RoboBoat 2026: Technical Design Report Kapsül Yazgit Eternal Team

Ceren Güneş, Muhammed Can Önaçan, Gayenaz Kaynar, Ravza Gül Doğan, Uğur Atalay, Fatih Uluçam, Ayşenur Ramazanoğlu, Ahmet Fatih Akyüz, Muhammed Ertuğrul Dursun , Beyza Nur Bakır, Mert Özen, Onur Danişment
Konya Technical University, Turkey, Konya

***Abstract*— This report presents the engineering design processes and operational strategies of the autonomous marine platform developed by Team Eternal for the RoboBoat 2026 competition. A rigorous systems engineering discipline was adopted during the design phase, optimising component integration and system validation. This facilitated maintenance and iterative development processes.**

The vehicle's propulsion system consists of two independent propulsion units, providing the dynamic manoeuvring capability required for complex maritime tasks. The software architecture is based on the ROS (Robot Operating System) framework. Data communication between subsystems is managed wirelessly via high-speed radio links. An Intel RealSense depth camera is integrated, forming a robust perception layer enhanced with special computer vision algorithms for environmental awareness and object detection. To minimise operational risks and maximise software efficiency, the system's digital twin was extensively tested in the Gazebo simulation environment prior to physical field trials. This integrated methodology aims to ensure the vehicle successfully performs autonomous tasks in real marine conditions.

Keywords- autonomous surface vehicle, robot operating system, systems engineering, modular design, digital twin, computer vision, object detection, RoboBoat 2026

ASV	**:Autonomous Surface Vehicle**
ConOps	**:Concept of Operations**
DMS	**:Decision-Making System**
ESC	**:Electronic Speed Controller**
FB	**:Fuse Board**
OCS	**:Operator Control Station**
PCB	**:Printed Circuit Board**
PM	**:Power Manager**
ROS2	**:Robot Operating System 2**
SBC	**:Single Board Computer**
SOM	**:Special Operations Module**
VCU	**:Vehicle Control Unit**
VPU	**:Vision Processing Unit**

I. COMPETITIVE STRATEGY

A. SYSTEM COMPLEXITY AND RELIABILITY BALANCE

Our autonomous surface vehicle (ASV) is equipped with a multi-sensor detection system to detect water surface reflections, buoys, and debris [21], [23]on Roboboat's complex and hazardous routes, avoiding them when necessary, and to provide reliable and instantaneous navigation mapping. The accuracy of autonomous tasks is proportional to the accuracy of the detection system. Although a detection system based on data from a single sensor may provide adequate performance in some conditions, it causes risks throughout the system. Such risks lead to detection errors and cause chain errors in decision-making and control layers [31], [32]. Accurate navigation information for Roboboat is critical for reliable autonomous behavior [29], [33]. A multi-sensor perception system has been used to mitigate these risks [4], [6]. However, since a large amount of data can increase software and integration complexity [27], [28] it has been simplified to a manageable level. As a result of these guidelines, cameras and LIDAR have been chosen [26] to provide visual and distance-based detection. This choice ensures controllable integration while increasing reliability, in addition to providing complementary detection.

II. SYSTEM DESIGN STRATEGY

A. GENERAL SYSTEM ARCHITECTURE

The general architecture of the ASV is designed with a modular and hierarchical approach, ensuring a seamless integration between hardware components and software layers [27], [28]. The system is divided into three primary domains: Physical/Mechanical, Electronic/Power, and Computational/Software.

Mechanical Domain: A catamaran-style fiberglass hull provides a stable and hydrodynamic platform [3], [23] for all components.

Electronic Domain: Centered around a dual-battery configuration (16000mAh and 5000mAh), the power distribution system ensures isolated and stable energy flow to high-current motors (via Flycolor 120A ESCs) and sensitive computing units (Jetson Orin NX and Pixhawk Orange Cube) [17], [25].

Software Domain: Built on the ROS2 (Robot Operating System 2) framework [9], [10] the software architecture utilizes a node-based communication structure. It integrates a Perception Layer (YOLOv8 and LiDAR fusion) [26], a Decision-Making Layer (Behavior Tree-based task management), and a Control Layer (ArduPilot/MAVLink) [29], [31] to execute complex autonomous missions.

This integrated architecture allows for real-time data processing, adaptive obstacle avoidance [21], [26] and robust mission execution, even under varying maritime conditions. The

use of a digital twin in the Gazebo environment ensures that this architecture is validated before physical deployment [44], [45].

B. PERCEPTION SYSTEM

To enable Roboboat to provide uninterrupted navigation in challenging water surface conditions (reflection, light changes), a Multi-Mode Sensor Fusion architecture has been adopted [21]. In the system, semantic data (object type/color with YOLOv8) from the Intel RealSense D455 camera is combined with precise geometric data (range/position) from the RP LiDAR S2. This hybrid structure eliminates individual sensor errors by validating visual data with point clouds and provides reliable mapping for autonomous tasks.

1) Computer Vision: ASV has a single camera architecture for environmental perception [26]. The image processing load is handled by the YOLOv8-S model running on NVIDIA Jetson, not on the sensor. This model was chosen for its long-range detection success and is optimized to minimize response times.

2) Obstacle Avoidance: The system applies the Camera-LiDAR Sensor Fusion principle to eliminate false detections [26]. Semantic data (class/color) from the camera is combined with geometric depth data from the LiDAR to provide input to the relevant task algorithms [31].

3) Positioning: GPS data is used for real-time position accuracy, while LiDAR data is used for local mapping and odometry. All data is combined using Extended Kalman Filter (EKF) or SLAM algorithms to ensure optimal route tracking.

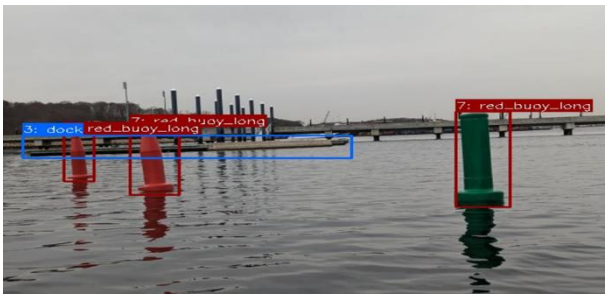


Figure 1-Bouy Detection

C. SENSOR FUSION AND DECISION VALIDATION

To increase accuracy and prevent potential errors during the competition, sensor fusion is positioned as a layer between perception and control [22]. Our fusion, developed using data obtained from RP lidar S2-based range perception and geometric information, along with shape and class data from the camera, makes decisions by verifying the reliability of perception [21], [23]. For an object detected in the lidar data to be accepted as a target, it must match the images processed by YOLOV8 on the camera [24]. This strategy both reduces false detections reflected from the water surface and increases the accuracy of the distance required for navigation [25]. As a result, reliable predictions are provided for both mapping and detection [26].

III. AUTONOMOUS SYSTEM DESIGN

The architecture of our system is based on the Robot Operating System (ROS2) infrastructure [27]. ROS2 offers a distributed software architecture that enables the interaction of nodes operating in parallel and communicating through topics [28]. Data exchange is carried out using pre-defined message types over these topics; each node can only read data, write data, or perform both operations within the communication structure [27]. This modular approach creates a flexible, scalable, and reliable environment among system components.

As shown in Addition Figure 6, the proposed autonomy architecture consists of four main components: the perception layer, the decision-making layer, the control layer, and the vehicle control layer [30].

A. Perception Layer

The perception layer is responsible for collecting environmental data and transmitting it to the autonomy system **[31]**. This layer uses RPLIDAR S2, Intel RealSense D455 sensor, and GPS/IMU data provided through Pixhawk **[32]**. Raw data from the sensors are processed through ROS2 nodes, converted into standard ROS messages, and forwarded to the decision-making layer **[27], [31]**. This layer aims solely to provide environmental awareness and does not include any decision-making mechanism **[31]**.

B. Decision Layer

The decision-making layer is a core component of autonomous navigation, continuously evaluating the system state by integrating environmental data from the perception layer with vehicle status information from the control layer [33]. In the proposed architecture, decision-making is implemented using a Behavior Tree (BT) approach [34]. BTs enable hierarchical and condition-based behavior selection, allowing tasks such as target seeking, approaching, obstacle avoidance, and mission completion to be prioritized and activated dynamically based on environmental conditions [35]. This structure allows the system to make flexible and reliable autonomous decisions, rather than following a fixed task sequence, making it well-suited for race scenarios [36].

