

# LABORATÓRIO DE SISTEMAS AUTÓNOMOS



## Relatório Final

Collision Sensor Array for Drones:  
ToF and Solid State LiDAR

Carlos Jorge Vieira da Silva 1160628

Mestrado de Sistemas Autónomos  
LaSET - Laboratório de Sistemas Embebidos e Tecnologias

14 de julho de 2021



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento e Motivação . . . . .	1
1.2	Cenários de aplicação . . . . .	2
1.3	Objectivos . . . . .	2
1.4	Estrutura do Relatório . . . . .	3
<b>2</b>	<b>Estado da Arte</b>	<b>5</b>
2.1	UAVs . . . . .	5
2.1.1	Introdução . . . . .	5
2.1.2	Tecnologias Anti-Colisão para UAVs . . . . .	8
2.2	LiDAR . . . . .	13
2.2.1	Introdução . . . . .	13
2.2.2	Princípio Time-of-Flight . . . . .	18
2.2.3	Tecnologia VCSEL . . . . .	20
2.2.4	Flash LiDAR . . . . .	21
2.2.5	Tecnologias Rotacionais . . . . .	22
2.2.6	Sistema Solid-State . . . . .	23
<b>3</b>	<b>Análise de Requisitos</b>	<b>29</b>
<b>4</b>	<b>Arquitectura de Alto Nível</b>	<b>33</b>

<b>5 Projecto</b>	<b>37</b>
5.1 Descrição do <i>Hardware</i> . . . . .	37
5.1.1 Sensores LiDAR . . . . .	37
5.1.2 Servo Motores Dynamixel . . . . .	42
5.1.3 Microcontroladores STM32 . . . . .	44
5.1.4 Esquema Elétrico . . . . .	46
5.1.5 Placa de Circuito Impresso . . . . .	55
5.2 Análise Financeira . . . . .	56
5.3 Descrição do <i>Software</i> . . . . .	60
<b>6 Implementação</b>	<b>65</b>
6.1 Desenvolvimento do Hardware . . . . .	65
6.1.1 Placa de Circuito Impresso . . . . .	65
6.1.2 Estrutura Mecânica . . . . .	68
6.1.3 Protótipo Final . . . . .	69
6.2 Desenvolvimento do Software . . . . .	71
6.2.1 Programação do Microcontrolador . . . . .	71
6.2.2 Integração com o ROS . . . . .	73
<b>7 Resultados</b>	<b>75</b>
7.1 Estudo da Linearidade do Sensor LiDAR . . . . .	75
7.2 Estudo Rotativo do Sensor LiDAR . . . . .	76
7.3 Scan e Visualização em ROS . . . . .	80
<b>8 Conclusão e Trabalho Futuro</b>	<b>81</b>
<b>A Esquema Eléctrico Total do Sistema</b>	<b>93</b>

# **Lista de Figuras**

2.1	Tipos de Estruturas para UAVs . . . . .	7
2.2	Sistema Anti-Colisão de 360° . . . . .	8
2.3	Sistema Anti-Colisão de 360° com Altímetro . . . . .	9
2.4	Sensor SONAR HC-SR04 . . . . .	11
2.5	Caméra Estereoscópica Intel Real Sense . . . . .	12
2.6	Scanning LIDAR . . . . .	14
2.7	Detector Arrays . . . . .	14
2.8	Detector Arrays Modulation . . . . .	14
2.9	Situações de Reflectividade da Luz . . . . .	16
2.10	As diferentes Tecnologias de Medição de Distância . . . . .	16
2.11	Resposta Espectral Genérica de Sensores LiDAR. . . . .	17
2.12	Princípio <i>Time-of-Flight</i> . . . . .	18
2.13	Estrutura de um VCSEL . . . . .	20
2.14	Topologias de Flash LiDAR . . . . .	21
2.15	Tecnologias LiDAR Rotacionais . . . . .	22
2.16	Esfera Delimitadora e Cone de Colisão . . . . .	24
2.17	Modos Operacionais de Matrizes Multi-Dimensionais . . . . .	25
2.18	TeraRanger Evo Mini Array Kit . . . . .	26
2.19	TeraRanger Multiflex . . . . .	27
2.20	TeraRanger One Kit . . . . .	27
2.21	TeraRanger Tower Evo . . . . .	27

4.1	Arquitetura do Sistema . . . . .	33
4.2	Esquema Geral do Hardware . . . . .	34
4.3	Arquitectura dos Módulos Genéricos . . . . .	36
5.1	Sensores FlightSense™ . . . . .	39
5.2	Sensores LiDAR . . . . .	40
5.3	Servo Motores . . . . .	43
5.4	Microcontrolador STM32F103RB . . . . .	50
5.5	Círculo de Alimentação . . . . .	50
5.6	Círculo Oscilador, Boot, SWD e Reset . . . . .	52
5.7	Controlo do Motor, Leitura dos Sensores e Comunicação CAN	54
5.8	Design Final da PCB . . . . .	55
5.9	Comparação de Custos Relacionados com a PCB . . . . .	58
5.10	Comparação de Custos Totais . . . . .	59
5.11	Arquitectura de Software dos Módulos . . . . .	61
6.1	Placa de Circuito Impresso Produzida . . . . .	66
6.2	Layers da Placa de Circuito Impresso . . . . .	67
6.3	Placa de Circuito Impresso noutras Cores . . . . .	67
6.4	Construções Mecânicas do Protótipo . . . . .	68
6.5	Ferramentas de Hardware Utilizadas . . . . .	69
6.6	Protótipo Final . . . . .	70
6.7	Análise de Baixo-Nível dos Dados do TFmini-Plus . . . . .	71
6.8	Software Dynamixel Wizard 2.0 . . . . .	72
6.9	Software STM32 CubeIDE . . . . .	72
6.10	Visualização dos Dados do Protótipo via Consola . . . . .	73
7.1	Gráfico do Estudo Linear do sensor TFmini-Plus . . . . .	76
7.2	Ambiente de Testes Rotativos do Sensor LiDAR . . . . .	76
7.3	Frequência de Amostragem do Sistema) . . . . .	77
7.4	Estudo Rotativo do sensor TFmini-Plus (Plot Angular) . . . . .	78

7.5	Estudo Rotativo do Sensor TFmini-Plus (Plot Cartesiano) . . .	79
7.6	Estudo Rotativo do Sensor TFmini-Plus (Plot Angular) . . . .	79
7.7	Visualização do Scan no RViz, em Tempo Real . . . . .	80
A.1	Círculo Total . . . . .	94

Esta página foi intencionalmente deixada em branco.

# **Lista de Tabelas**

2.1	Comparação de Especificações dos Produtos TeraBee . . . . .	28
5.1	Comparação de Sensores de Distância . . . . .	39
5.2	Comparação de Sensores de Distância do tipo LiDAR . . . . .	41
5.3	Comparação de Servo Motores Dynamixel . . . . .	43
5.4	Comparação de Microcontroladores STM32 . . . . .	45
5.5	Comparação Genérica de Sistemas de Conversão de Tensão CC	46
5.6	Consumo de Corrente de Componentes alimentados a 3.3 V .	47
5.7	Consumo de Corrente de Componentes alimentados a 5 V .	47
5.8	Especificações do AMS1117-3.3V . . . . .	48
5.9	Orçamento Simplificado da Fabricação da PCB . . . . .	56
5.10	Orçamento Simplificado dos Componentes Externos à PCB .	56
5.11	Orçamento Extensivo do Custo Total do Sistema . . . . .	57

Esta página foi intencionalmente deixada em branco.

# Capítulo 1

## Introdução

### 1.1 Enquadramento e Motivação

*Drones*, ou UAVs (*Unmanned Aerial Vehicles*) fazem parte de um conjunto de novas tecnologias que estão a alimentar a indústria global. O tipo de atividades específicas para este tipo de veículos têm aumentado exponencialmente em áreas como a agricultura, minas, segurança, construção, energia, produção multimédia e entretenimento devido ao seu nível de performance, custo e segurança de seres humanos.

O tema deste projeto foi proposto pela unidade curricular de LaSET que faz parte do Mestrado em Sistemas Autónomos do ISEP (Instituto Superior de Engenharia do Porto). Esta unidade curricular visa o desenvolvimento de competências de projeto de engenharia, nomeadamente nas fases do ciclo de desenvolvimento, capacidades de projeto e implementação real de sistemas complexos, como a integração deste mesmo sistema nos *drones* presentes no LSA (Laboratório de Sistemas Autónomos) do ISEP, de forma a proporcionar-lhes um sistema de desvio de obstáculos compatível com os sistemas embebidos já presentes nos mesmos. Tudo isto aliado a espírito de aprendizagem, perpetuando-se o lema ”Saber Fazer” do ISEP.

## 1.2 Cenários de aplicação

Existe possibilidade de colisão tanto em voos autónomos como remotamente comandados, que pode causar danos tanto no próprio UAV como em objetos, pessoas e/ou animais. O sistema proposto a ser desenvolvido pode ser aplicado em qualquer cenário de aplicação de atividades com *drones*, pois esse risco está sempre presente. Para além dos danos causados, questões judiciais e legais poderão afetar os donos e/ou pilotos destes mesmos veículos, sendo recomendado o uso deste sistema para prevenção dessas situações embarracosas.

## 1.3 Objectivos

O principal objetivo é o desenvolvimento de um protótipo de um *array* de sensores ToF (*Time-of-Flight*) *Solid State LiDAR*, e a sua integração nos *drones* presentes no LSA (Laboratório de Sistemas Autónomos) do ISEP, de forma a proporcionar-lhes um sistema de desvio de obstáculos compatível com os sistemas embebidos já presentes nos mesmos, como também uma plataforma de software para testes. Ao qual se agrupam os seguintes sub-objetivos:

- Bom custo-benefício,
- Boa eficiência energética e de processamento,
- Mecanicamente simples e leve,
- Comunicação de dados multi-protocolo.

Ao atingir estes objetivos pode-se afirmar que foi criada uma base funcional para o desvio de obstáculos, que poderá continuar a ser desenvolvida futuramente noutros contextos curriculares e/ou profissionais. Provando ser possível obter alternativas às soluções disponíveis no mercado.

## 1.4 Estrutura do Relatório

Este relatório está dividido nos seguintes capítulos:

- **Capítulo 2. Estado de Arte:** Neste capítulo é feita uma breve introdução aos vários tipos de UAVs e às suas estruturas. Posteriormente é realizado um levantamento das várias tecnologias anti-collision já existentes, os sensores utilizados e os processos e algoritmos de deteção. De seguida é feita uma abordagem mais extensiva acerca da tecnologia LiDAR, que se revelou ser extremamente interessante para esta aplicação. O capítulo termina com uma análise a algumas soluções disponíveis no mercado.
- **Capítulo 3. Análise de Requisitos:** Aqui, é apresentada, de forma resumida, a listagem dos requisitos que o projeto terá de respeitar. Esta lista teve em conta a opinião dos vários orientadores do projeto, assim como a análise do capítulo anterior e as restrições financeiras.
- **Capítulo 4. Arquitetura de Alto Nível:** Nesta secção são apresentadas várias arquiteturas de alto nível, que ajudaram à construção e organização da implementação do projeto, e resumem a estrutura utilizada.
- **Capítulo 5. Projeto:** A seleção e comparação de componentes é feita neste capítulo. É também desenhado o circuito que interliga todos os componentes, e a sua compactação no *design* de uma PCB. É também feita a análise financeira na aquisição dos componentes propostos e uma previsão das funções de *software* a ser desenvolvidas.
- **Capítulo 6. Implementação:** Aqui é mostrado o desenvolvimento final do *hardware* e *software*, assim como a construção final do protótipo.
- **Capítulo 7. Resultados:** Nesta última secção são demonstrados e avaliados os resultados produzidos pelo sistema.

Esta página foi intencionalmente deixada em branco.

# Capítulo 2

## Estado da Arte

### 2.1 UAVs

#### 2.1.1 Introdução

UAVs (*Unmanned Aerial Vehicles*) também conhecidos como *drones*, são nos dias de hoje uma escolha popular para aplicações autónomas devido ao seu tamanho reduzido, agilidade, capacidade de navegar no ar e pode conseguirem realizar funções que seriam dificilmente executadas pelo ser humano [1]. Algumas dessas aplicações autónomas estão relacionadas com atividades de vigilância, mapeamento de terreno, agricultura, meteorologia e atividades de busca e salvamento, a um custo/benefício bastante acessível em comparação com veículos aéreos tripulados. Um UAV contém, normalmente, uma grande variedade de sensores, entre os quais um IMU (*Inertial Measurement Unit*) e um receptor GPS (*Global Positioning System*), que são utilizados para navegação e posicionamento. As caracterizações possíveis têm normalmente em conta o seu tamanho, estrutura e tipo de utilização. Os grandes avanços nas tecnologias relacionadas com estes veículos autónomos aéreos permitem atualmente um elevado nível de estabilidade e performance no ar.

### Tipos de UAVs

Existem os mais variadíssimos tipos de UAVs, e cada um conta com diferentes funcionalidades, propósitos e processos de operação [2]:

- **HALE (*High Altitude Long Endurance*)**

Este tipo de veículo caracteriza-se principalmente por poder voar a grandes altitudes (até cerca de 15 km) possuindo um elevado nível de autonomia. O seu uso é praticamente exclusivo a missões militares de reconhecimento e vigilância.

- **TUAVs (*Tactical UAVs*)**

Apesar de também terem uso maioritariamente militar, estes veículos estão também a ser utilizados com fins comerciais para transportes de mercadorias até cerca de 20 kg e mapeamentos aéreos. São normalmente concebidos para operar entre os 100 e os 300 m.

- **MAVs (*Micro UAVs*)**

Estes veículos são os menores, mais leves e acessíveis, e por essa mesma razão conseguem realizar atividades que os tipos anteriormente apresentados normalmente não conseguem. São na sua grande maioria utilizados para finalidades recreativas e científicas.

### Tipos de Estruturas para UAVs

É possível fazer uso de diferentes tipos de estruturas especialmente desenhadas para as atividades de descolagem e aterragem tendo em conta a aerodinâmica e tipo de uso [3]:

- **Asa-Fixa**

- **Multi-Rotor**

Um veículo do tipo asa-fixa contém duas asas, o que lhe permite atingir grandes velocidades aéreas e planar em pleno ar. Estas características possibilitam um menor consumo de energia e por consequente uma maior autonomia para longos voos, quando comparado com estruturas multi-rotor. As desvantagens prendem-se com a limitação de propulsão horizontal para descolar, aterrissar e a dificuldade em pairar no ar. É comum ser utilizado para fins recreativos e militares.

Diferentemente dos veículos de asa-fixa, os veículos do tipo multi-rotor não possuem asas de voo e são capazes de pairar no ar. Estes podem ser classificados de acordo com o número de hélices e as montagens mais comuns são: *Quadcopters* (4 hélices), *Hexacopters* (6 hélices) e *Octacopters* (8 hélices). Uma das grandes vantagens está na possibilidade de poderem descolar e aterrissar de forma vertical ao solo, como também de permitir um maior nível de controlo e estabilidade em curtas/médias distâncias, quando comparado com os de asa-fixa. Algumas das desvantagens são uma menor velocidade máxima de voo e um maior consumo de energia, que por consequente leva a uma menor autonomia.



(a) UAV de Asa-Fixa [4]

(b) UAV Multi-Rotor [5]

Figura 2.1: Tipos de Estruturas para UAVs.

### 2.1.2 Tecnologias Anti-Colisão para UAVs

O desvio de obstáculos e possíveis colisões são algumas das características mais importantes e essenciais de drones modernos, tanto em navegação autónoma, como não autónoma, nomeadamente para as *Intelligent Flying Machines* (IFMs) [6]. A possibilidade de colisão com outros objectos pode causar tanto danos no próprio UAV como em objetos, pessoas e animais. Existem dois principais tipos de obstáculos que podem surgir durante o tempo de voo, sendo estes estáticos tais como paredes ou outros objetos imóveis ou dinâmicos tais como outros *drones*, pessoas ou animais [7]. Para um desempenho razoável, sistemas de deteção de obstáculos devem ser capazes de correr em tempo-real ou com uma alta taxa de atualização, com um *Field-of-View* (FoV) e alcance de deteção suficientes [8]. De forma a evitar possíveis colisões, o *drone* deve não só detetar o obstáculo como também recalcular a trajetória e seguir outro rumo. O conceito visual deste tema é apresentado na Fig. 2.2.

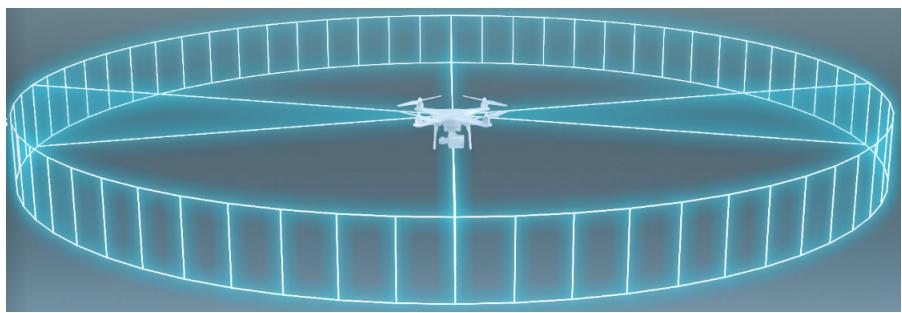


Figura 2.2: Sistema Anti-Colisão de 360° [9].

Já existem no mercado algumas tecnologias anti-colisão embbebidas em *drones* comerciais tais como o DJI Mavic 2 (sensores por todos os lados), DJI Phantom 4 Pro V2 (2 sensores ultra-sónicos e 4 sensores monoculares) e o Yuneec Typhoon H Pro (Câmara 3D Intel RealSense R200).

Um simples sensor orientado para o solo pode funcionar como um altímetro para que o veículo possa manter uma altitude constante enquanto que uma matriz de sensores consegue fornecer até 360° de monitorização de dados de proximidade, tal como ilustrado na Fig. 2.3. Contudo, um sistema não é apenas composto de sensores, são também necessários algoritmos sofisticados a trabalhar em conjunto tais como modelos matemáticos e tecnologias SLAM (*Simultaneous Location And Mapping*) [10]. A aplicação das tecnologias de desvio de obstáculos não servem só para evitar obstáculos mas também para detetar, seguir ou reconhecer objectos do ambiente [11].

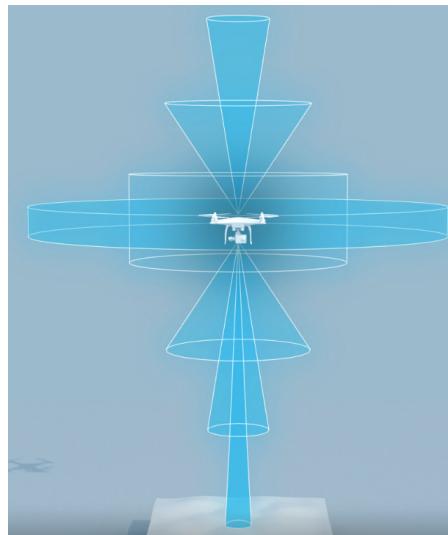


Figura 2.3: Sistema Anti-Colisão de 360° com Altímetro [9]

Soluções para evitar obstáculos estáticos são de fácil aquisição, enquanto que sistemas que permitem a deteção e desvio de objetos dinâmicos são de desenvolvimento mais complexo uma vez que é necessário uma rápida percepção do ambiente e capacidade de controlo [8]. Todas estas soluções sofrem de algumas contrapartidas normalmente distintas. A evasão de colisões é uma das capacidades chave para a navegação autónoma, que normalmente envolve quatro passos [12]:

### 1. Deteção

O primeiro passo envolve o uso de tecnologias sensoriais e/ou de posicionamento no espaço (ex: GPS) para determinar a presença de obstáculos no seu ambiente e para recolher dados de distância, que pode ser útil para prevenir uma potencial colisão, tais como a posição relativa e/ou a velocidade do obstáculo no caso de este ser dinâmico. Importante ter em atenção de que o sinal do GPS tem muitas das vezes uma precisão muito limitada dependendo do local de recepção do sinal.

### 2. Decisão

O próximo passo determina se o obstáculo detectado é uma ameaça e como deve ser evitado, tendo o UAV de ajustar a sua trajetória.

### 3. Ação

Após a deteção e decisão, é o momento de executar a manobra de anti-colisão e realizar o planeamento de uma nova trajetória.

### 4. Resolução

Este último passo determina se a navegação pode ser resumida em segurança.

## Sensores de distância

Várias tecnologias sensoriais têm sido propostas para deteção de obstáculos, tais como: Infravermelhos (IV), LiDAR, visão monocular e visão estereoscópica, assim como soluções híbridas [13][14][15]. No entanto, algumas destas tecnologias têm limitações, tais como: alto consumo de energia, requisitos de peso e/ou altura, e custo. Por este motivo, muitas empresas utilizam diferentes tipos de sensores de forma a complementarem-se e combaterem as

limitações de ambos [16]. Contrariamente às tecnologias mencionadas anteriormente, os sensores eletro-ópticos providenciam uma solução de tamanho, peso e custo reduzido na deteção de obstáculos para *drones* [17][18][19]. Estes sensores são portanto particularmente adequados para pequenos *drones* [20][21][22][23]. A criação de imagens térmicas infravermelho têm sido, também, consideradas para o mesmo propósito [24][25].

## SONAR

Sensores do tipo SONAR (*Sound Navigation and Ranging*) enviam sinais sonoros, esperam pelo retorno dos mesmos e calculam a distância ao obstáculo com base na velocidade do som. Estes sinais são normalmente modulados em altas frequências de forma a serem imperceptíveis por seres humanos e sofrerem menores perturbações ambientais [26]. No que toca à sua estrutura, é comum apresentarem duas aberturas, uma de transmissão e outra de recepção tal como demonstrado na Fig.2.4. Uma das suas desvantagens prende-se com a impossibilidade de detetar sons vindos de superfícies absorventes de som [27][28].

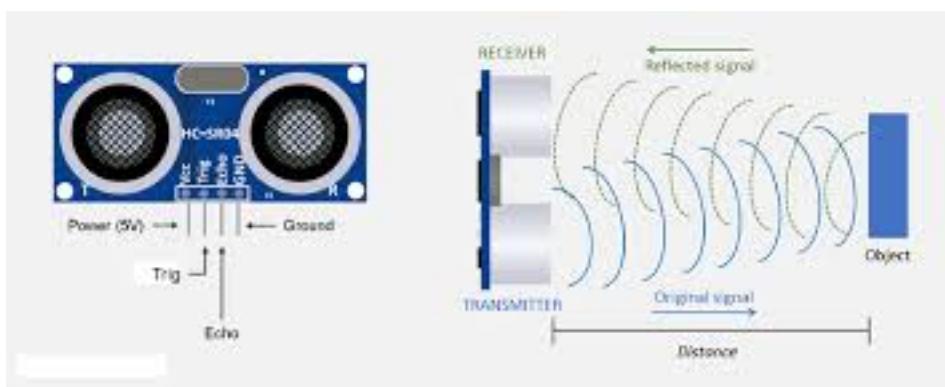


Figura 2.4: Sensor SONAR HC-SR04 [29].

