



ESCOLA NAVAL



talant de bi-faire

Afonso José da Silva Martins Nunes Ramos

Vela Autónoma – Melhoria do Sistema de Velas

Rígidas do USV *Barlavento*

Melhoria da Plataforma Barlavento

Dissertação para obtenção do grau de Mestre em Ciências Militares

Navais, na especialidade de Marinha



Alfeite
2021



ESCOLA NAVAL

talant de bi-faire



Afonso José da Silva Martins Nunes Ramos

***Vela Autónoma – Melhoria do Sistema de Velas Rígidas do
USV Barlavento***

Melhoria da Plataforma Barlavento

**Dissertação para obtenção do grau de Mestre em Ciências Militares Navais, na
especialidade de Marinha**

Orientação de: Professor Doutor Vítor Sousa Lobo

Co-orientação de: 2TEN EN-MEC Castro Fernandes

O Aluno Mestrando,

O Orientador,

ASPOF M Nunes Ramos

Professor Doutor Vítor Lobo

Alfeite

2021

Epígrafe

“Um pensamento original vale mil citações descartáveis”

Diógenes de Sinope

Agradecimentos

A realização desta dissertação de mestrado não teria sido possível sem o auxílio de várias pessoas e entidades, às quais venho agora agradecer a disponibilidade, paciência e apoio incondicional, durante todo o percurso da minha dissertação, Escola Naval e vida.

Em primeiro lugar à minha família, sem a qual nada seria possível.

À Sofia, pela sua inquebrável confiança em mim e apoio constante.

Aos meus orientadores, Professor Doutor Vítor Lobo e 2TEN EN-MEC Castro Fernandes, por toda a disponibilidade, apoio e motivação.

Ao GMAR M Barracosa Santos, pela amizade e inspiração.

A toda a guarnição do NRP *Sagres*, particularmente à Câmara de Oficiais, navio em que estagiei, pelo seu imprescindível valor e por todos os ensinamentos dados.

Ao CTEN ST-MEC Pinto Guimarães, SAJ ETS Henriques Bandeja, 1SAR ETS pelo seu apoio incondicional e constante tutela.

Aos meus camaradas do curso Capitão-tenente Raúl Alexandre Cascais, e em particular ao camarada ASPOF M Carvalho de Oliveira, pelo seu companheirismo e amizade.

Resumo

Veículos autónomos de superfície são uma tecnologia promissora, apresentando vantagens em missões longas, reduzindo o risco e o custo das mesmas. Esta tese propõe o aprontamento de um veleiro autónomo totalmente operacional, proporcionando à Marinha uma base para o desenvolvimento de futuras embarcações autónomas. Iniciou-se o projeto com o desenho de um incremento vertical ao casco, através do programa *SolidWorks*, de modo a albergar os novos componentes de controlo dos mastros, seguido do estudo e desenvolvimento destes componentes e sua montagem. O algoritmo de controlo de leitura dos novos sensores foi desenvolvido. De seguida foram implementados os novos sensores, nomeadamente anemómetro, bússola e placa de microcontrolador, operando no sistema NMEA¹. O trabalho será finalizado com provas de mar e análise dos seus resultados.

Palavras-chave: Veleiro Autónomo, Impressão 3D, Microcontrolador, CAD, Algoritmo de C2

¹ *National Marine Electronics Association*

Abstract

Autonomous surface vehicles are promising technology, offering improvements in extense missions, by reducing their risk and cost. This project proposes to assemble a fully operational autonomous sailboat, providing the Portuguese Navy with a basis for the development of future autonomous vessels. The work begun with the design of a vertical increment to the hull, using *SolidWorks*, for the accomodation of the new mast control componentes, followed by the study and development of such componentes and its assembly. The command algorithm was upgraded to encompass the new sensors. New sensors were implemented , namely an anemometer, compass and microcontroller, integrated in the NMEA system. Test results and its analysis will be accomplished after sea trials.

Key-words: Autonomous Sailboat, 3D printing, Microcontroller, CAD, Command and Control Algorithm

Índice

Epígrafe.....	v
Agradecimentos	vii
Resumo	viii
Abstract.....	ix
Índice de Figuras.....	xii
Índice de Tabelas	xv
Índice de Equações	xvi
Siglas e Abreviaturas	xvii
1 Introdução.....	1
1.1 Enquadramento.....	1
1.2 Justificação do tema	2
1.3 Objetivo da dissertação	2
1.4 Estrutura da investigação	3
2 Revisão da literatura	5
2.1 História da Navegação à Vela	5
2.2 História da Vela Autónoma.....	7
2.3 Vela Autónoma em Portugal	9
2.4 Vela Autónoma na Marinha Portuguesa	10
3 Atualização da estrutura física.....	13
3.1 Elevação do Convés	14
3.2 Mecanismo de Controlo dos Mastros.....	20
3.3 Atualização das portas de visita	23
3.4 Criação de uma verga de sinais	30

4	Atualização da componente eletrónica	35
4.1	Módulo de Controlador Solar.....	38
4.2	Módulo do Microcontrolador	42
4.3	Módulo dos atuadores	48
4.4	Verga de sinais	53
4.5	Atualização do Algoritmo de Controlo	54
5	Testes e Provas de Mar	55
5.1	Provas de Mar com o Catamarã X-2601.....	55
5.2	Testes ao sistema completo em terra.....	58
5.3	Testes na Bacia de Manobra da Base Naval de Lisboa.....	60
5.4	Exercício REP(MUS) 2021	62
6	Conclusões.....	65
	Bibliografia	67
	Apêndices.....	69
	Apêndice 1 – Bill of Materials dos Painéis Solares.....	69
	Apêndice 2 – Rasto em modo RC 1 - Catamarã X-2601	70
	Apêndice 3 – Rasto em modo autónomo 1 - Catamarã X-2601	70
	Apêndice 4 – Rasto em modo RC 2 - Catamarã X-2601	71
	Apêndice 5 – Código C de Comando – do Veleiro	71
	Apêndice 6 – Código controlo	92
	Apêndice 7 – Código de Leitura do NMEA	95
	Apêndice 8 – Código do GPS	100
	Apêndice 9 – Código do Leme	107
	Apêndice 10 – Código dos Waypoints	110
	Apêndice 11 – Código do Anemómetro	113

Índice de Figuras

Figura 2-1 - NTM Creoula.....	6
Figura 2-2 - NRP Sagres.....	7
Figura 2-3 - Atlantis em testes de mar (2001)	8
Figura 2-4 - Frota SailDrones, São Francisco.....	9
Figura 2-5 - Veleiro FAST,, durante a travessia Cascais-Sesimbra, apoiada pela Escola Naval.....	10
Figura 2-6 - Veleiro Barlavento.....	11
Figura 3-1 - Preparação inicial. Retiradas primeiras camadas de tinta.....	14
Figura 3-2 - Barlavento sem elevação	15
Figura 3-3 - Barlavento com elevação.....	16
Figura 3-4 - Medidas das portas de visita	16
Figura 3-5 - Split e perfurações	17
Figura 3-6 - Colagem das novas peças	18
Figura 3-7 - Aplicação de fibra de vidro.....	19
Figura 3-8 - Acabamento com TopCoat	19
Figura 3-9 - Veio dos matros	20
Figura 3-10 - Apoio dos Veios ao convés.....	21
Figura 3-11 - Suporte dos veios.....	21
Figura 3-12 -Fix dos veios nos mastros	22
Figura 3-13 - Tirante do mastro	22
Figura 3-14 - Tirante do mastro colocado	23
Figura 3-15 - Porta a ré.....	24
Figura 3-16 - Porta do mastro a ré	24
Figura 3-17 - Porta de visita do catamarã da CEOV	25
Figura 3-18 - Encaixe das porcas.....	25
Figura 3-19 - Colagem das caixas das porcas.....	26
Figura 3-20 - Desenho 2D das portas de visita.....	27
Figura 3-21 - Esquema peças do encaixe do retentor	27
Figura 3-22 - Perfil de retentor TC	28
Figura 3-23 - Caixa do retentor.....	28
Figura 3-24 - Tampa da caixa do retentor.....	29

Figura 3-25 - Aspeto final das portas de visita	29
Figura 3-26 - Verga de sinais.....	31
Figura 3-27 - Verga de sinais no topo dos mastros.....	31
Figura 3-28 - Ligadores	32
Figura 3-29 - Peça de união ao mastro	32
Figura 3-30 - Peça de união à verga	33
Figura 4-1 - LT1000 NRU	36
Figura 4-2 - Clipper Tactical Wind Mast Sensor - MHU-TACT	36
Figura 4-3- Caixas da eletrónica	37
Figura 4-4 - Tampa das caixas de eletrónica	37
Figura 4-5 - Disposição dos módulos no casco do veleiro	37
Figura 4-6 - Esquema Controlador Solar	38
Figura 4-7 - Tampa Controlador Solar	39
Figura 4-8 - Módulo completo.....	39
Figura 4-9 - Controlador Solar.....	40
Figura 4-10 – Bateria utilizada	40
Figura 4-11 - Testagem dos painéis e controlador solar.....	41
Figura 4-12 - Módulo finalizado.....	42
Figura 4-13 - Módulo LoRa.....	43
Figura 4-14 - Calibração do Módulo LoRa	44
Figura 4-15 - Esquema ligações LoRa.....	44
Figura 4-16 - Parâmetros LoRa	45
Figura 4-17 - Conversor RS232 para TTL.....	45
Figura 4-18 - Tampa módulo controlador.....	46
Figura 4-19 - Esquema módulo controlador	47
Figura 4-20 - Módulo de controlo finalizado.....	47
Figura 4-21 - Esquema de ligações da bússola	48
Figura 4-22 - Tampa módulo atuadores.....	49
Figura 4-23 - Controlador Veyron	49
Figura 4-24- Esquema módulo atuadores	50
Figura 4-25 - Aplicação móvel Electrodoc. Cálculo da resistência para LED.....	51
Figura 4-26 - Servo do leme	51
Figura 4-27 - Regulador de tensão.....	52
Figura 4-28 – Módulo dos atuadores completo	52

Figura 4-29 - Rádios TARANIS	53
Figura 4-30 - Esquema da verga de sinais	54
Figura 5-1 - Catamarã X-2601	55
Figura 5-2 - Rasto em modo RC (1ª pernada)	56
Figura 5-3 - GCS em modo RC	57
Figura 5-4 - Rasto em modo autónomo	57
Figura 5-5 - Rasto em modo RC (2ª pernada)	58
Figura 5-6 - Módulos completos integrados no veleiro	59
Figura 5-7 - Display da GCS	60
Figura 5-8 - Prova de mar do Barlavento	61
Figura 5-9 – Barlavento	61
Figura 5-10 - Display no OpenCPN	62
Figura 5-11 - Plataforma da Royal Navy MADFOX	63
Figura 5-12 – Bujador	64
Figura 0-1 - Bill Of Materials - Painéis Solares	69
Figura 0-2 - Rasto em Modo RC 1- Catamarã	70
Figura 0-3 - Rasto em Modo autónomo 1- Catamarã	70
Figura 0-4 - Rasto em modo RC 2	71

Índice de Tabelas

Tabela 1- Consumo energético estimado.....	41
Tabela 2 – Resultados dos testes aos painéis solares.....	42
Tabela 3 - Características Arduino Mega 2560	43

Índice de Equações

Equação 1 - Cálculo da resistência nos LED's.....	50
---	----

Siglas e Abreviaturas

CAD – Computer Aided Design

CEOV – Célula de Experimentação Operacional de Veículos não-tripulados

EN – Escola Naval

FEUP - Faculdade de Engenharia da Universidade do Porto

FHSS - Frequency-hopping spread spectrum

GPS – Global Position System

IH – Instituto Hidrográfico

LoRa – *Long Range* (nome dado a um sistema e protocolo de radio-comunicação digital)

Marocup - MARitime RObotic CUP

NATO - North Atlantic Treaty Organization

PLA - Polylactic Acid

RC – *Remote Control*

REP(MUS) – *Robotic Exercise Prototyping (Maritime Unmanned Systems)*

SAN – Serviço de Apoio Naval

STL - stereolithography

1 Introdução

O presente capítulo está dividido em cinco secções. Na primeira será enquadrado o tema da dissertação. Na segunda, será justificado o tema no contexto atual. Terceiramente, é apresentado o objetivo principal da dissertação, e respetivos objetivos secundários. Em quarto lugar está descrita a metodologia adotada para a realização do trabalho, de modo a atingir os objetivos supramencionados.

1.1 Enquadramento

Um veículo autónomo, para o ser considerado, deve ser capaz de desempenhar a sua função sem a intervenção constante de um operador, além de ser energeticamente independente. Na sua constituição, deve ter um computador de bordo, que constitui o centro do sistema autónomo, responsável pela receção de sinais, processamento dos mesmos, e reação adequada aos eventos que se sucederem, enviando comandos para atuadores, e gerindo as comunicações com o exterior. Esse computador deve estar integrado com sensores, atuadores, sistemas de armazenamento e/ou produção de energia, módulo de comunicação com estação terrestre e um sistema de controlo de emergência, para salvaguarda dos demais componentes.

Vive-se atualmente um período de rapidíssimo desenvolvimento tecnológico, e, subsequentemente, a crescente presença de automação em todos os setores de atividade (previamente) humana. Veículos autónomos de superfície assumem naturalmente uma maior importância no estudo da navegação e na sua evolução, apresentando novas características e melhorias em relação aos veículos convencionais. A segurança da navegação, um dos critérios essenciais no desenvolvimento de qualquer embarcação, pode ser mais facilmente atingida reduzindo um dos fatores mais difíceis de mitigar, o “Erro Humano”.

Este projeto insere-se no projeto eVentos, aproveitando um veleiro desenvolvido inteiramente pela Escola Naval em 2016 e cuja estrutura foi posteriormente melhorada em 2019 (Fernandes, 2016) (Santos, 2019).

1.2 Justificação do tema

O tema desta dissertação apresenta-se como um desafio e uma solução para os problemas que a Marinha Portuguesa enfrenta, que é o patrulhamento e vigilância de uma vasta zona do Atlântico Norte. De facto, os veículos autónomos apresentam vantagens promissoras na execução de missões de longa e muito longa duração, reduzindo muito substancialmente os custos financeiros, e ultrapassando obstáculos que navios tripulados têm nessas missões de grande *endurance*, como a manutenção da moral e descanso da guarnição e a necessidade de abastecimento regular de bens. Operações complexas e imprevisíveis são similarmemente adaptáveis a sistemas autónomos, fornecendo soluções baseadas na análise de dados para uma rápida tomada de decisão.

Surge assim a necessidade de desenvolver sistemas autónomos, internamente, apresentando valências em rocega de minas, monitorização oceânica dinâmica, recolha de dados meteo-oceanográficos, testagem de novas tecnologias e protótipos e deteção de atividades ilegais. Só com essa experiência de desenvolvimento interno à Marinha poderemos ter os conhecimentos necessários para compreender bem esses sistemas, quer em termos de conceção quer em termos de operação, e produzir ou adquirir no exterior, mas de forma sensata, os sistemas que melhor nos servem.

Este trabalho servirá como base para a construção de futuras embarcações a serem empenhadas em atividades de interesse para a Marinha, assim como trabalhos anteriores (Fernandes, 2016) (Santos, 2019) serviram como base a todo o sistema de comando e controlo dos sistemas autónomos de marinha Trimarã *X-2401*, e Catamarã *X-2601*, desenvolvidos pela CEOV.

1.3 Objetivo da dissertação

O objetivo será, sucintamente, aprontar a plataforma *Barlavento* para a realização de travessias oceânicas, através da melhoria e aperfeiçoamento dos sistemas de comando, assim como da instalação e integração de novos sensores. Estas melhorias resultam da experiência adquirida na prática com os trabalhos anteriores desenvolvidos na Escola Naval (Fernandes, 2016) (Barracosa, 2018), e dificilmente poderiam ser antevistas apenas com análises teóricas, necessariamente simplificadas em relação à realidade.

Os objetivos secundários estão seccionados para cada componente a melhorar.

Em relação ao controlo das velas rígidas, foi desenvolvido um novo mecanismo de controlo de mastros. Por forma a acomodar o novo mecanismo, surgiu a necessidade de aumentar o espaço interior do veleiro, e, para tal, desenhou-se uma elevação do casco, seccionada em dois espaços centrados nos mastros para o controlo dos mesmos, um espaço intermédio para estabelecimento do circuito elétrico e ainda uma secção à ré para o mecanismo de controlo do leme.

O algoritmo de controlo das velas necessitava de ser atualizado, sendo a versão anterior escrita para o sistema *Arduíno* e agora utilizada uma versão aplicada ao microcontrolador Atmel2560, em linguagem C.

1.4 Estrutura da investigação

O trabalho está dividido em seis capítulos. No capítulo *Introdução* é apresentado o âmbito do trabalho científico, fazendo o seu enquadramento teórico, além de uma descrição da investigação realizada, da metodologia adotada e dos objetivos determinados, terminando com a estrutura do trabalho em si.

No Capítulo 2 – Revisão da Literatura, aborda-se brevemente a história da navegação à vela, o percurso até à vela autónoma, descrição de outros veleiros autónomos e regatas de vela autónoma e a utilização de plataformas não tripuladas na Marinha Portuguesa.

No Capítulo 3 – Atualização da estrutura física, são estudados os resultados obtidos em trabalhos anteriores para o desenvolvimento de medidas a aplicar neste trabalho.

No Capítulo 4 – Atualização da componente eletrónica, é descrito o processo de criação dos vários componentes necessários e da implementação dos novos sensores.

No Capítulo 5 – Testes e Provas de Mar, é feita a análise dos dados obtidos por meio da experimentação física do veleiro no mar, validando os conceitos implementados.

As Conclusões são elaboradas tendo em conta os resultados obtidos, retirando-se ilações e planeando de ações futuras, terminando assim o ciclo de estudos sobre o veleiro *Barlavento*.

2 Revisão da literatura

2.1 História da Navegação à Vela

Navegar à vela tem sido um símbolo do povo português há mais de 5 séculos (Gonçalves, 2009), tendo sofrido poucas alterações desde então, mantendo-se como uma base da nossa cultura e tradição. Este projeto propõe trazer verdadeiramente esta prática ao século XXI, através do uso de veículos autónomos de superfície, que representam uma das maiores evoluções da atividade desde a invenção das velas latinas, e uso na epopeia das descobertas por parte do povo português, permitindo assim navegar contra o vento, ou à bolina. Novos mares nunca dantes navegados eram agora percorridos pelos portugueses, com rumo a portos cada vez mais distantes.

Após a era áurea de Portugal e dos seus Descobrimentos, surge cada vez mais a aplicação bélica da vela, sendo as potências mundiais determinadas por quem detinha o controlo do mar, e o domínio da arte da navegação (Oliveira, 1555) (Mahan, 1890).

O século XIX marca o declínio da vela como meio predominante de navegar, com o aparecimento dos navios movidos a vapor, independentes do vento e do mar, descritos pelo Oficial da Marinha Espanhola Blasco de Garay como “...uma máquina, por meio da qual se faria andar um navio sem a ajuda de velas ou remos.” (Archivo Popular, 1838).

Em Portugal, a prática de vela continuou a ter uma presença relevante, utilizando navios à vela para a pesca do bacalhau, contando com uma frota nacional, que efetuava travessias oceânicas (Villiers, 1950). O NRP *Creoula*², já não como pesqueiro, mas ainda ao serviço de Portugal, na Marinha Portuguesa, é um monumento vivo a esta cultura e tradição tão enraizadas na índole portuguesa, continuando a educar portugueses nas práticas de marinharia e vela, assim como o *Santa Maria Manuela*, navio irmão do NRP *Creoula* e o *Argus*, que será recuperado e transformado em museu.

² Site da Marinha. https://www.marinha.pt/pt/os_meios/veleiros/Paginas/ntm-creoula.aspx



Figura 2-1 - NTM Creoula

Outro exemplo a referir é o NRP *Sagres*³, que embora de construção original alemã (Gonçalves, 2009), é uma embaixada itinerante de Portugal, conjugando em si o peso da história e cultura portuguesa, carregando-o com orgulho a todos os cantos do mundo, saudoso na recordação de tempos passados, mas superando igualmente os desafios do presente.

³ Site da Marinha. https://www.marinha.pt/pt/os_meios/veleiros/Paginas/nrp-sagres.aspx



Figura 2-2 - NRP Sagres

2.2 História da Vela Autónoma

Enquadrando os veleiros autónomos como uma das aplicações do conceito de veículos autónomos, estes surgem como uma alternativa aos de propulsão convencional devido à redução dos custos energéticos, abrindo um leque de possibilidades para a realização de missões prolongadas, sendo os seus gastos energéticos limitados à operação do computador de bordo, dos seus sensores e atuadores.

Historicamente, a automação do leme foi a primeira aplicação do conceito (Stelzer & Jafarmadar, 2011). Aplicado em inúmeros navios e embarcações, permite essencialmente ter uma função de “piloto automático”. Este leme automatizado, seja mecânica ou eletricamente, assegura um rumo constante após indicação inicial, muito adequado a travessias oceânicas.

No decorrer das últimas duas décadas, foram desenvolvidos vários projetos de veleiros autónomos. O *Atlantis* (Elkaim & Kelbley, 2006), desenvolvido em 2001, era um catamarã autónomo, que navegava através do seguimento de trajetos segmentados. Posteriormente, aplicou-se esta tecnologia que foi aproveitada no catamarã *HWT-XI* (Elkaim & Boyce, 2007).

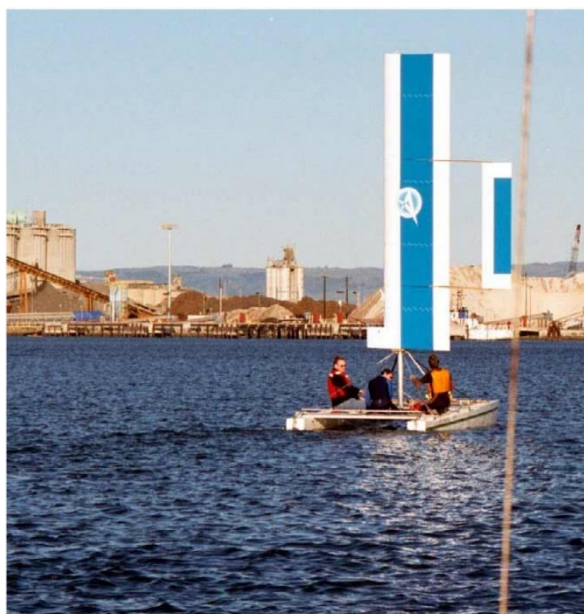


Figura 2-3 - Atlantis em testes de mar (2001)

Em 2014, a empresa *Ocean Aero* desenhou um monocasco, denominado *Submaran* (Mayerfeld, 2017), capacitado para navegação à superfície e subsuperfície.

