RoboSub 2026 Technical Design Report

University of Southern California (Robosub SC)

Lauryn Hills, Ekamresh Vasudevan, Russell Rodriguez, Doris Lai, Huey Thompson, Joseph Ibalio, Takatoshi Soeda, Alejandro Arteaga, Arsalan Ghogari, Isabella Wu, Nathan Leung

***Abstract*—We present Barracuda 2.0, a significant evolution of our maritime platform featuring completely overhauled mechanical, electrical, and software architecture. Following our 2025 return to the competition, the team channeled lessons learned into a rigorous system-wide redesign. Key upgrades include a re-engineered pressure vessel and sealing mechanism, a consolidated electrical stack driven by a Jetson AGX Orin, and a robust ROS 2 Humble software pipeline. With these advancements, Barracuda 2.0 demonstrates our commitment to modularity, serviceability, and high-performance autonomy.**

I. Competition Strategy

A. Introduction

RoboSub SC has represented the University of Southern California in autonomous underwater robotics for nearly two decades and has consistently competed in RoboSub throughout that time. Following the 2023–2024 academic year, however, the team entered one of the most difficult periods in its history. Nearly all senior members graduated, the team lost access to its primary workspace, and several major funding sources disappeared. For a period of time, the future of the organization was uncertain. Instead of allowing the club to dissolve, the remaining members chose to rebuild the team from the ground up. During the 2024–2025 academic year, the executive board secured a new workspace in USC's Baum Family Maker Space, strengthened the team's relationship with the USC Viterbi School of Engineering, acquired new funding, expanded membership from fewer than five active members to more than twenty, and built an entirely new submersible in roughly six months. These efforts led to a successful appearance at RoboSub 2025, where the team tested new mechanical, electrical, and software systems in water for the first time.

After returning to competition in 2025, the team shifted its focus from basic vehicle operation to reliability, autonomy, and system integration. For the 2026 season, the team focused on building a more capable and dependable platform. Over the past year, the team upgraded major subsystems across the vehicle, including the transition to the NVIDIA Jetson AGX Orin as the primary onboard computer and the adoption of ROS 2 Humble as the standard middleware platform. The team also improved chassis accessibility, revised the onboard sensing stack, and refined autonomy systems through repeated pool tests and full-system integration trials. During Spring 2026 operations, the software team collected real-world telemetry and SLAM data to improve localization accuracy, navigation performance, and sensor fusion underwater. This report describes the major upgrades made to Barracuda 2.0 and outlines the engineering goals and competition strategy for RoboSub 2026.

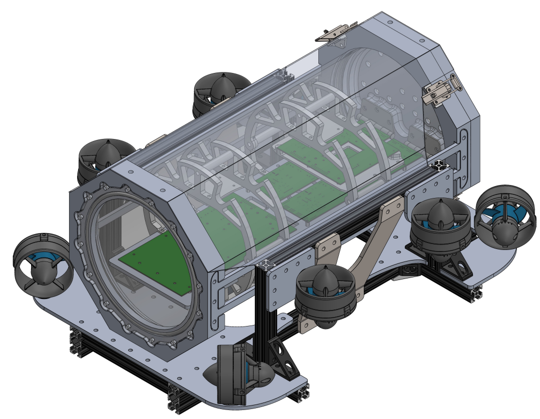

Fig. 1: Full CAD rendering of Barracuda 2.0 with revised pressure vessel and frame

B. Core Strategy

After a leak in the pressure vessel prematurely ended our sub's run last year, the mechanical team focused on a complete redesign of the main hull, focusing on ease of integration with the custom electrical stack and overhauling the sealing mechanism. Following intensive tests which validated Barracuda 2.0's mechanical redesign, our team outlined the following goals for the competition:

- Demonstrate pressure vessel sealing mechanism and decrease assembly time
- Simplify electrical stack integration and integrate peripheral functions
- Implement revised actuator system
- Build a strong foundational autonomy stack focusing on perception, localization and control

With these goals in mind, our team then developed an assessment of vehicle's ability to complete each of the tasks at Robosub.

C. Capability-to-Task Alignment

The tasks at Robosub require a complex mixture software, electrical and mechanical capabilities to be completed. Given the current competencies of our vehicle, we assign a confidence to each task based on pre-competition testing. This gives the team a realistic target for task completion, and we optimize our strategy to meet this.

Task	Capability Requirement			
	Actuation	Maneuverability	Vision	Confidence
Gate	No	Yes	Yes	100
Slalom	No	Yes	Yes	90
Bin	Yes	No	Yes	70
Acoustics	No	No	Yes	70
Octagon	Yes	No	Yes	70
Return Home	No	No	Yes	70

TABLE I: Per-Task Confidence

From the table above, we can clearly identify which tasks best match Barracuda 2.0's current strengths. The gate and slalom tasks are the team's highest-confidence objectives because they primarily rely on maneuverability, vision, and previously-tested autonomous control. The bin, acoustics, octagon, and return home tasks require greater integration between autonomous control, and the revised actuator system, which makes them higher-risk objectives. As a result, Barracuda 2.0's competition strategy prioritizes completing the gate and slalom before attempting more complex tasks that depend on actuation and advanced autonomy.

