Technical Design Report

Stanford Robosub

Prepared for the Robosub 2024 Competition

Contributors

Software: Selena Sun*, Scott Hickmann, Miyu Kojima, Yuanzhe Dong, Ali Ahmad, Rhea Malhotra, Nathan Elias, Thomas Deng
Mechanical: Lawton Skaling*, Yohannes Aklilu, Ruben Carrazco, Juan Castillo, Marcellina Chang, Vanessa Chen, Arden PBW, Lauren Sibley, Stanley Zhou
Electrical: Vassilis Alexopoulos*, Ryota Sato, Angelina Krinos, Coco Layton, Elijah Kim
Operations: Amy Dunphy*, Rhea Malhotra, Angelina Krinos, Vassilis Alexopoulos, Lawton Skaling, Selena Sun
*indicates team lead

0.1 Abstract

Given that Stanford Robosub was recently formed in February 2024, our strategy for the 2024 competition is based on reliability and simplicity focusing on three key tasks: Rough Seas, Enter the Pacific, and Hydrothermal Vent. Our goal for this competition is to compete with platform hardware that performs well on the aforementioned tasks, which can then be easily built upon in future years for increased complexity. Our targeted approach to task completion is informed by the belief that a high level of execution within a finite number of tasks is better than lackluster execution across more tasks. This report discusses the design decisions made to integrate software, mechanical, and electrical goals into Robosub's constrained underwater environment, as well as how those goals align with our broader competition strategy. We also discuss testing methodologies that have helped us prepare and iterate on our vehicle in preparation for Robosub 2024.

0.2 Acknowledgements

Participating in the Robosub 2024 competition has been a collaborative effort that extends beyond the core team. We would like to express our gratitude to the following individuals and organizations for their invaluable support:

- **Sponsors**: Stanford Student Robotics, Stanford Robotics Lab, Movella, Stanford Moonshot Club
- **Stanford Student Robotics**: Andy Tang, Michal Adamkiewicz, Amy Dunphy, and Archer Date for their wisdom and logistical leadership.
- **Lab64 @ Stanford**: Jeff Stribs for welcoming the club into the space and providing us with a workspace during the school year.
- **TartanAUV**: Micah Reich, Tom Scherlis, Rylan Morgan, and Rishabh Jain for their guidance on founding the club and technical implementation.
- **Professor Oussama Khatib**: For his technical guidance, support in providing course units, and enthusiasm.

Technical Content

1.1 Competition Strategy

When we started in February, our V0 AUV design was informed by the bare minimum hardware necessary to perform well at tasks necessitating precise movement and localization. Given the six-month crunch to build the software, mechanical, and electrical systems entirely from scratch, we decided to avoid overcomplicating our design with additional points of failure – such as torpedoes or environmental manipulation. We hope future Stanford Robosub teams will build upon V0, using it as a solid foundation for tackling these more advanced tasks.

As such, we narrowed our competition strategy down to three target tasks: Rough Seas, Enter the Pacific, and Hydrothermal Vent. Each task involves significant vision and localization requirements, which we prioritized in our system. Sensor fusion of our doppler velocity log (DVL), inertial measurement unit (IMU), and camera systems allows us to locate task markers using a "you only look once" (YOLO) detection algorithm while achieving precise, autonomous movement across our AUV's 6 degrees of freedom. This setup is sufficient for completing the aforementioned tasks.

Several design trade-offs were evaluated to reach our current configuration. Firstly, our camera and DVL are fixed relative to the AUV, with the camera facing forward and the DVL pointing downward. This setup simplifies the autonomy stack by ensuring that camera and velocity measurements are captured in consistent movement frames. However, it restricts navigation to AUV orientations that keep the DVL parallel to the pool floor. This particular DVL was generously provided by Professor Oussama Khatib due to cost constraints, and has a tight field of view that exacerbates the above orientation restrictions.

