Design Review of *Talos*: An Autonomous Underwater Vehicle by The Ohio State University's Underwater Robotics Team

Nathan Becker, Amber Dellacqua, Brach Knutson, Mitch Oinonen, Robert Pafford, Cole Tucker

***Abstract*—The Underwater Robotics Team at The Ohio State University dedicated the 2022-2023 academic year to enhancing the reliability and performance of their competitive Autonomous Underwater Vehicle (AUV), Talos, for RoboSub 2023. This was achieved through the improvement of rigidity in chassis design and novel design in the torpedo launcher and marker dropper mechanism, the redesign of all Printed Circuit Boards (PCBs) to enhance troubleshooting, and the addition of smart battery modules. Moreover, the team streamlined their internal software tools, creating an interactive GUI that visualizes AUV path planning, status updates, and aids navigation and control. These comprehensive advancements enabled Talos to successfully complete prequalification for RoboSub 2023 and gain a competitive advantage for this year's autonomy challenge.**

I. Introduction

The Underwater Robotics Team (UWRT) at The Ohio State University is an interdisciplinary student project team that specializes in the construction, design, and operation of Autonomous Underwater Vehicles (AUVs). Comprised of students from various majors, the team assumes responsibility for all aspects of vehicle development.

For more than half of a decade, UWRT has actively been developing AUVs for RoboSub with the goal of increasing robustness of a vehicle capable of consistently performing at its peak capability. Building upon the knowledge gained from their previous competition vehicle, UWRT embarked on the development of their latest AUV, Talos, for RoboSub 2023. Talos boasts an enhanced mechanical design, a redesigned electronics system for troubleshooting, and streamlined software tools that feature an interactive GUI for efficient navigation and control. These advancements optimize its performance for the autonomy challenge in this year's RoboSub competition.

This report covers UWRT's competition approach and design strategy, highlighting the importance of testing and experimentation in optimizing Talos' performance for the RoboSub competition.

II. Competition Strategy

UWRT's competition strategy for RoboSub 2023 revolves around enhancing the reliability and performance of Talos. Through strategic design adjustments in mechanical, electrical, and software aspects, the team aims to optimize their course approach and task execution.

A. General Strategy: Reliability

Based on UWRT's performance at the 2022 RoboSub competition, the team adapted its strategy to focus on improving the reliability and ease of vehicle operation. The primary objective of these changes was to increase the minimum number of points obtained from each run during the autonomy challenge. To achieve this goal, the team set four specific criteria while planning and designing Talos for optimal success at the 2023 RoboSub competition:

- Software behavioral reliability and ease of operation
- Rigid chassis structure that maneuvers quickly and reliably
- Dependable task mechanisms compatible with software and electronics
- Stable electronics with the ability to troubleshoot and locate faults quickly

Implementing the criteria above, UWRT was able to craft an ideal course approach for this year's competition.

B. Competition Strategy: Course Approach

The four criteria set in UWRT's general strategy were paired with a list of parameters to determine the tasks to prioritize for Talos' autonomy runs:

1) The point values of the task last year
2) Cost to support the task in terms of technical ability, people, and pool testing time
3) Risk of damage to the vehicle if a failure occurs
4) Past experience with completing the task

Based on these criteria and the state of the team coming out of the academic year, UWRT has determined that it will commit to completing the Destination, Start Dialing, and Goa'uld Attack tasks, as well as surfacing in the octagon at the end of each run. Using this reduced course approach, UWRT hopes to dedicate more resources to completing its tasks consistently and increase its minimum score per run.

C. Competition Strategy: Task Execution

1) Gate - Destination: Talos will first complete the gate task, as it is a prerequisite for subsequent tasks in the course. To complete the task, Talos will dive underneath the water and attempt to locate the gate structure using the search algorithm outlined in Appendix B. Upon successful completion of the search algorithm, Talos will align itself with the center of the gate and determine the position of the "earth" symbol. Talos will then move underneath the earth symbol, spin for extra style points, and move on to the next task.

2) Buoys - Start Dialing: Given that the Buoy task has similar systematic requirements as the gate task, UWRT opted to prioritize it. Talos will search for the overall figure of the buoy using the same pattern outlined for the gate. Upon locating the buoy, Talos will move in front of it and determine the location of the two symbols it is going to touch. Then, it will align with and touch both symbols using a strategically designed poker to complete the task.

3) Torpedoes - Goa'uld Attack: After completing the buoy task, Talos will move on to the Torpedoes task, which UWRT selected to complete due to its high point value and low risk factor. Talos will begin by searching for the overall figure of the torpedo props. When the prop is found, Talos will move in front of it and determine the location of the holes in both images. Next, Talos will align the torpedo launcher with each hole and fire the torpedoes through them in a pre-determined order.

4) Bins – Location: UWRT does not plan on attempting the bins task due to its proximity to the bottom of the pool, and risk of damage to the AUV. Rather than dedicating resources to the development of this task, the team has opted to focus on the validation of the mechanisms and behaviors for the previous tasks. However, if UWRT re-evaluates this decision, Talos will attempt to complete the task by locating the overall figure of the bins, pushing the lids off with the AUV's chassis, and using the marker system to drop the markers into the bins.

5) Octagon - Engaging Chevrons & DHD : After the prioritized tasks are completed, Talos will move to the octagon task. Despite the projected high point value of the chevrons, UWRT has opted to only surface in the octagon to allow the team to spend more time developing reliable behaviors for the previous tasks. Talos will use the DHD table to locate the octagon, then move to the surface to complete the task.

III. Design Strategy

In order to align Talos with UWRT's competition strategy for RoboSub 2023, the team implemented innovative mechanical design features, streamlined electrical architecture, and created new software tooling. This section elaborates on the improvements of Talos' mechanical, electrical, and software subsystems.

A. Mechanical Subsystems

1) Chassis: UWRT decided to improve upon the previous AUV chassis by increasing the rigidity of its structure as well as enhancing Talos' maneuverability in the water.

The large high-density polyethylene sheets on the previous AUV were swapped for 6061 aluminum, arranged into triangular shapes connected together by steel cables as seen in **Fig. 1**. Along with the material change, implementing tall cross sections throughout Talos' chassis increased the area moment of inertia, therefore improving the rigidity. To further verify the structural integrity of the new chassis, the team conducted finite element analysis simulations as seen in **Fig. 2**. These changes reduced the weight of Talos by 2.7 kg and the width by 47 cm, making transportation easier.

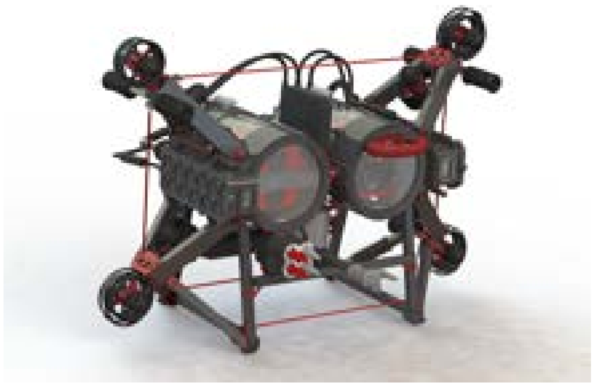


Fig. 1: Rendering of Talos

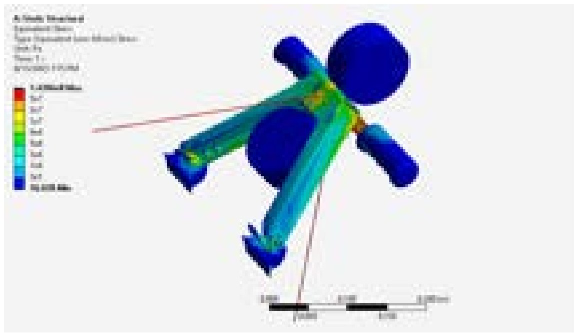


Fig. 2: Talos' upper triangular support FEA

Task mechanism placement was reworked to improve performance in a variety of ways. The modular task mechanism mounting system was integrated into the structural beam below the hull. This provided a centralized location that avoided thruster wash and placed task mechanisms in the camera's view.

Maneuverability also benefited from the new chassis design. Drag from the thruster cables was reduced by shortening the cable path and by fabricating a new lid for the cables to come out more streamlined to the vehicle. In addition, improvements to the center of mass and buoyancy locations made more orientations achievable. Adjustability in the center of mass location was made possible by shifting the batteries forwards and backwards on indexing rails. This new mounting system positions the batteries repeatably by snapping in with compliant clips, allowing for fast, toolless swapping. The AUV's center of bouyancy can be adjusted by moving buoyancy foam located in between the hull and on the bottom beam. These changes help allocate more thruster power towards maneuverability instead of counteracting drag, buoyant, and gravitational forces.

2) Torpedo and Marker Launcher: Along with chassis improvements, the team sought to improve the reliability and performance of the torpedo and marker launcher systems. The team decided to switch to large 3D printed torpedoes and markers for more inertia and added fins for extra stability. Additionally, the torpedoes and marker droppers were combined into the same system to reduce mechanical and electrical complexity. This was done because both tasks require the same basic action, and the vehicle can simply be pointed downwards to launch markers into the bin. To differentiate between the tasks, the torpedoes were colored scarlet, and the markers were colored gray as seen in **Fig. 3**.

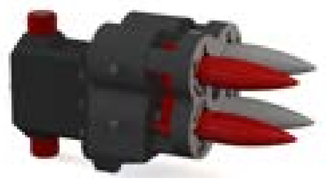


Fig. 3: Rendering of torpedo/marker droppers

The launch mechanism itself was also updated to avoid magnetic interference issues with the AUV's IMU. The new system avoids the use of magnets entirely, swapping the electromagnetic coils of the previous system for a spring launch mechanism actuated by a cam. A torpedo or marker can be inserted into the slot and rotated, which locks it in place due to the spring compression. When ready to fire, the cam rotates it back in the opposite direction, aligning the fins with the slots and launching it. The cam is driven by a Dynamixel servo, which is held in a waterproofed housing using a dynamic seal. This single servo also actuates both pairs of torpedoes and markers. The new system was tested and proved to be both reliable and accurate to 1.5 m, a tenfold increase compared to the previous torpedoes that tended to tumble shortly after firing.

