RoboBoat 2026:Technical Design Report

Takeshi Yamagami[1], Kenta Futami[2], Reo Fujioka[2], Yota Nozaki[2], Kosuke Iwazu[2],
Saku Itozono[1], Ryohei Onishi[1], Fumiya Matsuzaki[3]

[1] *Graduate School of Life Science and Systems Engineering*, *Kyushu Institute of Technology*, Fukuoka, Japan
[2] *School of Engineering, Kyushu Institute of Technology*, Fukuoka, Japan
[3] *Joint Graduate School of Mathematics for Innovation*, *Kyushu University*, Fukuoka, Japan

***Abstract*—OUXT Polaris, a team from the Kyushu Institute of Technology, competes in the RoboBoat and Maritime RobotX Challenges. Committed to open-source principles, the team actively shares its hardware and software developments with the community. For RoboBoat 2025, we introduced the "Mini-V," a scaled-down iteration of the WAM-V platform. However, stability issues led to a capsizing incident during autonomous navigation, preventing task completion. Consequently, for RoboBoat 2026, our primary focus has been enhancing the Mini-V's hardware to ensure platform stability and operational reliability. This Technical Design Report (TDR) details the hardware refinements and the integrated software architecture of the upgraded Mini-V.**

***Keywords—Maritime systems, Robotics, Unmanned surface vehicle*.**

Fig. 1 Mini-V hardware

I. Design Strategy

A. Hardware Concept

OUXT Polaris is competing in the Maritime RobotX Challenge as well as RoboBoat, and its greatest requirement is that Mini-V act as a compatible machine for WAM-V. While the WAM-V requires half a day for assembly alone, the Mini-V can be transported by one person and assembled in a few minutes, making it an affordable platform for experimental validation. The largest component, the Hull, can be folded for extremely compact storage. The vessel's communication network is also Ethernet-based, and the protocol is unified with WAM-V (Fig. 1).

*1) **Improved Stability (Anti-Capsize Measures):*** In RoboBoat 2025, the vessel capsized due to a high center of gravity and the instability caused by the previous curved, high-rocker pontoons (banana-shaped hulls). Therefore, the hull design was changed to flat-bottomed pontoons, and the mounting position of the electronics enclosure was lowered to reduce the center of gravity(Fig. 2).

*1) **Sensor Compatibility:*** Have enough buoyancy margin to carry the same sensors as WAM-V [2], which is about the size of a car.

*2) **Actuator Compatibility:*** Compatible with WAM-V in terms of actuator layout. For tasks such as Task2 and Task3, Blue Robotics T200 thrusters were employed to maintain a balance between speed and maneuverability.

*3) **Software Compatibility:*** Embedded middleware and computer configuration to allow the same software to be used with WAM-V and Mini-V.

B. Body Frame

Mini-V uses a MISUMI aluminum frame for the framework. This frame has been modified to accommodate the new hull design described later, specifically to lower the overall center of gravity. It is designed for easy mounting of various sensors.

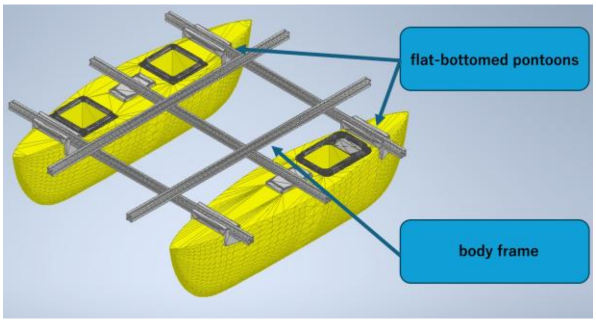

Fig. 2 Mini-V body frame

A LiDAR and camera are mounted in front. This mount is manufactured by sheet metal fabrication, but can be easily reproduced by MISUMI's Meivy service. A Blue Robotics T200 compatible motor [1] is mounted at the rear to provide differential turning motion.

C. Sensors

To achieve autonomous navigation and task execution, the upgraded Mini-V is equipped with a suite of sensors for both internal state estimation and external environmental perception. The sensor configuration is summarized below.

- M5 Stack (IMU & GNSS): Internal state estimation and positioning.
- Tier IV C1 Camera: Object recognition and color detection.
- Velodyne VLP-16 (LiDAR): Obstacle avoidance and 3D mapping.
- Pixhawk Flight Controller: Sensor fusion and low-level control.

The IMU and GNSS data are integrated into the Pixhawk flight controller, which performs state estimation using an Extended Kalman Filter (EKF). For external perception, the Velodyne VLP-16 and Tier IV C1 Camera are utilized to identify and localize competition elements such as buoys and gates.

