Troy High SPEAR Technical Design Report

Yireh Ban, Aidan Chen, Yifei Zhang, Ryan Zhou, Shri Krishna Sivakumar, Humza Shahzad, Gavin Gibson, Matthew Yen, Troy Song, Rener Li, Jocelyn Mathew, Ryan Jian, Wesley Shen, Arjun Reddy, Elvina Liou, Karthik Sajeev, Kirthi Reddy, Mateus Noronha, Kathy He

Abstract — **Troy High SPEAR's AUV, Krabby Patty, has been refined for the 2025 RoboSub competition with an emphasis on reliability and strategic task execution. Notable mechatronic upgrades include a double-jointed claw for versatility, an NVIDIA Jetson Orin Nano, and dual ZED 2i stereo cameras that offer a fully integrated perception and localization solution. The software architecture was upgraded to a modular behavior tree for mission planning, while the vision pipeline employs a YOLOv8 model integrated with ZED 2i spatial data for precise 3D localization. System validation progressed from algorithm unit tests to Gazebo simulations and in-water trials, preparing the integrated systems for the competition's challenges.**

I. Competition Strategy

This year's competition, *Protect the Deep*, introduces several unique twists on tasks from previous years. Despite this, the course remains to consist of 5 main components:

(i) Collecting Data (Gate)
(ii) Navigate the Channel (Slalom)
(iii) Drop a BRUVS (Bins)
(iv) Tagging (Torpedoes)
(v) Collect Samples (Octagon)

As a younger team, we focus on the fundamental elements of the course, such as the gate and channel, before progressing to more complex tasks. This strategy aimed to maximize our ability to score points consistently within the allotted time.

A. Task Prioritization

Through experience, we recognized that allocating equal time to each task hindered our ability to complete any of them effectively. As a result, we adopted a more strategic planning approach by assigning different time allocations to each task based on the following priority: Collecting Data (Gate), Navigate the Channel (Slalom), Drop a BRUVS (Bins), Tagging (Torpedoes), and Collecting Samples (Octagon). The prioritization largely followed the capabilities and limitations of our sensors and tools.

Task order was also determined based on competition requirements, such as the mandatory gate navigation, along with point assignments and overall task difficulty. Efficiency was also considered, ensuring a more optimized path without needing to retrace routes. These considerations were reflected in both the vehicle's physical design and software architecture.

II. Design Creativity

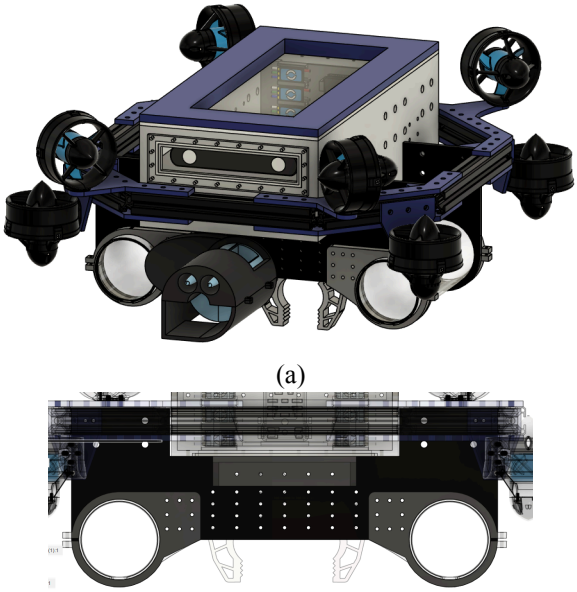
A. Mechatronics (Overall Design)

The primary objective for Krabby Patty was to develop a reliable and modular AUV, that allows for future modifications and is more forgiving. Our designs also considered the manufacturing tools we had access to. Most custom components, such as the claw and torpedo, were 3D printed using PLA, while the frame and main compartments were machined from 6061 aluminum. Our design decisions were also heavily influenced by anticipated implementation challenges from the electrical

and software subteams, with an emphasis on balancing functionality and feasibility.

(i) Frame

Last year, we encountered significant drag issues due to a tall, rectangular frame design. To address this, we transitioned to a shallow, octagonal frame that improves maneuverability and hydrodynamics. The BlueROV2 heavy motor configuration was retained from the previous year, which provides six degrees of freedom for maximum maneuverability. The frame is constructed from 1010 aluminium extrusions cut to length, while the remaining components are CNC-milled in-house from ¼-inch 6061-T6 aluminium.



