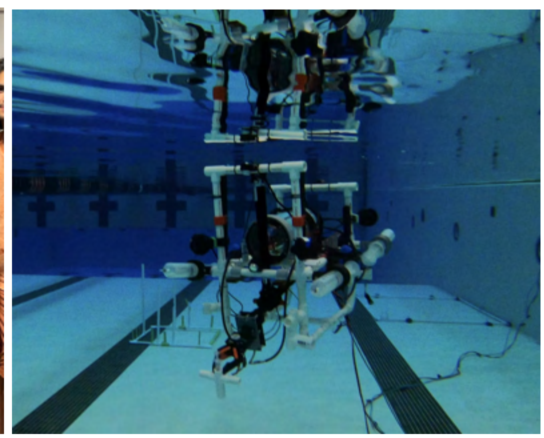
# NUWAVE

Northeastern University
Boston, MA,
2023 MATE ROV Competition



Jonah Jaffe: CEO and Co-Founder
Adri Lanza: Mech Lead, Co-Founder
Thomas Davies: Mechanical Lead
Mel St. Cyr: Electrical Lead
Joshua Cheng: Software Lead
Aditi Purandare: CSO
Lily Miles: Electrical
Bavesh Matapathi: Electrical/Software
Tom Consi: Mentor

Dylan Wolter: Mechanical
Noah Mitchell: Mechanical
Bryan Pitts: Electrical/Software
Charlotte Stieve: Software
Nikolas Sanderson: Electrical
Samira Hajj: Mechanical
Josie Kummer: Software
Mia Yim: Software
Jack Yu: Software

**Abstract** ..... **2**
**Design Rationale** ..... **3**
**Systems and Subsystems** ..... **4**
Electronics Housing ..... 4
Overall Part Selection Metrics ..... 4
Power Management System ..... 4
Control System Flow ..... 5
Thrusters ..... 7
Electrical System Integration ..... 8
Waterproofing Mechanisms ..... 9
Arm ..... 10
Frame ..... 13
Buoyancy ..... 15
Control system ..... 16
Coral Head Modeling ..... 17
**Safety** ..... **19**
Safety Procedures ..... 19
Safety Checklist ..... 20
**Challenges** ..... **20**
**Future Improvements** ..... **21**
**Reflections** ..... **22**
**Budget and Cost Accounting** ..... **23**
**Schedule** ..... **23**
**Acknowledgements** ..... **24**

# Abstract

NU Wave has been created over the past two years to design and construct underwater remotely operated vehicles (ROVs) to provide solutions for numerous complex underwater operations. As a newly developed company, we have been able to build from the ground up around our two pillars of ROV Design: extensibility and precision. Our ROV, Mariana, has the requisite functionality to install floating solar panels, remove biofouling, model coral heads and much more.

Mariana is the product of almost 2 years of research, development, and fabrication. It is the result of our iterative design process. It has a lightweight, modular PVC frame, adjustable buoyancy, an arm with 4 points of actuation, a watertight electronics enclosure, and a high precision software system. This technical document

describes the design and construction of our ROV and how Mariana addresses the numerous climate challenges outlined in the United Nations: Decade in Climate.

NU Wave is a quickly growing company with 17 consistent members who have years of collective engineering and software industry experience.

## Design Rationale

At NU Wave, we begin designing by identifying the set of problems we would like to be able to solve. Our goal is to design solutions that can be applied to multiple challenges at once. This philosophy allows us to spend a lot of time optimizing a few designs rather than working on many smaller systems to deal with our large set of problems. We also believe in designing with the ability to both change and add to our design rapidly. We tend to think of it similarly to a large software project, if the person setting up the architecture hard codes every step, it becomes difficult for future developers to make changes and update the system.

That leads to the second pillar of our design rationale, a strong belief in the iterative design process. We believe that by building on previous designs, we learn something new at every step and can converge on a close to optimal solution for our problems. In practice, that looks like us going through 3 versions of the arm design, 3 versions of the frame, and moving through 3 control system variations before reaching our final product. Each variant improved on the flaws of a previous design, leading to a much more effective ROV in the process.

Lastly, we have a policy of build first, buy second. More specifically, in any situation where we have an idea we want to make, we try to build it for as cheap as possible first, and then if that initial build has some material flaws, we spend money to account for them. This can be seen with both our arm and our cameras where we tried to build them for around $30 each, but then moved on to better solutions that other companies had developed to ensure reliability. This concept was initially born out of necessity, as for the first year of this project, we had almost no funding. As such, buying anything expensive was a last resort. While we have since done significant fundraising through primarily grant-writing, we believe that the build first policy is both fiscally responsible and leads to a scrappier, more resilient team culture.

## Systems and Subsystems

## Electronics Housing

The electronics housing used on our robot is an off the shelf system from Blue Robotics. We went with a 6 inch cylinder positioned in parallel with the forward range of motion to have improved hydrodynamics and enough space to house all of our electronic components.

## Overall Part Selection Metrics

When choosing electronic components, there were many factors to consider which models would work best.

A large consideration was footprint within our electronics cylinder: a smaller footprint is ideal and could even lead to decreasing our cylinder size in the future. Thus, smaller electronics were chosen when possible. Alternatively, we also chose some fully waterproof parts so that these parts could be mounted onto the robot chassis and kept outside of the tube. This is especially important for parts that are larger in size or give off more heat, in order to decrease both the physical size of electronics within the enclosure and heating as well.

Another consideration was ease of integration. As our system developed, parts that operated within our already existing power rails were ideal, as well as parts that were easily integrated into our control system.

Ease of assembly was also considered with this system design, choosing electronic solutions that required minimal soldering and other time consuming electronics assembly tasks were ideal for time management. Additionally, electronics parts that could be easily attached and detached from our system during early design phases were key to making iterating more efficient.

## Power Management System

A 48V 1500W power supply from Mean Well is used to supply power to the ROV, and managed by an emergency switch, as well as a 30A fuse. The power was converted to 12V and 24V by two waterproof buck converters, each rated for 30A, attached to the frame of the robot. Another buck converter was kept topside and used to step down the 48V to 9V, which was then sent to the router. This was an adjustable voltage converter with part number SYRE-BUCK-6T60.

