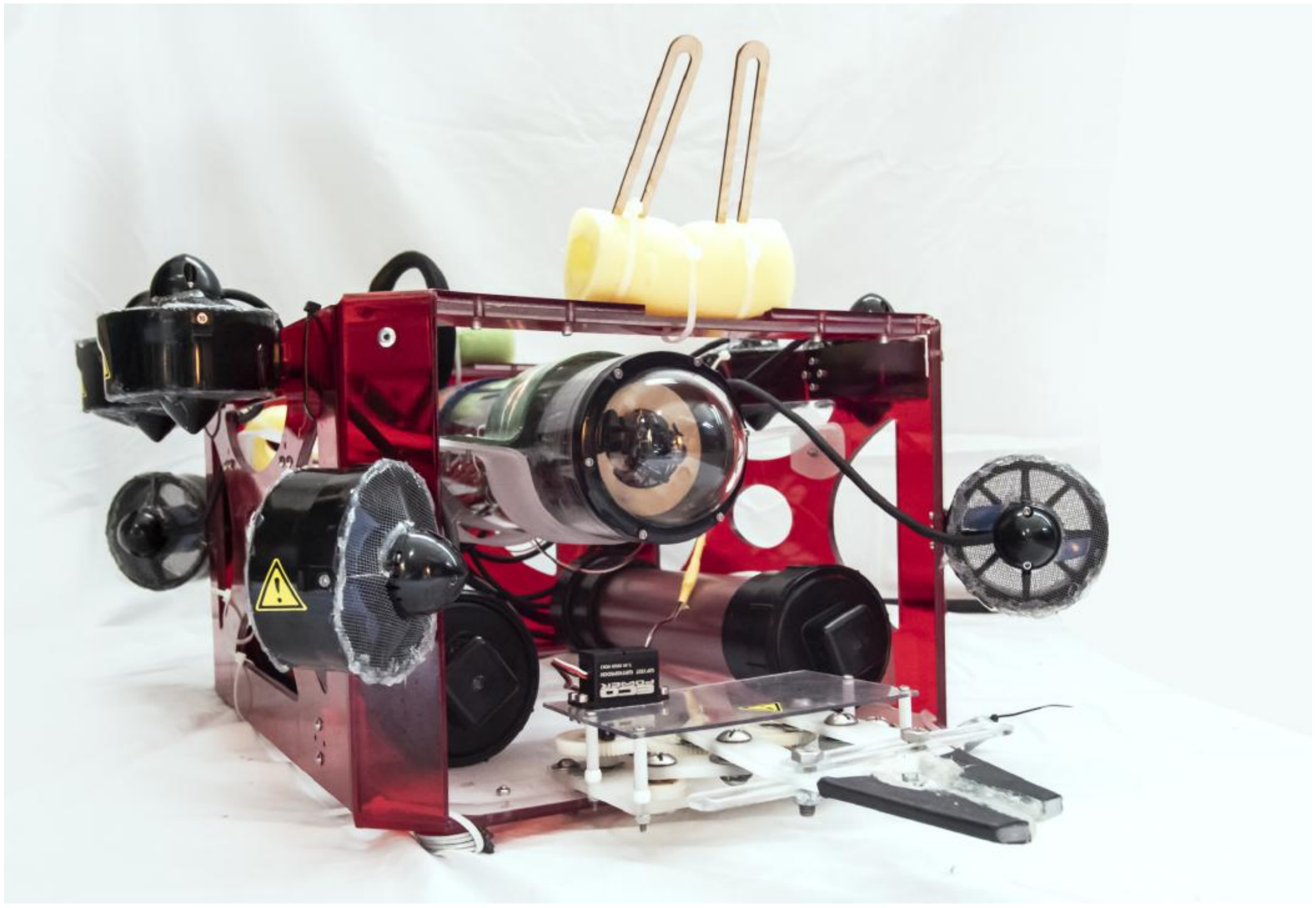
BLUESHIFT ROBOTICS



FLOAT-E

Seattle Academy, Seattle, Washington, United States

Top:
(New) Meghan Jimenez, Team Mentor
(Returning, '22) Kavi Dey, CEO, Programming Lead, Electrical Lead, Co-Pilot
(New, '24) Haedyn Darling Hill, Pilot, Electrical Engineer

Middle:
(New, '23) Jenna Li, Software Engineer, Visual Systems Engineer, Tether Manager
(New, '22) Emily Myint, Chief Safety Officer, Chief Financial Officer, Electrical Engineer
(Returning, '23) Della Smith, Manipulation Systems Engineer, Mechanical and Design Lead

Bottom:
(Returning, '22) Nalu Farrell, Chief Compliance Officer, Mechanical Engineer

ABSTRACT

Blueshift Robotics is a student-led team focused on the development of robots that are optimized to improve the state of our oceans through the MATE ROV Competition. In our 5 years as a team, we have developed a remote-operated vehicle (ROV) with safety, efficiency, and flair in mind. Our ROV, FLOAT-E, is designed to be cost-effective and efficient while facing the challenges associated with deep-water exploration.

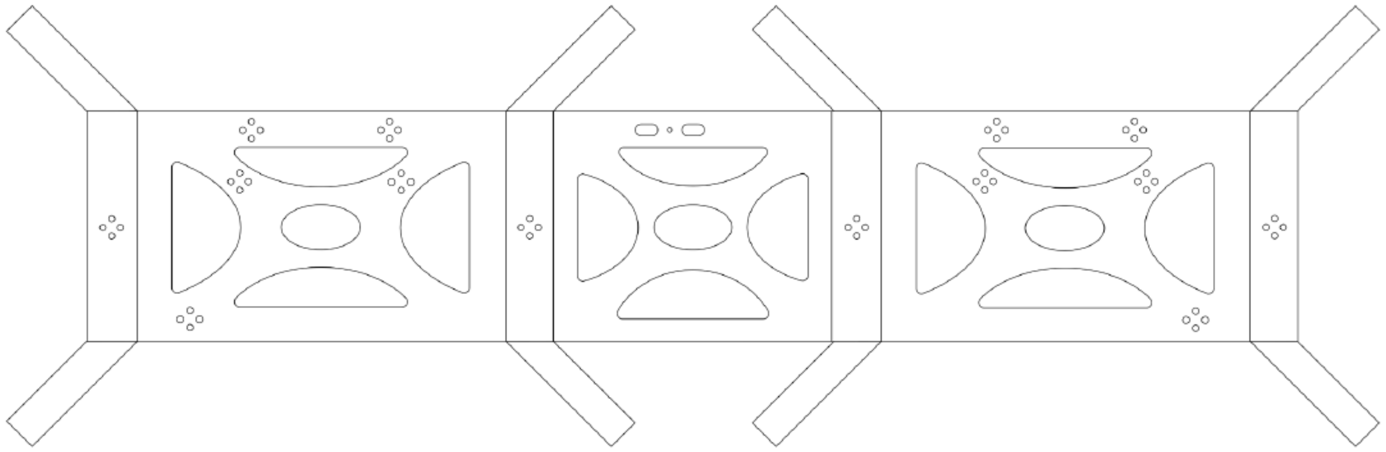
With a multi-purpose gripper, camera, custom video compression and computer vision, the 6.4kg submersible ROV is equipped for any challenge. FLOAT-E is capable of completing numerous tasks, from autonomous differentiation of coral bodies to the retrieval of underwater samples and the maintenance of coral reefs.

Our carefully engineered ROV is built to effectively traverse underwater landscapes and complete challenges in a timely manner. FLOAT-E runs on T100 motors, allowing for smooth travel to points of interest. Our gripper easily carries samples, local flora, fauna, and debris for relocation or removal. In addition to being highly efficient, our robot is decorated from color-matched ballast tanks to underwater-themed paintings decorating the supports. Not only does this provide an artistic element to our ROV, but it brought our team closer together in a time when we had to be physically apart for safety. Despite our challenges coming together physically, we still met diligently as a team and were able to accomplish so much.