(a)

(b)

Figure 1. (a) AUV overview, (b) Bottom panel

One particularly notable feature is the customizability of the bottom panels, shown in figure 1(b), which incorporates an array of holes to allow for modular mounting of components like the torpedoes or claw. This flexible layout also makes the frame more reusable for future iterations.

(ii) Main Compartment

The electrical bay, affectionately nicknamed the "patty," houses all the AUV's primary electrical and hardware components. Custom fabricated in-house from 6061 ¼-inch aluminum and welded, as in figure 2, it features two windows and ten ports to allow for camera visibility and scalability. This marks a major improvement from the previous seasons, where we relied on off-the-shelf acrylic tube enclosures from Blue Robotics. For waterproofing, we implemented custom silicone o-rings and sealed potential leak points with epoxy. The lid, machined from ¾-inch 6061-T6 aluminium, features a custom o-ring and is secured with four strategically placed latches to ensure even pressure and consistent watertight sealing. The components inside are mounted by custom 3D-printed electronics trays, designed for modularity and customized for the integrated components.

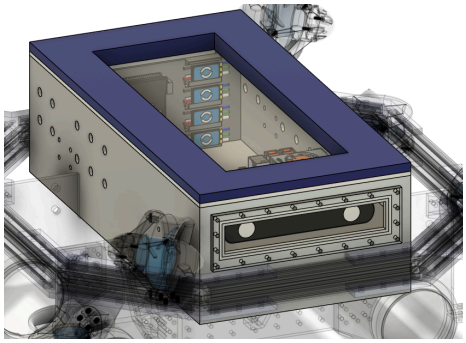


Figure 2. Main Compartment

(iii) Cameras

The AUV is equipped with two ZED 2i cameras from Stereolabs, which feature a built-in accelerometer, inertial sensors, and an SDK that allows integration of YOLO models for object detection on the course. Due to its IMU capabilities, the cameras allowed us to phase out the DVL system used in the previous season, which proved too complex for our team to utilize effectively. The stereo setup enables accurate triangulation and distance estimation through dual-lens depth sensing, while also supporting the precise generation of mesh maps for more efficient localization.

(iv) Claw

The claw, seen in figure 3, utilizes a double-jointed servo-powered mechanism to ensure equal force distribution. This is an essential feature for handling objects that require precision and accuracy. A key design rationale was keeping the claw's "hands" parallel at all times, enabling reliable manipulation of variously sized objects, such as the cup and ladle. The claw is capable of retracting upwards for storage without damage when docked, and extends beyond the submarine to retrieve items underwater. This season, the claw hands were custom-designed for specific tasks. The top and bottom surfaces have additional grip to secure the cup, while an oval cutout in the center improves control of the lid and ladle. When the ladle is grasped by its stem within the oval, the spoon and hook prevents it from slipping free.

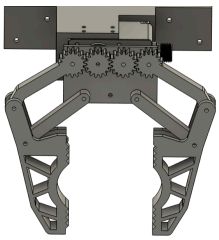


Figure 3. 2025 claw design

(iv) Torpedo and Dropper System

As shown in figure 4, the combined torpedo and dropper system is a servo-powered mechanism that deploys its components at specific actuation angles within a 360-degree loop. This is made possible by a compound gearbox that amplifies the servo's 70-degree range. The torpedoes are held in tension by compression springs, and are actuated at 30 and 60 degrees. The dropper is loaded from the top and relies on gravity to fall through aligned holes in the housing and the barrel. These holes are rotationally offset to ensure each dropper is deployed at the correct location. The barrel itself is divided with internal ridges that act as locking detents, preventing unintended release during rotation.

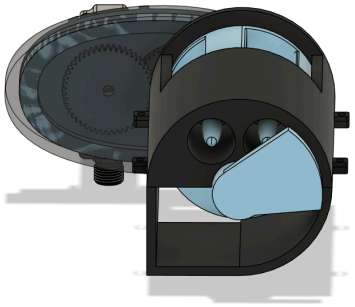


Figure 4. 2025 torpedo and dropper design

(v) Computer

The onboard computer is the NVIDIA Jetson Orin Nano Developer Kit, responsible for running all mission-critical software, including the object detection algorithm and mission decision making. Our team chose the Jetson over alternatives like the Raspberry Pi due to its significantly faster I/O capabilities and powerful tensor cores, which are essential for real-time computer vision tasks. Additionally, its compatibility with our software stack allowed for a more straightforward integration.

