# Team Caleuche Technical Design Paper for RobotX 2026

Rodrigo Aguilera, Gabriel Aldunate, Sebastián Arzola, Matías Azócar, Nicolas Azócar, Pedro Bahamonde, Joaquín Bocaz, Ignacio Campillay, Martin Castro, Joaquín Coñomán, Sebastián Cornejo, Mario Flores, Ignacio Gajardo, Leandro Garrido, Raimundo Gazzolo, Jaime Gibson, Felipe Gonzáles, Daniel Haddad, Joaquín Handal, Ricardo Indo, Linn Karlsen, Camila Lara, Agustín Llambías, Tomas López, Sebastián Madrid, Lucas Martin, Vicente Merino, Oscar Merino, Anatole Monroy, Benjamín Moreno, Juan Moya, Ixchel Nasser, Vicente Navarrete, Héctor Núñez, Fiorenza Oyanedel, Maximiliano Pisfil, Benjamín Reyes, Ignacio Segura, Pablo Solar, Benjamín Ulloa, Martin Valdes, Lucas Zúñiga, Christopher Rhomberg.

***Abstract*—Team Caleuche, from the Pontificia Universidad Católica de Chile, competes in RobotX 2026 with three vehicles: Caleuche II 1.0, a BlueBoat surface vehicle; Cochayuyo 1.0, a BlueROV2 underwater vehicle tethered to it; and Chucao 2.0, a custom aerial vehicle. After reaching the semifinals in 2024, the team found that its main weakness was not the number of capabilities it had developed, but how little they had been tested on the water. The 2026 system therefore favors reliability over additional capability, targeting the Core and Advanced tiers of all four tasks.**

**Two main rules shaped system design: first, existing commercial or open-source solutions were used wherever they met a requirement, and second, any new component had to serve more than one task. This includes using vehicles and the modular systems carried over from 2024. The design strategy focused on cooperation between vehicles and combining sensor data to enhance precision, robustness and versatility. Each capability progresses from simulation to bench, integrated and water testing, and the system has logged 110 hours of flight and water testing combined.**

## I. Competition Strategy

Team Caleuche's approach to 2026 was shaped by three factors: the platforms the team had to adopt, the limited hours available to develop and test them, and what the 2024 edition revealed about the team's weaknesses.

The new USV and weight limits and the addition of the underwater domain meant that the WAM-V used in previous editions could no longer compete, and the team did not have the funds to build or buy new vehicles. The team therefore applied to the granted platform program and received a BlueBoat and a BlueROV2, which arrived in March 2026. Starting from commercial vehicles with proven hulls and propulsion meant that very little development time had to be spent on the platforms themselves, and the team extended that principle to the rest of the system. Wherever an established framework or vendor solution met the requirements of a task, it was adopted, and the team's main effort went into integrating the three vehicles and developing the autonomy the tasks require.

Changing platforms did not mean starting from zero. Since 2024, the team has designed its subsystems as modules that are not tied to a specific hull [1], and most of them transferred from the WAM-V to the BlueBoat with only minor adaptations. The sensor platform keeps similar sensor positions and structure established in 2024. The electrical system keeps the 2024 architecture, with pelican-enclosed distribution, DC converters and breakers, and the LoRa-based remote emergency stop. The ROS software stack and the VRX-based simulation workflow carried over as well. This reuse is what made a competitive system feasible within a development window that opened only in March.

In 2024 the team reached the semifinals. The post-competition review showed that the team's main limitation was not the number of capabilities it had developed but how little those capabilities had been tested on the water. The

2024 development cycle produced one water test day every three weeks. For 2026, the team therefore traded capability for reliability: fewer tiers targeted and more frequent water testing, achieving at least one field test per week.

### *A. Task Selection*

The team targets the Core and Advanced tiers of every task. The Disruptive tiers were not targeted, because responding to changes introduced during a run, such as dynamic safe passages or a moving obstacle, requires a level of reliability that only extended water testing provides.

Within that target, development was ordered by how many tasks a capability serves. The UAV is required in every task, and because the UUV is tethered to the USV, the surface vehicle takes part in all four as well. Capabilities on these two vehicles, including perception, navigation, communication and actuators, were developed first, since each contributes to every task. Capabilities used in a single task were developed afterward.

| Capability | Confidence (0-10) | Tested up to |
|---|---|---|
| **Navigation** | | |
| Waypoint navigation | 7 | Water |
| Obstacle avoidance | 6.5 | Water |
| Station keeping & docking | 6 | Simulation |
| UUV relative localization | 6.5 | Simulation |
| **Perception** | | |
| LiDAR buoy detection | 8.5 | Water |
| Aerial buoy mapping | 6 | Flight |
| Dock detection | 7 | Simulation |
| Platform detection | 6 | Water |
| Pinger localization | 5 | Simulation |
| **Actuation** | | |
| Water delivery | 2 | Not tested |
| Magnetic probe | 1 | Not tested |
| Tin pickup & delivery | 4 | Bench |

Table 1. Capabilities confidence level

### *B. Complexity and Reliability*

With limited development time, every hour spent building a new component was an hour not spent testing an existing one on the water. Throughout the cycle, the team chose reliability over additional capability, and simple, low-cost solutions over complex ones. This trade was made sharper by a development window that began only when the vehicles arrived in March 2026 and by remodeling of the team's lab, which limited in-person work for most of the cycle.

Two rules kept the system's complexity under control. First, the team adopted before building. The vehicles, the navigation framework and the UUV's tethered configuration were all taken from established vendor or open-source solutions, and custom development was reserved for integration and autonomy, where no alternative solution existed. Second, a new capability or piece of hardware had to serve more than one task. The acoustic array on the USV locates the Task 2 pinger and tracks the UUV. Development effort was concentrated on the USV and UAV because together they take part in every task.

By not attempting the Disruptive tiers, the team gives up the points available at that level in every task. The single-task capabilities developed last, such as the water delivery and UUV magnetic probe, have been validated only in simulation and on the bench, so their first water tests will take place in October the 3rd. Keeping the UUV tethered limits it to operating within 6 m of the USV, so the surface vehicle must accompany it throughout Task 2.

## II. Design Strategy

### *A. Design Philosophy*

The design aims for subsystems that are simple, modular and low-cost. Commercial and open-source products are adapted where possible instead of building custom ones, and hardware is built from standard parts that can be replaced or modified without redesigning the vehicle. Subsystems are not tied to a specific hull, which let most of the 2024 system move to the BlueBoat. Custom development focused on the USV and UAV, and the UUV is kept close to its stock configuration.

The software on all three vehicles is split into ROS 2 packages with one responsibility each,

such as mission management, navigation and perception. Low-level hardware control runs separately from mission logic and vision processing, so a delay or failure in one package does not stop the others. This structure also lets the subteams develop in parallel and add new capabilities without changing tested code.

To achieve this, the software design follows the SOLID principles of object-oriented and system-level design [2].

Each node handles a single task, which isolates failures and simplifies testing. New capabilities are added as new nodes rather than by changing tested code, and nodes subscribe only to the ROS 2 messages they need. Mission logic depends on hardware-independent interfaces rather than specific sensors or actuators. The UAV perception pipeline (Fig. 1) shows this: uav_perception defines a Camera Base and a Detector Base, so any camera (OAK-D, ROS topic or TCP stream) and any detector (color, ArUco or SIFT) can be swapped at runtime, and uav_mission_manager receives the same detection format regardless of which is used.

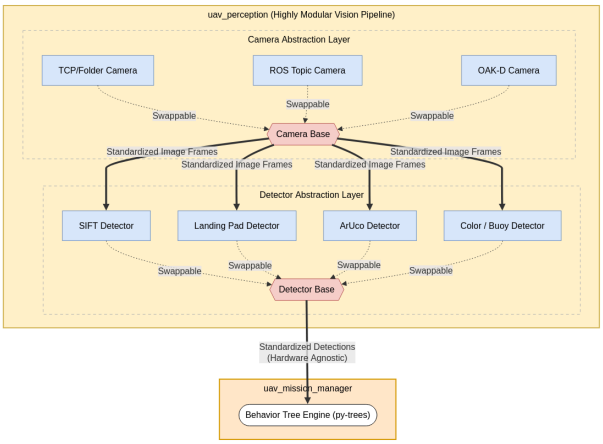



Fig 1. UAV Software Stack

## *B. Systems Engineering*

The team's development runs on a weekly cycle. Subteams work on their designs during the week, and every Wednesday a general meeting attended by all three subteams serves as the design review. The review is where designs are approved, requirements are set and, since the team does not maintain a formal interface document, interfaces between subsystems are agreed. When a water test is scheduled, the rest of the week follows the bench deadlines and Friday freeze described in Section III.

