Cornell Cup Robotics

Xplorer XRP

Educational Submersible Robot with Data Collection

Final Documentation

Fall 2025

Table of Contents

Xplorer XRP Overview

Background

The Xplorer that started with a previously developed project out of FSU.

Purpose

The purpose of the Xplorer is…

Semester Objectives

The primary goal for Fall 2025 was to prepare a working prototype of Xplorer for a live test in a pool or in collaboration with a program with Cayuga Lake.

The mechanical team focused on…

The electrical team focused on enabling reliable underwater movement by integrating thrusters with a robust power distribution system and long-distance data communication. Key challenges included waterproofing a 50-foot tether carrying both power and data connections, as well as designing systems to support camera streaming and sensor feedback.

The computer science team…

Project Overview

Purpose

What is the project trying to achieve?

Semester Objectives

What is the project trying to achieve this semester?

Subsystems

Explain what the subsystems are for the project, including nicknames used to refer to them. E.g. nobody outside of Cup would know what we mean when we refer to a MechE "basekit"

Mechanical

Project A

Explain mechanical objectives on project A

Subsystem A.1

Explain the purpose of the subsystem and objectives for the semester. From the below subheadings, pick the ones relevant to your specific project:

Issues with previous design

If you're redesigning a system, why does it need to be redesigned?

[New Design]

Explain your new design

Previous Iterations

Explain (briefly) the older versions you went through to get to the final design

Fabrication Plan

What is your fabrication plan?

Fabrications

How did fabrication go?

Project B

Explain mechanical objectives on project A

Subsystem A.1

Explain the purpose of the subsystem and objectives for the semester. From the below subheadings, pick the ones relevant to your specific project:

Issues with previous design

If you're redesigning a system, why does it need to be redesigned?

[New Design]

Explain your new design

Previous Iterations

Explain (briefly) the older versions you went through to get to the final design

Fabrication Plan

What is your fabrication plan?

Fabrications

How did fabrication go?

Subsystem A Propeller Control System

The mechanical objective of this system is to build a stiff connection between motors, propellers, and fix within the electrical box without breaking the water-proof characteristics of the mechanical system.

Issues with previous design

We are building a submarine robot system from the beginning trying to scale it down and make it accessible economically.

New Design

To enable the propeller rotation to downward direction providing thrust to the robot, we consider having a motor to control its orientation. Therefore, we developed a series of connections: connector between rod and motor, wall of box, gasket, bearing, rod, connector between rod and propeller. In detail, I designed the mount from metal rod to propeller. It was previously press-fit on both ends; later it is optimized to have doves to better fix to the connector with the rod.

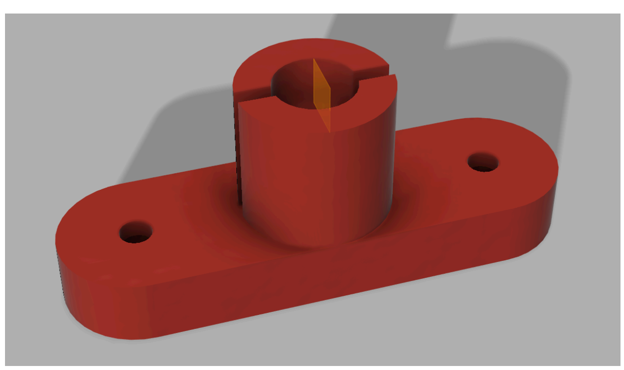


Figure I: connector between propeller and rod connector

Fabrications

We started with a box and manufactured it with two holes on the right/left sides to enable the rotation of the propeller by metal rods. To handle the risk from water leaking, we researched and designed layers in hope of sealing the connection between bars and the box mentioned above.

During the manufacturing and assembling process, we found the uncertainty in water proof: the accuracy of holes on the wall is supposed to be slightly smaller than the outer circumference of the bearing so that the bearing can seal the possible leaking. However, the manufacturing of the hole cannot be achieved with the expected level of accuracy with the process of drilling and small rotary sawing. Besides, drilling a small hole on each side of the wall to connect wires of propellers to the power source would have similar processing problems. For the successful demonstration of the idea in MIT, the hot glue is applied to seal the gap.

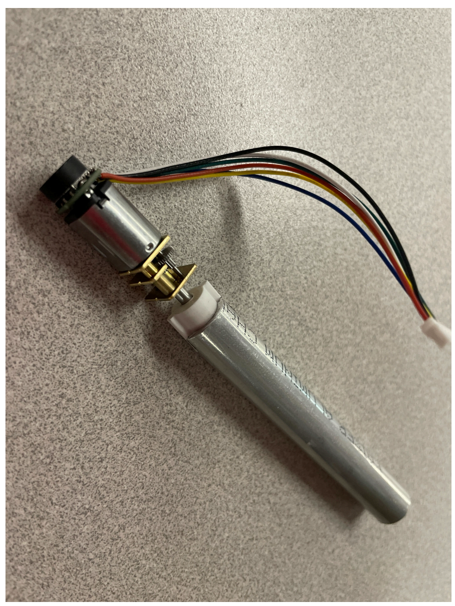


Figure II: layers of design on motor side

The problem of uncertainty of water proof has been brought up and discussed at an early stage. However, it did not get successfully resolved eventually. The lesson would be to make the design divided into different segments, by other saying to make it modular, so that water-proof property of the connection module can be tested individually and integrated into the design.

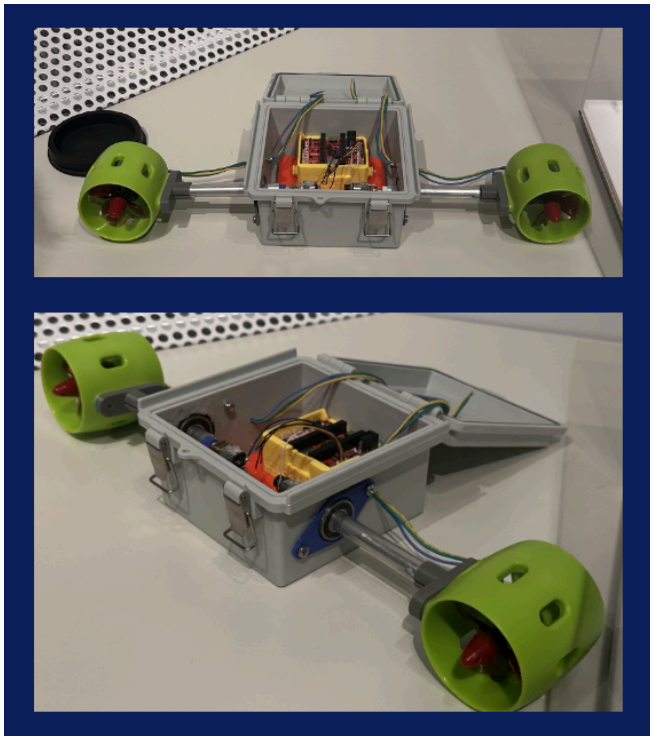


Figure III: Xplore Version One

Due to concern about water proof property, Xplore shifted to another design of cylinder body proposed by Edwin. It was considered and chosen based on the consideration of momentum of inertia, which implies the stability of submarines under water with less possibility of overturning. In this stage, I was involved in the design of the mount for the temperature sensor, the design of the water-sample collection device, and the rail system that lays inside the cylinder.

Subsystem B.1 Temperature Sensor

[New Design]

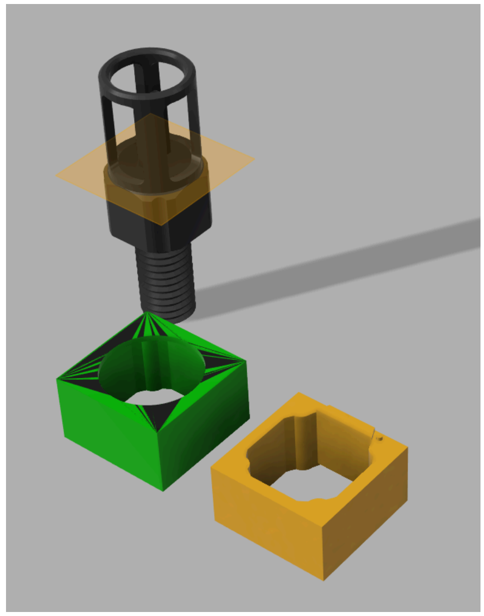


Figure IV: Temperature Sensor Mount

I designed these mounts by focusing on attaching to the segment below the shown orange plane. The outer contour of the segment matches with the inner contour of the mounts; two colors represent two methods that I approach the fixation of mount onto the sensor. Green one has the transition from its contour to a smaller circle as shown on its top; it works like a cap: they are attached together once the sensor goes though the mount by top fixation. The yellow one is exactly following the contour of the sensor, so in order to enable better press fit, the contour gets offset inward and a small clip is designed on the side to make the sensor slide in the mount horizontally and get fixed in location by clipping in the small door.

Fabrication Plan

It was simply 3D printed, so no more details about fabrication and assembly detail to expand.

Fabrications

A takeaway from this would be aware of the dimension of the design which can directly affect the effectiveness of the design. However, this can also be overcome by practicing design and prints.

Subsystem B.2 Rails

I designed the rails attached to the cylinder which provides support and fixation of separation boards where electronic parts mount to.

Issues with previous design

We referred to design from blue robotics a lot in the container for electronic parts. However, they do not have access to a 6-in diameter rail which can potentially maximize the space inside for various electronic parts. This is the basic need for this design.

