosat Systems Small Satellite Solutions Australia's First PocketQube Satellite," 2017. [Online]. Available: <http://picosat.systems>
- [59] I. I. Fernández-Morales, R. Lechuga-Rodríguez, S. Escobedo-Juárez, L. Bañuelos-Delgado, and I. A. Arriaga-Trejo, "Design and development of an educative platform based on the PocketQube 60 Specification," Zacatecas, Mexico, 2019. doi: 10.13140/RG.2.2.20809.49761.
- [60] I. I. Fernández-Morales *et al.*, "Educative Platform based on the PocketQube60 Standard," Zacatecas, Mexico, 2019.
- [61] I. I. Fernández-Morales *et al.*, "Educative Platform based on the PocketQube60 Specification," in *33rd Annual AIAA/USU Conference on Small Satellites*, 2019.
- [62] "PocketQube Kit v1.0 EM – Alba Orbital." <http://www.albaorbital.com/hardware/pocketqube-kit> (accessed Jan. 20, 2023).
- [63] "GitHub - FOSSASystems/FOSSASAT-1." <https://github.com/FOSSASystems/FOSSASAT-1> (accessed Jan. 20, 2023).
- [64] "ESAT - Educational satellite - CubeSatShop.com." <https://www.cubesatshop.com/product/esat/> (accessed Jan. 20, 2023).
- [65] J. Fernando Gómez, "ESAT DATASHEET," Madrid, Spain, Jan. 2021.
- [66] "ISTSat-1 – The first Portuguese CubeSat." <http://istsat-one.tecnico.ulisboa.pt/~istsat-one.daemon/> (accessed Jan. 20, 2023).
- [67] "NessieSAT - Hardware." <https://sites.google.com/bhdyn.com/nessiesat/hardware> (accessed Jan. 20, 2023).
- [68] "What is Kitsat? | Arctic Astronautics / Kitsat." https://kitsat.fi/what_is_kitsat (accessed Jan. 20, 2023).
- [69] "Sanosat-1 - ORION SPACE." <http://orionspace.com.np/SanoSat-1/> (accessed Jan. 20, 2023).
- [70] C. Brunskill, "UBO: A POCKETQUBE KIT FOR EDUCATION ON SATELLITE SYSTEMS AND NEW MISSION CONCEPTS," Harwell, UK. [Online]. Available: www.buildubo.co.uk
- [71] R. Bankston, A. Johnston, V. Smith, E. Char, and B. Paige, "Apogee View; Educational Relations Update; My First Simulated Satellite; Building AMSAT CubeSatSims in the Classroom; 2022 AMSAT Field Day Results," *The AMSAT Journal*, vol. 45, no. 4, Jul. 2022.

- [72] A. Johnston, "The CubeSat Simulator Project Page," *AMSAT Journal*. <https://cubesatsim.org/> (accessed Jan. 21, 2023).
- [73] A. Johnston, J. McLaughlin, D. White, and P. Kilroy, "A New Design for the AMSAT CubeSat Simulator," *The AMSAT Journal*, vol. 43, no. 2, Mar. 2020.
- [74] L. Dudas, L. Szucs, and A. Gschwindt, "The spectrum monitoring system by Smog-1 satellite," in *Proceedings of 14th Conference on Microwave Techniques, COMITE 2015*, Jun. 2015. doi: 10.1109/COMITE.2015.7120316.
- [75] L. Dudas and A. Gschwindt, "The communication and spectrum monitoring system of smog-1 pocketcube class satellite," in *2016 21st International Conference on Microwave, Radar and Wireless Communications, MIKON 2016*, Jun. 2016. doi: 10.1109/MIKON.2016.7491999.
- [76] ECSS, "Space product assurance: Thermal vacuum outgassing test for the screening of space materials," Noordwijk, Netherlands, 2008.
- [77] "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," *IEEE Std 610*, pp. 1–217, 1991, doi: 10.1109/IEEESTD.1991.106963.
- [78] "ISO/IEC 2382-1:1993(en), Information technology – Vocabulary – Part 1: Fundamental terms." <https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:-1:ed-3:v1:en> (accessed Jan. 24, 2023).
- [79] S. Y. Diallo, H. Herencia-Zapana, J. J. Padilla, and A. Tolk, "Understanding interoperability," in *Industry and Academia Symposium 2011*, 2011.
- [80] E. Becnel, Stuart McAndrew, L. Strass, T. Walkinshaw, and K. Worrall, "PQ 60 Standard Document," 1.1, Mar. 03, 2015
- [81] Maxim Integrated Products, "+3V Voltage Monitoring, Low-Cost μ P Supervisory Circuits," MAX706 Datasheet, Rev 5, 2006
- [82] Linear Technology, "LTC4121/LTC4121-4.2 - 40V 400mA Synchronous Step-Down Battery Charger." LTC4121/LTC4121-4.2 Datasheet, 2015.
- [83] RICOH, "Li-ION/POLYMER 2-CELL PROTECTOR OUTLINE." R5460x2xx SERIES Datasheet, 2020.

ANEXO A – ESQUEMÁTICO OBC

Este anexo apresenta o esquemático utilizado no projeto da placa OBC, abordados na secção 3.4.1. É de salientar que o esquemático está dividido em 4 páginas, sendo elas apresentadas nas páginas a seguir pela ordem: Esquemático geral OBC (1/4), *OBC Connectors* (2/4), OBC Power Management (3/4) e OBC Bus Protection (4/4).