Each design starts within its subteam, where the first question is whether an existing commercial or open-source solution meets the need; custom development begins only if none does. The first draft is presented within the subteam, then at the general meeting as a preliminary design review (PDR), with slides, CAD, simulation results and an itemized parts list with costs. A design is approved only when the other subteams accept it, since each has to power, mount or control what the others build. If the review finds problems or missing features, these become requirements, the work is redistributed, and the design returns with a deadline for its next review. A design rejected outright is dropped and its members move to other work. Significant purchases are approved only after the design they depend on passes review.

Approved work is tracked in GitHub Projects, and software follows a Gitflow workflow in which pull requests are reviewed and pass several checks and simulation tests before merging. A design is finished only after passing its assigned tests (Section III). Since many members are new each cycle, tutorials and welcome events prepare them before they join the review process.

The same process addressed the weaknesses found in 2024. The single computer failure that left the USV without perception led to independent behavior trees on each vehicle, monitored by a heartbeat, and to failover between the USV's onboard computers, currently in development. Midpoint navigation without obstacle detection was replaced by Nav2 [3], and water testing only once every three weeks led to the fixed test schedule in Section IV.

## *C. USV - Caleuche II 1.0*

The first iteration of the USV retained a very similar model as the original WAMV. Due to the weight restrictions of the platform a decision was made to reduce the size of the onboard Pelican Case by including most of the vehicle’s

equipment inside its hull. As such, only the sensors and power distribution box remained outside.

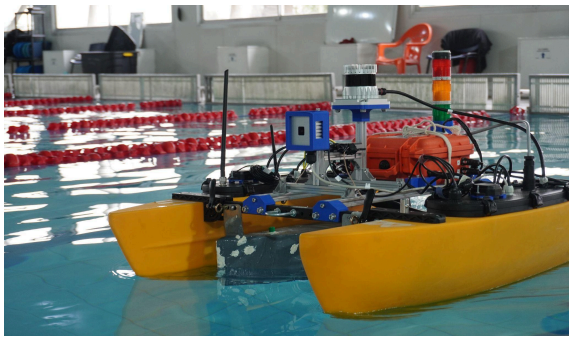


Fig 2. USV

### 1) *BlueBoat Modifications*

The main challenge for migrating from WAM-V to BlueBoat was the high difference of payload between both vehicles, passing from over 200 kg to just 15 kg, so the last competition's system had to be changed in line to reduce its weight. To adjust that limitation, the team developed a payload extension system, which added close to 20 kg of load capacity to the USV, consisting of a set of floats made of fiberglass reinforced with polyester, with a hydrodynamic profile to reduce the possible drag force on the boat.

Another modification was the development of a structure to integrate the LiDAR, PoE camera, GNSS/GPS system, status light and Pelican 1200 electronics enclosure within the vehicle's payload and space constraints. The design uses 20×20 mm aluminum T-slot profiles and PETG mounts, preserves the sensor positions established by the team, and allows longitudinal and transverse adjustment without modifying the main frame

In line to complete Mission Task 3, the team designed a water cannon based on a modified wireless pressure washer with a remotely actuated trigger, mounted on the USV's sensor tower with two degrees of freedom—one tilt axis (0°–45°) and one horizontal pan axis (75°–105°). The mechanism is driven by two NEMA 17HS4401S stepper motors controlled by A4988 drivers and an ESP32 microcontroller, which receives targeting commands from the main USV control system via serial UART, operating through a mechanical reduction system. This configuration was chosen to hold the water stream steadily on target for several continuous seconds, as required by the task.

The final system is 0.93 m wide, 1.2 m in length and 0.85 m in height, weighing approximately 35 kg using all its modifications.

### 2) *Power and Safety*

The team built the new power and distribution architecture within the BlueBoat's pre-existing electrical system rather than replacing it outright, adapting the original wiring to the team's requirements. At the center of this architecture is a distribution box housed in a Pelican 1200 case, which grouped 24 V and 12 V converters, DC circuit breakers for overcurrent protection, and terminal bus bars, from which power is distributed to the boat's sensors and computers. Waterproof LP12 connectors and PG9 cable glands sealed each connection point entering the hull and floats, with ferrule-terminated cabling secured to the internal distribution bars to keep the whole power path protected against water ingress. 5V and battery voltage entered the pelican from the already existing Blueboat's fuse board.

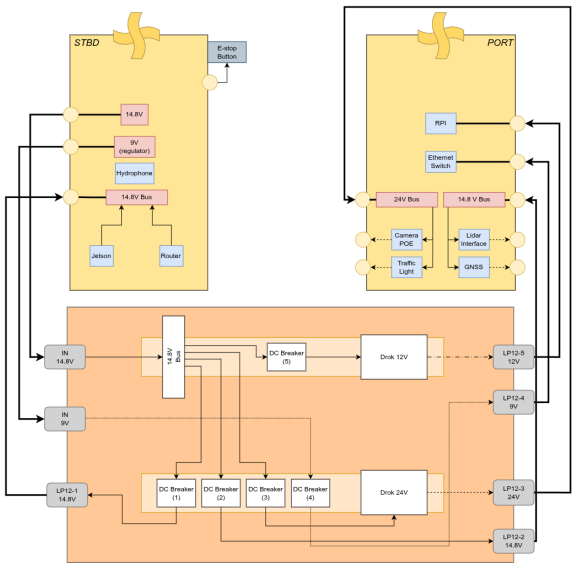



Fig 3. Power distribution diagram for the Caleuche USV

Electrical safety relies on two independent, redundant emergency-stop paths that act exclusively on the actuators, without cutting power to the computers or sensors. The physical e-stop uses a normally closed, manually resettable switch with an accessible red button on the starboard side, which directly interrupts actuator power. The remote e-stop relies on two normally open solid-state relays controlled over a 920 MHz LoRa link, backed by a dead-man's-switch failsafe that automatically kills the actuators if the wireless connection was lost. A traffic light status light turns red whenever either e-stop was triggered.

*3) Localization*

The design strategy for vehicle localization can be understood as a combination of two separate processes: internal device odometry (implemented as a /odom to /base_link transform) and global pose estimation (/map to /odom transform). This localization architecture is designed to be modular and scalable, with distinct implementations for each vehicle in accordance to their system requirements and their available sensor inputs.

USV local and global odometry are obtained through sensor fusion in Extended Kalman Filters (EKF) implemented with the ROS2 robot_localization package, especially with the ekf_local, ekf_global and navsat_transform_node. The latter, is configured to receive a datum input, which fixes the pose of the map frame in relation to world coordinates.

*4) Perception*

Strategic sensor fusion is at the core of Team Caleuche perception team design strategy. By computing intrinsic and extrinsic parameters through camera calibration, Camera-LIDAR sensor fusion is achieved by projecting the 2D image topic onto the 3D point cloud, and vice versa. This enables color-based filtering used for dock detection.

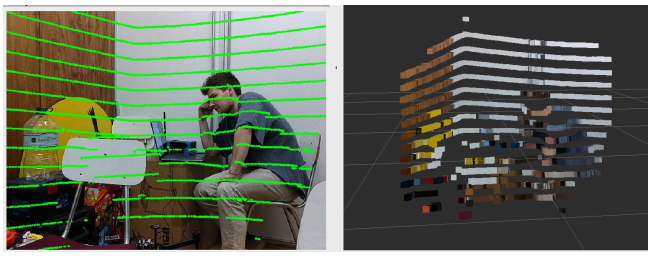


Fig 4. Camera-LIDAR sensor fusion

To tackle the infrastructure survey and repair task, a heterogeneous perception architecture for buoy localization. In this system, the acoustic array serves as a long range coarse orientation detector, while the LIDAR sensor provides a fine grain metric of buoy pose. The implemented acoustic processing algorithms employ TDOA (Time Difference of Arrival) technique to resolve angular direction. Utilizing an array of 4 hydrophones, results yielded an error of 0.12 m and a 5.59° when testing on simulation. Further testing will be conducted once the newly purchased hydrophones and pinger arrive.