C. Control Layer

The control layer is responsible for converting the high-level navigation decisions generated by the decision-making layer into movement commands that the vehicle can physically execute **[37]**. In this layer, speed and direction (yaw) control is performed, and the obtained control outputs are produced in the form of cmd_vel messages **[27]**. The generated motion commands are transmitted to the vehicle control layer through the ROS–MAVROS interface **[38]**. This structure serves as a reliable bridge between autonomous algorithms and low-level vehicle control **[38]**.

D. Vehicle Control Layer

The vehicle control layer is managed by the ArduPilot software running on the Pixhawk flight controller **[39]**. This layer is responsible for low-level control operations, ensuring precise driving of the motors and thrusters **[40]**. Speed,

orientation, and stabilization control are achieved using PID controllers, ensuring the boat moves smoothly and stably **[41]**. Furthermore, the vehicle control layer feeds back the boat's position and orientation information to the upper layers via ROS **[27], [38]**.

An autonomous surface vehicle competing in a race environment must exhibit predictable and stable system behavior under varying operational conditions **[42]**. For this reason, the Robot Operating System (ROS2) infrastructure was selected due to its maturity, extensive documentation, and widespread use in real-world and academic applications **[27], [28]**. The well-established ROS2 ecosystem offers strong support for system integration, modular development, and debugging, which significantly reduces the risk of unexpected behaviors during competition scenarios and ensures reliable overall system performance **[29]**. As a result, the adoption of ROS2 contributes directly to the robustness, reliability, and competitive readiness of the autonomous surface vehicle by enabling consistent and controllable system operation throughout the race **[36], [42]**.

IV. AUTONOMOUS NAVIGATION SYSTEM ARCHITECTURE

The autonomous navigation system architecture refers to the configuration of software components within the overall system architecture in a way that enables the vehicle to move safely and mission-oriented based on its environmental perceptions **[43]**. The architecture consists of nodes that run on the ROS2 infrastructure and communicate on a topic-based system **[27], [28]**.

At the core of the system is a Behavior Tree (BT)-based task manager that continuously evaluates environmental data from the perception layer and vehicle status information **[34]**. This structure addresses navigation behaviors such as target search, target approach, obstacle avoidance, and task completion in a hierarchical and conditional manner, ensuring that the most appropriate behavior for the current situation is dynamically selected **[34], [35]**.

In accordance with the active behavior enabled by the Behavior Tree, the navigation sub-modules calculate the target direction and speed information; the obtained outputs are transmitted to the control layer in the form of cmd_vel messages **[27], [37]**. This way, high-level autonomous navigation decisions are converted into control commands that directly affect the vehicle's physical movement, and the system exhibits flexible and reliable navigation performance that is compatible with racing scenarios **[36], [42]**.

A. Software Implementation Details

The Python programming language was used in the development of autonomous navigation software **[44]**. Python's high-level structure allows for the rapid processing of raw data obtained from sensors and the transformation of complex mathematical calculations into logical decision-making mechanisms **[45]**. This allows condition-based navigation decisions, such as "turn left when the right buoy is detected," to be implemented within a flexible and readable software architecture.

B. Autonomous Flow System Security Decision

While the system is operating, sensor, hardware, and overall system health are continuously monitored; detected error conditions are classified by type, and dynamic decisions are made for Safe Return in critical situations and Continue in Task in non-critical situations.This structure is designed to be compatible with a behavior tree-based decision-making approach. Figure 2 shows the safety-focused decision-making and error management architecture designed for an autonomous navigation system

V. TASK STRATEGY

Throughout the RoboBoat 2026 Autonomous Competition, the ASV is intended to perform all tasks autonomously [13], [14], [40]. In this regard, motion control is guided by computer vision-based perception outputs, while task management and behavior selection are carried out by a behavior tree-based decision-making mechanism [**27], [29].**

A. Task 1

Under Task 1, the ASV must navigate completely autonomously in the navigation channel consisting of two red and green buoys **[13], [14].** The behavior tree for this task can be simplified to the following steps:
Unless Asv gets through the door:

- Find the door.
- Set a waypoint after the door and proceed.
- Repeat for the next door.
- Mission accomplished.

B. Task 2

After completing its first task, the ASV will search for the small green and red buoy pair that marks the beginning of the second task **[13], [14]**. In the debris cleaning task, our vehicle will follow the behavior tree below:
Unless the ASV detects a red-green buoy:

- Find the buoy pair
- Enter the wreckage
- Report the red buoys
- Circle the green buoys
- Avoid the black buoys
- Identify the end of the course
- Mission accomplished.

C. Task 3

After completing the second task, the ASV will search for the gate consisting of a pair of small green and red buoys that marks the beginning of the third mission **[13], [14].** In the *Emergency Response Acceleration* task, our vehicle will follow the behavior tree below **[27],[29]:**

After finding the ASV starting gate:

- Sense color from the light buoy
- Avoid obstacles
- Reach the yellow buoy
- Turn according to the light color
- Exit from the starting gate
- Mission accomplished.

D. Task 4

After the third mission is completed, our vehicle autonomously delivers materials to rescue ships fixed along the track during the Material Delivery mission **[13],[14]**. The yellow boats represent the clean water needs, and the ASV sprays water onto the black triangles on these boats. The black boats represent the urgent need for medical supplies, and the ASV delivers objects to these boats. The task execution and action sequencing are managed through a behavior tree-based mission structure **[27], [29].**

E. Task 5

After the 4th task is completed, our ASV vehicle will search for the docks which mark the beginning of the Marina Tour task. Once the docks are found:

- Enter the harbor.
- Perform a 360-degree scan
- If the marina is empty
- If the marina is marked with a green indicator
- Approach the marina
- Task completed successfully.

F. Task 6

After completing the task, within the scope of the *Port Alarm* mission, the ASV immediately terminates its current task when it detects an auditory emergency signal **[13], [14].** A single detected signal triggers the ASV to head toward the emergency response area, while two sound signals trigger the ASV to return to the designated marina. This task tests the system's ability to respond quickly and accurately to dynamic and high-priority emergencies, which is a key requirement in modern autonomous marine vehicle mission management architectures **[27], [29].**

VI. DESIGN STRATEGY

This section details the architectural designs of the core subsystems and the engineering rationale behind these designs.

A. Mechanıcal Desıgn

Performance analyses conducted after the last racing season have shown that system success depends not only on the individual efficiency of the subsystems but also on the sustainability of the system integration and field test processes. The previous ASV had a single-body structure and a large-scale architecture; it was determined that it could not fully meet the challenges in logistics processes and the dynamic stability required by competitive conditions **[3], [24].** For this reason, *lightness, modularity, and portability* have been prioritized to increase operational flexibility, which is consistent with modern ASV design trends emphasizing modular platforms and reconfigurable system architectures **[5], [11], [22], [25].**

B. Body and Configuration Selection

Since the previous ASV could not satisfy current requirements, a new platform was designed from scratch **[13], [14].** Monohull, trimaran, and SWATH concepts were evaluated based on stability, maneuverability, balance, and hydrodynamic resistance, and a catamaran was selected for RoboBoat scenarios due to its superior lateral stability and lower drag compared to the former monohull design **[3], [24].**

The dual-hull configuration increases lateral stability and metacentric height, improving resistance to environmental disturbances and providing a stable sensing platform **[24].** The ASV's weight and dimensions were minimized to enhance logistics and portability, reducing the total mass to 20 kg, with further optimization ongoing **[5], [11].** A fully symmetrical hull geometry was adopted to improve maneuverability, docking accuracy, and predictability of autonomous control responses **[15], [19].**

To mitigate the negative effects of roll on perception performance observed in previous projects, an inverted bow geometry was implemented to reduce wave-induced roll amplitude and acceleration **[3], [24].** Additional stability gains were achieved by reducing composite layers, optimizing body dimensions, and relocating heavy components such as batteries and ESCs to the pontoons, lowering the center of gravity and increasing metacentric height **[11], [22], [24].**

The hull is manufactured in collaboration with Aybig Design using fiberglass (GRP) casting with wooden molds, selected for its low weight, high impact and corrosion resistance, and low surface roughness that reduces drag **[3], [20], [22].** Overall, the design follows a modular, systems engineering-driven approach that builds on a proven hardware baseline while prioritizing software and perception enhancements to minimize operational risks and ensure repeatable mission success **[1], [5], [14].**

