RoboBoat 2024: Technical Design Report

Autoboat, Cornell University
Ithaca, New York, United States

I. Abstract

AutoBoat is a team of undergraduate students representing Cornell University in the RoboBoat 2024 competition. We aim to accomplish the navigation tasks while prioritizing stability, accessibility, and simplicity in our increasingly complex design. This fall we designed and constructed a brand new trimaran boat equipped with a custom PCB electronic system, an expanded sensor suite, increased communication system capabilities, and more robust computer vision and autonomous software.

II. Competition Goals

A. Approach

Our primary strategy for the RoboBoat 2024 competition is to perform all navigation tasks (tasks 1-3, 5 and 8). We directed the majority of our focus towards these tasks as they make up the bulk of point awards and serve as an essential capability for any intelligent ASV.

The design of our boat aligns closely with our competition goals. Given the focus on navigation, stability and maneuverability were prioritized in the hull design process. We also strived to create a design we can build upon in future years. For example, we enlarged the size of the electronics bay within the boat and main deck atop the boat to allow for the expansion of our electronic and robotic systems.

As our second year competing in person, we learned tremendously from our experience at RoboBoat 2023. We were able to identify our weak points and understand the steps necessary to strengthen them. While we aim to optimize our performance at competition, we also hope to show incredible growth each year.

B. Trade-off Studies

A big challenge this year was our shortened timeline due to the new competition dates. This forced us to reconsider the scope and new complexity we initially hoped to introduce. Ultimately, to reduce the amount of time needed for research and simulations, we decided to continue with a similar physical design as last year and to build off our existing software systems rather than implement entirely new ones. However, we still took on the challenges of manufacturing a new boat, incorporating new sensors, creating custom PCBs, and reworking our code framework. As a team that values innovation and hard work, the shortened timeline became an opportunity to see what we are capable of when pushed.

III. Design Strategy

A. Mechanical System

Hull and Frame Design

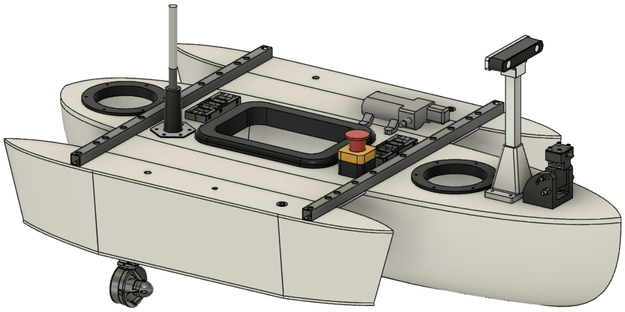


Figure 1: Full Mechanical Assembly.

Our hull design aims to optimize boat stability and minimize camera movement for improved image quality and continuity. A trimaran design was chosen with a large displacement hull in the center to maximize stability, utilizing amas for additional support during turns. To optimize water displacement, the long and slim amas were designed to barely touch the water until the boat starts to roll, maximizing volume without inducing extra drag. Balancing buoy navigation and stability, we have carefully considered the placement of the amas.

Our electrical system is housed in our hollow displacement hull, strategically lowering the center of mass and greatly enhancing overall stability. However, this design presents challenges in accommodating all components within the limited space and ensuring the waterproofing of the enclosure at water level.

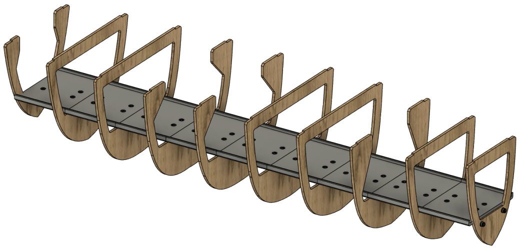


Figure 2: Frame & E-Bay Tray

To address these concerns, we redesigned the frame to be slimmer with tighter tolerances, prioritizing weight reduction and facilitating easier access to the electronics-bay. The electronics bay was redesigned to support all the electronic components on a modular extended tray with velcro fasteners. The design also incorporated off-the-shelf waterproof hatches, modular multi-cord grips and metal-rubber bonded sealing washers to streamline and expedite the waterproofing process.

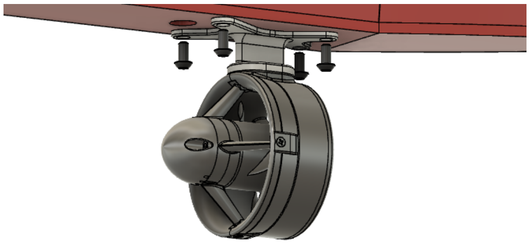


Figure 3: Propulsion Mount

T200 thrusters showed to be difficult to install and prone to failure from lateral loading with off the shelf mount. To combat this, custom mounts were developed and tested. These mounts fail first under load to protect other equipment, and take less than 2 minutes to replace.

B. Electrical System

Power System

The boat is powered by two Blue Robotics 14.8V Li-Ion batteries. We added a second battery to ensure the Ubiquiti Bullets received full power and maintained a strong connection.

Sensor and Computer Hardware

The boat's main computer is a Stereolabs Nvidia Jetson Xavier NX. We use a ZED 2i stereo camera for object detection and depth sensing. A goal this year was to improve the accuracy of our localization, which was previously informed by the ZED's positional tracking capabilities. Experimentation showed the camera's vulnerability to confusion from disturbances, so we opted to integrate a compass and GPS this year for more reliable data.

The compass module is a DEVANTECH CMPS12 Tilt-Compensated Magnetic Compass. The GPS module is a ZED-F9P Sparkfun Micromod GNSS carrier board coupled with an ESP32 processor. The GPS and compass modules communicate with the Xavier constantly, providing it with localization data. The GNSS boards were set up following [3], [4].

PCB Design

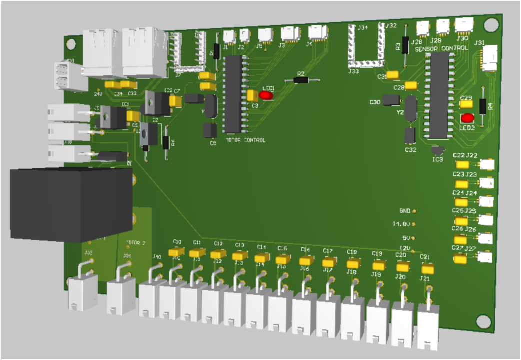


Figure 4: Main Power & Signal Distribution Board
See Figure 16 in Appendix D for a detailed schematic.

