RoboBoat 2026 Technical Design Report Humber ASV – Loon-E

Amelia Soon, Ian Cameron, Carson Fujita, Emiliano Rodriguez
Humber Polytechnic, Toronto, ON, Canada

***Abstract*—The Loon-E is Humber ASV's first Autonomous Surface Vehicle (ASV), developed for participation in the 2026 RoboBoat competition. This paper presents the rationale behind design decisions which contributed to the development of the Loon-E, including the team's objectives, overall system logic, and a proposed methodology for the execution of the competition's various tasks. The design strategy section includes a description of each subsystem and the process of their development, including the hardware, electrical, and software design. Finally, the various tests performed to verify the functionality of each subsystem are discussed, whether in simulation or in real life.**

Keywords—Autonomous vehicles, Artificial intelligence, Marine engineering

I. Competition Strategy

This will be Humber ASV's first year participating with an ASV, as the team participated without a boat at RoboBoat 2025. The team's objective was to design and build a functional remote-controlled boat and add any systems necessary for full autonomy. Autonomy requires the integration of sensors (camera) as well as the software which integrates these sensors into the overall control flow (see Figure 1 and Appendix A).

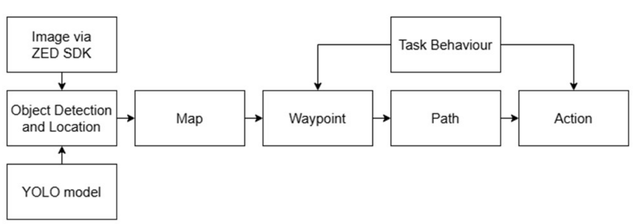


Fig 1. Control Flow

Successful completion of Task 1 will be sufficient to demonstrate functionality of the system, and as such is the team's main objective. However, the execution of subsequent tasks is considered for the purpose of future development, and the intention is to test the effectiveness of proposed solutions for other tasks during the competition. Should these methodologies be executed effectively, the Loon-E will demonstrate core capability in Task 1 and advanced capability in Tasks 2 and 3, and it should be possible to at least perform Tasks 1 and 2 in sequence.

Tasks 4 (Supply Drop) and 6 (Harbor Alert) are not addressed as they require additional task-specific subsystems (water cannon, racquetball launcher, and microphone) which do not otherwise contribute to the system's autonomous function. Similarly, Task 5 (Navigate the Marina) requires more complex logic which is not used for any other task. These were not developed due to time constraints and as such are not possible to attempt with this year's system.

A. *Task 1 (Evacuation Route & Return)*

Given an image of the task, the closest large green and red buoy can be identified. After calculating the midpoint between the two buoys, the Loon-E will drive via PID control [1], aiming to keep this point in the center of the frame. This process will be repeated until there are no longer any visible large buoys.

B. *Task 2 (Debris Clearance)*

The first half of the debris field (containing the red and green buoys) can be completed in a similar method to Task 1. Any black buoys found will replace the nearest green or red buoy when calculating the midpoint.

After this, the ASV will locate the green beacon, circle around it, and return to the start point. This is done in three steps:

1. Generate and follow the path to the side of the beacon until it is no longer visible.
2. Travel in a semi-circle around the beacon until the starting point is visible.
3. Return to the starting point with the same path planning used in Step 1.

Once done, the ASV will return to the beginning of the task by travelling back through the first half of the debris field.

C. *Task 3 (Emergency Response Sprint)*

The ASV will identify the location of the yellow buoy as well as the beacon used to indicate the direction in which the ASV should travel. Given this, three waypoints will be identified: One at the start position and one to the left and right of the yellow buoy. The colour of the beacon will indicate whether the path should go to the left (green) or right (red) point first, using the strategy described for the second half of Task 2.

II. Design Strategy

As a first attempt at an ASV, there will inevitably need to be adjustments made if this project should continue for future years. As such, iterability and adaptability is a large focus throughout development. For this first iteration, components were selected based on preliminary research with particular focus on the equipment used by other teams.

A. *Hull Design*

The hull was designed considering parting lines and draft directions required for traditional molding techniques such as to be manufacturable using a variety of methods, such as fiberglassing, rotomolding, and 3D printing.

The hull notably features panels on the front and back of the boat on which various modules (ex. racquetball launcher, water cannon) may be installed. This makes it so that if a new element is required (such as a microphone for the new Harbor Alert challenge), they can be easily added to the hull with minimal changes to design.

After an initial fiberglass test, it proved too difficult to pursue this methodology and the team opted instead for 3D printing; while originally intended to be printed in one piece on a sufficiently large 3D printer, the team ran into logistical issues and could not do so. As an alternative, the hull was segmented into parts, each of which was printed individually with ABS on a smaller printer. The pieces were then glued, assembled, sealed using melted ABS, and further waterproofed with a layer of epoxy.

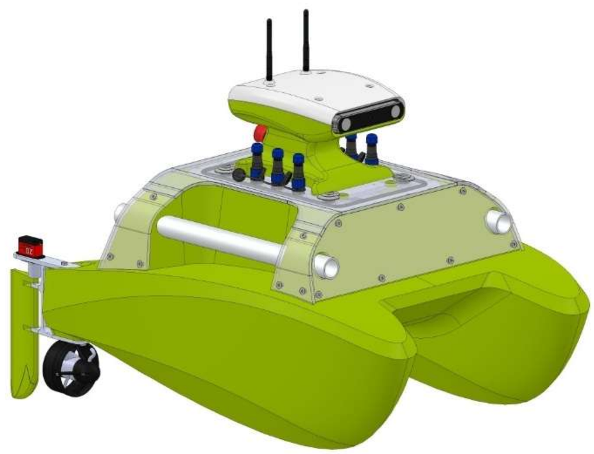


Fig. 2. Loon-E CAD

B. *Electrical System*

A full electrical system was implemented for the first time this year. With the base system that was prototyped last year, the team had implemented the necessary safety features such as the remote and on-board e-stops as well as basic propulsion control.

This year, the system plan has received significant improvements. The system now has two different power sources: Two 20 V, 6 Ah lithium-ion batteries, which will only be used to power the propellers, and a 14.8 V, 18 Ah battery which will be used to power the rest of the system. The goal of this was to utilize the maximum thrust of the two propellers as well as to increase the maximum run time of the boat. For this last point, a switching system has been added using a relay that will allow the propellers to use the 14.8 V battery in case the power of the 20 V battery is depleted, allowing the boat to be brought back to shore.

The electrical team also developed a custom PCB to alternate between the RC and Autonomous modes. It utilizes 2:1 multiplexer ICs that choose between the signals coming from the Jetson or from the RC receiver. It also sends a signal back to the Jetson indicating which mode the boat is being driven in. Additionally, the team designed a light indicator located on the exterior of the hull. It mainly consists of an RGB cob and a driver IC. The colour of each panel can be controlled though individual PWM signals raging from 0.3 V to 2.5 V. The design of the PCB was done using the EasyEDA software and diagrams for these circuits can be found in Appendix B.

Throughout the year the electrical team made prototypes of the bigger system to ensure its functionality and performed testing in both dry and in-water environments. Further details are included in Section III.

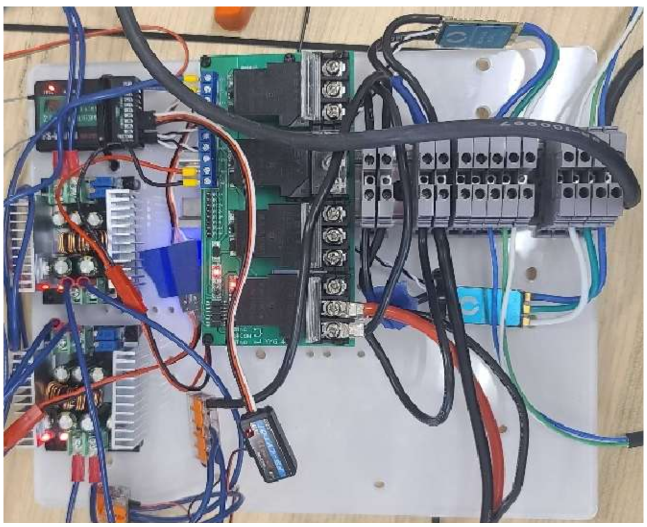


Fig. 3. Electrical components in boat

C. *Safety Systems*

The vessel carries the base safety features developed in the prototype from last year, namely the remote and on-board e-stop. In case the vessel loses connection with the RC remote, the receiver is configured to turn off the channel corresponding to the remote e-stop, shutting it off and cutting power to the propellers.

