Second Year of engineering school Internship
2023-24

" Work with autonomous system especially maritime vehicle, and software development "

22/04/2024 - 24/08/2024

Author
Arthur Varré

Supervisor
Ola Benderius
Krister Blanch

Referent teacher
Cédric Anthierens

Specialization
SYSteme MEchatronique et Robotique

Engagement de non plagiat.

Je soussigné, …Arthur Varré………………………………………………………………………………

N° carte d'étudiant : …22204729………………………………………………………………………

Déclare avoir pris connaissance de la charte des examens et notamment du paragraphe spécifique au plagiat.

Je suis pleinement conscient(e) que la copie intégrale sans citation ni référence de documents ou d'une partie de document publiés sous quelques formes que ce soit (ouvrages, publications, rapports d'étudiants, internet, etc….) est un plagiat et constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.

En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour produire et écrire ce document.

Fait le …18/08/2024

Signature(s)

Arthur VARRÉ

Ce document doit être inséré en première page de tous les rapports, dossiers et/ou mémoires.

Document du chapitre 10 annexe 5, issu de la Charte des examens adoptée en Conseil d'Administration le 11 juillet 2013 après avis du CEVU du 27 juin 2013 - Délibération N°2013-73 – Modifié suite au CFVU du 12/03/2015.

1 Words summaries

- Responsibility
- Independence
- Research
- Integration
- learning

- Responsabilité
- Indépendance
- Recherche
- Integration
- Apprentissage

Contents

2 Glossary

ABS	Acrylonitrile butadiène styrène
ASA	Acrylonitrile styrène acrylate
CTH	Chalmers University of Technology
C2B2	Mistra Co-Creating Better Blue
ESC	Electronic speed control
IMU	Inertial measurement unit
KTH	Royal Institute of Technology in Stockholm
LTH	Lunds Tekniska Högskola
PC	Polycarbonate
PCB	Printed circuit board
PETG	Polyethylene terephthalate Glycol
PLA	Acide polylactique
ROV	remotely operated underwater vehicle
USV	Unmaned Surface vehicle
UV	Ultra violet

3 Acknowledgements

I would first like to thank Ola Benderius, my supervisor, for his trust throughout this internship. He allowed me a great deal of freedom in my work and was always enthusiastic about new ideas. Secondly, I want to thank my co-supervisor, Krister Blanch, who taught me a lot and was incredibly helpful whenever I got stuck. He was also the co-worker who always believed in the realization of the project. I also appreciate Daniel Poveda, the R&D engineer, for his invaluable help at the REVERE. As Anna the lab manager who's allow me in high quality workplace.

I would like to thank the entire vehicle safety department for their warm welcome during these four months. Lastly, a special thanks to Norah, Fanny, Ia, Linda, and Bjornborg for their help and, most importantly, their friendship, which provided me with exceptional comfort in my work.

4 Introduction

This internship took place at Chalmers University of Technology. I worked and assisted researchers in the Department of Maritime and Vehicle Safety. My primary focus was to bring expertise in maritime studies and contribute to software development. This broad internship subject translated into providing quick assistance on various projects, and most importantly, developing a maritime Unmanned Surface Vehicle.

These weeks felt like a continuation of my studies at engineering school, adding a professional dimension and providing insight into the work of a researcher. Additionally, the focus on maritime vehicles, which often involves mechanics, electronics, and software development, perfectly aligned with my specialization in mechanics and robotics. That specialization also provided me with many tools that were useful during this internship and helped me to be as helpful as possible to the different co-workers I interacted with.

Working in a foreign country was also a new experience. Given the importance of English proficiency for engineers, an internship abroad was an excellent opportunity to confirm my ability to think and communicate my research in English and to be well understood in an international context.

5 Chalmers University

5.1 The University

Chalmers University of Technology in Goteborg, Sweden, is a private institution established in 1829. It is one of Sweden's three main technology universities, alongside the Royal Institute of Technology (KTH) in Stockholm and the Faculty of Engineering at Lund University (LTH). The university operates under the governance of the Chalmers University of Technology Foundation (Chalmers tekniska högskola AB), which is accountable to the Swedish government.

Chalmers is known for its wide education spectrum with more than 40's different masters in 13 main fields. Moreover Chalmers is today an international university with researchers and students coming from all around the world and masters that are taught entirely in English.

The university combines theoretical knowledge but also practical application with a strong link with the local industry such as Volvo. And some space as the FuseLab allowed the realisation of prototype and construction in general. It also encourages student to take part in a lot of project. Chalmers is also a university that tried to challenge environmental issues and with some projects such as biodiversity's studies or sustainable transformation of energies.

Overall, Chalmers University of Technology is not only a center of academic excellence but also a huge community of research lab and experimental project for the future.

5.2 Lab and Research

Chalmers is organized into 13 research departments, each specializing in different fields of technology and science. These departments cover a broad range of disciplines, ensuring a comprehensive approach to research and innovation. The university's research strategy emphasizes interdisciplinary collaboration, integrating expertise from various fields to tackle complex global challenges.

Key areas of research at Chalmers include sustainable energy systems, materials science, information and communication technology, and life sciences. The university is particularly noted for its work in environmental technology and sustainable development, aiming to contribute significantly to the global sustainability agenda.

My internship took place in two different departments: firstly the vehicle engineering and autonomous system, and secondly the REVERE (Research on Vehicle Research) lab. This lab hosts a variety of projects and has recently expanded into maritime research. The following sections detail my work in this lab during the internship.

6 Mission

My mission at Chalmers was initially broad and not precisely defined, as my internship subject stated that I was expected "to work with autonomous vehicles, especially marine vehicles, and other software-related tasks." This began with conceptual work on a project named "Sea Pickle", which served as an introduction to my main internship project and provided insight into the lab's work processes.

The main focus of my internship was the "Sea Dragon" project. This project aims to improve the efficiency of rescue operations for drowning individuals, an idea originating from Umeå University. The concept was developed in a master's thesis and later brought to

Chalmers University, where it became part of a joint research project under my co-supervisor, Krister Blanch and professor Andreas Claesson from Karolinska Institute. You can find attached (22) a timeline of my work.

7 Sea Pickle

7.1 Swedish Approach to the Blue Economy

The primary purpose of the Sea Pickle project is to raise awareness about marine life and the importance of preserving the sea. This project exemplifies the strong environmental consciousness of the Swedish people.

As global warming increases and environmental issues become more significant, the Mistra Co-Creating Better Blue (C2B2) initiative was born. According to the documentation found at the website dedicated to C2B2 [2]: This initiative focuses on transformative change for the blue economy (Maritime economy) through scientific data collection, participatory governance, and industry involvement. Overall, the C2B2 initiative seeks to co-create solutions for healthier marine and coastal environments through collaborative research, stakeholder involvement, and innovative practices. The Sea Pickle fits into the data collection aspect of this initiative. The project has been featured on national TV, generating significant viewer interest and demonstrating the potential impact of the C2B2 initiative.

7.2 Sea Pickle Description and Improvement

The Sea Pickle is originally a pickle tank used to place sensors inside and submerge it underwater. The idea of the using something like a pickle tank is to make it really easy to use and cheap.

Currently the sea pickle use a tube made of acrylic and close by two leads in aluminum the whole things comes from Blue Robotics. This project was developed at the Chalmers

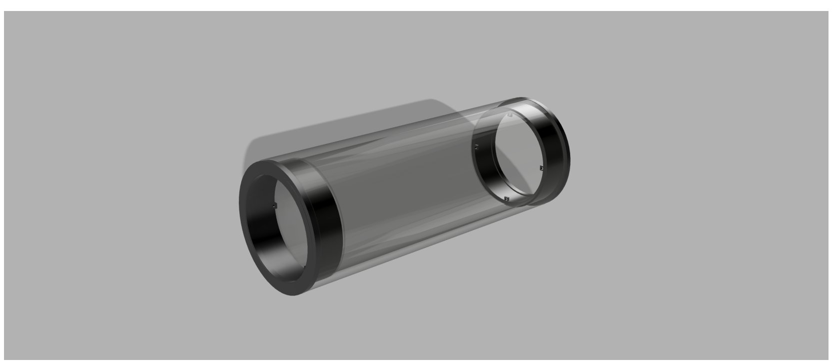
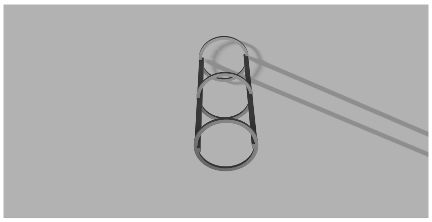


Figure 2: Sea Pickle Tank structure supporting the sensors is simply placed inside.

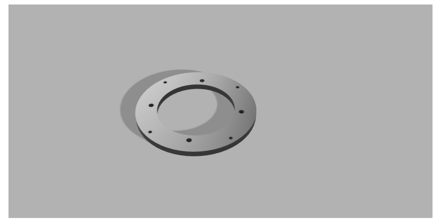
Laboratory of Mechanics and Maritime Technology to create a passive ROV that can monitor underwater activities near Goteborg as fish activities but or just biodiversity's in general.

The project is nearing completion when I arrived. But the team that handles this project encountered some issues during the test. One problem was that the structure was not securely fixed inside the tank. That non secure inside structure can cause some moves on the sensor and in the worst case unplug cables or make faulty connection. So we needed to devise a solution for that.

Initially, I considered a rail system that could attach directly to the plugs of the two parts of the tank (see Figure 3a). This option seemed promising because it would make it easy to remove the contents. However, the 3D printing lead time was too short, so we had to come up with an alternative solution, which I also designed. The alternative solution is a platform



(a) Rail structure idea



(b) Platform structure idea

Figure 3: External Structure idea

that attaches to the cap of the tank (see Figure 3b). One can secure all the equipment on top of it, similar to assembling a cake.

We completed the assembly and went to the Börno water monitoring station to do a submersion test. Currently, it is still deployed in the fjord near this station. This compact tank houses two cameras, two temperature sensors, one depth or pressure sensor, and a light controlled by a Raspberry Pi. All the equipment is connected to the surface via a 50-meter cable. The data are collecting to be stream in live for everybody in Sweden.

The Sea Pickle was the first project I worked on. I spend a bit more than a week on this project and didn't encounter difficulties. The main purpose for me was to integrate myself in the work team and get use to how people work there. After the test at Börnö, I moved on to the main project of my internship.

8 Sea Dragon

8.1 General Presentation and Objectives

Drowning accidents are a significant issue in Sweden and in the world. Drowning represents the third cause of mortality with more than 3400in the world accidents each year [3]. Several factors contribute to this, including Sweden's vast water coverage (almost 9% of the territory) and over 11,000 km of coastline. Swimming in cold water is common, and many of these accidents are related to swimming.

