Aryan Sharma

CYSF LOGBOOK 2023-2024

September to October 2023

This month, I spent my time researching prominent current issues and determining a potential project idea. I started by brainstorming general areas of science that interested me. Last year, I completed a project related to environmental science, which I really enjoyed doing since it combined my passion for environmentalism and STEM. This year, I wanted to continue to do a project related to solving a large environmental crisis.

After determining the overarching theme of my project, I started to look into specific environmental issues, such as climate change, plastic pollution, and resource depletion. I also looked into the major causes behind these issues, what is currently being done to tackle them, and what more needs to be completed. In addition to my research online, I tried to identify a problem that I observed in my everyday life. Over the course of the summer, I noticed that the pond behind my house had fewer and fewer Canadian geese coming to it. In 2022, hundreds came, nested, and used my pond area as a migration area. It was an amazing and lovely thing to look at everyday. This year, I am noticing very little of this. And so I decided to look into the issue further. I initially thought that the geese found the water to be too cold, or too many people were walking near them. But upon close inspection, I could see plastic, lots of plastic, in the pond. They were in there for so long that many began to break apart. Some plastics were big, and some were smaller than a pellet. The water consisted of bottle caps, plastic bags, and much more. This initial exposure sparked my interest in environmental stewardship, fueling my passion to do something to solve the problems that I was seeing in my community. Through further research, I discovered that the minuscule plastic particles, often indistinguishable from marine objects like organic matter, pose a significant threat to aquatic life, as they can be mistakenly ingested, leading to severe complications and fatalities in marine life. On a larger scale, microplastics can even end up on our plates, since microplastics stay in a marine animal's body and can end up in seafood.

This event really resonated with me, and I finally decided to narrow down my project idea to the detection and filtration of microplastics. I want to create an autonomous, cost-effective, and effective underwater robot that is able to filter plastic from bodies of water, such as the pond behind my house.

October to November 2023

After figuring out my project idea for this science fair, I began to research the problem of microplastic pollution in our world. I wanted to gain a scope of how big the problem really is and what its far reaching impacts on the environment were. During my research I found the following statistics and information:

Aryan Sharma

What does plastic pollution look like in our world?

- Plastics that are frequently used include nylon, polypropylene, polystyrene, and polyurethane. The physical, chemical, and biological effects of the environment cause these plastics to break down gradually.
- It may take more than a century for these plastics to decompose entirely
- Since the 1950s, with annual global plastic production has increased dramatically every year
- Merely 9% of the total plastic waste in the world gets recycled each year, while the remaining 12% has been burned. Most of the remaining 79% have entered aquatic environments
- Plastics can have detrimental effects on all organisms in the food chain once they enter the body of a single organism. A phenomenon called biological magnification means that a predator will ingest more plastic than it will its prey
- Plastic pollution has negatively impacted more than 400 animal species, including fish, turtles, zooplankton, shellfish, seabirds, and mammals

What is a microplastic, and where do they come from?

- Plastic particles smaller than five millimetres are referred to as microplastics.
    - Two types:
        - <u>Primary microplastics</u> are produced plastic particles that are initially smaller than 5 mm; they are typically utilized in abrasives, manufacturing plastic powders, and personal hygiene items like microbeads. Typically, primary microplastics are pellets or microbeads.
        - <u>Secondary microplastics</u> are found as fragments or as synthetic textile fibres. Due to the widespread usage of plastics across all fields, microplastics can be discovered in sediment from rivers and the ocean as well as in lakes, rivers, and the ocean itself.
- Microplastics have entered the food chain through these routes and have since been detected in a variety of forms that humans may consume, such as drinking water, aerosolized dust particles, fish, and other marine mammals. Littering, sewage, and unlawful dumping all contribute to the large amount of plastic waste that ends up in the rivers and oceans.
- Every piece of plastic that enters the ecosystem, regardless of source, eventually breaks down into microplastics!
- **Statistics related to microplastics**
    - Our oceans currently contain up to 51 trillion microplastic particles
    - It is estimated that at least 50,000 microplastic particles are consumed by humans globally through food and water

Aryan Sharma

Having gained a comprehensive understanding of what the microplastic problem is, in the next month I want to start looking into what is currently being done around the world to address this pressing problem.

November to December 2023

I started looking into existing solutions this month. My overarching goal was to identify the gaps in existing research that I could address with my project to ensure that it is a novel contribution. I found the following information:

1. Raman Spectroscopy:
    a. Uses laser light to analyze the molecules in a sample.
    b. Can distinguish between different types of plastics, such as polyethylene or polypropylene, based on their molecular patterns
2. Infrared Spectroscopy
    a. Fourier-transform infrared spectroscopy (FTIR) is a sophisticated method used to analyze the vibrations of molecules in a sample. In this technique, the sample is exposed to infrared radiation emitted by a light source, leading the molecules to absorb specific wavelengths based on their vibrational modes. Within an FTIR system, an interferometer divides the infrared light into two beams—one traverses the sample, and the other goes through a reference material. Upon recombining, these beams generate an interference pattern, transformed into an infrared spectrum through Fourier transformation. This resulting spectrum displays absorption peaks that correspond to the chemical bonds and molecular structures present in the sample.

While both spectroscopy methods are great, they do have some drawbacks:

1. Spectroscopy methods are lab-based and are extremely expensive (costs over $50,000). They also require specialized equipment and trained personnel.
    a. Makes it less accessible to researchers or organizations with limited resources, just like me!
2. Spectroscopy requires somebody to collect samples in advance and transport them to a laboratory for analysis
    a. Not time effective
    b. Potential for sample contamination
3. Spectroscopy does not provide real-time insights into the extent of microplastic pollution in a
given environment

Other current solutions:

- Filtering and Cleanups

    - Big drawback: a filter by itself can't tell microplastics apart from other materials like organic matter. And since it is open to anything, when undesired objects, such as organic materials and non-plastic debris, can become stuck in the filters
    - Volunteer human cleanups are ineffective and time-consuming
- <u>Government Bans</u>
    - Many world leaders have also proposed bans on single-use plastics in their countries
    - Drawback: Not everyone follows bans

After identifying current solutions and the problems that need to be addressed despite them, I created the following objectives to guide my project and measure its success:

1. Cost-effectiveness
2. Real-Time Use
    a. The ROV must be able to efficiently and precisely detect and filter microplastics in water bodies in real-time, which means when it detects a microplastic, it filters it immediately, which directly addresses the urgent issue of microplastic pollution.
3. Autonomous Control

December 2023 to January 2024

At the beginning of my ROV journey, I had to start understanding how to actually make my robot. so I started with the most simple part which was the casing. the casing is the outer layer of the robot which holds all the electronics and Motors is what I initially thought:

a. Shell
    i. At the beginning of my ROV journey, I explored pre-made underwater robots, thinking I could modify them to my own needs. My main requirement was finding one with an SDK (software development kit) so I could code it to do what I wanted. SDK is a software where you can add your own code to the robot. However, underwater drones with SDKs were rare and expensive. I found one called the BlueEye Pioneer from Blue Eye Robotics, but it was very pricey, starting at $20,000. The cheapest alternative was Chasing Drones underwater robots, but its $500 “Dory” model had a faulty camera and no SDK. Faced with these limitations, I decided to build my ROV to meet my specific needs and allow for tailored solutions to any questions that might arise. In constructing the ROV, I researched submersibles and underwater aerodynamics. To start off my project, I had to figure out the encasing, or the shell of the robot. Initially, I considered 3D printing for the casing, and in this process I educated myself on 3D modeling platforms like TinkerCAD and Fusion360 solely through YouTube videos. However, after doing some tests on waterproofing the prints, I discovered that 3D printing filament isn't waterproof due to microscopic pores in the printed material.

To solve this, I opted for a simple yet effective solution: a regular airtight food storage container. This uncomplicated approach proved surprisingly effective underwater, showcasing good aerodynamics, and a very good watertight seal. Unlike 3D prints, plastic doesn't have pores, preventing water infiltration.

I tested this process by submerging the plastic container underwater. I moved it around, up and down, side to side, and no water came through it.

After this, I started to understand different types of coding languages and code boards in our world today. I had to figure out one of these coding boards and languages because it would be the main source of all my power and knowledge for my robot. I tested on many different types of code boards including Raspberry pies, and Arduinos. At the time raspberry pies were actually very very expensive and they were around $150. This was actually just due to inflation costs and high demand with low supply. however at the time I wasn't going to take the risk and I found that Arduino was generally cheaper and also had pins for Motors that would just help me with the easier approach to my project.

i. After realizing that off-the-shelf SDK Bots weren't suitable for my project, and determining the design of my shell, I went into the first step of electronics—the code board and programming language. Keeping accessibility in mind, I aimed to choose a coding platform that was user-friendly and open to modifications. After lots of research, I narrowed down my options to Arduino or a Raspberry Pi. Raspberry Pi showed impressive computing power, functioning as a mini-computer with an operating system. This made it ideal for projects involving extensive data processing or complex algorithms. Additionally, its built-in Ethernet, Wi-Fi, and Bluetooth capabilities impressed me. On the other hand, Arduino included analog inputs, making it better suited for projects requiring sensor readings or motor control. Another key consideration was cost—Arduino boards were significantly more budget-friendly compared to Raspberry Pi devices. Arduino boards were available for around $15, while Raspberry Pi devices were priced at approximately $150 during that period. While Raspberry Pi was undoubtedly superior in performance, the high cost was a deterrent at the time. My decision leaned towards Arduino due to its affordability and pins suited for my application. However, I acknowledged that Raspberry Pi's capabilities would be beneficial for more complex tasks in the future. I planned to reevaluate my choice when Raspberry Pi prices became more reasonable, considering the potential advancements and improvements it could offer to my project. After buying my arduino, I had to learn the language of the board, which is C++. I educated myself in under 10 hours by just youtube videos! I also had a class in school related to robotics so I learned there as well.

