Roboboat 2025: Technical Design Report AGH Solar Boat Team

Anastasiia Vudvud, Mikolaj Slowikowski, Krzysztof Furtak, Przemyslaw Domagala, Szymon Nowak, Damian Brzana, Filip Nawalaniec, Kamil Zajac, Michal Dydek
AGH University of Krakow, Krakow, Poland

***Abstract*—Autonomous boat Barka enters the RoboBoat competition for the 3rd time and richer in past experiences applies new solutions and technical advancements. Strategy of the team emphasizes using sensor fusion of IMU, LiDAR, and ZED2i camera to achieve precise real-time navigation, object detection, and decision-making across the tasks. Design adaptations include hardware enhancements such as LiDAR integration, a ball launcher for object delivery, and improved waterproofing and cooling systems, ensuring reliability in diverse environments.**

The three-layer software architecture balances scalability and real-time performance, employing advanced processing techniques, including YOLOv10 for vision-based detection and Euclidean clustering for LiDAR data. Detection fusion combines vision and LiDAR inputs for robust global object tracking. Hardware improvements such as enhanced battery management, stabilized LiDAR mounts, and upgraded communication systems complement these software capabilities.

Testing strategies ensure operational readiness despite seasonal constraints, relying on iterative dry and on-water tests. This alignment between strategy and engineering underpins the ASV's ability to perform reliably, efficiently, and safely in competitive scenarios.

Competition Strategy

Building on our recent success with vision-based steering, we aimed to enhance the system by incorporating LiDAR. Our system is designed to integrate data from all onboard sensors, including the IMU, LiDAR, and ZED2i camera, to create a unified and highly accurate understanding of the environment. This sensor fusion enables the boat to navigate, avoid obstacles, and interact with its surroundings with exceptional precision.

Across all tasks we are using real-time detection, localization and decision-making based on a constantly updated model of the environment. Advanced algorithms process this data to identify objects and determine the best course of action to achieve the task objectives while avoiding collisions and maintaining efficiency. To meet the unique requirements of each task, we have implemented necessary hardware modifications that ensure compatibility with all scenarios.

I. Navigation Channel & Follow the Path

For the Navigation Channel, our approach involves a strategy to ensure precise navigation through the designated path. We start by calculating the localization of all nearby gate buoys using the boat's integrated sensors, including the IMU, LiDAR and ZED2i camera, to accurately determine their positions relative to the boat. This data is processed to establish the arrangement of the buoys, allowing us to identify the boundaries of the gate with high precision. By maintaining a central path, we minimize the risk of collisions and ensure compliance with course guidelines. Our system continuously updates the localization of the gate buoys in real-time to account for environmental factors, such as water currents and wind, which could cause drift. Approach to Follow the Path utilizes algorithms similar to those described above, but also accounts for the presence of obstacles such as yellow buoys and other vessels. Obstacle avoidance algorithms determine the optimal path to prevent collisions with boats or yellow buoys, while the tracking algorithm enables the counting of the objects.

II. Speed Challenge

The key to success in the speed challenge is identifying the distinct blue buoy as quickly as possible while avoiding the black buoys. Our strategy incorporates the newly implemented obstacle avoidance algorithm, detailed below, to navigate between the black buoys and determine the optimal path to bypass them. Waypoints are utilized to steer around the blue buoy and return by placing a waypoint at the center of the gate.

III. Docking

Vision detection powered by a neural network will identify the front and side of the dock and this will allow the vessel to align parallel to the dock's side, change direction, and steer itself to the center of the dock's front. At this point, the system will be able to identify all three banners and select the designated shape. The final step involves checking the availability of the dock to ensure it is not occupied by another boat. If it is occupied, the vessel will return using pre-established waypoints and navigate to the opposite side. With the dock's position stored, the ASV will initiate the procedure on the other side.

IV. Water & Object Delivery

Our approach to Water & Object Delivery tasks focuses on completing objectives immediately upon detecting an orange or black vessel. By leveraging advanced detection systems and efficient decision-making algorithms, the ASV can identify target vessels and initiate delivery actions without delay or unnecessary navigation at the end of a run. Additionally, tracking and localizing objects within a coordinate system allows us to determine whether the currently detected boat has already been served, ensuring that the ASV can recognize if it has gone off course.

V. Return to Home

Our plan involves introducing a "home" waypoint at the beginning of the Navigation Channel. This is implemented with the use of cartesian coordinates with our (0,0) point designated as the starting position. Along the route to the waypoint, the boat will also search for two black buoys; upon locating them, it will navigate through the gate between them.

Design Strategy

Given the strong performance of the hardware last year, we decided to implement only minor improvements. There were no significant changes, but we added a ball shooter to enable the completion of the Water and Object Delivery task. Most of our time was dedicated to enhancing the software system. We introduced a LiDAR sensor and began developing new software to integrate it into our system and refine our algorithms. Additionally, we are implementing a new approach that combines object localization using vision, point cloud data, and GPS.

I. Hull Design

The design featured a catamaran structure for easy disassembly, ensuring convenient transport, including suitability for air travel. The hull design remains largely unchanged.

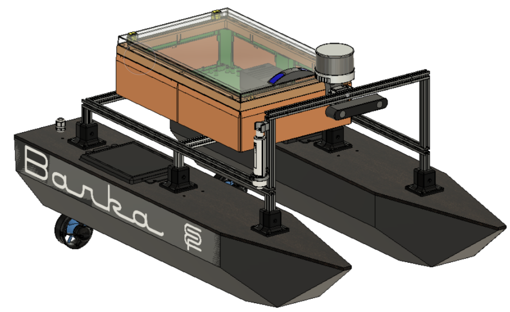


Fig. 1: CAD model of Barka

A. *Waterproofing*

We applied a new layer of epoxy varnish after removing the old enamel. To address issues with the previous sealing system, which caused condensation, PCB corrosion, and battery explosion risks, we replaced the 3D-printed covers with ones made with basalt fiber, consistent with the material used for the rest of the boat. Its transparency to magnetic fields prevents interference with electronics. The new covers incorporate a labyrinth sealing system with soft sponge, making the design waterproof and effectively preventing condensation inside the hull.