To address this issue, an idea was developed at Umeå University (see [5] for more details): build an Unmanned Surface Vehicle (USV) that can detect drowning individuals and alert rescue services for rapid intervention. This project, named "Sora", was transferred to Chalmers University for further development as another master's thesis, looking at the potential for additive manufacturing for this project. When I arrived, the thesis had been completed, and the next iteration was already well-underway under the supervision of Krister Blanch, a PhD student at Chalmers and my co-supervisor. I discovered the project as a prototype consisting of a buoy and a wooden structure housing a BlueView sonar. This prototype, named Sea

Dragon, had already been tested few times to get data from the sonar for Krister Blanch for unrelated PhD research.

My first task was to add remote control capabilities, but my broader objective was to develop a more comprehensive design for the USV, considering real-world conditions and the necessary electronics.

8.2 Prototypes: Issues and Improvements

The initial prototype, consisting of a buoy and wooden structure, required enhancements to include remote control capabilities. My first job was to design a low-cost structure that could house a watertight box for electronics and accommodate thruster mounts. Given the need for quick testing, the solution had to be both time-efficient and budget-friendly. I addressed these challenges with the design represented Figure 4. I selected a box with IP67 rating for

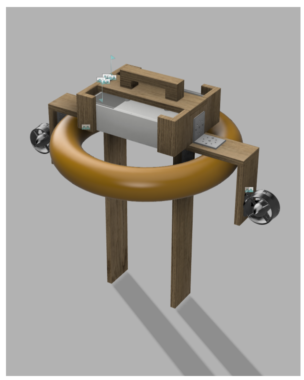


Figure 4: Remote control prototype design

temporary waterproofing of the electronics. Inside the box, we used a Raspberry Pi with a custom cap designed for another project. This setup included a buck converter to connect a 12V battery and power the Raspberry Pi, which also powered the sonar directly through the 12V battery (with a fuse for safety). The sonar was connected to the Raspberry Pi via Ethernet for data collection and transmission, with an integrated antenna for communication. The electronics scheme of the Sea Dragon is shown Figure 5.

The testing of this prototype revealed several issues. This test simply consists in using the USV as we want to, so in the sea in remote control. First, the USV's autonomy with a 12V lead battery was only about 15 minutes, which is insufficient for practical use. Additionally, the buoyancy was inadequate, as the electronics and wooden structure caused the USV to sink in rough waves. We addressed the buoyancy issue by adding a buoy above the sonar, which allowed for further testing and improvements in data processing and remote control using a gamepad.

These tests were crucial for designing the next version of the Sea Dragon, providing insights into the project's requirements, challenges, and potential improvements.

8.3 Sea Dragon 2.0

The goal of this version of the Sea Dragon was to address previous issues while creating a professional-looking and enhanced USV. The design included improvements such as an IMU

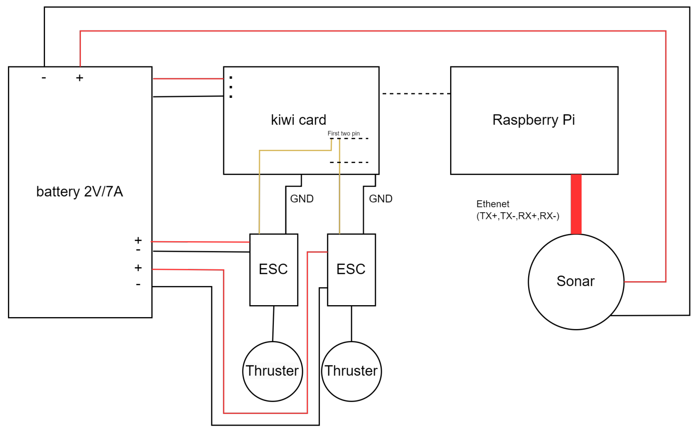


Figure 5: Electronics scheme

and adjustable sonar angles. The requirements were as follows:

- Can be printed on a 3D printer (dimension: 200x200x200 mm)
- Avoid buoyancy issues
- Can be thrown from a boat or dock into the water
- Operates in both the littoral and lakes around Goteborg
- Integrates two thrusters and a sonar
- Allows for remote/autonomous behavior
- Protects all electronic components from water
- Improves hydrodynamics to conserve energy
- Can be used as a temporary buoy

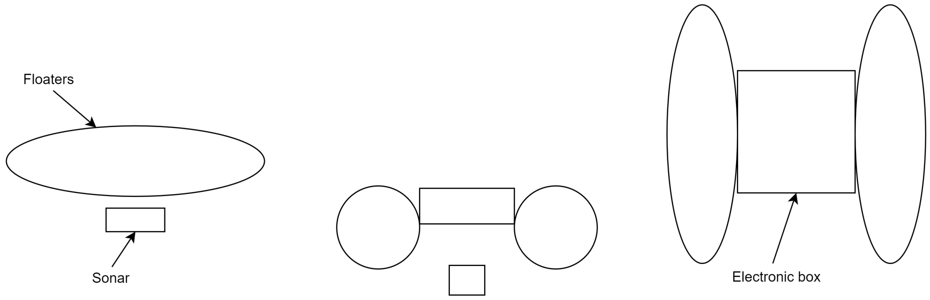
To ensure stability at sea, I proposed a catamaran design with two floaters (illustrated on Figure 6). The electronics are safely housed on the middle bridge, and the sonar, being quite heavy, helps lower the center of mass.

8.3.1 Conception

Archimedes' Principle. Addressing buoyancy was the first priority. Table 1 gives a list of the items housed on the Sea Dragon and their weights in water.

The floaters use a 4mm plastic hull, considering a plastic density of $\rho_H = 1.28 g/cm^3$ (This choice will be explained in section 8.3.2). The volume of air inside the 4mm hull needed for sufficient buoyancy was calculated using the static balance of forces (see Figure 7). The forces are defined as:

$$F_{Arch} = \rho_{\text{liq}} * V_{\text{submerged}} * g$$



(a) Side view (b) Front view (c) Top view

Figure 6: Concept sketch

Name	Weight (Kg)
Sonar	1.2
Box	1
Electronics card	0.2
Battery	2.5
Thruster	0.16
Total	5.06

Table 1: List of the items housed on the Sea Dragon and their weights in water.

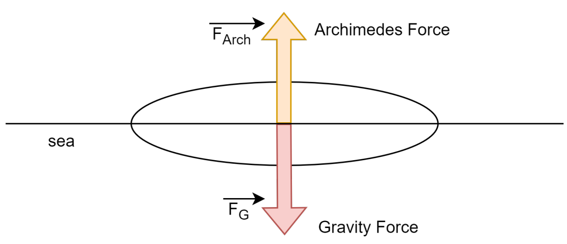


Figure 7: Force Balance Scheme of the Sea Dragon's floaters.

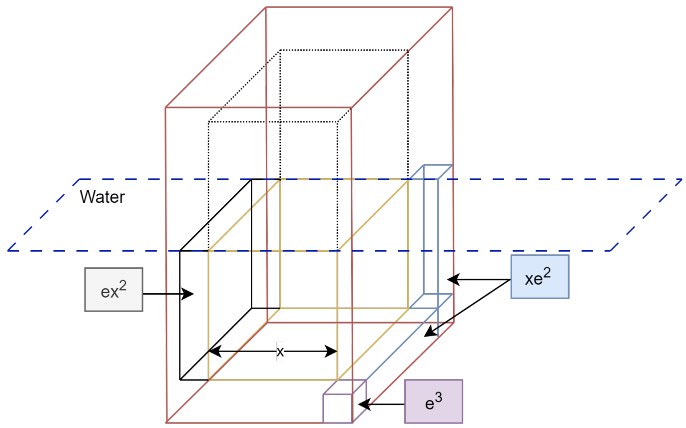


Figure 8: Scheme of the hull and air simplification

and

$$P = g * (M_{stuff} + V_H * \rho_H)$$

where:

- ρ_{liq} is the density of water (pure or salt)
- $M_{stuff} = 5.06$ Kg, the weight of components
- V_H and ρ_H are the volume and density of the hull

To ensure buoyancy, P must equal F_{Arch}. The floaters were modeled to submerge halfway. The floaters' air volume was determined considering the floaters' design constraints and material properties, such as density for plastic and different types of water (pure or salted).

Plus we can established if we considered x^3 as the volume of air inside a hull of thickness named e (see fig 8) :

$$V_{\text{submerged}} = x^3 + 4e^3 + 5ex^2 + 8xe^2$$

and

$$V_H = 2 * (5ex^2 + 4e^3 + 8xe^2)$$

Plus we can estimate that with all the mounting interface we can add 150% of the weight of the the hull. And add one kilo for the structure that will hold the sonar.

We can now associate two function that depend of x for the Gravity force and the Archimedes force.

$$F_{\text{Arch}} = 1000 * (x^3 + 0.02x^2)$$

and

$$F_{\text{Gravity}} = 6.06 + 2.5 * 2 * 1280(0.02x^2)$$

we simplify the terms $4e^3$ and $8xe^2$ at 0 because there are enough close to 0 to not influence the result. Finally we get the curve (see fig 9) and $x = 0.22$ so a volume of air of $V = 0.0107m^3$

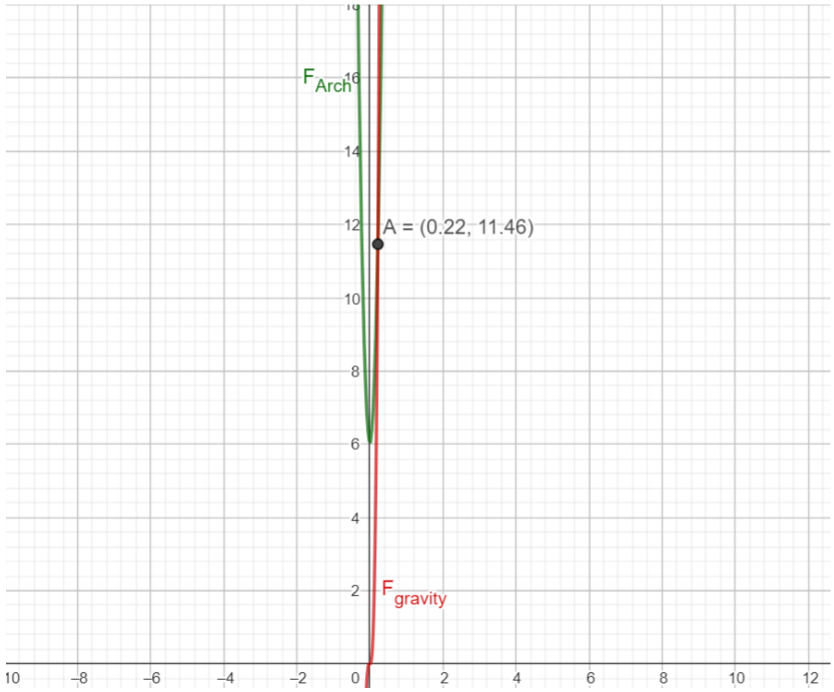
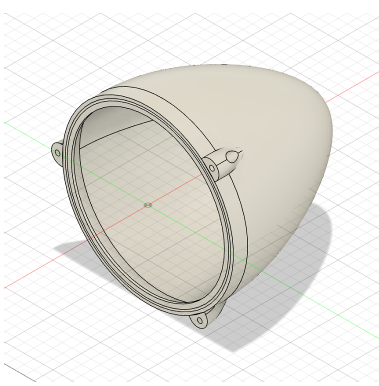


