RoboSub 2026 Technical Design Report

Project Poseidon's Code (Bangladesh RoboSpace Consortium)

Arkoprotik Acharya, Ishtiaque Ahmad Eyamin, Rupkotha Roy, Omera Fidan, Mohaimen Ul Haq, Adiyat Yeamim Rayan, Syed Faheem Hassan, Naziat Ul Islam Lammim, Adnan Sami Pathan, Soroforaz Hossain, S.M. Abtahi Noor, Aranya Abeer Khan Prapya, Sarrinah Reza

***Abstract*—This paper details the RoboSub 2026 entry by Project Poseidon's Code, Bangladesh's first high-school-university joint team. Prioritizing reliability over breadth, our strategy targets the consistent completion of core tasks. The AUV features an 8-thruster, 45° fully vectored layout for holonomic 6-DOF control, housed in a lightweight carbon-fiber and machined-aluminum chassis. The autonomy stack integrates a Pixhawk v6x and Jetson Orin Nano, running YOLOv8 via an OAK-D stereo camera. To preserve buoyancy trim, interchangeable payloads are swapped between runs. This report outlines our competition strategy, design rationale, and testing methodology.**

I. Competition Strategy

A. General Strategy and Team Context

Project Poseidon's Code is the first high-school-university joint team to enter RoboSub from Bangladesh. The team is structured into four subteams: Mechanical, Electrical, Software, and Documentation. The team operates under the technical mentorship of the UIU Centre for Artificial Intelligence and Robotics (CAIR) and the institutional support of Dhaka Residential Model College. RoboSub 2026 [13] is the team's debut international competition.

Our approach is to develop a robot that can perform a few basic functions well and in a consistent manner, rather than trying to do everything. We use real pool footage for testing and training our system because real underwater conditions are different from simulation. Given the time and resources constraints, our emphasis was stable, simple, and competition performance. As Bangladesh's first entry into RoboSub, the documented baseline we bring back from 2026 matters more than the headline score.

B. Mission Task Strategy

Gate and Slalom-The mission starts with the gate task, where the vehicle selects the Survey and Repair side and performs the roll bonus after passing through. The custom vectored thruster layout enables stable rolling and forward motion simultaneously. Using the front-facing OAK-D Camera, IMU, and onboard LEDs, the AUV maintains accurate alignment and visibility underwater. After passing through the gate, the vehicle proceeds directly to the slalom task. The lightweight carbon fibre structure and full 6-DOF control allow smooth turning, fast correction, and minimal drift while navigating through the buoy field inside the boundary. Bin and Torpedo-For the bin task, the camera enables accurate alignment for marker dropping using stable lateral movement without unnecessary rotation. During the torpedo task, the OAK-D Camera and onboard LEDs assist target tracking, while the stable frame improves firing accuracy. Octagon and Return Gate-For the octagon task, the gripper payload is mounted separately to keep the vehicle lightweight and manoeuvrable. The six carbon fibre buoyancy tubes maintain stable ascent and resurfacing during the task. After completing the octagon, the vehicle immediately navigates back through the return gate using onboard vision, IMU feedback, and efficient power usage from the lightweight design.

C. Vehicle Capability Envelope

Our AUV is kept very lightweight to ensure stable and efficient 6-DOF control. The 45° lateral and 35° vertical vectored thruster setup works properly only with this low mass, allowing smooth surge, sway, and heave motion with low power use. If the vehicle becomes heavy, it loses maneuverability, consumes more power, and cannot respond accu rately to control inputs, making mission performance unreliable. To avoid this, we do not mount the dropper, torpedo launcher, and gripper together. Carrying all of them increases weight, drag, and reduces stability. Instead, we use interchangeable payload modules, installing only the required system for each task. This keeps the AUV light, stable, and fully controllable underwater.

II. Design Strategy

A. Mechanical Design and Improvements

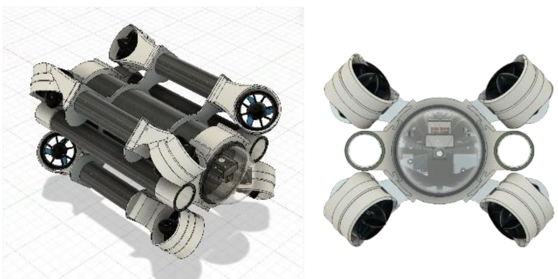

Fig. 01: Poseidon-1.0 AUV Render

1) Structural Design and Material Selection

Our AUV is a lightweight and highly controllable platform for RoboSub missions, with an approximate weight of 12 kg. It is built using a carbon fibre frame and six carbon fibre buoyancy tubes, which actively support buoyancy control by adjusting internal air volume based on displacement and mission load. This allows stable underwater operation with a slight positive buoyancy of about 0.3–0.5 kgf, keeping nearly 83% of the vehicle submerged during static tests. The vehicle uses a custom vectored thruster system for full 6-DOF motion, with each thruster mounted at 45° lateral and 35° vertical angles for precise and efficient manoeuvring in all directions. The lightweight structure improves balance, responsiveness, and control stability. An aluminium ring-based structural support combined with a compact 3D-printed ABS filament optimised frame design enhances rigidity while keeping weight low. The modular tray and hull system simplifies assembly, maintenance, and subsystem integration. Overall, the design is purpose-built for RoboSub tasks, ensuring reliable performance in complex underwater environments.

Thruster	Surge (X)	Sway (Y)	Heave (Z)	Roll	Pitch	Yaw
T1	+0.8192	-0.4056	+0.4056	-0.0728	-0.2917	-0.1446
T2	+0.8192	+0.4056	+0.4056	+0.0728	-0.2917	+0.1446
T3	+0.8192	-0.4056	-0.4056	-0.1505	-0.2917	-0.0123
T4	+0.8192	+0.4056	-0.4056	+0.1505	+0.2917	+0.0123
T5	+0.8192	-0.4056	-0.4056	+0.0728	+0.2917	-0.1446
T6	+0.8192	+0.4056	-0.4056	-0.0728	+0.2917	+0.1446
T7	+0.8192	-0.4056	+0.4056	+0.1505	+0.2917	-0.0123
T8	+0.8192	+0.4056	+0.4056	-0.1505	+0.2917	+0.0123

Table 1: Thruster Configuration Matrix (TCM).

2) Payload System

Our AUV supports three interchangeable payloads: dropper, torpedo launcher, and gripper. All systems are 3D-printed and custom-designed based on this year's RoboSub task objects, ensuring better fit, accuracy, and mission reliability. Each payload mounts on a shared aluminium interface, allowing quick swapping while maintaining structural balance.

- Dropper: A lightweight mechanism designed for precise marker release during the bin task. Its simple vertical drop system that improves accuracy during lateral positioning.
- Torpedo Launcher: A rigid system fixed to the frame to avoid launch deflection. It fires at a fixed angle for consistent and repeatable targeting underwater.
- Gripper: A structure with TPU soft fingers, designed for secure object handling in the octagon task. The flexible TPU contact improves grip and reduces slippage on mission objects.

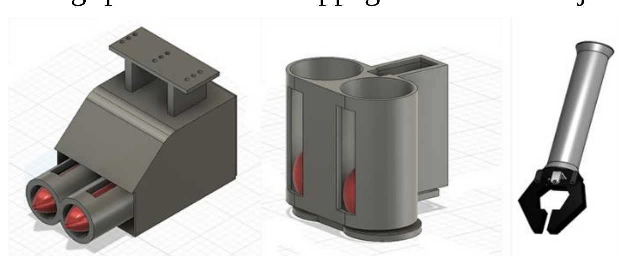

Fig. 02: Renders of payload systems.