Entretanto, também uma empresa, *SailDrone* (Voosen, 2018) e (Gentemann, et al., 2020), construiu um veleiro com vela-asa, colaborando com a *Marine Science and Technology Foundation*. Capaz de suportar condições de mar bastante adversas, tem autonomia para até doze meses. Capacitado com painéis fotovoltaicos, produz a energia necessária ao seu funcionamento, e, com uma massa de cem quilogramas, transporta diversos sensores de monitorização meteo-oceanográfica e transmite os dados recolhidos a cada seis horas, através do *Router-based Unrestricted Digital Internetworking Connectivity Solution* (RUDICS), um modem independente do resto do sistema do veleiro para garantir a continuidade da comunicação.

O seu sucesso, provado pela realização de navegações oceânicas de longa duração (até 34 dias), é notável na sua situação atual, contando já com uma frota de mais de cem unidades⁴, realizando missões de proteção ambiental, controlo das populações de peixe, mapeando o fundo do mar com tecnologia sonar e recolhendo dados meteorológicos.

⁴ Site Notícias CNet, <https://www.cnet.com/news/autonomous-saildrones-are-the-newest-weapon-in-fighting-climate-change/>



Figura 2-4 - Frota SailDrones, São Francisco

Apesar do sucesso dos veleiros autónomos no setor económico, o esforço de investigação desta vertente de veículos autónomos é comparativamente baixo, com um total de setenta e um (71) artigos científicos relacionados com o tópico, de 1985 a 2020 (Yang, Liu, Liu, An, & Xiang, 2021).

A *World/International Robotics Sail Conference* é, neste momento, a maior potenciadora para a elaboração de estudos na área, consistindo numa competição amigável e cooperativa entre veleiros completamente autónomos com até quatro metros de comprimento. A Escola Naval participou na conferência e regata WRSC de 2016 (Fernandes 2016).

2.3 Vela Autónoma em Portugal

O melhor exemplo de vela Autónoma em Portugal é o *FASt* (Alves & Cruz, 2008), criado e desenvolvido pela Faculdade de Engenharia da Universidade do Porto (FEUP), com o intuito de participar no *Microtransat Challenge*, e ter a capacidade para realizar tarefas de observação oceanográfica e de vigilância, cumprindo com os requisitos da *Microtransat*. Este veleiro tem também a valência de poder carregar um *payload* adicional, possibilitando a sua utilização em missões que não apenas as pretendidas inicialmente, dando alguma margem de manobra para os projetos futuros que se viessem a desenvolver.

O veleiro está munido de sensores, entre eles um recetor *Global Positioning System* (GPS), uma agulha magnética, anemómetro, sensor de posição das velas e inclinómetro. Possui ainda sensores de luminosidade, de temperatura e módulo de monitorização de

bateria. O veleiro tem dois modos de operação, sendo eles radio-controlado (RC) ou autónomo, seguindo um planeamento pré-determinado. No módulo de comunicações conta com antena de receção rádio, de *Wi-Fi*, e uma antena satélite para o sistema IRIDIUM (Marques, 2015). A Escola Naval colaborou com os esforços do INESC-TEC no desenvolvimento do FASt, quer apoiando testes experimentais como a travessia Cascais-Sesimbra (2014), quer envolvendo Aspirantes com o FASt nas suas dissertações de Mestrado (Marques, 2015).



Figura 2-5 - Veleiro FASt, durante a travessia Cascais-Sesimbra, apoiada pela Escola Naval

2.4 Vela Autónoma na Marinha Portuguesa

No âmbito do projeto eVentos, têm sido desenvolvidos trabalhos desde 2010, desde um veleiro *Laser* que funcionava em modo RC, adaptado e melhorado para efetuar navegação autónoma e mais tarde culminando no veleiro *Barlavento*, criado em 2016, apto a participar em regatas, tais como a *World Robotics Sailing Championship* em 2016 (Fernandes, 2016). Foi novamente alvo de revisão em 2019, tendo sofrido alterações estruturais e munido de novas velas, assim como novos sensores. Já se puderam observar os frutos deste trabalho com a construção de dois sistemas autónomos por parte da CEOV, que utilizaram as lições aprendidas no projeto do *Barlavento*.



Figura 2-6 - Veleiro Barlavento

3 Atualização da estrutura física

Um veleiro, é, na sua conceção, uma estrutura que será sujeita a diversas forças, sendo o resultado das mesmas a ditar o comportamento do veleiro. Para navegar, um veleiro é afetado pelo vento, através das velas, gerando-se o efeito de escoamento, e as duas forças que o caracterizam, *drag* e *lift*. As velas, o casco e o patilhão devem agir em uniformidade, e apresentar características que potenciem o seu funcionamento. Tendo sido identificada a necessidade de reestruturação do casco, deve ter-se em conta a reavaliação da estrutura das velas e do patilhão, de modo a que o seu equilíbrio se mantenha.

Este capítulo irá abordar, faseadamente, o processo de atualização da componente estrutural do veleiro, abordando a preparação do mesmo, a elevação do convés, o desenho das portas de visita, o mecanismo de controlo dos mastros, elementos de estanqueidade, a criação de uma verga de sinais para albergar os sensores utilizados e reforço das velas.

O modelo existente apresentava os seguintes problemas:

- Portas de visita de difícil acesso e quebras de estanqueidade;
- Componentes eletrónicos montados de forma confusa e pouco *user-friendly*;
- Sistema de controlo de mastro complexo, sujeito a falhas;
- Sensores inexistentes ou pouco fiáveis.
- Falta de espaço para componente



Figura 3-1 - Preparação inicial. Retiradas primeiras camadas de tinta

3.1 Elevação do Convés

A primeira tarefa desempenhada foi o desenho da elevação do casco do veleiro, visto o espaço disponível se mostrar insuficiente para a montagem do novo sistema de controlo das velas e da nova organização da eletrónica, além de passar a garantir maior estanqueidade, com a integração de anteparas estanques, seccionando os compartimentos sobre o convés original em cinco secções. Um à proa, vazio, servindo como reserva de fluabilidade. Duas caixas centradas em cada mastro, para montagem do mecanismo de controlo dos mesmos. Uma caixa entre ambos os mastros para os módulos de eletrónica, com a cablagem a atravessar compartimentos a partir de orifícios no pavimento de cada secção, não penetrando as anteparas estanques. Por fim o acesso ao compartimento do mecanismo do leme foi otimizado, sendo o quinto compartimento superior localizado acima deste, incluindo também nova porta de visita.

Foi utilizada a ferramenta de CAD 3D e simulação *SolidWorks 2021 Premium Edition*.

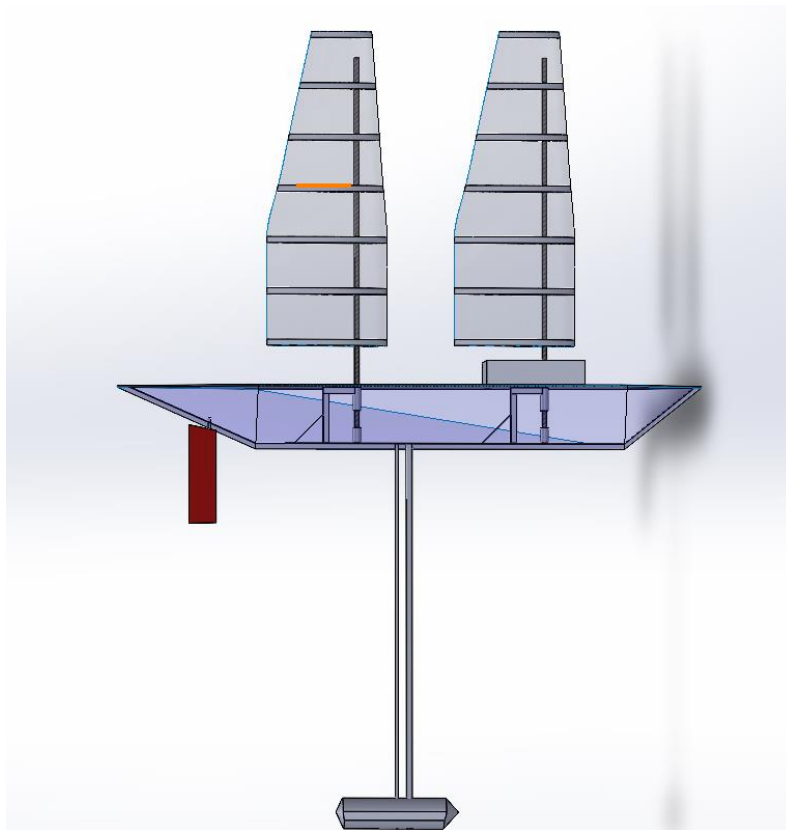


Figura 3-2 - Barlavento sem elevação

Determinou-se que a altura necessária da elevação seria de oitenta milímetros (80 mm), permitindo assim o acomodamento de todo o material sem ter impacto nas velas. Através da ferramenta *Draft*, criou-se uma *Shell*, ou um componente oco, sendo coincidente na borda do veleiro, e com uma inclinação de vinte e dois graus (22°) convergindo, dando assim um perfil aerodinâmico e visualmente apelativo ao casco.

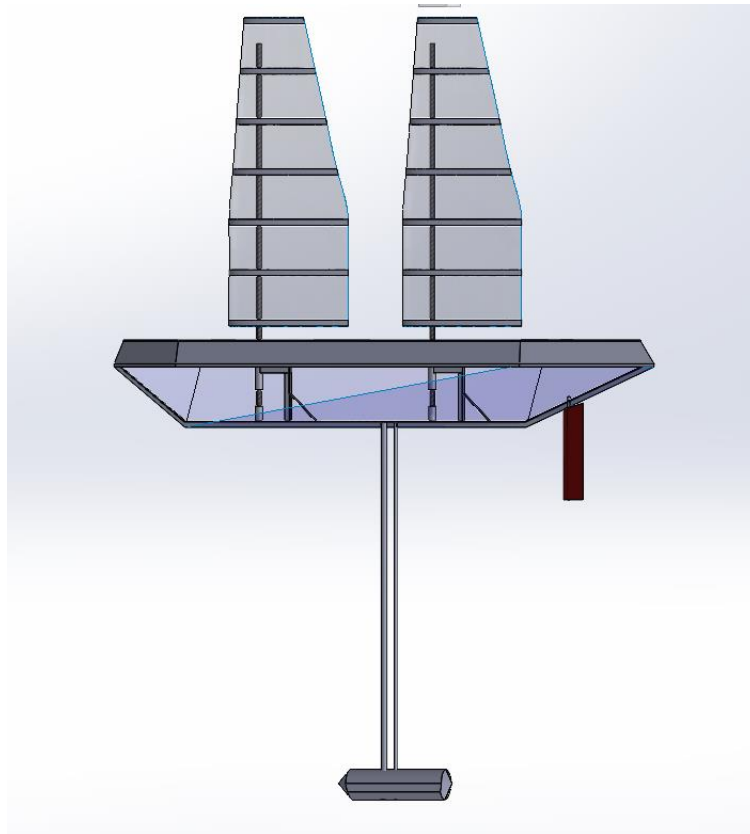


Figura 3-3 - Barlavento com elevação

Numa fase seguinte, determinaram-se os comprimentos necessários a cada porta de visita, estando de acordo com a figura seguinte, sendo as unidades milímetros. A secção de proa não tem associada uma porta de visita, pois não conterá componentes.

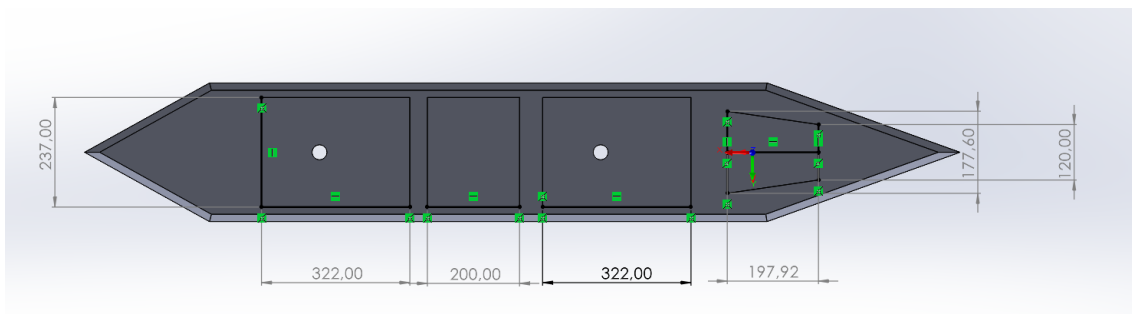


Figura 3-4 - Medidas das portas de visita

O método de fixação das tampas é um compromisso entre estanqueidade e facilidade de acesso, sendo que no último projeto (Santos, 2019), deu-se prioridade à facilidade de acesso. Os resultados demonstraram que a estanqueidade deverá ser a prioridade, e o método adotado foi a utilização de parafusos de quatro milímetros de diâmetro para fixação das tampas. No projeto efetuaram-se as perfurações adequadas, distando sete milímetros e meio da borda de cada porta, com exceção dos vértices, e distribuindo quatro

parafusos por lado, distanciados equitativamente e. após o *Sketch* das mesmas, utilizou-se a função *Extrude Cut*.

Por forma a imprimir a *Shell* em 3D, procedeu-se ao seccionamento da mesma em planos distando duzentos milímetros (200), de ré para vante, sendo que as duas partes de proa, continuando ainda assim demasiado grandes, foram seccionadas a cento e vinte e oito (128 mm) e cento e dezoito milímetros(118 mm), a Bombordo e Estibordo, respetivamente. Para tal, foi utilizada a ferramenta *Split*.

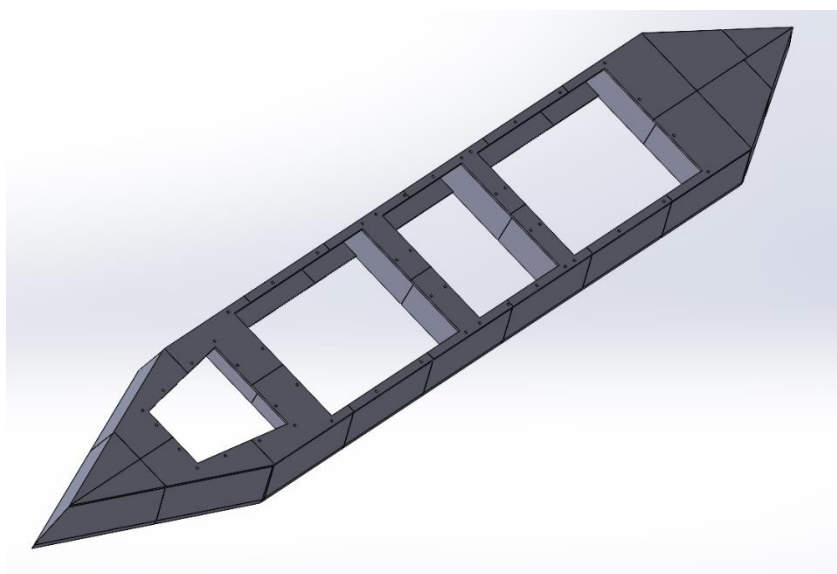


Figura 3-5 - Split e perfurações

Terminado desenho da *Shell*, deu-se início à conversão dos ficheiros de cada parte individual para formato *stereolithography* (STL), importação dos mesmos para *Ultimaker CURA*, por forma a realizar o *Slicing*, ou acomodamento das peças para impressão na máquina, orientando-as da maneira mais eficiente e posterior exportação para formato *G-CODE*⁵, formato esse que é lido pela impressora.

A impressora utilizada foi a *Creality Ender 3*, que possui uma cama com as dimensões 220mm por 220 mm por 250mm., fator relevante para a divisão das peças. O material utilizado foi PLA (*polyactic acid*), pois embora menos resistente que outros materiais, é mais prático e adequado à impressora, sendo o defeito da resistência solucionado através da fibragem posterior.

⁵ O Código G, em inglês G-Code, é a nome dado à linguagem de programação criada a partir da necessidade de máquinas industriais que fazem uso de sistemas de Comando Numérico Computadorizado (CNC).

À medida que as peças foram sendo impressas, recorreu-se a resina de *epoxy* para as unir, formando pares em primeiro lugar e juntando-os de seguida, até se obterem 4 partes principais. Com o convés previamente lixado e pronto para a montagem das novas peças, foram colocadas no lugar devido as quatro componentes, utilizando -se a resina e betume de preenchimento, seguido de uma passagem com lixadeira, para alisar a superfície e compensar certas depressões e elevações do convés, assim como em todas as uniões.

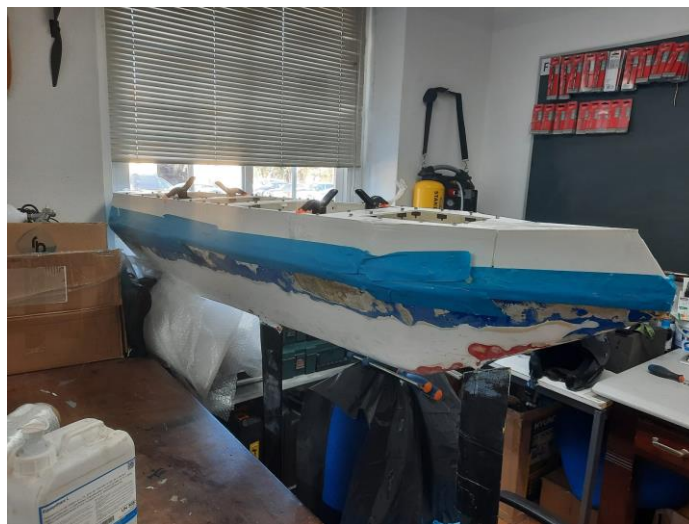


Figura 3-6 - Colagem das novas peças

O passo seguinte foi a aplicação de duas camadas de fibra de vidro, revestindo por completo o casco, e finalizando o processo de união das peças ao casco. Para esse fim voltou a utilizar-se resina de *epoxy*. Começando pelo reparo de uma bolsa de ar resultante de fibragens antigas, tendo sido recortada a fibra nesse local e reforçada com resina e betume de poliéster, seguido do recorte da malha de vidro de modo a ficar acomodada ao formato do casco, cortando-se a fibra de modo a sobrepor-se ligeiramente em todas as arestas. Foi aplicada uma camada de resina preliminar no casco, e outra imediatamente por cima da fibra. Após a primeira camada secar, foram limadas as falhas recorrendo à lixadeira, usando-se uma lixa de grão 320. A segunda camada foi aplicada usando o mesmo método, realçando que durante todo o processo as perfurações dos parafusos se encontravam preenchidas com cola UHU *patafix*. Utilizou-se posteriormente um berbequim para reabrir as entradas, e a fibra em excesso sobre as portas de visita foi recortada.



Figura 3-7 - Aplicação de fibra de vidro

O último passo deste processo consistiu na pintura do casco, primeiro com uma camada de *TopCoat*, à base de *epoxy*, e depois com tinta de acrílico. O acabamento tem o objetivo de assegurar a estanqueidade de toda a estrutura, preenchendo qualquer possível lacuna nas fases anteriores. Por forma a melhorar a estética do veleiro, antes da aplicação do *TopCoat*, utilizou-se novamente betume de poliéster para o alisamento e homogeneidade da superfície, A sua aplicação foi feita por meio de trincha, com alisamento entre cada demão por meio de lixadeira. Após três camadas, considerou-se suficiente e que estava garantida a proteção da fibra e a estanqueidade e para acabamento, procedeu-se ao processo de lixar finamente a superfície, com recurso sucessivo das lixas de grão 320, 400 e 600.



Figura 3-8 - Acabamento com *TopCoat*

3.2 Mecanismo de Controlo dos Mastros

O controlo dos mastros, naturalmente, assume elevada importância, sendo o principal elemento de propulsão do veleiro, e, caso necessário, também de controlo da trajetória, não sendo imperativo a existência de um leme para a manobra. O projeto das caixas para o mastro foi realizado tendo em vista a utilização de um mecanismo *WormDrive*. Estes mecanismos apresentam vantagens quando é necessário um rácio de redução de velocidade notório entre eixos que não se intercetem. Este mecanismo consiste numa engrenagem de maior diâmetro e num parafuso unido este por meio de dentes. À medida que o parafuso gira, também a engrenagem gira. O tamanho do conjunto é geralmente baseado na distância entre o centro do parafuso e da engrenagem (Oberger, 1920).

No processo de montagem e aprontamento do veleiro para navegação, a limitação do tempo revelou-se mais impactante que o previsto inicialmente, e assim, ficou estipulado que a fase de implementação deste sistema de controlo seria abandonada. Foi adotada uma solução provisória, que permitiu o seu funcionamento básico por meio de molas, abdicando do seu controlo e monitorização.

Com a implementação das novas portas de visita, surgiu a necessidade de adotar um sistema que permitisse o suporte dos mastros, e, simultaneamente, o seu desacoplamento sem a necessidade de abrir as portas de visita. Para esse fim, foram adaptados dois veios de alumínio com 300mm de altura, com 20mm de diâmetro exterior e 16mm de diâmetro interior. Nestes, encaixam os mastros, e, por estrangulamento, passam a girar em união. Estes veios são de alumínio, e têm soldada uma anilha do mesmo material, com 60mm de diâmetro, utilizada para segurar o veio no eixo vertical.

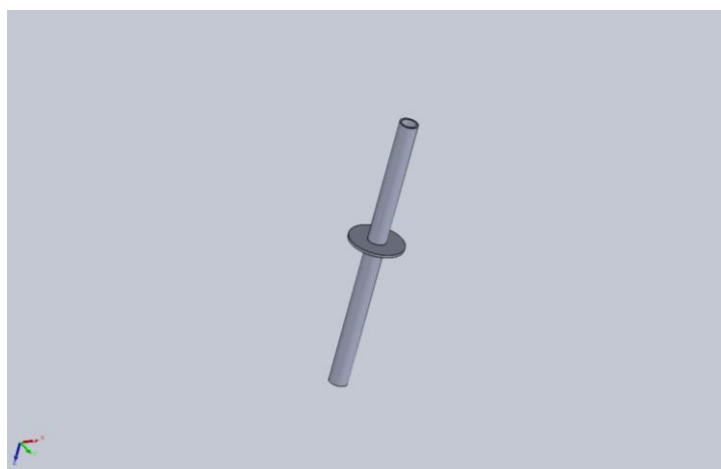


Figura 3-9 - Veio dos mastros

A anilha encontra-se ao nível do convés original, encaixando numa peça que, aparafusada ao convés permite a sua rotação, mantendo-a alinhada e segura verticalmente. Esta peça foi desenhada em *SolidWorks* e impressa em PLA.



