RoboSub 2026 Technical Design Report

Aditya Gupta, Ayden Gardner, Ben Alexander, Blake Smith, Cameron Tucker, Ethan Rosati, John Darnley, John Ulm, Josh Fishman, Morgan Birely, Vishwa Senthil Kumar, William Dailey

***Abstract*—UWRT is deploying two autonomous underwater vehicles at RoboSub 2026 to balance competition reliability with continued platform development. Talos, the team's established vehicle, has been refined to improve consistency on complex autonomous tasks, while Triton introduces a smaller, agile platform for parallel operation and future system validation. This division of roles allows the team to expand task coverage while preserving the reliability of its primary vehicle. Simulation, subsystem validation, and in-water testing are used to evaluate the integrated two-vehicle system.**

Fig. 1: Talos

Fig. 2: Triton

I. Competition Strategy

A. General Strategy

UWRT's strategy for RoboSub 2026 centers on multi-vehicle collaboration to maximize task completion and pursue all available points. The team refined its existing competition vehicle, Talos (Figure 1), to improve consistency and reliability for high-complexity tasks. A new platform, Triton (Figure 2), was developed to complete tasks in parallel with Talos and validate concepts for future vehicles.

B. Vehicle Task Allocation

Talos and Triton both provide full six-degree-of-freedom (6-DOF) control for underwater navigation, but are assigned different course roles based on each task's sensing and actuation requirements. Triton is designed for agile operation, making it well suited for the *Gate / Return Home* tasks. Talos is equipped with a broader set of actuation and sensing capabilities including manipulation systems, pinger localization, and navigation using a Doppler Velocity Log (DVL), enabling it to execute tasks such as *Slalom*, *Bin*, *Torpedo*, and *Octagon / Table*. Both vehicles support vision-based perception and acoustic communication for Inter-Vehicle Communication (IVC). The task and capability allocation between Talos and Triton is summarized in Table I.

Vehicle	Task	Required Capability			
		Actuation	Maneuverability	Vision	Acoustics
Triton	Gate / Return Home	No	Yes	Yes	No
Talos	Slalom	No	Yes	Yes	No
Talos	Bin	Yes	Yes	Yes	No
Talos	Torpedo	Yes	Yes	Yes	Yes
Talos	Octagon / Table	Yes	Yes	Yes	Yes
Talos / Triton	IVC	No	No	No	Yes

TABLE I: Task Allocation Matrix

C. Dual-Platform Development Strategy

UWRT's dual-platform development strategy separates competition reliability from future development. Talos was maintained as the team's stable primary AUV, allowing development to focus on improving autonomous task consistency without introducing unnecessary mechanical or electrical risk. This kept Talos available for high-value competition testing throughout the season.

Triton was developed in parallel as a focused secondary platform for new AUV concepts, absorbing higher-risk development work while preserving the primary vehicle's competition readiness. This gave the team a dedicated testbed for future mechanical, electrical, and software architectures, increasing testing throughput and providing a practical path to evaluate new systems before considering them for a primary competition vehicle.

II. Design Strategy

A. Talos Mechanical

Talos's mechanical development focused on targeted task mechanism improvements while preserving the vehicle's proven chassis and system layout. The team refined manipulation for *Octagon / Table* and projectile consistency for *Torpedo*, improving Talos's ability to complete high-precision tasks without adding unnecessary integration risk to the primary competition vehicle.

1) Claw Redesign

For RoboSub 2026, one of UWRT's goals was to update the claw to attempt the *Table* task and accommodate various potential prop shapes. After researching claw designs used in industry [1], the team developed a solution using compliant robotics principles, shown in Figure 3. The jaw portion of the claw utilizes a ribbed design 3D-printed out of TPU, allowing the claw to conform around a wide range of shapes; the gripping faces of the claw are also treated with Dragon Skin 20 silicone [2] to increase surface friction, improving object retention during manipulation. The claw is actuated using a rack-and-pinion mechanism to open and close the jaws, allowing the claw to operate with a single servo while retaining strong inward gripping force.

Fig. 3: Updated claw actuator and mount

To streamline the transport and deployment process of Talos's claw actuator, an adjustable hinged mount was designed. The mount uses retaining pins to lock the arm at adjustable extension lengths and deployment angles, allowing the claw's reach and orientation to be reconfigured for each year's competition requirements. When the claw is not in use, the arm can be rotated upward to protect the end effector during transportation and allow Talos to rest safely on flat surfaces.

2) Neutral-Buoyancy Torpedo Redesign

UWRT's previous projectile design was intentionally negatively buoyant, which worked well for marker drops in the *Bin* task but was less ideal for *Torpedo*, where vertical drop reduces shot consistency over distance. For RoboSub 2026, the team separated these use cases: the sinking version will continue to be used as a marker, while a new neutral-buoyancy variant was developed for the *Torpedo* task.

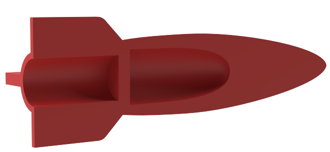

Fig. 4: Neutrally-Buoyant Torpedo

The updated torpedo remains 3D printed and retains the same exterior geometry as UWRT's previous design, preserving compatibility with the existing launcher. A hollow internal cavity, shown in Figure 4, was added to better balance the torpedo's mass and displacement while shifting buoyancy toward the rear, helping the torpedo maintain orientation during travel.

B. Talos Electrical

Talos's electrical development emphasized reliability and consistency within the existing vehicle architecture. Updates such as IVC integration and improved actuator firmware supported new multi-vehicle capability and improved task-mechanism feedback while maintaining Talos's proven electrical foundation.

1) IVC Firmware & Protocol

UWRT's IVC system uses an in-house circuit board to enable underwater acoustic communication between Talos and Triton. The board is built around an RP2040 [3] microcontroller, which drives a custom transducer using pulse width modulation for transmission and receives data by sampling an Aquarian Audio H1C hydrophone [4] using its onboard analog-to-digital converter.

The communication protocol was designed with reliability as the primary requirement. Frequency shift keying was selected because it is simple to generate in firmware and robust to underwater signal distortion. Message data is encoded as a binary stream using two separated frequencies, along with a synchronization signal that indicates the start of communication and defines symbol timing. The transmit frequencies of 51, 52, and 53 kHz were selected to remain within the effective bandwidth of the acoustic hardware while maintaining enough separation for reliable detection after filtering. A diagram of a typical IVC packet can be found in Appendix F.

2) Claw Firmware Improvements

The team enhanced the servo firmware running on Talos's actuator board to improve reliability, add support for multiple simultaneous servo connections, and enable continuous rotation across arbitrary angles. Since the servos natively permit angles only within a 0-240° range, the firmware detects over/underflows in the servo's internal 32-bit position value to track multi-turn motion. The tracked position state is stored by the actuator board's onboard microcontroller, allowing actuator positions, such as the claw's, to persist across reboots. A stall detection algorithm was then built atop the position tracking to determine if the claw picked up an item. That information, along with the servo's persistent position, is relayed to the autonomy system to better support *Table* task behavior.

C. Talos Software

Talos's software development aimed to improve the consistency of its existing competition stack through targeted updates. Upgrades to controller stability, perception reliability, and autonomy tree structure helped Talos adapt to new course requirements while preserving its mature software foundation.

1) Controller Stability

Talos's controller was refined to improve stability during precision tasks, specifically *Torpedo* and *Table* tasks. The team resolved issues in the angular Proportional-Integral-Derivative (PID) controller's integrator that had previously caused undesirable accumulation leading to oscillation during attitude control. With the integrator behavior corrected, Talos became more predictable when holding roll, pitch, and yaw setpoints.

The team also improved feedforward tuning and updated the hybrid controller that combines PID control with Sliding Mode Control (SMC). For the hybrid PID/SMC controller, shown in Figure 14, tuning prioritized stability over speed so Talos could hold commanded positions more reliably before executing task actions. This tradeoff improves performance in *Torpedo* and *Table* tasks, where consistent positioning and attitude control are more important than rapid maneuvering.

