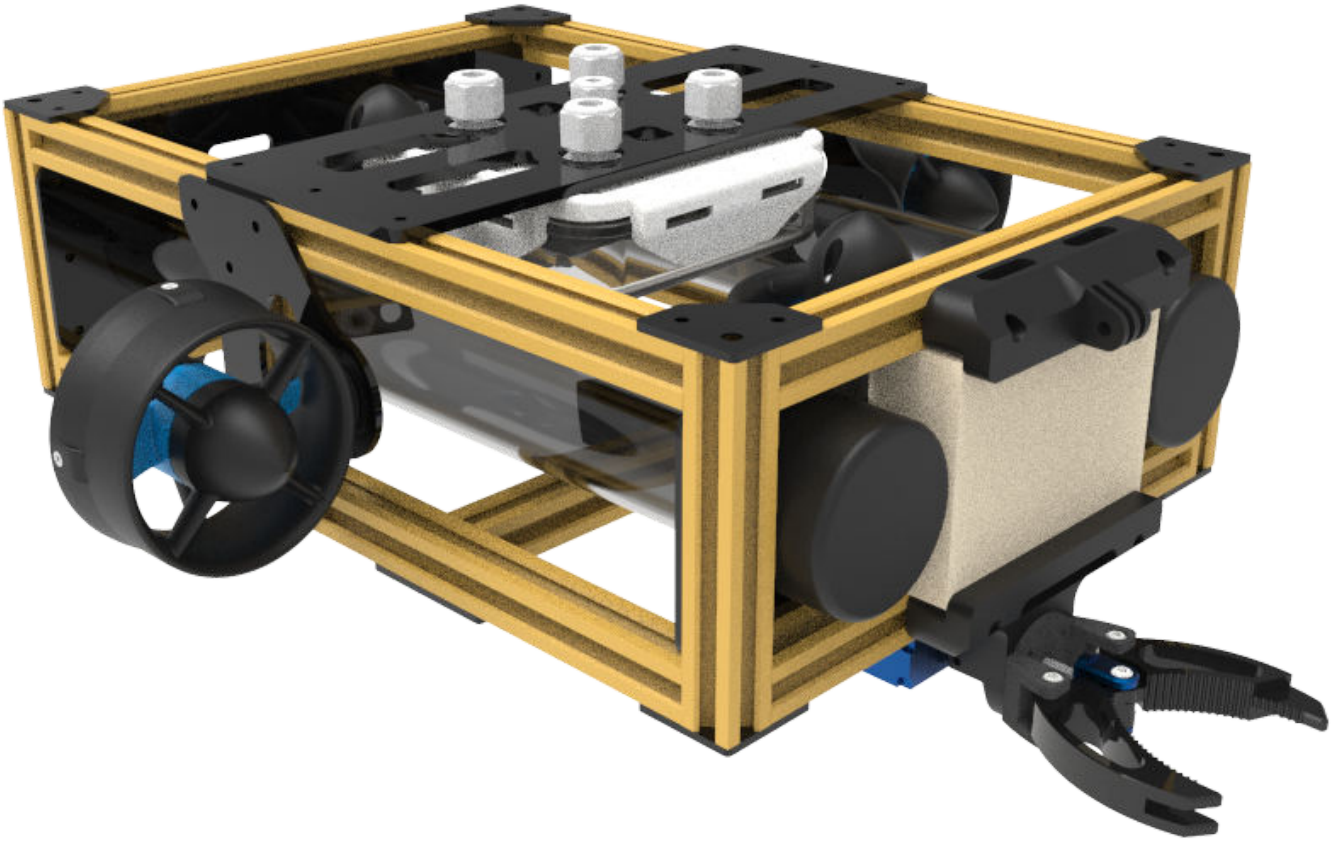
One Degree North

[Singapore American School, Singapore]



		Mentors
Ming Jin Yong	CTO, Head of Mechanical	
Emilio Orcullo	Mechanical Engineer	Barton Millar
Anay Gokarn	Mechanical Engineer	Meredith White
Jefferson Zhang	CEO, Head of Electrical	James Harvey
Alex Tao	Electrical Engineer	
Ayam Babu	Head of Software	
Liam Kelly	Software Engineer	
Nitya Arora	Software Engineer	
Rimi Chakravarti	CFO, Documentation Manager	

Contact: robotics@sas.edu.sg

1. - Introduction

1.1. - Abstract

For the past four months, our team has worked tirelessly to design, fabricate, build, and set up our newest ROV, nicknamed the “Yellow Submarine.” Understanding that plastic pollutants of every shape and form are accumulating in waterways worldwide, we looked to design an ROV that could adapt to as many tasks and environments as possible. The ROV is packed with features such as automated leveling, high-powered thrust, low-profile buoyancy tubes, adaptable aluminum v-slot structure, and a two-position claw.

Despite the COVID-19 pandemic, we continued to persevere in building this ROV, knowing that it is not the only problem we will be facing in the coming years. We spent countless weeks shipping our ROV back and forth from our technicians and completed the hardware just in time for the competition. With this ROV, we hope to not just compete in the MATE Ranger competition but to create a product that is low-cost and adaptable to solve our problems in our waters.

1.2. - Table of Contents

1. - Introduction
- 1.1. - Abstract 2
- 1.2. - Table of Contents 2

2. - Mechanical Design
- 2.1. - Structure 3
- 2.2. - Buoyancy 4
- 2.3. - Propulsion 5
- 2.4. - Enclosure 6
- 2.5. - Manipulator 7
- 2.6. - Surface Capture 8

3. - Electronics Design
- 3.1. - Architecture 9
- 3.2. - Supplied Power 10
- 3.3. - Cameras & Sensors 11
- 3.4. - Motor Control 12
- 3.5. - Communication 12

4. - Software Design
- 4.1. - Onboard 14
- 4.2. - Control Station 15
- 4.3. - Driver Controls 16

5. - Project Management
- 5.1. - Company Profile 18
- 5.2. - Work Timeline 19
- 5.3. - Safety Protocol 20
- 5.4. - Budgeting & Cost 22

6. - Acknowledgements 23

7. - References & Appendix
- References 24
- Appendix i. - Full SID 25

2. - Mechanical Design

Technical Specifications: 0.55m x 0.5m x 0.2m ; 10kg

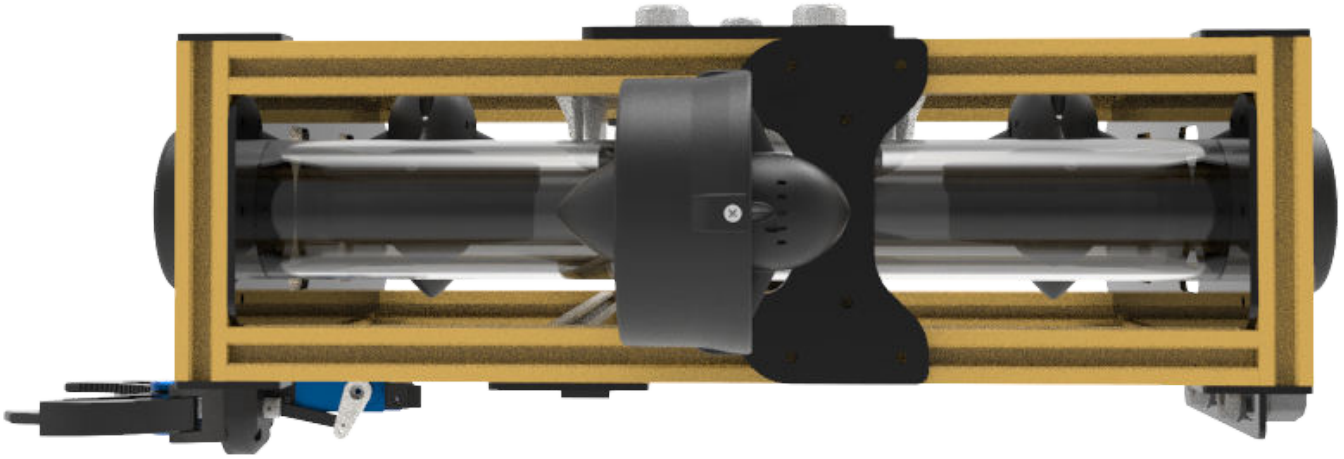


*Fig. 2.A - KeyShot Render of Yellow Submarine**

The mechanical hardware on this ROV was one that we took strides in developing. We find it unique in the style it was built in, with its aluminum frame, custom-designed mounts, calculated air tubes, cable glands for wire management, and flexible mounting rails.