## LiDAR

LiDAR (*Ligth Detection and Ranging*) tem um funcionamento base semelhante ao SONAR, mas neste caso utilizando luz ao invés de som, o que traz algumas vantagens e desvantagens. Este conceito irá ser mais explorado no Cap. 2.2. A sua principal desvantagem está relacionada com ambientes de muito alta luminosidade e a presença de fumos ou nevoeiro [27][28].

## Visão Monocular e Visão Estereoscópica

A técnica de visão monocular visa o processamento de imagem a partir de uma única lente ou câmara, de forma a se poder obter uma estimativa de profundidade. A visão esterescópica utiliza não uma mas duas lentes ou câmaras para atingir o mesmo objetivo. Correspondendo as imagens da câmara do lado esquerdo e do lado direito, é possível criar um espaço de pontos tridimensionais que contêm informação de cor e distância relativa ao dispositivo. Um exemplo de uma câmara esterescópica, uma Intel Real Sense, é apresentado na Fig.2.5.



Figura 2.5: Câmera Estereoscópica Intel Real Sense [30].

## 2.2 LiDAR

### 2.2.1 Introdução

LiDAR (Light Detection And Ranging), também conhecido como LADAR (LAser Detection And Ranging) é uma tecnologia de medição ativa de distâncias, geralmente utilizado para obter a distância a um ou mais pontos de um objeto [31]. Esta tecnologia é baseada em medições precisas do atraso de tempo entre a transmissão de um sinal de pulso LASER e a sua receção [32]. Sensores LiDAR podem ser divididos em diferentes categorias, com base nas suas características de iluminação e deteção do alvo [31]:

- ***Scanning***, Fig. 2.6:

- Ilumina o alvo através de varrimentos de um raio LASER e deteta o sinal de retorno com um simples detetor.
- Envia pulsos discretos de LASER e depois aguarda o pulso de retorno no detetor.

- ***Detector Arrays***, Fig. 2.7 and Fig. 2.8:

- Ilumina todo o alvo de uma vez só e deteta o retorno com uma matriz de detetores.
- Modula o sinal do LASER e posteriormente deteta a mudança de fase do sinal retorno.

- ***Spatial Light Modulators***:

- Ilumina porções do alvo seguindo um padrão e depois deteta o sinal de retorno com um único detetor (com a ajuda de algoritmos de deteção).
- Consiste em codificar uma sequência de números pseudo-aleatórios para o LASER e depois realizar uma auto-correlação do sinal valor do sinal de retorno

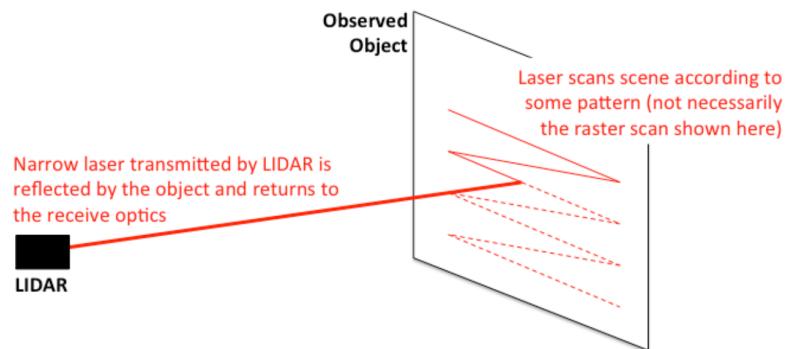


Figura 2.6: Scanning LiDAR [31].

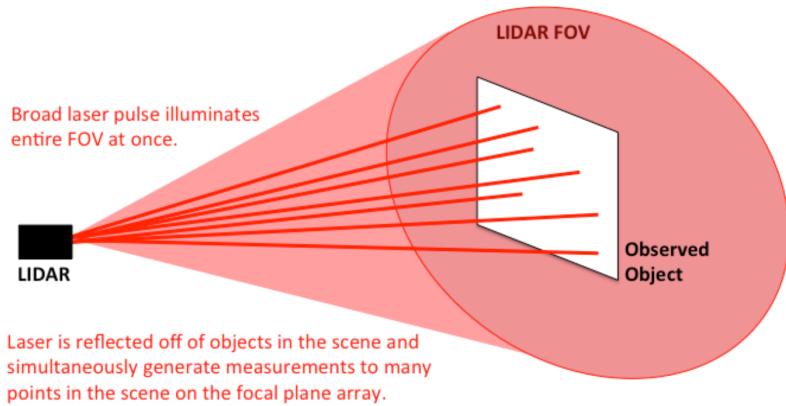


Figura 2.7: Detector Arrays [31].

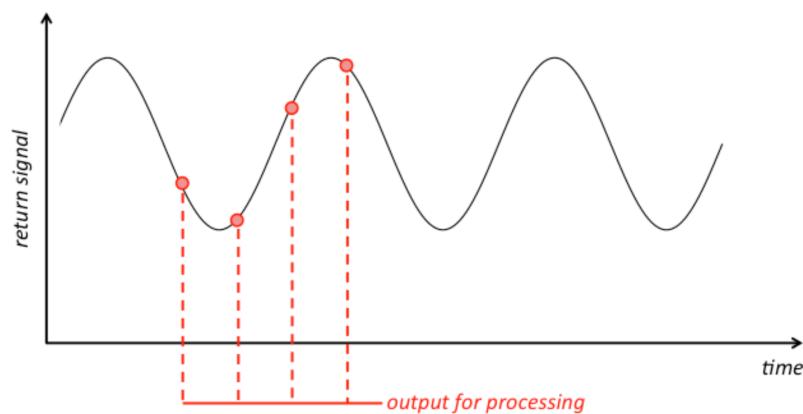


Figura 2.8: Detector Arrays Modulation [31].

### Vantagens e Desvantagens

Os sensores LiDAR distinguem-se por possuírem sistemas *built-in* de processamento de sinal e um estreito FoV, gerando dados críticos acerca da distância, de forma eficiente para deteção de obstáculos, assegurando uma navegação segura. Estes também podem trabalhar em baixos níveis de contraste e sombra, e até em escuridão. Apesar do grande potencial de aplicação de sensores do tipo LiDAR, este tipo tecnologia ainda não está presente em grande escala no nosso dia-a-dia. Há três principais limitações que limitam o uso convencional destes sensores [33]:

- **Saúde ótica**

É quase obrigatório utilizar proteção nos olhos quando se lida com LASERs e requere-se uma supervisão constante,

- **Custo**

Devido a ser uma tecnologia relativamente recente, o custo ainda é bastante alto para muitas aplicações,

- **Confiabilidade**

Apesar do rápido avanço desta tecnologia, a falta de confiabilidade em sistemas de LASER convencionais têm sido um problema para o seu uso de rotina.

Outra desvantagem prende-se com a reflexão da luz em objectos apesar de a refração ser diferente em diferentes tipos de superfície, sendo os espelhos e vidros os maiores “inimigos” do uso de sistemas de LASER. Existem também reflexões indiretas, reflexões vindas de outros pulsos, reflexões múltiplas onde os fotões incidentes atingem uma superfície, saltam para outra e retornam após a reflexão primária, reflexos difusos de superfícies ásperas ou irregulares e muito mais. Por este motivo é necessário realizar uma análise mais aprofundada do sinal, de forma a maximizar o SNR

(*Sound Noise Ratio*) e minimizar a distorção através de algoritmos complexos. Superfícies opacas e irregulares podem também causar complicações significativas para as medições, pois pode ser difícil para o sensor entender de onde vieram os fotões. Em pequenos espaços, os fotões fazem ricochete em paredes e outros objetos, confundindo ainda mais a distância medida. Estes conceitos são visualmente apresentados na Fig. 2.9.

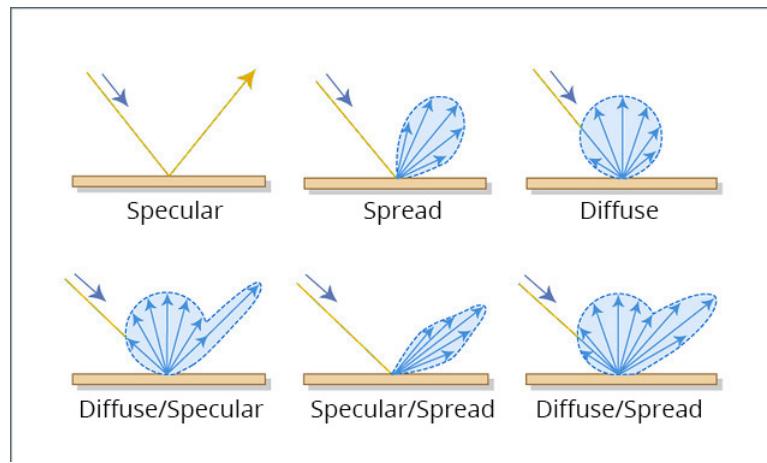


Figura 2.9: Situações de Reflectividade da Luz [34].

Também é possível realizar-se uma ligeira comparação entre vários tipos de tecnologias de medição de distâncias, tal como mostra a Fig. 2.10.

	Ultrasonic	Camera	RADAR	LiDAR
<b>Cost</b>	Low	Low	Medium	High
<b>Size</b>	Small	Medium	Small-medium	Medium-Large
<b>Speed Detection</b>	Low	Low	High	Medium
<b>Sensitivity to color</b>	No	High	No	No
<b>Robust to weather</b>	High	Low	High	Medium
<b>Robust to day and night</b>	High	Low	High	High
<b>Resolution</b>	Low	High	Medium	High
<b>Range</b>	Short	Medium-long	Short, med, long	Long

Figura 2.10: As diferentes Tecnologias de Medições de Distância [35].

### Espetro Eletromagnético, Foto-Emissor e Foto-Receptor

Para diferentes tipo de aplicação existem diferentes comprimentos de onda eletromagnéticos usados em sensores LiDAR, sendo que existem vantagens e desvantagens atribuídas a cada comprimento utilizado. O uso mais comum costuma ser perto dos infravermelhos (NIR - *Near Infrared*) [36]. A Fig. 2.11 mostra-nos alguns comprimentos de onda utilizados em contraste com a sensibilidade ou emissão de luz solar.

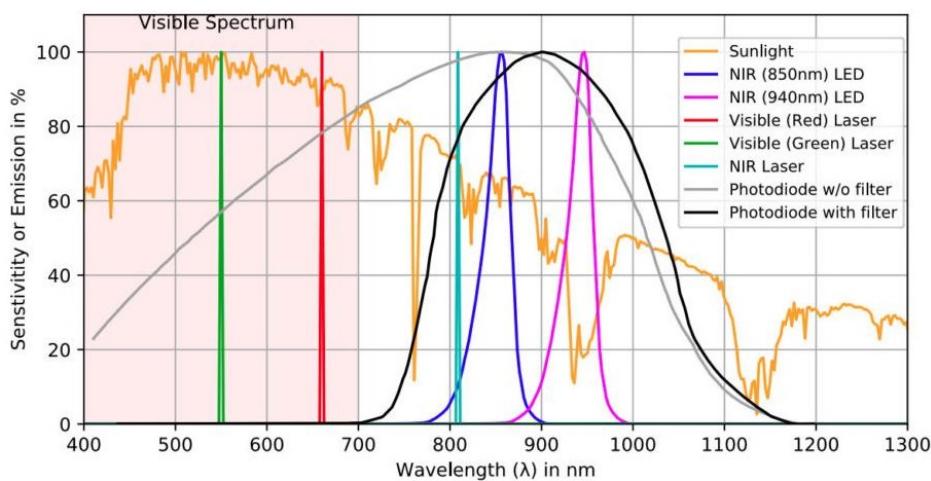


Figura 2.11: Resposta Espectral Genérica de Sensores LiDAR para diferentes Comprimentos de Onda [36].

Inicialmente, o único tipo de emissor com características favoráveis era o PLD (*Pulsed Laser Diode*), operando normalmente a 950 nm devido à baixa absorção atmosférica. Mas recentemente tem havido uma tendência a utilizar LEDs infravermelhos, pois são mais facilmente controláveis e a sua potência de saída é atualmente aceitável para estes propósitos. Os LEDs infravermelhos são *eye-safe*, sem necessidade tomar precauções extra. Requerem também menos potência elétrica do que os PLDs e são mais eficientes.

De forma a capturar os fotões refletidos e convertê-los em sinais elétricos, são utilizados dois tipos de sensores. Os mais utilizados são os foto-díodos PIN (*Positive-Intrinsic-Negative*) e os APD (*silcion Avalanche PhotoDiode*).

### 2.2.2 Princípio Time-of-Flight

A tecnologia *Time-of-Flight* (ToF) utiliza pulsos de luz e a medição do seu “tempo de voo” para determinar a distância a um objeto utilizando a reflecção de um pulso ou de um conjunto de pulsos, tal como ilustrado na Fig. 2.12. Isto baseia-se na associação inequívoca de cada sinal de retorno em relação ao pulso emitido que o causou [37]. Esta tecnologia tornou-se a chave para medições de profundidade numa grande variedade de aplicações, abrangendo diferentes áreas tais como: industrial, consumo e automóvel. Pode ser utilizada em distâncias desde 1 m até vários quilómetros [38].

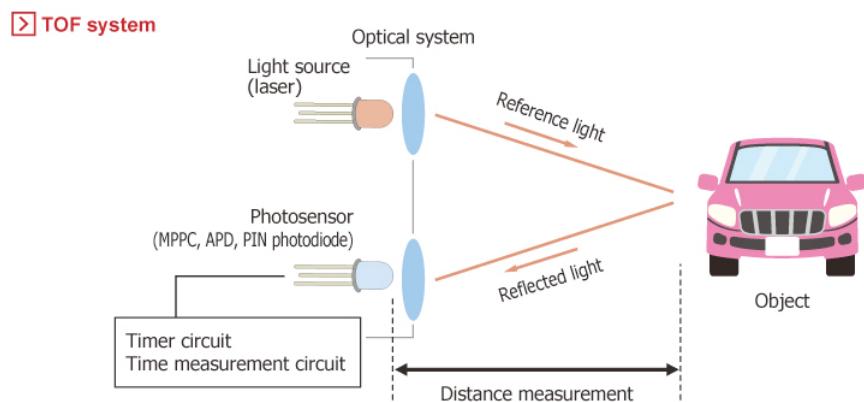


Figura 2.12: Princípio *Time-of-Flight* [39].

Esta medição pode ser feita em várias formas, utilizando detetores de alta velocidade e/ou *gating* de sensores de imagem [40].

Avanços na miniaturização de semicondutores levou a soluções mais compactas e em formatos mais pequenos, tornando assim possível a integração de sensores ToF em vários dispositivos, como por exemplo *smartphones* [41].

Num ambiente ideal e com um sensor de distância perfeito, a distância pode ser calculada como demonstrado na Eq. (2.1), contudo, devido a limitações de processamento, alguns sensores recorrem à Eq. (2.2):

$$r = \frac{ct}{2}, \quad (2.1)$$

$$r = \frac{c}{2} \frac{1}{2\pi f} \Delta\phi, \quad (2.2)$$

onde:

- $r$  Distância (m),
- $c$  Velocidade da luz (m/s<sup>2</sup>),
- $t$  *Time-of-Flight* (s),
- $f$  Frequência do sinal (Hz),
- $\Delta\phi$  Ângulo de mudança de fase (rad).

Existem dois tipos de sensores ToF [42]:

- **Diretos (DToF)**

Sensores do tipo DToF utilizam pequenos pulsos de luz, medindo diretamente o tempo de retorno desse pulso vindo do objeto (Eq. (2.1)) e são considerados mais apropriados para medições de grandes distâncias.

- **Indiretos (IToF)**

Os sensores do tipo IToF emitem um sinal contínuo de luz modulada. Quando o fotoreceptor deteta a reflexão, mede o tempo e a fase da luz refletida para calcular a distância ao objeto (Eq. (2.2)). Esta alternativa tem vindo a ser implementada em várias aplicações ao nível do consumidor comum, para mapeamento de profundidade e de alcance. Estes têm demonstrado uma alta resolução espacial com a capacidade extra de detetar múltiplos objetos ao longo de um grande campo de visão (FoV). Contudo, estes são limitados a alcances de pequena distância [43].

### 2.2.3 Tecnologia VCSEL

Os *Vertical Cavity Surface Emitting Lasers* (VCSELs) são utilizados numa grande variedade de aplicações e tornaram-se a tecnologia de infravermelhos dominante em sistemas de deteção a curta distância. Isto deve-se a várias características vantajosas, tais como alta eficiência, baixo ruído de coerência, e o seu perfil de feixe óptico. Podem ser também facilmente produzidos em massa, em ambiente industrial. O resultado é um LASER *surface-mountable* que tem um alto pico de poder óptico e baixa divergência [44]. A representação da sua construção é ilustrada na Fig 2.13.

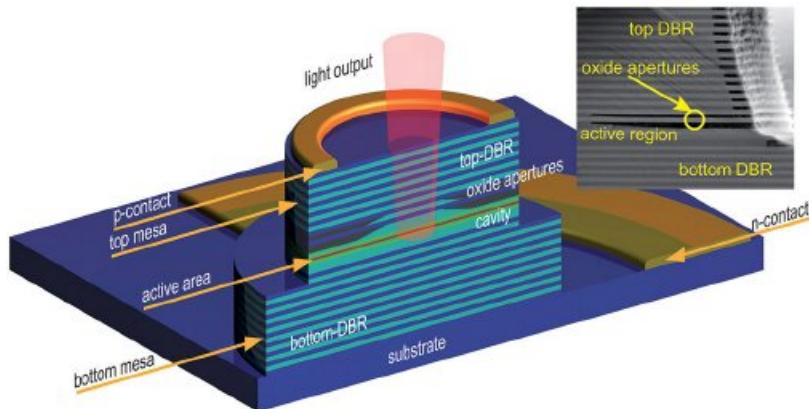


Figura 2.13: Estrutura de um VCSEL [44].

“The anodes of the individual VCSEL elements are connected in parallel to a metal pad on the sub-mount via solder bumps that are reflowed to the metallization of the sub-mount. The cathode contact is common for each die and is achieved by intentionally shorting a similar mesa structure to a current-carrying return layer in the epitaxial structure of the VCSEL array die. These cathode contacts are similarly bonded to common metal layers on the sub-mount” [45].

### 2.2.4 Flash LiDAR

A tecnologia Flash LIDAR utiliza, tipicamente, um feixe expandido para iluminar a cena ao longo de um maior campo de visão em comparação aos sistemas ToF comuns [40]. Os sinais de retorno são então detetados por um vetor detetor (ou *imager*), que tem a capacidade de medir o tempo de retorno dos pulsos da fonte de luz, relativa aos objetos na cena. A fonte do LASER tem uma potência de pico elevada e tem de enviar pulsos estreitos para providenciar a densidade de potência necessária para o alcance desejado [40]. A tecnologia VCSEL costuma ser uma fonte fiável para estes sistemas. O tamanho do vetor detetor determina o campo de visão, enquanto que tamanho do vetor de elementos define a resolução. Estes podem ser usados com um campo de visão extenso, de forma a detetar uma grande quantidade de objectos, assim como na Fig. 2.14 a), ou com um campo de visão estreito, de forma detetar objetos a longas distâncias, como ilustrado na Fig. 2.14 b). Uma solução híbrida pode ser utilizada para reduzir o pico de potência necessária [45].

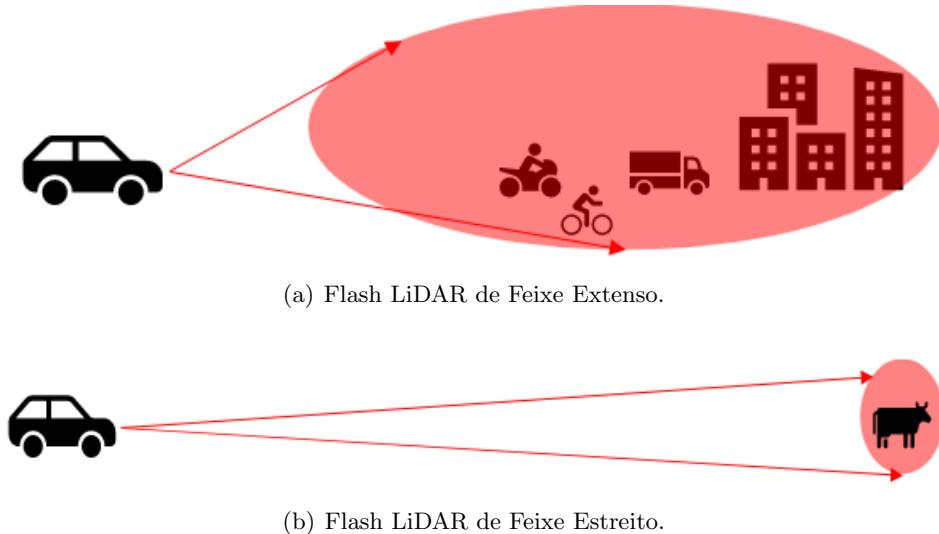


Figura 2.14: Topologias de Flash LiDAR [45].

### 2.2.5 Tecnologias Rotacionais

Sensores LiDAR rotacionais conseguem rodar em torno do seu próprio eixo e permitem uma visão a 360°, fazendo uso de um feixe de laser altamente culminado, que é analisado, tipicamente, utilizando rotação de espelhos para analisar o feixe no espaço. Várias combinações de espelhos e elementos rotativos podem ser utilizados para analisar o campo de visão a duas dimensões. Múltiplas fontes podem ser rodadas para providenciar linhas de resolução vertical. O detetor é geralmente um único elemento que mede o tempo de voo do impulso de luz e é automaticamente correlacionado com a orientação conhecida. O resultado é uma nuvem de pontos densa [46]. Um exemplo de uma nuvem de pontos resultante é apresentado na Fig. 2.15, que mostra um campo de gelo com pessoas a patinar.

Existem várias empresas que produzem este tipo de sensores. Uma delas é *SLAMTEC*, criadora do *RPLIDAR A1* [47], a primeira alternativa de baixo custo de LIDARs de rotação, em 2012. Este sensor é exibido na Fig. 2.15 e é baseado no princípio de triangulação, tendo um alcance de 12 metros em 360° com uma frequência de amostragem até 8000 Hz.

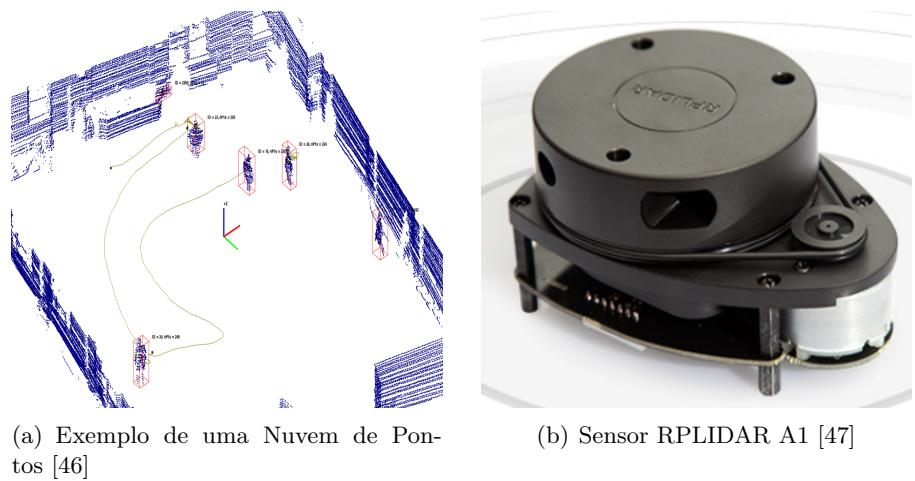


Figura 2.15: Tecnologias LiDAR Rotacionais.

