# Roboboat 2024: Technical Design Report
# UPRM Roboboat Team: AquamaRUM

Nomar Rodríguez Díaz
*Team Captain*
*Electrical Engineering*
Mayagüez, Puerto Rico
nomar.rodriguez@upr.edu

Yaidimar Pallens Rivera
*Team Co-Captain*
*Mechanical Engineering*
Mayagüez, Puerto Rico
yaidimar.pallens@upr.edu

Drazen Poventud
*Mechanical Lead*
*Mechanical Engineering*
Mayagüez, Puerto Rico
drazen.poventud@upr.edu

Pedro Pagan Torres
*Software Lead*
*Software Engineering*
Mayagüez, Puerto Rico
pedro.pagan6@upr.edu

Angel Rivera Garcia
*Electrical Lead*
*Electrical Engineering*
Mayagüez, Puerto Rico
angel.rivera120@upr.edu

Isabel Perez Diaz
*Business Lead*
*Operations Management*
Mayagüez, Puerto Rico
isabel.perez@upr.edu

Anthony Poyet Vargas
*Robotics Lead*
*Electrical Engineering*
Mayagüez, Puerto Rico
anthony.poyet@upr.edu

***Abstract*—The UPRM RoboBoat Team designed and manufactured a new vehicle for the 2024 competition: AquamaRUM. This report details all the specifications of the Autonomous Surface Vehicle (ASV) possesses to complete our main competition goal: to operate autonomously. AquamaRUM features a modular design that aims to improve stability, precision and adaptability of the ASV. The testing plan for AquamaRUM encompasses a dual-phase approach, incorporating a simulation to assess software algorithms that is followed by in-water testing sessions to validate real-world performance.**

## I. COMPETITION GOALS

All knowledge gained during Roboboat 2023 Competition provided the foundation for this year's competition strategy and goals. Our strategy consists of completing the following tasks: Task 1 - The Navigation Channel, Task 2 – Follow the Path, which demonstrates the ability to identify and maneuver through buoys avoiding contact with them, Task 5 – Speed Challenge, that taps into the same ability with object recognition and decision making. Nevertheless, after completing those 3 tasks we are hopeful AquamaRUM can accomplish Task 3 – Docking and Task 4 – Duck Wash that both tap into the precision of our ASV. Increasing the system complexity to accomplish these tasks allows the team to gain more points but the design and development time was extended which meant less in-water testing time. Also, increasing reliability requires simplifying the system, limiting the design. AquamaRUM is equipped with new technologies that push the boundaries to what has ever been accomplished by the team highlighting the need for a well-balanced and high-performance autonomous system.

## II. DESIGN STRATEGY

### A. Vehicle Design:

AquamaRUM features 2 symmetrical hulls, an aluminum extrusion custom made frame for the addition of additaments and two component boxes

#### 1. Hull Geometry and Arrangement:

For the Roboboat Competition 2024: Duck OverBoard, we decided to design and manufacture a completely new catamaran. This newly designed catamaran uses hulls that are 4ft long, 5in wide and when connected with a custom made aluminum extrusion frame, 26inches apart as shown in Figure 1.

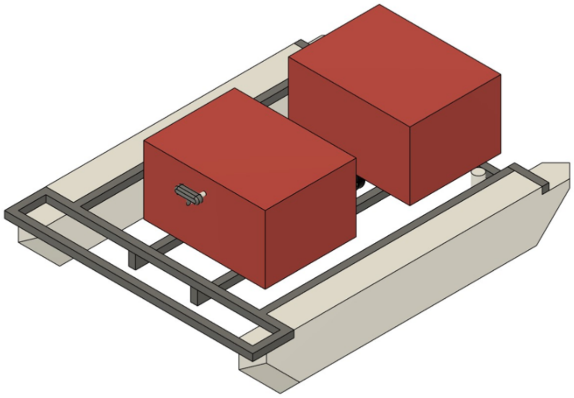


*Figure 1*

The material that was chosen for the hulls is fiberglass, due to their high strength-to-weight ratio. The hulls were designed to be thinner and longer than the previous ASV's hulls, to of the vessel, as well as improve its resistance to being pushed around by the wind by having more of the hull sit underneath the water. A rough estimate of the dimensions was designed in Fusion 360, and by using bouyancy calculations, the hull was was further iterated on to be able to comfortably support 45 pounds while leaving a few inches of clearance between the boxes and the surface of the water.