With the inclusion of additional and higher power batteries, it was important to ensure that the system remained safe above all other conditions. For instance, more robust components had to be implemented, namely terminal blocks, waterproof connectors, relays, as well as utilizing 10 AWG wire in the propulsion system circuitry. Similarly, to ensure consistent and secure connection, ferrules, WAGO connectors and screw-on terminal blocks were used around the system. The reasoning behind this was to avoid soldering as much as possible: soldering connections are permanent and time consuming and also introduce potential points of failure as a proper connection is hard to standardize and difficult to visually confirm. However, soldering was utilized on the exterior of the boat (ex. to connect to motors) as non-permanent connections are harder to waterproof.

Breakers were added to the 20 V batteries tripping at 30 A, slightly above the maximum operating current of 28.26 A. These batteries are monitored by a GoBILDA Battery Management System (BMS) that will stop the battery from discharging if it reaches the minimum safe voltage.

The individual level of each cell on the 14.8 V is also monitored through the BMS. If any of the cells is depleted, the output to the BMS is cut off. A voltage sensor that relays information to the RC remote was placed at the output of the BMS. In this manner, if the lithium-ion battery ever falls under the minimum safe voltage, the voltage will not have a reading. It is the user's responsibility to monitor the voltage reading of the voltage sensor, in case the read ever drops to zero. Although this implementation is not automatic and has human error, it is a scenario that will rarely happen as the battery has more than enough power to outlast the propeller batteries, and the 14.8 V battery will only be used to bring the boat back to shore, minimizing the possibility of error even more.

To connect devices on the exterior of the hull, waterproof connectors were used. These can be removed and sealed when not in use. To wire devices on the outside, watertight heat shrinking tubing was used to prevent water ingress to the bare connections.

The electronics hatch is made watertight with latches and a rubber seal that is squeezed when sealed. The electronics devices are suspended on the air by being mounted on the lid (see Appendix C, “Electronics Hatch”). This should provide extra protection in case of a any leak maintaining the water at the bottom of the cavity and away from the electronics.

D. *Component Selection*

The most notable system components include those used for propulsion as well as sensors to determine the state of the boat and its environment. For propulsion, the Loon-E uses two Blue Robotics T200 thrusters mounted to the back of the boat. For more precise steering, rudders have also been installed behind each propeller and are controlled by servo motors. Components are listed in Appendix D.

A stereoscopic camera such as Stereolabs' ZED X is crucial as it will provide data on the distance between the boat and surrounding objects; an IMU is also included in the ZED X camera for possible use in a future implementation, reducing part complexity. Data from the camera can be obtained through use of the ZED SDK [2], and up to three additional Stereolabs cameras may be added should it be deemed necessary in future years.

E. *Software Design*

The program runs on a Jetson Orin Nano Developer Kit using ROS 2 architecture, with nodes for each of the system's core processes (outlined in the sections below) each contained in a single package. The process for executing each task, as described in Section I, is also outlined in its own node and is executed when required (ie. the boat will be set to execute the Task 1 program when attempting Task 1). This architecture allows for a more straightforward representation of each process, supporting future development. Python is used for this project due to the team's familiarity with the language.

1) Object Detection: The initial dataset was made up of 520 images taken during the 2025 RoboBoat competition. 100 of these images were hand-labelled in Roboflow [3] and used to train a preliminary model which supported the labelling of the rest of the images. Variations of all images (with noise added) were used to train a final YOLO11 model [4]. The model will need to be retrained with new images added to the dataset to account for changes in present objects; these will be taken during competition to maintain consistency within the dataset.

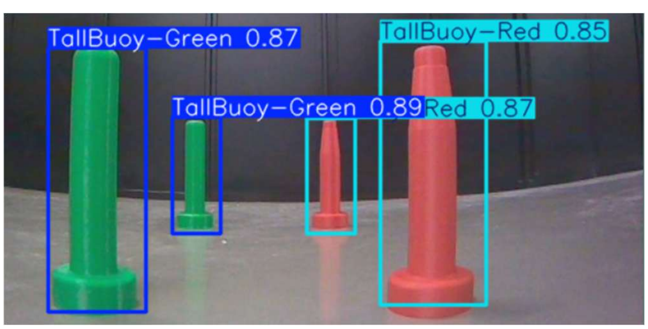


Fig. 4. Object Detection Output

2) Mapping: Using the ZED SDK [2] it is possible to measure the position of detected objects in the image relative to the camera and place these in a map. The map represents what is currently visible through the ZED X camera from a top view, including the classes of detected objects, and each cell represents a square with sides of 0.5 m. This is the approximate size of the largest obstacles we expect to interact with (navigation channel lane markers), and each obstacle is assumed to take up the nearest space on this grid regardless of its size. Positions relative to relevant objects within this map can then be used as waypoints for path planning.

3) Path Planning: Provided a point on the map and the boat's current position (using the current mapping methodology, this will always be the top center of the map), a straight line is drawn between these two points. If an object is within 1 m of a point on this path, the A* algorithm [5] will be used to adjust the path such that it avoids the obstacle. This allows for faster path planning in scenarios where there are no obstacles to avoid. To allow for a smoother transition between points, it is ensured that points are at least 2 m (two boat lengths) apart, after which the horizontal distance between the boat's current position and the next point in the path can be used to determine the boat's movement.

4) Following the Path: The Loon-E drives primarily using differential thrust, where the two propellers may rotate at different speeds to allow for turning. Turning can be further supplemented by the use of rudders which can be positioned at one of three positions: -35°, 0°, and 35° (selected as 35° is the proposed maximum angle for a rudder [6]). Both use the error signal found in the path planning step to determine the output, either through PID control or on/off control respectively. In the case where a path is unavailable (for example, if the object used to define the waypoint is not visible in the image), the boat will stop and turn until it is able to see the relevant object again. Other scenarios (such as circling buoys) may not require a path at all and can instead be done through hard-coded motor control.

III. Testing Strategy

Testing was done in various phases, starting with testing of individual sensors or subsections and expanding over time until the full system could be tested.

A. *Electrical Testing*

First, isolated testing on the propellers was done directly off the RC remote. The purpose of this test was to verify the current draw of the propellers against the specifications and to confirm that they were below the trigger point of the breaker.

Testing to verify proper power cutoff for the remote e-stop and the on-board e-stop was done. Similarly, it was ensured that the breaker would trip at above 30 A. This was done by powering the propellers in reverse at 20 V and removing the cap on the PWM signal sent to the ESC.

To safely discharge the 14.8 V lithium battery, a BMS that monitors each of its cells was used and a voltage sensor was placed on its output. To test the functionality of the setup without needing to discharge the battery to an unsafe level, a variable power supply was connected, simulating a cell. The voltage level was set to 4.2 V, the maximum level of a cell, then brought down to 3.0 V, the minimum level of a cell. At this point, the voltage sensor stops having a reading, displaying 0 V on the RC remote.

A PID control algorithm [1] and testing environment were also developed to demonstrate the basic form of boat control. In this setup, an IMU was used to gather heading using its magnetometer, as this was originally how the boat's heading would be gathered. A DC motor was used to turn the IMU until it was aligned with the target heading. The results of this test were a greater understanding on PID control, as well as tuning methods. It was determined that acquiring heading using the magnetometer is unreliable mainly due to the wires carrying fluctuating current through them. Various calibration methods for the magnetometer were tried with little success. The PID controller would be later used in the boat motor control node.

