Blue Lobster

Technical Paper

One Degree North, Singapore American School (Singapore)

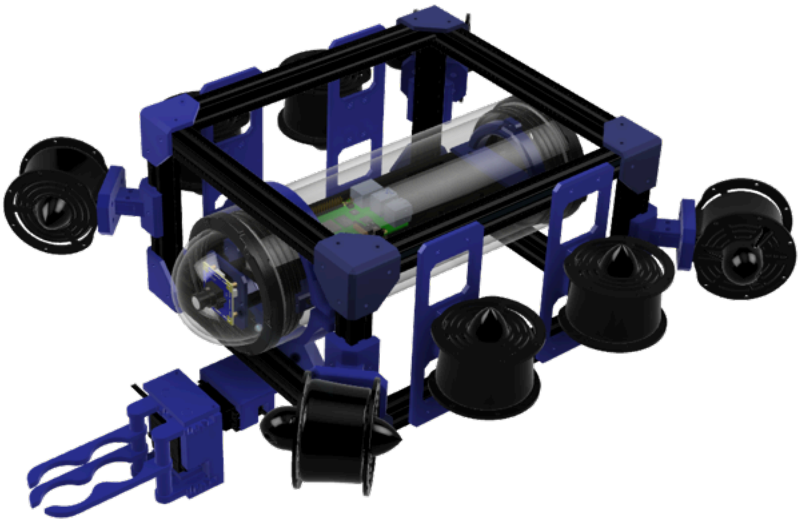


Fig. 1 - Finalized ROV

Leadership

Alan Chen (CEO)
Krishna Kundalia (COO)
Kaito Liventals (CTO)
Angela Soultati (CFO)
Genis Kim (Lead Mechanical Engineer)
Sumedh Mittal (Mechanical Officer)
Savitur Swarup (Lead Software Engineer)
Austin Thompson (Lead Electrical Engineer)
Vyjayanti Vasudevan (Editor-in-Chief)
Alice Yang (Marketing Director)

Mentors

Paul Booth
James Harvey
Simone Torreon
Emilio Orcullo
Jefferson Zhang
Aidan Lim

Mechanical Engineers

Billy Steckler
Jaiveer Bedi
Kirin Chadha
Michael Chu
Nathan Jacobes
Nicholas Lee
Royce Koh
Daniel Xu
Weihao Wang

Electrical Engineers

Layton Welker
Nick Lai
Shreyas Chopra

Software Engineers

Jinoo Na
Sid Maheshwari
Vir Rao

Acknowledgments

The One Degree North MATE team sincerely thanks Marine Advanced Technology Education II for allowing us to evolve as engineers as we construct and engage in the MATE ROV competition program.

Fig. 2 - Marine Advanced Technology Education II logo

Additionally, we extend our heartfelt appreciation to Singapore American School for generously providing us with the financial support, resources, and workspace necessary to participate in this competition.

Fig. 3 - Singapore American School logo

Finally, we wish to convey a special acknowledgment to our club sponsors, Paul Booth and James Harvey, along with Simone Torreon, Emilio Orcullo, Jefferson Zhang, and Aidan Lim, for their unwavering support and invaluable mentorship throughout our journey, guiding us through triumphs and challenges.

Table of Contents

Abstract

Rapid prototyping enables a team to develop a product iteratively, empowering members to take the initiative. However, effectively designing an ROV for rapid prototyping necessitates essential decisions regarding team organization and ROV design. During rapid prototyping, issues often arise due to unexpected interactions between prototypes and existing work and misunderstandings between teams. We addressed these issues by modularizing the ROV into distinct systems and delegating each to individual subteams. Furthermore, we modularized the individual core systems, allowing for the rapid replacement of standalone electromechanical and software components. We organized our team around weekly meetings to facilitate system integration, encouraging a mutual understanding of how the various systems operated. This approach enabled us to test several experimental solutions before integration, resulting in an innovative and reliable ROV. In designing the ROV, we aimed to be as cost-effective as possible, utilizing 3D printing and laser cutting for components whenever feasible.

Design Rationale

Mechanical

Overview

The design of Blue Lobster's ROV was centered around modularity. We emphasized convenience and flexibility in making multiple iterations to improve the mechanical design. To do so, we had to first decide upon the basic components of our ROV: cost, size, and weight. To have neutral buoyancy, ODN featured a lightweight frame (to counteract the drag of the tether) consisting of aluminum t-slot extrusions, aiding with maneuverability thrusters with integrated ESCs and WetLink penetrators that are quick to install. The negative space in the t-slot also allowed pieces to attach to the frame easily. The negative of using this material for our frame would be its cost. However, after team discussions, ODN determined that the size of the ROV being smaller meant that the cost of materials wouldn't be a significant factor. The thruster arrangement of eight thrusters also provided our ROV with all three degrees of freedom (pitch, yaw, and roll).

At the center of the mechanical design, the focus was the electronics mount located inside the Blue Robotics tube. The team began to design the mount by first laying out the design of our electronics and deciding on the placement of built-in gaps for our wires, electronics, and power distributors to avoid the wires getting tangled. To do this, we uploaded our design base, which consisted of purchased products such as our blue robotics tube, cameras, Adafruit Feather, etc. We also used self-built components such as thruster guards and strain relief for this specific purpose.

This approach allowed us to comfortably modify designs while still having a reliable structure if the modifications do not perform their functions as intended.

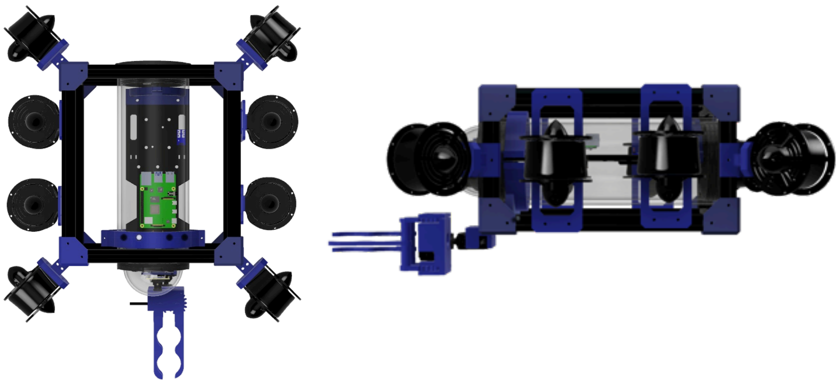
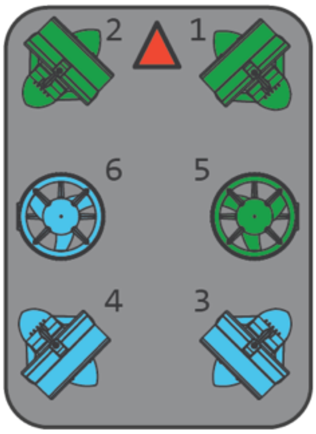