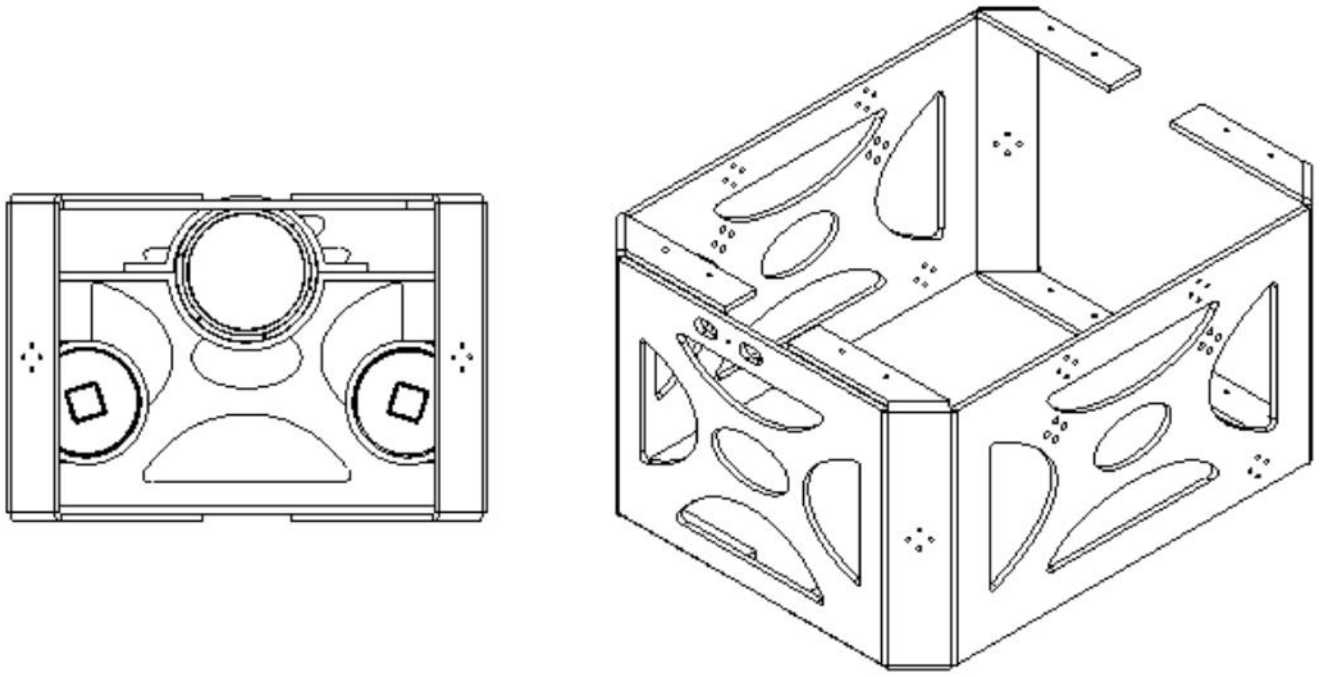
DESIGN PROCESS AND RATIONALE

Chassis and Propulsion

During the 2019 MATE competition, our ROV used a PVC pipe-based chassis design. After working with it for a season, we came to the conclusion that the design came with drawbacks and limitations that included limiting stable mounting points and an overall lack of stability, that a fully custom chassis would be in our best interest. Our design team went through multiple design iterations before settling on our current one, pictured below, dubbed the Milkcrate 2.0 due to how it was manufactured. This new chassis design was created in Solidworks sheet metal to be easily cut on a router table out of PETG, heated up, and bent into its final shape. This material design choice allowed us to implement a more complex design scheme while still keeping the fabrication time and costs to a minimum. This new design also allowed us to mount our BlueRobotics T-100 thrusters in a way that allowed our ROV to navigate on the horizontal plane using a holonomic drive system, which allows for more complex lateral movement while keeping our vertical movement to a simple up-and-down thrust control. This simplistic vertical axis drive system allowed us to control the pitch of the robot with an increased amount of ease, in addition to helping our buoyancy system.



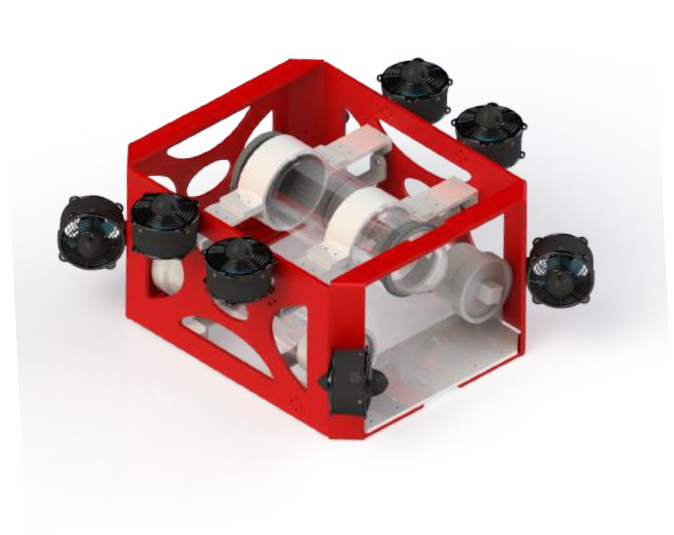
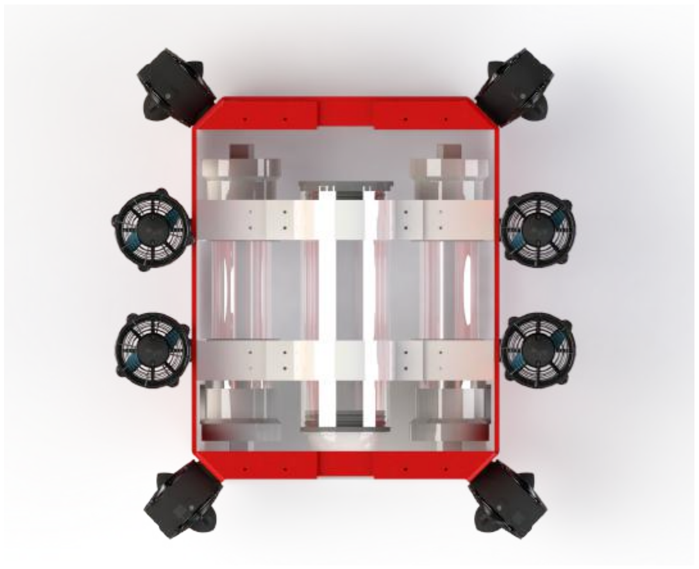
Unfolded view of robot chassis (Image by Nalu Farrell)



Folded robot chassis (Image by Nalu Farrell)

Buoyancy

The buoyancy system on our robot can be broken down into two major parts, the centrally mounted electrical tube, and the two mirrored ballast tanks. Our central electronics tube provides the main form of buoyancy for our robot, acting as a central air tank that evenly applies an upward force to the chassis. Even with this system, our robot is still negatively buoyant and extremely polar. Instead, we want it to be neutrally buoyant to allow for greater ease of control, and we want it to be less polar so we can pitch the robot to our desired angle with ease. We solved both of these problems with the implementation of the two low-mounted ballast tanks. Firstly, they enable us to remain perfectly neutrally buoyant by controlling the exact amount of water and air in each. Secondly, they help keep the robot less polar because when the ROV pitches forward, the upward force from the air in the ballast tanks will be distributed evenly to optimal positions due to the forces of gravity.



Robot chassis renders (Image by Nalu Farrell)

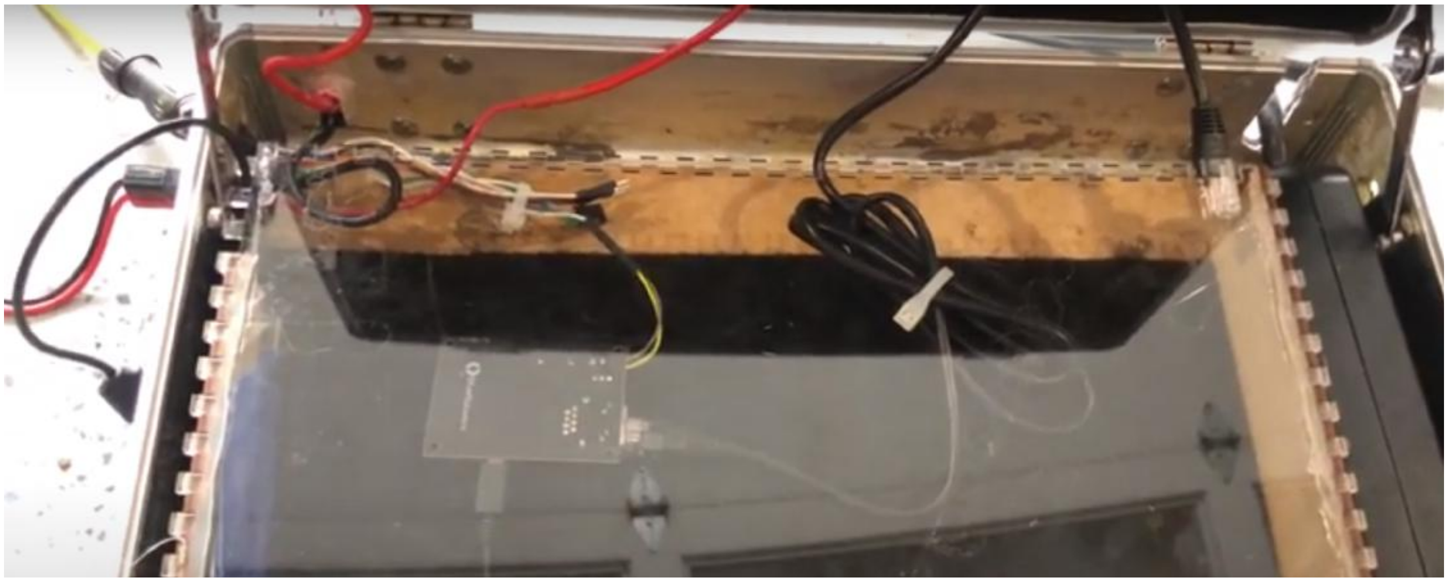
Tether

With a length of 30 feet, our tether consists of a pair of 12 AWG silicon wires and two twisted Fathom X wires. The high-power 12 AWG wires offer power to the robot, which allows us to move and function. As is mentioned in the following electrical section, the twisted data wires enable long-distance reliable communication.

