Team Unsinkable Technical Design Report 2026

Logan Holmboe, Tatum Kelley, Adam Ahmad, Sirio Jansen-Sánchez, Jesse Puricelli, Emily Coello
Embry-Riddle Aeronautical University, Daytona Beach, FL, 32114

Abstract— **Team Unsinkable's 2026 RoboSub effort focused on improving autonomous reliability, modularity, and task efficiency through continued development of the Hammerhead Autonomous Underwater Vehicle (AUV), and the introduction of a secondary lightweight platform, Remora. Mechanical improvements included redesigned electronics trays, adaptable camera mounts, updated payload deployment systems, and enhanced safety features. Electrical upgrades emphasized improved power distribution, tether reliability, and robustified onboard sensor integration. Software development centered on ROS2-based architecture, behavior-tree autonomy, waypoint navigation, and computer-vision pipelines utilizing both YOLO object detection and feature-based keypoint matching. Extensive benchtop, simulation, and underwater testing validated subsystem integration and operational performance while supporting continued development of autonomous mission execution capabilities for RoboSub 2026.**

Keywords—Autonomous Underwater Vehicle (AUV), ROS2, YOLO, RoboSub

I. Competition Strategy

Our competition strategy for the 2026 RoboSub season was to improve upon previously attempted tasks and increase reliability in our task completions. Rather than pursuing every available task, we prioritized refining systems associated with tasks we have previously seen success in. This approach allows the team to improve reliability while continuing development of new task attempt strategies and encourages technical development of newer members.

A major focus for our team this season was the implementation of efficient behavior trees in order to streamline our decision-making process and allow for concise movement. This decision is due to our desire to focus on what we have already attempted and use it as a learning experience for younger members, while our soon-to-graduate members pass on their experience to upcoming leadership. We also wanted to make a proper attempt at bringing a secondary autonomous underwater vehicle (AUV) into our roster. This lightweight AUV, Remora, is designed to prioritize maneuverability and speed. We chose this so it could make up for any faults in our primary sub and reduce our time per attempt.

II. Task Priority and Selection

The goal of this year's task selection was to maximize achievable points while remaining within our AUV's operational capabilities. Tasks were evaluated using the RoboSub 2026 Team Handbook [1], with emphasis placed on estimated completion time and expected reliability.

Primary focus was placed on tasks involving navigation, maneuvering, and computer-vision-assisted object detection. These objectives aligned with our team's current strengths in movement control, waypoint navigation, and behavior-tree autonomy.

Task	Vehicle
Heading Out	Hammerhead
Begin Assessment	Hammerhead, Remora

Style	Remora
Avoid Debris	Hammerhead
Resupply	Remora
Recon	Hammerhead

Table 1 - Task Distribution

With the addition of the second vehicle, the team divided task responsibilities between both AUVs based on their individual strengths. Hammerhead was chosen for larger and more time-intensive tasks such as slalom and the dropper task. Remora was designed to focus on faster and more maneuverability-based objectives, including style-point movements and octagon navigation. This division of tasks allowed both vehicles to operate within their intended design goals while maximizing overall scoring opportunities.

III. DESIGN STRATEGY

Figure 1 - Hammerhead Render

Figure 2 - Remora Render

A. Mechanical Systems

1) Hammerhead

a) Dropper

This year, we decided to do a complete overhaul of last year's design for the Dropper task. We originally decided on a pin pulling mechanism for release of our markers; however, we had issues with friction building in the mechanism. As a result, we decided to switch to an easier floor-release system. This included a swivel arm that can turn to release each marker individually without complex movement. A stabilizing slot was added for the markers to sit in so that they would drop more accurately onto the target. We even made sure that we would be able to keep the same servo and marker design for convenience.

Figure 3 - Dropper

b) Camera Mounts

With our greater reliance on vision systems this year, there came a need for a more adaptable and efficient camera system. We decided to redesign our computer vision camera mounts and make them easily adjustable. The mounts allow us to position our cameras in 5 different ways, enabling many possible views. Along with this, they allow us to fit two cameras on each mount to support multi-directional coverage.