B. Electrical Stack

The system employs a 24V 15000mAh 6S custom built lithium ion (Li-ion) battery pack for safe and reliable power distribution through several domains isolating high propulsion loads electrically from compute and control systems using a high-power DC-DC buck converter and power modules. Power is distributed from rails of Power distribution Board (PDB). A complete inventory of vehicle components, vendors, and procurement costs is provided in Appendix F.

1) Power Architecture

The system architecture consists of four domains:

- Propulsion domain
- Control domain
- Compute domain
- Sensor and payload domain

Propulsion domain consists of m080 thrusters driven by 60A brushless ESC's converting battery DC power into electronically commuted 3 phase signals. The control domain containing Pixhawk v6 providing PWM signals to the ESC's. The compute domain which includes Jetson Orin nano super handles computer vision tasks like object detection and communicates with the Pixhawk via MAV Link. The sensors suite includes IMU and auxiliary navigation sensors whose outputs help in 3 axes pose estimation for stable navigation underwater.

2) Microcontroller and Compute

The AUV is designed with a distributed architecture having Pixhawk V6X as the low-level controller and Jetson Orin Nano as the high-level compute unit. Pixhawk is used for real-time stabilization using Bar30 pressure sensor data. IMU data is also directly forwarded to Jetson for perception and navigation assistance. The thrusters are PWM/CAN controlled. Jetson handles computer vision, AI inference, navigation, and RoboSub tasks. Pixhawk and Jetson communicate over USB with MAVlink to send telemetry and control the mission in real time.

3) Power Distribution, Filtering, and Safety

The propulsion domain is powered by a 24 V, 15 Ah, 6S lithium-polymer pack, sized for the dominant load on the vehicle. The eight M080 thrusters each draw a peak current of 16 A (128 A aggregate, 3,072 W instantaneous), but mission profiling assumes a 60 % continuous duty cycle at 10 A per thruster — a 1,920 W sustained propulsion budget that the carbon-fiber chassis and low-drag geometry are expected to under-run in practice. The electronics domain draws an additional 53.3 W peak across the Pixhawk V6X (6 W), Jetson Orin Nano (20 W), IMU (2 W), Bar30 depth sensor (0.3 W), OAK-D camera (5 W), and actuator servos (20 W). LC filtering on each ESC output suppresses switching-noise injection into the shared bus, in-line fuses protect against thruster-stall fault currents, and a top-mounted kill switch cuts the propulsion contactor within one second of activation per RoboSub safety requirements.

C. Software Stack

The software runs on ROS 2 Humble [1] across a twelve-package modular workspace, each package enforcing a single-responsibility boundary and communicating only through typed auv_msgs interfaces. The architecture follows a layered autonomy hierarchy — a 10 Hz behavior-tree mission planner, a 20 Hz drift-aware trajectory planner, and a 50 Hz 6-DOF PID controller — separating strategic, kinematic, and dynamic concerns into independently tested tiers. Every numerically non-trivial module (state estimator, thrust allocator, trajectory kinematics) is implemented as a Python class with zero ROS dependencies, allowing isolated unit testing before any hardware integration.

The autonomy stack is anchored on a single thesis: vision-anchored, drift-budgeted autonomy with graceful degradation. Because the vehicle does not carry a Doppler velocity log or hydrophone array, every blind translation is bounded by a measured drift budget enforced as a py_trees decorator, every task is decomposed into perception-fixed waypoints, and every behavior is paired with a fallback path.

A complete system-level data flow diagram appears in Appendix E (Architecture Block Diagrams), and the full architecture details and YOLOv8 training pipeline are documented in Appendix C (Software Test Plan).

1) Perception

The forward-facing OAK-D stereo camera [11] streams 1080p RGB at 30 Hz to the Jetson Orin Nano, which executes a TensorRT FP16-quantized YOLOv8m detector [2] with a measured inference latency of 4.5 ms per frame. The four-class detection schema is task-aware and switched at runtime from the mission planner blackboard. CLAHE preprocessing [4] compensates for underwater color attenuation and backscatter, and ByteTrack [3] provides persistent object identities across the inevitable frame drops in turbid water. Monocular range estimation from known target dimensions is cross-checked against the OAK-D's hardware stereo depth, and the higher-confidence value is published.

A classical OpenCV detector based on HSV color thresholding runs in parallel as a graceful-degradation fallback: if the neural detector returns zero detections for over one second, the pipeline switches to the classical branch, which targets only the high-contrast gate divider. The full training dataset composition, augmentation strategy, mAP validation, and confidence-gating thresholds are documented in Appendix C.

2) Localization

State estimation is handled by a custom 15-state Error-State Kalman Filter (ESKF) [5] chosen over a conventional EKF for two reasons: the error-state formulation operates in a vector space, avoiding the quaternion-wrapping bugs that plague Euler-angle EKF implementations, and the small error state is reset after each measurement update, keeping the filter numerically stable over long missions. The nominal state — position, velocity, attitude quaternion, and IMU biases — is propagated at 100 Hz by direct integration of the Wheeltec N100 nine-axis IMU. Measurement updates from the Bar30 pressure-derived depth sensor (≈0.5 cm RMS noise, 20 Hz) and the N100's magnetometer-stabilized attitude estimate inject corrections into the nominal state via the standard ESKF inject-and-reset cycle. Innovation gating discards measurements whose residual exceeds a covariance-bounded threshold, protecting against extreme sensor readings. The full state representation, propagation equations, and Jacobian derivations appear in Appendix C.

3) Control System

The vehicle's control layer uses six independent parallel PID loops running at 50 Hz on the Jetson Orin Nano, matching state estimates from the custom ESKF against waypoints from the trajectory planner. Each tracking axis implements a conditional-integration anti-windup clamping algorithm [7] that freezes the error accumulator whenever an individual thruster saturates, preventing spatial oscillations. Acceleration feedforward terms are added to the surge and heave loops to instantly counteract structural hydrodynamic drag. The computed multi-axis control effort vector $\tau \in \mathbb{R}^{6\times1}$ is mapped to the eight physical actuators using the Moore-Penrose pseudoinverse [6] of the CAD-extracted Thruster Configuration Matrix ($B \in \mathbb{R}^{6\times8}$):

$$u = B^{+}\tau$$

The 45° geometry trades single-axis peak thrust for multi-axis simultaneous control authority. A heavy vehicle would require dedicated-axis thrusters to overcome inertia along its primary direction; our 12 kg lightweight carbon-fiber chassis operates comfortably with the multi-axis projected forces, so no thrust is wasted on dead-mass translation. The result is coupled trajectories such as active roll-assisted surge for style points through the gate, while maintaining strict power efficiency. Commands are sent at 20 Hz via MAVROS [9] to the Pixhawk V6X, where ArduSub [8] outputs the final motor PWM.