Figura 3-10 - Apoio dos Veios ao convés

Este veio foi desenhado para assentar na quilha, assim como os mastros, mas verificou-se que a enora e a carlinga não coincidiam perfeitamente, o que impedia que os veios assentassem na quilha. Foi desenhado e impresso um apoio que assegurasse o posicionamento correto dos veios, e, por meio de estrangulamento, a sua orientação vertical e suporte efetivo dos mastros. Esta peça, constituída por duas metades separadas, inclui furações para parafusos M4, iguais aos utilizados nas portas de visita. A separação das metades foi feita retirando 3mm de material do seu plano central, gerando uma folga que permitiu o estrangulamento. A parte superior detém diâmetro interior superior que envolve o tubo que constitui a enora.

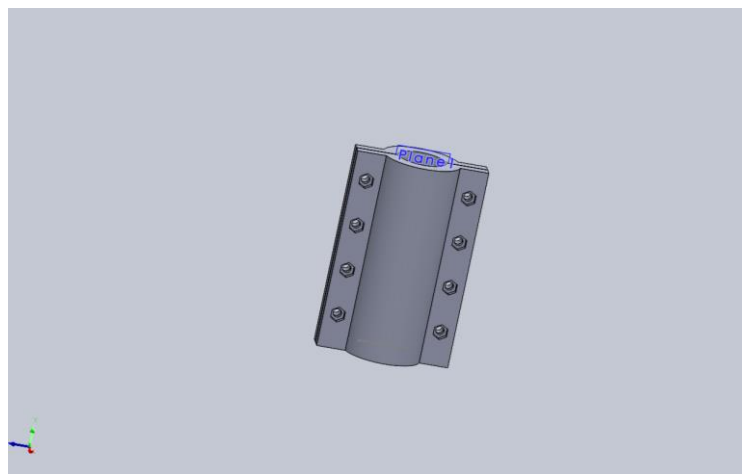


Figura 3-11 - Suporte dos veios

Similarmente, uma peça com a mesma função foi desenhada e impressa. Este *fix* dos mastros une o veio aos mastros, sendo a peça fundamental para a retirada das velas prévia à abertura das portas de visitas. É colocada no topo dos veios, sobre dois cortes longitudinais simétricos efetuados na boca superior de cada veio. Imperativamente os mastros devem girar em uníssono com os veios, e, assim que se age nos veios, a mesma ação deve verificar-se nos mastros. Assim, esta é uma peça que sofre uma grande pressão, e, nas primeiras impressões e testes, notou-se que se quebrava facilmente. Para colmatar esta limitação aumentou-se a densidade de impressão para 30%, e a peça foi colocada a imprimir num plano horizontal, de maneira que as camadas da impressão ficassem paralelas à aplicação da força, garantido o seu reforço. Não se verificaram novas quebras nos componentes. Por forma a garantir o aperto dos mastros, foram recortados dois lanhos paralelos no topo do veio, recorrendo a uma rebarbadora.

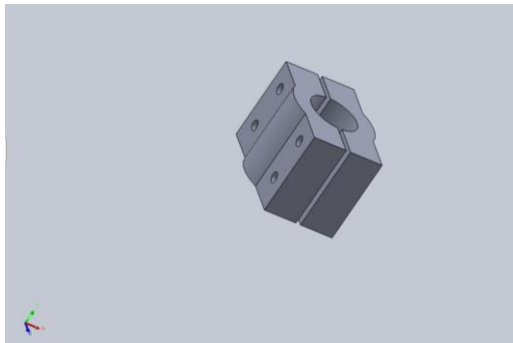


Figura 3-12 -Fix dos veios nos mastros

Com a implementação de todos os componentes, tornou-se possível a fixação dos mastros no casco, permitindo a sua rotação e a sua união aos veios. Para o mecanismo de controlo dos mastros utilizou-se um simples eixo, desenhado e impresso em PLA, unindo-se ao veio com o mesmo princípio do *fix* das velas.

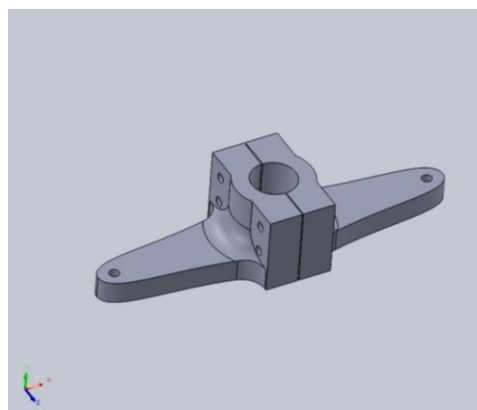


Figura 3-13 - Tirante do mastro

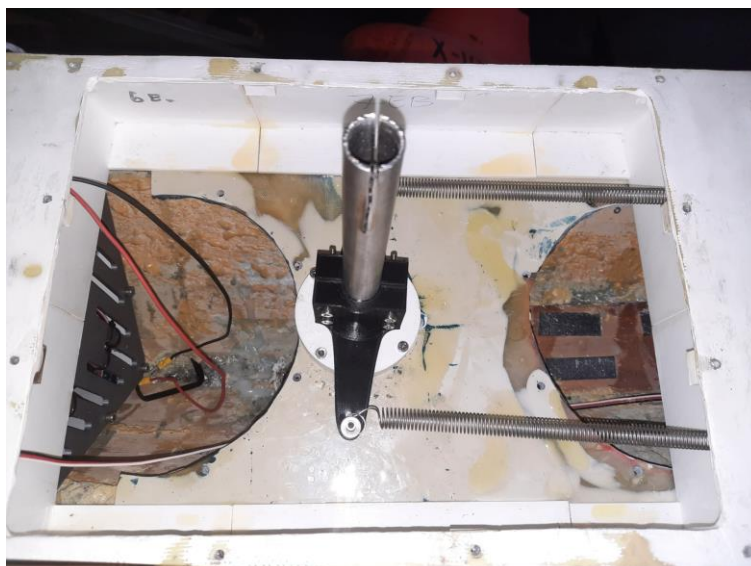


Figura 3-14 - Tirante do mastro colocado

3.3 Atualização das portas de visita

O veleiro deve contar com portas de visita, tendo sido já determinado que seriam quatro portas, de encaixe feito por meio de parafusos de 4mm (M4). A estanqueidade será assegurada por uma camada de espuma colada na face inferior das portas, que ao ser espalmada com o aperto dos parafusos, tornará as tampas estanques. O veleiro irá contar com duas portas de visita que abraçam os mastros, sendo perfuradas pelo mesmo, e duas portas nas quais não se comprometerá a sua integridade, na casa do leme e no compartimento da eletrônica.

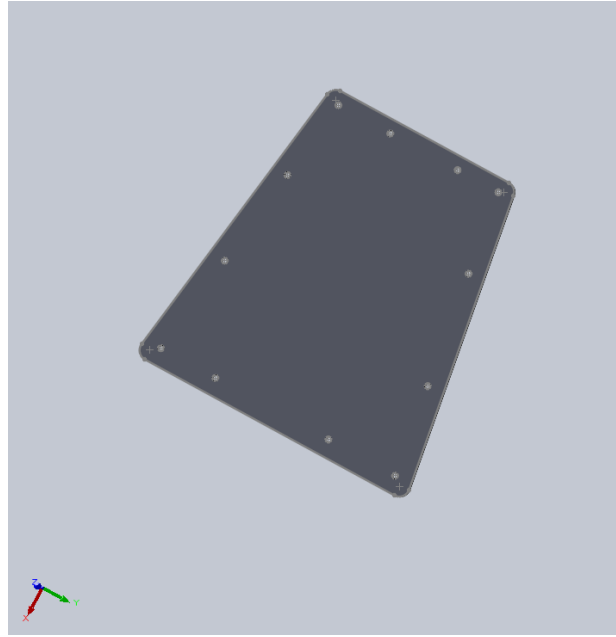


Figura 3-15 - Porta a ré

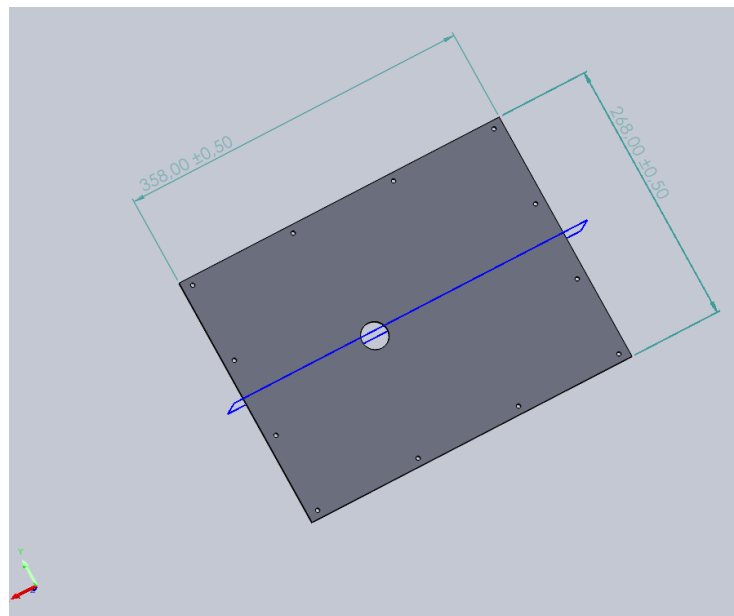


Figura 3-16 - Porta do mastro a ré

Para esta decisão, levou-se em conta o estudo do catamarã a ser desenvolvido pela CEOV para uso do Instituto Hidrográfico, que utiliza o mesmo sistema nas suas portas de visitas. O método revela-se algo moroso na montagem, e a utilização de uma parafusadora permite reduzir esse esforço. Ao contrário do que acontece com as portas do catamarã, os parafusos do veleiro não estão seguros ao mesmo, sendo colocados quando se apronta o mesmo, e são as porcas que estão integradas no veleiro.



Figura 3-17 - Porta de visita do catamarã da CEOV

O sistema aqui observado apresenta os parafusos incrustados no convés, orientados na vertical e bastante salientes, o que apresenta a vantagem de facilitar o encaixe das portas de visita, no entanto, torna-se visualmente desagradável, além de dificultar o trabalho na superfície do convés, reduzindo a área utilizável.

De maneira a poder efetuar o encaixe dos parafusos numa orientação descendente, foi desenhada e impressa uma pequena caixa, onde foram incrustadas as porcas dos parafusos numa ranhura hexagonal, com recurso ao ferro de solda. O material, PLA, devido à temperatura aplicada, tornou-se moldável no seu interior, e, após o arrefecimento, ficaram as porcas efetivamente seguras na sua posição.

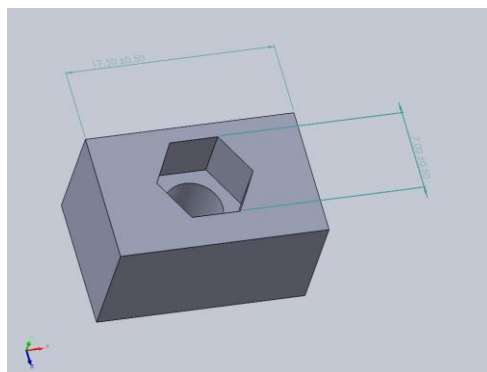


Figura 3-18 - Encaixe das porcas

Estas caixas foram unidas à estrutura utilizando novamente resina de *epoxy*. O desenho destas peças apresentou um grau de sucesso variável. No caso de as peças terem ficado

efetivamente coladas, o que sucedeu em quase todas, o design revelou-se suficiente e adequado, tendo sido possível garantir o aperto e a estanqueidade do veleiro ultimamente. Não obstante, num número de ocasiões distintas, surgiu a necessidade de substituir uma ou mais destas caixas, devido a várias razões, entre elas:

- Esforço exagerado num parafuso singular, causando a libertação da porca no interior da caixa e a sua rotação livre;
- Esforço num ângulo não ideal, o que levou à quebra da caixa;
- Necessárias várias tentativas para colar certas peças.



Figura 3-19 - Colagem das caixas das porcas

As portas de visita inicialmente desenvolvidas por Fernandes (Fernandes, 2016), eram montadas usando o mesmo princípio de aparafusamento, no entanto, não só eram de menores dimensões, (180mm por 160mm, ao invés que no novo desenho têm 358mm por 268mm), como também não apresentavam enora. Verificou-se ainda que o material em PLA, em que foram impressas, não era adequado para suportar as altas temperaturas a que o veleiro foi sujeito, tendo sofrido deformações (Santos, 2019).

Com vista a solucionar este problema, foi utilizado liga de alumínio 7075, na manufatura das novas portas de visita, através da colaboração com a empresa *Ricardo & Barbosa*. Os desenhos foram elaborados em *SolidWorks Drawings*, as cotas inseridas e o material foi recortado a laser nas suas instalações.

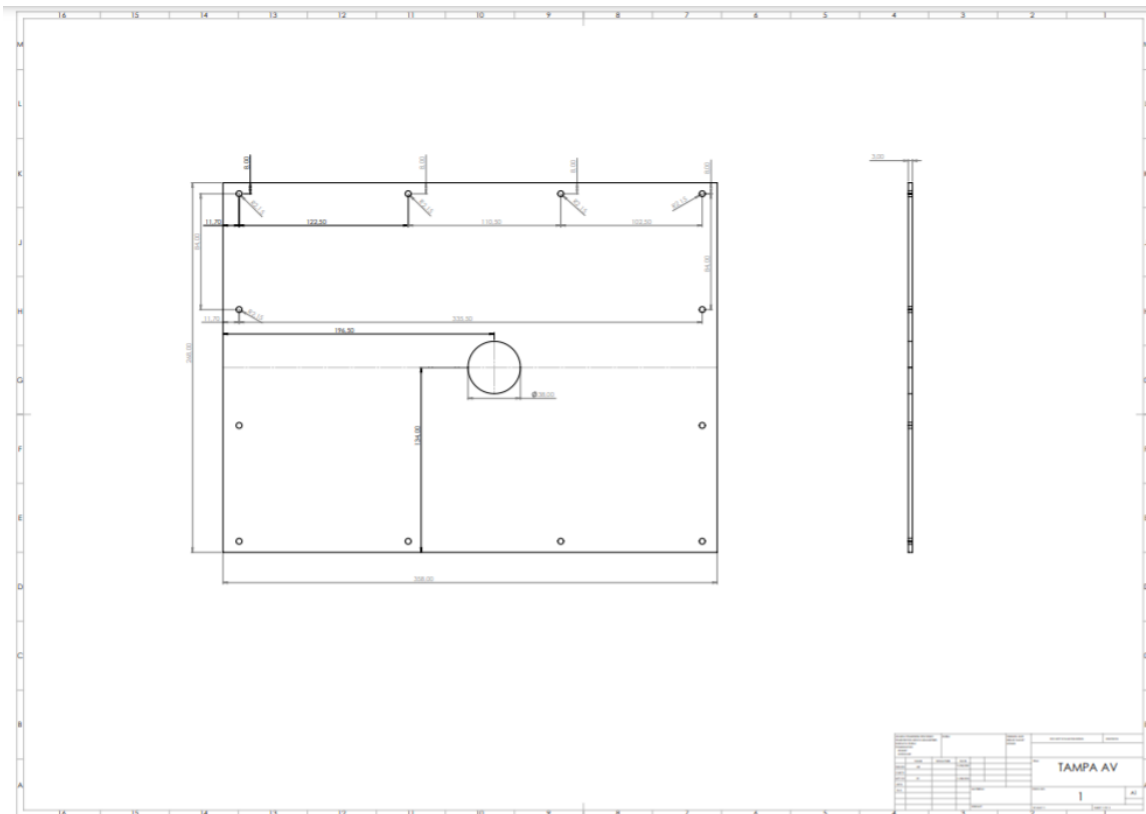


Figura 3-20 - Desenho 2D das portas de visita

Por forma a garantir a estanqueidade destas portas, foi utilizado um retentor de fluidos, adquirido no mercado, consistindo num perfil de metal em L, revestido a borracha e com uma mola autoajustável. Uma aba de borracha principal onde está integrada a mola e uma aba menor adicional para maior eficácia. Este perfil é denominado de TC.

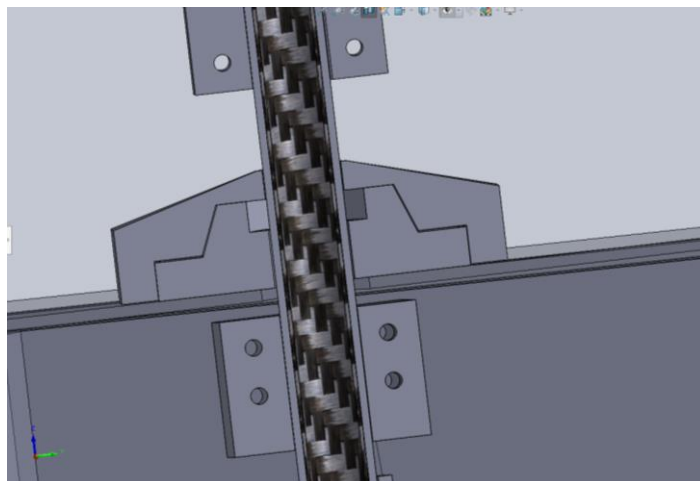


Figura 3-21 - Esquema peças do encaixe do retentor

TC



Figura 3-22 - Perfil de retentor TC

O diâmetro interior é de 20mm, abraçando justamente o veio, ainda que permitindo a sua rotação, 30mm de diâmetro exterior e 7mm de altura. Partindo destas medidas, foi desenhado uma caixa para o retentor, que assentando na enora da porta de visita, seria aparafusada a esta, contanto com uma camada de espuma autocolante na base, fornece estanqueidade com o mesmo princípio das portas de visita em si, selando por completo passagem do mastro. Para garantir uma maior eficácia ainda da estanqueidade, uma peça de encaixe sobre esta caixa foi também desenhada e impressa, sendo colocada sob stress na caixa do retentor e apresentando um perfil octogonal com faces descendentes, de maneira a que a água escorra do veio para fora, adicionando uma camada de proteção e redundância.

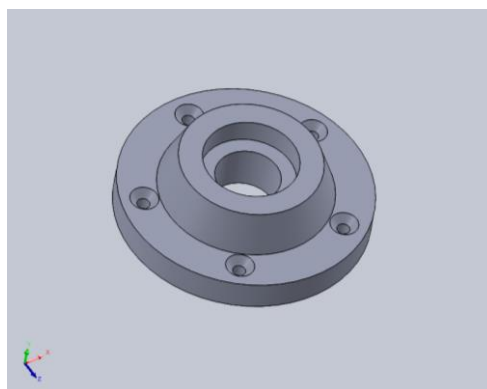


Figura 3-23 - Caixa do retentor

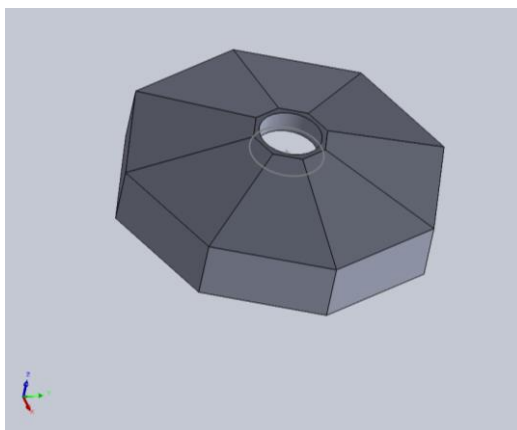


Figura 3-24 - Tampa da caixa do retentor

O último passo da atualização das portas de visita consistiu na furação de ambas as portas das caixas dos mastros com o objetivo da passagem da cablagem dos sensores. Utilizou-se para o efeito um bucim para cabos, que consiste numa rosca central e duas peças que nela enroscam. A superior possui nova rosca com anilha de espuma e cabeça que, ao apertar, ajusta a anilha ao cabo que o atravessa. A peça inferior é um O-ring simples que é esmagado contra a face inferior da porta.

No bucim em que passa o cabo do anemómetro e o da antena do módulo LoRa, por serem cabos mais finos, o bucim não gera aperto suficiente, tendo sido reforçado com o uso de cola quente na saída dos cabos. Ambos os bucins levaram um rebordo de cola quente na base para assegurar a estanqueidade, devido à ausência de anilha deformável.

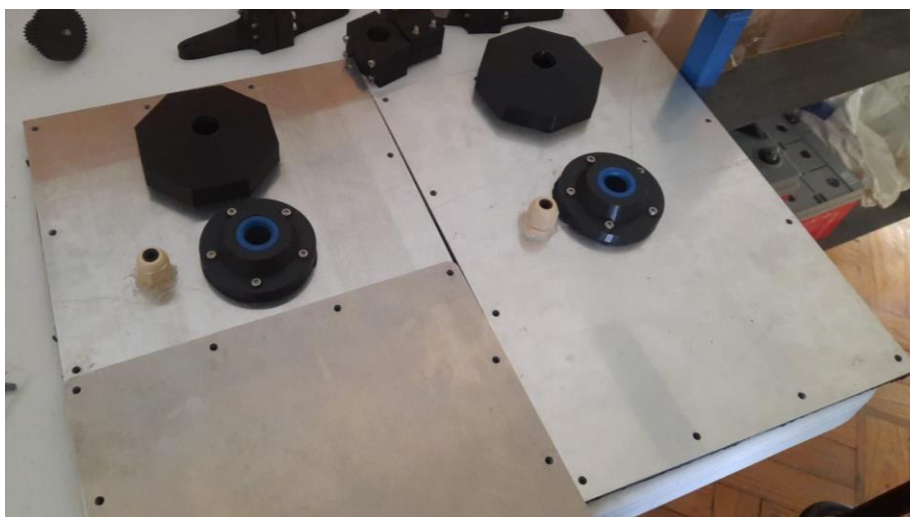


Figura 3-25 - Aspeto final das portas de visita

3.4 Criação de uma verga de sinais

O posicionamento dos sensores num veleiro deve ser tal que permita a obtenção dos dados o mais fielmente possível. O anemómetro merece especial relevo, devendo encontrar-se o mais elevado possível na estrutura do veleiro, de modo que sofra o mínimo de interferência na leitura pelas obras-vivas do veleiro. Deste modo, decidiu colocar-se o anemómetro na posição mais dianteira da verga, além de mais elevado que a bússola.

