# RoboSub 2024 Technical Design Report

Keith Chen (team lead), Aleksander Talamantez (Deputy), Khushee Goel, Avika Prasad, Leonard Wright

***Abstract*- This year, Team Inspiration's new team composed of mainly middle and high school students is building a STEM pipeline for the years to come. With less robotics experience, the team leveraged past lessons learned and capabilities, and will deploy a more capable autonomous underwater vehicle (AUV) Onyx to perform the missions while the smaller AUV, Græy, is serving as a testbed and an enabler to demonstrate intersub communication. The team integrated a new Fiber Optic Gyroscope (FOG) and Doppler Velocity Log (DVL) algorithm which combined with hydrophones, cameras and an onboard inertial measurement unit (IMU) are enabling more accurate navigation. The team continued to improve software, incorporating more robust localization algorithms. The team's open source RoboSub 101 Guide [1] is updated to accelerate and document learnings as a reference document for new RoboSub participants worldwide.**

## I. INTRODUCTION

Team Inspiration is a multi-institution, multi-grade organization that prioritizes learning STEM by developing AUVs and other autonomous vehicles for RoboNation and other competitions and educational purposes. In previous competitions, the team has developed two AUVs, Onyx and Græy (Fig. 1), which we are planning to utilize in RoboSub 2024.

Fig. 1: CAD of Onyx (left) and Græy (right).

With team members from multiple distant locations (e.g., different states across 3 different time zones), ranging from middle school to college, and most with no robotics experience, we had many challenges to manage to enable us to compete in RoboSub 2024 while working on RobotX. Supporting development for RobotX and RoboSub in the same year with a very small team is very challenging, especially when competing against teams with dozens of team members at college level and years of RoboSub experience. Excited by the challenge of competing in a university-level competition, the youngest members of our team decided in February 2024 to be trained in RoboSub by more experienced team members, who are preparing for RobotX.

Starting out with only one middle school and one high school student, Team Inspiration sped up onboarding to grow the team to seven members. We soon realized the need to further develop our RoboSub 101 guide (see Appendix C) to ease our journey in RoboSub. When we realized this guide could be useful for any RoboSub participant, we decided to share our RoboSub 101 by placing it in GitHub and webpage so teams from around the world can use it to onboard new team members and cooperate.

## II. COMPETITION STRATEGY

*A. General Strategy: Optimization*

We decided to use Onyx as the main competition sub and Græy as a backup, test bed, and communication sub. With six months, the new team opted to prioritize optimization of three key systems in Onyx:

- Improving navigation capabilities through integration of new sensors: DVL algorithm, FOG, and hydrophones.
- Overhauling the mission planning system to increase fault tolerance.
- Utilizing more efficient machine learning algorithms to identify objects.

*B. Competition Strategy: Task Execution*

The team plans to complete the missions as follows:

*1) Coin Flip and Gate:* Team Inspiration will request a coin flip, randomizing Onyx's orientation at the dock. Upon start of the run, Onyx will yaw clockwise to locate the gate and approach upon detection via the front camera. Onyx will then identify the counterclockwise symbol and yaw two revolutions after passing underneath it. Onyx will then proceed to the Buoy task, utilizing the Path for direction.

*2) Buoy:* Onyx will search for a path element using a downward-facing camera, align itself parallel to the path, and proceed forward until detection of the buoy. After this, Onyx will circumnavigate the buoy counterclockwise for one revolution. Upon completion of the revolution, Onyx will continue circumnavigating until detection of a second path element, which it will use to navigate to Bins in the same manner used to navigate to the Buoy.

*3) Bins:* Onyx will search for the bins while moving forward. Upon detection, Onyx will position its marker dropper over the bin and drop two markers into the blue/cold side of the bin before moving on to the next task.

4) *Pinger Detection:* The team plans to request a random pinger, randomizing the order of completion of *Torpedoes* and *Octagon*. For each task, Onyx will use three hydrophones to detect the acoustic signal from the pinger and triangulate its location. Onyx will then approach the pinger to perform step 5 or 6 below.

5) *Torpedoes:* Upon traveling to the acoustic pinger, Onyx will search for the map via cameras, and upon identification align itself to shoot torpedoes into the smallest hole and then the second smallest hole. Onyx will travel to the Octagon via camera perception and a priori knowledge of the general Octagon direction to perform Octagon function.

6) *Octagon:* Upon arrival at the pinger location and camera detection of the platform, Onyx will orient itself to surface. After surfacing, Onyx will dive to collect samples (in no particular order) via a grabber. It will resurface with each sample before dropping them in individual bins. Onyx will travel to Torpedoes mission via camera perception and a priori knowledge of the general Torpedoes target direction to perform Torpedoes function.

7) *Græy's Operations:* Græy will be deployed towards the end of Onyx's run, with a goal of demonstrating intersub communication with Onyx. Onyx will communicate with Græy to yaw simultaneously and flash their lights so the judges can observe the synchronization.

## II. DESIGN STRATEGY

The team decided to focus and optimize Onyx because its electronics enclosure is 50% longer than Græy's, providing room to integrate new sensors. Græy served primarily as our software test bed with the addition of a DVL, serving also as a demonstration of inter-sub communication. Design development focused on the mounting of new navigation sensors, improving our perception pipeline, and our mission planner.

We continued to use a perception-localization method to complete each mission. At close distance, a perception loop is used, utilizing camera streams to provide input into scripts for each mission for object detection and decision making. This provides that game elements are detected through camera input, allowing our AUV to make effective decisions. For longer distance navigation and localization, we utilized a sensor fusion system, including our DVL, FOG, hydrophones, and IMU. We used an onboard mission planner to choose the missions to complete, which are passed into the perception-localization method for execution.

*A. Mechanical Subsystem*

Our AUV's frames consist of 80/20 extruded aluminum. It is a simple and sturdy design that allows us to mount actuators and thrusters. This design is not hydrodynamic, but this is not a concern in low-speed operations. The low cost, modularity, adjustability, scalability, ease of assembly, and ease of vector thrust alignment to the direction of travel made an ideal hull structure that served well in past competitions.

The AUVs were designed with battery enclosures separate from the main electronics enclosure. This, along with the removable electronics platforms in the main enclosure, allowed for ease of maintenance and addition of new features. The subs were also designed with safeguards for reliable waterproofing. The end cap sealing the main electronics enclosure has two O-rings to provide redundancy, minimizing flood risk. In case of a leak, a moisture detector helps the sub quickly abort the mission and surface as a red light-emitting diode (LED) flashes to indicate this status. The external sensors communicate with systems in the main electronics enclosure via connectors which are sealed from the inside and the outside of the main end cap, rather than simple penetrators that can fail with weak epoxy.

We replaced our OAK-D Lites with OAK-D Wide cameras, which increased the horizontal field of view by 26° and the vertical field of view by 20° [2], [3]. OAK-D Wides also have onboard IMUs, which provide gyroscopic measurements for navigation. We continued to utilize USB low-light cameras when machine learning (ML) models were not necessary. Adding these cameras gives us higher-resolution images in low-light conditions.

*B. Electrical Subsystem*

As new team members, we found the high-level documentation pertaining to the electrical subsystems on Onyx lacked sufficient detail to familiarize ourselves with the low-level electrical subsystems. This familiarity was required before integration of new sensors. As we went about this familiarization, we documented all of the devices and wiring configurations in the AUVs to ease the familiarization process for future recruits (see Appendix C).

Our electronics enclosure on Onyx contains a three layer backplane, which ensures an organized layout with effective wire organization. This allows us to modularly expand our system to mount sensors and other electronics into our electronics enclosure and troubleshoot quickly. Our custom Power Distribution Board (PDB) provides hot-swap capabilities, and with two battery enclosures on Onyx. When the board detects that the voltage of one battery is too low, it will automatically switch to the other one. This gives us the capability to replace one battery without shutting down the system. The capabilities enable unlimited runtime to support testing.

Our current printed circuit board (PCB) for our hydrophones focuses on offloading as much processing as possible to our custom hardware. Each of our three hydrophones has a daughter board containing noise-isolated circuitry to filter the signal as much as possible. Signals are analyzed with custom algorithms, eliminating the need for computationally expensive Fast Fourier Transforms (FFTs). Our microprocessor, a Teensy 4.1, sends a pinger heading directly to the on-board computer.

