



Capstone Final Report

Development of an Autonomous Surface Vessel

Abrisam Durrani 1802806

Claudia Koh 1802496

Mohamed Haseem 1801522

Raymond Tan 1800790

Project Supervisor: Mr Mark Teo

Project Co-Supervisor: Mr Albert Eng

Acknowledgement

We would like to extend our thanks to anyone who has contributed to the success of our Capstone project. Without the help received from them, we would not have been able to achieve the objectives of our ASV.

To Dr Tang Liang, thank you for coordinating and facilitating all admin matters of the Capstone project. We would also like to show appreciation for your constant guidance and readiness to help whenever we approached you for advice, especially during the initial stages of our project when you advised us on selecting the most suitable operating system and MCU.

To our advisor, Mr Albert Eng, your kindness and willingness to help us will not be forgotten. Without your enthusiasm and expertise, we would have struggled greatly. Thank you for your constant support, and for spreading joy and optimism every time we are around you. Your expertise and skills were invaluable and thank you for importing us with your wisdom.

Special thanks to Ms Amanda Niu, who helped us with the design and calculation of our hull.

Special thanks to Dr Liaw Hwee Choo and Mr Gan Lian Chai for their invaluable assistance during the period of time when we were programming for the project. Without them taking the time out of their busy schedules to answer our numerous queries, this project would not be successful.

To ST Engineering Marine, thank you for providing us the opportunity to work collaboratively for such an esteemed company. The guidance and industrial advice from experts were extremely valuable and has helped us elevate our project.

Finally, to our supervisor, Mr Mark Teo, thank you for your constant support and guidance. Your leadership and mentorship ensured that we strived to excel and that we carry out our Capstone project to its fullest potential. You were always reasonable and supportive of our decisions and gave critical response and advise whenever the team required it. You gave us the privilege to start off an important project that will hopefully benefit SIT and its future students. We are eternally grateful for the opportunities and support that you have provided to us, and we hope we have met your expectations.

Thank you.

Abstract

The capstone project is a major obstacle for every student in SEEMS. The capstone project will span two trimesters (8-Months) and aim to implore students to apply the knowledge and skills that they have picked up during their university tenure and internship. Students will undertake complex, real-world and industry related projects in groups to demonstrate their engineering, project management and interpersonal skills.

This document will provide a detailed report on our Capstone project; the Development of an Autonomous Surface Vessel (ASV) project for the maritime industry. This report will provide the overview of the Capstone project, the overall design of the project, and the work carried out thus far. The challenges faced and future works will also be touched on.

Table of Contents

Acknowledgement.....	1
Abstract.....	2
Table of Figures.....	6
List of Tables	8
Abbreviations	9
1. Introduction.....	10
1.1. Purpose	10
1.2. Project Background.....	10
2. Project Overview.....	11
2.1. Problem Statement	11
2.2. Requirements	11
2.3. Objectives.....	11
2.4. Scope of Work	12
2.5. Project schedule	12
2.5.1. Trimester 1	12
2.5.2. Trimester 2	16
2.6. Testing and Validation	20
2.6.1. Trimester 1 Testing Criteria.....	21
2.6.2. Trimester 1 Testing Scenarios	21
2.6.3. Trimester 2 Testing Criteria.....	21
2.6.4. Trimester 2 Testing Scenarios	22
2.7. Responsibility Matrix	22
2.7.1. Roles and Responsibilities.....	23
2.8. Budget.....	24
3. Overall Design	25
3.1. Architecture.....	25
3.1.1. Hull	25
3.1.2. Operating System and Microcontroller	25
3.1.3. Sensor/Obstacle Detection	26
3.1.4. Navigation.....	26
3.1.5. Other components	26
3.2. System Architecture Diagram	27
3.2.1. System Overall Architecture Diagram	27
3.2.2. System Architecture for Trimester 1	27
3.3. Wiring Diagram	28
3.3.1. Overall Wiring Diagram.....	28