O material utilizado foi um perfil do tipo T, em alumínio, com 3 mm de espessura e 30mm por 30mm de lado. O material foi requisitado ao Serviço de Apoio aos Navios, e foram utilizadas as instalações do mesmo para o corte e soldadura do material. Para o encaixe da bússola, era requerido um tubo de alumínio com 25mm de diâmetro exterior, e diâmetro interior não inferior a 10mm. Devido à inexistência de um tubo com estas características, foi utilizado um varão com o diâmetro exterior adequado, e furado o seu interior recorrendo a broca de 10mm. Foi também efetuado um furo lateral para a passagem do cabo do sensor.

O suporte do anemómetro requereu um varão de 12mm de diâmetro exterior, tendo sido necessário tornear um varão de 20mm até ao diâmetro pretendido. De seguida, foi cortado esse mesmo varão, e as duas partes cortadas a direito numa ponta, e na outra, com um ângulo de 45 graus. Isto permitiu a soldadura das duas partes num ângulo reto, visto que o encaixe do anemómetro requeria um plano horizontal.

O ajuste dos instrumentos ao respetivo suporte foi feito através dos mecanismos já presentes nos mesmos.

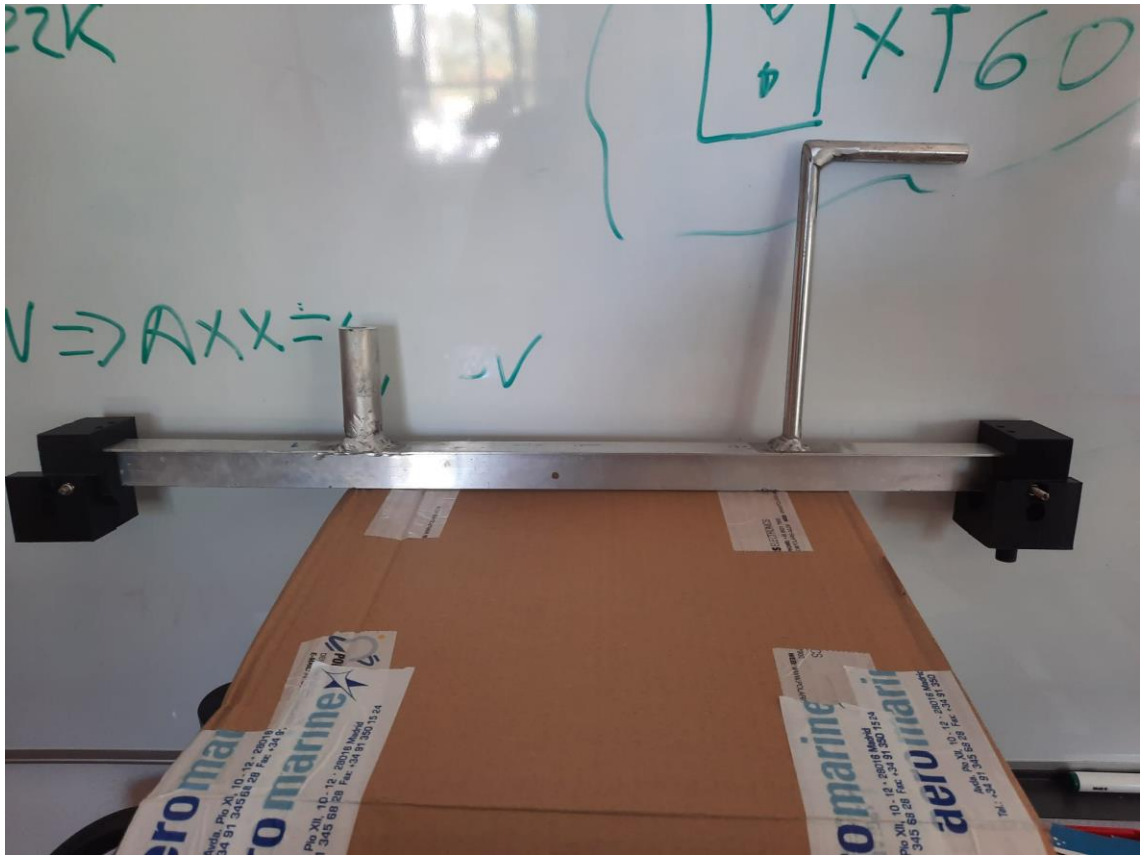


Figura 3-26 - Verga de sinais

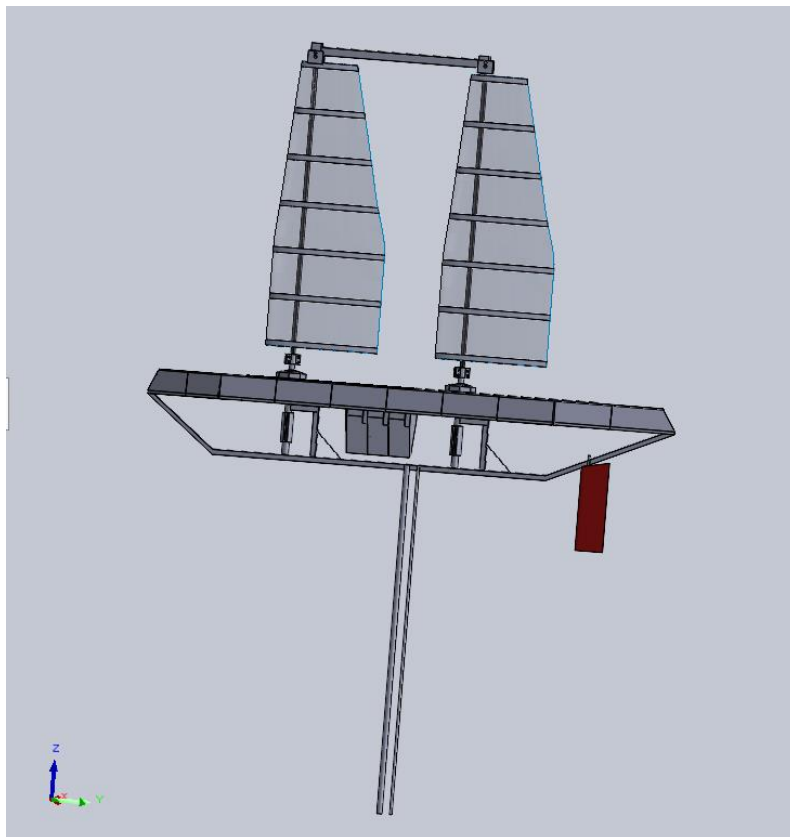


Figura 3-27 - Verga de sinais no topo dos mastros

Desenharam-se dois encaixes para a verga de sinais e impressas em PLA. Estes encaixes são constituídos cada um por quatro peças separadas:

- Encaixe em formato cilíndrico, a ser colado à face interior do galope dos mastros, com um anel mais largo superior onde liga às restantes peças;
- Peça com cavidade em T, para inserção das extremidades da verga;
- Ligadores simétricos que apertam por meio de parafusos, unindo o encaixe da verga e o do mastro, com espaço para rotação do anel superior deste, permitindo a rotação livre dos mastros.

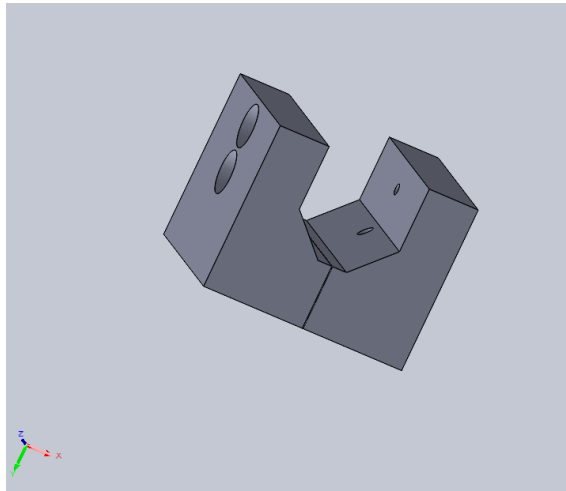


Figura 3-28 - Ligadores

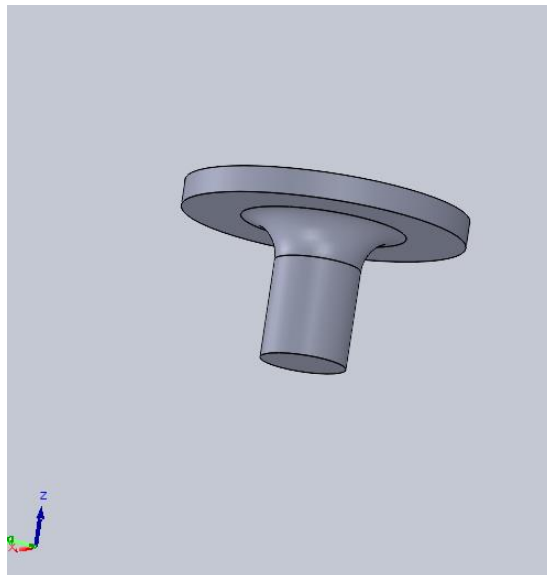


Figura 3-29 - Peça de união ao mastro

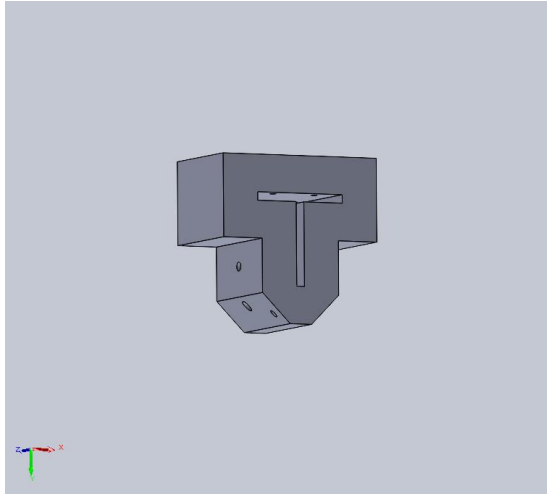


Figura 3-30 - Peça de união à verga

4 Atualização da componente eletrónica

Terminada a componente estrutural do veleiro, deu-se início à substituição de toda a eletrónica. Nos trabalhos anteriores não havia um espaço dedicado unicamente à eletrónica, estando os seus componentes espalhados pelo interior, o que gerava problemas de acesso, dificultando o processo de efetuar quaisquer alterações que fossem necessárias, além de estarem unidos ao casco de maneira muito precária. O principal objetivo da sua atualização foi facilitar o acesso, criando três módulos distintos, ou seja, três caixas desenhadas e impressas em PLA. Estas permitem a organização dos componentes, o controlo do estado dos mesmos, e a fácil remoção das caixas do seu berço permite efetuar reparações ou alterações expeditas.

A navegação oceânica caracteriza-se pelos seus longos trajetos, e períodos extensos, em que a embarcação se encontra totalmente desligada de terra, sem hipótese de abastecimento até atingir o seu destino. Com o objetivo em mente de efetuar uma navegação oceânica, a utilização de painéis solares torna-se também algo essencial a incluir na estrutura do veleiro.

Os sensores foram substituídos, com especial relevo no anemómetro, sensor absolutamente essencial num veleiro. O anemómetro e catavento utilizado foi o *Clipper Wired Tactical Wind Mast Sensor*, com *output* NMEA0183, com uma taxa de envio de dados de dez transmissões por segundo. O recetor de GPS, uma LT1000 NRU, fornece muito mais informação, tais como o *roll*, *pitch*, hora e data no Universal Time (UT), rumo, proa, abatimento, velocidade verdadeira, pressão e temperatura.

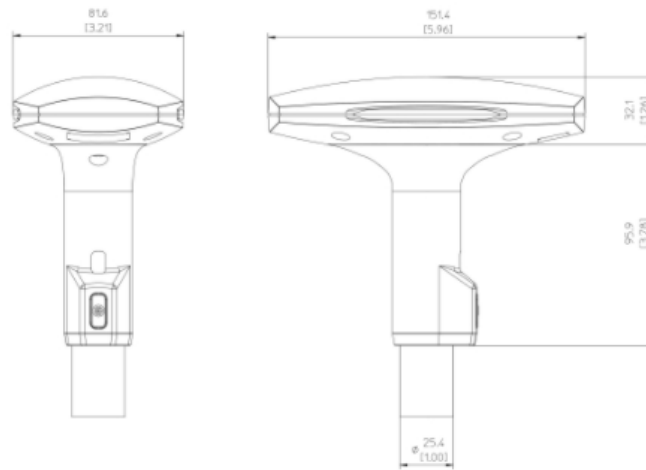


Figura 4-1 - LT1000 NRU



Figura 4-2 - Clipper Tactical Wind Mast Sensor - MHU-TACT

O microcontrolador utilizado foi uma placa *Arduíno* Mega 2560, por ser de fácil utilização e versátil nas suas aplicações. Como atuadores o veleiro conta apenas com um servo do leme, embora esteja capacitado também com um controlador *Veyron* para a integração de motores para controlo das velas.

As caixas foram desenhadas e impressas com recurso a PLA, sendo o design adaptado às medidas específicas do compartimento, por forma a caberem três caixas iguais, sendo que uma será fechada com tampa lateral (módulo do controlador solar), e as duas outras caixas encostam às costas da primeira e subsequentemente, aumentando o tamanho viável

das mesmas. A união das caixas à tampa lateral, assim como à superior foi feita por meio de parafusos M4, de cabeça oval, tal como nas portas de visita.

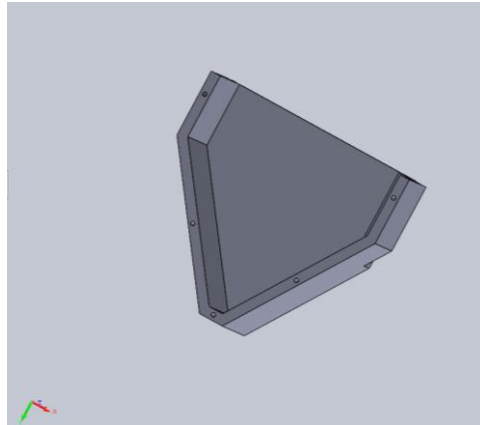


Figura 4-3- Caixas da eletrônica

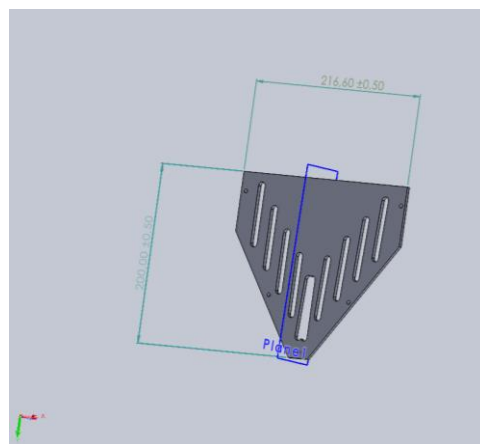


Figura 4-4 - Tampa das caixas de eletrônica

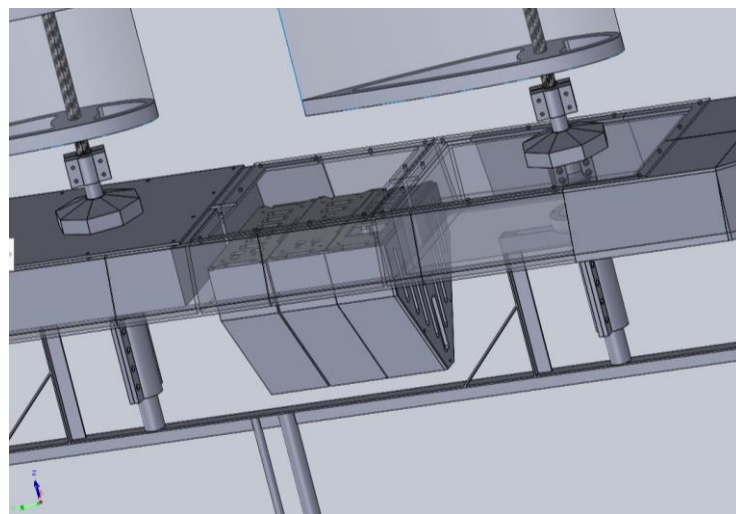


Figura 4-5 - Disposição dos módulos no casco do veleiro

4.1 Módulo de Controlador Solar

A adição dos painéis solares ao sistema do veleiro adicionou um novo grau de complexidade ao mesmo, tornando-se claro que se deve monitorizar o desempenho energético em tempo real do veleiro, assegurando o seu bom funcionamento e permitindo prever possíveis falhas.

O módulo em si irá efetuar a ligação entre as fontes energéticas, painéis solares, que alimentam as baterias e o restante sistema eletrónico, assim como das baterias em si para o veleiro, sempre que o rendimento dos painéis se encontre abaixo da carga das baterias. Por forma a monitorizar o desempenho dos painéis solares e da bateria, foram também incluídos sensores de corrente, nomeadamente na saída dos painéis solares, na entrada da bateria, e na saída da corrente para distribuição aos restantes módulos. Estes sensores comunicam diretamente com o microcontrolador, através das entradas para sinal analógico.

Com recurso ao programa *QElectrotech*, versão 0.8 para Windows, realizou-se o seguinte esquema:

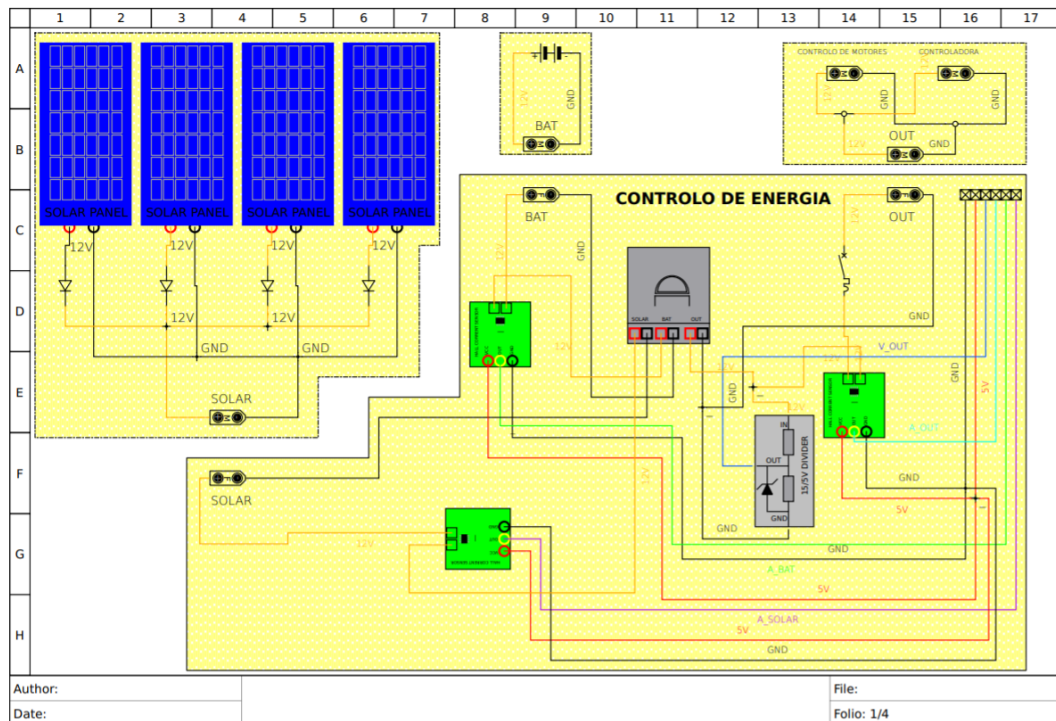


Figura 4-6 - Esquema Controlador Solar

Após finalizar a esquemática, desenhou-se e imprimiu-se a tampa superior da caixa, onde passam as ligações para os módulos adjacentes. Neste módulo a tampa necessitava de pelo menos três fichas XT-90, para a alimentação, uma régua de seis pins para soldar os cabos de sinal e alimentação dos sensores de corrente e um interruptor geral para ligar e desligar todos os sistemas.

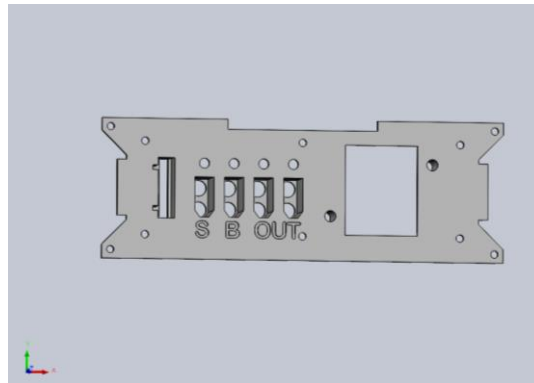


Figura 4-7 - Tampa Controlador Solar

O aspeto final será, portanto, o seguinte:

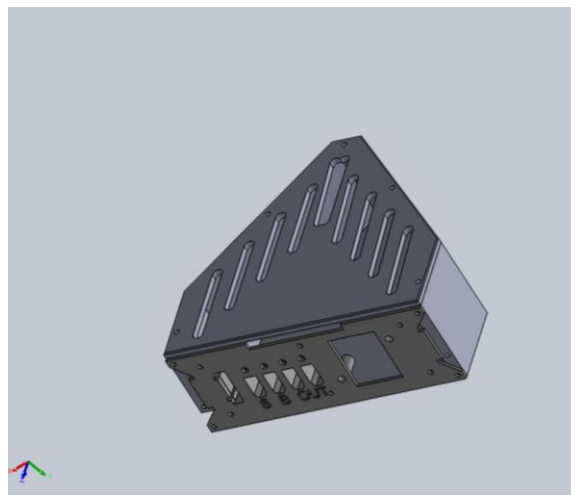


Figura 4-8 - Módulo completo

O controlador solar protege a bateria de sobrecargas geradas pelos painéis, e aumenta a eficiência da mesma. Possui também indicadores auditivos e visuais da carga da bateria, embora num contexto de navegação estes não sejam úteis, daí a utilização dos sensores de corrente. O controlador solar utilizado no sistema é um controlador do tipo PWM, que não tem uma eficiência tão boa quanto os controladores *Maximum Power Point Tracker* (MPPT), contudo, era o componente disponível para integrar no sistema e foi assumido que o desempenho estaria condicionado.



Figura 4-9 - Controlador Solar

As baterias utilizadas foram as NP7-12 YUASA, de ácido-chumbo e com tensão nominal de 12 V, com capacidade de 7 Ah, e de dimensões 151x65x97.5mm. Ligadas em paralelo ao controlador, alimentam ou são alimentadas consoante o rendimento dos painéis.