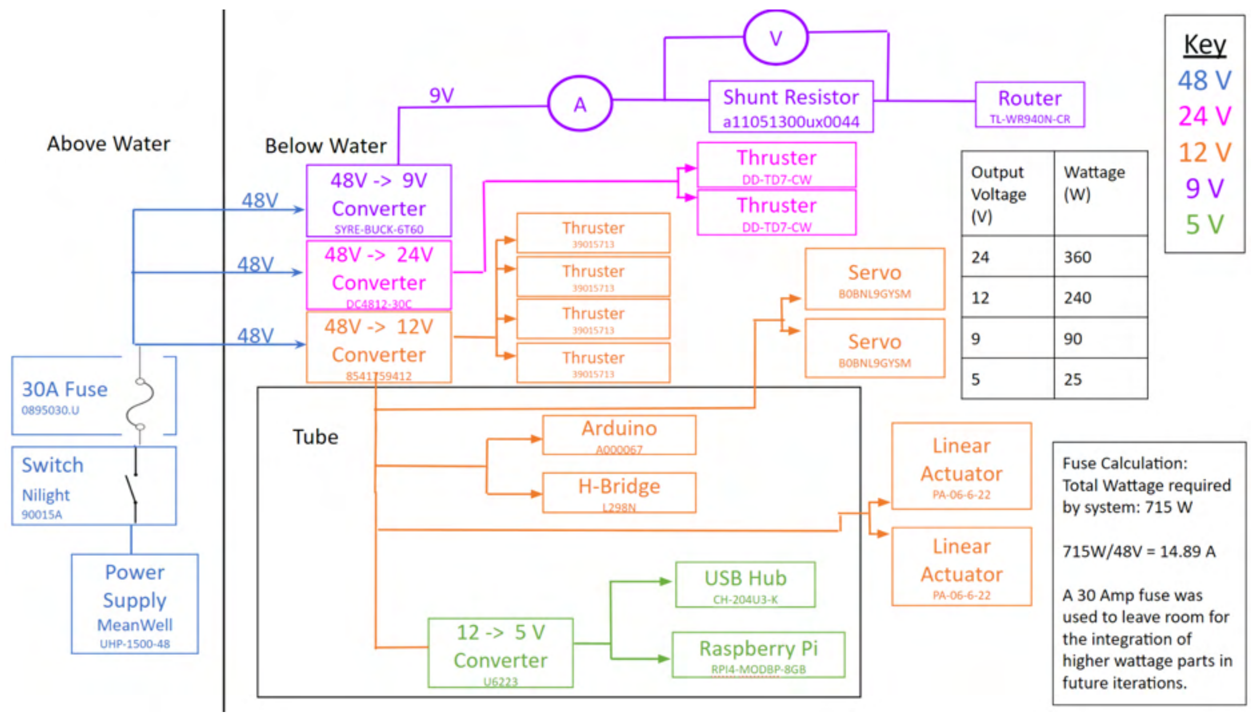



Figure 1: Power Diagram SID

The 12V line is used to supply four thrusters used for forward and backward motion as well as yaw rotation. It also powers two servos used for the arm's wrist and claw control, two linear actuators for extension and height manipulation of the arm, an Arduino Mega, and the H-Bridge motor driver. The 24V line is used to supply power to two larger thrusters for upward and downward motion. The decision to use larger and more powerful thrusters for this motion was largely fueled by the increased pressure when at increased depths and the likely frequency of traveling from the top to bottom of the pool.

The 12V line was then stepped down to 5V with a U6223 buck converter. 5V were used to power the cameras via the USB Hub, as well as the Raspberry Pi.

## Control System Flow

To control all of the electronics, a Raspberry Pi 4 was set up within the electronics tube and connected over serial to an Arduino Mega. We then used a library called PyFirmata to allow us to write Python code to control our Arduino. The Arduino Mega is a hub for many of the control signals of our robot, including signals that ultimately control our six thrusters, two servos, and H-Bridge motor controller that determines the orientation of the power and ground connections, in turn deciding the motion of the two linear actuators. The Raspberry Pi is also connected via a USB Hub to two cameras. An Ethernet cable connects the Raspberry Pi to our topside PC for control via an XBox controller. This ethernet connection from our Raspberry Pi to our router on topside is the only cable used to relay signal to the ROV. This decision was made as a means of making our tether as manageable as possible.

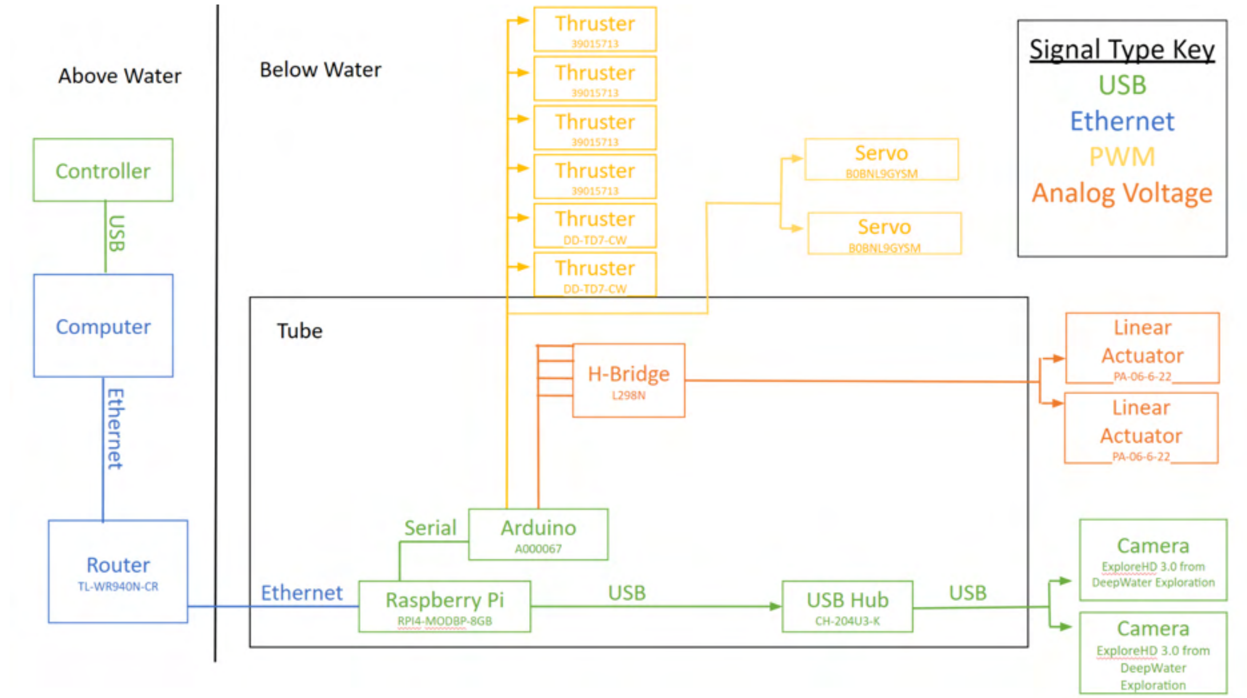



Figure 2: Signal Diagram SID

In order to communicate with the Raspberry Pi, an asynchronous server is run on the bottom side that receives messages sent down from the top side. The reason we chose to use an async server as opposed to a typical sockets server is because async allows us to schedule tasks that send out messages at a much higher frequency than can be communicated from the surface. This is useful for sending out square waves to stepper motors where the variation of the period of the wave determines the speed. This capability allowed the control of stepper motors that were employed throughout testing.

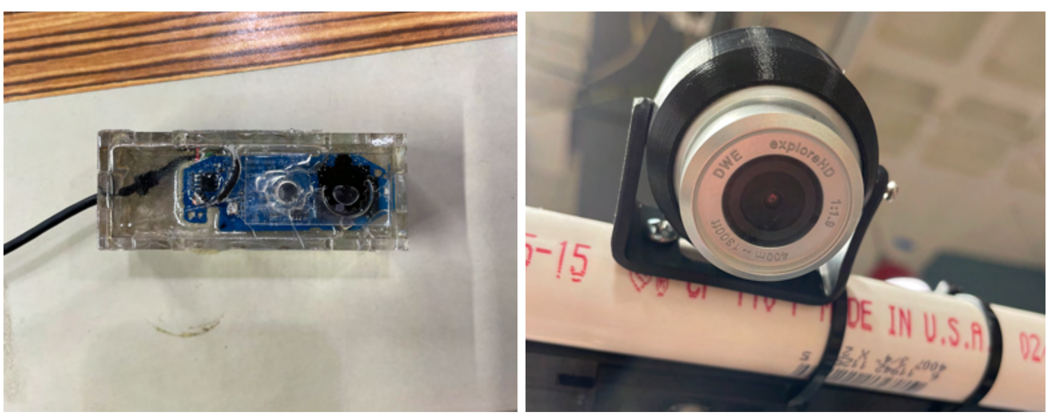
On the topside, input is received from two usb joystick controllers with a library called Pygame. One of the controllers is for the arm, while the other is for piloting, but the commands they send down are joined into one message. These control messages are only sent when the joystick values are changing in order to reduce network congestion. The control computer then connects to our router over ethernet for a fast connection and the router is connected to the Raspberry Pi to complete the control flow.