AutoBoat's electrical systems are centered around the main PCB, which acts as a power and signal distribution board. The functions of the board are: supply power to the Jetson Xavier, Ubiquiti Bullet modem, GPS module, compass module, Motors/ESCs, RC receiver, kill switch relay coil, sensors, and microcontrollers, as well as facilitating analog, digital, and serial communication between these components.

The PCB is a 4-layer board. The top and bottom layers are for power and signal traces. The middle two layers are planes, one for 14.8V power from the main battery and one for ground. The 14.8V plane powers the Jetson Xavier and the voltage converters. For the microcontrollers, sensors, RC receiver, GPS module, and compass module, an LM7805 linear voltage regulator steps the voltage down to 5V. For the kill switch relay coil, an LM7812 regulator steps the voltage down to 12V. The 14.8V, 12V, and 5V nets have 6 outlets each on the board.

For the Bullet modem, a DFR0946 buck-boost converter boosts the voltage up to 24V. This voltage is then routed to the power pins of an RJ45 ethernet connector. The TX and RX data lines are routed as differential pairs on the board from another RJ45 connector, which the Xavier ethernet cable plugs into. This way, the board acts as a POE injector for the Bullet modem.

The motor power stack uses a separate 14.8V battery, and is isolated from the rest of the board due to the high power draw of the motors. The motor power runs through a Panasonic AHES3191 2-form-a relay, before going to the ESCs. The relay coil is controlled by running a 12V line through an onboard mechanical switch. A flyback diode is connected across the coil to protect the board from any discharge spikes.

The board has two ATMEGA-328P-PU microcontrollers, one for motor control (PWM signals to the ESCs, RC control, remote kill switch activation, and receiving commands from the computer), and another to transmit sensor data to the Xavier. The ATMEGA was chosen because it is used on the Arduino Uno and is easy to interface with using our existing Arduino IDE code framework. It is used instead of the Arduino development board to keep the system more compact and reduce the amount of separate components and wired connections. The PCB allows access to all of the I/O ports from both microcontrollers. Their UART serial pins are used for communication via a USB-Serial converter. The sensor chip's digital and analog pins are used to read data from a Blue Robotics SOS leak sensor and several Texas Instruments LMT85LP temperature sensors. The motor control chip's analog pins receive signals from the RC receiver, two of its digital pins send PWM signals to the ESCs, and one of its digital pins receives a high/low signal from the relay, indicating whether the onboard kill switch is pressed.

The kill switch status signal is created by running a trace from the switch to a Vishay IRF640PBF N-channel MOSFET. The MOSFET's source pin is connected to the board's 5V net. When the MOSFET experiences a gate voltage, the signal coming from its drain pin is pulled high and sent to the microcontroller.

C. Software System

System Overview

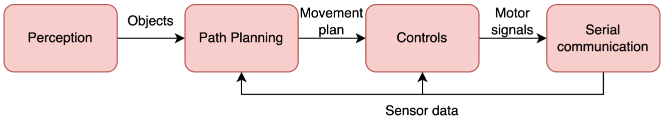


Figure 5: Overview of communication between software components. See Appendix D Figure 17 for a more comprehensive diagram of the ROS framework.

The software system of AutoBoat consists of four main components: Perception, Path Planning, Controls, and the Ground Station. The boat receives information about its surroundings, position, and orientation from the perception system in conjunction with serial communication from the sensors. The path planning system uses this information to identify the task at hand and plan a path or sequence of movements to successfully accomplish the task. This plan is then given to the controls system which determines how to move the motors to achieve the desired movement. The ground station system allows us to wirelessly control and monitor the boat. All of these systems are held together by our ROS Noetic framework which facilitates parallelism and communication between each component. Importantly, it allows us to see (perception), think (path planning), and move (controls) simultaneously.

Perception

We use an object detection model to identify competition buoys and targets for autonomous tasks. Our model is trained on over 18,000 images using the You Only Look Once (YOLO) neural network. Specifically, we started with the YOLOv8L weights and trained for 400 total epochs.

The computer vision model is capable of recognizing seven different object classes: 'red-buoy', 'yellow-buoy', 'green-buoy', ‘blue-buoy’, ‘black-buoy’, ‘red-column-buoy’, and ‘green-column-buoy.’ Notably, we need to recognize other objects like docking targets and yellow duckies in the competition, but those entities are mapped as their associated colored buoys in our dataset. Doing so allows us to effectively build a color detector, and since we never need to differentiate docking targets and buoys at the same time, there is never ambiguity with the model’s output.

We also implement a persistent memory algorithm that uses information from the previous frame to correct model predictions in the current frame. Essentially, if an object is seen previously but not seen currently, there is a short cooldown period of a couple frames in which the object is “hallucinated” into the present output. This additional step allows our system to be resistant to frames where the model fails to identify anything when we know something should be there, a problem we encountered during integration testing.

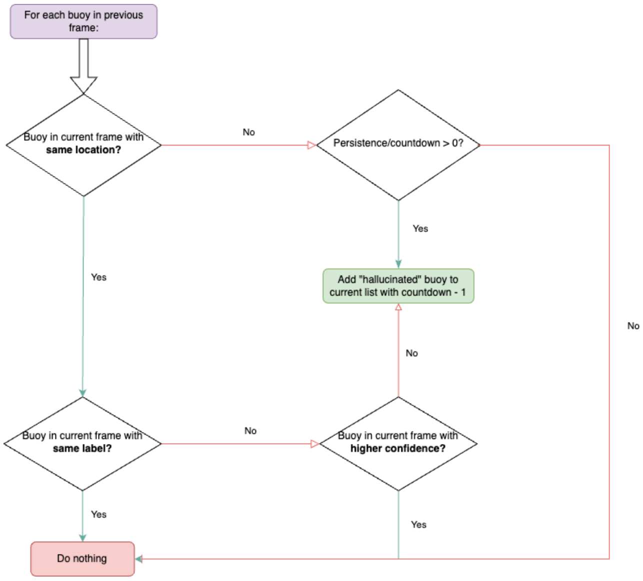


Figure 6: Persistent Memory Algorithm Flow

Path Planning

We use a similar approach for accomplishing all navigational tasks. First, the boat explores the environment until it gains sufficient information regarding the task at hand. Then, it uses the information about its current state (GPS location, compass heading) and environment (list of objects detected and their locations relative to the boat) to plan a path of GPS waypoints. After primary waypoints are selected we inject intermediate waypoints every meter along the path and smooth any harsh angles to produce more natural paths of movement.