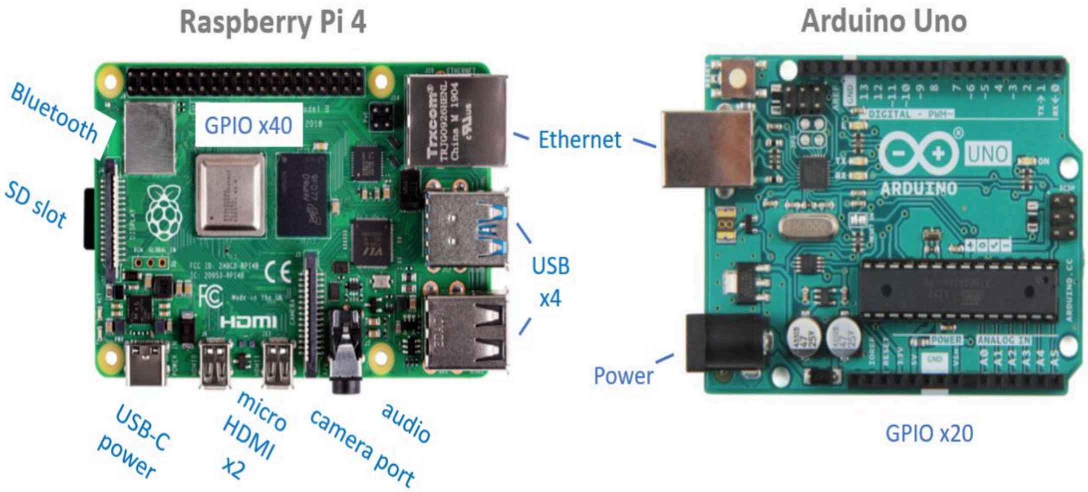


Figure 1: Comparison of Arduino Uno to Raspberry Pi 4. While Raspberry Pi is considerably better, the Arduino Uno possessed specific capabilities tailored to what I needed in my project

And finally I had to figure out electronics and motors for my project, essentially what I was going to be using to be building my ROV:

i. After planning and building both my code board and casing for the robot, I needed to first understand the fundamental electronic principles of robotics. For this purpose, I opted to purchase the Elegoo kit, an affordable Arduino kit encompassing essential components like wiring, batteries, LEDs, resistors, and more. The entire kit cost around $50, and I used and learnt each component. Online tutorials from various YouTube channels played a crucial role in enhancing my understanding of electronic principles and addressing issues like short circuits. I also learned with tutorials and a school class focused on robotics and engineering, providing a solid foundation to enhance my underwater robot. After I fully mastered the Arduino, I had to move on to the components of my robot. To start, I needed motors. I found reasonably priced motors from a company called Apis Queen, costing only $32 for both clockwise and counterclockwise motors. The choice of using both types of motors was influenced by the Law of Angular Momentum. If both motors were clockwise for example, then the robot would never go forward and constantly be turning. After acquiring the motors, I attached them to both sides of the robot and started wiring the Arduino to respond to controls. In the initial stages of the project, I opted for a controller, utilizing two joysticks (later I made autonomous control - I made a simple controller to check for any problems). When both joysticks were pushed forward, the machine moved forward. If the left joystick was pushed forward, the machine turned left, and when the right joystick was pushed forward, the machine turned right. Similarly, both joysticks pushed backward made the machine move backward. To enable this control, I required a lengthy cable(as I would be going into deep depths of water), leading me to a 100-ft ribbon cable online. This cable consisted of 4 wires which helped me connect them to my breadboard. I drilled a hole into the middle of the container for this step. For the battery, I chose the URGENEX 1000 mAh Li-po RC lithium battery. This battery stood out for its durability and offered a runtime of 10 hours under my circumstances. I also added an on/off switch to my robot to make the turning-on process much easier than having to take the container lid off every time I wanted to turn on/off the robot. I attached the on/off switch to the bottom left of the lid of the container. Now that I had all my main materials, I had to code the bot and attach specific Arduino components to make my robot work. Of course, there are breadboards, jumper wires, etc. But the most important component for all of my motor control was the “H-Bridge” or an L293D component (Fun Fact: It is called an “H-Bridge” because it looks like an “H” on a circuit diagram). An H-bridge is a crucial electronic circuit for motor control and controls a load's direction and speed. By manipulating the state of its switches, the H-bridge makes bidirectional current flow through the load, enabling control over the motor's rotation. It creates voltage control by adjusting the timing of transistor switching, which is important for regulating motor speed and power. Additionally, the H-bridge supports dynamic braking to swiftly halt the motor's rotation, taking away kinetic energy. Pulse Width Modulation (PWM) further enhances speed control by rapidly switching transistors on and off. All of the code for these components will be explained later. Finally, to put everything together, I bought many different adhesives. I started with gorilla glue’s silicone sealant. While this did work well, a silicone sealant is very flexible and prone to being cut. So I double-layered the glue, the first layer being silicone sealant, and the second layer being JB’s Marine Epoxy Weld. This epoxy was extremely strong and water-resistant.

Finally, after creating my base ROV, I created a DIY method!

Method

1. Charge your batteries so they are ready for use later.
2. H-bridge (pins numbered counterclockwise from the top left)
    a. Pin 1 to Arduino pin 11
    b. Pin 2 to Arduino pin 12
    c. Pin 3 to the right motor negative wire
    d. Pin 4 to ground
    e. Pin 5 to ground
    f. Pin 6 to the right motor positive wire
    g. Pin 7 to Arduino pin 9

   h. Pin 8 to 7.4 V from the battery
   i. Pin 9 to Arduino pin 10
   j. Pin 10 to Arduino pin 8
   k. Pin 11 to left motor positive wire
   l. Pin 12 to ground
   m. Pin 13 to ground
   n. Pin 14 to left motor negative wire
   o. Pin 15 to Arduino pin 7
   p. Pin 16 to 5 V from Arduino
3. Left joystick
   a. L/R+ to 5 V from Arduino
   b. L/R to Arduino analog pin A0
   c. GND to ground
4. Right joystick
   a. L/R+ to 5 V from Arduino
   b. L/R to Arduino analog pin A1
   c. GND to ground
5. Battery
   a. Double-check all of your wiring before connecting the battery.
   b. Connect the positive wire to the Arduino's Vin pin.
   c. Connect the negative wire to the Arduino's GND pin.
   d. Recommended: connect the switch in series with the battery's positive wire so you can easily turn your ROV on and off without unplugging wires.
   e. Make sure all breadboard ground buses are connected to Arduino GND so the entire circuit has a common ground. (Do not connect the left and right side power buses on the main breadboard to each other. This will create a short circuit between the 5 V supply from the Arduino and the 7.4 V supply from the lithium battery)
6. Download the code (the code is provided and explained at the end of this tutorial)
   a. Make any required changes to the code as needed, such as adding code to control additional motors or sensors. In my case, early in the project, the motors weren’t responding to the signals because not enough voltage was going in them). I fixed this by mapping the code (making a large value to a smaller value, so the program has something easier to work with)
   b. Upload the code to your Arduino.
   c. Test your motor controls. It can be hard to see which way the propellers are spinning, but you can put your hand near them to see which way they are blowing air.

   d. If needed, make any required corrections to your circuit or code. For example, if a motor is spinning backwards, you can reverse the two control wires connected to the Arduino or switch the two assigned pins in the code.

7. Once you have your circuit working, you are ready to start mounting parts to your waterproof container.
   a. You should carefully plan out where everything will go before you start drilling holes in your container.
   b. After drilling holes, remove any sharp burs from their edges, since these could cut the insulation on wires that you push through the holes.
8. After you have drilled all the required holes for mounting hardware, passing wires through, etc., follow the instructions for your glue (I used double layer - silicone sealant then epoxy) to seal around the edges of the holes. Make sure you wait for the glue to dry completely before you continue.
9. After the silicone sealant has dried, it is time to conduct leak testing to make sure your container is completely watertight. Fill a container of water (sink, bathtub, etc.) large enough to completely submerge your ROV. Remove the Arduino, breadboard, and battery from the container. Put the lid on the container, making sure that it is sealed tightly. Slowly submerge your ROV in the water. Watch carefully for any air bubbles coming out of the container, or water leaking into the container, especially around any holes that you drilled and the edges of the lid. Hold your ROV underwater and gently move it around and rotate it as you continue to watch for leaks. Remove your ROV from the water. Use a towel to completely dry off the exterior. Check to see if any water has accumulated inside the container. If you find any leaks, you will need to patch them (by adding more silicone sealant) and repeat your testing until all leaks have been sealed. If your container leaks around the lid, you may need to use a different container (to be extremely safe - I even sealed the container shut)
10. If all works, then you have a working ROV!

January to February 2024

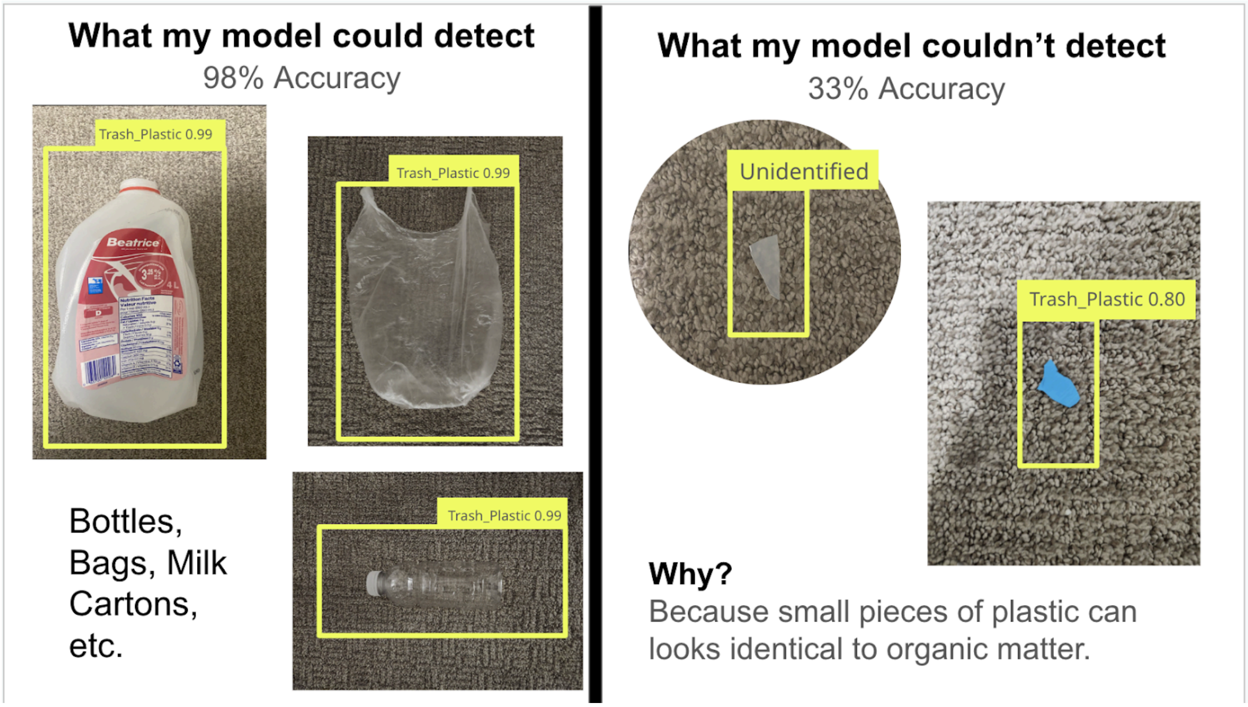
This month, I focused solely on sensor detection. It was definitely a very hard part of my project.