Figure 4 - Camera Mount

c) Safety Features

The safety of those handling the AUV is our top priority; we are constantly finding ways to improve operational safety. This year, we made two main additions to our vehicle that were purely chosen for their ability to increase the safety of those operating the vehicle.

We first added thruster guards to every AUV in our team's arsenal. Doing this ensures that, while in close contact with the operating AUV, nobody can be injured by the thruster blades. We also created small end caps for the frame of our AUV to reduce bump damage to people, and the environment, as it moves in the water.

2) Remora

a) Electrical Trays

We chose to update the contents of our second AUV's internal systems from the ESCs to the IMU. In order to do this, we needed to completely scrap the internal frame that held the outdated components and begin redesign. We started with a single center-plate design, but ultimately decided on a two-layered build to increase the surface area of workable space. The trays are held in place with two end caps that improve stability and provide an area for a front mounted camera as well as a slot for our e-stop relay near the back.

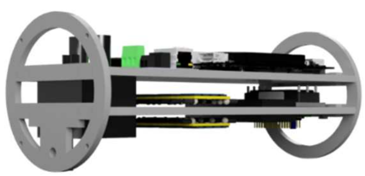

Figure 5 - Remora Electronic Trays

b) Vertical Stabilizers

As we upgraded our second AUV and began to fit it with more sensors we need to make a space for the DVL to fit. We created an easily attachable mount to hold it in place but then ran into a problem with where the DVL sat in relation to the ground. In order to avoid interference with the DVL's sound waves we needed to extend the AUV's height. We accomplished this by adding brace like stabilizers that will raise the DVL away from the floor. Importantly the design was made to hold the AUV's full weight and was tested to show resistance to large deformation when dropped.

B. Electrical systems

1) Hammerhead

a) Tether Repairs

This year we had a major issue occur with our only tether and had to make major repairs in order to have a chance to continue testing while members were available. We found during testing that there was a loss of connection through one of the pins coming from our previous graft point. We prioritized repair of this part so that testing could continue as fast as possible. The repair required us to fully remove our previous splice joint and create a brand new one. This gave us a chance to increase the safety of our solder joints at the point of connection and improve the sealing to ensure the mistake couldn't happen in the future.

b) Camera USB Hub

As the number of cameras utilized aboard Hammerhead increased as did the bandwidth requirements and number of ports. To accomplish this, the Deepwater Exploration 8-Port hub was utilized. This allowed for the compressed transport of the camera feeds and reduced the number of required ports to the main computer unit to a single USB 3.0 port.

c) Bus Bar Upgrade

There was a persistent issue with the previously implemented power distribution system. This was due to the lack of proper sizing and material usage on the bus-bars. This led to the insecure attachment of the power input to the bus bars, as well as the insecure attachment of the ESCs to the bus bars. This presented as the melting of the tray material, PETG. The failing system was replaced with adequately sized zinc-plated copper bus bars, which are insulated with glass-filled resin to provide a greater thermal barrier between the bus bars and the mounting tray. While remodeling the bus-bars the wires feeding them were also replaced with more flexible and appropriately gauged cable.

2) Remora

For our second AUV we did a full rebuild of its electrical systems. We made a plan to use a similar system to our main AUV so that it would act as a learning opportunity for younger members. We upgraded every component from the IMU to the bus bars using many of the same parts found on the main AUV. Many components had to be chosen based on size do to the struggle of confining everything into a singular small tube. We also made sure to account for parts that provided necessary computing power for behavior tree implementation.

C. Software Systems

This year's software design strategy was to optimize the existing systems in our AUVs, such as vehicle navigation, stereo vision, and computer vision. To support this objective, the system architecture was developed using the Robot Operating System 2 (ROS2), an industry-standard robotics middleware framework that enables centralized and efficient communication between distributed software components within an AUV.

Unlike previous implementations, the current system architecture reduces the reliance on dead reckoning for vehicle navigation. Instead, a behavior tree-based mission framework was implemented to provide greater modularity, adaptability, determinism, and interoperability with other onboard subsystems, including computer vision pipelines and payload deployment mechanisms such as dropper systems.