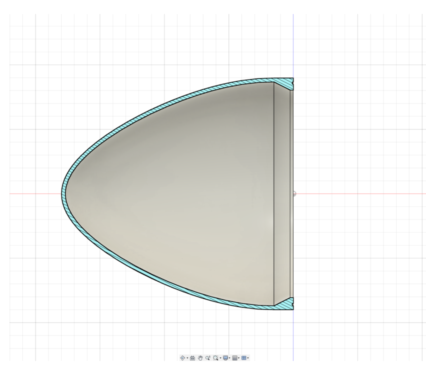
Figure 9: F_{Arch} and $F_{Gravity}$ intersection in function of x

Floaters Design. Given the printer's size limitation (200mm x 200mm), the floaters were designed in four parts, joined by screws and O-ring. They include a mounting interface for the electronics box and the thrusters, with aluminum used for structural strength due to the weight of the box and sonar. The floaters also feature reinforcement triangles for strength, especially if used as a buoy.

The floater's design achieved a sufficient internal air volume to ensure buoyancy, accounting for additional weights from mounting interfaces and screws. The floater ends (leads) were designed for hydrodynamic efficiency to reduce thruster energy consumption (see Section 8.3.1).



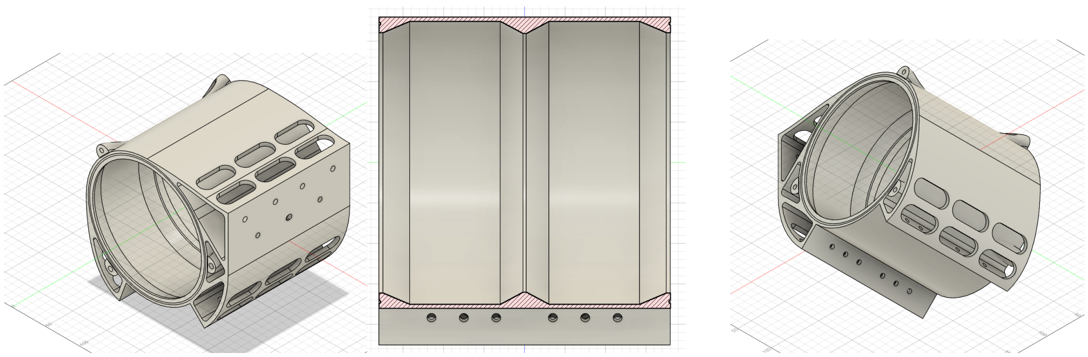
(a) Up Forward view



(b) Side Slice

Figure 10: Lead of the floaters

The leads were simpler, primarily designed to mount with the middle parts and provide space for seals, ensuring watertightness (see Figures 10 and 11). The overall air volume in the floaters exceeded the buoyancy requirements, allowing for safety margins and additional weight from components.



(a) Up Forward view (b) Side Slice (c) Down Backward view

Figure 11: Middle part of the floaters

The final component of the floaters was the thruster mounting interface. The Blue Robotics T200 thruster mounting plates were reused, and a custom piece was designed to secure these plates. With the floaters completed, the focus shifted to design the structure to house the electronics box and sonar.

Hydrodynamics. As we know that the drag coefficient depends on the shape, we use a table that resumes the drag coefficient for common shapes (see Table 2). So now we would like

Shape	**Drag Coefficient (C_d)**
Sphere	0.47
Half-sphere	0.42
Cone (vertex first)	0.55
Cube	1.05
Long cylinder (perpendicular to flow)	0.82
Short cylinder (parallel to flow)	1.15
Streamlined body (like an airfoil)	0.04
Flat plate (perpendicular to flow)	1.28
Rectangle (perpendicular to flow)	2.1

Table 2: Common Shapes and Their Drag Coefficients

to know the efficiency of the new floaters design compared to the old one. For that, we will consider a 2D representation of both prototypes to establish the push force from the thruster and the drag force due to the shape of each prototype in function of the speed.

Firstly we know from the Thruster Data-Sheet (see [6]) that they provide a forward force of $F_T = 36N$. Secondly we have to know the drag force D for each shape (the old and the new ones). We will consider both of them running straight on the x axis without any acceleration and $D = 1/2\rho v^2 C_d A$ (see [15]) with C_d the drag coefficient and A the slice area.

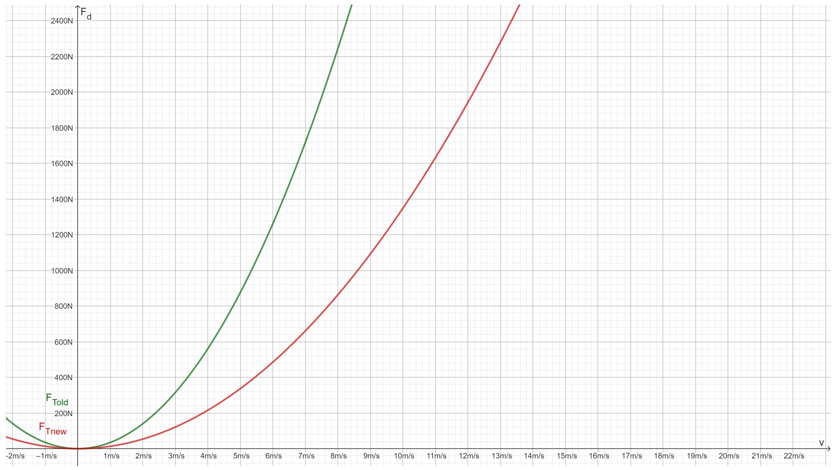


Figure 12: Comparison between D_{old} and D_{New}

- On the old prototype the slice area on the x axis will be a rectangle of 0.6m long and 0.1m wide (dimension of the buoy) that gives me an area $A_{old} = 0.06m^2$ and $C_{d_{old}} = 1.17$ the drag coefficient for a disk with is really close to the shape of the buoy. This gives $D_{old} = 1/2 * 1000 * v^2 * 1.17 * 0.06$
- On the new prototype we can refers to the NASA studies on the drag coefficient of a parabolic shape [9]. Indeed the conditions are similar. We take $C_{d_{new}} = C_{d_{cone}}/2$ (cones can have different drag coefficient in my case 0.55 seems pretty accurate because we can put an inner cone of 30° angle inside the lead of the floater refers to the course [10]). Plus the slice area of a floater is a disc of $0.18m$ but only half of the floater will be in the water so $A_{new} = 0.5m^2$ it finally give us a force that we have to multiply per 2 because we have two floaters $D_{new} = 2 * 1/2 * 1000 * v^2 * 0.27 * 0.05$

That gives the graph presented Figure 12 where we can spot a big difference between the two drag forces but not on really slow speed. That's why now we would like to have an idea of the speed we can reach.

From the static balance of force we can state:

$$F_T = D = 1/2\rho v^2 C_d A$$

and

$$v = \sqrt{\frac{2 \times 36}{\rho C_d A}}.$$

Then we find for the old prototype: $v = 1.01m/s$ and for the new $v = 1.63$. Once we know that the USV will move at least around 1m/s, we can know the efficiency $Ef = \frac{P_{out}}{P_{in}}$) of both prototypes at a speed of 1m/s where $P_{out} = (\vec{F_T} + \vec{D}).\vec{v}$ and $P_{in} = \vec{F_T}.\vec{v}$ since v is only on x axis we can simply the scalar product and we get $Ef = \frac{F_T - D}{F_T}$ and finally get :

$$Ef_{new} = 0.62$$

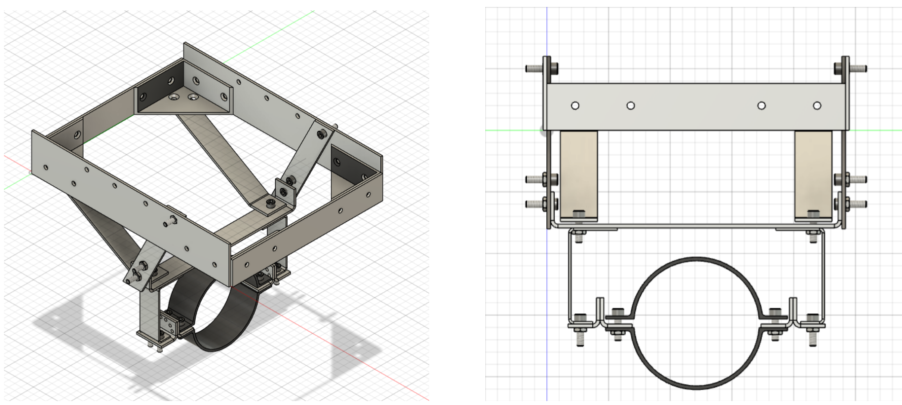
and

$$Ef_{old} = 0.025.$$

So finally with the new design we made a gap of 60% on the efficiency of the propulsion on the Sea Dragon.

Sonar and Box Structure. The sonar and electronics are the heaviest parts of the Sea Dragon, necessitating a metal structure for durability and vibration minimization. The design also needed to allow for different sonar angles (0°, 7.5°, 15°) and provide handles.

The initial structure design was overly complex for local fabrication capabilities, necessitating a simpler design that could be produced with available resources. This simplified structure included a reinforced bar for added strength (see Figure 13) and was successfully manufactured with the help of the FUSE lab.



(a) Up Forward view (b) Forward View

Figure 13: Sonar and box structure

The completed USV (see Figure 14), combining the floaters and the structure, marked the next phase of the project: material selection for manufacturing.