2) Perception System

UWRT's vision system builds on the perception pipeline used during the previous two competition cycles, with targeted updates for RoboSub 2026. The same core pipeline is shared across Talos and Triton, while the detection model was updated to YOLOv26n-seg [5] and the training data was expanded using both augmented and generated datasets.

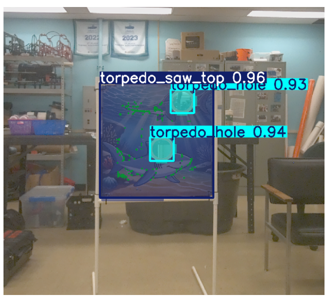

Fig. 5: YOLO Detections

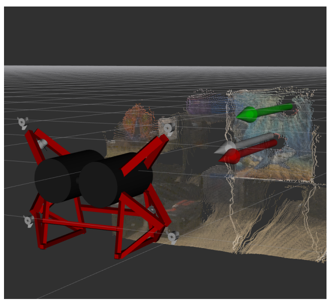

Fig. 6: 3D Pose Estimates

For stereo-camera inputs, YOLOv26n-seg first detects and segments task elements, shown above in Figure 5. Within each detected region, OpenCV's Good Features to Track algorithm [6] extracts 2D feature points, and the segmentation mask rejects points that do not lie on the object. The remaining points are mapped onto the camera depth map to generate 3D points. For planar task elements, singular value decomposition fits a plane to estimate object orientation, while the detection centroid and depth data provide the object's relative position, as seen in Figure 6.

Non-planar objects, such as *Slalom* poles, are handled by sampling the closest valid depth point within the segmented region rather than fitting a plane. To improve robustness, the team expanded its training data with traditional augmentations, such as rotation, mirroring, cropping, scaling, and compositing, as well as generated underwater scenes that vary object angle, lens distortion, water color, visibility, lighting, and caustic patterns. This improves model generalization across different course layouts and water conditions.

3) Autonomy

Talos continues to use UWRT's behavior-tree-based autonomy system for mission execution. The tree structure organizes each task as a reusable sequence of search, approach, alignment, validation, and action nodes, allowing high-level mission logic to remain modular while still using feedback from perception, localization, and the controller. For RoboSub 2026, the autonomy work on Talos focused on adapting this existing system to new course logic.

The main update was restructuring the top-level tree to support the random role assignment required for the *Gate* task's Disruptive scoring category. The selected role is determined at the start of tree execution and passed into all relevant task subtrees. Each downstream behavior then uses the assignment to select the correct target labels, task ordering, and action parameters. This structure keeps role-specific logic at the top level, allowing the task subtrees to remain reusable and independently verifiable.

D. Triton Mechanical

Triton's mechanical system was developed to support a maneuverable secondary AUV while reusing major existing vehicle components. The team designed a new chassis around Triton's hull, thruster configuration, and modular mounting needs, allowing the vehicle to meet its RoboSub task requirements while providing a flexible layout for evaluating future vehicle designs.

1) Thruster Configuration

Triton's thruster layout was selected through a combination of literature review and vehicle-specific simulation. A recent thruster-configuration study [7] compared common eight-thruster layouts using capability boundaries, maximum velocity, energy consumption, maneuverability, and trajectory tracking, showing that no single layout is universally optimal across all vehicles and missions. Instead, the best configuration depends on the vehicle geometry and task requirements.

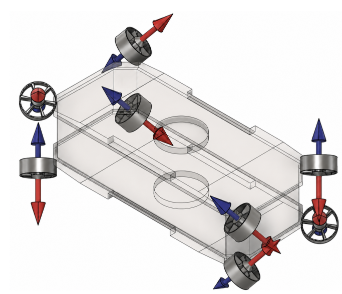

Fig. 7: TC-16 Thruster Layout

Using this approach as a reference, UWRT tested multiple eight-thruster configurations in simulation before selecting a TC-16-inspired layout for Triton, shown in Figure 7. This configuration provided a strong balance of 6-DOF maneuverability, surge capability, and rotational stability given the constraints of Triton's reused hull. By distributing force generation across multiple thrusters rather than assigning individual thrusters to isolated axes, the layout supports combined translation and rotation while reducing large thrust reversals during control.

2) *Chassis*

Triton's chassis was designed around two major constraints: reusing an existing cylindrical hull and supporting the simulator-validated eight-thruster layout in Figure 7. To address the limited mounting points of the cylindrical hull, six semicircular clamps were designed to wrap around the hull and transfer loads through friction and the front and rear flanges. Additional anti-rotation supports prevent the chassis from shifting under load, creating a reliable hull-to-chassis interface without compromising structural integrity.

Beyond supporting the selected thruster layout, the chassis was designed to improve control stability. Its wide, flat profile increases the moment arms of the thrusters, improving rotational response, while also providing a stable arrangement of the center of mass and center of buoyancy. Open areas at the front, rear, and sides of the chassis allow thruster wash to pass through with minimal obstruction, reducing thrust losses while preserving structural rigidity.

The chassis itself was designed from aluminum box channels and laser cut sheet metal parts. Aluminum 6061 alloy was selected for its rigidity, machinability, and low cost. The chassis design aimed to minimize the number of unique parts, allowing metal components to be ordered in bulk; this reduced manufacturing costs while simplifying and accelerating vehicle assembly. After fabrication, the aluminum components were anodized to improve corrosion resistance during repeated cycles of underwater operation.

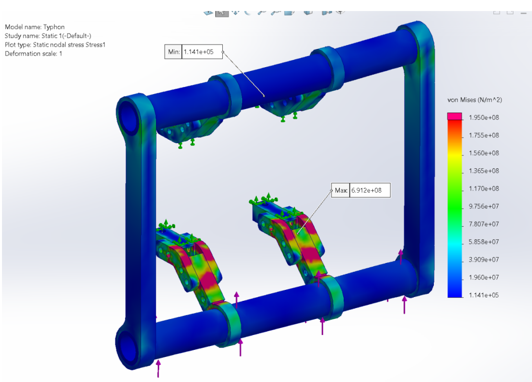

Fig. 8: Static load of Triton chassis segment

Finite Element Analysis (FEA) was used to verify that Triton's chassis could withstand expected handling and competition forces. The analysis used 200 lb vertical and 100 lb shear benchmark load cases, chosen to guarantee a factor of safety of at least 2. Initial FEA results (Figure 8, above), showed that the first chassis layout deflected approximately 0.5 in under load, indicating that additional stiffness was needed. The chassis was then iterated alongside the thruster layout, allowing support beams to be repositioned without obstructing thruster flow.

The final chassis achieved the desired structural benchmarks while adhering to the principal constraints of component reuse and the simulator-validated thruster layout.

3) *Modular Mounting*

Triton's design incorporates a modular mounting system, allowing for customizable attachment points for sensors and actuators on top of the main hull. This structure supports Triton's role as both a RoboSub 2026 vehicle and a development platform for future systems. The final design was chosen for its rigidity, accessibility, and ability to mount essential components such as the IVC system and kill switch.

E. *Triton Electrical*

Triton's electrical system was developed to preserve compatibility with UWRT's existing vehicle architecture while fitting within a smaller platform.

1) *Board Compatibility*

The Triton electrical systems were designed to maximize compatibility with the existing Talos architecture. Both systems are built using a dedicated RP2040 microcontroller on each circuit board, with boards exchanging ROS2 messages over a custom CAN bus [8] protocol. This allows Triton to reuse Talos's proven circuit designs and firmware, maintaining electrical compatibility between systems. Both robots also use the same custom external battery pack, allowing batteries and charging equipment to be shared between vehicles.