2.1. - Structure

2.1.1. - Aluminum Frame

Our company made an early decision to use aluminium v-slot linear rail (20x20mm). Aluminum V-slot is cheap and flexible, allowing us to adjust our thrusters, cameras, and payload positions late into the season without significant modifications to the frame. Being easily adaptable to new tasks was something our team strived to achieve from the start, and this seemingly small decision has helped contribute to our success. Our time spent on initial prototyping drastically decreased from two weeks to one and consequently provided us with more time in fabricating/tweaking the final parts.

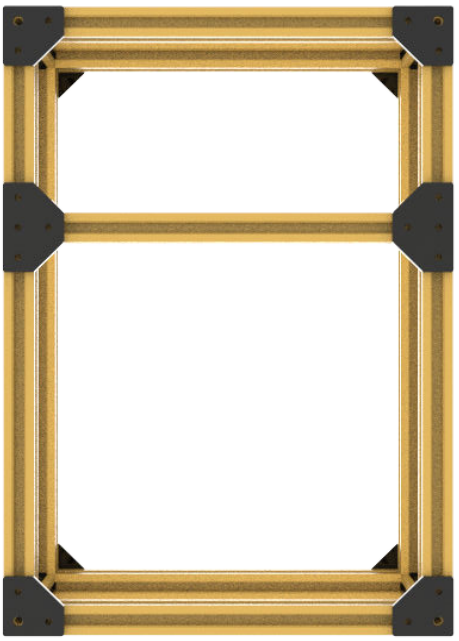


Fig. 2.B - Render of frame structure

2.1.2. - Delrin Gussets

Holding these nine structural pieces together are “business card sized” laser cut gussets made from Delrin sheets which lock to the structure using T-Nuts and screws.

We had three materials to choose from when creating these plastic gussets: polycarbonate, acrylic, and Delrin. We considered the safety, efficiency, accuracy, and cost of creating the gussets to select the best material. Polycarbonate releases unsafe fumes while being laser-cut and can only be machined, which would increase the cost as we would need to outsource. Acrylic was a material that our previous MATE teams have historically used. We decided not to continue because of its low durability and high tendency to crack under small shock loads. As such, Delrin was the best option as it is non-shattering, safe, and cost-effective.

2.2. - Buoyancy

2.2.1. - Air Tubes

After looking at a few options for buoyancy, such as low-density foam and pool noodles, we found through research that the buoyant force of air does not change with pressure (only temperature). This meant that we could get the ROV into a neutral buoyancy using a set volume of captured air. We derive this volume from the weight of the ROV, density of water, density of air, and gravitational acceleration.

2.2.2. - Volume Calculation

Weight of ROV	*	**Gravitational Acceleration**
Force of Gravity (-) =		

Buoyant Force (+) =				
Density of Water	*	**Volume of Air**	*	**Gravitational Acceleration**

Fig. 2.C - Buoyancy calculations

When both equations above are equal, the ROV will become neutrally buoyant. As every variable (except for the volume of air in the PVC) is a physical constant, basic algebra can derive the volume.

With the volume of air fully calculated, we found some 1” diameter PVC tubes and cut them to length. Because they were two times the length of the ROV frame, we split the tube on the midline and sealed them off using four PVC end caps. These tubes were then mounted beside the vertical thrusters within the v-slot frame and held in place using laser-cut Delrin mounts.

2.2.3. - Polyurethane Foam

To counter the additional weight of the aluminum claw at the front, we had to level the ROV by adding buoyant material to the front. To achieve that, we sanded the block down bit by bit on one face and observed the ROV using a bubble leveler once attached. We removed approximately half an inch off the original block, then modeled and made a mount to hold the block in place.

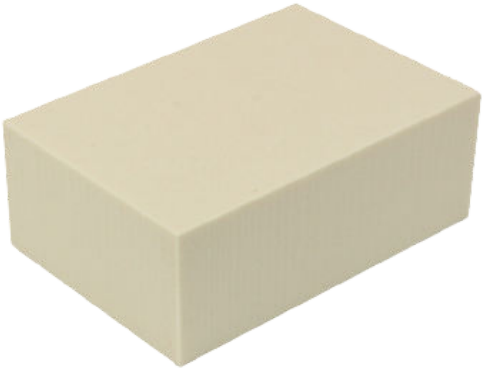


Fig. 2.D - Polyurethane Foam Block

2.3. - Propulsion

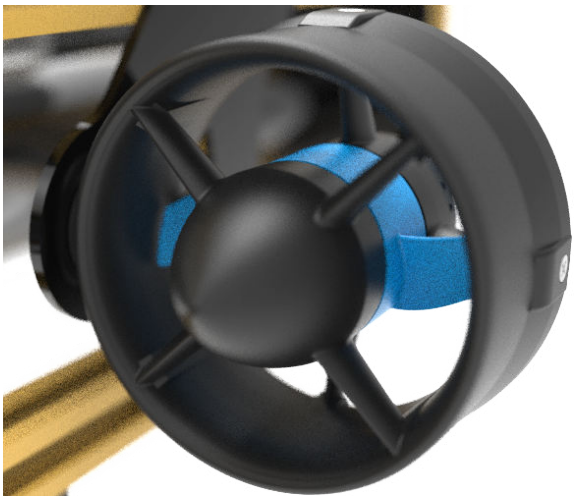


Fig. 2.E - Render of a T100 Thruster

2.3.1. - T100 Thrusters

To maneuver our ROV in the water, we used Blue Robotics’ T100 thrusters. Due to their use in years prior, re-using them reduced our development costs significantly, which allowed us to spend our limited budget on other components. In total there are four thrusters. T100 thrusters deliver approximately 18N of thrust each, providing adequate thrust for the maximum weight of 5N in our tasks. The thrusters are typically used at 10-60% speed, delivering 2-10N of thrust.

2.3.2. - Thruster Placement

On our ROV, we have four thrusters in total. We mounted two on the side for control over yaw + horizontal translation and two inside the frame for control over pitch + vertical translation. This minimized our power consumption as we only chose the essentials to having a maneuverable ROV.

2.3.3 - Propeller Guards

Safety is our number one priority within our company. Before running any thruster tests, we secured two IP-20 rated guards onto both sides of the thrusters to keep any fingers off the propeller. These were 3D-printed, screw mounted to both the backing, and zip-tied to the front of the thruster. The propeller's position allowed for easy maintenance and replacement in case of damage.

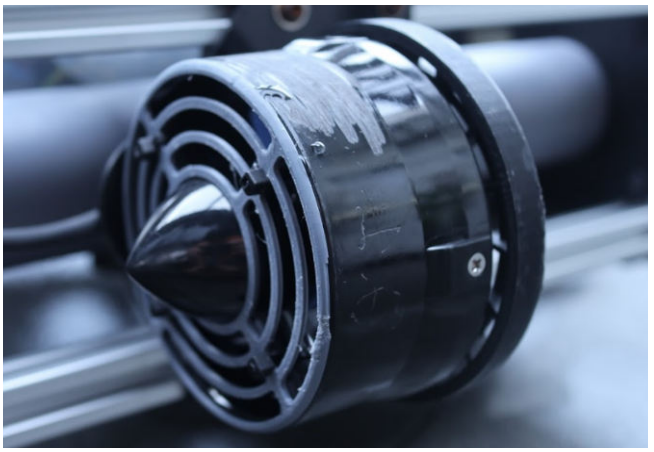


Fig. 2.E - Propeller Guards

2.4. - Enclosure

2.4.1. - Tupperware Container

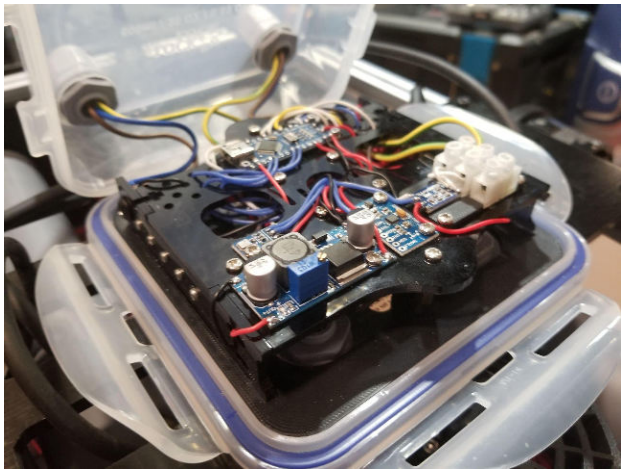


Fig. 2.E - Tupperware container for housing onboard electronics

We tested various storage containers and found that a "Lock & Lock Food Container - 870ml" worked incredibly well in keeping its internal contents dry because of its round corners, rubber seal, and tight edge locks. We validated it by filling the different containers with rocks, keeping them in a pool for 24 hours, then checking the internals for moisture after. This container costs only $10 and fits right into our low-cost concept of the ROV, storing all of our required electronics.