## Thrusters

The four thrusters for yaw and lateral movement are powered by 12V and the two larger thrusters for vertical movement are powered by 24V. Both types of thrusters were sourced from a company called Diamond Dynamics. These thrusters were chosen over the industry standard BlueRobotics thrusters for three main reasons. Firstly, they are cheaper. Secondly, they have built in waterproof ESCs. The potential of heat building up in the tube was a concern because of the inherent increase in pressure, and to avoid that, it was ideal to have ESCs and high voltage power conversion outside of our air-tight electronics cylinder. Lastly, the up and down thrusters from Diamond Dynamics have better specifications. Off of 20V, the BlueRobotics thrusters have 5.05kg of thrust while the Diamond Dynamics ones have 6.7kg. Additionally, 20V is not a common voltage to output, and so an additional converter would be necessary outside of the electronics cylinder, adding heat and clutter to the system.

## Cameras

The cameras relay signals through the USB hub to the Raspberry Pi. We chose a slim USB hub model to decrease the footprint within our electronics tube. Additionally, the USB hub allows us to take in and power more cameras than would be possible through only the Raspberry Pi due to its limited ports and power output constraints. Our first attempt at waterproof cameras was to laser cut a box out of acrylic and then put the insides of a logitech webcam inside of it. It worked, but the box was hard to mount and unreliable. We wanted a dependable solution because vision is highly important underwater. ExploreHD 3.0 cameras from DeepwaterExploration (a company started by a former MATE competitor) were chosen. The cameras are rated to 1,300 feet underwater and have preprocessing color correction for underwater environments onboard. They also come with a custom streaming system that helps reduce bandwidth issues.