Figura 4-10 – Bateria utilizada

Os painéis solares, por sua vez, devem encaixar bem nas velas, e apresentar um rendimento adequado ao gasto energético do veleiro. Para determinar as necessidades energéticas do veleiro, partiu-se do consumo energético calculado por Santos (Santos, 2019), de 4,86W e efetuou-se novo cálculo, tendo em conta os novos componentes. Com a nova arquitetura do sistema:

Componente	Corrente (mA)	Tensão (V)	Taxa do ciclo ativo à potência máxima	Potência (W)
LoRa6500Pro	<1800 (Tx)	12	5%	1,8
LT-1000	-	12	100%	<1
Anemómetro	-	12	100%	<1
Arduíno	65	5	100%	0,325
Servo leme	285	5	50%	0,7125
LED's	300	12	25%	0,9
Total				<5,7375

Tabela 1- Consumo energético estimado

Foram adquiridos painéis solares com a capacidade de produzir até 10 W, a 12 V, e, com as dimensões de 440x190x3mm, sendo possível acomodar dois em cada vela, um de cada bordo. Por forma a testar a sua eficiência, foram montados os quatro painéis solares, ligados ao controlador solar, e este por sua vez à bateria. Foram testadas diversas configurações dos painéis, além de se ter considerado necessário a utilização de díodos em série com os painéis, sacrificando uma ligeira queda da tensão (0,3V), pela garantia que não haveria consumo de carga por parte destes durante a noite ou períodos sem luz solar.



Figura 4-11 - Testagem dos painéis e controlador solar

Utilizou-se um wattímetro e voltímetro para medir potências e tensões com os painéis em várias configurações de ligações, ângulos e número de painéis expostos à luz solar. Simulou-se a carga do sistema utilizando duas resistências em série de 15 Ohms e 5 W.

Obteve-se 4,8W de carga simulada, descontando os 5 a 10 % de erro associados ao valor das resistências. Esta carga foi suportada estavelmente, mesmo com percentagens de exposição variadas, correspondendo ao número de painéis tapados.

% Exposição	Paralelo	Série	Mesmo bordo em série, Bordos diferentes em paralelo
25%	11,6V	-	-
50%	11,74V / 4,6 W	-	-
75%	12,4V / 5,2 W	22V / 5,67W	-
100%	12,6V / 6,3 W	24,2V / 5,9 W	12,66V / 5.0 W

Tabela 2 – Resultados dos testes aos painéis solares

Ficou estipulado que seria adotada uma organização com os quatro painéis em paralelo, para otimização da produção a 12V.

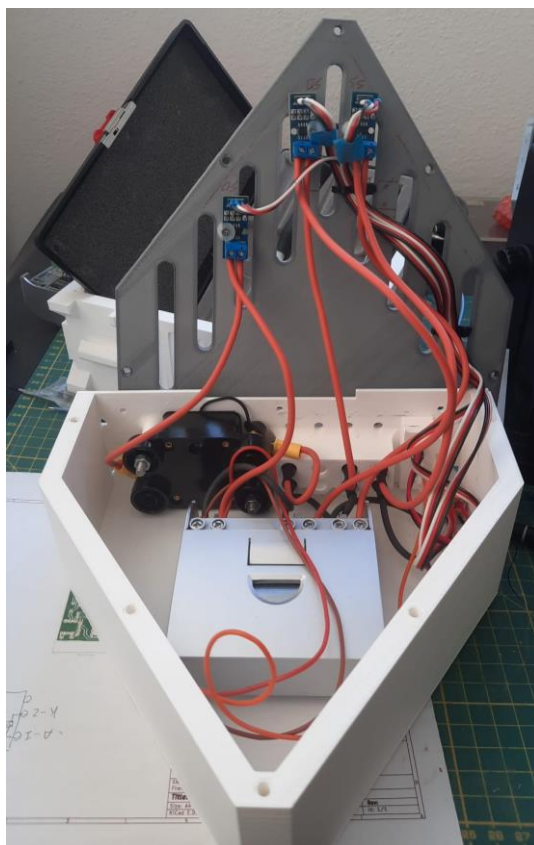


Figura 4-12 - Módulo finalizado

4.2 Módulo do Microcontrolador

Entre a gama de microcontroladores disponíveis, decidiu utilizar-se uma placa Arduíno Mega 2560, ao invés do PLC previsto inicialmente. Esta placa, de fácil manuseio, baixo consumo energético e com uma capacidade de processamento adequado

às tarefas necessárias, além de ser fisicamente diminuta, permitiu uma maior flexibilidade na sua montagem no módulo, otimizando a organização dentro do mesmo, visto que o sensor de comunicação ocupou grande parte do interior. As suas características são as que se seguem:

Microcontrolador	ATmega2560
Tensão de Funcionamento	5V
Limites de Alimentação	6-20V
Pins I/O Digitais	54 (dos quais 15 têm output PWM)
Pins Analógicos	16
Corrente DC por I/O Pin	20 mA
Corrente DC dos Pins 3.3V	50 mA
Memória Flash	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Comprimento	101.52mm
Largura	53.3mm
Massa	37g

Tabela 3 - Características Arduino Mega 2560

Para as comunicações com a *Ground Control Station*, utilizou-se um módulo de comunicações LoRa6500Pro. Este modelo faz parte de uma série de módulos de comunicação *wireless*, que utiliza o *Frequency-hopping spread spectrum* (FHSS), ou seja, um método de transmissão de sinais rádio, alterando rapidamente a frequência portadora de maneira a obter alta sensibilidade. (Torrieri, 2018).



Figura 4-13 - Módulo LoRa

Este módulo, munido com uma antena, permite uma ligação com outro módulo igual em terra, associado à GCS, permite a monitorização e o controlo do veleiro em tempo real. Através de uma configuração básica, usando os seguintes parâmetros, podemos calibrar ambos os módulos, passando estes a estar em contacto direto:



Figura 4-14 - Calibração do Módulo LoRa

O módulo comunica diretamente com o *Arduíno*, através dos pins de Tx/Rx:

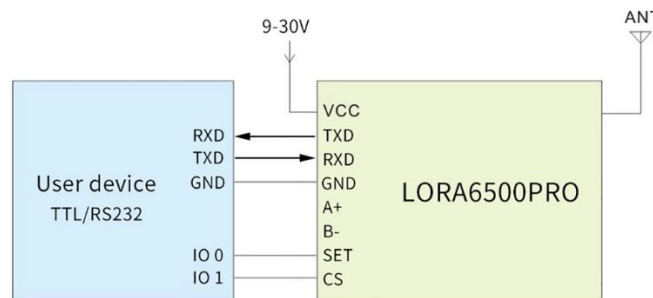


Figura 4-15 - Esquema ligações LoRa

De realçar também a tabela de parâmetros de funcionamento do módulo, nomeadamente o seu consumo energético, o mais elevado por uma grande margem de todos os componentes do veleiro, com 1,8A no caso da transmissão de sinais:

Parameters	Min.	Typ.	Max.	Unit	Condition
Working condition					
Voltage range	9	12	30	V	
Operating Temperature	-40	25	+85	°C	
Current consumption					
Rx current		<18		mA	@12V @TTL level @ 37dBm
Tx current		< 1.8		A	
Sleep current		< 7		mA	
RF parameters					
Frequency range	414.92	433.92	453.92	MHZ	@433MHZ
	470.92	490.92	509.92	MHZ	@490MHZ
Data rate	91	656	17353	bps	LoRaTM
Output power	28	/	+37	dBm	@12V,@433MHZ / 490MHz
Sensitivity		-139		dBm	91bps

Figura 4-16 - Parâmetros LoRa

Além do microcontrolador e do LoRa, o módulo inclui ainda os conversores de sinal RS232⁶, em que operam a bússola LT1000 e o anemómetro, para *Transistor to Transistor Logic* (TTL), em que opera o microcontrolador.

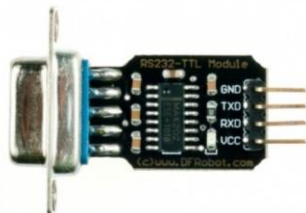


Figura 4-17 - Conversor RS232 para TTL

Surge assim o desenho da tampa superior deste módulo, que deverá cumprir com os seguintes requisitos:

- Alimentação a 12V, por meio de ficha XT60;
- Fusível em paralelo com um LED vermelho *standard*, assegurando que não há sobrecarga do sistema;
- Seis terminais de encaixe com seis saídas
 1. Controlo do servo do leme e motores das velas;

⁶ Protocolo padrão de troca série de dados entre um terminal e um comunicador

2. Alimentação e sinal da bússola LT1000;
3. Alimentação e sinal do anemómetro;
4. Sensores de corrente do módulo do controlador solar;
5. Equipamento de comunicação satélite;
6. Eventual *payload* ou novo equipamento.

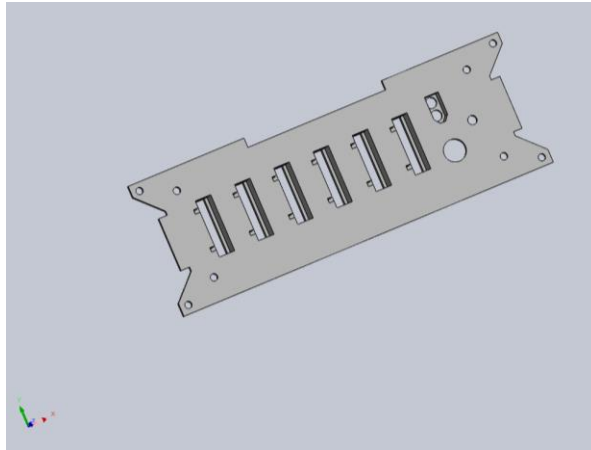


Figura 4-18 - Tampa módulo controlador

A alimentação do módulo, a partir da ficha XT60, passa por um fusível de 3A, em paralelo com um LED vermelho, que acenderá caso ocorra a queima do fusível, poupando-se assim o consumo contínuo por parte do mesmo. De seguida, por meio de ligadores rápidos de cinco entradas, alimenta-se diretamente a 12V o microcontrolador, o módulo LoRa, o anemómetro e a bússola. Destes, apenas o *Arduíno* não funciona a 12V, no entanto conta na sua disposição com um conversor de tensão, podendo ser alimentado deste modo. Os conversores de sinal serão por sua vez alimentados a partir do pin de 5V do *Arduíno*. Com o programa *QEletroTech*, elaborou-se o seguinte esquema:

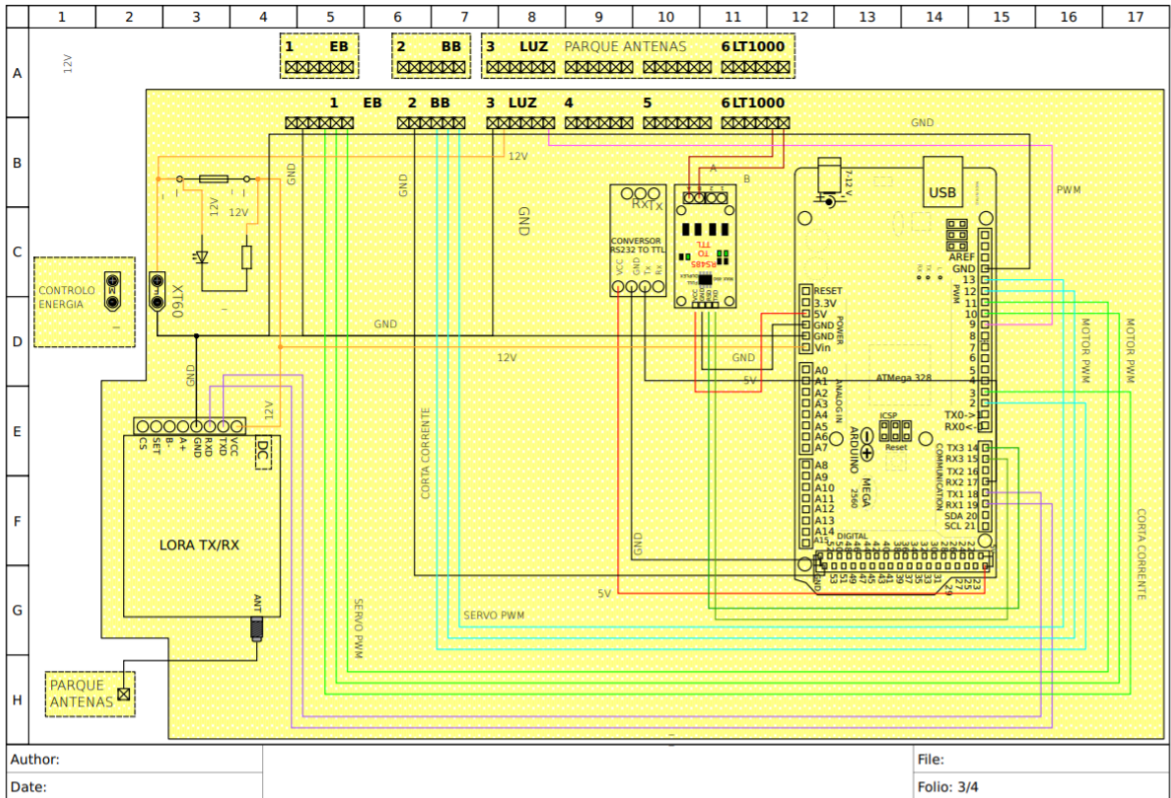


Figura 4-19 - Esquema módulo controlador

Para as ligações de sinais utilizou-se cabo do tipo *ribbon*, soldados num chicote aos pins dos terminais de encaixe, e no outro chicote foram aplicados ponteiros, ponteiros esses que permitem encaixar nos pins da placa controladora, mantendo-se estáveis no local, e, ao mesmo tempo, facilmente removidos e trocados os pins.

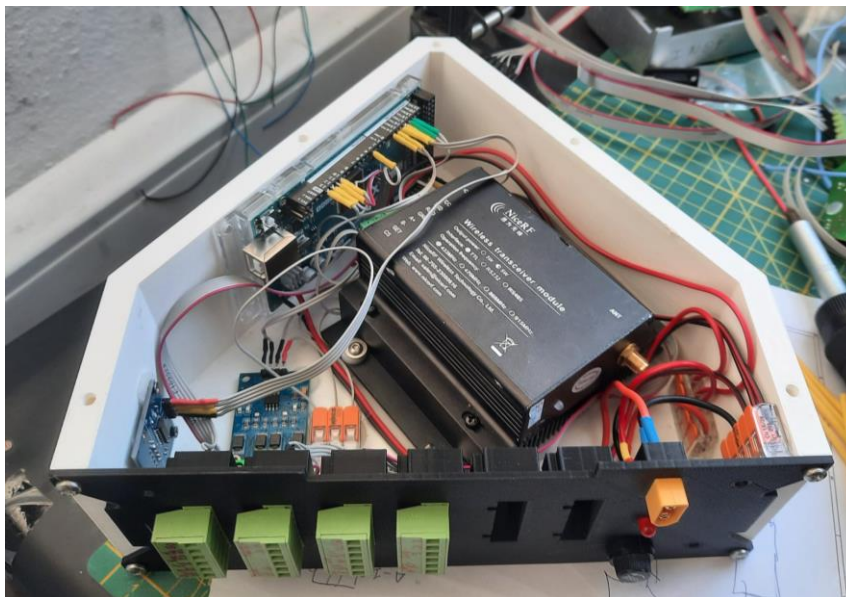


Figura 4-20 - Módulo de controlo finalizado

Foram efetuados testes individuais ao módulo, através do carregamento de código pela porta USB da placa controladora, tendo sido verificado que o conversor RS232/TTL do anemómetro estava a sofrer um aquecimento não regular, tendo sido o mesmo substituído posteriormente. As ligações associadas aos sensores de corrente foram estabelecidas, usando as portas analógicas do *Arduíno*. No entanto, devido aos constrangimentos de tempo não foi possível programar o microcontrolador para efetuar a sua leitura e tratamento. O terminal de encaixe associado aos sensores foi reaproveitado para a alimentação do anemómetro, tendo restado dois encaixes para terminais livres, para utilização futura.

O passo seguinte foi associar os sensores ao módulo, em primeiro lugar a bússola e seguidamente o anemómetro. Através do estudo dos *datasheets* de ambos os equipamentos, foi possível fazer a sua ligação, além da calibração da bússola, que estava a operar em RS422. Foram utilizados apenas os cabos de transmissão número 1 e 2 (castanho e amarelo), além dos de alimentação.

The LT-1000 NRU 8-pin female connector and the multi cable (simple-cut) interconnect details are listed in Table 1 and Figure 1.

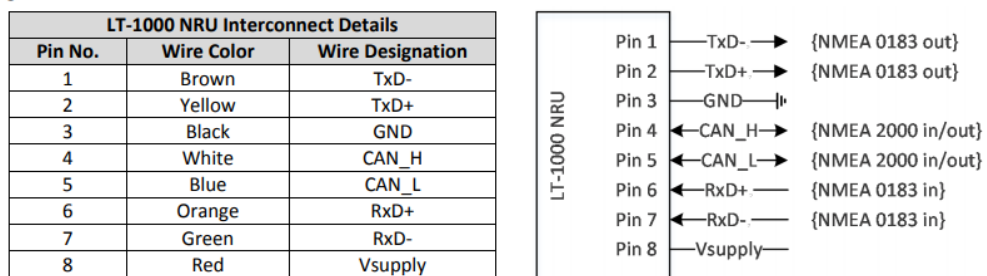


Figura 4-21 - Esquema de ligações da bússola

Para a integração do anemómetro foi necessário, além do estabelecimento das ligações físicas, elaborar um simples algoritmo de leitura dos dados.

4.3 Módulo dos atuadores

O terceiro e último módulo de eletrónica, contendo os controladores dos atuadores, é responsável pela alimentação dos mesmos, assim como do relé dos sinais provenientes do microcontrolador. Foi também desenvolvida uma placa para o controlo em regime de controlo remoto, presente neste módulo, não tendo, no entanto, sido necessária a sua

utilização, servindo apenas como redundância ou meio de controlo degradado caso ocorra uma falha na comunicação com a GCS.

Para este módulo, a tampa necessitava de:

- Três fichas XT60;
- Dois terminais de encaixe;
- Dois orifícios para fusíveis;
- Dois orifícios para LED's.

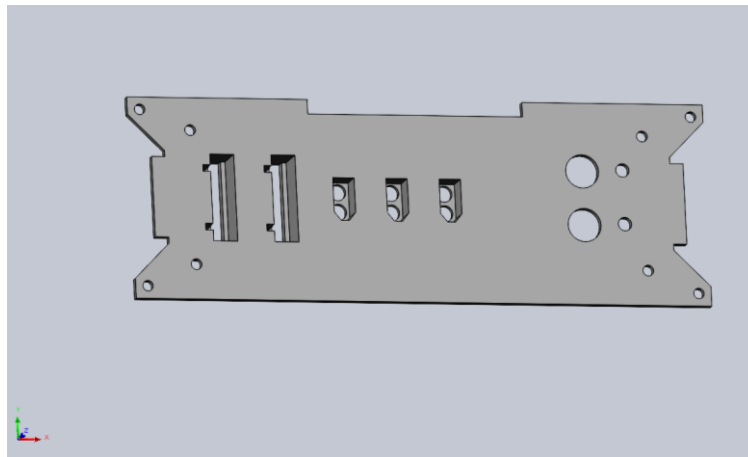


Figura 4-22 - Tampa módulo atuadores

Similarmente ao executado no módulo do microcontrolador, a alimentação de entrada passa pelos fusíveis de 3A em paralelo com dois LED's, sendo esta divisão entre a alimentação do controlador *Veyron* e a do conversor de tensão (*bug converter*), que irá por sua vez alimentar o servo do leme.



Figura 4-23 - Controlador Veyron

Foi assim elaborado o esquema do módulo:

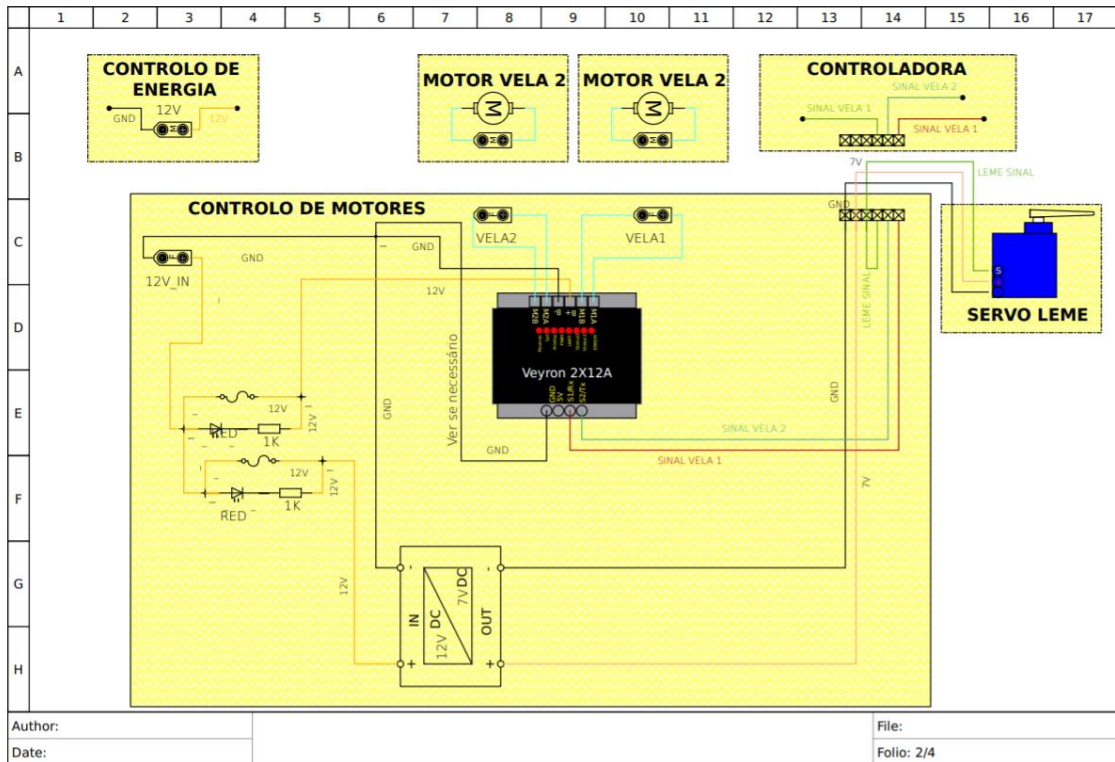


Figura 4-24- Esquema módulo atuadores

O cálculo para as resistências usadas nos LED's foi efetuado com recurso à equação:

$$R = \frac{V}{I}$$

Equação 1 - Cálculo da resistência nos LED's

Com a alimentação de 12V, utilizou-se a aplicação móvel *Eletrdoc*, a partir do qual foi selecionado o LED vermelho *standard*, indicando-nos a corrente de 10mA e a resistência de 1 kΩ, colocada em série com o LED.