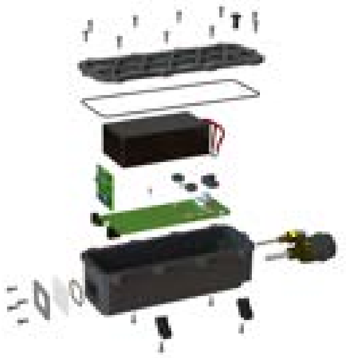


Fig. 4: Rendering of Talos' Smart Battery exploded assembly

3) Smart Battery Housings: The Smart Battery assembly consists of two elements, the mechanical housing and internal electronics as seen in **Fig. 4**. Each of the Smart Battery housings were CNC milled from aluminum blocks with iso-griding placed on each lid to reduce the total assembly weight by 0.68 kg. The lid is secured to a sealing face with an o-ring, generating a more reliable seal compared to gaskets. The battery hulls are also fitted with a pressure relief valve for safety and two SubConns for power and telemetry. Each battery housing features a window to view an informational display, allowing an operator to quickly determine the battery state.

B. Electrical Subsystems

The lessons learned throughout the design and operation of Mark 1 electronics became the basis for all Mark 2 design decisions, which focused on improving uptime. This year, the team made several large changes to the core electronics architecture. This involved distributing microcontrollers to each individual board, as well as modifying the physical layout of the electronics into a stack configuration. The team also developed electronics for the Smart Battery module, containing designs to improve insight into the vehicle's battery performance. Each board's microcontroller firmware additionally includes an improved core library to enhance stability and provide advanced telemetry.

1) The Mark 2 Electronics: Many of the design elements of Mark 2 carry over from the Mark 1 electronics. Both feature a Power Board, ESC Board, Camera Cage Breakout Board, and an Actuator Board seen in **Fig. 5**.

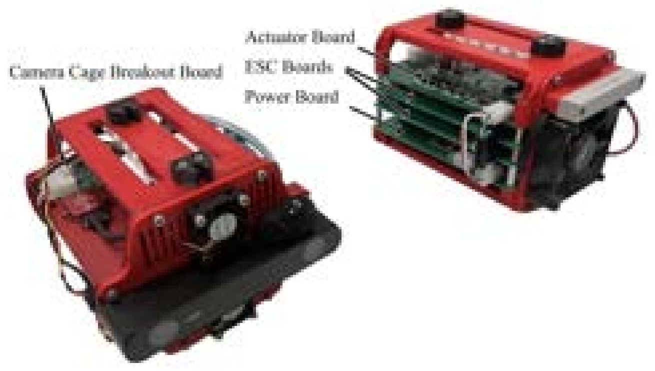


Fig. 5: Image of Mark 2 electronic cages in Talos

The Power Board is responsible for balancing battery power, creating the regulated 15V and 5V rails, providing a kill switch circuit to remove power from the thrusters, and sensors to monitor the voltage on the vehicle. The ESC Boards are carrier boards for the Electronic Speed Controllers (ESCs). The Camera Cage Breakout Board is responsible for distributing power and other signals throughout the Camera Cage, which houses the commercial off-the-shelf parts used for vehicle navigation and control. The Actuator Board handles all functionality to control the competition task mechanisms on the vehicle, including compatibility across three generations of actuator systems. This ensures that a failure of a task mechanism will allow the team to fall back to a previous task mechanism.

The largest design shift between the Mark 1 and Mark 2 electronics is the transition from a hub-based to a distributed control architecture as shown in Appendix C. The Mark 1 electronics forwarded all control signals between the electronics boards and the primary computer through a single microcontroller handling the communication. Mark 2 instead places a microcontroller block on every board, which are linked together using a Controller Area Network (CAN) bus. CAN bus introduces several features, such as improved signal integrity over longer distances, error detection, and multidrop communication with the primary vehicle computer.

This additionally reduces wiring in the vehicle, as now only power and two data lines travel between electronics boards.

The physical layout of the electronics system also changed with this generation, switching from a card-edge layout to a stack based layout, with each board mounted vertically on top of each other. This design was selected as it increases the available surface area for circuit layout by 118%, permitting more flexibility in the design. The increased area as well as a switch to 2 oz/ft^2 copper thickness on the PCBs allowed for higher power designs to meet the increased demands of the vehicle.

A large focus of the Mark 2 design was to increase the telemetry available from the electronics system. Part of improving uptime is ensuring that when failures occur, they can be quickly and easily traced to locate the failure. Mark 2 was designed with the expectation that system failures will occur at some point. A priority was set to provide as much information as possible via the software interface to determine and troubleshoot the cause of the failure, without needing to externally probe and test the system. The ESC board now implements the DShot protocol, providing real time RPM data from the thrusters to validate that they are spinning when commanded. The Actuator Board also provides feedback, communicating with the torpedo servo to ensure that it has rotated to the target position, and reporting if any hardware error events are raised by the servo.

2) Smart Battery Electronics: Talos now features two circuit boards in the battery housings to add protections and provide battery telemetry. The first board contains an integrated battery management system based on Texas Instrument's BQ40Z80 chip. This battery management system handles all cell balancing so that the batteries can be charged without being removed from the housings. It additionally includes several protection limits including undervoltage, overvoltage, overcurrent, and temperature protections, which will disconnect the battery in the event of any critical conditions.

The BQ40Z80 also monitors and reports approximate state of charge, allowing for more accurate reporting of battery capacity. The battery telemetry is then sent to the primary software system via an external CAN bus to be viewed by the operator. An OLED display and LEDs are also present, allowing for the batteries to show their state of charge. While disconnected from the vehicle or charger, the display can be triggered by a magnet to wake the battery. This allows the batteries to enter a low consumption state when not in use.

3) Firmware: A set of standard libraries called Titan Firmware was written to manage the increased number of microcontrollers on the vehicle. These libraries allow the majority of code to be shared across Mark 2, while still providing the flexibility to be tailored for each board's specific task. The central library, known as Titan Safety, provides real-time guarantees for execution, crash reporting, safe propagation of kill switch state, and profiling. This library ensures that all microcontrollers are operating in a safe and reliable manner and provides valuable telemetry to debug crashes while in operation. Furthermore, a bootloader was written for these microcontrollers, allowing for in-field upgrades without needing to open the hull. This bootloader, when combined with the crash reporting data, has been critical in troubleshooting issues found throughout firmware development.

C. Software Subsystems

This year, UWRT's software team focused on improving the usability of Talos' codebase. The team found at RoboSub 2022 that the previous AUV's software was difficult to manage for several reasons:

- AUV behavior was hard to debug with little visual feedback.
- At least 4 terminal shells were required to bring up the system and start an autonomous run.
- The Nvidia Jetson AGX Orin computer was difficult to set up and interact with.
- Competition objects were related to the world frame, and were difficult to locate if the pool was not aligned to magnetic north.

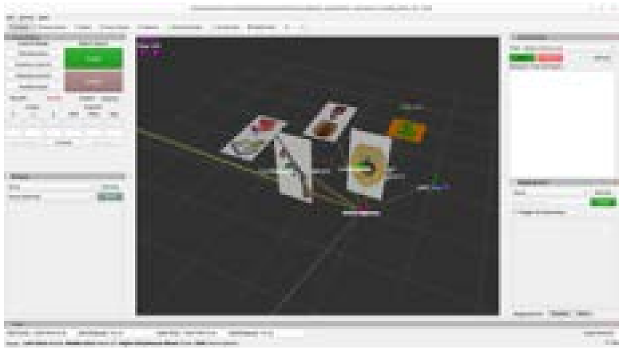


Fig. 6: UWRT's RViz layout used to operate the AUV

The software team addressed these user experience issues through the creation of a GUI system built into the ROS-based RViz platform. Shown in **Fig. 6**, these tools allow for visualization of the environment around Talos including path planning information, AUV status, and visualizations for the estimated positions of task objects. In addition to visualization, several plugins were developed for controlling the vehicle. One plugin allows the user to actively command the vehicle as well as control a software-based kill mechanism. The second plugin allows for remote startup of the software onboard the vehicle. It includes a staging mechanism to reduce unnecessary log spam and detect if the software was started properly. A third plugin allows controlling and monitoring the behavior tree autonomy system running onboard the vehicle for easy diagnostics and monitoring the task server's state during tethered runs. The final plugin can start, stop, and monitor ROS bags (data logs) running onboard the vehicle. Depictions of each plugin can be found in Appendix F.

UWRT reduced its number of Git repositories, streamlining the codebase and allowing for fewer systems to be improperly versioned. The team also introduced a "release" repository, which includes all the team's software repositories and drivers as Git submodules. This allows operators to tag and release specific versions of the codebase that have been tested in the pool. Using this system, operators can easily revert troublesome code to known working versions, allowing for faster isolation of issues.

To improve the setup of the Jetson, UWRT developed a cross-building system which allows users to build selected ROS2 packages in a containerized environment and automatically install them on the AUV. Building the packages also allows for the team to use ROS2 Humble on the Jetson, running Ubuntu 20.04 LTS. Once configured, changes between the operator computer and the Jetson can be easily pushed with a custom deploy extension for Colcon. This synchronizes selected packages from the operator computer to the Jetson and automatically builds them in one command.

The software team also focused on improving the robustness of its autonomy stack. The team created a fork of Groot, the behavior tree editor, and modified it to add a few features that UWRT saw to be desirable. These features included a save button, sharing of tree nodes in a workspace, and tree node port requirements. To allow better interaction with operators, autonomy software can display the status of the executing tree on four LED strips inside the hull. These LEDs display the AUV's status such as searching, success, or failure.

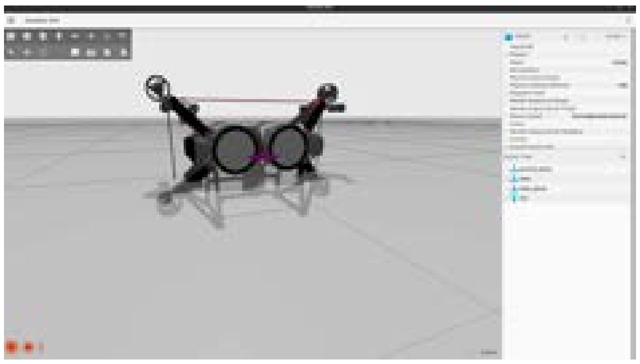


Fig. 7: Simulation of Talos in Gazebo Garden

To allow for testing of the AUV software systems in a native ROS2 environment, the team began development of a new simulation platform as seen in **Fig. 7**. A custom physics engine was developed using MathWorks' Simulink to provide realistic sensor feedback to the AUV. This engine computes the forces that act on the AUV including drag, buoyant, gravitational, and thrust forces in a discretized timestep. Gazebo Garden is used to display the results of the physics to the user, allowing for interaction with the vehicle during testing. A beta version of this simulation system has been tested and development is expected to finish next year.