3.3.2.	Sensors	29
3.3.3.	Motors.....	29
3.3.4.	Wiring diagram for rudder	30
3.4.	Work Breakdown Structure	30
4.	Component Selection.....	32
4.1.1.	MCU Selection	32
4.1.2.	Operating System Selection	33
4.1.3.	Motor.....	34
4.1.4.	Wi-Fi Module Selection.....	35
4.1.4.1.	Configuration of XBee	36
4.1.5.	GPS Module.....	36
4.1.6.	Lidar Sensor.....	37
4.1.7.	Finalised BOM List	38
5.	Trimester 1 – Development.....	39
5.1.	Hull Design	39
5.1.1.	Preliminary Draft	39
5.1.2.	First Iteration.....	40
5.1.3.	Second Iteration.....	40
5.1.4.	Challenges Faced During Hull Fabrication	41
5.1.5.	Solution and Final Hull Design	41
5.2.	Equipment Housing Deck.....	44
5.2.1.	Design of Housing Deck	44
5.2.2.	Fabrication of Housing Deck.....	45
5.3.	Camera connection to OS.....	47
5.4.	Lidar	48
6.	Trimester 1 – Testing	51
6.1.	Float Test and Pool Test.....	51
6.2.	Lidar Testing.....	51
6.3.	Pool Tests.....	52
7.	Trimester 2 – Development.....	55
7.1.	Physical Hardware.....	55
7.1.1.	Rudder.....	55
7.1.2.	Camera Mount.....	56
7.1.3.	RC Relay Circuit	57
7.2.	Software	58
7.2.1.	Wi-Fi-Communication	58
7.2.2.	Camera.....	62

7.2.3.	GPS Set up and Testing	64
7.2.4.	IMU.....	68
7.2.5.	Alternative GPS	68
7.2.5.1.	Alternative GPS: Adafruit Ultimate GPS Breakout Board.....	69
8.	ASV Operation.....	71
8.1.	Level 1 Operation	71
8.2.	Level 2 Operation	71
8.3.	Level 3 Operation	72
9.	Trimester 2 – Testing	74
10.	Overcoming Challenges	75
10.1.	Hull Design.....	75
10.2.	Running ROS.....	75
10.3.	Camera Live Feed	75
10.4.	Multiple Commitments	76
11.	Lessons Learnt.....	77
11.1.	Software Competencies	77
11.2.	Robot Operating System	77
11.3.	Naval Knowledge	77
12.	Future Works.....	78
12.1.	Sensor Integration.....	78
12.2.	Development of Obstacle Avoidance Algorithm	78
12.3.	Usage of data retrieved from GPS and RPI.....	78
13.	Conclusion	79
	References	80
	Appendix A	81

Table of Figures

Figure 1: Grades of Automation	12
Figure 2: Trimester 1 Project Schedule	16
Figure 3: Trimester 2 Schedule	20
Figure 4: Organisation Tree	23
Figure 5: System Architecture Diagram for Trimester 1	25
Figure 6: Overall System architecture diagram.....	27
Figure 7: System Architecture for Trimester 1	27
Figure 8: Overall Wiring Diagram of ASV for Trimester 1	28
Figure 9: Wiring Diagram of Sensors	29
Figure 10: Wiring Diagram for motors	29
Figure 11: Wiring diagram for rudder.....	30
Figure 12: Work Breakdown Structure	30
Figure 13: STM32H723ZG	33
Figure 14: Raspberry Pi 4. Image retrieved from Official Raspberry Pi website.....	34
Figure 15: Mabuchi RS-540 Torque Tuned Motor	35
Figure 16: XBee Wireless Communication.....	35
Figure 17: XB2B-WFWT-001	36
Figure 18: X-NUCLEO-GNSS1A1	37
Figure 19: RPLIDAR A1	38
Figure 20: First Preliminary Draft Design.....	39
Figure 21: CAD model of first Iteration	40
Figure 22: Second Iteration of ASV	40
Figure 23: The provided 3D printed Hull design	42
Figure 24: Holes for water pump	42
Figure 25: Hole for propeller shaft.....	43
Figure 26: Waterproofing the propeller shaft hole.....	43
Figure 27: Underside of hull with silicon applied.....	44
Figure 28: CAD model of Housing Deck.....	44
Figure 29: Exploded View of Housing deck with components.....	45
Figure 30: Equipment Housing Deck.....	45
Figure 31: Housing Deck attached to the Hull	46
Figure 32: Fabricated camera mount	46
Figure 33: Final ASV Design	47
Figure 34: Setting up connection to Raspberry Pi	47
Figure 35: Snippet of the code for the connection of the PiCam.....	48
Figure 36: Live Feed generated from the PiCam.....	48
Figure 37: A1-M8 Lidar	49
Figure 38: Example lines of code used to download ROS packages.....	49
Figure 39: Example lines of code used to download RPLidar Packages	49
Figure 40: Command used to launch visualisation	50
Figure 41: Visualisation of the data from the Lidar	50
Figure 42: Displaying the raspberry pi to the laptop via remote desktop.....	50
Figure 43: Conducting Float Test	51
Figure 44: Testing of Lidar from a distance of 50cm.....	52
Figure 45: Testing of the Lidar from a distance of 100cm.....	52
Figure 46: SP Swimming pool used for the Pool Test.....	53
Figure 47: Pool Test.....	53
Figure 48: Pool Test with Camera and Lidar	53
Figure 49: Store bought rudder	55