*C. Software Subsystem*

The majority of development this season focused on software and its importance in competition performance. Our system uses Robotic Operating System (ROS) to facilitate communication between different nodes, allowing for modular expansion. Our modularity is demonstrated by our perception pipeline architecture, which uses a camera stream to send frames to scripts designed for each mission to determine AUV movement patterns. This means we can preserve our software architecture while modifying low-level code based on each year's game element specifications. The team developed four subsystems: perception, sensor integration, mission planning, and intersub communication.

*1) Perception:* We noted that most game elements this year can be detected accurately with conventional computer vision (CV) methods. We decided to utilize OpenCV methods for the missions *Hydrothermal Vent*, *Path*, and *Ocean Temperatures* (Fig. 2), and YOLO (You Only Look Once) v8 Convolutional Neural Network (CNN) models for *Enter the Pacific, Collect Samples* and *Mapping*. Since the YOLOv8 model is computationally expensive, this decision allowed us to save power without significantly impacting object detection.

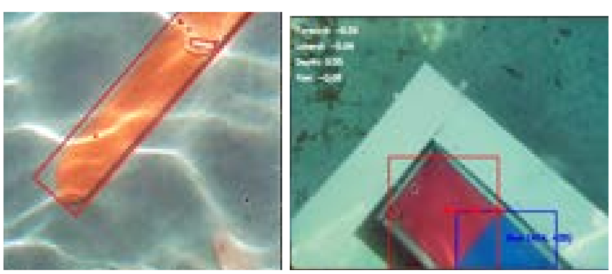

Fig. 2. Running OpenCV detection methods on *Path* and *Ocean Temperatures* training data.

The results of our object detection and localization are fed into our mission planner to aid in deciding Go/No-Go. Once inside the perception feedback loop, all data is given through camera streams, and all thruster and servo output is controlled by the processing of the position of objects on the screen. The perception loop is ended upon successful mission completion or when the assigned time for the mission has elapsed (Appendix C, Fig. 9).

*2) Sensor Integration:* To complete a mission, the AUV must navigate to the mission, as perception is only effective if the AUV localizes a mission. We utilized a sensor fusion system, combining our FOG, DVL, gyroscopes on OAK-D Wide cameras, Pixhawk IMU data, and hydrophone array on Onyx.

The FOG (Fig. 3) is a new sensor for Onyx that further enhances angular navigational capabilities. While previous compasses are susceptible to electromagnetic interference, the FOG removes this susceptibility. The team is currently configuring software to accurately collect headings from the sensor (Appendix B).

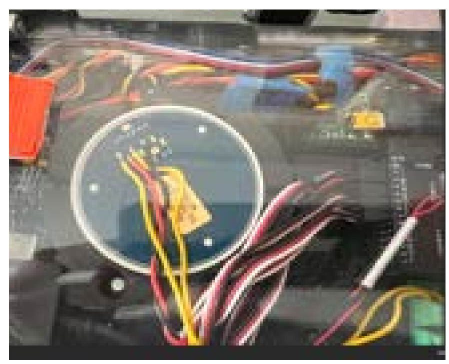

Fig. 3. FOG inside Onyx's electronics enclosure.

We also continued work from the previous year integrating a Teledyne Explorer DVL into Onyx. The DVL has four cells to produce sound waves, and the change in frequency of the echoes returning from the ground is used to calculate the AUV's velocity. Velocity is then integrated over time to calculate the AUV's change in position. We are currently configuring software to collect accurate position readings from DVL data.

The linear positional information from DVLs and heading information from gyroscopes give our AUVs effective navigation and localization on the xy plane. On the z plane, we incorporate our DVL data with barometer data, converting pressure readings into depth readings. For navigation, we input this data into proportional, integral, derivative

(PID) controllers to calculate our thruster outputs in the form of pulse-width-modulation (PWM) values sent to the thrusters. This allows for effective navigation and station keeping. When detecting an acoustic pinger, our hydrophone board calculates the relative heading change to orient with the pinger, whose value is passed into our motion functions to yaw accordingly.

*3) Mission Planning:* The mission planner inherited from the previous year's team executed the tasks in a preset sequence. While this approach was simple, it lacked fault tolerance as missions later in the run depended on completion of earlier missions. Therefore, the team decided to make a new mission planner module that considers the history of success, missions completed, and AUV location to decide which mission to complete next. Once the mission has been decided, the mission planner will call the specific mission module, which will handle all low-level processes, including running both the localization and perception loops needed to complete the mission. The mission module will then pass back certain variables to inform the mission planner of what should be done next.

*4) Intersub Communication:* While focusing on deploying Onyx to complete missions, we continued Team Inspiration's tradition of developing an intersub communication system via modems. Following our strategy of optimization, we preserved our previously built system and focused on integration of the data received from the other AUV into the onboard mission planner processing algorithm. We also looked to improve the content of data sent, focusing on data compression to allow more data to be sent with less bytes used.

## III. TESTING STRATEGY

With our contracted timeline and small team roster, efficient testing of components was imperative. We identified Græy as a parallel testbed for software development due to its similar sensors and software architecture, which made scripts transferable to Onyx upon completion of testing. We utilized unit and bench testing to verify individual components whenever possible to minimize the time and risk for each test. When completing water tests, we utilized detailed test plans (see Appendix B, Table I and Table II) for quick understanding of goals and documentation of successes and failures. A team "hot wash" after a test allowed us to identify lessons and next steps. For more in-depth testing information, see Appendix B.

### *A. Perception Tests*:

We began our perception tests by collecting in-water footage of the task elements via our AUVs. We then used the footage to train OpenCV models and YOLOv8 models. Finally, we performed bench testing using the footage or by holding the elements in front of the AUV cameras.

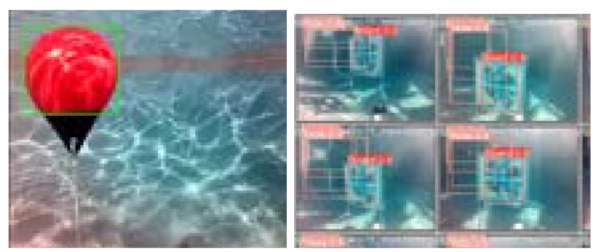

Fig. 4. Examples of detection testing for the *Hydrothermal Vent* (left, using OpenCV color thresholding) and the *Mapping* mission (right, YOLOv8 detection method). In this example, the *Hydrothermal Vent* detection test succeeded while the *Mapping* detection test failed.

### *B. Sensor Testing*:

We test each sensor individually for accuracy before integration on our AUVs, as locating points of failure is easier in simpler systems. However, some sensors required testing in-water to prevent overheating. When performing in-water tests, we followed a water testing protocol (see Appendix D) to ensure that electronics were protected from the underwater environment.

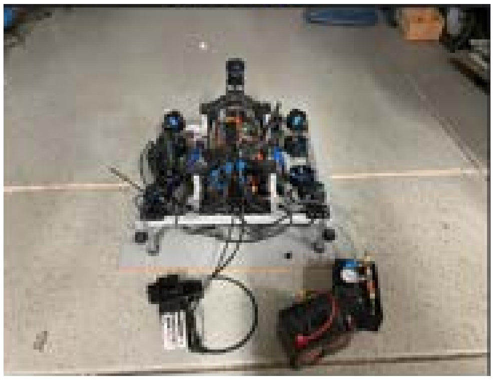

Fig. 5. Onyx vacuum test before water testing.

*C. Mission Runs*:

To perform mock mission runs and obtain training data for our ML models, it was imperative to fabricate task elements quickly and obtain access to a pool. For collecting training data and running a single mission, we used a personal pool. For running multiple missions or a single mission in a realistic format, we utilized a larger community pool.

## IV. SUMMARY

Team Inspiration is constantly looking to push its limits, and this year, we did just that. With our initially incomplete understanding of our sensors, hardware, and software, we failed fast and utilized quick iteration to accelerate our learning about RoboSub and our AUV systems to effectively compete this August. We built off our baseline and lessons learned, and continued to optimize our AUVs with navigation sensors and the improvement of our perception pipeline. Building off our experiences, we created documentation for new RoboSub students worldwide to onboard them and jumpstart new teams.

## V. ACKNOWLEDGEMENTS

Team Inspiration would like to thank the sponsors below:

*Diamond level*: Gilman Charitable Fund, Advancing Science Technology and Art, Qualcomm, and HP.
*Platinum level*: Nvidia and Intuit.
*Gold level*: Tinyvision.ai, Kenautics, Leverett Bezanson, and Medtronic.
*Silver level*: Blue Trail Engineering, Hologic, Chris Freeman, and San Diego Foundation.
*Bronze level*: SolidWorks, JetBrains, UC San Diego, MathWorks, Vinatech Engineering, RJE International and ePlastics.

