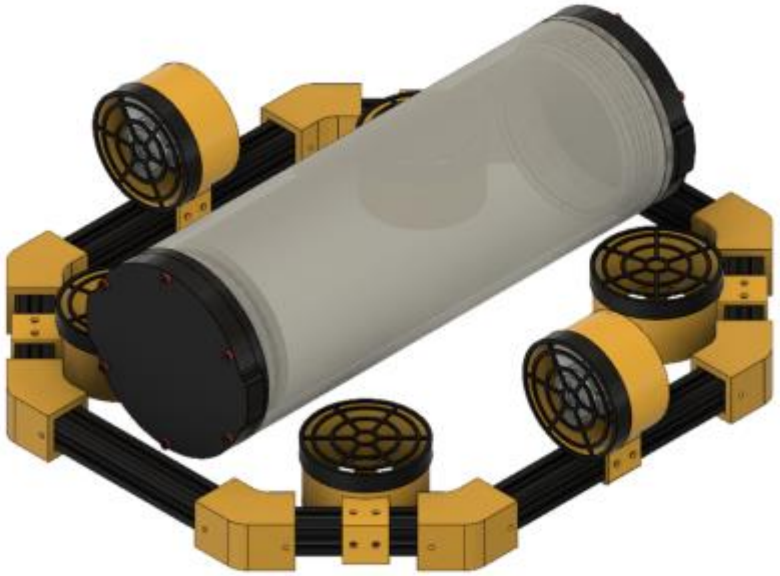
AQUILA INTEGRATED SOLUTIONS

AMERICAN INTERNATIONAL SCHOOL
HONG KONG



MEMBERS

Ronald Ng, Class of 2017
Chief Executive Officer

Brian Mellon
Mentor

Eunsup Kang
Mentor

Jeffrey Kam, Class of 2017
Chief Technology Officer &
Head of Mechanical Engineering

Tony Korari, Class of 2019
Senior of Mechanical Engineering

Tony Ahn, Class of 2017
Junior Engineer

Archit Kithania, Class of 2017
Head of Electric Engineering

Tom Li, Class of 2017
Senior of Electric Engineering

Zian Lin, Class of 2019
Junior Engineer

Steven Wong, Class of 2017
Head of Software Engineering

Tim Chen, Class of 2019
Senior of Software Engineering

Sungrae Kim, Class of 2017
Junior Engineer

Table of Contents

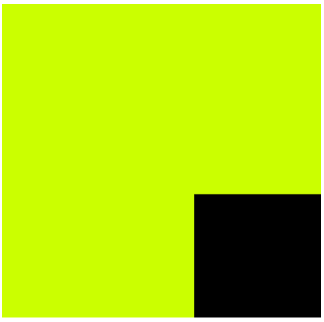
Abstract

The ocean covers 70% of the Earth's surface and has contributed so much to maintaining life and providing the resources we need. But despite what the ocean has done for us, we have only a limited amount of knowledge about it. To date, we have investigated less than 5% of the ocean, meaning that there is so much more to learn and discover. To tackle this problem, we have created a company for talented and passionate students to build ROVs: Aquila Integrated Solutions. We hope to raise awareness to the marine field and help students develop the skills necessary to enter careers in technical fields by joining the MATE competition, and on that note, we are very proud to present to you the Aquila II.

The Aquila II underwent many changes since last year. We completely changed our control system and created an auto-balancing system to improve the maneuverability of the ROV. We also introduced a VR system for the camera, allowing us to achieve a wider viewing perspective from the ROV. With so many new features, the Aquila II is ideal for exploring the deep and mysterious ocean.

Theme

Safety is one of the biggest concerns in our club. Therefore, we designed our team's theme with that in mind. This year, we based our team's theme color around lime yellow and black to highlight and accentuate our ROV while it is underwater. The contrast between the lime yellow and black creates sharpness in our ROV's appearance to ensure that everyone can see our ROV underwater- thus preventing accidents from happening. The color combination also creates higher visibility in the warning texts on the ROV, which immensely improves the safety aspect of our ROV.



Design Rationale

We based the design of the ROV on the core of our system – auto-balancing. With auto-balancing in mind, we built our ROV so that each corner could fit one bilge pump while still having enough room in the middle for dry housing. In the end, we decided to assemble our ROV in the following way (shown in Figure A.1).

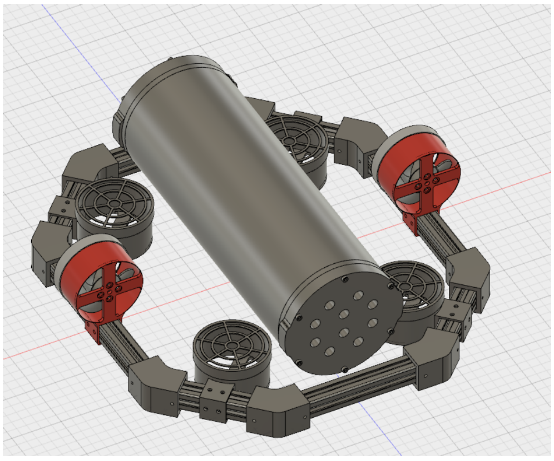


Figure A.1: A 3D drawing of our ROV

The four up and down motors on the corners ensure an equal distribution of the weight of the ROV so that the ROV will not tilt on any one side. Also, our main source of buoyancy - the watertight enclosure - is placed in the middle so that the ROV will not be buoyant on only one side, but also on the other.

Mechanics

Frame

This year, we decided to use aluminum for our frame because its sturdiness and feather weight properties that it brings to the ROV. We specifically bought a type of aluminum metal that makes it easy for us to implement components onto (see figure B.1).

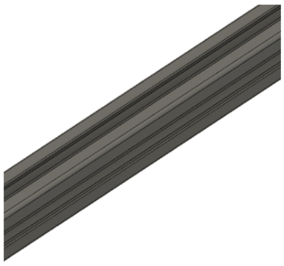


Figure B.1: A 3D drawing of the aluminum

We carefully treat each and every aluminum piece used for the ROV with professional equipment provided by MakerBay. With the equipment there, we were able to trim the size of the aluminum to exactly what we wanted and also grind the aluminum so that would be no sharp corners, preventing anyone from potentially getting hurt from it.

Our frame is designed using pieces of aluminum of 3 different lengths - 200mm, 150mm, and 105mm. This unique design allows us to form an octagon shape that is easy to manipulate.

We also 3D printed all of our connectors (Found in Appendix B.1), and it is effortless to take and put them back on. Doing so allows us to change the shape of our ROV at any given time during the process of making the ROV, thus giving us the ability to test different designs and adapt each the change.

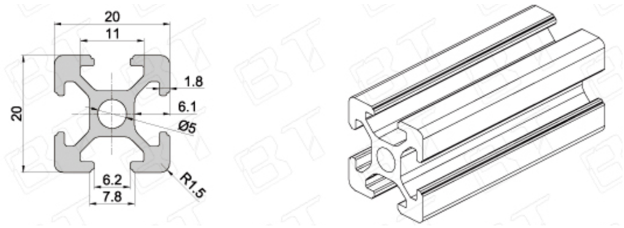


Figure B.2: The detailed specifications of the aluminum

Dry Housing

This year, we chose to use the watertight enclosure produced by BlueRobotics (see Figure B.3 and B.4) instead of making it ourselves because of the sturdiness and the maneuverability that it provides. Compared to other watertight enclosures on the market, the BlueRobotics watertight enclosure allows us to change the wires that are going in and out of the tube, which helps when we are testing the ROV. Also, their watertight enclosure ensures that it is 100% waterproof because of the epoxy in the entrance of the wires and the vacuum created by the tube.

Figure B.3: A 3D drawing of the watertight enclosure

Figure B.4: A photo of the watertight enclosure

Tether

Learning from our previous experiences in this competition, we realized that the tether of ROV is an important factor to consider when building our ROV. This year, with meticulously designed wire management, we were able to decrease the number of wires going down from land to water and properly tether the wires. Figure B.5 shows how those wires are organized.

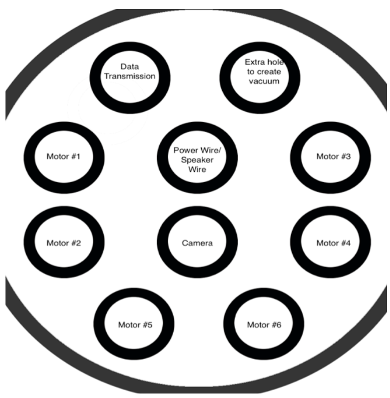


Figure B.5: A visual diagram of the tether organization

Instead of using electronic tape (which we used previously), we used Velcro straps (shown in Figure B.6) to ensure that the wires do not tangle, saving us precious time from untangling the wires every day.