Figure 50: Store Bought Rudder after modification.....	56
Figure 51: CAD model of camera mount design.....	56
Figure 52: 3D printed camera mount.....	57
Figure 53: RC Relay Circuit.....	57
Figure 54: Detailed RC Relay Circuit.....	58
Figure 55: Xbee S68	59
Figure 56: XCTU-Software.....	59
Figure 57: Configuration for the Creator (Laptop).....	60
Figure 58: Configuration of Joiner (ASV).....	61
Figure 59: Testing the Wi-Fi modules by transmitting and receiving data	62
Figure 60: Testing Intel RealSense Viewer application in Raspi.....	63
Figure 61: Testing of the python script's output.....	64
Figure 62: Snippet of the code	64
Figure 63: GPS Module connected to MCU	65
Figure 64: Snippet of GPS code.....	66
Figure 65: Results retrieved from GPS.....	67
Figure 66: Coordinates from GPS in Google Maps	67
Figure 67: Sample of Output from IMU displayed on terminal	68
Figure 68: IMU attached to STM32 mounted on ASV.....	68
Figure 69: Arduino Uno and Adafruit Ultimate GPS Breakout Board	70
Figure 70: Sample of output from the GPS.....	70
Figure 71: Screenshot of Terminal Control view on laptop	71
Figure 72: Flowchart illustrating logic for Level 2 Operation	72
Figure 73: Window to input Waypoint.....	72
Figure 74: Flowing illustrating logic for Level 3 Operation	73
Figure 76: Testing at SP Pool	74

List of Tables

Table 1: Stakeholder Requirements for ASV	11
Table 2 Responsibility Matrix	23
Table 3: Description of Roles	24
Table 4: Comparison table for MCU	33
Table 5: Features of the STM32H723ZG	33
Table 6: Features of Raspberry Pi 4.....	34
Table 7: Features of the XB2B-WFWT-001	36
Table 8: Features of X-NUCLEO-GNSS1A1	37
Table 9: Features of Lidar Sensor	37
Table 10: Hull Fabrication methods comparison.....	41
Table 11: Evaluation of GPS Options.....	69
Table 12: ASV Modes of Operation.....	71

Abbreviations

Acronym

ASV

GPS

IMU

IMU

MASS

MIT

MSC

PCB

PDB

PSA

PSA Marine

ROS

RTOS

SIT

STE

STM

TDOA

USV

Definition

Autonomous Surface Vessel

Global Positioning System

Inertial Measurement Unit

Inertial Measurement Unit

Maritime Autonomous Surface Ships

Massachusetts Institute of Technology

Maritime Safety Committee

Printed Circuit Board

Power Distribution Board

Port of Singapore Authority

Port of Singapore Authority Marine

Robot Operating System

Real Time Operating System

Singapore Institute of Technology

Singapore Technologies Engineering

STMicroelectronics

Time Distance of Arrival

Unmanned Surface Vessel

1. Introduction

1.1. Purpose

The capstone project is a major group project that students are required to undertake. Students are expected to tackle complex, industry related issues to demonstrate their engineering, project management and interpersonal skills gained during their university tenure and internships.