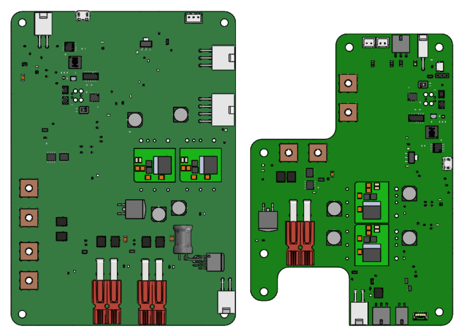

Fig. 9: Talos PCB (left) and equivalent Triton PCB (right)

2) *Board Density*

As seen above in Figure 9, the new Triton boards are approximately 36% smaller than their equivalent Talos counterparts while preserving the same core functions: power regulation, telemetry gathering, motor speed control using APD 80F3 ESCs, and IVC. This reduced board size is necessary for Triton's smaller hull, allowing the vehicle to retain comparable electrical capability within a smaller internal volume. Full internal architectural diagrams are provided in Appendix D and board-level block diagrams can be found in Appendix E.

F. *Triton Software*

Triton's software system was developed to make UWRT's software stack more efficient, configurable, and reusable across multiple AUV platforms. During the 2025-2026 development cycle, the team rewrote substantial portions of the existing stack and refactored others so Triton could reuse mature Talos components while still supporting vehicle-specific configuration and development. This allowed Triton to operate as a RoboSub vehicle while also serving as a software testbed for future AUVs.

1) Rewritten Software Stack

Rewriting major portions of the software stack was especially important for Triton because its smaller hull limits onboard computer size, requiring the shared stack to run efficiently with less compute headroom. Beyond runtime performance, the rewrite also addressed maintainability issues that become more significant as the same stack is deployed across multiple vehicles.

To improve developer experience, documentation was expanded through automatically generated Sphinx [9] pages, and the code review process was updated to require documentation for new features. The project structure was also standardized across modules, and coupling between subsystems was reduced so that new vehicles can be specified primarily through configuration files. This allows Triton to share a common software stack with future AUVs while still supporting their different hardware layouts and mission roles.

2) Visual Odometry

Triton uses a Stereolabs ZED X Mini [10] to evaluate visual-odometry-based localization for future UWRT vehicles; the ZED X Mini is already used on Talos as part of the team's stereo camera system, but Talos relies on a DVL for velocity feedback. Since Triton does not currently have a DVL, the ZED SDK's visual odometry output is used to provide linear velocity estimates in the x and y directions for Triton's Unscented Kalman Filter.

This setup keeps Triton's localization stack lightweight while still covering the main feedback needed for autonomous testing. The ZED X Mini contributes planar velocity, the depth sensor provides z-position feedback, and the onboard IMU supplies roll, pitch, and yaw measurements. This makes Triton a useful platform for testing visual odometry as a lower-cost velocity feedback option for future vehicles.

3) Autonomy Improvements

One of the largest software improvements for RoboSub 2026 was replacing UWRT's aging BehaviorTree.CPP (BT.CPP) [11] V3 autonomy workflow. Rather than directly upgrading to BT.CPP V4, the team developed a customized behavior tree framework built around BT.CPP with UWRT-specific extensions. These additions include a more robust variable management system with static typing, scoped variables, and global constants, reducing common behavior tree errors during mission development.

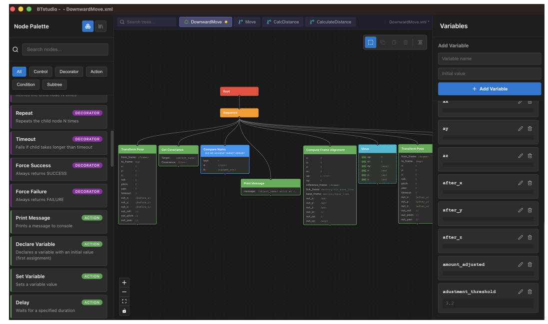

Fig. 10: Behavior Tree in BT Studio

To support this new autonomy framework, UWRT also developed BT Studio, a custom behavior tree editor designed for the team's workflow. BT Studio, shown in Figure 10, provides a visual node-based interface for editing mission trees while still generating BT.CPP-compatible XML. Features such as keyboard shortcuts, quick search, and automatic workspace management reduce the time required to create and modify autonomous behaviors.

Together, the behavior tree framework and BT Studio give the team greater control over its autonomy tools while improving developer efficiency. Automated tooling connects the two systems by generating node manifests from the autonomy codebase, allowing BT Studio to stay synchronized with available behavior tree nodes. This workflow makes it easier to develop, reuse, and maintain autonomous missions across Triton and future vehicles.

III. Testing Strategy

A. Parallelized Water Testing

To make the most of in-water time, UWRT parallelized testing across Talos, Triton, and IVC development. This involved simultaneous operation of both vehicles, with Talos validating advanced autonomous behaviors while Triton tested new technologies. IVC testing was also parallelized by placing one endpoint on Talos and one endpoint poolside, allowing the team to access serial debug output and probe exposed test points on the IVC board with an oscilloscope. This strategy allowed more subsystems to be validated during each pool test and helped the team identify issues earlier in development, making better use of available resources and driving significant progress this year.

B. Simulation-Based Autonomy Testing

UWRT used a custom ROS2 simulation environment to validate autonomy behavior before committing pool time to physical vehicle testing. The simulator can be run in a standalone software environment or in a Hardware-In-the-Loop (HIL) configuration using the onboard compute stack from Talos or Triton. In both modes, vehicle-specific autonomy trees receive simulated localization, perception, and controller feedback, allowing mission logic to be tested without changing the autonomy code used on the vehicles.

The simulator models vehicle motion from physical properties and commanded thruster outputs, then publishes simulated state and sensor data back into the ROS2 stack. This allowed the team to test full task routines for both vehicles, including search patterns, task transitions, timeout conditions, recovery behavior, and action sequencing. Standalone simulation was used for rapid iteration during development, while HIL testing helped verify that ROS2 interfaces, autonomy trees, and onboard compute environments behaved correctly before pool deployment.

For RoboSub 2026, simulation was especially useful for tasks with updated autonomy logic. The random role assignment for the *Gate* task's Disruptive scoring category was tested by setting both possible assignments at the start of tree execution and verifying that it propagated through the relevant subtrees. The *Table* routine was also tested in simulation to validate grabbing behavior, with HIL simulation allowing claw commands to be verified using real actuator hardware. Together, these tests reduced pool time spent debugging tree structure and allowed water tests to focus on vehicle behavior, perception quality, and real-world hydrodynamics.

C. IVC Testing

IVC testing emphasized system reliability by validating each layer of the implementation individually in a challenging acoustic environment. Testing began by placing a transducer and hydrophone 1 m apart in the pool and measuring the received amplitude of a single-frequency signal. This procedure was repeated across frequencies from 5-100 kHz in 5 kHz increments and distances from 1-5 m in 0.5 m increments. The results showed that frequencies in the 50 kHz range provided the best balance between transducer response and transmission distance. At 1 m, a 50 kHz signal was measured 20 dB above the noise floor, giving the team confidence that the selected frequency range would provide resilient communication in a noisy environment. A graph of this test, Figure 11, is shown below.

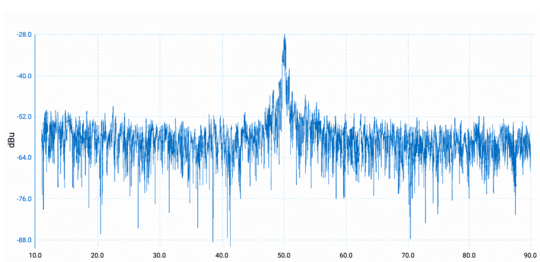

Fig. 11: 50 kHz transmission frequency-domain graph

Once the signal-handling layer was established, the next step was testing the implementation of data transfer and selecting an appropriate data rate. Transfer speeds from 1 Bit Per Second (BPS) to 100 BPS were tested, with a 20 BPS speed providing acceptable transfer rates while successfully transmitting packets in more than 75% of cases. To further improve reliability, a 4-bit cyclic redundancy check was appended to each packet, allowing malformed packets to be detected and re-sent. With this retransmission logic in place, the team was able to reliably transfer a given packet, even if several attempts were required.

D. Claw Prototype Evaluation

During the claw redesign process, the team focused on developing a claw that could adapt to a wide range of potential *Table* task objects; this reduced the likelihood of requiring a redesign for future competitions. Multiple claw archetypes were prototyped and evaluated based on reliability, adaptability, and maintainability.