#### 2. Component Boxes:

To achieve the major protection of our components we decided upon two Husky storage containers with dimensions of 13.2in x 9.9in x 17.6in. These containers feature an internal gasket in the lid that ensures a waterproof seal and are connected to each other with a flexible PVC conduit pipe to secure all connections. Figure 2 shows the boxes and the connection between them.

*Figure 2*

#### 3. Thruster Assembly:

The thruster mounts were adapted from 1-½ PVC pipe, as shown in Figure 3, due to its high strength and water resistance. Their cylindrical shape reduces drag in the water when compared to another shape or surface.

*Figure 3*

### B. Electrical System

AquamaRUM features an optimized high-capacity power distribution system, with enhanced energy efficiency and a fail-safe kill switch to ensure reliability and performance of the ASV.

#### 1. Thruster Battery Pack:

To power the Blue Robotics T200 thrusters, the system needs a battery that can handle some time in a full throttle position while also powering the boat during the duration of the challenges and gates. Currently, we have two Zeee batteries (one of them for replacement) with a configuration of 4S1P delivering 14.8V with a capacity of 10Ah.

### 2. *System Battery Pack:*

The team has acquired two 7,000 mAh 77.7Whr 11.1V Gens Ace Battery Pack, by connecting them in parallel we will have 14,000mAh of capacity that will power the 12V components through a fuse block. Based on calculations, the system will consume 65.6 Whr during an hour of use with all components at max power. Therefore, the boat can operate at maximum power for 1.5 hours and still have 37% of the energy in the battery as reserve.

### 3. *Kill Switch:*

Besides from the physical kill switch, the vessel also counts with a wireless kill switch binded to one of the RC controller's switches. The vessel was tested at full throttle and by flipping the switch it immediately stopped and disconnected power to the thrusters.

## *C. Software Arquitecture*

AquamaRUM has a reliable and expandable software stack that runs on Python and can accomplish all navigation tasks. The code runs two primary processes: the vision pipeline, and the control system while on a separate computer, a GCS (Ground Control Station) is run to communicate with the vehicle. The main computer on AquamaRUM is a Jetson TX2, that connects via Mavlink to an Orange Cube flight controller who controls the thrusters and provides telemetry using its internal inertial measurement units (IMU) and the Here3+ GPS.

### 1. *Vision Pipeline*

The vision pipeline polls a RealSenseD435 to obtain color and depth frames. Color frames are fed to a YOLOv5 object detection model (ODM) that was trained in-house. Once the model performs interference and identifies the segments most likely to contain buoys, a color detection algorithm is run through the segments to determine the color of each buoy. For each object, the pipeline publishes its type, color, and relative radial coordinates (distance and angle) from AquamaRUM.

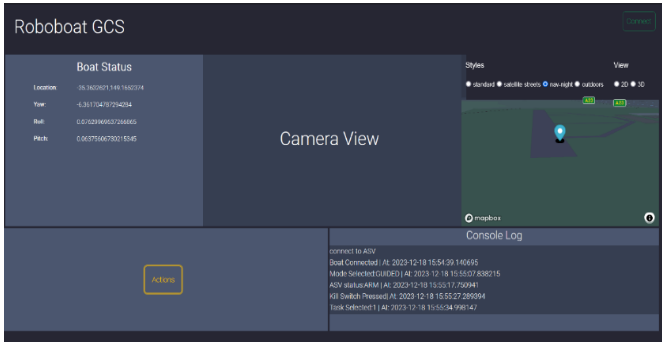
### 2. *Control System*

The control process constantly "listens" for messages from the GCS and publishes telemetry and status data for the GCS. It controls the status of the vehicle, the connection to the flight controller, and serves as a launcher for the subprocesses that control movement. AquamaRUM has two modes of autonomous operation: missions and pathfinding. The primary mode of autonomous operation is pathfinding or AI subsystem. This system is composed of a series of decision trees that receive input from the vision system and send movement commands to the flight controller. Decision trees are developed per-task and can be dynamically stitched together at runtime. Missions are run from pre-loaded mission files written in a domain-specific language developed in-house. These consist of a series of relative waypoints and simple commands that AquamaRUM can execute in order.

### 3. *Ground Control Station*

The Ground Control Station software monitors and controls the ASV and its autonomous operations. The user interface allows the user to visualize Real-Time Data (location, heading, speed, etc.) and provides action buttons to interact with AquamaRUM. It also provides a map to visualize AquamaRUM via satellite map.