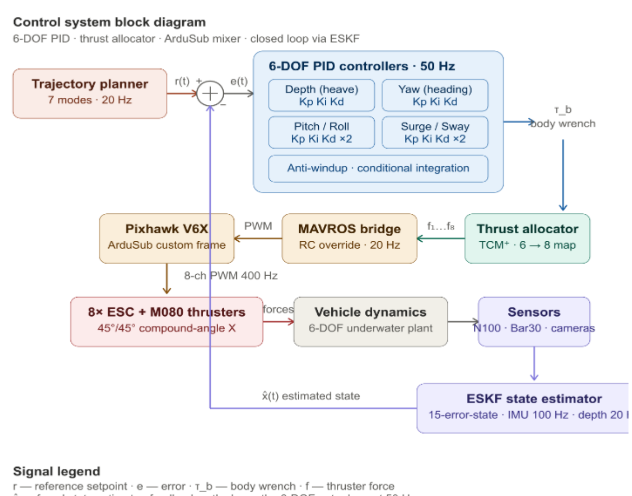

Fig. 03: Control system block diagram.

4) Mission Planner

The mission planner is built on py_trees [10], selected over BehaviorTree.CPP and finite-state machines for its native ROS 2 integration, runtime tree introspection, and the Python ergonomics that allow behavior iteration between pool sessions without recompiling. Eleven reusable leaf behaviors — covering depth and yaw setpoints, object detection queries, visual alignment, approach, search patterns, marker drop, surface, and arming — compose into every mission tree. Each behavior is a class with initialise, update, and terminate lifecycle methods returning SUCCESS, RUNNING, or FAILURE. Two architectural patterns carry most of the mission-planner load. A drift-budget decorator wraps every blind-motion behavior with a tunable time budget (default six seconds) and forcibly interrupts the motion if exceeded — a direct response to operating without a Doppler velocity log. Capability gates are evaluated once at boot from the parameter server; missing capabilities replace their gated subtree with a no-op leaf, honoring the deliberate exclusions of Section I-C at the code level. An example behavior-tree fragment is shown in Fig. 04.

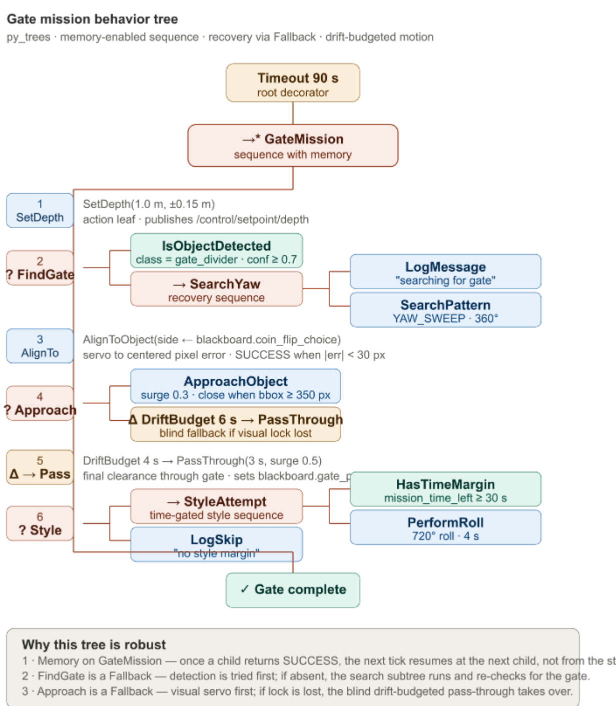

Fig. 04: Example of behavior tree specifying mission logic.

5) Network and Telemetry

During tethered development and dry-lab testing, high-bandwidth communications rely on a Gigabit Ethernet link encapsulated within a ruggedized, neutrally buoyant tether. Real-time parameter tuning and vision state verification are monitored using a custom web-based ground control station that parses native ROS 2 diagnostic topics.

For autonomous competition runs, the physical tether link is severed to ensure unconstrained 6-DOF movement, and all telemetry streams are routed completely internally. The AquaCore logging configuration saves synchronized sensor data from the primary Wheeltec N100 IMU, YOLOv8 detection confidences, and raw camera images directly to an onboard high-speed NVMe storage drive at a compressed rate of 25 MB/s using a rosbag2 logging daemon for post-run diagnostic and regression evaluation.

D. Innovation and Creative Aspects

Five design choices in this vehicle are non-standard for student-team RoboSub entries: (1) the eight-thruster fully vectored allocation at compound 45°, providing holonomic six-DOF motion with single-thruster fault tolerance (Section II.A.1); (2) carbon fiber external buoyancy tubes that couple structural function and displacement in one element, where most student vehicles use foam or PVC; (3) multi-function aluminum structural rings that simultaneously serve as thruster mounts, tube-to-hull interfaces, low-CoM ballast, payload mounts, and thermal conductors; (4) the 100 percent infill plus acetone-vapor-cured ABS process, rendering 3D-printed components effectively non-absorptive across the deployment; and (5) the interchangeable payload swap strategy, preserving the buoyancy trim and thrust allocation calibration across both Bin and Torpedo tasks rather than compromising both.

III. Testing Strategy

A. Test Strategy Overview

Our testing strategy focuses on reliability in real underwater conditions rather than simulation-only results. The team manually collected and annotated over 5,000 real pool images for each mission task using different lighting conditions, angles, and water clarity levels to improve detection performance underwater.

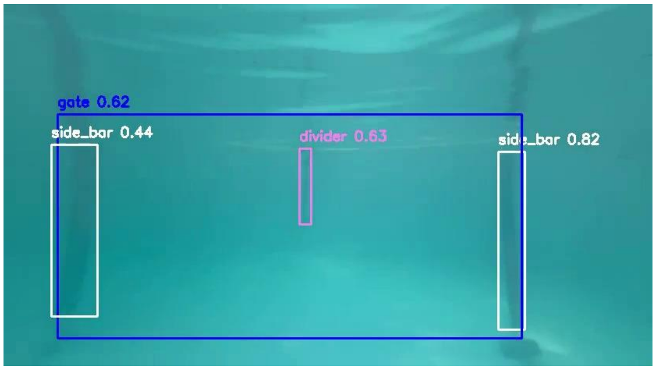

Fig. 05: Mission Gate Detection via AUV Vision System

Development follows a step-by-step process. The AUV is first trained and validated on the gate task before moving sequentially to slalom, bins, torpedo, octagon, and full mission runs. A new task is added only after the previous one performs consistently. To improve robustness, tasks are tested under difficult conditions such as blur, low light, partial target loss, and unstable movement. All pool tests are recorded using ROS bag logging for offline analysis, allowing the team to evaluate detection accuracy, controller response, drift, and recovery behavior after each run. Before every pool session, the complete software stack is verified through bench testing to ensure stable communication between perception, control, and navigation systems. Final testing focuses on full autonomous mission runs under competition-like conditions to validate overall system reliability and recovery performance.

ACKNOWLEDGEMENTS

We thank our Chief Advisor Dr. Hasan Sarwar, Dean of the School of Science and Engineering at United International University, and our Chief Patron Brigadier General Dr. Mohammad Zaber Hossain, Principal of Dhaka Residential Model College, for the overall support provided by the faculty and administration of both institutions that made this project possible. We are deeply grateful to our Technical Advisor, Md Abid Hossain (Lecturer, Department of CSE, UIU), for daily technical guidance and project leadership across every phase of the build. Our mentors Mr. Saif Al Saad (Team Lead, UIU Mars Rover Team) and Mr. Mushfiqur Rahman provided sustained mechanical and electrical expertise throughout development. We are also grateful to Mr. Alex Szeto and his team "Team Inspiration" for their support and valuable guidance. UIU CAIR provided laboratory access, equipment, and research mentorship as our Collaborative Partner; Dhaka Residential Model College provided the institutional foundation and the student core of the team as our Institutional Partner.