*5) Navigation*

The USV navigation architecture prioritizes high-level control and operational reliability, governed by a strict mandate to leverage proven, resilient algorithms rather than redundant ground-up development. Following an analysis of high-performing Technical Design Reports from the RobotX 2024 competition, the USV navigation stack was transitioned to the Nav2 framework. As a production-grade autonomy system, Nav2 abstracts complex local reactive behaviors [3]. To achieve this, the team utilizes the Spatio-Temporal Voxel Layer (STVL) [4] plugin, which seamlessly integrates continuous onboard 3D sensor data to handle dynamic buoy avoidance. This delegation of low-level, reactive collision avoidance to a highly trusted framework enables the overarching path-planning module to operate strictly within global waypoint designations, significantly streamlining USV mission execution.

A key feature of Nav2 that the team leverages for this competition is the ability to designate keep-out zones. This feature provides a standardized way to handle the static keep-out

area for the Advance tier and the dynamic keep-out zone for the Disruptive tier.

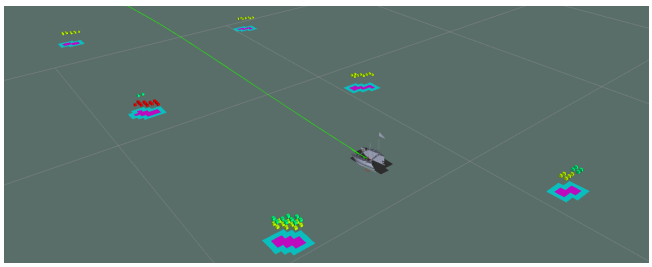


Fig 5. USV Nav2 mapping and obstacle avoidance.

Buoys detected by the UAV are paired into gates and ordered along the USV's direction of travel. To choose a pairing method, a synthetic map generator with known ground truth was built to reproduce eight conditions: straight, curved, S-shaped and tightly turning corridors, high positional noise, missing detections, long corridors, and a combined worst case. A greedy nearest-neighbor method and the Hungarian optimal-assignment algorithm were compared on thousands of maps per condition. The two performed the same in most conditions. The Hungarian algorithm minimizes total pairing distance across all buoys, which made it more reliable under positional noise (98.8% vs. 97.6% success). The greedy method commits to unambiguous pairs first and never revisits them, which made it slightly more reliable with missing detections (92.7% vs. 92.5%). On the water, wave motion, UAV drift and camera pose error are expected to cause positional noise more often than missed detections, so the Hungarian algorithm was selected.

### *D. UUV - Cochayuyo 1.0*

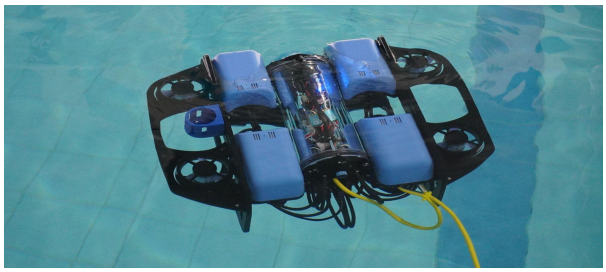


Fig 6. UUV

#### *1) Navigation*

To establish a robust underwater autonomy framework, the team implemented a comprehensive control and mission architecture for the BlueROV2 Heavy platform. The development focused on enabling closed-loop underwater navigation, RViz visualization, and mission-level sequencing driven by behavior trees.

In alignment with the project's overarching architectural philosophy, which prioritizes separating high-level decision-making from low-level stabilization—all low-level hydrodynamic control and motor mixing are delegated directly to the onboard BlueOS firmware. By offloading these high-frequency stabilization tasks, the ROS 2 companion stack is freed to employ a strictly waypoint-based navigation strategy. This approach is rigorously validated in a Gazebo-based BlueROV simulation equipped with ROS 2 bridge interfaces, ensuring reliable translation of high-level objectives into executable trajectories.

Crucially, the perception stack utilizes AprilTags not only for identifying environmental targets but also for the critical task of UUV-USV relative pairing. By detecting these tags, the UUV establishes a reliable local reference frame relative to the surface vessel, allowing it to continuously update and compute task-specific waypoints. This decentralized, waypoint-driven architecture enables the UUV to autonomously execute complex, multi-stage underwater navigation routines—such as traversing submerged structures or circumnavigating obstacles—with high precision and spatial awareness.

#### *2) Localization*

UUV localization posed one of the greatest challenges for inter-vehicle cooperation. Unlike the UAV, the Bluerov2 UUV does not have in-built odometry estimation, nor does it have enough sensor information to guarantee a drift-free estimate. Therefore, it is required for /map to /odom transform to be able to compensate for it. Two distinct localization strategies are

under development. The first, is hydrophone and pinger-based pose estimation, which employs a pinger mounted on a UUV and hydrophones to estimate its pose. Considering a worst-case scenario where no hydrophones or pinger are available, the UUV will instead use an apriltag on the USV to compute its relative pose. To further improve device odometry, ORB-SLAM is being implemented to reduce horizontal pose ambiguity.

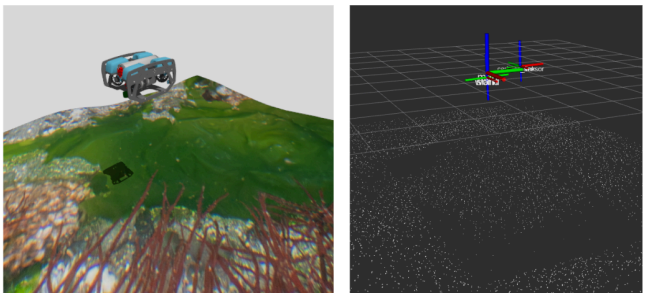


Fig 7. Orb-SLAM tested on Stonefish simulator

### *E. UAV - Chucao 2.0*



Fig 8. UAV

#### *1) UAV Claw*

To facilitate the autonomous retrieval and deployment of small ferromagnetic payloads, a modular magnetic pickup mechanism was designed for the UAV. Manufactured via PETG 3D printing, the end-effector incorporates a dual-electromagnet array capable of securely lifting the target objects.

To evaluate the structural robustness of the design, Finite Element Analysis (FEA) was performed under various impact conditions representing potential extreme scenarios during real-world operation. The results demonstrated that the structure withstands the evaluated loading conditions without failure. The analysis indicated a minimum safety factor of 1.84, which occurs only in highly localized stress concentration regions, while the majority of the component exhibits significantly higher structural margins, reaching safety factors up to 15.

Furthermore, to ensure effective and reliable contact between the electromagnets and the payload in dynamic environments, a passive damping system was integrated using a PETG housing and a 3D-printed TPU spring. During the retrieval sequence, the spring compresses to absorb impact forces and maintain continuous surface contact. This compliance allows the electromagnets to securely capture the payload during the final descent phase, without requiring the UAV to fully touch down, thereby increasing flight safety. Finally, a custom mount was integrated into the upper section of the mechanism to accommodate a Logitech HD270 camera. Positioned directly above the actuation zone, this sensor provides a top-down view of the workspace, allowing the onboard perception system to visually verify successful payload capture and significantly improve overall manipulation reliability.

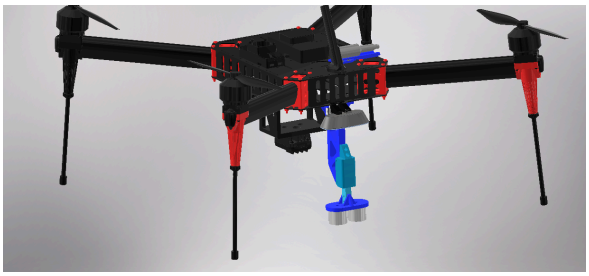


Fig 9. UAV Claw 3D model

#### *2) Localization*

The Team Caleuche UAV vehicle utilizes PX4 firmware which contains in-built EKFs for local and global odometry estimation. Internal sensor fusion also receives GPS inputs, reducing horizontal odometry drift. For global localization, the map datum is fixed to match that of the USV.