First Attempt: Utilizing Machine Learning
I started off with machine learning. I began with machine learning and image processing, mainly because it was a very hot topic at the time and I was curious to know more about it. But before I explain my results, we must first understand the basics.

Aryan Sharma

How did I make my model?

This program was created in Google Colaboratory, a free platform that allows Python to run without the need for any setup. The model in this study is trained to identify plastics using a deep learning algorithm. I used an ESP32 camera to look at images in real time. To identify large plastics, I used open-source models and then used the neural network, YOLOv5-S, to analyze the images quickly. The top-performing model was YOLOv5-S, which maintains near-real-time speed while operating at a Mean Average Precision (mAP) of 0.851 and an F1-Score of 0.89. The images were formatted into the Darknet and YOLOv5 PyTorch required sizes, 416x416. When my model is uncertain as to what the object is, it says “unidentified”, and when the model has a 80% or higher chance that an image is plastic, it labels the image as “trash_plastic”. At the start, I used an open source model and created a model to identify bigger pieces of plastic, and then I moved to smaller and smaller plastic pieces. However, it just got less accurate by utilizing smaller and smaller pieces. In the end, my first model, which could detect large plastic objects, like bags and cans, was at an accuracy of a whopping 98%! But as I went into smaller and smaller pieces of plastic, the percentage went lower. This was because a spec of plastic can look identical to a spec of wood, for example. If a human can’t decipher the difference, the machine is unlikely to. My model to identify plastic relied on looking at their texture and color, as in some cases these characteristics are different from organic matter. However, this was proven not to be a very reliable method, and the final accuracy for this was 32%. Nonetheless, I still learned valuable aspects of computer vision from this area and can largely help me in future projects.



What my model could detect

98% Accuracy

Bottles, Bags, Milk Cartons, etc.

What my model couldn’t detect

33% Accuracy

Why?

Because small pieces of plastic can looks identical to organic matter.

Figure 2: This figure shows how my model could easily detect plastic bottles, bags, etc. But it could not readily detect smaller pieces of plastic. The AI thinks the blue object is plastic, but only with 80% certainty (borderline until it is deemed as “unidentified”), which is right, but again it is not certain enough. The final microplastic machine learning model accuracy was 33%. After machine learning did not work out so well, I moved onto looking at how plastic is affected by external sources:

2nd Attempt at Sensing Microplastics: UV Sensing

The UV sensing technique involved two components: a TCS3200 Arduino Sensor and a 365-nanometer UV flashlight. The TCS3200 Arduino Sensor has a built in photodetector in it, making it perfect for what I need in my project. I bought a 365 nm UV flashlight because that is the range where plastics are affected the most.

After this, I attached my UV sensors to my robot:

Step 1: Attach light intensity sensor to the top left corner of the robot

Step 2: Attach the UV flashlight right beside the light intensity sensor so both are in the same range

I began by testing how UV affected the plastics. I positioned the UV light and the photodetector together so they were both in the same target range. I then put a piece of plastic approximately 7 centimeters away from the robot. The full range exceeds 15 centimeters, but I found that the best results come if the plastic is 7 cm away from the robot. I could only test LDPE AND PET Plastics, more testing for other plastics will be undergone soon.

LDPE Plastic (not under UV light)

R	86
G	98
B	145

LDPE Plastic (under UV light)

R	5
G	251

B	230

PET Plastic (not under UV light)

R	96
G	143
B	122

PET Plastic (under UV light)

R	6
G	251
B	232

As you can see from these results, all of the 2 plastics without UV light shined onto them have drastically different values. But when UV light is shined on them, they all generally have the same average value: R-5, B-251, and C-230. I also tested this on various pieces of organic matter, and the results were very different!

With that, my month was finished and I had a detection sensor and a working ROV!

February to March 2024

Now that I had both detection and the actual ROV done, I could move on to filtering and autonomous control.

Let's first look at filtering:

a. Filter no 1: Filter in Bot
    i. Originally, I wanted to put the filter inside the bot for a compact and easy-to-move design. It seemed good initially, but it didn't work out. The container was too small, and having the filter in the middle was risky—water

could easily leak in, and replacing parts was tough. For future 3D-printed designs, I might try it again since it could make things more maneuverable. But for now, my current design, which I'll explain soon, still moves well enough to get the job done quickly and effectively. (Show pictures of filter)

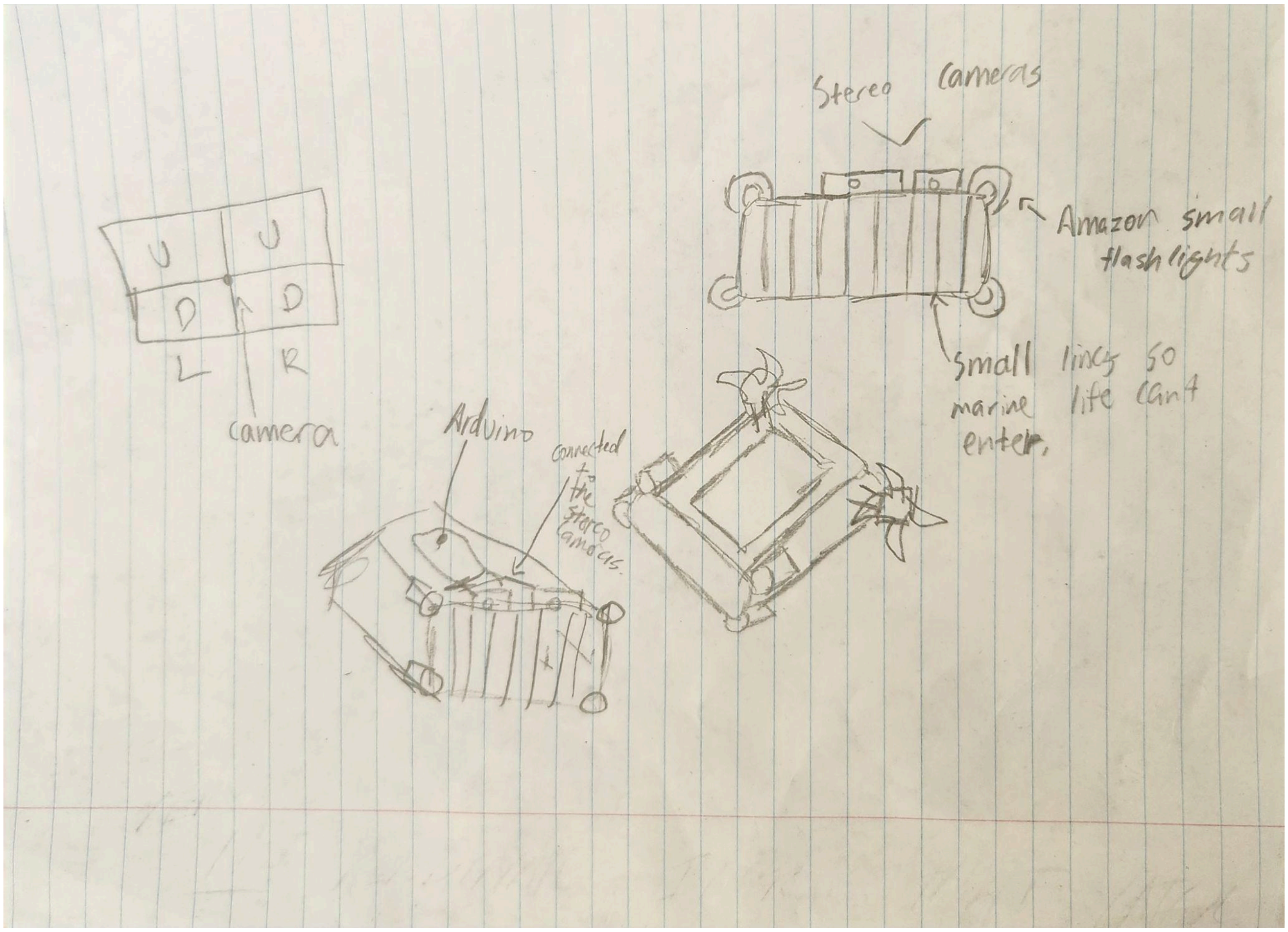


Figure 1: Sketches of the first filter prototype.