Old Camera Fig 3: Cameras New Camera

# Electrical System Integration

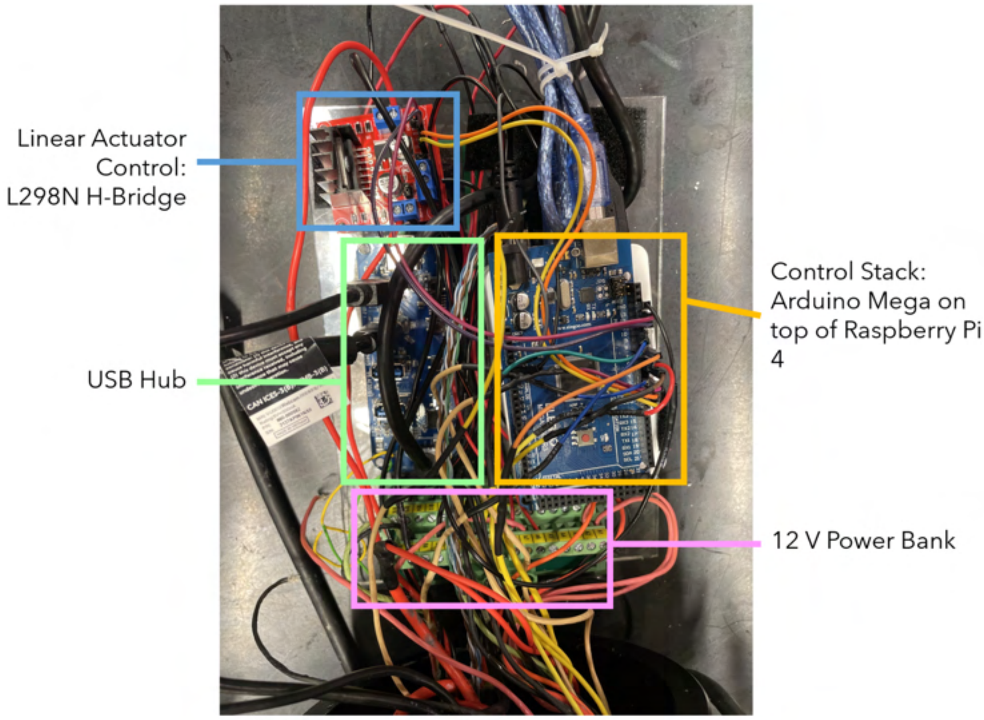



Figure 4: Top view electronics tube

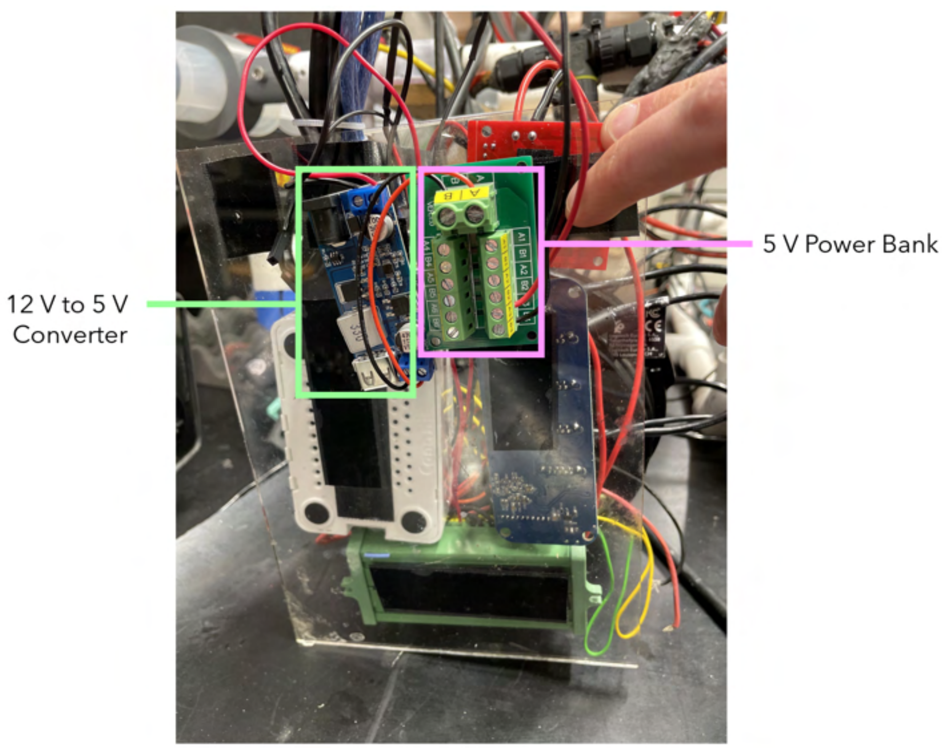



Figure 5: Bottom view electronics tube

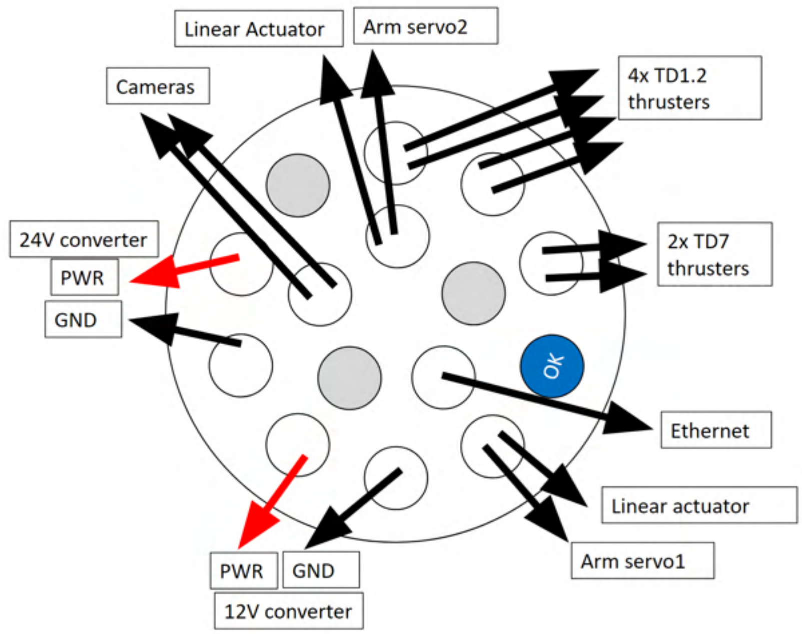


Figure 6: Electronics Tube Passthrough Map

## Waterproofing Mechanisms

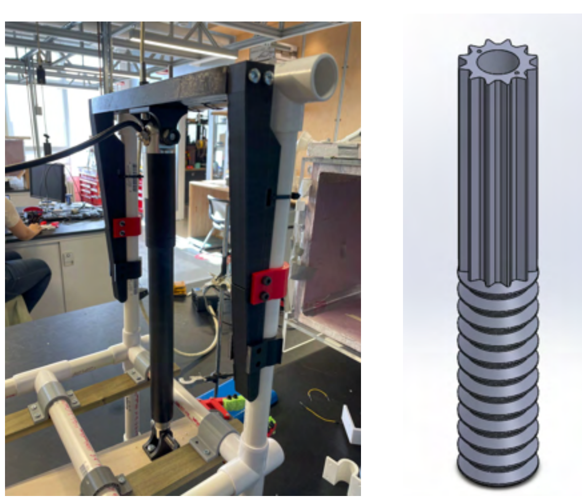
Our policy with waterproofing is to err on the side of caution. All of our waterproof wire connections use two layers of waterproofing to ensure no short circuits or water leaks are possible. We use watertight solder sleeves to make all connections and then cover them with industrial duct sealant to further ensure our connections are waterproof. We have two solder free connections for our tether on the ROV that are both IP68 rated and have additional industrial duct sealant on the entry points. They also help act as strain relief for the ROV. Our electronics tube has potted cable penetrators for all of our wires entering and exiting the tube. There are o-rings on every cable penetrator and 2 for the tube itself. Our in-water power converters are rated to be waterproof, but we added a second layer of protection using silicone sealant in all of the areas we deemed to be potential locations where water could enter the converter.

## Arm

In the initial design stages of the ROV's arm, the goal was to make it so the pilot would be able to plant the ROV on the bottom surface and still retain the freedom required to attempt tasks on the bottom of the pool. With this in mind, NU Wave designed the arm with the ability to extend, move up and down, rotate, and grip

Figure 7: Initial Design of Linear Actuator

objects of varying size. The up and down movement combined with the extension was critical so objects directly in front of the ROV could be picked up without the pilot needing to make small movements towards the object. The ROV has the ability to rely completely on the arm's movement instead of holding a position while suspended in the water. To achieve this, all of the parts were 3D printed using PETG to save money and reduce the weight of the final version of the ROV. The initial plan was to make a 3D printed linear actuator to mount a two way circular gear underneath the robot. In this application the linear actuator would move the arm up and down while the two way gear would allow for both extension and rotation of the claw. As time went on, NU Wave earned money from various grants which allowed us to pivot to a more reliable design which involved using a purchased linear actuator from Progressive Automations. Using this new part for the up and down motion of our arm, we managed to use the initial design of the mount for attaching it to the front of the frame and had the other end reach all the way to the lower half of the ROV. With the linear actuators movement,



. Figure 8: CAD of Cylindrical Two Way Gear the ROV uses hinges at its back to allow for a pivot point for the arm to lower and raise from.

The next task was designing a solution for the extension and wrist rotation. The initial solution as mentioned previously was to use a two way cylindrical gear. The set of teeth that run along the length would be paired with a motor which would rotate it and the set of teeth that run perpendicular to the length of the cylinder would be paired with a motor that would allow for the extension of our claw. The motors we decided would be waterproof NEMA 17 stepper motors from Automation Direct. The two way gear was planned to sit on linear bearings with a guide rail running down the middle to minimize the amount of friction the gear experienced. Once the design was printed and assembled, the imperfections in the printed parts proved to be difficult to work around. The overall length of the two way gear was the entire length of the ROV which came out to around 250 millimeters. This required us to split up the casing and the two way gear into multiple prints and then press fit them together with fasteners. Due to tolerance stacking between the printed parts, the rod when fit through the middle was not perfectly straight. This caused the two way gear to not extend or rotate as required. From here, we pivoted in designs to using a purchased linear actuator from Progressive Automations and using that with an in-house design for wrist rotation.

The wrist rotation gear box consisted of two 3D printed sheets separated by aluminum standoffs from McMaster-Carr. The side that attaches to the linear actuator used for extending has a cutout for a 9imod Brushless Servo Motor which gives a torque of 60 kg-cm. A gear was attached to this servo which mated with a thicker gear of the same exact size to give a 1 to 1 ratio. This thicker gear was attached to the back of the claw which fit through a flange bearing. The flange bearing was mounted on the opposite 3D printed sheet. This gear was screwed to the back of the claw to allow the thicker gear to stop any oscillations of the claw. The side that is attached to the linear actuator has a printed adapter which was designed to fit on the linear actuator.

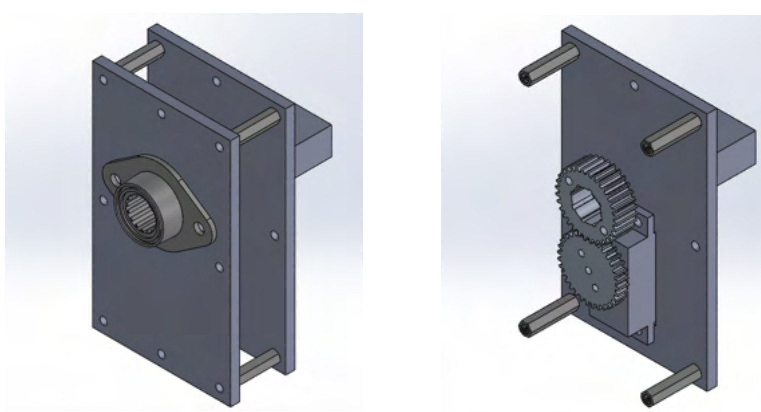


Figure 9: Wrist Rotation Gearbox

The final step in completing the entire arm assembly was the claw. The design was inspired by an online image of a mechanical claw. There were two arms stemming from the base which attached to the actual claw face. One of the arms is attached to a gear which allows both sides to move with each other. On the base of the claw, there was another mount made for a 9imod Brushless Servo motor which initially had a

Figure 10: First Three Iterations of the Claw

smaller gear to increase the torque output and the grip strength. From here, instead of having two arms to support the claw faces, two more were added on the other side along with a grippier material on the actual faces to increase the friction between the claw and any object it grabbed. From this design, a huge problem started occurring where the gear on the servo would skip teeth. To combat this, we went with a 1 to 1 gear ratio on the servo to reduce the amount of torque placed onto the claw and got new screws which would allow for a smooth rotation with no loosening or tightening. This was the first claw designed for the wrist rotation gearbox as well. The extrusion at the back is the part that fits through the flange bearing and attached to the gear. With

Figure 11: Final Iteration of the Claw

the new gear ratio, the gears continued to skip. This was found out to be for a few reasons. The first was that the base of the claw had a stress concentration point right in between the gear on the motor and the gears attached to the arms. Whenever a large amount of torque was applied, the whole base would bend slightly allowing the gears to slip. Another reason was how the gears were initially aligned. The one on the servo was slightly below the one on the claw. This played into the bend and caused the gears to skip easier. To combat these, the base of the claw was made stronger by experimenting with different infills and making it thicker. We changed the gears from regular spur gears to helical gears to have more contact area and deal with the higher load. We also changed the distance from the motor's shaft to the center of the gear attached to the claw.