D. Risk Assessment

Our 2026 strategy is centered on minimizing downtime by identifying high-probability failure points within Barracuda 2.0. The following table details our technical risk profile and established mitigation protocols developed during our spring testing cycle (II).

1) Mechanical: Pressure Integrity & Sealing: Our primary mechanical risk involves the failure of the main hull seals during the mission. To mitigate this, we perform a standardized vacuum-test protocol before every pool session. The new modular latching system is designed for high-tension sealing, but the risk of O-ring degradation remains. We have implemented a "dry-dock tear down" phase after every wet test to inspect all wet-link penetrators and valve assemblies for moisture, ensuring the integrity of our pressure vessel.

2) Electrical: Bus Communication & Actuation: During our May 2nd full-system operations, we encountered anomalous noise on the I^{2}C bus, causing intermittent thruster communication failures. Our current technical risk management involves bus isolation strategies: we have mapped specific addresses (`0x2d`/`0x2e`) to identify faulty thruster indexes in real-time. Moving forward, we are implementing a watchdog timer in our firmware to force a safe-state reset if the thruster node heartbeat is lost for more than 50ms.

3) Software: State Estimation & Localization: Relying solely on simulated motion models proved insufficient for the level of localization precision required at RoboSub. Our primary software risk is estimator drift during inertial navigation, where small IMU integration errors in the body frame can accumulate rapidly when propagated into the world frame, especially under imperfect initialization, sensor bias, or orientation misalignment. To mitigate this risk, we shifted toward Real-World Telemetry Integration by collecting actual pool telemetry data, including IMU and doppler velocity logger (DVL) measurements into our SLAM Dataset Log. This allows us to replay real mission conditions offline, cross-reference estimated trajectories against verified mission logs, evaluate drift behavior, and tune our EKF-based state estimation pipeline without requiring continuous in-water testing.

4) Operational: Field Reliability: Our reliance on the Century Apartments pool for testing introduces a risk of variable signal interference. To maintain stable Topside UI connectivity, we have deployed local travel routers that bridge our AUV network to the campus infrastructure. This ensures that even if external Wi-Fi fluctuates, our internal ROS 2 communication bridge remains consistent, preventing "blind" missions during critical competition tasks.

II. Design Strategy

A. Mechanical Architecture

The mechanical design of Barracuda 2.0 represents a fundamental shift from the previous iteration, prioritizing modularity, ease of maintenance, and fluid dynamic efficiency. By eliminating the old pneumatic systems and completely overhauling the vehicle's structural foundation, the mechanical team significantly reduced the vehicle's size while expanding its functional capabilities. All components were modeled and drafted in SolidWorks before manufacturing.

1) Main Pressure Vessel and Sealing: To improve sealing and internal accessibility, the main pressure vessel was entirely redesigned. The central hull now utilizes a 10" OD acrylic tube, a substantial volume increase from last year's 8" OD housing. This increased internal volume provides ample space for the expanded electrical stack and facilitates rapid troubleshooting of onboard hardware. Furthermore, the end caps are now fully CNC-machined from 6061 aluminum, programmed using Mastercam to ensure precise tolerances.

A major focus of this year's overhaul was minimizing downtime between pool tests. The team implemented a new quick-release latching system that drastically reduces the time required to open the hull for battery swaps and electrical maintenance. This design reduced routine turnaround time from 30 minutes to just 30 seconds, virtually eliminating on-land bottlenecks and maximizing valuable in-water testing time.

2) Modular Frame and Propulsion: Barracuda 2.0 features a completely new external chassis designed for rapid iteration. Moving away from a rigid monolithic structure, the new frame is constructed utilizing several 80/20 aluminum extrusions and 1/4-inch-thick aluminum plates. This modular approach significantly reduces the vehicle's overall footprint and drag profile, while allowing the team to easily reposition thrusters, external sensors, and actuation mechanisms without machining new mounting hardware.

Taking advantage of the modular frame, the thruster placement was optimized to maximize fluid dynamic efficiency. By adjusting the thrust vectors and minimizing structural blockage in the propeller wash, the sub achieves superior maneuverability and a noticeably higher top speed of approximately 1.2 m/s, which is critical for completing timed navigation tasks.

TABLE II: Risk Assessment and Mitigation Strategy

Component	Risk	Probability	Impact	Mitigation Protocol
Pressure Vessel	Seal compromise (O-ring / valve)	Medium	High	Bi-weekly vacuum testing and post-immersion diagnostic checks
Electrical	I^2C/PWM communication glitch	High	Medium	Bus isolation, modular thruster addressing, and active terminal diagnostic monitoring
Software	State estimation drift (SLAM)	High	Medium	Real-world telemetry logging using rosbag for offline EKF calibration
Computing	Jetson / Raspberry Pi power throttling	Medium	Low	Active thermal monitoring and Docker-containerized resource allocation

B. Actuation and Mechanisms

The mechanism setup was completely overhauled this year to remove the pneumatic dependencies of Barracuda 1.0. All active mechanisms now utilize fully-electric servo actuation, which eliminates the need for bulky pressurized air storage and significantly reduces potential points of failure within the system.

