Maritime RobotX Challenge 2026
Technical Design Report

OUXT Polaris — Kyushu Institute of Technology

Takeshi Yamagami, Kenta Futami, Reo Fujioka, Yota Nozaki, Amisha Rawat, Marina Okano, Shobhit Pokhriyal, Kosuke Iwazu, Saku Itozono, Ryohei Onishi, and Fumiya Matsuzaki

Abstract

OUXT Polaris is adapting its proven Mini-V autonomous surface vehicle into a reliable System of Systems for the 2026 Maritime RobotX Challenge. Our competition strategy prioritizes Task 1 Safe Passage and Task 3 Coordinated Logistics because both allow a mature USV to provide navigation and manipulation while a staged uncrewed aerial vehicle contributes overhead perception and, later, resource delivery. The USV retains its stable aluminum frame, dual-thruster drive, Jetson AGX Orin/ROS 2 computer, Pixhawk/ArduPilot navigation, camera, and Velodyne VLP-16. RobotX-specific work concentrates on inter-vehicle messages, RoboCommand reporting through an operator control station, a clearly partitioned propulsion safety path, and an implemented wireless Remote Kill that fails safe on transmitter or RF-link loss. The UAV uses a PX6 flight-controller baseline; its integration is deliberately gated by manual takeover, link-loss, autonomous flight, flotation/recovery, and payload tests before it is assigned a competition role. This report explains how reuse of a tested USV and frozen subsystem interfaces reduce risk while staged collaboration provides a path to advanced autonomy. It also defines a verification plan covering component, simulation, field, failure-injection, and multi-vehicle tests without presenting planned UAV functions as completed capabilities. Boat and drone operational tests were conducted on August 24 and August 30, 2026, respectively.

Index Terms*—*Maritime autonomy, System of Systems, USV, UAV, ROS 2, emergency stop, collaborative logistics.

I. COMPETITION STRATEGY

A. Strategic vision and task selection

RobotX 2026 rewards collaborative autonomy rather than a collection of independent vehicles. OUXT Polaris therefore treats the USV, UAV, operator control station (OCS), and safety controllers as one engineered system. The strategy is reliability-first: retain the parts of Mini-V exercised for RoboBoat, then add only the interfaces needed for RobotX collaboration and reporting. This approach follows the handbook's warning that complexity can consume the time needed for validation [1].

Task 1 Safe Passage is the primary collaborative objective. The USV supplies stable surface navigation and buoy perception, while the UAV is intended to observe overhead-only information and return a route map. The first acceptance target is the core behavior with safe autonomous operation. An advanced attempt is enabled only after the UAV-to-USV route message remains valid through flight, network, and perception tests. This keeps a temporary UAV fault from invalidating an otherwise mature USV run.

Task 3 Coordinated Logistics is the second objective. Mini-V is assigned approach, docking, and the surface-side water-delivery action. The UAV resource-pickup and delivery mechanism is a later stage because it introduces mass, balance, release, and recovery risks. Tasks whose planned capability requires an underwater vehicle are not assigned primary development time while our declared System of Systems contains only the USV and UAV.

B. Complexity and reliability trade-off

The largest trade-off is between immediately adding high-value aerial payloads and preserving a testable system. We selected a staged integration model. A capability is admitted to the competition configuration only after its safety behavior, interface contract, and evidence log are complete. The USV control loop, ProtoLink messages, and propulsion hardware remain isolated from experimental UAV code. Vehicle-to-vehicle collaboration uses



Figure 1: Mini-V USV retained as the RobotX surface-vehicle baseline.

small task-level messages—request, route, target, acknowledgement, and completion—rather than sharing each vehicle's internal control topics.

The same principle applies to communications. The XBee pair is reserved for Pixhawk/Mission Planner monitoring and telemetry [2]; it is not a control or emergency-stop path. Team data uses the 2.4 GHz TP-Link AX4400 network, and the OCS alone connects to the RoboCommand network. Safety links remain separate from task data so that a visualization, Wi-Fi, or RoboCommand failure cannot command propulsion.