For fairly straightforward tasks, like Navigation Channel and Docking, the primary waypoints are selected using mathematical calculations, such as identifying the midpoint of two buoys. After examining research on industry standard path planning algorithms, such as [2], we opted to apply the A* algorithm for the more complicated Follow the Path task given the algorithm’s simplicity and effectiveness. A* finds the shortest path through the maze according to a distance heuristic which avoids obstacles and stays in between the gate buoys.

The change to the Speed Challenge task this year of having an uncertain location for the blue buoy prompted us to implement a collision detection and avoidance feature which runs while a path is being executed. For example, in the Speed Challenge task, the boat plans a loop path to go through the gate and circle around the yellow buoy. However, while this path is being executed, the boat is repeatedly analyzing its surroundings and determining if any buoys identified intersect its path of movement. If so, it modifies the path to avoid the obstacle. We apply this strategy to the Return to Home task as well.

Controls

To follow the waypoint path outlined by the path planning system, the control system employs a combination of Pure Pursuit and PID control. Pure Pursuit is a path tracking algorithm which maintains a “lookahead” point on the path some set distance away from the boat. As the boat moves, the point advances along the path, so the boat is always chasing it. Our implementation is inspired by the Purdue SIGBot’s controller [5].

Pure Pursuit typically outputs an ideal linear and angular velocity for the vehicle. However, we found that controlling on velocity produces rough and shaky movements due to the sensitivity of our IMU, a sensor in the ZED 2i camera which produces velocity readings. This year we decided to transition to using heading as the variable we monitor since our compass readings are more reliable. Learn more about this method in [1]. The algorithm calculates the error in heading as the difference between the boat's current heading and its heading if it were pointed directly at the lookahead point. We then apply PID control to reduce this error. PID (proportional, integrative, derivative) control is a tunable equation which takes the heading error as input, multiplies the error, integral of error, and derivative of error by some constants, and outputs the offset which should be applied to the PWM signals sent to move the thrusters. A larger error produces a greater offset causing the boat to turn faster and exhibit course correction.

Ground Station

A major takeaway from competition last year was our need for a more robust communication system to wirelessly control and monitor the boat. As a result, we formed the Ground Station team. Inspired by the invaluable help of RoboBoat staff and fellow teams, we set up a wireless network using Ubiquiti bullets, a router, and POE injectors to facilitate a connection between our onboard computer, the Nvidia Jetson Xavier, and onshore laptops. This connection allows for 3 essential features: SSH connection into the Jetson, transmitting GNSS correction source data, and receiving state information to be used as input to our monitoring and visualization platform. To facilitate the latter two features we dedicate nodes in the ROS framework for launching WebSocket connections to enable communication with programs run on onshore laptops.

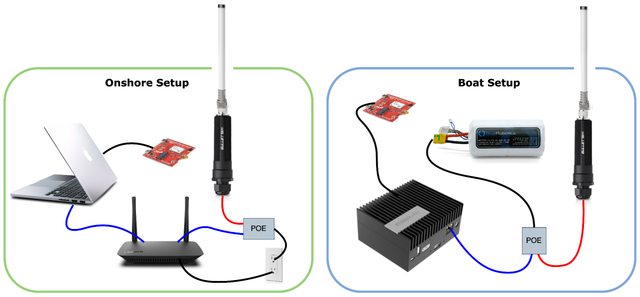


Figure 7: Communication system setup. Blue wires are LAN, red wires are POE, and black wires are power.

IV. Testing Strategy

Another main takeaway from last year's competition was the need to prioritize testing throughout our design and implementation process. We opted to split our efforts between land, simulation, and water testing.

A. Land & Simulation Testing

Mechanical Testing

During the design phase of the boat, our boat design & manufacturing team performed comprehensive simulations to examine the stability of the vessel. The team utilized Orca software to analyze the stability curves of various designs. Figure 8 shows the stability curve of the previous year's design (blue) to this year's design (green). Compared to last year's design, the stability curve shows an increase in the range of stability, peak righting arm, and a higher angle where that righting arm occurs.

This means there is a strong balance between minimizing capsize risk while maximizing stability at operating heel angles. This is in part due to the increased width of the vessel which corresponds to a larger righting moment from the amas when the boat heels. The initial slope of the stability curve is steeper on this year's designs compared to last year, which means the boat is stiffer at low heel angles. This is due to the slightly wider and longer amas designed this year to supply a stronger righting moment when the amas are submerged in the water.

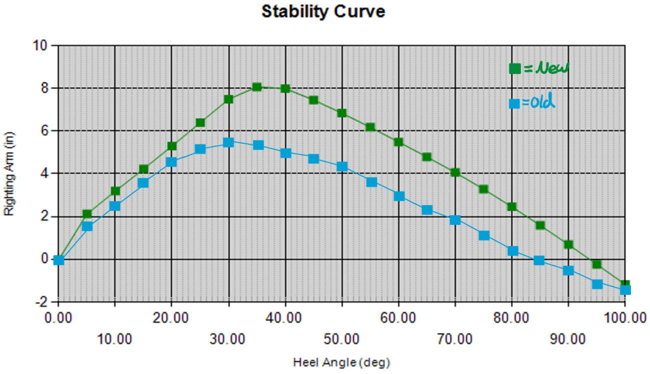


Figure 8: Stability curves of old vs new design.

Electrical Testing

The electrical testing strategy was to first verify every component of the system in isolation, without being connected to the motors, the batteries, or the Xavier, for the safety of the system. Power distribution was checked with a multimeter, and communications and commands were checked using a serial monitor on a computer. Once everything was tested in isolation, the board was connected to all of the components in the boat and tested in normal operating conditions. Refer to Appendix B for a more comprehensive testing plan.