### 2.2.6 Sistema Solid-State

Sensores LiDAR de rotação mecânica introduzem algumas preocupações a longo prazo, tais como desgaste e desvios em situações turbulentas. Contrariamente, a tecnologia *solid-state* LiDAR é promissora no sentido em que oferece resoluções superiores, é mais rápida, mais leve, mais robusta e menos cara do que os sistemas mecânicos tradicionais. Conseguindo também atingir dimensões extremamente pequenas que permitem uma excelente integração em UAVs. Os *solid-state* LIDARs são sensores baseados em silício e não requerem partes mecânicas de rotação. Um tipo de *solid-state* utiliza movimentos de espelhos baseados em tecnologias de sistemas micro-eletromecânicos (MEMS) para realizar o “*scan*” do ambiente. Outra técnica utilizada é o uso de um vetor de fase óptica (OPA), com múltiplas fontes independentes que direcionam e escolhem dinamicamente os pulsos em diferentes direções ao ajustar o vetor de cronometragem. Um sensor LiDAR baseado em OPA tem uma maior durabilidade e confiabilidade por não executar qualquer tipo de movimento, e permite também uma rápida análise. Têm um tamanho inferior em comparação aos baseados em MEMS mas costumam ser mais caros.

Nos dias de hoje, ainda não existe nenhum *solid-state* sem qualquer movimento, que conseguia fornecer um campo de visão de 360°. Como alternativa, costumam ser utilizados múltiplos sensores com alguma sobreposição. “Na realidade, uma unidade mecânica bem projetada pode ser conferir um grande nível confiabilidade e ser compacta, como resultado de décadas de experiência com a sua implementação e prováveis pontos fracos. Ainda assim, um projeto sem partes móveis é muito atrativo” [48].

### Deteção de objetos

A deteção e previsão de possíveis colisões tem sido explorada por vários investigadores no ramo da robótica [49]. Vamos assumir que um obstáculo consiste numa nuvem de pontos que representa a sua superfície externa. Um obstáculo é inicialmente desconhecido até que é detetado pelo *drone*. Sob um ponto de vista prático, o sensor LiDAR tem um FoV (*Field-of-View*) limitado, como também um alcance máximo. A junção dos dados adquiridos com os possíveis erros de medição é utilizada para formar uma esfera delimitadora e um cone de colisão [50]. O cone de colisão está diretamente relacionado com o ângulo de FoV do sensor, no qual o ponto pode estar inserido, enquanto que a esfera delimita o erro da medição feita pelo sensor.

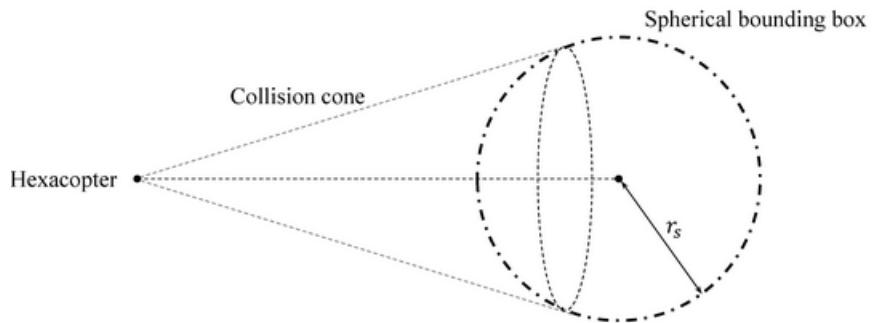


Figura 2.16: Esfera Delimitadora e Cone de Colisão [50].

A abordagem de cone de colisão é um dos métodos possíveis para o desvio de obstáculos. Se o vetor velocidade do *drone* estiver contido dentro do cone de colisão, então o obstáculo detetado é uma potencial ameaça ao UAV, a não ser que mude de trajetória. Neste caso, o veículo deve realizar uma manobra de desvio de colisão [50].

### Multi-Directional Arrays

Este tipo de soluções agrupam um conjunto de elementos independentes integrados num único dispositivo. Desta forma torna-se possível explorar um largo FoV, ideal para soluções de desvio de obstáculos.

A Terabee é uma empresa que projeta, desenvolve e produz sensores e soluções ToF, leves e pequenas, normalmente baseadas em infravermelhos otimizados para *drones* e aplicações tais como SLAM e desvio de obstáculos [51]. É parceira formal do CERN, *European Centre for Nuclear Research* e venceu vários prémios. A maior parte dos seus produtos atuais fornece APIs para o ROS (Robot Operating System), PixHawk Flight Controller e oferece sistemas de proteção contra *crosstalk*. As suas soluções oferecem vários modos de operação dependentes do tipo de ambiente ou aplicação (Fig. 2.17):

- **Simultâneo:** Todos os sensores ativados ao mesmo tempo, beneficia de uma elevada frequência de amostragem.
- **Sequencial:** Os sensores são ativados à vez. Isto é utilizado em ambientes com alta probabilidade de *crosstalk*.
- **Torre:** Os sensores são ativados quatro de cada vez, igualmente distanciados de cada um. Esta configuração costuma ser utilizada para monitorização de ambiente de 360º.

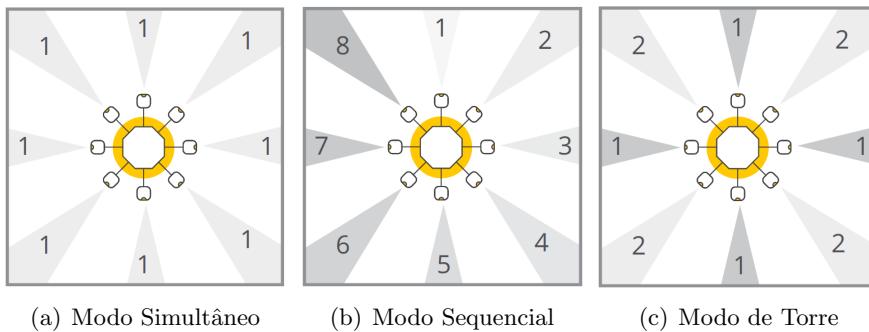


Figura 2.17: Modos Operacionais de Matrizes Multi-Dimensionais

Alguns dos seus produtos são: o TeraRanger Evo, o TeraRanger Multiflex, o TeraRanger One e o TeraRanger Tower Evo.

### **TeraRanger Evo Mini Array Kit (Fig. 2.18)**

Desenvolvido para pequenas distâncias, este kit modular de baixo custo está disponível em 2 configurações, com 4 ou com 8 sensores.

### **TeraRanger Multiflex (Fig. 2.19)**

TeraRanger Multiflex é um kit de deteção modular e de baixo custo que contém 8 módulos de sensores num *flexible bus*. Este é baseado num sensor ToF da STMicroelectronics chamado VL53L0X.

### **TeraRanger One Kit (Fig. 2.20)**

Este kit já contém uma moldura para inserir os sensores.

### **TeraRanger Tower Evo (Fig. 2.21)**

O TeraRanger Tower Evo vem com duas opções, com uma taxa de *update* de 600 Hz ou com um alcance de 60m. Especificações mais detalhadas acerca de cada sensor estão disponíveis na Tab. 2.1. É importante referir que a maioria utiliza microcontroladores das famílias STM32F0 e STM32F1.

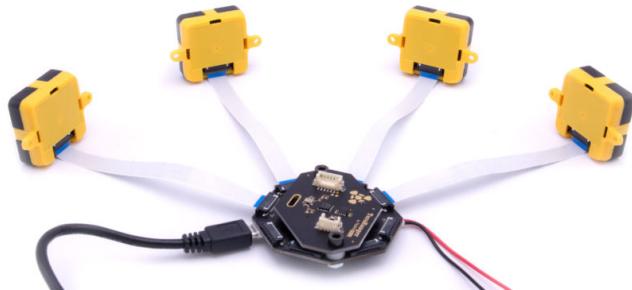


Figura 2.18: TeraRanger Evo Mini Array Kit [51].



Figura 2.19: TeraRanger Multiflex [51].

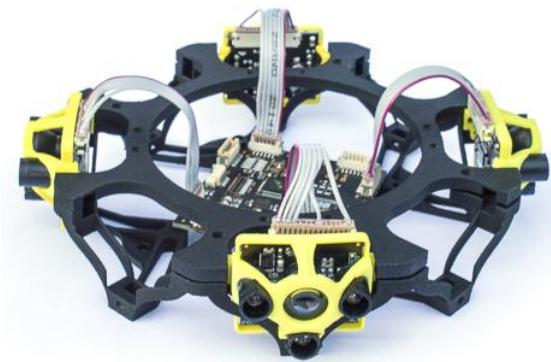


Figura 2.20: TeraRanger One Kit [51].



Figura 2.21: TeraRanger Tower Evo [51].

Tabela 2.1: Comparação de Especificações dos Produtos TeraBee.

	TeraRanger Evo Mini Array Kit	TeraRanger Multiflex	TeraRanger One Kit	TeraRanger Tower Evo 600 Hz	TeraRanger Tower Evo 60m
Tecnologia	Infrared ToF	Infrared ToF	Infrared ToF	Infrared ToF	Infrared ToF
Resolução (mm)	1	1	5	5	5 / 20
Comp. de Onda (nm)	940	940	940	940	940
Alcance (m)	0.03 - 3.3	0.05 - 1.1 / 2.0	5 - 6 / 14	0.75 - 8	0.5 - 60
Freq. Amostra. (Hz)	2 / 8 / 16	50/n ou 30/n	600 - 1000	320 / 600 per sensor	240 / 120 per sensor
Precisão (cm)	± 2	± 6 / ± 3	± 4	± 12	± 4 / 1.5%
Peso (g)	60 / 100	20	60	92 / 135	92 / 135
Campo de Visão (°)	27 per Sensor 108 / 216	20	3	2 per sensor	2 per sensor
Interfaces	USB 2.0 UART I2C	USB 2.0 UART I2C	USB 2.0 UART I2C	USB 2.0	USB 2.0
T. de Alimentação (V)	12	5	12	12 - 24	12 - 24
Consumo de Corrente (mA)	225 / 425	50	250	1100 (max)	1100 (max)
Preço (€)	179 / 279	99	250	399 - 599	399 - 599

# Capítulo 3

## Análise de Requisitos

Após uma análise cuidada do estado de arte e das orientações fornecidas pelos docentes, chegou-se à conclusão de que o sistema proposto deverá:

- **• Detectar obstáculos distantes, de forma rápida e precisa:**

Considera-se que o sensor escolhido não deverá ter um alcance menor que  $5\text{ m}$  em plena exposição solar. Isto é derivado à grande envergadura de alguns dos UAVs presentes no laboratório. A taxa de amostragem deverá ser de cerca de  $50\text{ Hz}$ , oferecendo alguma margem de distância percorrida pelo drone até detectar o obstáculo, travar em segurança, ou se for o caso, restabelecer uma nova rota.

- **• Utilizar entre 4 e 8 sensores LiDAR:**

Apesar de os UAVs presentes no laboratório se movimentarem preferencialmente com “nariz” para a frente (relembrando que os drones se podem mover para praticamente qualquer direção), deverão ser colocados vários sensores, que realizarão a deteção de obstáculos à volta do veículo. Desta forma é possível obter diferentes campos de visão à volta do UAV e evitar colisões em momentos em que o drone não esteja a voar com o “nariz” na direção do movimento, ou com obstáculos dinâmicos em seu redor.

- **Permitir o ajuste de parâmetros:**

De acordo com a missão ou ambiente em causa, o utilizar poderá preferir ou necessitar de realizar alterações na taxa de atualização dos sensores, ou na distância máxima detetada, que normalmente estão inversamente relacionados. Também deverá ser flexível definição da quantidade de sensores utilizados e o seu modo de funcionamento.

- **Receber e transmitir informação rapidamente:**

Para além da rápida deteção do obstáculo por parte do sensor, é também necessário que essa informação chegue ao computador de bordo da forma mais rápida possível. Por isso deverá ser incluído um sistema de interrupção no caso de deteção de um objeto por *threshold*, de acordo com a velocidade do mesmo. Existem alguns sensores que emitem interrupções e têm *thresholds* programáveis.

- **Flexibilidade de Protocolo de Comunicação**

O projeto deverá permitir, de acordo com as configurações do utilizador, a escolha do protocolo de comunicação a utilizar com o computador de bordo, entre os quais: RS232, RS485, USB e CAN.

- **Baixo Consumo, Economicamente Viável e Leve:**

Sendo um AUV um veículo aéreo alimentado puramente a bateria, é necessário ter em atenção o consumo de energia de forma maximizar o tempo de vôo. Por isso, o sistema de deteção de obstáculos deverá consumir o menor nível de energia possível. Para que seja realmente viável a sua construção e utilização em AUVs, é necessário que este seja economicamente viável, tendo, no limite, o mesmo custo de um AUV. Terá também de ser, obviamente, o mais leve possível, dado que os drones, em si, costumam ser também bastante leves e não têm muita capacidade de transporte de peso.

De forma a conseguir atingir os objetivos pretendidos, são necessários, pelo menos e principalmente, os seguintes elementos:

- **Sensores ToF *Solid State LiDAR*:**

O alcance requerido destes depende de variadíssimos fatores tais como a velocidade nominal do *drone*, a sua rapidez de travagem, a precisão da deteção do obstáculo, da frequência de amostragem do próprio sensor e do ambiente em que irá ser utilizado. Todas estas características estão interligadas e devem ser levadas em consideração. Os máximos alcances de sistemas já existente variam entre 1.1 *m* e 60 *m*, enquanto que as frequências de amostragem variam entre 2 e 1000 *Hz* por sensor. O conjunto de sensores pode, ou não, realizar a cobertura de 360. Os ângulos "mortos", definidos como os ângulos fora do campo de visão do sistema de deteção de obstáculos, são muitas das vezes complementados com o movimento natural do veículo.

- **Placa Eletrónica de Interface e Desenvolvimento**

Deverá ser desenvolvida, durante o processo de criação do projeto, uma placa eletrónica (PCB - *Printed Circuit Board*), que utilize um microcontrolador de rápido processamento, e que suporte os protocolos de comunicação referidos anteriormente, assim como a interface com os vários sensores. Esta deverá ser alimentada diretamente a 12 *V* que é a alimentação nominal das baterias, e tal como referido anteriormente, o conjunto com os sensores LiDAR deverá ser robusto, economicamente viável e de baixo consumo.

- **UAV de Teste**

De forma a poder efetuar testes em situações reais e comprovar a robustez do sistema, será necessária a utilização de um UAV para realização de testes.

Esta página foi intencionalmente deixada em branco.

## Capítulo 4

# Arquitectura de Alto Nível

A estrutura inicialmente planeada teve grande inspiração nos produtos da empresa TeraBee. O objetivo seria ter uma estrutura central que continha a placa de processamento e comunicação, com os vários sensores presos a esta e seu redor. Acontece que este tipo de arquitetura não é aplicável nos UAVs de grandes dimensões por limitação dos sensores, perdendo-se um alcance considerável devido à distância entre o perímetro do *drone* e a sua zona central. Por isso será uma melhor opção colocar os sensores nas zonas mais periféricas do veículo, tal como demonstrado na Fig. 4.1. Foi também adicionado um servo motor rotativo na zona frontal do *drone*, de forma a obter um maior FoV, e assim, poder detetar obstáculos mais facilmente.

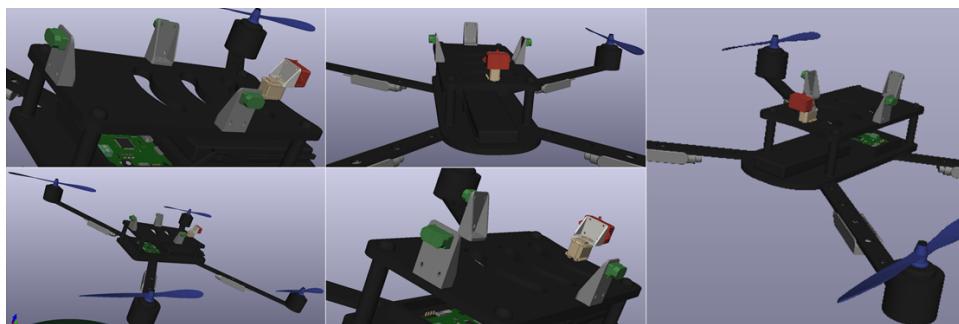


Figura 4.1: Arquitectura do Sistema.

A colocação dos sensores nas zonas mais periféricas traz algumas dificuldades extra, pois os protocolos de comunicação normalmente utilizados pelos sensores LiDAR (I2C e UART), são subjectivos a interferências nesta escala de distância, não sendo por isso robustos para este tipo de aplicação.

A solução passou por dividir o sistema em vários módulos genéricos e flexíveis, interconetados por comunicação CAN (protocolo mais robusto de comunicação do que os mencionados anteriormente) nos quais cada um é responsável por recolher informação de um sensor LiDAR, e controlar o servo motor, no caso do módulo da zona frontal (Fig. 4.2). Neste conjunto, existirá um módulo *master*, que poderá ser um qualquer, e que irá recolher também a informação dos outros módulos e enviar essa informação para o computador de bordo. A alimentação é também transmitida de módulo a módulo de forma a facilitar as ligações e a sua organização.

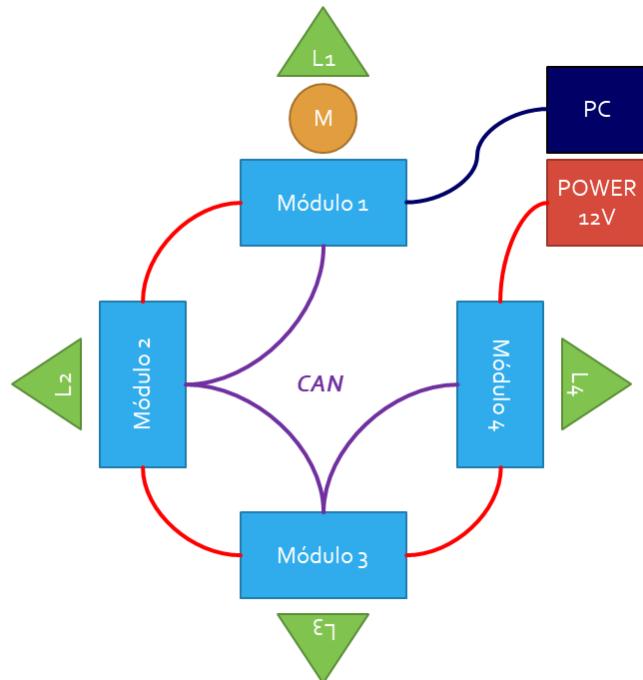


Figura 4.2: Esquema Geral do Hardware.

Tal como é possível visualizar na Fig. 4.3, a alimentação é sempre for-

## Capítulo 4

---

necida a 12 V sendo que os módulos terão de realizar a sua conversão para 5 V de forma a alimentar os sensores e alguns componentes, como também para 3.3 V com o objetivo de alimentar o resto dos componentes ativos.

A comunicação entre nós, ou módulos, como já foi referido, é realizada através do protocolo CAN, e para isso é necessário existir um *transceiver* que realize a transformação do sinal para os *standards* requeridos por este protocolo. O *jumper* da resistência de terminação é utilizado no caso de este ser o 1º ou o último módulo do sistema, de acordo com os padrões desse protocolo de comunicação. Existem conectores de *CAN INPUT* e de *CAN OUTPUT*, que também transmitem a alimentação de todo o sistema, sendo que as nomenclaturas *INPUT* e *OUTPUT* não devem ser interpretadas como tal, mas sim como uma forma de organização das ligações.

O microcontrolador terá um circuito oscilador dedicado (a cristal), por necessitar de grande precisão de *clock* na transmissão de informação. Terá também um conector SWD para a programação do mesmo. É também conectado um circuito de conversão lógica bi-direcional para comunicação com o servo motor e o computador de bordo, que utilizam 5 V como nível de comunicação lógico, ao contrário do microcontrolador que utiliza 3.3 V.

O controlo do servo motor (no caso do módulo frontal), poderá ser feito via comunicação TTL ou RS-485, sendo para isso necessário um *buffer* TTL ou um *transceiver* de RS-485. A seleção do protocolo é feita a partir da seleção de um primeiro *jumper* que define o protocolo a utilizar. Caso seja escolhido o protocolo RS-485, é necessário também selecionar outro *jumper* que determina a sua ligação ao servo motor ou ao PC, dado que só existirá um *transceiver* desse protocolo e que este tem apenas um *output*.

Por fim, a comunicação com o computador de bordo, caso este seja o módulo *master*, é feita através do protocolo RS-485, e para tal é necessário selecionar o *jumper* que define a direção do *transceiver* para o PC.

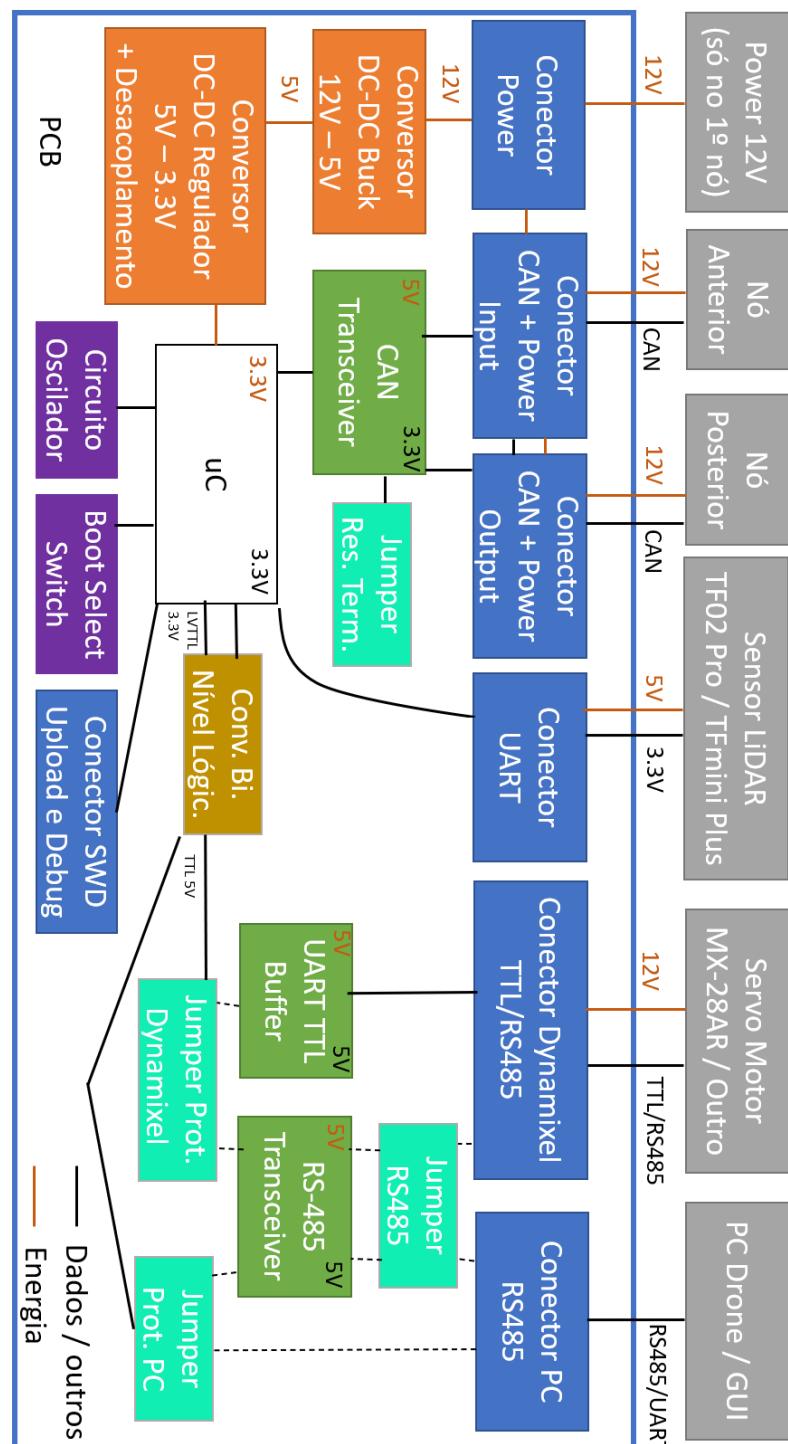


Figura 4.3: Arquitectura dos Módulos Genéricos.