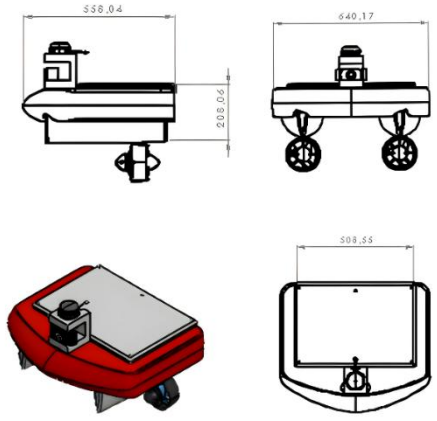


Figure 2-Technical Design

C. Suction System

Building on the design experience from our previous prototypes, ASV Eternal is equipped with a differential thrust propulsion system utilizing two T200 thrusters **[15], [19**]. This configuration maximizes maneuverability by enabling zero-turn capability, which is a common approach in modern small-scale ASV platforms **[3], [20**]. Furthermore, it provides system redundancy to ensure operational continuity in the event of a motor failure, while optimizing cruise stability by counterbalancing propeller torque **[11], [24].**

D. Object and Water Delivery Mechanism

To minimize weight and maximize energy efficiency, the system draws water directly from the sea instead of using an onboard tank, a method frequently adopted in lightweight ASV designs to reduce payload mass and improve endurance **[18], [20].** A hull-mounted pump delivers water to a flat fan nozzle, which ensures a wide spray angle to reduce targeting errors during dynamic mission conditions **[21].** Fully integrated with the stereo camera and LiDAR suite, the ArduPilot-based controller modulates the launch timing and angle in real time for precise mission execution **[9], [10], [27].** The ball launcher utilizes counter-rotating wheels wrapped with high-friction belts. This belt-driven architecture increases the contact surface area, ensuring the balls are compressed and accelerated without slipping, a mechanism commonly employed in compact robotic delivery systems **[23].** For mission execution, the ASV approaches the target vessel closely, enabling the system to deploy the balls onto the deck at a controlled, low velocity to avoid structural damage and ensure safe material transfer **[14], [20].**

VII. TESTİNG STRATEGY

The Gazebo simulation environment was used in the process of developing and testing the autonomous system.Gazebo enables a realistic simulation of the ASV's dynamic motion model, sensor outputs (LIDAR, camera, GPS/IMU), and environmental interactions.The developed behavior tree-based autonomy software was tested in the Gazebo environment before being used on actual hardware, and task scenarios, safety conditions, and error scenarios were verified.This has increased the system's stability and allowed for the early detection of potential errors, ensuring a safer transition to real-world field tests.

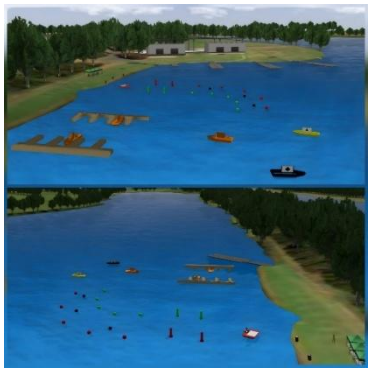


Figure 3- Gazebo simulation environment

VIII. GENERAL ELECTRICAL SYSTEM

A battery is a device that can store electricity. The types of batteries commonly used in society are as follows: Lead Acid, Ni-Cd, Li-Ion, and Li-Po. Among all the batteries mentioned above, Li-Po batteries are one of the most widely used.Because Li-Po batteries have many advantages such as high capacity in a small physical size, no memory effect, and the ability to provide currents up to 20 times higher.For these reasons, a Li-Po battery was preferred in our unmanned surface vehicle.

In electronic installations, power and signal are separated. The separation of power and signal was aimed at improving the efficiency of signal cables, which are more sensitive due to materials like motors and actuators that draw high current, and at preventing various issues such as noise.For this reason, two batteries were selected: a 22.2V 16000mAh LiPo and a 4S 5000mAh Li-Po. The 16000mAh LiPo battery was used as the main battery, and power for the DC water pump and DC servo motors was distributed via the Matek PDB power board.

Table 1-Noise and EMI Solitions

Problem: Noise and EMI caused by power cables.
Solutions:
The signal and power cables are separated by considering the balance of weight in the layout.
The power cables (+) and (−) are assembled together. (To prevent magnetic flux from canceling each other out.)
The power unit is housed in a box similar to a Faraday cage.
Devices like Jetson, Pixhawk Orange Cube, and microcontrollers draw power from a different battery rather than the power supply line.

A. Electronic System Strategy

The electronic subsystem designed for the RoboBoat 2026 competition was primarily tailored to the targeted missions **[13], [14].** The ASV electronic system architecture was conceived as a comprehensive framework encompassing all subsystems, including autonomous control, power distribution, image sensing, and communication **[27],[29].** In terms of material selection, factors such as efficiency and weight were considered first. All power and signal data connections are shown in Figure 3.

The The 16,000mAh main battery supplies power to the quadcopter motors, firing mechanism motors, water spray system pump, and servos. A separate 16,000mAh LiPo auxiliary battery powers critical components such as the Pixhawk Orange Cube, Jetson Orin NX, and LiDAR, which initially draws a 2A current, providing sufficient operational capacity for all mission tasks. For safety, an emergency stop button will be implemented to instantly cut off power. In addition, 100A fuses will be placed before each of the parallel-connected 120A ESCs, 20A automotive blade fuses before the 30A ESCs, and a 5A fuse before the Jetson Orin NX. Flycolor waterproof 120A ESCs are selected for the propulsion motors due to their high efficiency and waterproof design. The Pixhawk Orange Cube operates synchronously with the NVIDIA Jetson Orin NX, transmitting PWM control signals to motors and subsystems. Autonomous navigation is achieved using radar, LiDAR, and camera sensors to accurately detect obstacles and prevent collisions through algorithm-based avoidance.

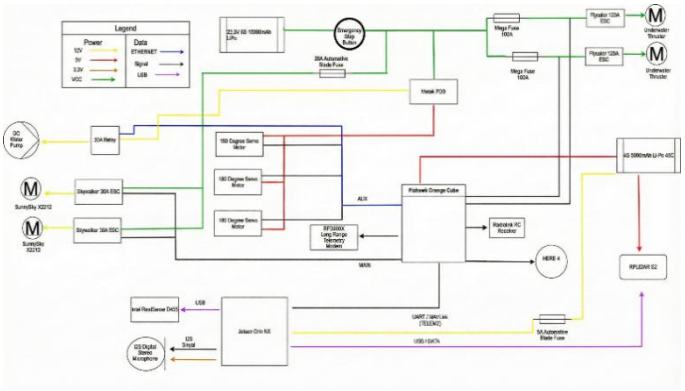


Figure 4-Electronic System

IX. COMMİNATİON SYSTEM DESİGN

The Orange Cube (Cube Pilot) autopilot hardware was chosen for its sensor redundancy and vibration damping features to manage autonomous navigation tasks **[29].** The internal damping system, which can isolate mechanical vibrations at sea, ensures the stability of sensor data (IMU) **[45].** Orange Cube works fully compatible with MAVLink, which we use as our communication protocol, playing a role in the lossless processing of data packets **[27], [28].**

A. COMMUNICATION PROTOCOL

The data flow between the vehicle and ground station is managed using the MAVLink 2.0 protocol **[27], [28].** MAVLink 2.0 reduces the data payload by transmitting telemetry data in binary packets **[27].** This optimization prevents network congestion by reducing bandwidth usage on the RFD900x line and preserves data integrity in long-range communication **[7], [33].**

B. TELEMETRY SYSTEM

The telemetry communication system design provides the command-and-control link between the ground control station (GCS) and the autonomous surface vehicle (ASV) using RFD900x long-range (24+ miles) telemetry modems **[7], [33].** The system operates in the 902–928 MHz ISM band, which is resistant to reflection effects (multipath fading) on the sea surface **[7].** The RFD900x creates a robust and secure data link against signal interference thanks to its high output power (1 Watt) and frequency hopping spread spectrum (FHSS) technology **[7], [33].** Signal quality is also enhanced through the use of antenna diversity **[33], [36].**

C. INTERNAL DATA LINK

There is a serial (UART) connection between the mission computer (Jetson) and the autopilot (Orange Cube). Jetson processes data from the lidar and camera to detect obstacles and calculate the route. This route is converted into MAVLink protocol packets and sent to the Orange Cube via the TELEM2 port. This architecture provides maximum efficiency and stability by distributing the total processing load.