Electrical System

Above Water

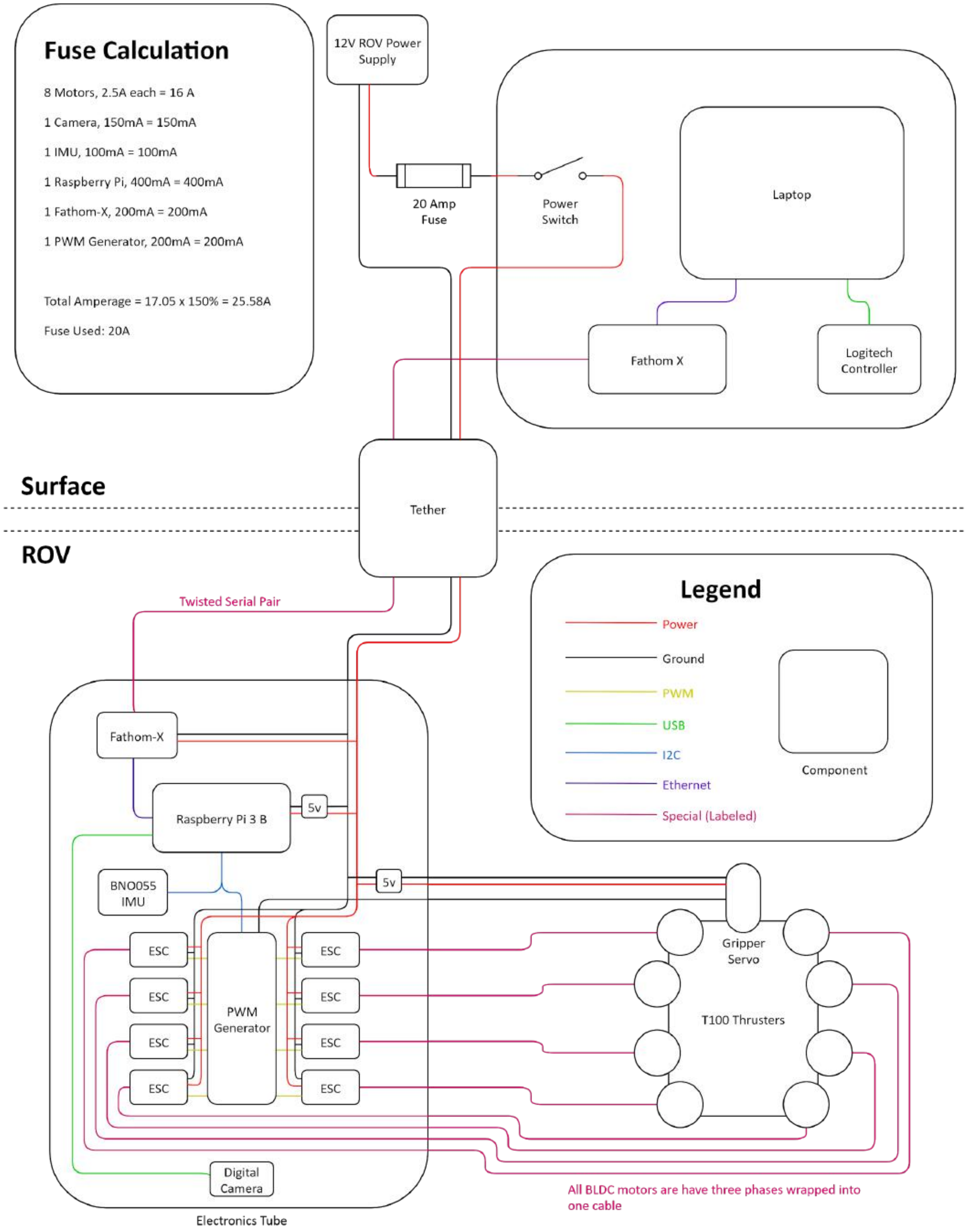
Above the water, our electrical systems are optimized for simplicity and serviceability. The wiring itself sits orderly in a custom organizer, allowing for easy access to all components. This is where we connect our power switch, which later leads to our carefully picked 20 Amp fuse, and then the power supply. Sitting atop this, we have our computer and driver station. Our driver station was carefully engineered to provide mobility and simplicity to our control systems. While sacrificing a second monitor, we are allowed the ability to use any available laptop. Due to this, we can easily set up our driver station wherever is necessary, whether it be at a pool, lake, or ocean. To remedy the lack of an extra screen, we have organized our video and instruments, so the driver can access everything necessary from one screen. This connects to the robot's controller and our Fathom-X. The Fathom-X allows us to have faster and more seamless communication to the robot across long distances. The Fathom-X converts Ethernet to two twisted wires, which have the capability of spanning more than 300 meters. We have considered serial and I2C communication to the robot, however, we decided on the Fathom-X, as it allows us to maintain the use of complex protocols found in ethernet use whilst still communicating over long distances. Blueshift Robotics has firsthand experience with this conclusion, as prior to the redoing of our electrical system in 2019, we used serial communication. This caused numerous communication issues and was incredibly unreliable. In contrast, the use of ethernet has been far less problematic and more efficient.


All wiring in the control box with the laptop removed (Image by Kavi Dey)

Below Water

The primary computer in the ROV is the Raspberry Pi, which controls cameras for the driver, as well as an IMU, which allows for instruments such as an attitude meter and a compass. Our IMU (BN0055) allows us to access absolute orientation data without having to run our own Kalman filter, which is well worth the price. The Raspberry Pi controls our gripper servo and also connects to the PWM Generator (PCA9685) over I2C. The PWM Generator, then, controls our ESCs. We used a PWM Generator because our Raspberry Pi did not provide sufficient PWM ports. Our ESCs are efficient BlueRobotics ESCs, which work off of trapezoidal control and these ESCs go on to control our powerful T100 motors. Organizing everything, we have a custom motherboard PCBA, which has allowed us to have many additional options for our electronics. Everything is placed in a clear acrylic tube from BlueRobotics. This tube allows for all electronics to be seen without dismantling anything, which makes debugging far easier. All of this is visualized on the SID on the following page.

SID



Fuse Calculation

8 Motors, 2.5A each = 16 A

1 Camera, 150mA = 150mA

1 IMU, 100mA = 100mA

1 Raspberry Pi, 400mA = 400mA

1 Fathom-X, 200mA = 200mA

1 PWM Generator, 200mA = 200mA

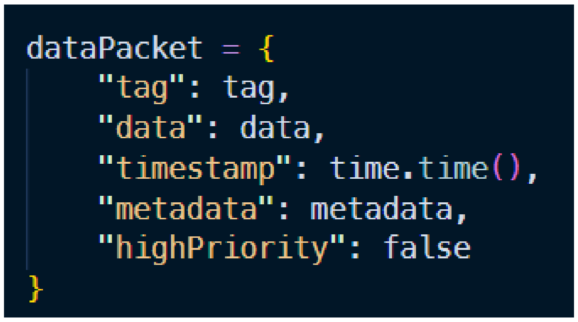
Total Amperage = 17.05 x 150% = 25.58A

Fuse Used: 20A

Code

Our code base consists of three nodes that are interconnected through sockets. We chose this structure for ease of organization as well as the flexibility it affords. The majority of our code is written in Python, except for a few files related to video streaming that was written in C++ for the sake of speed.

The three nodes are water, earth, and air. All inter-node and inter-process communication is done in the form of a custom JSON packet protocol. Because the data is in JSON format, it can quickly be parsed in Python, Javascript, or C++. Each packet has a tag (i.e. motorData, log, stateChange, settingChange, etc.) that defines what nodes and processes it needs to be sent to. Once the packet is created, it is automatically routed through the system to each of its target destinations. This enables low latency, automatic communication, and ensures that data is available wherever and whenever it is needed. See *Appendix 2* for a full system graph.