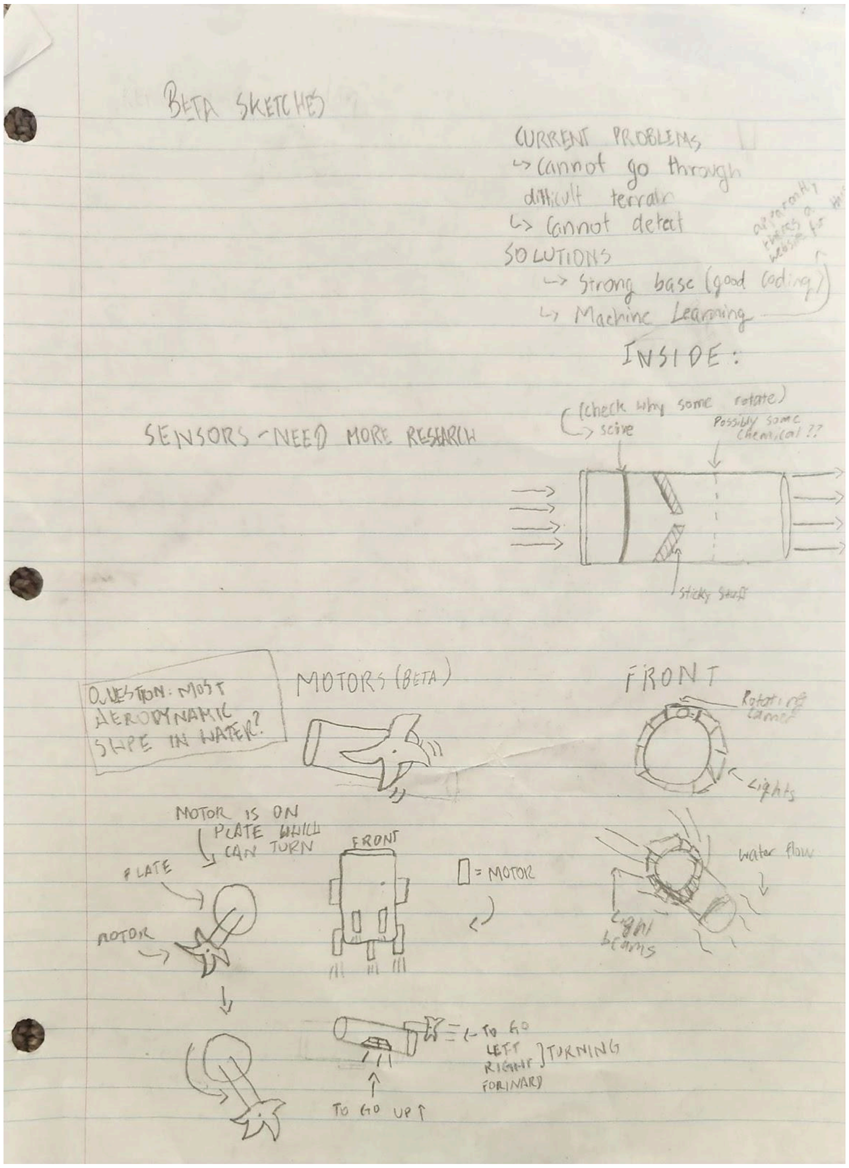


Figure 2: More sketches of the initial prototype

b. Filter No. 2: Tube design

i. After giving up on the idea of having a filter inside the bot, I opted for an external filter, though still close to the bot. My initial approach was straightforward—a tube with a mesh piece in the middle to let water through while catching plastic. Although the concept was right, using a small mesh made it prone to clogging easily, disrupting the robot's operation due to an excess load after being in the water for some time. In an attempt to refine the design, I explored electrostatic filters and even rotating filters with the idea of generating energy. However, upon revisiting my goals of simplicity and accessibility, these designs didn't align well. They were either too complex to construct or didn't fit into the overall objectives.

While these features might find a place in future iterations, for now, they would add unnecessary complexity and pose potential safety risks. There were inherent limitations, like poor underwater aerodynamics, associated with rotating filters that made them less suitable for my current project.

d.