*3) Navigation*

While the overarching project philosophy favors standardized frameworks like Nav2 [2], the architecture remains adaptable when standard solutions fail to meet specific operational constraints. Extending the Nav2 paradigm to the UAV's three-dimensional operational domain was initially considered but ultimately discarded due to a lack of mature, easily integrated volumetric alternatives that matched the performance of the existing systems. Alternative perception-based approaches, such as ORB-SLAM, were evaluated to enhance spatial awareness. However, empirical testing demonstrated that the maritime environment inherently lacks stable visual keypoints, leading to frequent tracking failures when the UAV operates at a distance from static references like the USV.

Consequently, the UAV's challenging maritime domain demands a highly reliable, customized flight control strategy. The system relies on a highly optimized, waypoint-based navigation stack located in the uav_navigation package. This architecture leverages ROS 2 action servers to seamlessly process high-level mission commands, such as automated takeoffs, search patterns, and point-to-point traversals.

These behavioral commands are then translated into executable MAVLink instructions via a dedicated PX4 interface using the uXRCE-DDS (MicroXRCE) middleware as a low-latency bridge. By delegating low-level, high-frequency flight stabilization directly to the PX4 v1.14 firmware, the onboard companion computer is freed to focus exclusively on executing complex path planning. This bifurcated approach effectively mitigates positional drift caused by wind and wave motion, maximizing overall flight stability while ensuring precise and resilient execution of the waypoint sequences designated by the mission manager.

*4) Perception*

The UAV color-space buoy detector serves as an extension to the USV's spatial understanding. This is achieved by utilizing a ray-casting projection that maps UAV 2D observations onto the USV's reference frame. These detections are then fused with USV LIDAR for persistent map construction. As part of the design strategy for this task, perception and the exploratory waypoint generation were designed jointly.

Additionally, YOLO (You Only Look Once) computer vision algorithms are used to achieve robust and effective platform detection from the UAV camera. Initial results with a dataset of only 100 images yield confidence metrics in the range 70-90% on gazebo simulation. Furthermore, a platform pose estimation feature is underway, which will further aid in drone control for resource delivery.

*F. System of Systems*

Each vehicle is designed to operate on its own, and cooperation between them is added on top. The vehicles share only the information another vehicle needs: the UAV sends buoy detections to the USV's map, and the USV provides the UUV's position relative to itself. Each vehicle runs its own software stack, ROS domain and decision-making. A lost link therefore reduces what the team can do in a task, but it does not stop the vehicle that lost it.

*1) Inter System Communication*

Because ROS 2 nodes rely on multicast for discovery, placing all nodes from the three systems within a single ROS domain generated excessive network traffic, resulting in increased latency and packet loss [5]. To address this limitation, each system was assigned a separate ROS domain, since only a small subset of topics requires inter-system communication. A domain bridge was then used to selectively forward these necessary topics between domains.

For the 2026 RobotX Challenge, the team developed an Operator Control Station (OCS) system that can communicate with the judges RoboCommand. Since the judges's server has a single endpoint, the OCS runs as a ROS 2 node translating vehicle state into RoboCommand's MQTT/Protocol Buffer interface. Telemetry is time-stamped and reported unknown once stale,

preventing a disconnected vehicle from appearing falsely autonomous. Operating mode is drawn from the existing operation_executive node. A standalone mode validates the sequence using synthetic data.

*2) Decision Making:*

Decision-making moved from the finite state machines (FSMs) used in 2024 to behavior trees (BTs), implemented with the open-source py_trees library and its py_trees_ros extension. BTs are easier to extend than FSMs: a new behavior is added as a new branch instead of adding states and transitions to an existing machine. The Python library was chosen because most members are undergraduates and Python lowers the time needed to write and test new behaviors. Since the BTs only make high-level decisions, and time-critical control runs in the navigation stacks and autopilots, Python's lower performance does not affect the vehicles.

The structure of the BTs follows the system-of-systems approach. Instead of one BT on the USV that controls the UAV and UUV as extensions of itself, each vehicle runs its own BT, and a heartbeat between vehicles monitors the links. This also keeps inter-vehicle traffic low, since vehicles exchange task information instead of low-level commands. A vehicle that loses connection with the others can keep working on its part of the task, so a momentary disconnection does not end the run.

Separate trees also let each vehicle's behavior be changed without touching the others. For example, a "soft abort" was added to the UAV's tree as a high-priority branch without modifying any existing behavior. Unlike the radio kill switch, which disarms the UAV, the soft abort holds the UAV in a hover at an obstacle-free altitude and interrupts the running mission. It also cancels the active goals on the navigation action servers, so the maneuver in progress is stopped instead of continuing in the background. The action clients already supported cancellation, and the BT provides a single way to trigger it from every task.

## III. Testing Strategy

The team's testing follows a fixed progression. A capability is first developed in simulation, then integrated on the vehicle, validated on a bench, and finally, tested on water. A capability does not advance to the next stage until it has passed the previous one. Because water testing was the team's scarcest resource, this progression ensured no untested capability reached the water. To date, the team has logged approximately 200 hours of simulation, 150 hours of bench and integration testing, 50 hours of water testing, and 60 hours of flight testing.

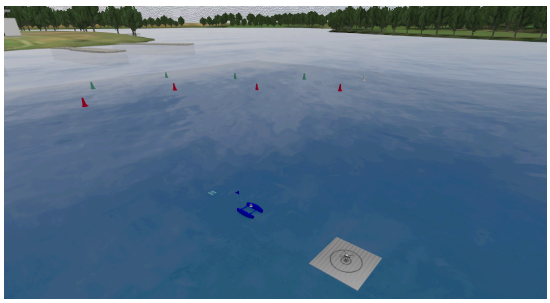


Fig 10. VRX test world

### *A. Simulation*

Simulation is used mainly to test mission logic, coordination between vehicles, edge cases and fallbacks, since it allows the structure of the system's responses to be changed and re-tested quickly. For that purpose, the team built custom VRX worlds for each task, each tier and variants for different configurations, such as alternative safe passage layouts and a final world combining all three tasks. The vehicle's models were adapted according to the official Blue Robotics data sheets to match the specifications of each vehicle. Before integration of the real sensors, the team also tested the USV in hardware-in-the-loop configurations, driving the real vehicle with simulated sensor data.

However, the simulations do not reproduce photorealistic conditions such as glare or underwater camera blur. Therefore, capabilities that depend on these conditions are verified with real sensors either on the bench or outside.

### C. *Field Testing*

Field tests follow fixed scheduling rules. They are held on Saturdays and planned two weeks in advance. Bench tests are allowed during the week of the test only until Thursday, and no changes are made on the vehicles on Friday unless they correct an emergency.

Each test is planned in advance with its objectives and the data to be collected, and available time is divided into blocks, each dedicated to a specific capability or data collection. Objectives are classified as primary or secondary, and secondary objectives are dropped if time runs short or problems arise.



Fig 11. USV testing in water

Each vehicle requires a software operator and a separate RC pilot, who is responsible for arming, disarming and identifying unsafe situations. If the latter does occur, a designated safety operator activates the system emergency stop. Before launch, the USV's load extenders are checked for leaks and the sensor platform and distribution box are checked for secure mounting. Finally a dry test of the motors using the RC control is performed to confirm their operation. The UUV is sealed following Blue Robotics' procedure and vacuum test [6]. The UAV undergoes pre-flight and arming checks before take off.

The USV and UUV have been tested mainly in the university's indoor pool, and the UAV on the university's football field. Both sites are on campus, which simplifies transport and allows short, frequent tests, but lack of a single open-water site has prevented the team from testing all three vehicles together. Furthermore, the university's pool roof prevents the use of GPS for the USV. UAV flight requires prior approval from the university for both space and the flight. Each flight progresses from manual control, to a basic autonomous routine, and only then to a full autonomous task.

## IV. Acknowledgements

Team Caleuche thanks the Office of Naval Research (ONR), the team's main sponsor since 2023, and RoboNation and Blue Robotics for granting the BlueBoat and BlueROV2 through the granted platform program. Without their support, the team could not have competed in this edition. The team also thanks the Department of Mechanical Engineering at Pontificia Universidad Católica de Chile for providing a workspace and administrative support, and its faculty advisor, Professor Sergio Aguilera, for his administrative support and for representing the team before the university. The team also thanks Markus Buchholz for giving access to his Gazebo SITL simulations of the BlueBoat and BlueROV2, which served as a reference when the team began developing its own simulation environment.

