S.O.N.I.A. AUV Technical Design Report

Roxanne Parent Thibeault, Team Leader
Olivier Juteau-Desjardins, Admin Team Leader
Caroline Boudreau, Admin Team Leader
Ewan Féminis, Software Team Leader
Zachary Proulx Electrical Team Leader
Félix Bolduc, Mechanical Team Leader
Clement Imfura, Software Team
Ethel Larios Garcia, Software Team
François Côté-Raiche, Software Team
Naomi Catwell, Software Team
Nimai Jariwala, Software Team
Raphaël Largeau, Software Team
Baptiste Jourdes, Electrical Team
Corentin Couëron, Electrical Team
Félix Charbonneau, Electrical Team
Jérémie Cachot, Electrical Team
Nathan André, Electrical Team
Alexis Rocheleau, Mechanical Team
Benito Alonso, Mechanical Team
Étienne Dutemple, Mechanical Team
Robin PetitJean, Mechanical Team
Romain Guedon, Mechanical Team
Thibault Esbert, Mechanical Team
Thierry Goyer, Mechanical Team

***Abstract*—S.O.N.I.A. is a robotics club dedicated to constructing underwater vehicles since 1999. Comprised of 24 student members, the club aims to improve the existing platforms in terms of efficiency, precision and overall robustness. An improved power distribution unit, an external battery system and new floaters have been developed for the latest prototype in addition to a redesign of its central hull. The software stack has also received important improvements to the vision and mission in addition to a major update to its framework. The improvements made to the team's submarine were made based on competition, design and testing strategies that will be presented in this paper.**

I. Introduction

S.O.N.I.A. (Système d'Opération Nautique Intelligent et Autonome) is a robotic club from the École de Technologie Supérieure, an engineering school in Montreal, Canada. For the past 25 years, the club has been building submarine vehicles. This year, the club will bring two autonomous underwater vehicles (AUVs) to the RoboSub competition: AUV7 and AUV8 (see Appendix G and H). The club had two goals this year: improve the reliability of AUV8 and enhance the skills and knowledge of new members. The latter goal is part of a larger multiyear plan for renewing the team composition, which was undermined by the Covid-19 pandemic. As such, last year was spent rebuilding the team, and this year was focused on improving the overall knowledge of the team, with the final objective being the creation of a new prototype within the next two years. To that end, the team chose to build on the knowledge of previous team members by improving the reliability of the last submarine (built between 2019 and 2021). These improvements, along with other existing systems, will be leveraged to support the team's competition strategy, which will be detailed in this paper. The design strategy used by the team will also be discussed, with a particular focus on the improvements made this year. Finally, the testing strategy employed to validate the team's design will be presented.

II. Competition Strategy

The overall competition strategy of the team has been to focus on tasks that present minimal dependencies between the submarine's systems. The rationale is that this would provide more flexibility during the competition. As such, the team has chosen to forgo the "collect samples" and "random pinger" tasks as they would represent a significant time investment while also presenting a consequential level of cross-system dependency. Moreover, the team has chosen to prioritize certain tasks over others based on risk-reward assessments. Four main criteria were considered for these assessments: the readiness of existing systems, the required time and financial investment, the implementation complexity, and the scoring associated with a task. The tasks chosen by the team are presented in the following subsections in order of priority. It should be noted that the team plans to use two submarines during the competition. The considerations related to this choice are presented in the last subsection.

A. *Coin flips, pass the gate and style points tasks*

These tasks have been prioritized by the team as they are considered low risk. Leveraging the existing 6 degrees of freedom[1], [2], [3], the team feels confident in their ability to complete the tasks. The main risks identified for these tasks are the adaptation of the controller to ROS 2 and the modifications to the physical model used by the controller to reflect the changes made to the hull of AUV8. Given that the controller is also required for the other tasks, this has been considered an acceptable risk. Side detection while passing the gate has been given a low priority, as other detection tasks take precedence.

B. Hydrothermal Vent Task

The simple shape and high contrast of the buoy should make it a relatively easy target to detect. As such, it is considered an excellent starting point for the new and improved AI vision system and has been prioritized accordingly. Given the high score associated with circumnavigation compared to touching the buoy, the former has been chosen. Moreover, a low FOV lens has been selected for the front-facing camera which, while facilitating the detection of distant objects, may make alignment with close objects harder. Therefore, circumnavigation is considered slightly safer by the team.

C. Mapping Task

The mapping task will be attempted this year. While this task presents significant challenges, the team believes that the scoring associated with it justifies the time investment. The team plans to navigate to the task without using hydrophones, instead leveraging the low FOV lens of the front-facing camera to detect the task's board from the buoy. The main risks identified with this task are the reliability of the torpedo and the alignment systems. The former has been addressed by developing a new trigger mechanism that uses a servo motor instead of a solenoid. While this results in a more complex system, the team believes that the consistency of the servo motor compared to the solenoid should make it better. The risk associated with the alignment system should be mitigated by performance improvements made to the vision system. Indeed, the previous system suffered from significant slowdown, causing the submarines to overshoot during alignment. This overshooting, coupled with the low FOV of the front-facing camera, sometimes resulted in target loss. This year's submarine uses a new version of the YOLO AI model which implements significant performance improvements.

D. Ocean temperature task

The Ocean temperature task is considered low priority mainly due to the low scoring associated with it, even if the risk associated with it is considered only moderate. As such, the task will only be attempted if the team is confident that other tasks are working sufficiently well. Past experiences at RoboSub have shown that a reliable, less ambitious strategy is preferable. The main risk identified with this task is the moderate difficulty and time required to find the bins using the bottom-facing cameras.

E. Dual Submarines Strategy and Consideration

As previously mentioned, the team is planning to use two submarines (AUV7 and AUV8) during the competition. Two reasons explain this choice: First, it gives the team the ability to attempt the inter-vehicle communication task (which is associated with a significant number of points). Second, the two submarines act as backup of each other. This point is crucial to highlight, as it enabled the team to compete in the last edition of RoboSub despite AUV8 experiencing significant failures. The main downside of using two submarines is that it divides the team's attention between them. To mitigate this issue, the two submarines will be doing the same task, which means that one mission can be developed for both submarines. However, this approach does increase the risk of collision. To address this, the submarines will rely on the inter-vehicle communication system to synchronize their actions. Moreover, a fail-safe mechanism will be employed in case of communication loss, where the submarine will abort its mission if the task cannot be completed within a certain timeframe.

III. Design Strategy

A. Mechanical

This year, the mechanical design work has been focused on reducing the risk of water leaks. The team has analyzed the conditions that led to two leaks on AUV8 and concluded that, while both times user error was at play, two other factors played a significant role. First, every change of batteries involved removing and replacing two external caps on the submarine, which increased the risk of a misplaced cap. Second, the pre-water inspection only involved a visual confirmation which was found to be risky and potentially insufficient. The former has been addressed by the addition of an external battery system as detailed in section III.A.1) and III.A.2); the latter has been addressed by addition of a vacuum test in the pre-water procedure (see test procedure in Appendix C). The addition of external batteries also prompted the design of new floaters. Additionally, a new torpedo system has been developed.

