RoboSub 2026 Technical Design Report

Mitchell Turton, Kieran Abbott, Erin Beazley, John Beeson, Jason Hsiao
Shrey Patel, Em Paul, Armando Rolon-Loperena, Matthew Woodward, Aaron Wu
Marine Robotics Group at Georgia Tech
Atlanta, Georgia

***Abstract*—The Marine Robotics Group at Georgia Tech is entering RoboSub 2026 with an overarching focus on reliable, foundational autonomy. This paper outlines the scope of our 2026 development, detailing the direct linkage of our competition priorities with our system architecture, design, and engineering decisions. Our competition strategy deliberately prioritizes the development of reliable autonomous navigation rather than specialized subsystems for environmental interaction. We focus on developing *shared capabilities*–behavioral primitives such as depth and heading control, waypoint following, and station keeping–because they support the largest set of selected tasks while generating interoperable perception and autonomy packages for future vehicles. Our design strategy realizes this priority with targeted system-level hardware and software upgrades: a more serviceable electrical layout, a reusable motor interface board, structured firmware communication, and shared approaches to perception and mapping that foster collaboration with our surface vehicle team. Our testing strategy uses simulation, competition-element mocks, bench tests, dry tests, and staged pool deployments. Together, these choices focus the 2026 vehicle on capability growth that remains observable, serviceable, and reliable.**

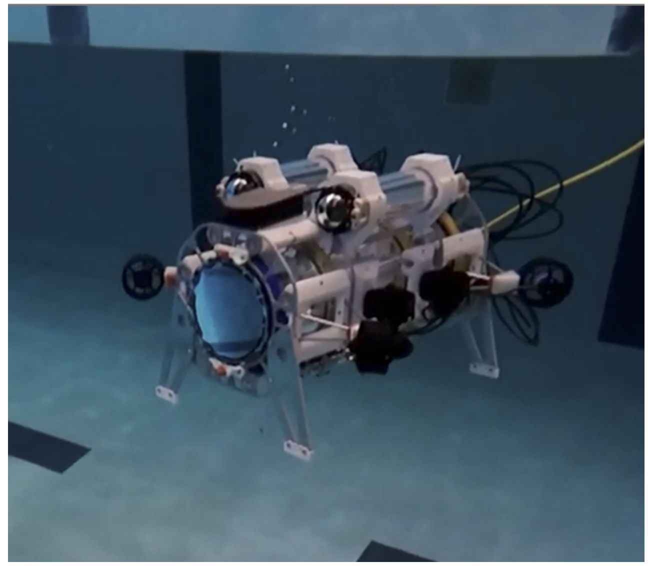

Fig. 1: Georgia Tech AUV during pool testing.

I. Competition Strategy

The 2026 strategy for the Marine Robotics Group at Georgia Tech (MRG) is to build an autonomous platform capable of making an actionable semantic representation of its environment while also reducing complexity in the software stack and maximizing hardware reliability. In order to achieve this, MRG focused on improving the capabilities that are reused across many RoboSub tasks. Rather than treating each task as a separate hardware project, the team uses the task list to identify common system requirements. By prioritizing tasks that strengthen these shared requirements, we mitigate the risk of subsystem propagation and actively manage the tradeoff between adding new complexity and improving existing reliability.

A. Strategic Vision

The 2025 vehicle established the team's baseline competition platform: a modular X-drive AUV with ROS 2 software [1], camera perception, DVL/IMU localization, simulation support, basic task autonomy, and a serviceable mechanical layout [2]. For 2026, the team is synthesizing those elements into a stable AUV architecture for robust navigation and future cross-vehicle interoperability in mind.

This approach is driven by the structure of the course itself [3]. The highest-yield engineering investments are those that enhance multiple tasks simultaneously: robust depth and heading control, waypoint navigation, reliable search logic, object perception, and station keeping. By perfecting these shared capabilities, the vehicle secures a highly competitive navigation score while establishing a proven baseline for future manipulation hardware.

The team's testing history also shaped the strategy. During previous integration and competition work, reliability problems were often caused by the interaction of subsystems rather than by a single missing feature: late software changes created rebuild risk, tether degradation forced internal reconfiguration, and a fragile reed-switch emergency stop motivated a more robust E-stop design. The 2026 strategy responds constructively by making new capability additions easier to test, easier to isolate, and easier to recover from when faults occur.

B. Task Selection and Capability Alignment

Before addressing specific tasks, we rigorously filter potential vehicle capabilities against this multi-year competitive philosophy. Table I scores core robotics behaviors based on their immediate contribution to foundational awareness, their scalability to future development cycles, and their field testability (the ability to validate systems offline via simulation or dry-bench testing).

This evaluation mathematically bounds our engineering scope. Shared navigation and perception behaviors form our targeted foundation because they yield massive systemic value

TABLE I: Strategic Capability Evaluation. Behaviors are scored (1–5) on their immediate value to foundational awareness/navigation and their scalability to future multi-vehicle and advanced manipulation goals.