New Design

The traditional rail has much smaller cross section area, while this version of rail got expanded intentionally to provide area for connection between segments by press fit. I tried to design dovetail on the traditional relatively small areas on trails; however, the limitation of dimension significantly deviated the effectiveness of dovetail for tighter connection. I learned to use a mirror function in CAD to improve the functionality of design this time,which makes up for the imperfection of the initial design and ensures the symmetry of the rail. Besides, I learned different orientations of extrusion to create slots on rails. One way is to create a plane along two inward edges and use the function of AAA to extrusion the slot, whose direction is perpendicular to the constructed plane, while the other way is to sweep which enables the extrusion along the radius which is not applied here though.

For the future, I have one point to develop better: one is to develop more mature design methods for segments like these which share similarity but not exactly identical. I totally remembered that I used multiple times of extrusion recording down lots of numbers to distance them so that they wouldn’t overlap with each other and modularize them for more convenient interaction with other parts. However, the length of each unfortunately are not the same at the end with potential for error happening in the middle.

Fabrications

It was 3D printed. Besides, some adjustments were made to ensure the press fit on each ends.

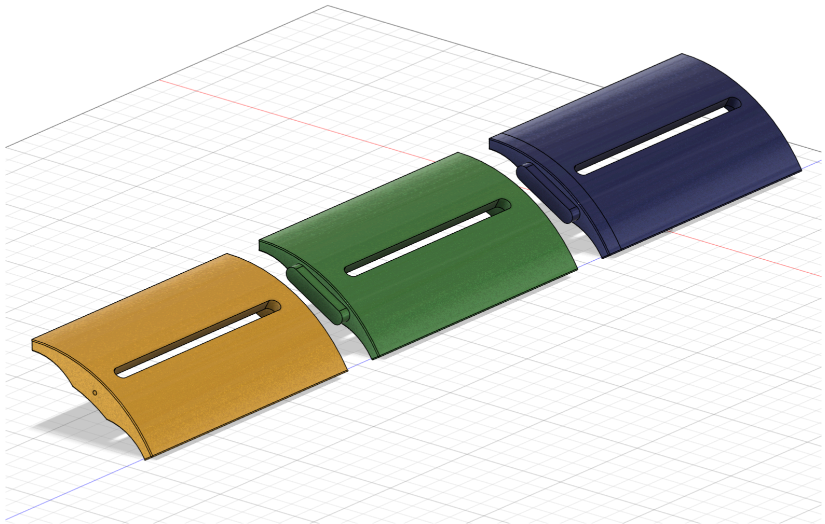


Figure V: Rails providing support for electronics design-mounted boards

Servo to Rod Mount

Purpose

The initial design of the Xplorer bot proposed using dual propeller propulsion, with the propellers attached to rods attached to a motor. These rods would be able to rotate, changing the pitch of the propeller thrust and enabling depth control. This mount was to be the connection between the servo motor and the rods extending out to the propellers. It was a preliminary design and not fully developed as we moved away from this method of depth control, but it has merit as a general servo horn clip.

Design Process

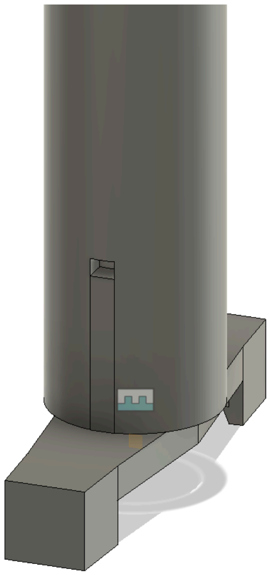
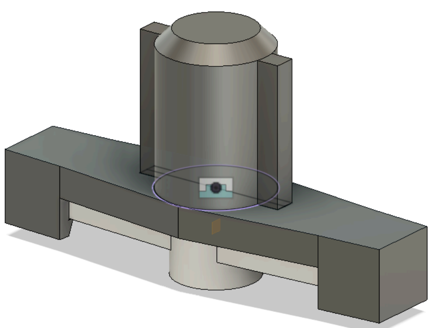


Figure XX: servo horn mount connects to rod



FigureXX: servo horn mount viewed from side connected to the horn

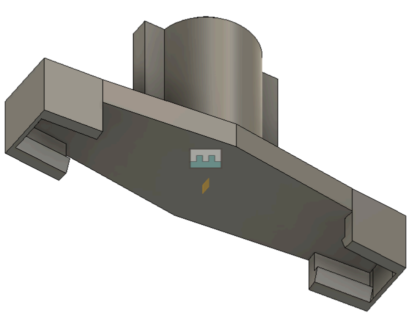


Figure XX: servo horn mount viewed from underneath showing snap fit geometry

This mount was designed to be a snap fit, snapping on over the servo horn. The thought process was that the servo horn comes with the servo and so fits well. Rather than trying to make something that fits equally well to the servo, we could just design something to snap onto the existing horn. It is a fairly straightforward design, with the mount snapping over the horn, and flanges underneath preventing it from coming off once snapped into place. The mount is fitted to the geometry of the rod so that the rod can snap on, with the flanges fitting into gaps cut into the rod to remove any concern about the ability of the snap fit to provide sufficient friction.

Manufacturing Process

It was easily 3D printed, with short print time and no assembly required.

Assembly

Once printed it could be immediately snapped into place and used. To disassemble simply pull until it clicks off.

Next Steps

The interesting thing about this mount is that the connection to the rod is a separate body in the CAD, apart from the servo horn mount. This enables this mount to be used on other projects where the servo is connected to a different geometry. Simply extrude a different connection off the surface of the mount geometry, and you can connect to any range of possible objects.

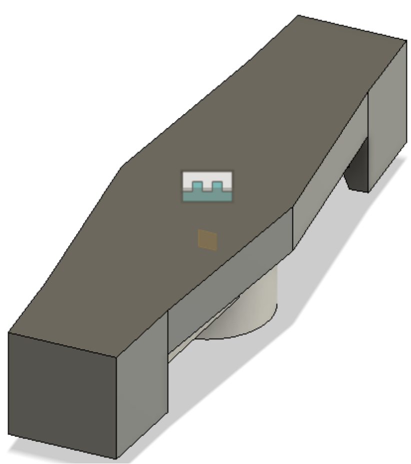


Figure XX: base servo horn clip without body that connects to the rod

If you decide to take this geometry and use it in this way, I would recommend testing and tightening the tolerancing of the mount, as it was never fully perfected with the Xplorer moving to a different design. The geometry resists rotation well and is capable of providing torque. However, the tolerancing was not perfected which led to some wobble during use. After a dozen or so cycles snapping on and off, the fit further weakened, leading to increased wobble which raises concerns for prolonged use.

N20 Mount

Purpose

When the Xplorer transitioned from servo motors to N20 motors, mounts were needed to hold the N20 motors in place. The N20 motor has an interesting and rather complicated geometry making it challenging to design a well performing mount. Several mounts were designed and tested, with this design being one of the ones proposed.

Design Process

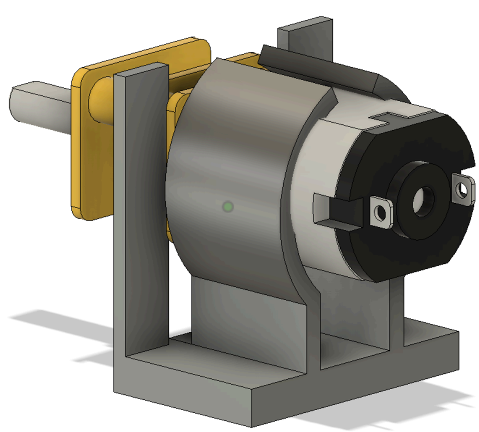


Figure XX: N20 mount viewed from the front attached to the motor

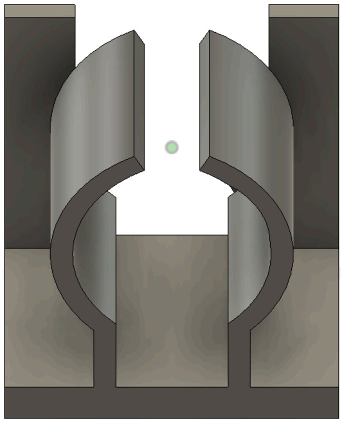


Figure XX: N20 mount viewed from the front

This mount was designed as a snap fit, with the curved pieces flexing outwards to snap onto the main body of the N20 motor, then snapping back into place tightly around it. This prevents movement radially (and is intended to prevent rotation, more on that later), but it does not prevent movement axially, which is why the slats fit behind the square cross section of the motor. The mount could be glued or drilled onto any flat surface, or a different bottom to the mount could be extruded to help it mount to a surface with a more complex geometry. This mount does a good job of preventing axial movement, and a decent but not great job of preventing radial movement. It is easy and quick to print, and the N20 can be easily snapped into place.

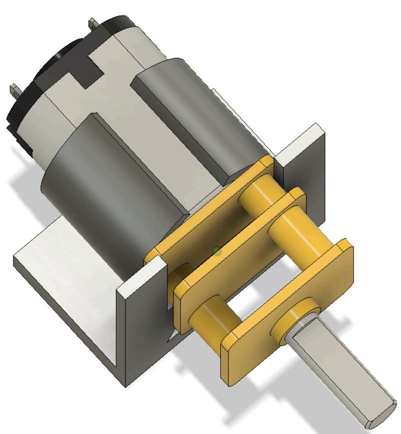


Figure XX: back view of the mount fit to the N20 motor

However, the major drawback of this mount, and the reason a different mount was used in the Xplorer final design, is that it does a poor job of preventing rotation. Steps were taken to improve this feature, with the slat changed to slide under a rod in the complex geometry at the back of the motor to prevent rotation. However, the problem remained, and solving it with the design approach would have been unnecessarily complex when alternative designs existed that were better suited for the task.