C. Run concept

Before each collaborative run, the OCS receives and validates the course configuration, publishes one `RunDeclaration` containing both participating vehicle identifiers and the closed UAV geofence, and then publishes a separate 2 Hz heartbeat for each vehicle [3]. Both vehicles report their actual killed, manual, or autonomous state. After switching to autonomous mode they position-hold; mission motion begins only when the OCS verifies that `RunStart` refers to the current declaration. The OCS then records the run identifier and routes task reports on behalf of the appropriate vehicle.

II. DESIGN STRATEGY

A. System-of-Systems architecture

Figure 2 shows the functional architecture. Mini-V executes surface navigation, local perception, and propulsion control. The UAV executes flight control and supplies aerial observations or payload actions after passing its integration gates. The OCS is a

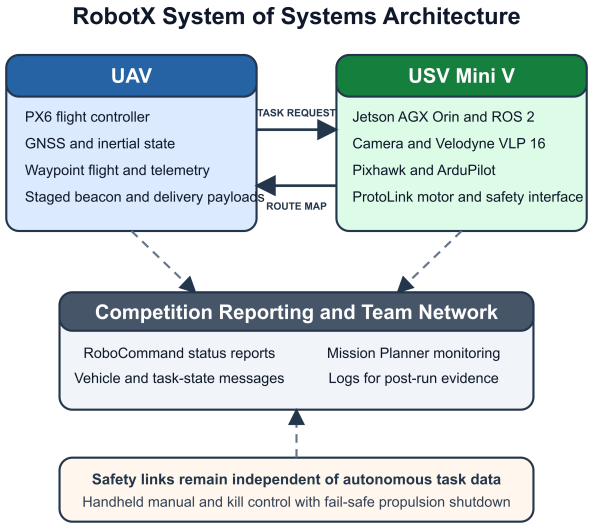


Figure 2: RobotX System-of-Systems architecture and information boundaries.

protocol boundary: it exchanges MQTT/protobuf messages with RoboCommand and distributes relevant course data to the team network, but it is not part of a vehicle's closed-loop control. Each vehicle can therefore enter a safe state if the OCS link disappears.

RoboCommand course, command, heartbeat, and task-report messages use the competition schemas without modification [3]. The OCS maintains an independent report sequence for the USV and UAV, verifies team and vehicle identifiers, rejects malformed payloads, reconnects automatically, and logs every state transition. This design supports evidence review while preventing logging or display failures from blocking vehicle control.

B. USV baseline and RobotX changes

The baseline for the USV is shown in Figure 1. We are using floats commonly found on canoes because we have experienced capsizing in the past. Additionally, we have lowered the center of gravity by wiring the battery and motor driver inside the floats. This makes it more resistant to rocking on the water and prevents capsizing. This hardware baseline was used in RoboBoat 2026 and will continue to be adopted.

One of the main changes in RobotX is thruster to provide differential thrust [4]. Before, we had problems about thrusters running because of the

lack of thrust and the overcurrent for ESC and motor driver. In order to resolve these problems, we have been adopted two Blue Robotics T200 thrusters now. In addition, the position of Velodyne VLP-16 was changed because it is too closed to water surface and we can't get collect point cloud data diffuse reflection. We set it at the top point of USV.

As the computer, Jetson AGX Orin runs ROS 2 Humble nodes for perception, mission logic, and hardware interfaces [5, 6]. A Pixhawk running ArduPilot supplies navigation state and low-level guidance through MAVLink [7]. A monocular camera and Velodyne VLP-16 provide complementary color and geometric observations.

However, the important RobotX changes are architectural rather than a wholesale rebuild: a formal USV–UAV task message boundary, OCS/RoboCommand reporting, a documented split-power propulsion path, direct relay control from the emergency controller, and a 360-degree auto/manual/kill indicator. The main computer and Ethernet switch are powered from a separate 5 V source. Left and right propulsion branches retain independent 18 V batteries and local DC-DC supplies for their motor-control Teensy 4.1 boards [8].

C. Propulsion safety path

The propulsion safety path is intentionally independent of motor commands. The emergency-controller Teensy 4.1 receives the hard-wired physical E-stop input, supervises the Jetson heartbeat and requested mode, drives the external mode indicator, and sends the release signal directly to both propulsion relays. It does not route the relay release through either motor-control Teensy. Loss of the computer/network heartbeat or removal of valid release conditions de-energizes the relays and removes power from the ESC/thruster branches.