We are grateful to our mentors who helped review our work and provided guidance on team management: Alex Szeto, Jack Silberman, Amit Goel, Venkat Rangan and Teresa To. We would also like to thank prior team members Eesh Vij, Colin Szeto, Mabel Szeto, and Lindsay Wright for their guidance and support as we moved through the development stage. We appreciate RoboNation and the Office of Naval Research for giving us the opportunity to compete in RoboSub. Lastly, we are indebted to our parents for their constant support.

## References

[1] Team Inspiration, RoboSub 101, https://docs.google.com/document/d/1p6zmOq8iW4hCqnPFx-JO1fsHeqHkgV8EnTyPkvKXcpc/edit#heading=h.45ur8fgkvllh (accessed Apr. 15, 2024).

[2] Luxonis, "OAK-D-Lite — DepthAI Hardware Documentation 1.0.0 documentation," *docs-old.luxonis.com*. https://docs-old.luxonis.com/projects/hardware/en/latest/pages/DM9095/#dm9095 (accessed Jun. 26, 2024).

[3] Luxonis, "OAK-D W — DepthAI Hardware Documentation 1.0.0 documentation," *docs-old.luxonis.com*. https://docs-old.luxonis.com/projects/hardware/en/latest/pages/DM9098w/#dm9098w (accessed Jun. 10, 2024).

[4] Ultralytics, "How to Train YOLOv8 Object Detection on a Custom Dataset," *Google.com*, 2024. https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolov8-object-detection-on-custom-dataset.ipynb#scrollTo=Y8cDtxLIBHgQ (accessed May 18, 2024).

[5] Aerospace Robotics Competition, "Aerospace Robotics Competition Tutorials! — ARC Tutorials 0.0.1 documentation," *arc-tutorials.readthedocs.io*. https://arc-tutorials.readthedocs.io/en/latest/index.html (accessed May 2024).

[6] "Index," *Game Manual 0*. https://gm0.org/en/latest/index.html# (accessed Jun. 30, 2024)

[7] S.W. Moore, H. Bohm, and V. Jensen, *Underwater Robotics : Science, Design & Fabrication.* Monterey, Ca: Marine Advanced Technology Education (MATE) Center, 2010.

APPENDIX A

COMPONENT SPECIFICATIONS

| Component | Vendor | Model/ Type | Specs | Custom/ Purchased | Cost | Year of Purchase |
|---|---|---|---|---|---|---|
| ASV Hull Form/ Platform (Frame, Waterproof Housing) | 80/20 Inc. | 1010 aluminum extrusion | Weight: 25 oz<br>Size: 22 in x 7.9 in | Purchased and machined in lab | $234 | 2019 |
| | ePlastics | Acrylic tube, 8 in. series | Size: 8" diameter | Donated | $343 | 2019 |
| Waterproof Connectors | Blue Trail Engineering | 10 cobalt series dummy plug<br><br>10 cobalt series locking sleeve<br><br>4 cobalt series cable termina-tion kit | https://www.bluetrailengineering.com/product-page/cobalt-series-dummy-plug<br>https://www.bluetrailengineering.com/product-page/removable-cobalt-locking-sleeve<br><br>https://www.bluetrailengineering.com/product-page/cable-termination-kit | Purchased<br><br>Purchased<br><br>Purchased | $201 | 2020 |
| Intersub communica-tion | Succorfish | Delphis V3.3 | https://succorfish.com/wp-content/uploads/2023/01/DELPHIS-Data-Sheet_V9_3.2.pdf | *Loaned by a sponsor* | *Loaned by a sponsor* | 2023 |
| Propulsion | Blue Robotics | T200 thrusters | Full Throttle FWD/REV Thrust @ Maximum (20 V)<br>https://bluerobotics.com/store/thrusters/t | Legacy | $200 apiece | 2019 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | 100-t200-thrusters/t200-thruster-r2-rp/ | | | |
| | Blue Robotics | T200 propellers | Max thrust: 49.82 N | Legacy | Included with thrusters | 2019 |
| Gripper | Blue Robotics | Newton Subsea Gripper | Grip Force: 28 N Jaw Opening: 2.75 in https://bluerobotics.com/store/rov/bluerov2-accessories/newton-gripper-asm-r2-rp/ | Legacy | $329 | 2019 |
| Torpedoes | Blue Trail Engineering | Under-water Servo SER-2000 | https://www.bluetrailengineering.com/product-page/underwater-servo-ser-20xx | Custom | $215 | 2023 |
| Marker Dropper | Blue Trail Engineering | Under-water Servo SER-2000 | https://www.bluetrailengineering.com/product-page/underwater-servo-ser-20xx | Custom | $215 | 2023 |
| Power System (Battery, Converter, Regulator) | Blue Robotics | Lithium-Ion Battery | 4s 14.8V 15.6 Ah | Legacy | Legacy | 2019 |
| | Blue Robotics | 5V, 6A power supply | 5V, 6A | Legacy | Came with BlueROV setup; Legacy | 2019 |
| | Mouser | Murata MYBSS054R6EBF | 54V Power supply<br>https://tinyurl.com/inspirationRegulator | Purchased | $22 | 2023 |
| | AliExpress | 100A Ideal Diode | 100A Ideal diode | Purchased | $72 | 2023 |
| | Custom | Unified Power Controller | Three: 12V 3A, 5V 3A, 3.3V 1.2A Power supplies | Custom | $400 | 2023 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Motor Control | Blue Robotics | Basic ESC | 30A brushless ESC https://bluerobotics.com/store/thrusters/speed-controllers/besc30-r3/ | Legacy | $36 apiece | 2019 |
| CPU | Nvidia | Nvidia Jetson Nano | 1.4 GHZ clock speed<br>4 GB RAM | Purchased | $99 (On Græy) | 2021 |
| | Nvidia | Nvidia Jetson Xavier NX | https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit | *Donated by a sponsor* | On Onyx | 2024 |
| Teleopera-tion | Blue Robotics | Fathom-X and Fathom-X Tether Interface (FXTI) | Communication: USB 2.0, Ethernet 10/100<br>https://bluerobotics.com/store/comm-control-power/tether-interface/fathom-x-r1/ | Legacy | Installed | 2023 |
| Inertial Measure-ment Unit (IMU) | Pixhawk | Invensense® MPU 6000 3-axis acceler-ometer/ gyroscope | 32-bit ARM Cortex M4 core with FPU<br>168 MHz/256 KB RAM/2 MB Flash<br>32-bit failsafe co-processor | Legacy | Included with Pixhawk on Græy and on Onyx (compass) | 2019, 2022 |
| | Dampener | XTORI Pixhawk dampener | Materials: plastic and rubber<br>Weight: 17 g | Legacy | $8 | 2021 |
| Doppler Velocity Log (DVL) | Teledyne | Explorer | Velocity Range: ±12 m/s<br>Long Term Accuracy: ±0.3% ± 0.2 cm/s<br>https://www.uniquegroup.com/wp-content/uploads/2022/10/Explorer_DVL.pdf | *Loaned by a sponsor* | *Loaned by a sponsor* | 2023 |
| | WaterLinked | A50 | Velocity Range: ±3.75 m/s<br>Long Term | Purchased | $ 5990 | 2020 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | Accuracy: ±1.01%<br>https://waterlinked.com/web/content/15701?unique=b4aef6d930bb256c64bae4e0ead8c56661bf12f0 | | | |
| Fiber Optic Gyroscope (FOG) | Fizoptika Malta | VG103 S2LND | Input range: 200 °/s<br>https://fizoptika.com/docs/fiber_optic_gyro_specification_vg103s-2lnd.pdf | Purchased | $3,060 | 2023 |
| Camera(s) | Blue Robotics | Low Light HD USB Camera | Pixel count: 2MP 1080P Onboard H.264 compression chip 32x32mm | Legacy on Græy<br><br>Purchased for Onyx | $198 (1x) on Græy (2x) on Onyx | 2019 |
| | Luxonis | OAK-D Wide Camera | 12 MP Resolution<br>60 FPS max frame rate<br>Focus (Full Frame): 60cm - ∞ | Purchased | $ 548 (1x on Græy) (2x on Onyx) | 2024 |
| Hydro-phones | Custom<br><br>Aquarian Audio & Scientific | Hydro-phone Circuit<br><br>AS-1 Hydro-phone | 100kHz sample rate for DTOA analysis. Variable gain control. 1-40kHz frequency lock range. Sub-watt and sub-degree precision under ideal conditions.<br><br>Linear range: 1Hz to 100kHz ±2dB Horizontal Directivity (20kHz): ±0.2dB Horizontal Directivity (100kHz): ±1dB Vertical Directivity (20kHz): ±1dB Vertical Directivity (100kHz): +6dB -11dB | Legacy<br><br>Legacy | $315 | 2020<br><br>2019 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Algorithms (acoustics) | Custom | Fast Fourier Transform (FFT) | Redundant | | Free | 2020 |
| Vision | Custom | Open Computer Vision | Color isolation, binary thresholding, contour approximation, erosion and dilation, area thresholding, and Contrast Limited Adaptive Histogram Equalization (CLAHE) | | Free/ Open Source | 2019 |
| | Ultralytics | YOLO | YOLOv8 | | Free | 2023 |
| Localization and Mapping | In-house | Custom | DVL, Hydrophones, CV | | Free | 2021 |
| Mission Planner | In-house | Custom | Mission planner | | Free | 2024 |
| Open Source Software | Open-Source (n/a) | OpenCV, Robot Operating System, Python, C++, Linux | Computer Vision, Inter-process communication, programming, computer operating system | | Free | 2019 |

