Team Inspiration's Autonomous Underwater Vehicles: Onyx and Græy

Carina Dumitrascu (team lead), Ruth Tu

Abstract— In its seventh year in RoboSub, Team Inspiration, having two proven vehicles, will be primarily focusing on the software enhancement of the Autonomous Underwater Vehicles (AUVs), Onyx and Græy (Fig 1). With many team members new to robotics and software, we leverage simple robot prototypes and Artificial Intelligence (AI) tools to foster the learning of fundamental skills. We applied the lessons we learned to better protect our critical components, thus increasing stability and reliability. To support efficient operations, we improved the WiFi and tether communication network. We apply modularity and continued integration to ensure efficiency in software development. Lastly, we refined the competition strategy of operating two subs.

Fig. 1: CAD of Onyx (left) and Græy (right).

I. INTRODUCTION

Team Inspiration is a community-based robotics team that has been involved in robotics competitions since 2011, participating in all major RoboNation competitions. Our team ranges from middle school and high school students to San Diego Mesa College students. We prioritize STEM Outreach to enforce our learning and share knowledge [15]. This year, our focus is on advancing novice members to match the capabilities of Onyx and Græy. In order to onboard them, an autonomous land robot, the MonkeyBot, was developed, which taught the team the fundamentals of systems engineering: the need for mechanical, electrical, firmware, and software considerations, with the ability to perform missions in mind.

Our diverse team dynamic allows for many learning opportunities with fresh perspectives and energy to the team, ultimately equipping students with the skills necessary to succeed in the current workforce.

II. COMPETITION STRATEGY

Building on lessons learned from last year, we prioritized reliability for this year. Since the capabilities of Graey and Onyx are competition-proven, we decided to primarily maintain the existing hardware system by refining the kill switch and electrical power management. Our priorities rest in establishing reliable communication between the base station and the vehicles via tether/Wi-Fi, electrical safety, and testing in two subs competition-like environments to elevate the skillset of team members.

Since the whole team consisted of inexperienced software members, even understanding the complexity of the AUVs was beyond what they were capable of when joining. With a three-month season and the members' limited schedules, we employed AI tools to assist in learning. Learning from an industry professional, our team learned how to use AI to enhance engineering, match industry requirements, and meet work expectations (Appendix C). The team learned to effectively prompt chatbots with specialized responses, using our understanding of engineering principles and requirements to further guide the prompt. By applying the resources available to us effectively, team members can quickly transition from learning about the system to applying their knowledge for its improvement. System engineering remains integral as we analyze systems, conduct testing, and solve problems. Our strong understanding of the system, supplemented with the use of AI, enhances our capabilities to prepare for competition in the limited time frame that we have.

Our strategy is to divide the tasks between our two AUVs, with an emphasis on Onyx, our main sub. While Graey is tailored for the *Slalom, Style,* and *Return Home* missions, Onyx is equipped to carry out all other missions. This setup provides a platform for inter-vehicle communication points and maximizes the time bonus points. We aim to enhance our navigation capabilities by refining our approach beyond dead reckoning to incorporate computer vision (CV) and station keeping. The general mission area is based on observations of the competition layout, similar to real-world mission planning, utilizing geographic features, and then using CV to fine-tune the final approach.

Operating two vehicles has proven to be difficult at the competitions. To mitigate future issues, the team will prioritize practicing with two subs during testing.

A. *Onyx and Græy Mission Sequence*

Onyx completes the mission sequence (Fig. 2) of *Gate, Slalom, Bins, Torpedoes*, *Octagon*, inter-sub communication, and *Return Home* while simultaneously, Græy completes *Coin Toss, Gate, Style Point,* and *Return Home*. By coordinating Onyx and Græy, they complete missions sequenced together in order to achieve inter-vehicle communication points.

Græy will pass through, save the gate location on an internal map, station hold while Onyx completes its mission sequence, and afterwards return to the *Gate* to roll two revolutions for style points.

Performing roll styling points and returning home by Graey will allow minimum navigation error impact on Onyx, and guarantee return points in the case of Onyx's long navigation error. However, Graey will have the capability to do *Slalom* and *Octagon* surfacing as needed.

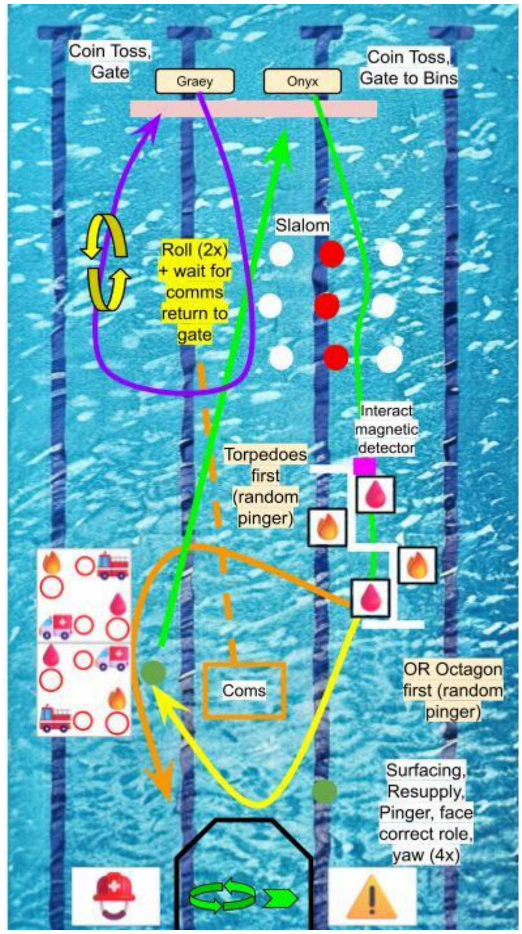

Fig. 2: Two-sub competition run strategy

B. *Order of Mission Development*

1) *Coin Flip*: This task is performed due to its simplicity. We will use a coin flip to randomize Græy's starting orientation. Græy will submerge and yaw to face the gate using the VectorNav Inertial Measurement Unit (IMU), and go through either side of the gate.

2) Gate: This mission is a prerequisite for both AUVs to perform further mission tasks. We aim for Disruptive capabilities this year through random role assignment via a coin flip. Our strategy is to have two mission sequences prepared for *Search & Rescue* or *Survey & Repair*. After the coin flip, the corresponding sequence for each role assignment is loaded. Each AUV utilizes CV via the forward camera to detect the assigned image and pass under it.