1) Behavior Tree and Autonomy

This year, autonomous navigation capabilities were fully integrated into the AUV control architecture through the implementation of a hierarchical finite-state framework based on behavior trees. The behavior tree architecture was developed using the BehaviorTree.CPP library [2], enabling modular task execution, improved decision-making, and simplified integration with other onboard subsystems. This library is compatible with a GUI interface, Groot2, which reduces the complexity of integration, allowing members of all experience levels to contribute to the development. This year, basic capabilities such as waypoint navigation, both absolute and relative, were added along with computer vision nodes capable of utilizing the aforementioned relative waypoint navigation. When arranged correctly, these nodes lead to deterministic behavior, allowing for the safe control and operations of the AUV.

2) Computer Vision

The vision system utilizes a combination of YOLO-based object detection and feature-based keypoint matching techniques. YOLO provides robust real-time object detection with high confidence and localization accuracy; however, these models are computationally intensive and require greater onboard processing resources. As a result, YOLO is primarily used for mission tasks that require reliable object classification and precise target identification

For less demanding vision-related tasks, a feature-matching approach is used to reduce computational overhead. Keypoint matching identifies and compares distinctive image features between reference and live camera frames to estimate object similarity or relative positioning. In this implementation, a Brute Force matching algorithm [3] was utilized to associate corresponding keypoints between images, as shown in Figure 3. Compared to YOLO-based detection, this approach is significantly simpler and more computationally efficient, though it generally provides lower robustness and reduced detection accuracy under varying environmental conditions.

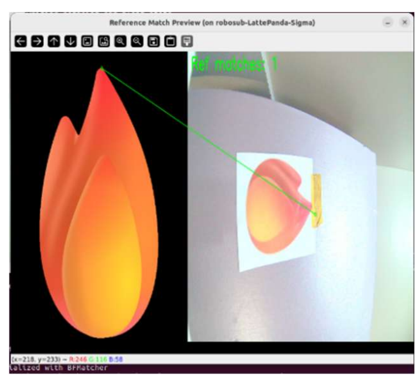

Figure 3 - Live Keypoint Matching Example demonstrating the identification of an image regardless of orientation

The use of feature-based matching also supports several mission-specific navigation and identification tasks, including gate traversal, bin localization, and slalom navigation. These tasks primarily require relative target positioning and consistent visual tracking rather than high-confidence object classification, making keypoint matching an effective solution for real-time operation. By integrating this approach into the overall vision pipeline, the system maintains sufficient task performance while preserving onboard computational resources for higher-priority autonomy and perception processes.

3) System Improvements and Ongoing Development

Additional development efforts this year focused on improving subsystem validation and hardware debugging capabilities. A dedicated suite of diagnostic and testing tools was developed to simplify verification of AUV thruster operation and overall propulsion system integration. These tools enable controlled arming and disarming of individual thrusters, independent thrust command testing, and sequential iteration through all eight thrusters to assist in identifying faulty hardware, configuration inconsistencies, or communication-related issues. The testing framework also supports thruster mapping and system-level debugging, improving maintainability and reducing troubleshooting time during integration and field-testing operations.

Additional quality-of-life and system reliability improvements were implemented through this development cycle. These improvements included parameter server–based PID gain modification for simplified controller tuning, internal pressure sensing and telemetry reporting, streamlined system initialization and ROS bagging procedures, and the automatic resetting of DVL localization states alongside the disabling of acoustic systems during shutdown operations. Collectively, these additions improved maintainability, reduced setup complexity, and enhanced operational consistency during testing and deployment.

Although numerous software and diagnostic improvements were successfully integrated, several ongoing technical concerns remain under investigation. The most significant issue currently affecting vehicle performance is the substantial drift observed within the IMU system, measured at greater than 10 deg/min under operating conditions. This drift rate is significantly outside the manufacturer's specified performance range and negatively impacts localization and navigational accuracy during autonomous operation.