The existing configuration implements wireless Remote Kill through the RadioLink T8FB/R8EF used for manual control [9]. Pressing Remote Kill, switching off or removing power from the transmitter, or losing the R/C RF link drives the vehicle to the killed state and removes the relay-release condition. This safety path does not depend on the Jetson, laptop, OCS, team Wi-Fi, or XBee telemetry link. For RobotX inspection, the team

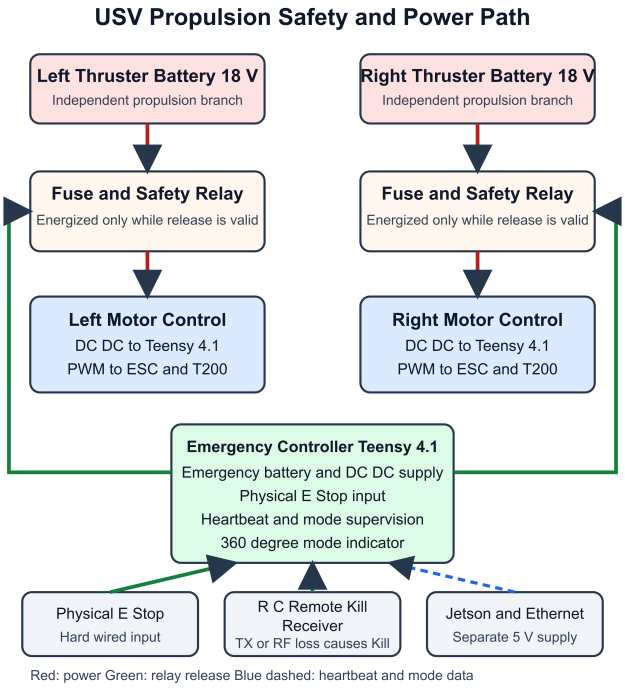


Figure 3: Mini-V propulsion power and independent emergency-stop path.

will demonstrate the implemented off-board E-stop, including transmitter power removal and RF-link-loss tests, and measure that propulsion power is removed within the required two seconds [10].

D. USV software and perception

ROS 2 separates sensing, localization, mission logic, and actuation. The Jetson receives vehicle state from Pixhawk over MAVLink and sends actuator intent through ProtoLink, a compact protobuf-based transport, to the motor-control Teensies [11]. The same transport carries the heartbeat and mode value to the emergency controller. A network interruption therefore removes propulsion authorization instead of leaving the last PWM command active.

The perception pipeline combines camera detections with lidar geometry. An Ultralytics YOLO detector identifies task objects in the image [12]; synchronized lidar points are projected into the camera frame to estimate range and reject isolated detections. The detector output is converted into task-level landmarks rather than being used as a direct motor command. This preserves the existing navigation safety envelope when the visual model is uncertain.

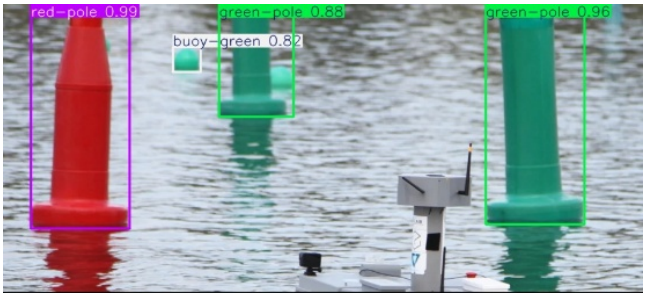


Figure 4: RoboBoat baseline object detection retained for RobotX development.

