TEAM SIMPLEXITY 2026 - *High Tide*

Lino Maricic, Thomas Fernandez, David Lu, Bo-Zhi Huang, Aiden Lee, Thejas Vakamudi, Liam Sagi

Abstract—Team Simplexity is a continuing private team composed of 6 passionate students from 3 high schools and one community college in San Diego, California. This season, we continued to optimize our sub design farther into convenience and prevention through reliabilityour High Tide vehicle from last year, iterating upon our state-of-the-art software, electronics, and mechanical systems. Our main AUV, *High Tide* incorporates the Navigator Flight Controller to process perspectives on both sensors and thrusters with the Jetson Nano and ZED Box Mini to process vision. Narrowing a focus on modularity and efficiency, *High Tide* was built to perform a multitude of tasks and provide a consistent modular foundation for future development, while *HydroX 2.0* continues to be a reliable testing platform for mechanical and software systems. Optimizing our sub design farther into convenience and prevention through reliability. Software believes if we create a robust computer vision pipeline for the sub, then we have localization and therefore a successful AUV.

Competition Strategy

Design of our High Tide Vehicle

High Tide is designed to be a highly modular and expandable AUV platform. The main structure of the frame is composed of 4 hollow steel tubes to allow easy component mounting and reconfigurability of electronics, sensors, and attachments, as well as the vehicle dynamics, such as centers of gravity, buoyancy, and thrust vectors. Unlike traditional AUV frame construction with a fixed size and geometry, our steel tube construction enables rapid iterations as the main structure of the frame does not need to be modified to make design changes; only the 3D printed clamping components need to be replaced or shifted when a design is updated. We had two goals for reducing hydrodynamic drag on the High Tide vehicle. Number one: reduce the frontal area, and number two: reduce the drag coefficient. With the components mounted in line with each other, the frontal area of the vehicle is minimized, and the hydrodynamic drag is reduced.

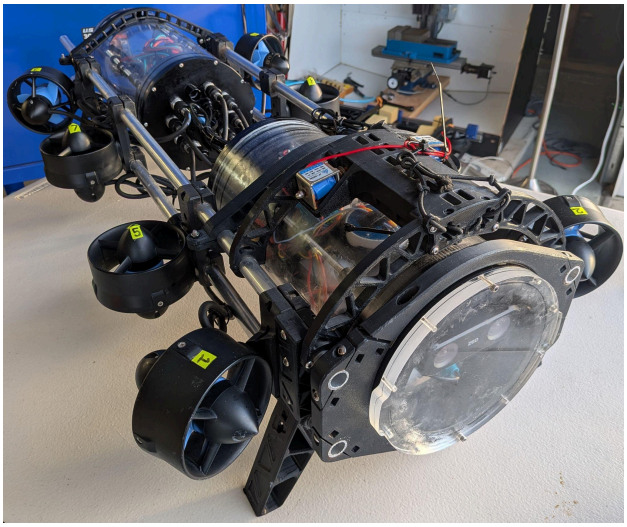

Figure 1: Physical AUV Hightide, with our custom CNC in background

Additionally, while thinking of ways to further optimize the efficiency, we realized that the majority of the AUV's movement will be forward and backward; strafing side to side is only required when aligning with certain tasks. Thus, mounting the thrusters at the standard 45° angle is non-optimal for efficiency. We opted for a thruster angle of 25°; through our calculations, we found that this thruster angle increases the forward-backward efficiency by 28.2% while

still leaving plenty of power for precise horizontal movements.

Furthermore, our overarching competition strategy focuses on reducing the risks associated with increasing system complexity. AUV's for example, operate in severe environments where one failed sensor can lead to the total failure of an entire mission run. To help mitigate this, emphasized reliable degradation of sensors over brittle, complex pipelines. We allocated far less time developing experimental, high-risk capabilities and instead engineered with safe fallbacks to ensure viability. A good example of this is how we managed our navigation. We recognized that while VSLAM is very precise, it is typically not reliable in a featureless pool environment. Therefore, we designed a 3-tiered navigation system:

- Tier 1 Visual Simultaneous Localization and Mapping (VSLAM): Spatial memory and full feature tracking (Requires >80% ZED tracking confidence)
- Tier 2 Visual Inertial Odometry (VIO): Falls back to pure visual-inertial odometry if spatial memory fails or if ZED confidence drops to 30-80%
- Tier 3 (Dead Reckoning): If visual data is stale (>1.0 seconds) or confidence drops below 30%, the system ignores the camera for navigation entirely and navigates via the flight controller's own INS and our FOG.