To better track the competition objects in the pool, the team introduced a dedicated map reference frame. The frame can be set to any position and simplifies the configuration of the task objects in the water. This is accomplished with a custom AprilTag to allow for reliable determination of the map frame. The AUV uses vision to sample the location of the tag, monitor for deviation in the data, then re-calibrate the map frame to a known location on the tag. This allows Talos to better understand object locations in under 20 seconds, even if its position has drifted.

IV. Testing Strategy

UWRT emphasized the importance of testing and refining Talos for optimal performance and reliability through in-water, out of water, and simulation-based scenarios. Ensuring validation occurred within core components and objectives assisted the team in solidifying their competition and design strategy for RoboSub 2023.

A. *Prequalification*

The primary objective of UWRT's testing strategy was to successfully complete prequalification. The team aimed to optimize Talos' capabilities and increase its chances of meeting the qualification requirements before arriving at competition. This allowed the team to make modifications to their competition and design strategy if failure occurred within any of Talos' subsystems. For a description on UWRT's prequalification test procedure, see Appendix G, page 17. **Fig. 8** shows Talos recognizing task objects in RViz during a test run before performing prequalification untethered.

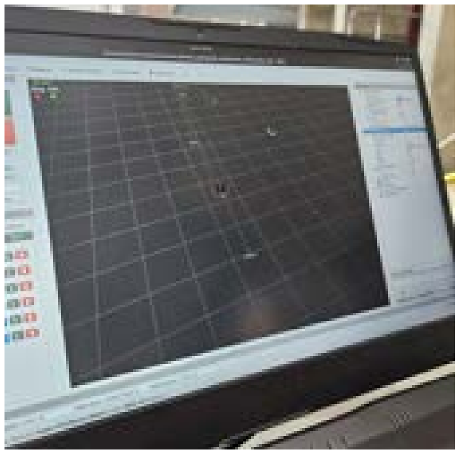


Fig. 8: Talos' prop recognition in RViz during its prequalifcation run

B. *Talos' Controller Calibration*

Pool testing time was limited this year for UWRT; therefore, higher level testing was prioritized. Focusing on tuning the vehicle's controller allowed the team to ensure Talos behaved in a stable and reliable manner. The team tuned the controller to be critically damped and stable in many scenarios. For a comprehensive description of the test procedure used by UWRT to fine-tune their controller during pool testing, see Appendix G, page 18.

C. *Smart Battery Electronics*

UWRT's bench testing of Talos' smart battery electronics, seen in **Fig. 9**, resulted in significant progress towards achieving stable electronics with efficient troubleshooting and fault localization capabilities. The real-time data obtained during testing provided valuable insights into the operational parameters of the batteries, contributing to a better understanding of

Talos' overall efficiency and performance. This achievement represents a notable advancement in UWRT's goal of creating dependable electronic systems, instilling greater confidence in the vehicle's performance. For a more detailed exploration of the test procedure and the findings on battery runtime, see Appendix G, page 20.

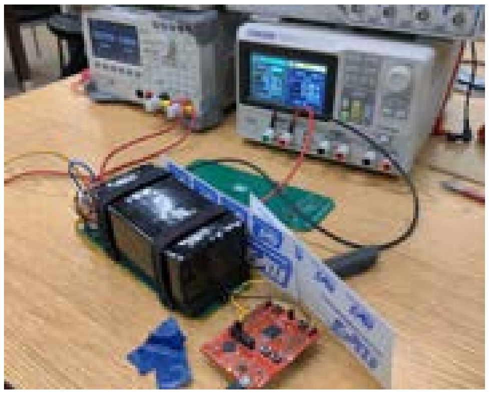


Fig. 9: Bench testing Talos' Smart Battery electronics

D. Torpedo Launcher

The reliability of the torpedo launcher installed on Talos was a key priority. The launcher needed to fire consistently without interfering with the vehicle's electronic systems. Preliminary testing of the torpedo launcher system is shown in **Fig. 10**. For a more detailed procedure outlining the test process conducted to ensure the torpedo launcher on Talos functions with a consistent firing distance, see Appendix G, page 22.

Fig. 10: Initial testing of Talos' torpedo launcher

V. Conclusion

UWRT's primary goal of reaching reliability in terms of AUV operation through advancements in mechanical, electrical, and software design is predicted to increase the quantity of points obtained from each run during the autonomy challenge. Optimizing the criteria discussed in the team's course approach in comparison to their previous year's competition strategy, UWRT was able to craft an ideal course run for their AUV, Talos. To achieve this, the team developed operator-friendly software, implemented a dynamic chassis design and task mechanisms that are more consistent, as well as constructed a more refined electronics system. With substantial testing of Talos' controller, torpedo launcher, as well as electronic systems, the vehicle was able to successfully complete prequalification and the team was able to solidify their competition strategy for this year's RoboSub competition.

In the future, UWRT hopes to expand on the development of inter-vehicle communication using acoustics, with their 2019 competition AUV (see Appendix E).

VI. Acknowledgements

The Underwater Robotics Team at The Ohio State University is located in Dreese Laboratories within Ohio State's Department of Electrical and Computer Engineering. UWRT would like to thank the staff members associated with the department and the building for their continued support and encouragement towards the team. The team is incredibly grateful for Dr. Saeedeh Ziaeefard, UWRT's advisor, for her guidance with the team's aspirations and achievements each year.

The progress made on UWRT's newest vehicle, Talos, and the continued progression of the team's STEMBot outreach program (see Appendix H) would not be possible without UWRT's dedicated team members. The team would also like to thank those who assisted with life-guarding, swimming, and organizing pool tests throughout the summer semester.

The team would also like to thank both its long-term and newly partnered sponsors such as Aptiv, Eaton, Ford, Ohio Space Grant, Caterpillar, and Honda. UWRT is also thankful for their non-monetary sponsors such as VectorNav and APD, who have graciously donated components that have further assisted the team in developing Talos for optimal performance.

Additionally, UWRT would like to thank RoboNation for committing themselves to maintaining RoboSub and gathering engineers from across the globe to collaborate and compete in a competition dedicated to underwater technologies. RoboSub has encouraged UWRT's team members to further dive into the field of robotics and explore concepts related to the competition.

VII. References

[1] Michele Colledanchise and Petter Ögren. Behavior trees in robotics and AI: an introduction. *CoRR*, abs/1709.00084, 2017.

[2] O-Ring Division. *Parker O-ring handbook*. Parker Hannifin, Parker Seal Group, 2007.

[3] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. February 2022.

[4] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.

[5] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022.

[6] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*. IEEE, Sep 2016.

VIII. Appendix A: Component Specifications

TABLE I: Talos' Component Specifications

Component	Vendor	Model/Type	Custom/Purchased	Cost	Purchase Year
Algorithms: Autonomy		BehaviorTree.CPP v3	Custom		
Algorithms: Localization/Mapping		Extended Kalman Filter/Custom mapping system	Custom		
Algorithms: Vision		YOLOv5	Custom		
AUV Chassis		Talos, Aluminum 6061	Custom	$900.00	2023
Battery	MaxAmps	Li-Po 8000 5S2P 18.5v	Purchased	$599.98	2023
Buoyancy Control	Blue Robotics	Subsea Buoyancy Foam; R-3312	Purchased	$70.00	2023
Camera	Stereolabs	Zed 2i	Purchased	$499.00	2022
Communication Network		CAN Bus	Custom		
Converter	TDK-Lambda	I6A4W020A033V	Purchased	$68.34	2022
CPU	Nvidia	Jetson AGX Orin	Purchased	$2,374.00	2022
Doppler Velocity Log (DVL)	Nortek	DVL1000	Purchased	$15,000.00	2018
Inertial Measurement Unit (IMU)/Compass	Vectornav	VN-100-T	Sponsored	$1,300.00	2022
Microcontrollers	Raspberry Pi	RP2040	Purchased	$0.07	2021
Motor Control	APD	80F3	Sponsored	$245.00	2022
Open Source Software		ROS2/OpenCV/Pico-SDK/BTCPP			
Programming Languages		C/C++/Python			
Smart Battery Housings		Aluminum 6061, Polycarbonate	Custom	$634.00	2023
Task Mechanism Servo	Robotis	DYNAMIXEL XL430-W250-T	Purchased	$49.90	2023
Thrusters	Blue Robotics	T200	Purchased	$1,432.00	2021
Waterproof Connectors	MacArtney	MC/HP/D Series	Purchased	$4,000.00	2015-2023
Waterproof Main Housing		Aluminum 6061, Polycarbonate	Custom	$2,233.00	2021