Fig. 4 & 5 - Finalized ROV rendered on Autodesk Fusion 360

Propulsion

At the beginning of the 2024 season, Blue Lobster used a three-part clockwise and three-part counter-clockwise design from ArduSub (denoted as green or blue), allowing for quick and efficient movements by minimizing the effect of torque generated on the ROV from turbulent flow. However, a limitation of this layout is that the ROV could only adjust its roll and yaw. Additionally, with two servos with high power consumption in our claw, very little power was going to the thrusters. As a result, we had to make sure that we had a backup plan in case either our servo or thrusters stopped working to complete Task 3.1. Having all three degrees of freedom in our ROV was a priority since, in case our servo stopped working, we would still be able to complete task 3.1 Probiotics 2As with relative ease. Based on a different thruster configuration from ArduSub, our current layout allows for pitch. Blue Lobster expanded on our previous layout limitations by adding two more thrusters on the left and right while allowing for an agile ROV. Despite the added cost of two more thrusters, it provided us with a more modular and reliable design.



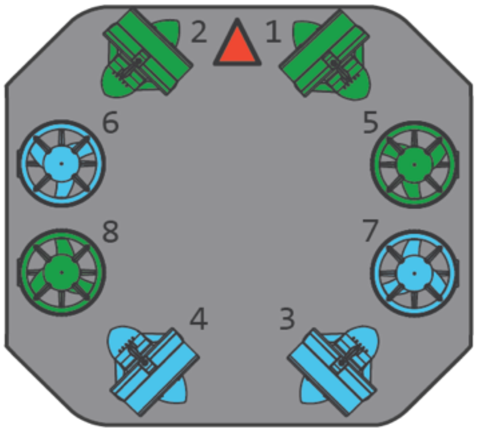


Fig. 6 - ArduSub's thruster configuration (initial layout on the right)
Fig. 7 - ArduSub's thruster configuration (modified layout on the left)

The ROV is equipped with eight Diamond Dynamics TD1.2 thrusters, capable of generating 1.2 KG of thrust. A compelling factor in our selection of this thruster model is its integrated waterproof electronic speed controller (ESC) outside the enclosure. By using these thrusters, we would no longer need to allocate space inside the closure for the ESCs. Additionally, the soldered connections on the ESCs frequently got damaged when opening and closing the enclosure, making the task of resoldering ESCs commonplace. Due to these challenges, Blue Lobster allocated a portion of our budget to upgrade our thrusters to the Diamond Dynamics TD1.2. This upgrade has saved our team considerable time and effort while preventing potential future problems related to the ESC.

Fig. 8 - Diamond Dynamics TD1.2 thruster

Enclosure

Blue Lobster used the ∅ 110mm x 300mm BR-100895 waterproof enclosure by Blue Robotics, rated for depths of up to 135 meters, to house the electronics. The enclosure comprises three main components: a tube, a dome, and an end cap fitted with cable penetrators to ensure a watertight seal for the components. Blue Robotics was chosen as the supplier for Blue Lobster's enclosure due to its affordability and established reputation in the industry since 2014. We ordered a new enclosure because last year's enclosure was quite damaged and no longer waterproof, rendering it unusable.

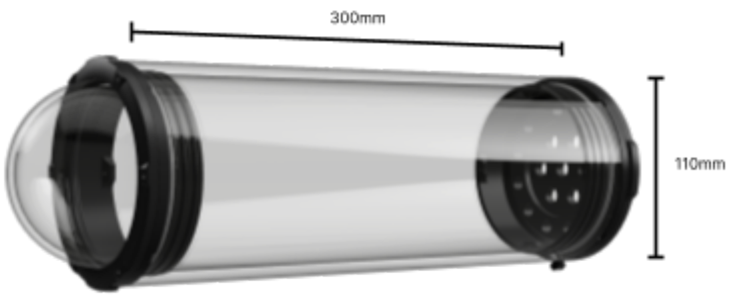


Fig. 9 - Fully assembled enclosure diagram

Using a tube over another type of enclosure, such as the Bocube used in previous years, offers several advantages. Firstly, the Bocube was only rated IP68, meaning it could withstand submerging 1.5 meters underwater for 30 minutes, less than the minimum required depth of four meters. Although the Bocube could be modified to meet these requirements, a tube is significantly more straightforward and reliably waterproof. However, purchasing a new enclosure did come with inevitable trade-offs, the most notable being that it is less environmentally friendly than reusing the enclosure from previous years. Nonetheless, by picking a good and reliable enclosure, we could reuse it in future years.

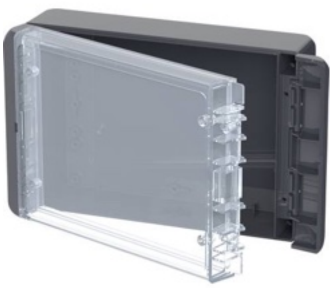


Fig. 10 - Bocube from RS Electronics (B 181306)

The ROV combines Blue Robotics WetLink cable and potted penetrators to prevent water from entering the enclosure. While WetLink penetrators were easier to work with and did not require any epoxy, making them reusable, they were expensive and incompatible with certain cables. Therefore, Blue Lobster employed potted cable penetrators for part of the wires, thereby minimizing costs while ensuring reliability.

Fig. 11 - Blue Robotics WetLink Penetrators

Fig. 12 - ROVMaker potted penetrators

Claw

We brainstormed some more when generating designs for the claw. One of the tasks listed in the manual would require our ROV to grasp onto an object and “unscrew” it. This resulted in us highly prioritizing the claw's ability to rotate. Following the idea of rapid prototyping, we began by creating a simple single servo claw that could function at the most basic level. Subsequently, we chose to alter the design so that it would have two servos, one to control the grasping function of the claw and one to rotate the claw. The claw underwent multiple iterations to reach its current form.

Initially, we settled on a laser-cut design using the ROVMaker 40 KG servo to open and close the grippers. As this was a first iteration, though functional, it had numerous issues, such as the inability to grasp larger objects, close fully, or rotate (required for Task 3.1 Probiotics 2). Furthermore, it necessitated the ROV getting close to the object it was attempting to grab, which was an issue for the driver.