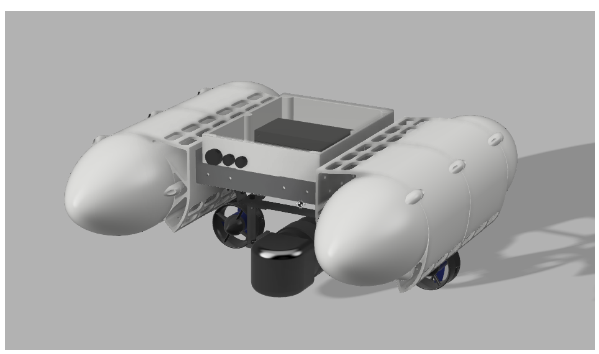


Figure 14: Complete Sea Dragon

8.3.2 Printing and Structure Material

The choice of material was particularly challenging for me, as it was a completely new area of study. In contrast, I felt more confident with the other aspects of the project due to the various courses I had taken at Seatech throughout the year.

Printing Material. In 3D printing, there are some basic filament types, such as ABS, PLA, and PETG, which are commonly used, and the knowledge about them is well-advanced. However, there are also many blends, like carbon fiber with PETG or glass fiber with ABS, and other fibers such as nylon, ASA, or PC. With all this diversity, I realized how complex material choice can be and how important it is to know exactly the intended use conditions of the USV.

The Sea Dragon will be used in lakes and at sea, which means our floaters must be watertight and resistant to saltwater. Moreover, the sun can be a threat as well, and finally, the material must have good shock resistance, as previously required.

I started by studying PLA and ABS (which are the two filaments already available in my workplace). This study is available in the annexes. To clarify the properties of each material, all the filaments and their properties is provided Table 3.

Material	**Advantages**	**Disadvantages**
Acrylonitrile Butadiene Styrene (ABS)	Good impact resistance, UV and chemical resistance, easy to post-process (sanding, painting)	Tendency to warp during printing, may require a heated print chamber for best results
Polyethylene Terephthalate Glycol-Modified (PETG)	Good water resistance, chemical and UV resistance, easier to print than ABS, good impact resistance	Can be less rigid than ABS
Nylon (Polyamide)	Excellent water and abrasion resistance, very impact-resistant, flexible	Absorbs moisture, which can affect mechanical properties, more difficult to print, requires good moisture management before and during printing
Acrylonitrile Styrene Acrylate (ASA)	UV and weather resistance, good impact and chemical resistance, similar to ABS but with better UV stability	May require a heated print chamber
Polycarbonate (PC)	Excellent impact resistance, high heat resistance, good water resistance	Difficult to print, requires high printing temperatures and a heated print chamber, can warp

Table 3: Filaments and their properties (according to multiple studies [13, 14]).

As we need large, watertight pieces with good shock resistance and ease of printing, PETG seems to be the best choice for our case, especially since rigidity is not our main concern.

Structure Material. The second material choice was for the structure. The requirements are quite simple: We need something with good rigidity and strength, that is also lightweight and not overly sensitive to saltwater.

Aluminum seems to be a very good choice. It is relatively inexpensive, can be coated to protect against water, is easy to manufacture, provides good strength, and is the lightest of the common metals.

8.3.3 Simulations

As the structure is finished and I'm sure of the material I will use, it's time to ensure that our buoyancy calculation is still accurate, even with all the structural elements that do not contribute to buoyancy.

To resume, we have an inside air volume underwater of

$$V_{tot} = V_l + V = 0.0084 + 0.0043 = 0.0127\, m^3$$

and the entire structure weighs around

$$M_{tot} = 2M_{floaters} + M_{Structure} + M_{Stuff} = 2 * 2.780 + 1.270 + 5.06 = 11.86\, kg.$$

We want to have a balance between Archimedes' force and gravity force, and we found from the previous results

$$F_{Arch} = \rho_{water}\, g\, V_{tot} = 1000 * 9.81 * 0.0127 = 124\, N$$

and

$$G = M_{tot}\, g = 11.86 * 9.81 = 116\, N.$$

So, we have the right buoyancy with an 8-newton margin. These results are quite safe because they are calculated using just half of the total buoyancy capacity of the USV, giving us good behavior even if someone were to hold onto it.

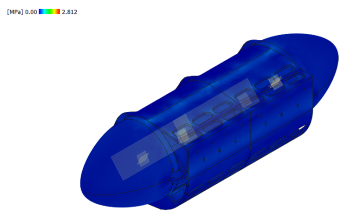
The objective of the simulation was to have a general idea of how the piece will behave. We didn't aim for very precise results but needed to know if the structure made sense. Simulations will not be done on the entire USV: the data calculation is too heavy for my computer. Thus I decided to conduct simulations on each major part: the floaters and the aluminum structure. The results were interpreted using the Von Mises criterion to determine if the parts remain within the elastic domain. The details of this theory will not be covered here since I was assisted by the simulation software used for the analysis, but one can find further explanation in [8]. It was also an opportunity to detail precisely the forces that will apply to the USV, their values, directions, and application points.

Floaters Simulation. First, I wanted to check if the structure connecting the floaters and the box is strong enough to hold the aluminum structure and the box. To do this, we fix the floaters and apply a force :

$$P = \left(\frac{M_{stuff}}{2} + \frac{M_{Structure}}{2} \right) g = \left(\frac{1.270}{2} + \frac{5.06}{2} \right) * 9.81 = 31\,\mathrm{N}.$$

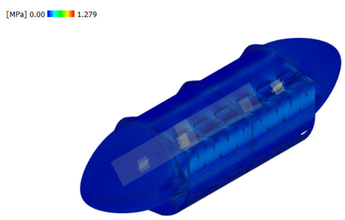
We divide the mass by two because there are two floaters, each supporting half of the weight.

However, it is also possible that one of the floaters might support the entire structure under certain conditions, such as waves. In such a scenario, the direction of the gravitational force vector could change, potentially becoming perpendicular (90°) to the interface. As observed, the maximum stress on the part in both (Figure 15 and 16) cases is below the elastic limit. We can conclude that the floaters should be sufficiently resistant. However, a real test is needed to confirm this.



Property of PETG	Minimum	Maximum
Young's Modulus (MPa)	1523.00	1523.00
Elastic Limit (MPa)	23.00	23.00
Von Mises Stress (MPa)	0.00	2.812
Global Deformation of the floaters (mm)	0.00	0.026

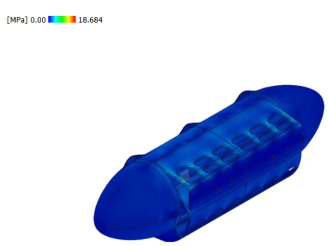
Figure 15: Result on the behaviour of the floaters under gravity stress on a flat ground by holding the electronic box.



Property of PETG	Minimum	Maximum
Young's Modulus (MPa)	1523.00	1523.00
Elastic Limit (MPa)	23.00	23.00
Von Mises Stress (MPa)	0.00	1.27
Deformation (mm)	0.00	0.012

Figure 16: Holding Box at 90° Result.

The second main simulation is to check if the floaters themselves remain in their state and have any deformation, as this could affect their watertightness if someone uses them as a buoy. To simulate the worst case, I will apply the weight of an upper body to the middle part of the floaters. For the use of the USV as a buoy, we consider the gravity force implied by an upper body (assuming the worst case) as $F_{UpBody} = M_{upper}\, g = 261\,\mathrm{N}$. I apply this force only on the top of one middle part (see Figure 17).

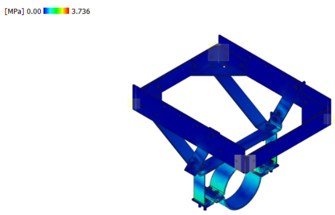


Property of PETG	Minimum	Maximum
Young's Modulus (MPa)	1523.00	1523.00
Elastic Limit (MPa)	23.00	23.00
Von Mises Stress (MPa)	0.00	18.68
Deformation (mm)	0.00	0.73

Figure 17: Upper Body.

As we see somebody holding on the floaters can create a deformation close to 1mm so it could create leak in the floaters. Our decision was to keep this design for the moment as we want to build a prototype quite soon. However we keep in mind that we will have to find a solution for the final version.

Finally we can check if the aluminium structure is strong enough to hold the sonar, considering the sonar outside of the water (I take the worst case, it will be lighter in the water). As we see the elastic limit of the Aluminium is wide above the one we hit with the sonar weight (Figure 18. Plus the shape of the structure make the aluminium strip work on different axis so it give me a rigid structure.



Property of Aluminum	Minimum	Maximum
Young's Modulus (MPa)	68900.00	68900.00
Elastic Limit (MPa)	275.00	275.00
Von Mises Stress (MPa)	0.00	3.73
Deformation (mm)	0.00	0.02

Figure 18: Behaviour of the Structure under stress due to Sonar weight

8.3.4 Integration and Software Development for an IMU

Software Development. With the evolution of the USV, we would like to include an IMU on board. To detect drowning people, the IMU can help us precisely locate the detection. We plan to use the IMU 6550 from Invensense. This IMU includes a gyroscope and an accelerometer. Initially, I used this IMU on the Adafruit board and had to develop software to read the data and calibrate the IMU.

To get data from the IMU, I first checked the register map (see [11]) of the IMU, which shows where all the data are stored and details how the IMU works. IMUs generally operate with a sleep mode, where they collect data and, at each time interval, send an interruption to the system that uses the IMU, which then retrieves all the data from storage (LIFO, FIFO, etc.). This mode saves a lot of battery. However, since the sonar already consumes a lot of power, we do not need this sleep mode, so we will directly access the data at the addresses of the accelerometer and gyroscope.

Finally, we used an I2C connection between our Raspberry Pi and the IMU board. The address to find the IMU on the Raspberry Pi is 0x68 (see [11]) . Several steps must be completed before using the IMU. First, we wake up the IMU by setting the power supply register on, by writing the bit 0x01 to the power supply register.

We will do the same to select the range scale of the accelerometer and the gyroscope. Once this is done, we can start reading the data by querying the accelerometer or gyroscope registers. Data from the accelerometer and gyroscope can be read from two registers: 0x3F (high-order bits) and 0x40 (low-order bits). To obtain the desired data, we need to combine and convert these two bytes. The code performing these steps can be found in annexes (see Section 11.2).