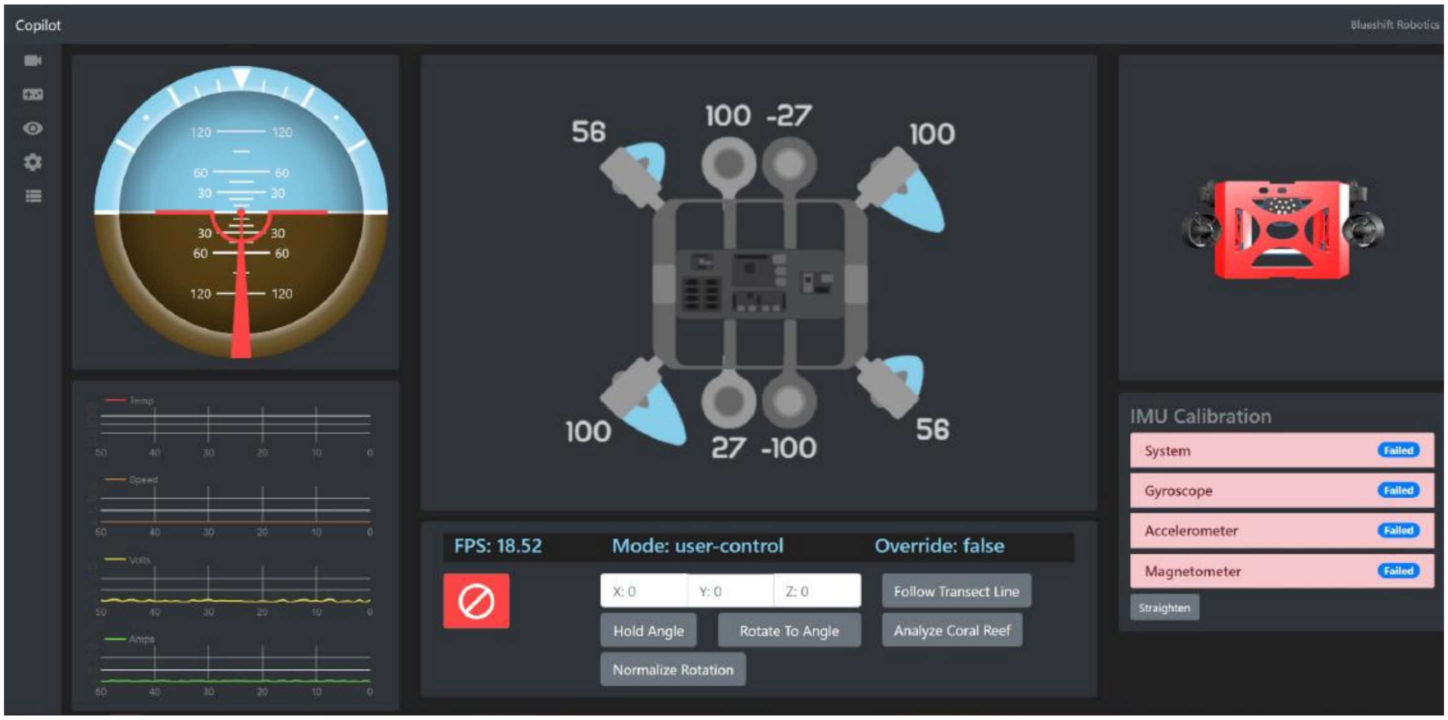
```
dataPacket = {
    "tag": tag,
    "data": data,
    "timestamp": time.time(),
    "metadata": metadata,
    "highPriority": false
}
```

Sample packet (Image by Kavi Dey)

The water node runs on the Raspberry Pi installed within FLOAT-E. It is responsible for decoding and passing on the commands received from the earth node to the motors, as well as encoding and sending the videos captured by the robot cameras to the earth node for processing. Due to the bandwidth limit from the Fathom X ethernet communication, it is necessary to stream compressed video. The Raspberry Pi is not fast enough to compress video with adequate quality, so we developed a custom camera driver, written in C++, using the V4L2 API. It allows us to read compressed MJPEG images directly from the camera, lessening the load on the Raspberry Pi, and giving us a motion to photon latency of only 150ms at 1080p30.

The earth node, as the name implies, runs on a computer above the water or on the ground--connected to the robot through the tether. It is the intermediate point between capturing data such as the video and displaying it to the driver. This is the stage where we process sensor data and run computer vision(i.e. mussel counting, recognizing coral reefs, generating images for the subway car, line-following, etc.) And as touched upon before, it is also responsible for sending the data from the driver's joystick to the water node.

The code for displaying all of the videos captured by the robot is within the air node. This code runs on the same computer as the earth node. The air node is a fully modular web UI. It is built out of a set of individual components (Attitude meter, real-time graphs, compass, IMU/Robot status indicator, etc.) that can be switched in and out based on driver preference. This enables the driver to effortlessly get any information they need, and more effectively control the robot.



FPS: 18.52 Mode: user-control Override: false

X: 0 Y: 0 Z: 0

System Failed
Gyroscope Failed
Accelerometer Failed
Magnetometer Failed

One possible configuration of the air node (Image by Kavi Dey)

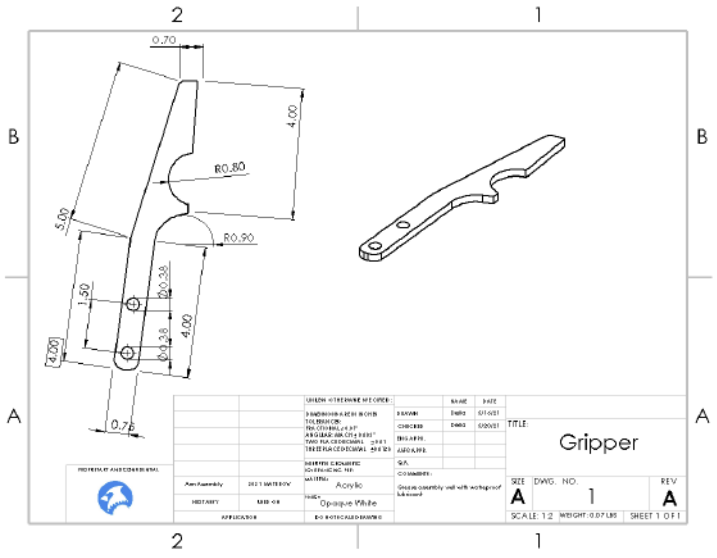
Auto-Stabilization

We designed a feature within the water node where the simple press of a button from the driver will direct the robot to automatically maintain its current angle, despite the movement of the water. This is achieved by recording the angle at the moment our driver presses the button and using it in combination with a well-tuned PID loop and our IMU to maintain the specified angle. This is especially useful in tasks like mapping the coral reef or the subway, where the pilot needs to hold the ROV in a specific orientation while moving through an environment.

TOOLS

Gripper Design

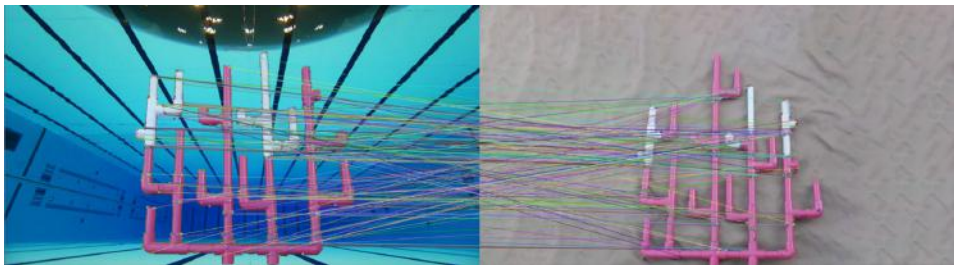
Our gripper was based on a simple but reliable gripper system that relies on a servo-driven gear to move the grippers. The gripper appendages are specifically designed to latch onto various small and multiple different gauges of PVC piping. The grippers themselves are coated in a layer of plastic dip to aid with the gripping surface and also use an upper and lower stabilizer to give it more stability and stop overlap. The gripper itself needs to be well lubricated with quality marine-grade grease to ensure the smoothness of the action. After the initial application of grease, service to the gripper is heavily reduced due to its simple nature.

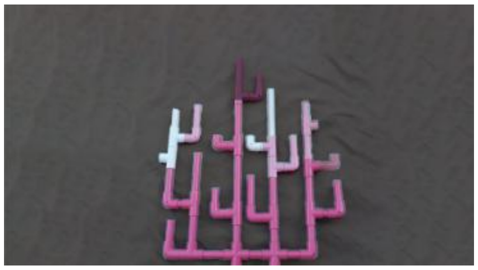


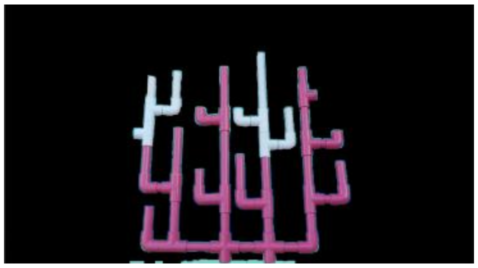
Gripper technical drawing (Image by Nalu Farrell) Gripper photo (Image by Kavi Dey)

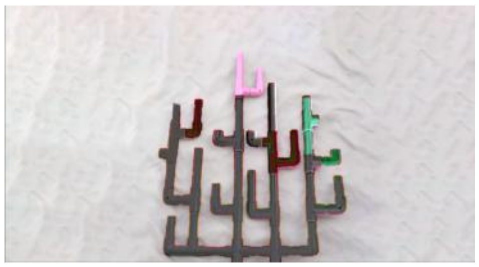
Coral Reef Health Evaluation

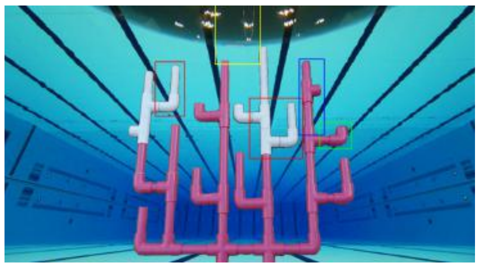
To determine the health of the coral reef, we compare a snapshot of the feed from the camera with the provided reference image of the reef. First, we align the reference image with the new one using the ORB algorithm. Next, we remove the pool background of the image, and then perform a pixel-wise subtraction between the two images, creating a physical representation of the mathematical difference between the two images. Each color in the different image represents a different coral reef change state. For example: if the color was the same in both images, they cancel out producing black, while an area of growth shows up as light green. Finally, we look for and highlight areas of change in the difference. This method of dynamically comparing the two images, allows our algorithm to run on any size or shape of coral reef.