Figura 4-25 - Aplicação móvel Eletrodoc. Cálculo da resistência para LED

O servo do leme utilizado é o único componente que foi mantido dos trabalhos anteriores, da marca *Hitec*, possui um binário máximo de 13,2Kg/cm, e alimentado a 7V.



Figura 4-26 - Servo do leme

Instalou-se um conversor de tensão no módulo, que permite regular a tensão de saída, através de semicondutores, tais como díodos e integrados, desde que a tensão de entrada seja superior à de saída (Donald G. Fink & H. Wayne Beatty, 1978).

Possui também um *display*, o que permite a regulação expedita da tensão, sem a necessidade de utilizar um voltímetro. Após a regulação para os 7V que alimentarão o servo do leme, o parafuso foi fixo na posição com recurso a cola quente.



Figura 4-27 - Regulador de tensão

No terminal de encaixe, estão como sinais de entrada os sinais de controlo dos mastros e do leme, sinal esse que volta a sair pelo mesmo terminal, permitindo assim que um cabo triplo leve a alimentação e o sinal ao leme. Os sinais de controlo dos mastros são recebidos pelo *Veyron*

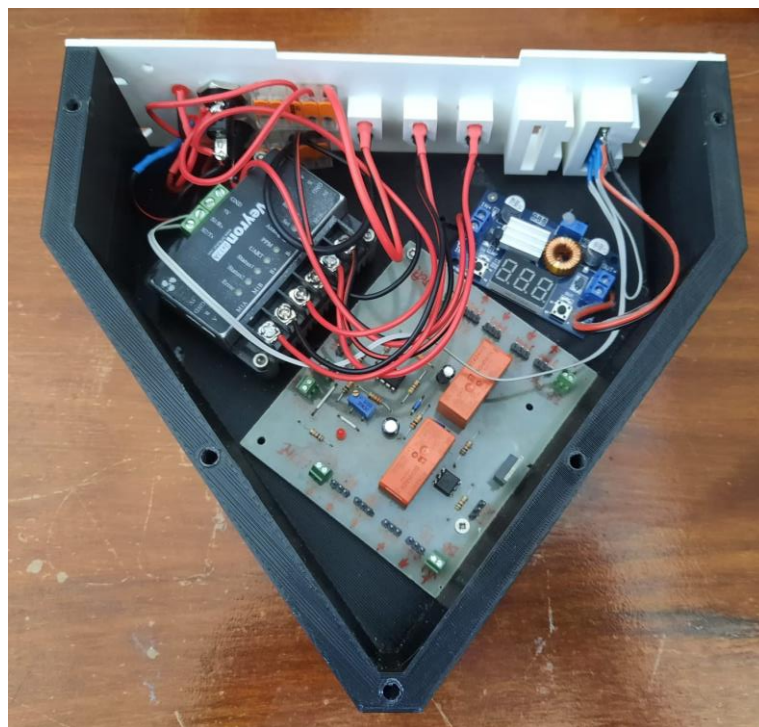


Figura 4-28 – Módulo dos atuadores completo

A placa de controlo remoto de atuadores, faz uso de um kit de rádios TARANIS X9D PLUS Telemetry Radio, que opera na banda dos 2,4GHz. Apresenta módulo recetor portátil, baterias próprias de 2AH e dezasseis canais disponíveis, dos quais quatro foram utilizados, nomeadamente para:

- Leme a Bombordo;

- Leme a Estibordo;
- Motor a Bombordo;
- Motor a Estibordo;



Figura 4-29 - Rádios TARANIS

4.4 Verga de sinais

Na verga de sinais foram estabelecidas ligações locais aos sensores, e além disso, um circuito simples de alimentação de quatro LED's. com o intuito de servirem como iluminação noturna do veleiro foi também preparado. Utilizando LED's de alta intensidade, mas reduzindo a sua taxa de funcionamento, pudemos obter dois pontos de iluminação, sobre cada mastro. O seguinte esquema foi desenvolvido em *QEletroTech*:

5 Testes e Provas de Mar

Este capítulo irá incidir sobre as provas de mar realizadas, que representam o culminar de todo o trabalho realizado, sendo esta a última etapa do ciclo de investigação. Foram efetuadas provas de mar para efeitos de caso de estudo com o Catamarã *X-2601*, desenvolvido pela CEOV para uso do IH.

5.1 Provas de Mar com o Catamarã *X-2601*

No dia 6 de julho de 2021, a CEOV realizou provas de mar com o sistema autónomo *X-2601*, um catamarã desenvolvido com o intuito de ser utilizado pelo IH para realizar trabalhos diversos, com especial destaque para levantamentos hidrográficos. Esta experiência contribuiu com dados valiosos para o estudo e aprontamento do veleiro *Barlavento*.



Figura 5-1 - Catamarã *X-2601*

Similarmente ao veleiro, esta plataforma comunica com a GCS utilizando o protocolo NMEA 0183, e foi utilizado o *software open-source OpenCPN* para a representação visual dos dados recebidos. O módulo de comunicações utilizado é um módulo LoRa igual ao do veleiro, e calibrado com os mesmos parâmetros. Assim, permite-nos obter e visualizar a posição GPS, rumo, velocidade, direção e força do vento aparente, ângulo de leme, *pitch* e *roll*. O programa está capacitado para apresentar dados de mais sensores que sejam integrados na plataforma de estudo, tais como barómetros, sondas, odómetros, radares, termómetros e AIS.

A GCS utilizada é o padrão dos sistemas não tripulados produzidos pela CEOV, que são baseados na GCS original do *Barlavento* (Fernandes, 2016). Apresenta as vantagens de ser propriedade da Marinha, além de oferecer compatibilidade e interoperabilidade entre os sistemas desenvolvidos.

Durante o aprontamento do catamarã, o signatário acompanhou todos os procedimentos executados, desde o estabelecimento dos sistemas elétricos à ligação direta com a GCS, carregamento do algoritmo de controlo e testes rápidos à resposta da plataforma no modo de RC, inserindo ângulos de leme e regimes de motor. De seguida foram estabelecidos oito *waypoints*, a partir de inserção manual das coordenadas (retiradas do *OpenCPN*) no algoritmo de controlo, o que representa uma limitação do sistema, pois não é possível alterar o percurso após o carregamento direto do código no microcontrolador.

A partir da GCS, podemos efetuar o controlo da plataforma utilizando dois métodos, método de RC nos atuadores, regulando diretamente o ângulo de leme e a velocidade. Ao ser colocado na água, este foi o modo adotado, tendo a plataforma navegado 34 minutos, a uma velocidade média de 1,1 nó, e percorrido 0,63 NM.

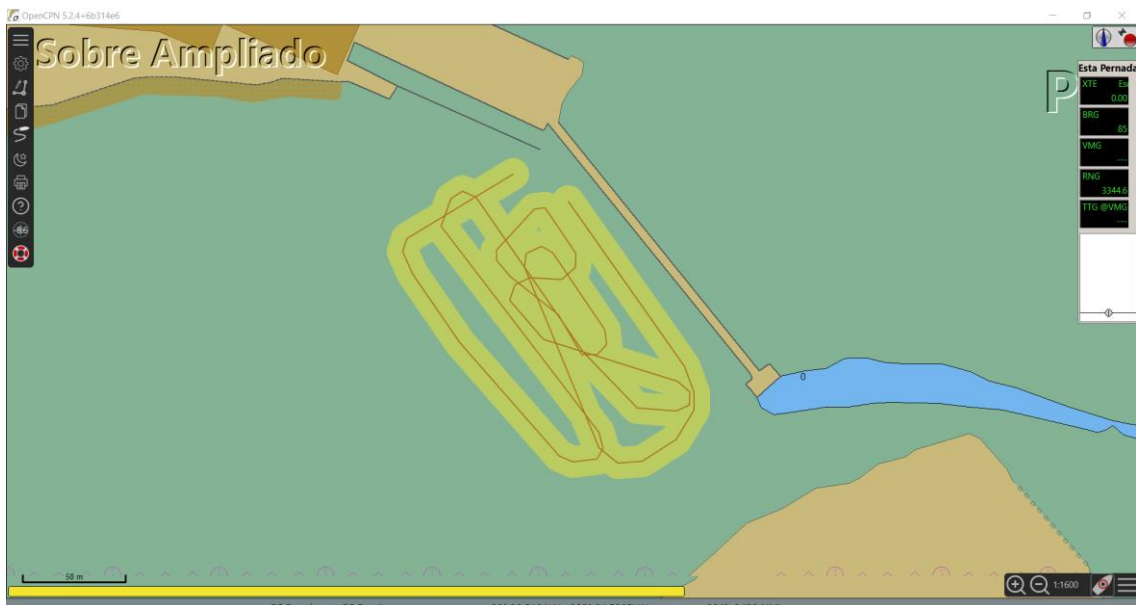


Figura 5-2 - Rasto em modo RC (1ª pernada)

No decorrer da fase de RC, o signatário tomou controlo da plataforma, por forma a ganhar familiaridade com o sistema e sensibilidade para o tempo de resposta associado a cada manobra e alteração de regime.



Figura 5-3 - GCS em modo RC

Na etapa seguinte, o catamarã navegou em modo autónomo, sendo notável a diferença na precisão do seguimento do planeamento. Navegou-se assim durante 43 minutos, a uma velocidade média de 2,2 nós, percorrendo 1,61NM.



Figura 5-4 - Rasto em modo autónomo

Concluíram-se as provas de mar com nova etapa de modo semiautónomo.



Figura 5-5 - Rasto em modo RC (2ª perna)

5.2 Testes ao sistema completo em terra

No decorrer da semana de 22 de agosto até à prova de mar a 26 de agosto, realizaram-se vários testes ao sistema completo, com o veleiro ainda em terra. Estes testes foram efetuados visando confirmar e assegurar diversos sistemas e condições, listados abaixo:

- Alimentação de todo o sistema autonomamente;
- Fiabilidade e estima da autonomia do veleiro;
- Testar comunicações com a GCS;
- Testar receção de dados fiel dos sensores de bordo;
- Testar a resposta dos atuadores do veleiro;
- Avaliar funcionamento dos sistemas de proteção dos módulos (fusíveis e LED's).

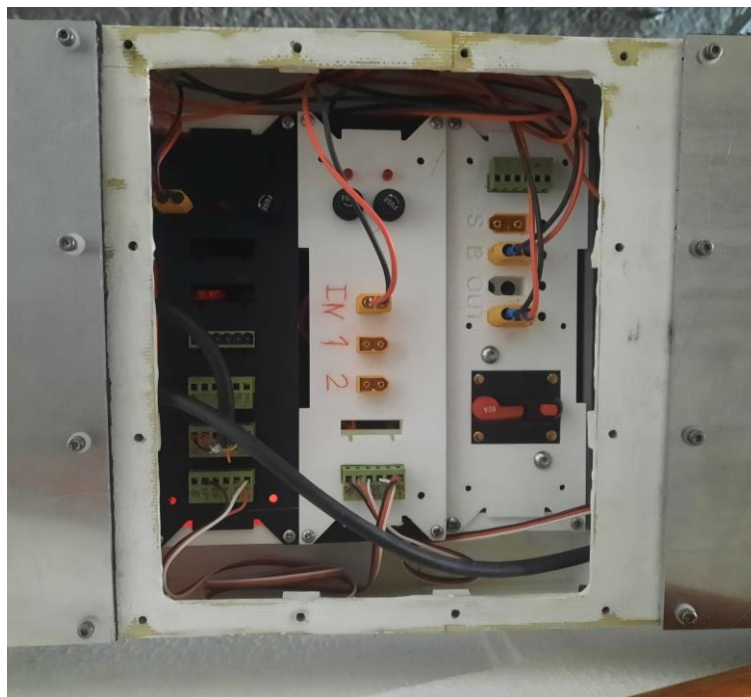


Figura 5-6 - Módulos completos integrados no veleiro

Começou-se por acionar o interruptor, e confirmou-se a alimentação total do veleiro pela observação dos LED's que se acenderam. O módulo LoRa iniciou a transmissão de sinais, e na GCS a informação foi corretamente recebida. Simultaneamente, o servo do leme trancou a sua posição. Através do manuseio livre da bússola, não foi possível observar qualquer ajuste por parte do leme. Após leitura dos dados enviados pela bússola, confirmou-se a falta de posição GPS. Concluiu-se que a posição GPS não era recebida dentro do edifício da CEOV, além de demorar alguns minutos a fixar quando no exterior.

Assim que foi o GPS começou a enviar dados corretos e completos, observaram-se ajustes no leme, no entanto, sempre no sentido inverso ao suposto. O código de controlo foi revisto e corrigido. Também a posição de leme a meio foi ajustada, pois tinha-se verificado um desvio de 5 graus para bombordo. Aplicadas as alterações, o leme passou a reagir adequadamente à alteração da proa do veleiro.

A compatibilidade eletromagnética do sistema deve ser estudada por forma a serem detetadas incompatibilidades nos componentes utilizados. O âmbito desta dissertação, nunca foi o estudo aprofundado da radiação emitida e da imunidade à radiação do sistema, contudo, durante os testes de bancada dos módulos de eletrónica, foi detetada uma falha intermitente do microcontrolador. Após *debugging*, detetou-se que o sistema falhava quando era iniciada uma transmissão pelo módulo de comunicações. Foram adicionados condensadores de desacoplamento nas alimentações, tendo-se verificado que a taxa de

ocorrência da falha do microcontrolador reduziu, verificando-se assim que existia *surge current* que causava a falha do sistema. Assumiu-se que as falhas que existiam após este melhoramento seriam causadas por ondas eletromagnéticas que excitariam uma parte do sistema do microcontrolador, provocando o seu *reset*. Para solucionar esta falha, afastou-se o transmissor do microcontrolador, não se tendo verificado mais nenhuma falha no sistema eletrónico.

5.3 Testes na Bacia de Manobra da Base Naval de Lisboa

No dia 26 de agosto realizou-se a navegação experimental do veleiro autónomo *Barlavento*, após aprontamento do veleiro, e utilizando como meio de apoio uma semirrigida do CNOCA.

O objetivo desta primeira navegação foi a de verificar as condições de estabilidade, após uma grande alteração às obras mortas do veleiro, assim como a adição da verga de sinais, condições de estanqueidade, operacionalidade dos sistemas e eficiência do mecanismo dos mastros.

Foi estabelecida a GCS e inseridos três *waypoints* no código de controlo do veleiro.

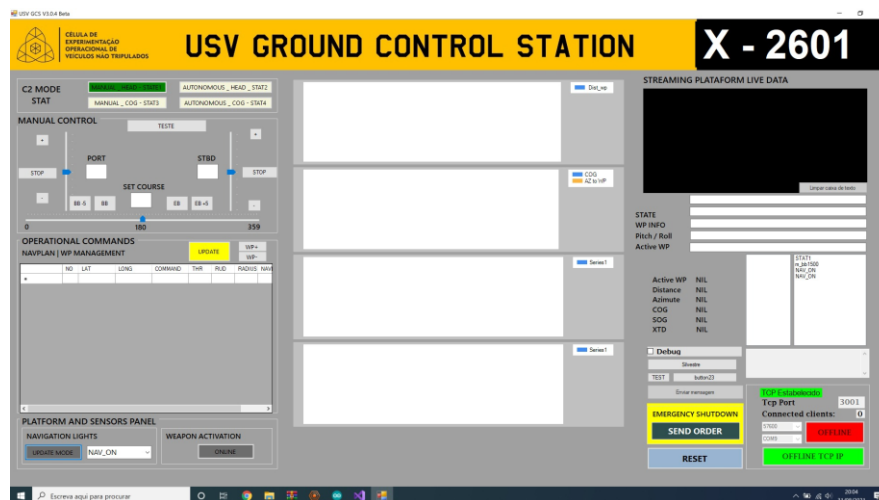


Figura 5-7 - Display da GCS

O veleiro foi colocado na água na rampa do CNOCA, com o auxílio da cegonha aí presente. Ao ser colocado na água pôde rapidamente observar-se que a estabilidade do veleiro tinha sido comprometida com a adição da verga de sinais. O veleiro adornava aproximadamente 70 graus e revelou-se incapaz de recuperar a posição inicial.



Figura 5-8 - Prova de mar do Barlavento



Figura 5-9 – Barlavento

Todos os sensores e sistemas mantiveram-se operacionais e o veleiro ganhou seguimento. Nesse dia as previsões de vento indicaram velocidades de 9 a 11 nós, com rajadas até aos 13 nós, valores confirmados pelos dados do anemómetro.

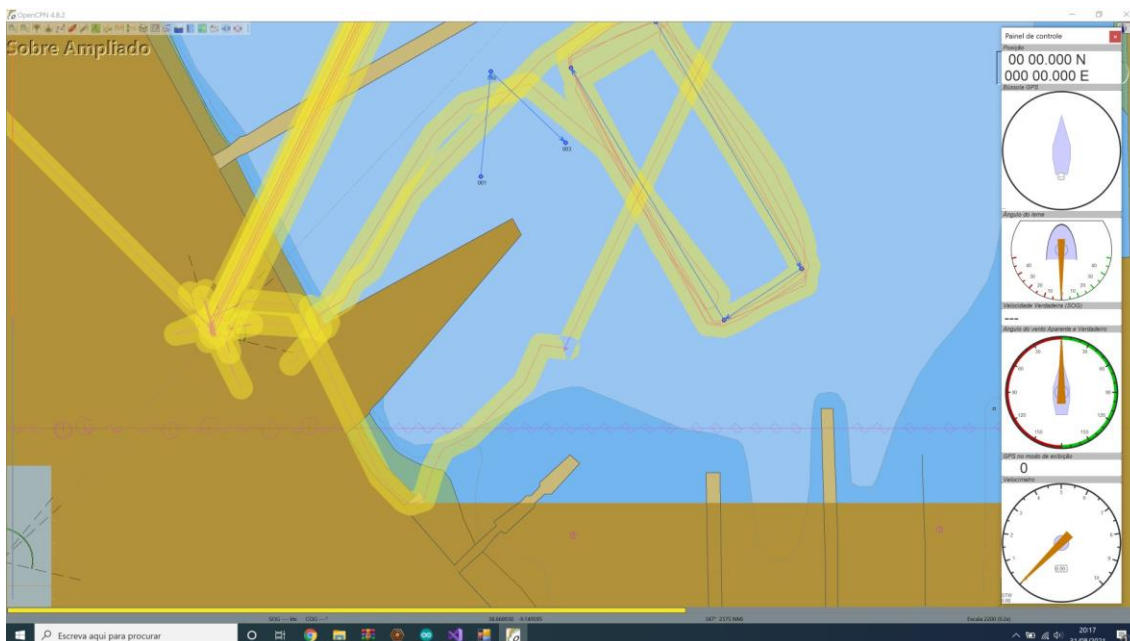


Figura 5-10 - Display no OpenCPN

Durante a realização da prova o veleiro manteve contacto com a GCS constantemente, apresentando no *display* a posição GPS, o ângulo de leme introduzido, a direção do vento e sua velocidade, e a velocidade do veleiro. Lamentavelmente, durante o reboque do veleiro para a sua posição de partida do planeamento, um erro de manuseamento levou à quebra de um dos suportes da verga de sinais, causando a queda à água da verga e dos sensores. Para salvaguarda da LT1000, a prova foi interrompida antes de tempo e o veleiro rebocado de volta a terra.

5.4 Exercício REP(MUS) 2021

Durante o período de seis a catorze de setembro, o signatário participou no exercício REP(MUS) 2021, realizado no Ponto de Apoio Naval de Troia, exercício de experimentação a grande escala, realizado anualmente, em que comunidades operacionais trabalham em conjunto com instituições académicas e empresas para desenvolver e testar conceitos operacionais e avanços tecnológicos nas operações marítimas. O exercício conta com a participação de Marinhas de dezassete países da NATO, empregando meios autónomos aéreos, de superfície e subsuperfície para realização de missões de guerra de minas, vigilância ou proteção de áreas, análise de amostras de água e interoperabilidade dos vários sistemas em operação simultaneamente.



Figura 5-11 - Plataforma da Royal Navy MADFOX

Acompanhando a equipa técnica da CEOV, foi desenvolvido trabalho no aprontamento das plataformas autónomas *Bujiador*, embarcação semirrígida a jato de água e reconvertida para operar autonomamente e o Catamarã *X-2601*, munido de um sonar lateral para deteção de objetos.

A plataforma *X-2601* será utilizada pela primeira vez em operações de deteção de objetos no fundo marinho em conjunto com outras embarcações autónomas, sendo a interoperabilidade do controlo do sistema com os restantes meios envolvidos o principal foco do seu empenhamento, testando simultaneamente novas hélices e novo sensor AIS.

A semirrígida *Bujiador*, meio orgânico dos navios da classe Tejo da Marinha Portuguesa, plataforma escalável e versátil, aprimorada para operar nas seguintes vertentes:

- Replicar ataques de ameaça assimétrica de embarcações pequenas;
- Aferição de eficácia, precisão e prontidão operacional do sistema de armas dos navios;
- Treino de fogo real de superfície;
- Treino de proteção e combate contra ameaças assimétricas;
- Utilização do veículo de superfície como alvo direto ou rebocador de alvos (alvo rebocado insuflável);
- Operação em condições de estado do mar 4 (Escala Douglas) permitindo que o treino ocorra em condições mais adversas;
- Interoperabilidade e intermutabilidade entre navios.