*GCS User Interface*



Boat Status

Location: -35.3632621,149.1652374
Yaw: -6.361704787294284
Roll: 0.07629969637266865
Pitch: 0.06375606730215345

Console Log

```
connect to ASV
Boat Connected | At: 2023-12-18 15:54:39.140695
Mode Selected:GUIDED | At: 2023-12-18 15:55:07.838215
ASV status:ARM | At: 2023-12-18 15:55:17.750941
Kill Switch Pressed| At: 2023-12-18 15:55:27.289394
Task Selected:1 | At: 2023-12-18 15:55:34.998147
```

*Figure 4*

### 4. *Networking*

The team has developed a network strategy that communicates the ASV and GCS via Wi-Fi. After last year's competition, we identified that a reliable network needed good range, extended bandwidth, decent customizable default settings (to deal with interference changing channel width), variety of operation modes and an easy setup. Based on those needs, we implemented a strategy with two extenders, which can work at 5G frequency. Providing extender bandwidth and faster data transmission. Additionally, these extenders can operate in Router, Repeater and AP mode, which allow us to build the strategy as wanted.

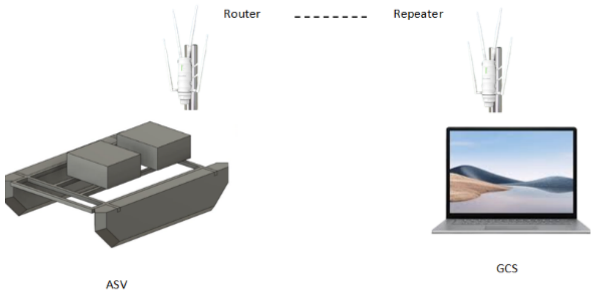


*Figure 5*

## *D. Task Approach:*

### 1. *Object Detection:*

Identifying the colors of objects is a crucial part of all navigation tasks. Since computation time comes at a premium, it is impractical to analyze every pixel in every region identified by the ODM. Depending on the type of object (i.e. ball buoy or tall buoy), different parts of the region are discarded. Besides cutting down on the amount of computation needed for each image, this also discards parts of the background which could affect color detection.

### 2. *Pathfinding:*

Once the data of objects seen is retrieved (color, type, location), the team implemented algorithms to determine AquamaRUM's movement based on vision data. Some of the same algorithms for pathfinding are implemented in similar tasks, like calculating midpoint coordinate between gate buoys, move to a calculated point, move until gate buoys are close enough for midpoint to be calculated, etc.

- **Navigation Channel**

In this task, AquamaRUM would be placed heading towards task's gate buoys. Then, AquamaRUM will detect if gate buoys are seen and close enough to calculate the midpoint between buoys (not farther from 2 meters). After ASV been close enough to the gate, midpoint between buoys is calculated, and AquamaRUM moves to that point. For a visual representation, see Figure 6.

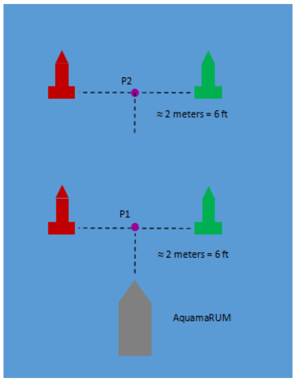


*Figure 6*

- **Follow the Path**

AquamaRUM's task involves dividing the path into sections between pairs of buoys. The Autonomous Surface Vehicle (ASV) positions itself in the middle of the starting pair, scanning the surroundings to the right and left until it detects buoys on both sides. If no other objects are seen during this time, it indicates the course might have ended. The ASV then scans for objects and the next pair of buoys, counting all objects in between. To navigate through objects, it uses the locations of the next pair of buoys and the objects. The ASV adjusts its path closer to the center of buoy pairs to avoid obstacles. This pattern continues until there are no more objects to scan, at which point the boat moves away from the course and performs necessary circles to count yellow objects. For a visual representation, see Figure 7.

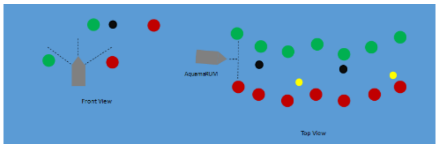


*Figure 7*

- **Docking**

In this task, the approach centers around the implementation of AquamaRUM having the ability identify the designated dock's shape or color and move towards a dock's slot. The first layer of implementation involves AquamaRUM perception module, utilizing computer vision algorithms to identify the designated shape or color associated with its target dock. Once AquamaRUM successfully recognizes the specific visual cue, the navigation module is activated, instructing the vehicle to propel itself towards the designated dock. Additionally, the team's implementation incorporates error-handling mechanisms to address unexpected situations, allowing the ASV to recalibrate its approach and successfully complete the docking task.