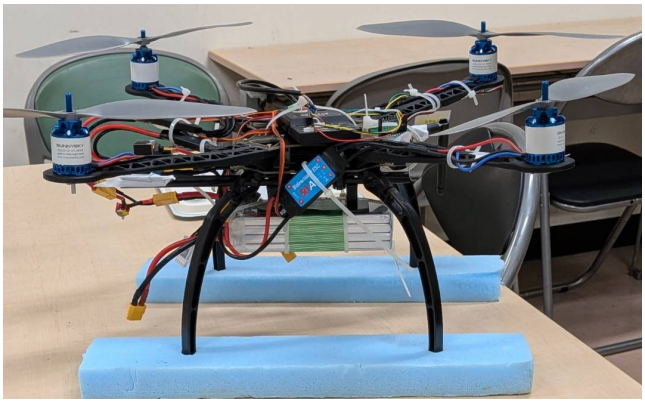


Figure 5: Air-V baseline image.

E. UAV baseline and system

The UAV baseline image is Figure 5. The aerial component of our System of Systems is the "Air-V", a customized multirotor UAV based on the COTS Holybro S500 V2 frame kit.Flight control is managed by a Holybro Pixhawk 6X running PX4 firmware, which provides reliable attitude stabilization and waypoint navigation. Power is supplied by a 4S 14.8V 6500mAh LiPo battery, offering an operational endurance of approximately 10 to 15 minutes. To mitigate the risk of water ditching, foam floats are permanently attached to the landing gear to ensure buoyancy and safe recovery.

For the RobotX collaborative tasks, Air-V carries a dedicated payload of approximately 200 g. This payload includes an onboard camera, Raspberry Pi as a companion computer, and a delivery gripper. The primary autonomous mission involves executing vertical take-off from the land pad, performing aerial object detection using YOLO, transmitting the detected object coordinates to the USV, and autonomously returning for a land-based vertical landing.

Ground control is centralized through QGroundControl (QGC) communicating via MAVLink telemetry, while a RadioLink T8FB 2.4 GHz transmitter provides the manual control link. Safety and operational limits are strictly defined: the UAV operates within a 100 m geofence radius and a 50 m altitude limit.

Failsafe mechanisms are deeply integrated into the UAV's configuration. An emergency kill switch is mapped to Channel 7 of the transmitter, allowing the safety pilot to immediately stop all motors. The PX4 firmware is configured to automatically trigger a Return-to-Launch (RTL) mode if the manual RC link is lost for more than 0.5 seconds. Additionally, a tiered battery failsafe triggers a warning at 15% capacity, initiates an automatic RTL at 7

F. UAV integration strategy

The current UAV uses the PX6 flight controller and completed a drone operational test on August 30, 2026. The exact airframe, battery, remote controller, companion computer, perception payload, delivery mechanism, and flotation/recovery specifications remain to be documented in the submission. Instead of assuming that the operational test closes every competition requirement, the UAV enters the System of Systems through three gated increments.

- **Flight-ready increment:** dedicated manual controller, manual and autonomous takeoff/landing, waypoint and position-hold flight, valid geofence behavior, complete telemetry, link-loss motor-safe behavior, airworthiness, and a demonstrated flotation/recovery method.
- **Safe Passage increment:** an overhead perception payload detects the required beacon information, transforms it to the shared frame, and sends a versioned route-map message that the USV can validate or reject.
- **Coordinated Logistics increment:** a light resource-handling payload is added only after its mass, center of gravity, release reliability, and safe-abort behavior pass ground and flight tests.

This ordering makes flight safety independent of mission payload maturity. The UAV may be declared for a run only when it can truthfully publish its flight phase, position, state, and current task

through the OCS.

G. Collaboration interfaces

Inter-vehicle messages are intentionally sparse. For Safe Passage, the USV issues an observation request and the UAV returns a route-map object containing frame, timestamp, confidence, and validity interval. Additionally, aerial object recognition is performed using YOLO via a camera. The object's coordinates are also measured based on this information. For Coordinated Logistics, the USV issues a target/resource request; the UAV acknowledges, reports pickup, and reports delivery or abort. Receivers reject stale timestamps, unknown frames, impossible coordinates, and messages from a previous run. No inter-vehicle message can directly energize a relay or bypass a local controller's safety state.

III. TESTING STRATEGY

A. Verification pyramid

Testing follows a pyramid that places inexpensive fault discovery before integrated field time. Component tests cover power rails, relay logic, motor outputs, sensors, message serialization, and flight-controller I/O. Software-in-the-loop tests replay camera/lidar data, exercise USV routes, and validate UAV mission-state transitions. Hardware-in-the-loop tests then combine real controllers with simulated navigation and network faults. Only configurations that pass these stages proceed to controlled water or flight testing.