IX. Appendix B: Search Algorithm

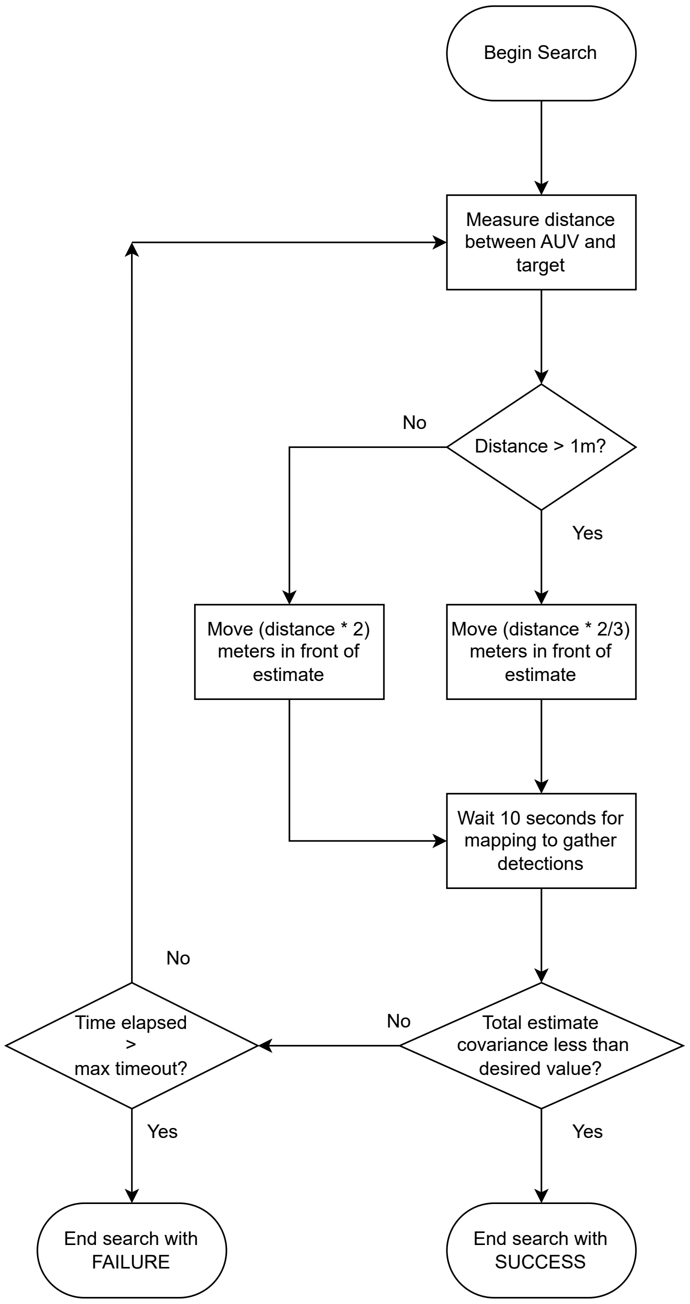

Begin Search
Measure distance between AUV and target
Distance > 1m?
No
Yes
Move (distance * 2) meters in front of estimate
Move (distance * 2/3) meters in front of estimate
Wait 10 seconds for mapping to gather detections
Total estimate covariance less than desired value?
No
Yes
Time elapsed > max timeout?
No
Yes
End search with FAILURE
End search with SUCCESS