# APPENDIX B

## TEST PLAN AND RESULTS

We split our testing plan into five parts: mechanical, sensor accuracy, perception, mission runs, and intersub communication. Additionally, we utilized detailed test plans when doing major tests. We made sure to save all results, so that troubleshooting and documentation would be as easy as possible.

### A. *Mechanical*

Mechanical testing was the most rudimentary yet arguably most important aspect of verifying our AUVs performance, as a failure in this system would result in other systems' failure as well.

1) *Græy SeaPerch Test (03/30/2024 - 04/06/2024)*: For our demonstration at the SeaPerch San Diego Tournament (see Appendix E), we used Græy to demonstrate an autonomous square maneuver. To do this, we had to test both Græy's mechanical and software ability.

On our first day of water testing Græy (03/30/2024), we nearly flooded our electronics enclosure due to an absent O-ring in one of our blank penetrators. Additionally, our dry interconnect penetrator for Græy's DVL was disconnected, causing water to flood the DVL bulk connector. From this, we learned the absolute necessity of a baseline checklist (see Appendix D) to ensure waterproof seals before submerging our AUVs. Græy and our DVL were fully restored, and there have been no incidents since.

2) *Onyx Water Test (06/19/2024)*: It was necessary to test Onyx much later than Græy, as there were complications with Onyx's computer wiring and changes in the electronics configuration (see Appendix C) of Onyx to include the FOG mount. We verified that our waterproof seals on the battery enclosures, electronics enclosure, and DVL worked after two hours submerged in a pool. Additionally, noting that our AUV was tilted at approximately a 30° angle upwards due to our electronics enclosure being present towards the stern of the frame, we added ballast on Onyx's aft side to balance out the vehicle.

3) *Servo Testing*: We realized that some 3-D printed servo parts on our marker dropper and gripper, due to their fragility, snapped off at a still unidentified time. Fabrication of new 3-D printed parts and testing of all three servos through software is our top mechanical priority.

### B. *Sensor Accuracy*:

We utilized unit testing and water tests to verify the accuracy of our FOG, DVLs, hydrophones, and IMUs.

1) *Fiber Optic Gyroscope (FOG)*: With our FOG having never been used before, we utilized both unit, bench, and water testing to verify the accuracy and availability of the sensor's data.

*a) Unit Test (05/19/2024):* We utilized unit testing to test our code that accesses the FOG, and to test the reliability of the data. Thanks to prior team members' work, we had a functional script and documentation to provide us with information on how to test the FOG. We verified that both the FOG was accessible and outputted accurate data (Fig. 6).


Fig. 6. Configuration of unit test for our FOG. By moving the physical FOG and looking at our outputs, we were able to verify that the FOG had an error of between 1° to 2°.

*b) Bench Test (06/13/2024):* The purpose of the bench test is to ensure that all sensors can be accessed through our onboard computer and that all actuators can be controlled through our flight controller. In the case of the FOG, we wanted to make sure that we could access its data. Initial testing proved that we needed to perform low-level software changes to access the correct USB port. After the small changes, we were able to access the FOG.

*c) Water Test (06/19/2024):* During Onyx's first water test (see Section A.2), we also took the opportunity to test the FOG in water. Several tests proved that the data output from our FOG was inaccurate – a rotation of Onyx by approximately 180° translated to only a 17.5° heading change on our FOG. We hypothesized that the 30° tilt on the aft-to-stern plane may have caused the inaccuracy of our FOG data. The following day, the results improved after this tilt was corrected but were still far from expected; a full 360° turn by Onyx was found to be only a 70° heading change by the FOG. Looking over all of our output data showed that the physical heading change to the output change was proportional, i.e. when a 180° degree turn was executed, the FOG would read approximately 35°. We hypothesize that this issue may be due to the positioning of the FOG on the electronics enclosure, rather than a hardware or a software issue.

*2) Doppler Velocity Log (DVL) test on Onyx (06/19/2024 - 06/22/2024):* Since our Teledyne DVL cannot be turned on outside of water without the chance of overheating and damaging vital components, we utilized in-water testing to test the accessibility and accuracy of the DVL. Testing proved our software that accesses and processes DVL data had several bugs which interfered with the time logs between messages. These errors caused our DVL connection to effectively shut down. Once this was fixed, we were able to collect DVL data. However, more processing was required because the DVL's heading is rotated 45 degrees clockwise from the AUV's heading. As forward and lateral data is collected relative to DVL orientation, the software must rotate the heading to output more accurate translational data.

*3) Hydrophones:* Our next testing step for verifying sensor accuracy is to test our hydrophone array. With a custom PCB and microcontroller that handles all processing, the testing process is to simply place an acoustic pinger inside a pool, and run code to access our hydrophone data. We expect to be given a relative heading to orient ourselves properly with the pinger.

*C. Perception*

Once we obtained the video footage necessary from our mock mission elements, we immediately worked on developing accurate YOLOv8 models and OpenCV detection methods. We ran our models and detection methods on our footage to ensure reliability.

*1) YOLOv8 (05/19/2024 - ):* We used the Jupyter Colab notebook provided by Ultralytics [4] to train a YOLOv8 model on a custom dataset, which we created using RoboFlow, a user-friendly method that we can use to annotate images to highlight the object of interest. Using the inbuilt ability of the notebook to test the model created on validation data, we found that it was necessary to perform preprocessing augmentations such as Gaussian Blur, rotations, and shear to increase the performance of the model. The *Enter the Pacific*, *Hydrothermal Vent*, *Path*, *and Ocean Temperature* tasks were able to be completed with 100% detection accuracy. However, our *Mapping* model performed extremely poorly. Specifically, the entire map that contains the holes through which we are to fire our torpedoes can be detected well; the holes themselves are detected with 0% accuracy. We are experimenting with performing more aggressive preprocessing and increasing iterations. We may also try using SIFT in as a replacement for a YOLOv8 model.

For further visualization purposes, we run our YOLOv8 models on our AUVs' cameras. Since we design and train our model based only on the object of interest, for example the red/hot section in the *Ocean Temperatures* mission, we use fabricated game elements in bench tests rather than using in-water mock mission setups. We found similar performances in our bench tests and our validation tests (Fig. 7).

Fig. 7. Example of testing the Buoy and Gate game elements using YOLOv8 models.

We tried using VLC as our media player for visualizing our camera streams. However, when running the YOLOv8 models, we realized that VLC did not have low enough latency to visualize a continuous camera stream; after experimenting, we found that MPV worked well enough to give us a real-time camera stream with visualized bounding boxes around objects of interest.

*2) Conventional Computer Vision (OpenCV, 06/05/2024 - ):* We utilized OpenCV color thresholding and histogram equalization techniques to complete the missions with simpler game elements, such as the *Path*, *Hydrothermal Vent,* and *Ocean Temperatures.* For the *Hydrothermal Vent* and *Path* missions in particular, we found morphology and color thresholding to work extremely accurately. While we tried to utilize similar techniques for *Enter the Pacific*, we found that the pool lighting was too subtle to detect the blue/cold side of the gate, and the red/hot side of the gate was inaccurate as well.