Fig. 13 - Claw V1 (laser-cut)

This led to two more designs we 3D printed to test, which would provide a bigger advantage. These designs were similar, with the main difference being that one could rotate using an additional servo while the other could not. We did this because we were still determining if we would have a second servo available by testing time. This new design proved to be a substantial upgrade, allowing for more precise movements through the rotation of the claw and enabling the grasping of larger objects through a new gripper design inspired by the interlocking of fingers.

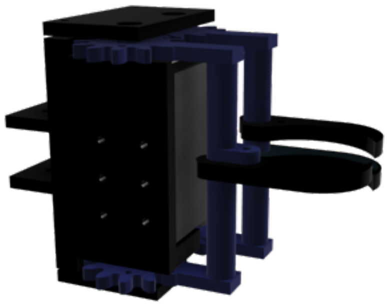
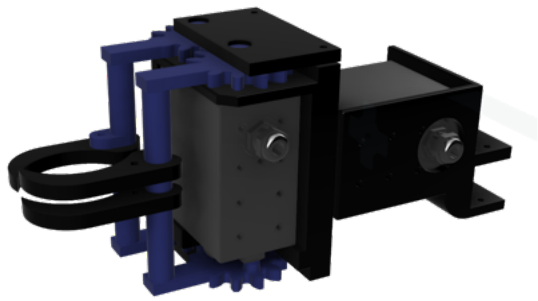



Fig. 14 - Claw V2 (3D printed, non-rotating on the left)
Fig. 15 - Claw V3 (3D printed, rotating on the right)

Although a significant improvement, this design presented several more problems. With the addition of another servo, the claw became too heavy and drew too much current. This led us to our final design, which followed the same general design as the previous claw but incorporated smaller, less powerful servos. We also redesigned the grippers to enable the driver to grab objects close and distant.

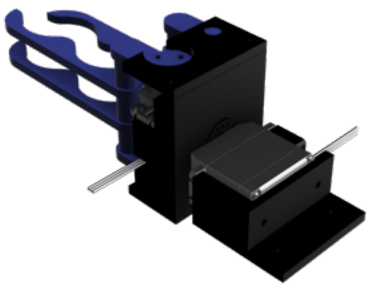


Fig. 16 - Claw V4 (3D printed, rotating)

Blue Lobster initially used the ROVMaker 40KG servo for the first three claw designs. This servo was waterproof up to 200 m and strong, with a peak torque of 40 N m at 12V, which provided a firm grasp on objects.

However, this servo was too powerful and could not detect when it was already grabbing ahold of an object, causing it to be in a “hard stop” and damaging the servo. As such, we used the weaker DSSERVO DS3225 with a peak torque of 25 N m at 6V since it was lighter and a good fit for our needs. However, the servos weren’t rated IPX8, so we needed to waterproof them manually. To do this, we followed a tutorial on YouTube (RCSubGuy, 2019), which required us to epoxy the outer casing together; fill the servo with oil to equalize pressure variations within the servo unit; and epoxy the screws shut.

Frame

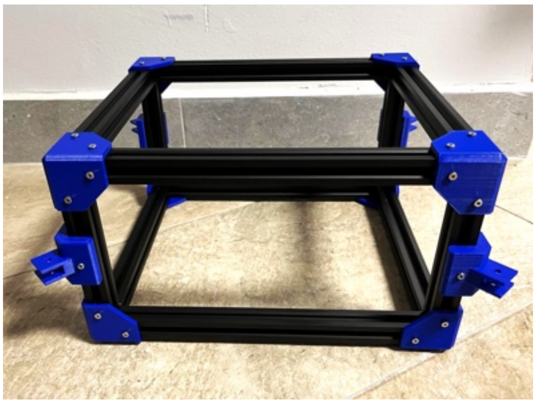


Fig. 17 - ROV frame without the enclosure or thrusters

This year, the team aimed to increase the flexibility of the ROV and expand its functions. Many of the tasks from this year require the ROV to be lightweight and compact. When brainstorming ideas, we found two options. The first would be to work with a PVC bot since it is low-cost, lightweight, and easy to fabricate. The Blue Lobster chose not to work with PVC this year because replacing large sections of the frame would be time-consuming and wasteful if we decided to make any changes. Additionally, to have neutral buoyancy with a smaller, more modular ROV (0.240,0.655,0.475 meters), we required a denser yet still lightweight material for our frame, which would result in a higher cost than PVC. As a result of this and the factors mentioned above we settled upon using aluminum T-Slot extrusions, allowing for a much more modular design, since components such as thruster guards and the camera can easily be attached or removed without permanently modifying anything.

Our teams previously used 3D-printed corner pieces and simple three-piece laser-cut Delrin corner pieces designed to slot aluminum extrusions into them to connect our T-slot extrusions. While the 3D-printed corner pieces looked nice and were entirely secure, they were often difficult to mount or remove, making it challenging to modularize the ROV. Although the laser-cut corner pieces were easy to change, they weren't sturdy and often fell apart. We used three side corner pieces from 3D-printed materials to address these issues.

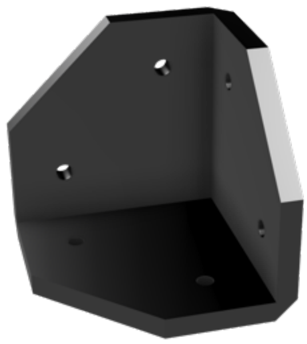


Fig. 18 - New 3D-printed corner mounts

Buoyancy

Blue Lobster aimed for neutral buoyancy to ensure stability and ease of control, with minimal force required to maneuver the ROV. The tether and claw rendered the ROV negatively buoyant, hampering its mobility. To counteract this, Blue Lobster decided to reuse foam flotation blocks from previous years, as they had proven to be a simple and effective solution for increasing buoyancy. Through trial and error, we used these blocks and adjusted the foam blocks attached to the frame until our ROV was near neutrally buoyant.

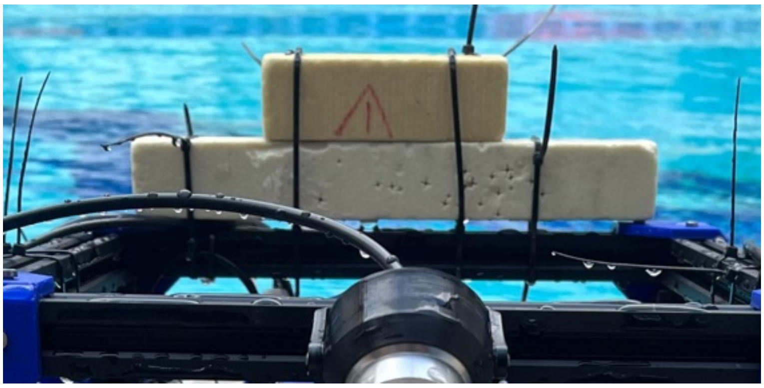


Fig. 19 - Foam blocks reused to counteract the weight of the tether

Pool noodles were also attached at one-meter intervals to prevent the tether from dragging on the pool floor and cut into 10-centimeter sections.