Core Behavior	Found. Nav & Aware.	Future Scalability	Field Testability	Total Score
Depth & Heading Control	5	5	5	**15**
Waypoint Following	5	5	4	**14**
Object Perception	5	4	4	**13**
Station Keeping	5	5	3	**13**
Semantic Mapping	4	5	3	**12**
Manipulation HW	1	2	1	**4**

for achieving a winning score. Conversely, isolated manipulation hardware scores exceptionally low; it offers minimal inherent navigation value, risks creating orphan subsystems, and is notoriously difficult to test outside of limited pool hours.

The final step is to reconcile this established foundation with task competitiveness. We approach the Autonomy Challenge with a strict heuristic: target tasks that maximize scoring using only our established navigation baseline, and explicitly exclude tasks that require high-risk, hull-specific mechanisms. Table II illustrates this process.

TABLE II: Task Competitiveness Matrix. Tasks are targeted if they maximize competitive scoring using only our established navigation foundation, minimizing the need for high-risk, isolated mechanical development.

Task Area	Uses Core Nav/Aware?	Added Mechanical Complexity	Target?
Gate	Yes	None	**Yes**
Slalom	Yes	None	**Yes**
Oct. (Surface)	Yes	None	**Yes**
Bins	Yes	Gravity-fed dropper (Low risk)	**Yes**
Torpedoes	Yes	Actuated pneumatics (High risk)	**No**
Oct. (Resupply)	Yes	Active manipulation (High risk)	**No**

As shown in Table II, our design logic proves highly effective. The first three tasks can be completed entirely by our robust navigation foundation, requiring zero custom mechanical engineering. This makes task selection a systems decision: complete the core navigation sequence reliably, add one low-risk manipulation action (Bins), and explicitly defer high-risk mechanical actions (Torpedoes, Resupply) to future years to guarantee a competitive score ceiling for the current platform.

C. Managing Capability Growth and Reliability

The central tradeoff of the Autonomy Challenge is capability growth versus reliability. While adding more mechanisms, sensors, and firmware states increases the theoretical ceiling of what the vehicle can attempt, every new component introduces compounding failure modes that threaten the core navigation baseline. The team's mitigation is to add complexity only when it either supports several tasks or can be validated independently before a full run. New systems are expected to be observable, serviceable, and testable outside the pool whenever possible.

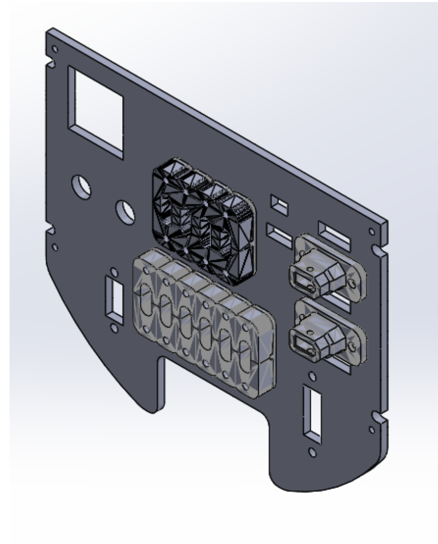

Fig. 2: CAD of electrical backplate organizer. Assembly is placed between the backplate and interior drawer for stable connection and organization.

This principle guides both task selection and operating procedures. The navigation tasks share a common capability base, so improvements to perception, mapping, controls, autonomy, and serviceability improve several task attempts at once. Additionally, the bin task provides a simple starting point to develop manipulation capability with a mechanically simple design while still developing software features –target approach, station keeping, alignment, and timed actuation– which will transfer well to the more difficult manipulation tasks. Procedure is treated as part of reliability as well: reviewed code, staged validation, checklists, and known-working pool configurations reduce the chance that limited water time becomes a rebuild or debugging session.

II. Design Strategy

The 2026 design strategy implements the competition strategy as a set of system-level requirements. The team identified reliability, serviceability, reusable capability growth, and staged testability as the main design priorities. Those priorities drove requirements for stable motion, accessible electrical integration, protected power distribution, robust emergency-stop behavior, low-ambiguity manipulation, modular firmware, and simulation-compatible autonomy.

A. Reliability and Serviceability Architecture

The vehicle retains the proven 24-inch acrylic main tube, aluminum end plates, guide rods, and X-drive thruster layout from the 2025 platform, but the 2026 design work focuses on making that platform easier to service and more tolerant of integration growth. The symmetric layout keeps the center of buoyancy above the center of gravity for passive roll and pitch stability, while the four horizontal X-drive thrusters and four vertical thrusters provide the control authority needed

TABLE III: Traceability from strategic priorities to 2026 design decisions.