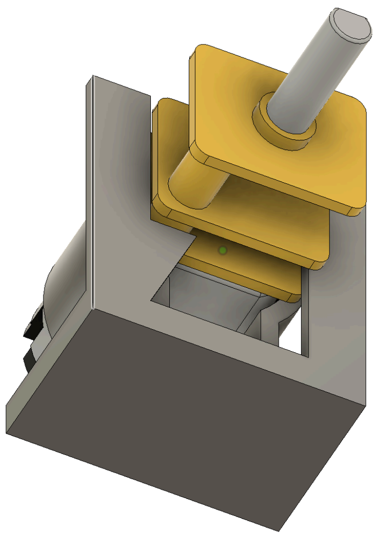


Figure XX: N20 mount viewed from below showing slat implemented to prevent rotation

Manufacturing Process

This mount is entirely 3D printed.

Assembly

The N20 motor snaps into the mount. The motor may need to be turned at an angle initially in order to fit. Make sure to line the vertical slats so that they slide in properly behind the square plate on the motor. To disassemble, simply pull the motor out of the mount.

Next Steps

This mount does not have a great future as an N20 mount in high torque applications, as it simply does not prevent rotation to the desired degree. However, it is quick and easy to print and use, which does give it some potential future use as an N20 holder/organizational piece.

Subsystem Waterproof Electronics Housing and Thrust Vectoring

The purpose of this subsystem was to house all electronics in a waterproof enclosure while allowing rotating impeller shafts and wiring to interface with the external environment. The objectives for the semester were to design a sealed housing, enable multi-degree-of-freedom motion using differential impeller control, and demonstrate a functional underwater prototype.

Issues with previous design

Early concepts relied on modifying currently existing, off the market parts, which introduced uncertainty in waterproofing and increased fabrication complexity. The design was also constrained by long lead times for off-the-shelf components and limited ability to iterate custom parts.

[New Design]

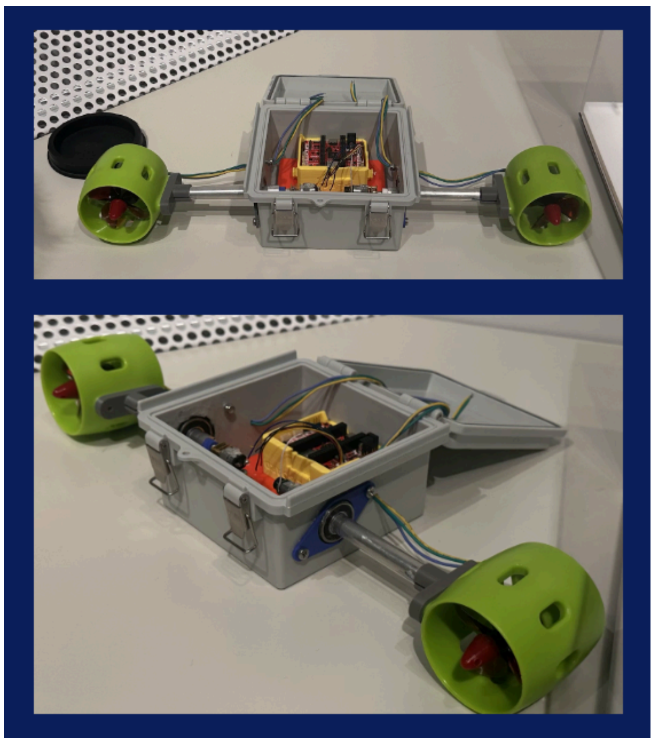


Figure I: First Xplorer Iteration

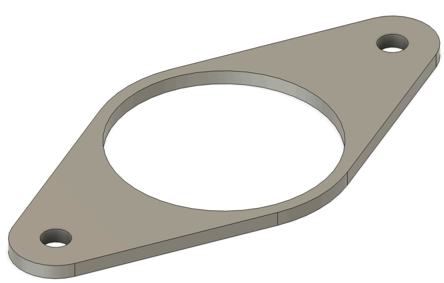


Figure II: Gasket Clamp

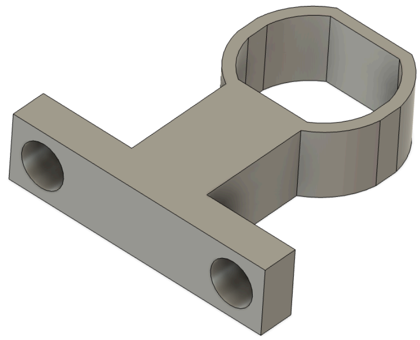


Figure III: N20 Clamp

The final design used a commercially waterproof Pelican case as the electronics housing. Rotating impeller shafts exited the enclosure through drilled sidewalls and were sealed using a layered approach consisting of sealed bearings, an O-ring surrounding the bearing, and epoxy to secure any manufacturing imperfections. Vertical motion was achieved by connecting the impeller rod to a servo motor that adjusted the thrust outlet angle, allowing ascent and descent using a single actuator. An external frame surrounded the housing to support sensor mounting and future attachments. Custom CAD components designed for this subsystem included impeller-to-rod connectors, a servo mount, a servo-to-rod adapter to translate rotary motion, and a gasket clamp to retain the O-ring.

Previous Iterations

Initial design ideas explored fully custom waterproof housings with gasketed wire exits. These were later replaced by the Pelican case–based design to improve waterproof reliability and reduce fabrication risk. The thrust vectoring mechanism was refined through multiple iterations and supported by free body diagrams and moment of inertia calculations to improve alignment and assembly.

Fabrication Plan

Fabrication relied on a combination of off-the-shelf hardware and FDM Bambu 3D printing. Sealing was achieved using commercially available sealed bearings and O-ring gaskets, with epoxy applied to reinforce sealing at penetrations. Holes were cut into the box itself to house the gaskets while the motors and electronics were mounted to a 3D printed slab that gave enough depth to screw in the N20 mounts.

Fabrications

Fabrication presented several challenges, primarily related to modifying the Pelican case. Cutting and drilling holes into the enclosure was difficult due to the thickness and stiffness of the material, which made it hard to maintain alignment. Several shaft holes were slightly off-center and exhibited jagged edges from the cutting process. These imperfections increased assembly difficulty and introduced uncertainty in shaft alignment and sealing. Additional epoxy was required to compensate for surface irregularities and improve stability. While the prototype was functional for demonstration purposes, these fabrication issues highlighted the limitations of modifying a commercially sealed enclosure and motivated consideration of alternative fabrication methods for future iterations.

Ballast-Controlled Xplorer — Prototype I

Purpose

Prototype I was developed to demonstrate the core motion architecture of the Xplorer. This prototype also served as an exploration of materials and geometry, and as a proof of concept that a syringe-based ballast system can provide effective vertical control, reducing mechanical complexity and cost relative to multi-propeller (three- or four-thruster) designs.

Design Process

The design of Prototype I was guided by ease of fabrication and assembly. The vehicle needed to support controlled vertical and horizontal motion, remain structurally sound when submerged, and be inexpensive and straightforward to put together using readily available materials. Emphasis was placed on creating a system that could be understood and built by students in a broad age range.

This prototype uses a single syringe-based ballast system, driven by a rack-and-pinion mechanism, to control buoyancy and move up and down in the water. Horizontal motion and turning are achieved by using different thrust levels between two propellers. This prototype was built to test whether these basic motion methods work together reliably in a single system.

Material choices focused on simplicity and ease of construction. PVC was used for the outer shell because it is naturally waterproof, inexpensive, and cylindrical, which helps distribute water pressure evenly around the vehicle.

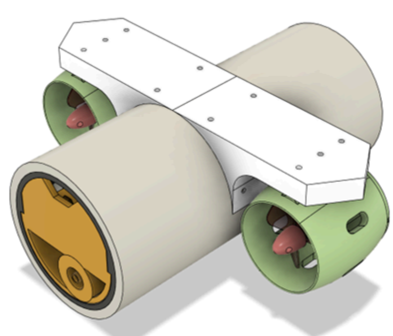


Figure X: Isometric view of the first ballast-controlled Xplorer prototype.

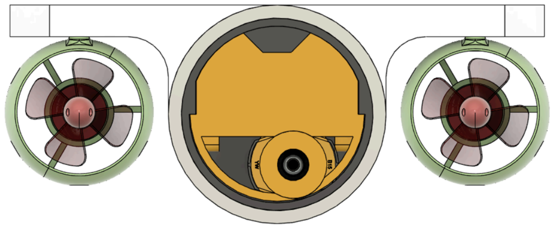


Figure X: Front view of the first ballast-controlled Xplorer prototype.

Implementation

Manufacturing Process

Internal components were designed in Fusion 360 and fabricated using 3D printing to enable rapid iteration, customization, and easy modification of mounting geometries.

Propulsion and in-plane steering was provided by two commercially available 12–24 V U01 brushless underwater thrusters. The outer hull, constructed from a PVC pipe, was cut to length using basic hand tools. The length of the pipe was chosen to accommodate all internal components while maintaining a compact form factor and balanced mass distribution along the vehicle's axis. This ensured sufficient internal volume for electronics and ballast while avoiding unnecessary excess length that could increase drag or reduce maneuverability.