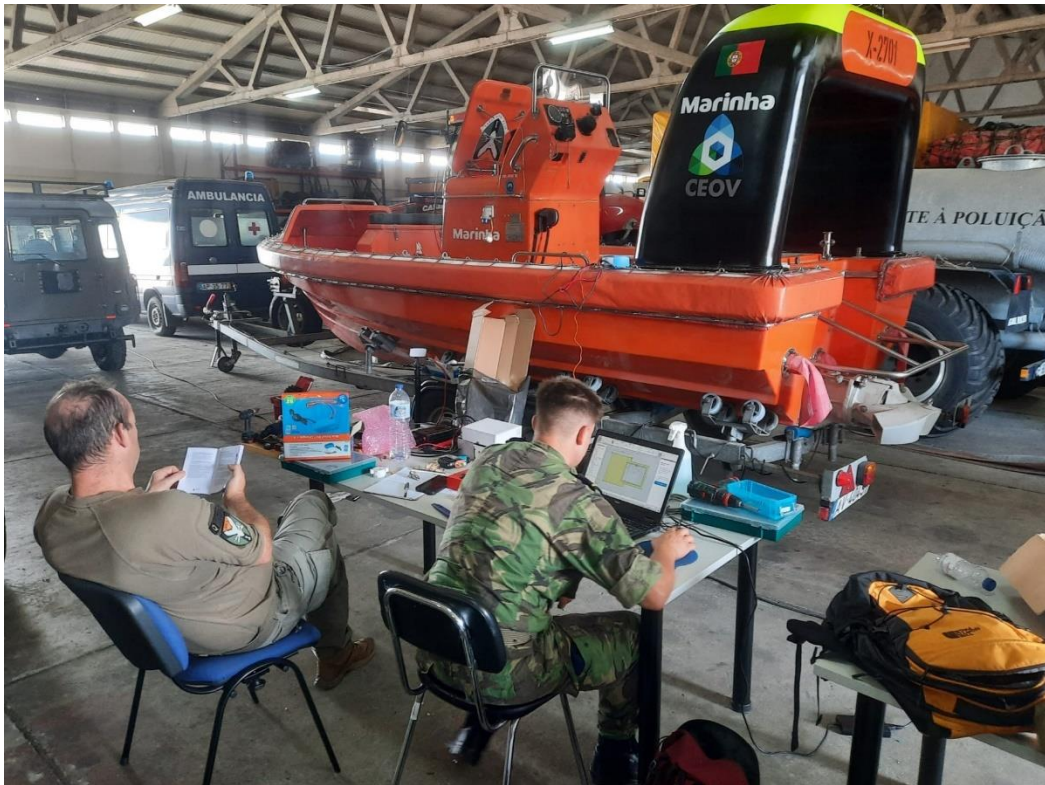


Figura 5-12 – Bujiador

Realça-se que todo o sistema de comando e controlo de ambas as plataformas teve origem no projeto do veleiro *Barlavento*, após atualização do algoritmo no decorrer do projeto. A GCS utilizada e os módulos de comunicação são adaptados do veleiro, garantindo a interoperabilidade dos sistemas. Os módulos de eletrónica foram montados utilizando desenhos das caixas dos módulos e esquemas da eletrónica desenvolvidos no projeto do veleiro.

6 Conclusões

Este trabalho iniciou-se com o estudo dos trabalhos realizados no âmbito do projeto eVentos e no veleiro *Barlavento*, assim como uma revisão de conceitos e matérias associadas aos sistemas autónomos, a sua história, conceção e valências. Criadas as fundações para o tema, procedeu-se à identificação ativa de objetivos específicos para o trabalho, tendo por base as observações realizadas nos trabalhos precedentes, e nos meios disponíveis. Tendo sido os objetivos delineados, deu-se início à aquisição dos materiais e recursos necessários ao cumprimento dos mesmos.

Foi utilizado o programa SolidWorks Premium para fazer o desenho da elevação do casco, todo o material foi impresso com recurso a impressoras 3D, disponibilizadas pela CEOV. A componente estrutural do trabalho envolveu a montagem do casco impresso no casco original, a sua colagem, fibragem, pintura e sucessivas passagens com lixa entre cada camada. Foram desenvolvidas e aplicadas portas de visita novas, enquadrado um novo mecanismo de controlo dos mastros e foi construída uma verga de sinais para sensores.

Na componente de eletrónica, foram criados módulos removíveis para os sistemas eletrónicos, desenhadas portas para os mesmos, estabelecidas seguranças, criaram-se condições para o estabelecimento não só dos componentes integrados, mas também de *payloads* e mecanismos a ser introduzidos no futuro. Foram adquiridos, testados e implementados painéis solares e uma placa de controlo remoto. O algoritmo de comando e controlo foi revisto e atualizado.

A plataforma resultante deste trabalho revelou bons resultados de estanqueidade, na solidez e robustez da alteração do casco, e realçam-se os desenvolvimentos feitos na componente eletrónica, tendo os objetivos para a mesma sido completados na totalidade.

A manufatura das portas de visitas foi atrasada, devido à incompatibilidade do desenho inicial com o material impresso, o que causou a manufatura de duas portas adicionais após correção do desenho. Sugere-se considerar a dilatação térmica das peças 3D no desenho das mesmas.

O peso do veleiro atual é de 26Kg, 5Kg a mais que o veleiro inicial (Fernandes, 2016), resultado do peso conjunto da elevação do convés, módulos de eletrónica e verga de sinais. Tendo em conta o peso adicionado, acima do centro de gravidade, a altura metacêntrica irá diminuir. Deve considerar-se alterar o bolbo para 5Kg, representando um incremento de 2Kg, que, de forma empírica aconselhada pela equipa técnica da CEOV iria manter a estabilidade do veleiro, passando o deslocamento a ser de 28Kg, valor abaixo do deslocamento máximo calculado para o casco (Fernandes, 2016).

Devido a constrangimentos de tempo, não foi possível desenvolver um mecanismo de controlo de mastros que foi inicialmente pensado, sugerindo-se para trabalho futuro.

Com o desenvolvimento do projeto o signatário adquiriu as seguintes competências:

- Utilização de programas de desenho em CAD;
- Utilização, manuseamento e manutenção de impressoras 3D;
- Escolha de sistemas eletrónicos e dimensionamento de consumos;
- Desenvolver placas eletrónicas, organização de componentes no navio e a fazer *interfaces*;
- Trabalhos com fibra e resolução de problemas mecânicos.

Bibliografia

- Aartrijk, M. Van, Tagliola, C., & Adriaans, P. (2002). AI on the Ocean: the RoboSail Project. ECAI, 653–657. Obtido de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.8172&rep=rep1&type=pdf> <http://frontiersinai.com/ecai/ecai2002/pdf/p0653.pdf> (1838). *Arquivo Popular*.
- Alves, J., & Cruz, N. (setembro de 2008). Fast - an autonomous sailing platform for. *OCEANS 2008*.
- Arquivo Popular*. (1838).
- Coimbra. (1555). *Arte da Guerra do Mar*.
- Donald G. Fink, & H. Wayne Beatty. (1978). *Standard Handbook for Electrical Engineers Eleventh Edition*. pp. 7-30.
- Elkaim, G. H., & Boyce, L. (2007). Experimental aerodynamic performance of a self-trimming wing-sail for autonomous surface vehicles. *IFAC Proc. Volumes 40*, pp. 271–276.
- Elkaim, G., & Kelbley, R. (2006). Station keeping and segmented trajectory control of a wind-propelled autonomous catamaran. *Proc. 45th IEEE Conf. Decis. Control*, pp. 2424-2429.
- Fernandes, P. (2016). *Projeto e Construção de um Veleiro Autónomo, utilizando materiais compósitos, Impressão a 3D e Aprendizagem Máquina*.
- Gentemann, C., Scott, J., Mazzini, P., Pianca, C., Akella, S., & Minnet, P. (2020). Saildrone: adaptively sampling the marine environment. *Bull. Am. Meteorol. Soc.* 101, pp. 744-762.
- Gonçalves, M. (2009). *Sagres - Construindo a Lenda*.
- Mahan. (1890). *The Influence of Sea Power Upon History: 1660–1783*.
- Marques, P. (2015). *Monitorização e gestão de energia no veleiro autónomo FASt*.

- Mayerfeld, P. (18-21 de setembro de 2017). Fluorometers: integration experiences with unmanned vehicles. *Proceedings of the OCEANS 2017 - Anchorage*, pp. 1-5.
- Oberg, E. (1920). *Spiral and Worm Gearing*.
- Oliveira, P. F. (1555).
- Santos, J. B. (2019). eVento - Desenvolvimentos no veleiro autónomo Barlavento. pp. 42-44.
- Stelzer, R., & Jafarmadar, K. (2011). History and Recent Developments in Robotic Sailing. p. 4.
- Torrieri, D. (2018). *Principles of Spread-Spectrum Communication Systems*, 4th ed.
- Villiers, A. (1950). *A Campanha do Argos*.
- Voosen, P. (2018). Saildrone fleet could help replace aging buoys. *Science* 359, pp. 1082–1083.
- Yang, S., Liu, C., Liu, Y., An, J., & Xiang, X. (11 de março de 2021). Generic and Flexible Unmanned Sailboat for Innovative Education and World Robotic Sailing Championship. *Front. Robot. AI*.

Apêndices

Apêndice 1 – Bill of Materials dos Painéis Solares

		PROJECT:	Destino Final		
		TITLE:	BILL OF MATERIALS (BOM) (per unit manufactured)		
DESCRIPTION	MANUFACTURER	PRODUCT REFERENCE	COST PER UNIT	QUANTITY	TOTAL COST
Solar pannel	XINPUGUANG	https://pt.aliexpress.com/item/32785137	15,35 €	5	76,75 €



Specification

Maximum Power (Pmax) : 10W
 Open Circuit Voltage (Voc) : 14.4V
 Short Circuit Current (Isc) : 0.91A
 Maximum Power Voltage (Vmp) : 12V
 Maximum Power Current (Imp) : 0.83A
 Maximum System Voltage (Imp) : 1000DC
 Size : 440*190*3mm
 STC:Irradiance1000w/m2, Temperature 25 C,
 AM=1.5

Product Structre



- 1 PET
- 2 EVA
- 3 Monocrystalline
- 4 EVA
- 5 Junction box

Figura 0-1 - Bill Of Materials - Painéis Solares

Apêndice 2 – Rasto em modo RC 1 - Catamarã X-2601

Propriedades do rasto

Base: Avançado

Nome: X-2601_JH_Manual_1

De: [] Para: []

Parâmetros de apresentação

Mostrar na carta Cor: Vermelho escuro Estilo: Por defeito Largura: Por defeito

Estadísticas

Distancia Total: 0.63 NMi Velocidade média: 1.10 Duração: 00:34

Pontos gravados

Perna	Distância	Azimute	Latitude	Longitude	Marca de tempo	Velocid...
---	3419.93 N...	085 °T	38° 38.548...	009° 04.77...	07/06/2021 14:23:34	--
1	0.05 NMi	145 °T	38° 38.505...	009° 04.74...	07/06/2021 14:25:21	1.78
2	0.01 NMi	160 °T	38° 38.498...	009° 04.73...	07/06/2021 14:25:35	1.87
3	0.00 NMi	179 °T	38° 38.493...	009° 04.73...	07/06/2021 14:25:45	1.62
4	0.01 NMi	212 °T	38° 38.486...	009° 04.74...	07/06/2021 14:26:01	2.02
5	0.01 NMi	237 °T	38° 38.480...	009° 04.75...	07/06/2021 14:26:21	1.90
6	0.01 NMi	266 °T	38° 38.480...	009° 04.76...	07/06/2021 14:26:35	1.63
7	0.01 NMi	309 °T	38° 38.483...	009° 04.76...	07/06/2021 14:26:47	1.90

Imprimir Dividir Estender Rasto Para rota Exportar

Cancelar OK

Figura 0-2 - Rasto em Modo RC 1- Catamarã

Apêndice 3 – Rasto em modo autónomo 1 - Catamarã X-2601

Propriedades do rasto

Base: Avançado

Nome: X-2601_JH_Manual_2

De: [] Para: []

Parâmetros de apresentação

Mostrar na carta Cor: Preto Estilo: Por defeito Largura: Por defeito

Estadísticas

Distancia Total: 0.80 NMi Velocidade média: 1.17 Duração: 00:41

Pontos gravados

Perna	Distância	Azimute	Latitude	Longitude	Marca de tempo	Velocid...
---	3419.90 N...	085 °T	38° 38.537...	009° 04.82...	07/06/2021 15:41:58	--
1	0.01 NMi	138 °T	38° 38.532...	009° 04.81...	07/06/2021 15:42:10	1.88
2	0.01 NMi	107 °T	38° 38.530...	009° 04.80...	07/06/2021 15:42:24	2.19
3	0.01 NMi	134 °T	38° 38.523...	009° 04.79...	07/06/2021 15:42:40	2.35
4	0.02 NMi	146 °T	38° 38.503...	009° 04.78...	07/06/2021 15:43:16	2.30
5	0.01 NMi	124 °T	38° 38.498...	009° 04.77...	07/06/2021 15:43:30	2.28
6	0.01 NMi	103 °T	38° 38.497...	009° 04.76...	07/06/2021 15:43:44	2.08
7	0.01 NMi	081 °T	38° 38.498...	009° 04.74...	07/06/2021 15:44:00	2.18

Imprimir Dividir Estender Rasto Para rota Exportar

Cancelar OK

Figura 0-3 - Rasto em Modo autónomo 1- Catamarã

Apêndice 4 – Rasto em modo RC 2 - Catamarã X-2601

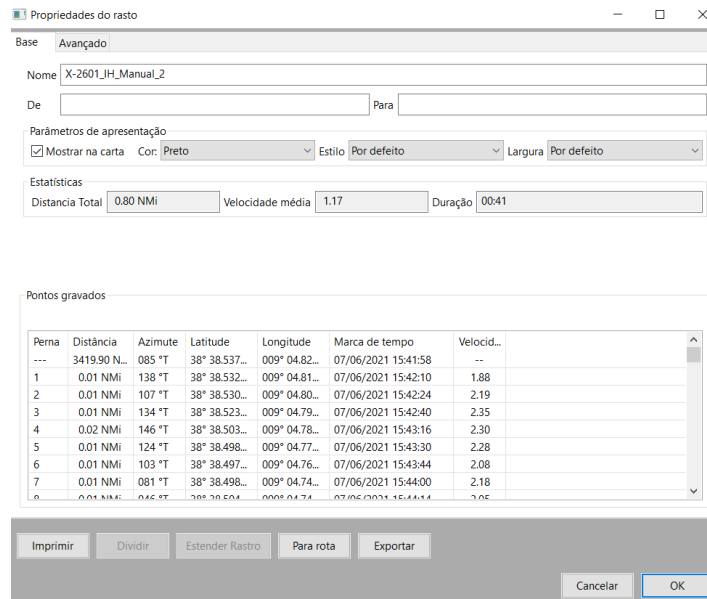


Figura 0-4 - Rasto em modo RC 2

Apêndice 5 – Código C de Comando – do Veleiro

```
#include <math.h>
```

```
#include <Servo.h>
```

```
#include <Wire.h>
```

```
#include <SPI.h>
```

```
#include <SD.h>
```

```
#include <Servo.h>
```

```
#include <PString.h>
```

```
#include "Main_GPS.cpp"
```

```
GPS gps;
```

```
#include "WayPoints.cpp"
```

```
WP wp;
```

```
#include "Rudder.cpp"

Rudder rudder;

#include "Decode_NMEA.cpp"

DECODE_NMEA Decode_NMEA;

#include "Control.cpp"

Control control;

#include "wind.cpp"

WIND Wind;

#define RUDDER_PIN1 11 // leme EB

#define RUDDER_1_TRIM 90 //VALORES PARA CENTRAR OS LEMES (TRIM)

#define MOTOR1_PIN 10 // motor EB esta

#define MOTOR2_PIN 12 // motor BB

#define MOTOR_PARADO 1500

#define NAV_LIGHTS_PIN 9 // PIN9 - NAVIGATION LIGHTS

//
```

```
// PINS 2/3 - RELES POTENCIA FLUTUADORS

//

//*****

float DistCalc = 1000; // float dist2wp = 0; // dist2wp is the distance (in meters) to the
next Waypoint2

float AziCalc = 270;

float Manual_head = 320;

float XTD = 0;

float head = 0;

float roll = 0;

float pitch = 0;

bool autonomous = true;

bool sail_stop = false;

bool sail_auto = true;

bool sail_upwind = false;

bool State1 = true;

bool State2 = false;

bool State3 = false;

bool State4 = false;
```

```
bool State5 = false;
```

```
long double millisBS = 0;
```

```
long double millis_sail = 0;
```

```
String Comand;
```

```
double timer = 0; //time it takes for the void loop run
```

```
float frequency = 0; //frequency of the code
```

```
File myFile;
```

```
Servo servo_rudder1;
```

```
Servo motor1;
```

```
Servo motor2;
```

```
void setup() {
```

```
    Serial.begin(57600); // connect serial
```

```
    Serial.setTimeout(50);
```

```
    Serial1.begin(57600); // connect Communications
```

```
Serial1.setTimeout(50);

Serial2.begin(4800); // connect anemometer

Serial2.setTimeout(5);

Serial3.begin(4800); // LT1000

Serial3.setTimeout(5);

Serial.println("Sistema a iniciar, seja bem-vindo");

servo_rudder1.attach(RUDDER_PIN1); // the rudder is connected to pin 12

servo_rudder1.write(RUDDER_1_TRIM);

Serial.print("Leme 1 na posição: "); Serial.println(RUDDER_1_TRIM);

Serial.print("Leme 2 na posição: "); Serial.println(RUDDER_2_TRIM);

motor1.attach(MOTOR1_PIN);

motor1.write(MOTOR_PARADO);

motor2.attach(MOTOR2_PIN);

motor2.write(MOTOR_PARADO);

pinMode(NAV_LIGHTS_PIN, OUTPUT);
```

```

digitalWrite(NAV_LIGHTS_PIN, HIGH);

Serial.print("Motores Parados: "); Serial.println(MOTOR_PARADO);

wp.Update();

Serial.print("Lista de wp's: ");

wp.wp_list();

wp.last_wp_number = wp.total_wp_number;

Serial.println("Tudo ok, vamos iniciar manobra.");

motor1.write(control.motor_eb_speed);

motor2.write(control.motor_bb_speed);

}

void loop() {

  Decode_NMEA.Update();

  Wind.Update();

  ReadComand();

  DistCalc = gps.CalcDist(Decode_NMEA.lat, Decode_NMEA.lon,
(wp.wplat[wp.number]), (wp.wplon[wp.number])); // Calcula a distancia entre os Wp's e
a posição atual

```

```
    AziCalc      =      gps.CalcAzi(Decode_NMEA.lat,      Decode_NMEA.lon,
(wp.wplat[wp.number]), (wp.wplon[wp.number])); // Calcula o Azimute entre os Wp's e
a posição atual
```

```
    XTD          =      gps.CalcCrossTrackError((wp.wplat[wp.last_wp_number]),
(wp.wplon[wp.last_wp_number]),      Decode_NMEA.lat,      Decode_NMEA.lon,
(wp.wplat[wp.number]), (wp.wplon[wp.number]) );
```

```
Update_GCS(1000); //250
```

```
if (State1) { // Semi_Autonomous - head
```

```
    servo_rudder1.write(rudder.UpdateRudder1(Decode_NMEA.head, Manual_head,
RUDDER_1_TRIM));
```

```
}
```

```
if (State2) { // Autonomous - head
```

```
    servo_rudder1.write(rudder.UpdateRudder1(Decode_NMEA.head,      AziCalc,
RUDDER_1_TRIM));
```

```
}
```

```
if (State3) { // Semi_Autonomous - COG
```

```
    servo_rudder1.write(rudder.UpdateRudder1(Decode_NMEA.COG, Manual_head,
RUDDER_1_TRIM));
```

```
}
```

```
if (State4) { // Autonomous - COG
```

```
    servo_rudder1.write(rudder.UpdateRudder1(Decode_NMEA.COG,      AziCalc,
RUDDER_1_TRIM));
```

```
}
```

```

wp.test(DistCalc);

}

void ReadComand() {

String validComand = "Invalid";

if (Serial1.available() > 0)

    Comand = Serial1.readString();

if (Serial.available() > 0)

    Comand = Serial.readString();

if (Comand == "") {

}

else {

    Comand = Comand.substring(0, Comand.length() - 1);

    Serial.print("Comando: "); Serial.println(Comand);

    Serial1.print("Comando: "); Serial1.println(Comand);

if (Comand == "STAT1") { // Semi_Autonomous - head

    State1 = true;

    State2 = false;

    State3 = false;

    State4 = false;

```

```

State5 = false;

validComand = "Valid";

autonomous = false;

}

if (Comand == "STAT2") { // Autonomous - head

State1 = false;

State2 = true;

State3 = false;

State4 = false;

State5 = false;

autonomous = true;

validComand = "Valid";

}

if (Comand == "STAT3") { // Semi_Autonomous - COG

State1 = false;

State2 = false;

State3 = true;

State4 = false;

State5 = false;

autonomous = false;

validComand = "Valid";

}

if (Comand == "STAT4") { // Autonomous - COG

State1 = false;

```

```

State2 = false;

State3 = false;

State4 = true;

State5 = false;

autonomous = true;

validComand = "Valid";

}

if (Comand == "STAT5") { // Autonomous - COG

State1 = false;

State2 = false;

State3 = false;

State4 = false;

State5 = true;

autonomous = true;

validComand = "Valid";

}

if (Comand == "reset") {

control.software_Reset();

}

if (Comand == "w+") {

wp.last_wp_number = wp.number;

wp.number++;

```

```

if (wp.wplat[wp.number] == 0) {
    wp.number = 1;
}

Serial.println("wp mais");
Serial1.println("wp mais");

validComand = "Valid";
}

if (Comand == "w-") {
    wp.last_wp_number = wp.number;

    wp.number--;

    if (wp.number < 1) {
        wp.number = wp.total_wp_number;
    }

    Serial.println("wp menos");
    Serial1.println("wp menos");

    validComand = "Valid";
}

if (Comand == "TestGPS") {
    Serial.println("Testing GPS: Vamos parar motores e colocar leme a meio");

    validComand = "Valid";

    servo_rudder1.write(RUDDER_1_TRIM);
}

```

```

motor1.write(1500);

motor2.write(1500);

gps.Test();
}

if (Comand.substring(0, 1) == "h" && Comand.substring(Comand.length() - 1,
Comand.length()) == "*")
{
if (Comand.substring(1, 4).toInt() >= 0 && Comand.substring(1, 4).toInt() < 360)
{
Manual_head = Comand.substring(1, 4).toInt();

Serial.print("nova proa: "); Serial.println(Comand.substring(1, 4).toInt());

Serial1.print("nova proa: "); Serial1.println(Comand.substring(1, 4).toInt());

validComand = "Valid";
}
else
{
Serial.println("Proa nao valida camarada!");

Serial1.println("Proa nao valida camarada!");
}
}

if (Comand.substring(0, 4) == "m_bb")
{

```

```

    if (Comand.substring(4, 8).toInt() > 999 && Comand.substring(4, 8).toInt() <
2001)

        control.motor_bb_speed = Comand.substring(4, 8).toInt();

        motor2.write(control.motor_bb_speed);

        Serial.print("motor de bb a: "); Serial.println(Comand.substring(4, 8).toInt());

        Serial1.print("motor de bb a: "); Serial1.println(Comand.substring(4, 8).toInt());

        validComand = "Valid";

    }

    if (Comand.substring(0, 4) == "m_eb")

    {

        if (Comand.substring(4, 8).toInt() > 999 && Comand.substring(4, 8).toInt() <
2001)

            control.motor_eb_speed = Comand.substring(4, 8).toInt();

            motor1.write(control.motor_eb_speed);

            Serial.print("motor de eb a: "); Serial.println(Comand.substring(4, 8).toInt());

            Serial1.print("motor de eb a: "); Serial1.println(Comand.substring(4, 8).toInt());

            validComand = "Valid";

        }

    if (Comand == "wp_list") {

        validComand = "Valid";

        wp.wp_list();

    }

    if (Comand == "NAV_ON") {

        validComand = "Valid";