REFERENCES

[1] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Sci. Robot.*, vol. 7, no. 66, May 2022, Art. no. eabm6074. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.abm6074

[2] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel, Oct. 2022, pp. 1–21. [Online]. Available: https://doi.org/10.1007/978-3-031-20047-2_1

[4] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV*, P. S. Heckbert, Ed. San Diego, CA, USA: Academic Press, 1994, pp. 474–485. [Online]. Available: https://dl.acm.org/doi/10.5555/180895.180940

[5] J. Solà, "Quaternion kinematics for the error-state Kalman filter," *arXiv:1711.02508*, Nov. 2017. [Online]. Available: https://arxiv.org/abs/1711.02508

[6] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. Hoboken, NJ, USA: Wiley, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119575016

[7] K. J. Åström and L. Rundqwist, "Integrator windup and how to avoid it," in *Proc. Amer. Control Conf. (ACC)*, Pittsburgh, PA, USA, Jun. 1989, pp. 1693–1698. [Online]. Available: https://ieeexplore.ieee.org/document/4790464

[8] ArduPilot Development Team, "ArduSub: Open source submarine autopilot." [Online]. Available: https://www.ardusub.com

[9] V. Ermakov et al., "MAVROS: MAVLink communication node for ROS." [Online]. Available: https://github.com/mavlink/mavros

[10] D. Stonier, "py_trees: A Pythonic implementation of behaviour trees." [Online]. Available: https://github.com/splintered-reality/py_trees

[11] Luxonis Holding Corp., "OAK-D Pro hardware documentation." [Online]. Available: https://docs.luxonis.com

[12] Eclipse Foundation, "Eclipse Cyclone DDS." [Online]. Available: https://cyclonedds.io

[13] RoboNation, "RoboSub 2026 team handbook," 2026. [Online]. Available: https://robonation.org/programs/robosub

APPENDIX A — MECHANICAL DETAIL

A.1. Hydrostatic Stability Calculation

Hydrostatic stability is established by ensuring the vertical separation between the center of buoyancy and the center of mass produces a positive metacentric height (GM). The CoB is computed as the volume-weighted centroid of the displaced fluid, dominated by the six external carbon fiber buoyancy tubes. The CoM is computed as the mass-weighted centroid of all components: the battery pack at the lowest internal position on the lower tray plate, the aluminum rings at low external positions, the carbon fiber hull and tubes distributed by their geometric centroids, and the internal electronics at their mounted positions on the upper tray plate.

A.2. Acetone Vapor Curing of ABS Components

Every printed ABS component on the vehicle is fabricated at 100 percent infill and finished with an acetone vapor curing step. Both choices are deliberate departures from common practice. Standard 3D printing uses 15 to 30 percent infill to save time and material, leaving an internal lattice surrounded by hollow voids. If the surface seal of such a part is ever broken in service, water enters the voids and cannot drain, adding parasitic mass and shifting the vehicle's center of mass between deployments. Printing at 100 percent infill removes those voids entirely. The acetone curing step then addresses the layer lines on the printed surface, which act as capillary paths that slowly absorb water even in a solid part. The component is placed in a sealed vessel with a small reservoir of liquid acetone, and the vessel is held at a controlled temperature for a defined exposure period. Acetone vapor reaching the part surface dissolves the outermost layer of the ABS, fusing adjacent layer lines into a continuous, sealed skin. The interior of the part is unaffected; only the outermost surface is modified. Together, these two steps render the printed components effectively non-absorptive across the duration of a competition deployment, a property untreated ABS does not provide.

A.3. Hull Sealing and Internal Layout

Hull sealing is provided by Buna-N O-rings at every hull-to-component interface. Under increasing depth, external pressure compresses the O-ring further into its precision-machined seat, which improves sealing performance with depth rather than degrading it. Acetoxy silicone is applied at internal joints prone to fine porosity, providing a secondary barrier without compromising the removability of the primary O-ring seal. Three alternative strategies were evaluated and rejected: epoxy resin coatings (rigid, prone to cracking under sustained thruster vibration, and irreversible); silicone-only application (no structural mechanical seal at hull interfaces); and adhesive-only bonding (irreversible, blocks all component access). The combined O-ring + acetoxy silicone approach, paired with acetone-cured ABS surfaces, was selected as the only strategy that satisfies all three operational requirements: pressure-tight sealing, vibration tolerance, and component removability.

A.4. Modeled Runtime vs. Vehicle Mass

The lightweight construction approach described in Section II.A.3 is justified through a comparative model of expected mission performance against vehicle mass. The model assumes equivalent battery capacity and similar hydrodynamic profiles, and varies only the vehicle dry mass to estimate the resulting power demand at cruise speed and the corresponding mission runtime.

APPENDIX B — ELECTRICAL INTEGRATION TEST PLAN

B.1. Power Budget and Load Analysis

To validate the electrical testing parameters, a theoretical power budget was calculated for the 24V, 15000mAh (360 Wh) 6S custom Lithium-ion battery pack. The autonomous underwater vehicle does not operate at a static continuous load; instead, it transitions between distinct operational regimes.

Propulsion Domain: The eight M080 thrusters draw fluctuating currents based on the commanded trajectory:

- Idle / Station Keeping: 2–5 A per thruster.
- Cruise (Normal Mission): 10 A per thruster. Total continuous current I_{cruise} = 8 × 10 A = 80 A.
- Aggressive Maneuver: 16 A peak per thruster. Total peak current I_{peak} = 8 ×16 A = 128 A.

Total Power and Runtime Estimation: During a nominal cruise profile, the propulsion network demands 1920 W (80 A × 24 V). The low-power electronics domain (Pixhawk v6x, Jetson Orin Nano, OAK-D, Bar30, IMU, and actuators) consumes a combined maximum of 53.3 W.

$$P_{total} = 1920\text{ W} + 53.3\text{ W} = 1973.3\text{ W}$$

Accounting for a 90% DC-DC converter efficiency and applying a 30% safety margin for transient spikes, voltage sags, and thermal derating, the system is designed to safely handle peaks up to 2540 W. Based on the 360 Wh total battery energy, the theoretical runtime under continuous cruise load is calculated as:

$$t = \frac{360\text{ Wh}}{1973.3\text{ W}} \approx 0.18\text{ hours} \approx 11\text{ minutes}$$

Because physical thrust demand is intermittent during actual pool navigation, the realistic in-water mission runtime safely exceeds the 11-minute threshold.

B.2. Integration-Phase Testing Feature Checklist

The following test plan translates the vehicle's electrical design parameters into a rigorous, hardware-in-the-loop verification checklist executed prior to in-water deployment.

Category	Items
Power Distribution	Methodology: The entire vehicle's propulsion loads, onboard compute stack, sensors, and payload actuators are powered simultaneously directly from the battery to verify stability across all power regimes. Pass Criteria: The integrated LC filtering successfully smooths electrical noise from the ESCs. Voltage stability across the isolated 24V, 12V, and 5V DC-DC buck rails remains within a strict ± 2% tolerance during simulated 80 A continuous load tests.