B. *Cooling system*

During the last competition, we faced overheating issues with the IMU/GPS and Jetson due to sun exposure and rising temperatures in the electrical box. To address this, we designed a cooling system consisting of two fans: one configured for intake and the other for exhaust. Both fans are housed in durable water traps 3D-printed from UV-resistant resin. Additionally, we covered the top of the electrical box with the window tinting foil to reduce the impact of solar radiation. This solution minimized PCB corrosion from UV light and lowered the internal temperature.

II. Hardware Design

A. *Construction*

The water nozzle system remains the same as last year, but a new ball launcher needed to be developed. A key focus was ensuring synchronization between the camera and LiDAR data, which led us to develop a stabilization system.

*1) **Stabilization**:* We developed a gimbal-like stabilizer with one degree of freedom along the pitch axis. The stabilizer is constructed using a combination of 3D-printed components, stainless steel for stiffness, and a servomechanism for actuation. Designing the system required careful consideration of the center of mass, which needed to remain below the pivot point for stability. This proved challenging, as the LiDAR, being the heaviest component, is mounted at the highest point of the structure. Servomechanisms were selected for this application because of their affordability and simplicity. The control protocol is the standard modeler's pulse-width modulation (PWM). Requiring only ground, power, and signal connections, this minimal wiring simplifies integration with the system's electronics.

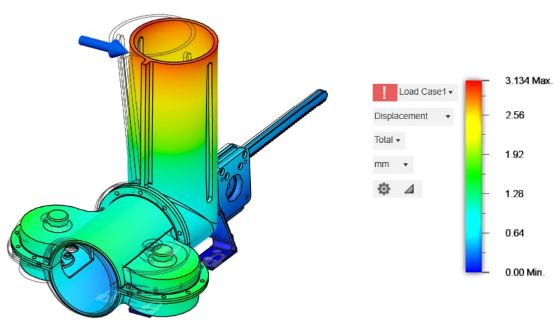


Fig. 2: Maximal displacement in previous load case in ball laucher

*2) **Ball launcher**:* The ball launcher utilizes two brushless motors and disks, all constructed using resin printing techniques. The launcher operates by first spinning up the disks to a high speed and then feeding a ball between them, propelling it forward with great force. A servomechanism is used to feed the ball into the launcher, and like the motor controllers, it is easily controllable using standard PWM signals. This simplifies integration with the control board, which manages multiple PWM signals for motor speed via ESCs and the ball-feeding servo.

B. *Electrical*

As part of the improvements to the boat's electrical system, significant upgrades were implemented to enhance its reliability and safety. The key enhancement was the optimization of the energy management system, which now efficiently monitors battery parameters such as current, voltage and temperature, protecting it from operating under adverse conditions. The system was also equipped with a buzzer to alert users of exceeded safe limits and a signal lamp indicating the boat's current operational state. Additionally, the wiring was reorganized using a dedicated PCB that serves as a power distribution board, providing additional safety by integrating fuses to protect electronic components.

*1) **Battery Management System**:* The advanced energy management system was upgraded to reliably monitor key battery parameters: current, voltage, and temperature. The system detects hazardous operating conditions and enables quick responses, ensuring the safe operation of the boat. Compared to last year, an additional electronic circuit consisting of transistors was added, which disconnects power from the thrusters in case of a critical threat [7].

*2) **Buzzer**:* The added buzzer serves as a warning signal. It emits sound when safe thresholds for parameters such as excessively high or low voltage, overcurrent, or abnormal temperature are exceeded, allowing immediate action to protect the system.

*3) **Power Supply Modernization**:* Instead of the previously used main power cable with multiple branches, a dedicated PCB was designed. The new construction serves as a power distribution board and includes fuses that effectively protect individual electronic components from overcurrent.

*4) **Status Light Post**:* The status light post provides an easy visual indicator of the boat's operational mode. It features a 360-degree LED light column with three distinct colors, each representing a specific state: manual control, autonomous operation, or disarmed mode. This simple yet effective system ensures that the current status of the boat is immediately clear, improving safety and operational awareness.

C. *Communication*

Over the summer, we encountered issues with Wi-Fi communication, where the connection would occasionally drop for up to a minute. To address this, we upgraded our connectivity by switching to a dongle, replacing the Ubiquiti BULLET antennas, which improved data transfer and enabled reliable remote connection. We decided to take greater advantage of the radio's capabilities by adding additional data to its screen. Furthermore, we developed a web-based GUI to display telemetry information.

*1) **Dongle**:* We decided to revise our approach for connecting with the boat from shore, choosing an RM520N-GL 5G dongle for our Jetson. Its connectivity enables fluent data transfer, allowing us to reliably monitor and control the boat remotely. This allowed us to utilize the telemetry system through a web-based Graphical User Interface.

*2) **Telemetry**:* To monitor battery voltages directly on the radio controller screen, we utilized the TBS Crossfire (CRSF) protocol by reverse-engineering publicly available packets and customizing them to our needs. This method enables reliable, real-time transmission of battery voltage data as long as the radio-receiver connection is active.

*3) **Graphical User Interface**:* We developed a web-based GUI using ROS2, React and Django to provide a centralized, real-time display of boat telemetry. This interface subscribes to multiple topics and aggregates the data into a back-end database for logging and analysis. Operators can monitor position, velocity, heading, and system status through a responsive dashboard, accessible through standard web browsers. This approach is more practical and convenient compared to using an FPV camera.

III. Software Design

To enable scalability and simplify the incorporation of new functionalities, our project is organized into a three-layer system. This approach offers a clear, modular framework that is both easily scalable and adaptable. We decided to implement our software in both C++ and Python using ROS2 [2] framework, as Python offers support for a broader range of libraries.

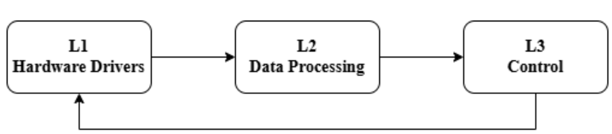


Fig. 3: Software system layers and direction of the data flow

A. *Layer 1 - Hardware Drivers*

This layer serves as the interface to hardware communication, encompassing messaging, CAN bus interactions, camera and LiDAR communication and motor control calculations. We are using zed_wrapper, hesai_ros_wrapper and our customly made can_driver to receive data from LiDAR, ZED2i camera and IMU/GPS. We changed SBG Ellipse-D IMU/GPS to Xsens MTi-680G, which includes RTK support that we plan to utilize in the future. To reduce sun glare, we have purchased a CPL filter, which we plan to install in front of the camera.