Additionally, our single forward-facing camera necessitates periodic 360-degree yaws for situational awareness, slowing down course navigation. This also complicates backward movement since there's no rear-facing camera for object detection and avoidance, requiring a 180-degree yaw when the direction of motion needs to be reversed. The absence of additional cameras simplified the mechanical design, as all cables interfacing between the inside and outside of the AUV are routed through a steel plate with wet link penetrators. A back-mounted camera would require replacing this plate with a transparent acrylic and a more custom wiring solution.

From a reliability perspective, reducing system complexity translates to fewer error modes across the subsystems that might need to be addressed. Our approach significantly simplifies the cross-team integration process, as having all components in a unified enclosure reduces the need for extensive coordination regarding waterproofing and mounting. By ensuring that our primary vision and localization components are robust and well-integrated, we were able to focus our limited preparation time on testing and refining these critical capabilities rather than managing the complexities of additional subsystems.

1.2 Design Strategy

1.2.1 Mechanical Design

Hardware design constraints for V0 were largely informed by target tasks and by software needs to achieve those target tasks. Early in the design process, it was clear that having six degrees of freedom for movement was necessary to achieve any level of accurate in-water localization. As such we utilized eight thrusters; four were mounted perpendicular to the pool floor for depth control, and four were mounted parallel to the pool floor, each at 45-degree offsets from one another (Figure 1.1). Mounting with 45-degree offsets allows the AUV to move forward, backward, and strafe with ease.

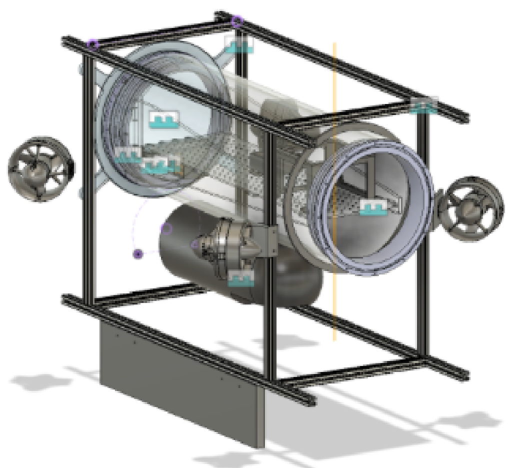

Figure 1.1: CAD Design

For computing, we elected to use a Jetson Orin – this runs all ROS2-related code, logs data from our various sensors, and runs YOLO/controller algorithms. Fast performance is key for movement accuracy, hence why we selected such a heavy option. We also selected an Xsens IMU that minimizes drift, Oak-D cameras for vision and a (generously donated) Teledyne ExplorerDVL.

The frame of the AUV is made from 15mm aluminum extrusion. We elected to trade size and weight for versatility, giving flexibility in where we mount different components to reduce integration and design complexity. The electronics tube is an 8-inch inner diameter acrylic tube from Blue Robotics, 24 inches long. This provides ample space for mounting. The tube is designed with all penetrators routed through the back, enabling the plastic shield to slide on and off without the need to unplug any wires, thereby streamlining maintenance and adjustments.

Two steel beams are attached to the bottom of the sub. This lowered the center of gravity to provide passive stability, and offset the large buoyant force caused by the electronic tube volume. Figures 1.2 and 1.3 show the mass and volume breakdown of the major systems of the sub.

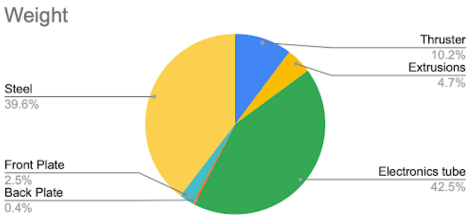

Figure 1.2: Weight Breakdown

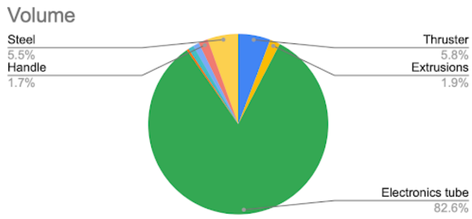

Figure 1.3: Volume Breakdown

1.2.2 Electrical Design

Given the above hardware, we analyzed system peak power requirements before choosing our battery chemistry: a 4S (14.8V) 20C 13000mAh LiPo. This battery's high discharge rate and amperage ensure continuous operation of all subsystems, even during peak power draw. Thrusters are driven directly off this 14.8V line, which passes through a current breaker for short protection. The thrusters are powered by a three-phase system supplied by off-the-shelf ESCs from Blue Robotics. These ESCs take in 14.8V and allocate three-phase power based on a PWM signal provided by a microcontroller.