Power_Management



File: Power_Management_OBC.kicad_sch

Connectors



File: Connectors_OBC.kicad_sch

Bus_Protection_OBC

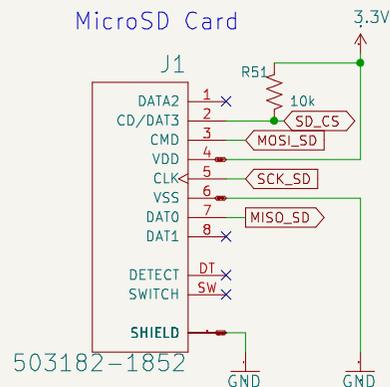
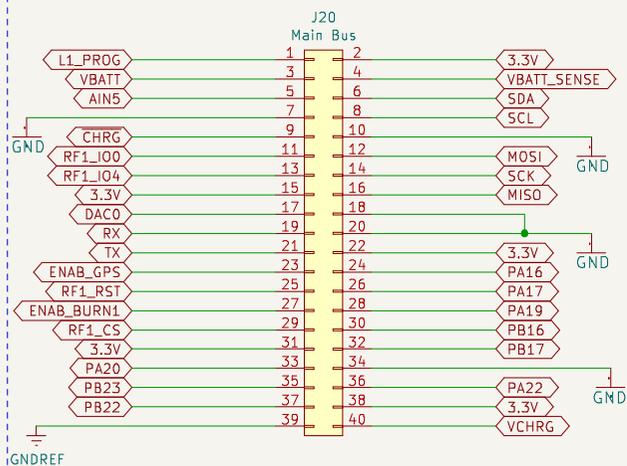
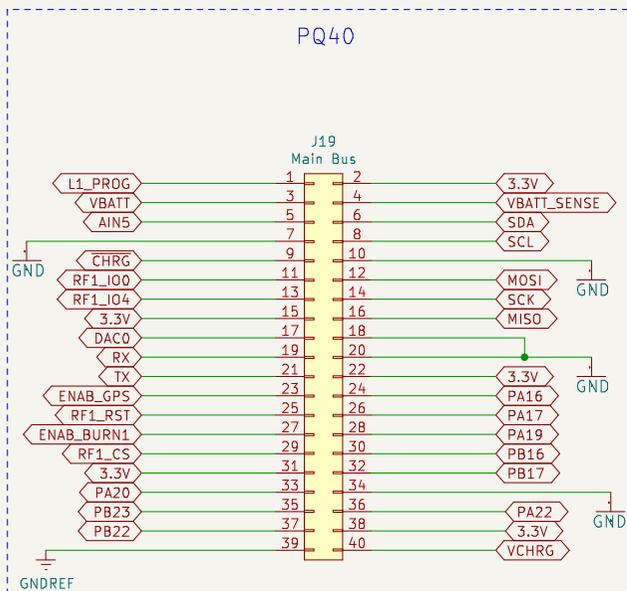
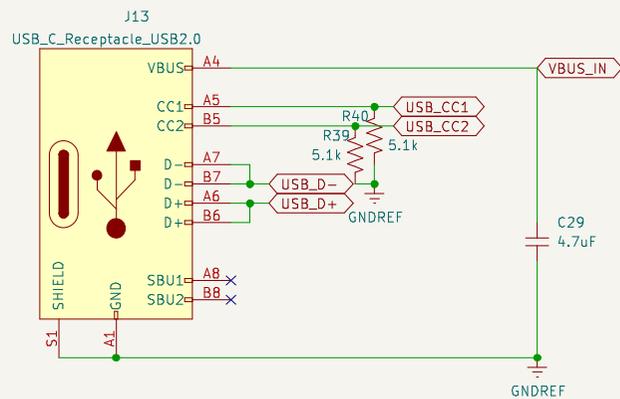


File: Bus_Protection_OBC.kicad_sch

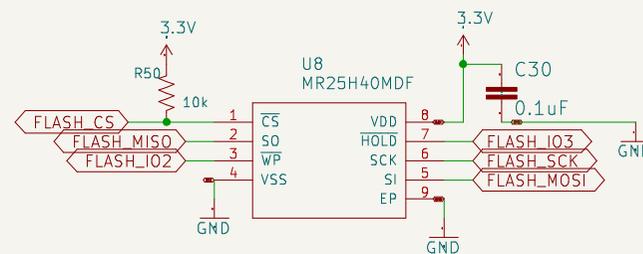
PROMETHEUS OBC

Universidade do Minho	
Sheet: /	
File: PROMETHEUS_OBC.kicad_sch	
Title: PROMETHEUS_OBC	
Size: A4	Date: 2022-03-24
KiCad E.D.A. eeschema (6.0.1)	Rev: Id: 1/4

Power Connectors: USB-C Power Delivery to 2S Li-ion Battery



MRAM – Nonvolatile Memory (4MB storage)