2.4.2. - Cable Glands

To maintain a waterproof electronics box, we used PG11 and PG7 plastic cable glands to keep a watertight seal between the wires and container. Before installation,

we drilled a 6.5mm hole into the Tupperware container and spread a loop of epoxy around it. Next, a gland placed into the hole, and a tightened nut, prevent water from leaking through the edges. The wire is then passed through the gland and tightened using its rubber ring. Finally, we potted the internals to ensure a fully waterproof seal as an additional layer of reliability.

Fig. 2.F - PG9 cable gland

2.4.3. - Multi-Layer Plate

We designed our electronics as compact as possible using a two-layered plate system made from laser-cut acrylic sheets. The lowest layer holds the electronic speed controllers for the thrusters. Beside them sit the positive and negative power distribution rails. The top consists of the main microcontroller, gyroscope, and pressure sensor.

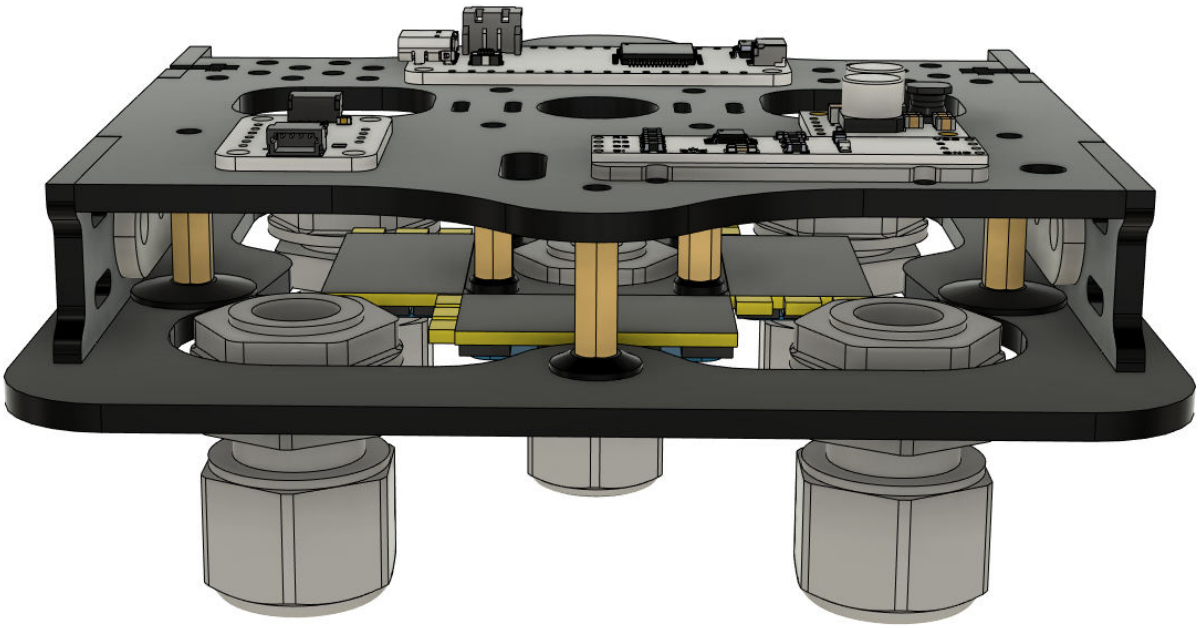


Fig. 2.G - Render of onboard electronics plate

2.5. - Manipulator

2.5.1. - Modified Newton Gripper

Our manipulator consisted of a Blue Robotics Subsea Newton Gripper reused from previous years in our initial design. However, a leak during testing fried its internal electronics, rendering the claw dysfunctional. Factoring in time and money, our team

decided to replace the broken claw with one we could design and fabricate in-house. In addition, the claw would reuse the jaws from the Newton Gripper kit as we had limited access to the CNC during the lockdown period.

2.5.2. - 25kg Waterproof Servo

This claw uses an LW-25MG Digital Waterproof Servo and a few 3D Printed components. Since manufacturing was done cheaply using a common FDM style printer and the servo was one we already had in stock, there was no time wasted in waiting for new parts to shop or money spent to order custom machined parts.

To grip onto coral, pins, and other game pieces, the servo's rotational motion is translated to linear motion through a central rod, pushing and pulling on the gripper's jaws.

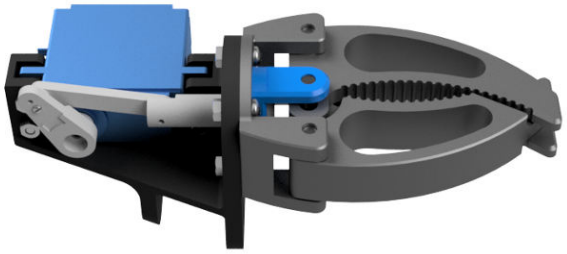


Fig. 2.H - 3D render of a closed claw.

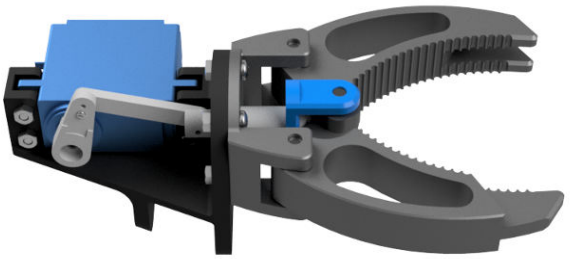


Fig. 2.I - 3D render of an open claw.

2.6. - Surface Capture

2.6.1. - Mesh Net Attachment

The surface capture attachment was designed to capture multiple ping pong balls from the surface of the pool in one continuous motion; and can hold up to five game pieces at any point.

This system works by using a tensioned mesh net constrained by three carbon fiber rods mounted at different heights. As the ROV rises and reverses into the pieces, they float up against the net and lock into position while water is allowed through.

3. - Electronics Design

3.1. - Architecture

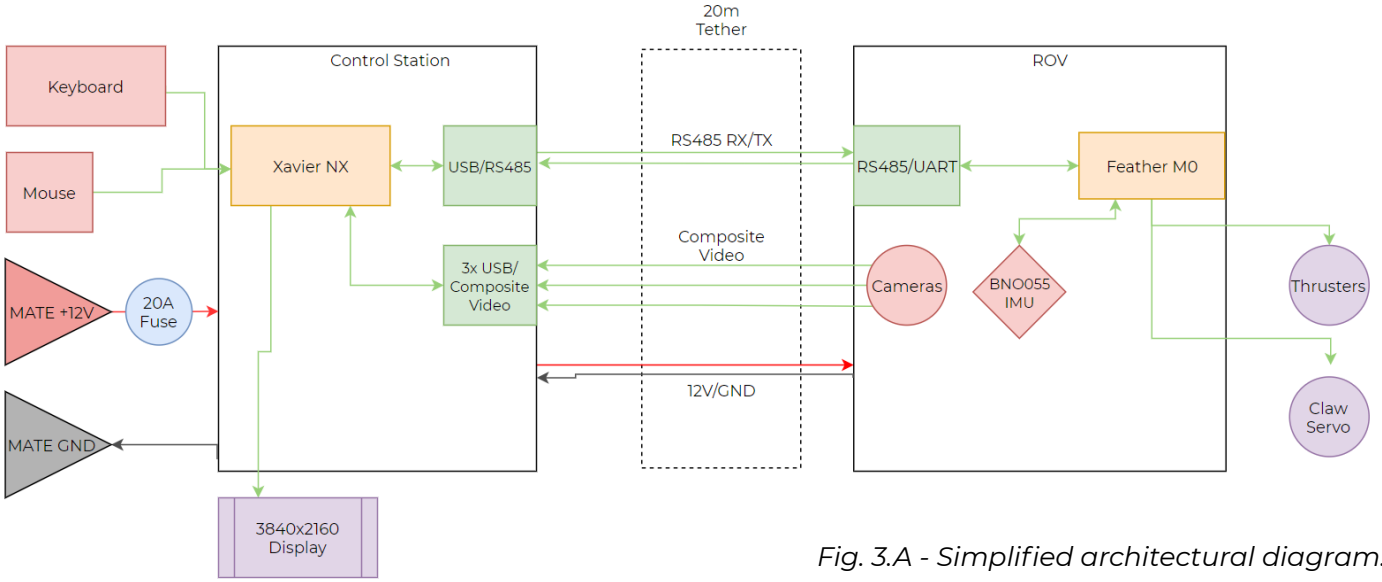


Fig. 3.A - Simplified architectural diagram.