Our team has undertaken the development of an Autonomous Surface Vessel for the maritime industry as our capstone project, in partnership with Singapore Institute of Technology (SIT) and ST Engineering Marine.

This document will put forward the progress of the project, and will highlight the challenges, the lessons learned, and the objectives met thus far. In addition to this, a brief project overview and description of the requirements, scope of work, schedule and budget will be provided. A brief summary of the project and future plans will also be provided.

1.2. Project Background

In recent years, the Singapore Port Authority have been battling a surge in port congestion. In March 2021, it was reported that there was an increase in the number of commercial vessels docking in Singapore. A report by Commodities pricing agency S&P Global Platts found that an average of 52 vessels per day stayed at Singapore's port for more than two days in February, a 60% increase from the same period last year [1]. The increase in port activity may bring about various issues, such as an increase in the number of marine accidents.

Therefore, the Port of Singapore is turning to technological solutions to keep up with the increase in activity [2]. Companies such as ST Engineering and PSA Marine are looking towards the development of unmanned vessels as the future of maritime. These unmanned vessels will cut labour costs and enhance safety while providing great opportunities for advancement in training and technology such as digital twin and data analytics. The potential of autonomous ships in providing solutions for problems in the maritime industry, such as prevention of accidents, will have a great influence on the entire maritime transportation infrastructure. The potential of autonomous ships in providing solutions for problems in the maritime industry, such as prevention of accidents, will have a great influence on the entire maritime transportation infrastructure.

As the idea of autonomous ships is still in its budding stages, SIT aims to join the effort in its development. As such, our group sees a great opportunity to contribute greatly to the industry by providing a proof of concept for SIT through the development of an ASV. Our group aims to tackle this challenge and aid the maritime industry in its journey to realising the use of autonomous ships on international waters.

2. Project Overview

2.1. Problem Statement

The maritime industry is looking towards the use of autonomous ships in the near future. However, the concept of autonomous ships is still in development and requires continued research and progress. Hence, there is a current shortage of actual prototypes and proof of concepts that are being implemented for industry use.

SIT aims to play a part in contributing to the research, however, there is no existing autonomous system or vessel being developed in SIT.

Our group will help facilitate the research by providing a proof of concept by developing an ASV capable of unmanned, autonomous operations and obstacle detection and avoidance. This project will be done collaboratively with SIT and ST Engineering.

2.2. Requirements

The high-level requirements for our ASV are as listed in the table below:

Identifier	Requirement
STR_1	The ASV shall be able to float
STR_2	The ASV shall be autonomous
STR_3	The ASV shall be able to detect obstacles
STR_4	The ASV shall be able to avoid obstacles autonomously
STR_5	The ASV shall be able to locate its current position
STR_6	The ASV shall be able to travel to specified waypoints autonomously
STR_7	The ASV shall be able to return to its original path after obstacle avoidance autonomously
STR_8	The ASV shall run for a minimum of 30 minutes

Table 1: Stakeholder Requirements for ASV

2.3. Objectives

The main objective of our ASV project is to develop an ASV that is capable of autonomous and unmanned operations that satisfies Level 3 and 4 of the Degrees of Automation.

Level	Name	Definition
Degree 1	Ships with automated processes and decision support	Seafarers are onboard to operate the ship. Some operations may be automated.
Degree 2	Remotely controlled ship with seafarers on board	The ship is controlled and operated from another location, but seafarers are onboard.
Degree 3	Remotely controlled ship without seafarers onboard	The ship is controlled and operated from another location. There are no seafarers onboard.
Degree 4	Fully autonomous ships	The operating system of the ship is able to make decisions and determine actions by itself.

Figure 1: Grades of Automation

Based on the preliminary requirements that was established, some of the preliminary objectives of our project are as follows:

1. To develop and demonstrate the unmanned capabilities of the ASV
2. To develop and demonstrate the obstacle detection and avoidance of the ASV
3. To develop a local positioning system for the ASV
4. To develop and demonstrate the autonomous capabilities of the ASV
5. To develop and demonstrate capabilities that meets Level 3 and 4 of the Degree of Automation

These objectives will serve as a guideline for the design, development, and scope of work for our ASV Capstone project.