Payload and Tools

In previous years, ODN used one camera for an ROV. However, only having one camera limited our field of view of the surrounding environment. This made it difficult for the driver to gauge where they were in the pool and where to go to complete the next task. As such, we decided to include two cameras in our ROV.

The first camera, a non-waterproof USBFHD06H USB camera, is mounted inside the enclosure and positioned parallel to the side of the ROV frame. This allows us to get a larger view of the surrounding environment, which is useful for navigating and photogrammetry.

Blue Lobster also mounted a more expensive waterproof IP69K exploreHD 3.0 FHD camera by DWE, aimed at a 40-degree angle of depression. Despite the price, the benefits gained with the camera's quality significantly outweighed its cost making it much easier to see what’s happening. This is especially important for the claw, as the driver requires a clear view of the claw to see what they are picking up. Furthermore, equipped with a fisheye lens, the camera lets the driver view the environment and claw.

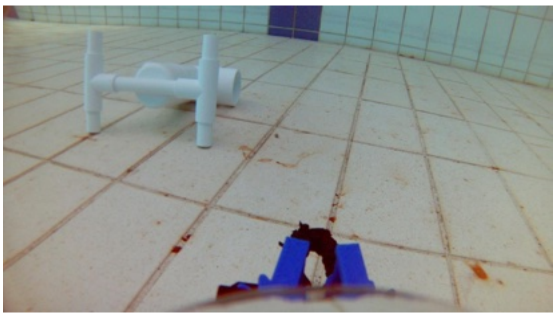
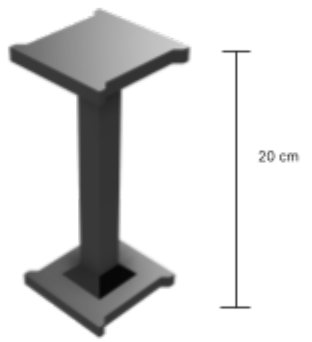


Fig. 20 - Camera feed from the ROV

In addition to the cameras, we used some sensors to help the ROV gauge its position and orientation. For example, we used a BNO055 gyroscope and accelerometer along with a depth sensor from ROVMaker, enabling us to use PID to control the ROV, which is further explained in the PID section of this document. For the SMART Cable task (Task 2), we used BlueRobotics' temperature sensor with ±0.1°C accuracy, as we required an accurate temperature reading.



As for the 3D coral model reconstruction task, a reference point must be placed next to the object to scale the scanned model. Blue Lobster opted for a simple design: a long rectangular prism with large blocks at both ends totaling 0.20 m long. This design facilitated the driver's ability to quickly grab or remove the object, regardless of its orientation.

Fig. 21 - Reference object used to scale model generated using photogrammetry

Sourcing Materials

By striving to create or 3D print as many components in-house as possible, Blue Lobster rapidly prototyped differing designs and significantly reduced costs. Producing these elements within the school premises allowed us to reduce transportation's environmental and financial costs, as there was no need for delivery. For example, while commercially available corner mounts for aluminum T-slot extrusions are available, our team was against incurring shipping expenses or creating a substantial carbon footprint. Therefore, we printed cutouts in CAD using our school's 3D printer. Although Blue Lobster aimed to fabricate as many components as possible, some had to be acquired from a reputable supplier to ensure reliability. Furthermore, Blue Lobster repurposed several materials, such as the foam blocks and tether from the previous year, as they had proven effective and required no further customization.

Electrical

Overview

Blue Lobster's electrical system is designed around maximizing expandability and reliability to help create multiple iterations. Regarding expandability, the ROV's internal mounting plate leaves extra space for cable management and additional devices. Furthermore, we delegated standardized power connectors for each device for quick assembly and disassembly, and our electronics design opted for reliability whenever possible. This is reflected in our decision to use Ethernet – a protocol with much support for development despite a larger overhead. Reliability was also the main concern for our thruster and servo choices.

Thruster and Servo Interface

Diamond Dynamics' TD1.2 thrusters have built-in electronic speed controllers, so Blue Lobster did not require an ESC housed within the enclosure. To control the eight thrusters and the two servers from ROVMaker that comprise the claw, we opted to use an Adafruit 8-channel PWM FeatherWing along with hardware PWM (pigpio) on the Raspberry Pi 4 for the remaining 2 PWM outputs. At first, we attempted to use only hardware PWM; however, the overhead was too high, leading to the PWM output not being consistent after five or so PWM outputs running concurrently. We opted for the PWM FeatherWing instead of Adafruit's 16 PWM output board because we had extras in stock, even though they both use the same PWM controller (PCA9685).

Communications

Blue Lobster uses 100 Mbit/s UDP over Ethernet to communicate with the surface station. Compared to a wireless connection and a serial RS485 that Blue Lobster used last year, it provides more reliable, low latency, high-throughput interface streaming functions to communicate with the Raspberry Pi 4. Ethernet also allows access to the camera stream, while serial communication offers two functions for receiving and transmitting commands. Ethernet also allows access over SSH to remotely execute commands and upload code without disassembling the tube. We set up an external router to SSH into the Raspberry Pi, simplifying connecting our devices to a router.

All the sensors and the PWM FeatherWing are connected to the same I2C bus. Only managing one communication protocol between the Raspberry Pi and the other devices on the ROV simplifies the implementation side of the Raspberry Pi and allows for easier debugging problems since a logic analyzer can be hooked onto the bus to see all the data being sent. In addition, unlike UART, I2C has built-in error detection.

Power System

The Diamond Dynamic thrusters individually draw a maximum of 7.5A. Blue Lobster employs a configuration with eight thrusters, necessitating limiting the draw for each thruster by reducing their maximum speed to 28%. This measure prevents stuttering in the thrusters and ensures their reliability. A buck converter was used to supply a power rail at 5V for the Raspberry Pi and other sensors.