1) External batteries and new center piece

The external batteries required the design of the battery enclosures and a redesign of the submarine central piece where the batteries will meet with the submarine. To avoid dangling wires and the risks associated, bulkhead connectors were chosen for the high current connection. Additionally, the connectors have been partially embedded in their supporting structures to bring the batteries closer to the center of the submarine. By doing so, the moment of inertia of the submarine is minimized (which was an issue on previous prototypes). The minimal wall thickness of those structures has been determined using finite element analysis with a safety factor of 1.4, allowing operation up to 10m.

2) Fixation system for the external battries

One of the considerations for the fixation system was to reduce the forces exerted to the bulkhead connectors. Therefore, the batteries are positioned by two square extrusions on the side plates and three complementary ones machined directly into the battery cases which prevent vertical translation and pitch and roll rotations (pink in Fig. 1). A quick-released T-handle with locking pins prevents movement in the remaining axes (orange in Fig. 1) resulting in a hyperstatic assembly. In addition, it allows for a toolless quick swap of batteries. According to the team's estimates, this quick disconnect system enables battery swaps in under one minute.

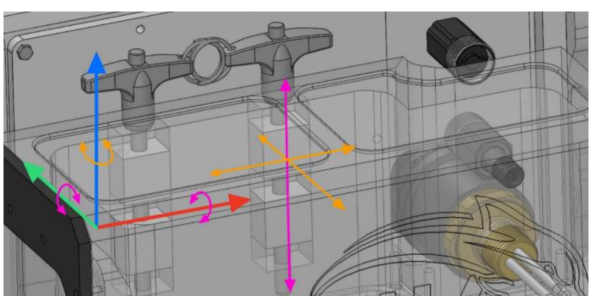


Fig. 1. Fixation system of the external batteries

3) New floaters

The change to the submarine volume and weight due to the addition of external batteries prompted the development of new floaters for AUV8. The side floaters are made from molded carbon fiber. This gives them two major upsides. First, the precision of 3D printed molds allows the floaters to be designed more hydrodynamically. Second, carbon fiber is more resistant to wear and tear than foam and tape (as was the case before). However, carbon fiber has one significant weakness: fragility. The risk being that a cracked floater could lead to a significant loss of buoyancy. To alleviate this scenario, safety foam has been partially added to the inside of the floaters. This way, marginal buoyancy can still be guaranteed in the event of a crack in a floater.

4) Torpedo system

The former torpedo system was triggered by a solenoid. While light and small, it exhibited significant non-linearity on applied force depending on the position of the metal core. This non-linearity, coupled with variability in the trigger placement resulted in failed launches.

B. Electrical Systems

1) Electrical Arhictecture

Fig. 2 present the shared architecture of both submarines. Both use a Nvidia Xavier as their main on-board computer. The Xavier receives positional data for an IMU, DVL and depth sensor and sends motor commands to the power management unit (PMU). The PMU is explored more thoroughly in another section (III.B.3). These systems form the navigational system of the submarine. The Xavier also has access to two cameras (front and bottom facing) for vision related tasks, an acoustic modem for inter-vehicle synchronization, a kill-switch for safely immobilizing the submarine and an actuator board to control the torpedoes and droppers.

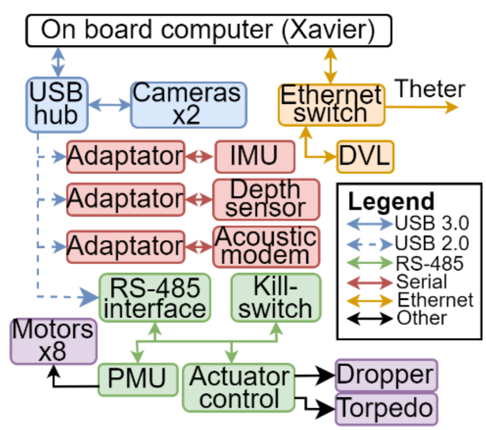


Fig. 2. Simplified submarine architecture

2) Communcation network

The submarine sub-systems communicate using a heterogeneous network (see Fig. 2). Each network was chosen based on their ease of use and on the subsystems needs. Due to their high-bandwidth requirements, the cameras are using a USB 3.0 link. On the other hand, most of the off-the-shelf components use serial connections, which were adapted to a USB 2.0 link. Similarly, all the custom PCB boards use RS-485 links for its ease of implementation, extensibility and robustness. For the DVL, an ethernet connection has been chosen as it allows easy external access (using the tether) for debugging.

3) Power Management Unit (PMU)

The PMU is responsible for both motor control and power distribution. Those two functions have been integrated together to optimize space usage and reduce ohmic loss due to the high current consumption of the motors. The main vision this year for the PMU was to improve its reparability. The team has chosen to move away from the previous monolithic design to a modular architecture with the idea that modules could be swapped instead of repaired. This new design is based on a main board which interconnect the LiPo batteries, the motors, the relays, the safety module (which controls the activation of the motors using relays), the MCU module (which generate the PWM signal to control the motor speed based on the on board computer command and monitors the power-consumption), and the motors modules (which contains the hardware to control the motor speed based on the PWM and integrates overcurrent protection for the motors). This architecture and the final design are presented in Fig. 3. Each module uses board-to-board connectors which provide power and a communication interface while also serving as the mechanical attach point. This way, a faulty module can be swapped in a matter of seconds. This should greatly improve the reparability of the PMU. Overall, the current consumption of the motor was a major consideration for both the choice of connectors as well as the layout of the main board and motor modules.

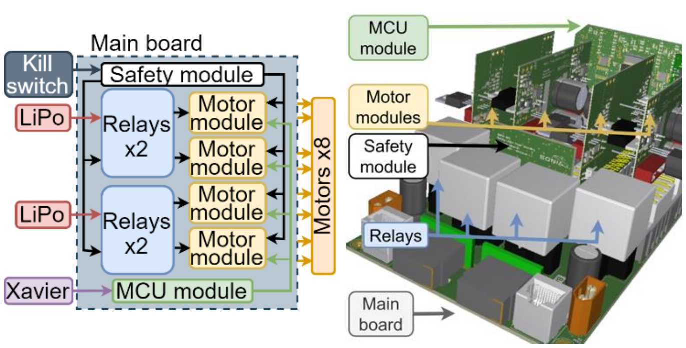


Fig. 3. Modular architecture of the PMU (left) and final desing (right)

C. Software

1) Overview

The submarine software stack's foundation is the ROS framework which provides the basic structure of the software subsystems. The software stack includes the low-level driver that serves as a bridge between the hardware of the submarine and the software system. The stack also includes a vision system for the detection of the different task object. The submarine's overall behavior logic is implemented in a mission system which gives it its autonomous abilities. Based on the mission system's decision, a planner generates the submarine's movement path. The position of the submarine is controlled based on the command from the planner using a model-based controller.

A major design consideration is the uniformity of the software architecture between the submarines. This greatly improves the speed at which new features can be developed for both submarines.

The software team has made significant changes preparing for this year's competition. The submarine will be equipped with the same components but will use ROS Humble Hawksbill (ROS2)[4] instead of ROS Noetic Ninjemys (ROS1)[5] and a different approach to vision algorithms as well as a new mission system. The objective for these changes is to make futures changes easier and improve the submarine effectiveness at completing missions.