Finally, the team is especially grateful to Ignacio Gajardo, who led the team through 2025, when its continuity was uncertain. He secured the team's participation in innovation and project events that opened new opportunities, and ensured the team had a space to work during the remodeling of the Department of Mechanical Engineering.



Fig 13. Several Caleuche team members

## References

[1] O. Águila, N. Arispe, I. Gajardo, D. Galleguillos, C. López, V. Sufán, J. T. Unger, and G. Vidal, "Team Caleuche Technical Design Paper for the RobotX Challenge 2024," Team Caleuche, Pontificia Universidad Católica de Chile, Santiago, Chile, 2024.

[2] Open Navigation LLC, "Nav2 - Your autonomous navigation framework," Nav2. [Online]. Available: https://nav2.org/ (accessed Sep. 28, 2026).

[3] R. C. Martin, *Clean architecture: A Craftsman's Guide to Software Structure and Design*. Pearson Professional, 2018.

[4] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," Int. J. Adv. Robot. Syst., vol. 17, no. 2, Mar.-Apr. 2020, Art. no. 1729881420910530, doi: 10.1177/1729881420910530.

[5] eProsima, "How to solve wireless network issues in ROS 2," Vulcanexus Documentation, 2022. https://docs.vulcanexus.org/en/jazzy/rst/tutorials/core/wifi/wifi_issues_tutorial/wifi_issues_tutorial.html

[6] Blue Robotics, "BlueROV2 assembly and build instructions," *Blue Robotics*, Sep. 04, 2026. https://bluerobotics.com/learn/bluerov2-assembly/

APPENDIX A
Component Specifications (Chucao 2.0)

| **Component** | **Vendedor** | **Model / Type** | **Specification** | **Custom/Purchased** | **Cost** | **Year of Purchase** |
|---|---|---|---|---|---|---|
| AirFrame | Hexoon | EDU-650 V2 | Multirotor Quadrotor; 500x500x400 mm | Purchased | 479.17 GBP | 2024 |
| Single Board Computer | Latte Panda | Delta 3 | LPDDR4 8GB, Intel® UHD Graphics | Purchased | 311 USD | 2024 |
| Flight Controller | Holybro | Pixhawck 6C | STM32H743 | Purchased | 165.99 USD | 2024 |
| Power Distribution | Holybro | PM07 | 2S-14S | Purchased | 36.66 GBP | 2024 |
| Reciber | RadioMaster | ER6 | 2.400-2.480 GHz; 0.079 limit - 0.4 W nominal EIRP | Purchased | 20.83 GBP | 2024 |
| Transmitter / Remote Controller | RadioMaster | Boxer | 2.400-2.480 GHz | Purchased | 116.66 GBP | 2024 |
| Camera | luxonis | OAK-1-W | up to 12MP (4056x3040) | Purchased | 199 USD | 2024 |
| Sensors | Holybro | ST VL53L1X Lidar | Communication via I2C | Purchased | 14.99 USD | 2024 |
| Motors | T-Motor | HS4012 | 370KV | Purchased | (included in AirFrame) | 2024 |
| ESCs | Hexoon | 40A | - | Purchased | (included in AirFrame) | 2024 |
| Propelleer | T-Motor | MF1503 | - | Purchased | (included in AirFrame) | 2024 |
| Battery | LiPo-Play | Lithium Polymer | 22.2 V, 10,000 mAH, 6S1P | Purchased | 239 990 CLP | 2024 |

| | | | | | | |
|---|---|---|---|---|---|---|
| External Comm. Network | Anjeilo | WiFi HaLow | Up to 1000 m | Purchased | 50.00 USD | 2024 |
| Flotation Device | In-house | - | Pool Noodles | Custom | - | 2026 |
| Payload (Gripper) | In-house | - | Electro-magnetic Gripper | Custom | - | 2026 |
| Flight Controller Firmware | | PX4 v1.14 | - | - | - | - |
| Testing Time: Simulation | - | - | 80 hours | - | | - |
| Testing Time: Real | - | - | 40 hours | - | | - |

APPENDIX B
Component Specifications (Caleuche II 1.0)

| Component | Vendor | Model / Type | Specification | Custom / Purchased | Cost (USD) | Year of Purchase |
|---|---|---|---|---|---|---|
| Hull / Platform | Blue Robotics | BlueBoat USV | 120 x 93 x 46 cm; 14.5 kg without payload | Purchased | 4,400 | 2026 |
| Waterproof Connectors | Zona Industrial | LP12 | 8-pin, industrial grade | Purchased | 50 | 2026 |
| Propulsion | Blue Robotics | 2 x M200 motors, weedless propellers | 12–16 V operating range; 470 RPM/V | Purchased | Included in BlueBoat | 2026 |
| Power System | In-house | 2 x 14.8 V Li-ion batteries, Pelican 1200 box | 10 A fuse board per hull; 4 breakers for segmented power; DC-DC converters per hull | Custom | 1,200 | 2026 |
| Motor Controls | Blue Robotics | ESC | PWM thrust control from Navigator FC | Purchased | Included in BlueBoat | 2026 |
| CPU | NVIDIA | Jetson AGX Orin 64GB | 12-core Arm Cortex-A78AE @ 2.2 GHz; Ampere GPU, 2048 CUDA / 64 Tensor cores | Purchased | 3,500 | 2024 |
| Teleoperation | RadioMaster | XR1 dual-band | 2.400–2.480 GHz; 0.158 W EIRP nominal; omnidirectional | Purchased | 85 | 2026 |
| Compass | HBK (Hottinger Brüel & Kjær) | 3DM-GQ7-GNSS /INS | 3-axis magnetometer + dual-antenna GNSS heading | Purchased | Included in IMU | 2022 |

| | | | | | | |
|---|---|---|---|---|---|---|
| IMU | HBK (Hottinger Brüel & Kjær) | 3DM-GQ7-GNSS /INS | Temperature-compensated; onboard EKF, cm-level precision | Purchased | 4,023 | 2022 |
| DVL | - | - | - | - | - | - |
| Camera | Luxonis | OAK-1 PoE | Water-resistant RGB; on-device NN inference | Purchased | 379 | 2024 |
| Hydrophones | Aquarian Hydrophones | 4 x AS-1 | 1 Hz–100 kHz linear range; omnidirectional, can also transmit | Purchased | 413 | 2026 |
| Lidar | Ouster | OS0 | 90° FOV; 35 m range @ 10% / 75 m @ 80% reflectivity; up to 5.2M points/s (128 ch) | Purchased | 18,000 | 2024 |
| Algorithms | Ultralytics | YOLO | Detection, classification, pose estimation, tracking | Vendored | 0 | 2025 |
| Localization and Mapping | Open Navigation, ROS 2 | Nav2, robot_localization | EKF odometry and map pose feeding Nav2 | Vendored | 0 | 2026 |
| Autonomy | Blue Robotics | Navigator Flight Controller | ArduRover firmware; EKF fusion, autonomous navigation | Purchased | Included in BlueBoat | 2026 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Software | ROS 2 | Humble | Custom repo: sensing, perception, localization, missions, comms with base station, UUV and UAV | Vendored | 0 | 2024 |
| Testing Time: Simulation | - | - | 80 hours | - | | - |
| Testing Time: In-water | - | - | 60 hours | - | | - |

APPENDIX C
BLUEBOAT SENSOR TOWER 2026

A lightweight, modular sensor tower was developed for the BlueBoat to integrate the LiDAR, PoE camera, GNSS/GPS system, status light and Pelican 1200 electronics enclosure within the vehicle's payload and space constraints. The design uses 20×20 mm aluminum T-slot profiles and PETG mounts, preserves the sensor positions established by the team, and allows longitudinal and transverse adjustment without modifying the main frame.