The system architecture consists of the ROV and the control station, connected via a 20-meter tether. The control station handles all computer processing and human interaction. The electronics on the ROV serve only to translate instructions sent by the control station over serial communications into real-world actions. This split increases the reliability of the ROV as the team can perform real-time checks on the communications on both ends, ensuring that all commands sent are valid, even at a higher communication speed. This split also eases development, as the control station implements all logic, all algorithms can be developed over the internet, remotely.

3.1.1. - Off-Board (Control Station)

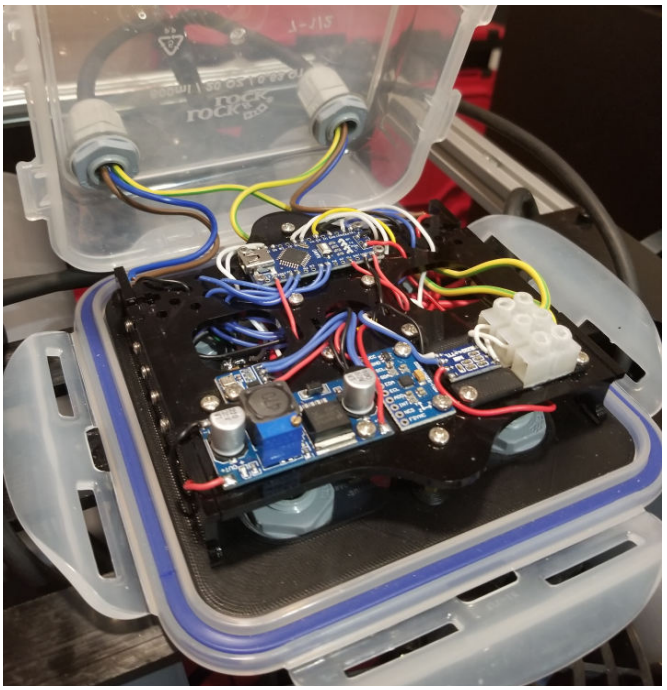
Left: the control station with keyboard and monitor.

Right: individual switches for powering each component of the ROV system.

The control station is housed in a pre-owned computer case and consists of an NVIDIA Jetson Xavier NX, three composite camera capture cards interfacing with the Xavier over USB, a keyboard, mouse, display, and various power systems supplying power to each component. The Xavier NX acts as our primary human interface, running our control GUI.

3.1.2. - On-Board (Electronics Box)

The ROV's onboard electronics are housed in a Tupperware container and are mounted onto laser-cut mounts, ensuring stability for sensors onboard. (See 2.4) A microcontroller, the Adafruit Feather M0, controls our thrusters and servos. The microcontroller acts as an interface between our sensors, motors, and commands from the control station. The onboard electronics consists of the Feather, power distribution blocks for 12V and GND, four electronic speed controllers (ESCs), a 5V regulator, and our BNO055 inertial measurement unit (IMU).



3.2. - Supplied Power

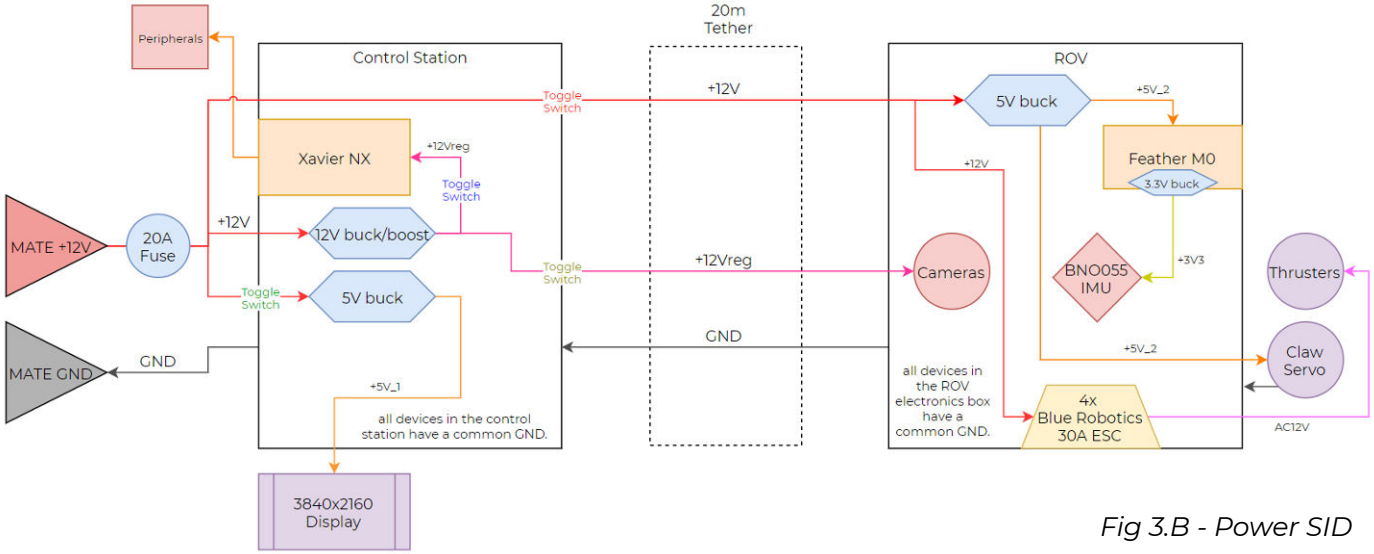


Fig 3.B - Power SID

3.2.1. - Electrical Characteristics

Characteristics	Symbol	Test Condition	Value	Unit
Theoretical Min. Voltage	Vmin	One thruster set to 1700us through command	8.5	V
Theoretical Max. Voltage	Vmax	One thruster set to 1700us through command	22.2	V
Standby Current	I	Vin = 12.2V, no thrusters running, all devices in sleep	330	mA
Typical Current	I	Vin = 12.2V, two thrusters running at 48% in water	6.1	A
Maximum Current	Imax	Vin = 12.2V, all thrusters set to 2000us in water	19.8	A

Fig 3.C - Electrical Characteristics

3.3. - Cameras & Sensors

3.3.1. - Analog Cameras

The ROV has three 2-megapixel composite video (analog) cameras onboard it, facing forward, downward, and focusing on the claw and its maneuvers. The forward-facing camera, the primary camera, is used to navigate the ROV through its environment. A downward-facing camera allows easier image capture while stationed above points of interest, and the claw focus camera aids the driver in visual feedback from claw maneuvers. Computer vision algorithms can use all three cameras, as they are interfaced digitally through EasyCAP capture cards.

3.3.2. - IMU Gyroscope

The ROV also uses an IMU to remain neutrally buoyant artificially and to affix rotation while moving, allowing for more precise driver movement and the ability to move autonomously. For this, a Bosch 9-axis BNO055 IMU is used, providing precise linear acceleration and absolute orientation data directly to the control station. Additionally, the ROV provides voltage data from a simple resistor-based analog input for monitoring and diagnosis.

3.4. - Motor Control

3.4.1. - Blue Robotics ESC

To control the ROV's 3-phase brushless T100 thrusters, we reused Blue Robotics' Basic ESCs to allow digital speed control via the Feather M0 microcontroller. These ESCs take readily producible 400Hz PWM signals and convert them into 3-phase AC instructions to power the thrusters.