3) Slalom: Once through the *Gate*, Græy localizes, adjusting depth to level and heading to center the pipes. Approaching the red pipe, Græy will utilize the front-facing OAK-D camera to reach it within a set distance, move laterally, and then forward until the next red pipe is detected, repeating the process on the same side of the *Slalom*, achieving Advanced capabilities. Based on a pre-loaded mission map manually generated by the team, after timed failure of detection, Græy will utilize waypoints to localize within a specified distance to the red pipe, looping the detection to proceed.

4) Octagon - Surfacing: Due to the high points and low risk, *Surfacing* is the fourth priority. Once Onyx navigates and localizes, it will use the front camera to approach the Octagon. The bottom camera centers, then Onyx yaws to face the corresponding icon and surfaces.

5) Return Home, Style, and Intersub Communication: The long-distance navigation demonstrated for surfacing can be applied to return to the starting gate. Past the *Slalom*, Græy will localize, enabling station-keeping and waiting. Once finished, Onyx continuously sends signals until Græy responds with an acknowledgment packet that will be physically signified through light-emitting diode (LED) lights. Græy returns through the gate, performing her roll maneuver at the end to mitigate navigation disruptions. Onyx will stop sending signals and proceed to *Return Home,* utilizing the saved location from Græy, and Græy will *Return Home* to signal the completion of the run.

6) Bins: Onyx will utilize the forward camera to approach the pipeline for the bins. Onyx will detect the light and navigate to the light, interacting with the magnetic detector to turn off the light. Proceeding, the bottom-facing camera then deciphers the image of the closest bin. If the assigned image is not detected, the forward camera navigates to the next bin, ascending in order for the bottom camera to view the image. After locking onto the bin, Onyx will account for the offset of the dropper from the camera, locking its station hold and dropping a marker into the bin. Onyx continues down the path until both marker droppers are in the corresponding bins.

7) Torpedoes: This task demands more precision compared to *Bins*. After localizing, Onyx notes its position, saving the location for station hold. After the target is centered in the camera, an offset is applied to center it to the launcher. Based on this procedure, Onyx will first fire a torpedo through the larger opening (🔥 🩸), then align and fire through the smaller opening (🚒 🚑).

8) Octagon - Resupply: This complex task requires localizing, identifying, grabbing, and releasing objects. Using its bottom-facing camera to localize and identify each item, Onyx's gripper will supply the item, surface with the item, then face the correct role image based on the number of items, and descend to depth and release each in its respective bin. After completion, Onyx will yaw the same number of revolutions as items in the bins.

9) Pinger Detection: This task requires the detection of the random pinger to guide Onyx for its mission sequence. Onyx utilizes a hydrophone array to detect the pinger. Based on which pinger is activated first, Onyx stops its current mission task and proceeds accordingly: *Octagon,* then *Torpedoes* or vice versa.

III. DESIGN STRATEGY

A. *Mechanical Subsystem*

Our innovative main vehicle frame utilizes 80/20 aluminum extrusions for its modularity, allowing us to easily adjust the mounts of the main electronics enclosure, thrusters, and payloads. The frame can be assembled quickly at a low cost. The AUVs have a holonomic thruster configuration with four vertical thrusters to maintain six degrees of freedom. With ease of vertical adjustment to the height of components, thrust vectors are balanced to the center of mass, ensuring the vehicle can travel forward and laterally while minimizing unintended rotational motion. More details about our mechanical subsystem can be found in our 2021 Hull Design video [10].

Leveraging a systems engineering approach, we apply the lessons we learned to identify and address failure modes that occurred in the previous season. For example, the kill switch easily detached from the 3D-printed mount during competition, so we mitigated repeated destruction of the latches by selecting larger-diameter fasteners, eliminating stress concentrations in the clamps. Utilizing 3D-printed mounts for rapid iteration, we also increased the polyacetic acid (PLA) density to 100% infill on the prints to enhance their strength.

Based on the tribulations from team members bending the pins on the Blue-Trail 8-pin connectors when making the tether connection, we implemented SubConn wet-mate connectors with larger-diameter pins to mitigate future risk. The team refined an open-source gripper design (Fig. 3) for functionality and strength by developing specific failure points, sustaining the main structure [14]. We upgraded from last year's straight-edge gear to herringbone, eliminating backlash issues.

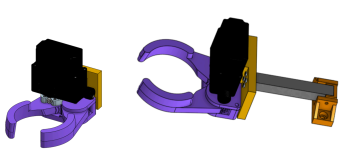

Fig. 3: Gripper Design. The orange component represents the failure point.

B. Electrical Subsystem

Onyx and Græy's electrical systems received minimal change, staying consistent with the previous year. We continue using our proven electrical system to minimize time on electrical redesign and integration. We focus on testing and expanding new capabilities rather than complicating systems that already perform reliably.

Continuing from last year's update to the kill switch, which resolved the risk of back electromagnetic fields (EMF) to the main computer, the power supply and relay are still susceptible to damage from back EMF spikes. To address this, we upgraded to solid-state relays including diodes, which ensure that the current flows in one direction, towards the motors, clamping and dissipating the spike before it could reach sensitive components.

C. Software Subsystem

Our codebase is written in Python and organized into distinct core modules (Fig. 4) for easier management and scalability (including Sensor Core, Localization Core, Robot Control Core, and Mission Planner Core). Robot Operating System (ROS) is utilized for interprocess communication.

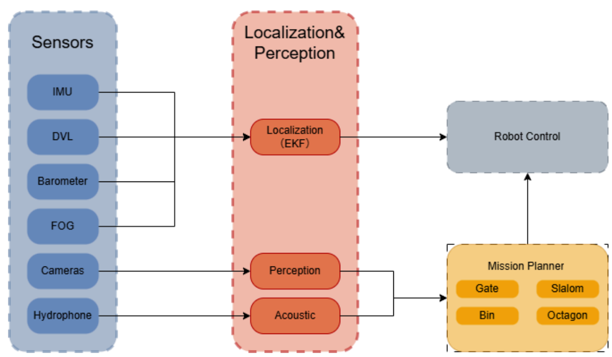

Fig. 4: Current software stack

1) Robot Control:

The movement of our AUVs is governed by the PixHawk Cube Orange flight controllers utilizing ArduSub. Our software utilizes ROS topics to transfer sensor data from the IMU, Doppler Velocity Log (DVL), and barometer to a class called RobotControl via our Application Programming Interfaces (APIs). The sensor information and Proportional-Integral-Derivative (PID) control govern the direction and magnitude of commands for the AUV's translational and rotational movement. These commands are sent via a Micro Air Vehicle Link (MAVLink) extendable node for Robot Operating System (MAVROS) topic to the flight controller, which interprets them to send Pulse Width Modulation (PWM) signals to the thrusters.