- **Speed Challenge**

For this task, AquamaRUM will locate the red and green buoys, determine the midpoint between them, position itself at that point, and record these coordinates as its 'home' position. Afterwards, it will locate and face the yellow buoy, then turn 40 degrees to the right. Subsequently, the ASV will move until the yellow buoy is out of sight. Following this, it will make an 80-degree turn to the left, bringing the yellow and blue buoys into view. At this juncture, the ASV will pinpoint the midpoint between these two buoys and navigate through that point until the buoys are no longer visible. Finally, the ASV will execute a 40-degree left turn and head back to the previously recorded 'home' coordinates. For a visual representation, see Figure 8.

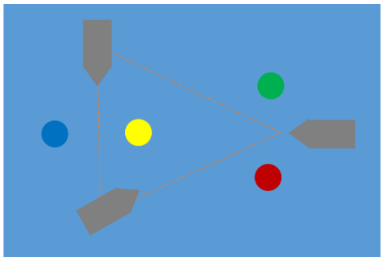


*Figure 8*

- **Duck Wash**:

In this task, AquamaRUM will detect the duck banner with the Computer Vision model from the Docking task. Once detected the boat will navigate in that direction to position itself in front of the duck banner. After, the boat will stabilize itself and activate a water pump that will shoot water for 3 intervals of 15 seconds to maximize the possibility of accurately shooting the banner for 5 seconds.

- **Collection Octagon**

In this task, AquamaRUM will have to position itself in front of the Collection Octagon. This is made possible with a Computer Vision model that recognizes black triangles, along with 2 arms holding a custom designed bucket that will be able to submerge and rotate with servomotors. After submersion the bucket will empty the water and then it will be raised to drop the items in the sorting box.

- **Delivery Octagon**

For this task we have a mechanical sorting box that will allow the rubber ducks to stay on the top half of the box and the racquet balls to stay on the bottom. Once the boat is in front of the corresponding object's banner it will open a door of the box of that object and tilt the box dropping the objects inside of the Delivery Octagon.

## III. TESTING STRATEGY

### *A. In-Water Testing*

The following testing plan aims to ensure AquamaRUM's reliability providing valuable insights into the ASV's performance. It consists of first, **Waypoint Navigation**, this assesses the ASV's ability to autonomously navigate predefined waypoint in controlled environments and evaluates the accuracy of the ASV's path following algorithms. Second, **Obstacle Avoidance,** introducing obstacles in the water to evaluate the avoidance and obstacle detection capabilities. Thirdly, the **Communication Systems**, we test the reliability and responsiveness of communication systems and data transmission between the ASV and remote operators. Lastly, **Endurance and Reliability,** we plan to achieve by conducting prolonged in-water testing sessions to test the endurance and reliability of the system.

AquamaRUM in-water testing sessions are scheduled to start in January 2024, due to unforeseen time limitations, leaving a month to test and improve performance. This experience has highlighted the importance of meticulous scheduling and contingency planning to account for unexpected challenges to ensure thorough testing and refinement processes.

### B. *Simulation Testing:*

To test the pathfinding algorithms without deploying the ASV, it is necessary to simulate two subsystems: the vision pipeline and the flight controller (Orange Cube). Using the interface-driven design described above, this was simply a matter of designing a new system that complied with the interface. To simulate the flight simulator and vehicle, the team used the Ardupilot SITL, an existing open-source software. Additionally, the team developed a vision simulator that can recreate any composition of buoys. Since scenarios are created using relative coordinates (i.e. "red buoy 5 meters to the front and 3m to the left of the ASV), they must be converted to global coordinates before the simulation begins. The result is a lightweight simulator that allows fast cycles of testing and iteration, as creating and modifying scenarios. The simulator sends messages using the same format as the real vision pipeline, so the pathfinding algorithms can react to the simulated surroundings and to the movement of AquamaRUM. For a visual representation, see Figure 9.