2.4. Scope of Work

The scope of work to be undertaken for this project is as outlined below:

- To build the hull
- To develop algorithm for autonomous movement and obstacle detection
- To develop route mapping via waypoints
- To develop autonomous control of the ASV
- To carry out testing and proof of concept procedures
- To determine location of work and testing facilities

2.5. Project schedule

2.5.1. Trimester 1

We plan to prioritise our project development with the following goals for Trimester 1:

- To build the hull
- Establish Remote control / Terminal Control
- Establish Wireless communication for camera and Lidar
- Environment monitoring

The project schedule for the first trimester as shown below in Figure 2.

Task Description	Category	Progress	Start	Days
Defining Items - Week 1				
Further Research on Components	On Track	100%	6-Sep-21	4
Hull Design Consultation	On Track	100%	6-Sep-21	1
Draft Hull Design	On Track	100%	6-Sep-21	4
Deck Creation for Weekly Meeting	On Track	100%	7-Sep-21	2
Final Proposal and Project Plan Drafting	On Track	100%	7-Sep-21	6
Weekly Meeting with Prof Mark	On Track	100%	9-Sep-21	1
Defining Items - Week 2				
Define Testing Requirements	On Track	100%	13-Sep-21	7
Define Components	On Track	100%	23-Sep-21	3
Define Components Procurement	On Track	100%	23-Sep-21	3
Refine Hull Design	On Track	100%	23-Sep-21	7
Weekly Meeting with Prof Mark	On Track	100%	16-Sep-21	1
Biweekly Individual Progress Report Submission	On Track	100%	18-Sep-21	1
Execution Start - Week 3				
Draft Test Scenarios	On Track	100%	23-Sep-21	7
Final Proposal and Project Plan Finalization	On Track	100%	20-Sep-21	7
Hull Design Finalization	Milestone	100%	20-Sep-21	1

Initial Deck for Preliminary Design Review	On Track	100%	21-Sep-21	
Weekly Meeting with Prof Mark	On Track	100%	23-Sep-21	1
Execution Start - Week 4				
Prep for Preliminary Design Review	On Track	100%	24-Sep-21	7
Final Proposal and Project Plan Submission	Milestone	100%	26-Sep-21	1
Biweekly Individual Progress Report Submission	On Track	100%	2-Oct-21	1
Execution: Procurement - Week 5				
Purchase/Gather materials and Components	Med Risk	100%	4-Oct-21	7
Midterm - Week 6				
Midterm Exams	On Track	100%	11-Oct-21	5
Biweekly Individual Progress Report Submission	On Track	100%	16-Oct-21	1
Recess Week - Week 7 (17 Oct 21)				
Start of Fabrication/Integration/Testing Phase	Milestone	100%	18-Oct-21	1
PiCam Live Feed Programming	On Track	100%	18-Oct-21	5
Midterm Report Finalization	On Track	100%	18-Oct-21	4
Lid and Deck for Component - Fabrication	On Track	100%	18-Oct-21	5
Progress Update with Prof Mark (PiCam Live Feed)	On Track	100%	20-Oct-21	1
Midterm Report Submission	Milestone	100%	22-Oct-21	1

Midterm Presentation Slides Submission	Milestone	100%	22-Oct-21	1
--	-----------	------	-----------	---

Fabrication/Integration/Testing - Week 8 to 11

Structure for sensors & GPS - Fabrication	On Track	100%	25-Oct-21	1
Assemble components onto lid/structure	Milestone	100%	2-Nov-21	1
Float Test with all Components	Milestone	100%	2-Nov-21	1
Progress Update with Prof Mark (Structure for sensors, Float Test)	On Track	100%	3-Nov-21	1
Biweekly Individual Progress Report Submission	On Track	100%	7-Nov-21	1
Start Prep for Demo	On Track	100%	15-Nov-21	7
Pool Test	Milestone	100%	19-Nov-21	1
Fabrication/Integration/Testing Phase End	Milestone	100%	19-Nov-21	1
Biweekly Individual Progress Report Submission	On Track	100%	21-Nov-21	1