2) Localization & Navigation:

a) *Computer Vision:* We developed a robust object detection algorithm using OpenCV2 and Ultralytics's You Only Look Once (YOLO) version 8. We prioritized pattern and shape detection since it proves more reliable than color-based detection in the presence of varying sunlight angles and blue hues from the surrounding water. In order to increase the detection confidence of mission elements, the frame is preprocessed using White Balance, which reduces the blue tint from the pool, and Contrast Limited Adaptive Histogram Equalization (CLAHE), which provides better overall contrast. For *Slalom*, we automated the Hue, Saturation, and Value-based (HSV-based) color segmentation labeling to quickly train YOLO models to accurately detect the pipes in varying lighting conditions.

b) *Navigation Algorithm:* Our navigation algorithm, similar to real-life mission planning, maps the competition into a grid and follows a pre-planned sequence. The algorithm utilizes a DVL to track translational movement, while the IMU tracks the azimuth heading. This enables the AUV to navigate to the general area of each mission, where objects are within the limited detection range of the cameras. The AUV will then search by yawing within a range of headings based on the general location of the course object. For example, the AUV may search for the *Slalom* channel by yawing 60 degrees in each direction, then approaching forward for a distance if the pipes are not detected.

c) *Station-Keeping:* We utilize the DVL for a station-keeping function during the *Torpedoes* and *Bins* mission and for Græy's standby before *Return Home*. When called, the function saves the AUV's position vector. A PID control loop will send forward and lateral commands to the flight controller to correct any drifting until another process signals the function to stop.

3) *Sensor Fusion*: We adopted and modified from the custom Kalman Filter that the University of California, San Diego (UCSD) capstone team fused the IMU alongside the DVL in 2025 [11]. Our current Kalman Filter uses the IMU roll, pitch, and yaw to provide orientation for the DVL-measured body velocities. A rotation matrix takes the DVL that converts the relative-body velocities with the position from the IMU to convert them to the world frame. The Kalman filter improves the accuracy of our navigation system.

4) *Mission Planning:* Each mission is governed by event flags, which can be a mission timeout or a boolean value indicating mission success/failure. Once a flag is triggered, the AUV transitions to the next mission. For example, after dropping the markers for *Bins*, a boolean value is sent to indicate mission completion, prompting navigation to the next mission. When all of Onyx's flags are activated, she will communicate with Græy to begin the *Return Home* sequence.

IV. TESTING STRATEGY

A. Testing Approach

We utilized the systems engineering approach, which breaks mission requirements into the simplest components, to guide our testing strategy. The use of unit/component tests, especially those on land or on a test bench, ensures rapid testing prior to integration. The subsystem tests are performed for individual functions, like the gripper, or for simple missions, such as *Coin Toss*. System testing includes multiple sensors and mission sequences. During each test, the team documents the state of the AUV and each step through a test plan template (Appendix B.1). Detailed blog information of our testing is provided on the team website [12].

B. Unit Test Case Studies

To effectively utilize the MATLAB simulation, we have to establish a baseline for configuring MATLAB 2025b with the Unreal Engine and running default simulations to understand how the AUV handles tasks [13]. Once this is set, the environment serves as a tool that allows the user to scan parameters, like mass and buoyancy or PID control, without the need for physical hardware, which is then advanced through algorithm training using simulated data from the AUV's IMU, DVL, and cameras generated by custom MATLAB scripts. We plan to utilize the simulation to tune PID controls for faster experimentation and ease of iteration.

The gripper undergoes unit tests on land and water to evaluate its accuracy in grasping objects, then undergoes a subsystem test. This staged approach enables rapid identification and resolution of issues prior to integration testing, where ongoing evaluation confirmed its effectiveness within the full system.

C. Subsystem Test Case Study: Inter-Vehicle Communication

In order to demonstrate inter-vehicle communication, LED lights signal the sending and receiving of packets. Once LED strips are unit tested for performance, a subsystem test is required to demonstrate the connection of the modem to the LED lights, then between the AUVs. We ran a test verifying the LED could signal LED flashes when sending and receiving messages, thus ensuring inter-vehicle communication by validating the modem and LED connection.

D. Systems Test Case Study: Prequalification Tests

The team prioritizes water time at competition and in practice. Thus, we completed pre-qualification within the first month of kick-off, allowing new members to learn controls and operations quickly. By focusing on pre-qualification, our new team ensures the function of basic capabilities for both AUVs and the understanding of students.

V. ACKNOWLEDGEMENTS

Team Inspiration would like to thank the supporters below for monetary and equipment donations:

Diamond: Advancing Science, Technology and Art, the Chen family, Robonation/ONR, and Gilman Charitable Fund.

Platinum: Nvidia.

Gold: L3Harris, Qualcomm, and tinyVision.ai.

Silver: Blue Trail Engineering, Blue Robotics, and Chris Freeman.

Bronze: ePlastics, Manna's Martial Arts, and Sunrez.

We are grateful to our mentors Jerome Bonhomme, Venkat Rangan, Jack Silberman, Alex Szeto, and Teresa To, who helped organize the team, continuously reviewed our work, provided technical suggestions, and guided team planning.

We would like to thank our alumni mentors Khushee Goel, Colin Szeto, Mabel Szeto, Eesh Vij, Chase Chen, Maxime Ellerbach, Lindsay Wright, and Leonard Wright for using their experience to weigh in on design decisions made by the team. Team Inspiration would also like to thank the Office of Naval Research and RoboNation for the opportunity to compete in a prestigious event such as RoboSub.

Finally, we would like to thank our parents for their support, without which none of our accomplishments would have been possible.

References

[1] C. Chen, C. Szeto, B. Tran, H. Tran, and L. Wright, "Team Inspiration's Autonomous Underwater Vehicles: Onyx and Græy," Team Inspiration, San Diego, CA, USA, 2025.

[2] K. Chen, A. Talamantez, K. Goel, A. Prasad, and L. Wright, "RoboSub 2024 Technical Design Report," Team Inspiration, San Diego, CA, USA, 2024.