D. *Circuit*

All circuits are integrated into the main unit's black case, except for the emergency stop system. The batteries installed in the Mini-V are all Makita BL1860B. This model was chosen because it provides sufficient power for over one hour of operation and is compliant with airline carry-on regulations, facilitating easy transport. Circuit design is performed using KiCAD, and an automation tool called KiBot is employed to automatically perform rule checking, Gerber generation, and BOM generation upon pushing the circuit data to GitHub. Ordering is also automated; with the push of a button, components can be purchased from Akizuki Denshi Tsusho, a Japanese electronic components retailer. Three types of custom circuit boards were manufactured for the Mini-V. The hardware is modular, mounted on 9×12 cm rectangular plates secured with arrow-shaped components. All mounts are fabricated with a 3D printer, allowing for on-site repairs in the event of damage.

1) *Computer Sensor Board*

The Computer Sensor Board supplies power to the Jetson AGX Orin computer [4] and other sensors mounted on the Mini-V. This board accepts 12 V DC as input and is equipped with a MOSFET to prevent reverse connection to prevent damage to the computer and sensors if accidentally connected (Fig. *3*).

2) *Mini-V Motor Controller Board*

The Mini-V Motor Controller Board supplies power and signals to the Blue Robotics T200 Basic ESC to control the motor. This board has a function to accept an emergency stop signal and receives a 3.3 V emergency stop release signal from the Mini-V E-stop Board. If the voltage of this emergency stops releases signal falls below 1.2 V, the relay that supplies power to the Blue Robotics T200 Basic ESC is shut down and the hull is safely brought to an emergency stop. This board receives the battery output directly to the board. This is because the drive voltage of the Blue Robotics T200 Basic ESC matches the voltage of the Makita BL1860B, so there is no need to use a DC/DC converter to keep the voltage constant (Fig. *4)*.

3) *Mini-V E-stop Board*

The Mini-V E-stop Board sends an emergency stop signal to the Mini-V Motor Controller Board. This board consists of two elements: a Teensy 4.1 controller for controlling LED tapes and a stop signal output circuit composed solely of hardware components. When the emergency stop button is not pressed, a voltage of 3.3 V is applied to pin 33 in the circuit diagram, and a 3.3 V emergency stop release signal is emitted from J7/J8. Pin 33 is pulled down so that when the emergency stop button is pressed, the voltage on pin 33 goes to 0 V, and the voltage of the emergency stop release signal consequently drops to 0 V. The Teensy controller that controls the LED reads the voltage on pin 33 and changes the color of the LED. This enables both the emergency stop and its status display to function without software intervention. (Fig. 5).

4) *Pixhawk*

The Pixhawk [6] runs ArduPilot firmware and handles remote communication with the ground control station (PC). It performs autonomous navigation using data from the connected GNSS and IMU, along with waypoints recorded in the firmware. Additionally, through serial communication with the onboard computer, it incorporates obstacle avoidance based on environmental data from the Camera and LiDAR. Finally, it generates and outputs the required control signals to the motor drivers.

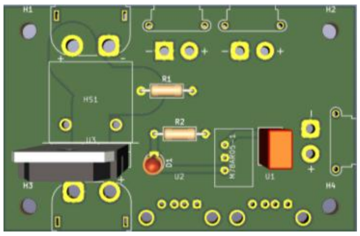

Fig. 3 Computer Sensor Board

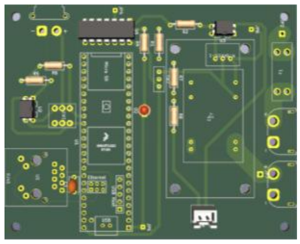

Fig. 4 Mini-V Motor Controller Board

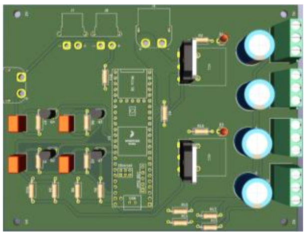

Fig. 5 Mini-V E-stop Board

E. *Software*

OUXT Polaris adopts Robot Operating System 2 (ROS 2) [7] as the core middleware for the Mini-V. Our software development philosophy centers on open-source contribution; all developed packages are released under the Apache License 2.0 and are publicly available on GitHub [8]. The architecture follows a modular design, enabling multiple team members to develop distinct subsystems (perception, control, communication) in parallel (Fig. 6).