B. Software

(i) Mission Planning

Instead of using a finite state machine this year, we decided to control the sub's decision making through the use of a behavior tree. Titled "SHRUB," or the Software for Handling and Regulating Underwater Behavior. The benefits of using a behavior tree over a finite state machine include the incorporation of a "blackboard" for shared memory in decision-making and the ability of the sub, as a moving object, to be more reactive to the specific environments and conditions the sub finds itself in. The root tree is moderated by respective nodes and iterates through each of the specific subtrees controlling the sub's behavior through each of the course's respective tasks, until reaching the end of the course. With more defined features, behavior trees provide more advanced control flows, enabling the accommodation of more complex environments.

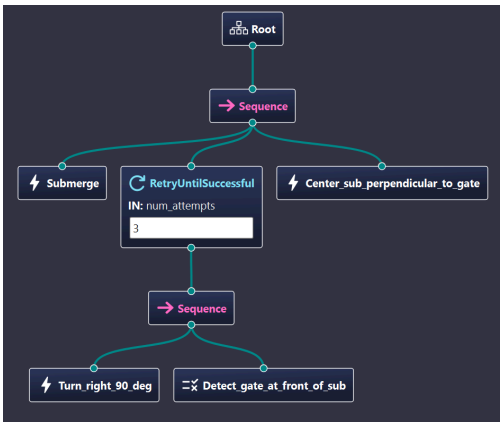


Figure 5. Behavior tree for ocean cleanup

(ii) Computer Vision

This year's computer vision system was significantly upgraded with the integration of the advanced ZED 2i stereo camera, creating a self-contained perception and localization solution. This architecture combines a YOLOv8 model for object detection with the ZED 2i's high-fidelity spatial data, including depth maps and 6-DoF positional tracking. This fusion enables real-time, 3D-aware detections, crucial for both navigation and precision tasks. To overcome the challenge of limited training data, we implemented a targeted data curation pipeline using PyTorch, applying augmentation to improve model robustness. The ZED 2i's high-quality stereo imaging also helps mitigate aquatic distortion issues experienced in past seasons, reducing the need for external post-processing and resulting in a more accurate and reliable perception system.

(iii) Architecture and Navigation

To enable efficient and autonomous operation underwater, our system relies on a coordinated system of processes, each employing specialized algorithms for navigation and task execution. Communication between these processes is handled using the publisher and subscriber model provided by ROS2 [4], ensuring modularity, and this is further expanded upon in the following subsection. For interfacing with the low-level hardware, we used MAVROS to bridge ROS with MAVLink, allowing communication with the Pixhawk PX4 flight controller, as demonstrated in figure 6. This architecture allows data exchange across all components from high-level mission planning to low-level motor control.

For navigation and localization, we used the IMU integrated within the ZED 2i cameras along with computer vision-based object detection to estimate the AUV's position within the environment. Leveraging data from the IMU, the implemented proportional integral derivative (PID) control loop dynamically adjusts the pulse-width modulation (PWM) signals sent to the thrusters, enabling more accurate and reliable movement, improving overall stability and precision.

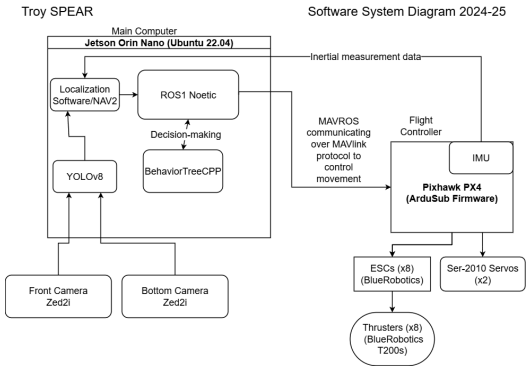


Figure 6. 2025 overall software diagram

Once an object is detected, a three-stage process is followed for effective engagement.

1. Initial Detection - The AUV explores the designated area of the pool, following the behavior tree decisions, until it recognizes an object significant to the current task.

2. Repositioning - Upon detecting a target, the AUV reorients itself to center the object within the frame of the front camera and aligns itself at the same depth level.

3. Distance Estimation - The ZED 2i stereo camera provides 3D depth data, which is leveraged to estimate the relative positions. This spatial information is then fed into the behavior tree and returns a resulting action.

(iv) ROS Nodes

ROS (Robot Operating System) is a modular framework that allows different processes, or nodes, to communicate efficiently using a publisher/subscriber protocol. This architecture allows integration of different subsystems within the AUV. The main 4 nodes are as follows:
- Detection: runs object detection, generates 3D bounding boxes, and publishes object locations.
- Localization: determines the AUV's relative position from the origin and supports loading or generating maps of the pool environment, with the help of NAV2.
- Behavior Tree: handles high-level decision making based on mission objectives and sensor inputs.
- Control: converts decisions from behavior tree into motion commands and transmits them via MAVLink to the Pixhawk.