3.4.2. - PowerHD Servo

The claw uses a PowerHD LW25MG servo, providing 180 degrees of motion for the updated claw mechanism. (2.5) This servo is powered directly from 5V, despite its rating of 6.0-7.4V, as it reduces the number of required power rails and lowers power consumption. The servo is interfaced through standard 500Hz PWM.

3.5. - Communication

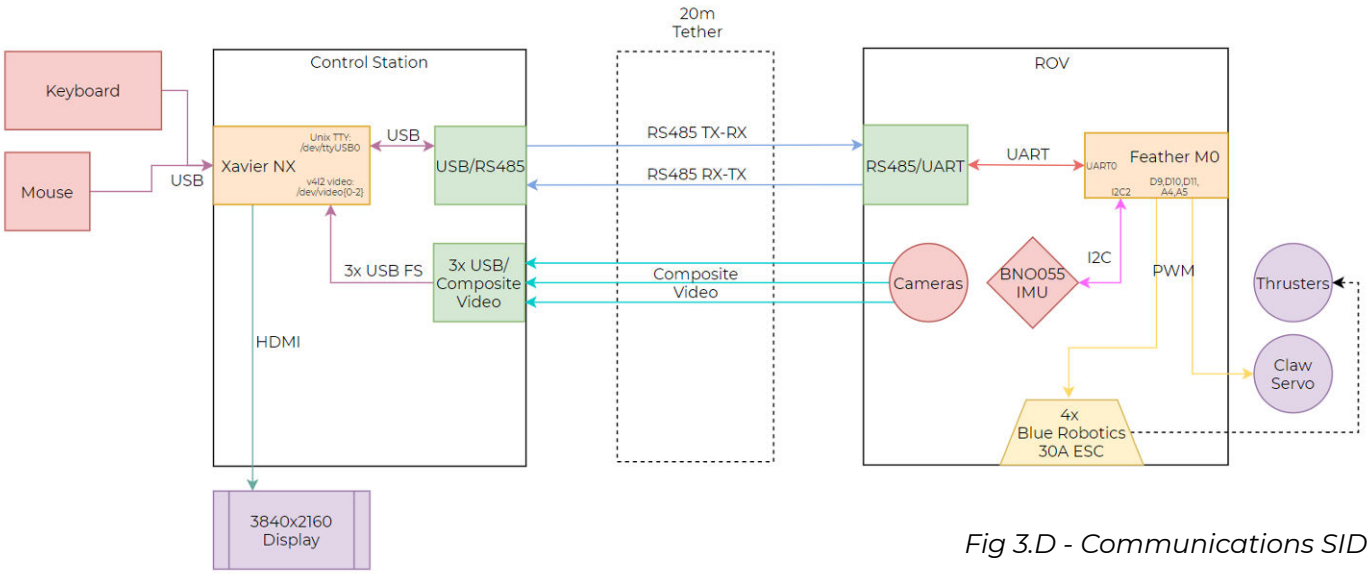


Fig 3.D - Communications SID

3.5.1. - RS485 Protocol

We communicate between the control station and ROV through long-range RS485 communication converted to more usable serial interfaces on either end. On the Xavier NX, RS485 communication is achieved with a USB-RS485 interface. On the ROV, serial UART from the microcontroller (Feather M0) is adapted into RS485 through a converter module.

As the RS485 interface delivers data in bytes only, we developed a communication protocol to allow more comprehensive data transmission between the two sides. The Xavier NX uses 8-byte packets containing a header, footer, and parity to

send commands to the ROV, and all data returning from the ROV is sent through 10-byte packets, also containing a header, footer, and parity.

The control GUI delivers commands to the ROV through a microcontroller interface library (mcu.py), which meticulously checks parameters and timings to deliver the data with a packet loss rate of just 0.2% at 230400 bits per second.

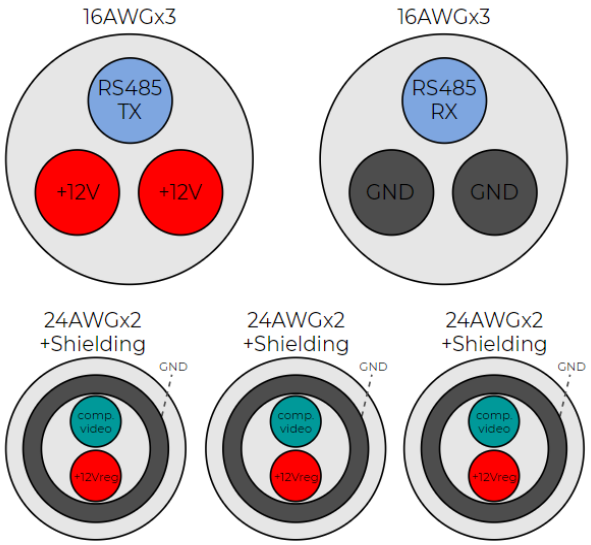


Fig 3.E - Tether Cross-section

3.5.2. - Wiring Scheme

Five tethers connect all components of the ROV to the control station. The ROV uses two cables for power and signal communication, with the remaining three for directly interfacing with the cameras. All tethers are detachable on the control station side through GX12, DC barrel, and RCA connectors. Two 16-gauge wires are used for power and ground, each delivering up to 22A with ROV voltage above 8.4V (30% voltage drop). The remaining two wires are used for high-integrity high-speed RS485 communication, running in half-duplex mode.

4. - Software Design

4.1. - Onboard

The software on the onboard microcontroller consists of two primary parts: a command parser and a device controller. As mentioned in 3.5.1, 8-byte packets of data, used as a command, are sent to the onboard microcontroller. These packets consist of a header, command, primary parameter, four bytes of data, and footer.

The command parser looks for valid header/footer combinations in 8 bytes, and if valid, parses the packet. When parsing the packet, it matches the packet against a list of commands. If any invalid data or bit-flips are detected, the command parser will abort parsing the packet.

Once packets are parsed, a state vector is updated, and the device controller moves the current real-world state closer to the desired state vector. While instantaneous movements are possible, voltage spikes and dips created by sudden movements are harmful for our components. Thus, a state system is used to control all system devices. All commands receive “OK” packets in return as feedback, and some commands receive specialized feedback, such as providing IMU data.

Command	Parameter	Data	Returns
0x10 - setMotorUs	Hardware motor (0x0-0x4)	uint16_t: microseconds	[0x1c] Motor Status [0x0a] OK
0x12 - setMotorCal	Hardware motor (0x0-0x4)	int8_t: percent	[0x1c] Motor Status [0x0a] OK
0x30 - getIMU	IMU component (0x15-0x18)	Any	[0x3a-0x3e] IMU [0x0a] OK
0x50 - setAutoReport	IMU component (0x15-0x18)	uint16_t: report speed	[0x0a] OK
0x0F - HALT	None (0x00)	Any	[0x0a] OK

Fig. 4.A - Important commands. More info on our GitHub (https://github.com/One-Degree-North/mate-rnd-ogbot)