After collecting data, we also need to create a calibration function. Ideally, calibration would involve aligning each axis straight up and checking if they measure around $9.81\mathrm{m/s}^2$. However, for ease of calibration, we will calibrate only the Z-axis for gravity, and the other two axes should be zero for the accelerometer. For the gyroscope, all values should be zero if the USV is not moving. To ensure calibration accuracy, I decided to take the median of 2500 values; if this median is within ± 0.1(deg/s for the gyro meter or m/s accelerometer) of the expected value, we consider the IMU calibrated. This value of calibration is high but it can be explained as the Imu we use is not expected to be precise but to roughly estimate our rate of turn.

Finally, the calibration values are used as offsets. Depending on the sign of the current data, we will add or subtract the offset. Once this is done, all data are sent from the USV to the local network of the Sea Dragon via the OpendLV middleware (see [17] and [18]). We then connect to this network to retrieve the data. I didn't work much on this part, but it uses the same system as for sending steering commands. The code is also available in annexes (see Section 11.2).

PCB Design. The final task I performed for this USV was designing a PCB to house the IMU and to power both the Raspberry Pi and the IMU from a 12V battery. For this purpose, we used two buck converters: one from 12V to 5.5V and another from 12V to 3.3V. Additionally, to integrate the IMU and the buck converters, I used some resistors and capacitors. Schematics and PCB design can be found on Figure 19. The PCB was designed to be directly plugged into the Raspberry Pi.

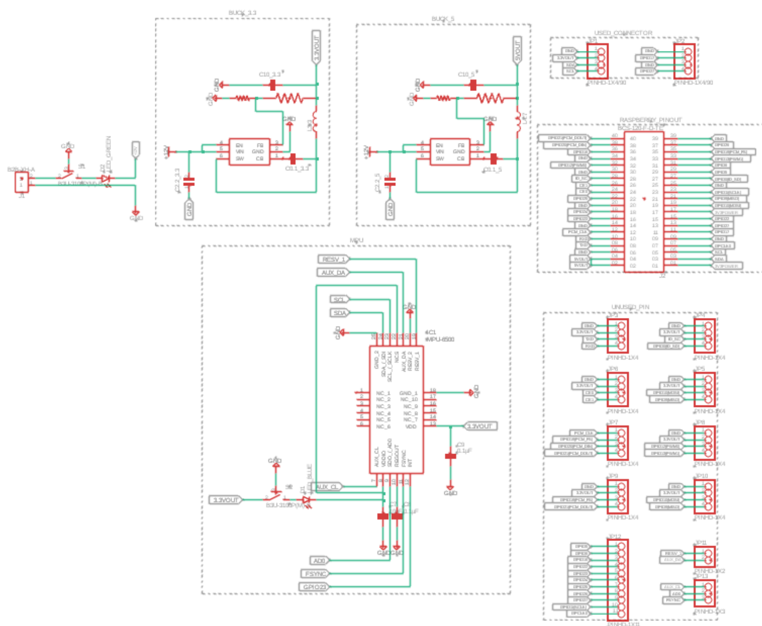


Figure 19: Schematics

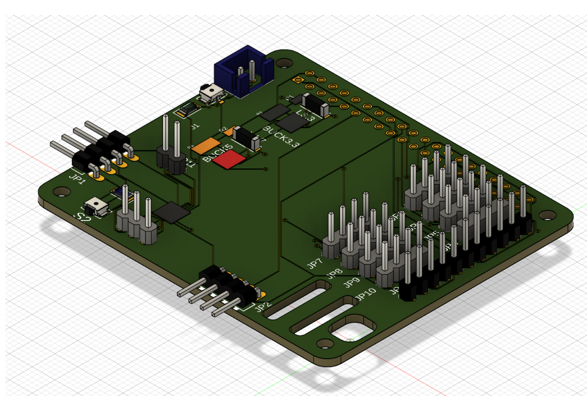
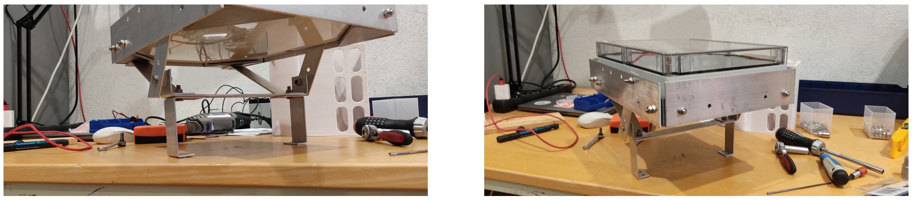


Figure 20: 3D PCB Design

For the rest of the electronics we stick to the first prototype changing only the caps as described above.

8.4 Construction and first test

Finally after all that work of conception it was the time to build the first prototype. I started with the realisation of the Sonar structure by manufacturing our own aluminium strip with the help of the FUSE lab (see Figure 21).



(a) Up Forward view (b) Upside down view

Figure 21: Sonar and box structure of the Sea Dragon 3.0.

Our pieces for the realisation of the floaters are to the limit of the printer that's why before printing the real one I made a try with ABS, ABS is close to PETG in term of print setting so it can give me a good feeling of what to modify or improve. The result was pretty successful unless some holes to make bigger and some straight angle to round to have better result.

I've done this print test on the second of August and the entire Sea Dragon should be ready for test at the end of my internship. We hope to make some tests before I leave. Unfortunately the delay on the deliveries we are waiting don't let us a lot of time. However I've seen some result with the aluminum structure that are satisfying.

9 Conclusion and open

I completed this 2nd year internship as a project assistant in the Department of Maritime and Vehicle Safety at Chalmers University in Gothenburg. During those 4 months, I had the opportunity to apply some of the skills I learned during my year at Seatech and to enhance those skills by adding a professional dimension to them.

Moreover, I had the chance to learn many new skills, such as PCB design and the science of 3D printing. It was also my first time sharing my opinions in an international context and switching all my work to English. I felt a great improvement throughout those months. I also enjoyed participating in projects with teammates who were gratefully driven by a concrete goal.

The main project of this internship was the Sea Dragon, and I am pleased to see the work I started continue. I made efforts to document my work thoroughly, providing future project assistants with the opportunity to build upon it and address any remaining issues that I, unfortunately, did not have time to solve.

Finally, I would like to highlight my first real experience in research. By the end of this internship, I am motivated to pursue a PhD in the years following my diploma.

10 References

[1] Chalmers Website https://www.chalmers.se/

[2] C2B2 Website https://c2b2.se/

[3] Stop Drowning Now Website https://www.stopdrowningnow.org/

[4] Sweden Wikipedia https://en.wikipedia.org/wiki/Sweden

[5] Johansson, J. E. "SORA: Search Optimization Rescue Assistance during Drowning Incidents". Master's thesis. Umeå University. (2021). https://umu.diva-portal.org/smash/get/diva2:1577369/FULLTEXT01.pdf

[6] T200 technical detail website page https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/

[7] MK2 technical detail Data Sheet https://www.teledynemarine.com/en-us/products/SiteAssets/BlueView/PLD20590-3%20Blueview%20M900-Mk2%20product%20leaflet.pdf

[8] Ansys explanation of Von Mises stress https://www.ansys.com/blog/what-is-equivalent-stress

[9] Johnson, Virgil E, and Thomas Rasnick. "The Drag Coefficient of Parabolic Bodies of Revolution Operating at Zero Cavitation Number and Zero Angle of Yaw". (Feb. 1961). https://ntrs.nasa.gov/api/citations/19980223945/downloads/19980223945.pdf

[10] Course on ballistic physics https://sagars24.wixsite.com/tam335-mp1/drag#:~:text=The%20thin%20disk%20has%20a,by%20up%20to%20115%25!

[11] Register Map of the IMU 6050 from Invensense https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf

[12] Data sheet of the IMU 6050 from Invensense https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

[13] Mehmet Kopar and Ali Riza Yildiz. "Experimental investigation of mechanical properties of PLA, ABS, and PETG 3-d printing materials using fused deposition modeling technique". In: Materials Testing 65.12 (Dec. 2023), pp. 1795–1804. https://doi.org/10.1515/mt-2023-0202

[14] Sheonandan Prajapati et al. "A review on comparison of physical and mechanical properties of PLA, ABS, TPU, and PETG manufac- tured engineering components by using fused deposition modelling". In: Materials Today: Proceedings (May 2024). https://doi.org/10.1016/j.matpr.2024.05.018.

[15] Wikipedia drag Force https://en.wikipedia.org/wiki/Drag_(physics)#:~:text=In%20fluid%20dynamics%2C%20drag%2C%20sometimes,a%20fluid%20and%20solid%20surface.

[16] Opendlv github repository `https://github.com/chalmers-revere/opendlv`

[17] Opendlv website `https://opendlv.org/index.html`

[18] Philip Masek et al. "Systematic evaluation of sandboxed software deployment for real-time software on the example of a self-driving heavy vehicle". In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (Nov. 2016), pp. 2398– 2403. `https://doi.org/10.1109/ITSC.2016.7795942`.

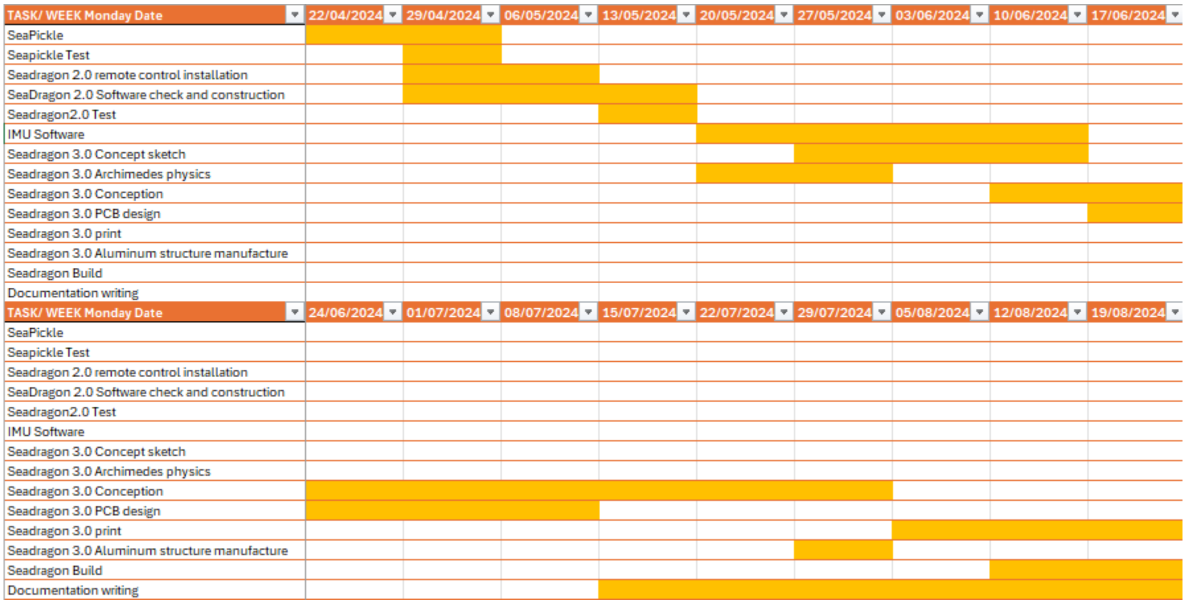


Figure 22: Timeline works

11 Annexes

11.1 Gant

11.2 IMU code

```

#include <cmath>
#include <cstdint>
#include <iostream>

#include "cluon-complete.hpp"
#include "opendlv-message-standard.hpp"


#include <algorithm>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <stdint.h>
#include <stdexcept>
#include <string>
#include <tuple>

// all the adress comes from the MPU6050 register map