Strategic Priority	Requirement	2026 Design Choice	Reliability Risk Reduced
Reliable navigation base	Stable motion and repeatable waypoints	Symmetric vehicle layout, X-drive configuration, DVL/IMU localization, position and velocity control modes	Failed task attempts due to unstable motion or poor control authority
Efficient pool use	Fast access to electrical interfaces and board-level validation	Electronics drawer, fixed backplate organizer, panel-mount connections, bench-tested distribution boards	Lost pool time from inaccessible connectors or wiring faults
Safe high-current operation	Protected thruster power with robust shutoff	Separated thruster/compute power systems, fused ESC distribution, 200 A relay, hall-effect E-stop	Electronics damage, unsafe actuation, and fragile E-stop behavior
Low-ambiguity manipulation	One manipulation path that isolates software and controls performance	Simple two-chamber dropper and negatively buoyant markers	Confusing mechanical unreliability with perception, control, or autonomy failures
Reusable embedded layer	Board behavior must be predictable across MRG platforms	Presto motor interface board, PlatformIO targets, Protocol Buffers serial bridge	One-off firmware, OS-specific integration and inconsistent board behavior
Simulation-first autonomy	Task logic must be testable before pool deployment	Modular ROS 2 task behaviors, Gazebo course mocks, shared autonomy and control code	Basic mission bugs discovered only during limited water tests

for the navigation and station-keeping capabilities identified in Table II.

Serviceability was treated as a reliability requirement, not only as a convenience. Internal electronics are mounted on drawer slides so that boards and harnesses can be accessed without removing the entire electronics stack. A fixed acrylic backplate organizer provides labeled connection points between the rear penetrator plate and the electronics drawer. This reduces strain on cables when the enclosure is opened and gives the team a stable interface when diagnosing faults. The previous tether degradation experience showed why this matters: when external communication faults occur, internal access must be fast and repeatable so that repairs do not introduce additional wiring errors.

Printed attachments are designed to avoid carrying trapped air. Low-infill PETG parts with gyroid-like internal paths are drilled at high and low points so water can enter while trapped air escapes, reducing unwanted positive buoyancy and improving repeatability between dry handling and submerged operation.

B. Power Distribution and Safety

The complete power system is divided into two common-ground circuits: thruster power system and computation power system. The thruster power system contains the high-current elements– emergency stop, relay, power distribution PCBs, fuses, ESCs, and motors. The computation power system powers the Jetson Orin Nano, embedded microcontrollers, IMU, DVL, Sonoptix Echo imaging sonar, and low-light cameras. This separation protects voltage-sensitive electronics from the high and variable current draw of the thrusters while keeping both domains conceptually clear.

The 2026 thruster harness was redesigned around one 200 A automotive relay and three fused power-distribution PCBs instead of the previous relay-heavy harness. Each distribution board supplies two to three ESCs and provides board-mounted connectors, fuse protection, and power indication. This approach increases capability by supporting eight thrusters while reducing wiring complexity and making failures easier to localize. The main thruster line uses a 125 A ANL fuse, individual ESCs use 30 A blade fuses, and software limits normal thruster current to 120 A even though the absolute maximum motor current is higher.

TABLE IV: Manipulation tradeoff used to select the 2026 subsystem. Software transfer is higher-is-better; complexity and test burden are lower-is-better.

Option	Transfer	Mechanical Complexity	Test	Decision
Dropper / bin	4	1	2	Selected for low hardware ambiguity
Smart torpedo / target	4	4	4	Deferred until base alignment is reliable
Arm / resupply	5	5	5	Deferred due to coupled interaction failures

Emergency-stop reliability was also improved. The previous reed-switch based E-stop was replaced with a hall-effect sensing circuit that actuates a low-current relay, which in turn energizes the 200 A thruster relay in a cascaded architecture. The design uses a more robust sensing approach and can be validated during pre-test checks before the vehicle enters the water. The emergency stop can be easily magnetically actuated by a diver in-situ.

C. Manipulation Subsystem Selection

The team selected the dropper as the only primary 2026 manipulation subsystem because it best supports the broader strategy: develop the shared manipulation software base while minimizing new mechanical uncertainty. Torpedoes and an arm-style resupply mechanism were considered, and the team previously prototyped a smart torpedo. However, all manipulation tasks require the same hard software problem first: perceive the target, maneuver precisely, hold station, confirm alignment, and command the mechanism at the correct time. Choosing the simplest mechanism makes it easier to determine whether failed attempts are caused by perception, controls, autonomy, or hardware.

The resulting dropper is a two-chamber 3D printed release mechanism powered by a BlueTrail waterproof servo. A triangular door covers both slots in its neutral position and exposes one slot at a time as the servo rotates, allowing two independent drops with one actuator. The markers are

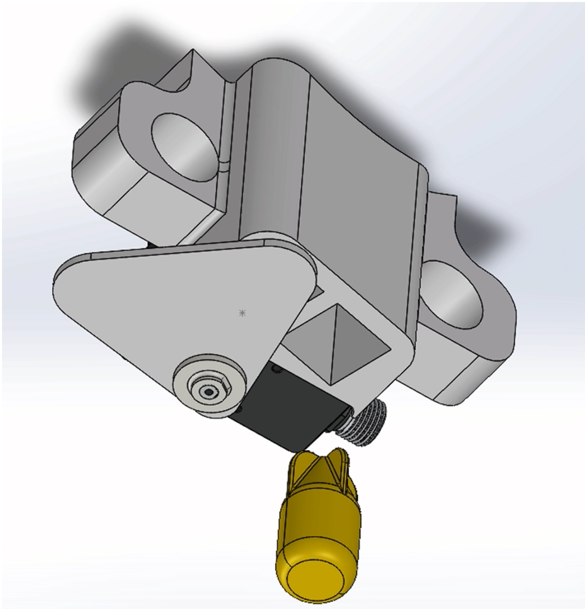

Fig. 3: Two-chamber dropper assembly with marker release path.

negatively buoyant and finned so they descend predictably after release [4].