2) ROS 2

The software team's biggest project for this year was updating the onboard computers communication system from ROS1 to ROS2. The main reason for this change is to remove the dependency of ROS1, as it's support is ending in May 2025.

In the original implementation of the communication network, the software team had decided to use Docker to isolate the environmental dependencies for each ROS1 node. With how ROS2 nodes now behave, it is no longer needed to isolate the nodes thus removing the need for Docker. This is one of the benefits of a decentralized system that uses peer-to-peer discovery. Other benefits include better performance and greater scalability. Overall, the migration to ROS2 provides a solid foundation for the software system while ensuring the continuation of the software platform.

3) Mission system

The new mission system for this year's competition will be built using BehaviorTree[6]. This is a stark change from the previous year's mission system which was built using FlexBe. One of the driving reasons for this change is that BehaviorTree is compiled and not interpreted, preventing potential errors at runtime. In the past, this issue considerably slowed down the development of the missions as the team had to deal with runtime errors.

One major change is that FlexBe is a Finite State Machine while the new BehaviorTree uses behavior tree. This required a redesign in the flow of logic when it came to how missions interacted with the rest of the system, leading to the recreation of all existing missions. Fortunately, behavior trees allow the use of nested trees (nested missions), allowing the reuse of smaller, more generalized mission blocks.

While this change adds to the complexity of the system, it has been found to be an acceptable comprise given the added reliability brought usage of a compiled language. Overall, the team is confident that this new system will reduce the chance of a mission failure during the critical moments of the competition.

4) Vision algorithms

Given the nature of the Robosub competition, many tasks require the usage of machine vision. The teams therefore rely heavily on the usage of vison systems. Traditionally, the team used a hybrid approach that incorporated both conventional vision and AI. However, during the previous edition of RoboSub, the team was able to obtain great results using only an AI system. Consequently, the effort this year was concentrated exclusively on the improvement of the existing deep learning vision system. With this approach, the team can dedicate more time to the single vision system, resulting in a more precise system overall. The team incorporated the capability to use different models and neural networks based on camera orientation (forward facing or bottom facing) to optimize results. Indeed, since both cameras operate on separate images, using two distinct models allows for more specialized results. The current system can use either YOLOv8n[7] or YOLOv10n[8] as the base convolutional neural network. The main design consideration behind this choice is the ubiquity and the performance of these models. The team chose to use both models since it is not clear which one offers the most performance. Once the team is more familiar with this new system, a single model will be chosen.

Additionally, major improvements have been made to the process used for the configuration of the training and data augmentation. The focus has been to improve the transparency of the system with the goal of providing a better user experience. Those improvements have been brought by the implementation of a new transparent interface. With this system, users can fine-tune the models in an efficient manner leading to the ability to rapidly explore various configurations and improved AI models.

IV. Testing Strategy

A. Electrical Testing Strategy

1) Power Management Unit (PMU)

The PMU testing strategy is based on two distinct phases. The first phase tests the PMU independently and is mainly used to validate the functionality of the PMU including motor speed control, motor emergency shutoff and software motor activation control. In this test, the PMU is first validated without any motor connected, the goal being to verify the basic functionality of the systems without introducing additional systems. This progressive approach helps the isolation of problems should they arise. At this stage, the circuit of the system has not been tested before. Consequently, a variable power supply with overcurrent protection is used to avoid damaging the PCB. A motor mounted on a pole is then connected to this system. This system then emerged in water, which provides a real-world test environment for the motor control. Notably, this setup is used to stress test the system by running the motors at full power for a long period of time. The test plan for this first test phase is presented in Appendix D.

A second test phase is where the PMU integration with the rest of system is validated. The approach for this phase is to test the PMU in the actual submarine environment. The strategy is therefore performing submarine movements in a pool to validate the PMU. First, simple linear moves are performed in various direction. The actual movement linearity is observed to confirm that the PMU is working fine. Once the basic integration of the PMU is established, a more thorough test is performed by executing a “trick shot” (where the submarine rotates around all its axis) as this the most taxing movement that the submarine can performed. The test plan for this phase is presented in Appendix D.

B. Mechanical Testing Strategy

1) Watertightness verification

For the watertightness of the new central piece and the external batteries, two tests have been created. The two tests are generic and can be reused for other projects. The first test consists of depressurizing the enclosures using a hand operated pump such that the enclosures experience a pressure equivalent a 10m water column (equivalent to 140% of the maximum expected operating depth of the submarine). The complete test plan for this verification is presented in Appendix E. This test is not perfect however, since water molecules are smaller than air molecules. The gold standard in leak tests is helium testing which wouldn’t be feasible in the case of the club. Consequently, a second validation test has been created which works by submerging the enclosure in a pool (see Appendix E).

C. Software Testing Strategy

1) ROS 2

To test the ROS2 nodes, they are run on both computers and the AUV’s on-board Xavier. Trying the node on a computer first allows for testing the compilation and checking for a correct startup. The AUV components can then be connected to the computer to verify the interaction with the hardware. When those are tests are completed, the node is then tested on both AUV’s Xavier to check the interaction with the other nodes and the hardware. Finally, when all other verifications are completed, the whole ROS2 network is tested in a pool to ensure it works in real world conditions also side the missions.

2) Mission

Since the missions with BehaviorTree are compiled, this checks if the code is valid before running it. This first verification confirms that there are no typos or invalid commands in the code. After the compilation is completed, the mission is tested in our simulation to check if the AUV reacts as expected in the competition environment. This allows for easy and fast verification without needing access to water. Finally, the mission is tested in a real pool with the obstacles to make sure everything works as intended. Those tests confirm that the mission is ready for the competition.

3) Vision system

To train the team’s models, datasets are used. These datasets are composed of images and labels containing the position of the objects in each image. The datasets are composed of images of the obstacles that were filmed by the team. The same amount of images containing each obstacle in different contexts are then selected and labeled manually. To ensure the quality of the neural networks, the datasets are separated into 3 sets (train, validation and test). This way, the overfitting phenomenon can be avoided by comparing training and validation losses. To compare the models, their performance is measured, and the accuracy of the test is set. The most accurate model that minimizes false positive for a minimum threshold of performance (10 fps) is then selected to be used. The model is then tested in real conditions during pool testing.

V. Acknowledgements

We wanted to thank the student club Cedille for hosting and maintaining our server.

We wish to express our gratitude to Prof. Pierre Bourque, Prof. Christian Belleau, Prof. Simon Joncas, Prof. Marc Paquet and Prof. Jean-Philippe Roberge of the École de Technologie Supérieure for overseeing the projects associated with our club. We also wanted to thank our sponsors for all their support:

Diamond: Altium, SolidWorks and Cégep du Vieux Montréal

Platinium: Drillmex, Fond de developpement ÉTS and Uzinakod

Gold: Labelbox, VectorNav, and Teledyne Marine

Silver: Digi-Key, Hakko, MacArtney, Impact Subsea and Parc Jean-Drapeau.