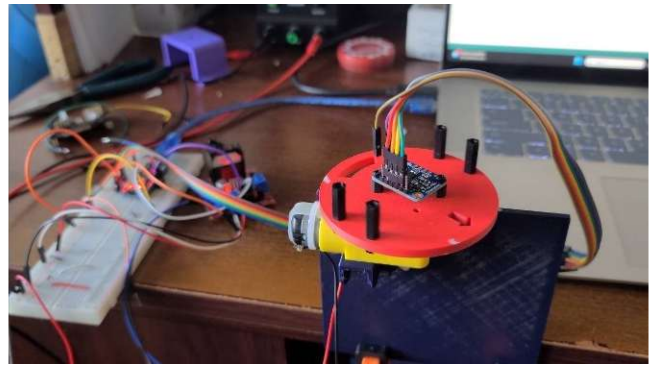


Fig. 5. PID and IMU test setup

B. *Object Detection Algorithm*

3D models of course objects were recreated and printed, specifically those of the tall red and green buoys. By running a preliminary program which loads the model and runs processed images through it, the accuracy of the object detection algorithm can be verified. The model used during testing would need to be updated such as to remove objects from the 2025 competition and add new objects for the 2026 competition. As such, the accuracy of this specific model is not as important as the verification that the overall methodology is effective.

Later, this was integrated into a preliminary version of the mapping and path planning algorithms to serve as a proof of concept and implemented in ROS 2. This was done with images from a single camera instead of the stereoscopic ZED X camera, and as such estimation of distance data was required. Errors encountered were a result of this estimation, but otherwise the methodology was shown to be effective.

C. *Simulation Environment*

The team has limited budget and test space, and by extension has limited access to course objects such as buoys and dock cubes. This means that it is not possible to fully recreate the course environment in real life. To remedy this, a simulation environment was developed using Isaac Sim. This software was selected due to its compatibility with the selected hardware, namely the ZED X camera.

After building the simulation environment in Isaac Sim, Isaac Lab was used to pull the simulation file in and run some number of training simulations concurrently. This allowed the boat to make thousands of navigation attempts in parallel to accelerate the learning process. Each epoch of training runs 50-200 parallel models, selects the best fit model based on a reward function, then begins the next epoch by loading the best fit model with an additional 50-200 iterations. The program runs 15 epochs for each training session, results are reviewed, and the training algorithm is adjusted before re-running the training in Isaac Lab.

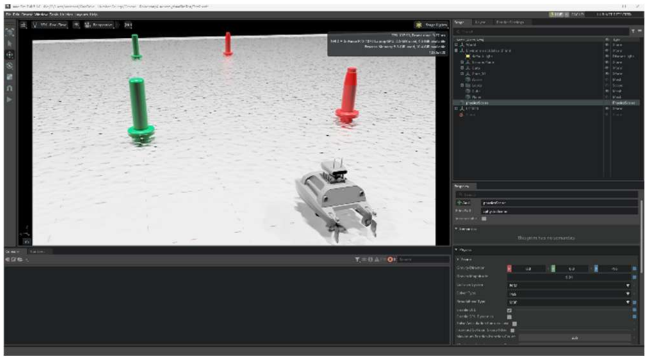


Fig. 6. Isaac Lab Simulation Environment

D. *Real-World System Testing*

To test the physical components of the boat, it was taken to a pond near Humber Polytechnic to verify the waterproofing of the system. This was done prior to the application of epoxy and consisted of placing the hull in the water for 10 minutes without any additional components. During this test, no water entered the interior of the boat where the electronics would be placed.

A second test was performed at a dock at Humber Bay, this time including any electrical components required to control propulsion. Key elements to be verified included the weight distribution and overall function of the electrical system. The boat was in the water for 15 minutes, during which it was observed that the boat was leaning back further than expected, even with the heaviest electrical components (ex. battery) being placed at the front of the boat. To account for this, any additional components should be placed at the front of the boat.

E. *Code Testing with Car*

Testing of the programming prior to the setup of the simulation environment was also performed using a car. This car, the JetBot ROS AI Kit [7], was considered a fair approximation of the Loon-E as the motors controlling the JetBot's two wheels would operate under the same logic as the motors controlling the Loon-E's two propellers through differential thrust (though tuning of program parameters would be required when translating the code from one vehicle to the other). The JetBot was used to verify the code used in Task 1 and was able to effectively detect the 3D-printed buoys and navigate the path in both directions.

Fig. 7. Testing with JetBot

Acknowledgment

Humber ASV would like to acknowledge the support of Humber Polytechnic professors who supported the team:

- Ali Taha provided general support
- Simon Heathcote provided funding
- Colin Buddin, Khaled Ibrahim, and Bruce McKinnon provided access to Humber equipment and facilities
- Joseph Tombe and Brigon Munkholm supported hull manufacturing
- Vlad Porcila supported PCB manufacturing
- Weijing Ma and Raji Subramaniam supported software development and provided the JetBot ROS AI Kit for testing
- Shaun Ghafari, Carl Oliver, and John Chu supported overall club operations
- Eric Forest supported website hosting and development.

Hanna O'neil, Hayden Knight, and Hartej Tapia also supported the production of the website and video.

Finally, the team would like to acknowledge the financial contributions of Dave Campbell and Jane Smith.

REFERENCES

[1] B. Beauregard, “Improving the Beginner’s PID,” Apr. 15, 2011. http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/

[2] Stereolabs, “Tutorial - Using Depth Perception,” *Stereolabs*, Nov. 12, 2025. https://www.stereolabs.com/docs/tutorials/depth-sensing

[3] Roboflow, “Auto Label,” *Roboflow Docs*, Jun. 23, 2025. https://docs.roboflow.com/annotate/ai-labeling/automated-annotation-with-autodistill

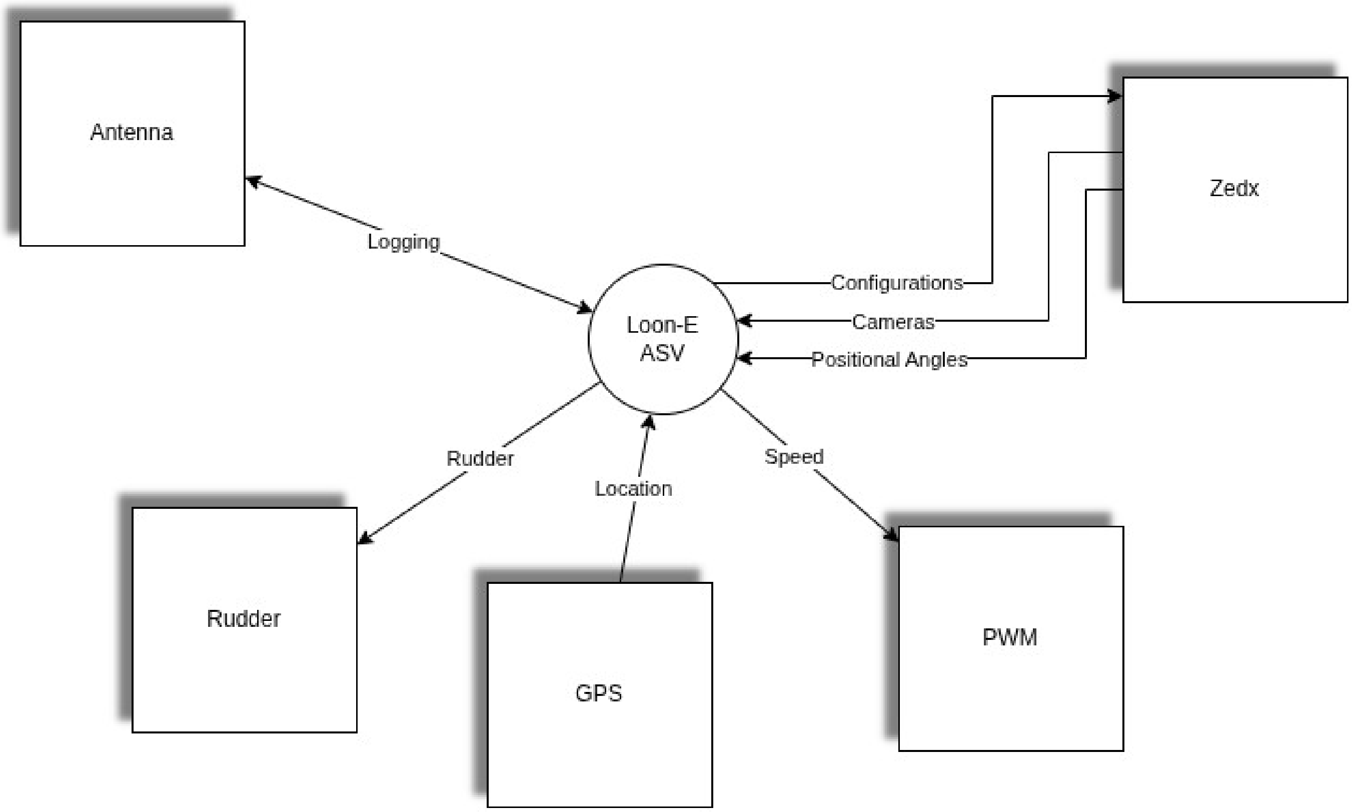
[4] G. Jocher, “Ultralytics YOLO11,” *Ultralytics YOLO Docs*, Nov. 15, 2025. https://docs.ultralytics.com/