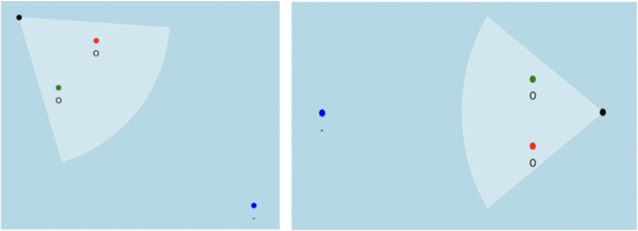


*Figure 9*

## IV. CONCLUSION

AquamaRUM represents UPRM Roboboat Team's dedication embodying a meticulously designed ASV for the 2024 Competition, after gathering data with our previous model, CatamaRUM from Roboboat 2023. There were several scenarios in which we could identify areas for improvement and findings were implemented to improve efficiency on our new design. AquamaRUM's modular architecture enhances stability, accuracy and efficiency. In addition, AquamaRUM possesses advanced algorithms that govern autonomous decision making and enhancements to the power distribution and communication systems. Alongside the team's testing plan, incorporating both a simulation and in-water testing, we are expecting readiness and reliability in the face of diverse challenges.

## V. ACKNOWLEDGEMENTS

The creation of AquamaRUM has been made possible through the dedicated efforts of our remarkable team and the generous support of our sponsors. We extend our heartfelt appreciation to each team member for their work and commitment which has been instrumental in every phase of AquamaRUM's development. Special thanks to our advisor, Dr. Ruben Diaz, who provided invaluable guidance and mentorship steering the project to success. Our gratitude extends to our sponsors, Boeing, Lockheed Martin, and GM, whose contribution has not only fueled the project financially but also enriched it with cutting edge technology and materials.

## VI. REFERENCES

(1) “The Cube Orange With ADSB-In Overview — Copter documentation,” ardupilot.org. https://ardupilot.org/copter/docs/common-thecubeorange-overview.html

(2) G. Jocher, “ultralytics/yolov5,” GitHub, Aug. 21, 2020. https://github.com/ultralytics/yolov5

(3) “MAVLink Interface — Dev documentation,” ardupilot.org. https://ardupilot.org/dev/docs/mavlink-commands.html

(4) “RoboBoat 2024,” RoboBoat. https://roboboat.org/programs/2024/

(5) “2024 TEAM HANDBOOK.” Accessed: Dec. 07, 2023. [Online]. Team HandBook

(6) "Catamaran Beam-to-Length Ratios Explained for Beginners," Catamaran Freedom. Available: https://ardupilot.org/copter/docs/common-thecubeorange-overview.html.

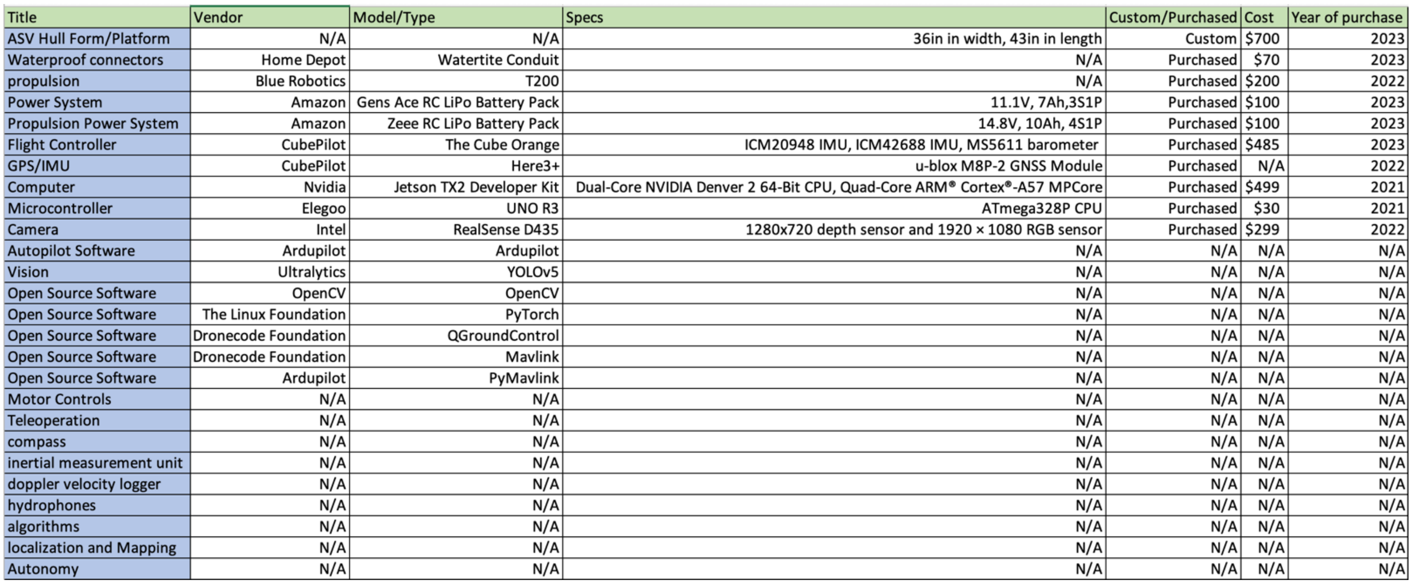
(7) “Emergency automatic switch from power supply to battery 12V module”, Power Switch Module. https://youtu.be/8CJFMBwNSQE?si=W2CI1U8HZG-EDX67