Bronze: Anodisation Québec, Blue Robotics, General Dynamics, Gosselin Photo and IVI.

REFERENCES

[1] A. Lamarre and A. Desgagné, “Identification d’un sous-marin autonome représenté à l’aide du quaternion et asservissement avec un contrôleur MPC.” Apr. 22, 2021. [Online]. Available: https://t.ly/4GB7v

[2] A. Lamarre, L. Rayeh, and G. Schena, “Conception d’une loi de commande linéaire et non linéaire pour un sous- marin autonome.” Dec. 06, 2021. [Online]. Available: https://t.ly/lN2fF

[3] *Mathworks Miniseries | Session 3: Test & Deploy an Adaptive Model Predictive Controller | RoboNation*, (Mar. 06, 2023). Accessed: Jul. 01, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=0meVOuq9y5w

[4] “ROS 2 Documentation — ROS 2 Documentation: Humble documentation.” Accessed: Jul. 01, 2024. [Online]. Available: https://docs.ros.org/en/humble/index.html

[5] “noetic - ROS Wiki.” Accessed: Jul. 01, 2024. [Online]. Available: https://wiki.ros.org/noetic

[6] “BehaviorTree.CPP.” Accessed: Jul. 01, 2024. [Online]. Available: https://www.behaviortree.dev/

[7] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO.” Jan. 2023. Accessed: Jul. 01, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[8] A. Wang *et al.*, “YOLOv10: Real-Time End-to-End Object Detection.” arXiv, May 23, 2024. doi: 10.48550/arXiv.2405.14458.

APPENDIX A
COMPONENT SPECIFICATIONS (AUV7)

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of purchase
Buoyancy Control	-	Added mass	Brass plates	Custom	150$	2022
Frame	-	CNC aluminium system	6061-T6 CNC machined, anodized and painted	Custom	-	2020
Waterproof Housing	Homemade	CNC aluminium system	6061-T6 CNC machined, anodized and painted	Custom	7000$	2020
Waterproof Connectors	MacArtney	Subconn connector	Wet Mate	Purchased	4000$	2020
Thrusters	Blue Robotics	T200 (x8)	0.02 kg f	Purchased	-	2016
Motor Control	HobbyKing	AFRO ESC	30A	Purchased	-	2016
High Level Control	-	LTV MPC	-	Custom	-	2022
Actuators	-	Droppers only	Solenoid Activation	Custom	-	2022
Battery	MaxAmps	4S 16000mAh	14.8V	Purchased	-	2022
Converter	Texas Instruments	LM25116MH	Synchronous Buck Controller	Purchased	-	2020
Regulator	-	Powersupplies	12V 15A Step-down	Custom	-	2020
CPU	Nvidia	Jetson AGX Xavier	16GB RAM	Purchased	-	2019
Internal Comm Network	-	RS485	2 twisted pairs Ethernet cables	Custom	-	2022
External Comm Network	ConnectTech	XDG021	1000 Mbps Switch	Purchased	-	2016
Compass	VectorNav	VN-100 Rugged AHRS	Standard calibration +25°C	Purchased	-	2022
Inertial Measurement Unit (IMU)	VectorNav	VN-100 Rugged AHRS	Standard calibration +25°C	Purchased	-	2022
Doppler Velocity Log (DVL)	Nortek	DVL500	300m	Purchased	-	2016
Vision	Flir	Chameleon 3 USB	55FPS, 3.2MP	Purchased	-	2022
Acoustics	Bruel & Kjaer	8103	0.1 to 180kHz	Purchased	-	2016
Inter-vehicule communication	Water Linked AS	MODEM M64	64 bits, omnidirectional	Purchased	2000$	2021
Algorithms: Vision	OpenCV TensorFlow2	- MobileNet	- -	Custom Custom	- -	2022 2022
Algorithms: Localization and Mapping	Matlab Octomap	Extend Kalman Filter 3D occupancy grid	TBD -	- -	- -	2021 2022
Algorithms: Autonomy	FlexBe	Finite-state-machine	-	Custom	-	2021
Open-source software	OpenCv, Behaviour Tree, AirFlow, TensorFlow, ROS, Unity Robotics, Docker, WikiJS, Github					
Team Size	24					
Expertise ratio	3/24					
Testing time: simulation	15 hours					
Testing time: in-water	5 hours					
Programming Languages	C/C++, C#, Python, Matlab					

APPENDIX B
COMPONENT SPECIFICATIONS (AUV8)

Component	Vendor	Model/Type	Specs	Custom/ Purchased	Cost	Year of purchase
Buoyancy Control	-	Added Volume	Carbon fiber	Custom	500$	2024
Frame	-	CNC aluminium system	6061-T6 CNC machined, anodized and painted	Custom	-	2020
Waterproof Housing	Homemade	CNC aluminium system	6061-T6 CNC machined, anodized and painted	Custom	7000$	2020
Waterproof Connectors	MacArtney	Subconn connector	Wet Mate	Purchased	4000$	2020
Thrusters	Blue Robotics	T200 (x8)	0.02 kg f	Purchased	200$/each	2019
Motor Control	Emax	Bullet Series ESC (x8)	30A	Purchased	15$/each	2019
High Level Control	-	LTV MPC	-	Custom	-	2022
Actuators	-	Droppers and torpedo	Servo motors	Custom	-	2024
Battery	MaxAmps	4S 16000mAh	14.8V	Purchased	2000$	2020
CPU	Nvidia	Jetson AGX Xavier	32GB RAM	Purchased	1000$	2020
Internal Comm Network	-	RS485	2 twisted pairs Ethernet cables	Custom	-	2020
External Comm Network	ConnectTech	XDG021	1000 Mbps Switch	Purchased	-	2020
Compass	VectorNav	VN-100 Rugged AHRS	Standard calibration +25°C	Purchased	-	2022
Inertial Measurement Unit (IMU)	VectorNav	VN-100 Rugged AHRS	Standard calibration +25°C	Purchased	-	2022
Doppler Velocity Log (DVL)	Teledyne	Pathfinder	600kHz,140m	Purchased	20000$	2020
Vision	Flir	Chameleon 3 USB	55FPS, 3.2MP	Purchased	1200$	2020
Acoustics	Bruel & Kjaer	8103	0.1 to 180kHz	Purchased	-	2020
Inter-vehicule communication	Water Linked AS	MODEM M64	64 bits, omnidirectional	Purchased	2000$	2021
Algorithms: Vision	OpenCV TensorFlow2	- MobileNet	- -	Custom Custom	- -	2022 2022
Algorithms: Acoustics	-	Time Differental of Arrival (TDOA)	FPGA Implementation	Custom	-	2022
Algorithms: Localization and Mapping	Matlab Octomap	Extend Kalman Filter 3D occupancy grid	TBD -	- -	- -	2021 2022
Algorithms: Autonomy	FlexBe	Finite-state-machine	-	Custom	-	2021
Open-source software	OpenCv, Behaviour Tree, AirFlow, TensorFlow, ROS, Unity Robotics, Docker, WikiJS, Github					
Team Size	24					
Expertise ratio	3/24					
Testing time: simulation	65 hours					
Testing time: in-water	15 hours					
Programming Languages	C/C++, C#, Python, Matlab					

APPENDIX C
AUV8 PRE-WATER TEST PLAN

PRE-WATER TEST AUV8

Pre-water test – AUV8

ID: TEST-GEN-AUV8-001-V1-0
Rev: V1-0 Applicable submarines: AUV8

Abstract

This document presents the pre-water test procedure for AUV8. The main goal of this procedure is to avoid water leak in the submarine. The other goal is to verify the core submarine functionalities prior to the insertion of AUV8 in the water.