Through comparative testing performed using the parent organization's remotely operated vehicle (ROV), Mako, the issue was determined to likely originate from insufficient hard-iron calibration under in-situ operating conditions. Experimental results obtained from the secondary platform demonstrated a reduction in drift from approximately 10 deg/min to nearly 1 deg/min following recalibration procedures. Based on these findings, additional calibration and sensor validation efforts are planned to improve navigational stability and overall autonomous performance for future deployments.

4) Future Improvements

The 2026 development cycle represented a significant advancement in the software architecture and autonomy capabilities of Team Unsinkable's AUV platforms. Despite these improvements, several major software and hardware upgrades are planned for future development iterations. One of the primary objectives is the integration of an NVIDIA Jetson AGX Orin onto the Hammerhead platform to provide substantially greater onboard computational capability for autonomy and perception tasks. This upgrade would support the deployment of more computationally demanding computer vision models, including larger and more accurate YOLO-based datasets, while also improving the team's ability to train, optimize, and deploy additional OpenCV-based vision pipelines.0

Future software development will also focus on expanding the behavior tree framework through the addition of new nodes, conditions, and modular task behaviors. Increasing the complexity and flexibility of the behavior tree architecture is expected to reduce reliance on handwritten mission scripts while improving scalability, maintainability, and autonomous task execution. Additionally, further development efforts are planned for the Remora platform to improve software functionality, subsystem integration, and overall operational reliability.

IV. Testing Strategy

Our testing strategy begins before departing the laboratory for the pool deck. Initial development for a new feature or behavior begins by creating a branch in our VCS, GitLab. The only requirement before merging with the development branch is successful compilation. The development branch is where experimental testing is performed on the benchtop, at the waterside, and in simulation. If possible, the feature is simulated; otherwise, it is tested by manually publishing inputs or by manipulating the environment (e.g., moving a picture) to cover all conditions and ensure that behaviors and inter-node communication are deterministic as expected.

Only after successful feature validation and benchtop validation can pool testing be considered. Before each pool test, a benchtop validation is performed, which includes spinning all thrusters, launching the master launch file, and performing a node tree analysis to ensure proper system startup. Finally, the AUV is brought to the pool for testing. We found that adhering to this process, while reducing time spent pool-testing, allowed for much earlier identification of issues, saved time traveling to and from the testing location for a fruitless test, and enabled timely reparations of errors.

The PIDs on Hammerhead have been tuned to the current configuration, which uses stainless steel ballasts to ensure the AUV remains level in the water and maintains only a slight positive buoyancy. Any significant changes to the internal structure, and thereby the weight distribution, include a pool test to ensure proper ballasting and control system response; if necessary, PID re-tuning is also performed during these tests. Similar to new-feature testing, benchtop validation of the entire AUV system is performed before departing for the pool.

Acknowledgment

Firstly, Team Unsinkable would like to highlight our faculty advisors at ERAU's College of Engineering, like Professor Daniel Penny, and Dr. Leo Ghelarducci, as the work of our team would not be possible without their support. We also thank the College of Engineering, William "Bill" Russo, and Kim May for access to the machine shop and guidance in designing. Furthermore, we thank Wanda Rodriguez from the Department of Electrical Engineering and Computer Science for her tremendous help with finances, funding, outreach, and her amazing attitude that makes the whole team smile. Additionally, we thank the Department of Electrical Engineering and Computer Science and the ERAU Fitness Center for granting us use of the Capstone Lab and the Fitness Center pool for both benchtop and underwater testing. Lastly, we are thankful for our generous sponsors: Infinity Center of Education, VectorNav, Ifixit, Aspen Utility, SendCutSend, PCBWay, and Quanta Telecom Solutions. Without their support, our team would not be in such a fortunate position. Team Unsinkable is excited to work with each one of our supporters to create an amazing working environment for future engineers to thrive in.

References

[1] "RoboSub 2026 | Team Handbook," Robosub, https://robonation.gitbook.io/robosub-resources (accessed May 27, 2026).

[2] "About | BehaviorTree.CPP," www.behaviortree.dev. https://www.behaviortree.dev/docs/Intro

[3] "Keypoint matching · Hugging Face," *Huggingface.co*, 2026. https://huggingface.co/docs/transformers/tasks/keypoint_matching (accessed May 26, 2026).