B. *Layer 2 - Data Processing*

This layer handles real-time object detection using both vision and point cloud data, processes information from the IMU/GPS sensor and manages trajectory planning. To ensure efficient object detection, a neural network based on the You Only Look Once (YOLO) algorithm was selected. YOLOv10 [3], [4], which is faster and more accurate than the YOLOv8 model used last year, performs object detection in a single iteration, making it highly suitable for real-time applications.

*1) **Autolabelling & Augmenation**:* To reduce the time required for image labeling, we implemented a partial auto-labeling process. This approach involves labeling a subset of the dataset, training the model on this subset, and then using the trained model to label the remaining data. Subsequently, we only need to review the generated labels, significantly decreasing the time needed for model preparation. We apply a variety of augmentations to the training set to prepare the model for weather conditions that differ from those during data collection, such as intense sunlight.

*2) **Quantitized Convolutional Neural Network**:* To achieve faster inference times and reduced memory exploitation, we implemented quantization to reduce the precision of weights and activation functions in our YOLOv11 model from 32-bit to 16-bit, ensuring minimal accuracy loss. This optimization uses frameworks like CUDA and TensorRT for efficient parallel computing. After receiving the initial vision data from the ZED2i camera, a 10-step warmup initializes GPU resources for stable processing. The workflow involves preprocessing the image by resizing and normalizing pixel values,

Fig. 4: Comparison of vision detections with and without augmentation in the strong sun

running inference on the GPU, and transferring results to the CPU. Finally, postprocessing interprets the output into a ROS2-compatible format for seamless integration into our autonomous system.

*3) **Clusterization & LiDAR**:* The Euclidean clustering algorithm was implemented to identify and segment objects within the processed point cloud data [9]. This method groups points according to their spatial proximity, using the Euclidean distance metric to determine cluster membership. The implementation was carried out using the Point Cloud Library (PCL) [10] integrated within the ROS 2 framework, facilitating seamless communication and data handling between different components of the system. This is crucial to our approach, as we rely on depth data to calculate the coordinates of objects.

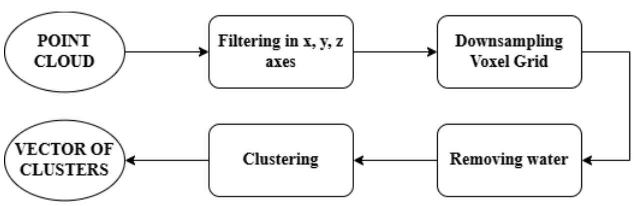


Fig. 5: Stages of the point cloud data processing pipeline

*4) **Detection Fusion & Tracking**:* The detection fusion system integrates data from two independent detectors, ensuring alignment of incoming detections based on similar timestamps through topic synchronization. It prevents issues caused by data arriving from different time periods. When a message is received, the system compares detections from the vision detector and the cluster-based detector by analyzing their angular positions. It evaluates the likelihood that both detections correspond to the same object and returns a boolean value to indicate the result.

Integration of data from LiDAR, IMU, GPS, and ZED2i enables the calculation of global object coordinates, with the origin point (0,0) defined as the autonomous vessel's starting location. The global coordinates of each new detection that successfully undergoes the fusion process are compared with the locations of previously detected objects. Based on a distance threshold, the new detection is assigned either a new ID or an existing one. This helps us to see the difference between different buoys and other elements.

C. *Layer 3 - Control*

Within this layer, an important component is the implementation of a state machine that governs the selection of algorithms executed by the software. Taking information from the L2 layer and guided by various conditions, the state machine serves as the decision-making engine, determining the subsequent algorithm to be executed.

*1) **Obstacle Avoidance**:* To ensure higher steering accuracy, we calculate the positions of objects around the vessel using object localization techniques. By continuously monitoring the surrounding environment, we identify and map obstacles and other vessels in real-time. These localized objects are then factored into the selection of optimal waypoints, enabling precise and efficient navigation while maintaining a safe distance from potential hazards [6]. This dynamic approach enhances the vessel's ability to follow a desired trajectory with accuracy and reliability.

Testing Strategy

As the majority of testing occurs in late autumn or winter, lake tests can only be conducted infrequently. This year, we were unable to conduct tests in the swimming pool due to a change in our approach to autonomous steering, which now is based on calculating waypoints from vision and point cloud data, as GPS signals are not very accurate indoors. Therefore, we strive to maximize the number of things we test on the water when we are at the lake and make every effort to accomplish as much as possible. We also try to conduct testing even in the absence of water access whenever possible.

I. Dry tests

To validate our hardware, we perform dry tests within our workshop. Before activating the boat, we thoroughly inspect all physical components of the system. Once the inspection is complete, we power up the boat and verify the radio connection. Next, we ensure that the thrusters, water nozzle, and ball shooter correctly respond to commands from the radio. Following these steps, we activate Jetson, the main component of the autonomous system, and confirm that signals can be sent to and received from it. Such tests are performed the evening before every test on water.

If necessary, we also test individual components of the boat. To assess the nozzle, we link it to the boat and manually issue commands to initiate and cease water flow. The same procedure is applied to the ball shooter. Both functionalities feature simple control, having only two states. These tests do not require powering the thrusters and can be performed at any time as needed in our workshop.

II. Tests on water

This year our only option for testing was Bagry lake in Krakow, where we have the capability to assemble parts of Follow the Path, Speed Challenge, Object and Water Delivery and Return to Home tasks. For the initial two tasks, we arranged buoys to simulate gate markers, comprising red and green buoys, as well as establishing a designated course for the Speed Challenge. In the case of Object and Water Delivery, two constructions which hold the plus and triangle sign, have been built with PVC pipes to replicate the testing environment. This approach ensures that our testing scenarios closely mirror the competition conditions, allowing us to refine and optimize Barka's performance in a controlled setting before deployment.

During the summer, as part of our preparation for the Njord competition, we collected a large dataset of our buoys on Bagry Lake, giving us a robust model for testing. This year, we have also prioritized recording sensor data to ensure that we can test our algorithms using real-world scenario data rather than relying solely on idealized simulation data. While on the water, we are able to thoroughly test hardware, software, peripheral elements, and their integration. The only aspect we cannot test is how our software transitions between tasks.