This trade-off allows the vehicle to not abort the run if it becomes temporarily blinded by sunlight or even the lack of a landmark to track onto. It will smoothly degrade its navigation tiers and continue on a slightly less precise course rather than failing entirely.

Task Execution

Our rigorous strategy allows for the maximum number of points earned within the opening of the run. When the vehicle's behavior tree initiates its PreDive sequence, it starts by looking around for the gate following the coin toss. Once the vehicle has achieved a stable position, we will guide the vehicle through the course in the following order: gates, slalom, bins, torpedoes, octagon, and return home. To further ensure reliability and maximize our point yield we use a dynamic time allocation heuristics to run concurrently with the mission tree. Instead of trying to complete a task until failure, our strategy compares points per second efficiency for each obstacle based on the assigned point value and the time elapsed. We have set script, weighted time-outs for each sub behavior. If the vehicle is having problems completing a complex task due to its time threshold, the behavior tree will purposely abort that node and allow the vehicle to proceed to the next task. This allows us to conserve our run time window and put our efforts where it counts.

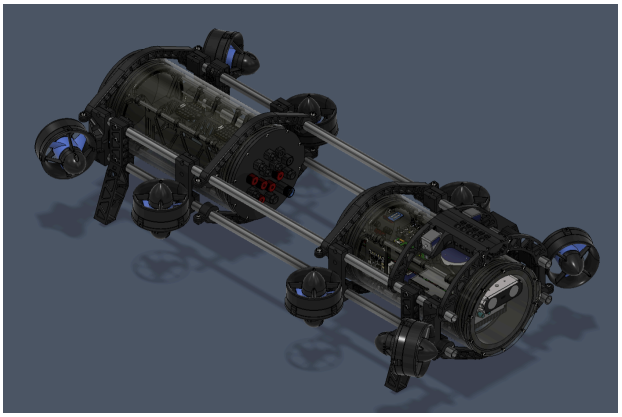

Figure 2: CAD model of our completed High Tide AUV

Software

This year, we made the decision to rewrite our software stack from an undiversified design to a distributed architecture based around ROS2 Humble. The majority of the processing is done on our companion computer, the ZED Box Mini, with low-level and flight controls being done by our flight controller, the Blue Robotics Navigator. With ROS2, we leveraged the Data Distribution Service (DDS), which allowed for zero-copy transfer of stereo video from our ZED X Mini camera, saving huge amounts of CPU time and possible interprocess communications bottlenecks. Our system is broken up into 7 decoupled ROS2 packages, each with their own purpose: tests, interfaces, drivers, perception, localization,

navigation, control, and mission.

Design Strategy

Mechanical and CAD

High Tide utilizes 2 waterproof 6-inch cylindrical acrylic enclosures from Blue Robotics. To maximize the available space for our electronics, we prototyped several methods for mounting them using 3D-printed parts to achieve a highly modular and stable structure. We created the foundation of our enclosure by designing a cylindrical piece that runs the length of the enclosure. The enclosure structure is bolted directly to the connector lid, so we can pull out all the electronics inside the main enclosure as a single unit without any hassle. The structure of the electronics is 3D-printed with acrylonitrile butadiene styrene (ABS) to be able to withstand high temperatures without losing rigidity. We optimally positioned our hardware inside the enclosure for ease of use when plugging in Ethernet, power, or other connectors. Figure 2 displays the CAD model of the 3D-printed structure of the enclosure.