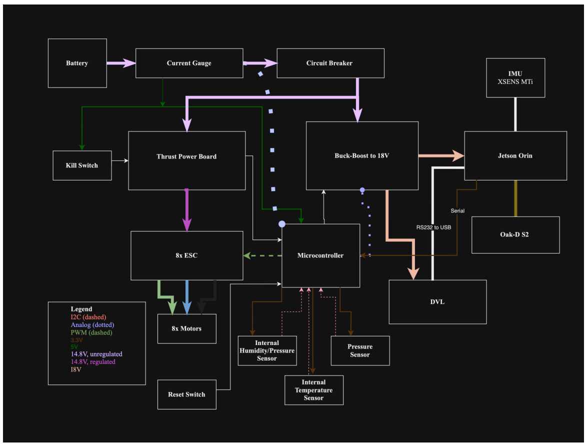

Figure 1.4: Electrical Sub-system Diagram

The Orin interfaces serially with this microcontroller. The basic flow involves the Orin providing thrust allocations to the microcontroller, which are then instantiated on eight PWM lines. The microcontroller also interfaces with I2C-based peripherals, including an inline current/voltage sensor on the battery, an internal temperature sensor to monitor component overheating, an internal pressure/humidity sensor to track leaks, and an internal display for key metrics. These readings are communicated to the Orin via a custom-designed packet structure.

The overall electrical system diagram is shown in Figure 1.4. A custom PCB facilitates power distribution of the 14.8V line via three power MOSFETs. These MOSFETs can turn the thrusters on or off when either the externally mounted kill switch is triggered or when the microcontroller sends a reset command. The PCB was designed with thermal tolerances in mind; it features large thermal reliefs, traces, and a thick, two-layer stack-up.

The microcontroller also has the capacity to perform a soft reset of all onboard systems by power cycling the Orin and the DVL via a series of 18V relays. We opted for a system design incorporating both an Orin and a microcontroller to simplify the software stack by keeping high and low-level code separate. This design also allows for a hardware watchdog in the form of the microcontroller in case of higher-level system failures, adding an additional layer of safety during autonomous operation.

1.2.3 Software Design

In our first year, we focused on building the core pillars of the autonomy stack. We designed the stack to be modular and extensible. Our stack was divided into four modules: localization, perception, mission planning, and GNC. We chose ROS2 as our software framework for its ease of use, integration with the Gazebo simulator for testing, and community support.

Localization

We use three sensors for localization: an IMU (Movella MTi-200), a DVL (Teledyne Explorer DVL), and a depth sensor (BlueRobotics 10m Depth Sensor). The state is a size 13 vector:

$$[x, y, z, \overrightarrow{q}, u, v, w, p, q, r, \dot{u}, \dot{v}, \dot{w}]$$

The state variables are computed as follows:

1. x, y, z: cartesian robot position. The IMU's accelerometer and DVL's measurements are fused via an EKF to give x and y, and depth sensor measurements are added to compute z.
2. $\vec{q}$: quaternion representation of the robot's orientation, found by integrating the IMU's gyroscope angular velocity measurements.
3. u, v, w: linear velocity of the robot, found through fusing the IMU's accelerometer and DVL's measurements via an EKF.
4. p, q, r: angular velocity of the robot, measured directly by the IMU's gyroscope.
5. $\dot{u}, \dot{v}, \dot{w}$: linear acceleration of the robot, measured directly by the IMU's accelerometer.