D. Embedded Interface and Firmware

The embedded layer was redesigned around reusable, observable interfaces [5]. A custom motor interface board provides Raspberry Pi Pico based actuation, ten servo outputs, E-stop sensing/actuation support, status LEDs, and radio compatibility. The design is intentionally interoperable across projects within the MRG development ecosystem, creatively allowing for board reuse for many platforms with similar actuation needs. On the AUV, the board is configured for eight thruster outputs, dropper control, safety status, command reception, and telemetry return. The board can be validated once and reused across platforms, reducing the number of new electrical and firmware variables introduced during AUV integration.

Firmware is organized as a product-line architecture: one shared codebase contains common modules for motors, safety signals, status reporting, and host communication, while PlatformIO build targets select the features needed by each board [5], [6]. Host communication uses Protocol Buffers over a framed serial link instead of micro-ROS on the microcontroller. This keeps the firmware independent of the host operating system and ROS middleware while allowing the Jetson driver to translate between ROS 2 messages and structured serial commands. Telemetry, actuator state, safety status, and debug information share the same link, allowing the team to quickly debug issues during testing.

E. Autonomy, Perception, Mapping, and Controls

The Pontus ROS 2 stack is the area of largest 2026 progress. It connects perception, mapping, localization, autonomy, controls, simulation, and firmware communication through modular interfaces. Perception nodes produce sensor-specific observations, mapping converts those observations into task-level course objects, localization provides vehicle state from DVL/IMU/depth sensing, autonomy selects task behavior and waypoints, controls convert desired motion into body commands, and firmware sends commands to hardware. This separation keeps failures diagnosable and avoids a monolithic mission script where perception, control, and task-state errors are difficult to isolate.

The perception and semantic mapping pipeline was redesigned so autonomy does not react directly to raw frame-by-frame detections. Low-light cameras provide YOLO-style semantic detections [7], while the Sonoptix Echo sonar provides geometric pointcloud information for larger objects that may be difficult to classify visually. Sonar returns are processed as pointcloud-based object candidates using standard pointcloud filtering and clustering ideas [8]. Each observation is passed to the mapping layer with its label, confidence, and estimated position.

This semantic map gives autonomy a course-level representation instead of isolated detections. For the slalom, the system can reason about red and white pole candidates as a valid row structure rather than unrelated blobs. For the gate and return-home tasks, mapped gate elements give autonomy a more stable reference than a single image detection. For the bin task, the same perception-to-map interface supports target search, centering, station keeping, and timed dropper release. This pipeline directly supports the high object-perception and semantic-mapping weights in Table II and allows camera and sonar improvements to benefit several tasks at once.

Autonomy is organized around reusable task behaviors: hold depth, hold heading, move to a waypoint, search for an object, center on a detection, hold station, command an actuator, and exit or skip when a task is not reliable. These behaviors support the gate, slalom, octagon, return-home, and bin tasks identified in the capability matrix. The mapping layer provides task objects through a common interface, so each task can be improved without rewriting the full mission pipeline.

Controls were updated to support mixed position and velocity operation. For example, the vehicle can hold depth while accepting planar velocity commands, or hold position while waiting for perception. The controller stack contains a position controller, velocity controller, and thruster allocator that rescales outputs to avoid saturation. Additionally, feedforward terms for drag and buoyancy were added at the velocity level to improve stability and minimize error [9]. The same autonomy, mapping, and control code is intended to run in Gazebo simulation [10], bag replay, and on the vehicle with only driver and hardware-interface changes.

F. Alignment with Competition Strategy

Overall, the team's design approach aligns with the competition strategy focused on maintaining high performance navigation and reliable, modular design. Electronics revisions simplified previous implementations, enhancing maintainability and platform reliability. Modular firmware has enabled efficient