### *D. Mission Runs*

At this time, no mock-mission runs have been attempted. However, we plan to use our mock mission elements and our team pool in order to perform in-water tests of individual missions. When attempting to test the ability of our AUVs to complete multiple missions in a single run, we will use a larger community pool to set up a mock competition configuration (see Appendix B, table II for our test plan for our first mission run).

### *E. Intersub Communication*

Because of our focus on optimizing Onyx, our intersub communication functionality is a stretch goal and has not been tested. However, we plan to water-test both AUVs and test their ability to communicate over long distances and execute a synchronized yaw with LED visual cues.

TABLE I

FIRST ONYX WATER TEST

<table>
<tr><td colspan="3" rowspan="3">First Onyx Water Test</td><td colspan="3">Approval Authority</td></tr>
<tr><td>Keith Chen (Team Lead)</td><td colspan="2"></td></tr>
<tr><td>Alek Talamantez (Mechanical Lead)</td><td colspan="2"></td></tr>
<tr><td colspan="2">Date: 2024-06-19</td><td colspan="4">Mission Title: Onyx Water Test w/ FOG, DVL, and Motor Testing</td></tr>
<tr><td colspan="2">Test #: Onyx 1</td><td colspan="2">Location: Team Inspiration Pool</td><td colspan="2">Risk: Medium (first time in nine months Onyx being put in the water)</td></tr>
<tr><td colspan="2">Software Version: Baseline 1</td><td colspan="2">Hardware Sensors Mounted: Barometer, Camera, IMU, DVL, FOG</td><td colspan="2">Hardware Sensors Used: DVL, FOG</td></tr>
<tr><td colspan="6">Scope: <u>Primary:</u> Check waterproofing of Onyx, enable autonomous movement.<br><u>Secondary:</u> Test DVL and FOG accuracy.</td></tr>
<tr><td colspan="2" rowspan="2">Roles</td><td colspan="2" rowspan="2">Walkie Talkies/ Cell Phones</td><td colspan="3">Times</td></tr>
<tr><td>Event</td><td>Time</td><td>Actual</td></tr>
<tr><td>Test Conductor</td><td>Leonard</td><td>Ground</td><td>N/A</td><td>Packup</td><td>3:30 pm</td><td>4:18</td></tr>
<tr><td>Sub Launcher</td><td>Keith</td><td>Pool Deck</td><td>N/A</td><td>Go to Pool</td><td>3:45pm</td><td>4:30</td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| **Ground Control** | N/A | **Lifeguard** | N/A | **Test** | 4:00 pm | 4:57 |
| **Data Collector** | Leonard | **Tether** | N/A | **Cleanup** | 6:00 pm | |
| **Photographer** | Keith | | | **Leave Pool** | 6:15 pm | |
| **Safety Checker/QA** | Alek | | | **Put Away** | 6:20 pm | |
| **Person in water** | Alek | | | **Hot Wash** | 6:45 pm | |
| **Status** | | | | | | |
| **Sub** | | GO/NO GO | | | | |
| **Ground** | | GO/NO GO | | | | |
| **Pool Deck** | | GO/NO GO | | | | |
| **Tether** | | GO/NO GO | | | | |

| **Attendance** | **Test Notes** |
|---|---|
| **Leonard**<br>**Alek**<br>**Keith** | **Resources Needed:** Onyx with fully charged battery, tether, ethernet cord, micro USB, ethernet to USB cord, personal laptop.<br>**Environment:** Team Inspiration Pool.<br>**Expected Results:** Able to get Onyx running autonomously in four ranges of motion; forward, lateral, yaw, vertical. Additionally to obtain FOG and DVL data.<br>**Prerequisites:** Make sure Onyx is waterproof – this should not take too much time, but we need to verify.<br>- Onyx WiFi IP: 192.168.0.198<br>- Onyx Ethernet IP through tether (untested): 192.168.2.2<br>**Test procedure:**<br>- Followed AUV setup using the baseline checklist.<br>- Run roscore, mavros. Then run control.py to enable keyboard control to run thrusters.<br>- Run dvl.py, fog_interface.py, to test the accuracy of the sensors. |

| | |
|---|---|
| | **Risk Management:** Alek will be in the pool, and he will be continuously checking any water leaks, including two dip tests (battery enclosure, fully submerged). |

**Results and Reflections**

- Onyx needs a hotfix method to enable SSH capabilities through tether. Command is "sudo ifconfig eth0 192.168.2.3 netmask 255.255.255.0" after SSH'd through Wifi.
- MAVROS node on Onyx never initializes; therefore unable to arm Pixhawk controller to run thrusters.
- DVL script is slightly faulty – time since last message received is not handled properly.
- FOG is inaccurate; gives us something like 5 times less of an angle change than is actually changed in reality

TABLE II

RUNNING THE GATE MISSION USING GRÆY TEST PLAN

<table>
<tr><td colspan="3" rowspan="3">Græy Gate Mission</td><td colspan="3">Approval Authority</td></tr>
<tr><td>Keith Chen (Team Lead)</td><td colspan="2"></td></tr>
<tr><td>Leonard (Network Lead)</td><td colspan="2"></td></tr>
<tr><td colspan="2">Date: 2024-07-02</td><td colspan="4">Mission Title: Græy Water Test w/ Perception, YOLOv8, and Motor Testing</td></tr>
<tr><td colspan="2">Test #: Græy 2</td><td colspan="2">Location: Team Inspiration Pool</td><td colspan="2">Risk: Low</td></tr>
<tr><td colspan="2">Software Version:<br>Baseline 1</td><td colspan="2">Hardware Sensors Mounted:<br>OAK-D Wide Camera, IMU,</td><td colspan="2">Hardware Sensors Used:<br>OAK-D Wide Camera</td></tr>
<tr><td colspan="6">Scope: <u>Primary:</u> Test execution of the Gate mission</td></tr>
<tr><td colspan="2" rowspan="2">Roles</td><td colspan="2" rowspan="2">Walkie Talkies/<br>Cell Phones</td><td colspan="3">Times</td></tr>
<tr><td>Event</td><td>Time</td><td>Actual</td></tr>
<tr><td>Test Conductor</td><td>Keith</td><td>Ground</td><td>N/A</td><td>Packup</td><td>3:30 pm</td><td></td></tr>
<tr><td>Sub Launcher</td><td>Leonard</td><td>Pool Deck</td><td>N/A</td><td>Go to Pool</td><td>3:45pm</td><td></td></tr>
</table>

<table>
<tr><td>Ground Control</td><td>N/A</td><td>Lifeguard</td><td>N/A</td><td>Test</td><td>4:00 pm</td><td></td></tr>
<tr><td>Data Collector</td><td>Leonard</td><td>Tether</td><td>N/A</td><td>Cleanup</td><td>6:00 pm</td><td></td></tr>
<tr><td>Photographer</td><td>Keith</td><td></td><td></td><td>Leave Pool</td><td>6:15 pm</td><td></td></tr>
<tr><td>Safety Checker/QA</td><td>Alek</td><td></td><td></td><td>Put Away</td><td>6:20 pm</td><td></td></tr>
<tr><td>Person in water</td><td>Alek</td><td></td><td></td><td>Hot Wash</td><td>6:45 pm</td><td></td></tr>
<tr><td colspan="7">Status</td></tr>
<tr><td>Sub</td><td></td><td colspan="2">GO/NO GO</td><td colspan="3"></td></tr>
<tr><td>Ground</td><td></td><td colspan="2">GO/NO GO</td><td colspan="3"></td></tr>
<tr><td>Pool Deck</td><td></td><td colspan="2">GO/NO GO</td><td colspan="3"></td></tr>
<tr><td>Tether</td><td></td><td colspan="2">GO/NO GO</td><td colspan="3"></td></tr>
<tr><td>Attendance</td><td colspan="6">Test Notes</td></tr>
<tr><td>Leonard<br>Alek<br>Keith</td><td colspan="6">Resources Needed: Græy with fully charged battery, tether, ethernet cord, micro USB, ethernet to USB cord, personal laptop.<br>Environment: Team Inspiration Pool.<br>Expected Results: Græy identifies and approaches the gate under the counterclockwise side, passing through.<br>Prerequisites: Waterproof test Græy<br>- Græy WiFi IP: 192.168.0.182<br>- Græy Ethernet IP through tether: 192.168.2.2<br>Test procedure:<br>- Followed AUV setup using the baseline checklist.<br>- Run roscore, mavros. Then run control.py to enable keyboard control to run thrusters.<br>- Run /missions/gateMission.py</td></tr>
</table>

| | |
|---|---|
| | **Risk Management:** Alek will be in the pool, and he will be continuously checking any water leaks, including two dip tests (battery enclosure, fully submerged). |