Figure B.6: Picture of the Velcro straps that we bought

Propulsion

Mobility was a key idea that we gravitated towards while deciding the outline of the ROV. To maximize movement of the ROV, we changed six 500GPH bilge pumps into thrusters that utilize propellers that we have 3D drawn.

One of the biggest decisions that we had to make this year was deciding between using bilge pumps and brushless motors. After a series of trials and errors, we concluded that we should use bilge pumps, as although brushless motors provide powerful thrust and are a lot smaller, they draw way too much current to be used for the competition. The brushless motors are connected in a parallel circuit, and each one is capable of pulling a maximum of 14 amps, which results in a total maximum of 84 amps on just propulsion, exceeding the 25 amps fuse safety limit. On the other hand, each bilge pump only pulls 2.2 amps maximum, resulting in the decision of using bilge pumps.

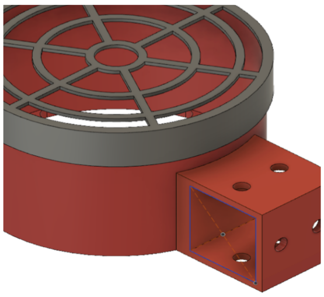


Figure B.7: A 3D drawing of the motor mount

Each bilge pump used is mounted using a mount we 3D drew and printed (shown in Figure B.7). This type of mount allows us to move the location of the bilge pump to anywhere within the frame, allowing us to change what we need while we're making the ROV. It also lets us take out the motors for testing when needed. The propeller of the system is also specially designed to maximize maneuverability, as the spiral shape boosts the amount of thrust provided by each motor (shown in Figure B.8).

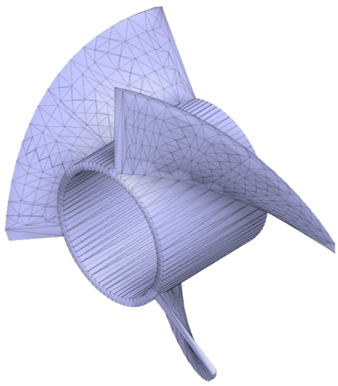


Figure B.8: 3D drawing of the propeller

Gripper

This multipurpose gripping tool was made by 3D printing its parts and using various screws and nuts to assemble it. It is powered by a servo that rotates the two claws with a gear to mimic a gripping motion. Inside the gear is an aluminum accessory which is attached to the gear, so that the servo's output axis does not wear down the plastic. A prototype of the 3D printed gripper is shown in Figure B.9. Also, we designed the gripper so that it could rotate on two different axis, which allows the gripper to rotate around objects and find a surface it can attach itself on.

Figure B.9: A picture of the 3D printed gripper

The servo is controlled by three wires and is connected to an Arduino Mega. The first version of the code only used keyboard inputs as the controller, but we later adapted it for wireless PS3 Dualshock 3 controllers. There are two modes: opening the claws all the way and a permanent attempt at closing the claws. The two modes give the claws a very strong grip on an object but does not allow it to grab fragile objects. The mission calls for picking up plastic parts that will probably not crush under the pressure of the claw. However, an unused line of code can be made active again to allow for the picking up of fragile objects.

The missions this year require the extraction of some plastic cylinders that mimic waste in the ocean. Many points are scored this way, and the gripper can rotate to pick these parts up. A hook was also added onto the ROV to assist in moving objects. Adding the hook follows closely with what the mission will require, as we need to move an object with a rope that is easy to hook. It also serves as a backup for the gripper, in case of water leakage in its electronics or other problems that may arise.

One decision we had to make with the gripper was whether to use pneumatics or a robotic gripper. We ended up going with a robotic gripper because it was stronger and more reliable. In our tests, the pneumatics usually failed more often when compared to the 3D printed gripper.

Electronics

Main Cameras

As with any other ROV, visuals were essential for the Aquila II. Without vision from the cameras, the team would not be able to navigate the waters and complete any tasks the ROV was purposed to do. With this in mind, we took utmost time and effort to precisely account for every possible mishap, and we worked to the best of our abilities to avoid them. The cameras are also crucial for monitoring whether all parts of the ROV are fulfilling their purposes. The ROV is equipped with three cameras: two front facing cameras and a rear camera.

The cameras used are the Spy-Hole B/W Camera K701. This camera provided the advantage of customization, as the parts were readily available and the size was easily adjustable in the ROV tubing. Since this camera requires a constant voltage of 5V and the power supply provided would be a varying 12V, a circuit was designed to regulate the voltage and provide the camera with the constant voltage it needed (see Figure C.1).

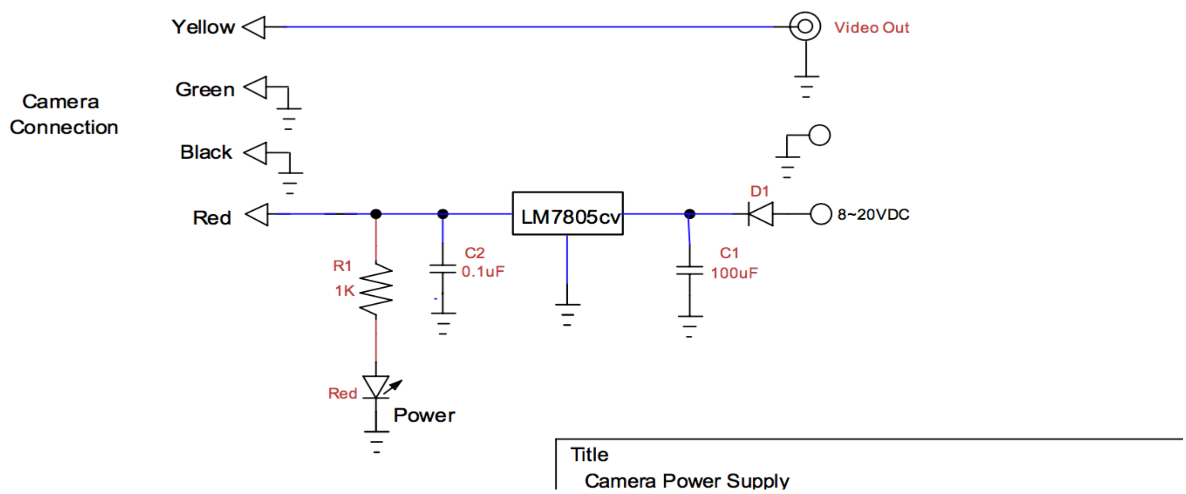


Figure C.1: The schematic of the camera circuit

The K701 camera has four wires connected to it:

1. Yellow: Sends the camera signal feed to a monitor
2. Green: Provides grounding for the camera equipment
3. Red: Positive terminal for the voltage supply
4. Black: Negative terminal for the voltage supply

The positive and negative wires undergo a circuit to ensure that a constant 5V is provided to the camera. An LM7805 voltage regulator is used to regulate the voltage of 5V, and two capacitors, one of 100μF and one of 0.1μF, were used to minimize voltage spikes and keep the voltage steady. All the circuits are fixed into a designated camera box.

The cameras themselves had to be modified to ensure waterproofing, which was achieved by 3D printing 9cm camera tubes to fit the camera perfectly. The tubes were then fixed with two step epoxy and then attached to their designated spots. All the waterproofing and circuitry ensure that the ROV can rely on its visuals to make its way towards victory. A finished product is shown in Figure C.2.

Figure C.2: Picture of the waterproofed camera

Control Box

The design for the control box of the Aquila II is based on the principle of simplicity, so we tried to make the box as compact and sleek as possible (see Figure C.3 and C.4). There are 5 main parts to the box: an Arduino Uno, Arduino Uno USB 2.0 Host Shield, Ethernet port for the VR circuit, Ethernet port for the underwater circuit, and a mechanical switch. The Arduino Uno is the brain above water that processes all the information and performs all the calculations before transmitting data to the underwater Arduino. We chose to use an Arduino as our computational system because it is open source, allowing us to find ample resources online while researching for solutions, and hardware compatible, which makes programming the hardware (i.e. PWM modules, Servos, and gyroscopes) much easier. The USB 2.0 Host Shield allows our Bluetooth PS3 Controller to communicate to our Arduino and control our ROV through a USB wired connection or a Bluetooth connection. The 2 Ethernet ports are for data transmission. The left port allows connection to the underwater circuit using an I^2C protocol, and the right port allows connection to the VR head tracking control circuit

Figure C.3: The control box in closed view

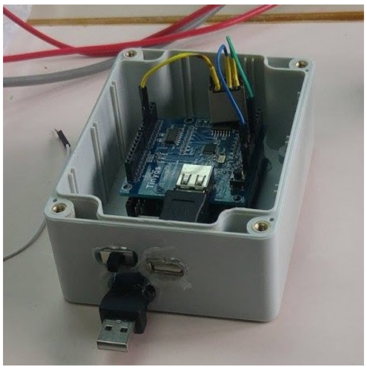


Figure C.4: The control box in open view (see Figure C.10) using Software Serial. Our team chose to use Ethernet ports because it is a standardized port for data transmission for web data, so our whole system could have a consistent port type for data transmission. Also, we chose to use Ethernet ports because we had a limited number of wires that could go into the enclosure, so the 8 small and low resistance wires inside a CAT 6 Ethernet cable allow us to have all the data using only 1 tether hole (see Figure B.5 below). The mechanical switch is for switching between VR head tracking control and PS3 directional buttons control for camera orientation.