The ballast system was actuated using a high-torque N20 DC gearmotor with a 100:1 gear reduction, controlled directly by the XRP board. The high gear ratio provided sufficient force to drive the rack-and-pinion syringe mechanism while prioritizing slow, controlled ballast motion.

Assembly

These thrusters were mounted symmetrically about the vehicle centerline and near the longitudinal midpoint of the hull. This placement minimizes unwanted pitch and roll moments during thrust changes and allows differential thrust to produce yaw control without introducing large rotational disturbances.

The internal components, including the XRP control board and battery pack, and ballast system were mounted inside a lining tube that slides into the main PVC hull. This lining tube included an integrated rail to hold the components in place and maintain alignment during insertion. The lining tube was press-fit into the outer PVC pipe, allowing it to be securely held without additional fasteners.

Basic waterproofing was achieved using two PVC end plugs that fit into the outer hull. These plugs provided a visual (yea, ONLY visual) seal at each end of the pipe, allowing the vehicle to be assembled and disassembled easily for testing and iteration.

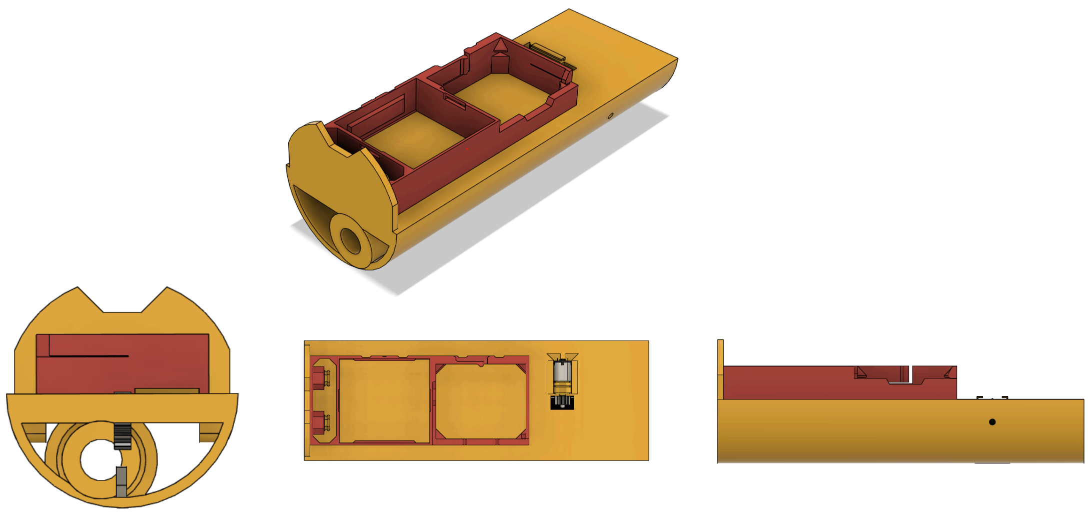


Figure X: Internal Assembly for the first ballast-controlled Xplorer prototype

Next Steps

Prototype I successfully met its demonstration goals by validating the core motion architecture and basic assembly approach of the Xplorer. The prototype functioned as intended and provided confidence that the ballast-based control and differential propulsion scheme could work together in a single system. A key takeaway from testing was the need for a sturdy, low-friction constraint opposing the pinion gear to prevent rack slippage and unintended rotation of the syringe ballast plunger during actuation.

Following this proof of concept, a newer, larger, and more robust prototype was quickly pursued.

Ballast-Controlled Xplorer — Prototype II

Purpose

Prototype II was developed to take the motion philosophy validated in Prototype I and implement it in a fully functional submarine capable of sustained underwater operation. In addition to preserving the ballast-based vertical control and differential propulsion scheme, this prototype was designed to support real-world use by housing a Pi camera, depth and pressure sensors, a temperature sensor, as well as water sampling capabilities, and the associated supporting electronics required for operation.

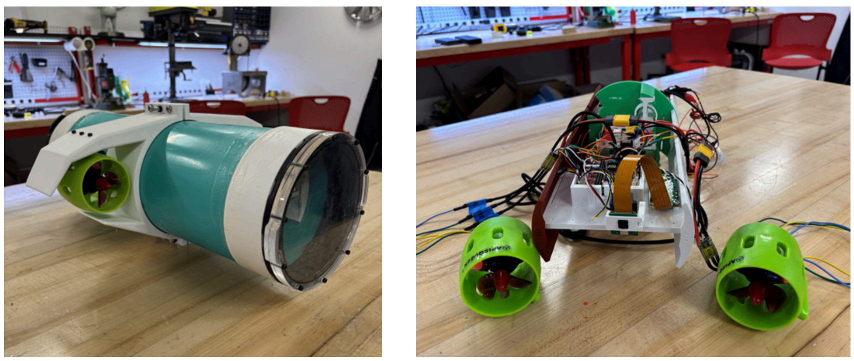


Figure X: Assembly of internal and external components for Prototype II

Design Process

Prototype II followed a design philosophy similar to Prototype I, with changes made to support reliable underwater operation and a larger overall system. PVC was again used for the outer hull, this time with a larger diameter to accommodate additional sensors, electronics, and a more capable ballast system.

The internal assembly was designed to slide in and out of the hull along internal rails. This allowed the internal components to be worked on, tested, and modified outside of the vehicle before being reinserted and sealed. This approach simplified debugging and reduced the risk of damaging components during assembly.

One of the main challenges encountered was a mismatch between the hull diameter and the waterproof end flanges that were available. This was addressed by 3D printing a PLA adapter piece to bridge the size difference. The joint was then sealed using a waterproof thread-locking compound and an additional flexible waterproof coating to ensure a reliable seal. One end of the hull was later replaced entirely with a pipe gripper, which preserved the ability to easily remove the internal assembly while maintaining a secure seal.

Another key challenge was the increased size and mass of the vehicle. A more complex ballast system was required to produce a noticeable change in depth. This was addressed by using two syringe-based ballast chambers driven through a compound gear train, allowing sufficient force and displacement to be achieved. Water sampling was implemented using a similar rack-and-pinion–driven syringe mechanism, but with a one-way check valve to allow controlled intake without backflow.

As the design increased in size and complexity, it also developed greater design inertia, becoming less modular and harder to modify once assembled. This highlighted the importance of prioritizing modular mounting strategies, allowing individual subsystems to be adjusted, replaced, or improved without requiring major disassembly of the vehicle.

Implementation

Manufacturing Process

All internal mounting components were designed in Fusion 360 and fabricated using 3D printing. Two 60 cc syringes were commercially purchased and integrated into the ballast system, and a 20 cc syringe was purchased for the water sampling. The same underwater propellers used in Prototype I were reused for propulsion.

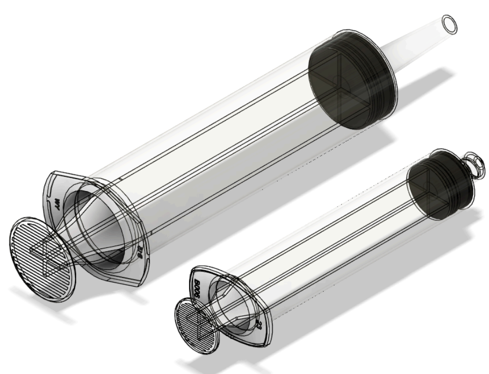


Figure X: 60 cc and 20 cc syringes

A custom propellor mount was 3D printed primarily for aesthetic purposes. Waterproof flanges and sensors were purchased from Blue Robotics.

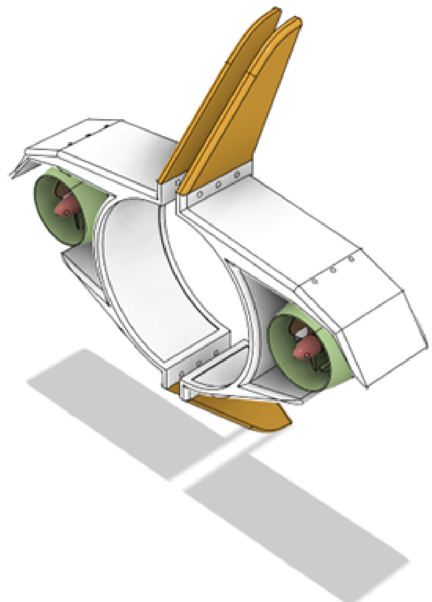


Figure X: Propellor Mount

Trays were 3D printed with electronics mounts integrated directly into their geometry, allowing components to be secured without additional brackets.

Assembly

The printed trays were slid into the internal rails, which were press-fit together to form a single continuous internal assembly. This created one removable unit containing the electronics and ballast components, allowing the entire internal system to be inserted into or removed from the hull as a whole.

A PLA adapter was installed on one end of the hull and sealed using a waterproof thread-locking compound and an external flexible waterproof coating to create a reliable seal. The propeller mount was secured to the motor shaft using a shaft collar to prevent axial movement during operation.

Holes were drilled through the hull to route the propeller wiring, and a WetLink penetrator from Blue Robotics was used to maintain a waterproof electrical connection through the hull. The ballast system was integrated between the trays, with the syringes positioned near the longitudinal midpoint of the vehicle to minimize pitch disturbances during ballast actuation.

Blue robotics parts can be found here:

https://bluerobotics.com/store/watertight-enclosures/locking-series/wte-end-cap-vp/

https://bluerobotics.com/store/cables-connectors/penetrators/wlp-vp/

Next Steps

Future work includes designing custom WetLink-style penetrators to reduce cost and achieve a more reliable waterproof seal tailored to the vehicle's specific geometry, particularly since non–Blue Robotics propellers are being used.