**Results and Reflections**

- N/A

# APPENDIX C

## ROBOSUB DOCUMENTATION

With four years of prior experience competing in RoboSub, and participating in RobotX, RoboBoat, and other robotics competitions, Team Inspiration has aggregated a vast amount of information which can be used to quickly onboard new team members. However, our team has never focused heavily on creating reliable documentation for such a purpose, as Team Inspiration had consisted of a group of students who have grown their knowledge together since 2019, and never needed documentation to be able to learn and continue to develop systems that previous team members in prior seasons had used in competition. However, starting in 2023, and especially this season, it became important to create documentation with the purpose of allowing new team members with minimal experience with RoboSub or robotics to be able to take a productive role in the development process. We created a RoboSub 101 guide [1] which is largely based on the Drone 101 document created by Aerospace Robotics Competition (ARC) [5]. The purpose of the document is to give new RoboSub team members a basic understanding of what designing an AUV for the RoboSub entails, and inculcating basic mechanical, electrical, and software principles, with a systems engineering style approach. This document can be shared with other prospective RoboSub teams to ease their learning curves as well, and has been shared on the RoboSub discord server. The document's final goal is to be similar to the Game 0 Manual for First Tech Challenge (FTC) teams [6], a living document in which other RoboSub teams can submit entries. Our other focus was to fully document our systems for new team members to quickly understand the high and low-level aspects and to be able to utilize the documentation for their own development processes.

### *A. RoboSub 101 Outline:*

We split our RoboSub 101 outline into multiple sections, each one covering a specific part of AUV development. We took content from the *Underwater Robotics* Textbook developed by Marine Advanced Technology Education (MATE) [7], and mentor and student experience and knowledge when creating the document.

*1) RoboSub 101:* The first section of our document, this is an introduction to basic terms such as what an AUV is, and basic mechanical concepts such as frames, batteries, and propulsion.

*2) Fundamentals:* The second section of our document, *Fundamentals* gives a basic description of several ROV/AUV competitions, such as SeaPerch, SeaGlide, MATE ROV, and RoboSub.

*3) The Theory Behind Underwater Robotics:* Gives a basic physics lesson on the theory behind the movement of underwater vehicles.

*4) Software:* Lists a series of websites and documents in which documentation can be found for software tools and documentation which many teams, including ours, use. These include ROS, Unity, Gazebo, among many others.

*5) Programming:* An overview of programming, discussing trade offs between different languages to build software. Additionally, this section discusses best practices when developing software, especially in a team environment.

*6) Mini-Lessons:* A series of links leading to videos recorded by Team Inspiration, teaching viewers about specific technical skills such as hull design and sensor optimization.

*7) Hardware:* Details all basic relevant hardware, with technical specifications of the hardware present on our AUVs.

*8) Autonomy*: Contains a layman explanation of the logic behind autonomous systems.

*9) SEIT: Systems Engineering, Integration, and Testing:* Describes the system engineering methodology. This is one of our most lengthy sections, and discusses in great detail the systems engineering processes, with tips and examples.

*10) Team Dynamics:* Discusses the importance and ideas behind team building and financial handling.

*11) Safety:* Because developing an AUV has the potential of becoming dangerous, we include a safety section that details basic safety when working in a lab environment.

*12) Technical Communication:* Based on our experiences doing outreach and competing in the Design Documentation portion of RoboSub, RoboBoat, and RobotX, we describe best practices for writing technical papers, creating team videos, webpages, presentations, and social media.

*13) RoboNation Resources:* Lists links to various RoboNation and RoboSub resources.

*14) Team Inspiration Resources & Contact:* Lists links to various Team Inspiration social media accounts.

*15) References:* Lists the resources whenever possible used to create the RoboSub 101 document.

*B. Systems Documentation*

Despite all of the technical knowledge aggregated over the years, our new team found it difficult to learn about the systems we inherited without continuous direct communication with team members from previous years. To further our own learning and to prevent this from happening in future seasons, we strove to document all low-level processes in our software, electrical, and mechanical subsystems.

*1) Software:* Our software architecture, while being well-designed and modular, was extremely convoluted from last year's competition, since, as our team pushed the limit of our capabilities, there was no time to organize the codebase, leading to deprecated files and functions spread throughout our system. Therefore, when reading through our code, it was difficult to understand which processes were used, and what function those processes served. We decided to move through the entire codebase and document all low-level processes, and for code that we reused from 2023, we made sure to preserve the documentation. For high-level processes, we utilized well-labeled diagrams to ensure that both high-level and low-level processes were able to be well-understood (Fig. 8, Fig. 9).

*2) Electrical:* Our electrical systems were severely in need of documentation, as our old interconnect diagrams, while giving us a high-level overview, did not help when trying to troubleshoot/expand low-level wiring, for example when installing a new Jetson NX onto Onyx, or when installing our FOG onto Onyx's electronics enclosure. We therefore created new low-level diagrams which detail all of the devices, wires, and connections with voltages on both Græy and Onyx (Fig. 10, Fig. 11).

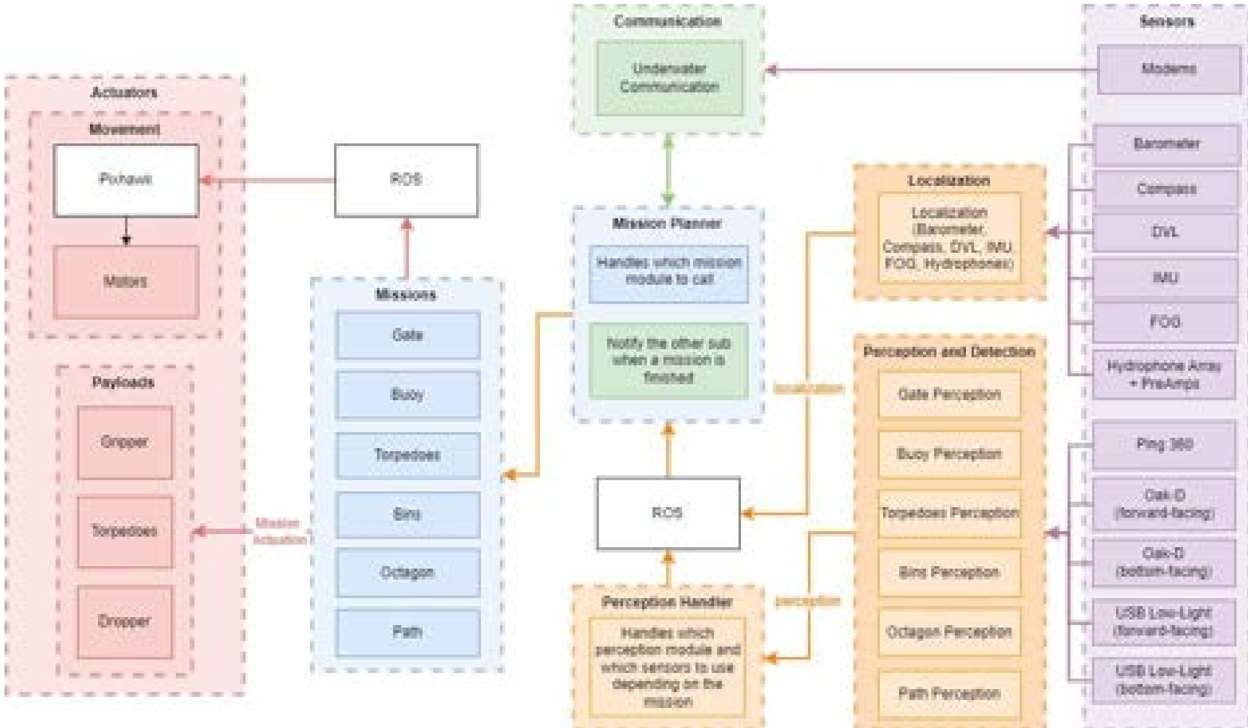


Fig. 8. High Level Software Architecture for Græy and Onyx.