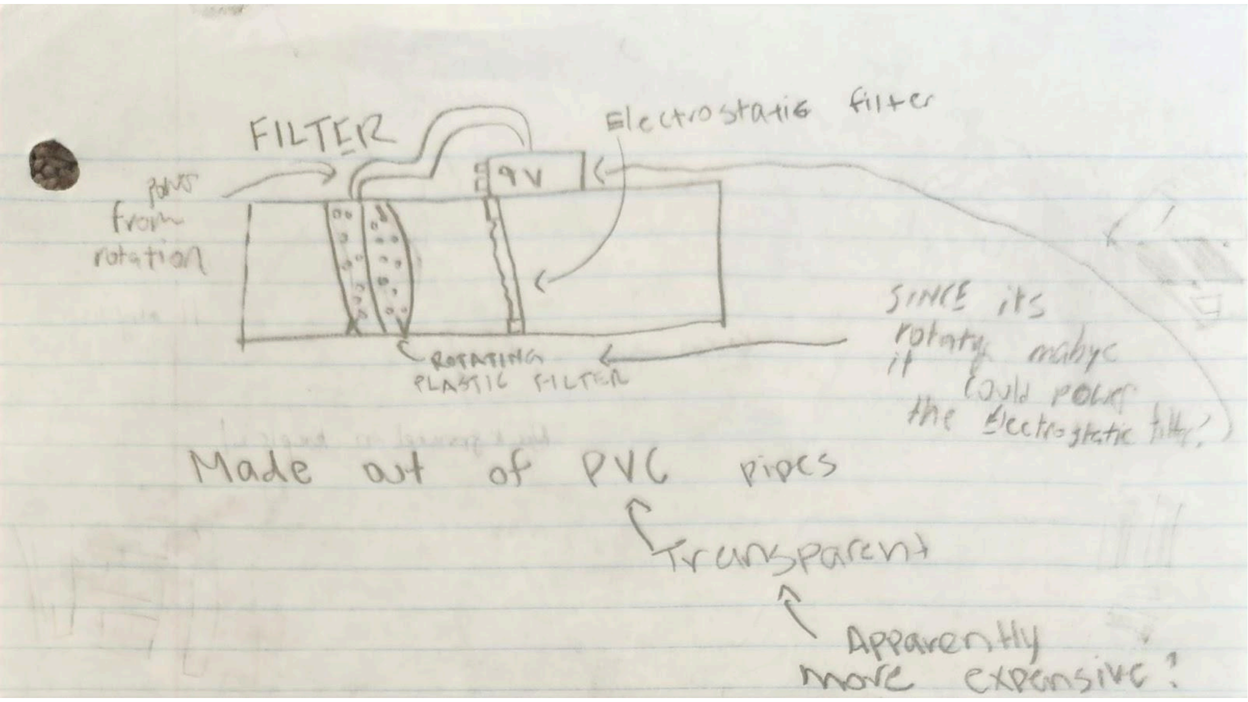


Figure 3: Electrostatic filter design

Aryan Sharma

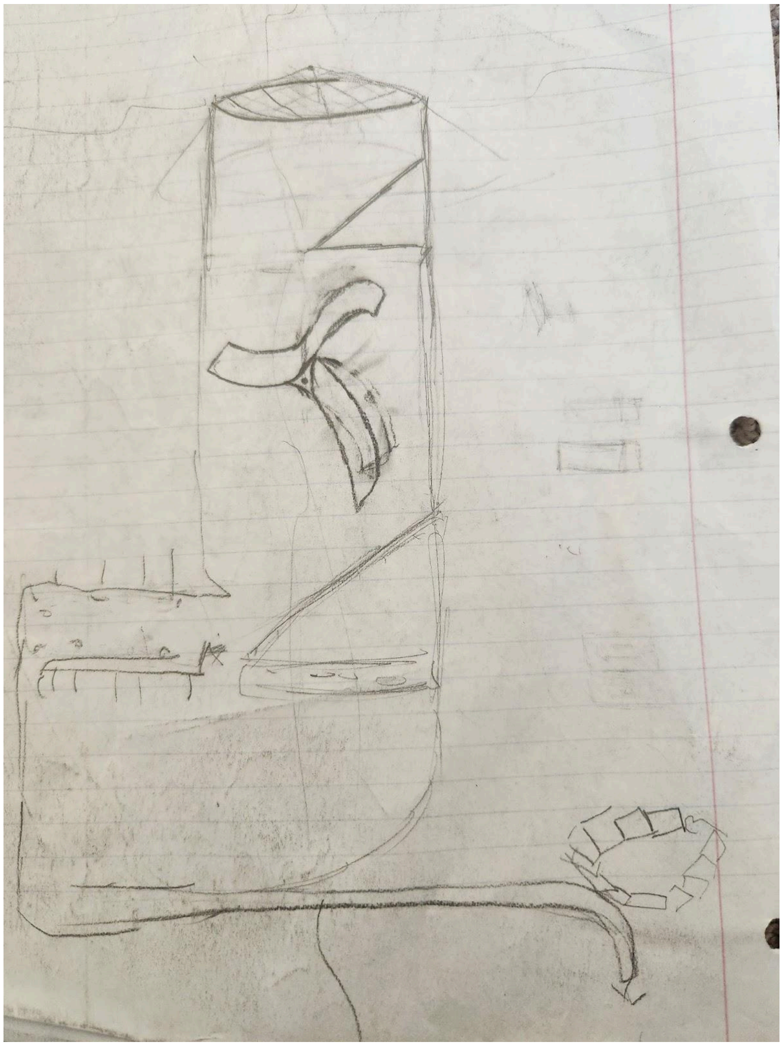


Figure 4: Another sketch of the rotating filter design

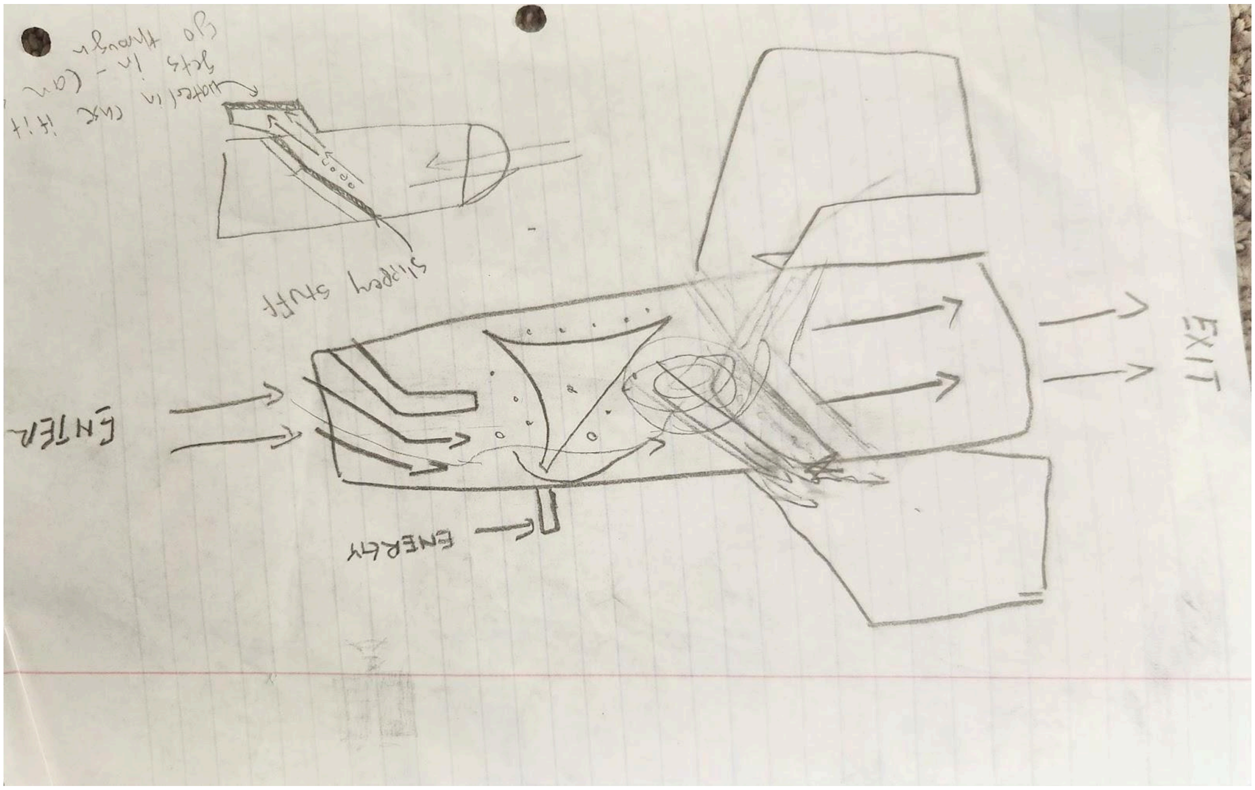


Figure 5: Rotating filter design with “wings” for microplastic collection

c. Filter No. 3: Final “gill” Design

i. After numerous attempts to perfect the filter, I circled back to my core objectives: simplicity and accessibility. I chose to retain a tube-like design but significantly enhance its effectiveness. The previous design was prone to clogging, a problem I aimed to solve while maintaining aerodynamic efficiency, requiring a compact and straightforward design. In the final iteration, I incorporated elements like mesh and 3D prints but with a novel approach. The design featured three pathways of increasing length, allowing plastics to travel through each one. This ingenious arrangement prevented clogging and, upon closer inspection, resembled a fish's gill—an inspiration drawn from their efficient aerodynamics. This redesigned filter not only excelled in aerodynamics but also efficiently collected microplastics. To optimize the mesh's effectiveness, I sized it to 1 mm, aligning with the typical size of many microplastics. Recognizing potential ecological concerns, particularly regarding organic matter, I sealed the filter entirely. Unwanted materials, if the filter were left open, could enter and be released only when the filter drained. To address this, I sealed the filter and incorporated a servomechanism to open it only when a plastic was detected by the main body machine. This not only prevented the entry of undesired materials but also maintained the filter's aerodynamic efficiency. Strategically placing the filter on the left side of the sensors addressed limitations in the detection area. Although I

> couldn't place a filter on the other side due to the robot's design, I envision overcoming this constraint in the future with a rotating sensor. A 360-degree detection angle would undoubtedly enhance efficiency, speed, and effectiveness. For the code implementation, I devised a simple algorithm: if plastics were detected by the machine, the servo would open, prompting the machine to move right and forward for two seconds. This brief maneuver ensured that microplastics were collected effectively, allowing the machine to seamlessly resume its regular tasks

Method for final "gill" design

Materials:

- 6 custom 3D prints
- 10 by 7 centimeter 1mm flexible mesh
- Flex Seal's Flex Glue

After numerous attempts to perfect the filter, I circled back to my core objectives: simplicity and accessibility. I chose to retain a tube-like design but significantly enhance its effectiveness. The previous design was prone to clogging, a problem I aimed to solve while maintaining aerodynamic efficiency, requiring a compact and straightforward design. In the final iteration, I incorporated elements like mesh and 3D prints but with a novel approach. The design featured three pathways of increasing length, allowing plastics to travel through each one. This ingenious arrangement prevented clogging and, upon closer inspection, resembled a fish's gill—an inspiration drawn from their efficient aerodynamics. This redesigned filter not only excelled in aerodynamics but also efficiently collected microplastics. To optimize the mesh's effectiveness, I sized it to 1 mm, aligning with the typical size of many microplastics. Recognizing potential ecological concerns, particularly regarding organic matter, I sealed the filter entirely. Unwanted materials, if the filter were left open, could enter and be released only when the filter drained. To address this, I sealed the filter and incorporated a servomechanism to open it only when a plastic was detected by the main body machine. This not only prevented the entry of undesired materials but also maintained the filter's aerodynamic efficiency. Strategically placing the filter on the left side of the sensors addressed limitations in the detection area. Although I couldn't place a filter on the other side due to the robot's design, I envision overcoming this constraint in the future with a rotating sensor. A 360-degree detection angle would undoubtedly enhance efficiency, speed, and effectiveness. For the code implementation, I devised a simple algorithm: if plastics were detected by the machine, the servo would open, prompting the machine to move right and forward for two seconds. This brief maneuver ensured that microplastics were collected effectively, allowing the machine to seamlessly resume its regular tasks.

Now that I had a filter working, I could finally move onto autonomous control!:

Aryan Sharma

After fully completing my robot with motor control, I then wanted to move on to autonomous control, as it was one of my objectives in this project. To do this, I went for a method of distance mapping. I tested three main sensors, the first two being ultrasonic and the final one being lidar. With autonomous control, a human doesn't have to be constantly monitoring the robot, allowing for more microplastics to be collected with fewer humans needed. In the end, the lidar sensor worked the best, and you'll see my process, code, and instructions further on.

First Prototype: Ultrasonic Distance Sensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. I used the basic Arduino HC-SR04 Ultrasonic Distance Sensor. The HC-SR04 is an affordable ($8.74 CAD) and easy-to-use distance measuring sensor which has a range from 2cm to 400cm (about an inch to 13 feet). The sensor is composed of two ultrasonic transducers. One is a transmitter which outputs ultrasonic sound pulses and the other is a receiver which listens for reflected waves. It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle in its path, it will bounce back to the module. Considering the travel time and the speed of the sound, you can calculate the distance.

The problem: As you can see the ultrasonic distance sensor is great because it is cheap and easy to use. However, since it uses sound to measure distance, putting it in the container will just mean the sound will bounce around in the container, not giving a reliable measurement. Sound waves are not strong enough to go through the container.

Second Prototype: Waterproof Ultrasonic Distance Sensor

I wanted to use an ultrasonic distance sensor that was waterproof as the sound waves still worked outside of the container. When I began researching waterproof ultrasonic distance sensors there were actually not very many. The ultrasonic sensors that were available had a rating called IP67 which means that they are working underwater for only a specific amount of time. In the case of what I saw, they only worked for around 30 minutes. Of course, I wanted to make these ultrasonic distance sensors work for as long as possible because I wouldn't be taking my machine off and on the water again over and over. The ones that were rated IP68 (IP68 means the sensor can be submerged underwater for any amount of time) range from 200-300 dollars which of course isn't in my budget and it is not worth it.

This led me to a sensor called the A02YYUW. The A02YYUW is an ultrasonic distance sensor that bears some resemblance to the HC-SR04 in that it has the same receiver and transmitter. The device is enclosed in a rubber-like material and is fully waterproof.

I still did get one of the sensors to actually just check if it really worked and to my surprise, it didn't even turn on! Maybe because I didn't get it from the right dealer? But after looking at a YouTube video online, the sensor could work but it actually does not work underwater. The

Aryan Sharma

YouTube video actually said that it can only work above water if it's reading signals from for example the shore to the bottom, which of course is not what I'm planning for.

Since the A02YYUW (Waterproof Ultrasonic Distance Sensor) is just a rubber-encased version of the HC-SR04(normal ultrasonic distance sensor), then the code and wiring are the same. Let's look at the wiring diagrams:

The sensor has 4 pins. VCC and GND go to 5V and GND pins on the Arduino, and the Trig and Echo go to any digital Arduino pin. Using the Trig pin we send the ultrasound wave from the transmitter, and with the Echo pin we listen for the reflected signal.

Third Prototype (Final Prototype): TF Luna Lidar Sensor

In design 3, I decided to take a different route in distance-sensing and autonomous driving control which was lidar sensors. Essentially, a lidar sensor is using lasers instead of sound waves to detect distance, this was helpful because the laser actually made it outside the container. The difference between laser and sound was that sound wasn't so powerful and so it bounced around in the container but laser is powerful so it can go easily through the plastic. Now of course the laser isn't too strong that it will penetrate the plastic but it is just strong enough to go through it. TF-Luna is a single-point ranging Lidar based on the TOF principle. With its unique optical and electrical design, it adopts an 850nm infrared light source to achieve stable, accurate, and highly sensitive distance measurements. After seeing this work I decided to figure out the values and what code I was going to be putting in the robot for this distance to actually work. My final result was I wanted the bot to move if an object was 20 cm or less from the bot. This way I could get a signal early on so the bot doesn't break under any circumstances.