PROMETHEUS OBC_Connectors

Universidade do Minho

Sheet: /Connectors/

File: Connectors_OBC.kicad_sch

Title: PROMETHEUS_OBC

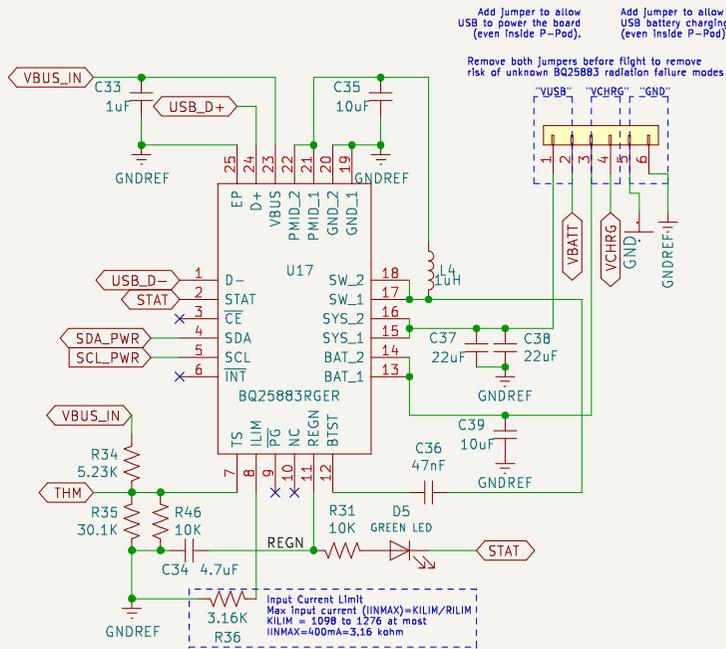
Size: A4 Date: 2022-03-24

KiCad E.D.A. eeschema (6.0.1)

Rev:

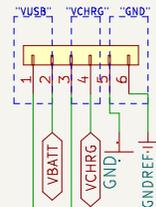
Id: 2/4

USB (Boost) Charging for 2-cell Li-Ion

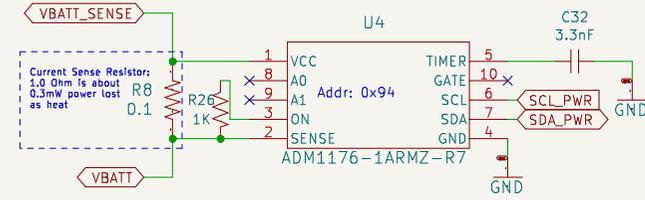


RBF Jumpers

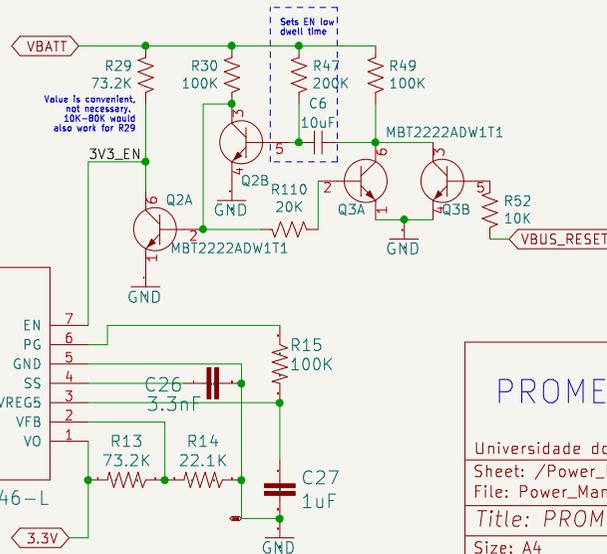
Add Jumper to allow USB to power the board (even inside P-Pod).
Add Jumper to allow USB battery charging (even inside P-Pod).
Remove both jumpers before flight to remove risk of unknown BQ25883 radiation failure modes



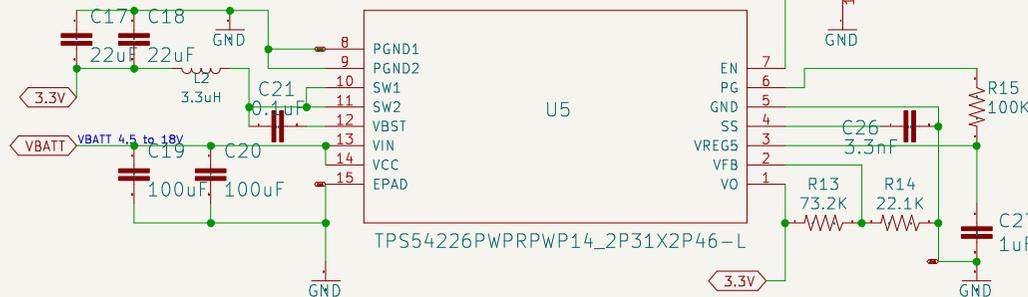
Battery Power Monitor



"One Shot" Regulator Reset



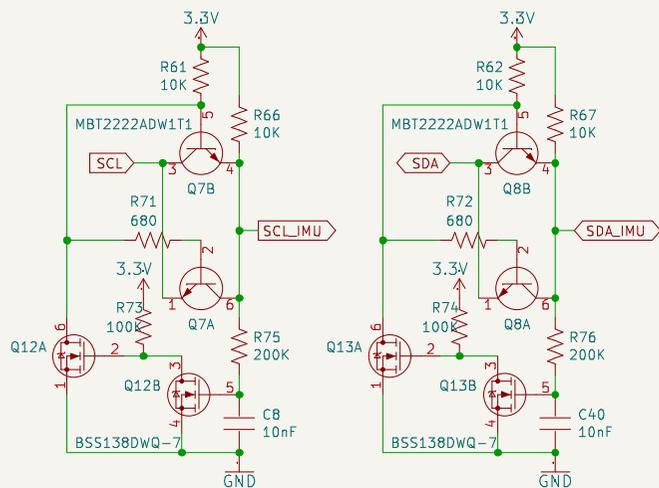
Regulator - 3.3V OUT



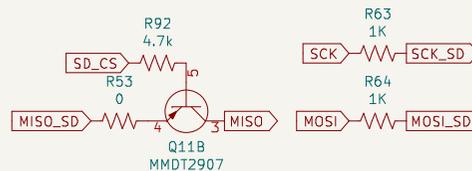
PROMETHEUS OBC_PowerManagement

Universidade do Minho
Sheet: /Power_Management/
File: Power_Management_OBC.kicad_sch
Title: PROMETHEUS_OBC
Size: A4 Date: 2022-03-24
KiCad E.D.A. eeschema (6.0.1) Rev: Id: 3/4