C. Electrical

The electrical subteam made its debut in the 2024-25 season, and due to its new nature, was focused primarily on redesigning the whole electrical layout of the submarine, which, up to this point, had been based off of the wiring of the Blue Robotics BlueROV2.

(i) Batteries

In past seasons, relying on a single battery led to frequent swaps, increasing failure risk and introducing unnecessary downtime. To extend runtime, we designed a battery combiner board that merges two identical LiPo batteries into a single parallel power source. By connecting the batteries' positive and negative terminals through low-resistance MOSFET protected paths, the board balances current draw while maintaining a consistent voltage. Built-in diodes prevent reverse flow between packs, and onboard fuses ensure safety in the event of imbalance or failure. This configuration effectively doubles available capacity without increasing system voltage, providing longer mission durations without altering the existing power infrastructure.

(ii) High Power

The high power board, in figure 7, serves as the central distribution hub for the AUV's moving parts, namely the servos and thrusters. It efficiently regulates and routes power from the main 14.8V LiPo batteries to the downstream modules. It features two buck converters configured to provide two independent 7.125V rails, accommodating the lower power requirement of the servos, and uses a separate buck regulator to supply a stable 12V rail for the thrusters. An earlier iteration of this board had problems with inrush current, so to mitigate such electrical surges, the board incorporates a 555 timer-driven precharge circuit that actuates a high-current relay via a MOSFET. A diode array provides overvoltage protection and screw terminals are mounted for easier troubleshooting. The kill switch directly cuts off power to this board, disabling propulsion while preserving power to the onboard electronics, preventing the AUV from moving without forcing a full system reboot.

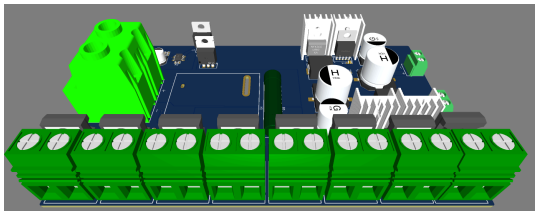


Figure 7. High power board 3D render

(iii) Low Power

The low power board supplies regulated power to the static components, including the onboard Jetson and tether interface board. Powered from the same central battery system as the high power board, it uses a dedicated buck converter to deliver a 5V rail. By separating low voltage logic from high current propulsion, this design improves electrical noise isolation and protects sensitive components during power transients. Crucially, this board remains active when the kill switch is triggered, meaning that the software stack does not need to be reset, significantly improving cycle times.

(iv) Kill Switch

The kill switch is a waterproof switch mounted at the rear of the AUV for easy access. It is run in line to the high power board, which, when activated, effectively disables the motors while continuing feeding low power, which includes the onboard computer. This design improves testing efficiency by eliminating the need to reboot the computer each time our submarine is killed. Especially in a competition setting–where testing time is limited–this feature significantly reduces downtime and cycle time between tests, giving our team more time to perform tasks and score points.

III. Experimental Results

A. Cameras

The first critical component tested on the AUV was the vision algorithm, which serves as the backbone of the AUV's object detection system. Given its central role in task execution during competition, early validation was necessary. The algorithm is designed with a flexible framework that allows for switching between different image databases with a single line of code. The models were integrated with the ZED 2i cameras, which made integration seamless.

Figure 8. ZED 2i camera object detection

Once the vision model was complete, we started testing by manually inputting images into the system. These included baseline images such as underwater pool shots, as well as homemade markers and reference pictures modeled after competition sets. After confirming the model's reliability, we repeated a similar process post-integration, being wary of the real-time detections performance.

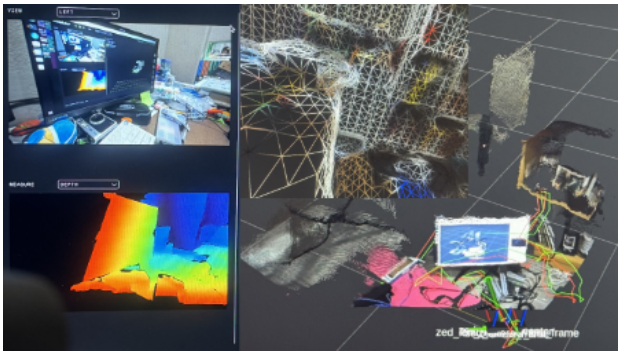


Figure 9. Localization software and depth testing

We tested the localization and depth perception capabilities of the cameras and their integrations, hoping to ensure reliable usage through testing and calibration. By leveraging the ZED SDK, we evaluated real-time positional tracking and 3D mapping accuracy in both simulated and controlled environments.