```

```

    digitalWrite(NAV_LIGHTS_PIN, HIGH);
}

if (Comand == "NAV_OFF") {
    validComand = "Valid";
    digitalWrite(NAV_LIGHTS_PIN, LOW);
}

```

```

Comand = "";
Serial.println(validComand);
Serial1.println(validComand);
}
}

```

```

void Update_GCS(int update_time) {
    if (millis() - millisBS > update_time) {
        millisBS = millis();
        Serial.println("Frecuencia: " + String(frequency));
    }
}

```

```

// _____
_____//

```

```

//COMPASS LT-1000

```

```

Serial.println("LAT NMEA: " + String(Decode_NMEA.lat));
Serial.println("LON NMEA: " + String(Decode_NMEA.lon));
Serial.println("HEAD: " + String(Decode_NMEA.head));

```

```

Serial.println("COG: " + String(Decode_NMEA.COG));

Serial.println("XTD: " + String(XTD));

//_____
_____//

//COMPASS NMEA

char NMEA_compass [35];

byte csm; //checksum compass

PString strm(NMEA_compass, sizeof(NMEA_compass));

strm.print("$HDM,");

strm.print(lround(Decode_NMEA.head)); // lround simply rounds out the decimal,
since a single degree is fine enough of a resolution

strm.print(",M*");

csm = checksum(NMEA_compass);

if (csm < 0x10) strm.print('0');

strm.print(csm, HEX);

Serial1.println(NMEA_compass);

Serial.println(NMEA_compass);

//_____
_____//

//COMPASS CEOV USV STAT

char CEOV_USV_STAT [75];

byte csm_stat; //checksum compass

PString strm_stat(CEOV_USV_STAT, sizeof(CEOV_USV_STAT));

```

```

strm_stat.print("$stat,");

strm_stat.print(autonomous); // lround simply rounds out the decimal, since a single
degree is fine enough of a resolution

strm_stat.print(",");

strm_stat.print(control.motor_eb_speed); // lround simply rounds out the decimal,
since a single degree is fine enough of a resolution

strm_stat.print(",");

strm_stat.print(control.motor_bb_speed); // lround simply rounds out the decimal,
since a single degree is fine enough of a resolution

strm_stat.print(",");

if (Manual_head > 99) {

    strm_stat.print(Manual_head); // lround simply rounds out the decimal, since a
single degree is fine enough of a resolution

}

if (Manual_head > 9 && Manual_head < 100) {

    strm_stat.print("0");

    strm_stat.print(Manual_head); // lround simply rounds out the decimal, since a
single degree is fine enough of a resolution

}

if (Manual_head >= 0 && Manual_head < 10) {

    strm_stat.print("00");

    strm_stat.print(Manual_head); // lround simply rounds out the decimal, since a
single degree is fine enough of a resolution

}

strm_stat.print(",M*");

csm_stat = checksum(CEOV_USV_STAT);

```

```

if (csm_stat < 0x10) strm_stat.print('0');

strm_stat.print(csm_stat, HEX);

Serial1.println(CEOV_USV_STAT);

Serial.println(CEOV_USV_STAT);

//_____
_____//

//MOTOR NMEA

/*

char NMEA_MOTOR_EB [23];

byte cs_motor; //checksum compass

PString strm_motor(NMEA_MOTOR_EB, sizeof(NMEA_MOTOR_EB));

strm_motor.print("$rpm,E,1,");

strm_motor.print(control.motor_eb_speed); // lround simply rounds out the
decimal, since a single degree is fine enough of a resolution

strm_motor.print(".0,,A*");

cs_motor = checksum(NMEA_MOTOR_EB);

if (csm < 0x10) strm.print('0');

strm_motor.print(cs_motor, HEX);

Serial1.println(NMEA_MOTOR_EB);

Serial.println(NMEA_MOTOR_EB);*/

//_____
_____//

//GPS NMEA

char NMEA_gps_GPRMC [100];

```

```

byte cs_GPRMC; //checksum compass

PString strm1(NMEA_gps_GPRMC, sizeof(NMEA_gps_GPRMC));

strm1.print("$GPRMC,");

strm1.print(String(Decode_NMEA.Time));

strm1.print(",A,");

strm1.print(Decode_NMEA.NMEA_Lat + ",");

strm1.print(Decode_NMEA.lat_Signal + ",");

strm1.print(Decode_NMEA.NMEA_Lon + ",");

strm1.print(Decode_NMEA.lon_Signal + ",");

strm1.print(Decode_NMEA.Speed_Kts);

strm1.print(",");

strm1.print(Decode_NMEA.head);

strm1.print(",");

strm1.print(gps.date);

strm1.print(",002.1,W*");

cs_GPRMC = checksum(NMEA_gps_GPRMC);

if (cs_GPRMC < 0x10) strm1.print('0');

strm1.print(cs_GPRMC, HEX);

Serial1.println(NMEA_gps_GPRMC);

Serial.println(NMEA_gps_GPRMC);

//_____
_____//

/**WIND NMEA

/*   char NMEA_wind [23]; //ex: $NRMWV,179,R,55,N,A*1e

```

```

byte cs_wind; //checksum compass

PString strm2(NMEA_wind, sizeof(NMEA_wind));

strm2.print("$NRMWV,");

strm2.print(wind.Direction);

strm2.print(",R,");

strm2.print(wind.Speed);

strm2.print(",N,A*");

cs_wind = checksum(NMEA_wind);

if (cs_wind < 0x10) strm2.print('0');

strm2.print(cs_wind, HEX);

Serial2.println(NMEA_wind);

Serial.println(NMEA_wind);*/

//_____
_____//

//RUDDER NMEA

char NMEA_rudder [50];

byte cs_rudder; //checksum compass

PString strm3(NMEA_rudder, sizeof(NMEA_rudder));

strm3.print("$RSA,");

strm3.print(rudder.NMEA_rudder);

strm3.print(",A,");

cs_rudder = checksum(NMEA_rudder);

if (cs_rudder < 0x10) strm3.print('0');

strm3.print(cs_rudder, HEX);

```

```
Serial1.println(NMEA_rudder);
```

```
Serial.println(NMEA_rudder);
```

```
//_____
```

```
_____//
```

```
//CEOV WP_GO
```

```
char WP_INFO [80];
```

```
byte cs_WP_INFO; //checksum compass
```

```
PString strm_WP_INFO(WP_INFO, sizeof(WP_INFO));
```

```
strm_WP_INFO.print("$WP_INFO,");
```

```
strm_WP_INFO.print(wp.number);
```

```
strm_WP_INFO.print(",");
```

```
strm_WP_INFO.print(DistCalc);
```

```
strm_WP_INFO.print(",");
```

```
strm_WP_INFO.print(AziCalc);
```

```
strm_WP_INFO.print(",");
```

```
strm_WP_INFO.print(gps.GPS_head);
```

```
strm_WP_INFO.print(",");
```

```
strm_WP_INFO.print(gps.Speed_kts);
```

```
strm_WP_INFO.print(",*");
```

```
cs_WP_INFO = checksum(WP_INFO);
```

```
if (cs_WP_INFO < 0x10) strm_WP_INFO.print('0');
```

```
strm_WP_INFO.print(cs_WP_INFO, HEX);
```

```
Serial1.println(WP_INFO);
```

```
Serial.println(WP_INFO);
```

```
// _____  
_____//
```

```
//PITCH ROLL NMEA
```

```
/*
```

```
char shrSentence [50]; //ex:$IIXDR,A,5.0,,PTCH,A,12.0,,ROLL,*hh<CR><LF>
```

```
byte csp;
```

```
PString strp(shrSentence, sizeof(shrSentence));
```

```
strp.print("$IIXDR,A,");
```

```
strp.print(pitch, 1);
```

```
strp.print(".,PTCH,A,");
```

```
strp.print(roll, 1);
```

```
strp.print(".,ROLL,*");
```

```
csp = checksum(shrSentence);
```

```
if (csp < 0x10) strp.print('0');
```

```
strp.print(csp, HEX);
```

```
Serial1.println(shrSentence);
```

```
Serial.println(shrSentence);
```

```
*/
```

```
// _____  
_____//
```

```
//Wind NMEA
```

```
Serial1.println(Wind.NMEA_String_wind);
```

```
Serial.println(Wind.NMEA_String_wind);
```

```

    }
}

byte checksum(char* str) { //Checksum function

    byte cs = 0;

    for (unsigned int n = 1; n < strlen(str) - 1; n++)

    {

        cs ^= str[n];

    }

    return cs;

}

```

Apêndice 6 – Código controle

```

#include "Arduino.h"

class Control {

public:

    //*****

    float head_to_go = 270;

    int correction_factor = 45;

    int help_rotate_factor = 10;

```

```

int motor_eb_speed = 1500;

int motor_bb_speed = 1500;

int correct_motor_eb_speed = 1500;

int correct_motor_bb_speed = 1500;

//*****

void Update_head_to_go(float head, float COG, float XTD, float AziCalc, float
DistCalc) {

    head_to_go = AziCalc + angle_dif(head, COG) + (correction_factor * ( XTD /
DistCalc ));

}

float help_rotate(float head) {

float dif = angle_dif(head, head_to_go );

if (abs(dif) > 30) {

    correct_motor_eb_speed = motor_eb_speed + (dif * help_rotate_factor);

    correct_motor_bb_speed = motor_bb_speed - (dif * help_rotate_factor);

}

else {

```

```

correct_motor_eb_speed = motor_eb_speed;
correct_motor_bb_speed = motor_bb_speed;
}

if (correct_motor_eb_speed > 2000) {
    correct_motor_eb_speed = 2000;
}

if (correct_motor_eb_speed < 1000) {
    correct_motor_eb_speed = 1000;
}

if (correct_motor_bb_speed > 2000) {
    correct_motor_bb_speed = 2000;
}

if (correct_motor_bb_speed < 1000) {
    correct_motor_bb_speed = 1000;
}
}

```

```

float angle_dif(int a, int b) {
    int d = abs(a - b) % 360;
    int r = d > 180 ? 360 - d : d;

    //calculate sign

```

```

int sign = (a - b >= 0 && a - b <= 180) || (a - b <= -180 && a - b >= -360) ? 1 : -1;

r *= sign;

return (r);

}

```

```

void software_Reset() { // Restarts program from beginning but does not reset the
peripherals and registers

```

```

    Serial.println("going for reset");

    asm volatile (" jmp 0");

}

};

```

Apêndice 7 – Código de Leitura do NMEA

```

#include "Arduino.h"

```

```

class DECODE_NMEA {

public:

    //*****

    int length = 100;

    char NMEA_Sentence [100];

    String NMEA_Array [30];

    char termChar = '$';

    float lat = 0;

    float lon = 0;

```

```

float COG = 270;

float head = 0;

float ATM_Pressure = 0;

float Air_Temp = 0;

float Speed_Kts = 0;

float Time = 0;

float Date = 0;

String lat_Signal = "N";

String lon_Signal = "w";

String NMEA_Lat = "0";

String NMEA_Lon = "0";

//*****

void Update() {

    if (Serial3.available())

    {

        int numChars = Serial3.readBytesUntil(termChar, NMEA_Sentence, length);

        NMEA_Sentence[numChars] = '\0';

        String NMEA_String = String(NMEA_Sentence);

        if (NMEA_String.charAt((NMEA_String.length() - 5)) == '*')

        {

            char copy[NMEA_String.length() - 4];

            (NMEA_String.substring(0, NMEA_String.length() - 4)).toCharArray(copy,

NMEA_String.length() - 4);

```

```

byte cs = checksum(copy);

if (String(cs, HEX).equalsIgnoreCase(NMEA_String.substring(
NMEA_String.length() - 4 , NMEA_String.length() - 2))) {

    int b = 0;

    int array_p = 0;

    for (int i = 0; i < NMEA_String.length() ; i++) { //Cria array com as info
NMEA

        if (1) {

            if (NMEA_String.charAt(i) == ',' || NMEA_String.charAt(i) == '*') {

                NMEA_Array[array_p] = NMEA_String.substring( b , i);

                b = i + 1;

                array_p++;

                if (array_p > 9)

                    break;

            }

        }

    }

    if (NMEA_String.substring(0, 5) == "HCHDT") {

        head = (NMEA_Array[1].toFloat());

    }

    else {

        if (NMEA_String.substring(0, 5) == "GNRMC") {

            lat = NMEA_Array[3].toFloat();

```

```

lat = lat / 100;

int i = 0;

while (i < lat) {

    i++;

}

lat = ((( ((1 - (i - lat)) / 60) * 100) + (i - 1)) * 100);

lon = NMEA_Array[5].toFloat();

i = 0;

lon = lon / 100;

while (i < lon) {

    i++;

}

lon = ((( ((1 - (i - lon)) / 60) * 100) + (i - 1)) * 100);

if (NMEA_Array[4] == "S") {

    lat *= -1;

    // NMEA_Array[3] = "-" + NMEA_Array[3];

}

if (NMEA_Array[6] == "W") {

    lon *= -1;

    // NMEA_Array[5] = "-" + NMEA_Array[5];

}

COG = NMEA_Array[8].toFloat();

```



```

char buf[a.length()];

a.toCharArray(buf, a.length());

int WVal = atof(buf);

return WVal;

}

byte checksum(char* str) //Checksum function
{
    byte cs = 0;

    for (unsigned int n = 0; n < strlen(str) ; n++)
    {
        cs ^= str[n];
    }

    return cs;
}
};

```

Apêndice 8 – Código do GPS

```

#include "Arduino.h"

#include <TinyGPS.h>

class GPS {

    TinyGPS gps_nr1;

public:

    //*****

```

```

long lat, lon;

float GPS_head;

float Speed_kts;

unsigned long fix_age;

unsigned long time, date;

unsigned long time_GPS;

bool GPS_LOCK = false;

float NMEA_lat;

float NMEA_lon;

float LAT_NMEA;

float LON_NMEA;

//*****

void Update() {

while (Serial2.available()) {

    if (gps_nr1.encode(Serial2.read())) {

        gps_nr1.get_datetime(&date, &time, &fix_age);

        gps_nr1.get_position(&lat, &lon, &fix_age);

        Speed_kts = gps_nr1.f_speed_knots();

        GPS_head = gps_nr1.f_course();

    }

if (fix_age == TinyGPS::GPS_INVALID_AGE) {

    GPS_LOCK = false;

```

```

}

else if (lat < 1) {

    GPS_LOCK = false;

}

else {

    GPS_LOCK = true;

}

}

NMEA_lat = (lat / 100000.0000000);

// Serial.print("NMEA: ") ; Serial.println(NMEA_lat,6);

NMEA_lon = lon / 100000.0000000;

if (NMEA_lon < 0)

    NMEA_lon *= -1;

// Serial.print("NMEA lon : ") ; Serial.println(NMEA_lon,6);

LAT_NMEA = Convert_GGmm_ddddd(NMEA_lat);

LON_NMEA = Convert_GGmm_ddddd(NMEA_lon);

}

```

```

float Convert_GGmm_dddddd(float coordenada) {

    int x = coordenada;

    float y = (coordenada - x) * 60;

    String z = String(x);

    if (y < 10) {

        z = z + "0" + String(y, 6);

    }

    else {

        z = z + String(y, 6);

    }

    float w = z.toFloat();

    //w = w*100;

    return w;

}

float CalcDist(float flat1, float flon1, float x2lat, float x2lon) {

    // flat1 = our current latitude. flat is from the gps data.

    // flon1 = our current longitude. flon is from the fps data.

    flat1 = flat1 / 100;

    flon1 = flon1 / 100;

```

```

float dist_calc = 0;

float dist_calc2 = 0;

float diflat = 0;

float diflon = 0;

diflat = radians(x2lat - flat1);

flat1 = radians(flat1);

x2lat = radians(x2lat);

diflon = radians((x2lon) - (flon1));

dist_calc = (sin(diflat / 2.0) * sin(diflat / 2.0));

dist_calc2 = cos(flat1);

dist_calc2 *= cos(x2lat);

dist_calc2 *= sin(diflon / 2.0);

dist_calc2 *= sin(diflon / 2.0);

dist_calc += dist_calc2;

dist_calc = (2 * atan2(sqrt(dist_calc), sqrt(1.0 - dist_calc)));

dist_calc *= 6371000.0;

return (dist_calc);
}

float CalcAzi(float flat1, float flon1, float x2lat, float x2lon) {

flat1 = flat1 / 100;

```

```

flon1 = flon1 / 100;

flat1 = radians(flat1);

flon1 = radians(flon1);

x2lat = radians(x2lat);

x2lon = radians(x2lon);

float heading;

heading = atan2(sin(x2lon - flon1) * cos(x2lat), cos(flat1) * sin(x2lat) - sin(flat1)
* cos(x2lat) * cos(x2lon - flon1)), 2 * 3.1415926535;

heading = heading * 180 / 3.1415926535;

float head = heading;

if (head < 0) {
    head += 360;
}

if (head > 360) {
    head -= 360;
}

return (head);

}

```

```

float CalcCrossTrackError (float lat_wp_ini, float lon_wp_ini, float flat1, float
flon1, float lat_wp_fin, float lon_wp_fin) { // Calcula XTD

    float t13 = CalcAzi(lat_wp_ini*100.0, lon_wp_ini*100.0, flat1/100.0,
flon1/100.0);

    float t12 = CalcAzi(lat_wp_ini*100, lon_wp_ini*100, lat_wp_fin, lon_wp_fin);

    float d13 = CalcDist(lat_wp_ini*100.0, lon_wp_ini*100.0, flat1/100.0,
flon1/100.0) / 6371000.0;

    t12 = radians(t12);

    t13 = radians(t13);

    float CrossTrackError = asin(sin(d13) * sin(t13 - t12)) * 6371000.0;

    return (CrossTrackError);

}

```

```

void Test() {

    Serial.println("A testar GPS");

    bool a = true;

    String stringOne = "";

    while (a) {

        while (Serial2.available()) {

            char c = Serial2.read();

            if (c == '\n') {

                Serial.println(stringOne);

                stringOne = "";
            }
        }
    }
}

```

```

    }

    else {

        stringOne = String(stringOne + c);

    }

}

if (Serial.available()) {

    String Comand = Serial.readString();

    Comand = Comand.substring(0, Comand.length() - 1);

    if (Comand == "Normal") { // End Cicle, if the user send the comand "Normal";

        Serial.println("Normal mode");

        a = false;

    }

}

}

}

};

```

Apêndice 9 – Código do Leme

```

#include "Arduino.h"

#include <math.h>

class Rudder {

public:

    int angle = 68;

    int correction_speed = 1;

```

```

int NMEA_rudder = 0;

int last_rudder_order;

int control_rudder = 98;

int UpdateRudder1(float compass_head, int azi, int middle_rudder) {

    int delta = compass_head - azi; // calcula a diferença

    if (delta > 180) {

        delta = 360 - compass_head + azi;

        delta = -delta;

    }

    if (delta < (-180)) {

        delta = 360 - azi + compass_head;

    }

    angle = (middle_rudder + (correction_speed * delta));

    if (angle > middle_rudder + 30) {

        angle = middle_rudder + 30;

    }

    if (angle < middle_rudder - 30) {

        angle = middle_rudder - 30;

    }

    NMEA_rudder = (angle - middle_rudder);

    NMEA_rudder = -NMEA_rudder;

```

```

return (angle);
}

int UpdateRudder2(float compass_head, int azi, int middle_rudder) {

    NMEA_rudder = 0;

    int delta = compass_head - azi; // calcula a diferença

    if (delta > 180) {

        delta = 360 - compass_head + azi;

        // delta = -delta;

    }

    if (delta < (-180)) {

        delta = 360 - azi + compass_head;

    }

    angle = (middle_rudder + (correction_speed * delta));

    if (angle > middle_rudder + 30) {

        angle = middle_rudder + 30;

    }

    if (angle < middle_rudder - 30) {

        angle = middle_rudder - 30;

    }
}

```

```

NMEA_rudder = (angle - middle_rudder);

NMEA_rudder = -NMEA_rudder;

return (angle);

}

};

```

Apêndice 10 – Código dos Waypoints

```

#include "Arduino.h"

#include <PString.h>

#define MAX_WP_NUMBER 20

class WP {

public:

    /*******

float wplat[MAX_WP_NUMBER + 1];

float wplon[MAX_WP_NUMBER + 1];

int total_wp_number;

int number = 1;

int min_dist = 15;

int last_wp_number;

    /*******

    void Update() { // 5 casa decimais - precisão ~~1 metro || Primeira posição dos WP
    é 1 ( wplat[1], wplon[1] )

```

```

wplat[1] = 38.66816;
wplon[1] = -9.14634;

wplat[2] = 38.66878;
wplon[2] = -9.14656;

wplat[3] = 38.66835;
wplon[3] = -9.14569;

}

void wp_list() {
    char wp_list [600];
    byte csm_wp; //checksum compass
    PString strm_wp(wp_list, sizeof(wp_list));
    strm_wp.print("$wp,");

    int i = 1;
    while (wplat[i] != 0) {
        total_wp_number = i;
        strm_wp.print(i);
        strm_wp.print(",");
    }
}

```

```

    strm_wp.print(wplat[i], 6);

    strm_wp.print(",");

    strm_wp.print(wplon[i], 6);

    strm_wp.print(",");

    i++;
}

strm_wp.print("M*");

csm_wp = checksum(wp_list);

if (csm_wp < 0x10) strm_wp.print('0');

strm_wp.print(csm_wp, HEX);

Serial1.println(wp_list);

Serial.println(wp_list);
}

void test(float DistCalc) {

    if (DistCalc < min_dist)

        number++;

    if ((wplat[number] == 0) || (wplon[number] == 0))

        number = 1;
}

```

```

byte checksum(char* str) { //Checksum function

byte cs = 0;

for (unsigned int n = 1; n < strlen(str) - 1; n++)

{

cs ^= str[n];

}

return cs;

}

};

```

Apêndice 11 – Código do Anemómetro

```
#include "Arduino.h"
```

```

class WIND {

public:

char NMEA_Sentence_wind [100];

String NMEA_Array_wind [30];

char termChar = '\n';

String NMEA_String_wind = "";

int length = 100;

```

```
void Update() {  
  
    int numChars = Serial2.readBytesUntil(termChar, NMEA_Sentence_wind,  
length);  
  
    NMEA_String_wind = String(NMEA_Sentence_wind);  
  
    }  
  
};
```