III. Simulation tests

We use simulation exclusively for testing software algorithms, as it removes variables like waves and wind, making comprehensive testing otherwise impossible. Virtual environment allows us to refine the logic in algorithms and make sure the detection processes work well. Consequently, we can check how strong and reliable our algorithms are in various situations. Such flexibility speeds up our development process. In the Gazebo, we have slightly updated the worlds we created for RoboBoat 2024 by incorporating new custom and VRX [5], [8] models.

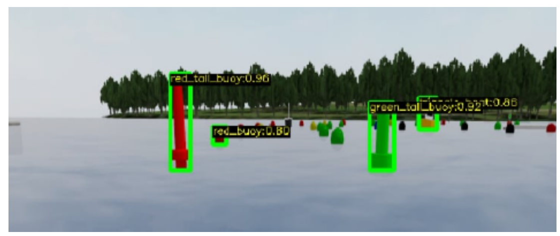


Fig. 6: Performance of the neural network model in real-time buoys detection during the simulation testing

Acknowledgments

We extend our heartfelt gratitude to AGH University for their invaluable support and resources throughout the development of our project. We are especially thankful to our academic advisors, Krzysztof Sornek and Wojciech Sajdak, for their expertise, guidance, and dedication.

Our deepest thanks also go to TME and Xsens for their crucial support in design and funding. We thank Michal Korzeniowski and Natalia Berestecka (MINE and KSAF) for filming our promotional video and providing media materials. We also appreciate the HORN sailing club for enabling in-water boat tests and the VRX developers for their dynamic simulation environment.

A special gratitude goes to our team members Karolina Nawrot, Krzysztof Furtak, and Mikolaj Slowikowski for preparing the application for the Ministry of Science and Higher Education scholarship, without which participation in this competition would have been much more challenging.

Finally, we extend our deepest gratitude to every member of our team for their dedication, time, and passion that made this project a reality.

REFERENCES

[1] Robonation Inc., 2025 Team Handbook, 2024, robonation.gitbook.io/roboboat-resources.
[2] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.
[3] Ultralytics, ”YOLOv10 - Ultralytics YOLO Docs,” [Online]. Available: https://docs.ultralytics.com/models/yolov10/.
[4] A. Wang et al., ”YOLOv10: Real-Time End-to-End Object Detection,” arXiv preprint arXiv:2405.14458, 2024
[5] B. Bingham et al., ”Toward Maritime Robotic Simulation in Gazebo,” OCEANS 2019 MTS/IEEE SEATTLE, Seattle, WA, USA, 2019, pp. 1-10, doi: 10.23919/OCEANS40490.2019.8962724.
[6] D. Bruijnen, J. van Helvoort and R. van de Molengraft, ”Realtime motion path generation using subtargets in a changing environment,” 2006 American Control Conference, Minneapolis, MN, USA, 2006, pp. 6 pp.-, doi: 10.1109/ACC.2006.1657385.
[7] Davide Andrea, Battery Management Systems for Large Lithium-Ion Battery Packs , Artech, 2010.
[8] N. Koenig and A. Howard, ”Design and use paradigms for Gazebo, an open-source multi-robot simulator,” 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 2004, pp. 2149-2154 vol.3, doi: 10.1109/IROS.2004.1389727
[9] K. Miadlicki, M. Pajor and M. Saków, ”Ground plane estimation from sparse LiDAR data for loader crane sensor fusion system,” 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 2017, pp. 717-722, doi: 10.1109/MMAR.2017.8046916.
[10] R. B. Rusu and S. Cousins, ”3D is here: Point Cloud Library (PCL),” 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 1-4, doi: 10.1109/ICRA.2011.5980567.

APPENDIX A
TEST PLAN RESULTS

A. Scope

We conduct monthly tests of our boat, starting with basic functionalities and gradually progressing to more advanced ones. Basic tests include verifying that the electrical system functions as expected and ensuring the boat can navigate using GPS waypoints. As the competition approaches, we focus on testing more advanced functionalities, such as the ball shooter and buoy tracking.

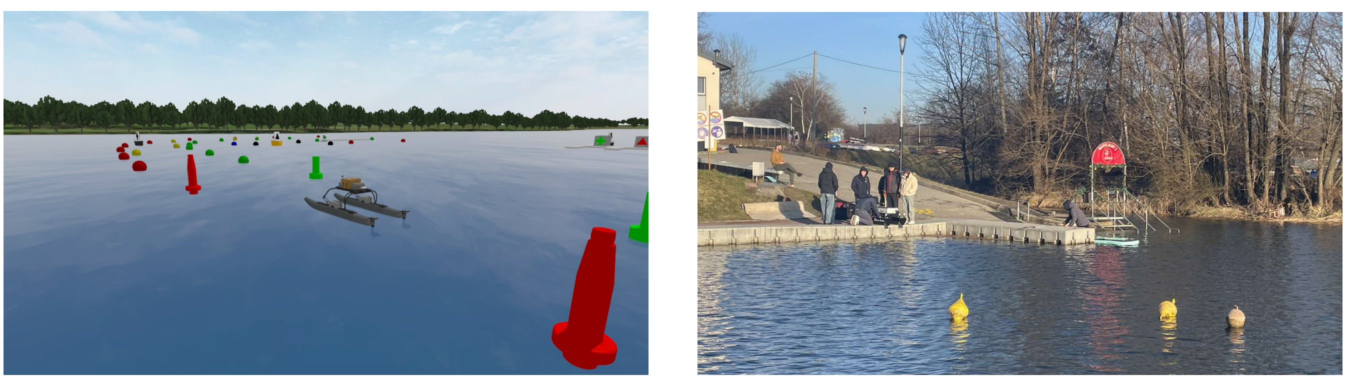
B. Schedule

	October	November	December	January	February	March (Sarasota)
Hardware	- Planning	- Telemetry on radio - Power supply for lidar	- Embedded code refactor	- Stabilization for lidar and camera	- Ball shooter - Buzzer - Power supply modernization	- CPL filter for reflections - Cooling system - Electrical box cover
Software	- Planning - Lidar integration - Clusterization algorithm	- Detection fusion - Record data with lidar and vision - Waypoint navigation	- Working on algorithms	- Tracking - Waypoint navigation - Better data gathering	- Collision avoidance - Telemetry with GUI - Follow the path	- Waypoints - Neural network (sun reflections)

Fig. 7: Plan of the workload and water tests

On the table, there is a schedule outlining work and tests. The months highlighted in blue indicate when water tests are planned. We have organized our tasks to ensure there is always something ready for testing, both in terms of hardware and software.