ORB Alignment between reference and new reefs


Aligned images overlayed


Aligned new reef with no bg

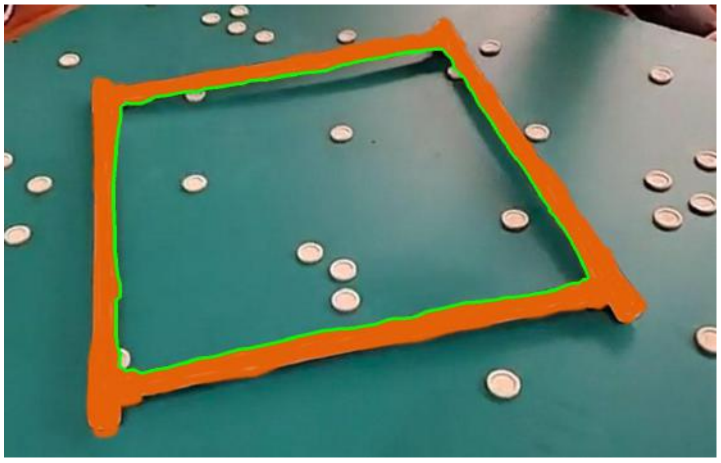

Difference image

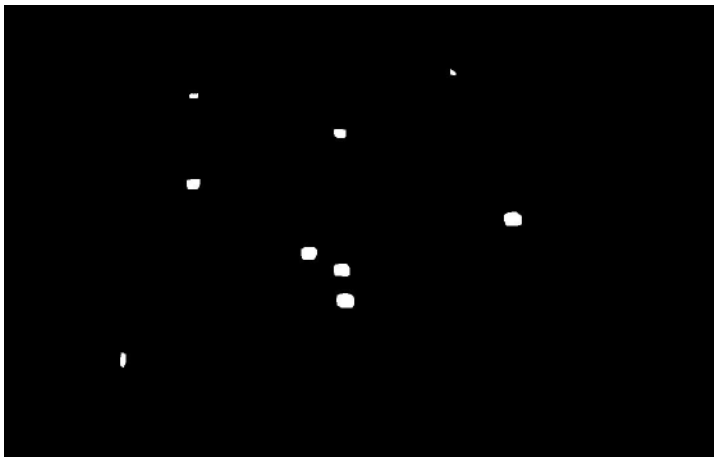

Final Output

(All images by Kavi Dey)

Mussel Counting

For the mussel counting code, we first identified all the mussels based on their color, and then using the same method of identification, we detected the quadrant and deleted all mussels that were not within it. While there is no point bonus to autonomously counting mussels, it saves a significant amount of time compared to doing it manually.

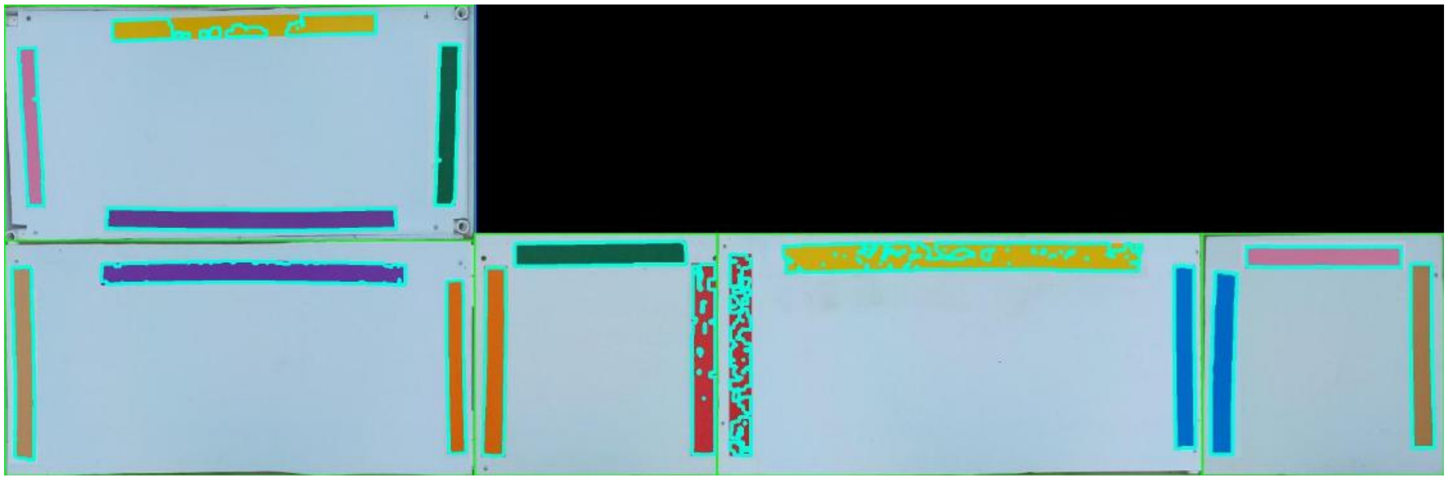

Inside of quadrant detection


Mussels inside quadrant mask

(All images by Jenna Li)

Photomosaic Generation

As for the subway car, we also relied upon identifying items through color. We first cropped the image to the outline of each panel using the previous method, and then we detected the colors on each panel. After determining where each color is on the image, we concatenated each image and generated a new image as the final result. The computer vision code for determining coral health is achieved by overlaying two pictures and finding the homography between them.



Photomosaic output with hue based tape identification (All images by Jenna Li)

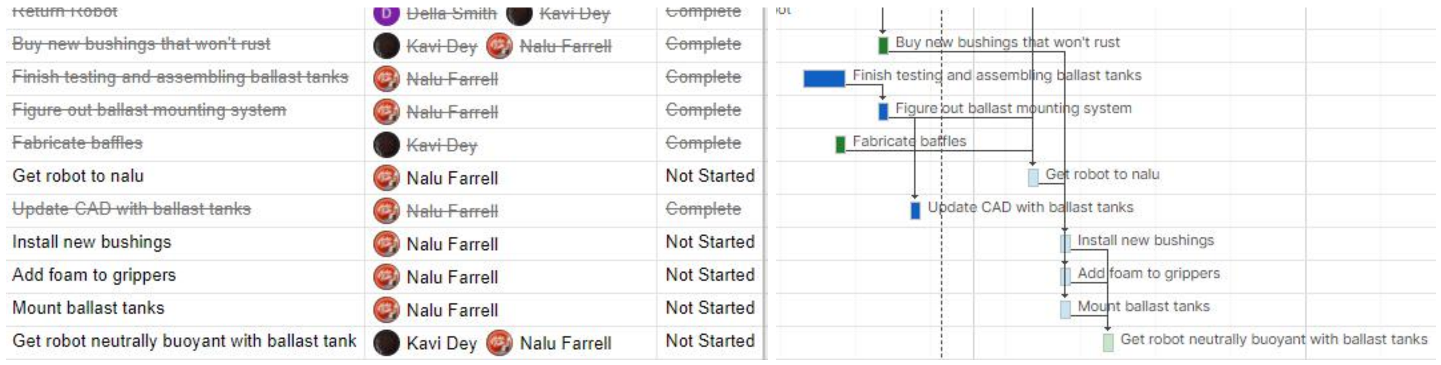
Flying a Transect Line

Like previous computer vision programs above, the code for flying a transect line is written in OpenCV. First, we detect the blue PVC pipes by their color. Following this, we calculate the angle and distance between the two pipes. By knowing the pipe angle, FLOAT-E can ascertain its current orientation, which allows it to make the necessary adjustments to avoid capturing the red pipes on the screen and ensure it moves straight down the transect line. Calculating the distance between the two pipes informs the robot of its distance to the ground. FLOAT-E will then move up or down based on this feedback. The point of this is a combination of evading red pipes as mentioned above and the need to ensure that the blue pipe is always captured on the screen.

LOGISTICS

Project Management

This year, we switched to a new, much more advanced system, generously provided by our sponsor, Smartsheet. While we do not have a large number of employees, a very detailed system was necessary to achieve our ambitious goals. We primarily used the GANTT chart function in Smartsheet. Using the Agile Framework, our CEO, subteam leads, and mentor regularly met to discuss whether the team was on track with our goals and adding new tasks where possible. The tasks are color-coded by their completion status, and which employees are necessary for the task.