Category	Items
Thruster and Kill System	Methodology: The external magnetic kill switch is triggered manually while the vehicle is simulating a live continuous thrust command on the bench. Pass Criteria: For emergency compliance, the kill switch successfully severs power to the propulsion domain in < 1 second. Crucially, the compute domain (Jetson), control domain (Pixhawk), and all sensors remain completely powered and in a safe state, ensuring telemetry logs and visual data are not corrupted during an emergency stop.
Sensors	Methodology: The Pixhawk v6x, primary Wheeltec N100 IMU, and Bar30 pressure sensor are subjected to dry static testing and known-depth water column submersion. Pass Criteria: Raw sensor data is actively received by the companion computer. The IMU is successfully calibrated on a flat plane to mathematically remove gyro bias, accelerometer drift, and scale errors prior to the ESKF fusion step.
Compute Bus	Methodology: The Pixhawk v6x, primary Wheeltec N100 IMU, and Bar30 pressure sensor are subjected to dry static testing and known-depth water column submersion. Pass Criteria: Raw sensor data is actively received by the companion computer. The IMU is successfully calibrated on a flat plane to mathematically remove gyro bias, accelerometer drift, and scale errors prior to the ESKF fusion step.

Table B.I: Integration-phase testing feature checklist.

Integrated hardware block diagram detailing the combined power distribution and data communication pathways. The schematic illustrates the 24V propulsion isolation via the kill switch, the regulated compute and control rails (19V and 5.3V), and the low-level data bridges (MAVLink, USB, I2C, SPI) connecting the Jetson Orin Nano, Pixhawk flight controller, and sensor suite.

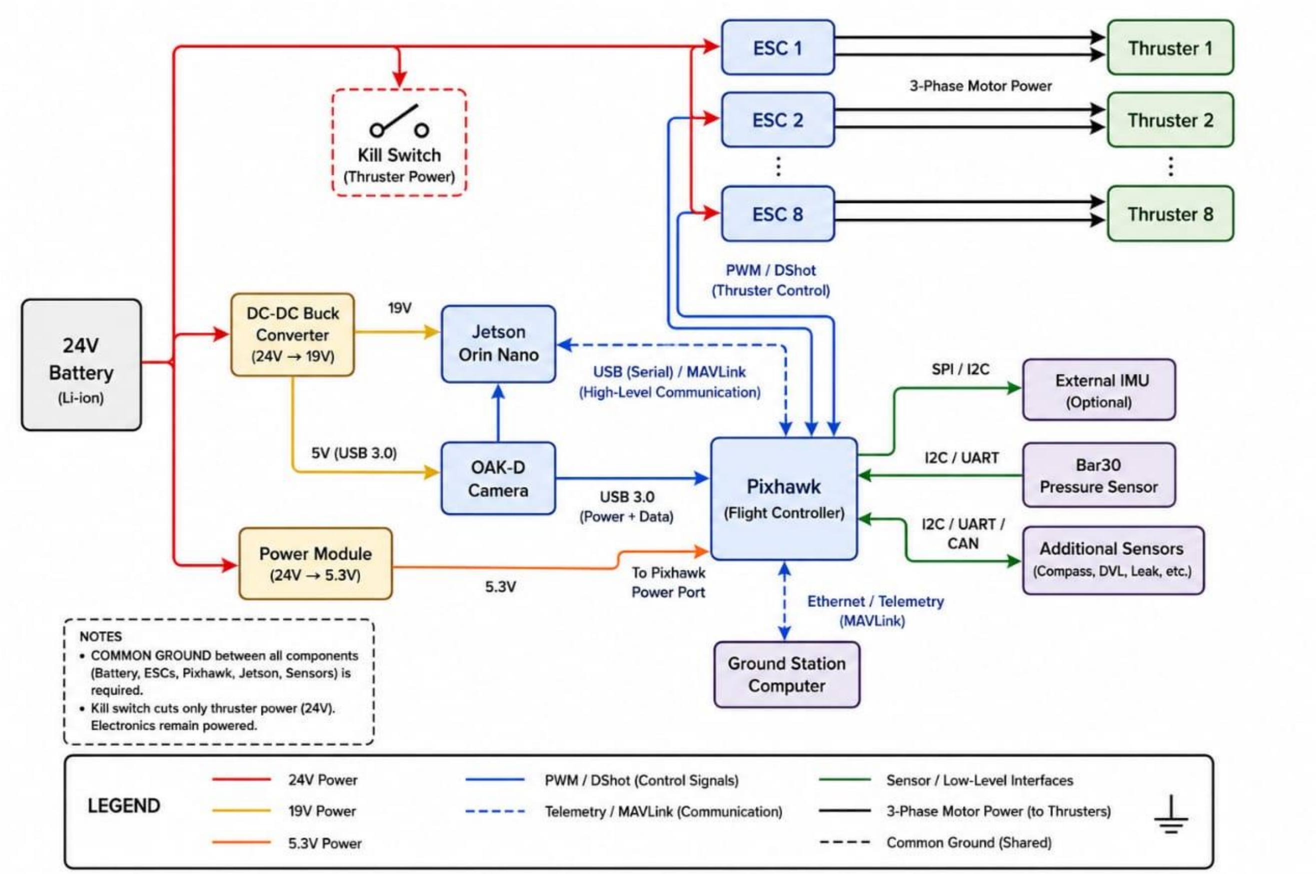

Fig. 06: Integrated hardware system architecture.

APPENDIX C — SOFTWARE TEST PLAN

C.1. Component Testing Schedule and Methodology

To systematically validate the AquaCore stack, testing is divided into distinct phases with strict pass/fail gates. The table below outlines the testing methodology and timeline for each core software capability.

Component / Capability	Description	Methodology	Timeline
Sensor Drivers	Validation of depthai-ros (OAK-D), GStreamer H.264 (OpenIPC), and Wheeltec N100 IMU drivers.	Bench rig testing using ros2 topic hz to verify that the OAK-D publishes at ≥25 Hz, the IP camera maintains <60 ms latency, and the IMU streams at 100 Hz without timestamp-skew.	Phase 3a (Pre-Pool Bench Integration)
Localization and Movement	Verification of the 15-state Error-State Kalman Filter (ESKF) and 6-DOF PID trajectory controllers.	Pure-math pytest suite to verify static convergence and depth tracking. In-water tuning using logged step responses to ensure <10% depth overshoot and <2° yaw drift over 5 minutes.	Phase 1 (Unit Tests) & Phase 3b (Pool Tuning)
Perception	Validation of the TensorRT-optimized YOLOv8m detector, ByteTrack multi-object tracking, and classical HSV color-segmentation fallback.	Offline playback of LFS-tracked rosbags to diff bounding box outputs of retrained models against real pool datasets; ensuring inference latency remains ≤4.5 ms on the Jetson Orin Nano.	Continuous (Regression Testing via Rosbags)
Mission Plans	Verification of the 10 Hz py_trees behavior tree, capability gates, and the 6-second drift-budget decorator.	ROS-wrapper mock integration testing against 5 scripted scenarios (e.g., late detection, mid-task occlusion) in <2 minutes, followed by HoloOcean SITL requiring >80% completion rate across 10 randomized poses.	Phase 2 (Mock Node) & Phase 2.5 (HoloOcean SITL)
Network and Telemetry	Validation of Eclipse CycloneDDS [12] middleware	Dry-lab boot sequence confirming all 12 packages start without DDS errors; verifying BEST_EFFORT QoS	Phase 3a (Pre-Pool Bench Integration)

Component / Capability	Description	Methodology	Timeline
	discovery and explicit Quality-of-Service (QoS) profile matching across nodes.	for sensor topics and RELIABLE QoS for control setpoints to prevent stale commands.	
Actuation	Testing the custom Moore-Penrose thrust allocator (mapping a 6-vector body wrench to 8 thrusters) and payload trigger commands.	Props-off bench test to confirm MAVROS RC override messages reach the Pixhawk v6x at exactly 20 Hz, verifying proper PWM output scaling via ArduSub's custom mixer before in-water use.	Phase 3a (Pre-Pool Bench Integration)

Table C.I: Component-by-Component Software Testing Plan

C.2. Continuous Integration and Regression Pipeline

To prevent new code from breaking validated vehicle behaviors, AquaCore implements a strict Continuous Integration (CI) gateway. Every mathematically intensive module—specifically the 15-state Error-State Kalman Filter (ESKF), the thrust allocator, and the trajectory planner—is constructed with zero ROS dependencies to allow rapid isolated testing.