System Integration Diagram

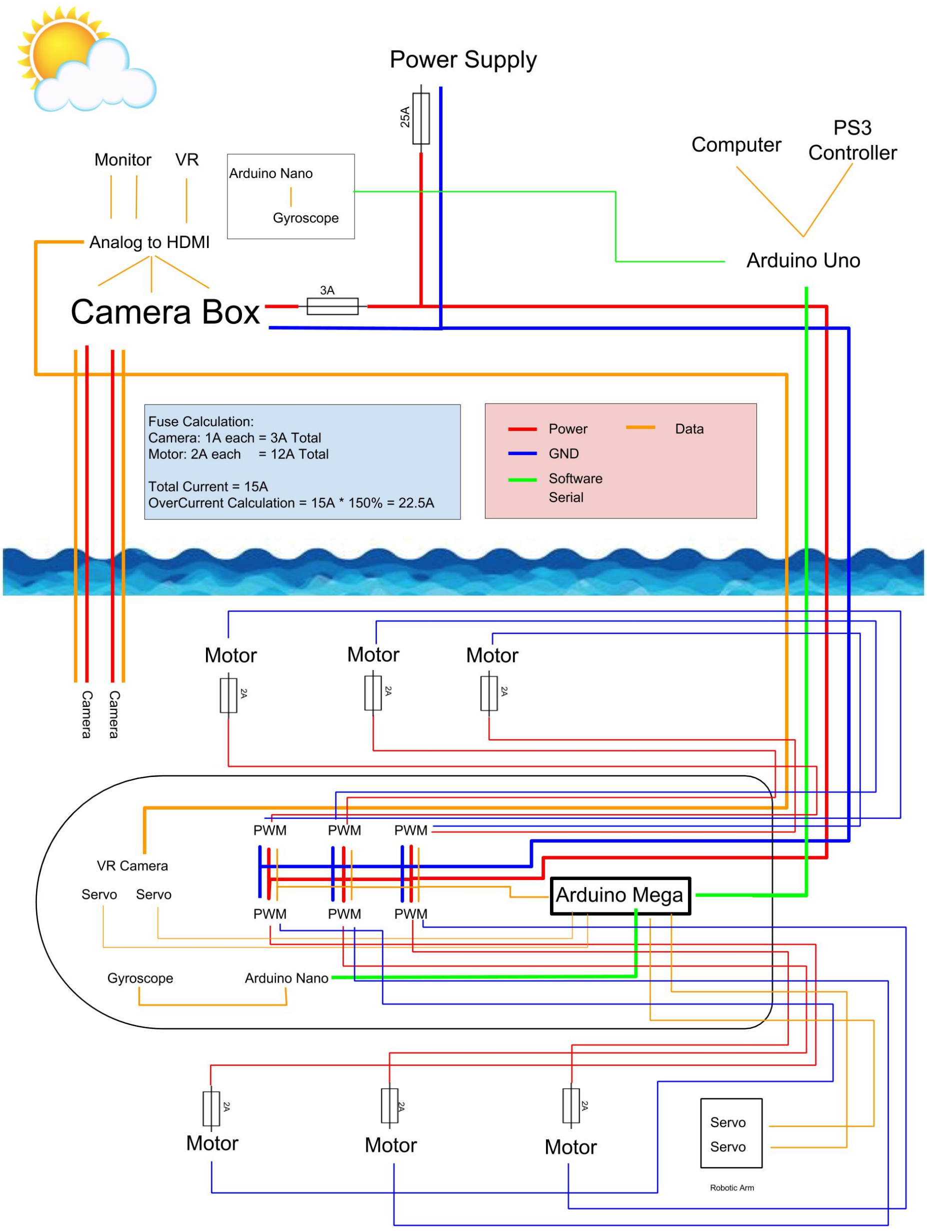


Figure C.5: System Integration Diagram

Pulse Width Modulation for Speed Control

Earlier this year, we decided to use brushless motors, so an Electronic Speed Control (ESC) was used to control the speed of the motors. However, after some calculations on the current draw of our system using brushless motors, we realized that it would exceed the 25 amps limit. Therefore, we decided to use bilge pumps and Pulse width modulation (PWM) modules to control the 6 motors mounted to the ROV. Unlike the ESC, the PWM speed control is achieved by modulating the frequency of on and off to control the speed of the motor. If one wants to move at 70% speed, one needs to set the width of the square wave at 70% for on and 30% for off. Learning from our experiences from the previous years, we paid closer attention to the PWM specifications before purchasing them (specifically the max current draw the PWM can handle). We purchased three types of PWM modules - ALFS PT01A, CCM2, and Risym XY-MOS (shown in Figure C.6), which can all support a maximum current greater than 3 amps, and allow us to determine the best fit for our ROV experimentally. CCM2 was chosen in the end because it performed the best in our experiments of thrust testing, and the control of it was most similar to that of our previous PWM modules by having switches for direction and analog data for speed. Also, using this PWM module eases our troubleshooting process as we can directly determine which PWM module is faulty and replace that single module, as each module corresponds to only one motor.

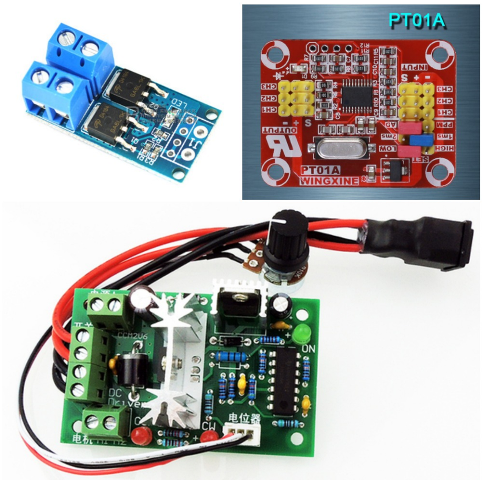


Figure C.6: The 3 PWM modules purchased (top left is Risym XY-MOS, top right is ALFS PT01A, bottom is CCM2)

Data Transmission System

The data transmission system is the spinal cord of the ROV and is responsible for all the signal transmissions between the underwater watertight enclosure and the above-water control box and VR control. This data transmission is achieved through the use of an I^2C protocol and a 6 meter Ethernet wire connecting the underwater circuit (see Figure C.8 below) to the above-water control box (see Figure C.3 above). However, I^2C is only optimized for distances less than 2 meters, so our system would not work under normal conditions. After some researching, we have realized that we could enhance the signals transferred by adding pull-up resistors. We have empirically determined (shown in Figure C.7) that the optimal resistance for the pull-up resistors is 1k ohms resistance for the SCL pin and 10k ohms resistance for the SDA pin. The data transmitted through a CAT 6 Ethernet cable, with its low resistance, will connect to the I^2C port of the underwater Arduino Mega to perform the actions accordingly. We also chose to integrate everything into a single circuit board for the underwater system because doing so will reduce the chances of data transmission failure, and it will then be easier to troubleshoot in regards to the wiring of the circuit.

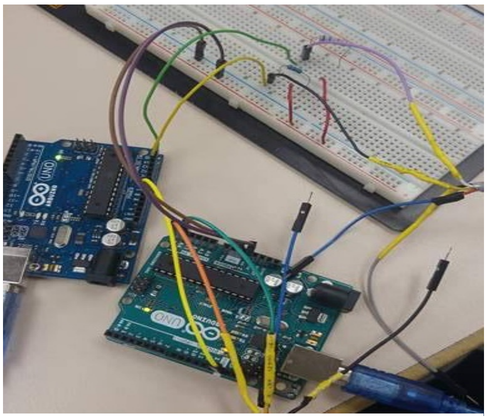


Figure C.7: Testing the data connection system

The data being transmitted from the control box will be in this format: process mode, arm angle X, arm angle Y, camera angle X, camera angle Y, left speed, right speed, top left speed, top right speed, bottom left speed, and bottom right speed. This string of data, sent from the control box, will be parsed into each respective individual variable by the Arduino Mega, and the corresponding actions are executed according to the values of each variable.