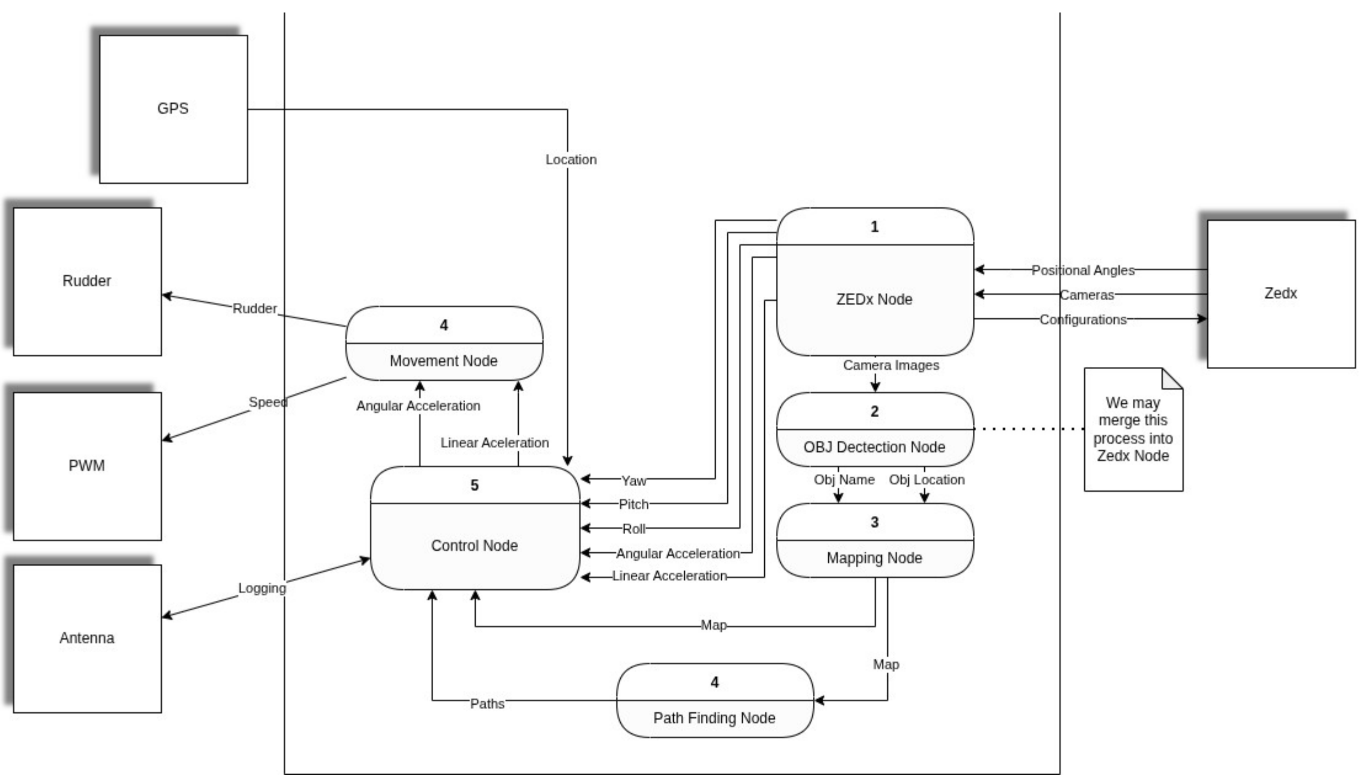
[5] P. Hart, N. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968, doi: https://doi.org/10.1109/tssc.1968.300136

[6] J. Liu and R. Hekkenberg, “Sixty years of research on ship rudders: effects of design choices on rudder performance,” *Ships and Offshore Structures*, vol. 12, no. 4, pp. 495–512, May 2016, doi: https://doi.org/10.1080/17445302.2016.1178205

[7] Waveshare, “JetBot ROS AI Kit,” *Waveshare Wiki*. https://www.waveshare.com/wiki/JetBot_ROS_AI_Kit