```

```
#define MPU6050_I2C_ADDR 0x68


// define the accel adress
#define REG_ACCEL_XOUT_H 0x3B
#define REG_ACCEL_XOUT_L 0x3C
#define REG_ACCEL_YOUT_H 0x3D
#define REG_ACCEL_YOUT_L 0x3E
#define REG_ACCEL_ZOUT_H 0x3F
#define REG_ACCEL_ZOUT_L 0x40

// define the gyro adress
#define REG_GYRO_XOUT_H 0x43
#define REG_GYRO_XOUT_L 0X44
#define REG_GYRO_YOUT_H 0x45
#define REG_GYRO_YOUT_L 0x46
#define REG_GYRO_ZOUT_H 0x47
#define REG_GYRO_ZOUT_L 0x48


#define REG_PWR_MGMT_1 0x6B
#define REG_ACCEL_CONFIG 0x1C
#define REG_GYRO_CONFIG 0x1B

int file = -1;
```

Listing 1: IMU code INITIALISATION

```

// Function for calculating median
float Median(vector<int> v) {
  int32_t n=vector.size();
    // Sort the vector
    sort(v.begin(), v.end());

    // Check if the number of elements is odd
    if (n % 2 != 0)
        return (double)v[n / 2];

    // If the number of elements is even, return the average
    // of the two middle elements
    return (double)(v[(n - 1) / 2] + v[n / 2]) / 2.0;
}

// Please note, this is not the recommanded way to write data
// to i2c devices from user space.
void i2c_write(__u8 reg_address, __u8 val) {
  char buf[2];
  if(file < 0) {
    printf("Error, i2c bus is not available\n");
    exit(1);
  }

  buf[0] = reg_address;
  buf[1] = val;

  if (write(file, buf, 2) != 2) {
```

```
    printf("Error, unable to write to i2c device\n");
    exit(1);
  }

}

// Please note, this is not thre recommanded way to read data
// from i2c devices from user space.
char i2c_read(uint8_t reg_address) {
  char buf[1];
  if(file < 0) {
    printf("Error, i2c bus is not available\n");
    exit(1);
  }

  buf[0] = reg_address;

  if (write(file, buf, 1) != 1) {
    printf("Error, unable to write to i2c device\n");
    exit(1);
  }


  if (read(file, buf, 1) != 1) {
    printf("Error, unable to read from i2c device\n");
    exit(1);
  }

  return buf[0];

}

uint16_t merge_bytes( uint8_t LSB, uint8_t MSB) {
  return  (uint16_t) ((( LSB & 0xFF) << 8) | MSB);
}


// 16 bits data on the MPU6050 are in two registers,
// encoded in two complement. So we convert those to int16_t
int16_t two_complement_to_int( uint8_t LSB, uint8_t MSB) {
  int16_t signed_int = 0;
  uint16_t word;

  word = merge_bytes(LSB, MSB);

  if((word & 0x8000) == 0x8000) { // negative number
    signed_int = (int16_t) -(~word);
  } else {
    signed_int = (int16_t) (word & 0x7fff);
  }

  return signed_int;
}
```

Listing 2: IMU code FUNCTION

```

```

```

// RawAcceldata function
std::tuple<float, float, float> GetRawAccelData ( int i2c_bus =1){
  char bus_filename[250];
  char accel_z_h , accel_z_l, accel_x_h, accel_x_l, accel_y_h, accel_y_l;
  int16_t x_accel = 0;
  int16_t y_accel = 0;
  int16_t z_accel = 0;
  float x_accel_g, y_accel_g,z_accel_g;


  // Reach the i2c port
  snprintf(bus_filename, 250,"/dev/i2c-%i", i2c_bus);
  file = open(bus_filename, O_RDWR);
  if (file < 0) {
    /* ERROR HANDLING; you can check errno to see what went wrong */
    exit(1);
  }
  if (ioctl(file, I2C_SLAVE, MPU6050_I2C_ADDR) < 0) {
    /* ERROR HANDLING; you can check errno to see what went wrong */
    exit(1);
  }

  // wake up the IMU
  i2c_write(REG_PWR_MGMT_1, 0x01);
  // config the acceleration register to scale range of 2g
  i2c_write(REG_ACCEL_CONFIG, 0x00);

  // Read the x accel data
  accel_x_h = i2c_read(REG_ACCEL_XOUT_H);
  accel_x_l = i2c_read(REG_ACCEL_XOUT_L);
  // Read the y accel data
  accel_y_h = i2c_read(REG_ACCEL_YOUT_H);
  accel_y_l = i2c_read(REG_ACCEL_YOUT_L);
  // Read the z accel data
  accel_z_h = i2c_read(REG_ACCEL_ZOUT_H);
  accel_z_l = i2c_read(REG_ACCEL_ZOUT_L);

  // convert the accel data
  x_accel= two_complement_to_int(accel_x_h,accel_x_l);
  x_accel_g = ((float) x_accel)/16384;

  y_accel= two_complement_to_int(accel_y_h,accel_y_l);
  y_accel_g = ((float) y_accel)/16384;

  z_accel= two_complement_to_int(accel_z_h,accel_z_l);
  z_accel_g = ((float) z_accel)/16384;

  return {x_accel_g,y_accel_g,z_accel_g};
}

//RawGyroData function
std::tuple<float, float, float> GetRawGyroData ( int i2c_bus =1){
  char bus_filename[250];
  char gyro_z_h , gyro_z_l, gyro_x_h, gyro_x_l, gyro_y_h, gyro_y_l;
  int16_t x_gyro = 0;
```

```
  int16_t y_gyro = 0;
  int16_t z_gyro = 0;
  float x_gyro_deg, y_gyro_deg,z_gyro_deg;
  // Reach the i2c port
  snprintf(bus_filename, 250,"/dev/i2c-%i", i2c_bus);
  file = open(bus_filename, O_RDWR);
  if (file < 0) {
    /* ERROR HANDLING; you can check errno to see what went wrong */
    exit(1);
  }
  if (ioctl(file, I2C_SLAVE, MPU6050_I2C_ADDR) < 0) {
    /* ERROR HANDLING; you can check errno to see what went wrong */
    exit(1);
  }

  // wake up the IMU
  i2c_write(REG_PWR_MGMT_1, 0x01);
  // config the gyro register to scale range of 250deg
  i2c_write(REG_GYRO_CONFIG, 0x00);

  // Read the x gyro data
  gyro_x_h = i2c_read(REG_GYRO_XOUT_H);
  gyro_x_l = i2c_read(REG_GYRO_XOUT_L);
  // Read y gyro data
  gyro_y_h = i2c_read(REG_GYRO_YOUT_H);
  gyro_y_l = i2c_read(REG_GYRO_YOUT_L);
  // Read z gyro data
  gyro_z_h = i2c_read(REG_GYRO_ZOUT_H);
  gyro_z_l = i2c_read(REG_GYRO_ZOUT_L);

  // Convert the gyro data
  x_gyro= two_complement_to_int(gyro_x_h,gyro_x_l);
  x_gyro_deg = ((float) x_gyro)/131;
  y_gyro= two_complement_to_int(gyro_y_h,gyro_y_l);
  y_gyro_deg = ((float) y_gyro)/131;
  z_gyro= two_complement_to_int(gyro_z_h,gyro_z_l);
  z_gyro_deg = ((float) z_gyro)/131;

  return {x_gyro_deg,y_gyro_deg,z_gyro_deg};
}
```

Listing 3: IMU code DATA COLLECTION

```

// Accelerometer Calibration function
std::tuple<float, float, float, bool, bool, bool> AccelCalibration(){
  float x_accel_cal=50, y_accel_cal=50, z_accel_cal=50;
  int32_t Check=0;
  int32_t cpt=0;
  bool x_check=false, y_check=false, z_check=false;
  std::vector<float> x_sample, y_sample,z_sample;
  std::cout << " Starting Accelerometer calibration" << std::endl;
  std::cout << " Starting x calibration" << std::endl;
  //Start with the x value
  if ( abs(get<0>(GetRawAccelData())-x_accel_cal)<-0.1 || abs(get<0>(
    GetRawAccelData())-x_accel_cal)> 0.1){
    // Get a sample of 2500 rawdata
```

```
    while (Check < 2500){
      Check +=1;
      x_sample.push_back(get<0>GetRawAccelData());
    }
    // Treat the sample to find offset value, find the median
    x_accel_cal=Median(x_sample);
    x_check=true;
  }
  else {
    std::cout << "x calibration fail" << std::endl;
  }
  // then the y value
  if ( abs(get<1>(GetRawAccelData())-y_accel_cal)<-0.1 || abs(get<1>(
    GetRawAccelData())-y_accel_cal)> 0.1){
    // Get a sample of 2500 rawdata
    while (Check < 2500){
      Check +=1;
      y_sample.push_back(get<1>GetRawAccelData());
    }
    // Treat the sample to find offset value, find the median
    y_accel_cal=Median(y_sample);
    y_check=true;
  }
  else {
    std::cout << "y calibration fail" << std::endl;
  }
  // Finally the z value
  if ( abs(get<2>(GetRawAccelData())-z_accel_cal)<-0.1 || abs(get<2>(
    GetRawAccelData())-z_accel_cal)> 0.1){
    // Get a sample of 2500 rawdata
    while (Check < 2500){
      Check +=1;
      z_sample.push_back(get<2>GetRawAccelData());
    }
    // Treat the sample to find offset value, find the median
    z_accel_cal=Median(z_sample);
    z_check=true;
  }
  else {
    std::cout << "z calibration fail" << std::endl;
  }

  return{x_accel_cal,y_accel_cal,z_accel_cal, x_check, y_check, z_check};
}

// Gyrometre Calibration function
std::tuple<float, float, float, bool, bool, bool> GyroCalibration(){
  float x_gyro_cal=50, y_gyro_cal=50, z_gyro_cal=50;
  int32_t Check=0;
  int32_t cpt=0;
  bool x_check=false, y_check=false, z_check=false;
  std::vector<float> x_sample, y_sample,z_sample;
  std::cout << " Starting gyroerometer calibration" << std::endl;
  std::cout << " Starting x calibration" << std::endl;
  //Start with the x value
  if ( abs(get<0>(GetRawGyroData())-x_gyro_cal)<-0.1 || abs(get<0>(
```

```
    GetRawGyroData())-x_gyro_cal)> 0.1){
    // Get a sample of 2500 rawdata
    while (Check < 2500){
      Check +=1;
      x_sample.push_back(get<0>GetRawgyroData());
    }
    // Treat the sample to find offset value, find the median
    x_gyro_cal=Median(x_sample);
    x_check=true;
  }
  else {
    std::cout << "x calibration fail" << std::endl;
  }
  // then the y value
  if ( abs(get<1>(GetRawGyroData())-y_gyro_cal)<-0.1 || abs(get<1>(
    GetRawGyroData())-y_gyro_cal)> 0.1){
    // Get a sample of 2500 rawdata
    while (Check < 2500){
      Check +=1;
      y_sample.push_back(get<1>GetRawGyroData());
    }
    // Treat the sample to find offset value, find the median
    y_gyro_cal=Median(y_sample);
    y_check=true;
  }
  else {
    std::cout << "y calibration fail" << std::endl;
  }
  // Finally the z value
  if ( abs(get<2>(GetRawgyroData())-z_gyro_cal)<-0.1 || abs(get<2>(
    GetRawGyroData())-z_gyro_cal)> 0.1){
    // Get a sample of 2500 rawdata
    while (Check < 2500){
      Check +=1;
      z_sample.push_back(get<2>GetRawGyroData());
    }
    // Treat the sample to find offset value, find the median
    z_gyro_cal=Median(z_sample);
    z_check=true;
  }
  else {
    std::cout << "z calibration fail" << std::endl;
  }

  return{x_gyro_cal,y_gyro_cal,z_gyro_cal, x_check, y_check, z_check};
}
```