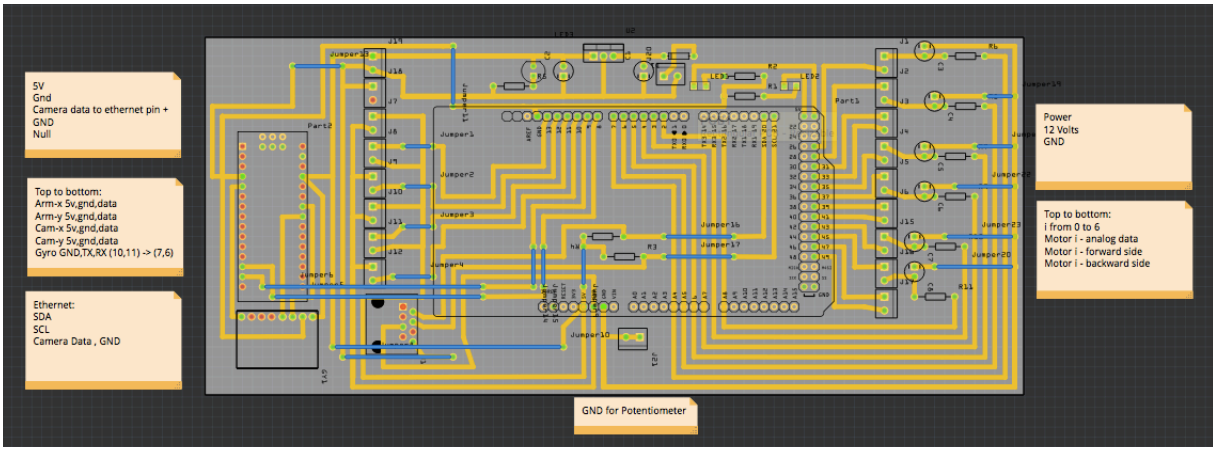


Figure C.8:The PCB view of the underwater circuit

VR Camera Electronics

Our team is using a VR box (see Figure C.9), one that allows you to put your phone inside, to make the VR system. On the VR box, we had to attach the head tracking

Figure C.9: The VR Box we are using

circuit (see Figure C.10) in a specific orientation, as we decided on a specific orientation of the gyroscope (the MPU-6050) that makes it possible to determine whether the gyroscope is being turned left or right. This is due to the problem that some orientations give the same value for both directions (such as 90 and 90 on left and right instead of -90 and 90). The chosen orientation is shown in Figure C.11. There is also a led that will light up to indicate when the gyroscope is stable, as when the program initially boots up, the values of the gyroscope fluctuate and require waiting before they stabilize. The head tracking circuit is designed as compactly as possible to ensure that it won't be too bulky for the VR box user. The gyroscope data is sent to the connected Arduino Nano, and the Arduino uses an Ethernet port to transmit the data to the Arduino in the control box using Software Serial, which will then be transmitted to the ROV through another Ethernet cable. Inside the underwater enclosure, there are two servos for the x and y axis connected to the VR camera rotational base that move according to the gyroscope values. The small camera is connected to the rotational base, and its video feed will be passed back to the control box using a wire inside the Ethernet cable. The video feed is then transferred to the phone inside the VR box through an AV cable and an analog to mini HDMI converter in between.

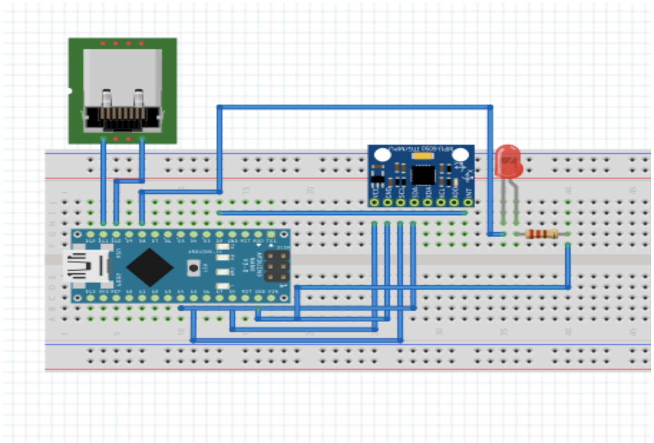


Figure C.10: Our team's VR head tracking circuit in breadboard view

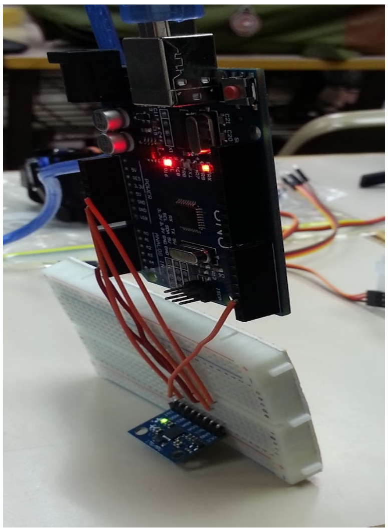


Figure C.11: A picture of our VR head tracking circuit placed in the correct orientation

Software

PS3 Control System

After experimenting with different types of controls, we concluded that an all-in-one system is the most efficient way to control the ROV and its related components. For all the choices of an all-in-one controller, a PS3 controller was considered the most optimal choice by our team as the Bluetooth functionality prevents the hassles of a wired connection.

There are 3 main modes for the ROV that can be selected from the controller. Mode 1, also known as the testing mode, will run through 3 testing stages for the motors: full-power forward, full-power backward, and stationary. This mode of control was added to our program because it could act as a preliminary test for the core functionalities of the ROV. With this mode implemented, we could fix the problems before we submerge the ROV underwater. Otherwise, we might be severely damaging the ROV without knowing it. Mode 2 is the stop mode of our ROV, which means when it is activated, all motors and controllable components in the ROV will come to a halt, and all control inputs will be blocked (except the square button on the PS3 controller to deactivate the stop mode). The stop mode was developed as a last option in case of emergency.

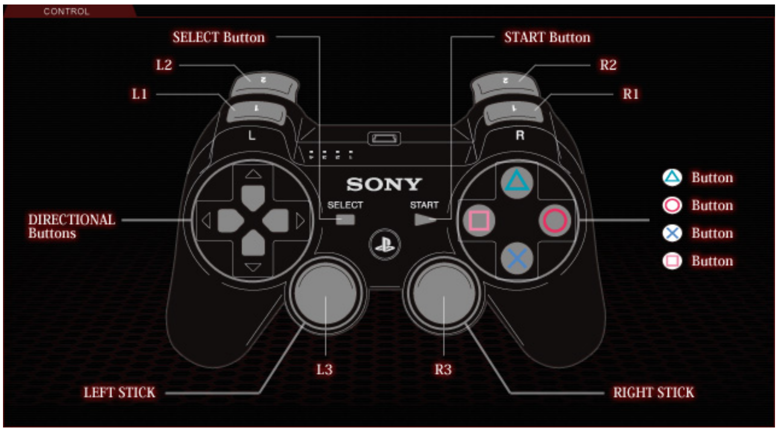


Figure D.1: A picture showing all the buttons' names of a PS3 controller

The final mode, Mode 3, is the active mode, which means that the underwater ROV can be directly controlled by the PS3 controller. There are 4 parts of the ROV that can be controlled by the PS3 controller: vertical speed (speed of moving up and down), horizontal speed and rotation, camera orientation, and the robotic arm. The vertical speed is determined by the pressure of the press on the L1 and R1 buttons (see Figure D.1). The pressure of the press will be returned as a number between 0 to 255 (0 = no pressure and 255 = max pressure), and the values will be constrained within the range from 0 to 240. These new values will be the PWM value for the 4 motors that governs vertical movements. The ideal values should be 255; however, the values are constrained to 240 as its maximum because we are taking into account that our PWM may not reach maximum efficiency. For the horizontal movement, our team is taking a simplistic and intuitive design. The two joysticks on the PS3 controller will be used to control the horizontal movement, with the left and right joystick controlling the left and right motor responsible for planar movements respectively. Our team disregarded the analog values of the x-axis of the joystick because only the magnitude of the y-axis is used in calculating the movement speed. We designed the program in a way such that if one wants to move the ROV forward, one can move both the joysticks forward; or if one wants the ROV to rotate left, one can perform a combination of having the left joystick backward and right joystick forward. Each joystick has a value ranging from 0 to 255 in the y-axis, so we divided the range in half such that values above or below the midpoint mean moving forward or backward respectively, and values within ± 10 of the midpoints mean stationary.

There are 2 different modes to control the camera inside the underwater enclosure: Virtual Reality (VR) Mode and PS3 Mode. The two modes cannot run simultaneously, and therefore we installed a physical switch onto the control box for switching between the two modes. The camera's orientation is controlled by the directional buttons on the left side of the controller with each button press corresponding to a change in camera orientation in that button's direction.