A primary mechanism developed for this year is the torpedo launcher. Replacing the previous pneumatic firing system, the new preliminary design utilizes a servo in conjunction with a spring-loaded mechanism. When triggered, the servo actuates to open a hatch, allowing the compressed spring to release and launch the torpedo with consistent firing dynamics. While this design is currently in the preliminary stages, it represents a major step toward reliable, electrically driven payload deployment, and the final iterations of the housings will be 3D printed for rapid integration.

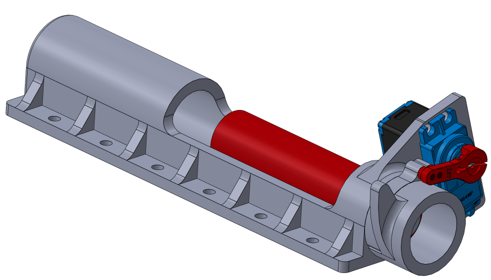

Fig. 2: CAD rendering of loaded torpedo launcher

Additionally, the team is conceptualizing a dropper mechanism designed to deploy two missile-shaped projectiles during the competition. Building on the architecture of the torpedo launcher, the dropper will utilize the same servos to actuate hatches and release the projectiles independently. Because this system is currently in the early design phase, the mechanical team is heavily prioritizing a modular mounting footprint. This ensures that once the CAD is finalized and motion-tested, the final printed mechanism can be seamlessly bolted onto the 80/20 frame without requiring chassis modifications.

C. Electrical Stack

After a successful implementation of the electrical stack at Robosub 2025, the electrical team sought to make further improvements to extend the functionality of the electronics. This included improving modularity, adding new peripheral boards, and pursuing extended testing of the electronics underwater.

1) Thruster Board: In order to control the 8 Blue Robotics T200 thrusters used to move the AUV, Barracuda 2.0 uses two identical thruster control boards. After receiving decoded force values from the Jetson AGX Orin, a Raspberry Pi transmits these values through the main I^2C bus to two Teensy 4.0s that sit on each of these boards. The Teensys then send a corresponding pulse width modulation (PWM) signal to any of the four Blue Robotics electronic speed controllers (ESCs) housed on each thruster board.

In the event of a necessary emergency shutdown, a hardware kill-switch system will immediately shut down power to the thrusters. Using an exterior magnetic latch and a Hall effect sensor, the sub is able to detect when the kill switch is pulled. With the kill switch activated, the board uses a high-side gate driver to shut down the thrusters. A 30A power MOSFET is switched on and off by a gate driver that is in turn controlled by the kill-switch. Through this implementation, Barracuda 2.0's thrusters receive a hard shutdown that is not dependent on any software - limiting the room for serious errors to occur.

Due to concerns related to large transient current spikes from the thrusters, both thruster boards receive isolated power from batteries (see electrical stack picture). This reduces the likelihood that the thrusters will degrade and interfere with other sensitive logic signal that are being transmitted (will change). In order to communicate with the other PCBs, the board use a set of opto-isolators to send digital information across different grounds.

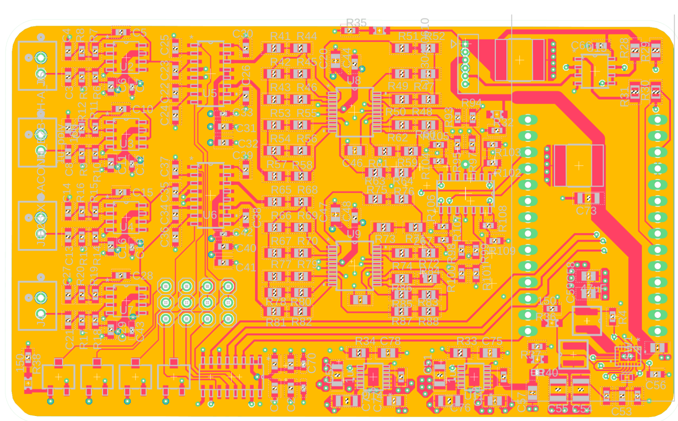

Fig. 3: Passive sonar board used to condition hydrophone signals for underwater acoustic localization

2) Passive Sonar Board: To navigate to tasks that are not visually noticeable, Barracuda 2.0 must make use of signals emitted by an underwater acoustic pinger. More specifically, the sub must determine the angle-of-arrival (AoA) of these acoustic signals using the formula

$$\theta_{\text{arrival}} = \arccos\left(\frac{\Phi\lambda}{2\pi d}\right) = \arccos\left(\frac{\Phi C}{2\pi d f}\right)$$

where λ is the wavelength, d is the distance between the hydrophones, C is the speed of sound in the pool, and f is the pinger frequency between 25 and 40 kHz.

The passive sonar equation estimates the received signal-to-noise ratio as

$$\text{SNR} = SL - NL + DI - PL,$$

where SL is the source level, NL is the noise level, DI is the directivity index, and $PL = 20\log_{10}(R)$ is the spherical spreading path loss for range R. Using this equation, we estimate that minimum amplitude signal we must receive to be around 200 μV_{rms}. Given the low amplitude of the input signal, we developed a passive sonar board (refer to Figure 3) to apply signal conditioning and then process these signals.