Reliability was tested by determining whether each claw prototype could consistently pick up objects from a variety of approach angles and precisely deliver them. Adaptability was assessed using *Table* task props from previous competitions to determine whether the jaws could grip and retain objects of varying geometries. Maintainability was evaluated based on how easily the mechanism could be manufactured and replaced, favoring modular designs with low 3D-printing and assembly times.

These tests guided the final claw design shown in Figure 3, which used compliant TPU jaws and a rack-and-pinion actuator to improve object retention while keeping the mechanism simple and serviceable. Underwater testing also showed that surface friction between the TPU jaws and smooth plastic props was a key factor in object retention. To address this, a thin layer of Dragon Skin 20 silicone was applied to the gripping faces.

The silicone coating did not reliably bond to TPU after repeated use, so the team tested several jaw-surface interfaces to improve retention of the coating. Each interface was tested using TPU sample prints with different surface textures and mechanical-locking features to compare how well the silicone coating remained attached after repeated grasp cycles. The final interface uses a hexagonal surface texture to increase contact area and dovetail notches along the jaw to mechanically lock the silicone coating in place.

Acknowledgment

UWRT owes its achievements this year to the generous support and contributions from many individuals and organizations. The team would first like to thank its sponsors: Altium, Anduril, Caterpillar, Funtrail, Hargrave Technologies, Nortek, VectorNav, Tektonix, and The Ohio State University College of Engineering. Special thanks go to the Electrical and Computer Engineering Department and Dr. Saideh Zia, its faculty advisor, for providing lab space, equipment, and invaluable guidance.

Testing facilities were generously made available by Sean Danekind, the Tirpak family, and the staff of The Ohio State Recreation and Physical Activity Center, providing the team with essential in-water time to validate its vehicles.

Finally, UWRT would like to thank its team members for their dedication and hard work that brought both vehicles to their present state, and RoboNation for their continued stewardship of the RoboSub competition.

References

[1] M. S. Sofla, H. Golshanian, E. I. Sklar, and M. Calisti, "Modeling and modification of fin-ray effect grippers to improve their load capacity and grasp stability," *Sensors and Actuators A: Physical*, vol. 392, p. 116711, 2025.

[2] Smooth-On, Inc., "Dragon Skin 20 Silicone Rubber." https://www.smooth-on.com/products/dragon-skin-20/, 2026. Accessed: 2026.

[3] Raspberry Pi, "Rp2040 datasheet." https://pip-assets.raspberrypi.com/categories/814-rp2040/documents/RP-008371-DS-1-rp2040-datasheet.pdf?disposition=inline, 2021.

[4] Aquarian Audio, "H1c Hydrophone." https://www.aquarianaudio.com/h1c-hydrophone.html, 2018. Accessed: 2026.

[5] Ultralytics, "YOLO26: Real-time object detection and image segmentation," 2026.

[6] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[7] A. K. K.S., J. Kadiyam, and S. Mohan, "Comparative performance investigations of the intervention-class underwater vehicle with different possible thruster configurations using eight identical thrusters," *Ocean Engineering*, vol. 288, p. 116147, Nov. 2023.

[8] ISO, "Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling," Tech. Rep. ISO 11898-1:2003, International Organization for Standardization, Geneva, Switzerland, 2003.

[9] Sphinx Team, "Sphinx — Sphinx documentation." https://www.sphinx-doc.org/en/master/.

[10] Stereolabs, "ZED X Mini Stereo Camera." https://www.stereolabs.com/products/zed-x-mini, 2022. Accessed: 2026.

[11] D. Faconti, "BehaviorTree.CPP: A C++ library to build Behavior Trees." https://github.com/BehaviorTree/BehaviorTree.CPP, 2026. Version 4.9, Accessed: 2026.

[12] Parker Hannifin Corporation, *Parker O-Ring Handbook*. Parker Hannifin Corporation, Cleveland, OH, 2007. ORD 5700.

[13] Eclipse Zenoh Team, "rmw_zenoh: ROS 2 RMW implementation using Zenoh." https://github.com/ros2/rmw_zenoh, 2024. Eclipse Foundation.

[14] The MathWorks, Inc., *MATLAB Stereo Camera Calibrator*. The MathWorks, Inc., Natick, MA, 2024. Computer Vision Toolbox.

[15] Blue Robotics, "T200 Thruster." https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/, 2024. Accessed: 2026.

[16] Open Robotics, "ROS 2: Robot Operating System." https://docs.ros.org/en/humble/, 2022. Humble Hawksbill Distribution.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[18] RoboNation, *RoboSub 2026 Competition Handbook*, 2026. Accessed: 2026.

APPENDIX A
COMPONENT LIST

TABLE II: Talos's Component Specifications

Component	Vendor	Model/Type	Custom/Purchased	Cost	Purchase Year
Acoustics Housings		Aluminum 6061, Polycarbonate	Custom	$98.00	2024
Algorithms: Autonomy		BehaviorTree.CPP v3	Custom		
Algorithms: Localization/Mapping		Extended Kalman Filter/Custom mapping system	Custom		
Algorithms: Vision		YOLO26	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 X Mini	Purchased	$639.99	2025
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	Sponsored	$15,000.00	2018
Downwards-Facing Camera	Stereolabs	Zed X Mini	Purchased	$639.99	2024
Hydrophones	Aquarian Audio	H1C	Purchased	$400.00	2018
Inertial Measurement Unit (IMU)/Compass	Vectornav	VN-100-T	Sponsored	$1,300.00	2022
Fiber Optic Gyro (FOG)	Fitzoptica	VG103S-2LND	Purchased	$2,870.00	2024
Microcontrollers	Raspberry Pi	RP2040	Purchased	$0.07	2021
Motor Control	APD	80F3	Sponsored	$245.00	2022
Open Source Software		ROS2/OpenCV/Pico-SDK/BTCPP			
Pressure Valve	IkeLite	1/2" Pressure Valve	Purchased	$125.00	2024
Programming Languages		C/C++/Python/MATLAB			
Silicone Coating	Smooth-On, Inc.	Dragon Skin 20	Purchased	$38.30	2025
Smart Battery Housings	Xometry	Aluminum 6061, Polycarbonate	Custom	$634.00	2023
Task Mechanism Servo	Hiwonder	Hiwonder HTD-45-H	Purchased	$24.99	2024
Thrusters	Blue Robotics	T200	Purchased	$1,432.00	2021
Transducer			Custom	$68.53	2025
Waterproof Connectors	MacArtney	MC/HP/D Series	Purchased	$4,000.00	2015-2023
Waterproof Main Housing	Xometry	Aluminum 6061, Polycarbonate	Custom	$2,233.00	2021
Waterproof Servo Housings	BlueTrail Robotics	SER 2000 Underwater Servo Kit	Purchased	$215.00	2024

TABLE III: Triton's Component Specifications

Component	Vendor	Model/Type	Custom/Purchased	Cost	Purchase Year
Acoustics Housing		Aluminum 6061, Polycarbonate	Custom	$50.00	2024
Algorithms: Autonomy		BehaviorTree.CPP v4	Custom		
Algorithms: Localization/Mapping		Unscented Kalman Filter/Custom mapping system	Custom		
Algorithms: Vision		YOLO26	Custom		
AUV Chassis	SendCutSend	Triton, Aluminum 6061, PVC	Custom	$608.95	2025
Battery	MaxAmps	Li-Po 8000 5S2P 18.5V	Purchased	$300.00	2023
Buoyancy Control	Nalgene	48oz Water Bottle	Purchased	$17.66	2025
Camera	Stereolabs	ZedX Mini	Purchased	$639.99	2025
Communication Network		CAN Bus	Custom		
Converter	TDK-Lambda	I6A4W020A033V	Purchased	$68.34	2022
CPU	Nvidia	Jetson Orin Nano	Purchased	$499.00	2024
Hydrophone	Aquarian Audio	H1C	Purchased	$200	2018
Inertial Measurement Unit (IMU)/Compass	Vectornav	VN-100-T	Sponsored	$1,300	2022
Microcontrollers	Raspberry Pi	RP2040	Purchased	$0.07	2021
Motor Control	APD	80F3v2	Sponsored	$245.00	2024
Open Source Software		ROS2/OpenCV/Pico-SDK/BTCPP			
Pressure Valve	IkeLite	1/2" Pressure Valve	Purchased	$125.00	2024
Programming Languages		C/C++/Python/MATLAB			
Smart Battery Housings	Xometry	Aluminum 6061	Custom	$634	2023
Thrusters	Blue Robotics	T200	Purchased	$1,432	2021
Transducer			Custom	$68.53	2024
Waterproof Connectors	MacArtney	MC/HP/D Series	Purchased	$4,000	2015-2023
Waterproof Main Housing	Xometry	Jaws2, Aluminum 6061	Custom		2015