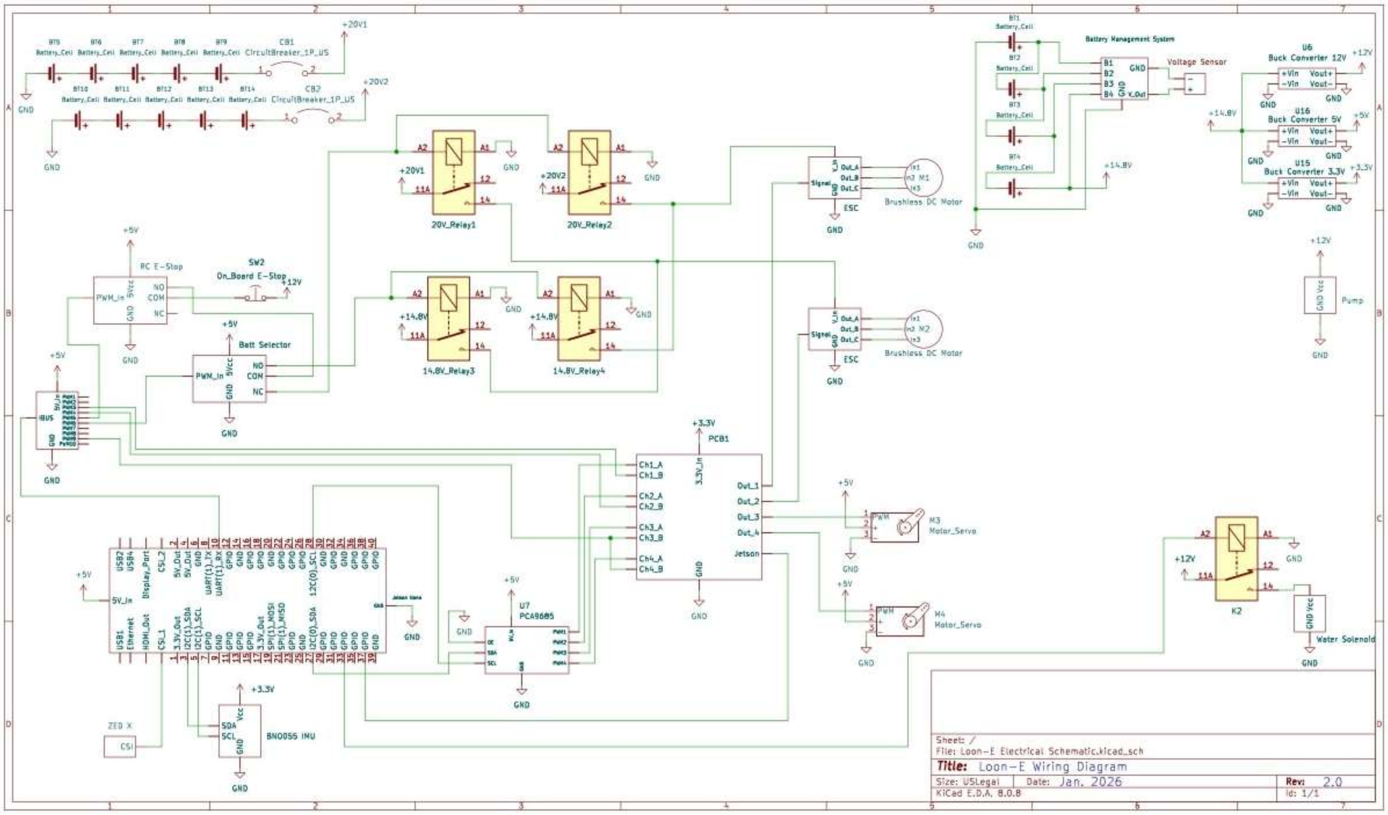
APPENDIX A: DATA FLOW DIAGRAMS



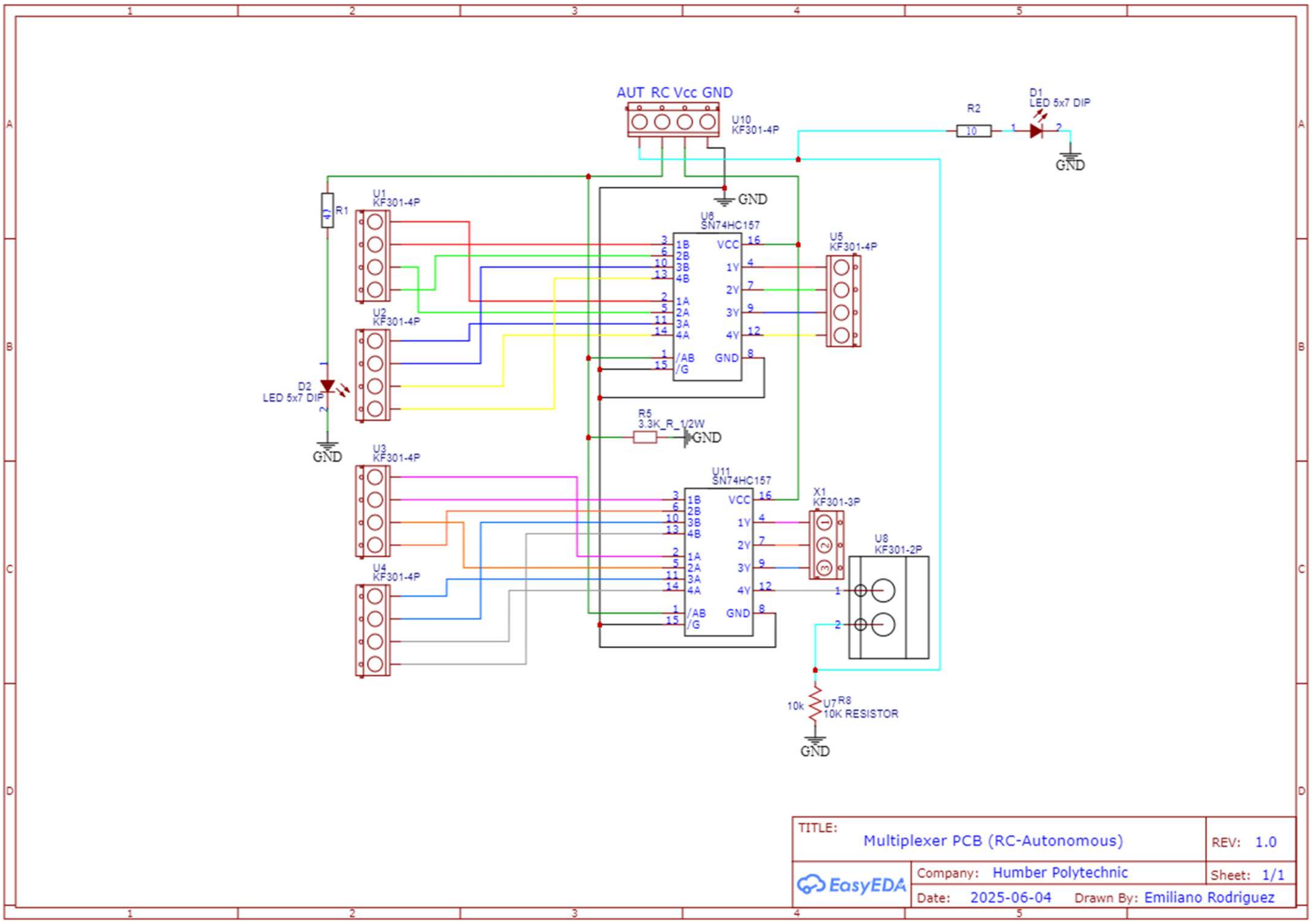


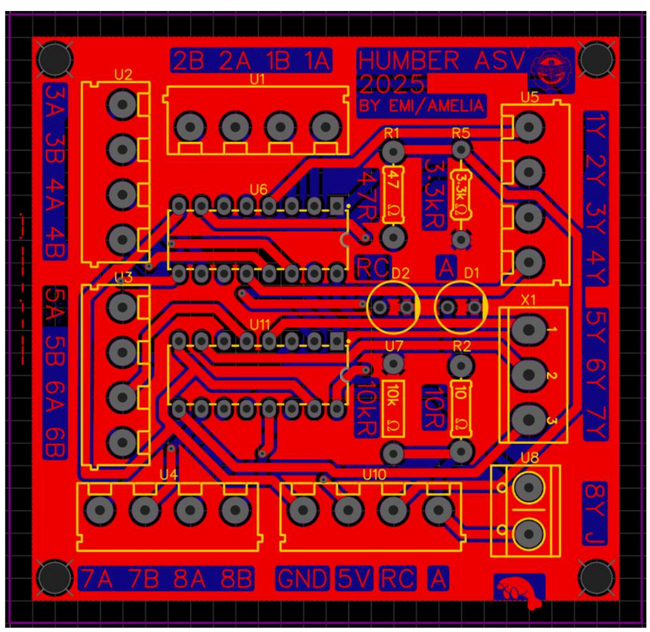
APPENDIX B: ELECTRICAL DIAGRAM AND PCB

Loon-E Electrical Schematic 2026

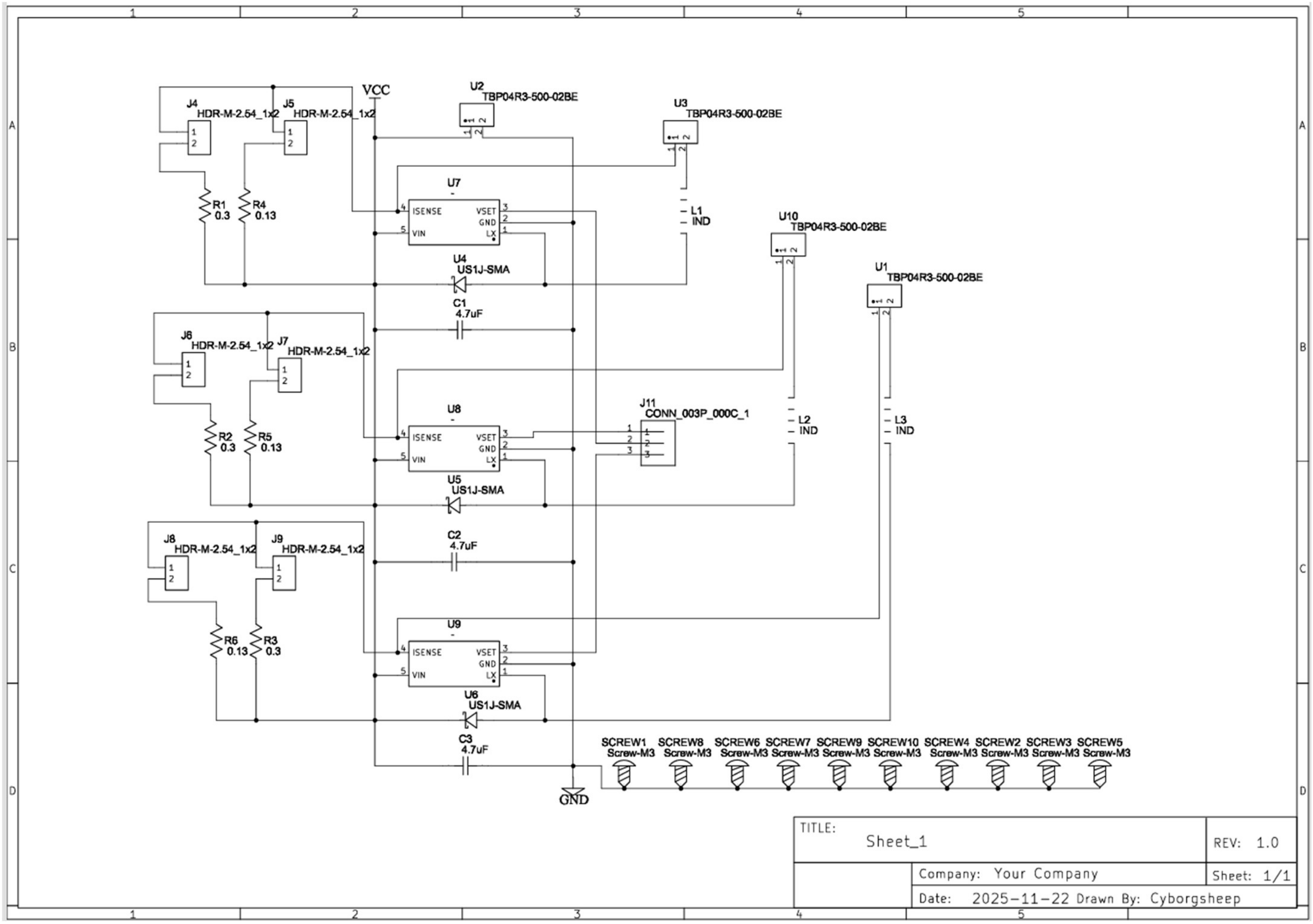


Multiplexer PCB

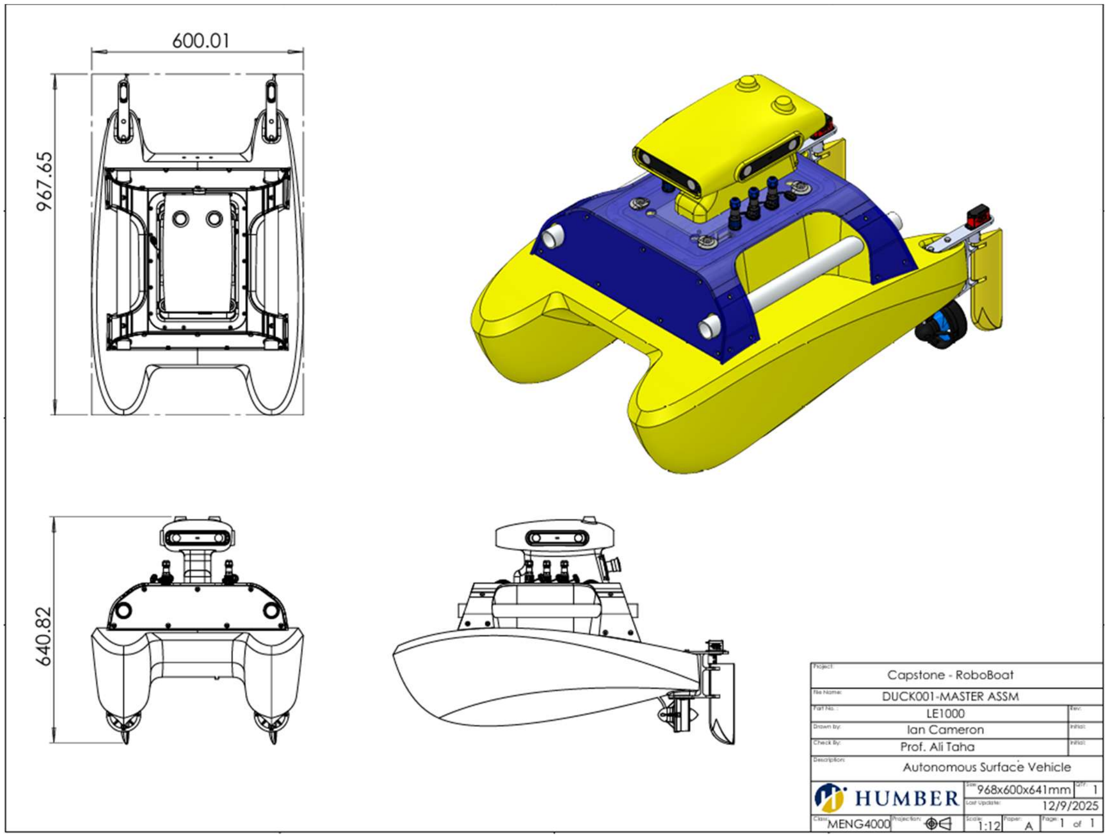


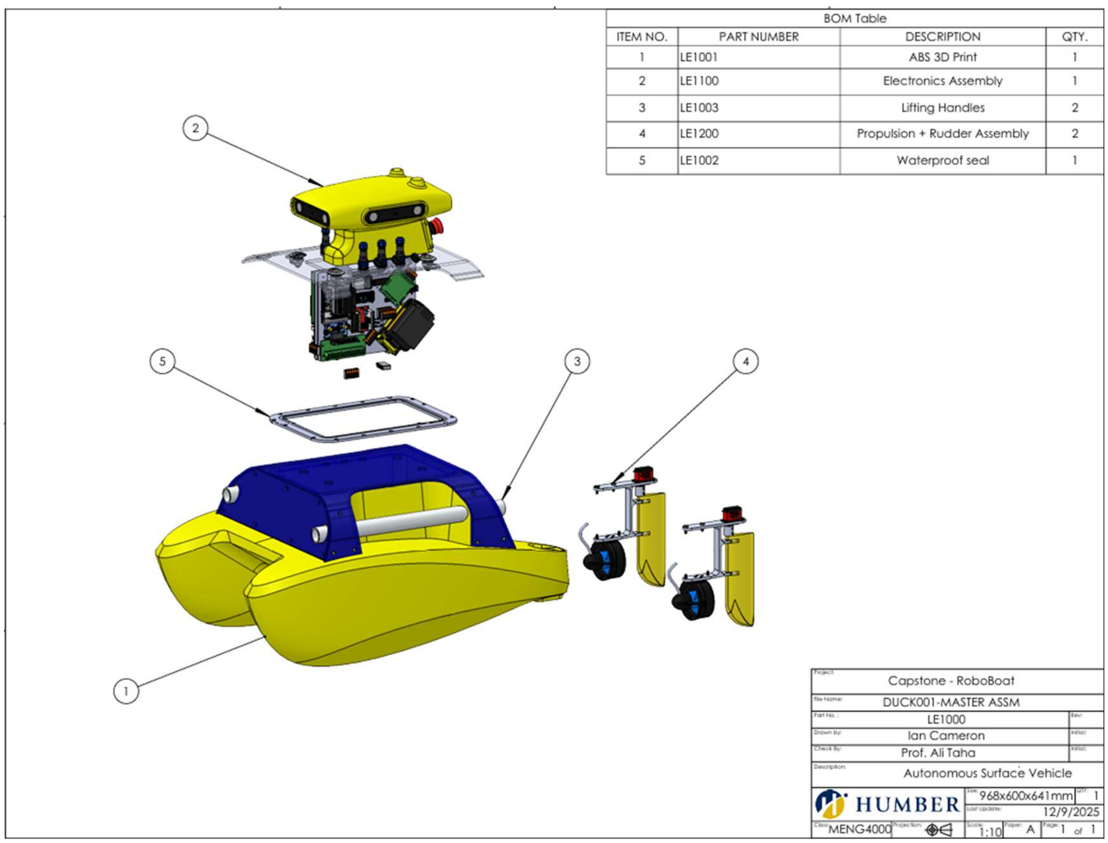


Light Indicator



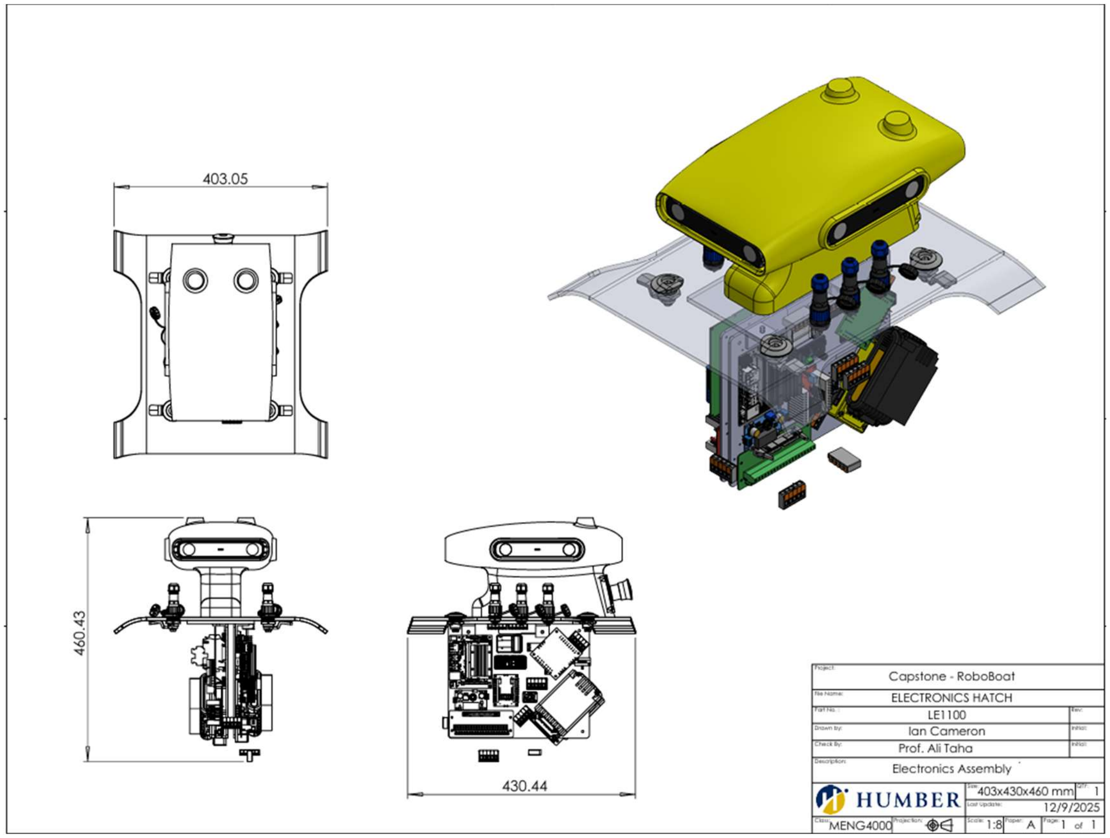
APPENDIX C: CAD DRAWINGS

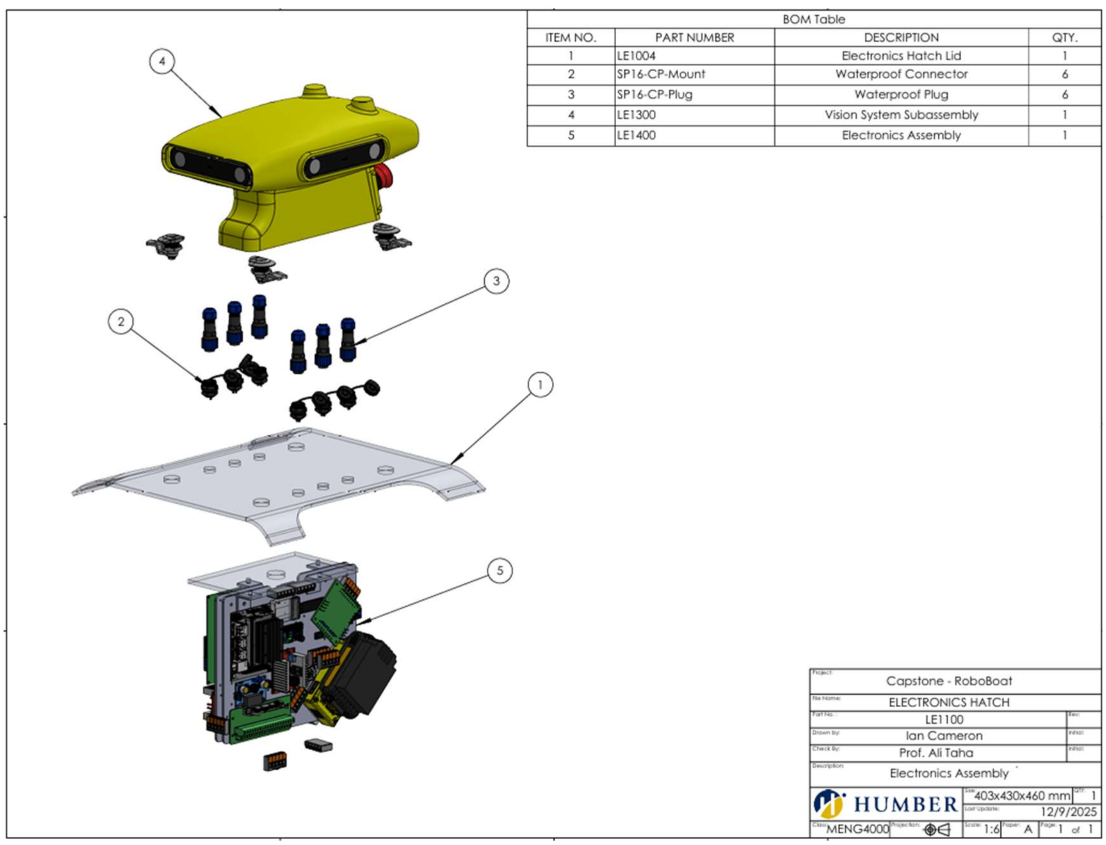




BOM Table

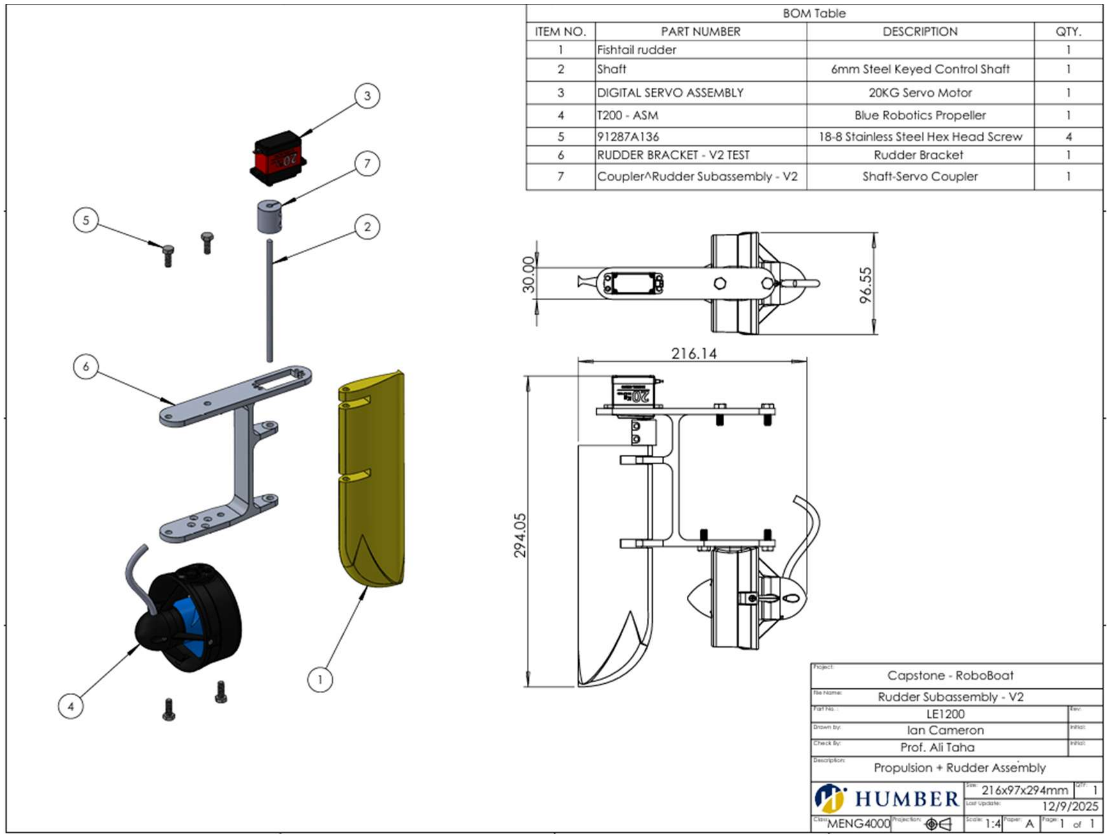
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	LE1001	ABS 3D Print	1