This test procedure shall be followed before putting AUV8 in the water in all instances where AUV8 was previously opened.

Revision history

Version	Date	Changes
V1-0	2024-06-29	• Initial document creation

Prerequisite

Hardware

1x AUV8
1x Portable network box (*duckbox*)
2x Batteries battery pack

Tools

1x Mityvac Hand Operated Vacuum Pump (from bluerobotics)
1x Pressure Relief Valve Backfill Adapter (from bluerobotics)
1x Vacuum Plug (from bluerobotics)
1x ¼'' Vacuum hose (from bluerobotics)
1x latop

Others

1x pool
3x team members
1x diver

Electrical Procedure

1. Using a multi meter in continuity mode, verify continuity between pin 1 & 4 (refer to Figure 1) of both battery connectors on the submarine hull. Verify the following criteria **(interrupt procedure in case of failure)** :
   a. **CRIT-1: No continuity between pin 1 & 4 on left connector**

   b. **CRIT-2: No continuity between pin 1 & 4 on right connector**

Figure 1

2. Using a multi meter in continuity mode, verify continuity between pin 1 (refer to Figure 1) of both battery connectors on the submarine hull. Verify the following criteria **(interrupt procedure in case of failure)** :
   a. **CRIT-3: No continuity between pin 1 of left connector and pin 1 of right connector**
3. Using a multi meter in continuity mode, verify continuity between pin 4 (refer to Figure 1) of both battery connectors on the submarine hull. Verify the following criteria **(interrupt procedure in case of failure)** :
   a. **CRIT-3: Continuity between pin 4 of left connector and pin 4 of right connector**
4. Connect both battery pack. Verify the following criteria **(interrupt procedure in case of failure)** :
   a. **CRIT-4: Battery level indicators of the PMU fully lights up**

Software procedure

5. Connect the laptop to the *duckbox* wifi network
6. Connect AUV8 and the *duckbox* using the tether.
7. After a ~ 1 minute, start a ping test with the address 192.161.0.31 on the laptop. Verify the following criteria **(interrupt procedure in case of failure)** :
   a. **CRIT-5: Responses to the ping request are present**
8. Start a SSH session with the Xavier. Start the submarine system using the script `launch_auv8.sh`
9. Launch the telemetry module. Verify the following criteria **(interrupt procedure in case of failure)** :
   a. **CRIT-6: Both batteries indicator on the telemetry are > 80%**
10. Pull the kill-switch. Verify the following criteria **(interrupt procedure in case of failure)** :

PRE-WATER TEST AUV8

a. **CRIT-7: Telemetry kill indicator in telemetry shows the kill as pulled**
b. **CRIT-8: LED labeled KILL on the PMU is off**

11. Push the kill-switch. Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-9: Telemetry kill indicator in telemetry shows the kill as pushed**
    b. **CRIT-10: LED labeled KILL on the PMU is on**
12. In the telemetry, click "activate all motors". Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-11: all motors start making a 3 notes sound (i.e. ESC startup sound)**
    b. **CRIT-12: the motor activation LED on the PMU lights up**
13. In the telemetry, set the submarine state to "dry mode". In the motor widget, click on "start dry test". Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-13: Each motor start rotating, one after the other, following the order defined in Figure 2**

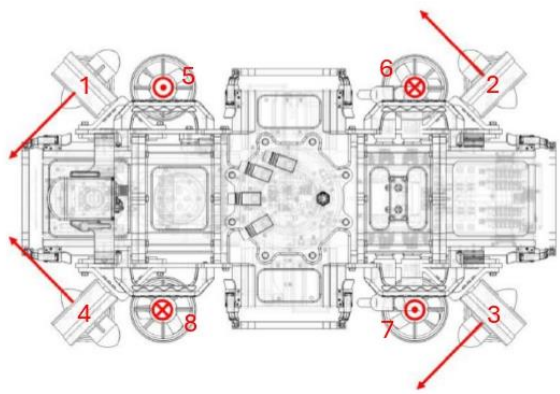


Figure 2: Motor numbering on AUV8 (front of AUV8 facing the left)

14. In the telemetry, set the submarine state back to "normal".
15. In the telemetry, open the camera viewer. Ask a team member to pass its hands on front of both cameras. Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-14: Both images are present, movement can been seen on both images**

PRE-WATER TEST AUV8

Mechanical procedure (caps)

16. Verify all of the acrylic caps on both side the submarine. Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-15: The o-ring of the front AND the rear cap is visibly compress along the entire permitter of the cap**
    b. **CRIT-16: All 4 clips of both front AND rear cap are close**
    c. **CRIT-17: All 4 clips of both front AND rear cap are locked (try to physically open all clips)**
17. The verification of the previous step should be redone by another team member
    a. **CRIT-18: The inspection of the caps shall be done by at least by two team members.**

Mechanical procedure (vacuum test)

The following test procedure is based on the bluerobotics vacuum procedures [1], [2].

18. Attach clear vacuum hose to the vacuum pump
19. Insert the barbed end of the Vacuum Plug into the vacuum hose.
20. Insert the rubber stopper into the hole on the Vacuum Plug to seal it up
21. Turn the knob on the side of the vacuum pump so it is in the "VACUUM" setting
22. Pump the hand pump to approximately 15 inHg (47.4 kPa), and monitor for about 30 seconds to a minute. Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-19: the pressure does not decrease by more than 0.5 inHg (1.7 kPa)**
23. Turn the knob on the side of the vacuum pump so it is in the "VACUUM" setting
24. De-pressured
25. The Pressure Relief Valve (PRV) plunger has threads at the top to accept the threads on the Backfill adapter. Screw on the adapter clockwise into the PRV. The O-ring on the Backfill adapter will seat into the PRV plug.
26. As you screw on the Backfill Adapter the plunger is pulled up, creating a sealed path into the enclosure. Continue turning the Backfill Adapter into the PRV plug all the way until it stops and cannot be turned further.
27. Insert the Vacuum Plug into the Backfill Adapter. At this point the Vacuum Plug and the Backfill Adapter can be used with a backfilling system to backfill the enclosure with an inert gas such as nitrogen or to pull a vacuum inside the enclosure.
28. Pump the hand pump, removing air and decreasing the pressure inside the enclosure to approximately 15 inHg (47.4 kPa).

PRE-WATER TEST AUV8

29. Monitor the gauge for about 5 minutes. Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-19: the pressure does not decrease by more than 0.5 inHg (1.7 kPa)**
30. De-pressured
31. Unscrew the Backfill Adapter counter-clockwise to remove it. As the Backfill Adapter is unscrewed the plunger is released and reseals the PRV.