The Ethernet connection that powers the submarine should also be routed through the end-of-pipe gripper, which has not yet been implemented. Additionally, now that full access to the internal diameter has been restored through the use of the pipe gripper, the internal trays should be redesigned in a more secure and rigid configuration to better support the electronics and ballast system.

Additionally implementing feedback control on the ballast system using depth and pressure sensor data, enabling closed-loop depth regulation rather than open-loop actuation should be an endgame goal.

Figure X: Snap-fit locking mechanism for the new trays to secure ensure inner components are rotationally fixed

Ballast System

Purpose

The ballast system was designed to provide depth control while remaining easy to assemble.

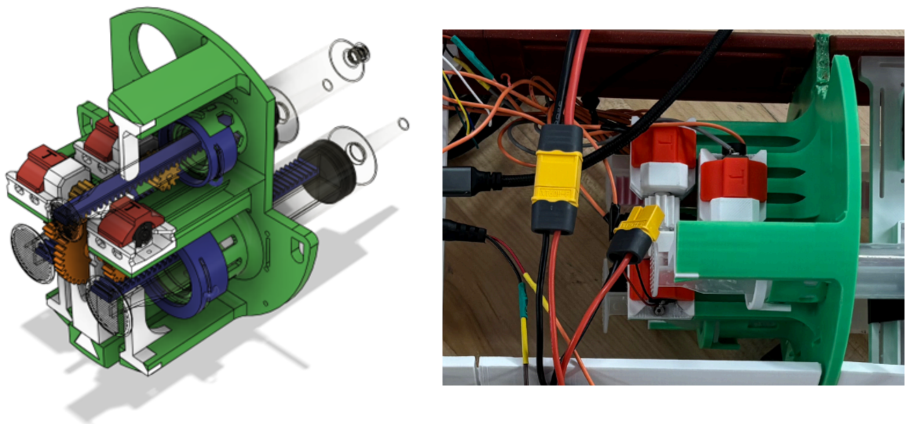


Figure X: Assembled Ballast System

Design Process

A key challenge in designing the ballast system was balancing printability with structural strength. Many components were interdependent, but printing them as a single piece would either require extensive support material or compromise strength at critical interfaces. To address this, the design was centered around a primary circular plate that incorporated all features that could be printed as one piece without supports. Dovetail joints were then integrated into this plate to allow additional components—such as securing elements identified as necessary in Prototype I and a C-clip for the drive axle—to be attached later at 90-degree orientations as separate printed parts. Holes were left for custom N20 mounts, and an assembly was made in Fusion to ensure everything fit together.

Another design goal was to ensure that the syringes were both removable and securely constrained while the plungers were driven back and forth by the pinion gear. This was achieved using a snap-fit mechanism similar to a pill bottle cap, where retention is provided through rotational engagement rather than linear insertion. This allowed the syringes to be locked in place during operation while still being easily removable for maintenance or replacement.

Because Prototype II uses multiple syringes, the system experiences greater static friction during actuation. This required additional mechanical advantage from a gear train. However, excessively gearing down the system would result in very slow ballast actuation, introducing an undesirable time delay in depth control. To avoid this, two N20 motors were used to drive the ballast system simultaneously, providing sufficient torque without overly reducing actuation speed. For spatial efficiency, the water sampling mechanism was integrated into the same assembly.

Implementation

Manufacturing Process

All parts beside the syringe and N20s were designed in Fusion and 3D printed.

Assembly

Racks were first attached to the syringe handles, and the syringes were secured in place within the ballast assembly. The 90-degree securing components were then press-fit into the main base plate, completing the structural framework of the system.

The compound gear was installed onto the drive axle and retained using a C-clip. The N20 motor mounts were then fastened to the base plate, and the N20 motors were installed and meshed with the driven gears. This completed the mechanical assembly of the ballast actuation system.

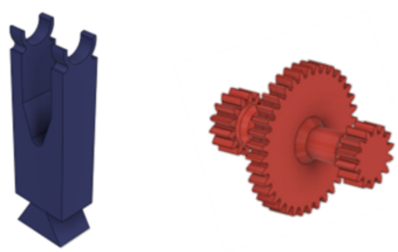


Figure X: C-clip and simple compound gear

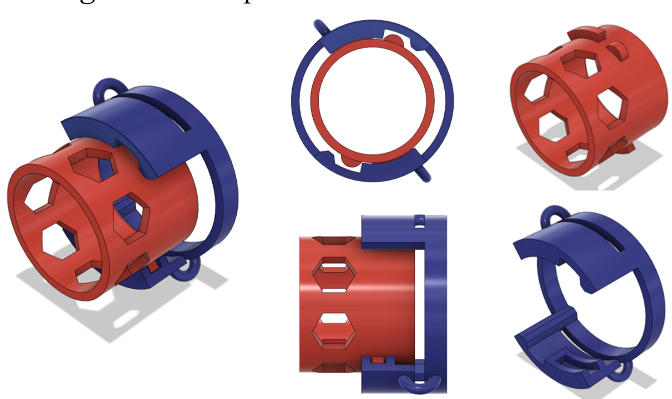


Figure X: Syringe Mount (Isolated)

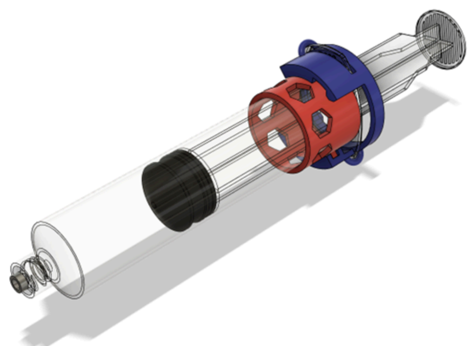


Figure X: Syringe Mount (Assembled)

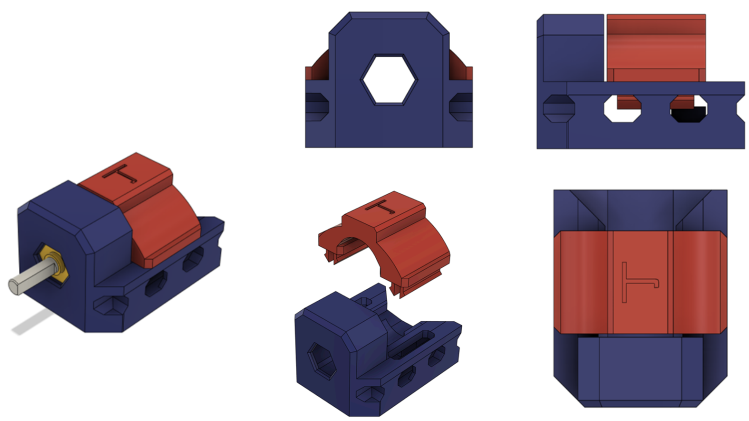


Figure X: N20 Mount

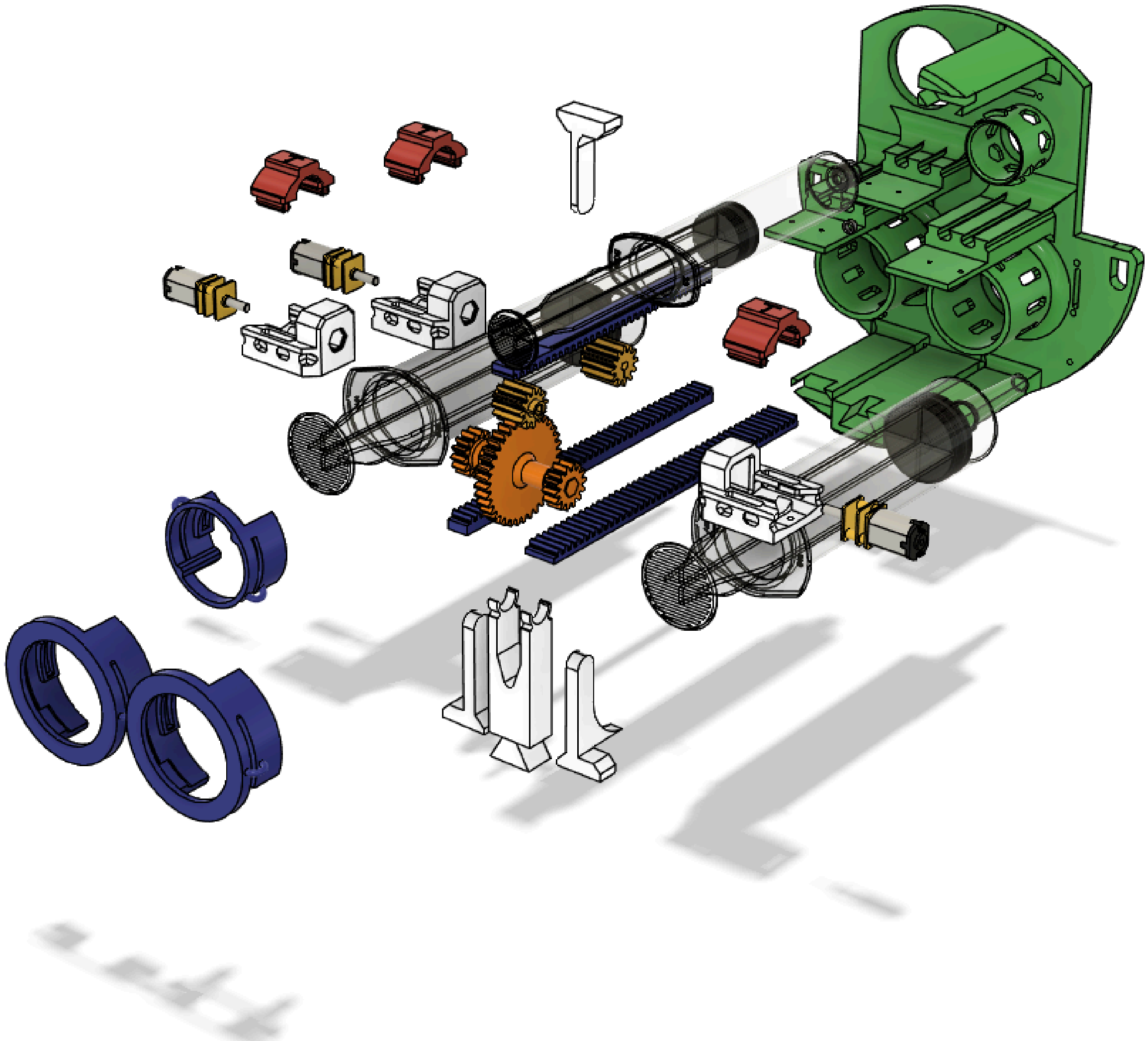


Figure X: Ballast Assembly (Exploded View)