Fig. 22 - LM2596 step-down buck converter

Fuse Calculations	
Component	Current
RPi4B	1.2A
BNO055	0.01A
PWM controller	0.05A
Depth Sensor	0.0125A
Temperature Sensor	0.014A
2x Camera	0.4A
8x TD1.2 Thruster (software limited)	16.8A
2x Servo	4.6A
Total	**23.087A**
Overcurrent (total x 150%)	34.63A
Fuse	**25A**

Fig. 23 - Fuse calculations

Tether

Blue Lobster connects to the surface control station using a 15-meter-long tether composed of four cables. We chose this tether length since we didn't want it to be long enough to make it difficult to manage, leading to the tether getting tangled, nor short enough not to allow the ROV to reach tasks further. By implementing an effective tether management protocol, where the designated tether manager would give just the right amount of tether, we could also minimize the drag of it against the pool floor. We settled on three two-core 14-gauge wires for power, providing Blue Lobster with 12 volts at 25A. Instead of using one or two wires, using three significantly reduces the electrical resistance, minimizing the voltage drop in the tether and enabling the thrusters and other electrical components to operate under higher loads.

The fourth cable is a category-6 twisted pair 8-core cable designed for Ethernet transmission. By employing a twisted pair Ethernet cable, Blue Lobster mitigates the risk of crosstalk between wires, ensuring more reliable communication over high-speed protocols like 100 Mbps ethernet, which is utilized to communicate with the Raspberry Pi.

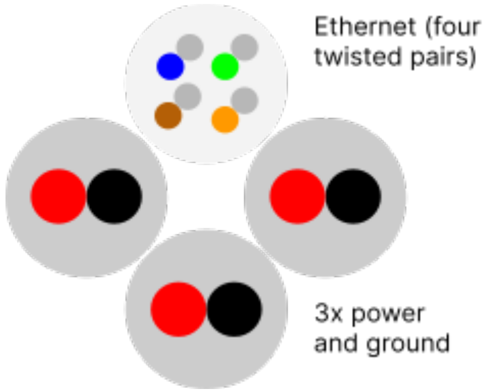


Fig. 24 - Tether cross-section diagram

Software

Overview

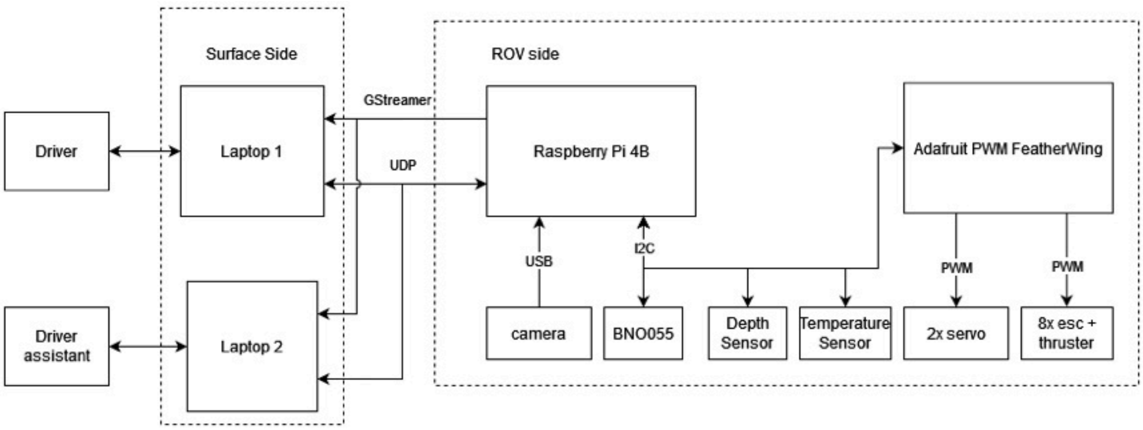


Fig. 25 - Software flow diagram

Blue Lobster's software architecture is split between three devices: the two surface laptops (connected to a VR headset) and the Raspberry Pi 4.

Laptop 1 runs the graphical user interface. Laptop 2 runs computer vision algorithms and controls autonomous tasks. This split in task delegation between the two laptops was made to offload GPU-intensive tasks to a computer with one and not impact the core functionality of manually controlling the ROV. The Raspberry Pi manages the ROV's high-level inputs and outputs, specifically receiving commands from the laptop to move the thrusters and servos, running the PID control software, streaming cameras, and controlling I2C outputs to send and receive data on the various devices on the ROV.

Each device is responsible for its layer of abstraction. The surface laptops operate on the most abstracted layer, with access to reliable commands for controlling the claw, setting target orientation, and more. The Raspberry Pi implements these functions. This allows our software to be developed efficiently and easily expandable, as it is clear what inputs and outputs are expected of each stage and device.

Cameras and Imaging

Since the driver cannot physically view the ROV while it is active, having low-latency and reliable cameras is important to ensure the driver can effectively complete tasks.

Although alternative solutions like MJPEG-Streamer FFmpeg had advantages and disadvantages, we ultimately settled on GStreamer because of its modularity. GStreamer lets us use highly performant lightweight nodes centered around a graph-based components system to build simple pipelines, allowing for almost instantaneous modifications to the camera stream to fit any possible changes to the ROV.

The final pipeline we settled on uses UDP-based streaming. Blue Lobster chose UDP over TCP for its lower overhead. We then encoded the video data in RTP, a standard for streaming video feeds over IP, and sent it to the client. From there, we decoded the pipeline and displayed it in the GUI. We also have a state where we can crop and record a frame rate-adjusted set of images for photogrammetry.

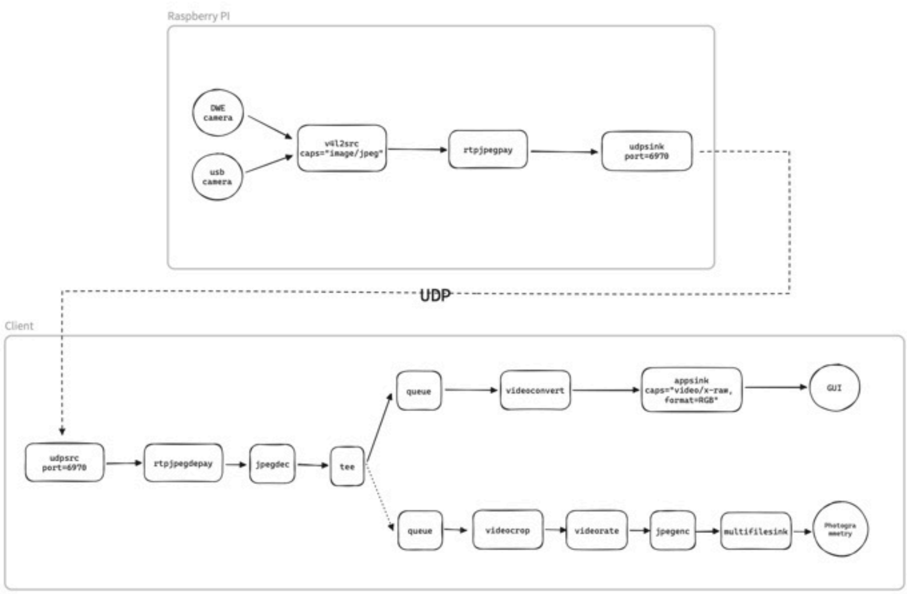


Fig. 26 - Diagram of the camera to the client data pipeline

PID and Thruster Control

Blue Lobster uses PID and a PID controller to automatically orient the ROV's roll and yaw angles and adjust for X, Y, and Z positions. Data for PID comes from the IMU, and the controllers were tuned using the Ziegler-Nicholas method (Ziegler, 1942). Using PID helps the ROV resist external interference by allowing the system to automatically correct deviations from the target orientation and velocity vector the driver sets.