The servos responsible for controlling the wrist and claw of the arm received pulse width modulation signals sent directly from the Arduino, while the linear actuators responsible for arm's extension and movement up and down received signals via the H-Bridge motor controller, which also received signals from the Arduino.

## Frame

During the initial stages of our design development, we encountered financial constraints and uncertainty regarding the specific requirements of our final design. Consequently, we recognized the need for an affordable and adaptable modular frame. After careful consideration, we opted for PVC Pipe as our frame material due to its cost-effectiveness and widespread availability, with 5 ft of ¾" pipe being obtainable for a few dollars at local hardware stores. PVC Pipe proved to be easily transportable and customizable, as it could be cut and fitted with inexpensive connectors of various types.The shape of the frame was inspired by a Blue Robotics robot, which one of our team members had encountered. We determined that an overall rectangular shape, shown in Figure (12), would be advantageous, providing ample space for our arm and all required electronics within the structure.

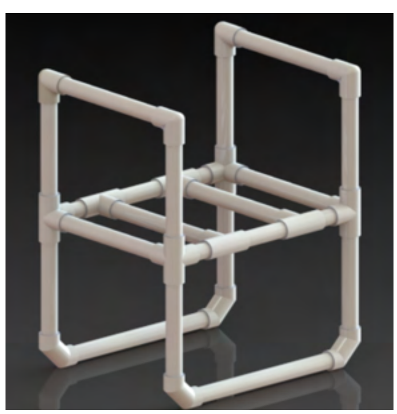


Figure 12: CAD of the Frame

For the mounting of the electronics tube, we opted for an 6"diameter tube placed at the center of the frame. In our initial design, we merely pulled the top of the tube down to the frame, but it had the potential to move forward, backward, and sideways. Subsequently, we improved the design by implementing a new iteration that held the tube firmly on all sides and clamped it to the frame. Rubber pieces were inserted between the tube and the 3D-printed mount to prevent undesired sliding. To attach the 3D-printed mount to the frame, we developed a specialized component, illustrated in Figure (13), that encompassed the PVC pipe and provided a flat surface for attaching any necessary equipment.

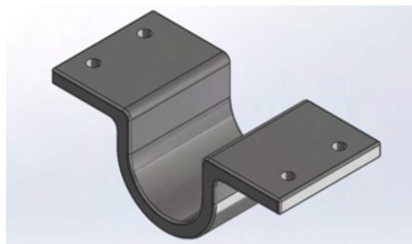


Figure 13: CAD of the PVC to Thruster Adapter

Regarding the attachment of the thrusters, we employed a similar approach as with the tube mounts. By resizing the aforementioned piece (see Figure (12)), we were able to secure the thrusters to the frame. To facilitate the attachment of the thrusters to the PVC mounts, we 3D-printed adapters that connected the thrusters to the PVC attachments. It's important to note that the adapter design differed for the side thrusters and the vertical thrusters, as depicted in Figure (14). The side thrusters were positioned at 45-degree angles, offering enhanced maneuverability when properly programmed, enabling straightforward movement in either a straight line or sideways. As for the vertical thrusters, they were positioned on the sides as close to the center of mass as possible.

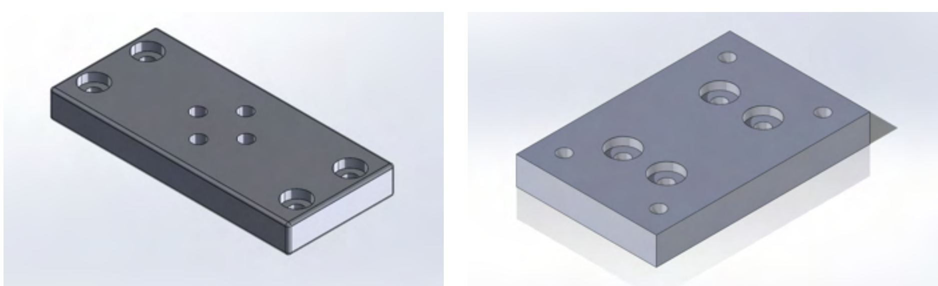


Figure 14: CAD of PVC to Thruster Adapter Mount

## Buoyancy

During initial testing of the ROV, it was believed that the ROV would be mostly neutrally buoyant based on the frame, electronics tube, and overall layout, and only minor changes would need to be employed in order to fix it. However, after the ROV's first wet test, it was noted that there were inconsistencies in the buoyancy, especially as it leaned to one side heavily. We first tried to rectify this with the use of pool noodles and fishing type weights. This worked short term but it was noted that it was hard to make small adjustments easily, and it was by no means a perfect fix. In order to combat

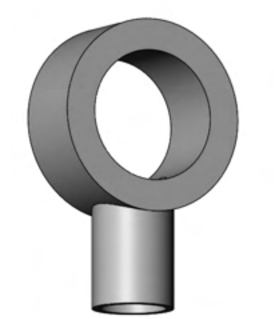


Figure 15: CAD of Syringe Mount

this issue, we looked for a highly adjustable system so that small adjustments could be made easily, and would be much more precise. Taking inspiration from a lego submersible video, we opted to use 350 ml syringes as a form of buoyancy. By filling the syringes with air and capping them before submersion, the syringes acted as adjustable bouyancy tanks that could be changed quickly and precisely. We then used four syringes, at each of the corners of the ROV, which we attached with a 3D printed coupling, which secured it in place. Then in order to adjust our buoyancy, if one of the the corners was less buoyant, we filled it with more air, this was especially useful with

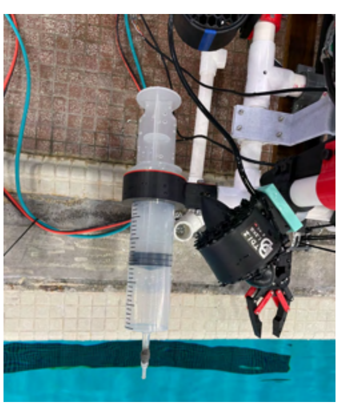


Figure 16: Syringe in Mount

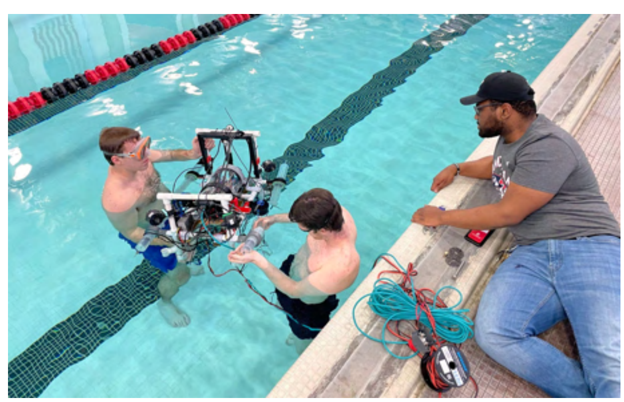


Figure 17: Buoyancy Testing

the syringe markings, as it allowed us to change the volume of air in uniform amounts. We spent one day at our university's pool with the ROV turned off in order to tune our buoyancy using this method, which proved successful. During this time, our other form of buoyancy control was tested, however it was determined to be obsolete. Initially, we believed we needed both control over being negatively and positively buoyant. However, we underestimated the weight of the linear actuators, and the ROV was negatively buoyant on its own, so only the syringes ended up being used. Despite this, our adjustment for being negatively buoyant, consisted of adjustable weights, similar to an adjustable dumbbell. A nut and bolt coupled with weights were used to make the bot negatively buoyant, and the number of weights on each nut and bolt could be changed if necessary.

| Overall Buoyancy of ROV | | | |
|---|---|---|---|
| Weight of ROV (kg) | Weight of Displaced Water from Electrical Tube (kg) | Weight of Displaced Water from Syringes when Full of Air (kg) | Weight of ROV in Water (kg) |
| 15.25 | 9.884 | 1.4 | 3.966 |

Figure 18: Buoyancy Table

## Control system

The software for the control system was implemented in Python using the pygame package. We initially defined the control scheme for the two controllers, focusing on ease of drivability. This framework is shown in the figures below.