Fig. 11: Talos' search algorithm to be used during competition runs

X. Appendix C: Mark 2 Architecture

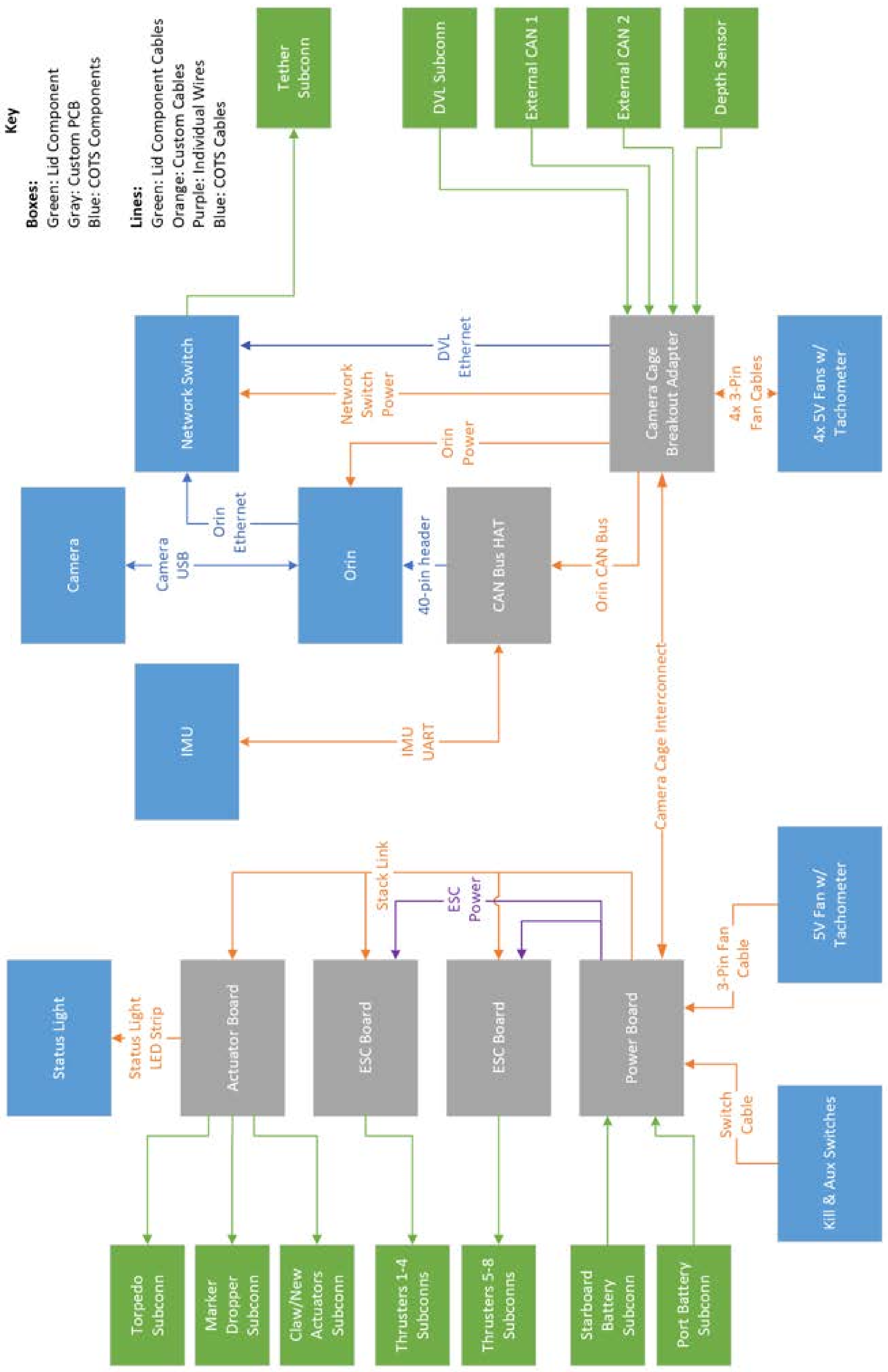


Fig. 12: Mark 2 architecture diagram

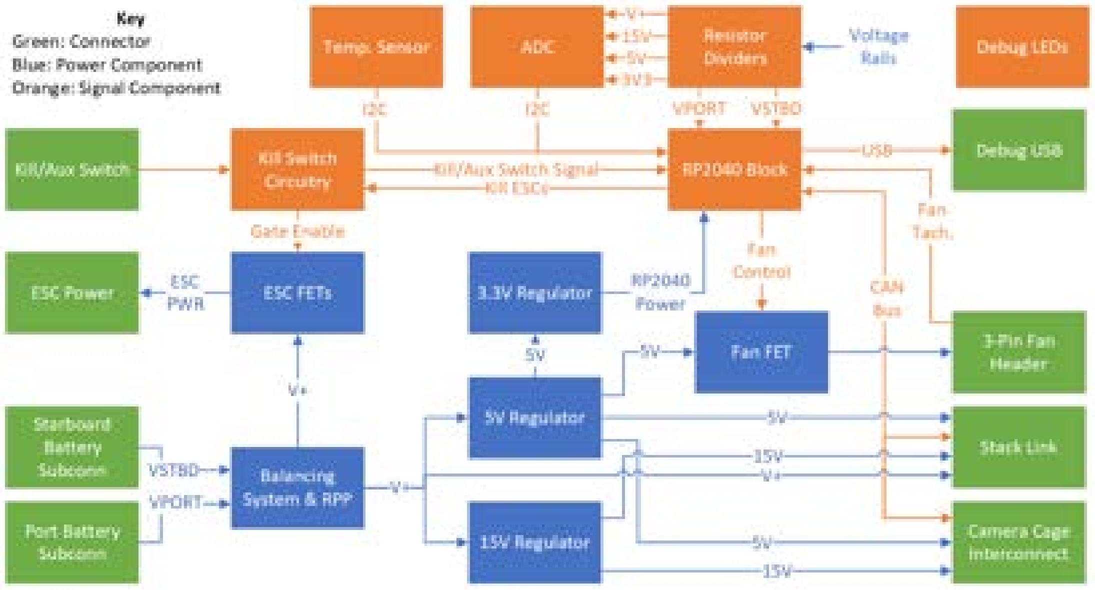


Fig. 13: Mark 2 Power Board diagram

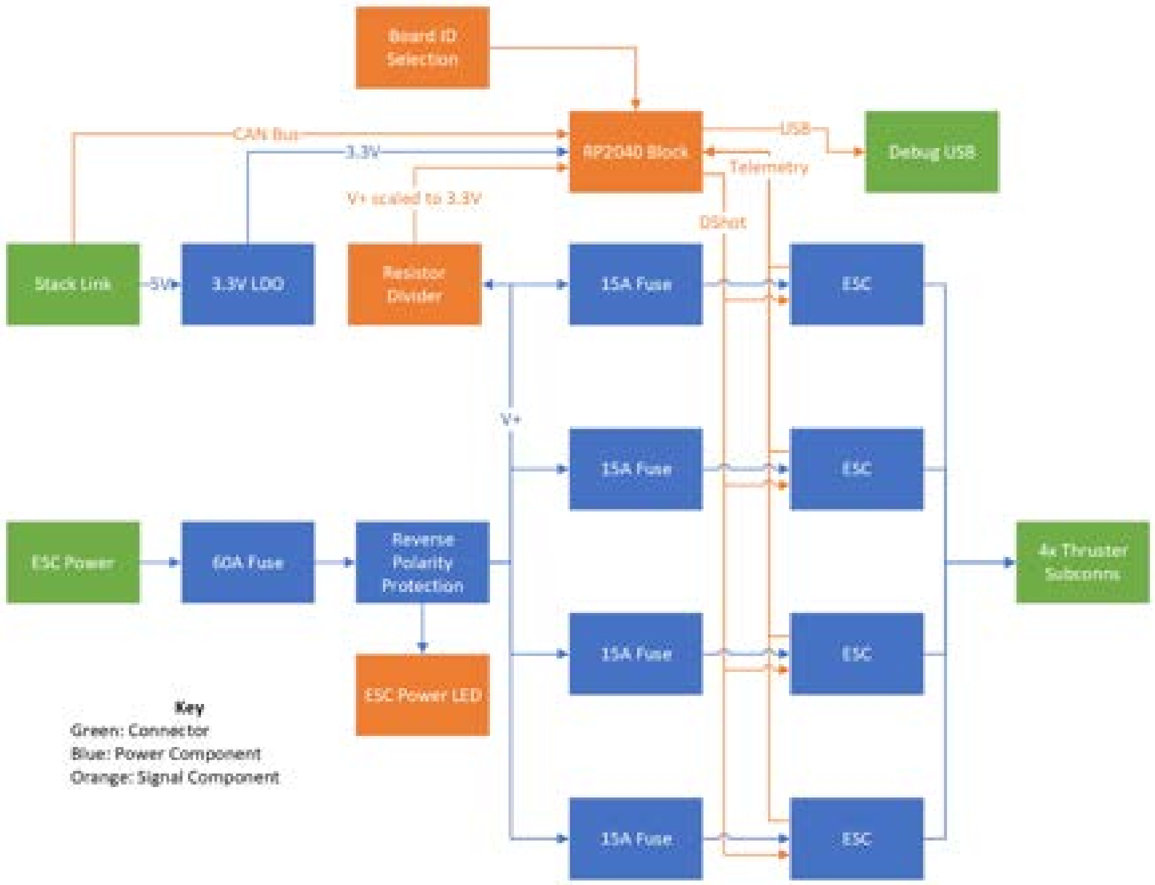


Fig. 14: Mark 2 ESC Board diagram

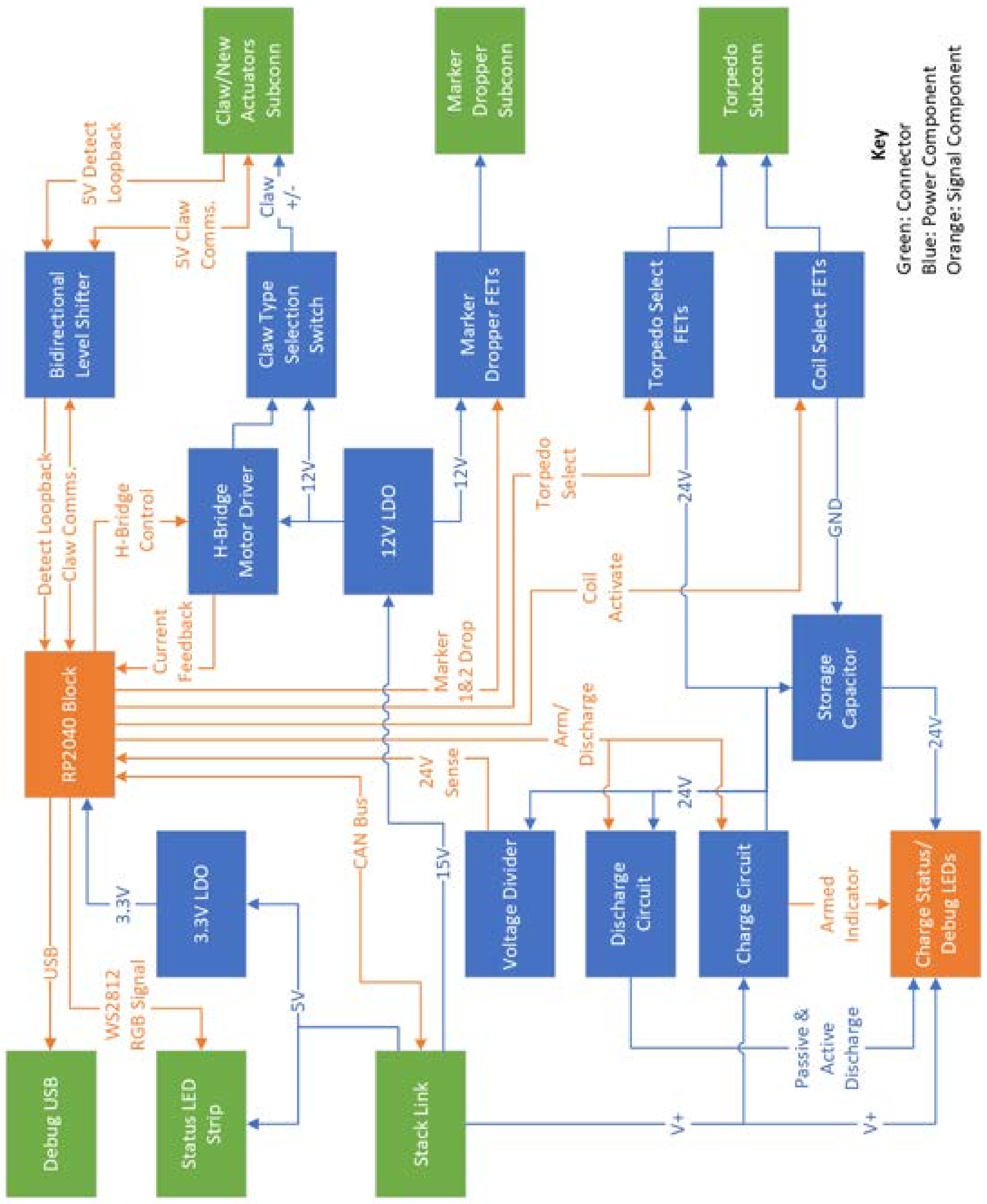


Fig. 15: Mark 2 Actuator Board diagram

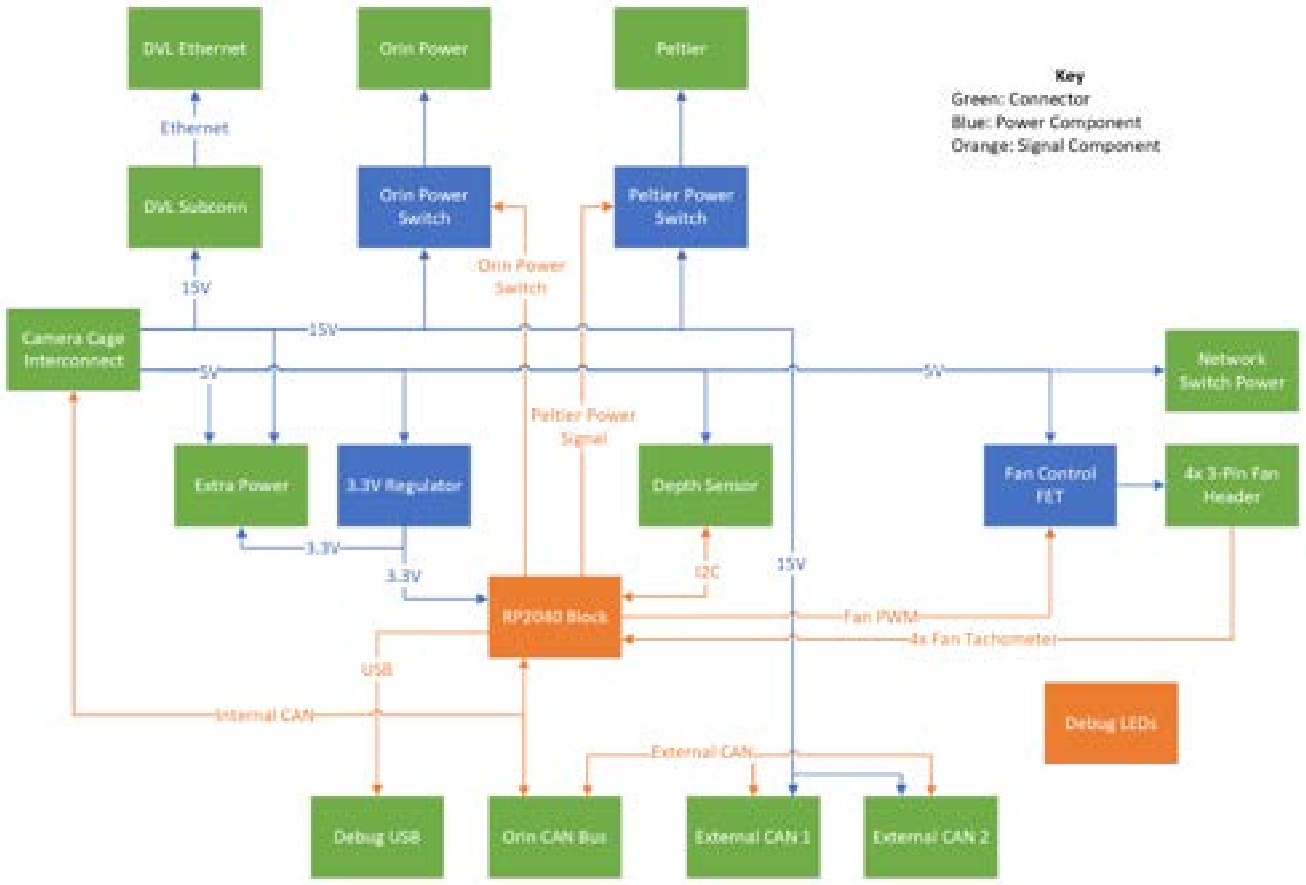


Fig. 16: Mark 2 Camera Cage Breakout Board diagram

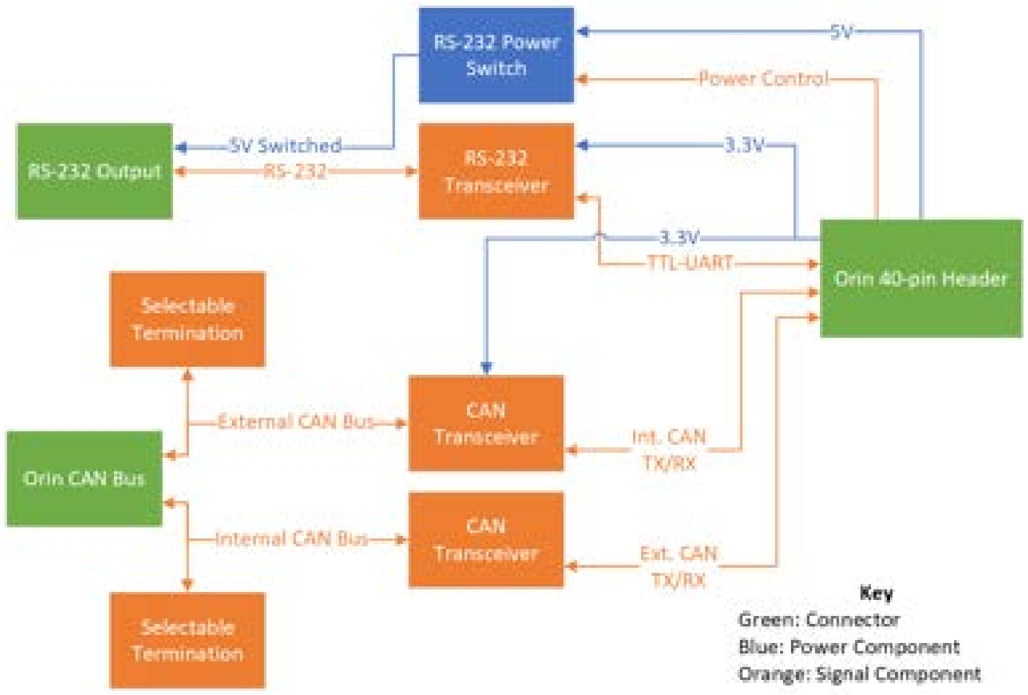


Fig. 17: Mark 2 CAN Bus Hat diagram

XI. Appendix D: riptide_software architecture diagram

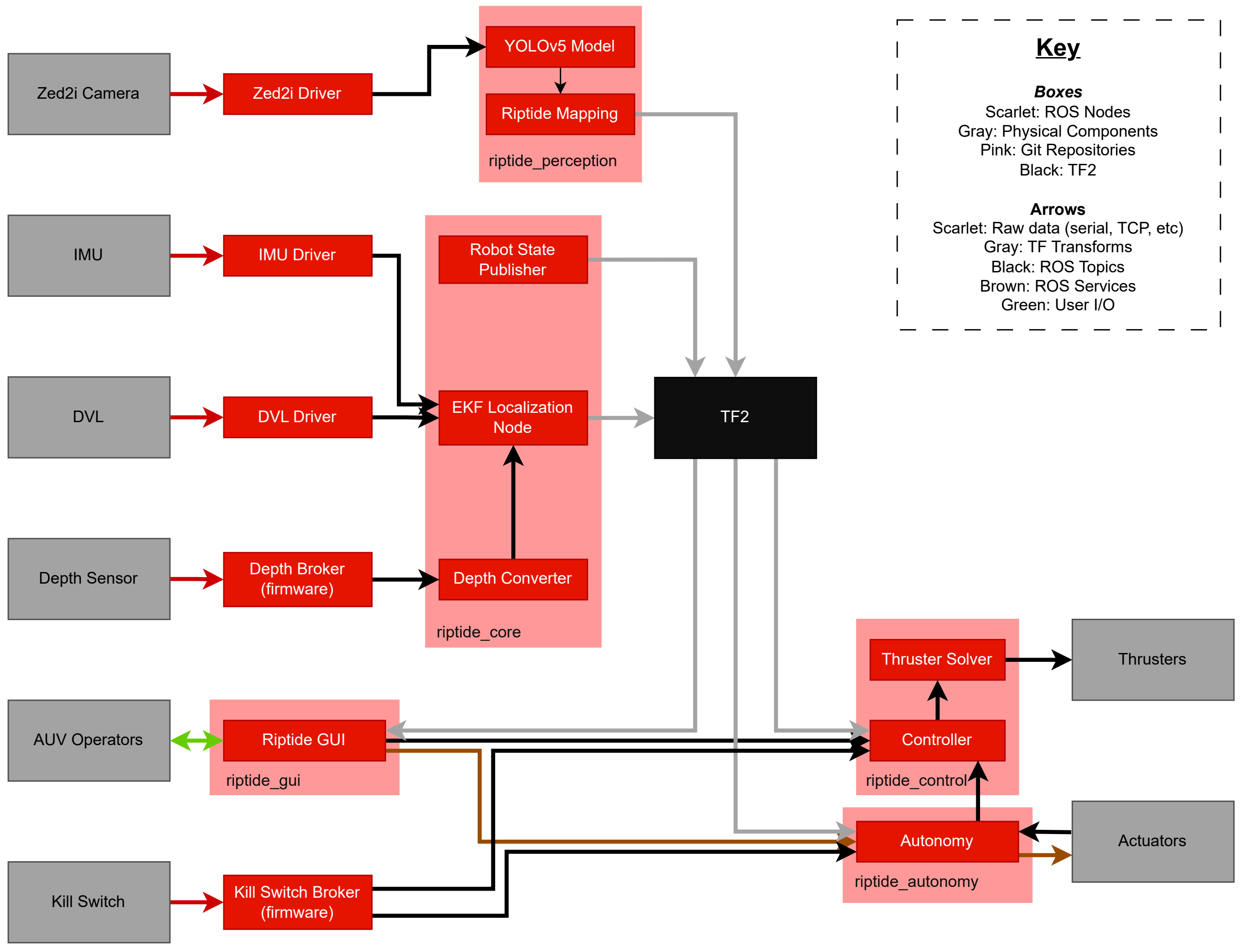


Fig. 18: Diagram of UWRT's software architecture

XII. Appendix E: Puddles Abstract

The Underwater Robotics Team at The Ohio State University produced their first AUV hull in 2016. In 2019, the team produced the AUV, Puddles using the same core hull design. In 2022, UWRT discontinued their use of the AUV and left it in a state of disrepair. The goal of this project was to restore the core of the vehicle and turn it back into a research tool capable of being operated by research teams to test