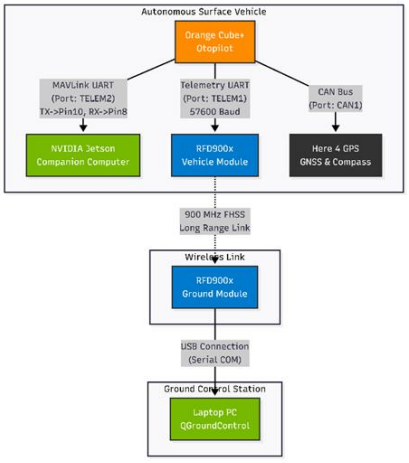


Figure 5-Configuration system block diagram

X. ACKNOWLEDGMENTS

We, the ETERNAL team, express our gratitude to all our stakeholders who have guided us and supported us at every stage of our project. First and foremost, we express our deepest gratitude to our advisor, Prof. Dr. Murat Ceylan, who guided our work with his knowledge and valuable advice. We thank the KAPSÜL Technology Platform for its significant contributions through the material support it has provided to our project. We are grateful to the AYBIG Design team for their technical contributions to our production process, especially in the areas of mold and body manufacturing. Furthermore, it was supported by transportation sponsorships provided under the sponsorship of TÜBİTAK and in cooperation with Turkish Airlines; we thank them for this valuable support. Finally, we express our gratitude to AGET Desing for the financial support they have provided for the sustainability and development of our project. Thanks to these supports, we are one step closer to our goals as the ETERNAL Team.

REFERENCES

[1] J. Moulton, N. Karapetyan, S. Bukhsbaum, C. McKinney, S. Malebary, G. Sophocleous, A. Q. Li, and I. Rekleitis, "An autonomous surface vehicle for long term operations," in *Proc. IEEE OCEANS*, Charleston, SC, USA, Oct. 22–25, 2018, pp. 1–10.

[2] J. E. Manley, "Unmanned surface vehicles, 15 years of development," in *Proc. IEEE OCEANS*, Quebec City, QC, Canada, Sept. 15–18, 2008, pp. 1–4.

[3] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi, and C. Laschi, "The HydroNet ASV, a small-sized autonomous catamaran for real-time monitoring of water quality: from design to missions at sea," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 3, pp. 710–726, 2015.

[4] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, p. 399, 2013.

[5] A. Martorell-Torres, M. Massot-Campos, E. Guerrero-Font, and G. Oliver-Codina, "Xiroi ASV: a modular autonomous surface vehicle to link communications," *IFAC-PapersOnLine*, vol. 51, pp. 147–152, 2018.

[6] Z. Zhou, J. Liu, and J. Yu, "A survey of underwater multi-robot systems," *IEEE/CAA J. Autom. Sin.*, vol. 9, pp. 1–18, 2022.

[7] M. Stojanovic, "Underwater acoustic communications," in *Proc. Electro Int. Conf.*, 1995, pp. 435–440.

[8] M. Carreras, J. D. Hernandez, E. Vidal, N. Palomeras, D. Ribas, and P. Ridao, "Sparus II AUV—A hovering vehicle for seabed inspection," *IEEE J. Ocean. Eng.*, vol. 43, pp. 344–355, 2018.

[9] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *Proc. ICRA Workshop Open Source Software*, 2009.

[10] Stanford Artificial Intelligence Laboratory, "Robot operating system," 2021. [Online]. Available: https://www.ros.org

[11] A. Odetti, G. Bruzzone, M. Altosole, M. Viviani, and M. Caccia, "SWAMP, an autonomous surface vehicle expressly designed for extremely shallow waters," *Ocean Engineering*, vol. 216, 108205, 2020.

[12] R. Millet, F. Plumet, and J. C. Dern, "Autonomous surface vehicles for oceanographic survey," in *Proc. Int. Autonomous Surface Ship Symp.*, Paris, France, Jan. 2008.

[13] K. Tanakitkorn, "A review of unmanned surface vehicle development," *Marit. Technol. Res.*, vol. 1, pp. 2–8, 2019.

[14] V. A. Jorge et al., "A survey on unmanned surface vehicles for disaster robotics: main challenges and directions," *Sensors*, vol. 19, p. 702, 2018.

[15] A. F. Zakki, A. Triwiyatno, B. Sasmito, and K. Budiono, "Design and control of autonomous surface vehicle to support bathymetry survey in the coastal environment," *Adv. Sci. Technol. Eng. Syst.*, vol. 4, pp. 458–461, 2019.

[16] M. H. B. M. Idris et al., "Design and development of an autonomous surface vessel for inland water depth monitoring," in *Proc. ICCCE*, Kuala Lumpur, Malaysia, 2016, pp. 177–182.

[17] H. Bayram, "Design and implementation of autonomous surface vehicle for inland water," *J. Inst. Sci. Technol.*, vol. 10, pp. 101–111, 2020.

[18] K. Sornek et al., "Development of a solar-powered small autonomous surface vehicle for environmental measurements," *Energy Convers. Manag.*, vol. 267, 115953, 2022.

[19] W. Wang et al., "Design, modeling, and nonlinear model predictive tracking control of a novel autonomous surface vehicle," in *Proc. IEEE ICRA*, Brisbane, Australia, 2018, pp. 6189–6196.

[20] D. F. Carlson et al., "An affordable and portable autonomous surface vehicle with obstacle avoidance for coastal ocean monitoring," *HardwareX*, vol. 5, 2019.

[21] H. Cao et al., "Intelligent wide-area water quality monitoring and analysis system exploiting unmanned surface vehicles and ensemble learning," *Water*, vol. 12, p. 681, 2020.

[22] G. Stanghellini, F. D. Bianco, and L. Gasperini, "OpenSWAP, an open architecture, low cost class of autonomous surface vehicles for geophysical surveys in shallow water," *Remote Sens.*, vol. 12, p. 2575, 2020.

[23] R. A. Regina et al., "Hull and aerial holonomic propulsion system design for optimal underwater sensor positioning in autonomous surface vessels," *Sensors*, vol. 21, p. 571, 2021.

[24] A. T. Olaoye and S. Brizzolara, "ASV operability at sea: size matters as much as hull form design," in *Proc. IEEE OCEANS*, Monterey, CA, USA, 2016.

[25] A. Odetti, M. Altosole, G. Bruzzone, M. Viviani, and M. Caccia, "A new concept of highly modular ASV for extremely shallow water applications," *IFAC-PapersOnLine*, vol. 52, pp. 181–186, 2019.

[26] T. C. Furfaro, J. E. Dusek, and K. D. Von Ellenrieder, "Design, construction, and initial testing of an autonomous surface vehicle for riverine and coastal reconnaissance," in *Proc. IEEE OCEANS*, Biloxi, MS, USA, 2009.

[27] J. Pinto et al., "Implementation of a control architecture for networked vehicle systems," *IFAC Proc. Vol.*, vol. 3, pp. 100–105, 2012.

[28] J. Pinto et al., "The LSTS toolchain for networked vehicle systems," in *Proc. IEEE OCEANS*, Bergen, Norway, 2013.

[29] N. Palomeras, A. El-Fakdi, M. Carreras, and P. Ridao, "COLA2: a control architecture for AUVs," *IEEE J. Ocean. Eng.*, vol. 37, pp. 695–716, 2012.

[30] N. Karapetyan et al., "Multi-robot Dubins coverage with autonomous surface vehicles," in *Proc. IEEE ICRA*, Brisbane, Australia, 2018, pp. 2373–2379.

[31] S. Manjanna et al., "Heterogeneous multi-robot system for exploration and strategic water sampling," in *Proc. IEEE ICRA*, Brisbane, Australia, 2018, pp. 4873–4880.

[32] Y. Singh et al., "A novel double layered hybrid multi-robot framework for guidance and navigation of unmanned surface vehicles," *J. Mar. Sci. Eng.*, vol. 8, p. 624, 2020.

[33] G. Vasiljevic et al., "Dynamic median consensus for marine multi-robot systems using acoustic communication," *IEEE Robot. Autom. Lett.*, vol. 5, pp. 5299–5306, 2020.

[34] J. McMahon, "Autonomous data collection with timed communication constraints for unmanned underwater vehicles," *IEEE Robot. Autom. Lett.*, vol. 6, pp. 1832–1839, 2021.

[35] B. Braginsky, A. Baruch, and H. Guterman, "Development of an autonomous surface vehicle capable of tracking autonomous underwater vehicles," *Ocean Engineering*, vol. 197, 106868, 2020.

[36] B. Arbanas et al., "Consensus protocol for underwater multi-robot system using two communication channels," in *Proc. MED*, Zadar, Croatia, 2018, pp. 358–363.