Every test records configuration identifiers, firmware revisions, ROS bag or controller logs, operator notes, and a pass/fail decision. A failed safety test blocks dependent autonomy tests. Planned functionality is kept separate from completed evidence so that competition declarations match the hardware actually inspected.

The 2026 test record includes a boat operational test conducted on August 24 and a drone operational test conducted on August 30. These completed activities establish that both physical platforms were operated before the RobotX integration phase. Their procedures, configuration identifiers, logs, photographs, and detailed results will be attached to the final test record.

B. USV regression and safety testing

The RoboBoat baseline is requalified rather than assumed. Dry tests verify both propulsion branches, PWM neutral, reversed-command protection, sensor timestamps, MAVLink state, and mode-indicator colors. Tethered water tests verify manual control, straight-line and turning response, station keeping, buoy detection, and recovery. Open-water tests then repeat the intended RobotX approach, gate, and docking trajectories with increasing speed only after lower-energy trials pass.

Failure injection is mandatory for the safety chain. Tests press and reset the on-board E-stop; activate the implemented wireless Remote Kill; remove its transmitter battery; remove the emergency-controller supply; stop the heartbeat process; disconnect the Jetson Ethernet cable; and remove Ethernet-switch power. In every case the relays must de-energize, propulsion must remain off until an intentional reset, and the 360-degree indicator must show the killed state. Relay response is measured at the contacts, not inferred from an LED.

C. UAV and collaborative testing

UAV testing begins without mission payloads. Bench and restrained tests verify controller power, manual takeover, radio loss, geofence, arming, and motor shutdown. Low-altitude flight tests validate takeoff, position hold, waypoint tracking, landing, emergency recovery, and telemetry. The aircraft is not integrated over water until flotation/recovery behavior and the competition operating constraints have been reviewed.

Collaboration is tested first with a simulated second vehicle, then with the real USV and UAV. The OCS must declare both vehicles, publish independent 2 Hz heartbeats, complete the `RunStart` sequence, preserve message sequence numbers, and recover after broker and team-network interruptions. Route and resource messages are tested for delay, duplication, loss, reordered delivery, stale timestamps, and invalid coordinates. The USV must continue to a defined hold or abort state instead of accepting unsafe data.

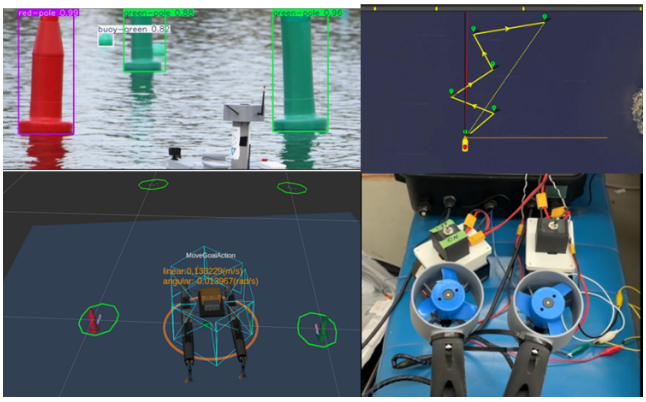


Figure 6: Existing Mini-V test activities form the baseline for RobotX regression.

D. Course rehearsal and acceptance

The final test level is a pseudo-competition rehearsal with launch, manual checkout, physical and remote E-stop demonstrations, mode-indicator demonstration, autonomous arming, `RunDeclaration`, `RunStart`, task execution, task reports, and recovery. A configuration is accepted only if the same software and wiring used for the rehearsal can be reproduced from the configuration record. Results from these rehearsals determine the task tiers declared on competition day.

ACKNOWLEDGEMENTS

OUXT Polaris thanks Kyushu Institute of Technology for laboratory, fabrication, and testing facilities; the faculty and technical staff who support system and safety reviews; previous OUXT Polaris members whose RoboBoat work established the Mini-V baseline; and sponsors and suppliers who provide components, advice, and competition support. Names and contribution details will be updated before submission to reflect the final RobotX 2026 team and confirmed sponsorship.