C. Environment



(a) VRX Simulation environment for the whole course

(b) Sailing club HORN in Krakow

Fig. 8: Testing environments

Our simulation is built on a world originally created for the VRX competition. Using models from VRX or custom-made ones, we recreated each task and the entire RoboBoat 2025 course. While this is an idealized environment, it is perfect for developing algorithms that require continuous testing.

For in-water testing, we use Bagry Lake in Kraków, specifically the harbor of the HORN Sailing Club. It provides a safe environment, often used by beginner sailors, and allows us to position buoys as needed.

D. Risk Management

Potential risks on the water include someone slipping on a slippery entrance to the water. To place buoys, we use paddle board or kayak and there is a risk that someone can fall into the water. We use this lake regularly for the tests of both our boats – autonomous boat and solar powered racing boat, so members of our team have been through the safety training. We also make sure that the people on the kayak or paddle board wear lifejackets.

E. Results

Throughout our monthly sessions, we gathered the necessary data for testing algorithms, clustering, and the neural network. During the tests conducted in November, we confirmed that the hardware functioned as expected, with no issues detected. We also verified that the fusion of detections was working correctly and recorded some data. However, we encountered problems with waypoint navigation, which we scheduled for retesting in January. We were able to successfully resolve the navigation issues.

In February, we plan to test the final version of the hardware that will be used during the competition. During our stay in Sarasota, we intend to test our cooling system and assess the neural network's performance in handling sun reflection issues. Unfortunately, due to weather conditions in Krakow, we are unable to conduct these tests locally.

APPENDIX B
NEURAL NETWORK QUANTITAZATION

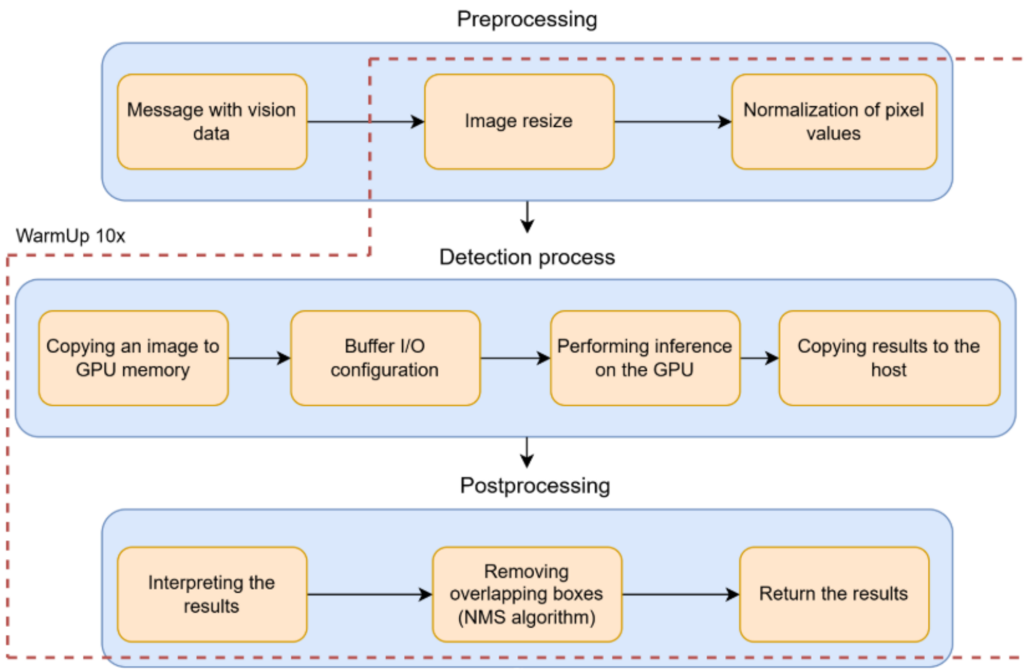


Fig. 9: Main pipeline for accelerating vision detection algorithm

Efficient and optimized implementation of detection based on CNN models using embedded devices like Jetson requires a quantization method, which goal is to minimize the accuracy loss and inference time for real-time processing applications. Quantization is the process of reducing the precision of the weights and activation functions of deep neural network model from 32-bit floating-point to lower bit width precisions, in our case to 16-bit. This technique aims to achieve faster inference times and reduced memory exploitation by leveraging powerful frameworks for quantization and parallel computing methods. Advanced tools like CUDA and TensorRT were used to maximize performance during model deployment and implementation of inference algorithm. The quantization of the convolution neural network model YOLOv11 proved highly effective in optimizing the performance of our autonomous boat's object detection system as the vision detector is no longer the bottleneck of the system.

Figure 10. shows the main pipeline for vision detection algorithm which uses CUDA for running inference directly on GPU architecture. After getting the first message with vision data from ZED2i we perform 10 times in a row warmup which is common practice for GPU resource initialization and ensuring stability in further data processing. The first step is preprocessing the image which includes resizing to 640x640 format and normalization of pixels values to the scope of [0, 1]. The second stage is the main detection process which is performing inference on GPU using CUDA framework. The algorithm copies image data directly to the GPU memory and then performs inference. Once this is done the results are copied back to the CPU memory. After this stage we are moving into the postprocessing stage which is focused on interpreting the results into proper form that can be fitted into our custom defined ROS2 message.