Computer Vision and AI

The primary task of computer vision is to create a 3D model reconstruction of the coral restoration area. The ROV uses video feed from the exploreHD 3.0 Camera to make a 3D reconstruction of the surrounding area.

When researching the task, we saw two possible solutions: Using PhotogrammetrySession in Apple's RealityKit or building a photogrammetry software ourselves. As this is the first year One Degree North is using photogrammetry in our ROV, and it was beyond our technical capabilities to build photogrammetry software from scratch, we decided to use Apple's PhotogrammetrySession, which is a simple and widely-used method of generating 3D models within applications on Apple devices. However, a challenge we faced with this method was that the video needed to be separated into multiple frames to build a realistic image, which is streamed directly to the auxiliary controller for tasks like photogrammetry. We solved this by overlapping the images provided and creating landmarks (or "scenes" generated in 3D space), which are stitched together to create a model of the object that is scanned.

GUI

The GUI serves as the primary interface for the ROV. From controlling the ROV's thrusters to reading connected sensor data, the client software must fulfill requirements to meet the team's standards. Blue Lobster had three main criteria when choosing the company's GUI software: minimal memory impact, highly adaptable, and low level. When the team explored libraries such as Qt and React, we found that they didn't meet the requirements above since they were either too rigid or took up too much memory. Thus, we decided to build our GUI on the ImGUI framework since ImGUI met our requirements and provided low-level access to integrate with GStreamer.

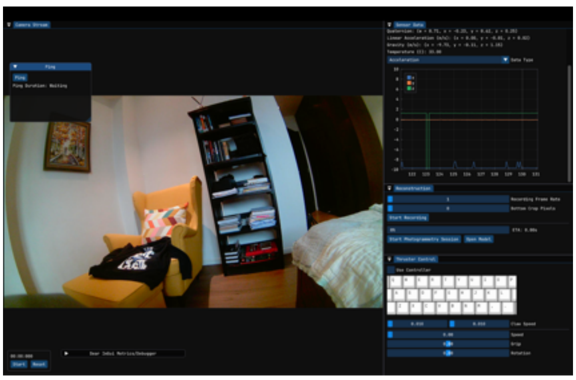


Fig. 27 - Completed GUI with camera stream

The GUI was designed with flexibility to enhance the user experience. We integrated every feature the GUI required into a separate ImGUI window through the docking branch of the ImGUI framework, allowing the driver to customize the GUI. Furthermore, windows are semi-transparent, so crucial information in the camera feed can be viewed at all times.

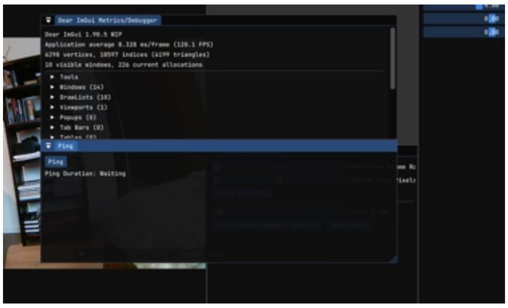


Fig. 28 - Translucent overlay of windows

Fig. 29 - Grouped windows

In addition to the layout, the GUI can be customized (including the color scheme, look, and feel of elements like buttons and frames)at any point through a configuration option tab. The GUI's fonts are embedded in its memory, so the GUI will display the same font with the same quality regardless of the system it is run on.

Blue Lobster's GUI also has a series of utilities to assist in debugging. As the ROV communicates over UDP, the client must know the server's correct IP and vice versa. To test the connection, we have a ping button to send a request to the server and wait for a response, measuring the duration. A debugger also informs the client about the memory allocated for the application, the elements drawn, and the framerate. For the camera stream, the server on the PI automatically starts the GStreamer pipeline described earlier. The GUI branch of the pipeline then renders to an app sink element, where every frame is polled for a new sample from the source. This is written to an OpenGL texture and drawn to the ImGUI window below.

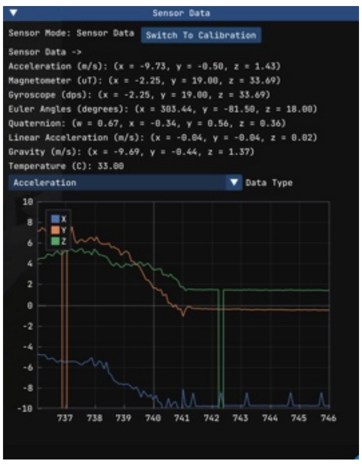


Fig. 30 - Debugger and ping windows on the GUI

The sensor data window receives and displays data for the BNO055, which sends continuous updates about the sensor to the window with a real-time graph to give feedback to the driver over time. For the photogrammetry task, Blue Lobster implemented two windows. The first window records at a modified frame rate and size, and the second is to process the recording into a 3D model that the driver can easily view.

The control section lets the driver control the ROV. Different keys are bound to the ROV's movements, which are visualized on the GUI. The speed sliders allow the driver to adjust the thrusters' speed. The remaining sliders enable the driver to easily control the claw's rotation and grip.

VR Implementation

This year at One Degree North Robotics, we experimented with VR technology. Inspired by drone pilots' use of headsets for enhanced situational awareness, a member suggested incorporating VR to improve visibility and offer the driver a clearer, more natural perspective. This helped facilitate easier control of the ROV when completing tasks previously hindered by a limited perspective. We chose Immersed as our desktop virtualization software due to its ability to utilize the same desktop-based client, making the switch to VR much easier. Furthermore, as one of our members already had a Meta Quest 2 that they were willing to lend to One Degree North, we did not need to purchase an additional headset.