To this end, our analog front-end contains a pre-amp stage, a variable gain stage, a 4^{th}-order low-pass Bessel filter with a 70 kHz cutoff, and a level-shifting ADC driver stage. Through these stages, the board targets a 40 dB signal-to-noise ratio (SNR) at the input of the Arduino Portenta ADC.

To minimize noise injection from our power supply, a few iterations were made to the on-board power management system. The final design consists of a dual-channel synchronous step-down regulator, with each channel corresponding to the analog and digital supplies, respectively. The analog supply is further stepped down using two low-dropout regulators, one for 3.3 V and one for 5 V. This allows us to provide power to the sensitive analog components with less than 3 mV of ripple on the supply lines.

3) Leak Sensor Board: Following a critical leak that prematurely ended Barracuda 1.0's run at the previous competition, the team prioritized the development of stronger safeguards against water leaks. To address this failure, the electrical team implemented a dedicated leak detection and shutdown system for the pressure vessel. The leak sensor board uses polarized probes and a comparator circuit to detect the presence of water inside the hull. If water enters the pressure vessel, it shorts the probes and causes the comparator to output a logic-high signal, notifying the Jetson that a leak has occurred. The Jetson then initiates a software kill sequence to shut down sensitive electrical components before further damage can occur. To improve detection reliability and prevent false triggering from noisy or unstable signals, hysteresis is included in the comparator circuit. This system provides Barracuda 2.0 with a more reliable method for detecting leaks and taking preventative action during operation.

D. Software Stack

One of the largest software changes implemented for Barracuda 2.0 was the transition away from the more fragmented compute architecture used during RoboSub 2025 toward a more centralized and maintainable onboard system. While the previous vehicle utilized a Kubernetes cluster of containers with ROS 1 Noetic workspaces, the 2026 software stack was redesigned around a single container and ROS 2 Humble workspace to reduce Docker memory overhead and better support long-term development and tighter integration between different execution layers. The software stack is primarily centered on the NVIDIA Jetson AGX Orin, and is organized into modular ROS 2 Nodes (see Appendix 3) that are individually responsible for peripherals interfaces, perception, state estimation, mapping, navigation, control, mission planning, and telemetry.

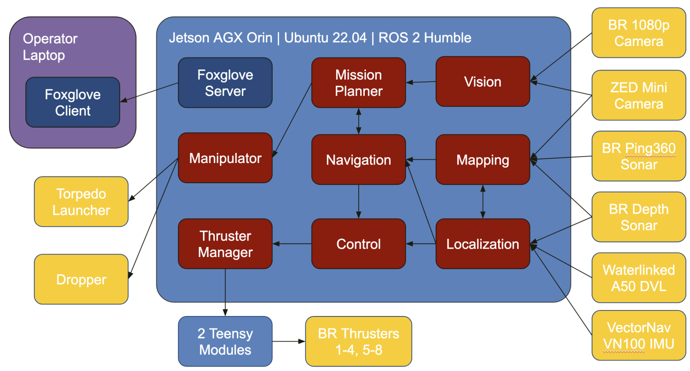

Fig. 4: Block diagram of the ROS 2 software and hardware pipeline on the Jetson AGX Orin and Operator Laptop.

In parallel with the ROS 2 migration, the software team has also worked on expanding the vehicle's onboard networking and telemetry infrastructure to support improved debugging, operator visibility, and subsystem monitoring during testing and competition operations. Current development efforts also focus on expanding vehicle mapping, planning, and control systems while improving integration between onboard sensing, autonomy, and actuation pipelines.

1) Onboard Compute and Networking: While the Jetson handles high-level autonomy computation, low-level hardware communication remains isolated through a Raspberry Pi 4B and two Teensy 4.0 micro-controllers responsible for interfacing with the vehicle's ESCs and peripheral devices. This separation allows the autonomy stack to remain insulated from lower-level actuator communication while improving fault tolerance during operation. Overall, the Teensy modules communicate directly with the Blue Robotics T200 ESCs over PWM channels while interfacing with the Jetson over I^2C bus lines.

The onboard networking (refer to Figure 10) was also expanded during the 2026 season following previous issues involving unreliable subsystem communication and limited telemetry visibility. The Barracuda 2.0 AUV now utilizes an internal Ethernet-based network architecture that connects major onboard devices, including the WaterLinked A50 DVL and auxiliary compute hardware, through an onboard switch mounted on. A Blue Robotics Fathom-X tether interface additionally made it possible for operators to monitor telemetry, inspect ROS topics, and debug autonomy behavior in real time during testing operations.

2) Perception and Sensor Integration: The perception stack integrates data from stereo vision, downward-facing vision, sonar systems, inertial sensing, and velocity estimation sensors to provide environmental awareness and task detection capabilities.

Barracuda 2.0 utilizes a ZED Mini stereo camera for depth perception and spatial awareness alongside a Blue Robotics low-light HD USB camera for downward-facing task alignment and object detection. Acoustic sensing is provided through a Ping360 scanning sonar and Ping sonar altimeter, while inertial and velocity measurements are supplied by the VectorNav VN-100 IMU and WaterLinked A50 DVL.

Perception nodes process incoming sensor streams independently before publishing standardized outputs to the remainder of the autonomy stack. This modular design improves debugging, fault isolation, and development iteration speed during integration testing.