# Capítulo 5

## Projecto

### 5.1 Descrição do *Hardware*

#### 5.1.1 Sensores LiDAR

Neste tipo de sensores, o alcance varia com as condições luminosas do ambiente e com a refletividade do objeto alvo. No pior dos casos, em condições *outdoor*, a luminosidade poder variar entre 32 Klux e 100 Klux [52].

#### Sensores FlightSense<sup>TM</sup>

A STMicroelectronics é uma empresa que produz diversos tipos de componentes electrónicos, entre os quais os sensores de distância da família *FlightSense<sup>TM</sup>*. Esta família de sensores recorre à tecnologia *ToF* referida no Cap. 2.2.2. Os sensores estão contidos num pequeno módulo, que contém o emissor do sinal, o receptor do mesmo, e um microcontrolador para processamento digital das medições realizadas. De acordo com a ST, a aplicação destes módulos é indicada para sistemas que requerem medição de distância precisa e de alta-velocidade, baixo consumo de energia e fácil integração mecânica, tais como sistemas de deteção de obstáculos para drones. Todas as opções apresentadas trabalham no comprimento de onda de 940 nm.

Estes sensores possuem uma saída de interrupção e uma entrada de *reset* [53]. A forma mais comum de alimentar os módulos é através do modo 2V8, utilizando uma única fonte de tensão.

Os módulos VL53L1 e VL53L3X são capazes de detetar vários objetos dentro de um mesmo FoV (Field-of-View) e tanto o VL53L1 como o VL53L1X possibilitam a programação customizada da região de interesse, permitindo a redução do FoV. O controlo destes sensores é feito via protocolo I2C, através de uma API (*Application Programming Interface*) compatível com microcontroladores STM32 (que ocupa 64kB de memória *flash*) e sistemas LINUX e Android. Existe também uma placa de expansão X-Nucleo, a X-NUCLEO-53L1A1. Uma breve comparação é realizada na Tab. 5.1 que demonstra as principais diferenças básicas entre os vários modelos. É necessário ter em atenção que alguns dos valores da Tab. 5.1 apenas conseguem ser obtidos em condições ideais, portanto não devem ser totalmente considerados para ambientes mais desfavoráveis. O sensor VL53L1 é o que verifica as melhores características *à priori*, mas que no entanto não atende aos requisitos mínimos em condições de grande luminosidade. O fabricante apresenta alguns resultados de testes de medição em ambos os sensores, mas nunca em condições semelhantes, o que torna a comparação de resultados **inviável**, e por essa razão não será considerada neste relatório. Estes sensores seriam no entanto uma opção bastante viável para pequenos *drones* de aplicações *indoor*.

---

<sup>1</sup>Preço unitário retirado do website: <https://pt.mouser.com>

Tabela 5.1: Comparaçāo de Sensores de Distācia.

Parāmetro	VL53L1	VL53L1X	VL53L3X
Alcance (cm)	800	400	500
Dist. Mínima (cm)	4	4	1
Full FoV (°)	até 27	15 a 27	25
Multi-zone	Sim	Sim	Não
Freq. Máxima (Hz)	125	100	125
Freq. Max. I2C (kHz)	1000	400	1000
Consumo Médio (mA)	16	16	16
Consumo de Pico (mA)	40	40	40
Tempo de inici. (ms)	40	40	40
Data lançamento	4/2017	2/2018	1/2020
Custo <sup>1</sup> (€)	6.21	5.45	4.19



(a) VL53L1

(b) VL53L1X



(c) VL53L3X

(d) VL53L5

Figura 5.1: Sensores FlightSense™, adaptado de [53]

### TF02-Pro e TFmini-Plus

Fabricados pela **Benewake**, o TF02-Pro e o TFmini-Plus são sensores LiDAR relativamente pequenos, *low-cost*, de baixo consumo e unidireccionais de médio alcance, baseados no princípio ToF. O TF02-Pro é capaz de detetar distâncias até  $40\text{ m}$ , com uma resistência à luminosidade de até  $100\text{ Klux}$ , enquanto que o TFmini-Plus é capaz de detetar distâncias até  $12\text{ m}$ , com uma resistência à luminosidade de até  $70\text{ Klux}$ . Ambos têm encapsulamento IP65, o que significa que são resistentes à água, poeiras e vibração. Estes módulos são baseados no dispositivo OPT3101 da Texas Instruments, que é um sensor ToF de alta velocidade e alta resolução. Estes módulos e outros podem ser identificados na Fig. 5.2 e comparados na Tab. 5.2.



Figura 5.2: Sensores LiDAR

Tabela 5.2: Comparação de Sensores de Distância do tipo LiDAR.

Parâmetros	TFmini-Plus <sup>2, 3</sup>	TF02-Pro <sup>3,4</sup>	LiDAR-Lite V3
Alcance (m)	12	40	40
Freq. Amostra. (Hz)	1000	1000	500
Campo Visão (°)	3.6	3	0.45
Imun. Lumi. (Klux)	70	100	—
Exatidão (cm)	±5 (0.1-6 m) ±1% (6-12m)	±5 (0.1-5 m) ±1% (5-40 m)	±2.5 (0-5 m) ±10 (5-40 m)
Resolução (mm)	5	10	10
Alimentação (V)	5	5	5
Cons. Médio (mA)	110	200	135
Nível Lógico (V)	3.3	3.3	3.3
Interface	I2C UART Trigger	I2C UART Trigger	I2C PWM Trigger
Dimensões (mm)	35 × 21 × 19	69 × 36 × 26	20 × 48 × 40
Peso (g)	12	50	22
Custo (€)	45	100	130

<sup>2</sup>Alcance baseado num teste *indoor* com um quadro branco regular (90% de refletividade a 25° como objeto de deteção)

<sup>3</sup>Encapsulamento IP65

<sup>4</sup>Alcance baseado em testes *indoor* com um quadro branco regular (90% e 10% de refletividade a 25° como objeto de deteção)

<sup>5</sup>Alcance baseado num objecto com 70% de refletividade, e tem encapsulamento IPX7

### 5.1.2 Servo Motores Dynamixel

A **ROBOTIS** é uma empresa desenvolvedora de vários sistemas eletromecânicos, tanto na área industrial como na robótica, fabricante da linha Dynamixel. Estes “*All-in-One*” *smart servo motors* são muito mais precisos e robustos do que os tradicionais servo motores de veículos RC, tanto analógicos como digitais. Os Dynamixel distiguem-se por conterem embebidos em si, um microcontrolador, o que proporciona as seguintes vantagens:

- Controlo PID, e de vários motores em simultâneo;
- Controlo em malha-fechada (*Joint Mode*) ou giro infinito com controlo de velocidade (*Wheel Mode*);
- Feedback de posição, velocidade, carga, temperatura e alimentação.

Serão aqui apresentadas aqui três opções de componentes para a aplicação em causa, entre os quais o **AX-12W** (Fig. 5.3(b)), o **MX-12W** (Fig. 5.3(c)) e o Parallax Feedback 360 (Fig. 5.3(a)) que é um servo motor analógico de rotação contínua sendo controlado através de um PWM de 50 Hz. Todas as opções apresentadas são controladas através do protocolo de comunicação UART, tendo apenas uma linha de comunicação que serve tanto para transmitir como para receber informação, de forma intercalada (*Half-Duplex*). Os níveis lógicos utilizados são TTL (5 V). Uma breve comparação é realizada na Tab. 5.3, que demonstra as principais diferenças básicas entre os vários modelos e onde é possível verificar que o MX-12W tem características superiores aos demais em praticamente todos os parâmetros, e permite uma rotação controlada a 360°, enquanto que o AX-12W apresenta uma *deadzone/deadband* de 60° [56]. Uma das desvantagens é que é justificadamente mais caro que os dois outros modelos de comparação mas mesmo assim muito abaixo de todos os outros modelos da linha Dynamixel com características semelhantes ou superiores. Existem “réplicas” *low-cost* de alguns modelos Dynamixel por parte da empresa LewanSoul/HiWonder.

Tabela 5.3: Comparaçāo de Servo Motores Dynamixel.

Parāmetro	AX-12W	MX-12W	MX-28AR
<b>Tensāo de Alim. (V)</b>	9 - 12	10 - 14.8	10 - 14.8
<b>Vel. Māx. (rpm)</b>	470	470	55
<b>Cons. sem carga (A)</b>	0.26	—	0.12
<b>Resoluçāo (º/pulso)</b>	0.2930	0.0879	0.0879
<b>Alcance Angular (º)</b>	300	360	360
<b>Sensor de Pos.</b>	Potenc.	Enc. Mag. Absoluto	Enc. Mag. Absoluto
<b>Baud R. Max. (bps)</b>	1,000,00	4,500,000	4,500,000
<b>Dimensões (mm)</b>	32 × 50 × 40	32 × 50 × 40	36 × 51 × 36
<b>Peso (g)</b>	52.9	54.6	77.0
<b>Custo<sup>6</sup> (\$)</b>	44.90	65.90	249.90



(a) Parallax, adaptado [57] (b) AX-12W, adaptado [56] (c) MX-12W, adaptado [56]

Figura 5.3: Servo Motores

<sup>6</sup>Retirado de: <https://www.sparkfun.com/> e <https://www.robotis.us>

### 5.1.3 Microcontroladores STM32

Os microcontroladores de 32 bits são extremamente poderosos e estão a tornar-se cada vez mais populares à medida que se vão tornando mais acessíveis em comparação com as alternativas de 8 e 16 bits. Dentro da categoria de 32 bits, a arquitetura ARM é uma das arquiteturas mais populares. Algumas das características mais avançadas incluem *Instruction Pipelining*, *Branch Prediction*, *Nested Vectored Interrupts (NVI)*, *Floating Point Units (FPU)* e proteção de memória. Conseguem também suportar Sistemas Operativos em Tempo Real (RTOS), que lhes fornece capacidades de *multi-tasking* [58]. Será então por estes motivos que será escolhida uma arquitetura ARM de 32 bits para esta aplicação.

No que à programação e desenvolvimento diz respeito, a STMicroelectronics fornece variadíssimas ferramentas de desenvolvimento, tais como o CubeIDE, que é capaz de gerar código de inicialização de periféricos sem necessidade estudar intensamente o manual de cada microcontrolador. Esta IDE (*Integrated Development Environment*) é compatível com os programadores e *debuggers* STLinkV2, também fabricados pela STMicroelectronics, o que facilita o processo de depuração de erros no código. Para testes e prototipagem de alguns dos seus microcontroladores, a ST fornece também placas de desenvolvimento NUCLEO e Discovery, que contêm interfaces STLink. Outra grande vantagem, é que microcontroladores pertencentes à mesma família, beneficiam de uma alta compatibilidade pino a pino, o que facilita a troca deste por um mais robusto ou mais económico, por exemplo. Por todos estes motivos referidos anteriormente, a STMicroelectronics será a fabricante de microcontroladores escolhida para potenciar este projeto.

Um microcontrolador viável para a aplicação em causa deve possuir, pelo menos: um barramento de comunicação I2C (Inter-Integrated Circuit), três barramentos de comunicação por USART/UART, capacidade de *debug* via *Serial Wire*, e 3 GPIO's (*General Purpose Input Output*) restantes.

Por fim, o microcontrolador deve estar disponível numa placa de desenvolvimento, ter um custo acessível, estar disponível em quantidades razoáveis e ser relativamente popular, o que facilita a pesquisa em fóruns e comunidades online sobre problemas que possam ocorrer durante o desenvolvimento do projeto [59]. Questões relacionadas com o nível de processamento necessário e quantidade de memória volátil só poderão ser quantificadas durante o desenvolvimento do sistema, e por isso, não serão tomadas em consideração precisa. Na Tab. 5.4 apresenta-se uma ligeira comparação entre três STM32 com características adequadas ao sistema pretendido. A escolha final acabou por recair no **STM32F103RBTx** por já ter sido introduzido pela unidade curricular de SiSem (Sistemas Embbebidos) do Mestrado em Sistemas Autónomos, o que permitirá um mais rápido desenvolvimento do sistema proposto em comparação com as outras opções aqui expostas.

Tabela 5.4: Comparação de Microcontroladores STM32.

Parâmetro	F103RBTx	F446RETx	L476RGTx
<b>Freq. (MHz)</b>	72	180	80
<b>IO (Qt.)</b>	51	50	51
<b>Flash (kBytes)</b>	128	512	1024
<b>RAM (kBytes)</b>	20	128	128
<b>Tipo de Package</b>	LQFP64	LQFP64	LQFP64
<b>Custo<sup>7</sup> (\$)</b>	2.454	3.641	4.316

<sup>7</sup>Retirado da aplicação CubeIDE

### 5.1.4 Esquema Elétrico

#### Circuito de Alimentação e Microcontrolador

Nas discussões acerca da alimentação do circuito, foi considerado o acesso à tensão regulada de 12 V, gerada por componentes externos à placa. No entanto teremos de realizar duas conversões, uma para 5 V e outra para 3.3 V. Existem vários sistemas de conversão de tensão CC (Corrente Contínua), nos quais se destacam os reguladores lineares e os chaveados (*Buck Converters*). Após a análise da Tab. 5.5, foi proposto realizar a conversão da maior queda de tensão por via de um *buck converter* (12 V para 5 V), e a menor queda de tensão por via de um regulador de tensão (5 V para 3.3 V). Desta forma obtém-se um equilíbrio entre a eficiência energética e a quantidade de componentes a utilizar.

Tabela 5.5: Comparação Genérica de Sistemas de Conversão de Tensão CC.

Característica	Lineares	Chaveados
Eficiência	Baixa ( $\approx 65\% - \approx 75\%$ )	Alta ( $>90\%$ )
Nível de ruído	Baixo	Alto
Qt. Componentes Externos	Poucos	Muitos
Custo	Baixo	Alto

De forma a fazer uma escolha correta dos conversores de tensão, terá de ser feita uma análise teórica acerca da corrente máxima que o circuito pode exigir. Caso contrário, o funcionamento do circuito poderá ser severamente afetado. Esta análise também é importante no que diz respeito à correção de possíveis falhas ao nível do *hardware*. O consumo máximo do sistema é apresentado nas Tab. 5.6 e Tab. 5.6. A conclusão mais importante a retirar destas tabelas é de que o regulador de tensão necessita de ser capaz de fornecer no máximo 220 mA de corrente ao circuito e que o *buck converter* necessita de ser capaz de fornecer no máximo 840 mA de corrente ao circuito.

No anexo A, pode ser analisado o esquema completo do circuito.

Tabela 5.6: Consumo de Corrente de Componentes alimentados a 3.3 V

Componente	Consumo Máximo (mA)
STM32F103RB	150
74HC126	70
Outros	$\approx 0$
<b>Total</b>	$\approx 220$ mA

Tabela 5.7: Consumo de Corrente de Componentes alimentados a 5 V

Componente	Consumo Máximo (mA)
TF02-Pro / TFmini-Plus	500
AMS1117-3.3V	220
TJA1051T	70
SN75176AD	70
Outros	$\approx 0$
<b>Total</b>	$\approx 860$ mA

Existem vários parâmetros a ter em consideração em relação a reguladores de tensão. O primeiro é a tensão de *dropout*, que é definido como sendo a tensão mínima entre  $V_{IN}$  e  $V_{OUT}$  para operar corretamente. O segundo é o limite de corrente que é capaz de fornecer. Seguido da corrente mínima de carga, que é o valor mínimo de corrente que o regulador precisa de estar a fornecer para que este opere corretamente. Por último é necessário considerar a temperatura máxima de junção.

Os condensadores de desacoplamento recomendados pelo *datasheet* do regulador de tensão podem não ser suficientes, e é recomendado utilizar condensadores de tântalo devido ao ser baixo valor de ESR e a sua relação capacidade/volume. Existem ruídos inerentes a praticamente todos os sistemas de alimentação, assim como efeitos indutivos nas trilhas de uma PCB, e por isso, é necessário adicionar condensadores de desacoplamento que filtrem esses ruídos, protegendo o circuito de variações de tensão que poderão prejudicar o seu funcionamento.

Fabricado pela **Advanced Monolithic Systems**, o AMS1117-3.3V é um regulador de tensão LDO (*Low Drop-Out*) de baixo-custo. As suas

especificações estão presentes na Tab. 5.8. A partir da Eq. (5.5) é possível deduzir que o regulador está a trabalhar na sua zona normal de operação.

Tabela 5.8: Especificações do AMS1117-3.3V.

Parâmetro	Valor
Tensão de Dropout $V_{DO}$	1.1 V
Corrente Máxima $I_{MAX}$	1100 mA
Corrente Mínima $I_{MIN}$	5 mA
Resistência Térmica Total (junção para o ambiente) $R_T$	45 °C/W
Temperatura Máxima de Junção $T_J$	125 °C

A queda de tensão ( $V_D$ ) no AMS1117 é dada por:

$$V_D = V_{IN} - V_{OUT}, \quad V_D > V_{DO}. \quad (5.1)$$

A corrente de saída ( $I_{OUT}$ ) do AMS1117 satisfaz a condição:

$$I_{MIN} < I_{OUT} < I_{MAX}. \quad (5.2)$$

A potência de dissipação ( $P_D$ ) do AMS1117 pode ser calculada por:

$$P_D = (V_D)(I_{OUT}). \quad (5.3)$$

A temperatura de junção ( $T_J$ ) dada a temperatura ambiente ( $T_A$ ) é:

$$T_J = T_A + P_D R_T. \quad (5.4)$$

Para o AMS1117-3.3V, resultam os seguintes valores:

$$\left\{ \begin{array}{l} V_D = 5 - 3.3 = 1.7 \text{ V}, \quad 1.7 > 1.1 \\ I_{OUT} = 0.220 \text{ A}, \quad 0.005 < 0.220 < 1.100 \\ P_D = (1.7)(0.220) \approx 0.374 \text{ W} \\ T_J = 25 + (0.374)(45) \approx 41.83 \text{ °C}, \quad 41.83 < 125 \end{array} \right. . \quad (5.5)$$

Por cada pino  $V_{DD}$  do micro-controlador foi colocado um condensador cerâmico de  $100\text{ nF}$ . Em caso de consumo elevado dos periféricos, pode ser necessário adicionar um condensador de tântalo de  $10\text{ }\mu\text{F}$  adicional. Para ruídos em alta frequência pode ser necessário adicionar mais um condensador de  $10\text{ nF}$  em cada desacoplamento. Estes condensadores devem [60]:

- Ser colocados o mais próximo possível dos pinos de alimentação para que tenham o maior efeito possível,
- Ser do tipo cerâmico e de menor tamanho possível, pois estes têm geralmente um menor valor de ESR (*Equivalent Series Resistor*), apesar de serem geralmente mais caros, seguindo o valor de referência igual ou menor a  $0.1\text{ }\Omega$  (ex: Condensadores da família X7R),

Os pinos  $V_{DDA}$  poderão necessitar de um maior cuidado caso se esteja a utilizar periféricos com propriedades analógicas tais como o ADC (*Analog-to-Digital Converter*) e o DAC (*Digital-to-Analog Converter*). Em circuitos com de alimentação estável, tal como uma bateria, ou em que não se utilize os periféricos analógicos do microcontrolador ou que a precisão dos mesmos não seja um requisito, pode-se conectar os pinos  $V_{DDA}$  aos  $V_{DD}$ . Caso contrário, é recorrente conectar uma bobina indutora entre os pinos  $V_{DD}$  e  $V_{DDA}$ , para as diferenças de corrente do resto do circuito não afetarem a parte analógica. A bobina indutora tem o papel de filtrar ruídos e grandes diferenças de corrente, e o seu valor depende das propriedades do circuito. Estas bobinas devem ter baixa impedância em nível DC e alta impedância em frequências elevadas. Se for necessária uma alta precisão para uma ampla faixa de temperaturas, é recomendável utilizar um CI (Circuito Integrado) para referência de tensão (ex: MAX610x). O resultado final do circuito de alimentação pode ser analisado visualmente nas Fig. 5.4 e Fig. 5.5.

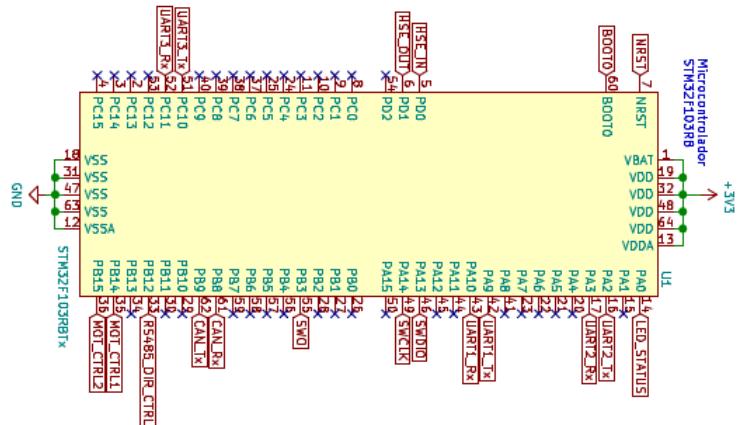


Figura 5.4: Microcontrolador STM32F103RB.

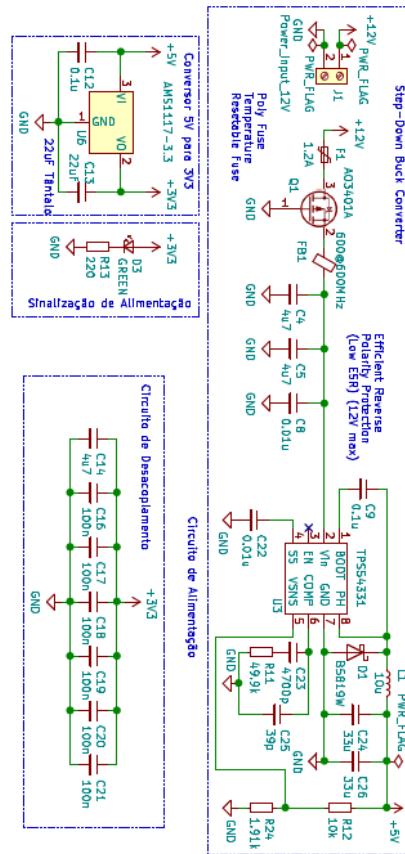


Figura 5.5: Circuito de Alimentação.