APPENDIX C
CALCULATION OF THE TORQUE REQUIRED FOR THE MOTOR

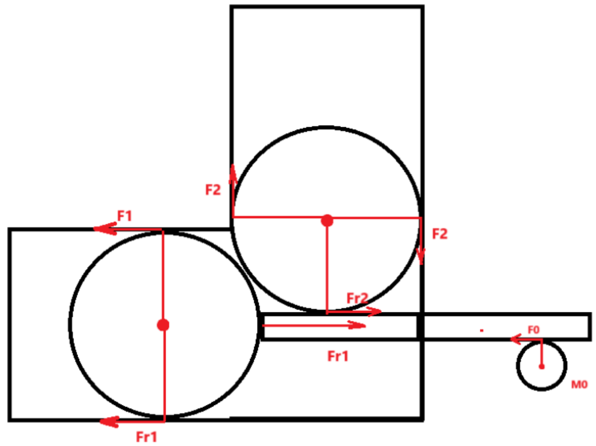


Fig. 10: Forces applied to the ball launcher

(1) $r_0 = 10\,\mathrm{N}$

$M_0 = F_0 = 0.01\,\mathrm{m}$

(2) $r_1 \cdot F_{11} = r \cdot F_{12}$

$F_{11} = F_{12} = F_1$

(3) $r \cdot F_{21} = r \cdot F_{22}$

$F_{21} = F_{22} = F_2$

(4) μ – plastic rubber friction coefficient

$\mu = 1$

(5) $F_{r_1} = F_1 + F_1 = 2F_1$

(6) $r \cdot F_{r_2} = r \cdot F_2 + r \cdot F_2$