Appendix

I. SOFTWARE SYSTEMS

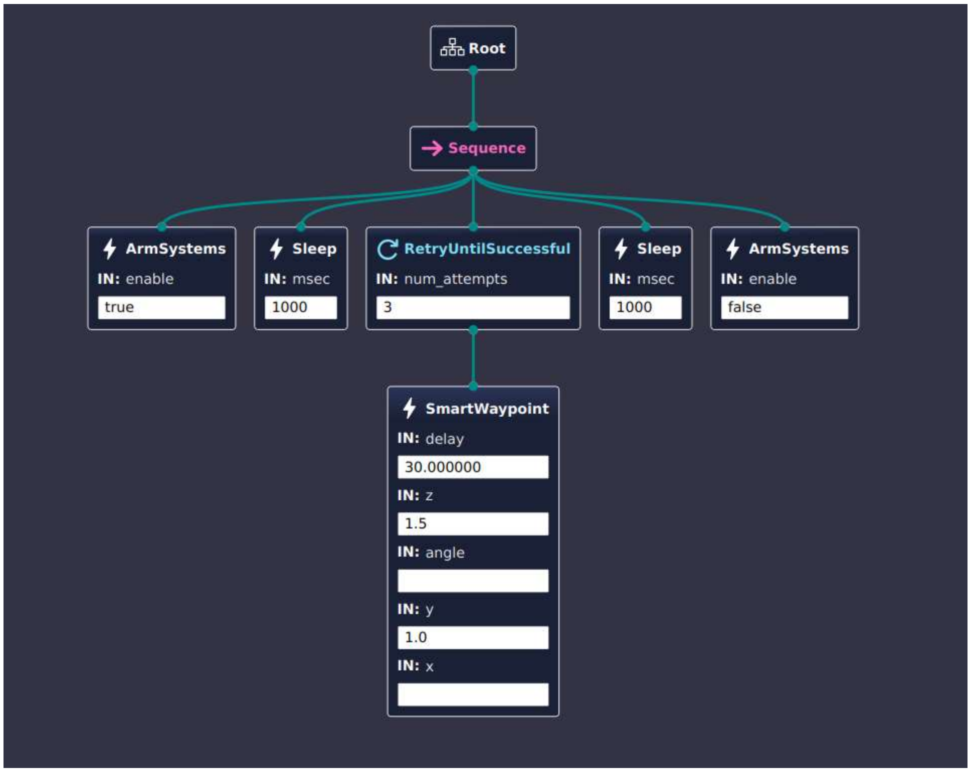

Figure 6: A sample diagram of a simple behavior tree system using Groot2.

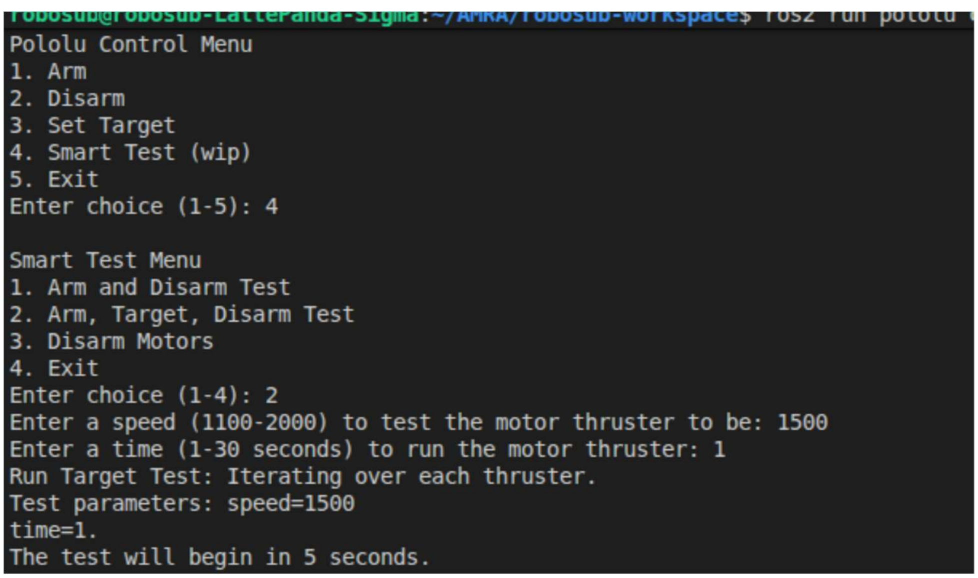

Figure 7: A capture of the command window of the thruster Smart Test script.