For the control of the arm, L2, R2, Select, and Start buttons are used to give the commands to the arm's servo. The arm consists of the gripping servo and the base rotation servo. Hence, we need to use 4 buttons to gain full control. The mapped functionality of each button is: L1 for closing, R1 for opening, SELECT for rotating left, and START for rotating right. Since the motion of the arm is just open or close and rotating left or right, the analog values returned will have no effect on the turning of the servos; as long as the value is not equal to zero, the servo will perform the actions of the button pressed. A visualized diagram of the general functions of each button is shown in Figure D.2.

Figure D.2: A visual diagram of the functionality of each button on the PS3 controller

Graphical User Interface

We decided to create a GUI so that while we were running and controlling the ROV, we would also be able to visualize different aspects of it simultaneously. The GUI program is built using Processing, as Processing is a relatively simple yet powerful way to create visuals based on Java code. When the program is first started, it will display “Please wait” with a loading circle to wait until the connection between itself and the Arduino is set up, as shown in Figure D.3a. Then, when it is ready to begin, there will be a button for the user to press, as shown in Figure D.3b. The button is red and changes to green once hovered on. Once the button is pressed, the program will display the data, as shown in Figure D.3c. The current mode of the ROV is shown on the top of the GUI so we can make sure we are in the right mode. The x and y angles of the camera and the robotic arm are shown on the right side in text form so that we can also check whether they are in the right positions or not. The left and right motor speeds (horizontal movement) are shown separately in a gauge similar to a car’s speedometer in order to show the speeds of both motors, with negative indicating that it is going backwards. To create the gauge display, we mapped the motor speeds from -255 to 255 to an angle from π to 0. We then converted the points from polar to cartesian, as it is easier to find the point in polar due to the input being an angle, but it needs to be graphed in cartesian. Next, we converted the cartesian point into the plotting system that Processing uses, as there are slight differences in its plotting format. The last value displayed is the motor speed for vertical movement, which is shown in a bar graph format that indicates the speed of the top left, top right, bottom left, and bottom right motors. These values are all sent to Processing through the Arduino, and then our program updates the GUI based on those values. With our GUI in hand, we found it much more straightforward

Figure D.3a: Loading screen

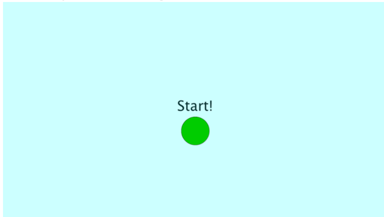


Figure D.3b: Start screen

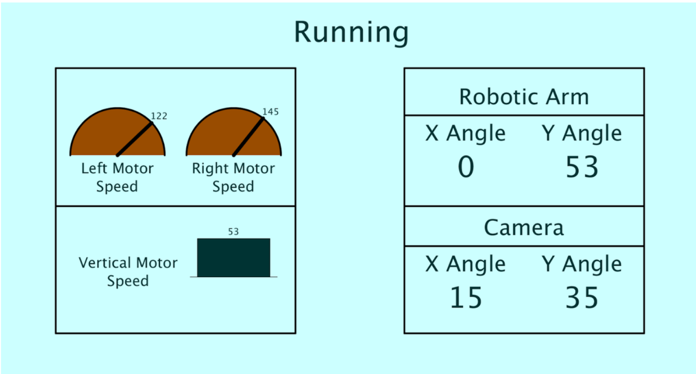


Figure D.3c: Active Screen (when the PS3 control code is running) to see how the ROV is currently moving as well as the what positions the camera and robotic arm are at. The 3 different screens are shown in order below.

VR Camera Head Tracking

Rather than using a fixed camera position and having to move the ROV in order to change the camera angle or using a joystick to control the camera angle, we decided it would be more advantageous to control the camera angle using our own heads' movements. By controlling the camera with our head movement and seeing the live feed through a VR glasses, it makes it possible to feel as if we are underwater and makes it more intuitive when maneuvering the ROV. This was done using a gyroscope (MPU-6050) to detect the head orientation and mapping it to the servo so that it can rotate the camera. The software records the initial orientation of the user and saves it as the center. Since the gyroscope measures from -180 to 180 degrees in the x direction, while the camera only has 180 degrees of rotation in the x direction, we use the initial orientation to determine which part of the -180 to 180 degrees from the gyroscope to use, as shown in Figure D.4. It then rotates the camera based on that initial orientation and maps the current orientation to the angle that the servo needs to turn to. The camera is mounted on the servo and therefore turns with it. The same approach is used in the y direction but with different degrees.

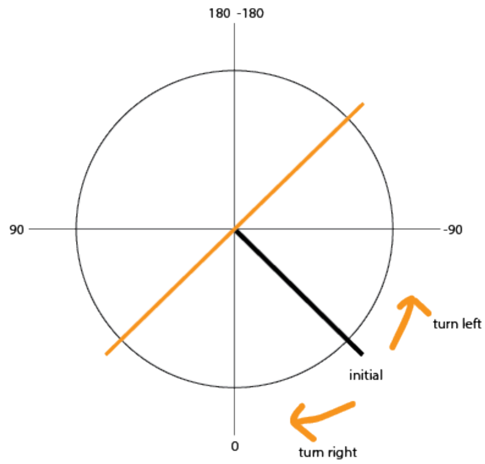


Figure D.4: A visual representation of the head tracking code

Auto-Balancing System

The auto-balancing system is the most important part of our ROV, as every other part of the ROV is built around this. A simple way to approach this problem is to provide thrust proportional to the error from the optimal yaw and pitch values from the gyroscope. For instance, if the value is lower than the optimal one, our system can activate the motors to provide upward thrust to counter such imbalance. However, this creates one huge problem - oscillation of ROV due to overshooting. To solve this, we must employ a Proportional-Integral-Derivative (PID) algorithm. Taking into account the previous cumulative errors (the integral section) and predicting future errors (the derivative section), this algorithm can smoothen the oscillation around the optimal value and prevent overshooting or undershooting so that the ROV can eventually stabilize. The core equation for the PID algorithm is shown in Figure D.5. Each section of the equation requires a constant, and each constant is determined through the Arduino PID Autotune library developed by Brett Beauregard. For the hardware, the gyroscope responsible for auto-balancing is integrated onto the main underwater circuit board to make sure that the optimal values will be same in all situations, as the optimal values of yaw and pitch are to be calculated and inputted into the code empirically.

$$u(t) = K_p e(t) + K_i \int_0^t e(q)dq + K_d \frac{d}{dt} e(t)$$

Equation D.5: PID algorithm core equation

Distance Calculation Program

As one of the missions require participants to “Determine the distance from the high-risk container to the other three containers”, it is crucial to figure out a simple yet effective way to find out such distances. To do so, our team has devised a distance calculation program using JavaScript and jQuery. The program takes an image and 4 mouse coordinates (inputted through mouse clicks) as inputs for the calculation. (See Figure D.6) The user must first have an object of known length in centimeters before using the program because all further calculations are based on that object’s length.

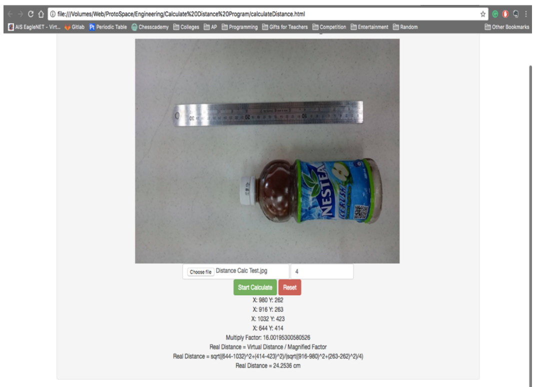


Figure D.6: A screenshot of the distance calculation program

First, the user must upload an image that contains the distance you want to find and the object of known length. Next, the user inputs the length of the known object in cm in the right textbox. Then, the user should click on the 2 endpoints of the length-known object (which should be the size inputted previously), and then click on the 2 endpoints of the unknown distance. The program will then use your first 2 x and y coordinates to find the virtual pixel distance using the Euclidean distance formula: $virtual\ distance = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$. The calculated virtual distance will be divided by the actual object’s size to find the multiply factor (how much the virtual distance is larger than the actual distance). The program then takes the 3rd and 4th mouse x and y coordinates to calculate the virtual distance of the unknown distance. The real distance of the unknown distance can be found by dividing the virtual distance of the unknown distance by the multiply factor.

Decision Making

Reused VS Used Components

Throughout the creation of our ROV, one important aspect we had to consider was which part to keep and which part to change from last year's design. Upon discussions, we decided to keep the used components for the backup ROV and buy new ones for the new ROV. This way, we could ensure that everything is brand new and will work when we make the ROV, and we will also have a backup plan which is the ROV we made last year. The only used components that are in the new ROV are the speaker wires and the camera wires. We have decided to reuse such components because the wires left from previous years are still new as we bought way more than enough last year and the reused camera wires are still in good condition from last year as no apparent rusting of wires is shown.