Current software development has focused on improving underwater object detection robustness through curated training datasets, synthetic data augmentation workflows, and depth-assisted object localization. Planned improvements include semantic mapping integration, dynamic object tracking using EKF-based filtering, and gravity-aligned image pre-processing to improve visual consistency underwater.

3) Localization and State Estimation: Localization is handled through a multi-sensor state estimation pipeline that fuses inertial, depth, and velocity measurements into a unified estimate of the vehicle's pose and velocity.

The navigation pipeline combines stereo vision, IMU data, DVL measurements, and depth sensing before passing these measurements into a factor graph optimization framework using GTSAM. The resulting optimized odometry estimates are propagated through the vehicle's TF tree and used by downstream mapping, planning, and control systems.

A major focus of the 2026 software redesign has been improving localization robustness in real underwater environments. During spring testing operations, the team began collecting real-world telemetry logs to analyze estimator drift, sensor synchronization issues, and navigation accuracy. These logs are replayed offline to evaluate localization performance and tune estimation parameters under realistic operating conditions.

This telemetry-driven workflow allows the software team to validate navigation behavior without requiring continuous in-water testing while significantly improving repeatability between test sessions.

4) Control System: The low-level thruster pipeline includes force-to-RPM conversion, per-thruster PID control, ESC telemetry feedback, and additional protection systems such as anti-windup compensation, rate limiting, and power derating safeguards. This layered approach separates high-level vehicle dynamics from low-level motor control, improving modularity and simplifying future hardware revisions.

5) Mission Planner: Current mission planning research focuses on evaluating both Behavior Trees and Hierarchical State Machines as scalable frameworks for autonomous task execution. These approaches improve readability, modularity, and debugging while simplifying the scheduling and monitoring of complex autonomous behaviors. This modular mission architecture also improves reliability during testing, since individual actions can be validated independently before being integrated into full mission sequences.

6) Simulation and Development Infrastructure: Simulation plays a major role in Barracuda 2.0's software development and validation workflow. The software team is developing a high-fidelity ROS 2 simulation environment using NVIDIA Isaac Sim and Ocean Sim to validate perception, mapping, planning, and control systems before deployment onto physical hardware.

The simulation environment integrates sensor models, thruster dynamics, and underwater environmental effects to improve sim-to-real transfer performance. Planned future improvements include Unity-based underwater rendering systems capable of simulating turbidity, lighting attenuation, and particle scattering for more realistic perception training.

Development infrastructure is built around GitHub repositories, containerized ROS 2 workspaces, and shared networked development environments. This architecture allows team members to collaborate on autonomy modules independently while maintaining reproducible deployments across testing systems and onboard hardware.

7) Topside Telemetry and Operator Interface: Barracuda 2.0 includes a topside telemetry and visualization system designed to provide operators with real-time insight into vehicle state, autonomy behavior, and subsystem health during testing and competition operations.

Telemetry tools provide live camera feeds, point cloud visualization, object detection overlays, trajectory monitoring, thruster telemetry, battery information, and manual control interfaces. Current development includes both lightweight web-based dashboards and more advanced Unity-based operator interfaces capable of displaying immersive 3D vehicle and environment visualizations.

Foxglove is used extensively for debugging and telemetry visualization during pool operations, allowing software developers to inspect ROS topics, localization outputs, and system behavior in real time.

The telemetry pipeline also supports extensive mission logging for offline replay and debugging, allowing developers to analyze failures, evaluate navigation performance, and improve autonomy pipelines without requiring repeated in-water testing.

III. Testing Strategy

Our validation pipeline follows a hierarchical approach, ensuring that every subsystem is robust before full-system integration.

A. Compartmentalized Electrical Testing

To isolate failure modes, we subject individual electrical modules to bench-top validation before installation into the hull. This involves verifying power distribution stability, checking logic level compatibility for sensor interfaces, and validating communication protocols (I^2C/PWM) with localized dummy loads. By verifying electrical integrity at the component level, we prevent transient faults occurring during main operations.

B. Integration Testing

Once components are cleared, we proceed to hardware-in-the-loop (HITL) integration. This phase focuses on the interaction between the Jetson AGX Orin and peripherals, ensuring that sensor data harvesting (DVL, Sonar, IMU) and actuator command signals operate within latency constraints. We utilize iterative dry-run sessions to monitor bus health and confirm that electrical noise does not interfere with critical telemetry transmission.

C. Software Testing and Validation

Our software validation relies on a dual-track approach: offline simulation and real-world telemetry replay. We use high-fidelity simulation to test mission logic and planner performance in a virtual environment. This is followed by validating our localization and vision stacks against real-world pool datasets, ensuring our EKF state estimation remains bounded. Before wet testing, we run a thruster initialization sequence to visually verify correct thruster mapping, direction, and responsiveness. During live pool sessions, we also support real-time human debugging through controller override, allowing operators to intervene, inspect behavior, and safely validate autonomy decisions under dynamic conditions. Final validation is achieved through iterative wet tests that verify vehicle response and mission task completion.