Appendix B
Software Architecture Diagrams

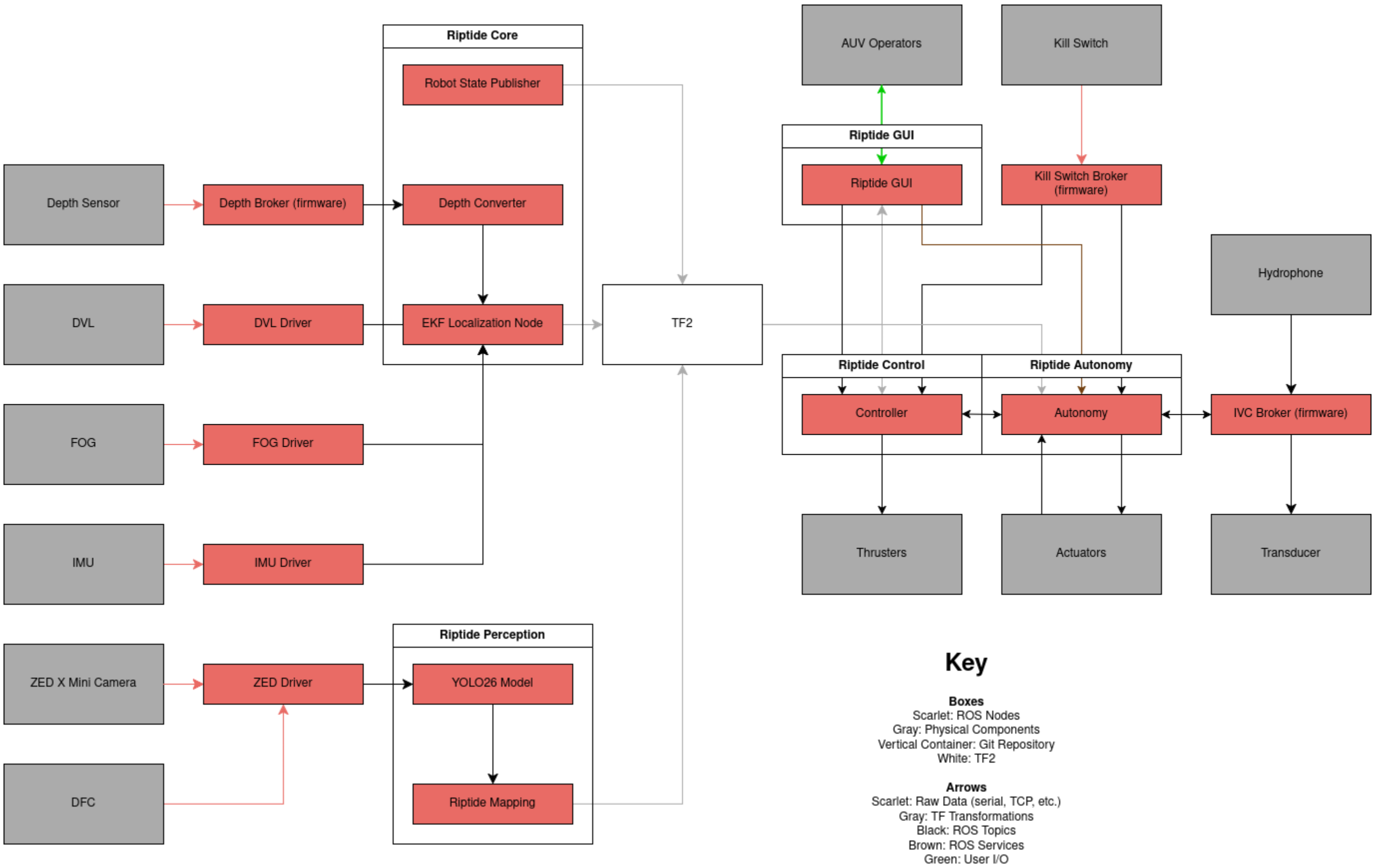

Fig. 12: Talos software architecture

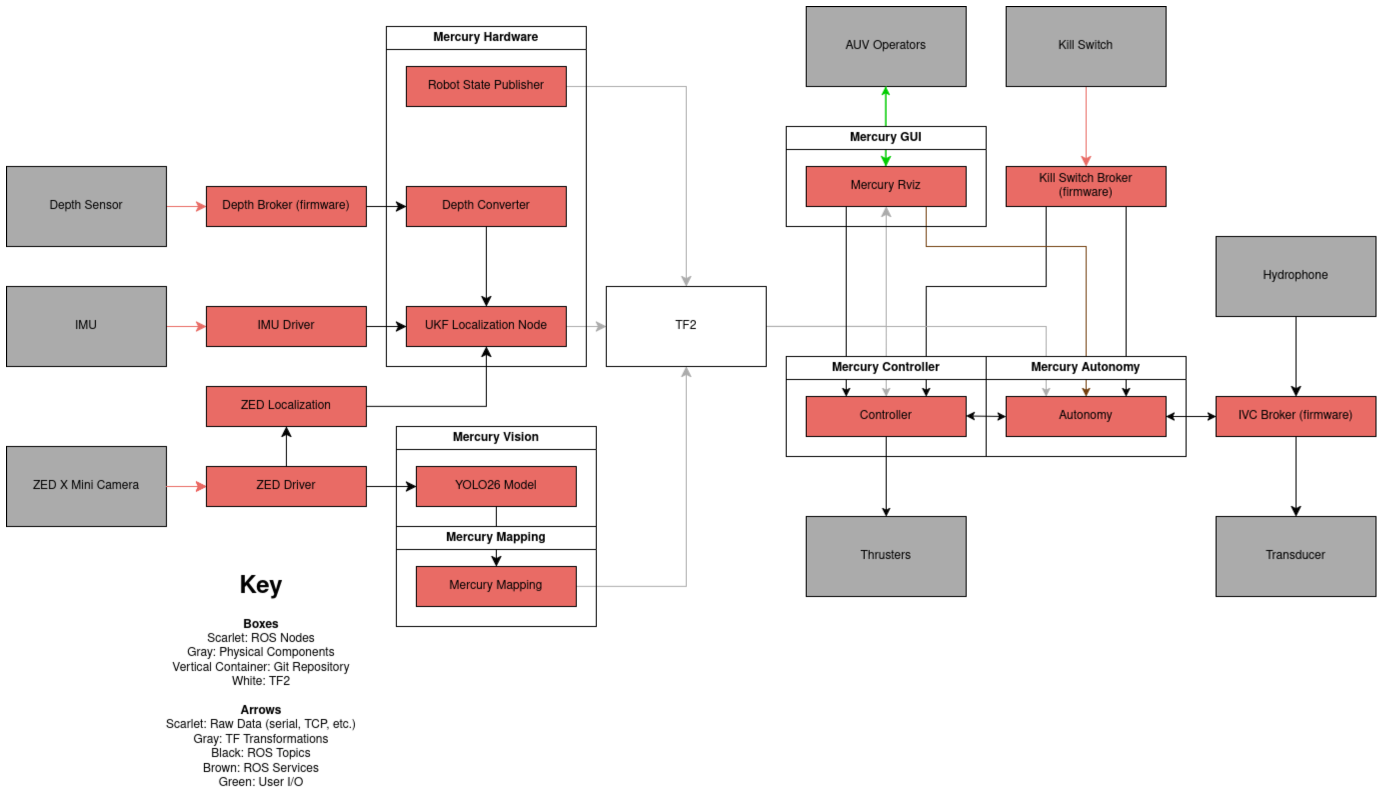

Fig. 13: Triton software architecture

Appendix C
Control System Diagram