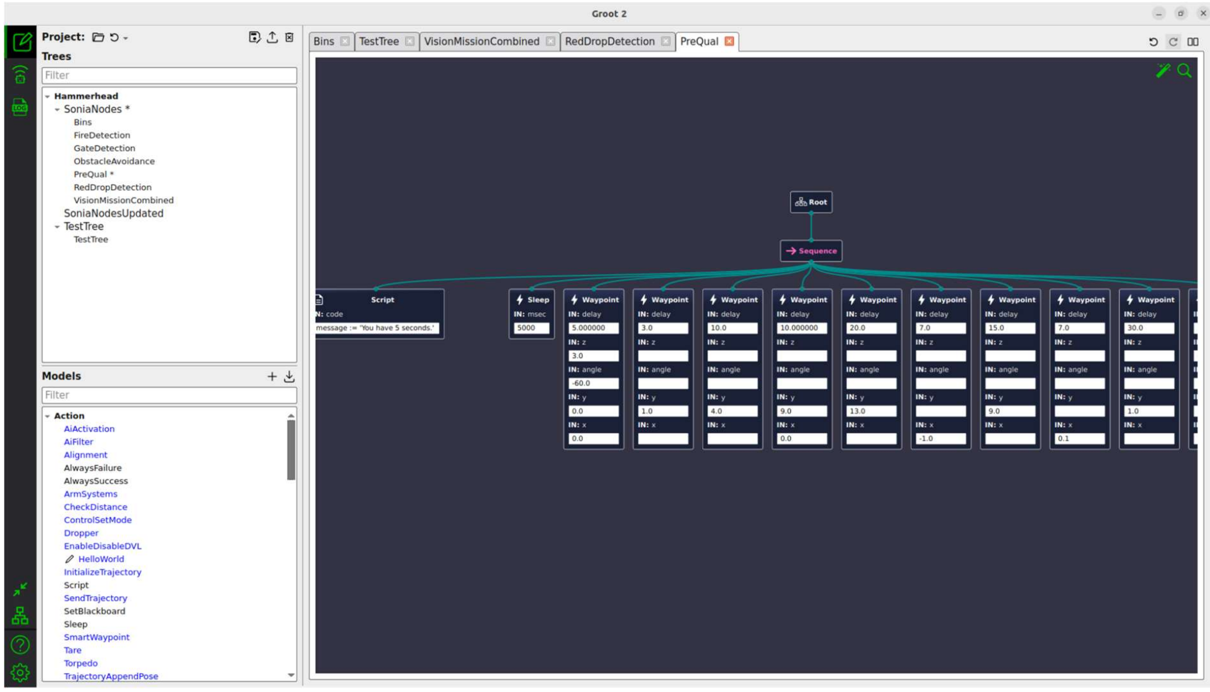

Figure 8: An image capture of the Groot2 Interface with a list of available nodes and actions. The current Behavior Tree system includes the current prequalification script converted over B.T. actions and nodes.