References

[1] "Pressure Relief Valve Installation and Usage," Blue Robotics. Accessed: Jun. 30, 2024. [Online]. Available: https://bluerobotics.com/learn/pressure-relief-valve-installation-and-usage/

[2] "Using the Vacuum Plug and Hand Pump," Blue Robotics. Accessed: Jun. 30, 2024. [Online]. Available: https://bluerobotics.com/learn/using-the-vacuum-test-plug/

APPENDIX D
PMU BENCHTEST PLAN

PMU TESTING PROCEDURE (BENCH TEST) V1-0

PMU testing procedure (bench test)

ID: TEST-ELE-PMU-001-V1-0
Rev: V1-0 Applicable submarines: AUV8

Abstract

This document presents a test procedure to the test the PMU individually

Revision history

Version	Date	Changes
V1-0	2024-06-29	• Initial document creation

Prerequisite

Hardware

1x PMU main board
8x PMU motor module
1x PMU MCU module
1x Safety module
2x banana to XT-96 wires
1x USB-C to USB-A cable
2x Ethernet cable (for RS-485)

Tools

1x Dual Variable PSU with 10 amps capability (or 2x Variable PSU with 10 amps capability)
1x Motor-pole assembly
1x Rs-485 interface

Tools (continued)

1x Kill switch simulation board
1xLaptop
1x Multimeter
1x 12V PSU with barrel connector

Others

1x Water container (at least 1m deap)
1x Local with water access, water drain and power outlet (e.g. D-1034)
1x folding table
1x electrical tape
2x team members
1xESD protection bracelet

Procedure

1. Clean the water container (if necessary)
2. Fill up the water container and install close to the water drain
3. Install the folding table ~1.5m away from the water container
4. Ground yourself with an ESD protection bracelet
5. Assemble the PMU modules with the PMU main board
6. **Make sure that PSU in NOT connected yet to PMU**. Set the voltage of **BOTH** channel to 16.6V, set the max current of **BOTH** channel

TEST-ELE-PMU-001-V1-0 1

PMU TESTING PROCEDURE (BENCH TEST) V1-0

7. Power the ON the PSU, verify that **BOTH** channels are at 16.6±0.2V, adjust if necessary
8. **Power OFF the PSU**
9. Follow the connection from Figure 1

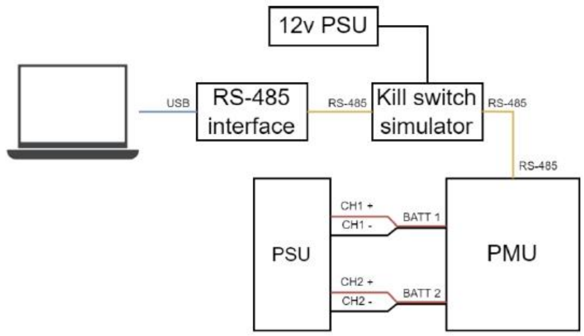


Figure 1: Connection diagram

10. Secure all wires using tape
11. Enter `lsusb` in the terminal of the laptop, verify that the RS-485 interface is recognized as FTDI device
12. Power ON the PSU. Verify the following criteria **(interrupt test in case of failure)**:
    a. **CRIT-1: PSU CH1 is in C.V. mode (constant voltage mode)**
    b. **CRIT-2: PSU CH2 is in C.V. mode (constant voltage mode)**
13. Activate the kill switch simulator. Verify the following criteria **(interrupt test in case of failure)**:
    a. **CRIT-3: PSU CH1 is in C.V. mode (constant voltage mode)**
    b. **CRIT-4: PSU CH2 is in C.V. mode (constant voltage mode)**
    c. **CRIT-5: The relays make an audible click sound**
    d. **CIRT-6: LED labeled "KILL" on MCU module lights up**
    e. **CIRT-7: Both battery level indicator on the MCU module fully light up**
    f. **CRIT-8: LED labeled "KILL" on all 4 Motor modules lights up**
14. Deactivate the kill switch simulator. Power OFF the PSU.
15. Connect the motor-pole assembly to the M1 Connector
    a. **Note**: One team member should operate the motor-pole assembly, the other team member should do the remaining to reduce the splashing the electronics. The table with the electronic setup should be ~1.5m away from the motor-pole assembly operator.
16. Power ON the PSU
17. On the laptop, launch the script "`launch_local.sh`", enter the command "`rqt`", the local telemetry should now be opened

PMU TESTING PROCEDURE (BENCH TEST) V1-0

18. Activate the kill switch simulator.
19. In the telemetry, switch the run mode to "drytest". Activate all motors. Verify the following criteria **(interrupt test in case of failure):**
    a. **CRIT-9-M1: PSU CH1 is in C.V. mode (constant voltage mode)**
    b. **CRIT-10-M1: PSU CH2 is in C.V. mode (constant voltage mode)**
    c. **CRIT-11-M1 : The motor make a 3 notes sound (i.e. ESC startup)**
20. Make sure that the motor-pole assembly is submerged in the water container and that he motor-pole assembly is securely hold by the operator
21. Set motor 1 to 20% power forward. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-12-M1: Motor is rotating in a clockwise direction as view from the back of the motor**
22. Set motor 1 to 20% power backward. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-13-M1: Motor is rotating in a counterclockwise direction as view from the back of the motor**
23. Set motor 1 to 100% power backward. Hold the motor to this power level for 10 minutes. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-14-M1: Motor is rotating at steady rate for the entire duration of the test**
24. Still with power level at 100% deactivate the kill-switch. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-15-M1: Motor immediately stop rotating**
25. Reactivate the kill switch. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-16-M1: Motor stays stopped**
26. Set motor 1 to 0%, deactivate motor 1, reactivate motor 1, to motor 1 to 10% forward. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-17-M1: Motor rotate**
27. Deactivate motor 1. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-17-M1: Motor immediately stop rotating**
28. Deactivate the kill switch simulator. Power OFF the PSU.
29. Disconnect the motor
30. Re do steps 15 to 29 with the motors 2 to 8. For the motor connector, use connector MX, where X is the motor number. The criteria identification, use MX, where X is the motor number (e.g.: **CRIT-17-M3** for motor 3)

TEST-ELE-PMU-001-V1-0 3

APPENDIX D
PMU INTEGRATION TEST PLAN

PMU TESTING PROCEDURE (INTEGRATION) V1

PMU testing procedure (integration)

ID: TEST-ELE-PMU-002-V1-0
Rev: V1-0 Applicable submarines: AUV8

Abstract

This document presents a test procedure for the integration of the PMU with AUV8

Revision history

Version	Date	Changes
V1-0	2024-06-30	• Initial document creation

Prerequisite

Hardware	Others
1xAUV8 with PMU installed 1x Portable network box (*duckbox*) 2x Batteries **Tools** 1x Laptop 1x Tether	1x Pool 3x team members 1x Diver