<u>Software Testing</u>

We primarily tested our computer vision model through integration testing on our Jetson Xavier. After training the model, we deployed it onto the boat's computer, staged buoys around the boat's environment, and used a combination of command line output and visualization output to sanity check our model. Compared to our YOLOv5 model last year, this year's iteration is much better, having consistent accuracy within roughly a 10m range, and rarely misclassifying objects. We tested our persistent memory algorithm as well using systematized unit testing and edge-case testing.

Given constraints on water testing due to weather and a lack of access to pools, we also invested time and resources into a Gazebo-based simulation framework for testing our code. We chose Gazebo based on recommendations from competition coordinators and fellow teams, as well as its compatibility with ROS. We used GZweb to connect to a Docker container to manage dependencies and run Gazebo headlessly. This framework will allow us to test our path planning and path execution code in more depth as we approach competition.

While the simulation framework was still under development, we utilized static testing frameworks to test path planning code. This included unit testing of mathematical functions and open loop visualizations. The visualizations show the path planned in a single iteration of our control loop with hardcoded sensor data input.

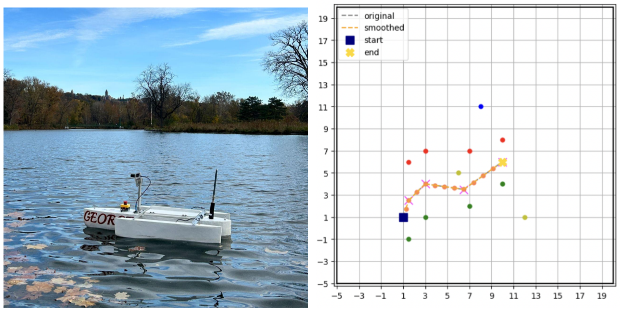


Figure 9: 2023 competition boat, George, on Lake Cayuga (L), and a static visualization for Follow the Path (R).

B. Water Testing

Our water testing includes outdoor testing at a local lake and indoor testing at local pool facilities. This testing consists of three stages: 1) safety testing to replicate the checks done at the onset of competition, 2) remote controlled buoyancy, speed, and stability testing to examine the maneuverability of the hulls, 3) autonomous control algorithm tuning (e.g. PID gains, pure pursuit lookahead distance), and 4) autonomous navigation testing. To test autonomous navigation we incrementally build up the complexity of tasks, starting by attempting to navigate to a predetermined GPS waypoint, then navigating to a waypoint determined by the vision system, and finally autonomously planning a series of waypoints to accomplish a competition task.

V. Acknowledgements

We extend our appreciation to everyone who lent their support to our team and helped enable us to innovate and grow as engineers. We thank the Cornell University College of Engineering for helping fund our team's travel and material expenses and providing access to facilities for boat construction and design. Our sincere thanks also go out to our individual supporters, including friends, family, and alumni of the AutoBoat team, whose generous donations have greatly contributed to our endeavors. The guidance and support provided by Dr. Hunter Adams, our faculty advisor, has also been instrumental in propelling our team towards success. We would also like to give a special thanks to the members of UM::Autonomy, RoboBoat staff including Nicholas Cassasa, Carloyn Judge, and Tanvir Reza, and all our fellow RoboBoat teams for giving us advice and support during and since our first in person competition last year. Additionally, thank you to the folks at Hotel Ithaca and the Ithaca YMCA for allowing us to use your pool facilities. We also appreciate the support from our corporate sponsor, Orca3D. Lastly, we acknowledge the efforts and dedication of former AutoBoat team members whose significant contributions paved the way for our current achievements.

VI. References

[1] A. Sears-Collins, “PID Control Made Simple”. October, 2022. Automatic Addison. https://automaticaddison.com/pid-control-made-simple/

[2] N. Sariff and N. Buniyamin, "An Overview of Autonomous Mobile Robot Path Planning Algorithms," 2006 4th Student Conference on Research and Development, Shah Alam, Malaysia, 2006, pp. 183-188, doi: 10.1109/SCORED.2006.4339335.

[3] N. Seidle, “Setting up a Rover Base RTK System.” SparkFun Learn. https://learn.sparkfun.com/tutorials/setting-up-a-rover-base-rtk-system

[4] N. Seidle, “How to Build a DIY GNSS Reference Station.” SparkFun Learn. https://learn.sparkfun.com/tutorials/how-to-build-a-diy-gnss-reference-station

[5] S, Xiang, “Basic Pure Pursuit.” Last Modified May, 2023. Purdue SIGBots Wiki. https://wiki.purduesigbots.com/software/control-algorithms/basic-pure-pursuit

VII. Appendix A: Component List

Component Name	Vendor	Model/Type	Specs	Custom / Purchased	Cost	Year of Purchase
ASV Hull Form/Platform	Self Developed	N/A	60in x 30in x 28in	Custom	$3000	2023
Propulsion	Blue Robotics	T200	Up to 5 kg Thrust / Each link	Purchased	$400	2023
Battery	Blue Robotics	Lithium-ion Battery 14.8V, 15.6Ah	14.8V, 15.6Ah Max draw 60A Max Burst 132A link	Purchased	$660	2023
Power/Signal Distribution Board	Self Developed	4-layer PCB	Dimensions (mm) 130 x 208 x 0.5717	Custom	$300	2023
Motor Controls	Blue Robotics	Bidirectional ESC	30 A max 7-26 V Runs on BLHeli_S link	Purchased	$72	2023
CPU	Stereolabs	Xavier NX	- NVIDIA® Jetson™ TX2-NX - GPU: 256-Core NVIDIA® Pascal™ - CPU : Dual-Core NVIDIA Denver 2 64-Bit and Quad-Core ARM Cortex-A57 MPCore - Memory : 4GB LPDDR4 - 51.2 DB/s	Purchased	$1390	2023
Remote Controller	FLYSKY	2.4G FS-CT6B 6 Radio Model RC Transmitter & Receiver	8 model memory, digital control, full 2.4GHz 6-channel radio, 4-Model memory, integrated timer, throttle cut, computer programmable, USB Socket	Purchased	$50	2021
Radios	Ubiquiti	UISP airMAX Bullet AC Dual-Band IP67	PtMP, PtP BaseStation radio, 5 GHz frequency band, weatherproof IP67 rate link	Purchased	$260	2023
Radio Antennae	Elecbee	Omni-Direction 2.4G Wifi Fiberglass Antenna	2.4GHz WIFI Antenna, Male, straight, standard, Threaded, 5dbi link	Purchased	$38	2023
Compass	Sparkfun	CMPS12	Tilt Compensation 3-axis Magnetometer 3-axis Gyro 3-axis Accelerometer	Purchased	$33.18	2023