[3] E. Vij et al., "The Design of Team Inspiration's 2023 AUVs," Team Inspiration, San Diego, CA, USA, 2023

[4] A. Goel et al., "The Design of Team Inspiration's 2021 AUVs," Team Inspiration, San Diego, CA, USA, 2021

[5] C. Szeto et al., "The Design of Team Inspiration's 2020 AUVs," Team Inspiration, San Diego, CA, USA, 2020

[6] M. Szeto et al., "Team Inspiration 2019 RoboSub Journal," Team Inspiration, San Diego, CA, USA, 2019

[7] E. Vij et al., "Maritime Robotics - Autonomous, Unmanned, Multi-Domain Vehicles Challenge," Team Inspiration, San Diego, CA, USA, 2024.

[8] K. Chen, C. Szeto, B. Tran, L. Wright, "The Design of Team Inspiration's 2025 RoboBoat Autonomous Surface Vehicle," Team Inspiration, San Diego, CA, USA, 2025

[9] RoboNation, "Team Handbook - RoboSub 2026," 2026. [Online] https://robonation.gitbook.io/robosub-resources

[10] Inspiration Robotics 11128, San Diego, CA, USA, *RoboSub 2021 - Hull Design*. (Aug. 2021). Accessed: Jun. 30, 2025 [Online Video]. Available: https://www.youtube.com/watch?v=5O3wI03UST4

[11] R. Nambimadom, June Park, Andy Zhang, Sharfa Pital, Justin Nguyen, Purush Ram, "Sensor Fusion and Localization for Autonomous Underwater Vehicles," University of California San Diego, San Diego, CA, USA, 2025.

[12] Team Inspiration, "Inspiration Robosub 2025 Team Website," [Online]. Available: https://www.inspirationrobosub.com/robosub-2025-1 (accessed June 30, 2025)

[13] Mathworks, "designfilt - design digital filter - MATLAB," 2025. [Online]. Available: https://www.mathworks.com/help/signal/ref/designfilt.html

[14] The Loony Squad, "Loony Claw," [Online]. Available: https://www.theloonysquad.com/open-source/loony-claw

[15] E. Chan et al., "ASTA/Team Inspiration Community and Outreach Activities," Team Inspiration, San Diego, CA, USA 2026

APPENDIX A

COMPONENT SPECIFICATIONS

Component	Vendor	Model/ Type	Specs	Custom/ Purchased	Cost	Year of Purchase
AUV Hull Form/ Platform (Frame, Waterproof Housing)	80/20 Inc.	1010 aluminum extrusion	Weight: 25 oz Size: 22 in x 7.9 in	Purchased and machined in lab	$234	2019
	ePlastics	Acrylic tube, 8 in. series	Size: 8” diameter	Donated	$343	2019
Waterproof Connectors	Blue Trail Engineering	10 cobalt series dummy plug 10 cobalt series locking sleeve 4 cobalt series cable termina-tion kit	https://www.bluetraiengineering.com/product-page/cobalt-series-dummy-plug https://www.bluetraiengineering.com/product-page/removable-cobalt-locking-sleeve https://www.bluetraiengineering.com/product-page/cable-termination-kit	Purchased	$201	2020
	Blue Robotics	1 7.5mm HC Wetlink penetrator	https://bluerobotics.com/store/cables-connectors/penetrators/wlp-vp/		$12	
Intersub communica-tion	Succorfish	Delphis V3.3	https://succorfish.com/wp-content/uploads/2023/01/DELPHIS-Data-Sheet_V9_3.2.pdf	*Donated by a sponsor*	*Donated by a sponsor*	2023

Propulsion	Blue Robotics	T200 thrusters	Full Throttle FWD/REV Thrust @ Maximum (20 V) https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/	Legacy	$200 apiece	2019
	Blue Robotics	T200 propellers	Max thrust: 49.82 N	Legacy	Included with thrusters	2019
Gripper	Blue Robotics	Newton Subsea Gripper	Grip Force: 28 N Jaw Opening: 2.75 in https://bluerobotics.com/store/rov/bluerov2-accessories/newton-gripper-asm-r2-rp/	Legacy	$329	2019
Torpedoes	Blue Trail Engineering	Under-water Servo SER-2000	https://www.bluetrailengineering.com/product-page/underwater-servo-ser-20xx	Custom	$215	2023
Marker Dropper	Blue Trail Engineering	Under-water Servo SER-2000	https://www.bluetrailengineering.com/product-page/underwater-servo-ser-20xx	Custom	$215	2023
Power System (Battery, Converter, Regulator)	Blue Robotics	Lithium-Ion Battery	4s 14.8V 15.6 Ah	Legacy	Legacy	2019
	Blue Robotics	5V, 6A power supply	5V, 6A	Legacy	Came with BlueROV setup; Legacy	2019
	Mouser	Murata MYBSS054R6EBF	54V Power supply https://tinyurl.com/inspirationRegulator	Purchased	$22	2023
	AliExpress	100A Ideal	100A Ideal diode	Purchased	$72	2023

		Diode				
	Custom	Unified Power Controller	Three: 12V 3A, 5V 3A, 3.3V 1.2A Power supplies	Custom	$400	2023
Motor Control	Blue Robotics	Basic ESC	30A brushless ESC https://bluerobotics.com/store/thrusters/speed-controllers/besc30-r3/	Legacy	$36 apiece	2019
CPU	Nvidia	Nvidia Jetson Nano	1.4 GHZ clock speed 4 GB RAM	Purchased	$99 (On Græy)	2021
	Nvidia	Nvidia Jetson Xavier NX	https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit	*Donated by a sponsor*	On Onyx	2024
Teleopera-tion	Blue Robotics	Fathom-X and Fathom-X Tether Interface (FXTI)	Communication: USB 2.0, Ethernet 10/100 https://bluerobotics.com/store/comm-control-power/tether-interface/fathom-x-r1/	Legacy	Installed	2023
Inertial Measure-ment Unit (IMU)	Pixhawk	Invensense® MPU 6000 3-axis acceler-ometer/ gyroscope	32-bit ARM Cortex M4 core with FPU 168 MHz/256 KB RAM/2 MB Flash 32-bit failsafe co-processor	Legacy	Included with Pixhawk on Græy and on Onyx (compass)	2019, 2022
	Dampener	XTORI Pixhawk dampener	Materials: plastic and rubber Weight: 17 g	Legacy	$8	2021
Doppler Velocity Log (DVL)	Teledyne	Explorer	Velocity Range: ±12 m/s Long Term Accuracy: ±0.3% ± 0.2 cm/s https://www.uniquegroup.com/wp-conten	*Loaned by a sponsor*	*Loaned by a sponsor*	2023