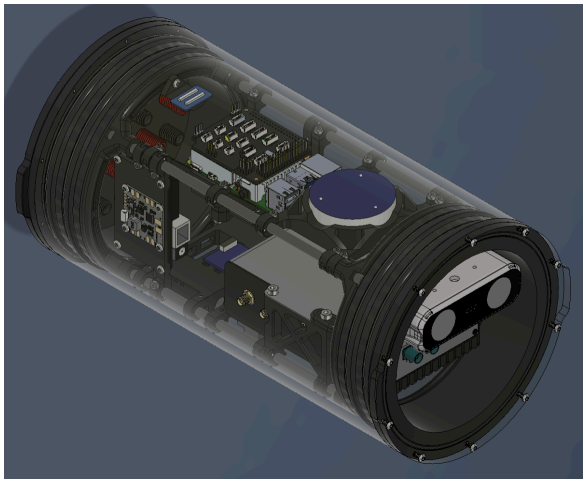

Figure 3: CAD model of the High Tide Vehicle's main enclosure

During experimental testing, we experienced thermal heating as a result of the battery and ESC being in the main enclosure. Therefore, we designed and built our own custom T6-6061 aluminum alloy enclosure for the ESCs and custom-built the battery enclosure of PVC pipe and custom-machined aluminum end caps. This design, however, caused a large discrepancy in buoyancy between the front and back of *High Tide*. Additionally, in more experimental testing, we found that the ESCs, when not running at excessive power levels, remain relatively cool without a heatsink. Therefore, we replaced the battery enclosure, ESC enclosure, and ABS ballast with a second, nearly identical acrylic capsule to the front, effectively evening our displacement. Our battery and ESC mounts are extremely modular, and the battery can be replaced even faster than last year, by removing the blank endcap and undoing a velcro strap.

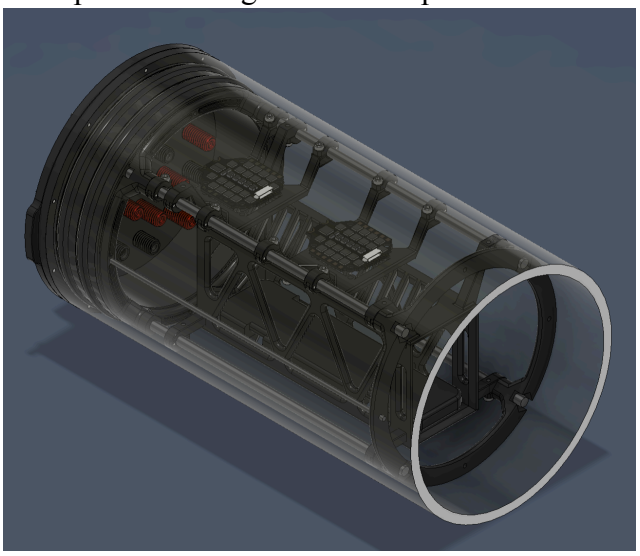

Figure 4: CAD of our custom battery and ESC mounting system. Top: Battery Enclosure, Bottom: ESC enclosure.

Solenoid Usage

While most other teams use servos to power their various mechanisms, we opted for solenoids for our torpedo launcher, dropper, and claw mechanisms. For some tasks, servos are preferred since they offer precise, variable positioning. However, solenoids are incredibly reliable for tasks that require only an on/off or open/closed state. Additionally, solenoids are highly reliable, cost-effective, and more resistant to water leaks.

Torpedo Launcher

Our torpedo launcher uses a bungee-loaded design with a solenoid-actuated release. The

torpedo itself has 4 fins with a minimized circular cross-section to reduce unnecessary hydrodynamic drag. Both our torpedo and launcher went through several iterations. For instance, by using computational fluid dynamics testing and physical testing, we improved the reliability of the torpedo and found that it travels straighter for a longer distance than past designs. We reduced the impact of the frontal surface area of the torpedo by smoothing out connections from the fins to the body of the torpedo, making it more hydrodynamic.

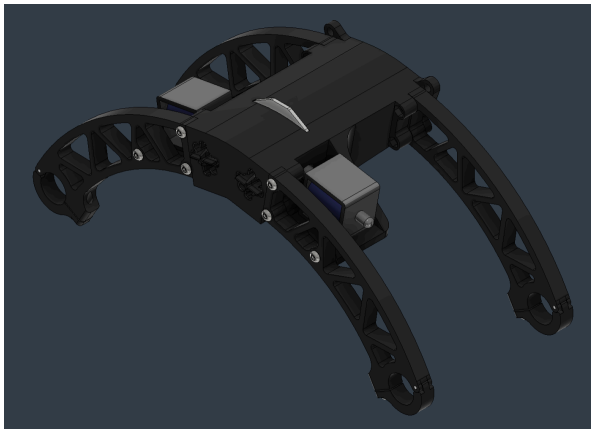

Figure 5: Torpedo Launcher, with mount, solenoids, and safety pin

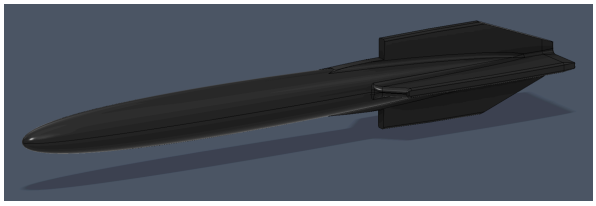

Figure 6: Torpedo Design

Dropper Mechanism

Our new dropper mechanism has a much smaller fo compared to last year, which minimizes drag and al the attachment of more peripheral devices to the sul dropper uses solenoids to pull back plates, releasing weights ~~with 100% reliability over 500+ underwate~~