## Círculo Oscilador de Relógio

Existem várias possibilidades para fonte de clock do microcontrolador:

- Oscilador RC Interno
- Osciladores Pierce (Cristal de Quartzo + Condensadores)
- Osciladores CMOS ou MEMS (Micro Electro Mechanical Systems)  
(ex: MAX 7375, que apenas utiliza um pino do microcontrolador, em modo bypass)

Iremos optar pelos osciladores Pierce devido à precisão necessária fazer uso dos periféricos de comunicação do microcontrolador e ao seu baixo custo.

Foram considerados os seguintes fatores na escolha de um cristal [61]:

- Estabilidade, com referência em valores iguais ou menores a 8 ppm,
- Faixa de temperatura de operação e estabilidade nessa faixa,
- Ganho mínimo.

Durante o desenvolvimento do *layout* PCB foram tidos vários cuidados a ter em consideração: O circuito de *clock* foi colocado próximo do microcontrolador, com trilhas curtas e simétricas, foi feito um ajuste adequado dos valores dos condensadores assim como uma disposição dos mesmos em forma simétrica, e não se permitiu a passagem de sinais de alta frequência próximos das trilhas de *clock*, assim como o aterramento dedicado e próximo do relógio. Uma resistência de *feed* nem sempre é necessária, mas é uma boa prática. O que esta resistência faz é limitar o nível de condução do STM32 que irá alimentar cristal oscilador. Se o nível de condução for demasiado alto, irá eventualmente sobre-conduzir o cristal, o que significa que irá gerar harmónicos adicionais e efetivamente ruído e distorção [61].

## Boot, Debug e Reset

Dependendo da configuração dos pinos *BOOT0* e *BOOT1*, o microcontrolador pode executar código de regiões de memória diferentes [62] [63]. Nesta aplicação apenas o pino de *BOOT0* será relevante:

- *BOOT0 LOW*: leitura e execução normal do código da memória,
- *BOOT0 HIGH*: permite o *upload* de um novo programa para a memória.

Para realizar o *debug* durante o desenvolvimento de *software* do projeto, foi adicionado um conector *standard* para fins de programação e depuração, tais como SWD (*Serial Wire Debug*) ou JTAG (*Joint Test Action Group*). Desta forma, pode-se conectar através de um STLink, ou utilizando uma placa de desenvolvimento de microcontroladores STM32 (ex: Discovery, Nucleo), que já contém um STLink embutido e este já inclui as resistências de *pull-up* necessárias. Foi utilizada a opção *Trace Asynchronous SW* que permite realizar operações de *printf()* pelo *debugger*, através do pino SWO.

De forma a evitar falsas interrupções por *reset*, foi adicionado um condensador de  $100\text{ nF}$  ao pino de *reset*, em paralelo com o botão de acionamento dessa interrupção. Este circuitos podem ser visualizados na Fig. 5.6.

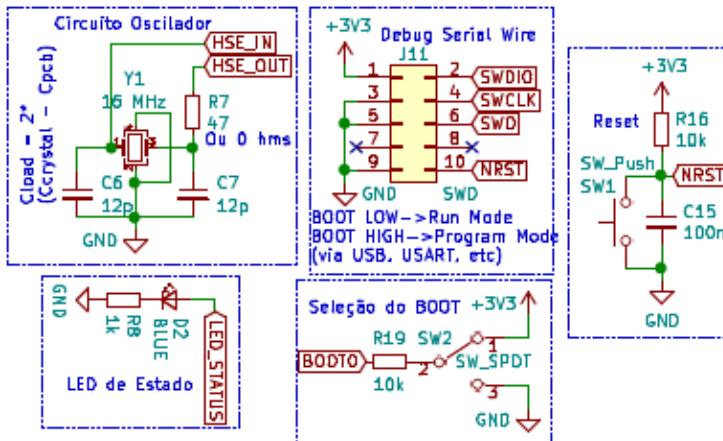


Figura 5.6: Circuito Oscilador, Boot, SWD e Reset.

### Círcuito Dynamixel

Tal como referido anteriormente, estes servo-motores são compatíveis com o protocolos de comunicação UART e RS-485, protocolos robustos e rápidos. Acontece que, em UART, a comunicação com o servo-motor apenas é possível em *half-duplex*, ou seja, apenas se pode transmitir ou receber informação de cada vez. De forma a efetuar o controlo da direção de comunicação, a ROBOTIS recomenda o uso de um *buffer* de transmissão, o CI 74HC126 (Fig. 5.7). O seu circuito sugere que se utilize também uma porta lógica *not*, conectada ao pino de direção. Ao invés de ocupar mais espaço da PCB, consumir mais energia, e complicar o circuito, irá se utilizar dois pinos do microcontrolador para realizar este controlo, realizando a operação de porta lógica *not* via código. Em comunicação RS-485, foi utilizado o *transceiver* SN75176AD, que realiza a tradução dos níveis TTL para os padrões deste *standard* de comunicação. Como o STM32 e o Dynamixel utilizam nível lógicos diferentes, foi utilizado um conversor de nível lógico, apesar de o Dynamixel ser compatível com 3.3 V (TTL) e o pino de UART utilizado ser tolerante a 5 V. A conexão é feita através dos conectores MOLEX 50-37-5033 que se inserem em conectores MOLEX 22-03-5035 na PCB.

### Círcuito TF02-Pro e TFMini-Plus

Tal como referido anteriormente, estes sensores são compatíveis com os protocolos de comunicação UART e I2C. Em ambos os sensores iremos utilizar o protocolo UART por ser um protocolo robusto e rápido.

### Comunicação CAN

Para a comunicação CAN, foi utilizado o *transceiver* TJA1051T, que realiza a tradução do microcontrolador para os standards elétricos deste protocolo. Foi adicionado um *jumper* de seleção da resistência de terminação para os dois extremos de comunicação, tal como recomendado pelo protocolo.

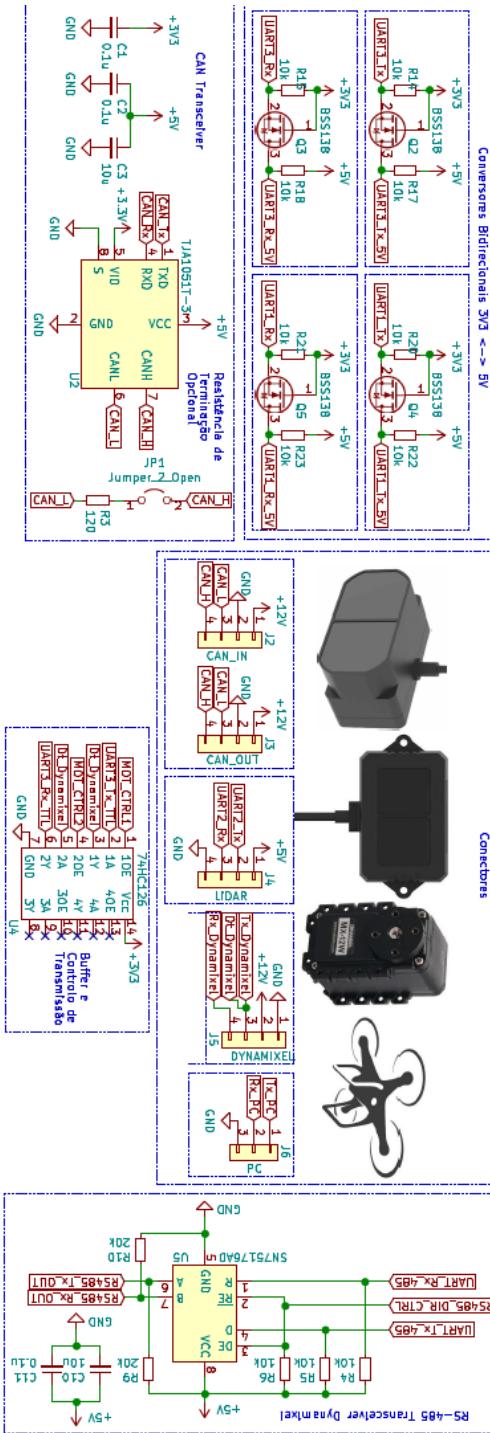


Figura 5.7: Controlo do Motor, Leitura dos Sensores e Comunicação CAN.

### 5.1.5 Placa de Circuito Impresso

Uma solução comum para uma estrutura elétronica organizada, com conexões sólidas é a criação de uma PCB. Outras vantagens incluem a melhoria da estética geral, assim como o fácil transporte e melhor implementação para uso real (menos susceptível ao movimento e contacto por parte de outrem).

Existe uma variedade de *software* disponível para desenho e criação de PCBs, entre as quais o KiCad, o Altium e o Eagle (agora fundido com o Fusion 360). O escolhido para este projeto foi o KiCad por ser multiplataforma, totalmente gratuito e *open-source* [64].

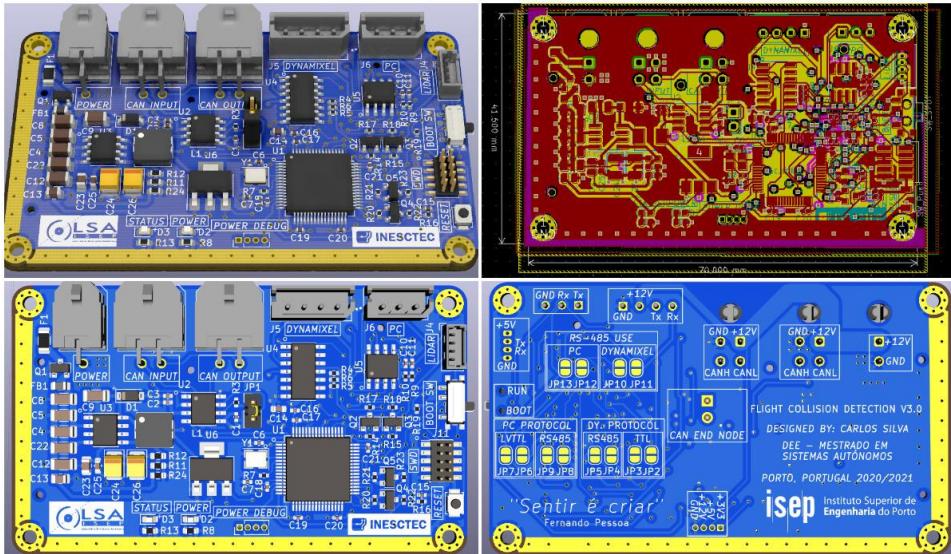


Figura 5.8: Design Final da PCB.

## 5.2 Análise Financeira

O orçamento simplificado de todo o projeto é apresentado nas Tab. 5.9 e Tab. 5.10, e de forma mais extensiva na Tab. 5.11. No entanto, nenhum destes inclui custos com o transporte, quantidades mínimas exigidas pelo fornecedor, custos relacionados com a impressão 3D e custos relacionados com a solda de componentes THT na placa. Na Fig. 5.9 é possível avaliar o custo relativo e absoluto de cada componente de fabricação da PCB, e o mesmo se aplica na Fig. 5.10 em relação ao custo total do projeto. Ambas estas figuras estão agrupadas por setores: o setor “TFmini-Plus” contabiliza a totalidade dos sensores TFmini-Plus utilizados, o setor “TF02-Pro” contabiliza o custo do sensor TF02-Pro, o setor “Dynamixel” contabiliza o custo do servo-motor Dynamixel MX-12W e o setor ”PCB” contabiliza o custo da PCB e de todos os seus componentes agrupados.

Tabela 5.9: Orçamento Simplificado da Fabricação da PCB

Descrição		Custo (€)
PCB		2.04
Assemblagem SMT	Componentes	15.82
	Taxa Componentes	9.88
	Setup	2.88
	Stencil	0.62
	Assemblagem	1.02
Componentes não SMT		7.69
<b>Total = 39.95 €</b>		

Tabela 5.10: Orçamento Simplificado dos Componentes Externos à PCB

Descrição		Custo (€)
Dynamixel	MX-12W	55.58
	MX-28AR	210.76
LiDAR	TFmini-Plus	60.87
	TFmini-Pro	85.81

Tabela 5.11: Orçamento Extensivo do Custo Total do Sistema

Componente	Fornecedor	Custo (Uni.) €	Qt.	Total €	Comentário	Link
Dynamixel MX-12W	Robotis	54.09	1	54.09		Robotis.us
TF02-Pro	Amazon	73.86	1	73.86		Amazon.com
TFmini-Plus	Sparkfun	36.89	3	110.67		Sparkfun.com
AMSI117		0.06	1	0.06	SOT-23	LCSC.com
Bobina 39 nH		0.01	1	0.01	0402	LCSC.com
Cl 74HC126		0.15	1	0.15	TSSOP-14	LCSC.com
Condensador 12 pF		0.01	2	0.01	0402	LCSC.com
Condensador 0.1 uF		0.02	1	0.02	1206	LCSC.com
Condensador 1 uF		0.01	1	0.01	0402	LCSC.com
Condensador 4.7 uF		0.01	1	0.01	0603	LCSC.com
Condensador 10 uF		0.01	1	0.01	1206	LCSC.com
Condensador 22 uF		0.01	1	0.01	1206	LCSC.com
Condensador 100 nF		0.01	6	0.06	0402	LCSC.com
Cristal 16 MHz		0.10	1	0.10	3225	LCSC.com
LED Verde	LCSC	0.02	2	0.04	0603	LCSC.com
Molex 22035035		0.04	1	0.04		LCSC.com
Molex 530470410		0.07	3	0.21		LCSC.com
MOSFET BSS138		0.03	1	0.03	SOT-23	LCSC.com
Push Button SMD		0.46	1	0.46		LCSC.com
Pin Header 01x04		0.02	1	0.02	2.54 mm	LCSC.com
Pin Header 02x05		0.73	1	0.73	1.27 mm	LCSC.com
Resistência 47 Ohms		0.01	1	0.01	0402	LCSC.com
Resistência 10 kOhms		0.01	2	0.02	0402	LCSC.com
Screw Terminal		0.09	2	0.18		LCSC.com
STM32F103RBT6		3.18	1	3.18	Só na JLC	LCSC.com
Switch SPDT PCMI12		0.57	1	0.57		LCSC.com
PCB	JLCPCB	2.54	1	2.54	Lead-Free	JLCPCB.com
<b>Custo Total: 244.62 €</b>						

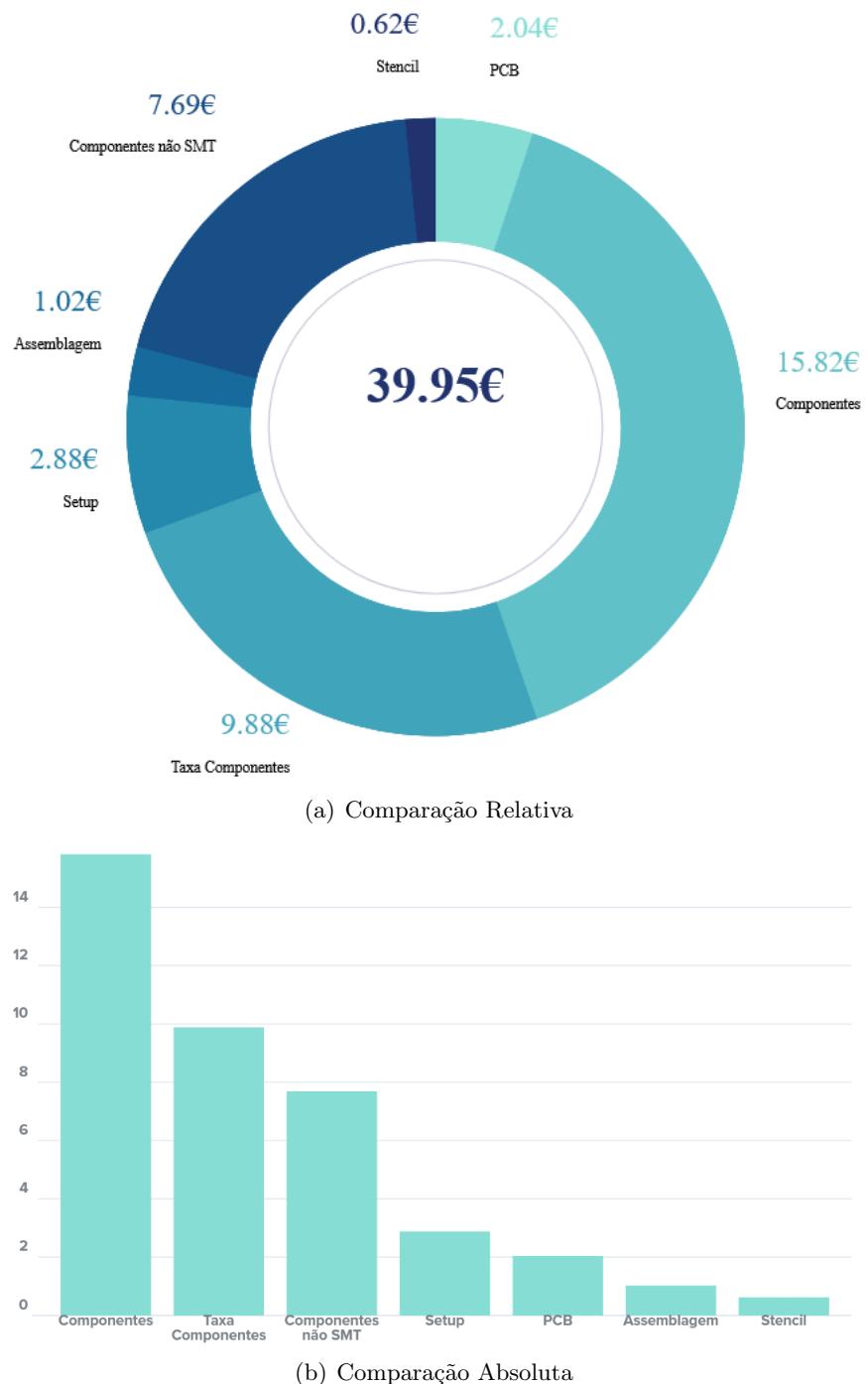


Figura 5.9: Comparação de Custos Relacionados com a PCB

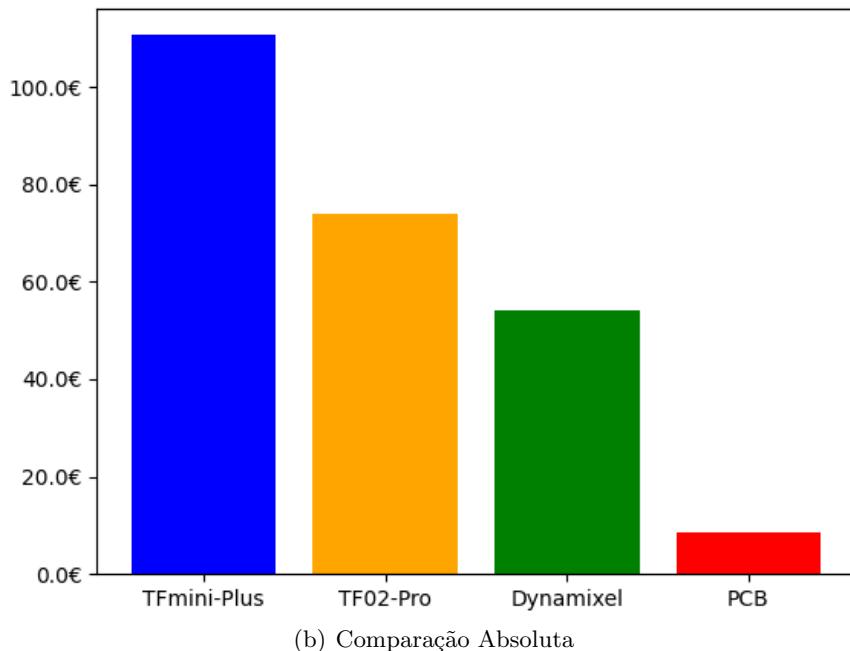
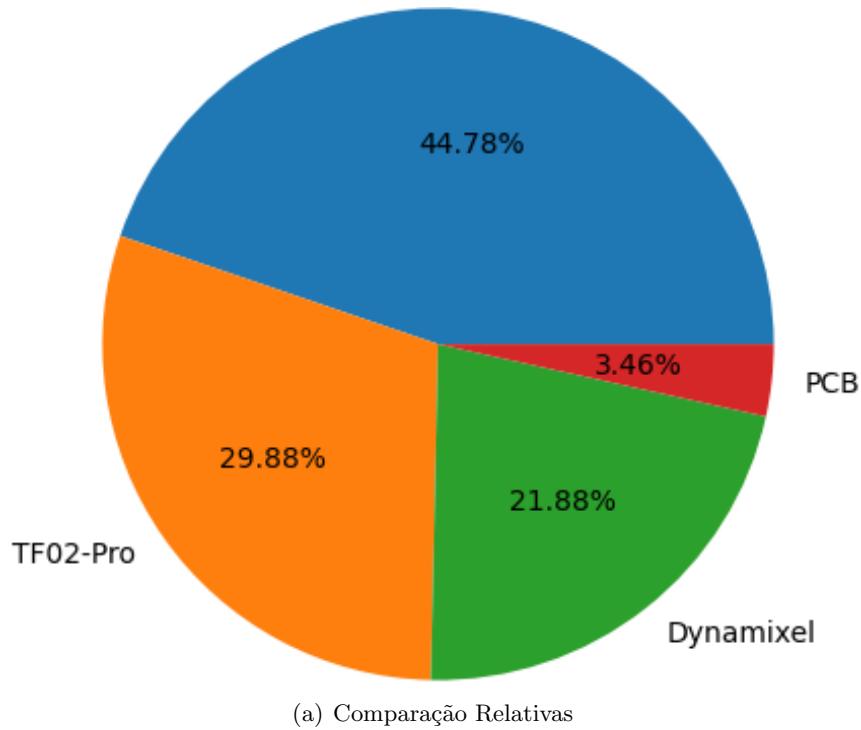


Figura 5.10: Comparação de Custos Totais

### 5.3 Descrição do *Software*

A ferramenta de trabalho principal para a programação do microcontrolador do módulo será o freeRTOS, um sistema operativo em tempo-real e *open-source* para microcontroladores. Este permite a realização de tarefas de forma aparentemente simultânea. A análise será feita para o módulo frontal que realiza a leitura de um sensor LiDAR e o controlo de um servo-motor dynamixel, e para um módulo que realiza a comunicação com o PC, que será o *master*, o resto dos módulos apenas realizam a leitura do sensor LiDAR e o envio dos dados para a placa *master*.