	WaterLinked	A50	t/uploads/2022/10/Explorer_DVL.pdf Velocity Range: ±3.75 m/s Long Term Accuracy: ±1.01% https://waterlinked.com/web/content/15701?unique=b4aef6d930bb256c64bae4e0ead8c56661bf12f0	Purchased	$ 5990	2020
Fiber Optic Gyroscope (FOG)	Fizoptika Malta	VG103 S2LND	Input range: 200 °/s https://fizoptika.com/docs/fiber_optic_gyro_specification_vg103s-2lnd.pdf	Purchased	$3,060	2023
Camera(s)	Blue Robotics	Low Light HD USB Camera	Pixel count: 2MP 1080P Onboard H.264 compression chip 32x32mm	Legacy on Græy Purchased for Onyx	$198 (1x) on Græy (2x) on Onyx	2019
	Luxonis	OAK-D Wide Camera	12 MP Resolution 60 FPS max frame rate Focus (Full Frame): 60cm - ∞	Purchased	$ 548 (1x on Græy) (2x on Onyx)	2024
Hydro-phones	Custom	Hydro-phone Circuit	100kHz sample rate for DTOA analysis. Variable gain control. 1-40kHz frequency lock range. Sub-watt and sub-degree precision under ideal conditions.	Legacy	$315	2020
	Aquarian Audio & Scientific	AS-1 Hydro-phone	Linear range: 1Hz to 100kHz ±2dB Horizontal Directivity (20kHz): ±0.2dB Horizontal Directivity (100kHz): ±1dB	Legacy		2019

			Vertical Directivity (20kHz): ±1dB Vertical Directivity (100kHz): +6dB -11dB https://www.aquarianaudio.com/as-1-hydrophone.html			
Algorithms (acoustics)	Custom	Fast Fourier Transform (FFT)	-		Free	2020
Vision	Custom	Open Computer Vision	Color isolation, binary thresholding, contour approximation, erosion and dilation, area thresholding, and Contrast Limited Adaptive Histogram Equalization (CLAHE)		Free/ Open Source	2019
	Ultralytics	YOLO	YOLOv8		Free	2023
Localization and Mapping	In-house	Custom	DVL, Hydrophones, CV		Free	2021
Mission Planner	In-house	Custom	Mission planner		Free	2024
Open Source Software	Open-Source (n/a)	OpenCV, Robot Operating System, Python, C++, Linux	Computer Vision, Inter-process communication, programming, computer operating system		Free	2019

Appendix B.1: Test Template

Test template		**Approval Authority**	
		[Names of Members]	**[Description of members' position]**

Date:	**Date:**			

Test #: [1]	**Location:**	**Risk:** [Low]
Software Version: [Baseline, Preliminary, Version 0.0]	**Hardware Sensors Mounted:**	**Hardware Sensors Used:**

Scope:	**<u>Primary:</u>** pre-qualification **<u>Secondary:</u>** **<u>Tertiary:</u>**

Attendance	**Test Notes**
	Resources Needed: **Environment:** **Expected Results:** **Prerequisites:** **Risk Management:**

	Validation Step Description	**Expected Result**	**Result**	**Reference to Data**	**Pass/Fail**

Appendix B.2: Test Plan and Results

Testing followed a structured, consistent approach aligned with our AUVs. Each requirement was verified through unit, sub-system, and system testing, ensuring that all components functioned and integrated.

Test Name	Requirements	Test Description	Planned Test Date	People Involved
Motor replacement	Verify replacement thrusters provide stable and reliable propulsion without electrical or mechanical failure	1>Bench testing to validate PWM response and thrust consistency. 2>Pool testing to verify waterproofing and stable integration with ESCs and Pixhawk controls during underwater operation.	05/30 - 05/31	Cesar, Yeng
Gripper Testing	Verify gripper integrates correctly with vehicle controls and mission logic	1> Test gripping force and jaw actuation using simulated competition objects. 2> Conduct underwater pickup and release trials to validate reliability under pool conditions.	05/9	Ivan
Gripper Testing	Verify gripper integrates correctly with vehicle controls and mission logic	1> Test communication between the gripper actuator and the onboard computer. 2> Validate mission script integration while performing autonomous pickup and release tasks	05/20 - 5/31	Ivan
Torpedo Launcher Testing	Verify the torpedo launcher can reliably fire projectiles with repeatable accuracy	1> Conduct land-launch tests to validate servo actuation and firing consistency. 2> Perform underwater launch tests to verify waterproofing, targeting alignment, and launch repeatability.	04/25	Carina &Yeng
Station Hold Testing	Verify AUV maintains a stable position and heading underwater using sensor feedback	1> Utilize DVL and IMU data to estimate underwater position and heading. 2> Implement PID control loops to measure drift correction and heading stability during stationary underwater operation.	05/9	Carina
YOLO Modeling	Verify YOLO-based object detection accurately identifies mission targets in underwater environments	1> Train YOLO models using underwater image datasets. 2> Validate detection accuracy under varying lighting and visibility	05/15	Ruth

		conditions. 3> Conduct pool testing to evaluate real-time underwater detection performance.		
Sonar Testing	Verify the sonar system can detect and localize acoustic signals reliably underwater	1> Test the hydrophone array communication and signal acquisition. 2> Validate signal-processing algorithms for pinger localization and noise filtering. 3> Conduct underwater to evaluate real-time detection performance in underwater conditions	5/01	Yeng