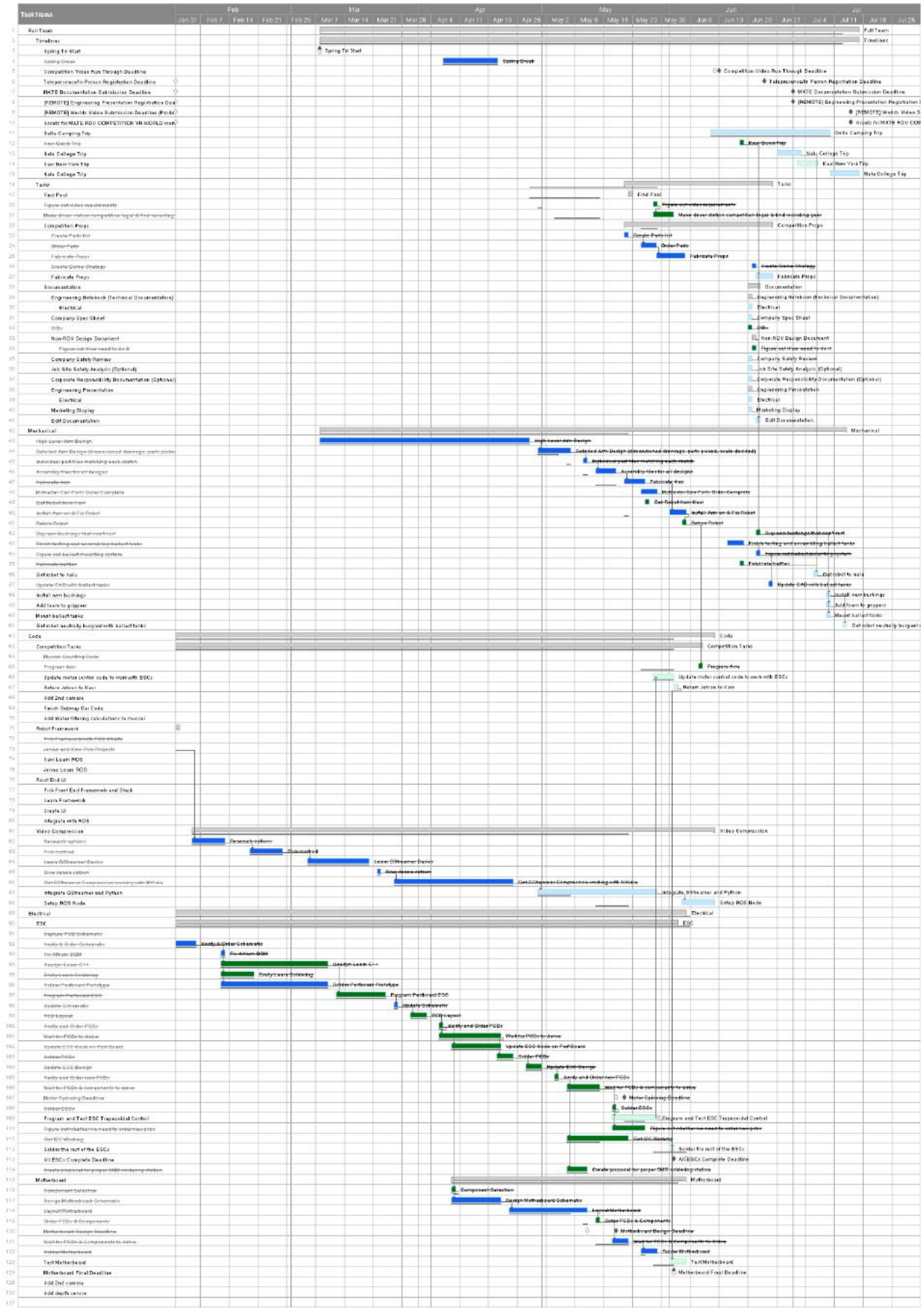
~~Return Robot~~	~~Della Smith~~ ~~Kavi Dey~~	~~Complete~~
~~Buy new bushings that won't rust~~	~~Kavi Dey~~ ~~Nalu Farrell~~	~~Complete~~
~~Finish testing and assembling ballast tanks~~	~~Nalu Farrell~~	~~Complete~~
~~Figure out ballast mounting system~~	~~Nalu Farrell~~	~~Complete~~
~~Fabricate baffles~~	~~Kavi Dey~~	~~Complete~~
Get robot to nalu	Nalu Farrell	Not Started
~~Update CAD with ballast tanks~~	~~Nalu Farrell~~	~~Complete~~
Install new bushings	Nalu Farrell	Not Started
Add foam to grippers	Nalu Farrell	Not Started
Mount ballast tanks	Nalu Farrell	Not Started
Get robot neutrally buoyant with ballast tank	Kavi Dey Nalu Farrell	Not Started

Buy new bushings that won't rust
Finish testing and assembling ballast tanks
Figure out ballast mounting system
Fabricate baffles
Get robot to nalu
Update CAD with ballast tanks
Install new bushings
Add foam to grippers
Mount ballast tanks
Get robot neutrally buoyant with ballast tanks

Sample Smartsheet task list (Image by Kavi Dey)

They are also split up by category (Full Team, Mechanical, Code, and Electrical). Within each category, we also have subcategories. For example, Code is split into Competition Tasks, Robot Framework, Front End UI, and Video Compression. We also use the Smartsheet feature Baselines, which automatically keeps track of the original deadlines that were set when a task was started.

At the beginning of each meeting, the CEO uses the chart to check in with every employee, making sure they are on track, and that they have a plan for that day's work.
Our spring trimester GANTT chart (March - July) is visible on the next page.



Task Name

Full Team
Timelines
Spring Tri Start
~~Spring Break~~
Competition Video Run Through Deadline
Telepresence/In Person Registration Deadline
MATE Documentation Submission Deadline
[REMOTE] Engineering Presentation Registration Dea
[REMOTE] Worlds Video Submission Deadline (Produ
Assets for MATE ROV COMPETITION YR WORLD watc
Delta Camping Trip
~~Kavi Quick Trip~~
Nalu College Trip
Kavi New York Trip
Nalu College Trip
Tasks
Find Pool
~~Figure out video requirements~~
~~Make driver station competition legal & find recording~~
Competition Props
~~Create Parts list~~
~~Order Parts~~
~~Fabricate Props~~
~~Create Game Strategy~~
Fabricate Props
Documentation
Engineering Notebook (Technical Documentation)
Electrical
Company Spec Sheet
~~SIDs~~
Non-ROV Design Document
~~Figure out if we need to do it~~
Company Safety Review
Job Site Safety Analysis (Optional)
Corporate Responsibility Documentation (Optional)
Engineering Presentation
Electrical
Marketing Display
Edit Documentation
Mechanical
~~High Level Arm Design~~
~~Detailed Arm Design (dimensioned drawings, parts picke~~
~~Individual part files matching each sketch~~
~~Assembly files for all designs~~
~~Fabricate Arm~~
~~McMaster Carr Parts Order Complete~~
~~Get Robot from Kavi~~
~~Install Arm on & Fix Robot~~
~~Return Robot~~
~~Buy new bushings that won't rust~~
~~Finish testing and assembling ballast tanks~~
~~Figure out ballast mounting system~~
~~Fabricate baffles~~
Get robot to Nalu
~~Update CAD with ballast tanks~~
Install new bushings
Add foam to grippers
Mount ballast tanks
Get robot neutrally buoyant with ballast tanks
Code
Competition Tasks
~~Mussel Counting Code~~
~~Program Arm~~
Update motor control code to work with ESCs
Return Jetson to Kavi
Add 2nd camera
Finish Subway Car Code
Add Water filtering calculations to mussel
Robot Framework
~~Pick Framework with ROS Basics~~
~~Jenna and Kavi Pick Projects~~
Kavi Learn ROS
Jenna Learn ROS
Front End UI
Pick Front End Framework and Stack
Learn Framework
Create UI
Integrate with ROS
Video Compression
~~Research options~~
~~Pick method~~
~~Learn GStreamer Basics~~
~~Give Jenna Jetson~~
~~Get GStreamer Compression working with NVidia~~
Integrate GStreamer and Python
Setup ROS Node
Electrical
ESC
~~Capture PCB Schematic~~
~~Verify & Order Schematic~~
~~Fix Altium BOM~~
~~Kaedyn Learn C++~~
~~Emily Learn Soldering~~
~~Solder Perfboard Prototype~~
~~Program Perfboard ESC~~
~~Update Schematic~~
~~PCB Layout~~
~~Verify and Order PCBs~~
~~Wait for PCBs to Arrive~~
~~Update ESC Code on Perf Board~~
~~Solder PCBs~~
~~Update ESC Design~~
~~Verify and Order new PCBs~~
~~Wait for PCBs & components to arrive~~
~~Motor Spinning Deadline~~
~~Solder ESCs~~
Program and Test ESC Trapezoidal Control
~~Figure out whether we need to order new pcbs~~
~~Get I2C Working~~
Solder the rest of the ESCs
All ESCs Complete Deadline
~~Create proposal for proper SMD soldering station~~
Motherboard
~~Component Selection~~
~~Design Motherboard Schematic~~
~~Layout Motherboard~~
~~Order PCBs & Components~~
~~Motherboard Design Deadline~~
~~Wait for PCBs & Components to Arrive~~
~~Solder Motherboard~~
Test Motherboard
Motherboard Final Deadline
Add 2nd camera
Add depth sensor

Spring Smartsheet timeline (Image by Kavi Dey)

Safety

Construction Safety

Safety is our number one priority at Blueshift Robotics. We believe that all team members must work together to create a safe working environment. Every decision is better with another set of eyes on it, especially when it comes to safety. We have ensured that Blueshift Robotics' robot complies with MATE's safety specifications and our safety checklist. Our goal is to keep our team members safe throughout the season in every process, including construction, transportation, and even prop building. In the creation of the robot, we ensured that there was always a safe environment to ask for help and collaboratively work on fabrication. This learning environment allowed for team members to supervise one another to make sure safety rules are followed. These rules include wearing safety glasses when using tools, gloves when working with sharp objects, and filtered masks when working with aerosols.