Next Steps

Future iterations should strengthen the dovetail joints to improve overall strength and long-term durability. The geometry of the base plate will also need to be updated to integrate with the redesigned trays and rail system.

Additionally, the N20 motor mount connection should be redesigned to eliminate fasteners, moving toward a fully snap-fit assembly to improve modularity and ease of assembly.

Electrical

The electrical team was tasked with exploring novel ways of powering and communicating with the XRP over long distances and through deeper water depths. Bluetooth and Wi-Fi are unreliable in underwater environments, so the team looked into PoE (Power Over Ethernet) as the primary means of data transmission and reliable power supply to the Xplorer. Batteries for onshore power and emergency onboard power were also researched in order to supply reliable power to the motors. Team members also explored integration of various sensors to collect information about the underwater environment and support underwater education through data collection and analytics.

Underwater Propulsion

To traverse through water, powerful thrust provided by higher voltage motors are necessary for reliable operation.

Thrusters

Purpose

The thrusters should provide the necessary thrust to be able to traverse through water and have waterproof cabling.

Testing

At first, parts were bought separately, specifically for the thrusters developed by APISQUEEN. While testing, the team decided to purchase the whole set along with some backup parts in case those broke. After they came in, watched the video on the apisqueen website and got the thrusters to work using the remote control. Eventually, the team got the thrusters to work using a remote control.

To get the thrusters working with the XRP, a PWM signal was required to be sent to the thruster encoder (ESC). The ESC is a small controller that controls power to the brushless motor, controlling motor direction and speed. The UBEC component was important as well because it provides stable low-voltage power to the receiver and motors from the main battery. This unit ensures stable voltage regulation even while the motor draws a lot of power.

Control and Implementation

The team developed PWM code typically used for servos. This code was tested and plugged the signal inputs of the thruster ESCs into the servo pins of the XRP, and eventually got the thrusters to spin using the XRP board. Then, functions were written to control the turning right, left and going forward and backwards. To power the thrusters in testing, 12VDC was used from a power supply.

The thruster control uses PWM, where a square wave of the sent pulse width determines the amount of time the signal is on. The thruster datasheet contained information about the PWM pulse, where backwards was stated to be 1 ms, stopped 1.5 ms, and forward 2 ms. By sending 2000 to the servo, it spins the propellers forward, 1500 stops the propellers, 1000 goes backwards.

For the controls, five functions were written: forward, left, right, backward, and stop. Each of the directions sends a different combination of 1000 and 2000, while stop sends 1500 to both propellers. In the PWM code, five functions were made for forward, backward, right, left, and stop. Each function sets each of two thrusters with a pwm number in order to provide the correct movement. There is a main loop that goes through each of the five movement types to test that they work. We specifically use the function `servo._servo.duty_ns(us * 1000)` where us is the pwm number.

Next Steps

Some next steps would be to test the thrusters more thoroughly with the POE power system and tune the power parameters as well as the thruster code to optimize the thruster's power usage with thruster speed.

Power Distribution and Data Streaming

The Xplorer relies on a robust power distribution and data transmission system to allow the user to interact with the robot.

Battery Research

Purpose

The goal is to find a battery that would be able to fit within the physical constraints of Xplorer and would be able to provide enough power for Xplorer to run for approximately 30 minutes as an emergency power backup.

Design Overview

The first step was determining the power needs of Xplorer. While the needs for the Pi with the sensors still is unknown, each propeller needs approximately 8 watts and the n20 motors each require about 0.4 watts, based on testing outside of water. So, for thirty minutes of power, the battery needs to be at least 8.5 Wh, preferably with plenty of wiggle room for running the sensors and camera. Additionally, the battery should provide at least 16 V because the propeller prefers to run off 16 V. The only one that fit was a 16.8 V, 13.44 Wh NiMH battery.

Next Steps

For the battery, it must be determined how it should be connected with the rest of the system. Additionally, since the battery is meant to be emergency power, it needs to be figured out how the battery will only provide power if it is needed. There are two aspects of testing that needs to happen, one is confirming the battery works by using it as a power supply, and determining that it can provide the necessary power. Second, after determining how to hook the battery up with the full system, simulating a need for emergency power and determining everything can still operate. Also, what is meant to happen when the battery is being used needs to be considered, as communication is necessary to operate the Xplorer, but communication and power comes through the same tether, so how Xplorer will operate while on battery power needs to be discussed.

Power-over-Ethernet (PoE)

Purpose

PoE allows for data and power transfer over a single ethernet cable. The goal is to minimize the quantity of cable and increase data throughput over a single jacketed cable.

Design Overview

We went through three iterations of different power configurations for the XRP submersible. This was our first time working with Power over Ethernet (PoE), which required a significant amount of experimentation and iteration. From the outset, our primary objective in using PoE was to enable high-speed, low-latency data transmission while minimizing the number of physical cables required. We opted for waterproof ethernet cables for this task.

PoE was selected because the submersible operates underwater and Ethernet provides reliable, high-bandwidth communication and can be extended over long distances without significant signal degradation. At the same time, we had to account for the varying power requirements of different system components, including the motors, Raspberry Pi, and XRP controller. Balancing data transmission needs with power distribution constraints was a key design challenge.

The general design entails two cables: one of which the user will connect to the XRP directly through USB extended over cat6 cable, and another cable which will carry power to the system as well as communication with the Raspberry Pi over a direct LAN connection.

The user will connect to the Raspberry Pi over an ethernet cable and SSH login to the hostname raspberrypi.local. This specific hostname will allow the user to automatically find a Raspberry Pi over the local area network.

The development of power and data transmission through the other cable can be described in three phases. Phase 1 is a preliminary design which was somewhat functional but lacks stability in powering the circuit. Phase 2 is an improved version of the power distribution system which attempts to solve issues regarding the distribution of voltages and stable power to different components. Phase 3 attempts to build upon the design methodologies of Phase 2, parallelizing the circuit components to simplify wiring and prevent thruster transients from interrupting the other parts of the circuit. These three phases are discussed in more detail over the next few sections.

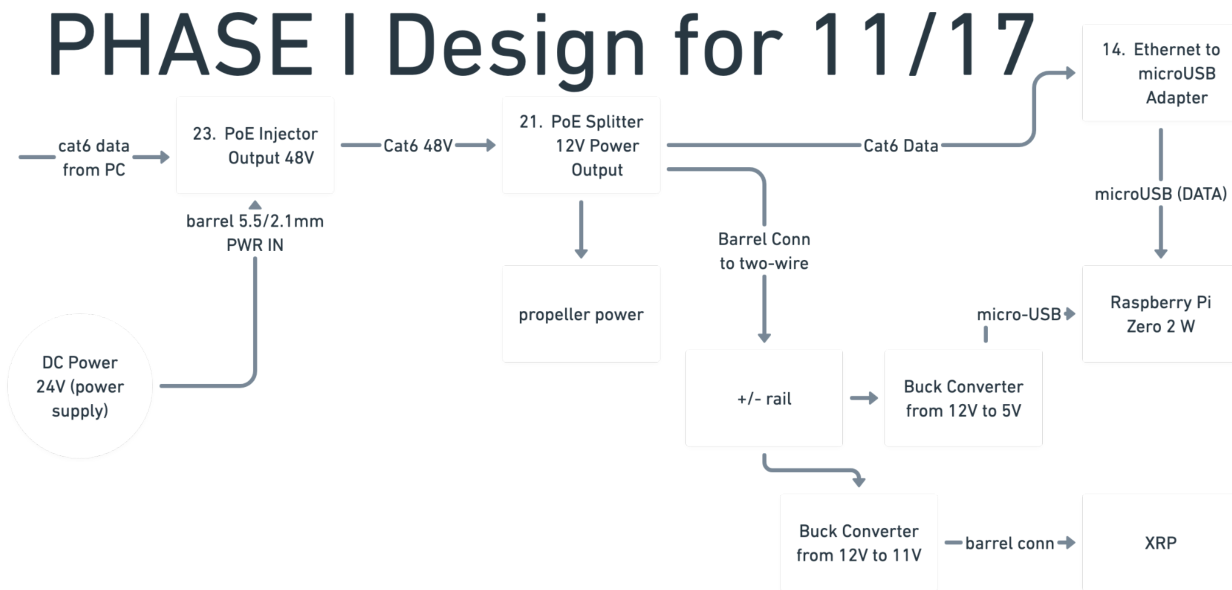


Figure X. Phase I Design

Phase 1 Design

The above diagram details the initial design, which was quickly assembled to meet the deadline for the National Geographic grant on November 17, 2025. The PoE injector serves as the primary power source, taking in 120VAC and supplying 48VDC at the output. The output power is then transmitted to the submersible via ethernet cable and then stepped down in various sequences to power the propellers, Raspberry Pi, and XRP controller. Because the Raspberry Pi and XRP require different operating voltages, multiple buck converters were used to further regulate and step down the supplied voltage for each board.