$F_{r_2} = 2F_2$

(7) $F_{r_c} = F_{r_1} + F_{r_2} = 2F_1 + 2F_2$

($V = const.$

(9) $V_1 = V_2$

$V_1 = \frac{4}{3}\pi r^3$

$V_1 = \frac{4}{3} \cdot 3.14 \cdot (0.0275)^3 = 0.00317\,\mathrm{m}^3$

(10) $V_2 = \iint_{-0.0275-x}^{0.0275-x} \pi \cdot (r+x)^2\, dx\, dx$

(11) $\Delta p = 0.095\,\frac{\mathrm{MN}}{\mathrm{m}^2}$

(12) $A_{\text{contact}} = \pi x^2 \approx 2.826 \cdot 10^{-5}\,\mathrm{m}^2$

(13) $F_\mathrm{N} = 2.68\,\mathrm{N}$

(14) $F_1 = F_2 = F_\mathrm{N} \cdot 1 = 2.68\,\mathrm{N}$

(15) $F_{\mathrm{rc}} = 4 \cdot 2.68\mathrm{N} = 10.72\,\mathrm{N}$

(16) $M = 0.01 \cdot F_{\mathrm{rc}} = 0.1072\mathrm{Nm} = 10.72\,\mathrm{Ncm}$

Appendix D
Electrical Schematics

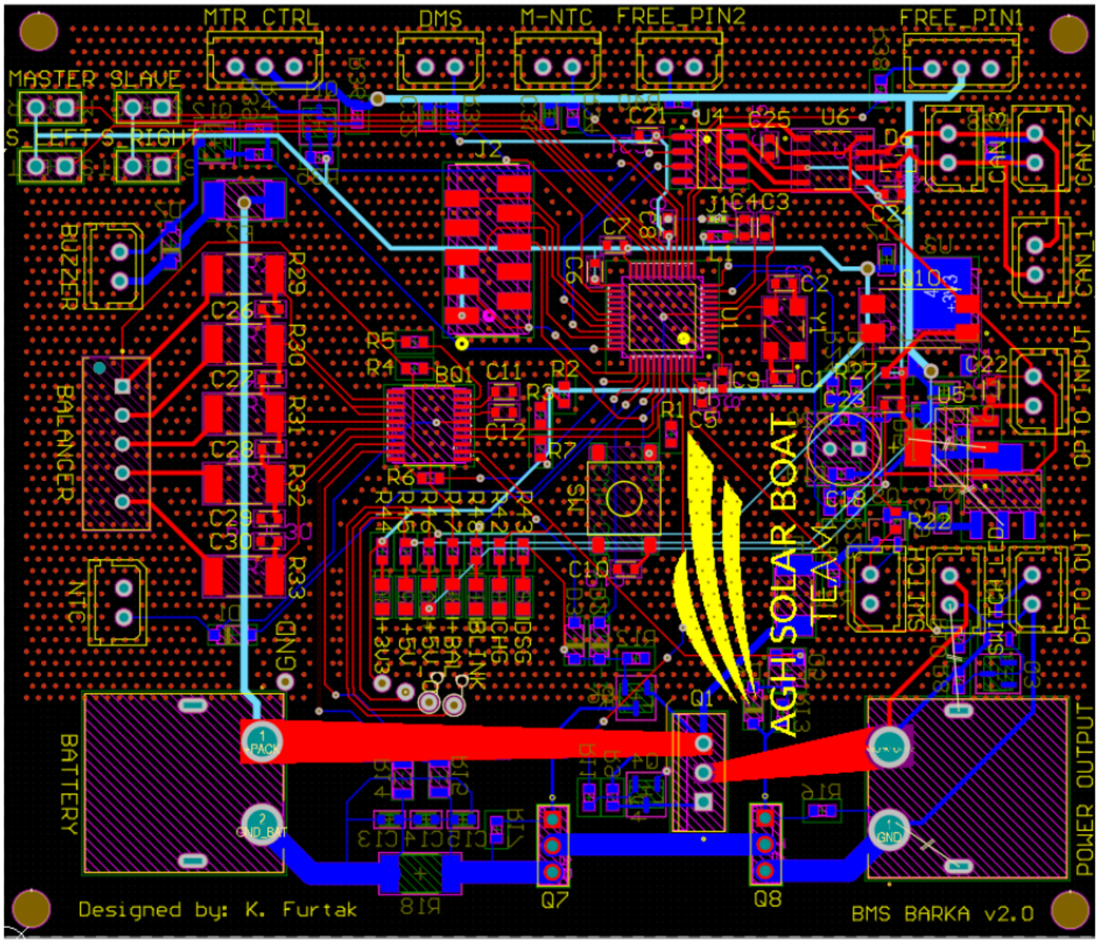
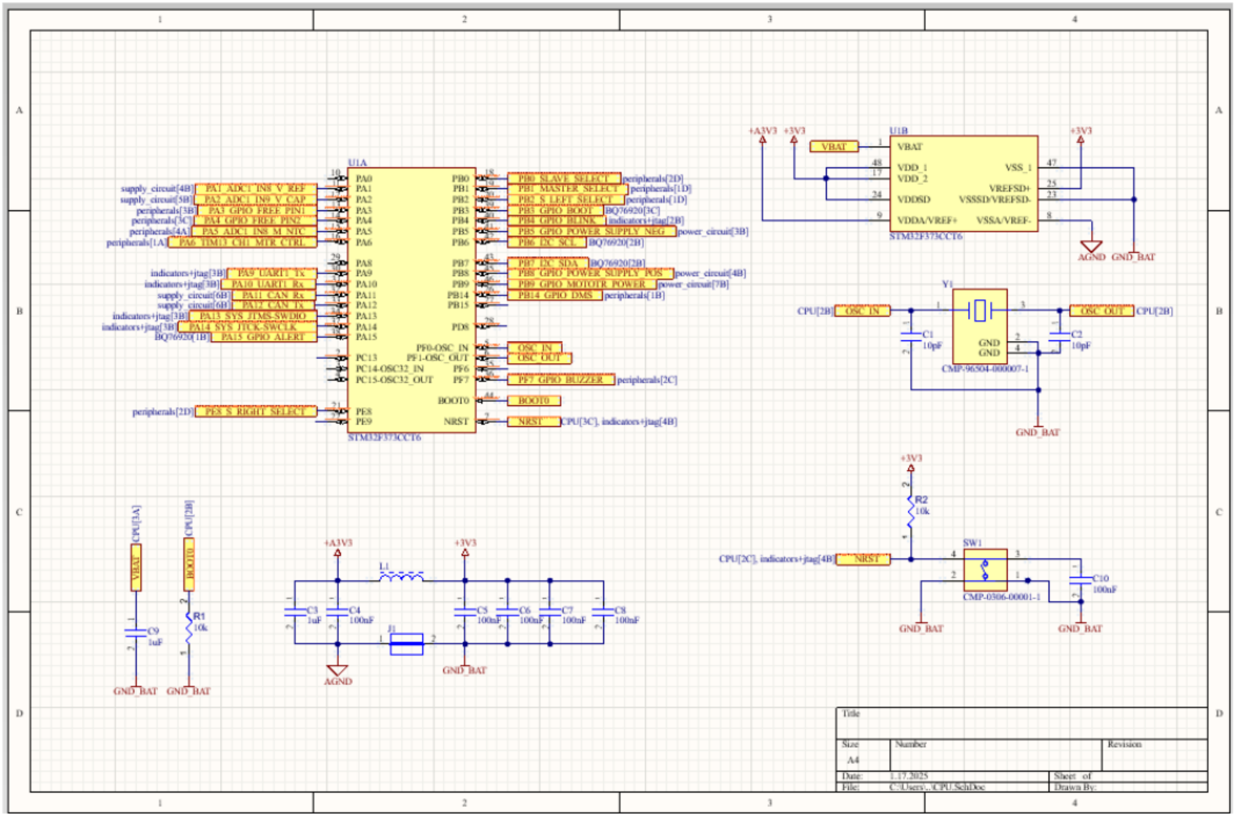
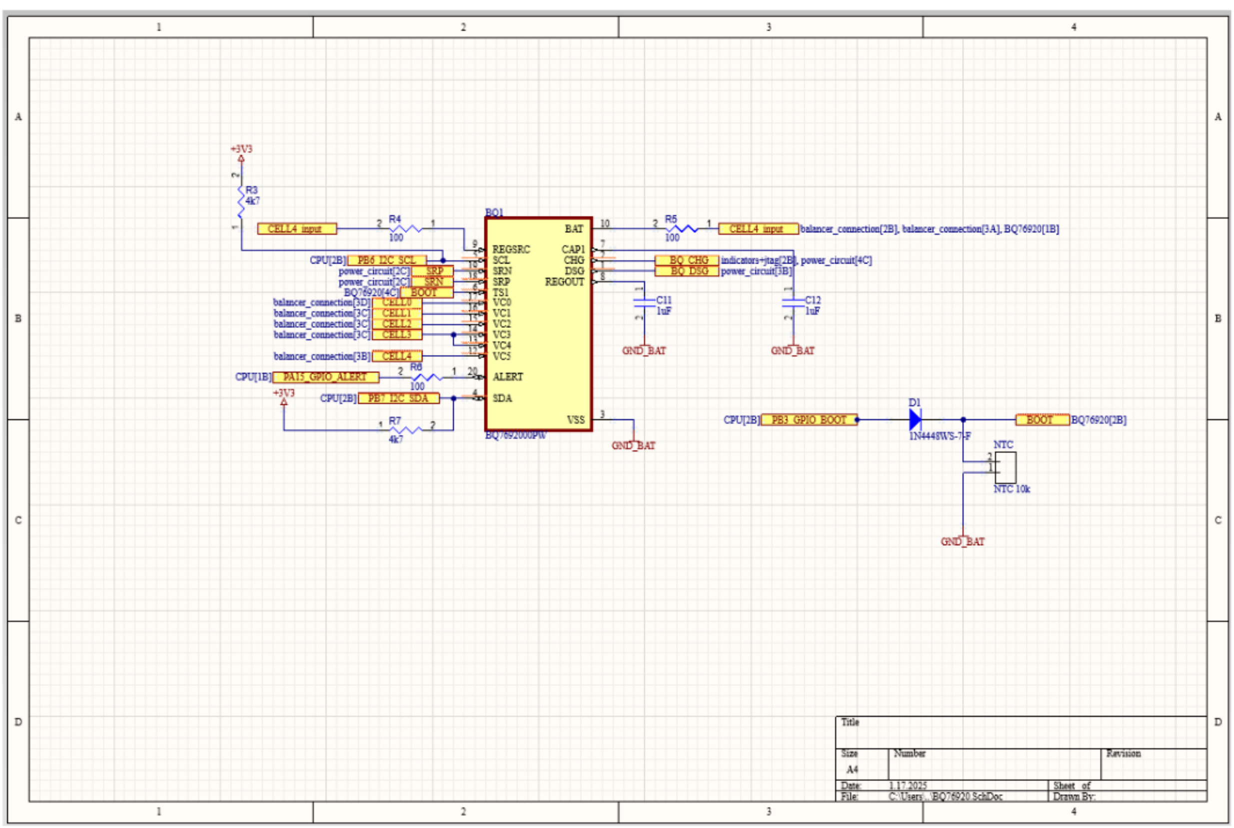
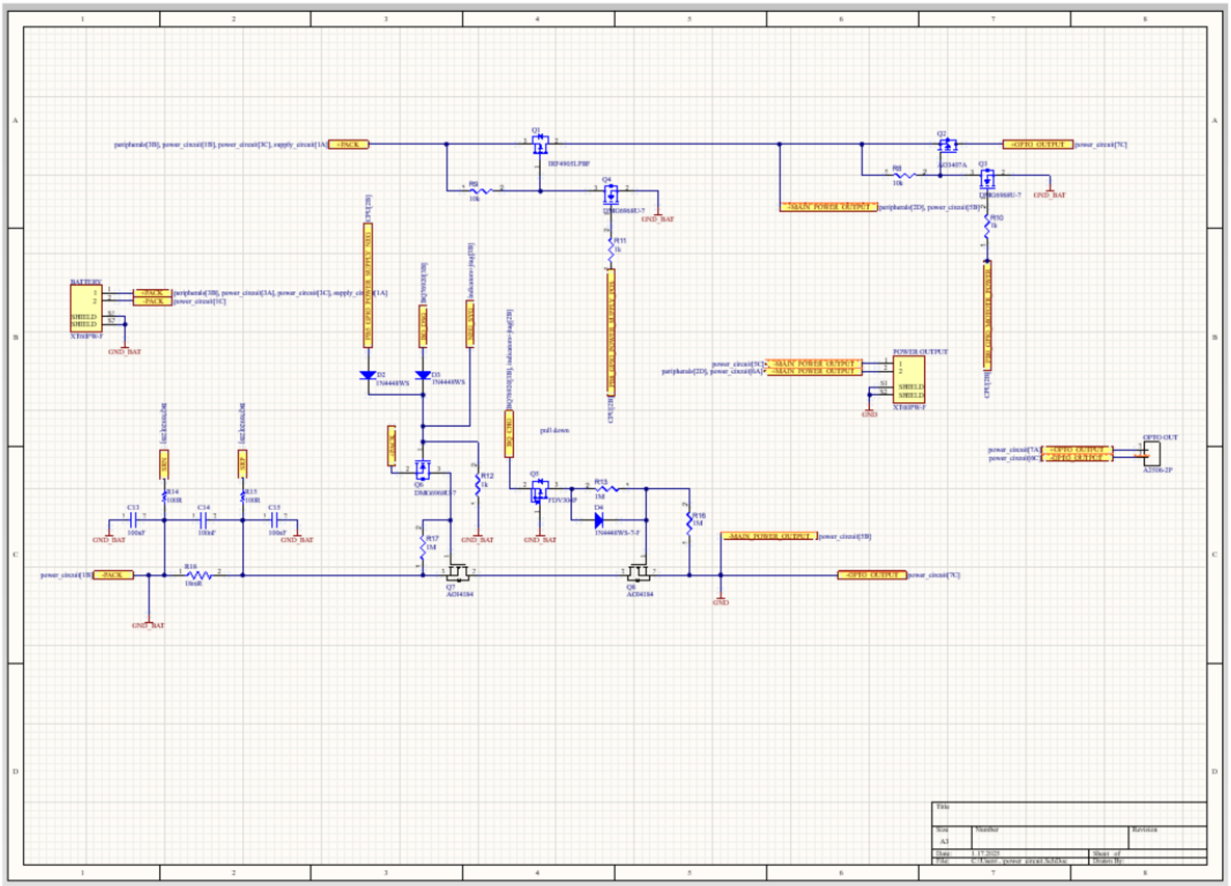
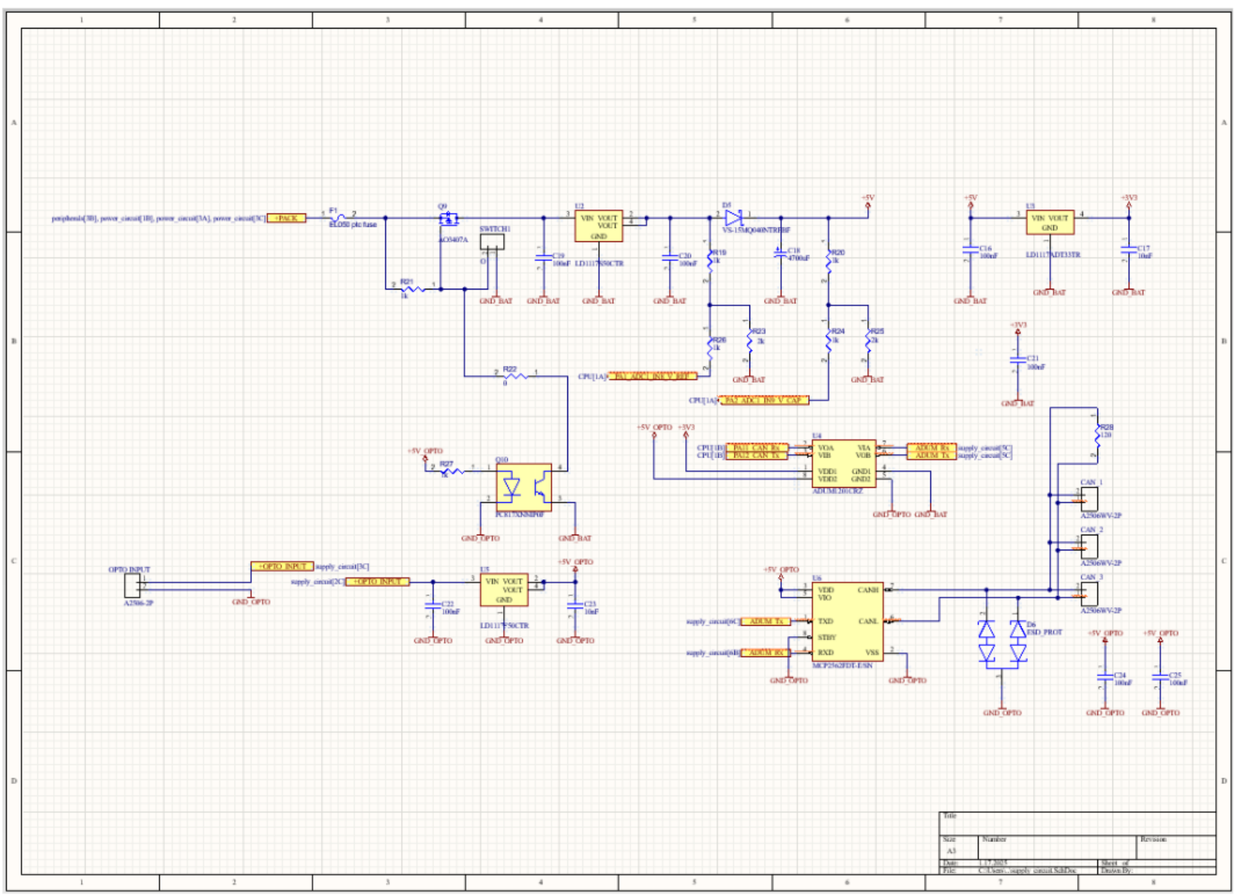


Fig. 11: Battery management system board design









APPENDIX E
COMPETITION STRATEGY