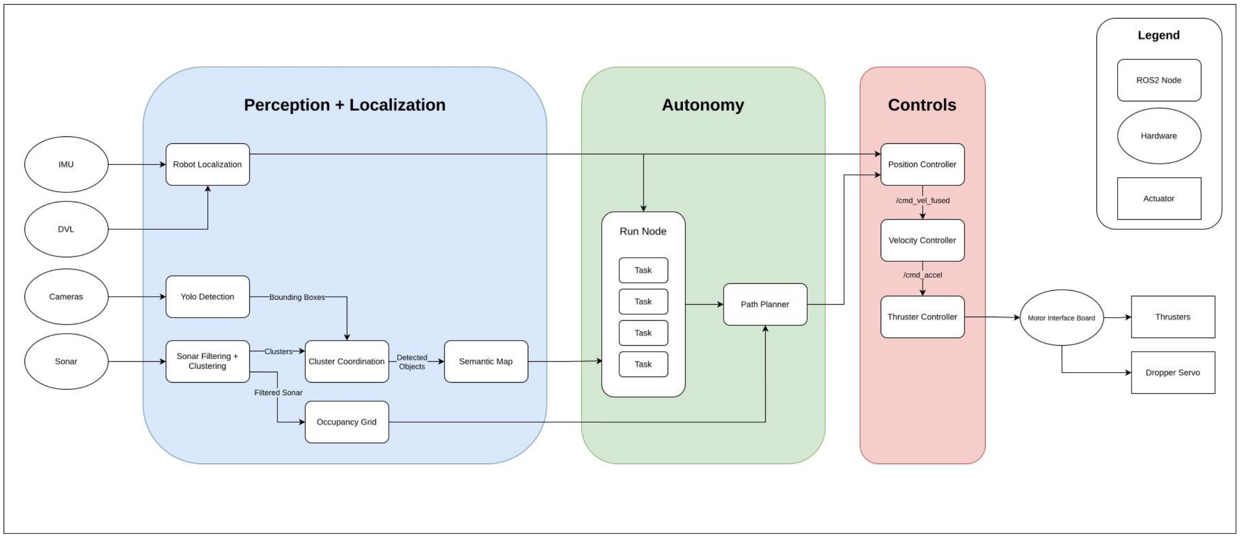

Fig. 4: High-level software flow from sensors through perception, semantic mapping, autonomy, controls, firmware, and actuators.

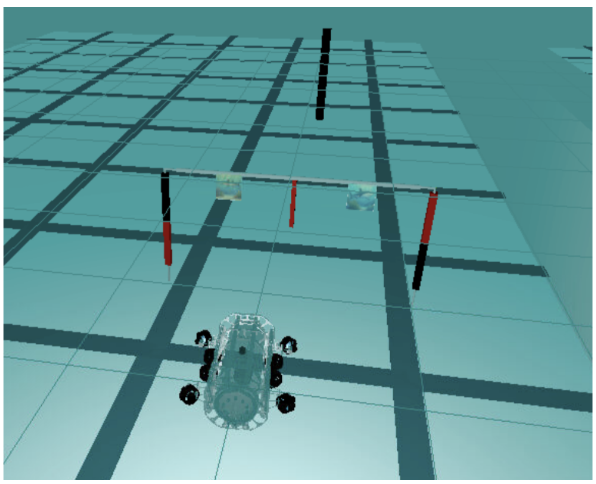

Fig. 5: Gazebo simulation of prequalification run with custom task elements.

reuse of code which has already been tested and validated in other contexts such as RoboBoat. The dropper subsystem was designed to be simple to align with a navigation-first task strategy with minimal additional risk. Software revisions especially focused on intelligent mapping and perception, which increase performance across tasks. This approach emphasizing reliability allowed for maximizing testing time, most of which was focused on validating the autonomy and controls stack.

III. Testing Strategy

The testing strategy mirrors the design strategy: verify the simplest observable piece first, then integrate only after the relevant subsystem has passed a lower-risk test. This gives the team a way to keep adding capability without allowing integration complexity to dominate pool time. The detailed test matrix used to track objectives, pass criteria, status, and evidence is included in Appendix A.

A. *Simulation and Course Mocks*

Simulation is used before water testing to validate mission state machines, search patterns, waypoint generation, controller mode transitions, and failure behavior. The Gazebo environment contains approximate vehicle dynamics, simulated sensor topics, and mock course elements for the gate, slalom, octagon, return-home, and bin tasks. The goal is not perfect underwater realism; it is to catch message-flow, state-machine, and controller-interface errors before a water test.

Physical course mocks are used in the pool when simulation is insufficient. Data from pool tests is recorded in ROS bags so perception and mapping changes can be replayed against the same sensor data after the test.

B. *Bench, Dry, and Pool Verification*

Bench tests validate boards and firmware before the vehicle is assembled. The team checks power rails, fuses, power-distribution boards, the motor interface board, E-stop sensing, serial command handling, returned telemetry, actuator neutral behavior, and dropper actuation. Dry vehicle tests then verify that the full assembled system powers on, communicates with sensors, reports safety state, and can be opened and reconnected without disturbing harness routing.

Pool testing is reserved for behavior that cannot be fully proven on land: waterproofing, vehicle dynamics, underwater perception, controller tuning, station keeping, and full mission integration. Pool tests follow checklists for enclosure sealing, power-on sequence, software launch, E-stop verification, tether configuration, and planned test objectives. Software changes are reviewed and exercised in simulation or dry tests before water testing so the team brings a known-working configuration to the pool whenever possible.

C. Progression to Competition Runs

Testing progresses from component validation to subsystem validation, full-vehicle dry tests, controlled pool tests, and then complete mission attempts. Early pool tests focus on sensor validation and controller tuning. Later tests focus on task-level autonomy: gate traversal, slalom association, octagon search, return-home behavior, and bin approach/drop. This staged sequence keeps each test tied to a capability from Table II and gives the team evidence for whether a task should remain enabled, simplified, or skipped in a competition run.