While all electrical connections were functional and the system successfully powered on, motor operation caused power instabilities. When a basic control program was deployed that simply toggled the motors on and off, the Raspberry Pi consistently shut down and rebooted. This behavior can be attributed to transient current spikes and global voltage sag introduced by the motors during startup. The sudden inrush current drawn by the propellers likely exceeded the instantaneous current capacity of the shared power distribution system, causing a temporary drop in voltage on the Raspberry Pi's supply rail and triggering an undervoltage shutdown. This revealed a key limitation of the initial power architecture in isolating sensitive logic components from high-current loads. The design of the next phase attempts to alleviate these issues.

PHASE II Design

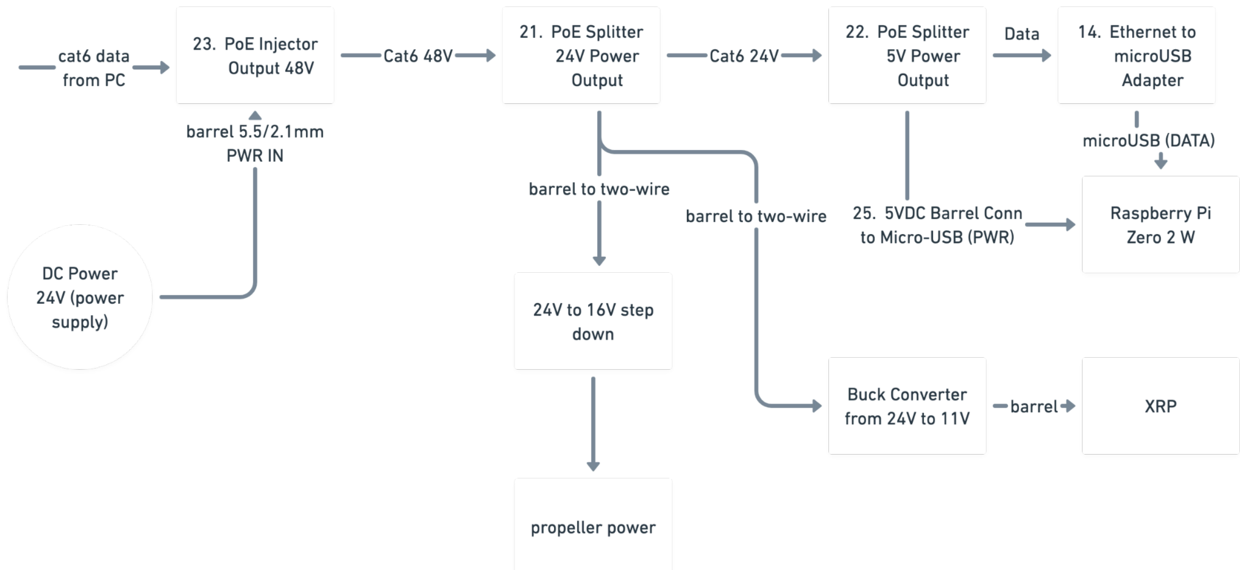


Figure X. Phase 2 Design

Phase 2 Design

The above design is a more optimized version of the previous iteration. The propellers operate most efficiently at 16 V, and this configuration delivers that voltage directly. Compared to the earlier design, fewer buck converters are required; instead, PoE splitters are used to handle most of the voltage step-down, simplifying the power distribution architecture.

Although the phase 2 design works in theory, testing results produced an undesirable outcome. Since part 21 (the PoE Splitter with 24V output) only outputs power over ethernet passively, the modules at the output are not able to provide a sufficient load for the passive splitter to detect that power is needed by the load modules. Thus, the rest of the circuit is inoperable and a new phase 3 design was developed.

PHASE III (Alt) Design

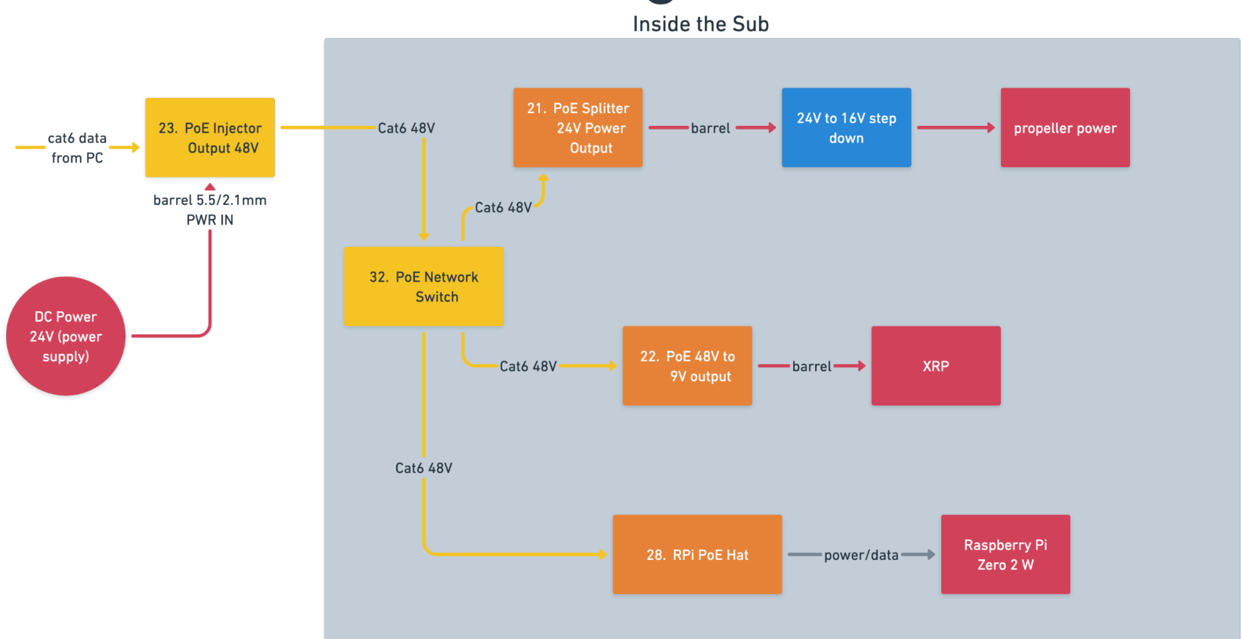


Figure X. Phase 3 Design

Phase 3 (Alt) Design

In the Phase 3 design, improvements were made in parallelizing data/power to the various systems and reducing series and branching connections. These changes helped to streamline the circuit design and help engineers to better understand where the different cables lead to. Instead of having splitting connections at each module, one network switch is used to create a parallel circuit with 48V PoE being provided to the three different components requiring different voltages, with the thrusters requiring 16V, the XRP requiring 11V, and the Raspberry Pi requiring 5V.

With this design, there is better voltage independence across the components without having to use multiple buck converters. This improved parallelism will hopefully reduce the effects of transients caused by the initial startup of the propellers which had affected the electrical stability of other components in the system in the Phase 1 design.

Next Steps

Next semester, the phase 3 design should be tested for functionality and power stability. In addition, waterproofing should also be explored for the two cables that will be tethered to the robot.

Sensor Integration

Temperature Sensor

Purpose

The temperature sensor will collect external water temperature data for environmental profiling and analysis.

Design Overview

The temperature sensor is a sealed, fast-response digital I²C sensor accurate to ±0.1 °C. It will be mounted externally on the submersible's enclosure with direct exposure to the ambient fluid but protected from mechanical damage. The sensor operates on a 3.3 V I²C logic interface. Data from the sensor will be routed through a 3.3 V I²C level converter to interface with the Raspberry Pi Zero 2 W's I²C bus.

Implementation/Testing

Pending acquisition of the I²C level converter, the sensor cannot yet be fully connected to the Raspberry Pi. Once the converter is installed, the implementation plan is to mount the sensor on the sub body with waterproof routing of wires. In addition, the team will connect sensor SDA/SCL to the level converter and then to the Pi's I²C pins. Communication between the components must then be validated with address scanning and raw reads. After the sensor is set up, the user can then record the temperature in Python.
GitHub - bluerobotics/tsys01-python: A Python module to interface with the TSYS01 temperature sensor.

Next Steps

The team will integrate I²C level converter and complete the wiring associated with the sensor. Driver code must also be written and validated on the Raspberry Pi. The temperature sensor will also need to be validated against calibrated thermometers.