Listing 4: IMU code CALIBRATION

```

// Calcul of the Accel data with the previous calibration
std::tuple<float,float,float>GetAccelData(std::tuple<float, float, float,bool,
    bool,bool> AccelCal){
  float x_accel,y_accel,z_accel;
  const auto AccelData=GetRawAccelData();
  // check if the calibration is higher than the raw data or not for x value
  if (get<3>(AccelCal)){
```

```
    x_accel=get<0>(AccelData)+get<0>(AccelCal);
  }
  else {
    x_accel=get<0>(AccelData)-get<0>(AccelCal);
  }
  // check if the calibration is higher than the raw data or not for y value
  if (get<4>(AccelCal)){
    y_accel=get<1>(AccelData)+get<1>(AccelCal);
  }
  else {
    y_accel=get<1>(AccelData)-get<1>(AccelCal);
  }
  // check if the calibration is higher than the raw data or not for z value
  if (get<5>(AccelCal)){
    z_accel=get<2>(AccelData)+get<2>(AccelCal);
  }
  else {
    z_accel=get<2>(AccelData)-get<2>(AccelCal);
  }
  return{x_accel, y_accel, z_accel};
}

// Calcul of the Gyro data with the previous calibration
std::tuple<float,float,float>GetGyroData(std::tuple<float, float, float,bool,
    bool,bool> GyroCal){
  float x_gyro,y_gyro,z_gyro;
  const auto GyroData=GetRawGyroData();
  // check if the calibration is higher than the raw data or not for x value
  if (get<3>(GyroCal)){
    x_gyro=get<0>(GyroData)+get<0>(GyroCal);
  }
  else {
    x_gyro=get<0>(GyroData)-get<0>(GyroCal);
  }
  // check if the calibration is higher than the raw data or not for y value
  if (get<4>(GyroCal)){
    y_gyro=get<1>(GyroData)+get<1>(GyroCal);
  }
  else {
    y_gyro=get<1>(GyroData)-get<1>(GyroCal);
  }
  // check if the calibration is higher than the raw data or not for y value
  if (get<5>(GyroCal)){
    z_gyro=get<2>(GyroData)+get<2>(GyroCal);
  }
  else {
    z_gyro=get<2>(GyroData)-get<2>(GyroCal);
  }
  return{x_gyro, y_gyro, z_gyro};
}
```

Listing 5: IMU code DATA ACTUALISATION

```

int32_t main(int32_t argc, char **argv)
{
  auto cmd = cluon::getCommandlineArguments(argc, argv);
```

```
if (!cmd.contains("cid")) {
  std::cout << argv[0] << " is an OpenDLV microservice." << std::endl;
  std::cout << "Usage: " << argv[0] << " "
            << "--cid=<conference id; e.g. 111> "
            << "[--verbose] "
      << "[--calibration]" << std::endl;
  return 0;
}

uint16_t const cid = std::stoi(cmd.at("cid"));
bool const verbose = (cmd.count("verbose") != 0);
bool const calibration = (cmd.count("calibration") != 0);
std::tuple<float, float,float,bool,bool,bool> AccelOffset;
std::tuple<float, float,float,bool,bool,bool> GyroOffset;


if (verbose) {
  std::cout << "Starting microservice." << std::endl;
}


if (calibration){
std::cout <<"starting calibration." << std::endl;
 AccelOffset=AccelCalibration();
 GyroOffset=GyroCalibration();
std::cout <<"end of calibration. Accel Offset :"
 << std::get<0>(AccelOffset)<< ","
   << std::get<1>(AccelOffset)<< ","
    << std::get<2>(AccelOffset)<< "| \t"
    << " Gyro Offset : "
    << std::get<0>(GyroOffset)<< ","
    << std::get<1>(GyroOffset)<< ","
    << std::get<2>(GyroOffset)<< "| \n";
 }


cluon::OD4Session od4(cid);

opendlv::proxy::AccelerationReading ar;
opendlv::proxy::AngularVelocityReading avr;

while (od4.isRunning()) {

  auto const AccelData=GetAccelData(AccelOffset);
auto const GyroData=GetGyroData(GyroOffset);

  //Send Messages
  auto time_ = cluon::time::now();
  ar.accelerationX(std::get<0>(AccelData)).accelerationY(std::get<1>(
 AccelData)).accelerationZ(std::get<2>(AccelData));
  avr.angularVelocityX(std::get<0>(GyroData)).angularVelocityY(std::get<1>(
 GyroData)).angularVelocityZ(std::get<2>(GyroData));
  od4.send(ar, time_);
  od4.send(avr, time_);

//Print Messages
```

```
    if (verbose) std::cout <<" | accel : "
      << std::get<0>(AccelData)<< ","
      << std::get<1>(AccelData)<< ","
      << std::get<2>(AccelData)<< "| \t"
      << " gyro : "
      << std::get<0>(GyroData)<< ","
      << std::get<1>(GyroData)<< ","
      << std::get<2>(GyroData)<< "| \n";
    std::this_thread::sleep_for(std::chrono::milliseconds(50));
  }

  if (verbose) {
    std::cout << "Closing microservice." << std::endl;
  }

  return 0;
}
```

Listing 6: IMU code MAIN

11.3 Structure Drawings

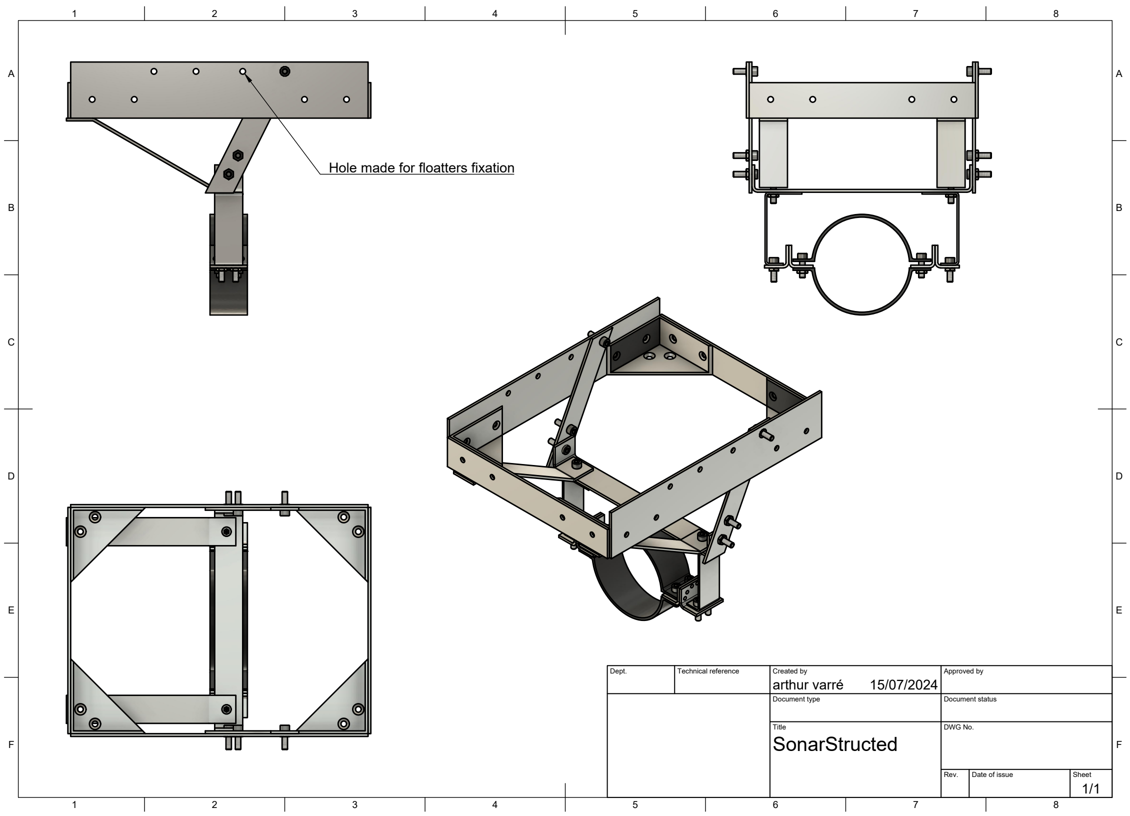


Figure 23: Sonar and Box Structure

Fiche d’évaluation étudiant

Ce document est à insérer dans le rapport. Le rapport est à fournir pour le 19 aout dernier délai.