			0.1 Degree Resolution Accuracy within 1% link			
GPS	Sparkfun	MicroMod GNSS ZED-F9P	25z Navigation Rate 0.01m Horizontal Positional Accuracy with RTK 2x USB-C Connectors link	Purchased	$325	2023
GPS Processor	Sparkfun	MicroMod ESP32 Processor	Dual-core Tensilica LX6 microprocessor 240MHz clock 520kB internal SRAM Integraterd Transceiver link	Purchased	$16.95	2023
GPS Antenna	Sparkfun	MagmaX2 - AA.200	Magnetic Mount IP67 Covers GPS L1/L2 and more link	Purchased	$83.50	2023
Camera & IMU	Stereolabs	ZED 2i Stereo Camera	120 FOV, built-in IMU, Barometer, Magnetometer, depth sensing, positional tracking, object detection, IP66-rated enclosure link	Purchased	$499	2021
Microcontroller	Atmel	ATMEGA328P-PU	Core: AVR Program Memory Size: 32kB Data RAM Size: 2 kB Package / Case: PDIP-28 Max Clock Frequency: 20 MHz Supply Voltage - Min: 1.8 V Supply Voltage - Max: 5.5 V	Purchased	$2.89	2023
Waterproof Boat Hatch	Pactrade Marine	Circular Hatch	- 6” Diameter - Foam Gasket link	Purchased	$32	2023
Waterproof Boat Hatch	Pactrade Marine	Rectangular Hatch	- 7.5" by 11.5" Opening link	Purchased	$34	2023
Propulsion Mounts	Self Developed	3D printed Brackets	- Modular Interface	Custom	$50	2023
Split Multi Cord Grips	McMaster	Surface-mount 10 inserts	- Modular interface with link	Purchased	$80	2023
Algorithms (Motion Controllers)	N/A	Pure Pursuit algorithm, PID controller	N/A	Custom	N/A	N/A
Vision	N/A	YOLOv8L	N/A	Custom	N/A	N/A
Localization	N/A	ZED 2i Stereo	N/A	Custom	N/A	N/A

and Mapping		Camera, custom sensor fusion				
Autonomy	N/A	A* algorithm, specialized waypoint selection	N/A	Custom	N/A	N/A
Programming Languages	N/A	Python 3, Arduino/C++	N/A	Custom	N/A	N/A
Programming Packages and Open Source Software	N/A	ROS 1 Noetic, Numpy, Pytorch, Websockets, ZED SDK, Docker, GZweb, Google Collaboratory, Roboflow	N/A	Custom	N/A	N/A
Simulation Software	N/A	ANSYS, Altium, Rhino3D, Orca3D, Gazebo, Virtual RobotX	N/A	Custom	N/A	N/A

VIII. Appendix B: Test Plans & Results

Electrical Testing Strategy and Possible Failure Modes

1. Have everything assembled.
2. Plug in the central board power from a power supply, test 14.8, 12, 5 volt nets with a multimeter.
3. Test code push and serial comms, first with arduino and then with the FTDI adapter.
    a. Push blink sketch
    b. Push a sketch that writes to the serial monitor
4. Test motor signals
    a. Push the main motor control code, have it print motor control values to serial
    b. Plug the receiver into the system
    c. Monitor serial using a computer to make sure values being sent are consistent with user input.
5. Test kill switch
    a. Plug in the motor power from power supply
    b. Wire kill switch
    c. Use a multimeter to verify that the switch and the relay mechanically break the circuit
    d. Use the serial monitor to check if the microcontroller detects the status change
6. Test POE injection
    a. Plug in the POE power using a power supply
    b. Check the 24V net with respect to ground
    c. Use a split ethernet cable to check the output voltage that the Bullet antenna would be receiving.
7. Test sensors
    a. Push sensor code
    b. Plug the sensors in
    c. Use the serial monitor to verify sensor readings
8. If all is good, test all of the above in the boat with all systems connected.

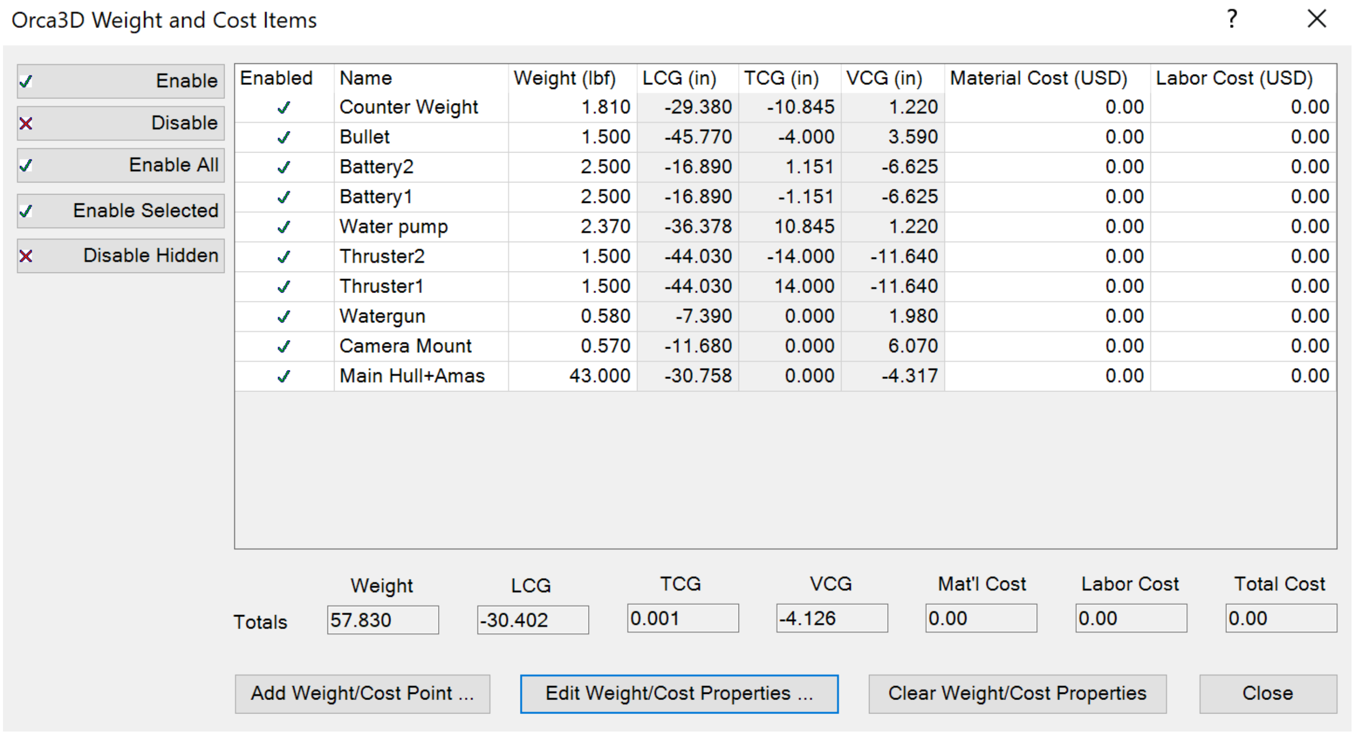
Possible Problems:

- POE not working; inconsistent
    - Use a split cable like last year, research how to handle more complex/high speed signals/power lines on a PCB for next year.
- FTDI not working
    - Make sure drivers are installed on all computers that interface with it
    - As a backup, use an empty arduino dev board as the adapter via the 3 pin connector
- Arduino as FTDI not working
    - ATMEGA could be bootloaded incorrectly, or not bootloaded at all. Make sure to bootload the controller to use the external 16MHz clock.
    - Could be a PCB routing issue, either hard-wire directly to the microcontroller leads, or order a new board with the problem fixed.
- Voltage nets not at the expected voltage
    - Double check the regulators, may have to switch them out. Make sure they are soldered in correctly.
- Motors not responding correctly
    - Try different I/O pins for the PWM output
    - Investigate motor commands with an oscilloscope

    - ESC may be defective, or the uncommon ground may be causing the problem
- Kill switch not working or status message wrong
    - Check wiring
    - Switch out relay
    - Check FET orientation
    - Double check the flyback diode orientation
- Sensors have incorrect readings
    - Sensors may be defective, or not calibrated correctly

Hydrostatics

In order to more accurately analyze the heel and trim of the boat, the weight and center of mass of each component were assigned in Orca to get an accurate center of mass of the boat (Figure 10). It is important to note that the LCG is negative due to the model orientation in Orca. The first test had each component in their respective position with no modifications or changes. With the default configuration, the trim was -0.1 degrees and the heel was -2.25 degrees (Figure 11). The trim should be lower to account for the positive trim supplied by the thrusters, and the heel should be close to zero when in a stable position. This heel angle was mostly caused by the offset position of the water pump, which is located on the right bridge deck. To fix these issues, a 2 lb counterweight was added to the opposite side of the bridge deck, which resulted in a trim of -1 degrees and a heel of -0.01 degrees (Figure 12).



Enabled	Name	Weight (lbf)	LCG (in)	TCG (in)	VCG (in)	Material Cost (USD)	Labor Cost (USD)
✓	Counter Weight	1.810	-29.380	-10.845	1.220	0.00	0.00
✓	Bullet	1.500	-45.770	-4.000	3.590	0.00	0.00
✓	Battery2	2.500	-16.890	1.151	-6.625	0.00	0.00
✓	Battery1	2.500	-16.890	-1.151	-6.625	0.00	0.00
✓	Water pump	2.370	-36.378	10.845	1.220	0.00	0.00
✓	Thruster2	1.500	-44.030	-14.000	-11.640	0.00	0.00
✓	Thruster1	1.500	-44.030	14.000	-11.640	0.00	0.00
✓	Watergun	0.580	-7.390	0.000	1.980	0.00	0.00
✓	Camera Mount	0.570	-11.680	0.000	6.070	0.00	0.00
✓	Main Hull+Amas	43.000	-30.758	0.000	-4.317	0.00	0.00

	Weight	LCG	TCG	VCG	Mat'l Cost	Labor Cost	Total Cost
Totals	57.830	-30.402	0.001	-4.126	0.00	0.00	0.00