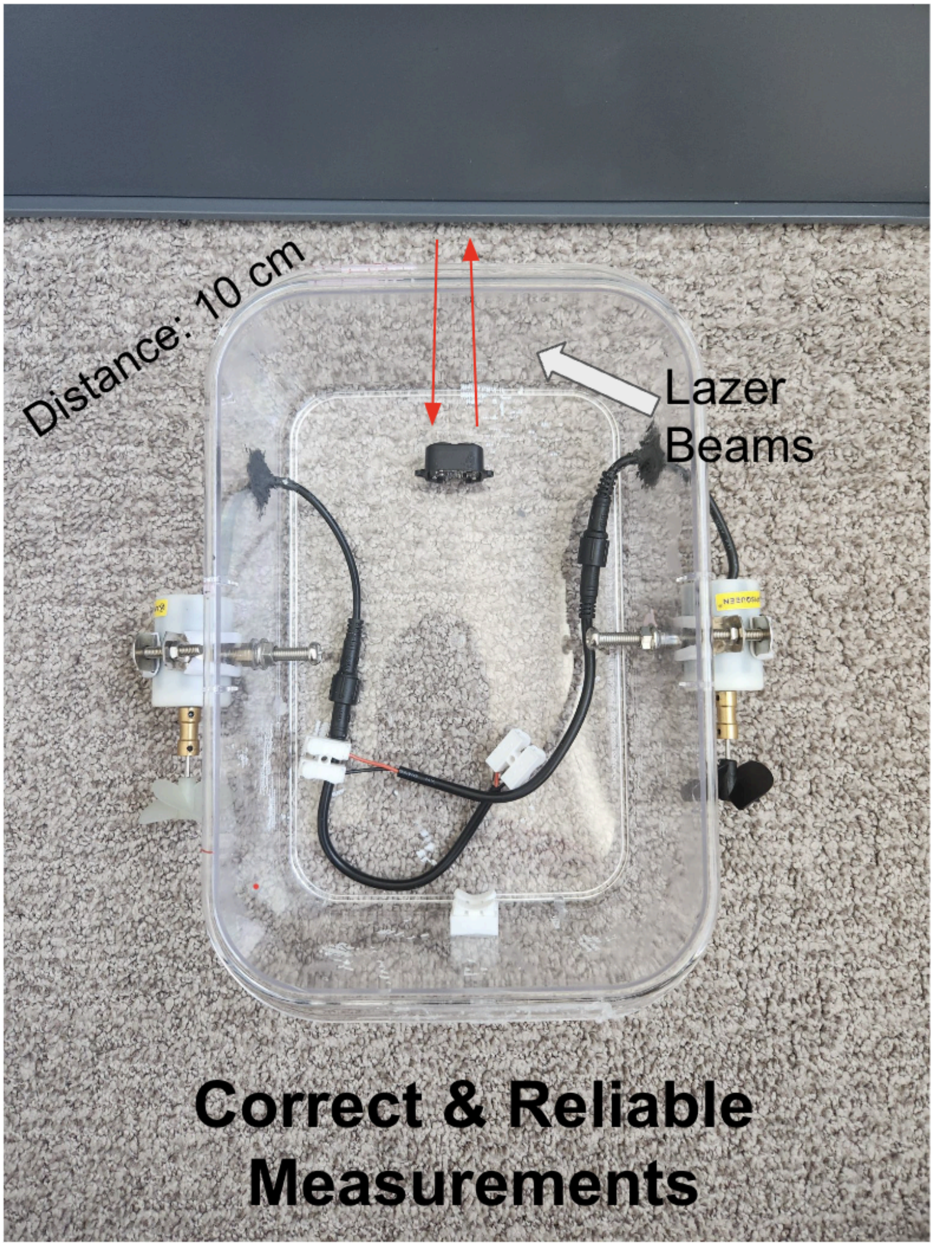
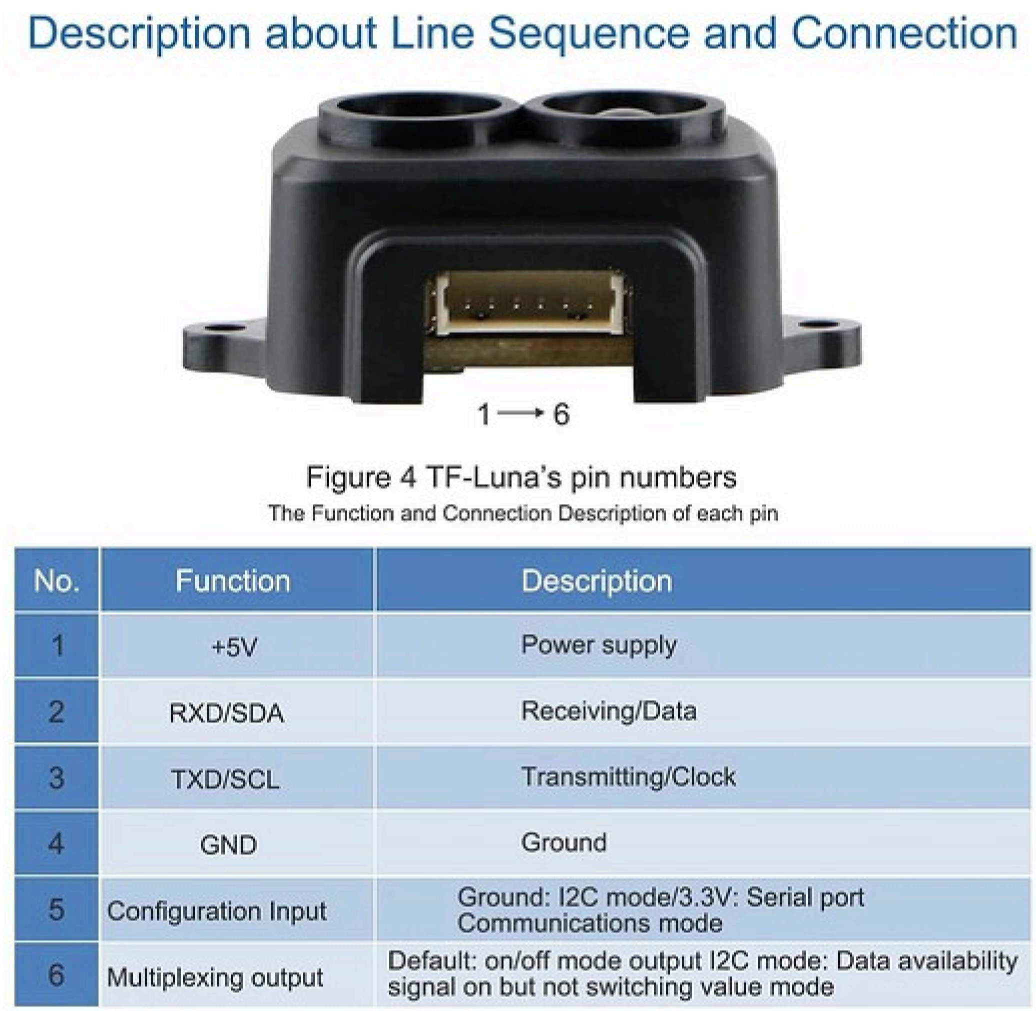


Figure 6: Picture showing how the TF-Luna's laser went through the plastic to get an accurate and reliable measurement for distance sensing

Pinout and Wiring Diagram for TF-Luna Lidar Sensor Module:



Description about Line Sequence and Connection

Figure 4 TF-Luna's pin numbers

The Function and Connection Description of each pin

No.	Function	Description
1	+5V	Power supply
2	RXD/SDA	Receiving/Data
3	TXD/SCL	Transmitting/Clock
4	GND	Ground
5	Configuration Input	Ground: I2C mode/3.3V: Serial port Communications mode
6	Multiplexing output	Default: on/off mode output I2C mode: Data availability signal on but not switching value mode

Figure 7: Pinout for the TF-Luna Lidar Distance Sensor

In my Arduino, I actually only use 4 of these pins, as pins 5 and 6 are not necessary. Pins 1 and 4 are necessary(power for the sensor), and pins 2 and 3 are connected to the Arduino RXD and TXD pins. A fun fact about these pins is that RXD and TXD actually don't allow for a serial monitor, so you can't see what values the sensor is getting. But I easily mitigated this problem by turning the Arduino LED on when an object is closer than 20 cm and turn off when an object is farther than 20cm. I also tried this with my motors, and it worked!

Aryan Sharma

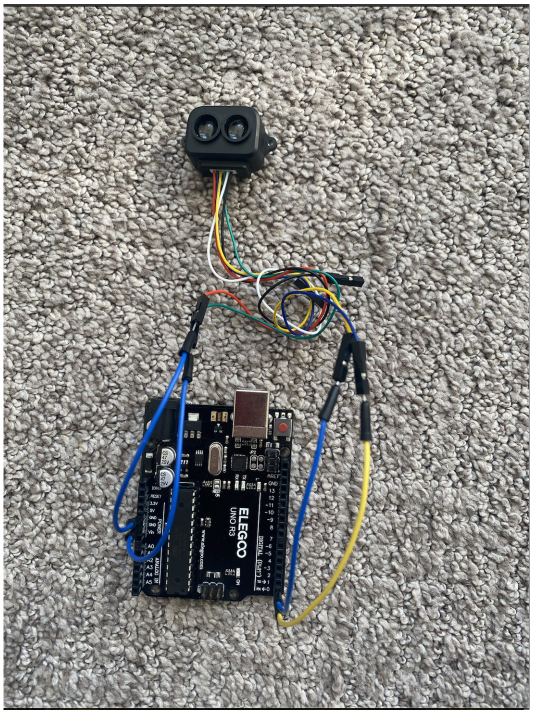


Figure 8: Circuit for the TF-Luna Lidar Distance Sensor (The sensor is not on the TinkerCAD platform)