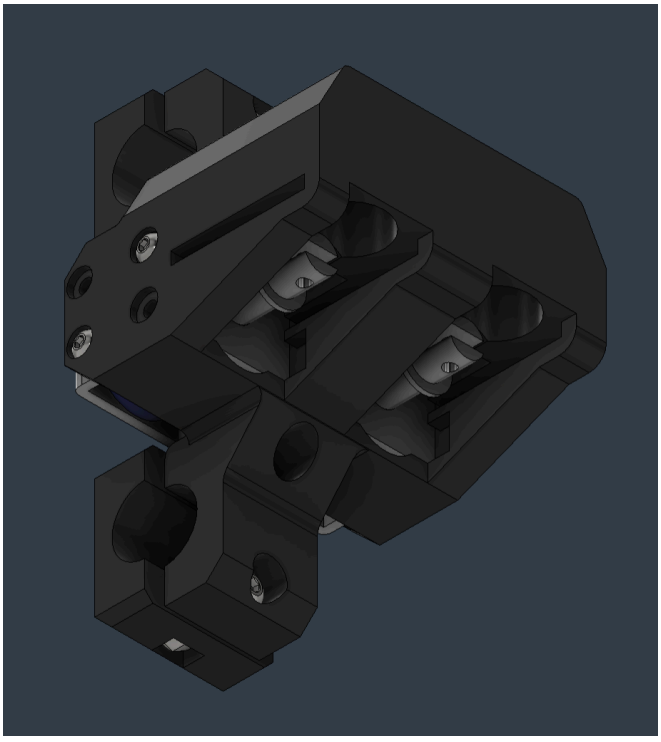

Figure 7: The dropper design. The two modules are mounted onto a plate connected to the sub's main body.

Communication

We custom-designed a tether box to house the *RAKwireless LX200V30* communication adapter. This module allows us to communicate with the AUV through a two-conductor tether, providing more freedom of movement while at the same time extending our maximum operating range to 300 meters.

Magnetic Kill Switch Design

We use a kill switch that disconnects the PWM signals from the Navigator to the ESCs. The switch is made up of a board connected to the PWM signals from the flight controller to the ESCs and the reed switch connected to the board. The reed switch forms a closed connection when an external magnet is affixed outside the AUV; this allows for the kill switch to activate by pulling off the magnet.

Solenoid Control Board

Since this year, we have modified the torpedo launcher, the dropper, and the claw mechanism to use solenoids. We made a custom board that can

provide current to each solenoid individually based on I^2C signals sent from the flight controller.

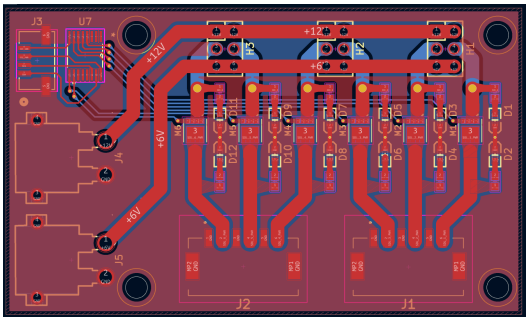

Figure 9: Solenoid Control Board PCB layout

On our HydroX 2.0 vehicle, we angled 4 thrusters towards the centerline instead of straight forward to allow the robot to move in the x-y plane (using strafing) without physically rotating. Due to limited battery capacity, we decided to implement this change so we can input less current into the thrusters to move the UAV forward, because less force is being wasted pushing against the motor across, creating no net force.

Electronics

The *Blue Robotics* T200 thrusters propelling the Robosub are a three-phase brushless motor optimized for underwater navigation. All 8 thrusters are controlled by two *Hobbywing* 4-in-1 ESCs, which support 60A of current on each channel at up to 20V.

The Navigator controls all the ESCs and the thrusters for autonomous movement. The Navigator communicates with the ESCs using the DSHOT protocol. The message then travels from the main enclosure to the ESC enclosure via CANBUS.

The Navigator is equipped with the *Sony* IMX577-AACK Sensor Module and *OMNIVISION*'s OV9282 used for image processing. The Navigator is connected to an MS5837 depth sensor, which can measure up to at most 30 bar (300m depth); the Navigator Nano is also connected to an RM3100 magnetometer and InvenSense's 6-axis IMU to obtain the Robosub's heading, acceleration, and velocity.