Original VS Commercial Design

Another challenge that we faced while making our ROV was deciding between making it ourselves or using a commercial design. Some important factors that we had considered were the amount of time needed and the quality of the product. Each component and part of our ROV were brainstormed and decided as a group before the actual creation. Initially, we decided to make our own watertight enclosure, but after 3 months of trial and error and research, the enclosure was still not fully waterproofed. As there were only a few weeks left before the competition, our team had decided to purchase the tube enclosure made by BlueRobotics. Some other commercially designed products we have used are the bilge pumps and the VR camera as we don't have the ability to make those from scratch. For computational units, we have decided to use Arduino, an open source platform, because it has an enormous amount of related resources online and will be easy to seek help from online communities.

Safety

Safety Philosophy

Safety is our primary concern, and no safety issue is too small for us. That is why we employ this philosophy to our design and working habit. We take extra precautions to ensure that no harm will come to all members of our team and each person using our ROV. Also, the Aquila II has been designed to meet the competition's safety requirements. After multiple checks and tests, we have determined that our product meets all the safety standards of the IET.

To guarantee the safety of all our members when they work on electronics and mechanical parts, we require everyone to wear safety equipment (such as goggles and ear protection gear) depending on the particular work they're doing, as shown in Figure E.1. A video showing one of our members working on a mechanical part is shown in Video E.2 in the appendix.

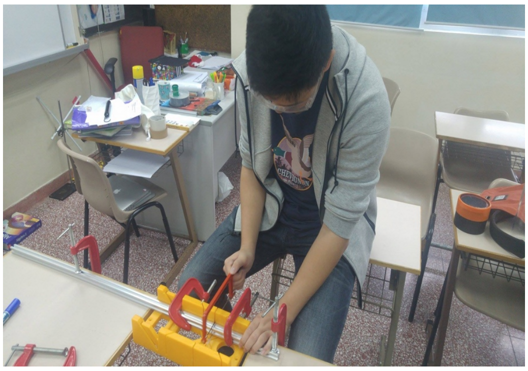


Figure E.1: Ronald Ng cutting metal with appropriate safety equipment

We also pay extra attention to the electronic parts of our ROV. We have a single inline fuse for each motor, and an overall 25 amps fuse for the whole ROV to prevent energy backlash.

A safety checklist is created for all users of our ROV to prevent any accidents while using it. Careful inspection of the checklist before using our ROV is strongly advised.

Safety Checklist and Protocol

	All items attached to the ROV are secure.		Propellers are completely shrouded.
	No sharp edges around the frame.		All wiring and devices for surface controls are secured.
	25A fuse is in place and ready to use.		No equipment is rusted.
	Workstation is completely cleared after work.		Soldering iron is unplugged after use.
	No exposed copper or bare wires.		Tether is properly secured on ROV.
	Long hair is tied back.		Gloves are worn when handling epoxy.
	Hearing protection and safety goggles are worn when operating power tools.		Single attachment point to power source.
	Metal part of the soldering iron is never touched.		Tether is organized to prevent tripping.
	Motor is securely attached.		Both ends of the tether are properly plugged in.

	Check if all the connections are connected securely.
	Check if fuses are all installed.
	Input power supply.
	ROV Start-up Testing (horizontal movement motors and vertical movement motors).
	Check if arm controls are functioning.
	Check if VR control is functioning.

Expense Report

Total Expenditures

Item	Total Cost of the Item
Control System Build Expenditures	$372
Watertight Enclosure	$1577
Camera Build Expenditures	$843
Others	$1948
Total	$5030

Control System Expenditures

Item	Cost of Each Item	Quantity	Total Cost of the Item
MKBP-G500-12 500GPH 12V Bilge Pump	RMB $55 Est HKD $62	6	$372
CCM2 PWM 120W	RMB $18 Est HKD $20	6	$120
Motor Mount	3D Drawn	6	N/A
Propeller	3D Drawn	6	N/A
PS3 Controller	Donated Est HKD $150	1	$150
Total			$642

Watertight Enclosure Expenditures

Item	Cost of Each Item	Quantity	Total Cost of the Item
Watertight Enclosure for ROV/AUV (4" Series)	USD $203 Est HKD $1577	1	$1577
Electronic Tray	Handmade using cardboard	1	N/A
Total			$1577

Camera Build Expenditures

Item	Cost of Each Item	Quantity	Total Cost of the Item
Camera (Model ENOCH-1032)	$60	3	$180
Camera Box Component + Box	$138	1	$138
Analog to HDMI Box Component + Box	$159	1	$159
RCA Female Connector	$10	3	$30
Clear acrylic tube (1m)	RMB $50 Est HKD $56	1	$56
15m of underwater cable	$200	1	$200
TV Monitor	Donated Est HKD $1300	1	N/A
VR Box	$80	1	$80
VR Stand	3D Printed	1	N/A
Total			$843

Other Expenditures

Item	Cost of Each Item	Quantity	Total Cost of the Item
Tools	$630	1	$630
Frame (2020 Aluminum Metal)	$150	1	$150
Fuse	$2	10	$20
Robotic Arm	3D Printed	1	N/A
Servo Motors	$55	5	$275
Speaker Wire (20m)	$325	1	$325
Arduino Uno	RMB $28 Est. HKD $32	1	$32
Arduino Nano	RMB $16 Est. HKD $18	2	$36
Arduino Mega	Donated Est. $350	1	N/A
3D Printing Cost	Donated Est. $300	N/A	N/A
Travel Expenses	Est. 500	N/A	$500
Total			$1968

Challenges Encountered

Technical Challenges

Data transmission has been a huge challenge for us to overcome. Originally, we wanted to use Software Serial as our means of transmission, yet no successful attempts were made after 3 weeks. We ended up researching on other methods and realized that I^2C, an industrial method for data transmission, could be a possible solution. Even though this method was feasible, our transmission distance greatly exceeded its optimal limit, so we had to think of ways to overcome this hurdle. One method we researched online was using pull-up resistors, so we set up the pull-up resistors and experimentally determined that 1k ohms and 10k ohms resistors were the ideal resistor combination for our system.

Another technical problem while creating the ROV was making our own circuits. Creating the circuit required us to work with Full Copper PCBs. To create a circuit like this, we needed to first create a circuit diagram on software such as Fritzing, print the SVG file onto a glossy piece of paper, superimpose the print using extensive heat onto a copper PCB, and finally etch the PCB to get rid of the excess copper. Creating the diagram and printing it wasn't problematic, but one of our major challenges was superimposing the circuit print onto the PCB. The heat distribution and applied pressure had to be uniform throughout the board, or else parts of the circuit board would not impose. Along with this problem, etching proved to be quite a challenge as well. Since our budget was limited, we had to synthesize our own etching solution, which didn't provide the results we needed as sometimes the circuit didn't etch completely and dissolved parts of the printed area. Although these challenges were quite frustrating, our team worked around them by adding solder to the disconnected circuit and providing a path for electricity to flow.

Intrapersonal Challenges

Since our team is getting larger every year, organizing tasks for each member was getting more difficult and inconvenient. For the first few weeks, members of the team came to the meetings but just sat there for most of the time, as they didn't know what to do. Instead of planning on paper and telling each member what to do during each meeting, we decided to enhance the process using Trello, a free online service for organizing and assigning tasks. We also created 3 different teams and assigned a leader for each team. (see Figure F.1) The CEO and CTO would assign tasks to each team, and the team leaders could assign tasks to their sub members. Using this method, our team has achieved higher efficiency, and members finally knew what they should do during each meeting.