I2C Bus Protection – IMU



SPI Bus Protection – SD Card



NOTE

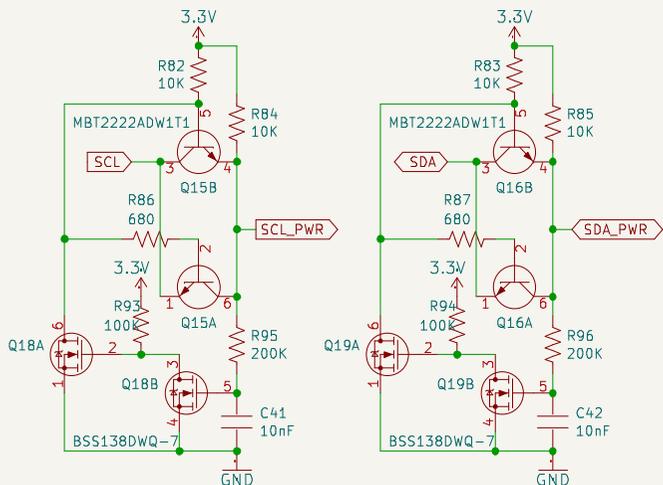
These novel bus protection circuits prevent traditional I2C/SPI failure modes where a single slave failure can disable the entire bus.

Learn more:
<https://doi.org/10.36227/techrxiv.15166620>

By default, slave clock and/or data lines can be held low and the Master (SAMD51) will still be able to communicate with the remainder of the bus.

They can individually be bypassed by removing the transistor(s) and soldering the 0ohm the jumpers below.

I2C Bus Protection – Power Monitor & USB Charger



PROMETHEUS OBC_BusProtection

Universidade do Minho
 Sheet: /Bus_Protection_OBC/
 File: Bus_Protection_OBC.kicad_sch

Title: PROMETHEUS_OBC

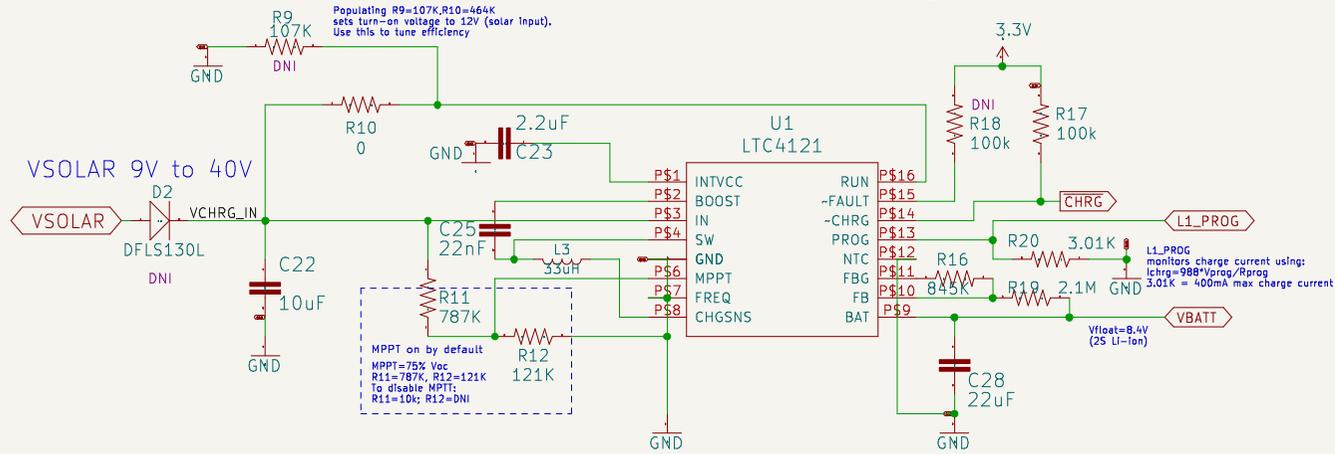
Size: A4 Date: 2022-03-24
 KiCad E.D.A. eeschema (6.0.1)

Rev:
 Id: 4/4

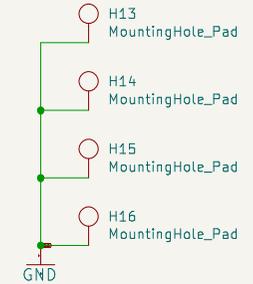
ANEXO B – ESQUEMÁTICO EPS

Este anexo apresenta o esquemático utilizado no projeto da placa EPS, abordados na secção 3.4.2. É de salientar que o esquemático está dividido em 3 páginas, sendo elas apresentadas nas páginas a seguir pela ordem: Esquemático geral EPS (1/3), *EPS Burn Wires* (2/3) e *EPS Connectors* (3/3).