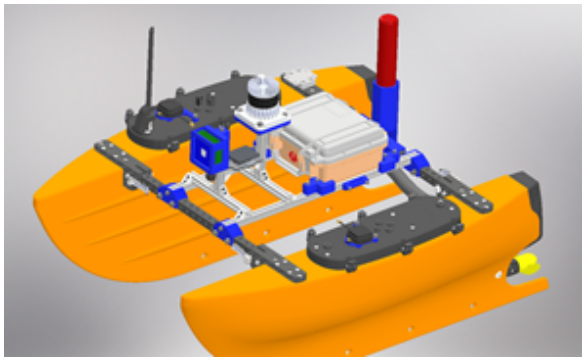


*Figure C-1. BlueBoat sensor tower CAD integration.*

| | |
|---|---|
| **Structure / size** | 20×20 mm Al T-slot + PETG · 653 × 287 × 378 mm |
| **Mass / payload target** | 2.04 kg structure · 3.604 kg with sensors · below 4 kg design target |
| **Adjustment / access** | x: −90/+40 mm · y: ±85 mm · Pelican lid remains fully accessible |
| **Static FEA** | 1000 N distributed load · 157.4 MPa · 0.8645 mm · minimum SF = 1.75 |

The tower mounts directly to the BlueBoat transverse bars through PETG interfaces and stainless M8 fasteners. Four printed restraints and a retaining strap secure the Pelican while preserving service access. The 1000 N FEA case was selected as a conservative accidental load, substantially above normal operating loads. *Detailed sensor configuration, mechanical design, manufacturing and FEA results are provided in Appendices C.A–C.D.*

## C.A · SENSOR CONFIGURATION

### C.A.1 Reference system and nominal positions

The z-coordinate is measured vertically from the flat upper surface of the pontoons. The x-coordinate is measured longitudinally from the physical center of the BlueBoat, with positive x pointing toward the bow.

| Sensor | Longitudinal position x | Height z |
|---|---|---|
| Camera | +270 mm | +150 mm |
| LiDAR OS0 | +100 mm | +250 mm |

The nominal longitudinal separation between the camera and LiDAR is 170 mm.

### C.A.2 Adjustment range (Camera and LiDAR)

| Direction | Available travel from nominal position |
|---|---|
| Longitudinal | −90 mm aft / +40 mm forward |
| Transverse | −85 mm / +85 mm |

Adjustment is achieved by loosening the T-slot fasteners and translating the sensor substructure along the profiles. This allows later tuning without manufacturing a new main frame.

### C.A.3 Installed sensors and devices

| Element | Qty | Mass | Role |
|---|---|---|---|
| LiDAR OS0 | 1 | 0.450 kg | 3D perception |
| ANN-MB-00 | 2 | 0.173 kg each | Multiband GNSS antennas |

| | | | |
|---|---|---|---|
| GPS-GQ7-ASY | 1 | 0.078 kg | GNSS/INS |
| Status light | 1 | 0.500 kg | Vehicle status signaling |
| OAK-1-POE | 1 | 0.190 kg | Visual perception |

## C.B · MECHANICAL DESIGN, MATERIALS AND MASS

### C.B.1 Main structure

The frame uses seven 20×20 mm aluminum T-slot profiles. All cut lengths use a ±1 mm manufacturing tolerance.

| Profile length | Quantity |
|---|---|
| 623 mm | 2 |
| 236 mm | 2 |
| 110 mm | 1 |
| 161 mm | 1 |
| 166 mm | 1 |

### C.B.2 Overall dimensions

| Dimension | Value |
|---|---|
| Overall length | 653 mm |
| Overall width | 287 mm |
| Overall height | 378 mm |
| Height above pontoon top surface | 361 mm |

### C.B.3 Structural components and fasteners

| Element | Quantity | Total mass [kg] |
|---|---|---|
| 20×20 mm aluminum T-slot profile | 2.155 m | 1.0750 |
| 20×20 mm 90° corner bracket | 10 | 0.0600 |
| M5 sliding T-nut | 32 | 0.0736 |
| M5 stainless Allen bolt, 12 mm | 2 | 0.0066 |
| M5 stainless Allen bolt, 25 mm | 4 | 0.0204 |
| M5 stainless Allen bolt, 30 mm | 8 | 0.0464 |
| M5 button-head Allen bolt | 20 | 0.0680 |
| M8 stainless Allen bolt, 50 mm | 12 | 0.3000 |
| M6 stainless Allen bolt, 20 mm | 10 | 0.0700 |
| PETG | 0.32 kg | 0.3200 |

Structural mass: 2.04 kg. Sensor mass: 1.564 kg. Total structure + sensors: 3.604 kg.

### C.B.4 BlueBoat fixation and Pelican integration

The tower is attached to the BlueBoat transverse bars using PETG 3D-printed mounts and M8 stainless Allen bolts. The Pelican 1200 is constrained in the horizontal plane by four printed supports and retained vertically by a strap anchored to additional printed mounts. Releasing the strap provides full lid access without removing the sensor assembly. Figures B-2 and B-3 show the mounting interface to the BlueBoat and the Pelican support arrangement, respectively.

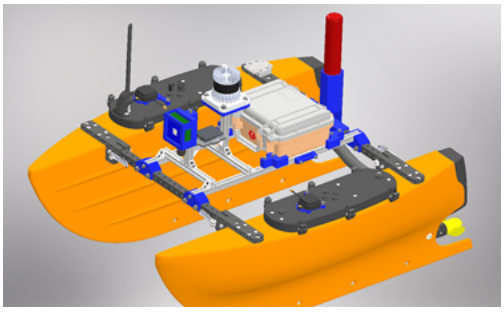


*Figure C.B-1. Overall mechanical integration of the tower, sensors and Pelican.*

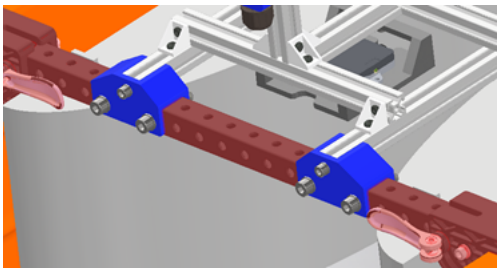


*Figure C.B-2. BlueBoat mounting interface*

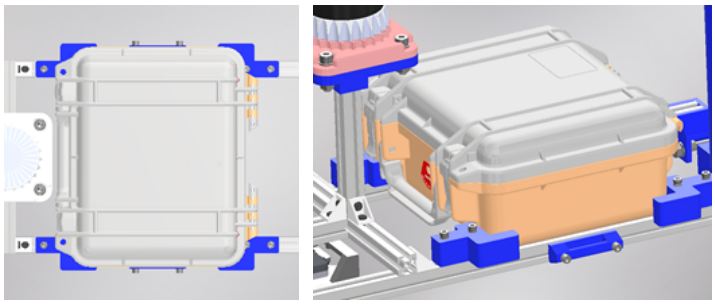


*Figure C.B-3. Pelican mounting arrangement*

## C.C · MANUFACTURING AND ASSEMBLY

### C.C.1 Profile fabrication

The 20×20 mm T-slot profiles are cut to the lengths listed in Appendix B using a ±1 mm tolerance. Connections use 90° brackets, Allen bolts and sliding T-nuts, avoiding welded joints and preserving disassembly and reconfiguration capability.

### C.C.2 3D-printed components

Geometry-specific mounts are manufactured in PETG. Printed components include:

- Tower-to-BlueBoat attachment mounts.
- LiDAR and camera mounts.
- GNSS antenna and GPS receiver mounts.
- Status-light mounts.
- Pelican 1200 lateral restraints.
- Pelican retaining-strap mounts.

Print orientation places the main service loads in the strongest layer direction wherever practical.

### C.C.3 Recommended assembly sequence

| Step | Operation |
|---|---|
| 1 | Cut, deburr and square the main T-slot profiles. |
| 2 | Install corner brackets, bolts and T-nuts without final tightening. |
| 3 | Install the movable LiDAR/GPS substructure and camera mount. |

| | |
|---|---|
| 4 | Install GNSS, status-light and Pelican mounting hardware. |
| 5 | Attach the tower to the BlueBoat transverse bars. |
| 6 | Set nominal sensor positions and complete final tightening. |
| 7 | Install the Pelican, secure the retaining strap and verify lid access. |

After installation, verify fastener tightness, absence of interference, sensor adjustment freedom and unrestricted Pelican access.