Testing Safety

Testing safety is equally as important as engineering safety in protecting team members and the environments we occupy. All dangerous and potentially hazardous robot components, like our propellers and the gripper, are marked with warning labels. Our propellers are properly shielded to protect team members' hands and the motors themselves. Our front camera is angled down to include the ground in its view to ensure the robot does not hit the bottom of the pool.

Safe thruster (Image by Kavi Dey)

Main power is connected through Anderson Powerpole connectors, and we installed a 20 amp fuse on the main power wire to prevent dangerous current spikes. We do not need to use fluid power, and thus we avoided many of the associated dangers. Our tether has strain relief to protect the wires and connections, and the details are further outlined in the *Electrical section*. All sharp edges are taken off the robot to protect team members and the environment around us. We pressure test and vacuum seal our electronics tube whenever there is any chance that the seal has been broken, whether by the team fixing wiring in the tube, transporting it between different locations, or any time we have any doubt that the tube is sealed. It is vital that no water makes it into the tube for the safety and integrity of our electronics. Safety decisions are collaborative, and all doubts of sufficient safety are addressed accordingly. During testing, every team member must wear safety glasses and use proper caution when interacting with the robot, especially when interacting with a potentially hazardous component. Please refer to our Safety Checklist (*Appendix 1*) and JSA for more safety details.

Troubleshooting and Testing

Fabrication

During our design and testing phase of the project, we had a number of times where we had to troubleshoot small mechanical issues, the first of which was after a handful of test runs our gripper stopped working consistently. After a bit of troubleshooting and group discussion about the problem, we disassembled and lubricated it in about 15 minutes and had it back working better than it had ever been. Another small issue we encountered with the grippers is that they would often close too far and get wedged against each other. In about an hour, our team was able to fabricate braces for the gripper that elevated the previous problem by doing rapid CAD and prototyping out of laser-cut wood before cutting our final design out of acrylic. The final troubleshooting was after our Worlds Qualifier video was filmed. We received feedback from our driver that the robot was too polar, which made it difficult for them to change the pitch of it. Our team, in the process of about a week, was able to design and fabricate a set of ballast tanks that alleviated the problem, making the ROV able to pitch forward and backward a lot easier.

Electrical

One of our biggest single electrical purchases this year was a Rigol DS1054Z Oscilloscope. We used it regularly when debugging electrical issues, and was much more powerful than the multimeter and handheld oscilloscope we were using previously. Much of our troubleshooting came from our ESC project which we talk about more in our *Future Improvements* section. We had little experience with motor control coming into the project, and thus we experienced many instances of troubleshooting. We became familiar with using research to aid in comparing our results with another. We also got familiar with slowly thinking through every detail of our projects. One particular instance of troubleshooting we had was working with our FOC code. We were having trouble getting our Clarke-Park transforms working properly and eventually learned that most sources used incorrect math for our application.

Code

Our entire programming codebase takes advantage of the logging library, which allows for extremely simple logging and printout management. It categorizes debugging statements into four different levels (CRITICAL, ERROR, WARNING, INFO, DEBUG). This, combined with our packet system and custom log viewer built into the air node, makes it very easy to identify how severe an issue is, and the exact location of the bug. Each process in each of our three nodes automatically logs all events that take place, ensuring that our copilot always has the information they need.

Budget and Cost

Income

Sponsors and Donors	Type of Aid	Amount Given
Seattle Academy	Funding	$2,250.00
Lake Ridge Swim Club	Space and Time	$300.00
LightArt	Donation	$225.00

Budget

Expense	Projected Cost	Total Cost
Purchased Components Tube	$600.00	$564.39
Purchased Components Motors/ Other	$400.00	$361.99
Custom Component Materials	$300.00	$253.28
Control Box Components	$50.00	$40.00
Testing Props	$200.00	$155.99
Pool	$300.00	$0.00
Tools	$100.00	$356.90
Other	$300.00	$133.01
Total	$2,250.00	$1,865.56

Expenses

Category	Description	Type	Value	Total Cost
ROV/ drive	CNC Milling	Donation	$100.00	N/A
	T100 Blue Thruster and Speed Controller (6)	Reused	$1,168.00	N/A
	T100 Blue Thruster and Speed Controller (2)	Purchased	$292.00	$292.00
Tube/ Electrical	Watertight Enclosure for ROV 4”	Purchased	$244.00	$244.00
	Fathom-X Tether Interface Board Set	Purchased	$169.00	$169.00
	Raspberry Pi 3 B	Reused	$35.00	N/A
	IMU	Purchased	$34.95	$34.95
	Servo Driver	Purchased	$14.95	$14.95
	PCB Manufacturing	Paid Service	$20.00	$20.00
	Interior Connectors of Tube	Purchased	$20.00	$20.00
	Voltage Converter 12V to 5V	Purchased	$2.49	$2.49
	Marine Epoxy	Purchased	$7.90	$7.90
	Wide Angle Camera	Purchased	$99.00	$99.00
Mission	Underwater Servo	Purchased	$69.99	$69.99
	Gears (3)	Purchased	$45.48	$45.48
	Bushings (9)	Purchased	$5.88	$52.92
	Binding Barrels (6)	Purchased	$17.75	$17.75
	Shaft Collar (2)	Purchased	$5.14	$5.14
Control	Control Laptop	Donation	$699.99	N/A
	Control Box	Reused	$30.00	N/A
	Control Box Components	Purchased	$40.00	$40.00
	Tether	Reused	$550.00	N/A
Other	Materials: Plastics and Metals	Donated	$125.00	N/A
	Materials: Plastics and Metals	Purchased	$75.00	$75.00
	Pool for Testing	Donated	$300.00	N/A
	Mission Task Props	Purchased	$155.99	$155.99
	Rigol DS1054Z Oscilloscope	Purchased	$349.00	$349.00
	Other	Purchased	$150.00	$150.00
Total			$4,826.51	$1,865.56

CONCLUSIONS

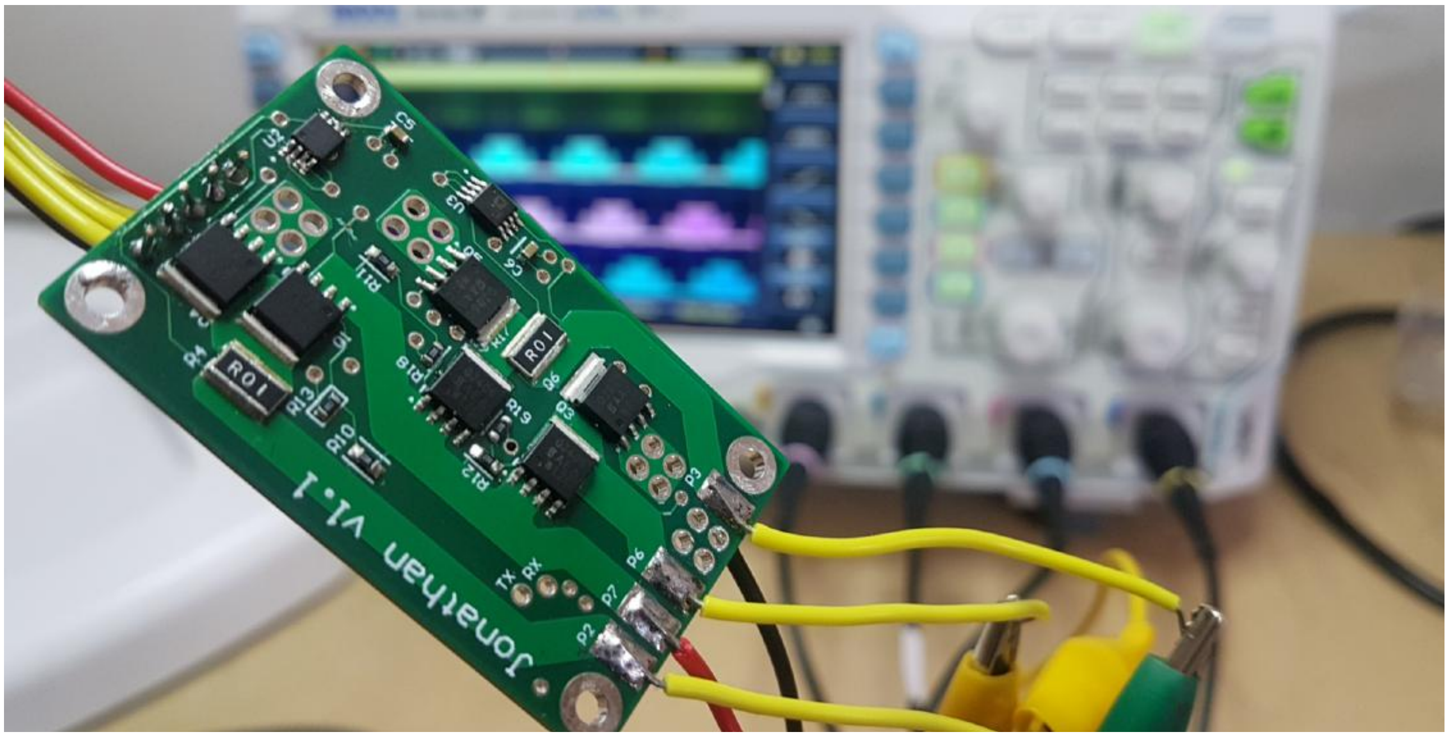
Lessons Learned

Throughout this season, our team has gained an impressive amount of knowledge of electronics, computer vision, buoyancy, and the mechanisms for building a gripper. We have learned to prioritize tasks, to see projects as a whole rather than details, and to work together as a team. For our video submissions, we learned the value of practice and preparation, as well as gained skills to communicate with businesses such as pools for mutual benefit. On top of all this, we have also mastered the skill of collaboration in a pandemic-ridden world. As a team, we learned that despite the forces acting against us, we could still dedicate time and effort into our collective love of robotics.