Organisme d’accueil :
Chalmers university
Sujet du stage : Work with Autonomous system especially maritime vehicle, and Software development

Evaluation qualitative du stage	I	S	B	E	N	Commentaires
Qualité de l’entreprise						Le lieu de stage était une université ; toutefois, les laboratoires dans lesquels j’ai travaillé peuvent accueillir des ingénieurs aussi. D’autres étudiants de Seatech ont été à Chalmers mais mon département ne connaissait pas Seatech.
L’entreprise offre-t-elle un contexte propice à une carrière d’ingénieur ?			X			
L’entreprise connaissait elle l’école (accueil de stagiaire, embauche, relations autres…) ?		X				
L’entreprise a-t-elle mis à votre dispositions les moyens nécessaires pour réaliser votre mission (documents, éléments d’information, matériels) ?				X		
Qualité de la missions						
Vos missions étaient-elles en rapport avec votre formation ?				X		
Les missions effectuées étaient-elles bien celles définies au départ ?				X		
Qualité de l’encadrement						
Votre tuteur organisme d’accueil a-t-il pris le temps de vous présenter le fonctionnement de la structure et l’équipe ?				X		
Votre tuteur organisme d’accueil vous a-t-il aidé et conseillé quand cela était nécessaire ?				X		
Votre enseignant référent vous a-t-il aidé et conseillé lorsque cela était nécessaire ?				X		

Explication des cotations	
I	(I) Insuffisant
S	(S) Suffisant
B	(B) Bien
E	(E) Excellent
N	(N) Non applicable

NB : Dans le but d’alléger la lecture du document, le genre masculin est utilisé sans discrimination pour le genre masculin et féminin.

Fiche d’évaluation étudiant

Autoévaluation : développement des compétences et trajectoire professionnelle En prenant un peu de recul sur votre activité durant le stage pensez-vous avoir travaillé / développé certaines des compétences du référentiel de la formation Seatech ? Lesquelles ? Pourquoi et comment ? D’autres compétences ?		
Compétence	**Critère**	**Commentaire**
Concevoir des solutions d’ingénierie	Analyser le besoin	Analyser les besoins était au centre de mon stage. En effet, la conception d’un USV requiert de savoir en globalité les exigences nécessaires à son fonctionnement.
	Répondre au besoin	Répondre au besoin était la suite logique. Lors de ce stage, j’ai pu développer cette compétence en conceptualisant et en construisant cet USV.
	Utiliser les outils appropriés	Ce point est plus relatif. Je dirais que lorsque j’avais accès aux outils appropriés, je les ai utilisés du mieux possible. Cependant, il est arrivé que les outils les plus appropriés me soient inaccessibles.
	Documenter ses choix et ses sources	C’est un point que j’ai particulièrement travaillé lors de ce stage car j’ai pu rédiger plusieurs documents de documentation qui nécessitaient de citer de façon précise d’autres publications dont j’utilise les résultats.
Mettre en œuvre des solutions	Analyser et améliorer une solution existante	Ayant travaillé sur un projet déjà existant initié par d’autres stagiaires ou assistants de projet, mon rôle était justement de poursuivre ces améliorations.
	Proposer une solution nouvelle	Mon amélioration de ce projet consistait en la conception d’une nouvelle structure et l’ajout de nouvelles fonctionnalités. Je pense donc avoir proposé une solution nouvelle.
	Utiliser les principes de l’amélioration continue	J’ai découvert cette notion au cours de ce stage. En effet, prendre en main un projet déjà existant avec l'intention de le transmettre après avoir contribué à son évolution me semble être une démarche en accord avec l’amélioration continue.
	Rédiger un document scientifique et technique d’appui	Sachant que mon travail sera repris par d’autres personnes il était nécessaire pour moi de réaliser un document scientifique d’appuis, c’était par ailleurs la première fois pour moi de réaliser cet exercice.
Développer une démarche R&D	Réaliser une veille technologique / un état de l’art	Lors de la rédaction d’une étude sur les propriétés des pièces imprimées en 3D, j’ai pu réaliser un état de l’art sur les données actuelles. L’exercice était d’autant plus intéressant qu’il concernait une étude scientifique (jamais réalisée auparavant pour moi).
	Formuler des hypothèses	RAS
	Proposer une démarche expérimentale, un protocole ou un modèle	Les démarches expérimentales théoriques n’étaient pas au centre de ce stage. Toutefois, la réalisation de l’étude sur les propriétés des pièces imprimées en 3D dans un environnement marin m’a permis d’en apprendre plus sur ce sujet.
	Adopter une démarche d’innovation	Je pense avoir adopté une démarche d’innovation dans la conception et la réalisation de cet USV.

Piloter des projets d'ingénierie	S'insérer dans ou conduire une ou plusieurs étapes d'un projet	Je pense que ma participation dans la conception d'un USV dans le projet général d'une plateforme autonome de détection m'a permis de m'insérer dans une équipe et dans un projet.
	Prendre en compte la gestion globale des organisations ou les règles de fonctionnement, économiques ou juridiques	RAS
	Utiliser les outils de gestion de projet et outils collaboratifs	RAS
	Identifier ou mobiliser les ressources appropriées	RAS
Encadrer une équipe	S'insérer et collaborer	Je pense m'être inséré dans l'équipe de recherche du département dans lequel j'ai fait mon stage et j'ai pleinement collaboré avec différents chercheurs (Ola Benderius, Krister Blanch).
	Assurer une responsabilité d'animation	RAS
	Travailler en équipe pluridisciplinaire et/ou internationale	Le travail d'équipe dans un contexte internationale était une constante lors de ce stage, en effet étant en suède nous communiquions en anglais principalement de plus la majorité de mon travail s'est réaliser en équipe.
	Communiquer (écrit et oral) de manière adaptée	Communiquer pour échanger mes idée mais aussi pour informer de mes prises de décision sur le projet était quelque choses de déjà familier mais que j'ai pu encore améliorer. J'ai surtout développer ma communication a l'écris pour la transmission de mon travail.
Agir en professionnel responsable	Prendre en compte les enjeux éthiques et sociétaux (RSE, DD, RGPD, …)	RAS
	Assumer la responsabilité de ses actes et décision	J'ai eu la chance d'avoir plusieurs responsabilités durant ce stage. J'ai dû notamment prendre à plusieurs reprises des décisions, notamment sur les matériaux à utiliser ou encore sur la démarche à suivre lors de la conception d'un USV.
	Porter un regard critique sur le sens de l'activité conduite	J'ai pu développer mieux cette compétence en m'interrogeant a chaque prise de décision sur le sens de celle-ci, je me suis notamment beaucoup documenter pour le choix de matériaux
	Être dans une dynamique d'apprentissage	Lors des projets mené pendant ce stage j'ai eu l'occasion d'apprendre de nouvel compétence notamment la réalisation de PCB je pense donc avoi été dans une dynamique d'apprentissage.

Fiche d'évaluation étudiant

Trajectoire professionnelle

A la suite de votre stage, avez-vous confirmé ou affiné votre projet professionnel d'être ingénieur (métier plus précis, secteur, contexte ou type d'entreprise, …) ? Si oui, quelles actions pensez-vous devoir entreprendre pour y arriver (renforcer certaines connaissances, développer certaines compétences, lesquelles) ?

À la suite de ce stage, je pense avoir développé un attrait supplémentaire pour la recherche, mais je souhaite tout de même m'orienter assez jeune vers le management. J'ai cependant pu déceler certaines lacunes en programmation réseau, que j'aimerais combler à l'avenir.

Date et signature de l'étudiant :
16/08/2024

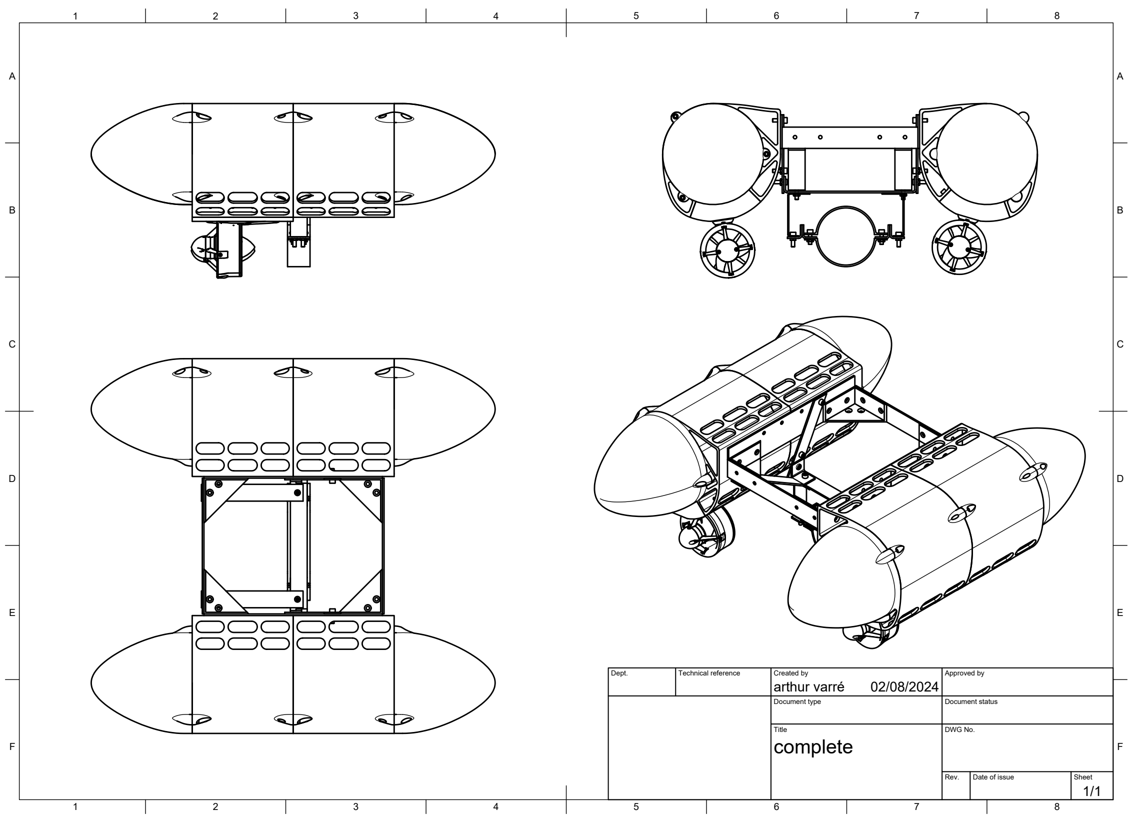


Figure 24: Sonar and Box Structure