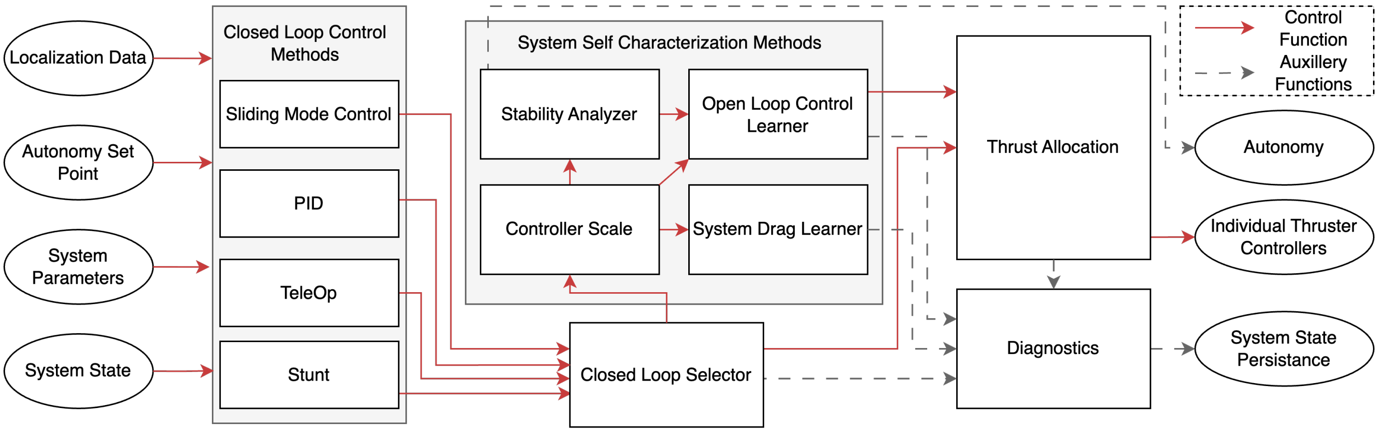

Fig. 14: Control system block diagram

APPENDIX D
ELECTRICAL ARCHITECTURE DIAGRAMS

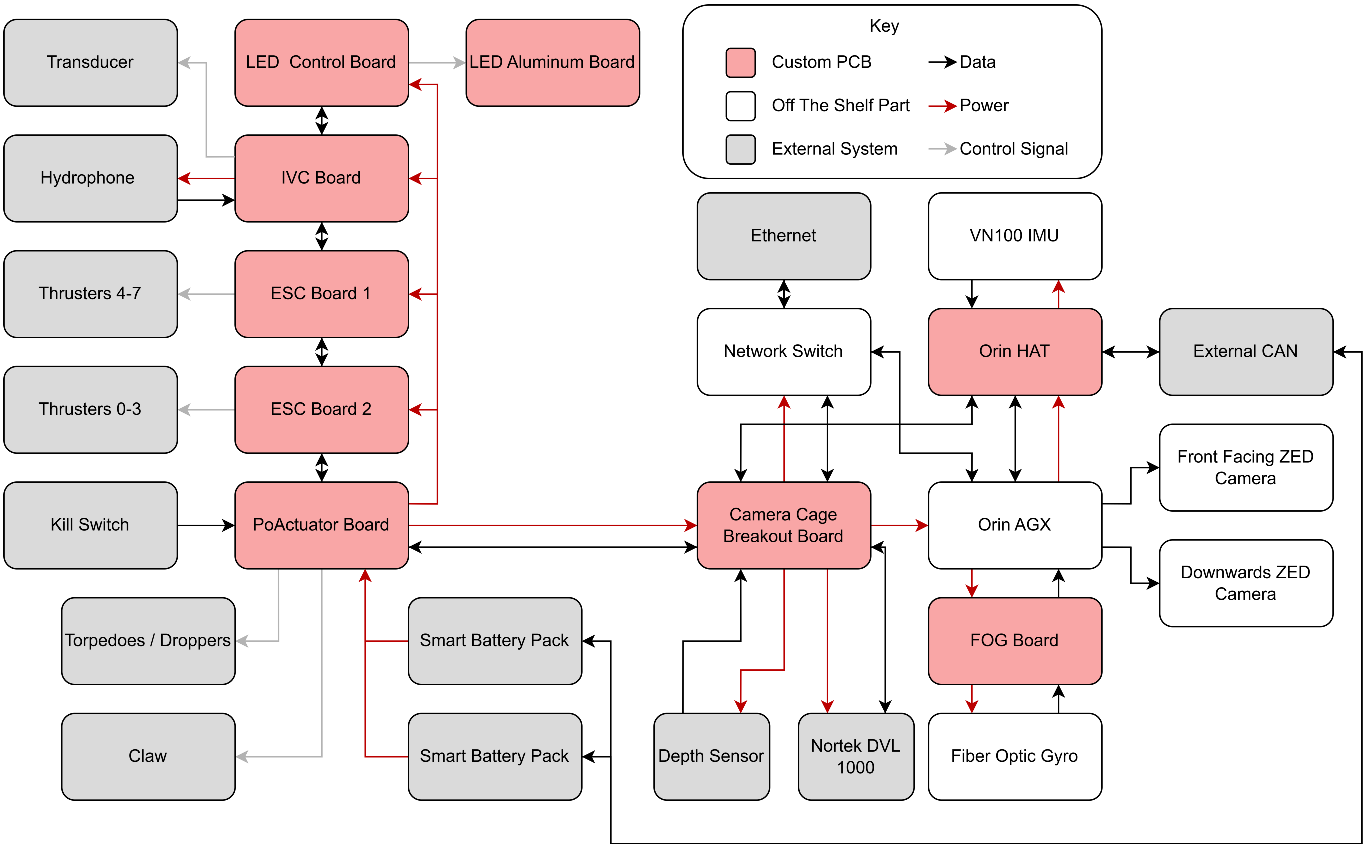

Fig. 15: Talos electrical architecture

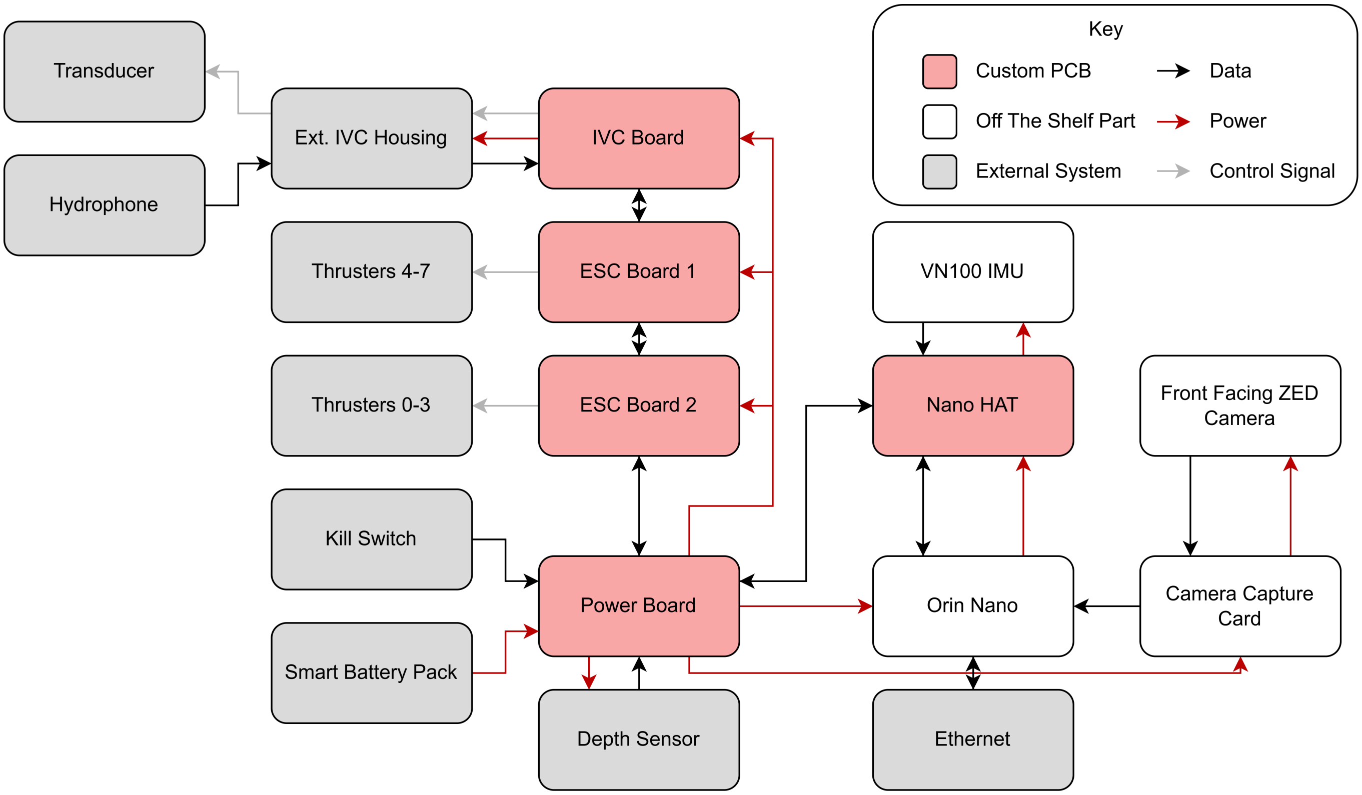