Appendix B.3: Prequalification Mission

<table>
<tr><td colspan="2" rowspan="3">Onyx and Graey Prequalification</td><td colspan="4">Approval Authority</td></tr>
<tr><td></td><td colspan="3">[Description of members' position]</td></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><td colspan="2">Date: 3/21</td><td colspan="2">Date: 3/21</td><td></td><td></td></tr>
<tr><td colspan="2">Test #: [1]</td><td colspan="2">Location: lab</td><td colspan="2">Risk: [Low]</td></tr>
<tr><td colspan="2">Software Version: [Baseline, Preliminary, Version 0.0]</td><td colspan="2">Hardware Sensors Mounted:</td><td colspan="2">Hardware Sensors Used:</td></tr>
<tr><td colspan="6">Scope: <u>Primary:</u> pre-qualification
<u>Secondary:</u>
<u>Tertiary:</u></td></tr>
<tr><td colspan="2" rowspan="2">Roles</td><td colspan="2" rowspan="2">Walkie Talkies/ Cell Phones</td><td colspan="2">Times</td></tr>
<tr><td>Time</td><td>Actual</td></tr>
<tr><td>Test Conductor</td><td>Carina, cesar, ruth</td><td>Ground</td><td>N/A</td><td></td><td></td></tr>
<tr><td>Ground Control</td><td></td><td>Lifeguard</td><td>N/A</td><td></td><td></td></tr>
<tr><td>Data Collector</td><td>ruth</td><td>Tether</td><td>N/A</td><td></td><td></td></tr>
<tr><td>Photographer</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Safety Checker/QA</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Person in water</td><td>Carina Cesar</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6">Status</td></tr>
<tr><td>Boat</td><td></td><td colspan="2">GO/NO GO</td><td colspan="2"></td></tr>
<tr><td>Ground</td><td></td><td colspan="2">GO/NO GO</td><td colspan="2"></td></tr>
<tr><td>Pool Deck</td><td></td><td colspan="2">GO/NO GO</td><td colspan="2"></td></tr>
</table>

Attendance	Test Notes
	Resources Needed: Base station, table, tether, batteries **Environment:** The pool **Expected Results:** **Prerequisites:** **Test procedure:** **Risk Management:** Print Heading output topic: rostopic echo /mavros/vfr_hud Print DVL output topic: rostopic echo /auv/devices/dvl/velocity Print PWM inputs: rostopic echo /mavros/rc/override

#	Validation Step Description	Expected Result	Reference to Data	Pass/Fail
	BRIEF PEOPLE ON KILLSWITCH: Might need to pull the killswitch if during an untethered run the AUV doesn't disarm. PREQUALIFICATION MISSION ONYX: In terminal, cd into robosub_2025/auv/missions/ Then in terminal type: nano prequal_mission_onyx.py Then copy paste the code from the prequal_mission_onyx tab of this document into prequal_mission_onyx.py Calibrate the sub heading to be perpendicular to the gate opening, (or imagine where the gate opening would be) When ready to run: Make sure you're in robosub_2025 directory then in terminal run Python3 -m missions.prequal_mission_onyx.py	PSEUDOCODE FOR MISSION: 1.) AUV Locks heading at **0 degrees**, keeping it perpendicular to the gate. 2.) AUV dives to a depth of **0.5 meters.** 3.) AUV surges forward **3 meters** through the gate and is now in front of the pole. 4.) AUV starts square sequence 5.) AUV sways **left 1 meter**, 6.) AUV surges forward **1 meter,** 7.) AUV sways right **2 meters**, 8.) AUV surges backwards **1 meter**		

		9.) AUV sways left **1 meter** 10.) AUV finished the square sequence 11.) AUV surges backwards **2.5 meters** through the gate. 12.) AUV then surfaces and disarms.		
	GRAEY: In terminal, cd into robosub_2025/auv/missions/ Then, in the terminal, type: nano prequal_mission_graey.py Then copy and paste the code from the prequal_mission_onyx tab of this document into prequal_mission_onyx.py			
3	Calibrate the subheading to be perpendicular to the gate opening (or imagine where the gate opening would be)			
	When ready to run: Make sure you're in the robosub_2026 directory, then in the terminal run			
	TETHERED TEST: Run the mission with the tether on to see if the program even works; comment out the 40-second delay.			
	IF TETHERED TEST WORKS WELL, TRY UNTETHERED TEST: First, make sure to uncomment the 40-second delay, and change it to what time best fits for you. Then follow this sequence: Calibrate heading -> Run program -> untether -> put AUV in water facing gate -> Script starts running -> Mission ends			

Results and Reflections:
We did not initialize properly at first, that's why Jayden's and my code did not work. I also think we need to clean up the 2026 templates so that they are up to date with the current repo
Our mission was able to run multiple times
Graey had a lot of trouble starting; there was a broken pipeline, and there was issues where the sensors were not working properly

- **Cesar resets all the sensors, and Graey is able to run properly**
- **So document in troubleshooting that if Graey has issues with not reporting information then we should reset sensors**
- **Able to do yaw maneuver for onyx with the gate, had some kinks that need working out, go 0 90 180 270 0 instead of 0 180 0 180**
- **Did not complete intersub communication**

Success:
Both subs were able to successfully perform the pre-qualification once we adjusted mission planning.

Appendix C: Engineering With AI: Meeting Review
Presented by Jerome Bonhomme

What You Just Learned And Where to Go Next

Agentic Software Engineering · Team Inspiration Robo Sub

The big idea

Software engineering is changing fast, but not in the way most people think. The change isn't that AI writes code for you. The change is that *coding*, the act of translating a clear idea into working instructions, is no longer the hard part. Large language models do that reasonably well. What remains hard, and what AI cannot do for you, is everything that comes before the code: figuring out what problem is actually worth solving, choosing the right solution for it, and describing both clearly enough that another intelligence can act on them faithfully.

That shift changes what it means to be a good software engineer. The most valuable skill is no longer how fast you can write functions. It's how precisely you can think, and how clearly you can write that thinking down.

What you learned today

Large language models are trained on vast amounts of text and learned to predict what comes next. There is no database, no lookup table, just prediction at scale that produces something remarkably close to reasoning. The model only knows what you give it in the current session. Think of it as a brilliant contractor with amnesia: skilled, but entirely dependent on the briefing you provide.

Agents are LLMs given tools, the ability to read files, write code, run it, read errors, and loop. Replit AI is one of these. When you ask it to "read spec.md first," it actually does. That's the leverage point you used today.

The naive one-shot trap is what happens when speed seduces you into skipping the thinking. You get something that compiles and might look decent, but it solves no particular problem for no particular user. The AI made every decision you didn't make. Most of those decisions were wrong for your context.

The three questions that every good product must answer in order: Is this problem worth solving? What is the best solution for it? What is the best implementation of that solution? AI has almost nothing to contribute to the first question. It can help you think through the second. It excels at the third, but only once the first two are answered clearly.

Documentation is not bureaucracy. It is thinking made visible. Writing spec.md didn't help the AI because it added words to a file. It helped because writing forced you to resolve ambiguity that felt fine in your head. If you cannot write something down clearly, you do not understand it clearly. Fix the understanding before you prompt the agent.