The IMU and DVL are clock-synchronized before they're read by their respective rosnodes. Synchronization is important because position is found via integration of both the IMU and DVL's velocity estimates, so timing errors will significantly degrade our position estimate over time.

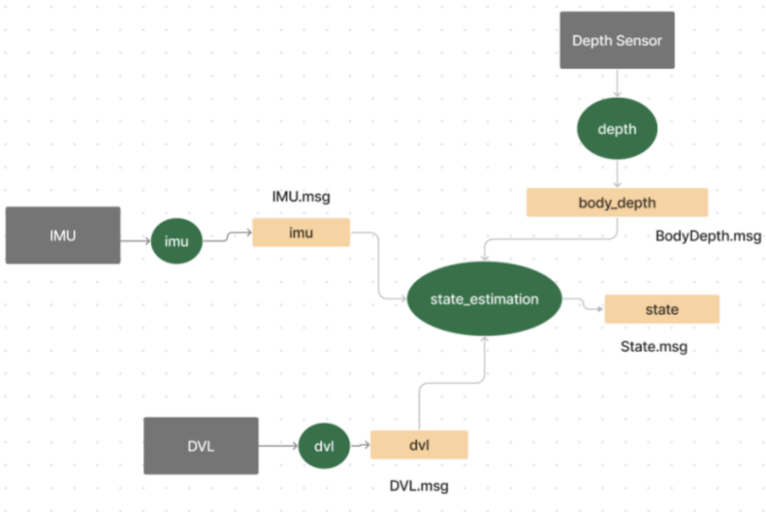

Figure 1.5: Rosnode diagram of the localization stack

Perception

We train a YOLOv8 model for classification of three objects: a red arrow, a blue arrow, and a buoy. Since we only intend to do the Enter the Pacific (Gate) and Hydrothermal Vent (Buoy) tasks, this three-way classification is enough to support the mission. We trained the model as follows:

1. Collected videos of the three objects in a pool
2. Extracted approximately 2,000 frames from the videos
3. Augmented the images (rotation, shear, zoom, position)
4. Labeled the images with bounding boxes
5. Trained a YOLOv8 model

We expect lighting conditions to be different at the competition venue, so we will likely need to re-collect data and retrain the model.

Figure 1.6: YOLOv8 inference on the blue arrow, red arrow, and buoy

The inference results in 2D bounding boxes, labels, and probabilities. The 2D bounding boxes are then mapped to 3D positions, and the estimated locations of the three objects are updated via a 3D Kalman Filter. The global positions of the objects are stored in a map.

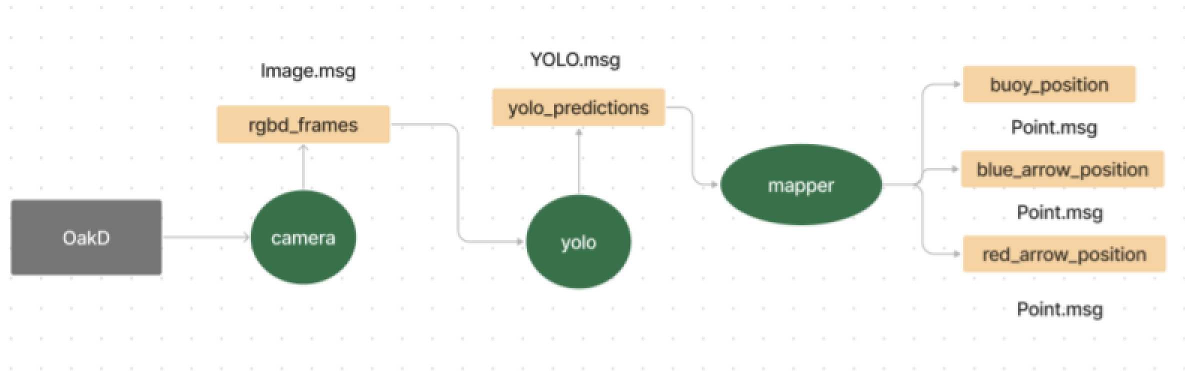

Figure 1.7: Rosnodes diagram of the perception stack

Mission Planning

The Mission Planner is responsible for coordinating the perception, localization, and control stacks to complete parts of the mission. The planner coordinates between tasks with a finite state machine, switching between and retrying tasks depending on the task outcome. The task outcome is decided by each Task Planner. The Task Planner surveys for its object of interest, determines high-level waypoints to complete the task, and evaluates the success of the task attempt.