Before any code is merged into the main deployment branch, a CI hook enforces a pytest suite execution. For example, the test_eskf_core.py suite validates static convergence, quaternion math edge cases, and yaw-rate integration, executing in under 0.8 s on the Jetson Orin Nano. Furthermore, when the YOLOv8m perception model is retrained, it is automatically evaluated against a stored corpus of LFS-tracked .rosbag pool recordings. This ensures detection consistency over verified datasets before the model is permitted on the physical vehicle.

C.3. Bounding the Sim-to-Real Gap

Because synthetic visual environments often fail to accurately replicate real pool turbidity and lighting caustics, our software testing strategy strictly divides simulation and perception validation. The robosub_holoocean ROS 2 bridge is utilized exclusively as a logic and control regression layer.

Five scripted YAML scenarios (gate, slalom, bins, torpedoes, octagon) test the py_trees mission planner and the 6-DOF PID responses against simulated hydrodynamic inertia. However, the YOLOv8m perception models are explicitly not validated in HoloOcean. Instead, perception is validated by streaming real pool video recordings—captured as .rosbag files during in-water testing—back through the Jetson's TensorRT pipeline. This bifurcation ensures our control logic scales efficiently while our computer vision remains firmly anchored in physical reality.

C.4. Software Failure and Remediation Log

Embracing a "graceful degradation" doctrine requires rigorous documentation of system failures. The following log details critical software vulnerabilities identified during Phase 3 integration testing and their subsequent architectural resolutions:

Vulnerability 1: Perception System Inference Instability

- *Observation:* Initial in-water testing utilizing COCO-pretrained weights yielded intermittent bounding box dropouts on the primary gate divider. This instability falsely triggered premature mission aborts within the behavior tree framework.
- *Architectural Resolution:* A deterministic classical-vision fallback mechanism was integrated into the auv_perception node (reference Fig. 18). If the primary YOLOv8m neural network fails to report valid detections for ≥ 1.0 s, the pipeline automatically degrades to a computationally lightweight HSV color-thresholding and contour-analysis loop optimized specifically for the red gate divider.

Vulnerability 2: Unbounded Open-Loop Translation Drift

- *Observation:* In the absence of a Doppler Velocity Log (DVL), open-loop translations generated unbounded positional drift, causing the vehicle to severely deviate from its designated search vectors.
- *Architectural Resolution:* A strict "Drift Budget" contract was codified within the trajectory planner. As depicted in the Mission Planner layer of Fig. 18, every blind motion mode is now wrapped in a py_trees decorator that forcibly interrupts actuation after 6 seconds. This intervention asserts a FAILURE state, actively triggering recovery branches (e.g., visual yaw sweeps or controlled resurfacing) to re-establish absolute localization.

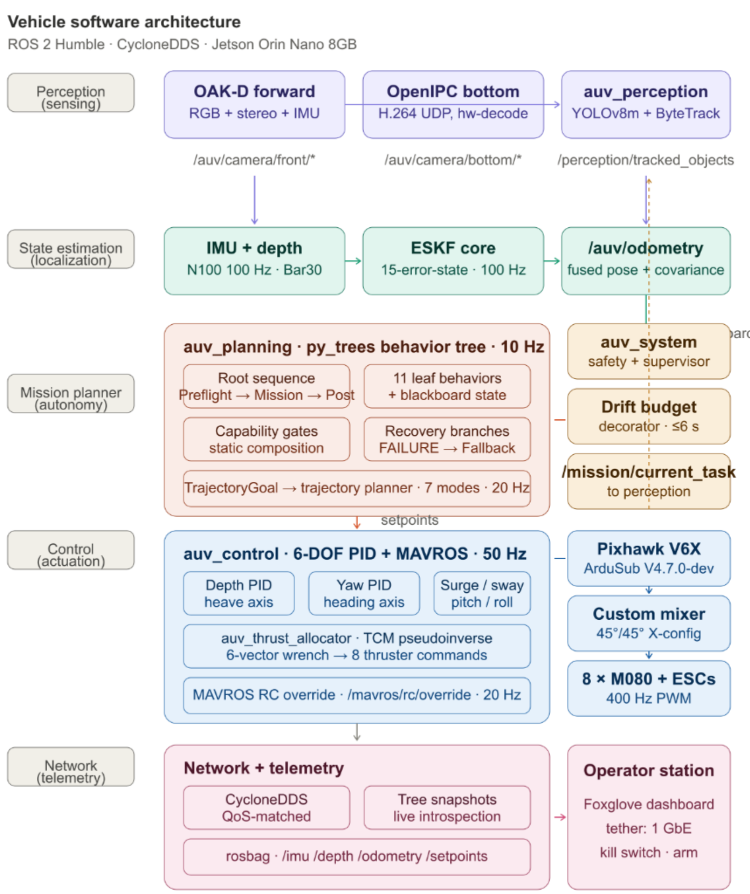

Fig. 07: Mission Gate Detection via AUV Vision System

APPENDIX D — POOL TESTS AND STRATEGY ADJUSTMENTS

Pool testing was conducted primarily at the Waterpolo BD swimming facility near United International University. The controlled testing environment allowed the team to evaluate vehicle maneuverability, navigation control, payload operation, and underwater perception performance under realistic operating conditions. Regular testing sessions were used for subsystem integration, driver practice, mission rehearsals, and validation of onboard sensing systems. Equipment transported to each session included the operator control station (OCS), monitoring laptop, battery charging setup, networking hardware, torpedo test props, spare components, and essential mechanical and electrical hand tools. Prior to each session, the team conducted a structured briefing to define testing objectives, assign operational responsibilities, and review deployment procedures. Test runs focused on navigation precision, underwater target visibility, payload handling, and stable vehicle operation during mission-oriented maneuvers. Environmental conditions including lighting variation and water temperature were logged during every session. Variations in lighting significantly affected onboard camera visibility and task recognition performance, while water temperature influenced battery behavior, sensor consistency, and hull sealing reliability. Maintaining these records improved repeatability between tests and supported more consistent performance evaluation.

Fig. 08: Torpedo board used for pool tests.

Fig. 09: Software team members during pool testing.

APPENDIX E — ARCHITECHTURE BLOCK DIAGRAMS

The following diagrams map the core system topologies of the AquaCore Gen 1 vehicle, detailing the strict architectural boundaries across our power distribution networks and communication buses.

Description: Vehicle power architecture illustrating the isolated load distribution from the central battery pack and regulators through to the ESCs, compute modules, sensor suite, and payload servos.