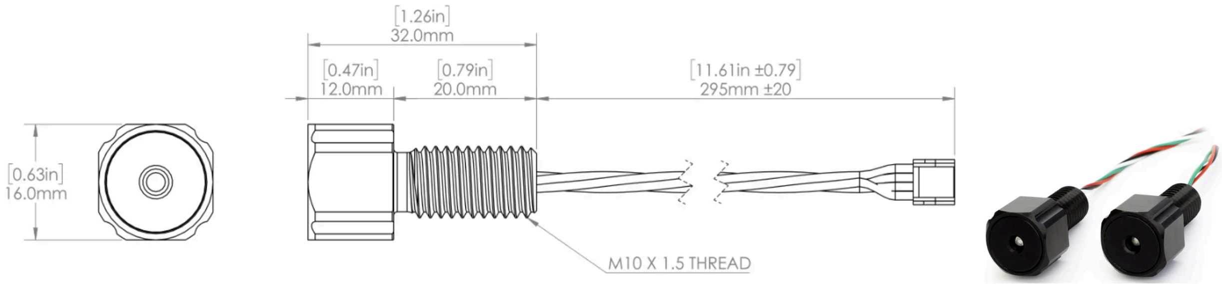


Figure X: Mechanical dimensions and external appearance of the depth and pressure sensor housing and cable

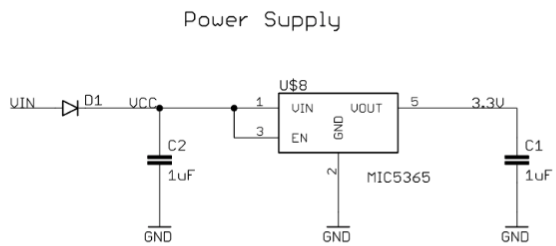


Figure X: Power supply circuit regulating input voltage to 3.3 v for sensor operation

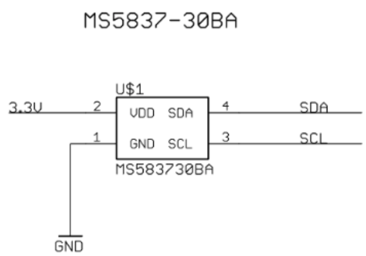


Figure X: MS5837 30ba depth and pressure sensor pinout and i2c interface connections

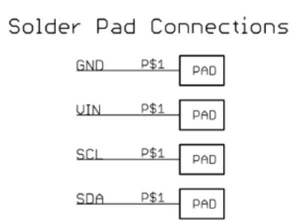


Figure X: Solder pad connections for ground input voltage SCL and SDA signals

Depth/Pressure Sensor

Purpose

Measure ambient fluid pressure and compute depth for navigation and environmental data logging.

Design Overview

The depth/pressure sensor is a high-resolution digital sensor suited for underwater usage, communicating over I²C. Depending on the specific model, it can measure pressure to depths up to ~300 m with millimeter-scale resolution. Like the temperature sensor, it requires a 3.3 V I²C level converter to interface with the Raspberry Pi Zero 2 W. The sensor will be mounted such that its pressure port has direct exposure to the ambient fluid while the submersible remains sealed.

Implementation/Testing

Full integration is pending with the arrival of a level converter. The team plans to mount the sensor in a low-flow location on the exterior. Then, SDA/SCL wil be wired through the level converter to the Raspberry Pi's I²C bus. The pressure sensor Python library will then be used to read raw pressure values. From there, pressure data can be converted to depth using standard fluid relationships.

Next Steps

The team shall install and verify the I²C level converter, develop and test depth readout software, and perform accuracy and repeatability tests in controlled depths (in a pool or tank).

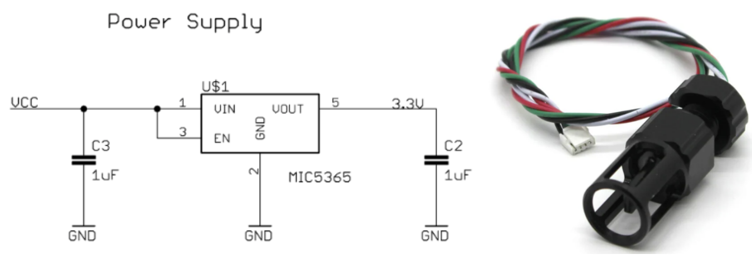


Figure X: Voltage regulation circuit providing 3.3 v output using the MIC5365 regulator

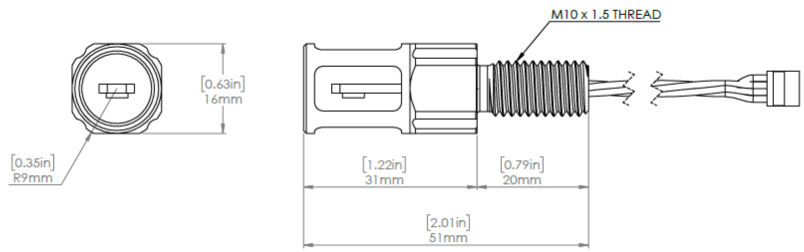


Figure X: Mechanical dimensions of the temperature sensor housing and threaded body

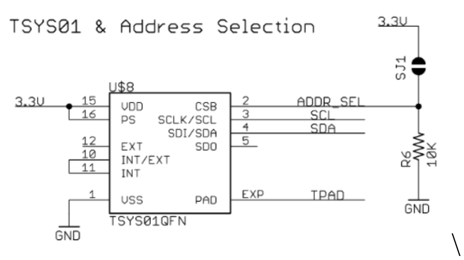


Figure X: TSYS01 temperature sensor pinout and i2c address selection configuration

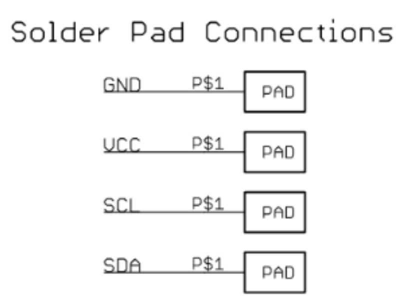


Figure X: Solder pad connections for ground power SCL and SDA signals

Resources

GitHub: https://github.com/cornell-cup/Xplorer/blob/main/motorControl/xplorer_thruster.py

PoE System Diagram: https://whimsical.com/xplorer-power-9oapA3h6ajEPVemrxYj5o9

Thruster PWM code:
https://github.com/cornell-cup/Xplorer/blob/main/motorControl/xplorer_thruster.py

Thruster set:
https://www.amazon.com/ApisQueen-Underwater-Thruster-Brushless-Propeller/dp/B0D9PZ47N5?th=1

CS

Dashboard

Purpose

We had to restructure the current XRP dashboard to be fit for the Xplorer. The minimum requirements for the dashboard was being able to see a live camera feed, using the temperature sensors, and controlling the thrusters.

Design

Tech Stack

- React
- Typescript
- Python

There are 3 main parts for the dashboard: the frontend, the Raspberry Pi backend, and the XRP backend.

The Raspberry Pi backend sends both the camera feed and the temperature readings to the frontend over Ethernet socket connection.

The frontend shows each of these as a sensor component. The frontend also has a separate controls application, which sends thruster information to the XRP.

Implementation

Repository: https://github.com/anishbh/xplorer-dashboard

Frontend

Branch: main

1. Clone the repository.
   ```
   > git clone -b pi-server --single-branch
   https://github.com/anishbh/xplorer-dashboard.git
   ```
   ```
   > cd xplorer_dashboard/pi_server
   ```
2. Install dependencies.
   ```
   > cd dashboard
   > npm install
   ```
3. Make sure the correct `PI_IP` of the Raspberry Pi is set in `dashboard/src/utils/constants.ts`
4. Run the server and view in the browser on http://localhost:3000. Might have issues on Chrome so use Safari / another browser.
   ```
   > npm run dev
   ```
5. The camera feed should already be visible. Press "Start" to see the temperature feed.
6. For thruster control, make sure the XRP serial port is correct in motor_bridge.py by running
   ```
   ls /dev/tty.usbmodem*
   ```
7. Connect thruster controls in a separate terminal and run
   ```
   python3 motor_bridge.py
   ```

Raspberry Pi

The Raspberry Pi will be connected to the laptop over Ethernet.
Branch: pi_server

1. On your laptop, `ssh pi@raspberrypi.local`. The password is usually minibot. (You can also directly just use the Pi's terminal if you want).
2. Make sure to take note of the IP of the Pi by doing `hostname -I`. This IP will need to be updated in the frontend to view the camera and sensor feed.
3. Clone the pi-server branch and cd into pi_server:
```
> git clone -b pi-server --single-branch
https://github.com/anishbh/xplorer-dashboard.git
> cd xplorer_dashboard/pi_server
```
4. Run the install.sh script.
```
> ./install.sh
```
5. Activate the venv, then run the server. This will run the sensor server and camera server.
```
> source venv/bin/activate
> python3 run_server.py
```

Next Steps

Currently, the Raspberry Pi only sends dummy temperature sensor info. This will need to be changed to actually retrieve the temperature sensor values.

Systems

System Requirements

What are the system requirements?

Timelines

Put overall timelines here

Interfaces

Describe the overall system

Mechanical

What are the mechanical-mechanical integrations?

Electrical

CS

Mechanical-Electrical

Mechanical-CS

Electrical-CS

Notes for final submission:

You must submit all project documentations in a combined document, including MechE, ECE, CS, and Systems documentation. Meaning, for Minibot, there must be a single document, and for C1C0 there must be a single document. These must be converted into Microsoft Word documents, sent in a zipped folder titled "CCRT [Fall/Spring] [Year]" (for example, CCRT Fall 2010). This zipped file needs to be uploaded to Box. The Box links can be found here: https://it.cornell.edu/box. Then, the file in Box needs to be shared with David Schneider's Cornell email.