Context beats prompt. The emerging term in professional AI engineering is *context engineering*, the discipline of deciding what information goes into the model's attention window, not just how to phrase a single question. A mediocre prompt inside a well-structured project with a clear spec outperforms a polished prompt dropped into a chaotic, undocumented codebase. spec.md, design.md, and sample_data.md were your context stack today.

The agent pipeline you ran manually, PM → Architect → Developer → Reviewer → QA, is what professional teams are automating at scale. Each role is a distinct AI interaction with a different brief and different context. You are the engineering manager in the center: the only one who knows what the product should feel like, who the users are, and what done actually means.

Going deeper

These are the most useful resources available right now, ordered from approachable to advanced.

Start here

- **Anthropic's Prompt Engineering Guide**, the best practical introduction to getting consistent results from any LLM. Free, browser-based, no account needed. docs.claude.ai/en/docs/build-with-claude/prompt-engineering/overview

- **Anthropic's Interactive Prompt Engineering Tutorial**, a hands-on 9-chapter course where you write and test prompts yourself. More valuable than any video. github.com/anthropics/prompt-eng-interactive-tutorial

- **DeepLearning.AI short courses**, Andrew Ng's platform has free 1–2 hour courses on prompting, agents, and building with LLMs. Start with *Prompt Engineering for Developers* and *Building Systems with the ChatGPT API*. learn.deeplearning.ai

Go further

- **"Effective Context Engineering for AI Agents"**, Anthropic's engineering blog post that introduced the term *context engineering*. Explains exactly why spec.md worked today, at a technical level. anthropic.com/engineering/effective-context-engineering-for-ai-agents

- **Anthropic's 2026 Agentic Coding Trends Report**, a research report on how professional engineering teams are actually using agentic tools right now: what they delegate, what they keep, and how the software development lifecycle is changing. resources.anthropic.com (search "2026 Agentic Coding Trends Report")

- **Replit's documentation on AI agents**, since you're already using it, understanding what the agent can and can't see in your project will make you significantly more effective.

docs.replit.com

If you want to build seriously

- **LangChain and LangGraph**, the most widely used frameworks for building multi-agent systems in Python. LangGraph is particularly good for the kind of pipeline you saw today (sequential agents with defined roles). python.langchain.com

- **OpenAI Agents SDK**, a lightweight Python framework released in 2025 for building multi-agent workflows. Well-documented, with a low learning curve if you know some Python. platform.openai.com/docs/guides/agents
-
- **"AI Engineering" by Chip Huyen**, if you read one book on this topic, make it this one. Covers the real-world engineering challenges of building with LLMs: evaluation, inference, deployment, and the gap between demos and production systems.

One thing to remember

The engineers getting the most out of agentic tools in 2026 are not the ones who know the most prompting tricks. They are the ones who can think most clearly about problems, who can talk to users, understand pain, make architectural decisions, and write all of it down precisely enough that another intelligence can act on it.

That skill is older than computers. It is clearer thinking and clearer writing. AI just made the value of it impossible to ignore.

"AI will not take your job. Someone who knows how to use AI will take your job."

Jensen Huang - Founder and CEO NVidia.