Figure 10: Weight cost assignment for more accurate hydrostatics

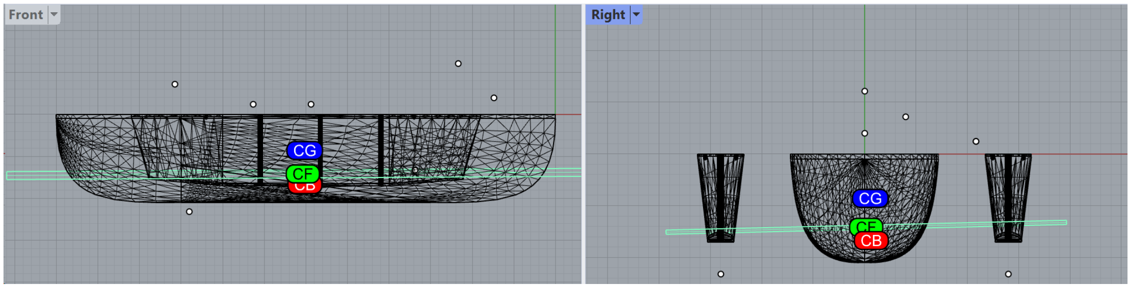


Figure 11: Waterline with first iteration of updated center of mass

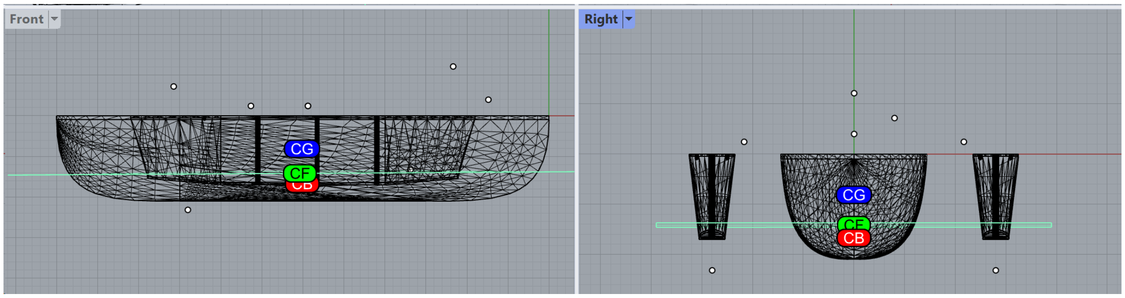


Figure 12: Waterline with updated center of mass including counterweight

Stability

The stability curve was then generated for the updated center of mass (Figure 13). For clarification, the stability curve shown previously in this report used the estimated weight of the main hull, amas, and extra components, but assumed this weight was acting at the center of mass of the main hull. This was done to understand the baseline stability of the hull design to have a fair comparison to last year's design. With an updated center of mass, it is important to also check the stability of the vessel. The only significant change is the maximum righting arm is slightly less than the original simulation, but not enough to cause any concerns. The transverse metacentric height (GMt) for the first test was $GMt = 25.4in$, while the updated has a $GMt = 24.7in$. While the overall stability has decreased slightly with the new center of mass, when comparing this to the GMt of last year's boat ($GMt = 22.1in$), this still shows an improvement in heel stability at the cost of some maneuverability.

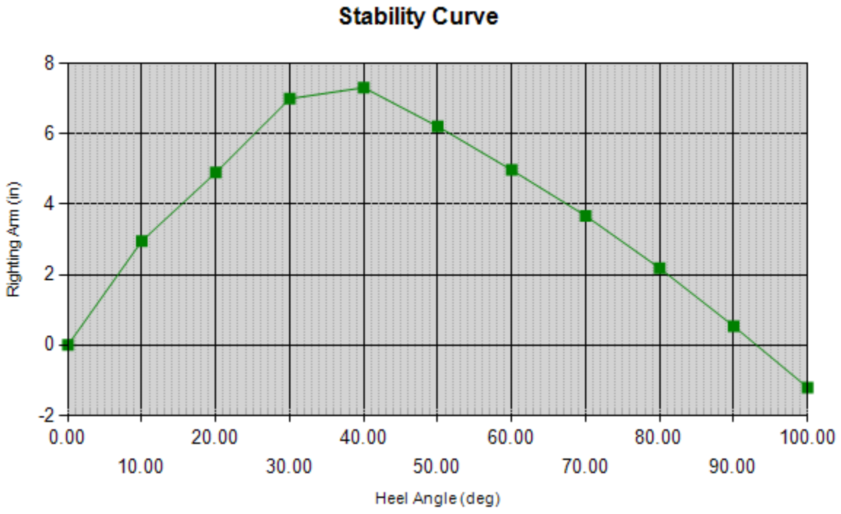


Figure 13: Updated stability curve with new center of mass

Hull Design Motivation

The main hull is designed to maximize the length of the waterline (LWL) to help maximize the hull speed and minimize the Froude number. The hull speed does not dictate the absolute maximum speed the boat is able to travel at, but rather the speed at which the boat will start to ride its own bow wave. This speed should not be exceeded to help minimize trim angle to ensure the camera stays as level as possible while navigating autonomously. With an overall length of 60in and a LWL of 57in, the LWL is 95% of the overall length. This corresponds to the main hull acting as a displacement hull at operating speeds, as shown by a Froude number of $Fn = 0.39$ which is just within the boundary of a displacement hull ($Fn < 0.4$). As this Froude number was calculated using the hull speed, it is important to consider the expected speed during autonomous runs to be around $u_{hull} = 0.5 m/s$. The Froude number at this speed is $Fn = 0.13$, which shows the design goal was achieved and the main hull will act as a displacement hull to maximize stability.

Hull speed: $u_{hull} = \sqrt{\frac{LWL \times g}{2\pi}} = 1.48 m/s$

Froude number: $Fn = \frac{u_{hull}}{\sqrt{g \times LWL}} = 0.39$

IX. Appendix C: Manufacturing Methodology

Center Hull, Frame & Deck

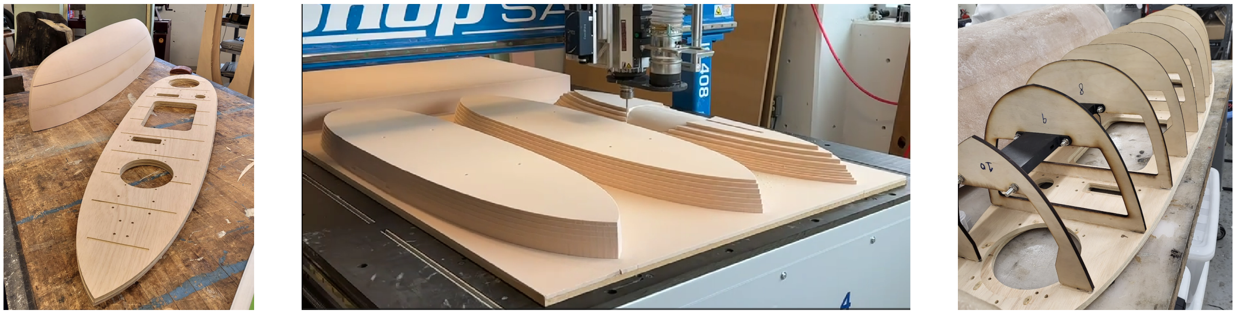


Figure 14: CNC foam mold and wooden deck (L), CNC mold fabrication process (C). Laser cut wood frame (R)

Initially, a male mold of the hull surface was created by assembling CNC cut sections of tooling board. Following the mold's completion, the hull was crafted using a hand layup of fiberglass mat and polyester resin. After the resin cured, the hull was separated from the mold. The structural elements of our frame were laser-cut from Baltic birch wood, assembled inside the fiberglass hull, and covered with our CNC cut wood deck. The junction where the deck met the fiberglass was sealed with fiberglass mat and resin. Epoxy fairing compound was applied to address major irregularities in the geometry. The entire structure was sanded for a smooth surface finish then topped with a barrier coat and paint to protect against the elements while boasting a flashy red appearance.