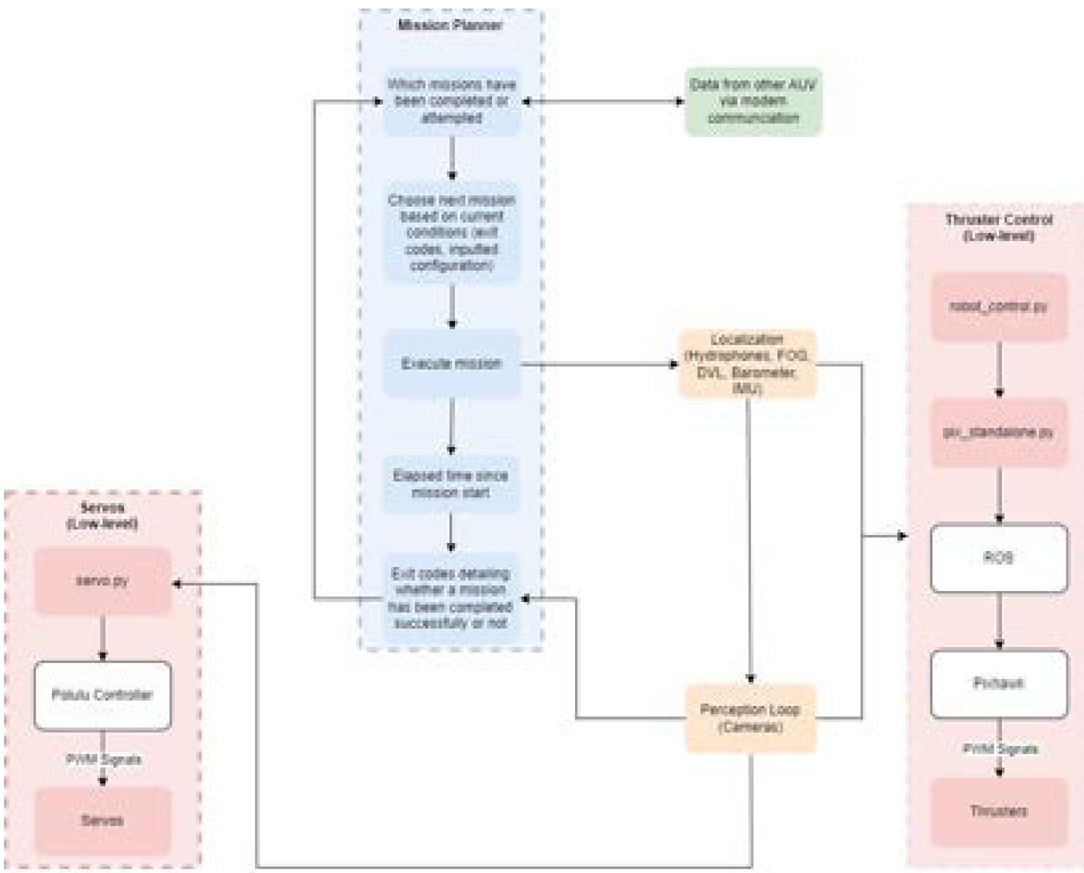


Fig. 9. Mission Planner Architecture.

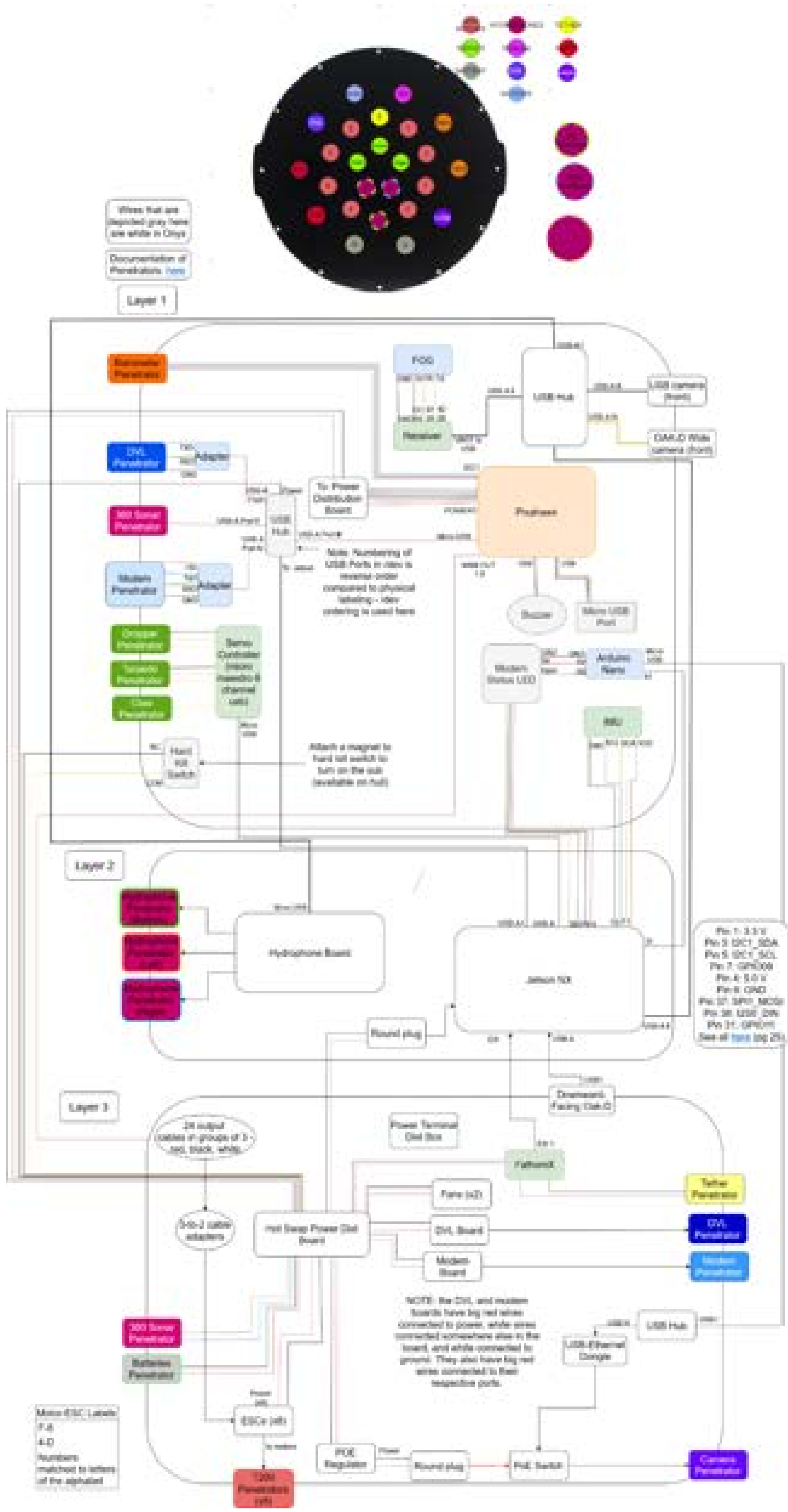


Fig. 10. Low Level Electrical Diagram for Onyx.