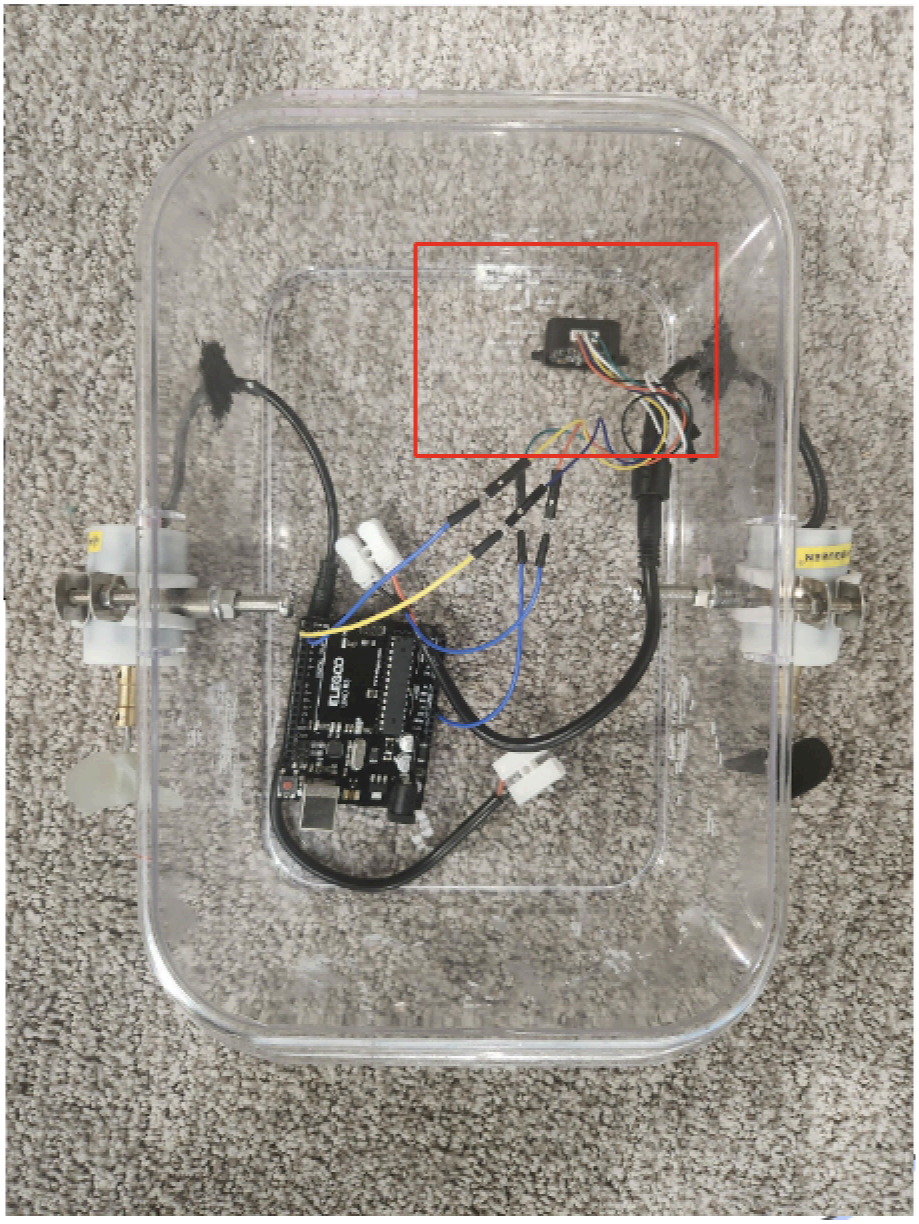


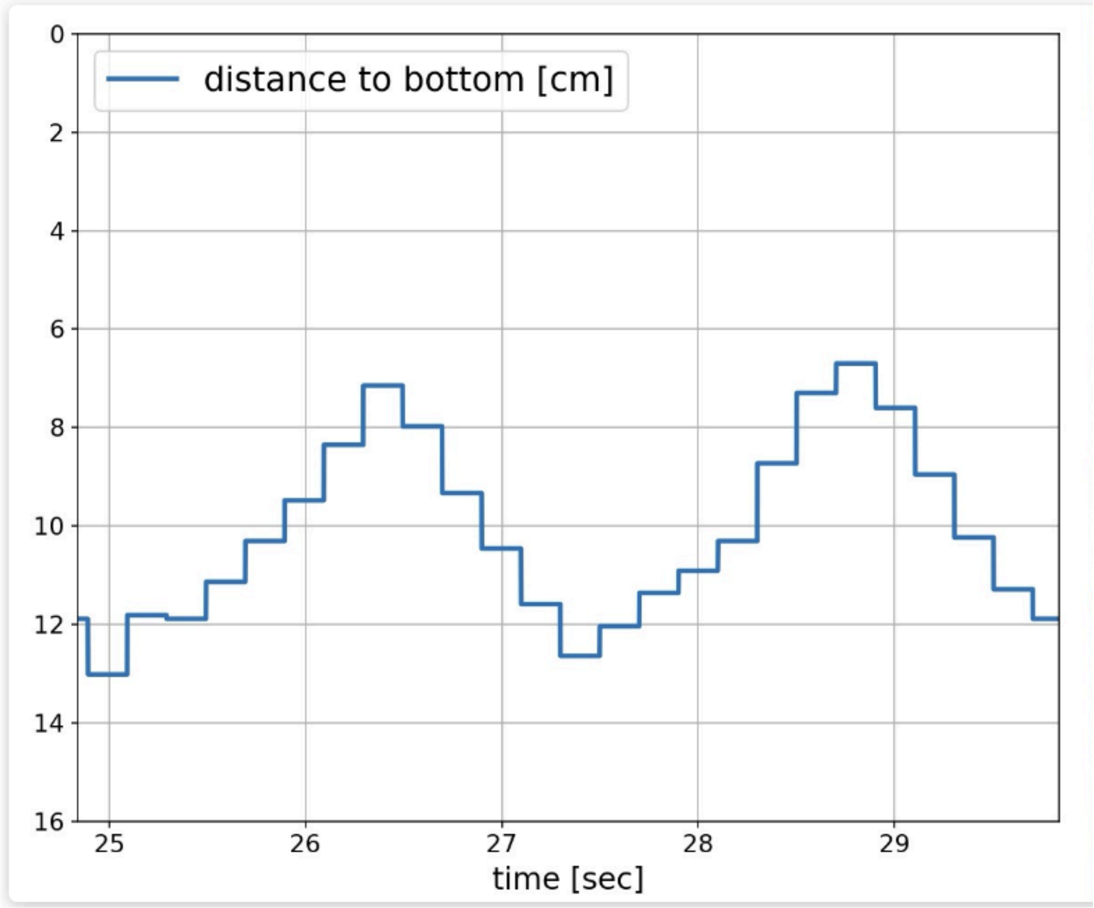
Figure 9: TF-Luna Sensor attached to the ROV (highlighted in red box)

Now I have all my compartments to my robot finished. I started analyzing everything I made:

Results from the TF-Luna Lidar Sensor

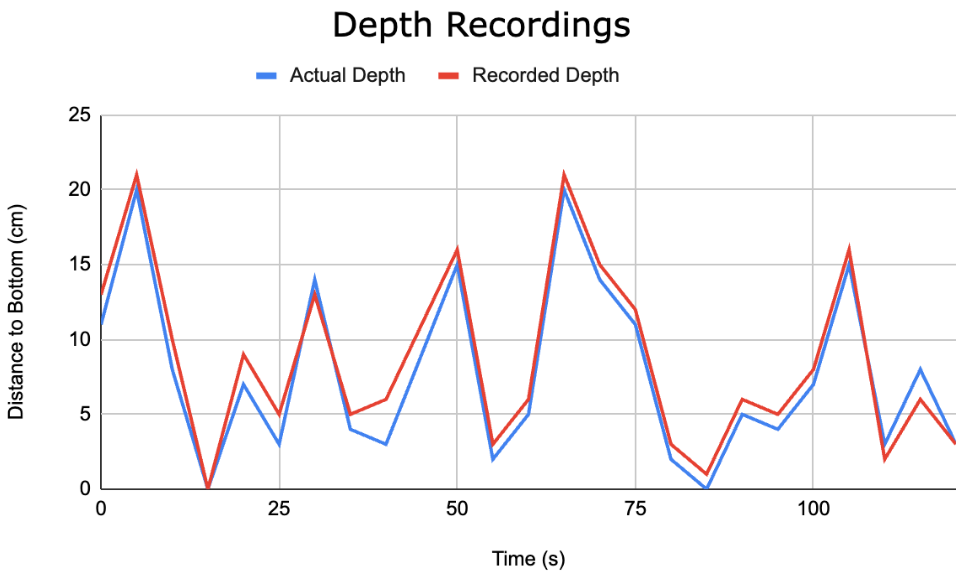
I tested the sensor on my bathtub filled with water since none of the ponds or lakes in Calgary had yet thawed at the time. However, I still tried to imitate a pond-like water environment in my bathtub even in this scenario. I added dirt, sticks, stones, etc. Of course, I did not add any animals because I had access to none. The hull was pressed against the water's surface so that no air was in between plastic and water. I lowered a robot into the water and started moving it up and down to try different distances.

Aryan Sharma



Graph 1: Testing depth accuracy in the TF-Luna Lidar Distance Sensor. I went for about 30 seconds until I stopped.

Beside me was a ruler, so I could see the distances it should be going to. The data looked accurate! I made sure the data was fully accurate by making a graph of the depths I put it into and the depths the sensor picked up.



Graph 2: Relationship between actual distance vs. sensed distance to the bottom of the bathtub

In this graph, you can see the relationship between the actual distance and the recorded distance to the bottom of the bathtub. The actual distance was pre planned, and so I went up and down according to the planned sequence in my bathtub. I wanted to see not only if the results were similar to what I planned, but also if the sensor can handle quick and high-pressure environments, where there are obstacles everywhere. You can see the sensor has performed very well in these tests, as the red and blue values are very close to each other. In a real-life situation, I am very sure this robot will easily be able to navigate the waters autonomously.

Then I tested the robot to see how well it responds. I coded the sensor so that if it detects an object that is 20 centimeters or less away, then it turns left for as much time as it gets a value that is higher than 20 cm. I tested this by putting it into a tub and making it go forward until one of the walls was 20 cm away. The boat turned left and since the wall beside it was very close it kept on turning for 180 degrees. This proved my code worked out well!

I could only show you the results for the TF-Luna sensor because the ultrasonic distance sensor didn't even work underwater! As I saw in my observations, normal non-waterproof ultrasonic distance sensors can't produce strong enough sound waves to go through objects, and waterproof ones don't even work. This was a downfall of my project, but in the end it opened up new doors for me to experiment with new sensors!

After analyzing the TF luna sensor, I began analyzing my final UV detection:

Aryan Sharma

Results from the Detection Sensors

The microplastic Sensing consisted of two modules. The first is machine learning and the second is UV sensing. In conclusion, UV sensing was far more effective than machine learning. In my tests of machine learning, I made a model that could detect bigger pieces of plastic such as milk cartons cans or plastic bottles. However, I tried to take the challenge of actually taking specks of these pieces and trying to use them to detect if it was microplastic or not. In my research and testing, I realized that machine learning wasn't a very feasible option as it couldn't tell the difference when they got smaller. The mean accuracy for bigger pieces of plastic was 98%, however, for microplastics, it was only a whopping 33%. After this, I went on to do UV sensing, and in the end, I could detect all six types of plastic: PS plastic, HDPE plastic, LDPE plastic, PET plastic, PVC plastic, and finally PP plastic. In comparison with machine learning, not only could the UV sensors detect far more plastics than machine learning (machine learning could only detect plastic bags, cans, and milk cartons), but they also have a very high accuracy of detecting them in comparison to organic matter.

How does UV light even work?

UV light induces fluorescence in plastics by exciting electrons within their molecular structures, producing characteristic glows or colors. Photodetectors then detect this fluorescence, allowing for the identification of different types of plastics based on their unique responses to UV illumination.