II. MECHANICAL SYSTEMS

Figure 9 - Remora Foot Design

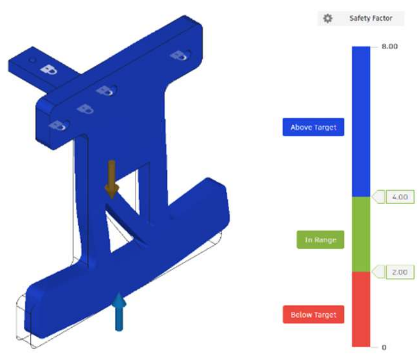

Figure 10 - Simulation of an Impact (50 N)

III. ELECTRICAL SYSTEMS

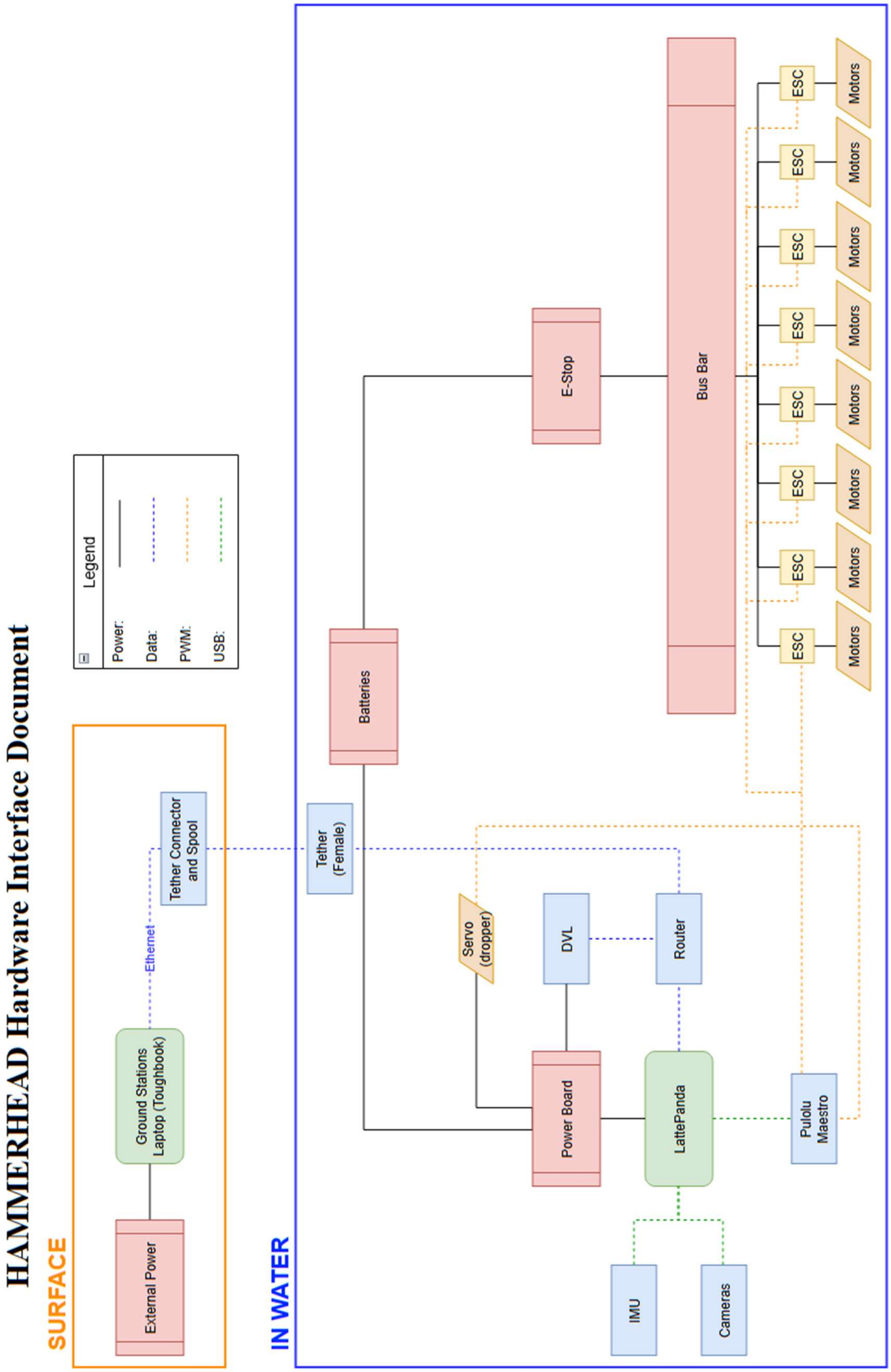

Figure 11: HammerHead Hardware Interface Design Document

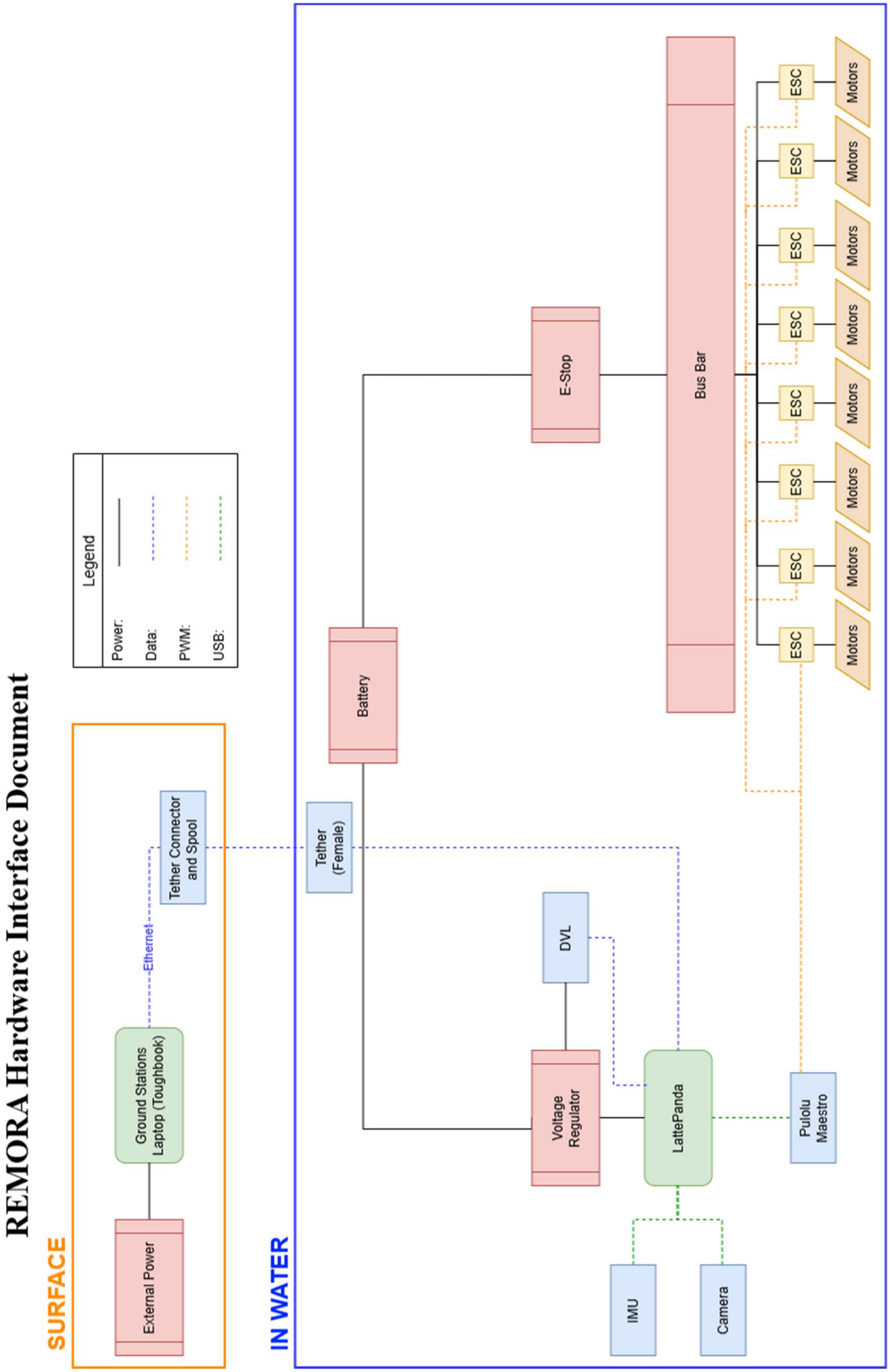

Figure 12: Remora Hardware Interface Design