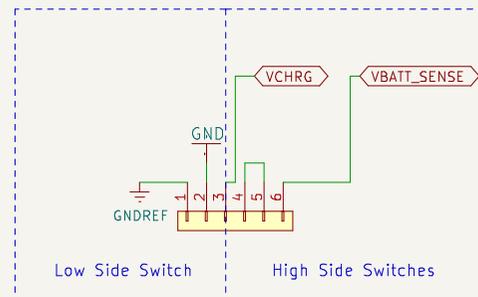
2S Li-Ion Battery Solar Charging Circuit



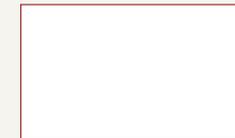
Mounting Holes



Power Connectors: Power Delivery to 2S Li-Ion Battery

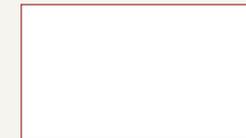


Connectors



File: Connectors.kicad_sch

Burn_Wires



File: Burn_Wires.kicad_sch

PROMETHEUS EPS

Universidade do Minho

Sheet: /

File: PROMETHEUS_EPS.kicad_sch

Title: PROMETHEUS_EPS

Size: A4

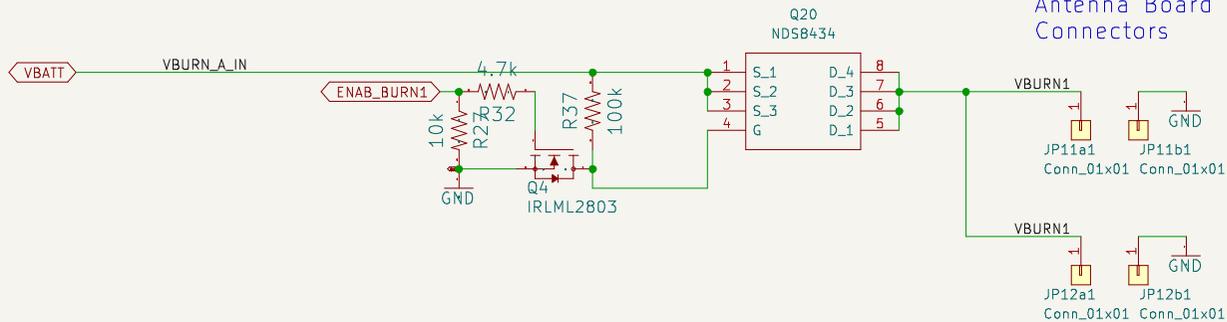
Date:

KiCad E.D.A. eeschema (6.0.1)

Rev:

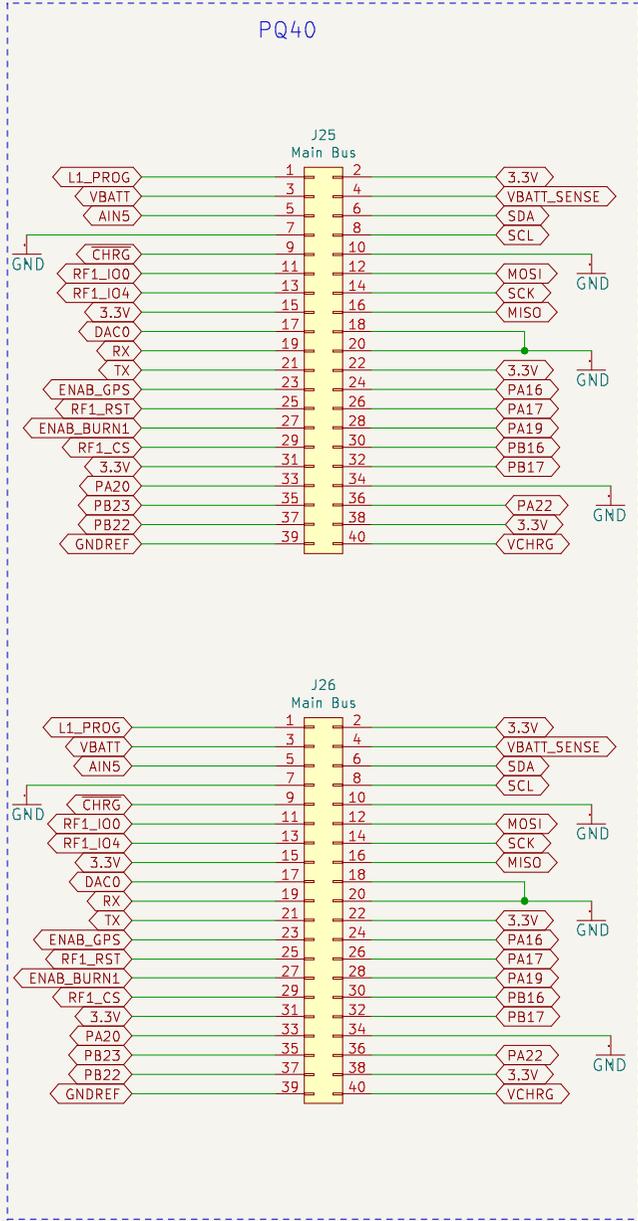
Id: 1/3

Burn Wire Control (Antenna Deployment)

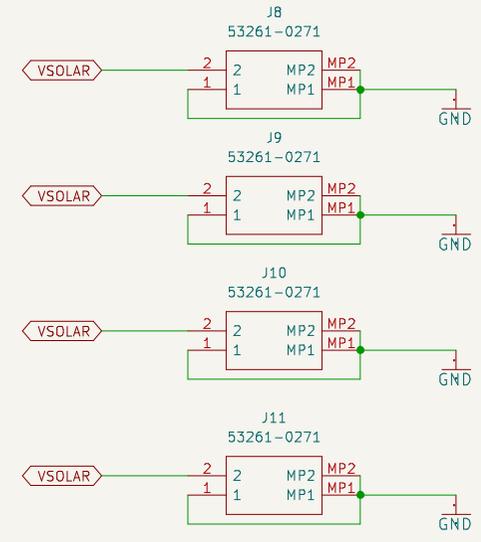


PROMETHEUS EPS Burn Wires

Sheet: /Burn_Wires/		
File: Burn_Wires.kicad_sch		
Title:		
Size: A4	Date:	Rev:
KiCad E.D.A. eeschema (6.0.1)		Id: 2/3