D. Mechanical Leak Testing

Catastrophic leaks present the highest risk to the vehicle's custom electrical stack. To mitigate this, the team enforces a vacuum leak test protocol before any water operations. An external pump is used to draw the internal pressure of the 10" acrylic hull down to a vacuum of 15 inHg. The valve is then sealed, and the pressure vessel is left to sit for a 15-minute hold period. If the pressure gauge indicates any loss of vacuum during this window, we know that there is a leak, and we can reinspect all seals and try again.

Acknowledgment

RoboSub SC would like to thank the USC Viterbi School of Engineering, the Baum Family Maker Space, Professor Laurent Itti, Kevin Giang, and the many mentors, sponsors, and students at the University of Southern California who have supported the team throughout this year. Their guidance, resources, and continued belief in our organization have made it possible for RoboSub SC to rebuild, grow, and proudly represent USC at RoboSub 2026.

References

[1] F. Dellaert and GTSAM Contributors, *borglab/gtsam* (Version 4.2a8). Georgia Tech Borg Lab, May 2022. [Online]. Available: https://github.com/borglab/gtsam. DOI: 10.5281/zenodo.5794541.

[2] NVIDIA, *Isaac Sim* (Version 5.1.0). [Online]. Available: https://github.com/isaac-sim/IsaacSim.

[3] J. Song, H. Ma, O. Bagoren, A. V. Sethuraman, Y. Zhang, and K. A. Skinner, "OceanSim: A GPU-Accelerated Underwater Robot Perception Simulation Framework," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

APPENDIX A
SPONSORS

Fig. 5: USC AUV Sponsorship Packet

USC RoboSub SC would like to thank the following organizations for their continued support of the team and the development of Barracuda 2.0.