Procedure

1. Follow the pre-water procedure for AUV8, as defined in TEST-GEN-AUV8-001-V1-0 (Pre-water test – AUV8)
2. Verify that the diver is ready to receive the submarine in the water
3. Ask two team members to take the submarine and introduced it the water with the help of the diver. Verify the following criteria **(continue test in case of failure):**
   a. **CRIT-1: Submarine is stable and flat relative to the pool water level**
4. Ask the diver to put the submarine in the middle of the pool
5. Start the submarine control
6. In the telemetry set a way point 2m in front, 1m below surface
7. Execute the waypoint mission in 3s. Verify the following criteria **(continue test in case of failure):**
   a. **CRIT-2: The submarine moves in a straight line to the programmed waypoint**

PMU TESTING PROCEDURE (INTEGRATION) V1

8. Execute the previous step with way points to the left, right and back of the submarine
9. Stops the submarine control.
10. Ask the diver to bring the submarine back to the center of the pool.
11. Execute the “trick shot” mission. Verify the following criteria **(continue test in case of failure):**
    a. **CRIT-3: The submarine shall rotate around the 3 axis**

APPENDIX E
ENCLOSURE GENERAL AIR VACUUM TESTING PROCEDURE

ENCLOSURE GENRAL AIR VACCUM TESTING PROCEDURE V1-0

Enclosure general air vacuum testing procedure

ID: TEST-MEC-GEN-001-V1-0
Rev: V1-0 Applicable submarines: N/A

Abstract

This document presents a generalized testing methodology to validate the watertightness of custom mechanical enclosure using air vacuum.

Revision history

Version	Date	Changes
V1-0	2024-06-30	• Initial document creation

Prerequisite

Hardware / Others	Tools
Hardware 1x Enclosure to test ?x Sealing hardware (Gasket/O-ring) of the enclosure to test **Others** 1x Molykote 111 silicone grease	**Tools** 1x Mityvac Hand Operated Vacuum Pump (from bluerobotics) 1x Pressure Relief Valve (from bluerobotics) 1x Pressure Relief Valve Backfill Adapter (from bluerobotics) 1x Vacuum Plug (from bluerobotics) 1x ¼'' Vacuum hose (from bluerobotics) ?x Appropriately sized torque wrenches

Procedure

The following test procedure is based on the bluerobotics vacuum procedures [1], [2].

1. Inspect all Gaskets and O-rings. Verify the following criteria **(abort test in case of failure):**
   a. **CRIT-1: all Gasket and O-ring shall show have no visual damage such as cracks**
   b. **CRIT-2: all Gasket and O-ring shall be exempt from any contaminants such as dust or sand**
2. Spread Molykote silicone grease over the surface of the Gaskets and O-rings
3. Carefully place gaskets and O-ring on the mating faces

ENCLOSURE GENRAL AIR VACCUM TESTING PROCEDURE V1-0

4. Close the enclosure using the appropriate torque spec for the screws, as defined in the specific enclosure documentation.
5. Install the pressure relief valve with greased O-ring on the enclosure. Use a torque of 3.5Nm.
6. Attach clear vacuum hose to the vacuum pump
7. Insert the barbed end of the Vacuum Plug into the vacuum hose.
8. Insert the rubber stopper into the hole on the Vacuum Plug to seal it up
9. Turn the knob on the side of the vacuum pump so it is in the "VACUUM" setting
10. Pump the hand pump to approximately 15 inHg (47.4 kPa), and monitor for about 30 seconds to a minute. Verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-3: the pressure does not decrease by more than 0.5 inHg (1.7 kPa)**
11. De-pressured
12. The Pressure Relief Valve (PRV) plunger has threads at the top to accept the threads on the Backfill adapter. Screw on the adapter clockwise into the PRV. The O-ring on the Backfill adapter will seat into the PRV plug.
13. As you screw on the Backfill Adapter the plunger is pulled up, creating a sealed path into the enclosure. Continue turning the Backfill Adapter into the PRV plug all the way until it stops and cannot be turned further.
14. Insert the Vacuum Plug into the Backfill Adapter. At this point the Vacuum Plug and the Backfill Adapter can be used with a backfilling system to backfill the enclosure with an inert gas such as nitrogen or to pull a vacuum inside the enclosure.
15. Pump the hand pump, removing air and decreasing the pressure inside the enclosure to approximately 29 inHg (98 kPa) corresponding to a water depth of about 10m.
16. Let the enclosure sit in a depressurized state for 12 hours
17. After the 12 hours, verify the following criteria **(interrupt procedure in case of failure)** :
    a. **CRIT-4: the pressure does not decrease by more than 0.5 inHg (1.7 kPa)**
18. De-pressured and Unscrew the Backfill Adapter counter-clockwise to remove it. As the Backfill Adapter is unscrewed the plunger is released and reseals the PRV.

References

[1] "Pressure Relief Valve Installation and Usage," Blue Robotics. Accessed: Jun. 30, 2024. [Online]. Available: https://bluerobotics.com/learn/pressure-relief-valve-installation-and-usage/
[2] "Using the Vacuum Plug and Hand Pump," Blue Robotics. Accessed: Jun. 30, 2024. [Online]. Available: https://bluerobotics.com/learn/using-the-vacuum-test-plug/

5. Install the pressure relief valve with greased O-ring on the enclosure. Use a torque of 3.5Nm.

Procedure

1. With one end of the rope, securely attach the weight to the enclosure
2. Attach the other end of the rope to the buoy
3. Using the rope, slowly emerged the enclosure in the deep end of the pool until it reaches the bottom
4. Wait 4 hours
5. After 4 hours, bring the enclosure out of the water
6. Untie the weight and the enclosure
6. Inspect visually the enclosure. Verify the following criteria**:**
    a. **CRIT-3: The enclosure shall be exempt from any visual damage**
    b. **CRIT-4: The interior of the enclosure shall be exempt from any water**

APPENDIX F
ENCLOSURE GENERAL WATER TESTING PROCEDURE

ENCLOSURE GENRAL WATER TESTING PROCEDURE V1-0

Enclosure general water testing procedure

ID: TEST-MEC-GEN-002-V1-0
Rev: V1-0 Applicable submarines: N/A

Abstract

This document presents a generalized testing methodology to validate the watertightness of custom mechanical enclosure.

Revision history

Version	Date	Changes
V1-0	2024-06-30	• Initial document creation

Prerequisite

Hardware / Tools	Others
Hardware 1x Enclosure to test ?x Sealing hardware (Gasket/O-ring) of the enclosure to test **Tools** 1x Pressure Relief Valve (from bluerobotics) ?x Appropriately sized torque wrenches	**Others** 1x Molykote 111 silicone grease 1x Pool (depth > 3m) 1x Weight 1x climbing rope (length > 4m) 1x buoy

Preparation

The following steps shall be followed before the pool test

1. Inspect all Gaskets and O-rings. Verify the following criteria **(abort test in case of failure):**
    a. **CRIT-1: all Gasket and O-ring shall show have no visual damage such as cracks**
    b. **CRIT-2: all Gasket and O-ring shall be exempt from any contaminants such as dust or sand**
2. Spread Molykote silicone grease over the surface of the Gaskets and O-rings
3. Carefully place gaskets and O-ring on the mating faces
4. Close the enclosure using the appropriate torque spec for the screws, as defined in the specific enclosure documentation.