Separating the Task Planner from a higher-level Mission Planner provides two key benefits:

- The mission can be easily extended to include additional tasks in the future.
- Each task can be isolated for focused testing and development.

Guidance, Navigation, and Controls

Given waypoint objectives from the Mission Planner, each Task Planner is responsible for generating a path that's free from obstacles. For most tasks, we generate a straight line trajectory with a trapezoidal motion profile (positive acceleration, no acceleration, negative acceleration) to reach a full stop at the next waypoint.

The motion planner takes a list of predetermined waypoints defined by desired positions, orientations, and generates a list of intermediate waypoints through linear interpolation. During runtime, a "follow the carrot" algorithm is employed to find the furthest point on the path constrained to some radius around the AUV. This point is passed into the controller which responds by moving the AUV towards the point. Once the AUV has moved, there is a new furthest point, and by repeating this process throughout the entire path, the AUV is able to follow a trajectory to a desired final position.

The controller gets a desired position from the motion planner and is responsible for sending commands to the thrusters to reach the desired position. The node takes an input of a desired state and the AUV's current position and velocity, runs 6 PID loops (one for each DoF), and generates a trajectory of 6-dimensional wrench vectors:

$$[f_x, f_y, f_z, \tau_x, \tau_y, \tau_z].$$

Each wrench is then allocated as forces to the eight thrusters using an allocation matrix, informed by the physical locations of the thrusters.

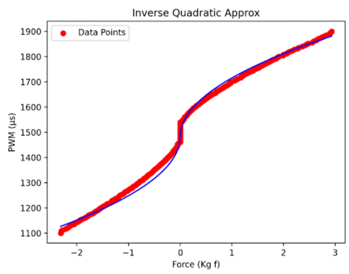

Figure 1.8: Inverse quadratic approximation

BlueRobotics publishes data of its T200 thruster's performance: $f(\text{battery voltage}, \text{PWM}) = \text{thrust (Kg-f)}$, where f is roughly quadratic. Since we want to find PWM given battery voltage and desired thrust, we fit a stepwise inverse quadratic function to the experimental data (Figure 1.8). This PWM value is used to adjust motor thrust.

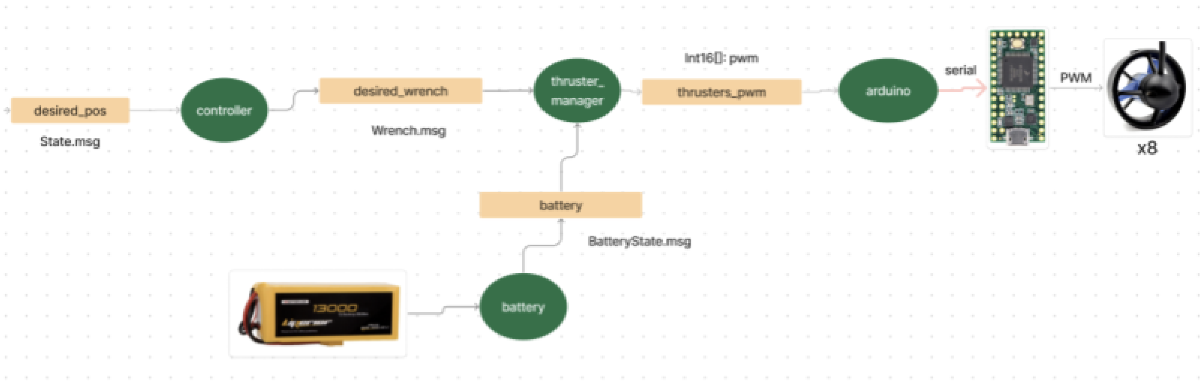

Figure 1.9: Rosnodes diagram of the controller

1.2.4 Testing Strategy

To refine and improve our existing capabilities, we conducted weekly pool tests, each addressing a specific aspect of our design. Our strategy included both in-water and simulation testing to thoroughly validate our system. This methodical testing process allowed us to systematically address and resolve potential issues, ensuring each component functioned optimally within the integrated system.