All of the sub's telemetry is able to be communicated using Ethernet via a wired tether to a computer outside the submarine in real time for testing.

All the electronics onboard are powered by a 10Ah, 4S *Lumenier* LiPo battery. The AUV draws an average current of 30A, so with its 10Ah capacity, the LiPo battery is able to power the sub for 20 minutes or two rounds between each charge.

Software

The software design methodology utilized is structured in a top-down philosophy such that every decision directly correlates to a requirement of the competition.

- Perception: TensorRT-Optimized YOLOv8 and Hardware Decoupling
  Why: Traditional methods of object detection (ie using cascade classifier and color-thresholding) often fail to perform under unpredictable lighting conditions like the ones in water. In contrast, deep learning networks provide robustness and accuracy, but standard models are generally not fast enough for real-time applications. Additionally, our limited compute limits our choice for such heavy models.
  How: We integrated YOLOv8 to achieve 30+ FPS on edge, with a custom TensorRT pipeline to utilize as much compute from our Jetson Nano for AI. We also used PyCuda to ensure zero-latency transfers between the CPU and GPU by allocating page-locked host memory and device memory.
- Localization: Sensor Fusion with EKF
  Why: Visual odometry alone can lead to drift due to a lack of visual features in a

scene.
How: An EKF is utilized to fuse our INS, FOG, and the ZED X Mini, which is generating 3D visual odometry data. The system evaluates the reliability of the visual odometry in real-time. The pose covariance from the ZED Odometry topic is being extracted within the navigation degradation node and the average value of the diagonal elements of the covariance matrix represents positional covariance and is used as a measure for mapping a dynamic tracking confidence score (0.0 to 1.0). This score is used to determine the degree of weight applied to the visual odometry data within the EKF to avoid significant weight being applied to faulty visual features in the calculation of the sub's pose.

- Mission Management: Behavior Trees (Py Trees)
  Why: Classically, finite state machines (FSMs) are used for state and behavior transitions. However, as mission complexity increases, FSMs lead to more complex and difficult logic to debug. We need a reactive architecture that can quickly abort any failing behaviors or tasks without having to traverse complex layers of state to do so.
  How: We utilized the Py Trees library to command our mission, with the blackboard feature helping us separate the behavioral logic from the asynchronous callbacks of ROS2. We designed our blackboard with subscriptions that push updates to the blackboard at 10Hz. This guarantees that when a behavior tree ticks, all of the discrete behaviors have the same atomic state that was taken at that same instant. By doing so, race conditions are prevented and ensure deterministic decision-making across parallel subtrees.

Testing Strategy

Having limited pool access, we developed a 3-phase testing strategy that separates hardware from software before field testing.

- Phase 1: Offline Component and Unit Testing
  Utilizing the pytest framework, our test packages are utilized to run as part of our continuous integration process before merging our code. These tests include checking things such as bounding box intersections, integrators, etc. Next, Node tests load an individual ROS2 node and attempt to provide mock topic data to confirm that it will be routed appropriately,
- Phase 2: Software in the Loop (SITL) Simulation
  We test our missions using a Gazebo hydrodynamics simulation, allowing us to prototype and improve on our mission logic and algorithms in-silico without the time and logistical constraints of field testing. This also included our CFD simulation.

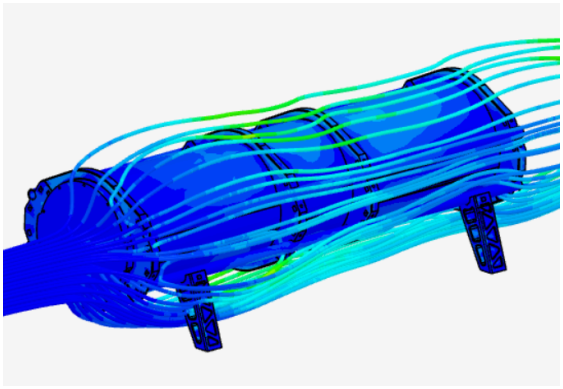

Figure 10: CFD Simulation of Hightide against the flow of water

- Phase 3: Pool testing
  To make the most out of our limited pool time, we do not run the complete master behavior tree. Instead, we created a series of isolated integration test scripts to specifically verify the operation of peripherals in tandem with software. By limiting our test scope to test each of the sensor/software interfaces individually prior to attempting complex, multi-task missions, we can reduce the debugging scope when integrating full software stack.