4.2. - Control Station

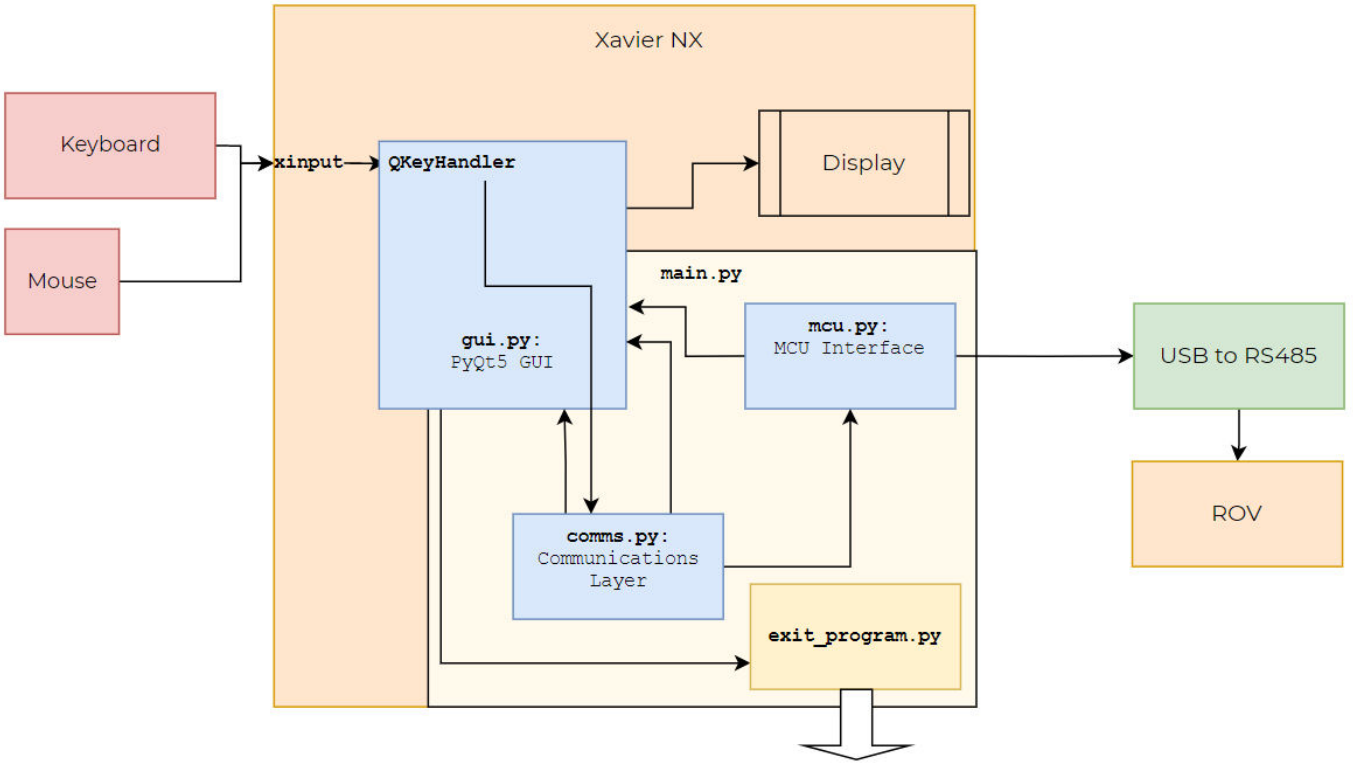


Fig. 4.B - Software diagram

The Graphical User Interface (GUI) uses the Qt framework, a visual application framework, through the PyQt5 library, a python interface for the framework. The primary modules used from PyQt5 were QWidget and QMultimedia. A QWidget represents the application window itself, with its widgets arranged in a grid layout. Group boxes were put on the right side of the display, featuring IMU, motor, and time information. The other parts of the grid were taken up by a sub-layout that displays all the cameras available. QMultimedia provided these camera widgets with its QCameraViewfinders being displayed to the end-user. Additionally, QMultimedia provided methods to capture images from the cameras and save them to the disk. PyQt5 was the optimal front-end library for our use-case due to its simple integration with Python and broad feature-set.

The user interface (UI) for our control software was an integral part of software development. Early on in development, we recognized the importance of a well-designed UI. Following this, we collaborated with our driver by spending lots of time collecting feedback to perfect our UI. Continually iterating and tweaking our UI allowed us to explore a variety of layouts and designs until a satisfactory interface to control the ROV was found.

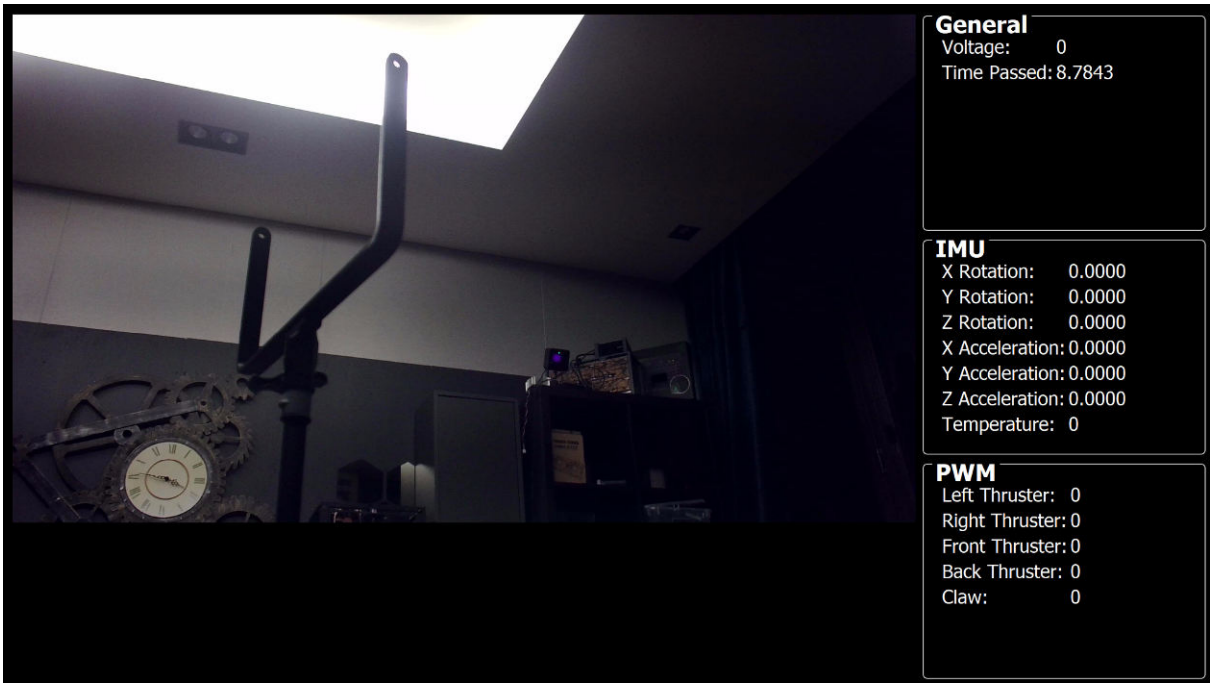


Fig. 4.C - Screenshot from our control program. Note that we are in single-camera and dark mode.

Initially, the keyboard inputs used the Pygame library because the programmers had more experience with Pygame. Despite being reliable in function, using two different graphics libraries was architecturally inefficient. Since PyQt5 was used to create the GUI, it was convenient to utilize the built-in keyboard event methods to detect keypresses instead of Pygame's keyboard inputs. Despite the lack of experience with using the PyQt5 library, utilizing the PyQt5 library shows the software team's willingness to experiment with different and more unconventional libraries and techniques.

As safety is the team's number one priority, the software department had to write a program that is able to efficiently and effectively close the program and stop the ROV from moving in the situation that something goes wrong. The program uses the sys Python module to safely close the GUI and the rest of the software subsystem. Before closing the software subsystem, the program causes the software subsystem to send a HALT packet, then it closes all communication with the microcontroller subsystem safely. Taking a systematic approach to terminating the system allows for a safe, efficient, and effective termination.

4.3. - Driver Controls

Controlling the ROV involves the keyboard predominantly. For the software department, it was crucial to design control schemes for it to be intuitive to most drivers after a quick read of a manual. To do so, the software team communicated with the driver to design ideal controls and a GUI interface. This allowed the driver to have an optimal experience when controlling the ROV, providing exceptional precision and minimal errors. We designed ideal controls by taking keyboard controls inspired by gaming, like WASD and arrow keys, to create a program that will be intuitive to control by most drivers while still providing flexibility as to how the bot can move.