Fig. 16: Triton electrical architecture

Appendix E
Board-level Block Diagrams

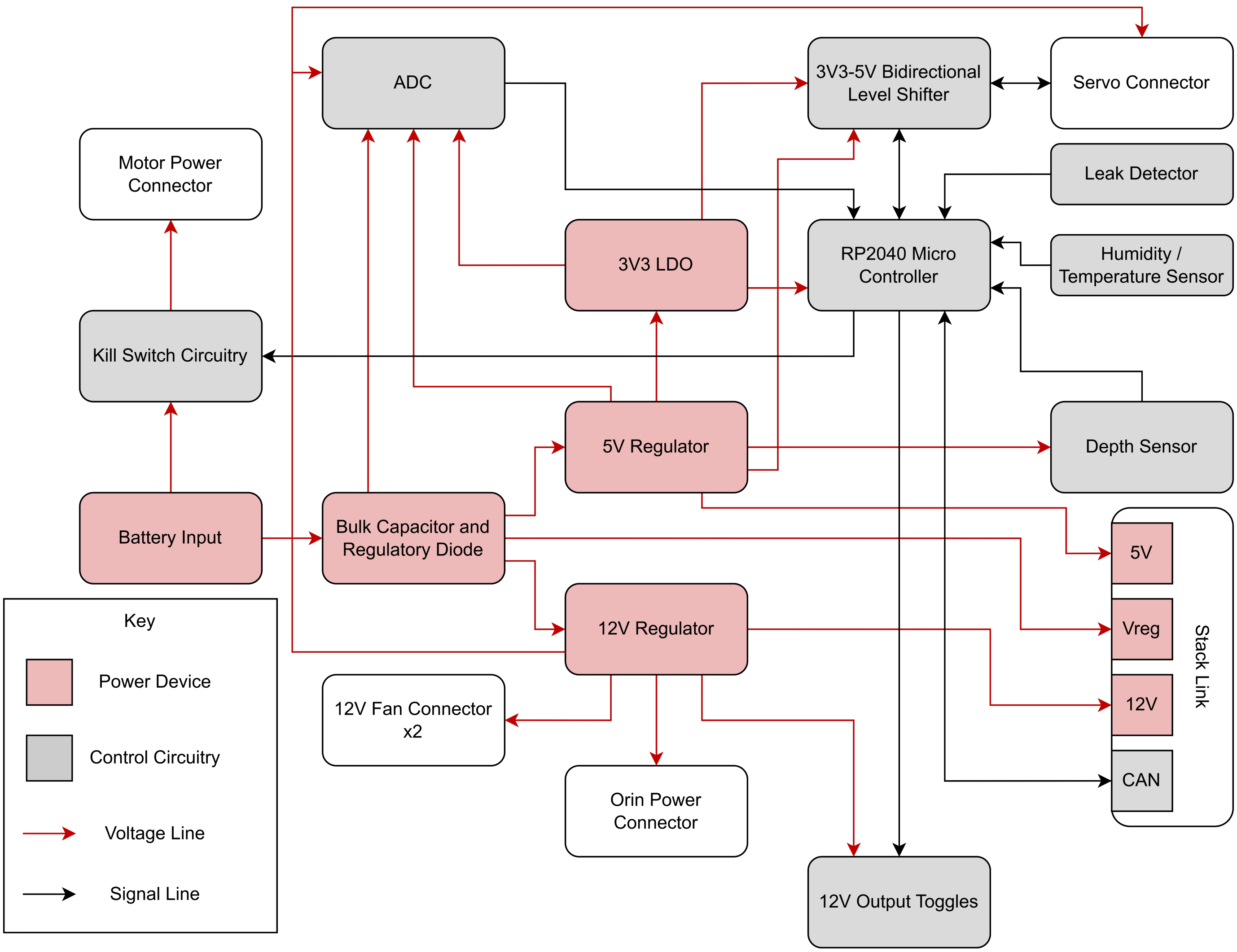

Fig. 17: Triton power distribution board

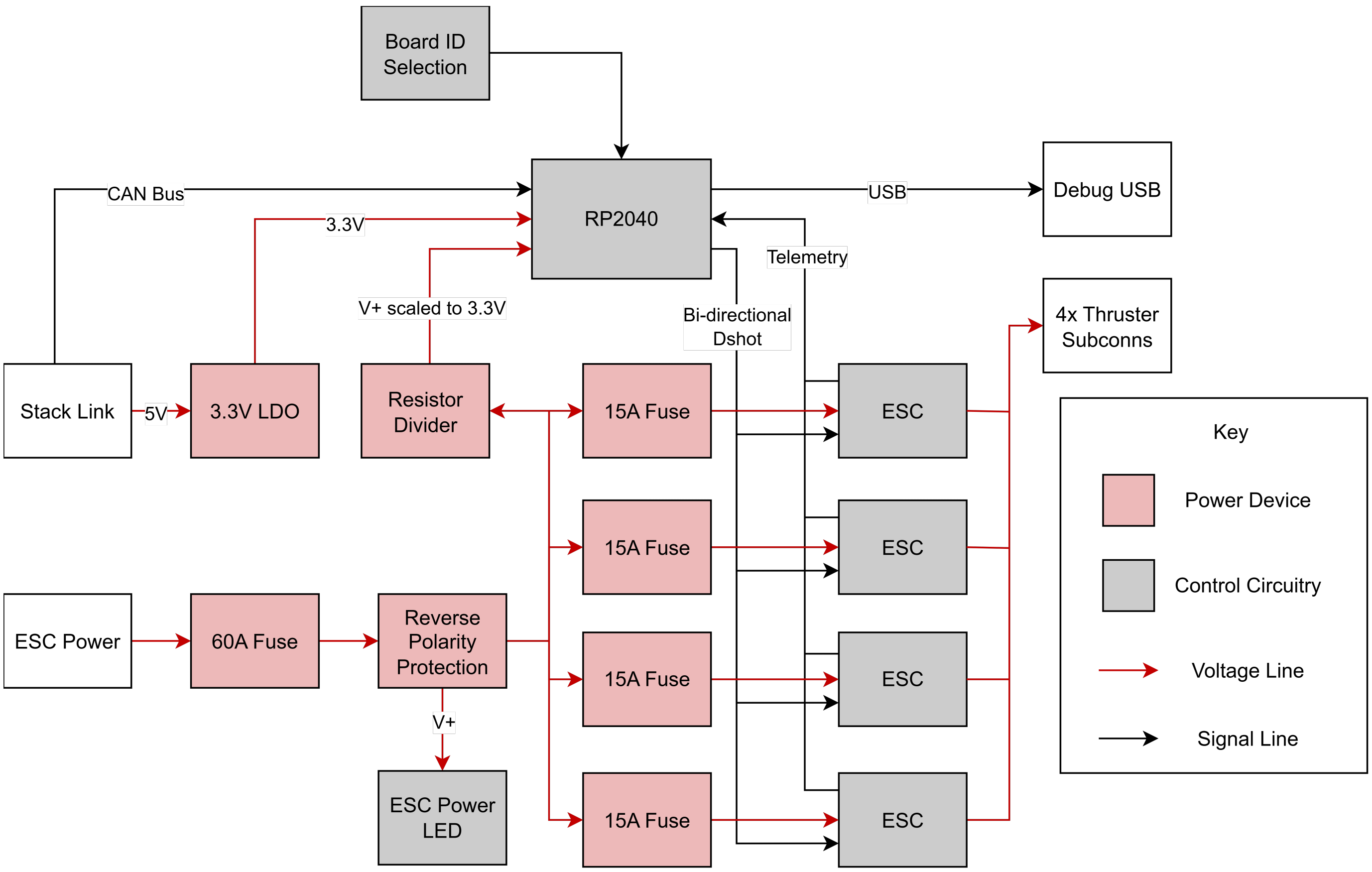

Fig. 18: Triton ESC board