REFERENCES

[1] RoboNation. (2026) RobotX 2026 Team Handbook—Technical Design Report. [Online]. Available: https://robonation.gitbook.io/robotx-2026-team-handbook/section-2-design-documentation/2.1-technical-design-report-tdr

[2] Digi International. Digi XBee 3 Zigbee RF Module. [Online]. Available: https://www.digi.com/resources/documentation/digidocs/pdfs/90001543.pdf

[3] RoboNation. (2026) RobotX 2026 Communications Protocol. [Online]. Available: https://robonation.gitbook.io/robotx-2026-team-handbook/section-3-autonomy-challenge/3.4-communications-protocol

[4] Blue Robotics. T200 Thruster. [Online]. Available: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/

[5] NVIDIA. Jetson AGX Orin Developer Kit. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-orin-developer-kit

[6] Open Robotics. ROS 2 Humble Documentation. [Online]. Available: https://docs.ros.org/en/humble/

[7] ArduPilot Dev Team. ArduPilot Documentation. [Online]. Available: https://ardupilot.org/ardupilot/

[8] PJRC. Teensy 4.1 Development Board. [Online]. Available: https://www.pjrc.com/store/teensy41.html

[9] RadioLink Electronic Ltd. T8FB 2.4 GHz Eight-Channel Transmitter. [Online]. Available: https://www.radiolink.com/t8fb

[10] RoboNation. (2026) RobotX 2026 Safety Requirements. [Online]. Available: https://robonation.gitbook.io/robotx-2026-team-handbook/section-5-rules-and-requirements/5.2-safety-requirements

[11] OUXT Polaris. protolink_drivers. [Online]. Available: https://github.com/OUXT-Polaris/protolink_drivers

[12] Ultralytics. Ultralytics YOLO Documentation. [Online]. Available: https://docs.ultralytics.com/

I. TEST PLAN AND EVIDENCE STATUS

Table 1 is the working acceptance matrix for the RobotX configuration. "Baseline evidence" means that the capability was exercised during the RoboBoat program and must still be regressed on the final hardware. "RobotX verification required" identifies work that must be completed and documented before the associated vehicle or task tier is declared.

Area	Method	Acceptance evidence	Current status	Gate
USV propulsion	Dry plus tethered-water command tests	Neutral, forward/reverse, current and response logs	Operational test conducted Aug. 24, 2026; evidence summary to add	USV
On-board E-stop	Actuate under commanded thrust	Relay contact trace and zero propulsion until reset	Existing path; final wiring test required	USV
Remote E-stop	Activate, switch off transmitter, remove battery, and interrupt RF	Power removal in < 2 s and latched killed state	Implemented Remote Kill; measured evidence to attach	USV
Heartbeat loss	Stop process, unplug LAN, remove switch power	Both propulsion relays de-energize	Behavior observed; measured evidence required	USV
Mode indicator	Command auto/manual/kill and inject faults	360° colors match actual safety state	LED correction and final video required	USV
OCS protocol	Demo broker or local test server	Course decode, declaration, 2 Hz heartbeats, `RunStart` log	Software integration in progress	SoS
UAV flight safety	Bench, restrained, low-altitude flight	Manual takeover, link loss, geofence, takeoff/landing	Operational test conducted Aug. 30, 2026; evidence summary to add	UAV
UAV water recovery	Unpowered flotation/recovery demonstration	Stable float and documented recovery method	Hardware not frozen	UAV
Safe Passage	Simulated then live USV–UAV trial	Validated route map; stale/invalid data rejection	Planned after flight gate	Task 1
Coordinated Logistics	Ground release tests then controlled flight	Pickup/delivery success and safe abort	Payload not frozen	Task 3
Full rehearsal	Pseudo-competition sequence	Safety check, declaration, start, task, reports, recovery	Pending final integration	SoS

Table 1: RobotX acceptance matrix.

Submission note: replace planning statuses with dated results, test locations, configuration identifiers, and links to logs/video after each gate is closed.