IV. Acknowledgments

The Marine Robotics Group thanks the organizations that support the team financially, technically, and logistically. The Aerospace Systems Design Laboratory at Georgia Tech provides the team's lab space, fabrication tooling, and advising. Tocaro Blue supports the team's marine perception and autonomy development through collaboration and technical feedback. VectorNav Technologies, TDK Lambda Corporation, Water Linked AS, Theia Technologies, and Blue Trail Engineering support the team through hardware donations and product discounts. Southern Fiberworx provides financial support and shipping support. NVIDIA provides financial support through matched donations from the Beeson-Battenburg Family Fund. Additional support from Georgia Tech comes from the AquaBots Vertically Integrated Project, Student Government Association, Student Organization Finance Office, Georgia Tech Student Foundation, and Campus Recreation Center. The team also thanks Carl Johnson for advising MRG, Dr. Michael West for RoboSub guidance, and Professor Dimitri Mavris for support of the broader organization. Additional sponsor and component details are included in Appendix B and C.

References

[1] Open Robotics, "Ros 2 documentation." https://docs.ros.org/.
[2] E. Beazley, J. Beeson, A. Betancourth-Pollett, S. Coutinho, J. Fang, S. Fish, Z. Greenberg, C. Tio, M. Turton, M. Woodward, and A. Wu, "Take two: Robosub 2025: Technical design report," 2025.
[3] RoboNation, "Robosub 2026 team handbook." https://robonation.gitbook.io/robosub-resources/.
[4] N. Hall, "Shape effects on drag," *NASA*, 2025.
[5] D. Battory, "Product-line architectures," *Smalltalk and Java in Industry and Practical Training*, 1998.
[6] PlatformIO, "Platformio documentation." https://docs.platformio.org/.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
[8] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.
[9] I. Schjlberg and T. Fossen, "Modelling and control of underwater vehicle manipulator systems," 1994.
[10] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2149–2154, Ieee, 2004.

Appendix A: Test Plan and Results

TABLE V: Staged test plan used to connect subsystem tests to competition-readiness evidence.

Subsystem	Environment	Objective	Pass Criteria	Status	Evidence / Result
E-stop and power	Bench / dry	Confirm safe thruster power enable and disable	Thruster relay remains open when not actuated; E-stop state reported before motor commands	Ongoing	Pre-test checklist item for every dry and pool test
Power distribution	Bench	Verify fused ESC distribution and rail stability	Correct polarity, fuse continuity, expected rail voltage, no heating under commanded load	Ongoing	Board-level inspection and bench power-on before integration
Firmware bridge	Bench	Validate serial command and telemetry contract	Motor/dropper commands parse correctly; timeout returns outputs to neutral; telemetry visible on host	Ongoing	Host-side serial tester and Jetson driver tests
Perception	Dry / pool bag replay	Detect course objects from camera and sonar data	Objects detected with stable labels and usable relative positions in recorded scenarios	Ongoing	ROS bag replay with updated perception and mapping nodes
Controls	Pool	Tune velocity and position controllers	Stable depth/heading hold; acceptable overshoot; no sustained oscillation during commanded maneuvers	Ongoing	Controller tuning logs from water tests
Autonomy	Simulation / pool	Validate task state transitions and skip/failure behavior	Gate, slalom, octagon, return-home, and bin states run without message-flow errors	Ongoing	Simulation runs before pool deployment
Dropper	Bench / pool	Confirm marker retention and release timing	Both markers release independently on command without servo stall or mechanical interference	Planned	Bench actuation followed by underwater release test
Full mission	Simulation / pool	Demonstrate integrated run sequence	Known-working software configuration completes selected task sequence or exits safely	Planned	Full simulation run before full pool mission attempt

APPENDIX B: COMPONENT LIST

Component	Vendor	Model/Type	Specs	Custom/Purchased	Cost	Year of Purchase
Buoyancy Control	—	—	—	—	—	—
Frame	In-house	Custom Waterjet	—	Custom	—	—
Waterproof Housing	Blue Robotics	Cylindrical Non-Locking Series 8” ID	8” ID, 8.5” OD, 26” Length Cast Acrylic Tube	Purchased	$220	2024
Waterproof Connectors	BlueTrail	Assorted Cobalt Connectors	3 Pin Power, 8 Pin, BlueROV Battery Cables, 6 Pin	Purchased	$50 ea	2025
Thrusters	Blue Robotics	T200	—	Purchased	$245 ea	2024
Waterproof Servo	BlueTrail	SER-2010	2.8 Nm, $\pm$ 70 degrees	Purchased	$375	2025
Motor Interface Board	JLCPCB	Custom PCB with Raspberry Pi Pico	—	Custom	$30	2026
High Level Control	NVIDIA	Jetson Orin Nano	—	Purchased	$250	2024
Battery	Socokin	4S Lipo Battery	6600mAh 14.8V, 2 sets of 2 LiPos wired in parallel	Purchased	$50 ea	2025
Regulator	DROK	Boost Buck Converter	DC 5.5-30V to 0.5-30V	Purchased	$17 ea	2024
Internal Comm Network	In-House	USB / Ethernet	USB3 / 1Gbps	Custom	—	—
External Comm Interface	BlueTrail	Ethernet Tether	1 Gbps	Custom	—	2025
Cameras	Blue Robotics	Low-Light HD USB Camera	2x Forward, 1x Downward	Purchased	$110 ea	2024
Inertial Measurement Unit (IMU)	LORD MicroStrain	3DM-GX3-25	—	Purchased	$2640	2017
Doppler Velocity Log (DVL)	WaterLinked	A50	—	Purchased	$7890	2024
Acoustics	Sonoptix	Echo Multibeam Imaging Sonar	—	Purchased	$8950	2024
Algorithms	Open-source	—	Color-thresholding, YOLO, Extended Kalman Filter, State Machine	Custom	—	—
Localization and Mapping	Open-Source	—	Extended Kalman Filter	Custom	—	—
Autonomy	Open-Source	—	State Machine	Custom	—	—
Open Source Software	Open-Source	—	ROS2, OpenCV, Gazebo	Custom	—	—
Programming Language(s)	Open-Source	—	Python, C++	Custom	—	—