Appendix D: Schematics

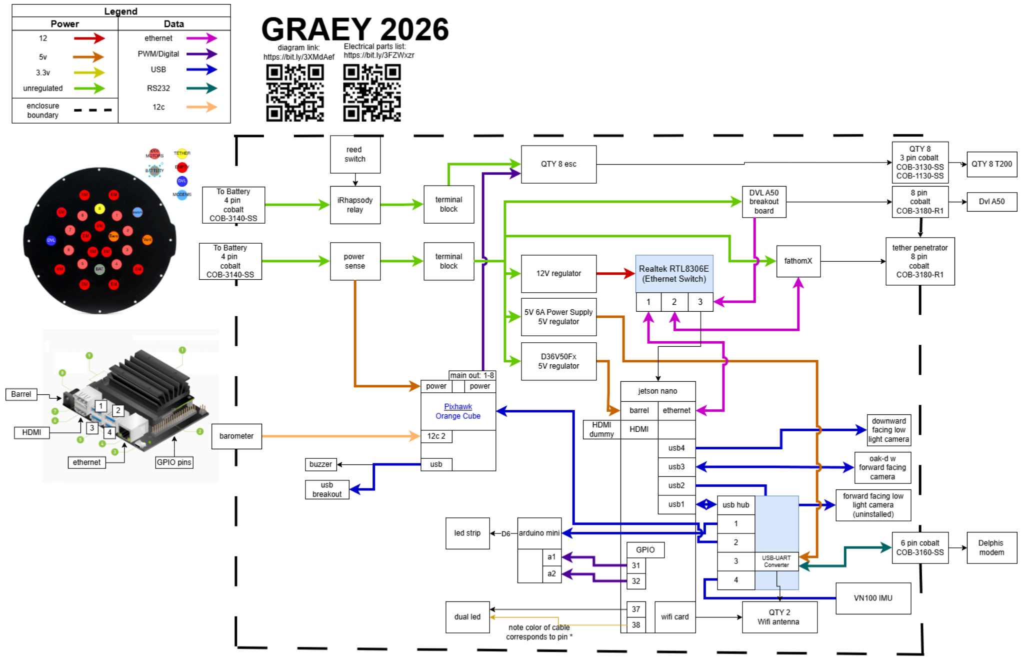

Fig. 5: Græy's Schematic

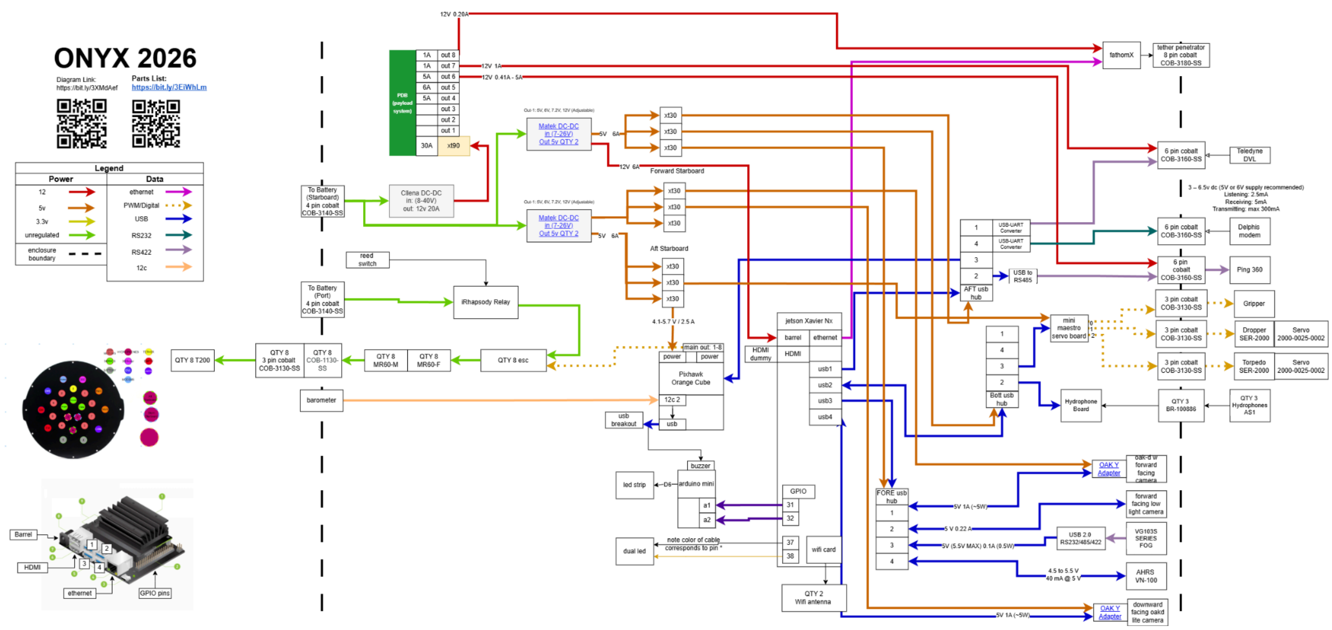

Fig. 6: Onyx's Schematic

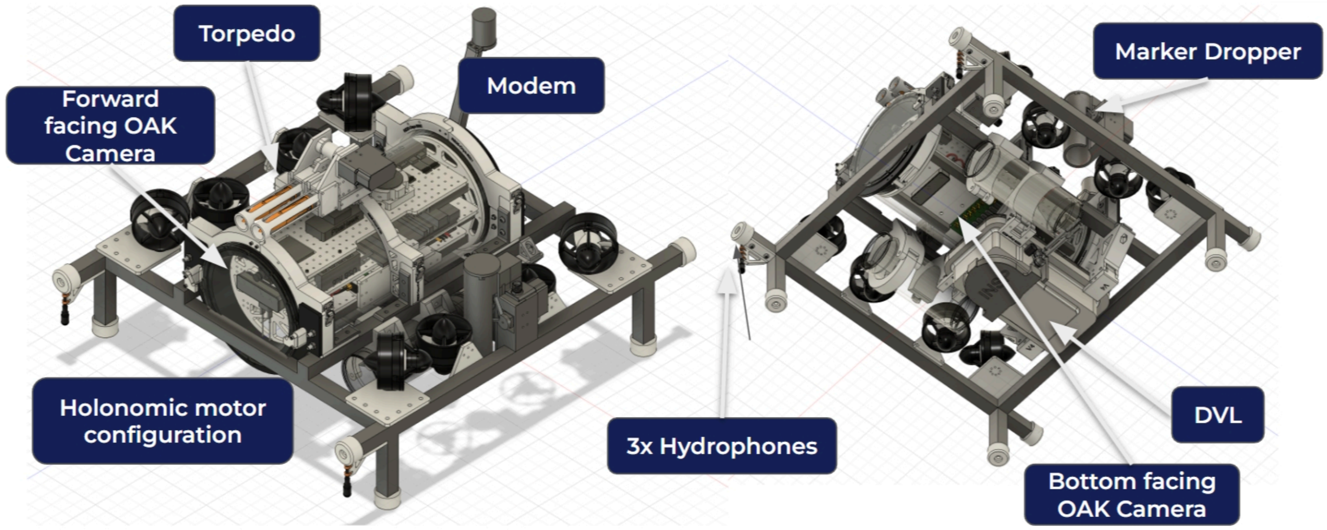

Fig. 7: Labelled CAD of Onyx

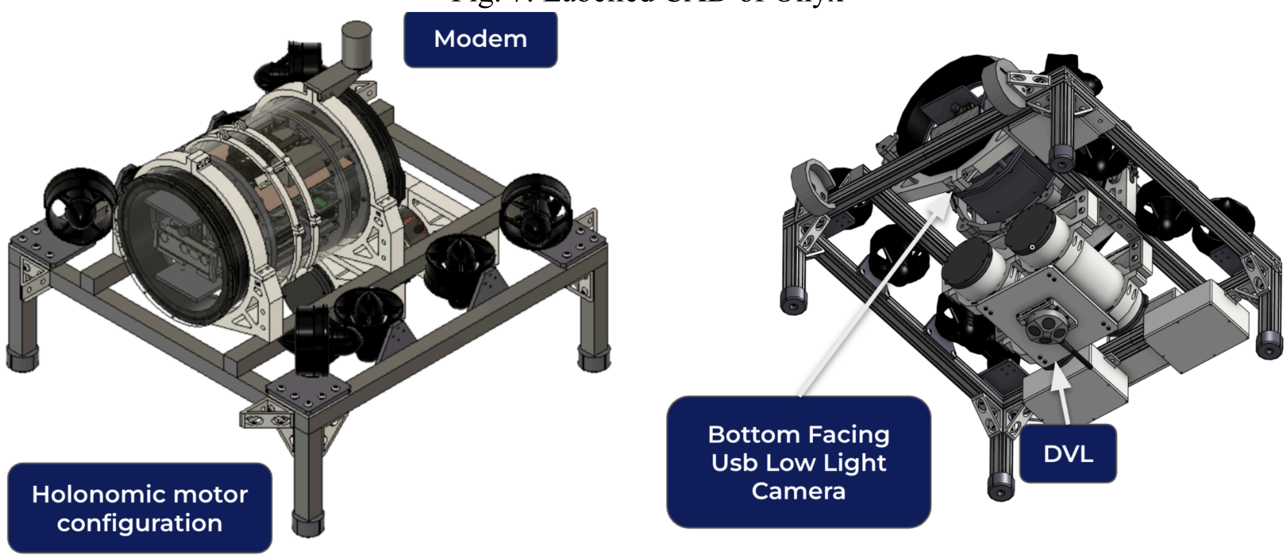

Fig. 8: Labelled CAD of Græy