Acknowledgements

We express our deepest gratitude to the individuals, organizations, and sponsors whose unwavering support and contributions were instrumental in our RoboSub 2025 season. Their expertise, resources, and encouragement enabled our team to design, build, and test our autonomous underwater vehicles.

Our mentors played a pivotal role in guiding our team through the complexities of this project. Particularly, our lead technical advisor, Bruce Meagher, and our head financial advisor, Paul Fernandez. Coach Bruce dedicated countless hours assisting us with mechanical and electrical design, as well as our autonomous navigation software. Coach Fernandez provided us with tremendous support in terms of financial organization and trip planning.

We are also immensely grateful for Qualcomm, General Atomics, Blue Trail Engineering, Blue Robotics, Motovisio, and RISE for their generous support.

Lastly, we also acknowledge the resources provided by each of our parents, enabling us to access our engineering lab and testing pool. The collaborative effort of our team members, mentors, and sponsors has been pivotal in our RoboSub 2025 season so far. We are profoundly thankful for their contribution and dedication.

References

1. Blue Robotics Navigator Flight Controller *https://bluerobotics.com/store/comm-control-power/control/navigator/*
2. TDK. *ICM-42688-P Datasheet* *https://media.digikey.com/pdf/Data%20Sheets/TDK%20PDFs/ICM-42688-P_DS_Rev1.2.pdf* (InvenSense 2020)
3. Blue Robotics Ardupilot SITL model https://github.com/ArduPilot/ardupilot/blob/master/libraries/SITL/SIM_Submarine.h#L69
4. RAKwireless Wislink 2014 https://store.rakwireless.com/products/lx200v30-plc-homeplug-av-module
5. PNI. *RM3100 Breakout Board.* https://www.pnicorp.com/product/rm310 0-breakout-board/ (PNI 2018).
6. Riedl, Max J. *Optical Design Fundamentals for Infrared Systems*. SPIE Press. pp. 40 (2001).
7. PX4. *ROS 2 User Guide (PX4-ROS 2 Bridge).* https://docs.px4.io/main/en/ros/ros2_comm.html (2021)

Appendix A

Components	Vendor	Model/Type	Specs	Number	Status
Frame	ePlastics	ePlastics Cut Enclosure	Acrylic	1	Installed
Waterproof Housing	Blue Robotics	6" diameter acrylic with end caps	Acrylic	1	Installed
Thruster	Blue Robotics	T200		4	Installed
Motor Control	Hobbywind	60A 4-in-1	60A / 6S/ 4 Ch	3	Installed
Propellers	Blue Robotics	T200		4	Installed
Battery	Lumenier	LiPo Battery	10000Ah 4s 25c	1	Installed
Control Computer	Blue Robotics	Navigator Flight Controller		1	
Internal Comm Network	Ethernet, I2C, SPI, RS-232, USB, CAN				Installed
External Comm Network	1Gbit Ethernet, WiFi				Installed
Ethernet Current Converter	Qualcomm	QCA7005 chip	QFN 68 pins	2 (1 on each end)	Installed
Compass / Magnetometer	Blue Robotics	MMC5983MA	±8G FSR / 18bits operation 0.4mG total RMS noise / Enables heading accuracy of 0.5º / Max output data rate of 1000Hz	1	Installed
IMU	Multiple	Included on Navigator	3-axis accelerometer w/ gyroscope	5	Installed
Vision	Qualcomm	Sony IMX577-AACK Sensor Module	4K, 30 FPS	1	Installed
AI Module	NVIDIA	NVIDIA Jetson Nano Module	128-Core NVIDIA Maxwell™ GPU Quad-Core	1	Installed

			ARM® A57 CPU 4 GB 64-Bit LPDDR4 10/100/1000BASE-T Ethernet		
Algorithms: Vision	OpenCV				In Use
Algorithms: Localization and Mapping	SLAM				In Use
Algorithms: Autonomy	PX4, ROS2 OpenCV				In use
Open Source Software	Yocto, Ubuntu, C++, Python, JavaScriptC, ROS2				All programs Used throughout the season
Schematic/Fritzing Software	Fritzing, EasyEDA Pro, LTspice				All programs Used throughout season
CAD software	Solidworks, Fusion 360, KiCAD				All programs Used throughout season
Expertise Ratio	4 Mechanical 0 Electrical 3 Software				In use
Team Size	7				In use
Underwater Testing Simulation	Gazebo				Used Throughout Season
Water Test Time	Backyard Pool; Long Course Pool	6ft; 12ft	Started:	20 hrs	N/A