1) *Low-Level Control: ArduPilot*

As described in the Hardware section, the Mini-V utilizes ArduPilot, an open-source autopilot software capable of controlling various autonomous vehicles including rovers and boats.

a) Sensor Fusion & Control

Running on a Pixhawk flight controller, ArduPilot performs self-localization and PID control by fusing data from the onboard GNSS and IMU with visual odometry information transmitted from the main ROS 2 computer.

b) ROS 2 Integration

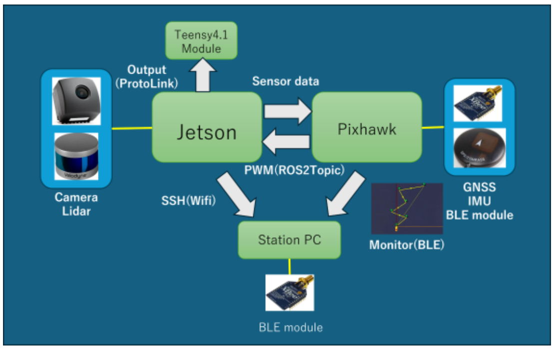

Fig. 6 System architecture

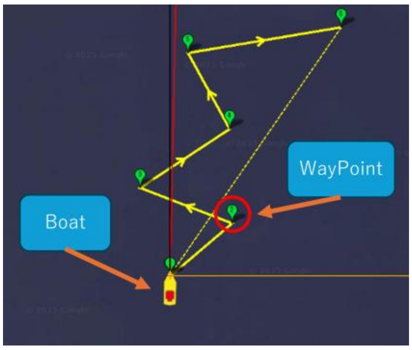

Fig. 7 Simulation with Pixhawk Mission planner

By utilizing dedicated libraries, we have enabled full compatibility between ArduPilot and ROS 2. Uniquely, the system is configured to output motor PWM values as ROS 2 topics. These values are then transmitted to the motor drivers via our custom "ProtoLink" interface (described below).

c) Simulation & Tuning

We utilize Mission Planner [9] for system calibration. This tool allows for precise tuning of PID gains and path planning parameters. It also supports Software-In-The-Loop (SITL) simulation, enabling us to verify behavior using internal models before deployment (Fig. 7).

2) *Perception: Hybrid Sensor Fusion*

Our perception stack processes environmental data in the following sequence to support autonomous navigation.

a) Visual Recognition (YOLO)

We employ YOLO [10], a vision-language model renowned for high-speed real-time processing, to extract object bounding boxes from camera images (Fig. 8).

b) LiDAR Clustering

Simultaneously, point cloud data from the Velodyne VLP-16 [11] is clustered using a rule-based algorithm.

c) Sensor Fusion

The system fuses the 2D visual data with 3D point cloud clusters to accurately determine the classification and 3D position of objects. This obstacle information is then transmitted to the Pixhawk to update the local cost map for avoidance.

3) *Embedded Communication: ProtoLink*

A significant challenge in robotics is bridging the communication gap between high-level ROS 2 computers (Jetson) and low-level microcontrollers (motor drivers/emergency stops). To address this, OUXT Polaris developed "ProtoLink," a mechanism to automate data serialization.

a) Technical Approach

Instead of running heavy DDS agents on microcontrollers (as with micro-ROS), ProtoLink focuses on lightweight serialization. The workflow is as follows.

- **Definition:** The system parses standard ROS 2 .msg files.

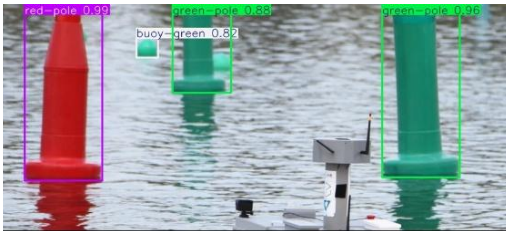

Fig. 8 Image inference via YOLO

- **Generation:** It automatically generates .proto files and subsequently creates C headers using nanopb, a lightweight ANSI-C implementation of Protocol Buffers designed for embedded systems. This library utilizes static memory allocation, preventing memory fragmentation and ensuring deterministic behavior on resource-constrained microcontrollers.
- **Communication:** Data is serialized and transmitted via USB-Serial or UDP without requiring a ROS layer on the MCU.

b) Advantages & Validation

This approach significantly reduces CPU load on the microcontrollers and eliminates manual protocol maintenance errors. Furthermore, because the communication protocol is defined by schemas but is independent of the ROS client library, it allows for robust teleoperation even if the ROS 2 system encounters issues. This middleware was recognized for its novelty and practicality, leading to an oral presentation at ROSCon JP 2024[12, 13].