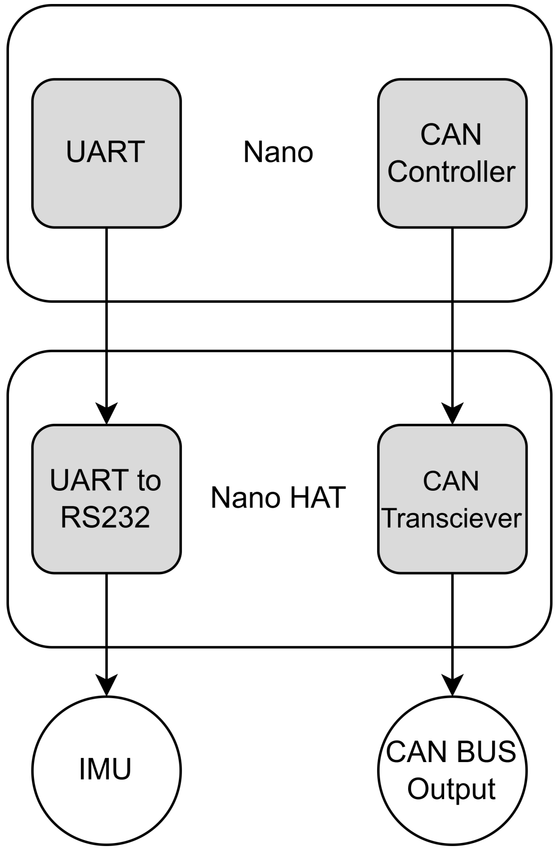

Fig. 19: Triton Orin Nano HAT

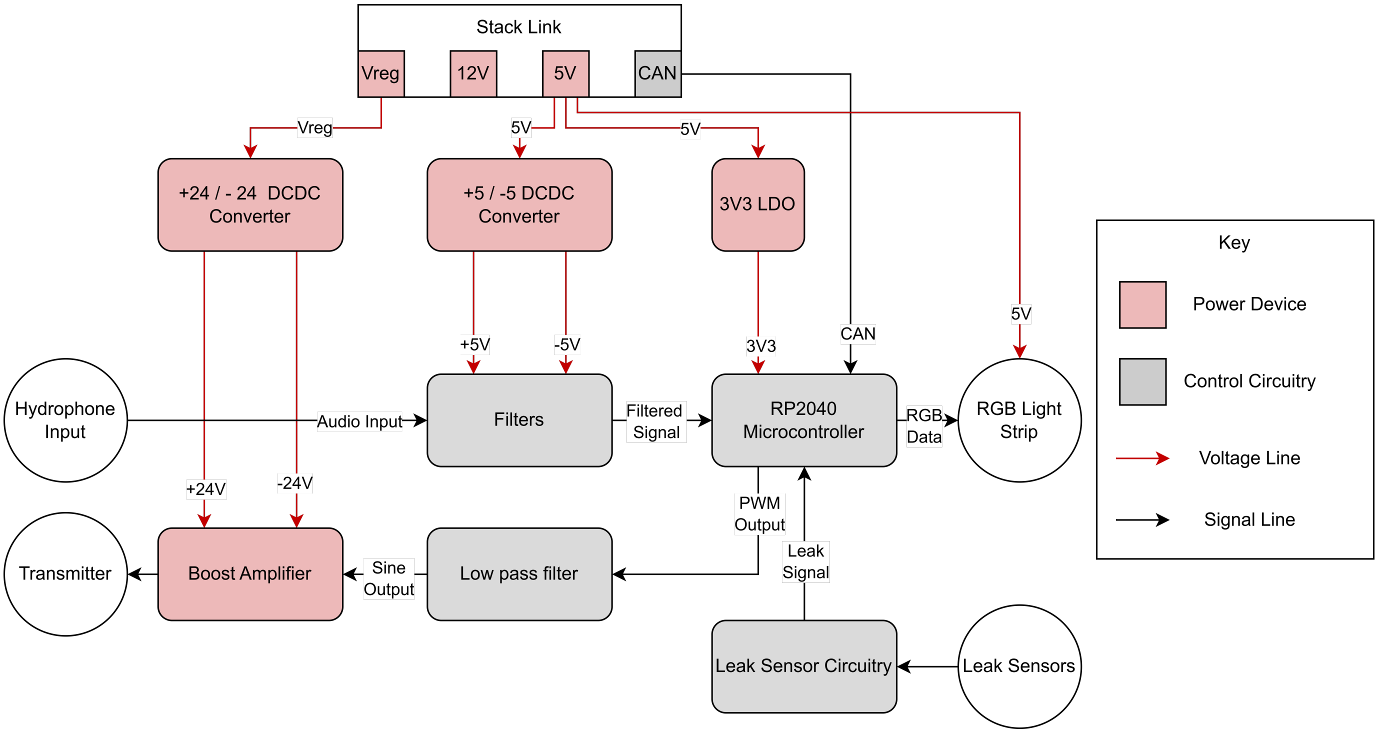

Fig. 20: IVC board

Appendix F
IVC Transmission Diagram

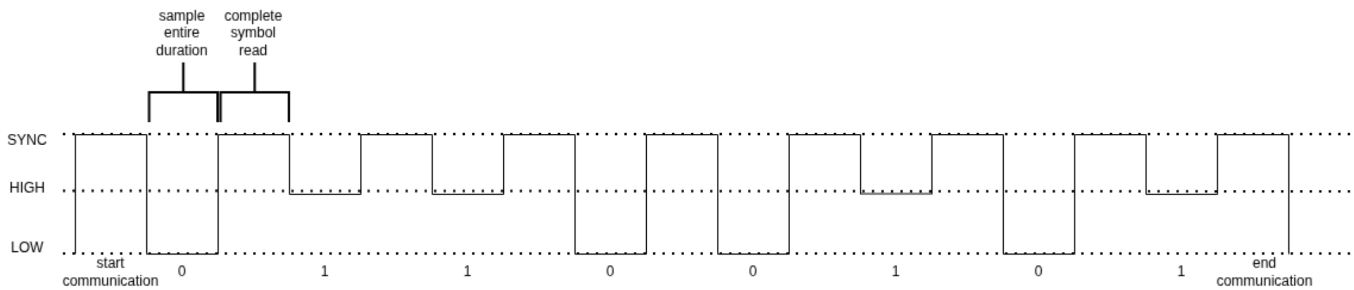

Fig. 21: Typical IVC transmission