(8) *Amazon.Com: (Installation video) WAVLINK AC1200 Outdoor WIFI Extender Weatherproof, Outdoor* ..., http://www.amazon.com/WAVLINK-Extender-Weatherproof-Detachable-Antennas/dp/B09ZDQWFNG.

(9) “Getting Started.” *Getting Started | Axios Docs*, axios-http.com/docs/intro. Accessed 8 Dec. 2023.

(10) “V3.3.1 JavaScript Library: All. *Mapbox*, docs.mapbox.com/mapbox.js/api/v3.3.1/.

# Appendix A. Component List



| Title | Vendor | Model/Type | Specs | Custom/Purchased | Cost | Year of purchase |
|---|---|---|---|---|---|---|
| ASV Hull Form/Platform | N/A | N/A | 36in in width, 43in in length | Custom | $700 | 2023 |
| Waterproof connectors | Home Depot | Watertite Conduit | N/A | Purchased | $70 | 2023 |
| propulsion | Blue Robotics | T200 | N/A | Purchased | $200 | 2022 |
| Power System | Amazon | Gens Ace RC LiPo Battery Pack | 11.1V, 7Ah,3S1P | Purchased | $100 | 2023 |
| Propulsion Power System | Amazon | Zeee RC LiPo Battery Pack | 14.8V, 10Ah, 4S1P | Purchased | $100 | 2023 |
| Flight Controller | CubePilot | The Cube Orange | ICM20948 IMU, ICM42688 IMU, MS5611 barometer | Purchased | $485 | 2023 |
| GPS/IMU | CubePilot | Here3+ | u-blox M8P-2 GNSS Module | Purchased | N/A | 2022 |
| Computer | Nvidia | Jetson TX2 Developer Kit | Dual-Core NVIDIA Denver 2 64-Bit CPU, Quad-Core ARM® Cortex®-A57 MPCore | Purchased | $499 | 2021 |
| Microcontroller | Elegoo | UNO R3 | ATmega328P CPU | Purchased | $30 | 2021 |
| Camera | Intel | RealSense D435 | 1280x720 depth sensor and 1920 × 1080 RGB sensor | Purchased | $299 | 2022 |
| Autopilot Software | Ardupilot | Ardupilot | N/A | N/A | N/A | N/A |
| Vision | Ultralytics | YOLOv5 | N/A | N/A | N/A | N/A |
| Open Source Software | OpenCV | OpenCV | N/A | N/A | N/A | N/A |
| Open Source Software | The Linux Foundation | PyTorch | N/A | N/A | N/A | N/A |
| Open Source Software | Dronecode Foundation | QGroundControl | N/A | N/A | N/A | N/A |
| Open Source Software | Dronecode Foundation | Mavlink | N/A | N/A | N/A | N/A |
| Open Source Software | Ardupilot | PyMavlink | N/A | N/A | N/A | N/A |
| Motor Controls | N/A | N/A | N/A | N/A | N/A | N/A |
| Teleoperation | N/A | N/A | N/A | N/A | N/A | N/A |
| compass | N/A | N/A | N/A | N/A | N/A | N/A |
| inertial measurement unit | N/A | N/A | N/A | N/A | N/A | N/A |
| doppler velocity logger | N/A | N/A | N/A | N/A | N/A | N/A |
| hydrophones | N/A | N/A | N/A | N/A | N/A | N/A |
| algorithms | N/A | N/A | N/A | N/A | N/A | N/A |
| localization and Mapping | N/A | N/A | N/A | N/A | N/A | N/A |
| Autonomy | N/A | N/A | N/A | N/A | N/A | N/A |