[37] M. Bresciani et al., "Cooperative ASV/AUV system exploiting active acoustic localization," in *Proc. IEEE/RSJ IROS*, Prague, Czech Republic, 2021, pp. 4337–4342.

[38] T. Pham Van and C. D. N. Dang, "Underwater searching based on AUV-ASV cooperation," in *Proc. Int. Conf. Ubiquitous Inf. Manag. Commun.*, Seoul, Korea, 2022.

[39] M. V. Jakuba et al., "Toward an autonomous communications relay for deep-water scientific AUV operations," in *Proc. IEEE/OES AUV Workshop*, Porto, Portugal, 2018.

[40] Georgia Tech Savannah Robotics, "ASV competition," in *Proc. MED*, Ancona, Italy, 2003.

[41] XiroiStack, "XiroiStack GitHub repository," 2021. [Online]. Available: https://github.com/srv/xiroi_stack

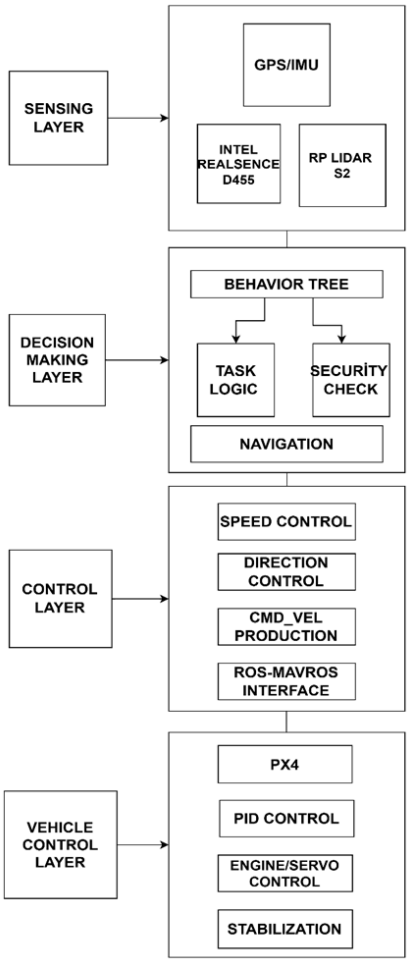
[42] L. Ribés Nebot, "IQUAview: una interfaz de usuario para operar vehículos submarinos basada en QGIS," 2019. [Online].

[43] IQUAview, "IQUAview Wiki," 2021. [Online]. Available: https://bitbucket.org/iquarobotics/iquaview/

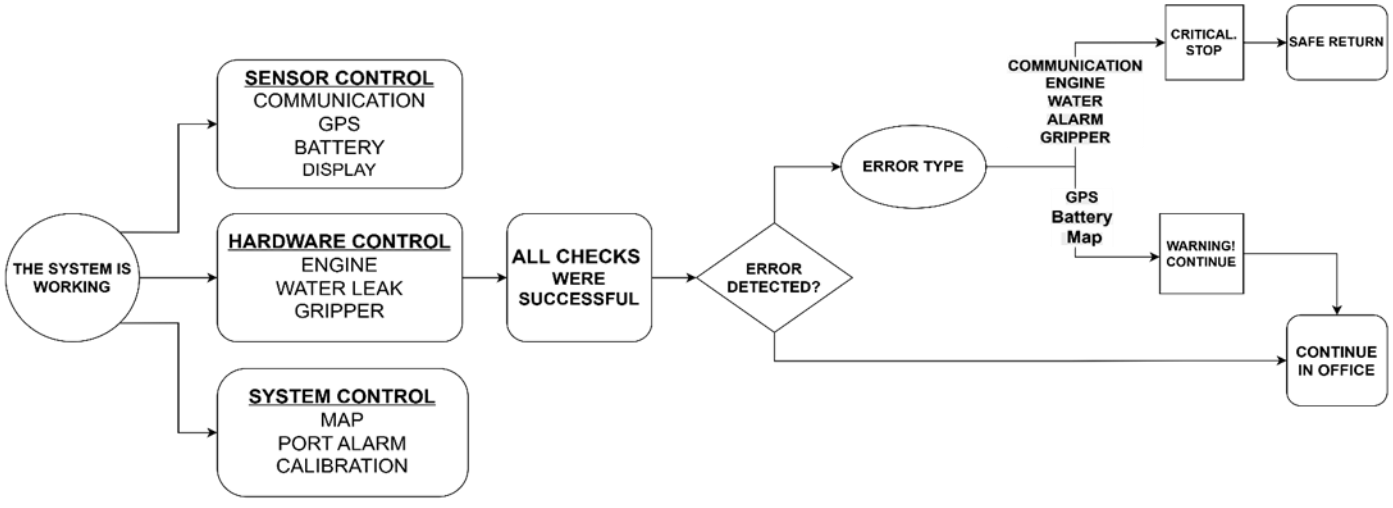
[44] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Proc. IAS-13*, 2014, pp. 335–348.

[45] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Rehab Robotics*, 2011.