No 1º caso da Fig. 5.11 temos primeiramente a inicialização dos periféricos e do próprio microcontrolador, entre os quais, os portos de comunicação UART para leitura do sensor LiDAR e controlo do servo motor Dynamixel. É também realizada a inicialização do periférico de comunicação CAN do próprio microcontrolador e a definição de um GPIO como *output* para ativar um LED. Posteriormente temos 4 rotinas, contínuas, a correr de forma aparentemente simultânea. A primeira é responsável pela leitura do sensor, a segunda pela movimentação do servo motor, a terceira pela comunicação via CAN dos dados sensoriais para o módulo *master*, a última serve como *debug* de situações de *starvation*, e é responsável por piscar um LED.

No 2º e último caso, é feita tal como anteriormente, a inicialização do microcontrolador, sendo que neste caso os portos UART servirão para a leitura do sensor e para comunicação com o computador de bordo. Existem também 4 rotinas contínuas, sendo que uma delas é semelhante à do caso anterior, responsável por piscar um LED para efeitos de *debug*. Após a inicialização, o módulo espera pela inicialização dos outros módulos, e após o reconhecimento de todos irá começar a realizar a leitura do sensor conectado a si e a fazer a recolha dos dados dos outros módulos. Em cada iteração de leitura dos sensores, é efetuado o envio dessa mesma informação para o computador de bordo, que e utilizará algoritmos de desvio de obstáculos.

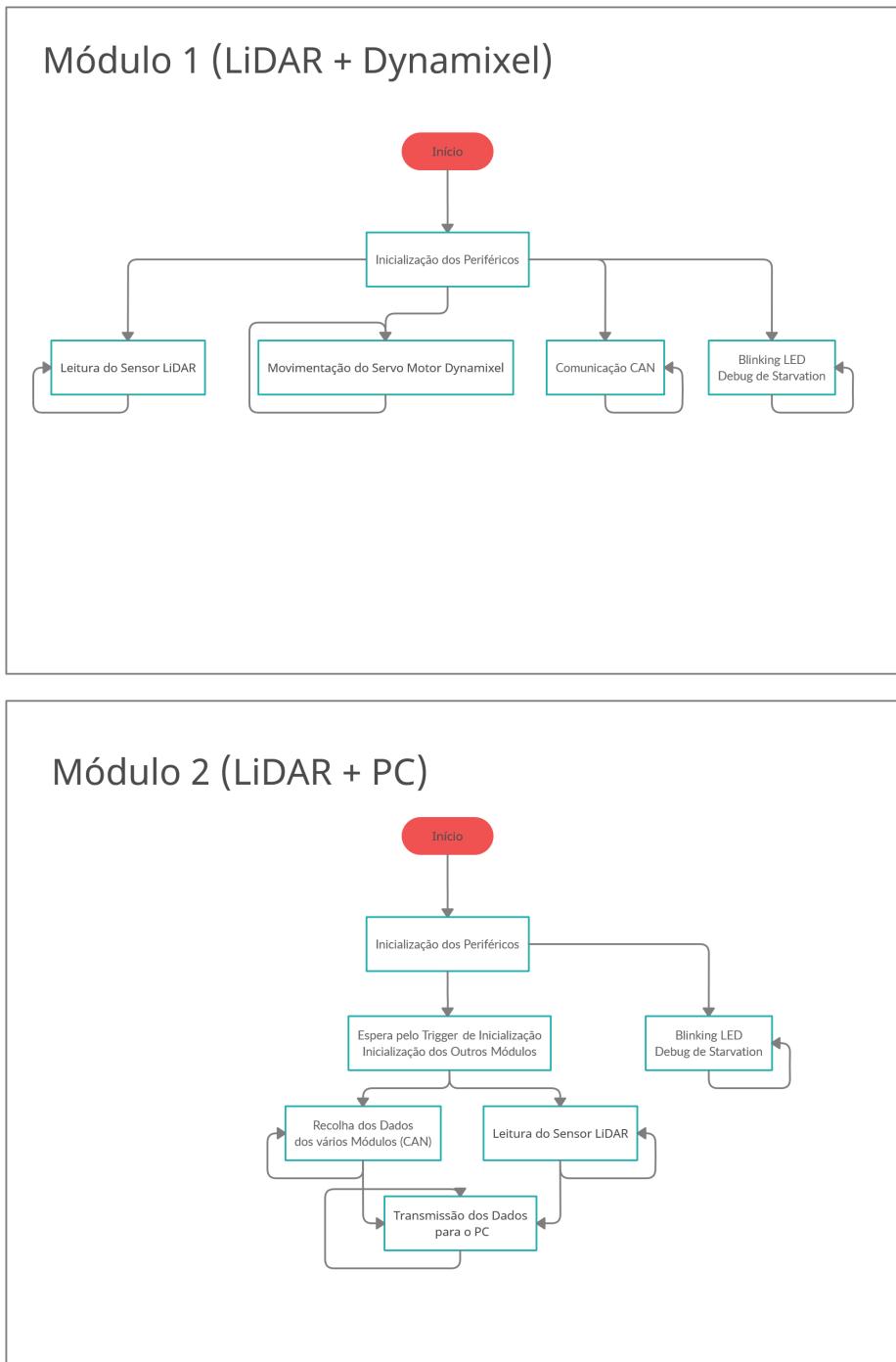


Figura 5.11: Arquitectura de Software dos Módulos.

Foi proposto o desenvolvimento de bibliotecas em linguagem C, para o STM32, de forma a facilitar a organização do processo, e proporcionar um maior nível de abstração em relação aos sensores LiDAR e ao motor Dynamixel. A leitura de dados dos sensores LiDAR, ao nível de software e interface, é realizada exatamente da mesma forma, e por isso a biblioteca é válida para ambos os sensores. Todas as funções permitem a escolha da porta UART a ser utilizada, no entanto, esta deve ser previamente inicializada para tal.

A 1<sup>a</sup> função é a mais importante e mais utilizada, que realiza a leitura dos dados provenientes do sensor, tanto de distância como de força de sinal, e temperatura do mesmo. Isto porque em todos os *frames* enviados pelo sensor, são recebidos estes parâmetros.

A 2<sup>a</sup> realiza um *reset* aos registos do sensor, enquanto que a 3<sup>a</sup> envia um sinal de pedido de dados (caso esteja no modo síncrono). A 4<sup>a</sup> função será também utilizada durante a inicialização do sensor, de forma a que o sensor retorna a distância em *mm* ao invés de *cm*.

As últimas três funções definem o bloqueio da saída de dados, a aplicação do *reset* de fábrica, e o armazenamento permanente das definições nos registos do sensor.

Listing 5.1: lidar.h

```

1 void lidar_readFrame(UART_HandleTypeDef* huart, unsigned short*  
    distData, unsigned short* streData, unsigned short* tempData);  
2 void lidar_sysReset(UART_HandleTypeDef* huart);  
3 void lidar_sendTrigger(UART_HandleTypeDef* huart, unsigned short*  
    distData, unsigned short* streData, unsigned short* tempData);  
4 void lidar_outFormat(UART_HandleTypeDef* huart, unsigned char value);  
5 void lidar_enOutput(UART_HandleTypeDef* huart, unsigned char value);  
6 void lidar_setFactory(UART_HandleTypeDef* huart);  
7 void lidar_saveSettings(UART_HandleTypeDef* huart);
```

A biblioteca do servo-motor é bastante extensa devido à enorme possibilidade de configurações do mesmo. A principal função utiliza na escrita de

valores para o motor foi a 6<sup>a</sup>, que define a orientação do eixo, e que depende da 1<sup>a</sup> função que realiza o envio dos parâmetros a um nível mais baixo de abstração. Na leitura, a mais utilizada foi a última função que indica se o motor se está a mover ou não, sendo que esta depende da 10<sup>a</sup> função.

Listing 5.2: dynamixel.h

```

1 // Escrita
2 void dyna_sendFrame(UART_HandleTypeDef* huart, unsigned char ID,
3                      unsigned char instruct, unsigned char params[], unsigned char
4                      paramLen);
5 void dyna_torqueEnable(UART_HandleTypeDef* huart, unsigned char ID,
6                        unsigned char value);
7 void dyna_led(UART_HandleTypeDef* huart, unsigned char ID, unsigned
8                char value);
9 void dyna_setPID(UART_HandleTypeDef* huart, unsigned char ID, unsigned
10                  char valueP, unsigned char valueI, unsigned char valueD);
11 void dyna_moveTo(UART_HandleTypeDef* huart, unsigned char ID, unsigned
12                   short value);
13 void dyna_moveSpeed(UART_HandleTypeDef* huart, unsigned char ID,
14                      unsigned short value);
15 void dyna_torqueLimit(UART_HandleTypeDef* huart, unsigned char ID,
16                      unsigned short value);
17 // Leitura
18 unsigned char dyna_readFrame(UART_HandleTypeDef* huart, unsigned char
19                               paramLen);
20 unsigned short dyna_readDoubleFrame(UART_HandleTypeDef* huart);
21 unsigned short dyna_presentPos(UART_HandleTypeDef* huart, unsigned
22                               char ID);
23 unsigned short dyna_presentSpeed(UART_HandleTypeDef* huart, unsigned
24                               char ID);
25 unsigned short dyna_presentLoad(UART_HandleTypeDef* huart, unsigned
26                               char ID);
27 unsigned char dyna_presentVolt(UART_HandleTypeDef* huart, unsigned
28                               char ID);
29 unsigned char dyna_readTemp(UART_HandleTypeDef* huart, unsigned char
30                               ID);
31 unsigned char dyna_registered(UART_HandleTypeDef* huart, unsigned
32                               char ID);
33 unsigned char dyna_moving(UART_HandleTypeDef* huart, unsigned char ID
34 );

```

Esta página foi intencionalmente deixada em branco.

# Capítulo 6

## Implementação

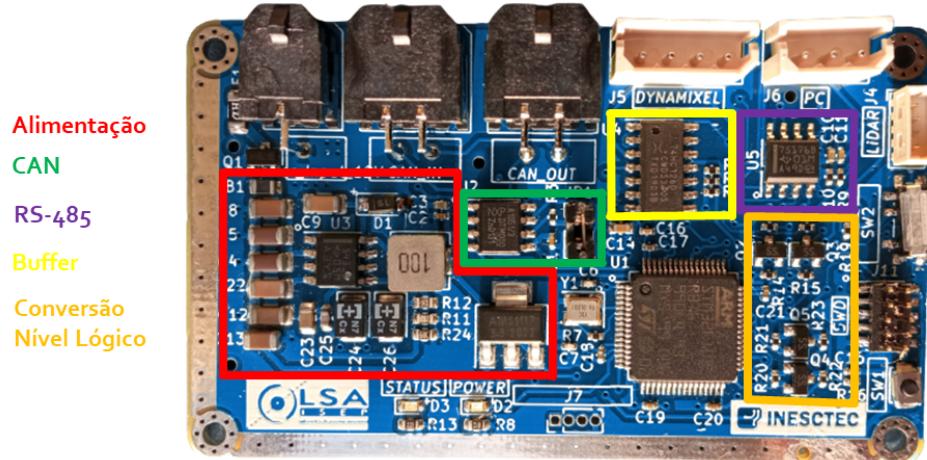
### 6.1 Desenvolvimento do Hardware

#### 6.1.1 Placa de Circuito Impresso

De forma a tornar o circuito robusto, mecânicamente sólido, e aplicável num UAV, foi desenvolvida uma PCB com todos os componentes necessário, ao invés do uso de placas genérica de desenvolvimento e várias *breakout boards*.

Na Fig. 6.1, podemos detetar os vários sub-circuitos que constituem a placa. Dentro da área a vermelho, podemos encontrar o circuito de conversão DC/DC. A verde encontra-se o circuito integrado responsável pela comunicação CAN, assim como o *jumper* que é utilizado nos módulos mais distantes um do outro. Na zona a violeta encontramos o *chip* responsável pela comunicação RS-485, a amarelo o circuito de *buffer* TTL e finalmente, a laranja o circuito de conversão lógica bi-direcional.

O maior CI (Circuito Integrado) é o microcontrolador e ao seu lado o cristal oscilador. Na zonas superior e lateral direita encontram-se os conectores, o botão de *reset* e o seletor de *boot*. As bordas inferiores e lateral esquerda, têm a zona de cobre exposta conectada ao GND, de forma a facilitar a conexão ao GND de um instrumento de medição, como por exemplo um osciloscópio, através de uma garra do tipo crocodilo.



(a) Face Superior da PCB



(b) Face Inferior da PCB

Figura 6.1: Placa de Circuito Impresso Produzida.

Tal como indicado na Fig. 6.2, a PCB foi dividida em 4 *layers*, sendo as camada superior e inferior dedicadas principalmente às trilhas de sinal, sendo revestidas por uma zona de GND, de forma a não receber nem transmitir interferências eletromagnéticas a outros dispositivos do *drone*. As camadas interiores foram dedicadas à alimentação dos componentes. A camada interior superior ficou dedicada apenas ao GND, enquanto que a inferior ficou dedicada às tensões positivas de alimentação, +5 V e +3.3 V.

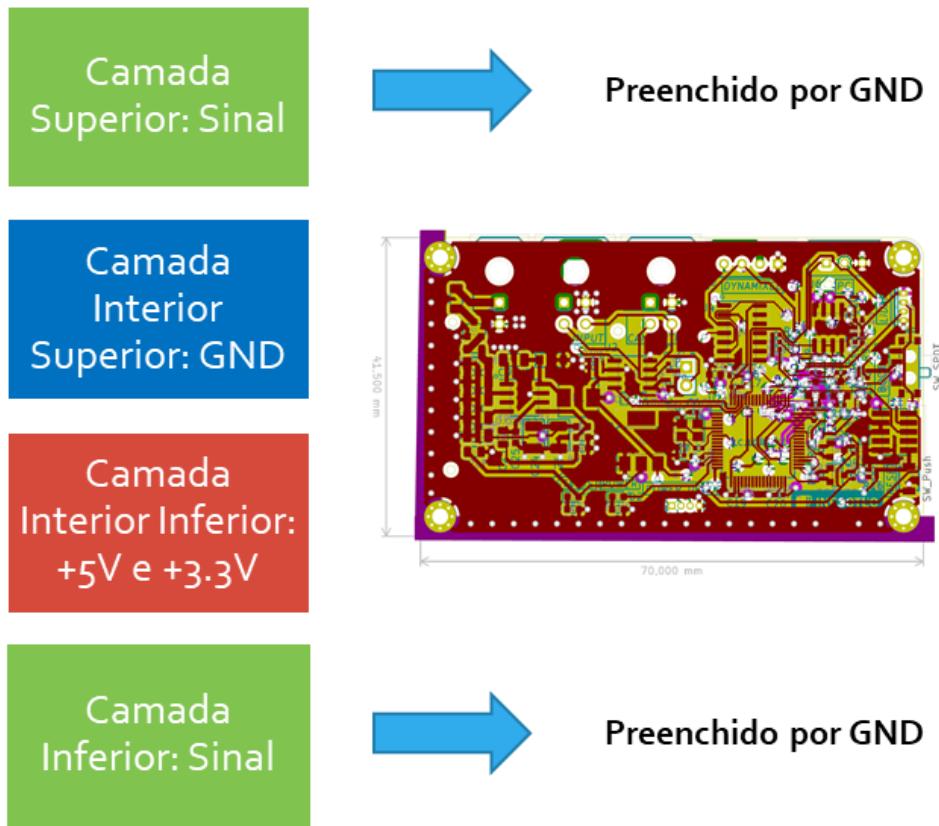


Figura 6.2: Layers da Placa de Circuito Impresso.

A Fig. 6.3 mostra o aspeto da PCB com diferentes cores de *soldermask*.



Figura 6.3: Placa de Circuito Impresso noutras Cores.

### 6.1.2 Estrutura Mecânica

Antes de colocar o sistema e o testar num veículo autónomo, surgiu a necessidade de o testar numa base fixa, ao mesmo tempo que se realizava a programação do sistema e a viabilidade do mesmo. Para isso foi desenvolvido um suporte para o servo-motor, outro para a PCB e um último para o encaixe dos sensores LiDAR no Dynamixel. Estes suportes foram adicionados a uma pequena base comum, que compacta todo o sistema.

Para testagem foram apenas utilizados dois módulos. Um módulo representativo da zona frontal do veículo, conectado ao servo-motor que contém também um sensor LiDAR acoplado, e um outro módulo cuja função será de *master* do sistema e comunicar a informação ao computador de bordo, que neste caso será um computador regular.



Figura 6.4: Construções Mecânicas do Protótipo.

### 6.1.3 Protótipo Final

Para iniciar o processo de testagem os componentes aparafulados à base do protótipo, tal como visualizado na Fig. 6.6. A alimentação proveniente das baterias foi simulada a partir de uma fonte regulada para 12 V.

Inicialmente esta fonte foi limitada a um nível de corrente mínimo de forma a detetar algum possível curto-circuito de forma rápida sem danificar algum componente. Não foi detetado nenhum curto-circuito, no entanto o LED indicador de energia não estava aceso e a temperatura do *buck-converter* estava a aumentar. Foi então descoberto que o diodo do circuito do *buck-converter* tinha sido soldado na orientação reversa. Solucionado esse problema, a placa demonstrou estar em condições de ser programada.

Para o desenvolvimento do projeto foi essencial o uso do programador/*debugger* SWD STLINK-V2, um conversor USB/Serial FTDI para comunicação com o computador (que não contém portas UART, apenas USB) e um analisador lógico para estudo dos protocolos de comunicação utilizados.

Estes componentes estão presentes na Fig. 6.5.

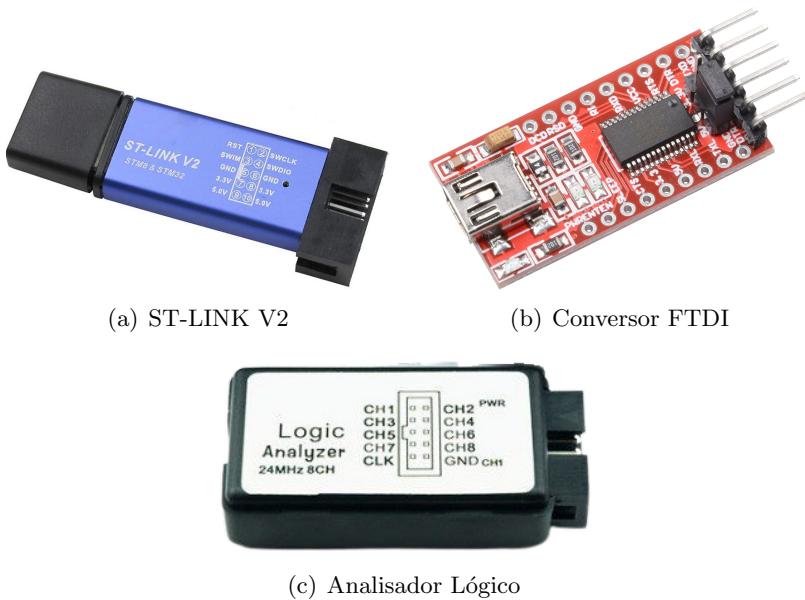
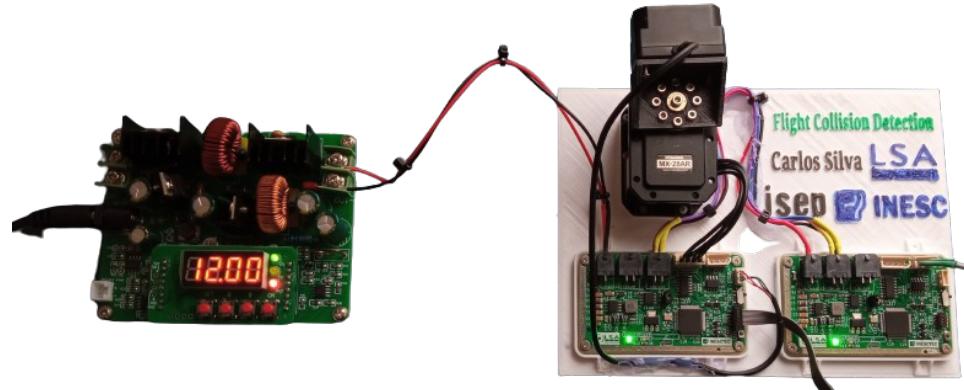
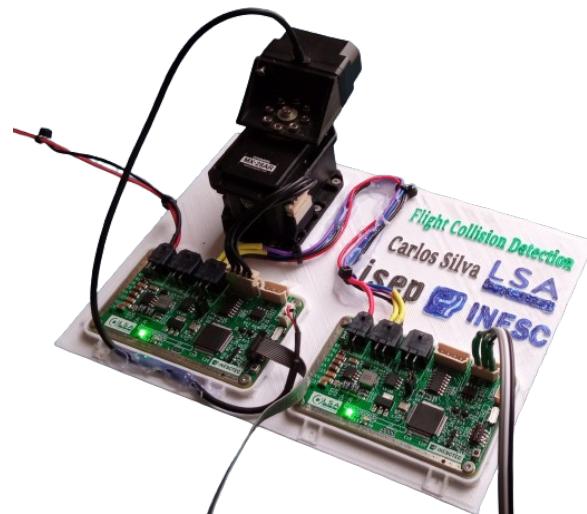


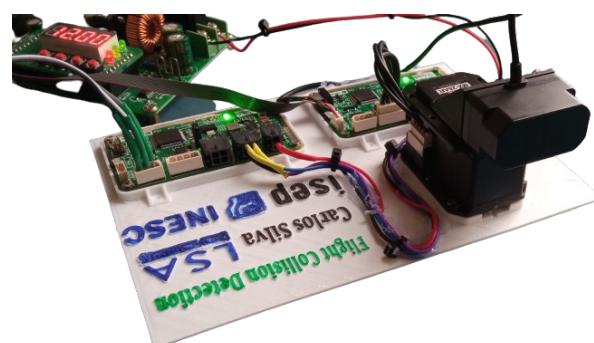
Figura 6.5: Ferramentas de Hardware Utilizadas.



(a) Perspectiva Superior



(b) Perspectiva Traseira



(c) Perspectiva Frontal

Figura 6.6: Protótipo Final.

## 6.2 Desenvolvimento do Software

### 6.2.1 Programação do Microcontrolador

Antes de iniciar a programação do microcontrolador foi alimentado o sensor LiDAR TFmini-Plus. Aqui foram testados os *frames* enviados por este sensor utilizando um analisador lógico. A Fig. 6.7 é uma captura desse processo, no qual são claramente identificáveis os *frames* referenciados no *datasheet* do componente, como por exemplo os dois *header frames* (89 e 89).

Um processo semelhante foi utilizado com o servo-motor, sendo que neste caso, a empresa fabricante também fornece um software de controlo do mesmo, onde também é possível ter acesso e controlar, a baixo nível, os registos do microcontrolador do motor, para além de visualizar os comandos recebidos e enviados. Esta ferramenta chama-se “Dynamixel Wizard 2.0” e foi fundamental para a compreensão do funcionamento do mesmo, e *debug* da biblioteca desenvolvida, dado que as informações no *website* do fabricante estavam erradas. Concluiu-se então que o *ID* = 1, e que estava a trabalhar com o protocolo 1.0 do fabricante, com um *baud rate* de 57600 *bps*. Uma captura de ecrã do uso desta ferramenta corresponde à Fig. 6.8.