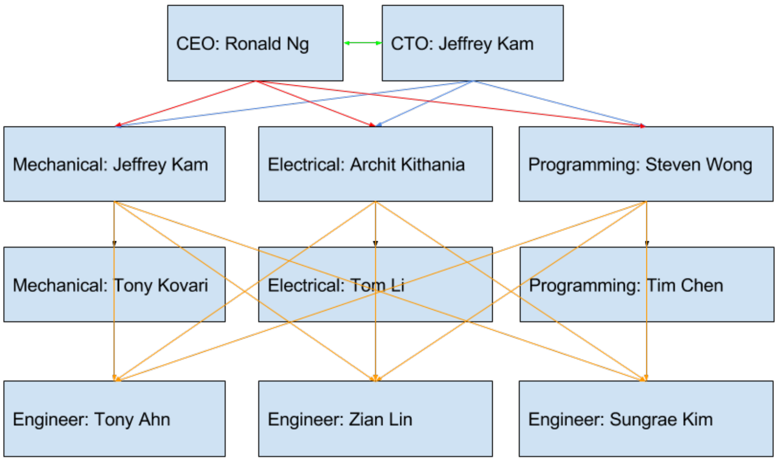


Figure F.1: A visual representation of the hierarchy of the roles in our team

Testing and Troubleshooting

To avoid mistakes that we made last year, we decided to do extensive testing on each and every component of our ROV. For instance, the motors we used have been tested 3 times before being mounted onto the ROV. Also, we designed a system (see Figure G.1) to test the thrust of our motors based on a lab report from one of our members. We also tested the thrust against various propeller sizes to find the maximum efficiency, since the thrust decreases after a certain propeller size as a

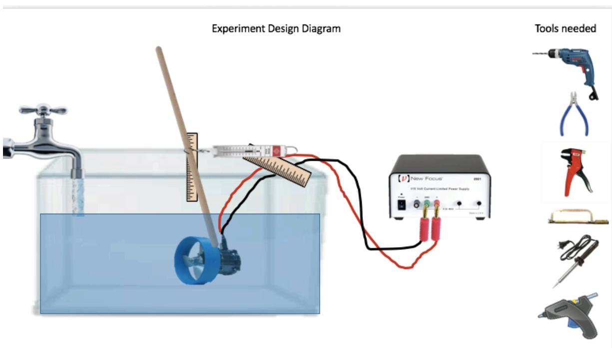


Figure G.1: A diagram of how to set up the testing system

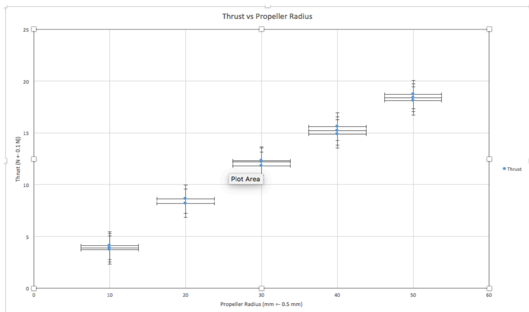


Figure G.2: A screenshot of the graph of the thrust vs propeller size data vacuum would be created. We recorded the data in an Excel spreadsheet and plotted them on the graph shown in Figure G.2.

It is important to follow a protocol when we encounter a problem in engineering, so we have developed a flowchart to help guide us through the troubleshooting process. There are incidents of software failure during the makings of the ROV, and they are quickly solved through peer review on Gitlab, as suggested by the flowchart (see Figure G.3). This flowchart has helped us and saved a lot of our time whenever we encountered problems.

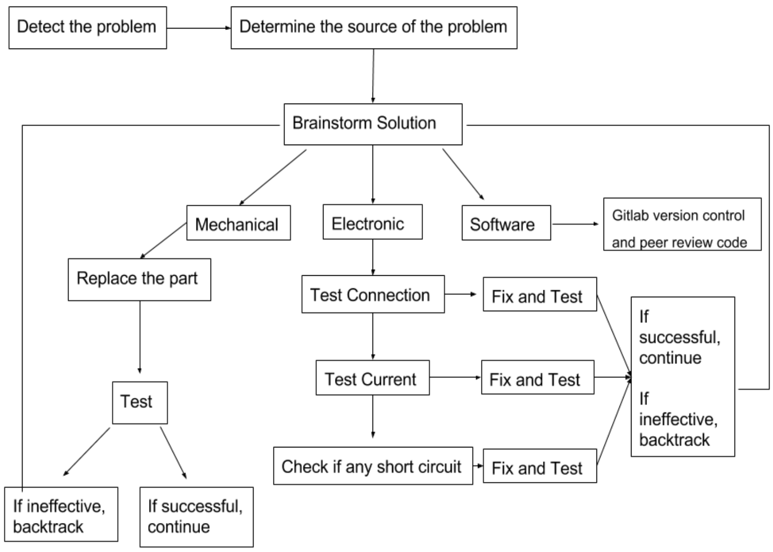


Figure G.3: A flowchart to show how to troubleshoot

Lessons Learned

Completing the ROV has been not only an exciting journey but also extremely educational for us. With each problem we encountered, we learned something truly extensive. For example, before this year, our team used three circuits, each regulating the voltage for an individual camera. This year, our team, with an application of Ohm's Law and Kirchhoff's Loop Rule, was able to create one circuit that could regulate the voltage of all the cameras applied to it. Even better, it also cut the current requirement by a third of the previous amount. We also learned better communication and teamwork skills. For the ROV to perform well, each part of it needed to work with perfect synchronization. The ROV had too many parts to be done alone, which required communication within the team. Although there were communicational problems in the beginning that led to some mishaps (like a tube being shut with epoxy before a few final adjustments were made), each issue helped our team work together better, and by the end of the experience, we ran like a well-oiled machine.

Future Improvements

Our team has 2 major areas of improvement needed: Lack of passion and resources. First of all, due to many members being neophytes in this subject, our teammates frequently experienced limitation in their participation, which led to little motivation. We now realize that it is imperative for our members to first become competent engineers so that they can gain adequate knowledge and skill to participate. In the future, to boost the passion of engineering in our members, they must first dedicate sufficient time to learn basic skills and further practice them. As for the lack of resources, we hope to find more sponsors to support our company, as the cost of development for this year's ROV is relatively high, and most of the monetary support comes from the members within. We hope to contact electronic component shops such as WECL and RS Electronics for material support, and also hope to continue our current sponsorships so we can work to our fullest potential without the fear of lacking the resources to do so.

Teamwork

Our company entered the MATE ROV competition for the third time; therefore, cumulative experiences from the senior engineers provided valuable knowledge to the whole team. As we were constructing the ROV, we discovered that teamwork was the most important aspect. The construction of the ROV had to take many different fields into consideration, such as designing, mechanical engineering, computer programming, and material engineering. Ronald and Jeffrey, as leaders of the team, created detailed plans every week for each member to follow and assigned roles for talented individuals who possessed talents and passions in their fields. We developed a hierarchy to determine which member should perform which task (See Figure F.1). We also used Trello as a central platform to organize tasks for each member (See Figure H.1). This year, Steven Wong, leading Tim Chen, is responsible for the programming section to create the control and VR program; Archit Kithania, leading Tom Li, is responsible for the electronic section to create the camera box and control box; Jeffrey Kam, leading Tony Kovari, is responsible for the mechanical section to design and construct the ROV and the gripper. As a team, we discussed with each other and embraced each other's ideas, and always seeking ways to improve. The result of this hard work was a collage of genius ideas combined together from each individual. Without a doubt, teamwork was the most important factor in creating this ROV.

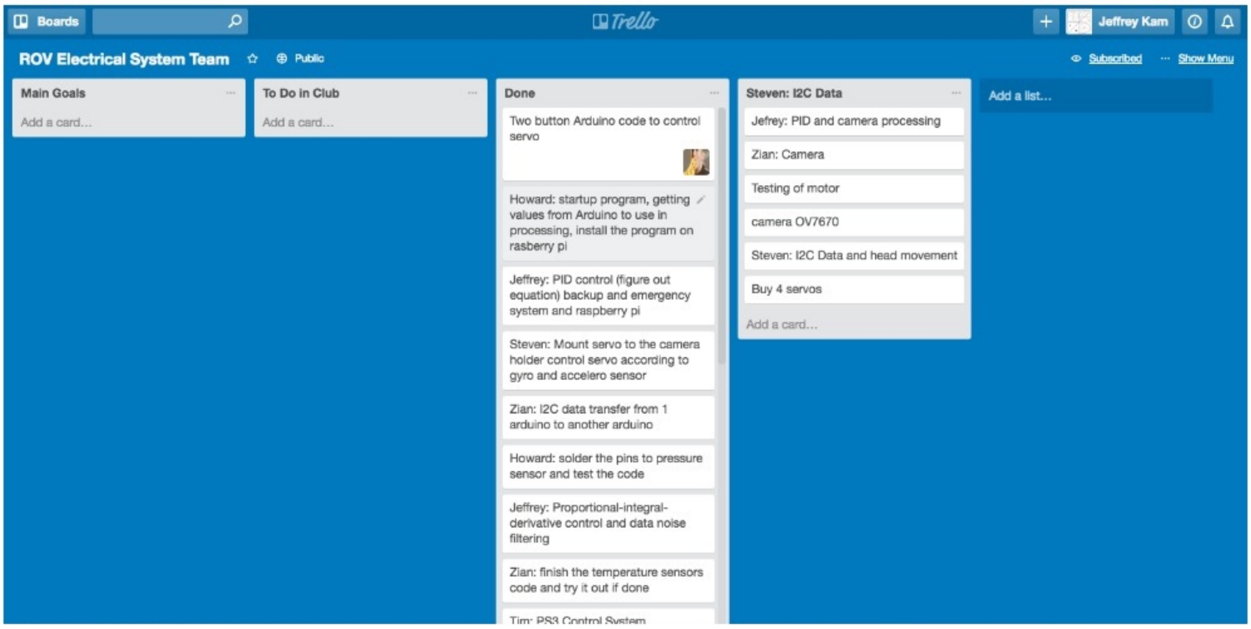


Figure H.1: The Trello pages we use for project management

Reflection

Overall reflection

Members' Reflections

Ronald Ng

For these three years I've worked on ROV, I've encountered various challenges, learned different variety of things and met different people. The first year we join this competition, our team was compact, and had similar idea; it was really smooth working with them. This year, our team grew exponentially, many new members joined coming up from middle school, each and every excited for learning new things about engineering. This year as I've become the CEO of this club, I've matured and learned to listen to each person's opinion and brainstorm it into something better. I've learned a lot of different ways to time manage and learned how to motivate students to work. As the CEO this year, I'm really proud of what we've accomplished and look forward to the future of ROVs.