5. Install the pressure relief valve with greased O-ring on the enclosure. Use a torque of 3.5Nm.

Procedure

1. With one end of the rope, securely attach the weight to the enclosure
2. Attach the other end of the rope to the buoy
3. Using the rope, slowly emerged the enclosure in the deep end of the pool until it reaches the bottom
4. Wait 4 hours
5. After 4 hours, bring the enclosure out of the water
6. Untie the weight and the enclosure
6. Inspect visually the enclosure. Verify the following criteria:
    a. **CRIT-3: The enclosure shall be exempt from any visual damage**
    b. **CRIT-4: The interior of the enclosure shall be exempt from any water**

APPENDIX G

AUV7 PRESENTATION

AUV7

Introduced in 2017, AUV7 marked the club's pioneering venture into a cross-shaped submarine design. This unique design is complemented by usage of carbon fiber to create the central hull of the submarine.

Over time, several overhauls were made to enhance the submarine's performance. In 2022, the electrical and software systems underwent a thorough revision to better integrate with AUV8's internal setup.

Features

- Central 6 layers carbon fiber hull
- 4 anodized aluminum access port for improved robustness
- 2 cameras
- 8 motors
- 6 degrees of freedom
- Navigation sensors: DVL (with pressure sensor), IMU
- Marker system
- On-board computer: Jeston Xavier AGX
  - 8 cores ARM processor @ 2.2 GHz
  - 512 cores graphic accelerator @ 1377 MHz
- Autonomy: 1-2 hours (depending on tasks).
- Passive sonar (3x Brüel & Kjær 8103 array)

APPENDIX H

AUV8 PRESENTATION

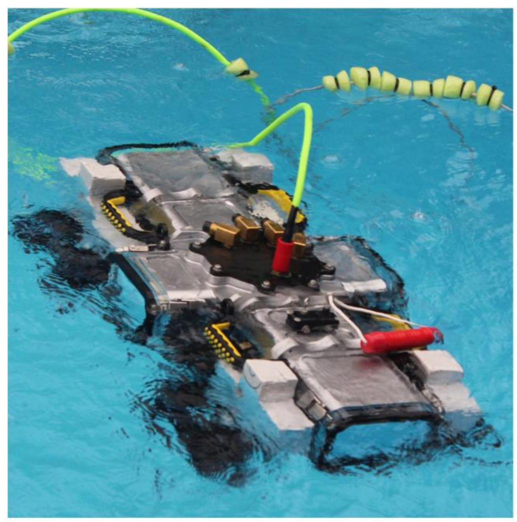
AUV8

Originally designed in 2020, AUV8's first participation in the robosub competition took place in 2022 due to health restrictions.

AUV8 follows the concept of a cross-shaped submarine with a few modifications for improvement. The left and right segments of the submarine were shortened to bring the heaviest components closer to the center, thereby increasing the submarine's maneuverability. Carbon fiber was replaced with a fully aluminum hull.

Features

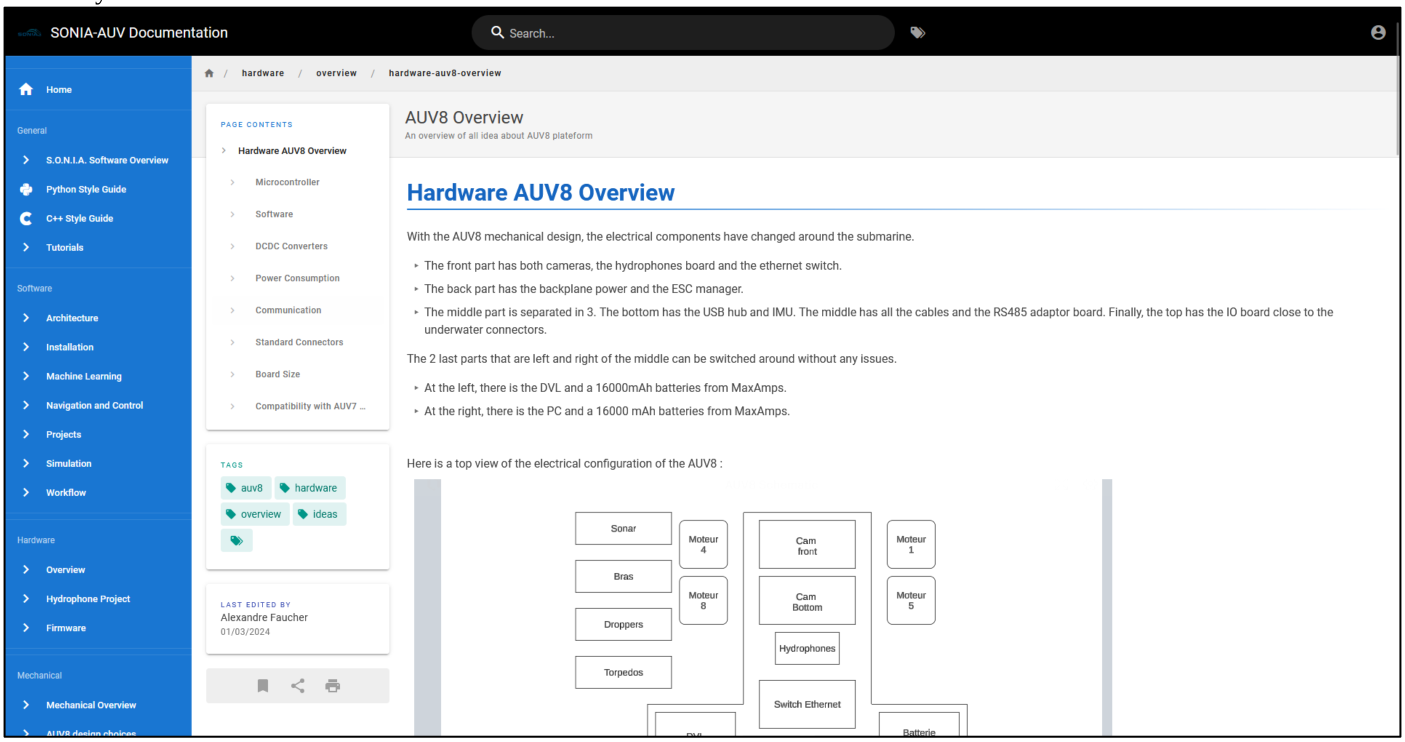
- Anodized aluminium hull
- 8 motors
- 6 degrees of freedom
- Navigation sensors: DVL, depth sensor, IMU
- Markers system
- Torpedoes system
- On-board computer: Jeston Xavier AGX
    - 8 cores ARM processor @ 2.2 GHz
    - 512 cores graphic accelerator @1377 MHz
- Autonomy : 1-3 h (depending on the tasks).
- Passive sonar (4x Brüel & Kjær 8103 array)



APPENDIX I

SONIA AUV PUBLIC DOCUMENTATION

The club SONIA AUV has been hosting a publicly accessible wiki (wiki.sonia.etsmtl.ca/) for the past four years. The wiki shows a view of the different project and design decision made on the submarines. The goal of the wiki is to share the knowledge accumulated over the years with both future team members and members from other teams.



The club also host its full software stack on GitHub (github.com/sonia-auv) under various open licences with over 100 projects publicly available.