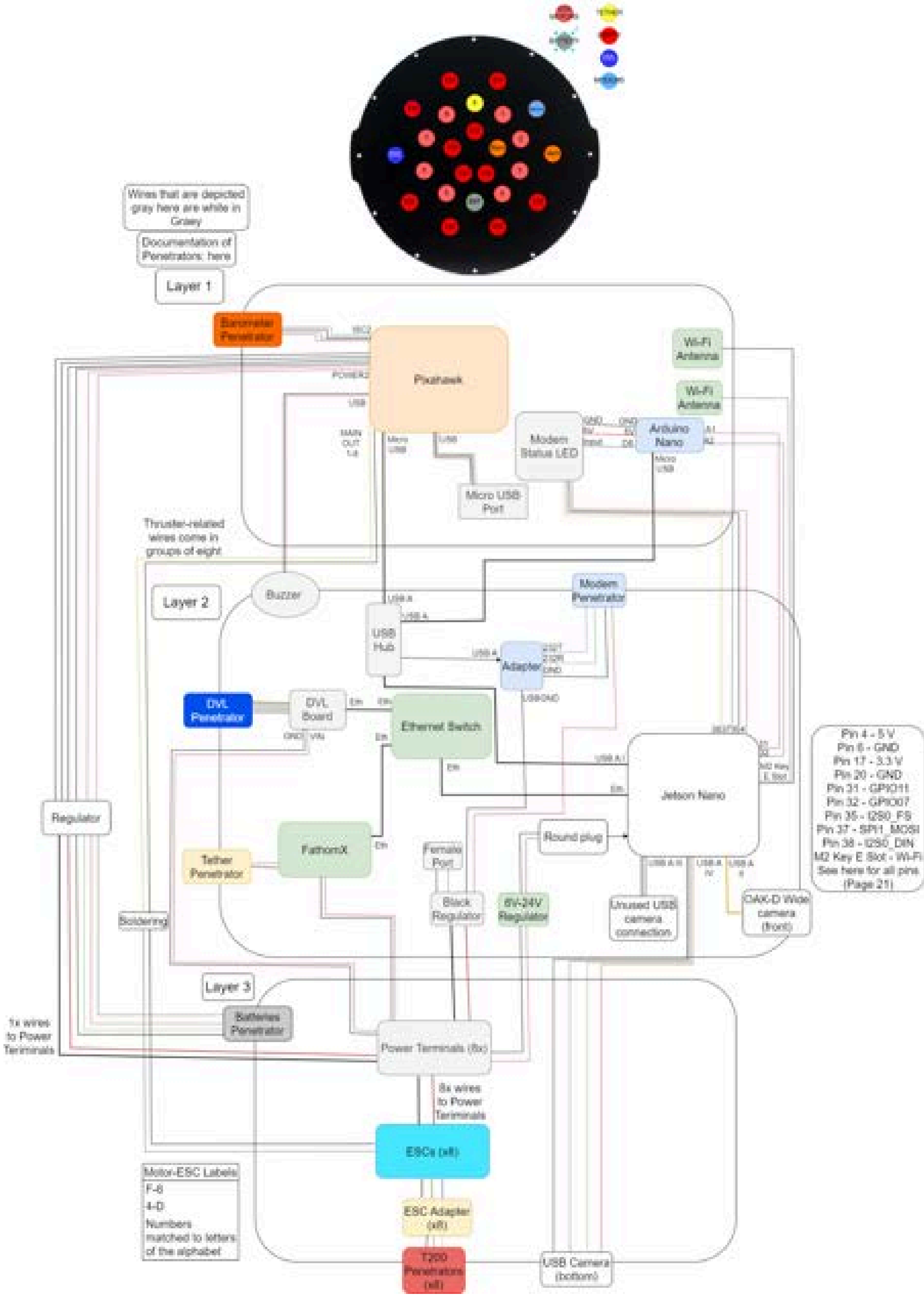


Fig. 11. Low Level Electrical Diagram for Græy.

APPENDIX D

WATER TESTING PROTOCOL FOR Græy and ONYX

With the complexity of ensuring our AUVs were watertight before water testing, we found that using a checklist to make sure all relevant systems were checked was extremely helpful.

## Pre-testing Checklist for Græy

- [ ] Fully charge BlueRobotics 14.8 V, 15.6 Ah battery
- [ ] Put battery in AUV
- [ ] Take out tether, connect it to the correct penetrator on AUV
- [ ] Take micro USB cable, ethernet cable, ethernet to USB adapter (if necessary), and plug into tether/computer.
- [ ] SSH through tether connection ("ssh inspiration@192.168.2.2"). Using an Ubuntu version rather than Windows is not necessary but more convenient.
  NOTE: Make sure a connection has been made before vacuum testing
- [ ] Plug in DVL and Modem
- [ ] Vacuum test AUV
    a. Unscrew the OK caps of both the electronics and battery enclosure
    b. Plug in the vacuum tester tubes to the now open plugs on both the battery enclosure and the electronics enclosure
    c. Pump until the gauge reads 10 in. Hg [34 kPa] vacuum. Wait 15 mins.
- [ ] Take Cover off of DVL
- [ ] Final check of penetrators
- [ ] Place sub in water (note start time in dive log) here

- [ ] If you want to run code, create a roscore screen ("screen -S roscore") and run "roscore". Then navigate back to the original terminal (Ctrl + A then press d).
- [ ] Navigate to the auv folder ("cd auv").
- [ ] From here you can run any code in the robosub_2024 repository as a module (python3 -m <SCRIPT DIRECTORY>, ex "python3 -m auv.device.camsVersatile")

## Post Testing:

1. Take sub out of water (note end time in dive log) here

2. Turn off electronics (latch side of main enclosure tube)
3. Wash down propellers with hose; **do not disconnect tether as water will get into the exposed female connector**
4. Tilt out water (so it doesn't pool at the bottom of the feet)
5. Pat down sub and store
6. Remove batteries (storage if testing is not scheduled within 7 days, charge if testing occurs within 7 days)

Check for:

1. Extension cord has been rolled up
2. Nothing is left in the water
3. Nothing is left on the tables
4. Gate has been locked

## Pre-testing Checklist for Onyx

- [ ] Fully charge 2 BlueRobotics 14.8 V, 15.6 Ah battery
- [ ] Put batteries in AUV
- [ ] Take out tether, connect it to the correct penetrator on AUV
- [ ] Take micro USB cable, ethernet cable, ethernet to USB adapter (if necessary), and plug into tether/computer.
- [ ] SSH through tether connection:
    a. SSH through Wifi onto Onyx
    b. Run "sudo ifconfig eth0 192.168.2.3 netmask 255.255.255.0" in order to be able to ssh through tether.
    c. SSH through tether ("ssh jetson@192.168.2.3")
- [ ] Using an Ubuntu version rather than Windows is not necessary but more convenient.
  NOTE: Make sure a connection has been made before vacuum testing.
- [ ] Plug in DVL and Modem
- [ ] Vacuum test AUV
    a. Unscrew the OK caps of both the electronics and one of the battery enclosures.
    b. Plug in the vacuum tester tubes to the now open plugs on both the battery enclosure and the electronics enclosure
    c. Pump until the gauge reads 10 in. Hg [34 kPa] vacuum. Wait 10 mins.
    d. Screw in the OK caps of the previously open electronics and battery enclosures; unscrew the OK caps of the DVL and other battery enclosure and run the same process.
- [ ] Take Cover off of DVL
- [ ] Final check of penetrators

- [ ] Place sub in water (note start time in dive log) here
- [ ] If you want to run code, create a roscore screen (“screen -S roscore”) and run “roscore”. Then navigate back to the original terminal (Ctrl + A then press d).
- [ ] Navigate to the auv folder (“cd auv”).
- [ ] From here you can run any code in the robosub_2024 repository as a module (python3 -m <SCRIPT DIRECTORY>, ex “python3 -m auv.device.camsVersatile”)

## Post Testing:

1. Take sub out of water (note end time in dive log) here
2. Turn off electronics (latch side of main enclosure tube)
3. Wash down propellers with hose; **do not disconnect tether as water will get into the exposed female connector**
4. Tilt out water (so it doesn't pool at the bottom of the feet)
5. Pat down sub and store
6. Remove batteries (storage if testing is not scheduled within 7 days, charge if testing occurs within 7 days)

Check for:

1. Extension cord has been rolled up
2. Nothing is left in the water
3. Nothing is left on the tables
4. Gate has been locked

## APPENDIX E

## OUTREACH ACTIVITIES

As part of our motto, *To Learn, To Share, To Innovate, and To Inspire*, Team Inspiration constantly strives to expose the next generation to STEM, and in doing so, we have found that some of the best learning comes through teaching.

Within the local community, we actively mentor rookie FIRST robotics teams, teach robotics camps, showcase our robots at a variety of local events (including the San Diego SeaPerch competition), reach out to youngsters at science museums and all major San Diego STEM fairs. During the summer time, we conduct more intensive summer workshops in partnership with the UrbanLife ministries and the USD STEAM Academy (Fig. 12), extending our reach to underrepresented and underprivileged students as well.

Fig. 12. Team Inspiration outreach with the UrbanLife Ministries (left) and USD STEAM Academy (right).

Beyond our local community, we extend our knowledge to the broader, global community. Actively coaching teams in Benin, Togo, Paraguay, and Ecuador, our team members connect with coaches from abroad and work to create a moving robotics experience for students, despite the barrier that distance creates. This year, our team members were able to reach a wider audience and were recognized at a speakership opportunity with the Embassy of Eswatini.

Within the last year, our team members have surpassed a total of 300 hours spent on outreach and volunteering. Especially focusing on students who have had less resources and/or lack of access to high-quality instruction and mentorship, we work to share our experiences, teach students about engineering fundamentals, and inspire students to further pursue education, especially in STEM. (Fig. 13).

Fig. 13. Team Inspiration outreach at the San Diego SeaPerch competition in 2024.