## C.D · STRUCTURAL FEA VALIDATION

### C.D.1 Analysis setup

| Parameter | Configuration |
|---|---|
| Analysis type | Static |
| T-slot profiles | Aluminum 6061 |
| Printed parts | PETG |
| Boundary condition | Tower fixed to two representative transverse bars |
| Primary load | 1000 N downward distributed load on the Pelican support region |
| Additional load | Gravity |
| Joint assumption | Properly tightened joints with no relative sliding |

### C.D.2 Mesh

| Parameter | Value |
|---|---|
| Average Element Size | 0.100 |
| Minimum Element Size | 0.200 |
| Grading Factor | 1.500 |
| Maximum Turn Angle | 60° |
| Part-based assembly mesh | Enabled |

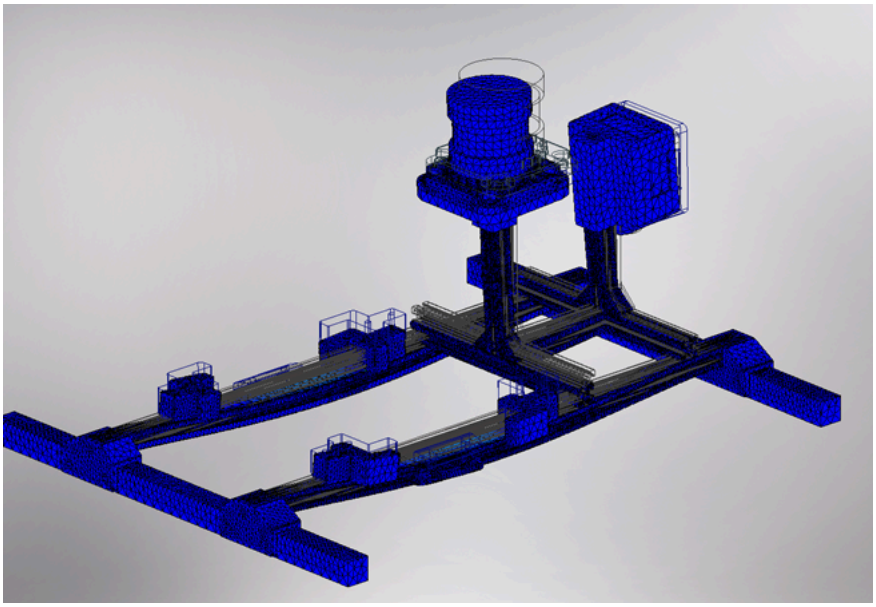


*Figure C.D-1. Structural model mesh.*

### C.D.3 Results

| Result | Value |
|---|---|
| Maximum von Mises stress | 157.4 MPa |
| Maximum displacement | 0.8645 mm |

| | |
|---|---|
| Minimum safety factor | 1.75 |

The maximum stress and minimum safety factor occur in localized regions of the model inside T-slot profile geometry. Maximum displacement remains below 1 mm under the conservative 1000 N load case.

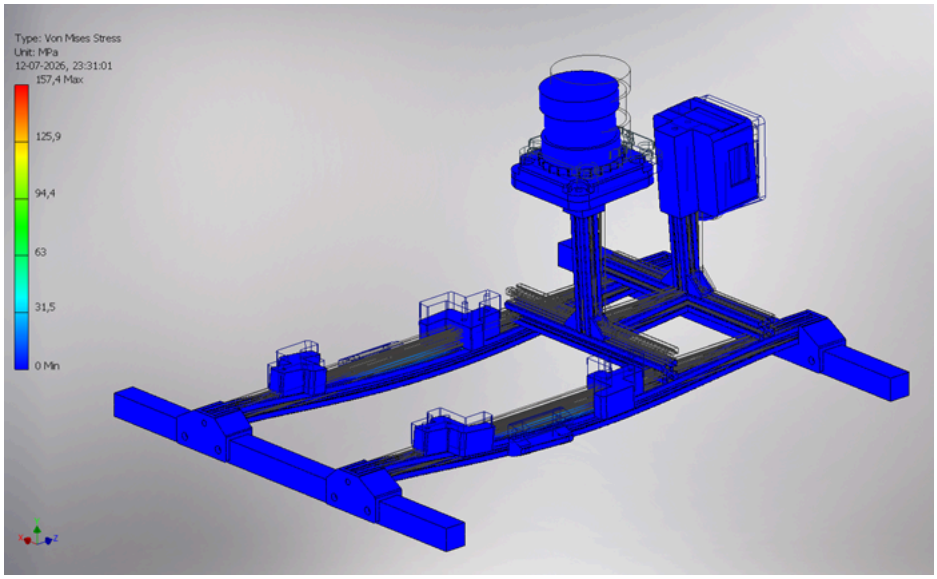



*Figure C.D-2. Von Mises stress distribution.*

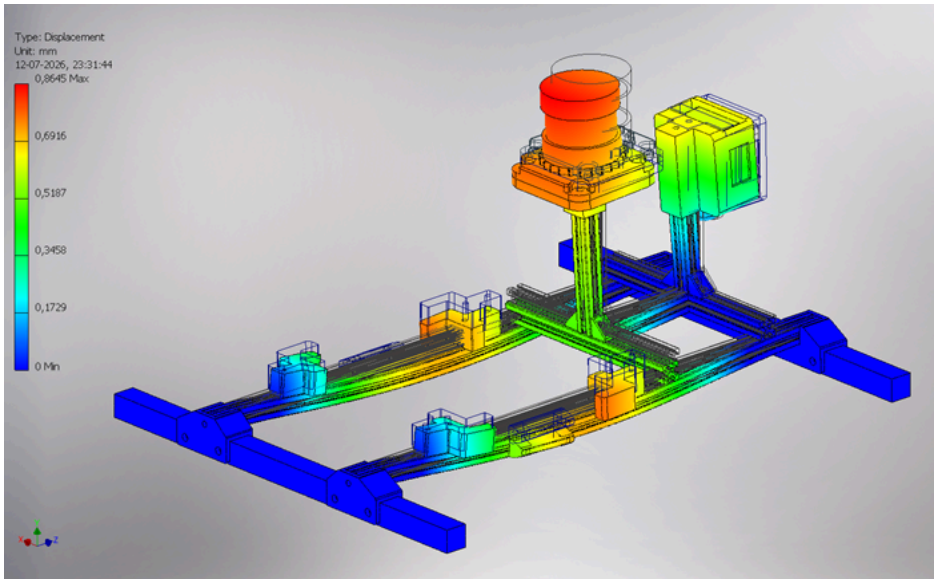



*Figure C.D-3. Displacement distribution.*

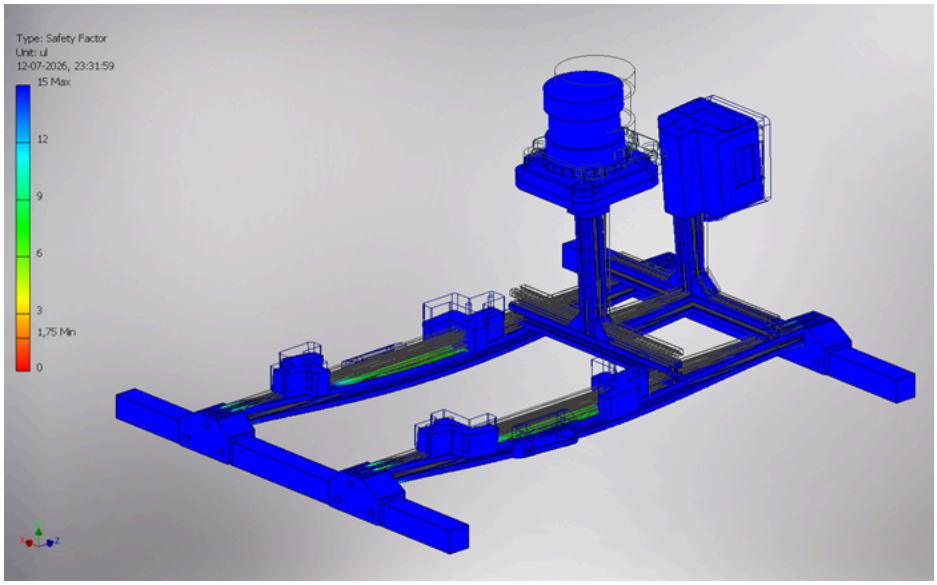



*Figure C.D-4. Safety factor distribution.*

**C.D.4 Validation scope**

The 1000 N load was selected as a conservative accidental load case, including the possibility of a person sitting on or falling onto the Pelican support area. The results support the tower's static strength and stiffness. The assembled tower was also physically tested on the BlueBoat, with no relevant vibration, loosening, or sensor instability observed during operation. The central cross-member and the short unsupported lengths of the sensor-supporting profiles contribute to a stiff load path and limit structural motion. Based on the analytical and experimental results, no additional vibration mitigation was considered necessary for the current configuration.