Solar Panel Connectors



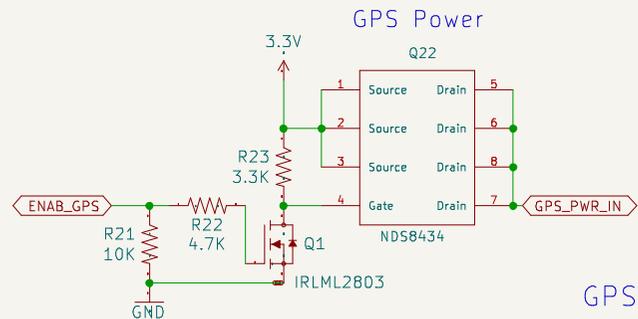
PROMETHEUS EPS Connectors

Sheet: /Connectors/		Date:	
File: Connectors.kicad_sch		Rev:	
Title:		Id: 3/3	
Size: A4	KiCad E.D.A. eeschema (6.0.1)		

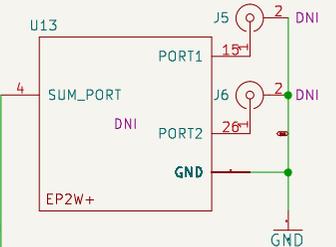
ANEXO C – ESQUEMÁTICO COMMS

Este anexo apresenta o esquemático utilizado no projeto da placa COMMS, abordados na secção 3.4.3. É de salientar que o esquemático está dividido em 3 páginas, sendo elas apresentadas nas páginas a seguir pela ordem: Esquemático geral COMMS (1/3), *COMMS Bus Protection* (2/3) e *COMMS Connectors* (3/3).

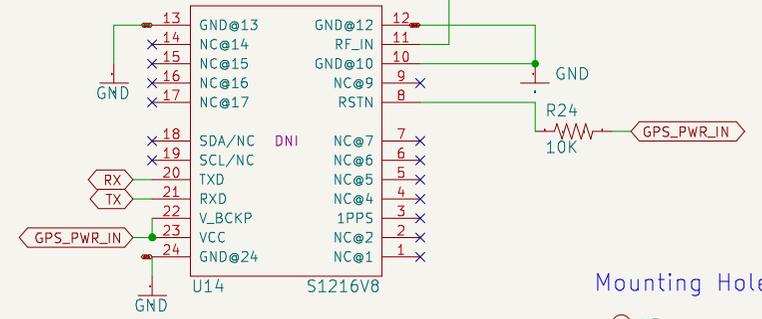
Modular Radio (HopeRF format)



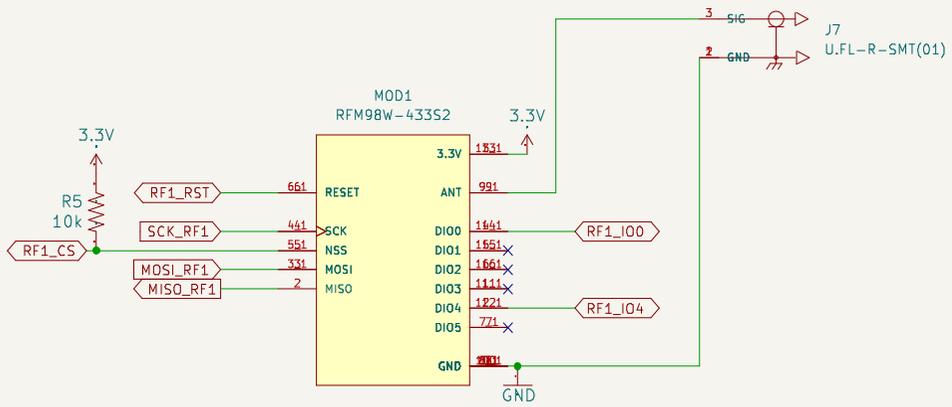
RF Splitter (2 Way, 0deg DC-Pass)



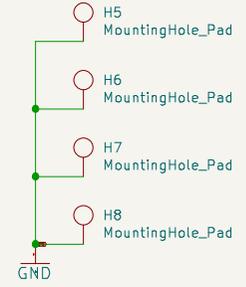
GPS Module



Radio #1



Mounting Holes



Bus_Protection_COMMS



File: Bus_Protection_COMMS.kicad_sch

Connectors

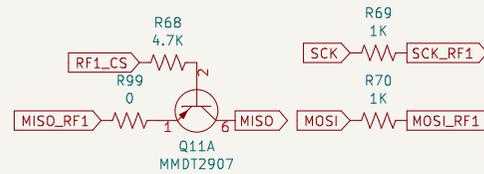


File: Connectors.kicad_sch

PROMETHEUS COMMS

Universidade do Minho	
Sheet: /	
File: PROMETHEUS_COMMS.kicad_sch	
Title: PROMETHEUS_COMMS	
Size: A4	Date:
KiCad E.D.A. eeschema (6.0.1)	Rev: Id: 1/3

SPI Bus Protection – Radio 1



NOTE

These novel bus protection circuits prevent traditional I2C/SPI failure modes where a single slave failure can disable the entire bus.

Learn more:
<https://doi.org/10.36227/techrxiv.15166620>

By default, slave clock and/or data lines can be held low and the Master (SAMD51) will still be able to communicate with the remainder of the bus.

They can individually be bypassed by removing the transistor(s) and soldering the 0ohm the jumpers below.