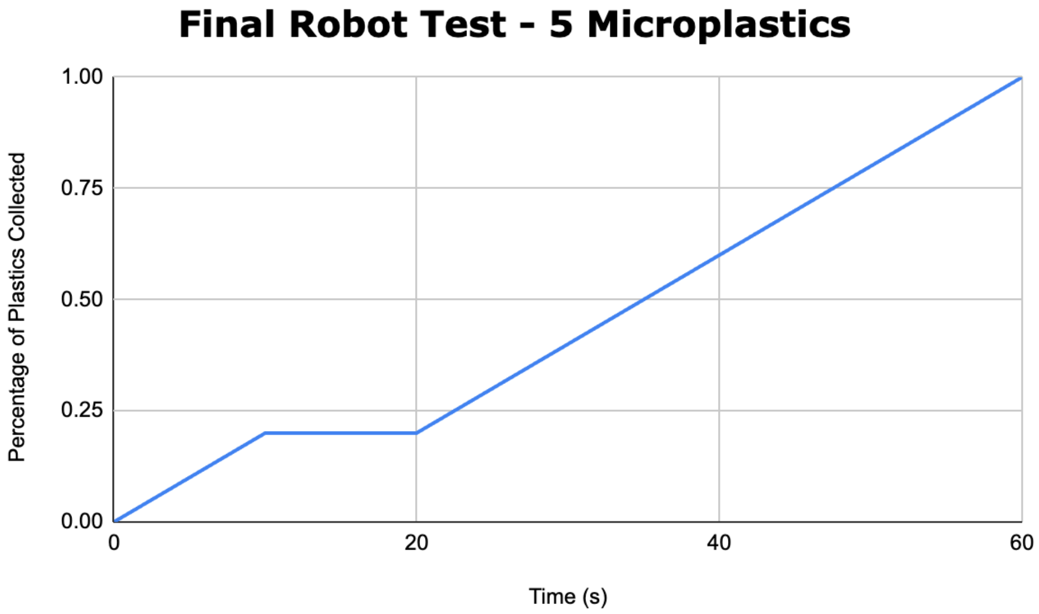
Finally, I looked at how my whole bot collected microplastics:

Results from the Final Robot

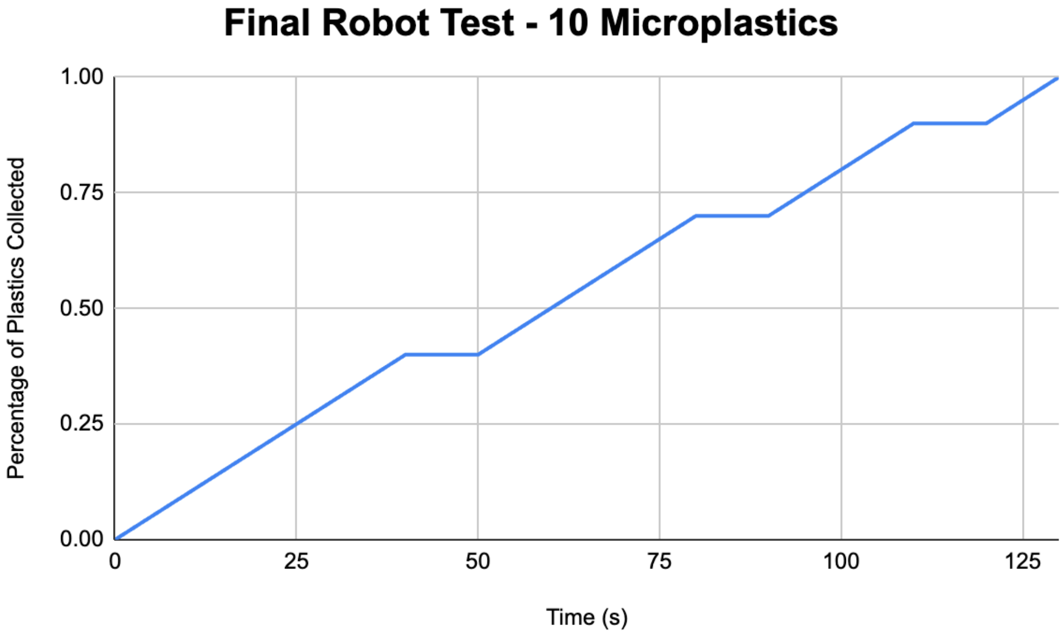
After testing everything by itself, I decided to put it all together and test my whole robot. The best way to do this is to see how well and fast it can collect microplastics. For this test, I filled my bathtub up with water (no ponds in Calgary had thawed yet), and began to add many items that simulated a pond or lake like environment. I did not add any animals as I did not have access to any. However, when the water bodies in Calgary thaw out I will get my robot there. After replicating my setting as best as I could, I began adding microplastics in the tub. For my first test, I put 5 types of microplastic in a straight line. I wanted to see if my robot could detect and collect all 5 types of plastic. All of these plastics were clear. I turned my robot on and put it into the tub, and it went on to detect and collect all 5 of the microplastics, in only 60 seconds! I also made sure to include many other pieces of organic matter, such as algae and sticks, so I could see if the robot would detect those as plastics. It didn’t, and only chose the right materials to filter! This was a huge success in my project. And so I went on to test further. I then tested on 10 and 15 microplastics in the water. And while the robot could differentiate between the plastic as and the non-plastics, it did take more time to find them, as I expected.

Aryan Sharma

5 microplastics:

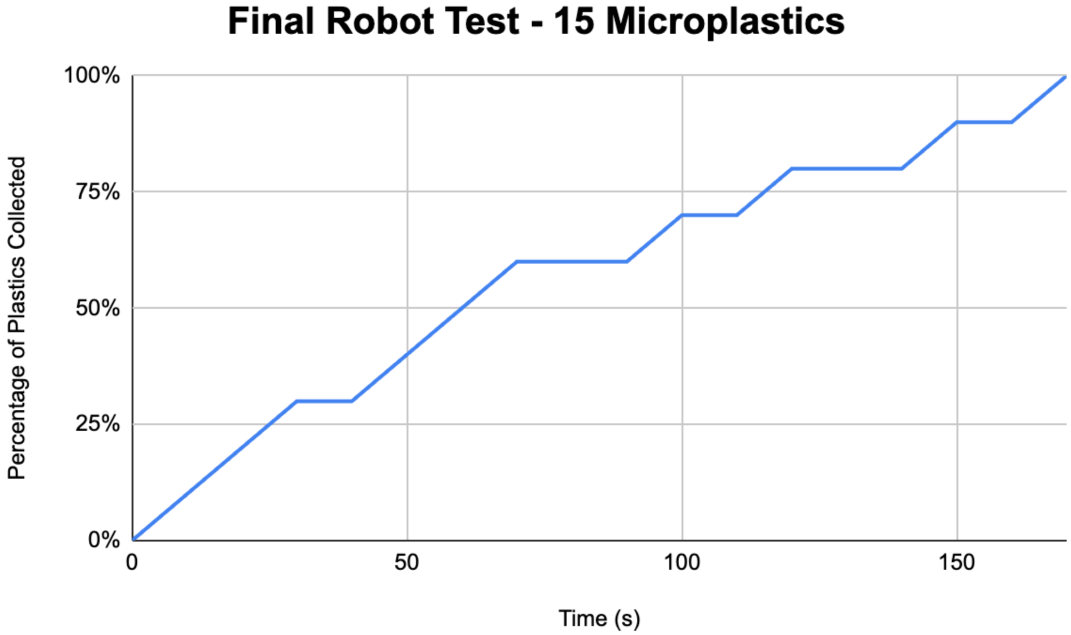


10 microplastics:



Aryan Sharma

15 microplastics:



My current robot is very helpful because it can do things other filtering robots can't. There are many great aspects to other robots in the world, however, I add real-time use, autonomous control, and low cost. I believe that with even more tweaking to my robot, I can even sell it!

March - March 15:

At this point I had finished my whole robot. It was a success! At this point, I started creating my trihold and other necessities to transport my bot and also create you know trifle things like that. I also created my conclusions and citations and acknowledgments.

Concluding Statement:

The purpose of this project was to make a low cost and autonomous underwater robot that can detect and collect microplastic in water bodies in real-time. After significant testing and variations throughout each 4 modules (Detection, Robot, Autonomous Control, Filtering), I have successfully created an underwater robot (ROV) that is $484 CAD, fully autonomous, and can detect all 6 types of plastic (LDPE, HDPE, PVC, PP, PE, PET) in real time.

Practical Applications:

Essentially my robot is a practical application. As I've proven to you by solving my three objectives: my first one being autonomous control, second one being real time efficiency, and the third one being cost-effectiveness. Now with this robot, I plan to go and navigate Calgary's Waters once they have thought out and try to take microplastics out of them. I have already tested my robot in my bathtub environment where I simulated a pond/lake like environment. The robot is equipped with sensors that analyze and detect these microplastics in real time. Now, I will go into real environments where there are animals and I can actually test out the full capabilities of my robot. This way I can expand to industrializing my robot and bringing it around the world to solve the microplastic pollution problem around the world. My robot can contribute to environmental cleanups and can provide data for future researchers. I hope my robot can be improved on and ultimately help save the world's marine microplastic pollution problem!

Future Improvements

1. Making my Robot more aerodynamic

   Although the robot that I made in this project was generally aerodynamic, it wasn't the best. In rough waters, the ROV could get entangled and broken. To solve this issue in the future, I am planning to create 3D models that are aerodynamically efficient and are generally better as a design for my ROV.

2. Even more range to detect

   In my robot, I could detect microplastics only on the top left corner of my robot. This range is 15 cm in height. Although it could detect plastics from that area, in the future, I will plan to make sure that the whole front side of my robot is filled with sensors so I can detect it from every way in that scenario. This will make it way more effective because it can detect far more plastics and make my robot more efficient.

3. Soldering

   I used a breadboard to connect all the wires. Breadboarding is an effective method because small mistakes can be tolerated. However in my future design I will solder my circuit. Soldering essentially makes the circuit so that the wires are actually fully connected to the sensors and they cannot be taken apart. Of course, this will only happen when I'm fully sure how the circuit will work. I already have a lot of knowledge on soldering and I can execute this part pretty easily. This way my ROV could bounce in very rough water and still not be affected.