# APPENDIX D
## Task 1 Gate Pairing Algorithm Testing

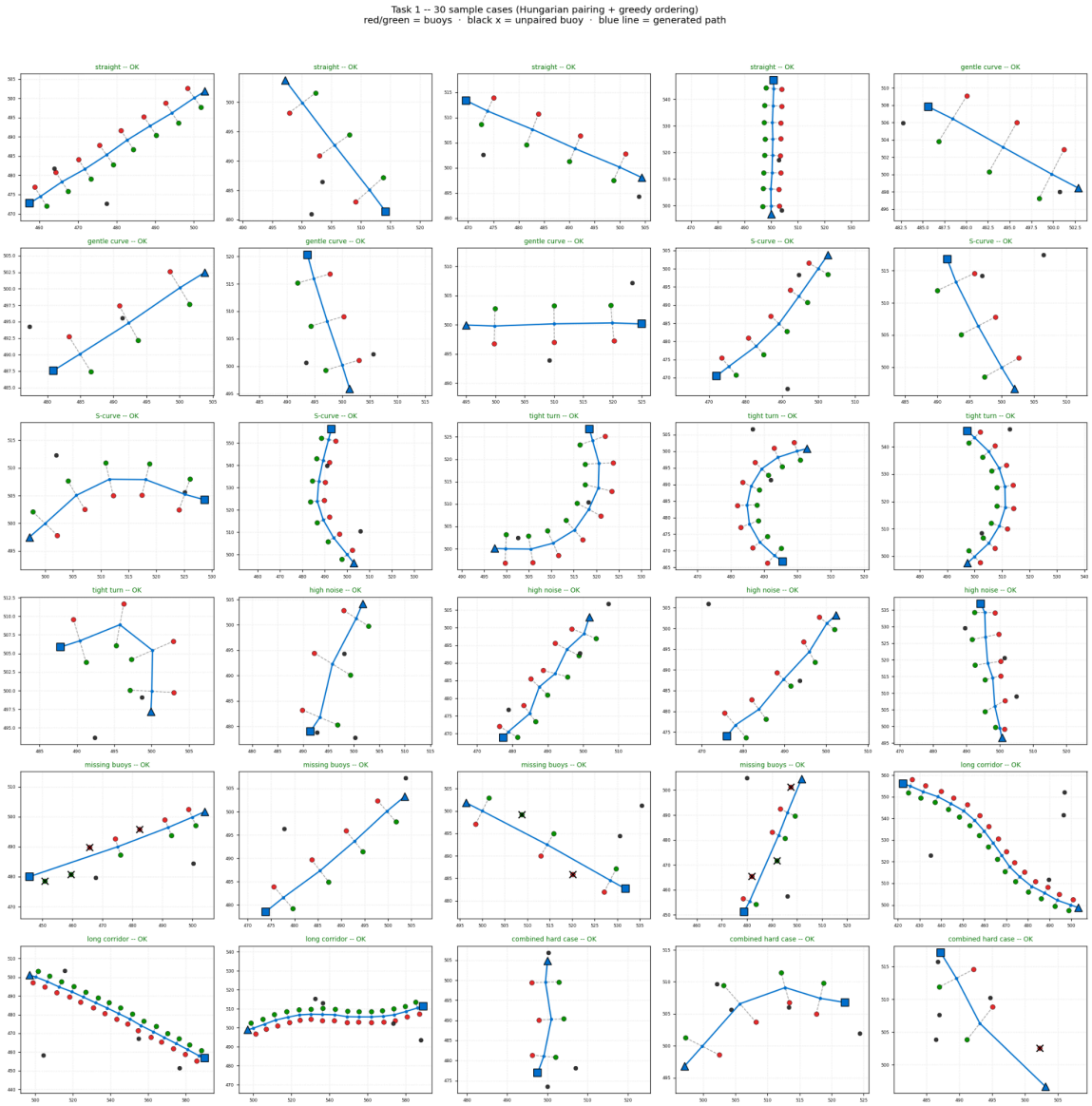



*Figure D-1. 30 sample cases for Task 1 gate pairing.*

APPENDIX E
USV Hardware architecture

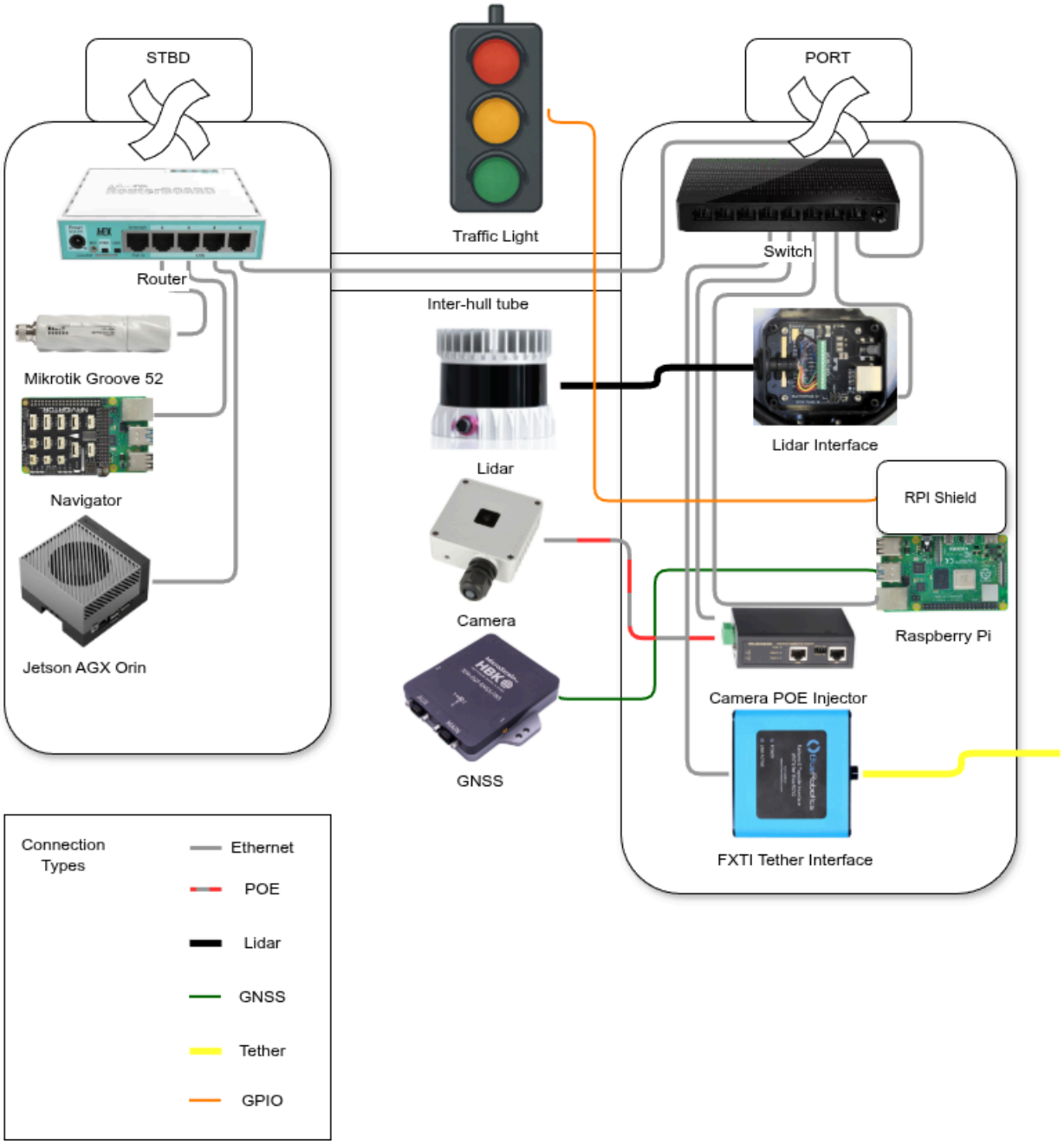



*Figure E-1. USV Hardware architecture*

The current USV hardware architecture diagram was designed in a way that any member of the team can learn how to connect the system. Real life photos were selected when possible so that inexperienced members could easily learn to recognize which equipment was which.