To develop the corresponding software, we used the pygame Python library. This package connects to the controller and uses an infinite loop to detect any interaction with the controller and updates the corresponding predefined dictionary.

For example, if an axis is pressed in order to move the thrusters, the dictionary for that axis is updated. The numerical value for movement is then calculated by axis control magnitude, necessary constants, and precision. Each of the controls are categorized as buttons or toggles. Toggles act as binary variables for situations such as precision. The precision variable controls whether the robot speed is multiplied by one of two predefined values: a lower (0.2) or relatively greater factor (1). The lower precision value allows us to move at lower speeds and complete tasks with a higher accuracy, while the higher factor supports speed for movements over farther distances.

Once the new values for the thrusters are calculated, they are combined in a string with a {pin number}:{value} format for each pin/thruster combination.


Figure 19: Arm Control Scheme


Figure 20: ROV Control Scheme

## Coral Head Modeling

Regard3d+algorithm: The photogrammetry software we chose to employ is Regard3d [3]. After uploading the images, the first step is to identify key points. It was experimentally determined that, for our data, a larger number of keypoints yielded a better model. As a result, we use a lower keypoint sensitivity. The program uses the AKAZE image registration algorithm. We also identified that Classic AKAZE enabled a better model than Fast AKAZE. After the keypoints are identified, they need to be matched to create correspondences. Among the algorithms available for matching, FLANN appeared to be the most promising. After matching, we use the default parameters for triangulation, densification, and surface generation. We tested 3d modeling of a roll of paper towels with a bowl on top. The original setup is shown in Figure 14 to the left, while the outcome is depicted on the right.

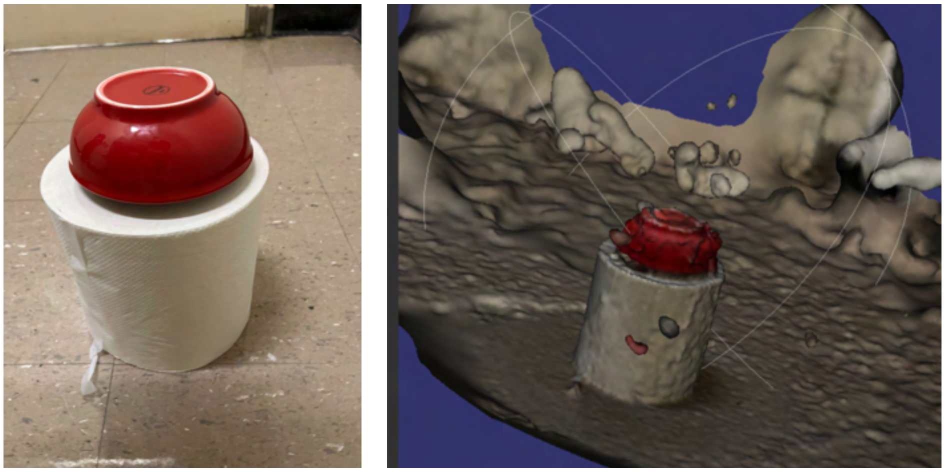


Figure 20: Original setup and 3D model

Literature review: In addition to Regard3d, we considered the implementation of previously successful research approaches. These included Papadaki et al. (2020), who emphasize a Random Forest binary classifier for keypoint identification. Prior to matching, eight parameters of a keypoint are used to train the RF. The classification determines if the keypoint is matchable or not. The process of eliminating unmatchable keypoints reduces unnecessary computations[1]. We chose not to implement this due to concerns regarding time. Additionally, it requires prior interaction with the image data. For example, the green channel parameter in the paper reflected the authors' knowledge of vegetation in the 3d reconstruction of the images. Another study was by Sharp and Ovsjanikov, who introduced PointTriNet, which uses networks for probabilistic triangulation. The classifier network computes a score for a given triangle. This is achieved by selecting triangles and points near the triangle in question, which are then encoded, and fed into a PointNet neural network. The triangles selected for this classifier network are determined by a proposal network, which inputs points, encodes them, and returns triangle neighbor predictions after another PointNet. [2] An advantage of this approach was that it required no prior labeled data. However, this was not implemented due to concerns about computing cost and time limits for the competition.

Blendr: Following the generation of a model for Regard3d, we can export a surface. In order to scale the model, we developed a plan to utilize a ruler for scale. Once the ruler is inside the model, we will use Blendr's internal ruler tool to determine if the physical ruler is the correct size. While it is not the expected size, we plan on adjusting the model scale accordingly.

Data pipeline: The model is planned to be developed from a machine separate from the camera feed. As a result, a script to scrape the Flask webpage containing the camera inputs is being developed. The videos will be sampled, and the saved frames will be used as inputs for Regard3d.

# Safety

## Safety Procedures

In order to ensure the safety of all team members working around the bot, we have made sure to include many safety features on the ROV. All of our thrusters are shrouded such that there is no way for anyone to stick their finger into the propeller. When any hot gluing, soldering, or power tool use occurs, goggles must be worn. People not involved in any activities must be at least 10 feet away and fire extinguishers must be present. Whenever the robot is being turned on we ensure that all present are aware that it is being turned on and ensure that the circuit is set up correctly. Part of the circuit being set up correctly involves checking the fuse to make sure the electronics will be protected in the event of a short-circuit.

The ROV itself has a number of safety features to keep both team members and others safe while the ROV is live. First is waterproofing. As mentioned in our waterproofing section, for all of our wire connections we have two layers of waterproofing, and we added extra protection to many of our waterproof parts to truly ensure safety with added emphasis around higher voltage wires. We have also installed a 30 amp fuze within 30 cm of our anderson powerpole connectors to ensure safety in the event that something does go wrong.

We go through a lot of iterations as a team, and in any situation where we are testing a new part we will test our robot in stages. First, we briefly test functionality dry to ensure every electrical component is functional. Then we put our ROV in the water with no power to check if it is watertight. We then take it out again, inspect the electronics tube and any new connections for signs of water inside. Once that is clear we power on the robot dry again and prove our electrical functionality. Last, we put our ROV in the water with power and proceed with our in pool tests.

## Safety Checklist

1. Ensure that there are no sharp objects on or near the ROV.
2. Make sure that there are no conductible, semi-conductible objects or fluids in, on, or near the ROV.
3. Make sure all wires are properly secured, insulated, and waterproof.
4. Check that all motor connections, coverings, and mounts are properly secured.

5. Check the tether to ensure that there is not any missing insulation or waterproofing and that the tether is not knotted.
6. Make sure the robotic arm is safely attached to the robot with no freedom to wiggle.
7. Ensure that all parts of the frame on the ROV are properly secure and will not disassemble from the ROV.
8. Be sure all personnel operating the vehicle are attentive and prepared to safely operate the ROV.
9. Check the fuses to make sure they are not blown.
10. Be sure that all pressure vessels on the ROV are sealed.
11. Make sure all personnel and onlookers are clear before operating the ROV.
12. Ensure that the electronics are on and running.
13. Check to make sure everyone is clear of the tether as the ROV is put in the water.