O IDE de programação utilizado foi o STM32 CubeIDE (Fig. 6.9) pela enorme quantidade de ferramentas facilitadoras da programação tais como o CubeMX. Contém também já instalado, a biblioteca HAL e uma distribuição do freeRTOS. Foram desenvolvidas e adicionadas ao projetos as bibliotecas dos sensores e atuador, e o desenvolvimento geral do projeto ao ficheiro *main.c*. A compilação foi concluída com 0 erros e 0 *warnings*.

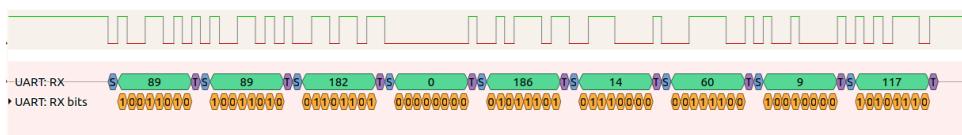


Figura 6.7: Análise de Baixo-Nível dos Dados do TFmini-Plus.

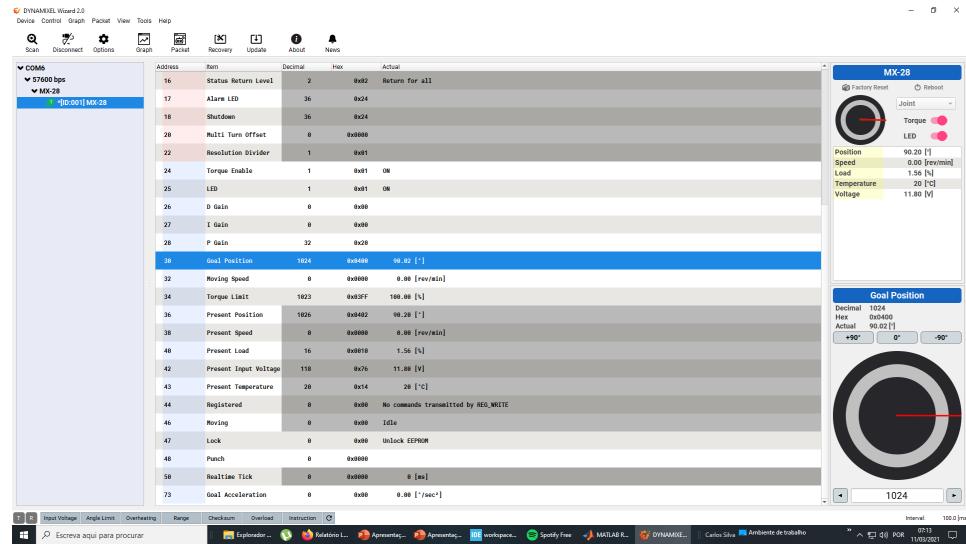


Figura 6.8: Software Dynamixel Wizard 2.0.

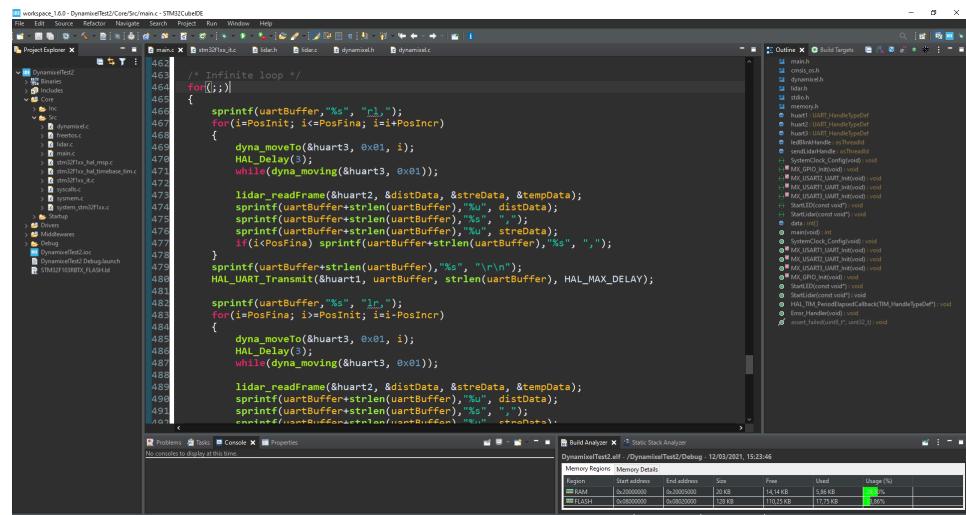


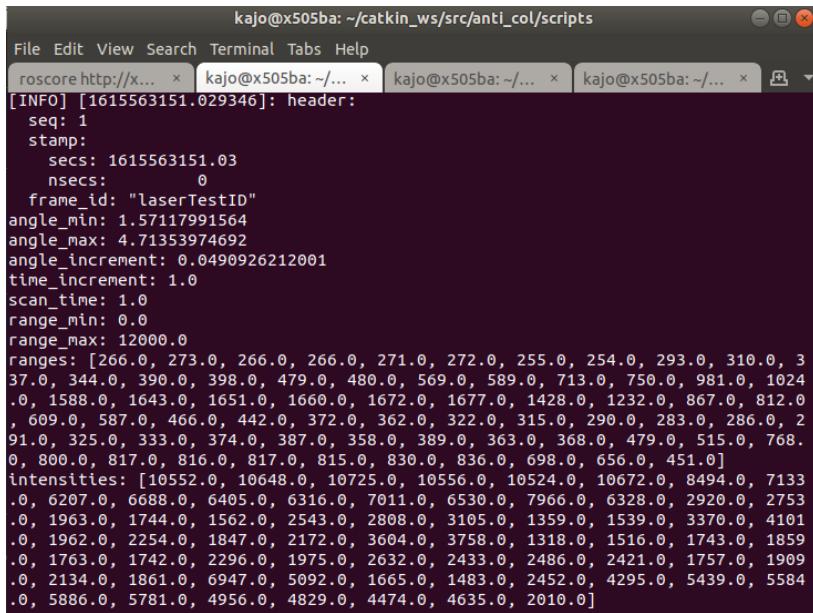
Figura 6.9: Software STM32 CubeIDE.

### 6.2.2 Integração com o ROS

Como os drones presentes no laboratório utilizam o ROS (Robot Operating System), torna-se necessário realizar a conversão dos dados do microcontrolador para esta *framework*. Para tal, foi criado um *package* ou nó, que lê os dados da porta série e os converte no formato de mensagem “**sensor\_msgs/LaserScan.msg**”, para que possam ser utilizados por outros nós. Este formato de mensagem é mais comumente utilizado para armazenar e transmitir dados de sensores LiDAR.

Para correr o código basta utilizar um comando semelhante ao seguinte, em que o primeiro argumento informa o ângulo inicial do *scan*, o segundo informa o ângulo final e o último define a resolução. Todos estes valores estão numa escala de 0 a 4098, em que 0 equivale a 0º e 4098 a 360º. Para a visualização dos dados basta aceder ao tópico da mensagem (Fig. 6.10).

```
$ rosrun anti_col lidar_scan.py 1024 3072 32
```



```

kajo@x505ba: ~/catkin_ws/src/anti_col/scripts
File Edit View Search Terminal Tabs Help
roscore http://x... kajo@x505ba: ~/... kajo@x505ba: ~/... kajo@x505ba: ~/...
[INFO] [1615563151.029346]: header:
  seq: 1
  stamp:
    secs: 1615563151.03
    nsecs: 0
  frame_id: "laserTestID"
angle_min: 1.57117991564
angle_max: 4.71353974692
angle_increment: 0.0490926212001
time_increment: 1.0
scan_time: 1.0
range_min: 0.0
range_max: 12000.0
ranges: [266.0, 273.0, 266.0, 266.0, 271.0, 272.0, 255.0, 254.0, 293.0, 310.0, 337.0, 344.0, 390.0, 398.0, 479.0, 480.0, 569.0, 589.0, 713.0, 750.0, 981.0, 1024.0, 1588.0, 1643.0, 1651.0, 1660.0, 1672.0, 1677.0, 1428.0, 1232.0, 867.0, 812.0, 609.0, 587.0, 466.0, 442.0, 372.0, 362.0, 322.0, 315.0, 290.0, 283.0, 286.0, 291.0, 325.0, 333.0, 374.0, 387.0, 358.0, 389.0, 363.0, 368.0, 479.0, 515.0, 768.0, 800.0, 817.0, 816.0, 817.0, 815.0, 830.0, 836.0, 698.0, 656.0, 451.0]
intensities: [10552.0, 10648.0, 10725.0, 10556.0, 10524.0, 10672.0, 8494.0, 7133.0, 6207.0, 6688.0, 6405.0, 6316.0, 7011.0, 6530.0, 7960.0, 6328.0, 2920.0, 2753.0, 1963.0, 1744.0, 1562.0, 2543.0, 2808.0, 3105.0, 1359.0, 1539.0, 3370.0, 4101.0, 1962.0, 2254.0, 1847.0, 2172.0, 3604.0, 3758.0, 1318.0, 1516.0, 1743.0, 1859.0, 1763.0, 1742.0, 2296.0, 1975.0, 2632.0, 2433.0, 2486.0, 2421.0, 1757.0, 1909.0, 2134.0, 1861.0, 6947.0, 5092.0, 1665.0, 1483.0, 2452.0, 4295.0, 5439.0, 5584.0, 5886.0, 5781.0, 4956.0, 4829.0, 4474.0, 4635.0, 2010.0]

```

Figura 6.10: Visualização dos Dados do Protótipo via Consola.

Esta página foi intencionalmente deixada em branco.

# Capítulo 7

## Resultados

### 7.1 Estudo da Linearidade do Sensor LiDAR

É sempre boa prática confirmar os desempenhos publicitados pelos *datasheets* dos componentes, por isso, foi realizado um estudo de linearidade da deteção de obstáculos. Este teste consistiu em medições regulares de uma caixa preta, de 20 em 20 *cm*, ao longo de aproximadamente 3 *m*. Foi escolhida uma caixa preta por este tipo de objetos apresentar um índice de reflexão menor que objetos mais claros, aumentando a robustez de resultados.

O gráfico da Fig. 7.1 mostra os resultados desse teste, onde é visível uma boa relação de linearidade até aos 2.4 *m*, sendo que a partir daí surge um pico indesejável na medição com um erro bastante significativo, voltando depois a aproximar-se na linearidade nos 3 *m*.

Idealmente o teste deveria ser realizado de forma a testar o sensor ao longo de todo o seu alcance, no entanto isso não foi possível por não existirem condições para tal. O seguimento de testes consistiu na realização de testes com a rotação do sensor, e a integração do sistema no ROS, com visualização no mesmo.

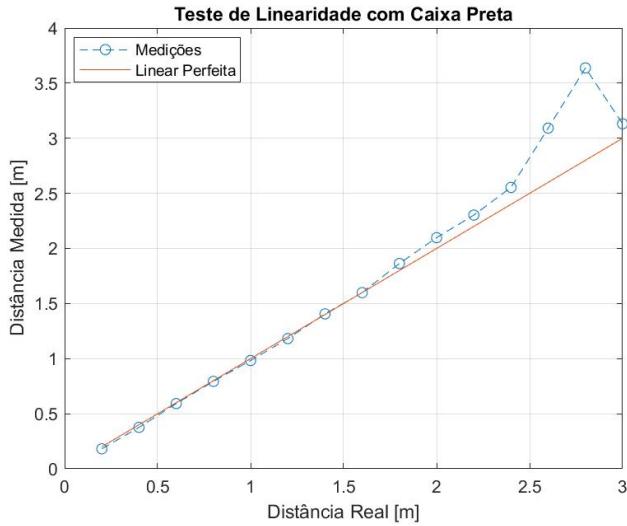


Figura 7.1: Gráfico do Estudo Linear do sensor TFmini-Plus.

## 7.2 Estudo Rotativo do Sensor LiDAR

Para este teste, o sistema foi montado no ambiente da Fig. 7.2, que ao nível do sensor, possui duas zonas de maior alcance, sendo estas a mesa de cabeceira e a zona entre a gaveta do lado esquerdo e a torre de computador por baixo da mesma. O projeto situa-se, na zona inferior da imagem em cima de um banco, e está conectado a um computador via cabo USB.

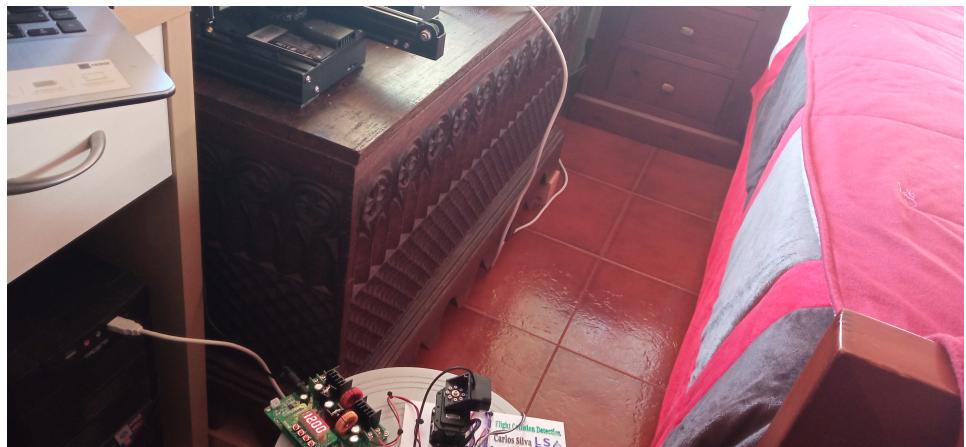


Figura 7.2: Ambiente de Testes Rotativos do Sensor LiDAR.

Foram então registados os dados de várias rotações do sensor, de forma a identificar parâmetros tais como a média e o desvio-padrão do *scan* nos dados de distância detetada e a intensidade do sinal, tal como demonstrado na Fig. 7.4. Segui-se uma visualização mais intuitiva da reconstrução do perímetro detetado pelo sensor, sendo esta apresentada em coordenadas cartesianas na Fig. 7.5 e em coordenadas polares na Fig. 7.6.

Em ambas as figuras, foram marcados os pontos detetados pelo LASER (a azul), o percurso que o feixe realizou (a verde), e os limites da realização desta rotação em específico (a vermelho), que foi limitada a  $180^\circ$ . Para além dos limites, fica a denominada de *blind zone*, zona que não foi alvo de deteção por limitação programada da rotação do servo-motor, dado que este eventualmente nunca terá necessidade de ultrapassar os  $180^\circ$  de rotação.

De notar e relembrar, que o perímetro detetado corresponde à zona entre a gaveta e a torre de computador (do lado esquerdo), e à zona da mesa de cabeceira (zona frontal com uma ligeira inclinação). A resolução pode ser aumentada ou diminuída de acordo com os parâmetros definidos na linha de comando, tendo em atenção de que quanto melhor a resolução, menor irá ser a velocidade do *scan*. A tabela da Fig. 7.3, demonstra as variações de frequência de amostragem registadas, de acordo com aumento da resolução, ou seja, o aumento da rotação do sensor por cada amostragem:

<b>n</b>	<b>Frequência de Amostragem (scan) [Hz]</b>	<b>Frequência de Amostragem (ponto) [Hz]</b>
32	0.20	12.8
64	0.29	9.28
128	0.43	6.88
256	0.71	5.68
512	1.00	4.00
1024	1.25	2.50

Figura 7.3: Frequência de Amostragem do Sistema.

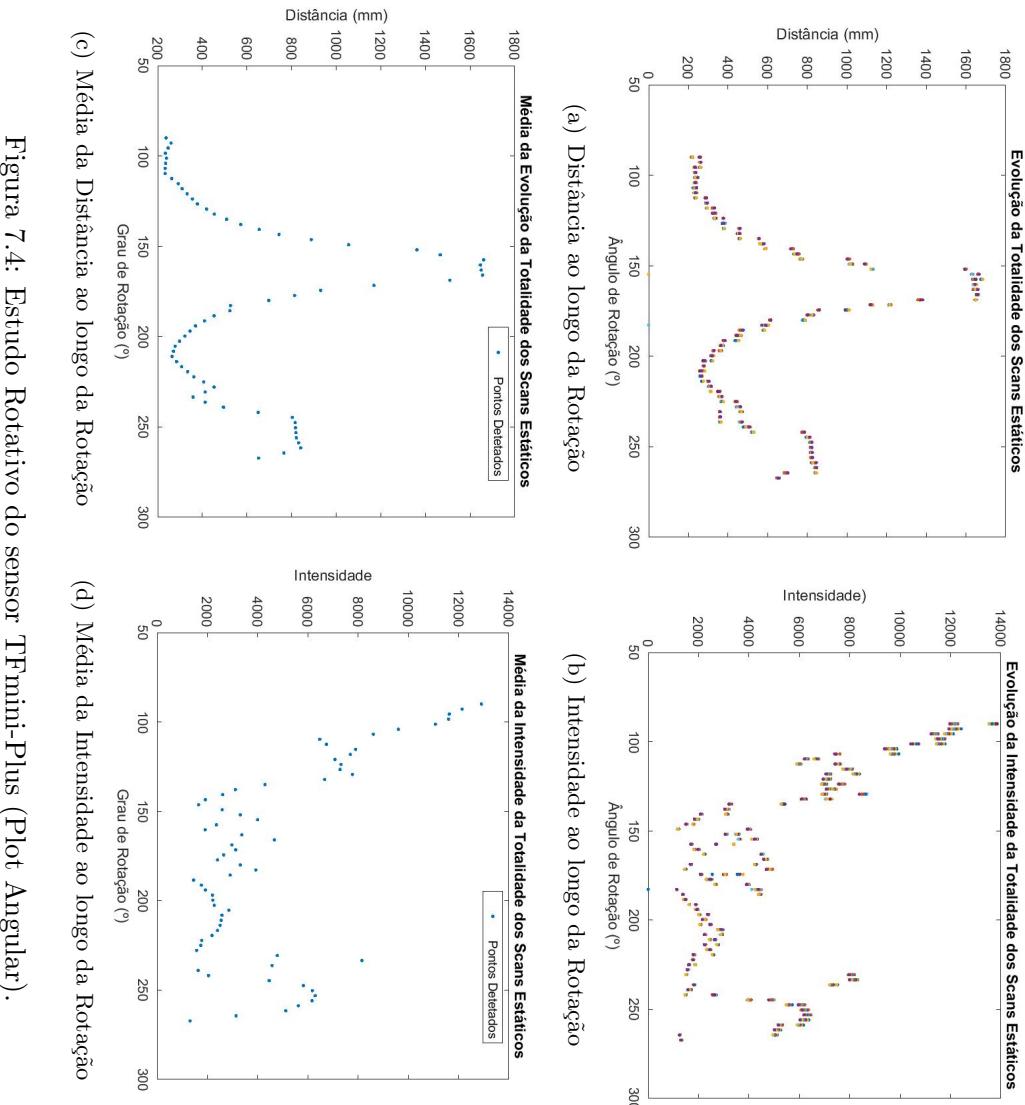


Figura 7.4: Estudo Rotativo do sensor TFmini-Plus (Plot Angular).

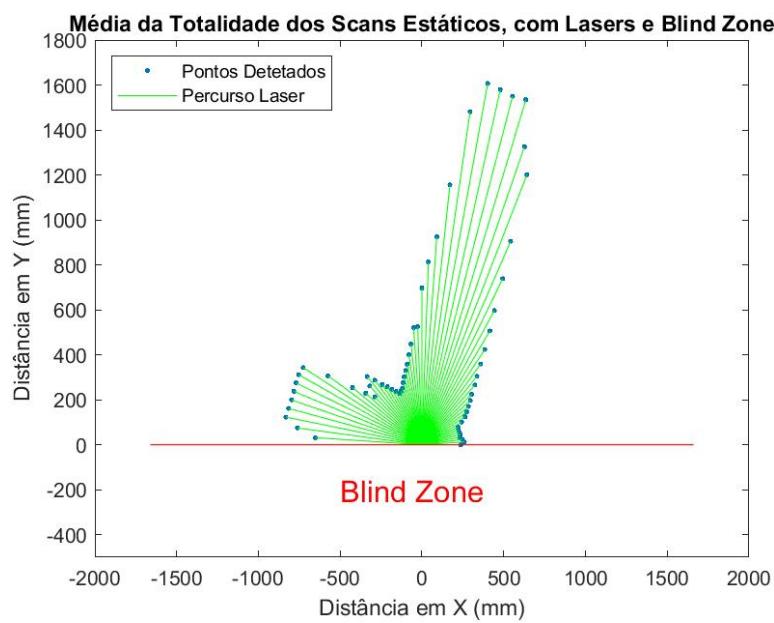


Figura 7.5: Estudo Rotativo do Sensor TFmini-Plus (Plot Cartesiano).

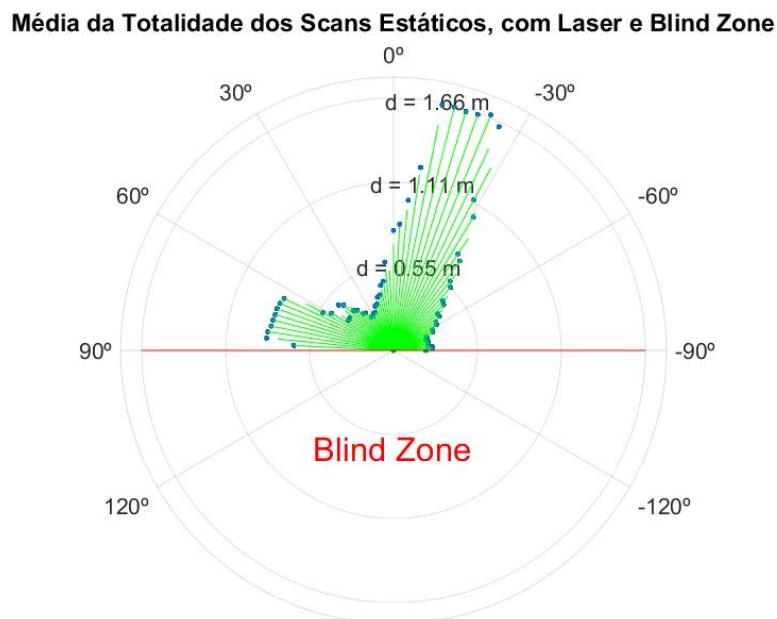
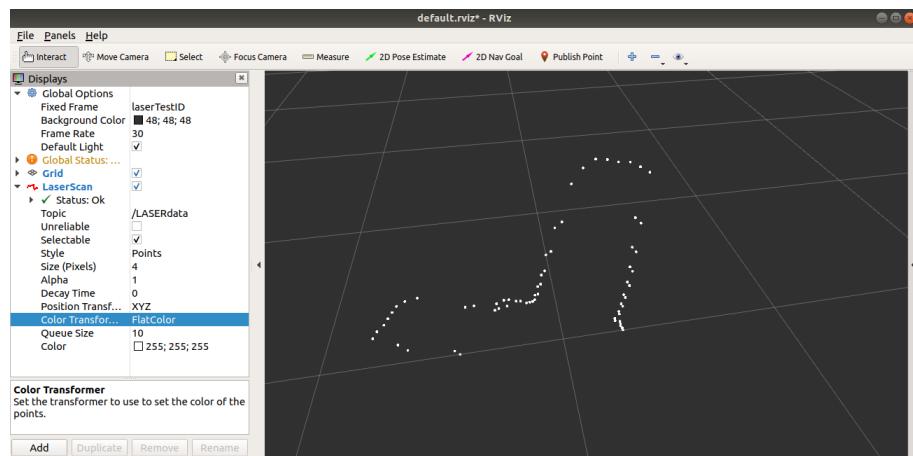


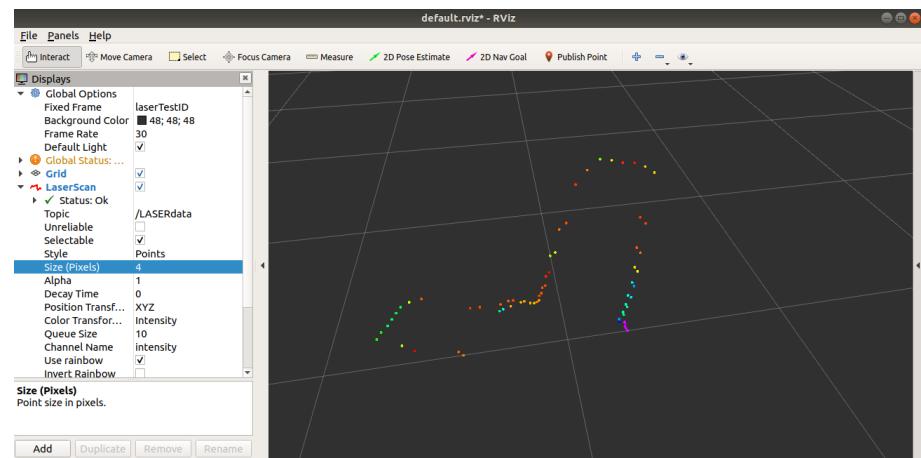
Figura 7.6: Estudo Rotativo do Sensor TFmini-Plus (Plot Angular).