Future Improvements

Electrical

Throughout the year, our team has had an ongoing electrical project to create our own ESCs from start to finish. We decided on this project because we were unsatisfied with the BlueRobotics ESCs and hoped to design functional ESCs with more features for a lower price. Going so far as creating a class at our school to focus on this project, we have gotten impressively far. We had hoped to complete our ESCs in time for the 2021 competition but were unfortunately not able to finish the control code in time. In the future, we plan to implement Field Oriented Control to most efficiently drive our motors and use them in the 2022 competition.



Functional ESC and oscilloscope (Image by Kavi Dey)

Code

We had planned to switch out our current operating system for another with higher bandwidth and lower latency. We had a few contending options-- namely LCM, ZeroMQ, ROS, or making a brand new operating system ourselves. We ultimately chose ROS as its advantages eclipsed those of the other options. It more than meets the objectives mentioned above as well as being user-friendly, updated and customizable. However, due to the time constraints of a remote season, we were unable to accomplish this switch for this year's competition so we plan to implement it before next year's competition.

Acknowledgments

Altium Systems - Their donation of Altium PCB design software
Dassault Systemes - Their donation of Solidworks software
Lake Ridge Swim Club - Their generous donation of pool time
Seattle Academy - Its generous donation of funding
MATE Center and Marine Technology Society - Sponsoring this year's competition
Meghan Jimenez - Their time, experience, and guidance this year
Smartsheet - Their donation of Smartsheet licenses
LightArt - Its donation of CNC time and plastic
Our Families - Their unending support

References

Python. "logging — Logging facility for Python." Python. Python, 30 June. Web.
Blue Robotics. "T100 Thruster Documentation." Blue Robotics. Blue Robotics, 22 Aug 2015. Web. 30 June 2021
MATE. "MATE ROV Competition Manual Ranger." MATE. MATE, 3 March. 2019. Web. 30 June 2021

APPENDIX

1. Safety Checklist

Team Members Should:

- ☐ Wear safety goggles when using power tools and tools that have the potential to damage eyes.
- ☐ Acquire adult supervision especially when working with heavy machinery and any tools on campus, peer supervision needed for handheld tools like drills and soldering irons off-campus.
- ☐ Long hair must be kept up, accessories that could potentially harm the user or another must be removed.
- ☐ Respect all team members, do not bother team members when they are operating machinery, maintain a trusting environment.
- ☐ Maintain professionalism especially when at the pool when there are several hazards to keep track of.

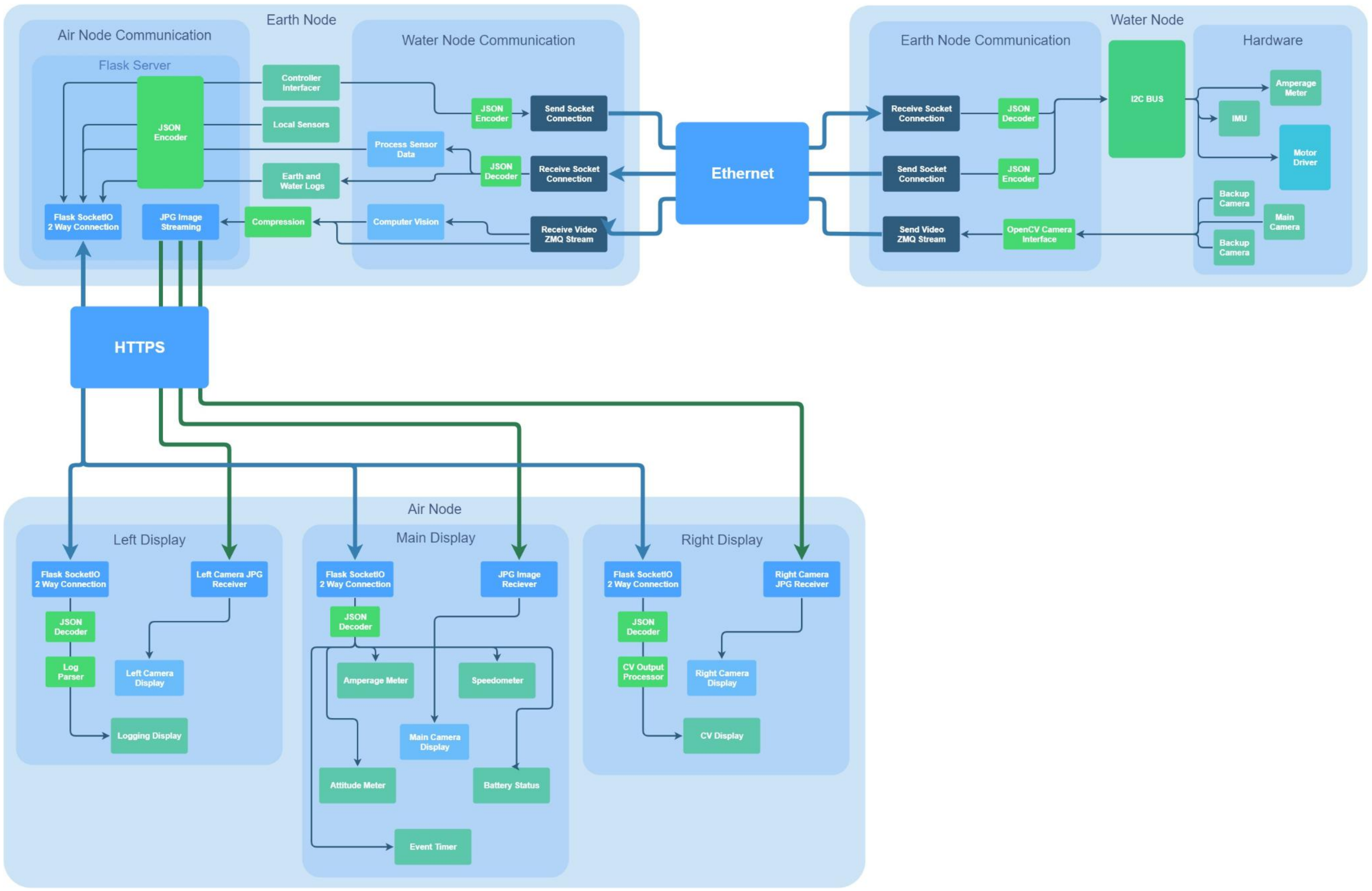
Robot Safety Checklist

- ☐ 20 amp inline fuse on the main power wire within 30 cm of the power source
- ☐ All propellers are shrouded, and there should be some sort of a barrier on all moving parts to protect the component and those who interact with it.
- ☐ Potential hazards such as all moving parts must be marked with a hazard sticker
- ☐ Check for and eliminate all sharp edges on the robot
- ☐ No shorts or broken electrical connections. If these are outside of the tube, also ensure they are properly waterproofed. Ensure there are no exposed wires.
- ☐ All connectors and components should be secured tight so that no water can get into the tube, the wires should not be twisted to the motors to put less strain on the connections
- ☐ Place the control box in a safe place away from splashing, rain, or other hazards.

Testing Checklist

- ☐ Pressure test and vacuum seal the tube when any changes are made inside the tube, or a team member is concerned.
- ☐ Make sure all control box plugs are connected including the tether, power, and the camera.
- ☐ Notify everybody on deck when the robot is active with “robot on,” members on the pool deck must acknowledge that they heard by repeating, “robot on.”
- ☐ Fully test the motors, gripper, and camera before placing it into water to identify issues. When testing motors or the gripper, the driver says “motor test” or “gripper test” respectively.
- ☐ Communicate clearly when the robot enters the water with a countdown: “3, 2, 1, launch”
- ☐ Ensure that the tether is accounted for and managed by a team member.
- ☐ A team member must always have eyes on the ROV in order to retrieve mission items.

2. Control Node Diagram



(Image by Kavi Dey)