2	LE1100	Electronics Assembly	1
3	LE1003	Lifting Handles	2
4	LE1200	Propulsion + Rudder Assembly	2
5	LE1002	Waterproof seal	1



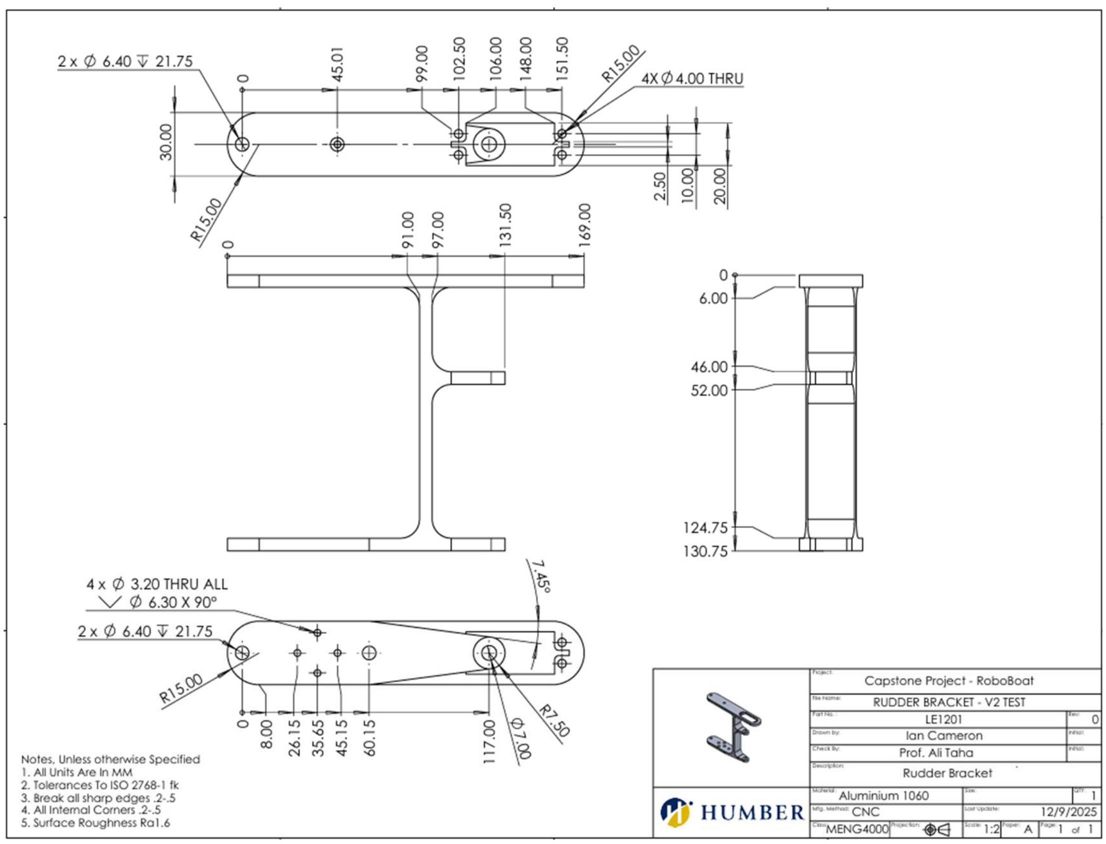


BOM Table

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	LE1004	Electronics Hatch Lid	1
2	SP16-CP-Mount	Waterproof Connector	6
3	SP16-CP-Plug	Waterproof Plug	6
4	LE1300	Vision System Subassembly	1
5	LE1400	Electronics Assembly	1



BOM Table			
ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Fishtail rudder		1
2	Shaft	6mm Steel Keyed Control Shaft	1
3	DIGITAL SERVO ASSEMBLY	20KG Servo Motor	1