II. Competition Strategy

A. *Overview*

Our team's performance in RoboBoat 2025 was compromised by a capsize during autonomous navigation, which prevented task completion. Consequently, our strategy for RoboBoat 2026 shifts from "attempting all tasks" to "guaranteed completion of fundamental tasks." We have adopted a phased approach, prioritizing stability and the sequential validation of our autonomy stack.

B. *Primary Objectives*

Our highest priority is the reliable completion of Task 1 and Task 2, as they form the foundation of maritime autonomy [14].

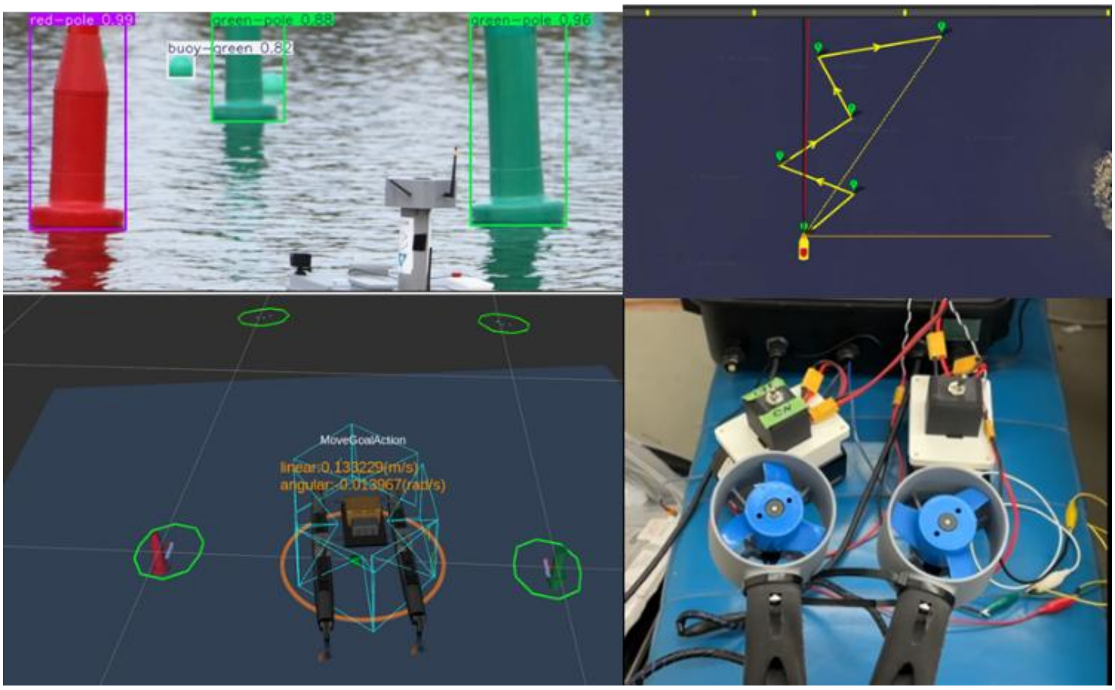

Fig. 9 Testing Strategy

1) ***Task 1 (Entry & Exit Gates)***

The ASV utilizes a sensor fusion of Camera (YOLO) and LiDAR to detect the nearest pair of large red and green markers forming a gate. The system calculates a relative waypoint located at the geometric midpoint between these markers. The vessel navigates to this waypoint, repeating the process until it successfully passes through two gates.

2) ***Task 2 (Navigation Channel)***

This task extends the navigation logic of Task 1. While traversing the channel, the ASV uses YOLO to detect black obstacles and executes avoidance maneuvers if one is identified. For the mapping component,

- Red Indicator: Upon detection, the system records the object's position by cross-referencing it with GNSS data.
- Green Indicator: Similarly, the system records the position and executes a circumnavigation maneuver around the buoy using LiDAR data to maintain a safe distance.

C. *Secondary Objective*

Upon the successful execution of the primary objectives, we will attempt Task 3 and Task 5. These tasks leverage the same perception stack but require more complex maneuvering logic.

1) ***Task 3 (Maneuvering)***

The ASV halts in front of the gate to analyze the light signal. Based on the signal color, the system generates a sequential waypoint path to describe a square shape around the yellow buoy—either in a clockwise or counter-clockwise direction—before returning to the gate.

2) ***Task 5 (Automated Docking)***

The ASV employs YOLO-based image recognition to identify the dock numbers and status indicators, selecting the optimal bay that meets the competition criteria. The final approach and precise stopping position are determined using range data from the LiDAR.