A. Trojan Sponsors

USC Viterbi School of Engineering

B. Gold Sponsors

Lockheed Martin

Northrop Grumman

Texas Instruments

APPENDIX B
ELECTRICAL STACK DIAGRAM

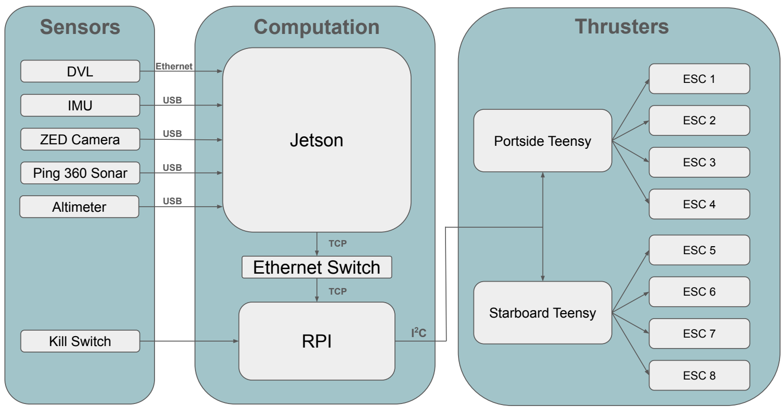

Fig. 6: High-level onboard electrical and compute architecture of Barracuda 2.0

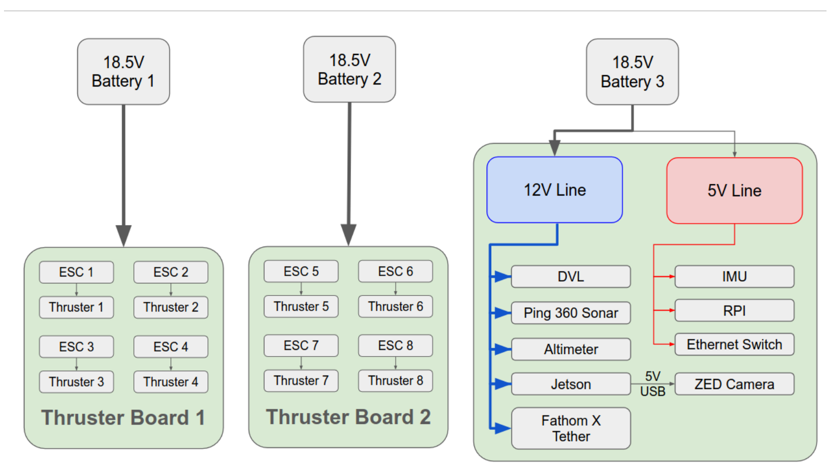

Fig. 7: Power diagram of Barracuda 2.0

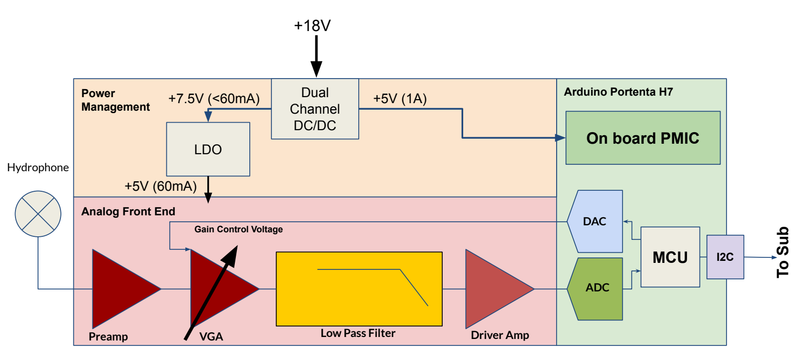

Fig. 8: Sonar processing and integration pipeline for Barracuda 2.0

Appendix C
Software Architecture

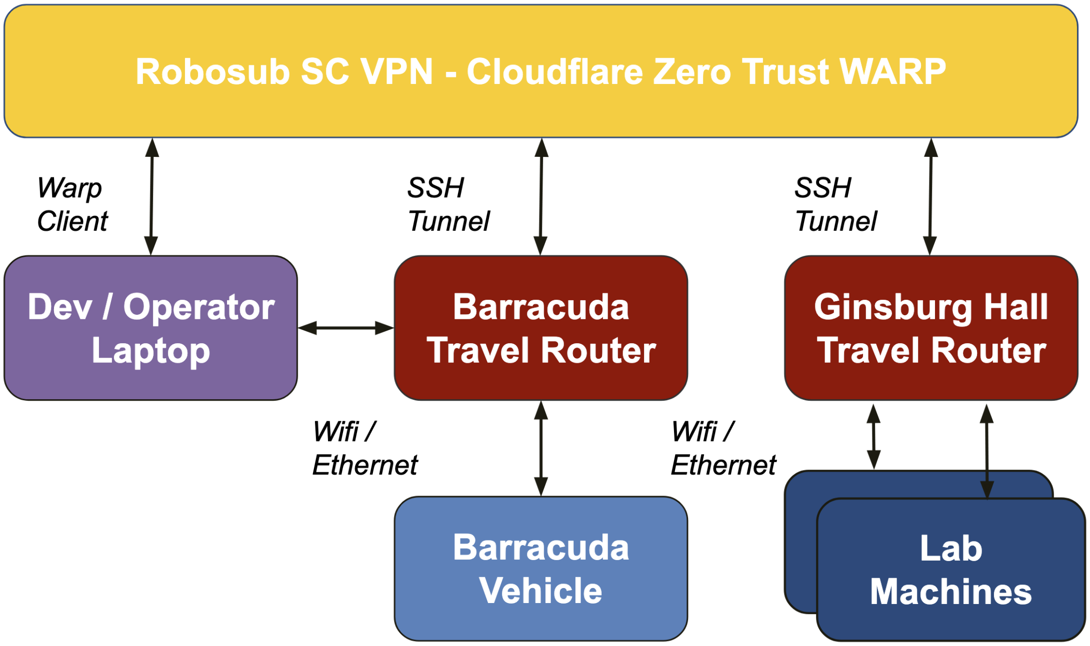

Fig. 9: **Network Infrastructure.** Topology diagram of the cross-site virtual private network utilizing Cloudflare Zero Trust WARP. The architecture facilitates secure remote development and data collection by bridging operator laptops with the Barracuda vehicle and Ginsburg Hall lab machines via dedicated travel routers and SSH tunnels.

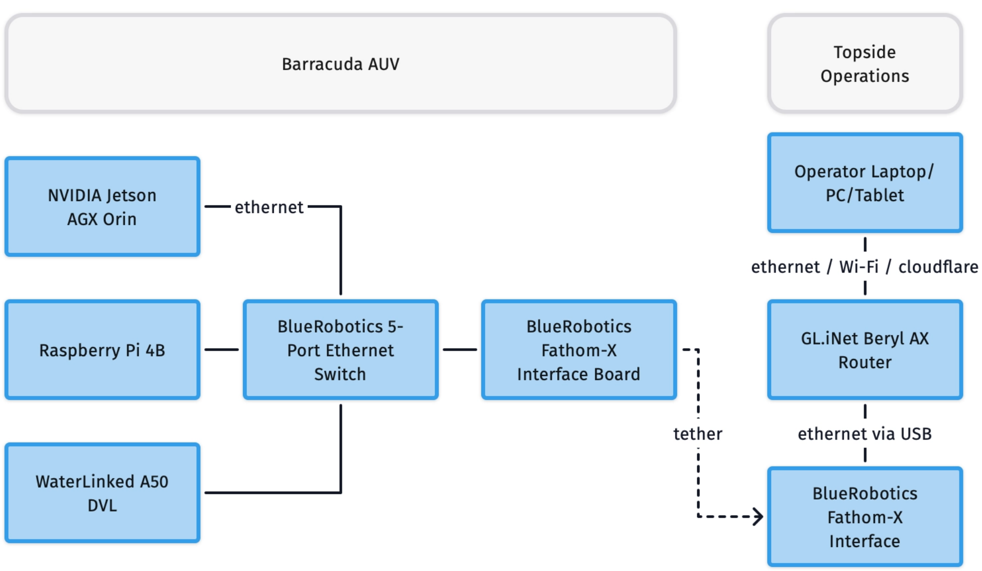

Fig. 10: **Vehicle Area Network.** Topology of the Barracuda AUV's internal and tethered network. Onboard compute and sensor nodes (NVIDIA Jetson AGX Orin, Raspberry Pi, and DVL) connect via a 5-port ethernet switch to a Fathom-X interface. This bridges communications over the physical tether to topside operations, routing operator traffic through a GL.iNet travel router.