new control algorithms, underwater communication, localization techniques, object recognition, and more. In order to get the AUV back to this operational state, new electronics needed to be designed and tested, new hull components fabricated, and new software written to interface and control the vehicle. The new AUV design would then be tested to prove it was ready to perform all of the possible research goals. During testing of the AUV, 11 of the 15 tests were considered passing while 4 were considered failures. Of the four that failed, each have root causes and solutions identified, allowing for Puddles to serve its new mission, underwater research in all forms.

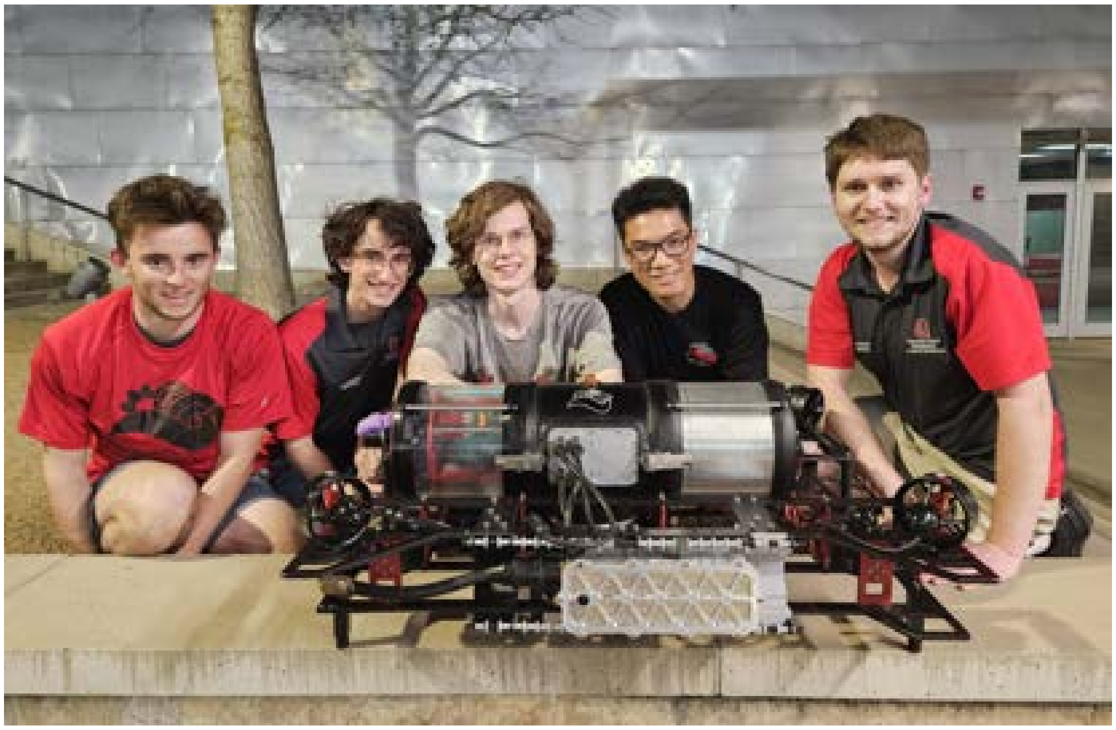
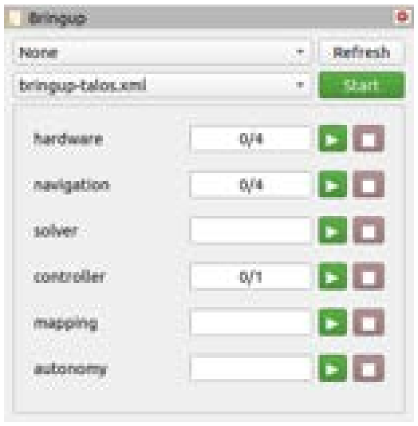


Fig. 19: The team working to restore puddles back to working condition. From left to right: Alex Schuler, Hunter Seachrist, Robert Pafford, Mark Fong, Cole Tucker

XIII. Appendix F: RViz Panels



None
bringup-talos.xml

hardware	0/4
navigation	0/4
solver	
controller	0/1
mapping	
autonomy	

Fig. 20: The RViz bringup panel is used for remotely starting the robot. This panel allows for staging and monitoring startup of the software.