Waterproofing

Research

Because of previous challenges with waterproofing, we dedicated more time to researching solutions from leading providers such as Blue Robotics, ROVMaker, and RS Electronics. Blue Lobster chose these three companies because they were established and reputable and had proven to work well for other teams in previous years.

While deliberating between these three options, Blue Lobster ultimately settled on three major categories to consider: usability, reliability, and cost. Ultimately, we found that Blue Robotics matched all three criteria with a manual that addressed our waterproofing concerns. While a Bocube offered promising features, drilling and installing cable glands would’ve compromised its waterproofing integrity.

Testing

The team conducted vacuum tests using the MityVac vacuum pumps to test the enclosures ordered. For Blue Robotics enclosures, the manual instructed us to pull a vacuum of 10-15 inHg for at least 15 minutes, ensuring the value does not go down by more than 0.5 inHg, which tested the enclosure at its maximum rated depth.

Fig. 31 - MityVac vacuum pump used for testing

This reduced the time and effort required to verify if the enclosure was fully waterproof, as we weren't required to conduct frequent tests in the pool.

Safety

Overview

At One Degree North, we take safety very seriously. While working on or using the ROV, there is always the threat of something going wrong. To mitigate this risk, Blue Lobster has implemented several strictly enforced measures.

Members always wear appropriate personal protective equipment when working on the ROV, whether soldering, cutting, or drilling. For example, members can protect themselves from solder splatter by wearing safety glasses. To protect members from ear damage with loud equipment such as saws, members working in the workshop must wear ear protection whenever the noise reaches 70 dB.

Fig. 32 - Paul Booth mentoring a SAS ODN robotics member

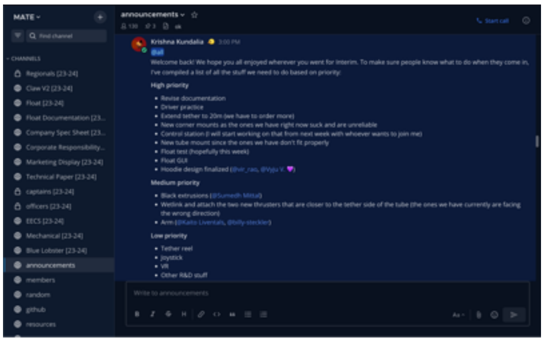
The ROV also poses many hazards to the crew working on it while it is operational. To mitigate these risks, Blue Lobster implemented thruster grates to reduce the chance of entanglement, a 25A fuse to protect against overdraw, and an emergency stop switch built into the control station to quickly and safely disable the ROV.

Safety Procedure Checklist

The following precautions must be taken to ensure the smooth operation of the ROV:

ROV	Ensure that the enclosure is fully waterproofed and sealed	Ensure that all cables and wires are not exposed	Ensure that all electronics properly function	Ensure that nothing is loose on the ROV (screws, electronics, attachments, etc.)
Control	The Control Station is at least one meter away from the pool		The anti-splash barrier is placed alongside the Control Station.	
Driver	Inform the crew when the ROV is turned on		Test all thrusters and servos to ensure all mechanisms function correctly.	
Crew	Have towels prepared at all times and aid in tether management so they don't weigh down the ROV.			
During Run	Before retrieving the ROV, turn off the power and wait until the ROV stops running.			

Teamwork



Since its founding in 2009, Singapore American School's ODN MATE robotics team has been working hard to bring marine education to the school. The school provides us with a budget, and we design the ROV, source parts, and build it ourselves.

Fig. 33 - Announcement on the MATE communications server detailing tasks

The lab is open Monday through Friday from 3:00 pm to 6:00 pm. Our typical workday includes team meetings, responsibility groups, and areas of specialization (referenced below). This ensures that all members have a holistic understanding of our ROV while keeping everyone productive and on task. To do this, we needed to ensure that members could efficiently collaborate to complete tasks.

This year, we chose to condense our entire program into one team. As such, Blue Lobster had several members who needed to be organized to ensure the team was working efficiently. Blue Lobster used several strategies, including our Robotics communications platform, Mattermost, to inform members of the remaining tasks. If a member ever wants to order a part, they must add the exact part, their reason for wanting it, and its cost to the resource management spreadsheet. This is then verified or rejected by our captains once they see whether it's within our budget, whether or not it can be reused and integrated into our ROV holistically, and whether it's a resource we already have by checking with our sponsors. James Harvey and Paul Booth. If approved, the order is placed by our resource manager, Simone Torreon.

We also used a whiteboard to assign members to specific tasks with deadlines. This helped members understand their roles in the team and complete work promptly. Those in leadership roles would also guide and assist newer members to ensure that all members contributed to the ROV.

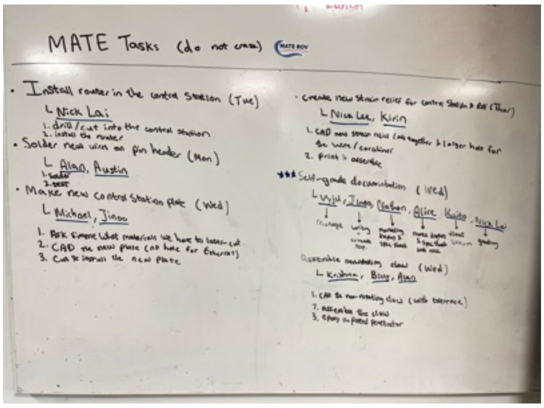


Fig. 34 - Schedule in the robotics lab assigning tasks to members with deadlines

Team Structure

Over the past few years at ODN Robotics, we've changed our team structures several times. With members having varying levels of experience, we try to accommodate by having two to three months dedicated to training in their respective fields at the beginning of each school year. We first gauge members' interests and categorize them as electrical, mechanical, or software. Two years prior, we had more divisions among these categories: instead of electrical, we had systems that required knowledge of Arduino C and separated software into backend and frontend programming. However, we later switched this system to account for the different time commitments of members. For example, front-end members didn't have to wait for back-end members to join meetings to be able to finish their job; instead, with basic knowledge in all software languages, they could work productively even when not all team members were present.

This year, in MATE, we organized the roles according to the above categories (electrical, software, and mechanical). We then separated members into groups of two to three with an assigned Responsibility Manager. This manager creates a channel in Mattermost (One Degree North's communication platform) and ensures all members are on task with the schedule (Fig. 34).