For water infiltration testing, we conducted initial proof tests by pulling a vacuum and reducing pressure to -27 psi, equivalent to a 19m depth, and then submerged the sub at 10 ft, confirming no water infiltration. Before each test, we reduced pressure by 15 psi to ensure proper electronics cooling. These tests were repeated every time a significant addition was made to the AUV.

Electrical subsystems were extensively tested under peak power draw to evaluate thermal management, transient responses to sudden load changes, prolonged operational demands, and other potential failure modes, both in the lab and underwater. We invested significant effort in optimizing our communication protocol between the Orin and the microcontroller to ensure low-latency thrust allocation. Additionally, we conducted long-duration tests of our tethered communications to the subsystem to ensure stable and reliable data transfer. We also performed rigorous edge-case testing of our kill switch under varying thermal and electrical loads to guarantee safety and reliability across diverse environments.

In simulation, we used the ROS2 Gazebo simulator to test the Mission and Task Planners, the Extended Kalman Filter (EKF), tune PID controllers, and verify waypoint routes before deployment. Additionally, we implemented joystick control to replace the motion planner for control stack verification. For in-water testing, we conducted bench and in-water tests for sensors and control systems according to detailed plans, followed by iterative system integration tests to ensure subsystem integration and robustness.

Appendix A: Components

1	Component	Vendor	Model/Type	Specs	Custom/Purchased	Cost	Year of Purchase
2	T200 Thruster	Blue Trail Engineering		With 3 pin power connector	Purchased	2450	2024
3	Cobalt Series Bulkhead Connector	Blue Trail Engineering		3 Pin	Purchased	460	2024
4	Cast Acrylic Tubing	ePlastics		8.500" OD x .250"	Purchased	438.22	2024
5	Rear Bulkhead	Sendcutsend			Custom	169.34	2024
6	Fathom Spool	Blue Robotics		Standard	Purchased	680	2024
7	Depth/Pressure Sensor	Blue Robotics	Bar02	Ultra high resolution	Purchased	75	2024
8	O-ring Flange	Blue Robotics	8 inch series		Purchased	220	2024
9	O-ring	Blue Robotics	8 inch series		Purchased	40	2024
10	Fathom tether	Blue Robotics		4 twisted pairs	Purchased	375	2024
11	Motor ESC	Blue Robotics	Basic		Purchased	342	2024
12	Vacuum Plug	Blue Robotics			Purchased	8	2024
13	M10 Enclosure vent and plug	Blue Robotics			Purchased	60	2024
14	Pressure relief valve	Blue Robotics			Purchased	28	2024
15	Pressure relief backfill adapter	Blue Robotics			Purchased	11	2024
16	Aluminum Extrusion	Rev Robotics	15mm spacing	1m length, clear anodized	Purchased	135	2024
17	Steel Weights	Alan Steel	Hot Rolled LC Steel	5/8 x 4" Bar, machined in house	Purchased	210	2024
18	8 pin wetlink tetherconnector	Ocean Innovations	DIL8m		Purchased	411.9	2024
19	8 pin wetlink bulkhead connector	Ocean Innovations	DBH8FSS		Purchased	505	2024