XI. APPENDIX A- SYSTEM ARCHITECTURE AND TASK DIAGRAMS



Addition Figure 6- ASV Status Monitoring and Safe Return Mechanism



Addition Figure 7- System architecture

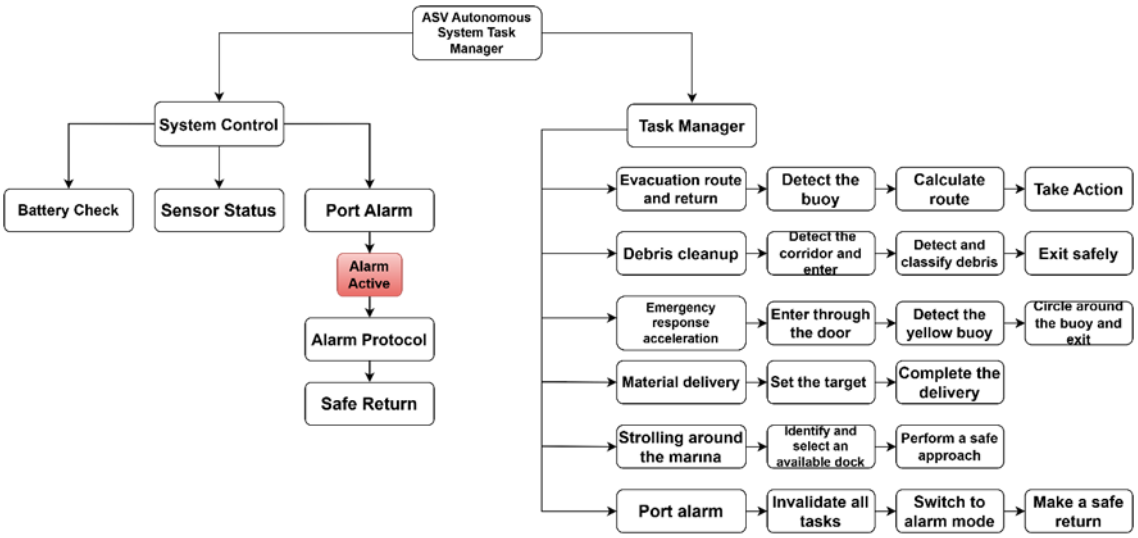


Figure 8- ASV Autonomous System Task Manager

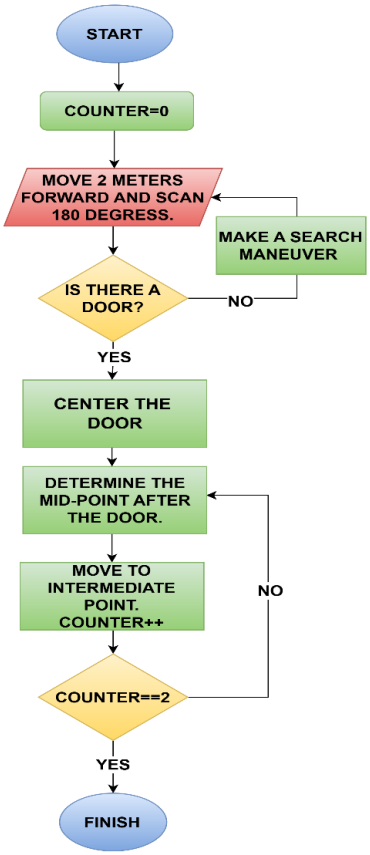


Figure 9- Task 1

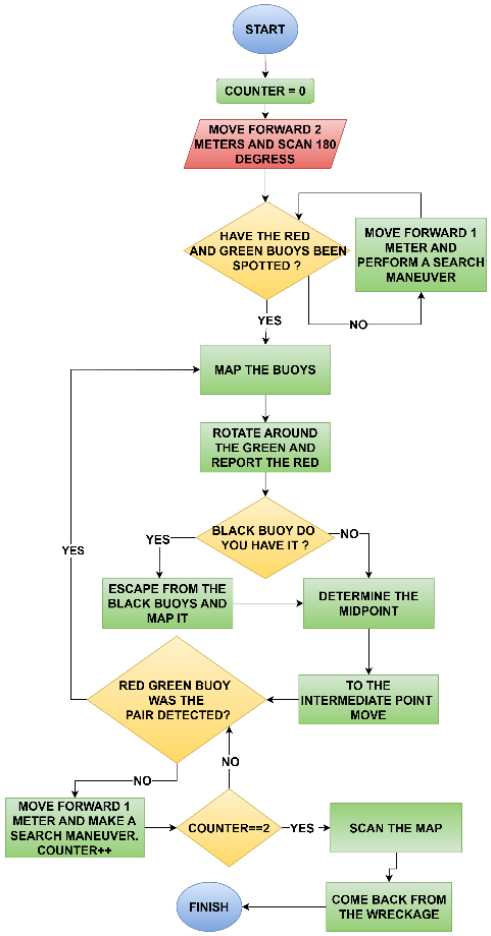


Figure 10- Task 2

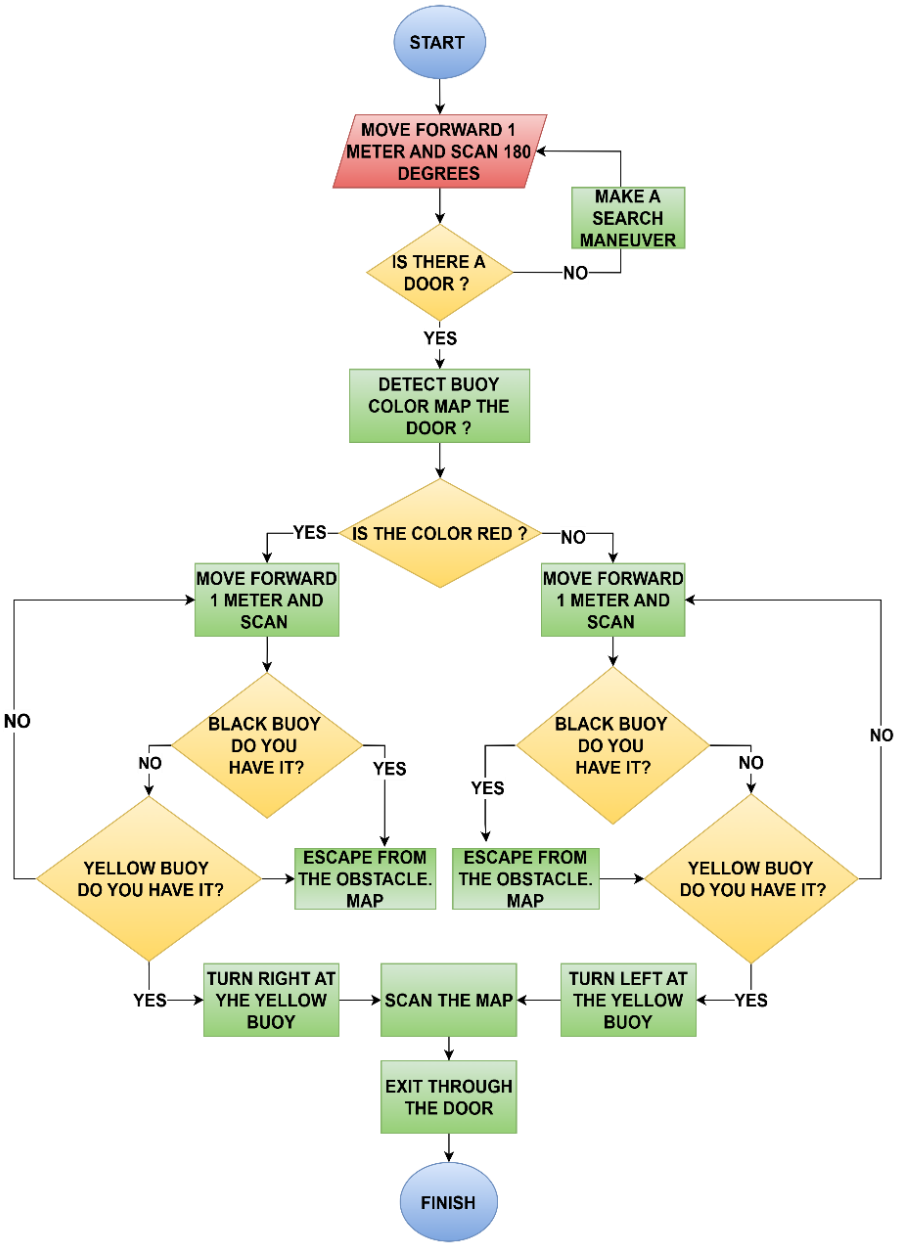


Figure 11- Task 3

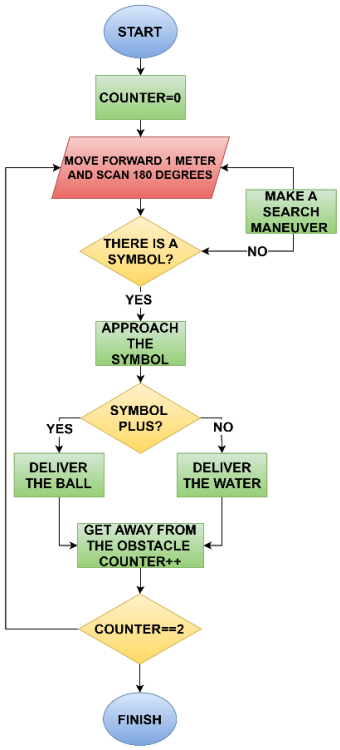


Figure 12- Task 4

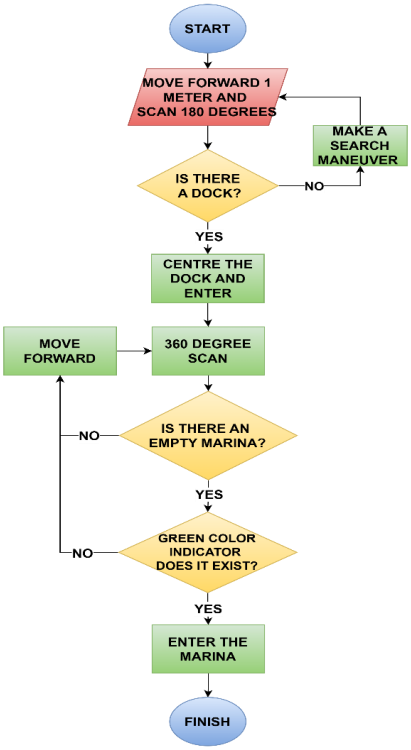


Figure 13- Task 5

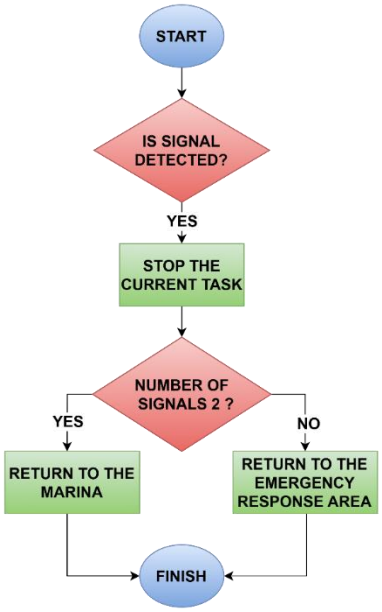


Figure 14- Task 6

XII. APPENDIX B- TEST PLAN & RESULTS

Tablo 2- Lidar Datas

Actual measurement(m)	Measurement of Lidar in Rviz (m)	Error margin(m)
1,524m	1,5281m	+0,0059m
1,82m	1,825m	+0,005m
3,048m	3,056m	+0,008m
3,66m	3,67m	+0,01m
7,62m	7,6325m	+0,0125m
8,636m	8,64m	0,004m
12,192m	12,1591m	-0,0329m
29,48m	29,415m	+0,065m

The LİDAR data we tested here has been tested both in the real environment and in Rviz. We calculated the margin of error by measuring distances similar to those on the course using lidar. We observed a maximum error margin of 0.27%. We decided that this margin of error was tolerable and tested that it could provide us with sufficiently accurate data for a Roboboat 2026.