APPENDIX D
MECHANICAL INFRASTRUCTURE

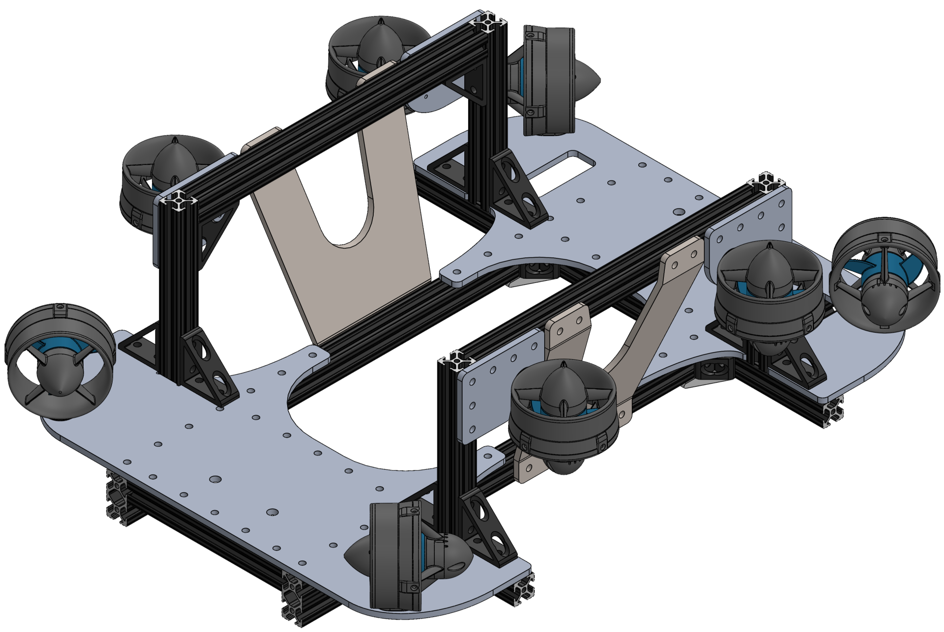

Fig. 11: CAD model of the frame assembly using 80/20 extrusions

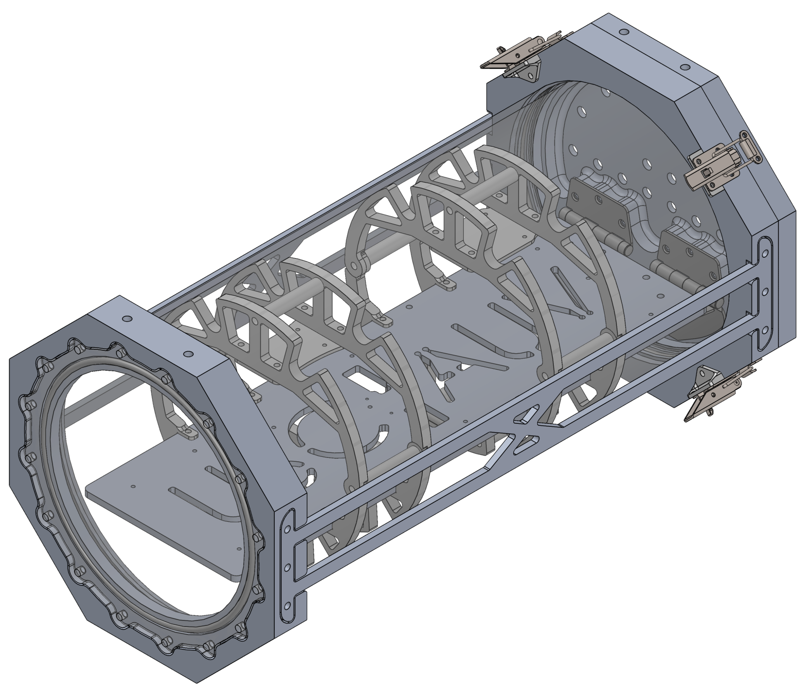

Fig. 12: CAD model of the main pressure vessel

Appendix E
Specifications

TABLE I: Key Vehicle Specifications

Parameter	**Value**
Dimensions (L×W×H)	26 in × 20 in × 15 in (approx.)
Dry/Submerged Weight	35 kg

TABLE II: Sensor Specifications

Sensor	**Category**	**Specifications / Notes**
ZED Mini Stereo Camera	Vision	1280×720 stereo depth imaging for visual SLAM and perception
IMU	Inertial	Orientation, acceleration, and motion estimation
DVL-A50	Navigation	Underwater velocity estimation and localization
Ping360 Sonar	Acoustic	Obstacle detection, environmental mapping, and target localization
Ping Sonar Altimeter	Acoustic	Altitude and depth estimation for navigation support

TABLE III: Embedded Compute and Networking Hardware

Device	**Vendor**
Raspberry Pi 4B	Raspberry Pi Foundation
Jetson AGX Orin	NVIDIA
Teensy 4.0	PJRC
5-port Ethernet Switch	Blue Robotics

APPENDIX F
HIGH SCHOOL COLLABORATIONS

Fig. 13: Team members talking with the students

Fig. 14: Team members and students operating and troubleshooting their FTC robot

At the conclusion of the 2025 Fall semester, Robosub SC took a trip to Crossroads School in Santa Monica. There, our team worked the school's FTC team and advised middle school students on avenues forward to bolster their robotic systems.