4	T200 - ASM	Blue Robotics Propeller	1
5	91287A136	18-8 Stainless Steel Hex Head Screw	4
6	RUDDER BRACKET - V2 TEST	Rudder Bracket	1
7	Coupler^Rudder Subassembly - V2	Shaft-Servo Coupler	1



APPENDIX D: COMPONENTS LIST

SUBSYSTEM	COMPONENT	VENDOR	CHARACTERISTICS	COST/UNIT (USD)	QTY.
Propulsion	T200	Blue Robotics	31.21 A @ 20V	238	2
	Basic ESC	Blue Robotics	7-26 V	38	2
	Servo Motor	Miuzei	5V, 20 kg	15.32	2
Hull	Custom Hull	-	ABS	-	1
Remote Operation	FS-i6X	FlySky	10 channels	69.66	1
Navigation	Jetson Orin Nano	Nvidia	8GB	250	1
	ZED X Stereo Camera	ZED	Polarizer, 4mm	905	1
	ZED Link Capture Card	ZED	Duo	550	1
Electrical	Lithium-Ion Battery	Blue Robotics	14.8V, 18Ah	380	1
	20V Lithium-Ion Battery	DeWalt	20V, 6Ah	239	2
	Smart Dock	GoBILDA	20V	170	2
	Voltage sensor CVT01	Flysky	100V	17	1
	PCA9685	Adafruit	16 Channels	11.15	1
	Relay Module	YWBL-WH	30A	19.83	1
	HE Waterproof connector	HangTon	3, 4, 12 Pin	10	6
	Remote Control Electronic Switch	Fockety	3-30V, 20A	11.15	2
	E-Stop	McMasterCarr	2.5 A @ 24 V DC	52.44	1
	285-Series Circuit Breaker	Blue Sea Systems	30 A	75	2
	DC to DC voltage converter	Sayal Electronics	20A	18	3
	Custom Multiplexer PDB	-	16 input, 8 output	20	1
	Custom Multiplexer PDB	-	RGB	20	1