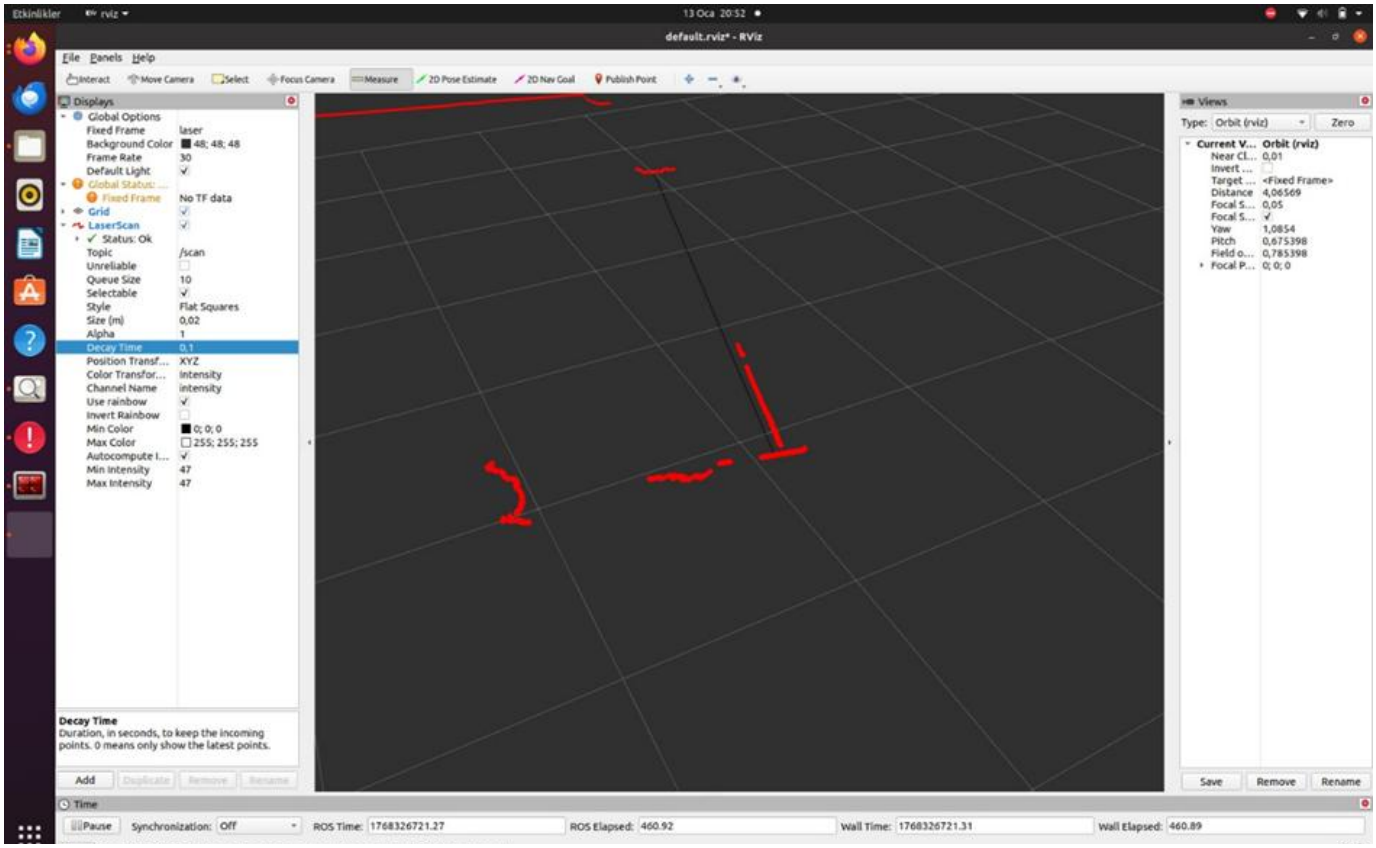


Figure 15- Lidar Detection

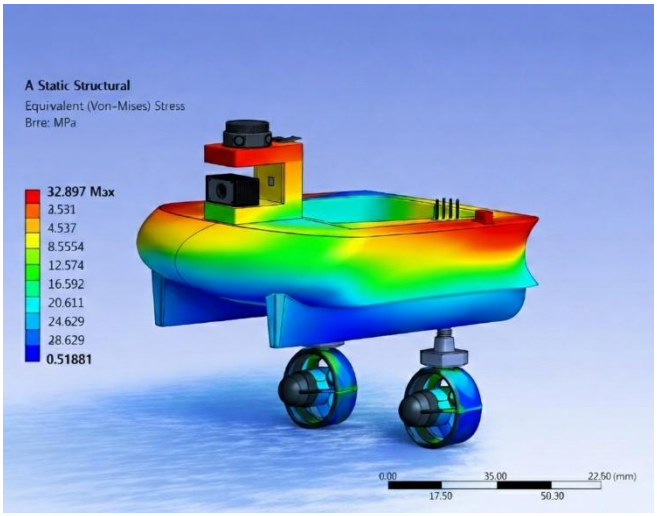


Figure 16- static analysis

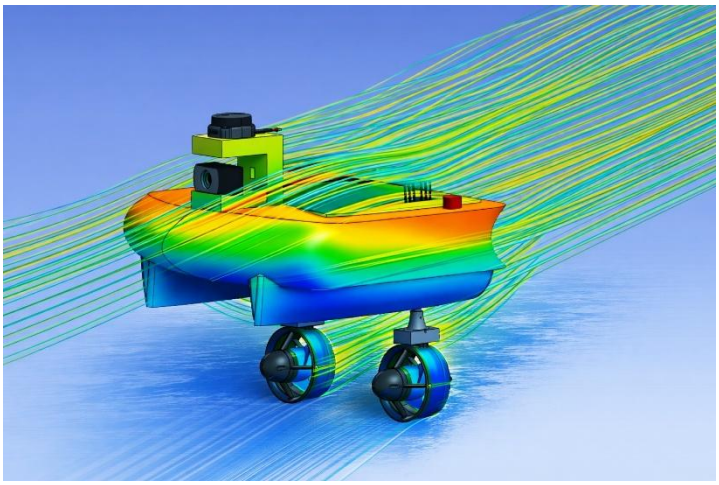


Figure 17- CFD analysis

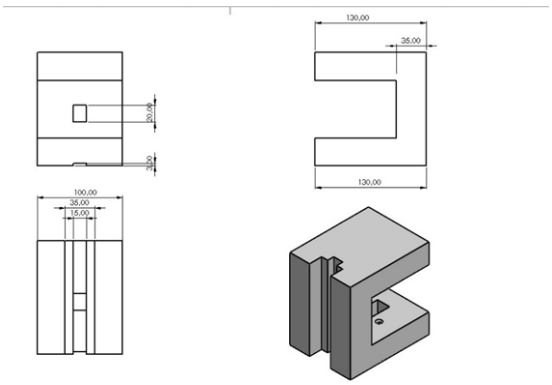


Figure 18- camera and lidar stand

In this section, the operational efficiency and structural integrity of the vehicle are evaluated through Computer-Aided Engineering (CAE) methods. Specifically, Static Structural Analysis and Computational Fluid Dynamics (CFD) simulations were conducted to validate the design.

To ensure the vehicle can withstand operational loads, a Von-Mises stress analysis was performed.

- **Maximum Stress Concentration:** The simulation results indicate a peak stress of **32.897 MPa**, primarily located at the upper sensor/camera mounting bracket. This region, highlighted in red, represents the primary load path where structural reinforcement is most critical.
- **Material Compatibility:** Given that the hull is manufactured using **Fiberglass (Glass Fiber Reinforced Polymer)** over a wooden mold, this stress value is well within the safe operating limits. Standard fiberglass laminates typically exhibit tensile strengths ranging from 100 to 250 MPa. Therefore, the design maintains a high **Factor of Safety (FoS)**.
- **Structural Stability:** The main hull and thruster attachment points show significantly lower stress levels (blue zones, ~0.5 - 4.5 MPa), confirming that the geometry effectively distributes mechanical loads without risking plastic deformation or fatigue failure.

A CFD analysis was executed to visualize the flow field and evaluate the drag characteristics of the vehicle during transit.