PROMETHEUS COMMS Bus Protection

Sheet: /Bus_Protection_COMMS/
File: Bus_Protection_COMMS.kicad_sch

Title:

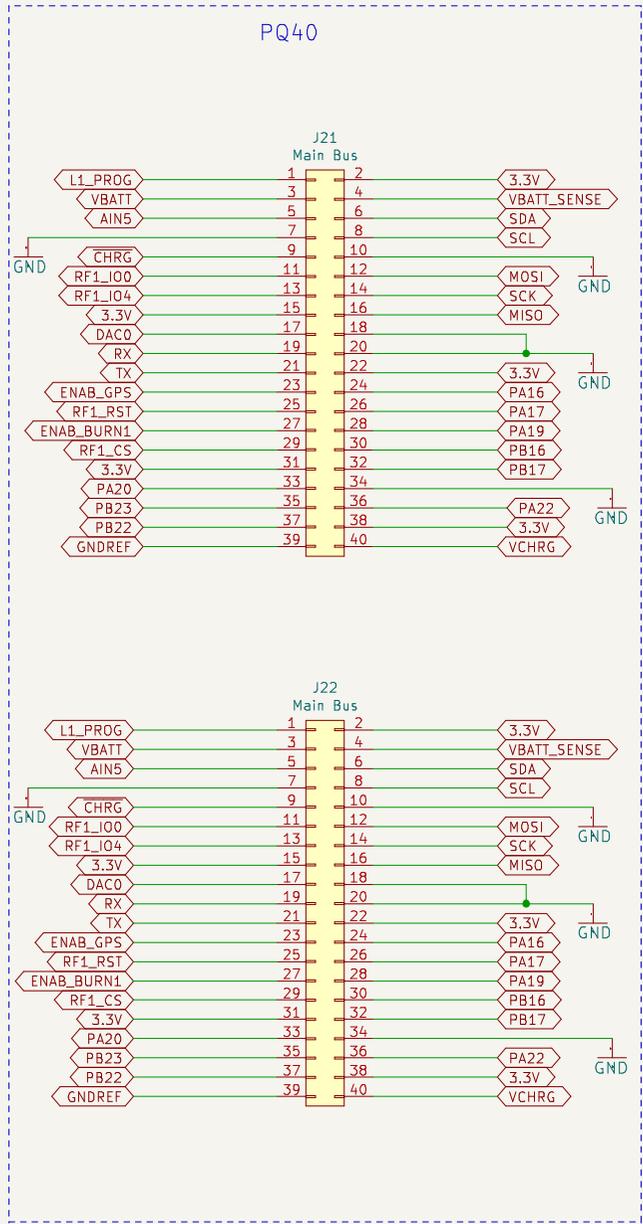
Size: A4

Date:

Rev:

KiCad E.D.A. eeschema (6.0.1)

Id: 2/3



PROMETHEUS COMMS Connector

Sheet: /Connectors/
File: Connectors.kicad_sch

Title:

Size: A4

Date:

KiCad E.D.A. eeschema (6.0.1)

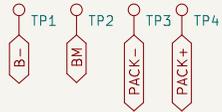
Rev:

Id: 3/3

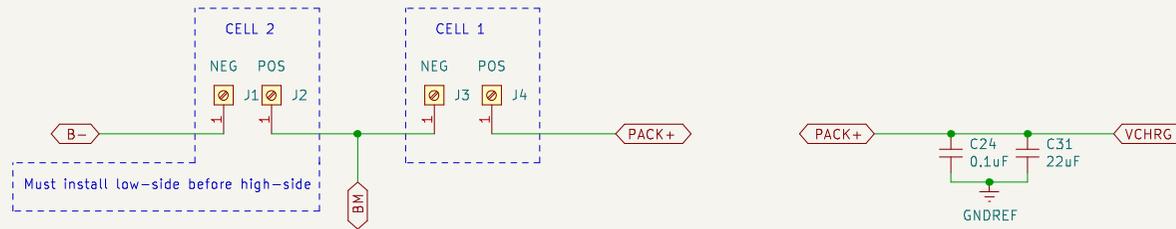
ANEXO D – ESQUEMÁTICO BATTERY HOLDER

Este anexo apresenta o esquemático utilizado no projeto da placa *Battery Holder*, abordados na secção 3.4.4. É de salientar que o esquemático está dividido em 2 páginas, sendo elas apresentadas nas páginas a seguir pela ordem: Esquemático geral *Battery Holder* (1/2) e *Battery Holder Connectors* (2/2).