For testing movement and testing without having to use the AUV, we used a simulator powered by Gazebo. By using an emulated flight controller, Ardupilot's SITL, we were able to send commands, sending feedback from the simulator to the virtual AUV in a loop.

IV. Acknowledgements

Our team's development would not have been possible without our mentors, Cdr. William Lauper and Lt. Roger Fronek, and support from our generous sponsors. We would like to thank the following organizations for sponsoring our team: Troy High School NJROTC, Blue Robotics, Blue Trail Engineering, Onshape, Armed Forces Communications and Electronics Association, Navy League of the United States Inland Empire Council, and the Navy League of the United States STEM Institute. Lastly, we would like to thank Cornell AUV and AV Botz for helping us better understand the competition and how to set up our AUV.

V. References

[1] “Resources.” *RoboSub*, Available: robosub.org/resources/ (2023/06/16).

[2] Akkaynak, Derya. “Sea-Thru.” *Derya Akkaynak*, Available: www.deryaakkaynak.com/sea-thru (2023/06/16).

[3] Supeshala, Chamidu. “YOLO v4 or YOLO v5 or PP-YOLO?” *Chamidu Supeshala*, Available at: https://towardsdatascience.com/yolo-v4-or-yolo-v5-or-pp-yolo-dad8e40f7109 (2023/06/16).

[4] Iranmehr, Masoud. “YOLO v4 or YOLO v5 or PP-YOLO?” *Masoud Iranmehr*, Available at: https://pypi.org/project/mavros-python-examples (2023/06/16).

[5] “BlueROV2 ROS Simulation” *UUVControl*, Available at: https://github.com/UUVControl/bluerov2 (2023/06/16).

[6] “BlueROV2 ROS Simulation *UUVControl*, Available at: https://github.com/UUVControl/bluerov2 (2023/06/16).

[7] “What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector,” Muhammad Yaseen, Available at: https://arxiv.org/abs/2408.15857 (2025/4/23).

[8] “ZED SDK,” stereolabs, Available at: https://github.com/stereolabs/zed-sdk (2025/05/24).

Appendix A: Component Specifications

Component	Vendor	Model/Type	Specs	Qty	Total Cost
Submarine	Fabricated in-house	Custom	1010 Aluminum extrusions Custom silicone O-Ring 3d printed components (PLA) 6061 aluminium	1	$750.00
Battery	Turnigy	Turnigy Graphene Panther 5000mAh 4S 75C	-	4	$206.08
Battery	Blue Robotics	Lithium Polymer Battery (14.8V, 10Ah)	-	2	$480.00
CPU	Nvidia	Jetson Orin Nano Developer Kit	GPU and 4 GB of RAM	1	$429.97
Camera	StereoLabs	ZED 2i Camera	120mm Stereo Baseline, Built-in IMU, barometer, & magnetometer 1080p @ 30fps	2	$998.00
3D Printer Filament	Elegoo	ELEGOO PLA Filament Black & White	1.75mm 2kg	1	$28.69
Battery Bags	Amazon	Tenergy 2 Pack, Fire Retardant Lipo Bags	-	1	$11.99
Kill Switch	Amazon	Hmknana IP67 Waterproof Inline Cord Switch	IP67 12V-24V 20A	1	$14.99
Tether Spool	Amazon	Fathom Spool	-	1	$784.00
Controller	Amazon	Logitech F310 Wired Gamepad Controller	-	1	$15.19
Torpedo Propulsion	Amazon	SP 9706 Mechanical Compression Spring Stainless Steel Extension Spring	1/2-inch by 1-1/2-inch (6 Pieces)	1	$6.99
Algorithms: vision	-	-	YOLOv8 Object Detection, ZED SDK	-	-
Algorithms: localization and mapping	-	-	ZED SDK, Nav2	-	-
Open source software	-	-	ROS Melodic, OpenCV, YoloV8, PyTorch, Pymavlink	-	-

Team size (number of people)	-	-	19 persons	-	-
Expertise ratio (hardware vs. software)	-	-	6 Mechanical 4 Electrical 9 Software	-	-
Testing time: simulation	-	-	14 hours	-	-
Test time: in-water	-	-	7 hours	-	-
Programming languages	-	-	Python, C++	-	-