- **Streamline Behavior:** The streamlines illustrate a predominantly laminar flow around the bow of the vehicle. The curved geometry of the hull successfully directs fluid with minimal separation, which is crucial for reducing energy consumption.
- **Thruster Interaction:** The flow vectors remain stable as they approach the ducted thrusters. The positioning of the propulsion units ensures that they operate in a "clean" wake, maximizing thrust efficiency and maneuverability.
- **Surface Finish Impact:** Since the vehicle will be produced via a wooden mold-based fiberglass process, achieving a high-quality surface finish (Gelcoat application) is essential. A smooth surface will minimize skin friction, ensuring that the real-world performance aligns with these simulated hydrodynamic results.

As a result, the choice of fiberglass as the primary material offers an ideal balance between strength-to-weight ratio and corrosion resistance. The analysis confirms that the radii and fillets incorporated into the design are optimized for composite layup, preventing stress risers at sharp corners. During the production phase, additional fiber layers (reinforcement) will be applied to the high-stress zones identified in the FEA results to ensure long-term durability.

XIII. APPENDIX C- BALL LAUNCHER MECHANISM

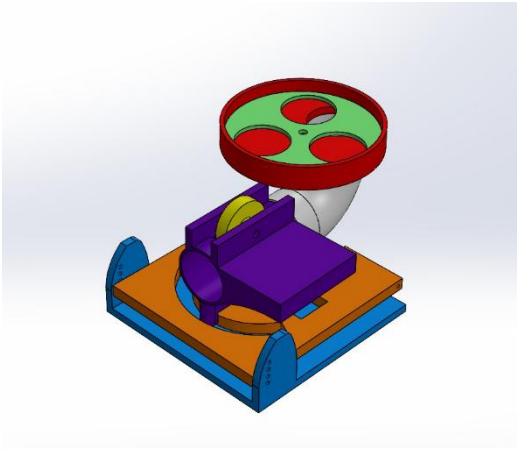


Figure 19- ball launcher mechanism

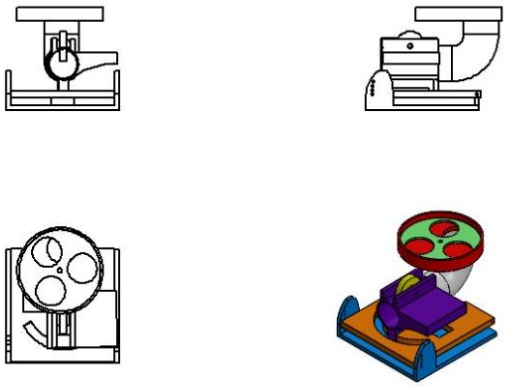


Figure 20- technical drawing of the ball launching

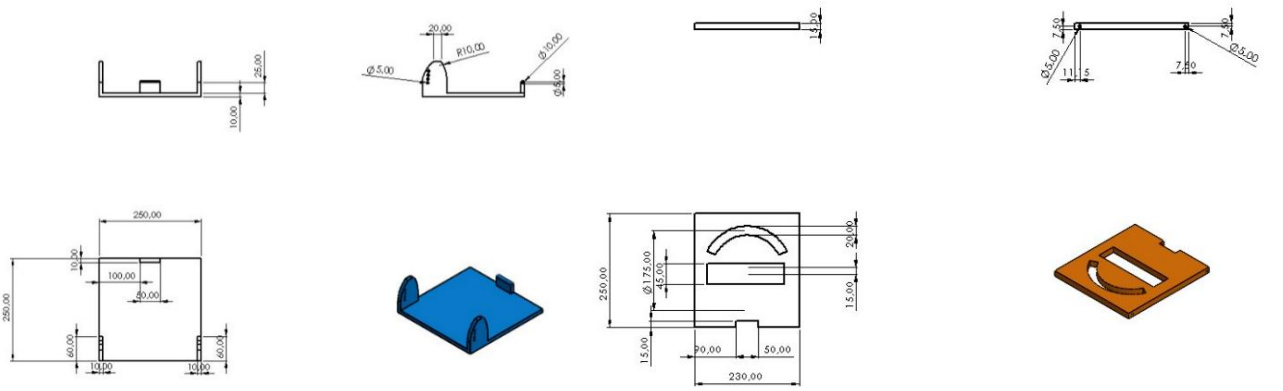


Figure 21- Part 1 of the ball-throwing mechanism

Figure 22- Part 2 of the ball-throwing mechanism

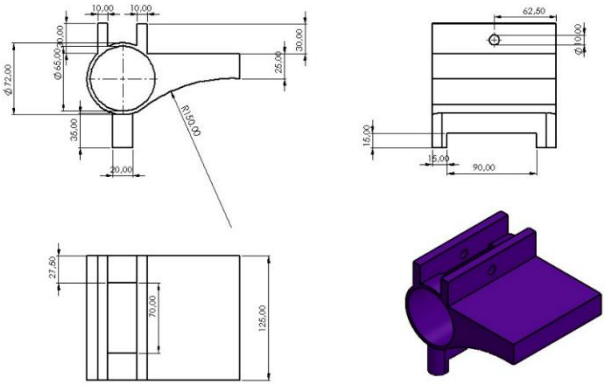


Figure 23- Part 3 of the ball-throwing mechanism

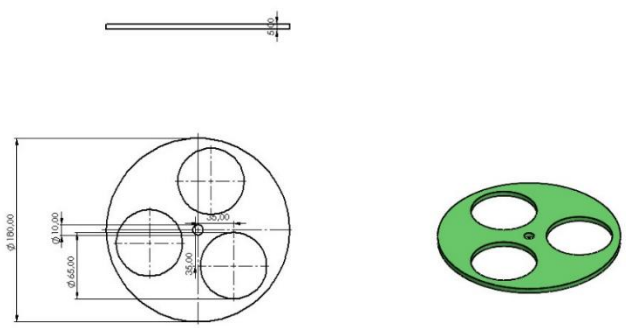


Figure 24- Part 4 of the ball-throwing mechanism

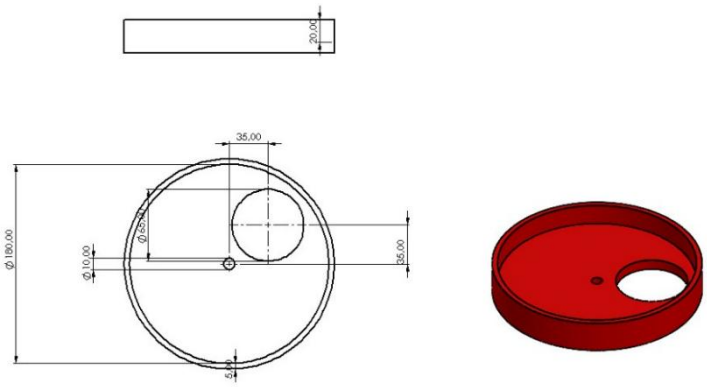


Figure 25- Part 5 of the ball-throwing mechanism

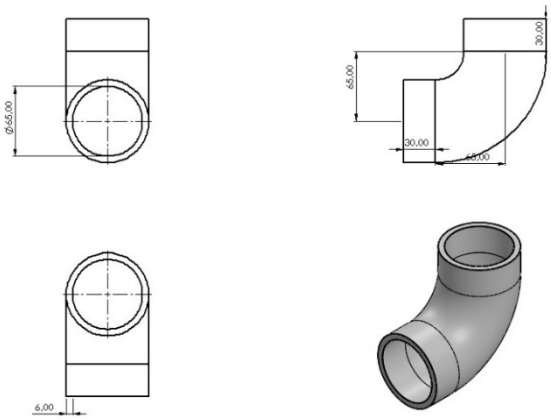


Figure 26- Part 6 of the ball-throwing mechanism

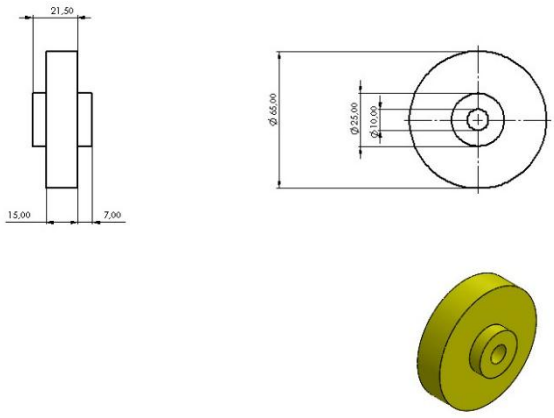


Figure 27- Part 7 of the ball-throwing mechanism