D. *Strategic Trade-off Regarding Payload*

To maintain the "Mini-V" concept of portability and low center of gravity, we made a strategic decision to omit the heavy payloads required for **Task 4 (Object Delivery)** and **Task 6 (Sound Signal)**. This decision minimizes the risk of instability and maximizes our probability of completing the prioritized courses safely.

III. Testing Strategy

The most fundamental verification of our system is performed through live tests on the water. These tests are primarily conducted at the swimming pool located within the Kyushu Institute of Technology campus. During these sessions, we validate the operation of the underwater vehicle's critical subsystems, including thruster actuation, battery management, emergency stop mechanisms, and remote communication links. Furthermore, we verify data acquisition and performance for various sensors such as GNSS, IMU, cameras, and LiDAR.

However, relying solely on on-water testing presents several challenges. It is difficult to fully replicate complex competition-specific tasks in the limited space of a pool. Additionally, the logistics of transporting the vehicle and the extensive preparation required create a significant burden on the team. To address these limitations, our team established a simulation environment utilizing Pixhawk and Docker containers. This platform allows us to conduct autonomous navigation experiments, such as waypoint tracking, in a virtual space that exceeds the physical constraints of the university pool.

Components undergo initial verification in the laboratory. We conduct fundamental electrical and functional tests to ensure reliability prior to integration (Fig. 9).

Acknowledgements

OUXT Polaris gratefully acknowledges the financial support provided by the Kyushu Institute of Technology (KIT), Eight Knot Inc., Japan Marine Heavy Industries, Inc., and Alumnote Inc.

We also express our deepest appreciation to our faculty advisor, Associate Professor Yuya Nishida of the Kyushu Institute of Technology, for providing the essential expertise required for our system development. Finally, we would like to extend our sincere thanks to our alumni for their generous technical guidance and mentorship throughout the project.

References

[1] Blue Robotics, "T200 Thruster," [Online]. Available: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/. [Accessed: Jan. 9, 2026].

[2] Ocean Power Technologies, "Wave Adaptive Modular Vehicle (WAM-V®)," [Online]. Available: https://oceanpowertechnologies.com/products/unmanned-surface-vehicles/. [Accessed: Jan. 9, 2026].

[3] OUXT Polaris, "OUXT Polaris Development Automation Tool: Circuit," [Online]. Available: https://ouxt-polaris.github.io/ouxt_automation/circuit/computer_sensor_board/. [Accessed: Jan. 9, 2026].

[4] NVIDIA, "NVIDIA Jetson Orin," [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/. [Accessed: Jan. 9, 2026].

[5] PJRC, "Teensy® 4.1 Development Board," [Online]. Available: https://www.pjrc.com/store/Teensy41.html. [Accessed: Jan. 5, 2026].

[6] Pixhawk, "Pixhawk Homepage," [Online]. Available: https://pixhawk.org. [Accessed: Jan. 5, 2026].

[7] Open Robotics, "ROS 2 Documentation (Humble)," [Online]. Available: https://docs.ros.org/en/humble/index.html. [Accessed: Jan. 5, 2026].

[8] OUXT Polaris, "OUXT-Polaris GitHub Repository," [Online]. Available: https://github.com/OUXT-Polaris. [Accessed: Jan. 7, 2026].

[9] ArduPilot Dev Team, "Mission Planner," [Online]. Available: https://ardupilot.org/planner/. [Accessed: Jan. 7, 2026].

[10] Ultralytics, "YOLO11 Documentation," [Online]. Available: https://docs.ultralytics.com/. [Accessed: Jan. 7, 2026].

[11] Ouster, "VLP-16 (Velodyne Lidar)," [Online]. Available: https://ouster.com/products/hardware/vlp-16. [Accessed: Jan. 7, 2026].

[12] OUXT Polaris, "Autonomous Navigation Hardware and Integration with ROS 2 for the Maritime RobotX Challenge 2024," presented at ROSCon JP 2024, Oct. 2024. [Online]. Available: https://roscon.jp/2024/presentations/06.pdf. [Accessed: Jan. 7, 2026].

[13] OUXT Polaris, "Autonomous Navigation Hardware and Integration with ROS 2 for the Maritime RobotX Challenge 2024," [Video]. ROSCon JP 2024. Available: https://vimeo.com/showcase/11452054?video=1029114561. [Accessed: Jan. 7, 2026].

[14] RoboNation, "RoboBoat Section 3: Autonomy Challenge," [Online]. Available: https://robonation.gitbook.io/roboboat-resources/section-3-autonomy-challenge. [Accessed: Jan. 4, 2026].