# Challenges

As a new MATE team, we only have two members who have ever been to a MATE regional competition and none who have competed at the international level. As such, we have had to spend a lot of time onboarding members. Additionally, a lot of people's initial instincts about how to design and fabricate things don't translate to underwater robots, so people have to learn through doing. Those with underwater experience have taken a very hands off approach to teaching newer members about the uniqueness of marine challenges with the one exception to that being safety. While out the gate, not giving people the answer we know will work has slowed us down in building a successful robot, to us, the lessons learned are much more important. We are already starting to see this approach pay dividends, as over time, members have grown to understand our design constraints on a much higher level and can independently design successful subsystems. This strategy also has the unintended benefit of members coming up with great ideas that those of us with underwater experience would never think of. While our lack of experience has oftentimes made progress slow, the learning that is happening should set the team up to be successful for years to come.

We also had some struggles with testing. Up until less than a month before our qualification video was due, we lacked the ability to do any testing at the Northeastern pool. We were lucky enough that Dr. Andrew Bennet and the people at MIT Sea Grant allowed us to use their testing pool that was about 4 feet deep. That pool entirely sustained our team's testing needs for most of NU Wave's existence and we are incredibly grateful to the people at MIT Sea Grant for their help. That being said, it wasn't an entirely optimal testing environment. It was a few miles away, so we had to drive our ROV to the test tank and find transportation for our members. Additionally, we couldn't bring all of our tools, so sometimes when something broke, we were stuck and couldn't make progress.

# Future Improvements

Despite the syringes working well, we are looking towards other ways of controlling our buoyancy. One idea we may pursue is automatic buoyancy control, executed in a way similar to that of a BCD. By integrating sensors and using a pressurized tank, it could be possible for the ROV to automatically adjust its buoyancy by inflating or deflating bladders of air located on the ROV. This could ensure that the ROV stays level and would automatically adjust, without having the need to spend time manually changing the buoyancy.

Another improvement is tether management. The team is looking to better the tether on the robot, so that it automatically has the right amount of slack. In order to combat this issue, the team is looking towards an automatically retracting tether system, similar to that of a tape measure. By having the tether constantly taught, but able to be changed easily, it would remove the stress on the team member poolside responsible for this. At the same time, it would be out of the way and allow the ROV to move freely without worry of being caught or tangled.

One of the other problems we ran out of time trying to solve for this year's competition was trying to reduce the thermal buildup within the electronics tube. The acrylic material that was used for our enclosure is poor at conducting heat. Even though our ROV is fully submerged, we have noticed in some instances, a large amount of pressure buildup inside the tube to push out the end caps slightly. To combat this, our plan was to create an aluminum mounting plate that runs through the tube and is open to the water in the pool. The electronics would be in direct contact with a material that could transmit their heat and transport into the surrounding water. This was the first idea we had and will hopefully get the chance to refine it. Our goal is to bring a working version of our thermal management system to next year's competition.

Lastly, for most of the year we have been working on automatic depth stabilization, and are having some issues integrating our depth sensor into our system. That being said, once we overcome that hurdle, it would be incredible to be able to pick a depth that we are at, and implement a PID loop where our thrusters are automatically controlled to keep us at that depth. That would let pilots get down to where they want to be, and then only have to think about solving the problem in two dimensions instead of three.

# Reflections

Bryan Pitts - Originally, I joined Northeastern Robotics through the free “Introduction to Robotics” class that the club had offered. Towards the end of the class, MATE had come in to present their project and my interest in their project and the competition had piqued my interest. After joining, I was ecstatic to see that the Robotics concepts and skills that I learned while in the Introductory course were applicable to the MATE team. Now, as a member of the Electrical Engineering team I look forward to competing this June and the many other things my teammates will teach me!

Mel St.Cyr - I have been a part of Northeastern University’s Underwater Robotics team for about two years. It has been amazing to see how the size and dedication of the team has grown alongside the steady improvements to our robot. As the lead electrical contributor to this robot, I have learned how to make independent design decisions and how to delegate and work with others interested in the electronics system. I am grateful for the community this team has brought to my undergraduate experience and hope to hear them excel in the future.

Jonah Jaffe - This has been a dream for me for a long time, and after two long years of work, seeing it finally become a reality has been incredible. I’ve really focused on education as a leader, and seeing the growth in the members of this team has been a great source of pride for me. That being said, this has certainly been one of the hardest things I’ve done in life, and there have been many moments where this has felt challenging, but seeing what we’ve been able to accomplish makes it all worth it, and I feel like our future is bright.

Thomas Davies - Finding the MATE team at Northeastern as an intro to the combination of marine science and robotics was incredibly lucky. I have learned so much from troubleshooting problems with a small team tl leading a subgroup effectively to meet deadlines. This will be my second year working with the Northeastern’s MATE team and I have loved every second of it. After working for this competition, I know that I want to pursue a career in the underwater robotics industry.

Lily Miles - I was drawn to the field of underwater robotics by the unique challenges it presents, and the innovative solutions required for a robot to exist and thrive in an underwater environment. My electrical engineering skills have improved exponentially since joining MATE, and I’m so lucky I get to work with such amazing engineers and students. I look forward to learning from this year's experience in competition, and improving for next year.

# Budget and Cost Accounting

Our funding comes primarily from grants that we've written as a team. We have had 5 individual team members bring in over $15,000 from grants, and as such, we put a heavy emphasis on learning how to write grants. We have also received support in the form of parts from Automation Direct, Igus and Bulgin. Given our build first, buy second policy, we spend a lot of time on research and development before we make final purchases for the vehicle. Given how much we learn from each test, we have reallocated funding to areas that need it throughout the season, that being said, we try to give each subteam a sizable cushion of funding to work with at the beginning of the season, and reevaluate when larger items need to be purchases. Note that for budgeting for our ROV and for our travel, we budgeted for the entire thing and not subsections of it.

| Income | Budget | Type |
|---|---|---|
| NURobotics Club Funding | $2,450.00 | Income |
| PEAK Grants | $6,750.00 | Income |
| MIT STREAM Grant | $4,995.94 | Income |
| NHRL Travel Grant | $5,000.00 | Income |
| Khoury Student Experience Fund | $3,800.00 | Income |
| **Total:** | **$22,995.94** | |

| Parts Donations | Appx. Value | Type |
|---|---|---|
| Automation Direct | $1,000.00 | Donation |
| Igus | $500.00 | Donation |
| Bulgin | $500.00 | Donation |
| **Total:** | **$2,000.00** | |

| R & D Costs | Total Budgeted | Total Spent | Type | Category |
|---|---|---|---|---|
| Additional Thrusters | $1,000.00 | $338.00 | Purchased | Hardware |
| Education Fleet | $800.00 | $799.96 | Purchased | Hardware |
| Mechanical | $1,500.00 | $619.87 | Purchased | Hardware |
| Electrical | $1,500.00 | $1,457.22 | Purchased | Electronics |
| Software | $500.00 | $210.99 | Purchased | Sensors |
| Testing and Props | $560.00 | $217.00 | Purchased | Hardware |
| **Total:** | **$5,860.00** | **$3,643.04** | | |