20	Wetlink locking sleeve	Ocean Innovations	DLSA-F		Purchased	18	2024
21	Wetlink locking sleeve	Ocean Innovations	DLSA-M		Purchased	18	2024
22	Wetlink splice cable	Blue Robotics			Purchased	20	2024
23	Wetlink penetrator	Blue Robotics	High Compression	7mm cable	Purchased	24	2024
24	DVL	Teledyne Instruments	Teledyne	Custom connector	Donated	unknown	2024
25	RJ45 ethernet cables	Amazon			Purchased	12	2024
26	BME280	Amazon	I2C	Pressure, Temperature, Humidity Sensor	Purchased	8	2024
27	DC Relay	Amazon	24V		Purchased	10	2024
28	RJ45 pin adapters	Amazon			Purchased	10	2024
29	MCP9808	Adafruit	I2C	Temperature Sensor	Purchased	10	2024
30	Wires, various gauges	Amazon			Purchased	100	2024
31	Silicone conformal coating	Amazon	422C-55MLCA 422C		Purchased	20	2024
32	DC jack	Amazon	90 degree, 5.5mm x 2.5mm		Purchased	14	2024
33	Hall sensor	Amazon	NPN		Purchased	8	2024
34	Thrust Power PCB	PCBWay	custom		custom	150	2024
35	Capacitors	Digikey	0.1u		Purchased	5	2024
36	Capacitors	Digikey	1uF		Purchased	5	2024
37	Capacitors	Digikey	1u (electrolytic)		Purchased	5	2024
38	Flyback Diodes	Digikey	SBRT15U50SP5-13		Purchased	5	2024
39	Zener Diodes	Digikey	1SMA5931BT3		Purchased	5	2024
40	Leds	Digikey	IR26-21C_L110_TR8		Purchased	10	2024
41	Connectors	Digikey	1135333		Purchased	40	2024
42	Inductors	Digikey	74477420		Purchased	5	2024
43	Resistors	Digikey	10k		Purchased	5	2024
44	Resistors	Digikey	3k		Purchased	5	2024
45	Resistors	Digikey	10		Purchased	5	2024
46	Resistors	Digikey	100		Purchased	5	2024
47	Resistors	Digikey	0 (100A)		Purchased	5	2024
48	IC	Digikey	INA300AIDSQT		Purchased	15	2024
49	IC	Digikey	SN74LVC1G97DBVR		Purchased	15	2024
50	IC	Digikey	AUIR3241STR		Purchased	15	2024
51	Power Mosfets	Digikey	BSC0702LSATMA1		Purchased	30	2024
52	Mosfets	Digikey	NVTR01P02LT1G		Purchased	10	2024
53	Sense Resistors	Digikey	WSLP59312L000FEA_1, WSLP59312L000FEA_2		Purchased	10	2024
54	Hall effect sensor	Digikey			Purchased	12	2024
55	Magnet	Digikey			Purchased	8	2024
56	XT60 connectors (male and female)	Amazon			Purchased	40	2024
57	Bullet connectors	Amazon			Purchased	20	2024
58	Batteries	Lumenier	LiPo	13000mAh 4S 20C	Purchased	400	2024
59	LiPo safe bags	Amazon			Purchased	25	2024
60	Current/voltage sensor	Lumenier	Holybro Pixhawk PM02 V3 12S Power Module		Purchased	20	2024
61	Microcontroller	Amazon	Teensy 4.0		Purchased	60	2024
62	Depth/Pressure Sensor	Blue Robotics	Bar02 Ultra High Resolution 10m		Purchased	75	2024
63	Switches	Blue Robotics			Purchased	40	2024
64	LEDs	Blue Robotics			Purchased	20	2024
65	Circuit breaker	Amazon		80Amp	Purchased	12	2024
66	IMU	Movella	MTi		Donated	~	2024
67	Cameras	Oak-D	Oak-D Lite + Cable		Purchased	340	2024
68	Computer	Nvidia	Jetson Orin		Donated	~	2024
69	Grease	Amazon	Silicon Lubricating Grease		Purchased	12	2024
70	RS232 cables	Amazon			Purchased	30	2024

References

- Handbook of Marine Craft Hydrodynamics and Motion Control link
- Guidance and Control of Ocean Vehicles link
- Kalman Filter + State Estimation link
- ROS2 EKF Package link
- Printed Circuits Handbook, 7th Edition link