APPENDIX C: SPONSOR INFORMATION

Name	Tier	Gift-In-Kind	Material Contribution	Fiscal	Donation Amount	Additional
Tocaro Blue	Platinum	Y	Furuno DRS4D-NXT Radar, Furuno SCX-21 Satellite Compass, Tocaro ProteusCore Development Kit	Y	$5000	—
VectorNav Technologies	Gold	Y	VN-100, VN-300	N	—	—
Theia Technologies	Gold	Y	Optic Lens	N	—	—
TDK Lamda	Silver	Y	DC Power Supply	N	—	—
WaterLinked Technologies	Silver	N	—	N	—	Discount on Modem M16
Blue Trail Engineering	Friend of MRG	N	—	N	—	Product Discounts
Southern FiberWorkx	Friend of MRG	N	—	Y	$1200	Transportation Assistance
NVIDIA	Friend of MRG	N	—	Y	$1250	Matched Contribution from Individual Donor

APPENDIX D: CAD OF THE AUV

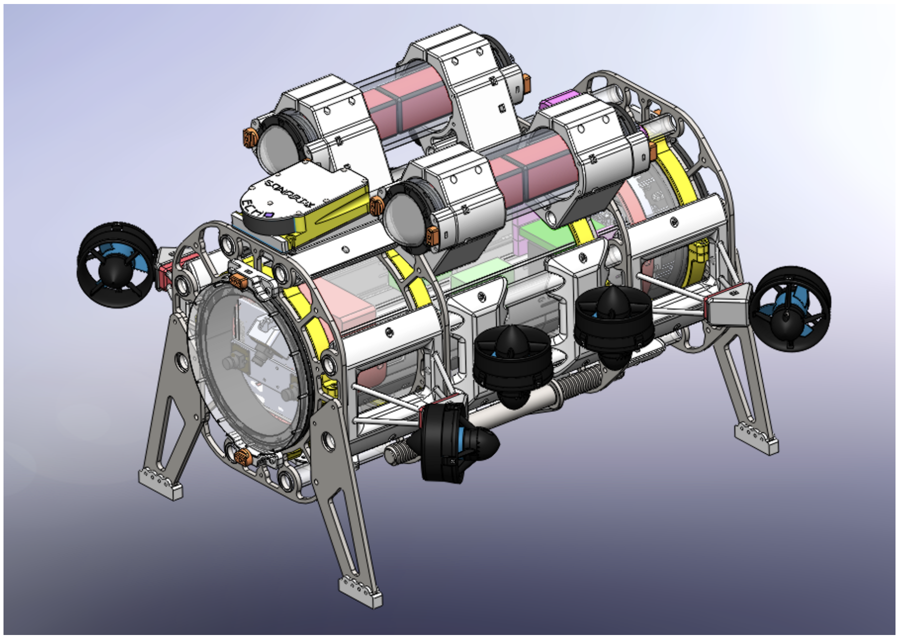

Fig. 6: 3D isometric view of the fully assembled AUV

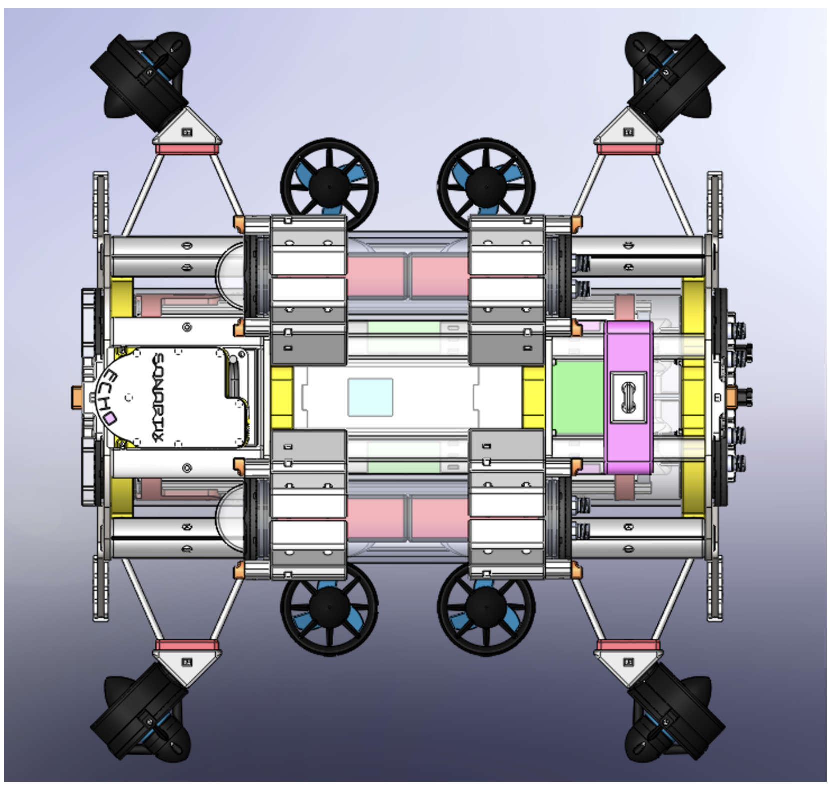

Fig. 7: Top view of the fully assembled AUV

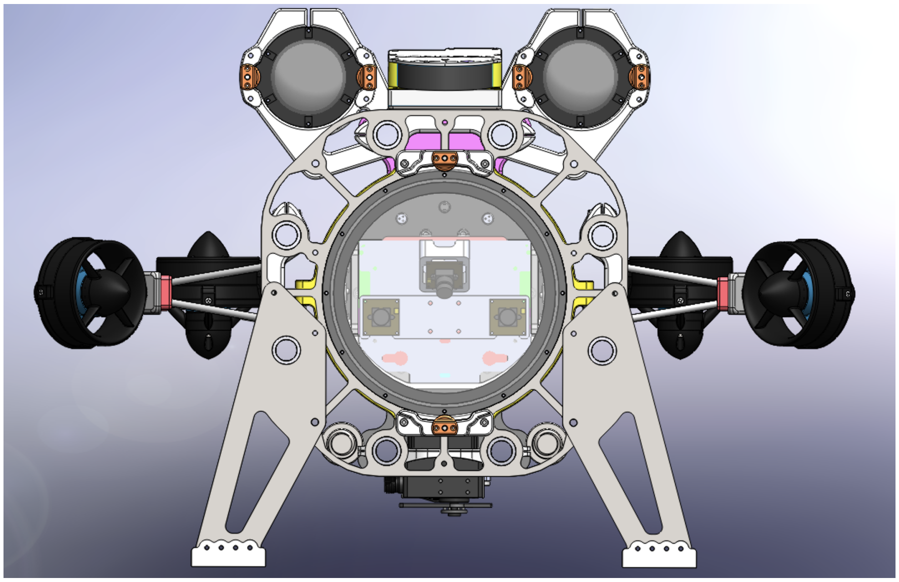

Fig. 8: Front view of the fully assembled AUV

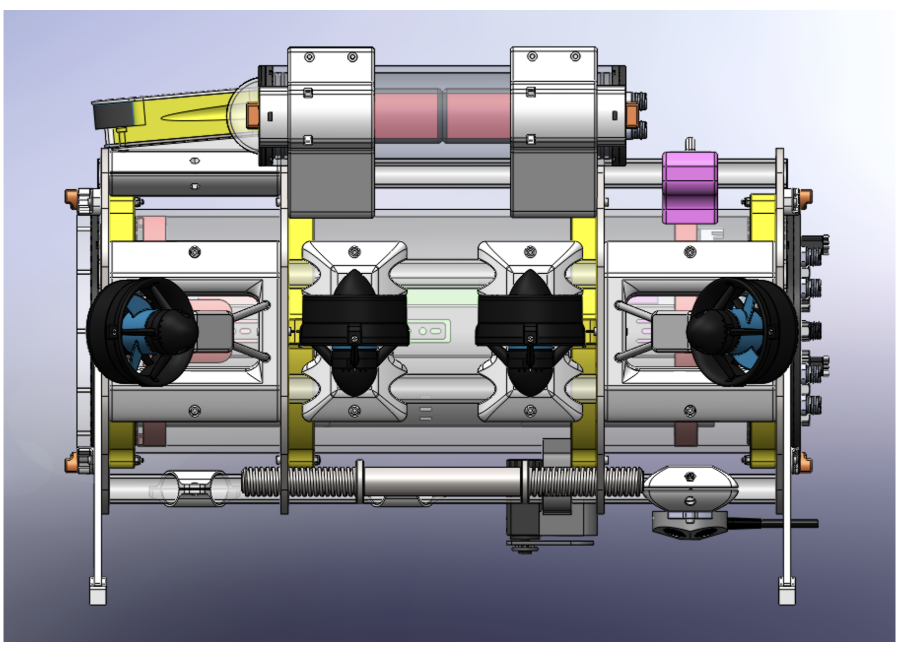

Fig. 9: Side view of the fully assembled AUV