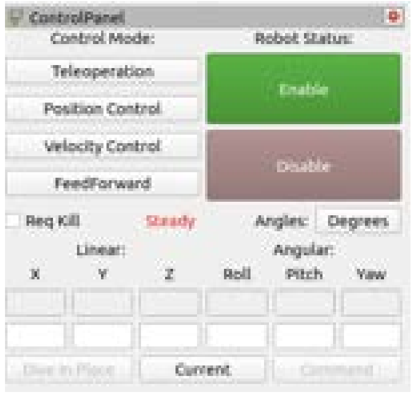
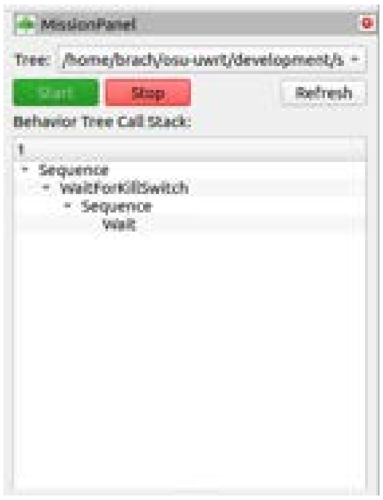


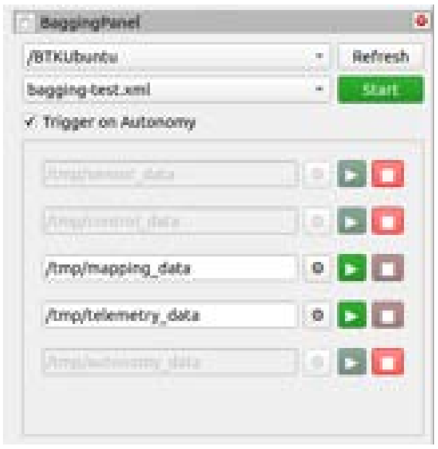
Fig. 21: The RViz control panel is used for commanding the vehicle while in the water. It allows for sending commands directly to the robot and reading out its current estimated location.



Tree: /home/brach/osu-uwrt/development/s

```
1
▾ Sequence
  ▾ WaitForKillSwitch
    ▾ Sequence
        Wait
```

Fig. 22: The RViz mission panel is used for commanding the behavioral system onboard the AUV. It is able to manage the executing tree as well as see the current stack trace of the tree to allow for ease of debugging.



/BTKUbuntu
bagging-test.xml
✓ Trigger on Autonomy

/tmp/sensor_data
/tmp/control_data
/tmp/mapping_data
/tmp/telemetry_data
/tmp/autonomy_data

Fig. 23: The RViz bagging panel is responsible for configuring, starting, and stopping logging onboard the AUV during runs. It also can synchronize with the mission system to allow for automatic starting and stopping of logs when behaviors are executing.

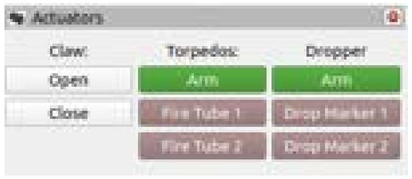


Fig. 24: The RViz actuators panel allows an operator to have direct control over the task mechanisms onboard the AUV. It also includes safety interlocks required for operation of critical systems like the torpedoes to prevent accidental discharge.

XIV. Appendix G: Test Plan

A. *Test: Pre-Qualification*

1) Required Equipment:

- Talos AUV
- AUV ethernet tether
- UWRT underwater AprilTag
- 2 Smart Battery assemblies
- Pole marker
- Qualification gate
- UWRT Safety Stack or network router
- Operator laptop with UWRT software installed
- A pool with minimum size of 35m x 7m x 2m (L x W x D)

2) Steps:

1) Connect Smart Battery assemblies to the AUV.
2) Connect ethernet tether to AUV and to UWRT Safety Stack.
3) Connect operator laptop to UWRT Safety Stack and power the laptop on.
4) Launch UWRT's RViz configuration for the Talos AUV.
5) Place the UWRT AprilTag into the water near the edge of the pool and place a weight on top to secure.
6) Place the AUV in the pool near the AprilTag.
7) Face the AUV towards the AprilTag and ensure the main body of the tag falls within the camera view.
8) Run the map frame calibration command and allow 10 to 20 seconds to update the map frame location.
9) In the RViz MissionPanel pane, select one of the PreQualTree.xml files.
10) Press ”Start”, underneath the tree selection dropdown.
11) Press ”Enable” in the RViz ControlPanel pane.
12) Have the swimmers remove the tether from the vehicle, and allow the operator to pull it clear.
13) Have the swimmers insert the kill switch and release the vehicle without interfering with the DVL.
14) Allow the robot to complete the run according to the RoboSub specification.

3) Pass / Fail Criterion:

1) The test is considered passing if the AUV dives below the water, passes through the gate, navigates around the pole, passes back through the gate, and surfaces.
2) The test is considered a failure if the AUV misses the gate or pole or was otherwise unable to navigate around the objects. Correct the positions of the gate and pole in the mapping config and, if necessary, re-calibrate the map frame with the apriltag.

4) Results: This test was executed at UWRT's 6/6/23 pool test. The AUV was at first unable to complete the procedure due to drift in the physical objects. After better anchoring the objects, the AUV was able to successfully complete the procedure by submerging and navigating through the gate, around the pole, and back through the gate.

B. Test: Controller Calibration

1) Required Equipment:

- Talos AUV
- AUV ethernet tether
- 2 Smart Battery assemblies
- UWRT Safety Stack or network router
- Operator laptop with UWRT software installed
- A pool with minimum size of 10m x 10m x 4m (L x W x D)

2) Steps:

1) Connect Smart Battery assemblies to the AUV.
2) Connect ethernet tether to AUV and to UWRT Safety Stack or network router.
3) Connect operator laptop to UWRT Safety Stack or network router and power the laptop on.
4) Launch UWRT's RViz configuration for the Talos AUV.
5) Launch RQT GUI and start the Dynamic Reconfigure plugin.
6) Launch a text editing program to edit the AUV's description file.
7) Have the swimmers insert the kill switch and release the vehicle without interfering with the DVL.
8) In RViz, place the AUV into the feed-forward control mode by pressing ”Enable”, then ”Feed Forward”, then ”Command”.
9) Observe robot behavior in feed-forward mode and adjust the AUV description's center of buoyancy value as necessary.
10) Disable the robot by pressing ”Disable” in RViz and having the swimmers remove the kill switch.
11) Repeat steps 8-10 until the robot stays mostly stationary while in feed-forward mode.
12) Use Dynamic Reconfigure to set the controller velocity P-gains to 0.
13) Choose an un-tuned linear axis to tune (X, Y, or Z) and set it's velocity P-gain to a positive value.
14) Configure the PlotJuggler program to plot the current and desired AUV velocity along the chosen axis in the same plot space.
15) Set the AUV into a zero-velocity control loop by pressing ”Enable” in RViz, populating all control fields with ”0”, and pressing ”Command”.
16) With the AUV in a zero-velocity control loop, have the swimmers push the robot in any direction within the axis being tuned.
17) Observe the velocity plot and use Dynamic Reconfigure to adjust the p-gain as necessary.
18) Repeat steps 16-17 until the AUV responds to the impulse by quickly returning to zero velocity along the chosen axis with minimal overshoot.
19) Repeat steps 12-19 until all linear axes are tuned.
20) Repeat steps 12-20, tuning the angular axes rather than the linear ones.
21) Set all velocity p-gains to their tuned values.
22) Using Dynamic Reconfigure, set all position p-gains to 0.
23) Choose an un-tuned linear axis to turn (X, Y, or Z) and set its position p-gain to a positive value.
24) Configure the PlotJuggler program to plot the current and desired AUV position along the chosen axis in the same plot space.
25) In RViz, set the desired position of the AUV to some position approximately 1 meter away in the chosen axis.
26) Set the AUV into position mode by pressing ”Enable”, ”Position Control”, then ”Command”, while the divers push the AUV underwater.
27) Observe the position plot and use Dynamic Reconfigure to adjust the p-gain as necessary.
28) Repeat steps 25-27 until the AUV is able to quickly achieve the commanded position along the chosen axis with minimal overshoot.
29) Repeat steps 23-28 until all linear axes are tuned.
30) Repeat steps 23-29, tuning the angular axes rather than the linear ones.

3) Pass / Fail Criterion:

1) If the steps are completed and the AUV is able to quickly achieve and reliably hold commanded poses, the test has passed.
2) If the AUV is unable to achieve and hold commanded poses in a stable manner, the test fails. Ensure the AUV description is correct, and adjust the p-values, drag coefficients, maximum velocities, and maximum accelerations as necessary until the controller is stable.

4) Results: This test was executed across three of UWRT's pool tests in the April and May months. After tuning the P-gains initially, UWRT found that Talos was unable to hold an upright position, and tended to have five to ten degrees of error along the pitch and roll axes. This issue was corrected by increasing the maximum allowed velocity and acceleration along the angular axes, and adjusting the pitch p-values as necessary. The controller has since displayed stable control over the AUV's pose in the water.

C. Test: Smart Battery Electronics

1) Required Equipment:

- Assembled Smart Battery Housing Battery Management Board (BMB) using a Texas Instruments BQ40Z80 Battery Pack Management Chip
- Maxamps LiPo 8000 5S2P 18.5v battery pack
- Texas Instruments EV2400 PC Fuel Gauge Interface Module, or equivalent interface
- 2 2-ch Isolated Variable DC Benchtop Power Supplies
- 1 1-ch Isolated Variable DC Benchtop Power Supply
- Calibrated Programmable DC Electronic Load
- Calibrated Voltmeter
- Test Leads
- Windows PC with Texas Instruments bqStudio installed

2) Steps:

1) Connect EV2400 to PC and launch bqStudio. Select the BQ40Z80 chip under the list of devices.
2) Power up the BMB with DC power supplies.
   a) Ensure nothing else is connected to the BMB (except the thermistor).
   b) Ensure that SB3 and SB1 are not populated on the PCB.
   c) Connect the EV2400 to the unpowered BMB via the LCD Board Connector.
   d) Set each of the variable power supplies voltage to 3.7V with a current limit of 250 mA, without turning on the output.
   e) Connect the variable power supplies together in series.
   f) Connect the negative-most power supply's negative lead to the BAT- terminal block on the BMB.
   g) Connect the positive-most power supply's positive lead to the BAT+ terminal block on the BMB.
   h) Connect each of the intermediate power supply's positive lead to the corresponding balance pin on the BMB, with the negative-most power supply's positive lead corresponding to cell 1.
   i) Enable the output on each of the power supplies sequentially. The negative most power supply should be enabled first, with the positive-most power supply enabled last.
   j) Wake the BQ40Z80 out of shutdown by either, a) briefly connecting the BAT+ to the PACK+ terminal through a 1k ohm resistor, or b) briefly connecting the Wake pin of the LCD connector to 3.3V.
   k) Ensure that the BQ40Z80 is detected in bqStudio on the PC.
3) Flash the golden BQ40Z80 image file, containing the correct configuration data for this battery application through bqStudio.
4) Calibrate the BMB through bqStudio to compensate for any tolerance in the board's components. This follows procedures in Texas Instrument Document SLUUBZ5: "bq40z80EVM Li-Ion Battery Pack Manager Evaluation Module".
   a) Perform procedure listed in SLUUBZ5 Section 3.3.1 to calibrate the Voltage Sense components of the BQ40Z80. Ensure that a calibrated voltmeter accurate to at least 3 decimals is used for the reference measurement.
   b) Increase the current limit of the DC power supplies to above 2 A.
   c) Connect the positive terminal of the programmable load to the BAT+ terminal block.
   d) Connect the negative terminal of the programmable load to the PACK- terminal block.
   e) Perform procedure listed in SLUUBZ5 Section 3.3.3 to calibrate the Current Sense components of the BQ40Z80. Ensure the programmable load is accurate to at least 3 decimals.
5) Verify the BMB is able to safely operate and support current loads prior to connecting the LiPo battery.
   a) Check the status flags on the BQ40Z80 from bqStudio to ensure that no faults are present, charge/discharge is not disabled, and the chip is not in a test override mode.
   b) Measure the voltage across the PACK+ and PACK- pins. Ensure that there is no voltage present across those nodes.
   c) Connect the DETECT pin to PACK- on the 6-pin external charge connector.
   d) Measure the voltage across the PACK+ and PACK- pins. Ensure that 18.5V is present across those nodes.
   e) Remove the DETECT to PACK- jumper. Connect the programmable load across the PACK+ and PACK- terminal blocks.
   f) Configure the programmable DC power supplies to 10A, or their maximum output current. Note this should be at least above 3A to fully validate the BMB.
   g) Connect the DETECT to PACK- jumper.
   h) Apply slightly under the maximum current set on the DC power supplies on the programmable load.
   i) Ensure that the current is properly reported in bqStudio. Note that due to the current scaling configuration set, this value will be the actual current / 4.

j) Remove the DC load.
k) Verify that the protection mechanism functions by disconnecting a balance cell lead from a programmable DC power supply.
l) Ensure that the BQ40Z80 status bits report an undercell voltage fault and that the pack output has turned off.
m) Ensure that no voltage is present across the PACK+ and PACK- pins, even though the detect jumper is still present.

6) Connect a LiPo cell to the BMB.

a) Remove the test equipment from connected in the previous steps.
b) Ensure that the BMB is adequately insulated on the bottom surface with Kapton tape, a foam cutout is on the top of the BMB to evenly distribute battery weight, and that the 3D printed terminal guards are placed on the PACK+, PACK-, BAT+, and BAT- terminals.
c) Kapton tape the battery thermistor to the underside of the cell facing the foam.
d) First, connect the balance terminal of the battery to the pack.
e) Then, screw the positive terminal of the battery pack, followed by the negative terminal to the BAT+ and BAT- terminals, respectively.
f) Plug the thermistor into the BMB.
g) Connect the EV2400 to the BMB.
h) Wake the BQ40Z80 using methods described previously.
i) Ensure that the BQ40Z80 is recognized by bqStudio, and no faults are present.

7) Perform initial battery gauge learning cycle on the LiPo pack. This is not necessary if the pack is new, as the golden image contains the correct learning data for a fresh MaxAmps LiPo 5S 8000 battery pack. However, if installing a previously used battery pack, this step must be run. This follows procedures in Texas Instruments Document SLUA848 – "How to Complete a Successful Learning Cycle for the bq40z80".

a) Following procedure in SLUA848 Section 4.2.1, discharge the battery to empty using the programmable load, connecting the load to PACK+ and PACK-, with the DETECT jumper in place.
b) Following procedure in SLUA848 Section 4.2.2, let cell relax for 5 hours.
c) Following procedure in SLUA848 Section 4.2.3, charge battery to full.
d) Following procedure in SLUA848 Section 4.2.4, let cell relax for 2 hours.
e) Following procedure in SLUA848 Section 4.2.5, discharge battery to empty.
f) Following procedure in SLUA848 Section 4.2.6, let cell relax for 5 hours.
g) If the flags match the state as specified in SLUA848, the pack has been successfully learned.
h) Charge the battery pack back to storage voltage .

8) Perform final load test on battery pack.

a) Apply a load of around 5A to the battery PACK+ and PACK- terminals.
b) Ensure that the battery properly delivers the load without raising faults on the BQ40Z80.

3) Pass / Fail Criterion:

- Pass: The BQ40Z80 did not raise faults during the bringup, and is able to successfully deliver the 5A load during step 8.
- Fail: The BQ40Z80 raised a fault during bringup, cell learning failed, or cell was unable to deliver the 5/item A load during step 8.

4) Results: This test was fully performed on the first smart battery housing assembled, with the second smart battery partially following the procedure omitting step 7. During the bring up of the first smart battery housing, a few bad solder joints were found due to faults being raised on the BQ40Z80 chip in step 5a. However, after addressing this problem, the battery pack passed the test above. The second pack additionally passed the test above without issue. These two packs have since been the primary batteries for Talos. With this increased insight into the battery state of charge, the smart battery housings were able to successfully run a 6 hour pool test on a single charge.

D. *Test: Torpedo & Marker Function*

1) *Required Equipment:*

- Torpedo and marker launcher assembly
- Talos AUV
- Tape measure
- Underwater camera
- 2 Torpedoes
- 2 Markers

2) *Steps:*

1) Command Talos to hold position at a depth of 1m.
2) Orient the servo to the topmost position.
3) Load both pairs of torpedoes and markers.
4) Hold up the tape measure from the tip of the torpedoes outwards from the robot's front face.
5) Launch all 4 torpedoes/markers and observe the flight path with the underwater camera.
6) Record the distance travelled by each torpedo/marker before tumbling.
7) Repeat steps 2-6, launching 16 torpedoes/markers in total.

3) *Pass / Fail Criterion:*

1) Pass: If all torpedoes and markers travel a minimum distance of 1 meter, the system passes.
2) Fail: If any torpedoes or markers travel under a distance of 1 meter, the system fails. Adjust the spring compression using spacers to increase the distance travelled until the system passes.

4) Results: Because of promising test results from smaller, in-lab experiments, pool time was prioritized on software calibration. As such, only 8 torpedoes/markers were launched and recorded at a pool test. The average distance travelled was 1.52 meters, with none travelling less than 1 meter. Therefore, for the amount of torpedoes launched, the system passed.

E. Test: Watertight Verification

1) Required Equipment:

- Assembly being tested
- Water tub
- 5-gallon bucket
- Pump
- Hose
- Smartphone with timer
- Pool towel
- Paper towels
- 5lb weights
- Torque wrench

2) Steps:

1) Remove any critical components in the assembly, such as electronics.
2) Place paper towels inside the assembly near each sealing surface.
3) Use the torque wrench to torque all sealing screws to their required amount.
4) Position tub near the spigot, and fill with the hose. While it is filling, adjust the water to a mild temperature.
5) Turn off once the water level is high enough to completely submerge the assembly.
6) Submerge the assembly completely in the tub, and look for bubbles near the seals. If excessive bubbling does occur, proceed to Step 8.
7) Place the 5lb weight on the assembly to hold it underwater, and set a timer for 10 minutes.
8) Once the time has past, remove the assembly from the tub without rotating it, and dry the outside completely with the pool towel.
9) Open the tested assembly to inspect interior and paper towels. Closely inspect all sealing surfaces.
10) Once all testing is complete, empty the tub into the drain basin using the hose, pump, and 5-gallon bucket.

3) Pass / Fail Criterion:

1) Pass: If there are no signs of water anywhere in the interior, the test passes.
2) Fail: If there is any water inside the assembly, the test fails. Determine the point at which the assembly leaked, and determine if a replacement is needed for the seal, or if a redesign of the assembly is necessary.

4) Results: Before any pool test, the main hull and the smart battery housings are verified to be watertight. So far, the hull and smart battery housings passed each test and have not leaked. However, one actuator housing failed during a pool test, which short-circuited the servo. It was revealed that this was caused by assembly error and has been corrected with a design change to prevent this failure from occurring again.

XV. Appendix H: Outreach Activities

UWRT's STEM outreach initiative and goal of educating others about the field of underwater robotics expands from Ohio State's campus to the greater Columbus area. The team engages youth in the community through a five-week after-school program called STEMBot. UWRT returned to Rosemore Middle School with an improved STEMBot program that allowed middle school students to gain more hands-on experience with developing underwater robots. Students had the opportunity to assemble a STEMBot both in TinkerCAD, a kid-friendly CAD software, as well as its physical form both electrically and mechanically. Students also learned the basics of programming and were able to compete with their robots against each other in an underwater obstacle course. All the students enjoyed the after-school program and constantly asked if UWRT would be returning the following year!

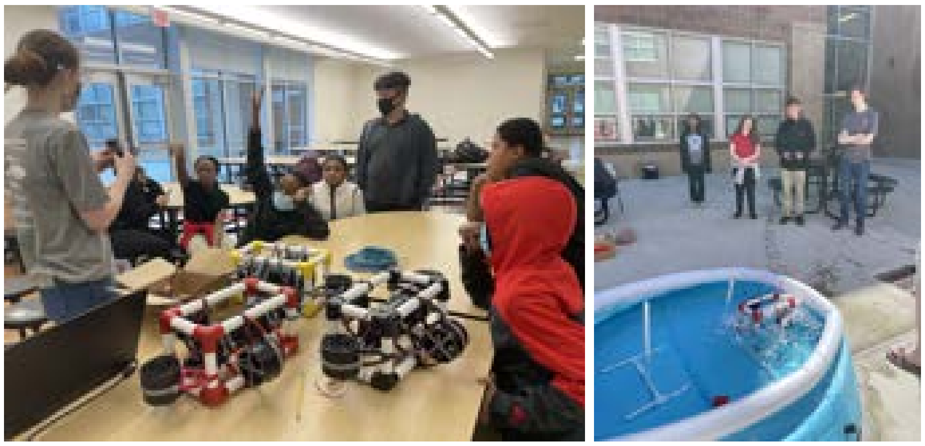


Fig. 25: STEMBot outreach activities at Rosemore Middle School