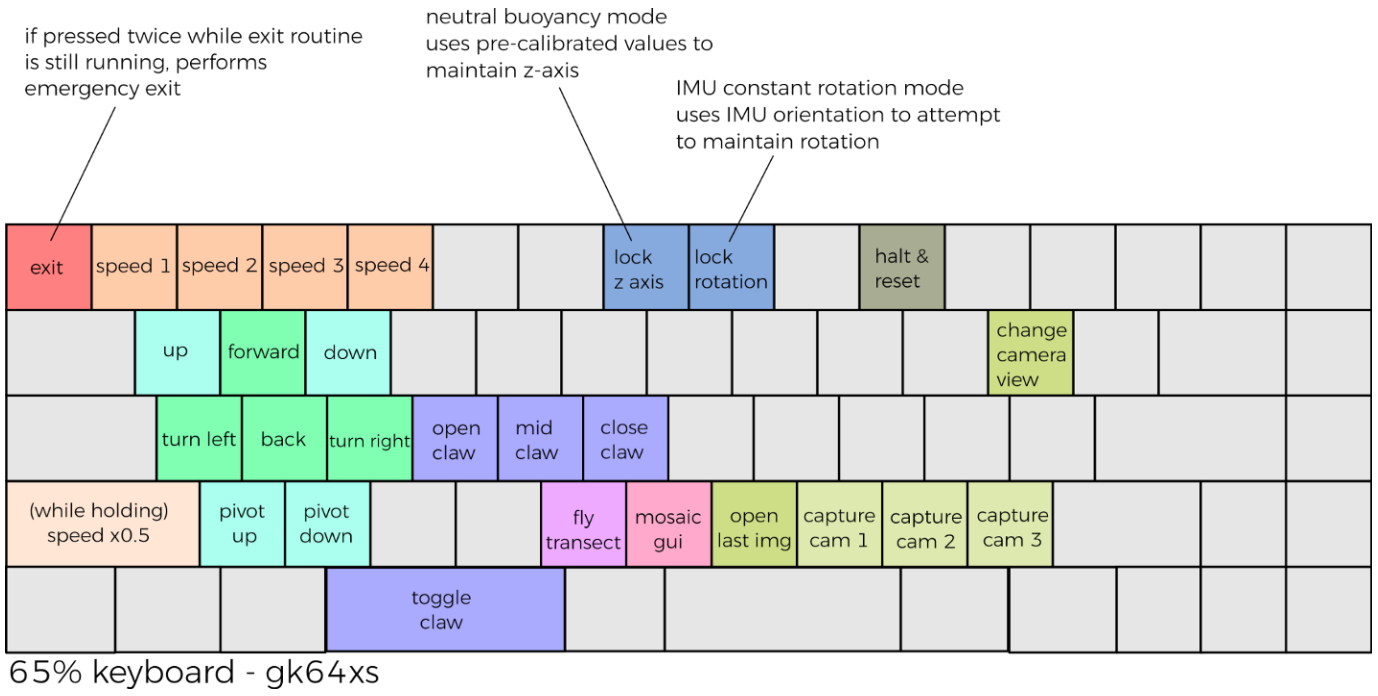


Fig 4.D - Keyboard input map

5. - Team Management

5.1. - Company Profile

5.1.1 - Team Composition

Team members were chosen for the project based on prior experience with MATE and the ability to dedicate to the project over the summer. As our team lacks any members who have previously properly competed in MATE, former MATE training team members and members from other robotics programs were chosen to fit effective roles. Members were assigned roles based on their skills and interests, and were split into two teams: the in-person mechanical/electrical team and the remote software development team. Our final team composition can be seen in the diagram below:

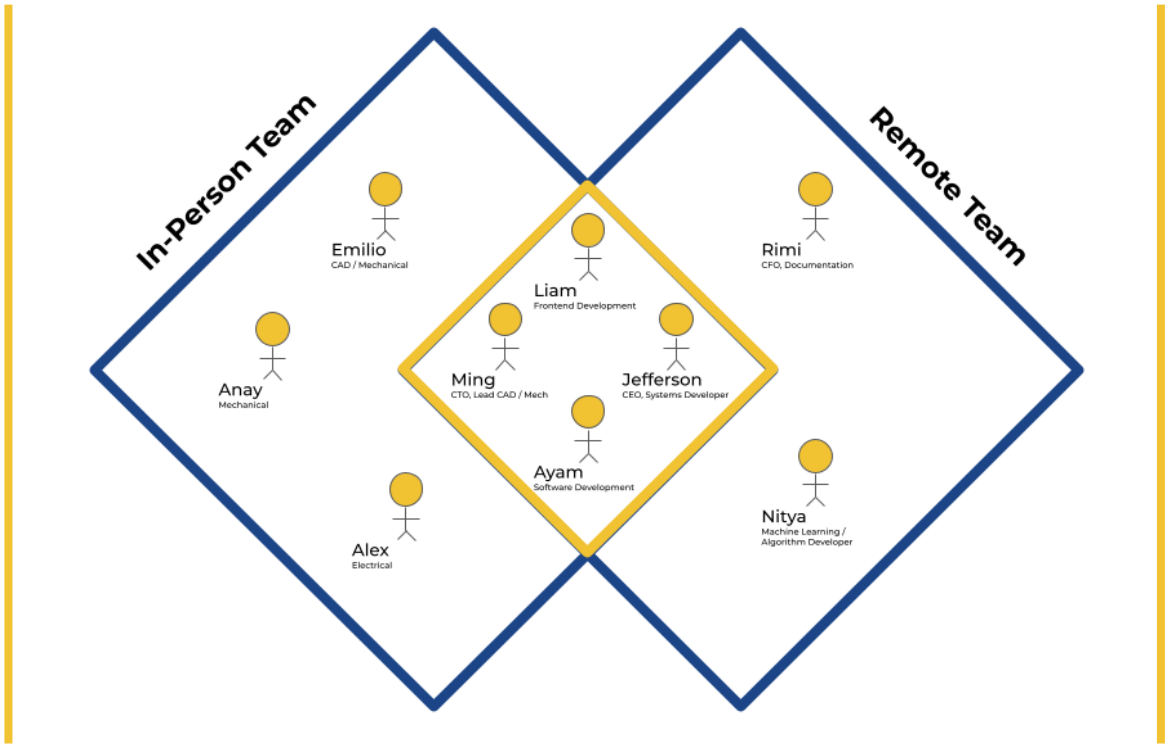


Fig. 5.A - Team Composition Diagram

5.1.2 - Project Management

Effective communication was achieved through a unified communication network over Mattermost (open-source alternative to Slack) in addition to fast communication over Discord and WhatsApp. Project management was organized through GitHub, with GitHub issues representing both hardware and software issues,

organized through tags and categories. Communication and collaboration through these means allowed our team to easily keep track of what needs to be done at all times, with members often working on our issues remotely from home.

5.1.3 - Problem Solving

As problems emerged throughout the process of assembling and programming the bot, a standard procedure was followed to debug, evaluate and troubleshoot the issue:

- Isolate the problem by testing all possible problematic factors one at a time
- Evaluate its root cause, through communication with other team members
- Decide on solution, balancing reliability, time consumption, and price
- Implement the solution, caring to make sure no unrelated issues pop up
- Evaluate the solution. If the solution is not desirable, create a new modified issue.