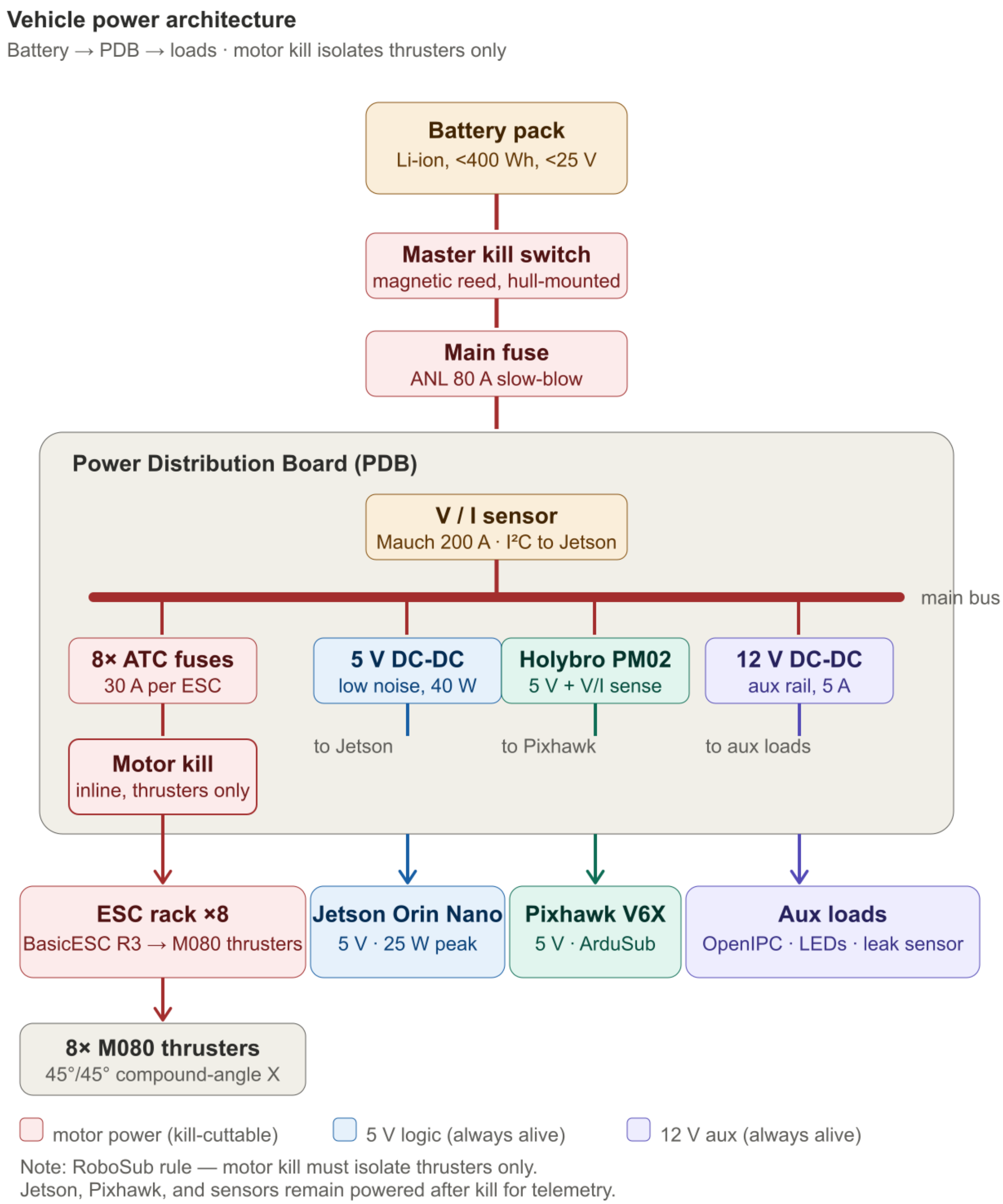

Fig. 10: Vehicle Power Architecture

Description: Vehicle communication architecture detailing the bus topology, MAVLink UDP bridges, and ROS 2 DDS network interfaces between the Single Board Computer (SBC), low-level flight controller, ESCs, and operator station.

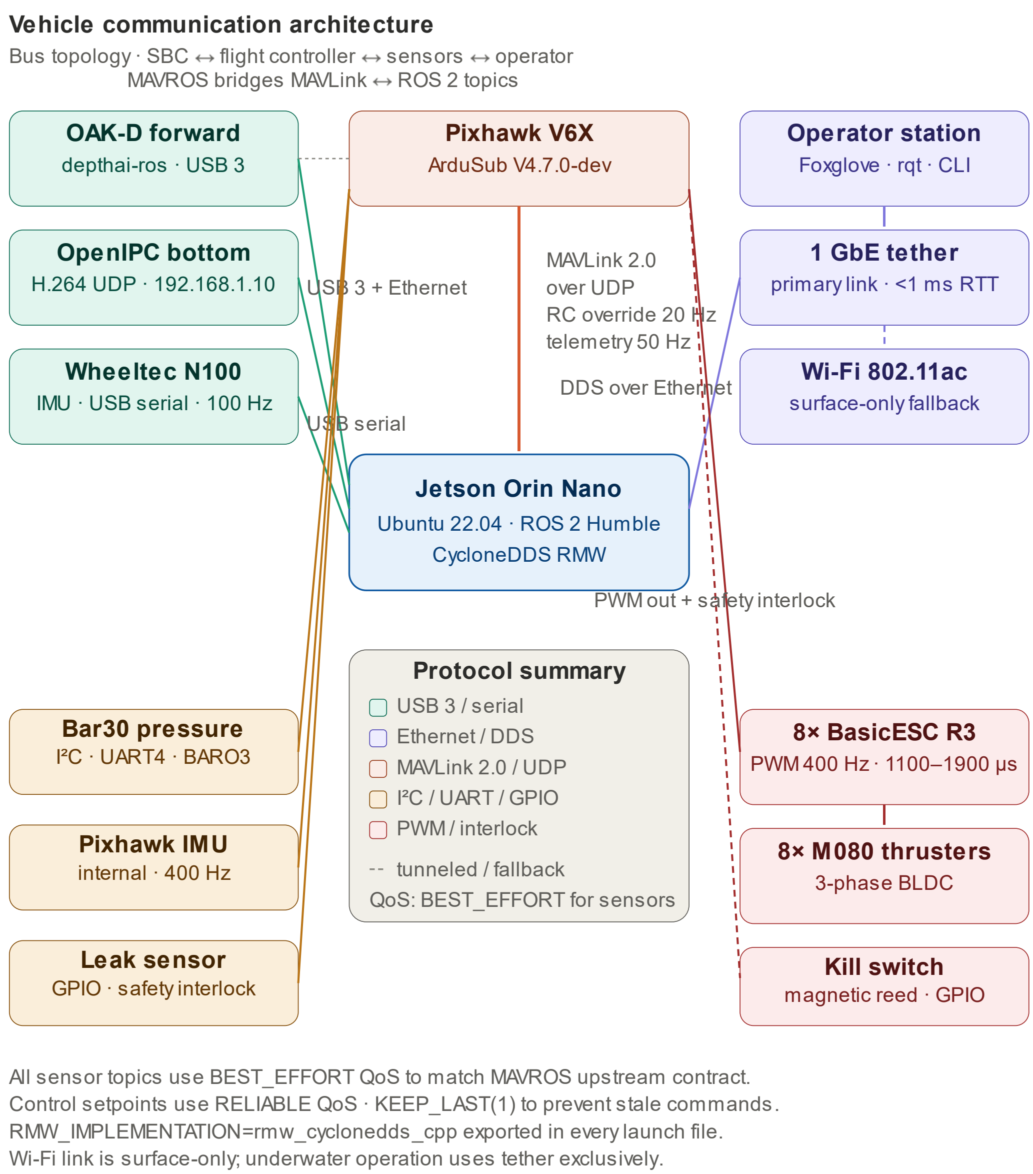

Fig. 11: Vehicle Communication Architecture

APPENDIX F — COMPONENT SPECIFICATIONS AND PROJECT BUDGET

F.1. Operational Vehicle Component Expenditure

All values in USD. Components listed below are the procured units for the operational vehicle. Quantities reflect single units unless otherwise specified.