| Production Costs | Total Spent | Type | Category | |
|---|---|---|---|---|
| Thrusters | $634.00 | Purchased | Hardware | |
| Linear Actuators | $520.00 | Purchased | Hardware | |
| Linear Actuator Mounts | $92.97 | Purchased | Hardware | |
| Electronics Tube | $249.00 | Purchased | Hardware | |
| Passthroughs | $90.00 | Purchased | Hardware | |
| Cameras | $558.00 | Purchased | Sensors | |
| Raspberry Pi | $169.99 | Purchased | Electronics | |
| Other in Tube Electronics | $147.53 | Purchased | Electronics | |
| Servos | $131.94 | Purchased | Hardware | |
| Printer Filament (PETG) | $100.00 | Purchased | Consumables | |
| Waterproof Electronics | $72.98 | Purchased | Electronics | |
| Topside Control Station | $700.00 | Purchased | Electronics/Hardware | |
| Tether | Wire Donated | Donation | Electronics | |
| **Total Spent:** | **$3,466.41** | **Total Budget:** | | **$5,000** |

| Travel Costs | Total Spent | Type | Category | |
|---|---|---|---|---|
| Flights | $4,104.15 | Purchased | Travel | |
| Hotels | $1,548.00 | Purchased | Travel | |
| Other Budget Items | $500.00 | Purchased | Travel | |
| **Total:** | **$6,152.15** | **Total Budget:** | | **$8,000** |

| Overall | Amount | Type |
|---|---|---|
| Total Budget | $22,995.94 | Income |
| Total Spent | $13,261.60 | Purchased |
| Total Remaining | $9,734.34 | |

Figure 21: Budget Sheet

# Schedule

Once the tasks were posted online for the 2023 competition, our CEO created an initial schedule, as depicted in Figure (#). This schedule was just a rough draft, and as the teams started working, the CEO made updates to make it more realistic. The team leads collaborated with the CEO to do their best in following the schedule, aiming to have the bot ready for video presentation by May and competition-ready by June.

| Mission Tasks | Subtasks (Purple - Mechanical, Orange - Electrical) | Points (Green - Easy, Yellow - Medium, Red - Hard) | Needed When | Notes |
|---|---|---|---|---|
| 1.1 - Install a floating solar panel array | position a solar panel array amongst floating wind turbines | 10 | March | Needed for video submission, high priority |
| | moor the panel array to three anchor | 5-15 | April | Needed for video submission, high priority |
| | remove the power port cover | 5 | March | Requires gripper with wrist rotation, Lucas Screw is highest priority for m |
| | install the power line connector into the power port | 10 | March | Easy mechanical goal, should be done quickl |
| 1.2 - Remove biofouling from the foundation and mooring lines of floating wind turbines | remove biofouling from turbines | 5-15 | March | Requires control of thrusters, need to stay at specific height |
| 1.3 - Piloting into "resident ROV" docking station | docking autonomously | 15 | May | Reach goal for software team |
| | docking manually | 10 | April | Thruster control needed, thrusters need to be attached to bot by March t software time to get controls ready |
| 2.1 - Create a 3D model of a diseased coral head | autonomously | 0-30 | April | Initial goal of the software team, needs to be done by April or we will abo |
| | manually | 0-20 | May | Secondary goal if autonomous control fails |
| 2.2 - Identify reef organisms using eDNA | collect a water sample from above the coral head | 10 | Febuary | Easy goal to attain to, mechanical should have this done quickly |
| | use the eDNA data to identify coral reef fish species | 5 | April | Reach goal for software. Not needed immediately |
| 2.3 - Administer Rx to diseased corals | position the simulated UV light source over the diseased area of coral | 5 | April | Neccesary goal, software should start immediately |
| | irradiate the diseased area of coral with simulated UV light | 5 | April | Neccesary goal, software should start immediately |
| | place a tent over the diseased area of coral | 10 | March | Needed for video submission |
| | insert a syringe filled with "probiotic" fluid into the port | 5 | March | Needed for video submission |
| | inject a "probiotic" fluid into the tent | 5 | May | Reach goal for mechanical, requires reliable gripper |

# Acknowledgements

Dr. Tom Consi (Mentor/Faculty Advisor): For meeting with our CEO every week and helping to guide the team towards success while letting us learn from our own mistakes.

Dr. Andrew Bennett: For allowing us to test at MIT Sea Grant and for helping us out when we needed supplies for immediate repairs.

Paul McGuinness: For inspiring the passion for underwater robotics that made this team possible.

Lauren Colby: For helping us finally get access to the Northeastern pool and for being very accommodating.

Hanumant Singh: For lending us the cases needed to transport our ROV.

Norwalk Havoc Robot League NHRL: For supporting us with travel funding that we desperately needed.

Automation Direct (Chip McDaniel): For the donations of a range of items from wires, to sensors to stepper motors, Automation Direct were the first to help and Chip made everything easy.

Igus: For their donation of wire.

Bulgin (Carlos Rodriguez): For their donation of connectors and Carlos for being incredibly helpful with the whole sample process.

NURobotics: For helping grow this team from two people into the project it is today.

Jill Zande: For her incredible and rapid support whenever we had questions or hurdles to overcome.

Matt Gardener: For his answers to our technical questions about the competition and his understanding of our struggles.

Tawni Gott: For helping us figure out hotels when we were in a bit of a bind.

Northeastern Undergraduate Research and Fellowship: For supporting our underwater research endeavors through their PEAK awards.

MIT Sea Grant:For giving us the space to test and for the MIT STREAM Grant.

MATE: For hosting the competition and allowing us to compete.

# References

[1] A. I. Papadaki and R. Hänsch, "Match or No Match: Keypoint Filtering based on Matching Probability," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 4371-4378, doi: 10.1109/CVPRW50498.2020.00515.

[2] Sharp, N., Ovsjanikov, M. (2020). PointTriNet: Learned Triangulation of 3D Point Sets. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science(), vol 12368. Springer, Cham. https://doi.org/10.1007/978-3-030-58592-1_45

[3] https://www.regard3d.org/index.php

[4] Ștefan, Amado-George, et al. “Theoretical and Experimental Aspects Regarding the Forced Mounting of a Cylinder Containing the Electronics of a Mini Submarine.” *MDPI*, Multidisciplinary Digital Publishing Institute, 8 Aug. 2021, https://www.mdpi.com/2077-1312/9/8/855/htm.

[5] N. Harsamizadeh Tehrani, M. Heidari, Y. Zakeri and J. Ghaisari, "Development, depth control and stability analysis of an underwater Remotely Operated Vehicle (ROV)," IEEE ICCA 2010, 2010, pp. 814-819, doi: 10.1109/ICCA.2010.5524051.

[6] N. D. Jayasundere and S. H. K. K. Gunawickrama, "Underwater ROV with Fuzzy Logic Motion Control," 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), 2016, pp. 1-6, doi: 10.1109/ICIAFS.2016.7946564.

[7] M. Cook, G. Cook, E. Celkis and T. Crandle, "ROVBus: A modular electronics backbone for flexible ROV capability integration," OCEANS 2016 MTS/IEEE Monterey, 2016, pp. 1-4, doi: 10.1109/OCEANS.2016.7761165.

[8] Barbieri, Loris, et al. “Design, Prototyping and Testing of a Modular Small-Sized Underwater Robotic Arm Controlled through a Master-Slave Approach.” *Ocean Engineering*, Pergamon, 16 Apr. 2018, https://www.sciencedirect.com/science/article/pii/S0029801818304645.

[9] Barbosa, Gustavo Franco, et al. “Digitalization of a Standard Robot Arm toward 4th Industrial Revolution - The International Journal of Advanced Manufacturing Technology.” *SpringerLink*, Springer London, 7 Nov. 2019, https://link.springer.com/article/10.1007/s00170-019-04523-2.