Team

Krishna	Billy Steckler	Nick Lai	
Kaito	Jaiveer Bedi	Nathan Jacobes	Savitur Swarup
Alice	Layton Welker	Sumedh Mittal	
Alan	Daniel Xu	Michael Chu	
Austin	Nicholas Lee	Weihao Wang	
Genis	Kirin Chadha	Royce Koh	
Angela	Shreyas Chopra	Sid Maheshwari	
Vyjayanti	Jinoo Na	Vir Rao	

Citations

Ziegler Nichols Method. (n.d.). Retrieved from https://www.sciencedirect.com/topics/computer-science/ziegler-nichols-method#:~:text=The%20Ziegler%2DNichols%20Method%20is,transfer%20function%20with%20dead%20time.

Building a Vehicle Frame. (n.d.). Retrieved from https://www.ardusub.com/quick-start/vehicle-frame.html

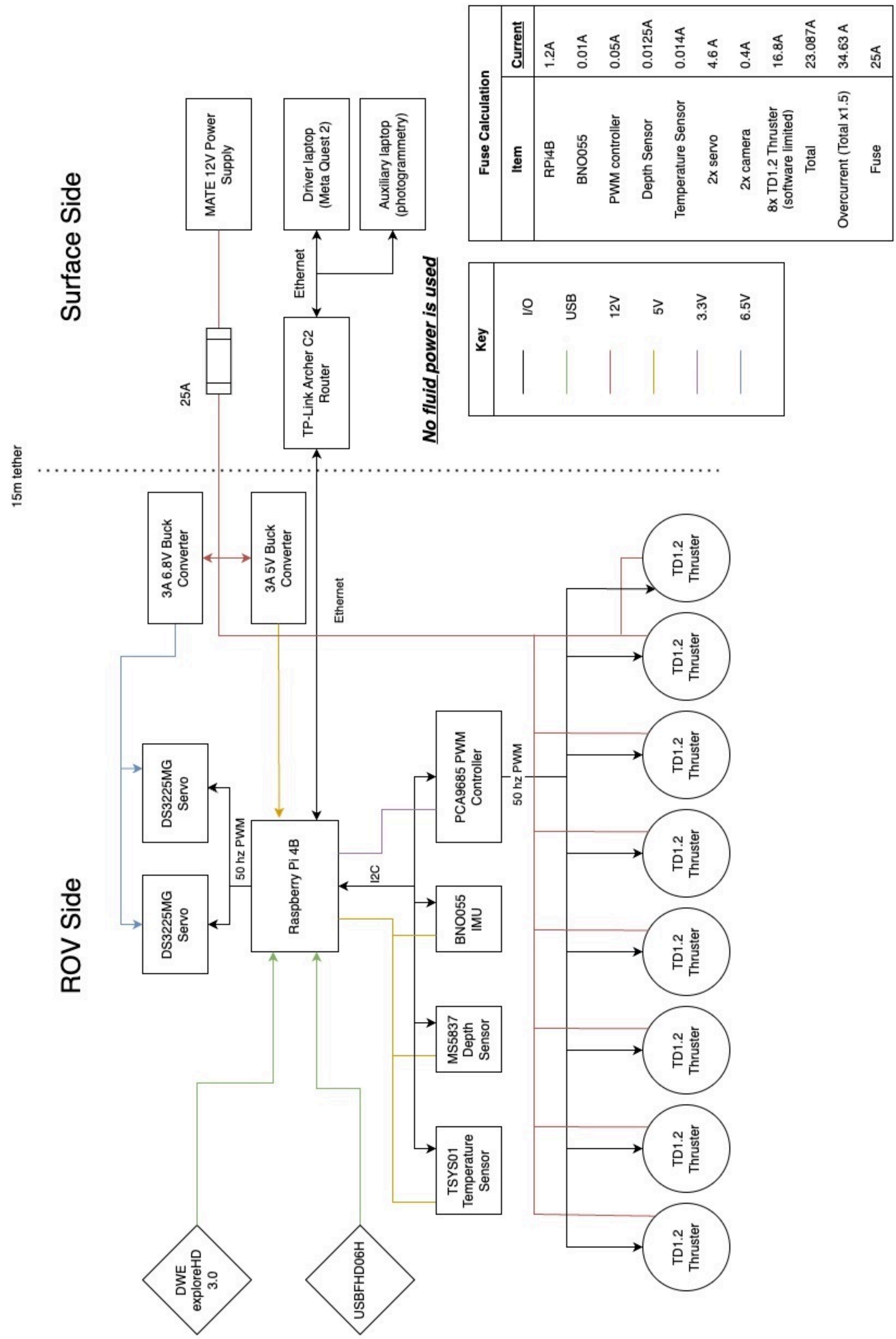
How to make a TRUE waterproof servo! (2019). Retrieved from https://www.youtube.com/watch?v=iSKlw3ZUEwU

Accounting

(All prices in USD)	Method	Quantity	Budgeted	Actual	Running
Mechanical					
Blue Robotics 4" enclosure (set)	Purchased	1	$1200	$402.00	**$402.00**
BlueRobotics WetLink penetrators	Purchased	9		$90.00	**$492.00**
15m tether (3x power, 1x ethernet)	Re-used	1		$ -	**$492.00**
Diamond Dynamic TD1.2 thrusters	Purchased	8		$526.08	**$1,018.08**
Electronics					
Raspberry Pi 4 8GB	Purchased	1	$1300	$123.00	**$1,141.08**
Adafruit 8-Channel PWM	Purchased	1		$9.95	**$1,151.03**
BNO055	Re-used	1		$ -	**$1,151.03**
High-Resolution Depth/Pressure Sensor	Purchased	1		$49.00	**$1,200.03**
Celsius Temperature Sensor	Purchased	1		$70	**$1,270.03**
DSSERVO DS3225 Servo	Purchased	2		$33.59	**$1,303.62**
Xbox One Controller	Re-used	1		$ -	**$1,303.62**
DWE exploreHD 3.0	Purchased	1		$300	**$1,603.62**
Meta Quest 2	Re-used	1		$ -	**$1,603.62**
Other					
Travel expenses (individual)		14	$15,000	$15,456	**$16,759.62**
Total Expenses					**$16,759.62**
Income					
Singapore American School			$2,500	$1,603.62	**$15,456**
Personal funding			$15,000	$15,456	**$ -**
Total					**$ 0**

Appendix

Appendix 1 - SID



Fuse Calculation

Item	Current
RPI4B	1.2A
BNO055	0.01A
PWM controller	0.05A
Depth Sensor	0.0125A
Temperature Sensor	0.014A
2x servo	4.6 A
2x camera	0.4A
8x TD1.2 Thruster (software limited)	16.8A
Total	23.087A
Overcurrent (Total x1.5)	34.63 A
Fuse	25A