5.2. - Work Timeline

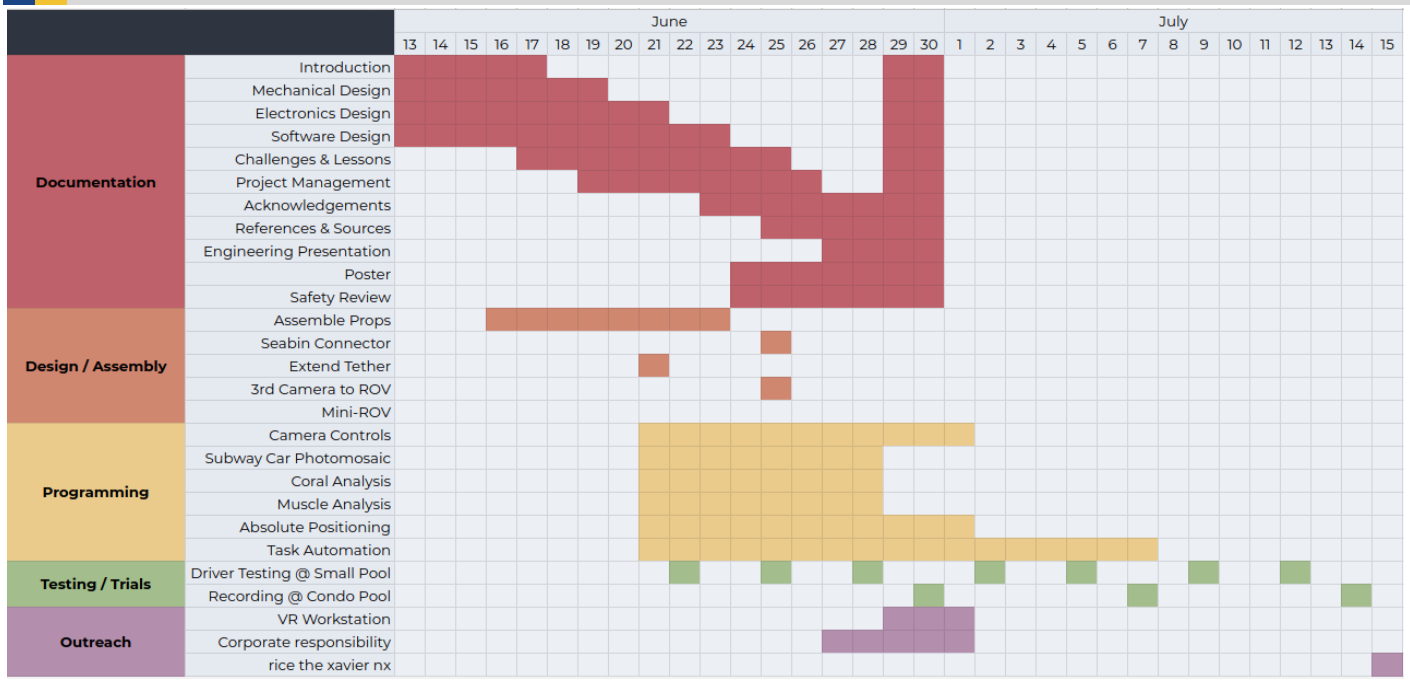


Fig. 5.B - Work Timeline Gantt Chart

5.3. - Safety Protocol

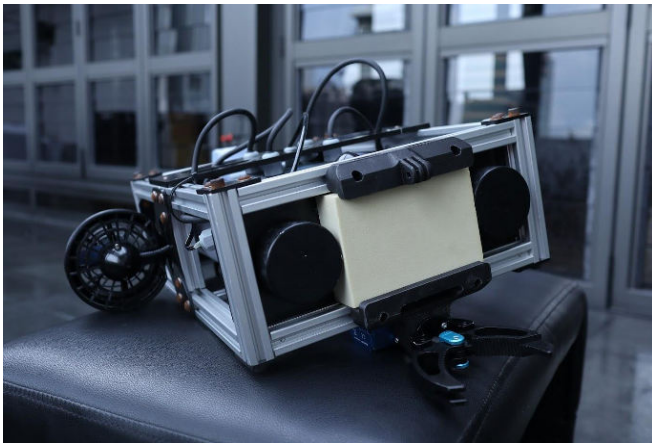


Fig. 5.A - (Left to Right) T100 prop guard, ROV resting on soft surface.

At team One Degree North, safety is our number one priority. We value the health and wellbeing of our employees before anything else. In order to maintain our safe work environment, employees are expected to adhere to rules and regulations set in place by MATE and by the company.

A towel is always nearby whenever the ROV is submerged in water in order to keep employees dry. Additionally, whenever pulling out the ROV from the water, we use the towel to dry it off before performing modifications.

Safety features have also been implemented on our ROV, for example IP20 rated propeller guards for T100 thrusters. In order to prevent accidental damage to the ROV, we also make sure to rest the ROV on a soft surface when not in use.

5.3.1 - Safety Checklist

- ☐ ROV
  - ☐ Electronics are properly sealed.
  - ☐ Structure is not loose.
- ☐ Control Station
  - ☐ Control station is at least 1m from the pool.
  - ☐ Control station has an anti-splash barrier in place.
- ☐ Driver
  - ☐ Driver alerts crew when power is given to ROV.
  - ☐ Driver does a test of all thrusters and auxiliary mechanisms.
- ☐ Crew
  - ☐ Towels are ready.

Fig 5.B - Pre-run safety checklist.

TASK	HAZARD	CONTROLS
1. FDM 3D Printing	1a. Skin contact with heated bed or extruder 1b. Extruder blockage	1a-I. Print 100% remotely 1a-II. Waiting for printer to cool down before handling 1b-I. Slice 3D prints appropriately
2. Soldering	2a. Skin contact with soldering iron 2b. Spilling solder 2c. Melting unrelated wires/surfaces	2a-I. Use proper soldering techniques 2b-I. Use proper soldering iron station and equipment 2c-I. Use proper soldering iron station and equipment
3. Handling 12V DC	3a. Short circuit 3b. Improper grounding	3a-I. Check for shorts before turning on 3a-II. Isolate sub-circuits 3b-I. Ensure proper grounding 3b-II. Ensure an EE is present
REQUIRED TRAINING		**REQUIRED PPE**
Soldering training Understanding of 3D printing principles		None

Fig. 5.C - Job Safety Analysis

5.4. - Budgeting and Cost

One Degree North Robotics Singapore American School	Period: Mar 23, 2021 to Jun 30, 2021

(All prices in SGD)		Projected	Budgeted	Actual
Mechanical				
Frame materials	Purchased	$100.00	$100.00	$74.23
Hardware for V-slot	Purchased	$50.00	$50.00	$31.70
Lock & Lock lunch box	Purchased	$10.00	$10.00	$8.99
PETG filament	Re-used	$75.00	N/A	N/A
Onboard Electronics				
T100 Thrusters	Re-used	$640.40	$750.00	N/A
Blue Robotics Basic ESCs	Re-used	$161.44	$200.00	N/A
Adafruit Feather M0	Purchased	$19.99	$20.00	$19.99
Adafruit BNO055 breakout	Re-used	$33.63	$50.00	N/A
Power regulators and devices	Purchased	$62.92	$50.00	$62.92
Tether cabling	Re-used	$250.00	$300.00	N/A
Cameras	Purchased	$109.99	$240.00	$109.99
Control Station Electronics				
NVIDIA Jetson Xavier NX	Re-used	$350.00	$350.00	N/A
14" Display	Re-used	$200.00	$200.00	N/A
Computer accessories & cables	Re-used	$80.00	$80.00	N/A
Camera capture interfaces	Purchased	$78.54	$80.00	$78.54
Assorted connectors	Re-used	$20.00	$20.00	N/A
Total Expenses		-$2,241.91	-$2,500.00	-$386.30
Income				
Singapore American School		$2,000.0	$2,000.0	$319.30
Personal funding		$0	$0	$67.00
Total		**-$241.91**	**-$500**	**0**

6. - Acknowledgements

Our team is grateful to the Singapore American School's Robotics Program, as well as the school's administration and financial teams for providing us with the opportunity to participate in MATE. Although we were not able to participate in person this year, we must thank SAS for allowing our relentless requests to enter the school to pick up items, even over the summer, and for providing us with the financial support necessary to complete our ROV.

Additionally, we are grateful for the MATE II foundation for hosting 2021's MATE season virtually, and allowing a telepresence option. MATE robotics has been one of SAS' core robotics programs for the past decade, and we thank you for the opportunity to participate, even during a global pandemic.

We would also like to thank our mentors, Mr. Millar and Mrs. White, for sponsoring our activities and helping us communicate with MATE representatives. The program could not exist without amazing mentors like you.

THANK YOU!

7. - References & Appendix

Sources:

"Buoyancy Calculator". Omnicalculator.Com, 2021, https://www.omnicalculator.com/physics/buoyancy. Accessed 30 June 2021.

“Development of a Low-Level Control System for the ROV Visor3.” Rúa, Santiago, and Rafael E. Vásquez. International Journal of Navigation and Observation, vol. 2016, July 2016, pp. 1–12. DOI.org (Crossref), doi:10.1155/2016/8029124.

"The Buoyant Force Does Not Get Smaller As You Sink (Article) | Khan Academy". Khan Academy, 2021, https://www.khanacademy.org/test-prep/mcat/physical-processes/fluids-at-rest/a/the-buoyant-force-does-not-get-smaller-as-you-sink. Accessed 30 June 2021.

Images:

"BF7-BF21DEMONTIERT." *reichelt elektronik,* https://www.reichelt.com/de/en/cable-gland-pg11-5-10-mm-pgbf-11-p126188.html

"Foam-16." *Blue Robotics*, https://bluerobotics.com/store/watertight-enclosures/buoyancy/float-r3318-r1/

Lock & Lock Square Classic Food Container. *EAMart Singapore*, https://www.eamart.com/product/list/best-sellers/Lock--Lock-Square-Classic-Food-Container-870ml-13149

Appendix i: Full SID