### 7.3 Scan e Visualização em ROS

Para finalizar o processo, foi realizada a implementação no ROS, que permite utilizar a ferramenta RViz para visualização da nuvem de pontos de forma prática e em tempo-real, tal como exemplificado na Fig. 7.7. Foi então selecionado um *fixed frame* chamado de “laseTestID”, e aberta a mensagem do tipo “LaserScan”, na qual foi selecionado o tópico criado denominado de “LASERdata”. Esta ferramenta possibilita também a visualização em *color depth* para distinguir as zonas mais próximas das mais distantes do sensor.



(a) Visualização Simples



(b) Visualização com Escala de Profundidade

Figura 7.7: Visualização do Scan no RViz, em Tempo Real.

## Capítulo 8

# Conclusão e Trabalho Futuro

A deteção de obstáculos em *drones* é atualmente um popular tema de pesquisa e desenvolvimento devido ao crescimento do número deste tipo de veículos, sendo que as medidas de segurança começam a ter um enorme impacto nesta área. Alguns *drones* comerciais de alta gama já incorporam sistemas embebidos de deteção de obstáculos.

O objetivo deste projeto foi o desenvolvimento de uma estrutura modular que possa ser colocada em grande parte dos AUVs de grandes dimensões. Este projeto é capaz de correr em tempo-real, proporcionando um FoV extensivo, assim como um grande alcance de deteção. Por isso, é capaz de indicar a distância relativa do veículo a um potencial obstáculo.

Foi proposta uma estrutura flexível e modular de deteção de obstáculos. A comunicação é realizada de forma circular, suporta vários tipos de configuração e um número elevado de detetores de obstáculos. Utilizado um sensor LiDAR, o TFmini-Plus, em conjunto com um servo-motor Dynamixel, proporcionou-se um FoV variável na zona frontal do *drone*, que já inclui o funcionamento base dos módulos com sensor LiDAR estático. O sistema consegue detetar obstáculos até uma distância de 12 m em condições de muita luminosidade e com uma frequência de amostragem máxima de 1000 Hz.

De forma a controlar todo o sistema, foram desenvolvidas e produzidas PCBs responsáveis por cada módulo, sendo extremamente flexíveis e robustas. Esta foi a parte mais trabalhada de todo o projeto e a mais complexa. No qual surgiram 2 versões anteriores à versão final, que foram sido melhoradas de acordo com os requisitos do projeto. As maiores dificuldades revelaram-se principalmente no desenvolvimento totalmente individual, o que obrigou a uma gestão de tempo apertada.

No futuro prevê-se o *port* do projeto da PCB para o *software* Altium Designer, por ser um programa mais sofisticado e profissional. Será proposta também a possibilidade de desenhar uma placa para cada configuração possível dos módulos, de forma a reduzir o tamanho das placas para que estas possam ser acopladas diretamente aos sensores. Desta forma será possível reduzir a quantidade de componentes e conectores, o que levará a uma menor *footprint* total. O microcontrolador também poderá ser adquirido num *package* de menores dimensões, dado que a memória utilizada não excedeu a quantidade de memória dos *packages* de menores dimensões, ou seja, trocar um invólucro do mesmo microcontrolador ( $10 \times 10 \text{ mm}$ ), pelo mesmo num tamanho de  $7 \times 7 \text{ mm}$ . Esta opção também poderá ser mais viável economicamente (em grande escala), por se reduzir o número de componentes não utilizados, o que leva a um menor custo de produção.

Existiram alguns pequenos erros na produção da placa, na camada de *silkscreen*, como a sinalização errada do cátodo de um dos diódos do circuito do *buck-converter*, e a legenda trocada entre os LEDs de *status* e de *power*. Foi verificado que não é necessária a existência de um *switch* de seleção do *boot mode*, e por isso será retirado. Apesar de não ter qualquer falha, prevê-se também uma melhoria na colocação do circuito oscilador na PCB.

Ao nível de *software* está prevista a definição dos parâmetros de inicialização de *trigger*, a construção de uma interface gráfica de testes e configuração, e o uso dos periféricos DMA e CAN, do microcontrolador.

## Capítulo 8

---

Foi uma experiência bastante rica no que toca aos objetivos concretizados, aprendizagens e desenvolvimento pessoal. Foi pela primeira vez desenvolvida, pelo autor, uma PCB com 4 *layers*, com um STM32 e uma com montagem SMT (componentes SMD). No qual o autor sentiu uma notória evolução das suas capacidades de *design* e desenvolvimento de placas de circuito impresso. Em termos de programação, fora das unidades curriculares de RSDIS e SOTER, foi a primeira vez que o autor utilizou o freeRTOS e o ROS, como solução para um projeto de escolha aberta.

As novas capacidades adquiridas estenderam-se ao uso de sensores de tecnologia LiDAR e ao uso de servo-motores da família Dynamixel, que são bastante populares na área da robótica. Em eletrónica analógica/eletrónica de potência, foi a primeira vez que o autor desenvolveu um sistema utilizando um conversor *buck converter*. Este projetou também favoreceu as capacidade de uso e exploração de ferramentas de CAD Design tais como o Fusion360. Finalmente, e para concluir, os principais ficheiros do projeto estão disponíveis, em formato *open-source* no seguinte repositório do GitHub: <https://github.com/CarlosVieira99/Colllision-Avoidance-System>.

Esta página foi intencionalmente deixada em branco.

# Bibliografia

- [1] Adrian Carrio, Sai Vemprala, Andres Ripoll, Srikanth Saripalli, and Pascual Campoy. Drone detection using depth maps. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1034–1037. IEEE, 2018.
- [2] Reg Austin. *Unmanned aircraft systems: UAVS design, development and deployment*, volume 54. John Wiley & Sons, 2011.
- [3] Jürgen Scherer, Saeed Yahyanejad, Samira Hayat, Evsen Yanmaz, Torsen Andre, Asif Khan, Vladimir Vukadinovic, Christian Bettstetter, Hermann Hellwagner, and Bernhard Rinner. An autonomous multi-uav system for search and rescue. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 33–38, 2015.
- [4] Delair. RPLIDAR UX11. <https://delair.aero/delair-commercial-drones/professional-mapping-drone-delair-ux11/>.
- [5] Ming-Chang Wen, Cheng-Hsuan Yang, EX Sung, Tzong-Hann Wu, and Shih-Chung Kang. Optimized integration of uvas surveys, and image-based modeling strategies for digital terrain model reconstruction. In *Proceedings of 14th International Conference on Construction Applications of Virtual Reality (CONVR) in Construction and Islamic Architecture*, 2014.

- [6] Jang-Ho Cho, Dong-Sung Pae, Myo-Taeg Lim, and Tae-Koo Kang. A real-time obstacle avoidance method for autonomous vehicles using an obstacle-dependent gaussian potential field. *Journal of Advanced Transportation*, 2018, 2018.
- [7] Nils Gageik, Paul Benz, and Sergio Montenegro. Obstacle Detection and Collision Avoidance for a UAV with complementary low-cost Sensors. *IEEE Access*, 3:599–609, 2015.
- [8] Adrian Carrio, Jesus Tordesillas, Sai Vemprala, Srikanth Saripalli, Pascual Campoy, and Jonathan P How. Onboard Detection and Localization of Drones Using Depth Maps. *IEEE Access*, 8:30480–30490, 2020.
- [9] LEDDARTECH. LiDAR for Drones and Colission Avoidance. <https://leddartech.com/app/uploads/dlm-uploads/2016/02/Solution-Overview-Leddar-for-drone-collision-avoidance.pdf>.
- [10] Jędrzej Drozdowicz, Maciej Wielgo, Piotr Samczynski, Krzysztof Kulpa, Jarosław Krzonkalla, Maj Mordzonek, Marcin Bryl, and Zbigniew Jakielaszek. 35 GHz FMCW Drone Detection System. In *2016 17th International Radar Symposium (IRS)*, pages 1–4. IEEE, 2016.
- [11] Phuc Nguyen, Mahesh Ravindranatha, Anh Nguyen, Richard Han, and Tam Vu. Investigating Cost-effective RF-based Detection of Drones. In *Proceedings of the 2nd workshop on micro aerial vehicle networks, systems, and applications for civilian use*, pages 17–22, 2016.
- [12] Tim Hutchings, Susan Jeffries, and SJ Farmer. Architecting UAV sense & avoid systems. In *2007 Institution of Engineering and Technology Conference on Autonomous Systems*, pages 1–8. IET, 2007.

- [13] József Mezei, Viktor Fiaska, and András Molnár. Drone Sound Detection. In *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 333–338. IEEE, 2015.
- [14] Maarten Uijt de Haag, Chris G Bartone, and Michael S Braasch. Flight-test evaluation of small form-factor LiDAR and Radar Sensors for UAVs Detect-and-Avoid Applications. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–11. IEEE, 2016.
- [15] Frank Christnacher, Sébastien Hengy, Martin Laurenzis, Alexis Matwyschuk, Pierre Naz, Stéphane Schertzer, and Gwenael Schmitt. Optical and Acoustical UAV Detection. In *Electro-Optical Remote Sensing X*, volume 9988, page 99880B. International Society for Optics and Photonics, 2016.
- [16] António Sérgio Lima Raimundo. *Autonomous Obstacle Collision Avoidance System for UAVs in Rescue Operations*. PhD thesis, 2016.
- [17] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Flying Objects Detection from a single moving Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4128–4136, 2015.
- [18] Tamás Zsedrovits, Ákos Zarányi, Bálint Vanek, Tamás Péni, József Bokor, and Tamás Roska. Collision Avoidance for UAV using Visual Detection. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pages 2173–2176. IEEE, 2011.
- [19] Yuanwei Wu, Yao Sui, and Guanghui Wang. Vision-based Real-time Aerial Object Localization and Tracking for UAV Sensing System. *IEEE Access*, 5:23969–23978, 2017.

- [20] Fatih Gökçe, Göktürk Üçoluk, Erol Şahin, and Sinan Kalkan. Vision-based Detection and Distance Estimation of Micro Unmanned Aerial Vehicles. *Sensors*, 15(9):23805–23846, 2015.
- [21] Arne Schumann, Lars Sommer, Johannes Klatte, Tobias Schuchert, and Jürgen Beyerer. Deep Cross-domain Flying Object Classification for Robust UAV Detection. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.
- [22] Eren Unlu, Emmanuel Zenou, and Nicolas Riviere. Ordered Minimum Distance Bag-of-words Approach for Aerial Object Identification. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.
- [23] Eren Unlu, Emmanuel Zenou, and Nicolas Riviere. Using Shape Descriptors for UAV Detection. *Electronic Imaging*, 2018(9):128–1, 2018.
- [24] Petar Andraši, Tomislav Radišić, Mario Muštra, and Jurica Ivošević. Night-time Detection of UAVs using Thermal Infrared Camera. *Transportation research procedia*, 28:183–190, 2017.
- [25] Adrian Carrio, Yucong Lin, Srikanth Saripalli, and Pascual Campoy. Obstacle Detection System for Small UAVs using ADS-B and Thermal Imaging. *Journal of Intelligent & Robotic Systems*, 88(2-4):583–595, 2017.
- [26] Ashley D Waite and AD Waite. *Sonar for practising engineers*, volume 3. Wiley New York, 2002.
- [27] Nischay Gupta, Jaspreet Singh Makkar, and Piyush Pandey. Obstacle Detection and Collision Avoidance using Ultrasonic Sensors for RC Multirotors. In *2015 International Conference on Signal Processing and Communication (ICSC)*, pages 419–423. IEEE, 2015.

- [28] Nils Gageik, Julian Rothe, and Sergio Montenegro. Data Fusion Principles for Height Control and Autonomous Landing of a Quadrocopter. In *Proc. UAVveek Workshop Contrib.*, 2012.
- [29] OSOYOO. HC-SR04. <https://osoyoo.com>.
- [30] Intel. Intel Real Sense. <https://www.intel.com>.
- [31] John A Christian and Scott Cryan. A Survey of LiDAR Technology and its use in Spacecraft Relative Navigation. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4641, 2013.
- [32] Lei Lei, Chunxia Qiu, Zhenhai Li, Dong Han, Liang Han, Yaohui Zhu, Jintao Wu, Bo Xu, Haikuan Feng, Hao Yang, et al. Effect of Leaf Occlusion on Leaf Area Index Inversion of Maize using UAV LiDAR data. *Remote Sensing*, 11(9):1067, 2019.
- [33] James D Spinhirne. Micro Pulse LiDAR. *IEEE Transactions on Geoscience and Remote Sensing*, 31(1):48–55, 1993.
- [34] TeraBee. A brief introduction to time-of-flight sensing, Oct 2020.
- [35] Michigan Tech Research Institute – Michigan Technological University. Benchmarking Sensors for Vehicle Computer Vision Systems, Oct 2020.
- [36] Texas Instruments. *Introduction to Time-of-Flight Long Range Proximity and Distance Sensor System Design User’s Guide*. Texas Instruments.
- [37] Peter Rieger. Range Ambiguity Resolution Technique applying Pulse-position Modulation in Time-of-Flight Scanning LiDAR Applications. *Optical Engineering*, 53(6):061614, 2014.
- [38] Cristiano Niclass, Mineki Soga, Hiroyuki Matsubara, Masaru Ogawa, and Manabu Kagami. A 0.18 m CMOS SoC for a 100 m Range, 10

- frames/s 200, 96-pixel, Time-of-Flight Depth Sensor. *IEEE Journal of solid-state circuits*, 49(1):315–330, 2013.
- [39] B Hamamatsu Photonics K.K. ToF System, Oct 2020.
- [40] Preethi Padmanabhan, Chao Zhang, and Edoardo Charbon. Modeling and Analysis of a Direct Time-of-Flight Sensor Architecture for LiDAR Applications. *Sensors*, 19(24):5464, 2019.
- [41] Sanggwon Lee, Keita Yasutomi, Masato Morita, Hodaka Kawanishi, and Shoji Kawahito. A Time-of-Flight Range Sensor using Four-tap Lock-in Pixels with High near Infrared Sensitivity for LiDAR Applications. *Sensors*, 20(1):116, 2020.
- [42] Aongus McCarthy, Robert J Collins, Nils J Krichel, Verónica Fernández, Andrew M Wallace, and Gerald S Buller. Long-range Time-of-Flight Scanning Sensor based on High-speed Time-correlated Single-photon Counting. *Applied optics*, 48(32):6241–6251, 2009.
- [43] Augusto Ronchini Ximenes, Preethi Padmanabhan, Myung-Jae Lee, Yuichiro Yamashita, Dun-Nian Yaung, and Edoardo Charbon. A  $256 \times 256$  45/65nm 3d-stacked spad-based direct tof image sensor for lidar applications with optical polar modulation for up to 18.6 db interference suppression. In *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 96–98. IEEE, 2018.
- [44] Hui Li, Philip Wolf, Philip Moser, Gunter Larisch, James A Lott, and Dieter Bimberg. Vertical-cavity Surface-emitting LASERs for Optical Interconnects. *SPIE Newsroom*, 25, 2014.
- [45] Mial E Warren, David Podva, Preethi Dacha, Matthew K Block, Christopher J Helms, John Maynard, and Richard F Carson. Low-divergence High-power VCSEL Arrays for LiDAR Application. In *Vertical-Cavity*

- Surface-Emitting Lasers XXII*, volume 10552, page 105520E. International Society for Optics and Photonics, 2018.
- [46] John Shackleton, Brian VanVoorst, and Joel Hesch. Tracking People with a 360-degree LiDAR. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 420–426. IEEE, 2010.
- [47] SLAMTEC. Rplidar a1. <https://www.slamtec.com/en/Lidar/A1>.
- [48] Blickfeld. What is solid-state lidar?, Oct 2020.
- [49] Andrea Cherubini, Fabien Spindler, and Francois Chaumette. Autonomous visual navigation and laser-based moving obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2101–2110, 2014.
- [50] Jongho Park and Namhoon Cho. Collision Avoidance of Hexacopter UAV Based on LiDAR Data in Dynamic Environment. *Remote Sensing*, 12(6):975, 2020.
- [51] TERABEE. Terabee. <https://www.terabee.com/>.
- [52] Paul Schlyter. Radiometry and photometry in astronomy. *Available: stjarnhimlen. se/comp/radfaq. html*, 1, 2009.
- [53] STMicroelectronics. Flight sense sensors. <https://www.st.com/>.
- [54] Benewake. Benewake products. <http://en.benewake.com/>.
- [55] Garmin. Garmin buy. <https://buy.garmin.com/>.
- [56] Robotis. Dynamixel. <https://www.robotis.us/dynamixel/>.
- [57] Parallax. Parallax standard servo. <https://www.parallax.com/product/parallax-standard-servo/>.

- [58] STMicroelectronics. Stm32 32-bit arm cortex mcus. <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus>.
- [59] Carmine Noviello. Mastering stm32. *A step-by-step guide to the most complete ARM Cortex-M platform, using a free and powerful development environment based on Eclipse and GCC*. Leadpub, 2017.
- [60] Clayton R Paul. Effectiveness of multiple decoupling capacitors. *IEEE Transactions on Electromagnetic Compatibility*, 34(2):130–133, 1992.
- [61] AN2867 Application Note STMicroelectronics. Oscillator design guide for stm8af/al/s and stm32 microcontrollers, 2009.
- [62] AN2606 Application Note STMicroelectronics. Stm32 microcontroller system memory boot mode, 2007.
- [63] AN3156 Application Note STMicroelectronics. Usb dfu protocol used in the stm32 bootloader, 2010.
- [64] KiCad. About kicad. <https://kicad-pcb.org/>.

## Apêndice A

# Esquema Eléctrico Total do Sistema

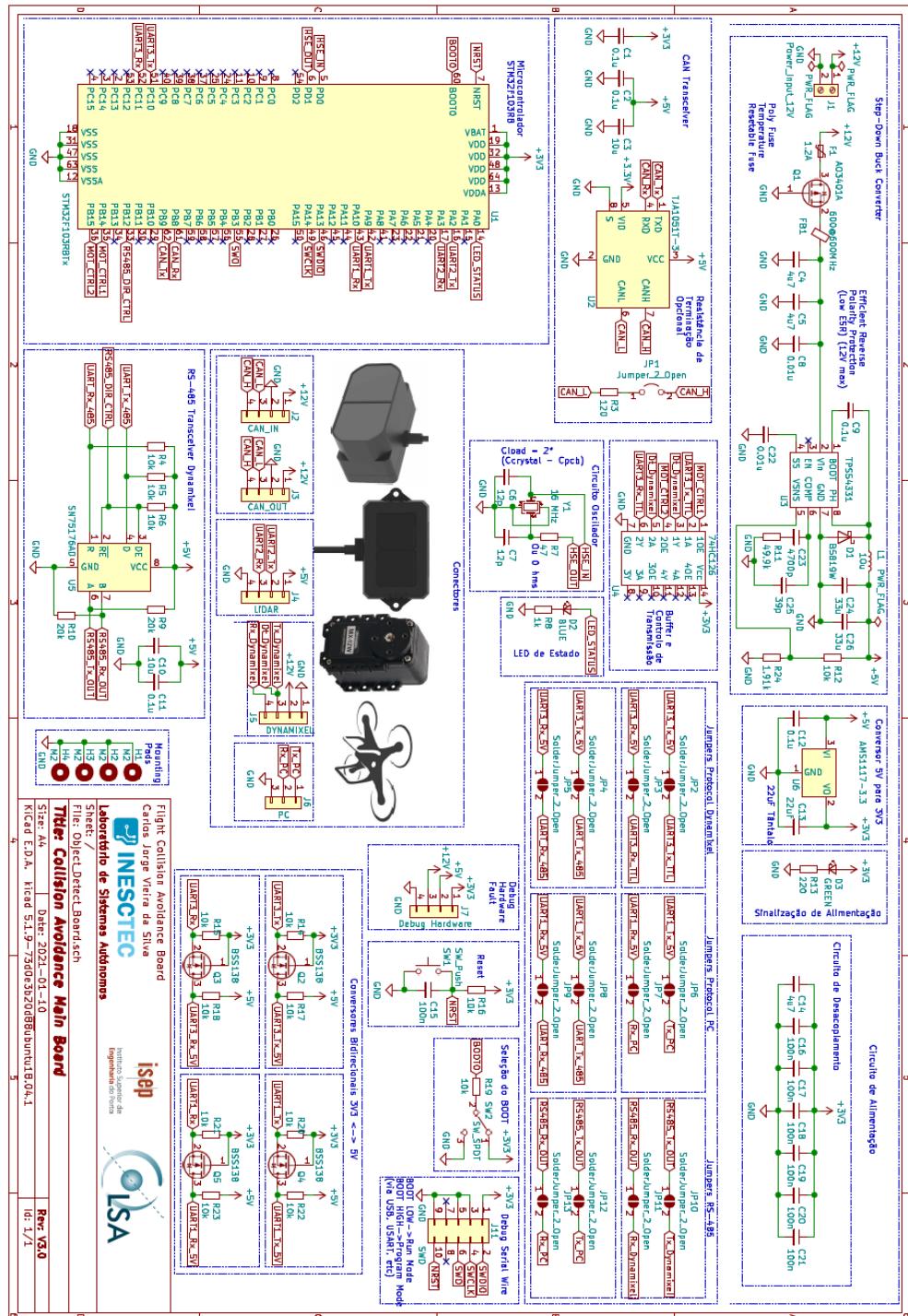


Figura A.1: Circuito Total.