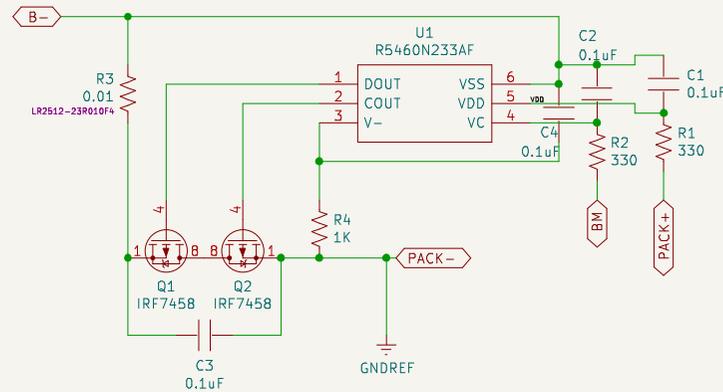
Test Points



2 li-po batteries in series



BATTERY PROTECTION IC



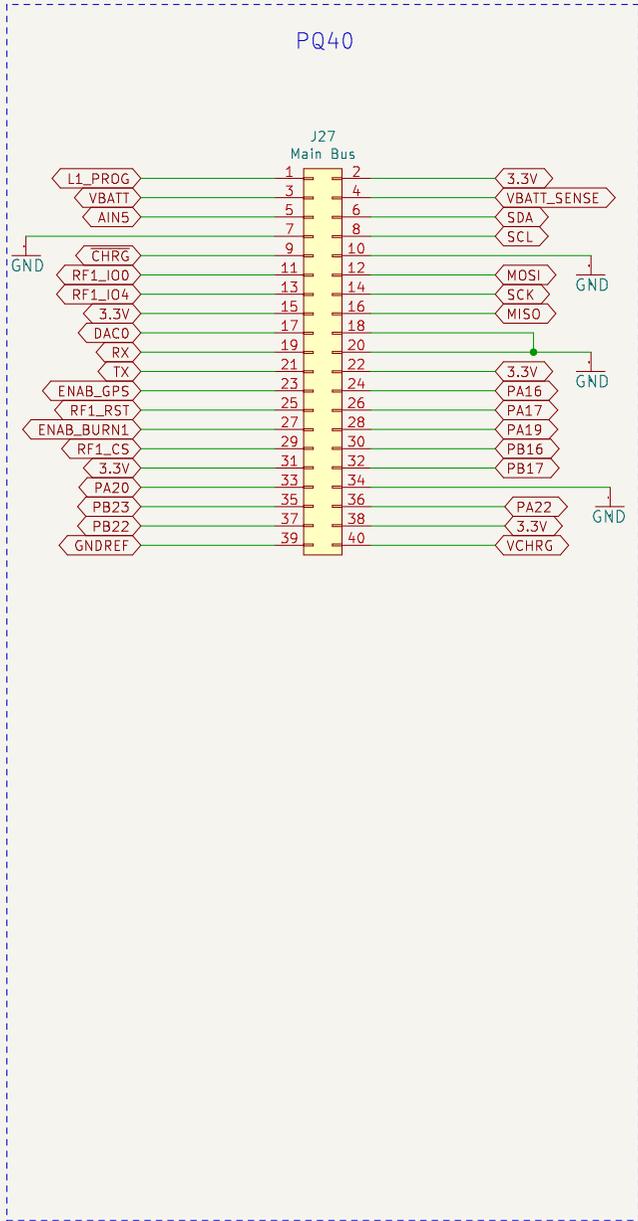
Mounting Holes



File: Connector_battery.kicad_sch

PROMETHEUS Battery

Sheet: /	
File: PROMETHEUS_BATTERY.kicad_sch	
Title:	
Size: A4	Date:
KiCad E.D.A. eeschema (6.0.1)	Rev: Id: 1/2



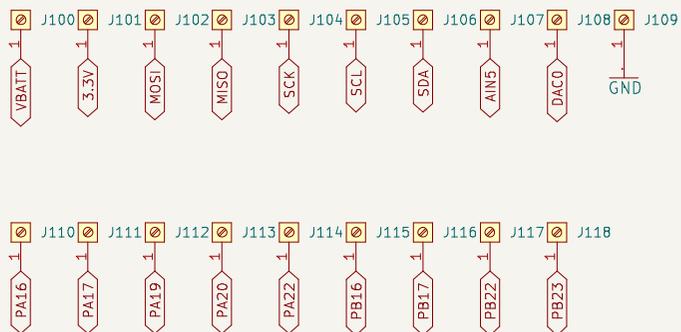
PROMETHEUS Battery Holder Connectors

Sheet: /Connector_battery/		
File: Connector_battery.kicad_sch		
Title:		
Size: A4	Date:	Rev:
KiCad E.D.A. eeschema (6.0.1)		Id: 2/2

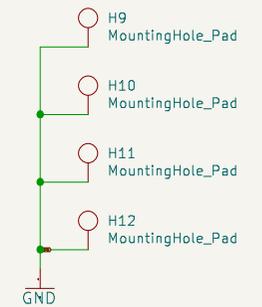
ANEXO E – ESQUEMÁTICO PAYLOADS

Este anexo apresenta o esquemático utilizado no projeto da placa *Payloads*, abordados na secção 3.4.5. É de salientar que o esquemático está dividido em 2 páginas, sendo elas apresentadas nas páginas a seguir pela ordem: Esquemático geral *Payloads* (1/2) e *Payloads Connectors* (2/2).

Payloads PinOut



Mounting Holes



Connectors_Payloads



File: Connectors_Payloads.kicad_sch

PROMETHEUS PAYLOADS

Universidade do Minho

Sheet: /

File: PROMETHEUS_PAYLOADS.kicad_sch

Title: PROMETHEUS_PAYLOADS

Size: A4

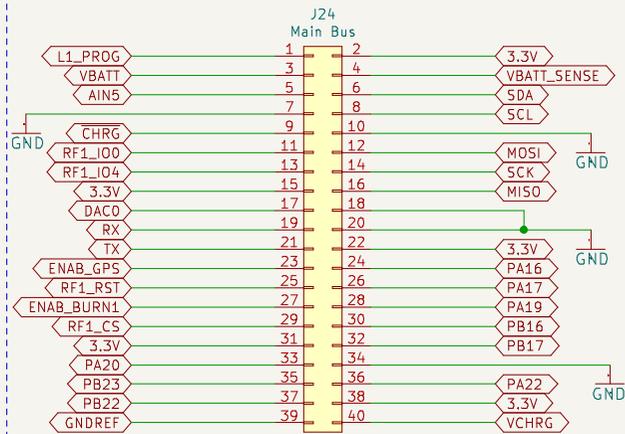
Date:

Rev:

KiCad E.D.A. eeschema (6.0.1)

Id: 1/2

PQ40



PROMETHEUS PAYLOADS Connectors

Sheet: /Connectors_Payloads/
File: Connectors_Payloads.kicad_sch

Title:

Size: A4
KiCad E.D.A. eeschema (6.0.1)

Date:
Rev:
Id: 2/2