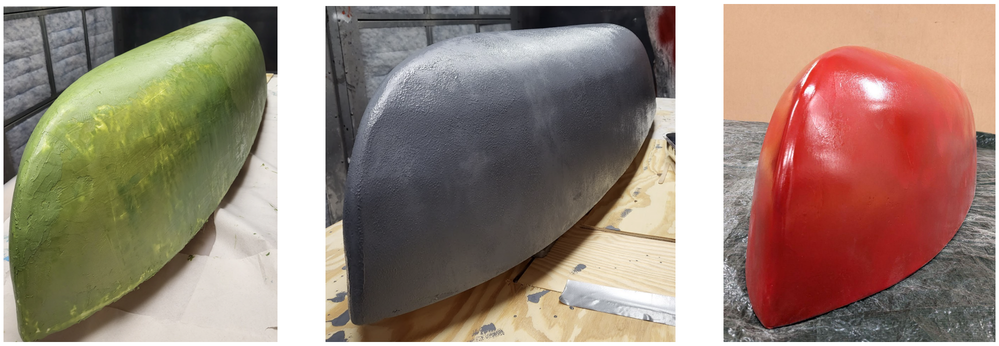


Figure 15: Fairing Compound Coat (L), Barrier Coat (C), Red Paint (R)

The amas were manufactured similarly to the main hull, with the distinction that the foam core was hand-cut to shape from low-density polystyrene. Additionally, the foam core was not removed from the composite part to enhance structural strength.

X. Appendix D: Additional Figures

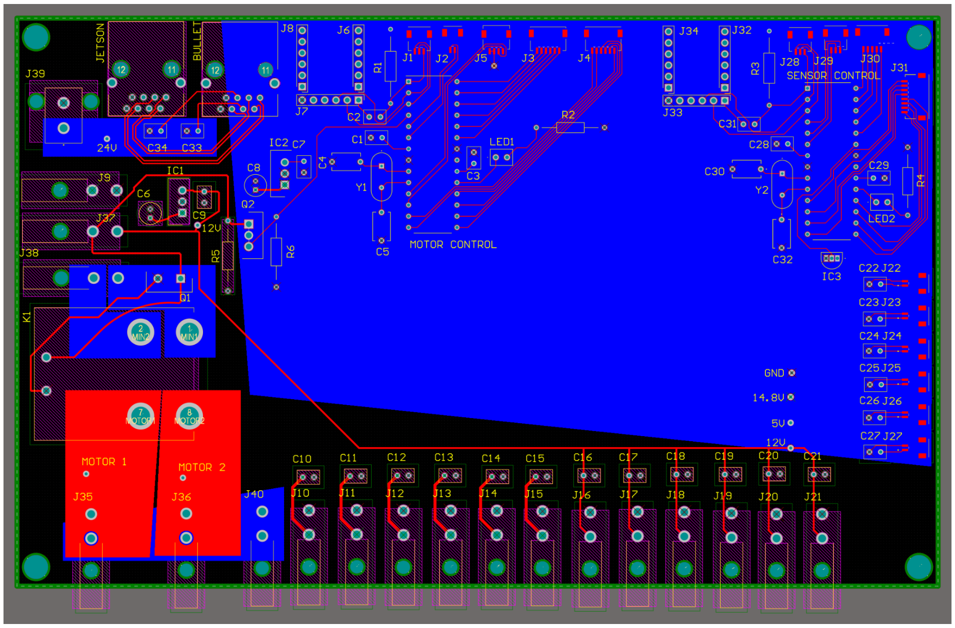


Figure 16: Schematic diagram of the PCB. Large blue polygon is a 5V plane which powers the microcontrollers and provides power to the 5V outlet. Red polygons represent traces to deliver power to the motors. The blue polygons on the middle left are also power delivery from the motor battery input to the relay. The blue polygon on the top left is a small power plane for 24V power delivery to the POE injector.

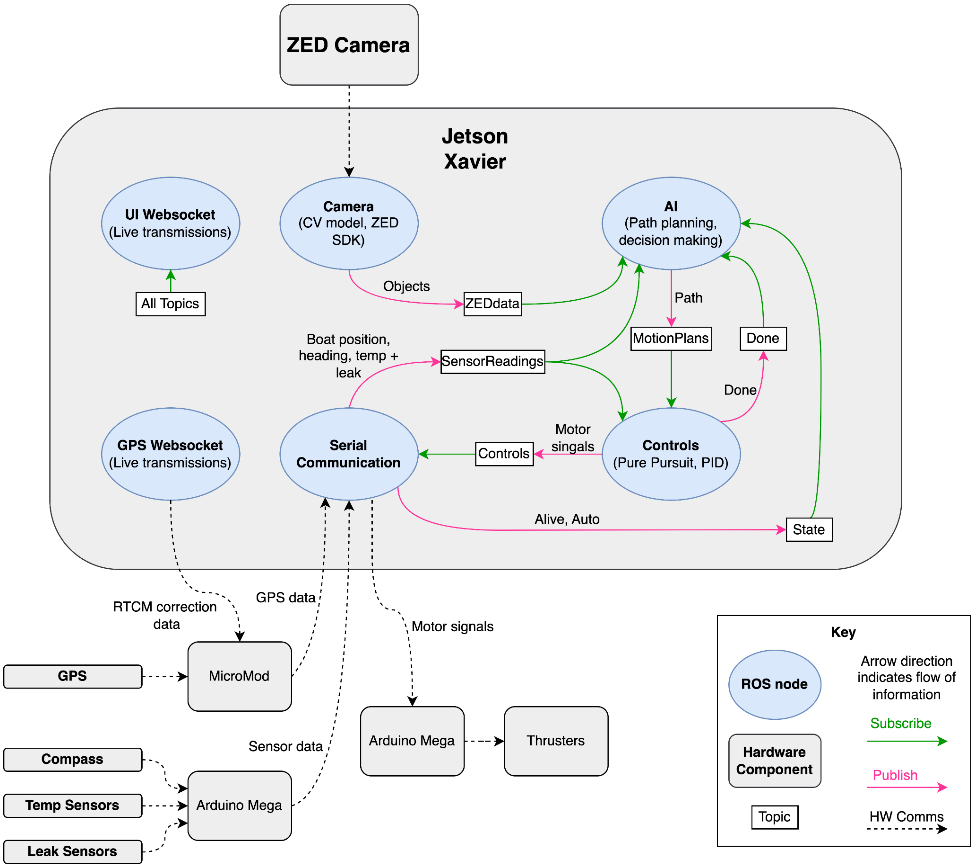


Figure 17: Publisher/subscriber node structure of the ROS framework and interaction between software and hardware devices, including the microcontrollers and sensors.