Component	Vendor	Model / Type	Specifications	Custom/ Purchased	Cost (USD)	Year
AUV Hull (Main)	Custom	Carbon fiber tubing	Main pressure hull + 6 buoyancy tubes	Custom	$989	2026
Acrylic Dome	Rovmaker	Optical Dome	Optical-grade camera dome	Purchased	$92	2026
O-ring Flange	Rovmaker	Flange Seal	End-cap waterproof seal	Purchased	$59	2026
Aluminum Sheet	—	Sheet stock	Structural ring stock	Custom	$30	2026
Acrylic Sheet	—	Sheet stock	Secondary structural plates	Purchased	$50	2026
Filament	—	PETG + ABS	3D-printed payloads and brackets	Custom	$208	2026
CNC Machining	In-house	Custom Aluminum Milling	Structural ring fabrication	Custom	$30	2026
Cable Penetrators (M16)	Rovmaker	M16 series	Large-conductor hull penetrators (×8)	Purchased	$14.80	2026
Cable Penetrators (M10)	Rovmaker	M10 series	Medium-conductor hull penetrators (×10)	Purchased	$8.80	2026
Cable Penetrators (M8)	Rovmaker	M8 series	Small-conductor hull penetrators (×8)	Purchased	$5.90	2026
Underwater Switch	Rovmaker	M10 magnetic	Kill switch	Purchased	$25	2026
Waterproof Servos	Savox	Savox SW-1210SG	Payload actuators	Purchased	$60	2026
Thrusters	—	M080	Brushless propulsion units (×8)	Purchased	$233	2026
Thruster Control	—	60 A ESC	Brushless electronic speed controllers (×8)	Purchased	$38	2026
Battery	—	S50 cells	Custom 24 V, 15 Ah, 6S Li-ion pack (40 cells)	Custom	$12.50	2026
Battery Monitoring	—	BMS 60 MPR	Battery protection circuit	Purchased	$25	2026
Microcontroller	CUAV	Pixhawk V6X	Low-level flight controller (STM32H753)	Purchased	$342	2026
Single Board Computer	NVIDIA	Jetson Orin Nano	High-level compute: AI + vision	Purchased	$508	2026
Inertial Measurement Unit	Wheeltec	N100	9-axis MEMS IMU, 100 Hz	Purchased	$195	2026
Depth Sensor	Blue Robotics	Bar30	High-resolution pressure sensor	Purchased	$46	2026

Camera (Forward)	Luxonis	OAK-D Pro	Stereo depth + onboard AI perception	Purchased	$610	2026
Camera (Surface link)	RunCam	WiFi link 2	Surface telemetry camera	Purchased	$86	2026
Network Switch	TP-Link	Ethernet → USB	Tether interface bridge	Purchased	$15	2026
Doppler Velocity Log (DVL)	—	—	Not equipped on this vehicle	—	—	—
Hydrophones	—	—	Not equipped on this vehicle	—	—	—
Internal Comm. Network	—	Ethernet / UART / I²C	1000 Mbps + MAVLink over USB-serial	Custom	—	—
External Comm. Interface	—	Gigabit Ethernet	1000 Mbps Cat6 tether	Custom	—	—
Algorithm: Vision	Ultralytics	YOLOv8m	TensorRT FP16 quantization on Jetson	—	—	—
Algorithm: Localization	Custom	15-state ESKF	Pure-Python core, ROS 2 wrapper	Custom	—	—
Algorithm: Perception Aux	OpenCV	HSV thresholding	Graceful-degradation fallback	—	—	—
Software: Autonomy	—	py_trees	Behavior-tree mission planner	—	—	—
Software: Framework	Open Robotics	ROS 2 Humble	Middleware: Eclipse CycloneDDS [12]	—	—	—
Software: Inference Runtime	NVIDIA	TensorRT FP16	YOLOv8m inference engine	—	—	—
Team size	—	—	14	—	—	—
Hardware / Software Expertise Ratio	—	—	1:4	—	—	—
Testing time: Simulation	—	—	10 hours	—	—	—
Testing time: In-water	—	—	25 hours	—	—	—
Subtotal (operational vehicle)					~$6,290	

Table F.1: Operational vehicle component expenditure.

Notes: BDT-priced items converted at 120 BDT per USD. Cells marked with Em-dash (—) indicates not applicable or not equipped.

APPENDIX G — OUTREACH ACTIVITIES

Project Poseidon's outreach strategy is designed to extend the impact of our marine robotics development far beyond the competition environment. As part of the broader Bangladesh AUV initiative, our primary long-term objective is to promote STEM accessibility in underserved regions. By introducing hands-on exposure to robotics and engineering to students with limited access to technical education resources, we aim to build a sustainable pipeline of future innovators.

A. Public Knowledge Exchange and Media Outreach

To establish early national awareness and inspire broader community engagement, our team initiated a public outreach campaign utilizing national mass media platforms. This included comprehensive television news coverage and interactive broadcasts on national radio (FM 88.6). During these sessions, team members discussed the constraints of limited engineering resources in developing contexts and advocated for the use of autonomous marine systems as practical, scalable solutions to real-world national challenges.

Fig. 12: National television news broadcast featuring the team's AUV initiative.

B. Community Advocacy and Inspiration

Our public messaging is strategically designed to motivate young learners by demystifying complex technology. We actively position robotics not as an exclusive or distant academic field, but as an accessible platform for grassroots innovation and problem-solving. By presenting our work on the AquaCore underwater robotics stack to the general public, we foster early curiosity and encourage sustained interest in applied engineering disciplines.

Fig. 13: Team meeting with the Principal of Dhaka Residential Model College.

Fig. 14: Advocacy meeting with the former State Minister for ICT.

C. Training and Mentorship Pipeline

To ensure the sustainability of the team and the continuous development of the AUV platform, our internal recruitment pipeline focuses on bridging the gap between theoretical education and applied engineering. New members are introduced to the vehicle's architecture through hands-on troubleshooting and iterative design tasks, ensuring they understand the practical realities of autonomous systems, hardware constraints, and technology-driven problem-solving before taking ownership of major subsystems.

D. Rural STEM Initiatives and School Collaboration

While formal rural STEM engagement programs are currently under active development, the team is designing structured initiatives targeted at local high schools and underserved communities. These upcoming programs comprise interactive workshops, live robotics demonstrations, and direct mentorship initiatives. The core objective is to take the technology out of the university lab and into rural classrooms, providing younger students with direct exposure to marine robotics and autonomous logic.