Jeffrey Kam

The two core values of engineering for me are exploration and problem solving. During the course of the ROV making process, I have enriched myself in those 2 values. I always try to explore new and more efficient solutions to the existing system, trying to improve it with better code and higher efficiency design. For example, from researching online, I realized that Ethernet wire at CAT6 has really low resistance, and thus useful for our design for transferring data between underwater and on ground Arduinos. This year, I have taught some of the freshmen how to design circuits and how to build robot efficiently with detailed explanations using physics. These precious experiences allow me to be a better communicator as i always keeps the idea and thinking process to myself and often explain concepts in an overcomplicated way. As the CTO and head of mechanical engineering in the club, I wish train some of the members on the team to have individual problem solving abilities and the heart to explore so after I left for university this year, those will be able to make the team thrive with their ability to problem solve and guide the team to explore for new solutions.

Steven Wong

This was my first year joining the Engineering Club and will be my last year in the club as I am graduating, but I felt that I have learned a lot from joining this competition that will greatly benefit me in the future. As I plan to major in Computer Science, I initially joined the club as I wanted to put my skills into practice and try to improve my coding ability. However, I have done much more than that. I learned how to use an Arduino to connect software I was creating with hardware that interacted with the real world. I also practiced basic engineering skills such as soldering and wiring. Aside from that, I also worked collaboratively with the other members in the Engineering Club, and these teamwork and communication skills will stick with me wherever I go. Although I will be leaving, the expertise I have garnered will never go away.

Archit Kithania

This was my first year as a part of an actual engineering team, as before this I had conducted projects by myself. As a senior working towards getting into the field of Electrical Engineering, I could not have been happier with the work required by this project. Not only was I given the opportunity to learn some new skills like using digital software to simulate real-life applications, I was also able to sharpen my pre-existing skills such as soldering, epoxying, and most importantly, team collaboration. With each frustrating problem the crew faces, I feel myself mature as a neophyte engineer and I can say without a doubt that the experiences from Project Aquila will be extremely helpful in the future.

Tim Chen

This is my second year in the engineering team. As a sophomore, my goal this year was just to learn and have new experiences. I was responsible for the backup ROV's control and the primary ROV's motor setup. I also tried to code the overall control program through Arduino. In the end, my code wasn't used, but a version adapted from it was used instead. I was just happy to be able to get some practice in coding. After that, I worked on the ROV's motor setup and various other little things, such as fixing and adjusting minor issues. Right now, I have just finished adjusting the main control for the backup ROV. During these processes, I not only learned about how to code a main control system, but also became more and more experienced with physical engineering

Tom Li

Being in engineering club for the first time this year, I had been very lucky to receive assistance and guidance from experienced engineers within this club. I was able to learn knowledge related to circuit boards, electricity, energy. First-hand experience in soldering wires, putting together camera parts, and understanding how wires work had been a fascinating and exciting journey. This is definitely something I will look into as I proceed into college.

Tony Kovari

Being a sophomore and second time competition participant, I wanted to learn more about making an ROV and actually make parts that went not only on the backup ROV but on the main one too. I learned a lot about 3D modelling software and 3D printing. Since our team is pretty small, I learned most of what I know from watching YouTube tutorials.

Zian Lin

This year was my first time in the AIS Engineering Club. My initial interest in this club was from wanting to get a taste of hands-on work and actually creating something by employing engineering concepts. This year, as it was my first year, I participated in several activities, such as constructing the backup ROV, soldering and working with a lot of wires, diagramming circuits, fixing some electrical problems, and some basic coding in Arduino. While doing these activities I learned how to code some basic things in Arduino (which is a particularly versatile language), learned how to solder things together, programmed a temperature sensor, learned how to diagram a circuit, and learned some basic electricity and wiring concepts along the way. Some difficulties I encountered were due to my lack of exposure to these activities before, but thanks to the other members in the club, I overcame these difficulties and usually got the job done. Overall, engineering club is a great experience for anyone, even if they're new to engineering like me, and I strongly recommend it.

Acknowledgements

Aquila Integrated Solutions wholeheartedly appreciates all the donations and sponsorships we have received. We couldn't have completed our ROV, the Aquila II, without our generous sponsors. Thanks to MATE/IET for providing us with the chance to compete and develop our critical thinking skills throughout the journey of making this ROV. We also appreciate the American International School for providing us with space for our team to meet up, and our mentor Mr. Eunsup Kang for his generous donations, and both Mr. Eunsup Kang and Mr. Brian Mellon for their massive amounts of time spent helping and guiding us to where we are now. Also, we are very grateful for the generous sponsorships provided by Makerbay - 4 full memberships for 6 months with a total worth of HKD 21,000, which allows us to do heavy metal and wood works there and use their high-power tools to refine our ROV. Lastly, we would like to thank Tim Chung for the drawing of our company's logo and report layout, Justin Bae for giving us his precious technological advice, and Hana Miyaji for proofreading.

References

"Algorithm for Self-balancing Robot." Frobot - a DIY Robot. N.p., 29 Apr. 2014. Web. 08 Jan. 2017. <https://frobotme.wordpress.com/2014/04/29/algorithm-for-self-balancing-robot/>.

Anagmate. "Get Distance between Two Points in Canvas." Javascript - Get Distance between Two Points in Canvas - Stack Overflow. N.p., 2014. Web. 29 Oct. 2016. <http://stackoverflow.com/questions/20916953/get-distance-between-two-points-in-canvas>.

"OpenROV 2.8 Kit." OpenROV | Underwater Exploration Robots. OpenROV, n.d. Web. 14 Nov. 2016. <https://www.openrov.com/products/openrov28/>.

HobbyTransform. "MPU6050: Arduino 6 Axis Accelerometer + Gyro - GY 521 Test & 3D Simulation." Instructables.com. Instructables, 12 May 2016. Web. 04 Oct. 2016. <http://www.instructables.com/id/MPU6050-Arduino-6-Axis-Accelerometer-Gyro-GY-521-B/>.

"Connecting Arduino to Processing." *Connecting Arduino to Processing*. N.p., n.d. Web. 14 Sept. 2016. <https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing>.

"MPU-6050 Accelerometer Gyro." *Arduino Playground - MPU-6050*. N.p., n.d. Web. 23 Jan. 2017. <http://playground.arduino.cc/Main/MPU-6050>.

"ROV and Marine Robotics Systems and Components." *Blue Robotics*. N.p., n.d. Web. 19 Jan. 2017. <https://www.bluerobotics.com/>.

Ustrobotics. "HKUST ROV 2015-STINGRAY." *YouTube*. YouTube, 21 June 2015. Web. 21 Nov. 2016. <https://www.youtube.com/watch?v=tYRYdTxP_7Y>.

OceanModules. "Search and Rescue ROV Training." *YouTube*. YouTube, 07 Feb. 2011. Web. 01 Feb. 2016. <https://www.youtube.com/watch?v=Zfdg6ZXSQ5Q>.

OpenROV. "MATE ROV Competition 2016: Williams Pool Stream." *YouTube*. YouTube, 13 July 2016. Web. 30 Dec. 2016. <https://www.youtube.com/watch?v=Gvp5R0p9fWM>.

"DOWNLOAD." *Arduino Playground - PIDAutotuneLibrary*. N.p., n.d. Web. 18 Oct. 2016. <http://playground.arduino.cc/Code/PIDAutotuneLibrary>.

Appendix

Overall all Pictures and Videos: https://goo.gl/5HHYJk

Pictures and Videos used for Section A: https://goo.gl/b0D0a5

Pictures and Videos used for Section B: https://goo.gl/QR1qQw

Pictures and Videos used for Section C: https://goo.gl/fn0Jic

Pictures and Videos used for Section D: https://goo.gl/4g1zij

Pictures and Videos used for Section E: https://goo.gl/zR2zu4

Pictures and Videos used for Section F: https://goo.gl/EuxsPI

Pictures and Videos used for Section G: https://goo.gl/mVWHCc

Pictures and Videos used for Section H: https://goo.gl/ZQaUWU