Demo - Week 12

Prep for Demo	On Track	100%	22-Nov-21	3
Progress Update with Prof Mark (Lidar, Remote Control)	On Track	100%	22-Nov-21	1
Pool Test and Video Taking	On Track	100%	11/24/2021	1
Progress Update with Prof Mark (Lidar, Remote Control)	On Track	100%	25-Nov-21	1

Final Presentation - Week 13

Final Presentation Slides Submission	On Track	100%	1-Dec-21	1
--------------------------------------	----------	------	----------	---

Final Presentation	Milestone	100%	2-Dec-21	1
Biweekly Individual Progress Report Submission	On Track	100%	5-Dec-21	1
Final Report	On Track	100%	5-Dec-21	1
Final Work Source Submission	On Track	100%	5-Dec-21	1
Final Exam - Week 14 (6 Dec 21)				
Final Exams	On Track	0%	6-Dec-21	14
Trimester 1 End				
End of Trimester 1	Milestone	0%	17-Dec-21	1

Figure 2: Trimester 1 Project Schedule

2.5.2. Trimester 2

We plan to prioritise our project development with the following goals for Trimester 2:

- To achieve situation awareness with Way point integration
- To achieve Autonomous with Collision Avoidance
- To conduct Extensive site testing for verification and validation

The project schedule for the second trimester as shown below in Figure 3.

Week 1 (Motor, Encoder, PWM)

Weekly Meeting with Prof Mark	10-Jan-22
-------------------------------	-----------

DC Motor

Code motor directions for DC	10-Jan-22
------------------------------	-----------

Code encoder for DC	10-Jan-22
---------------------	-----------

Code PWM for DC	10-Jan-22
-----------------	-----------

Servo

	10-Jan-22
--	-----------

Code motor directions for Servo	10-Jan-22
---------------------------------	-----------

Code encoder for Servo	10-Jan-22
------------------------	-----------

Code PWM for Servo	10-Jan-22
--------------------	-----------

Week 2 (Communication for terminal, terminal control)

Code UART	17-Jan-22
-----------	-----------

Testing of comms between ASV and laptop	17-Jan-22
---	-----------

C# Programming for terminal control	17-Jan-22
-------------------------------------	-----------

Movement control on terminal	17-Jan-22
------------------------------	-----------

Waypoint movement	17-Jan-22
-------------------	-----------

Week 3 (Communication between Pi & STM, Location/Mapping, Waypoint)

Code UART between Pi & STM 24-Jan-22

GPS location 24-Jan-22

Lidar Mapping 24-Jan-22

Terminal waypoint navigation 24-Jan-22

Week 4 (Waypoint)

Waypoint Navigation & Testing 31-Jan-22

Week 5

Set up real sense camera on Pi 7-Feb-22

Set up Depth Sensing 7-Feb-22

Retrieve data from depth sensor camera 7-Feb-22

Develop obstacle avoidance algorithm 7-Feb-22

Start Midterm Report 7-Feb-22

Midterm - Week 6

Midterm Exams 14-Feb-22

Retrieve data from depth sensor camera 14-Feb-22

Develop obstacle avoidance algorithm - code avoidance movement 14-Feb-22

Generate livestream on RVIZ/OpenCV 14-Feb-22

Recess Week - Week 7 (Code Obstacle Avoidance)

Integrating Testing	21-Feb-22
Waypoint Control	21-Feb-22
Pool Test	
Midterm Report Submission	25-Feb-22
Midterm Presentation Slides Submission	25-Feb-22

Week 8 (Code Obstacle Avoidance)

Integrating Testing	28-Feb-22
Pool Test	
Waypoint Control	28-Feb-22

Integration & Testing - Week 9 to 12

Start of Intensive Site Testing	7-Mar-22
Start Report & Slides	7-Mar-22
Fine tune code	7-Mar-22
Pool Test	
Reservoir Test	
ST Engineering Meeting	
Final Presentation - Week 13 (4/4/2022)	
Final Presentation Slides Submission	
Final Presentation	
Final Report Submission	
Final Work Source Submission	
Final Exam - Week 14	
Final Exams	11-Apr-22

Figure 3: Trimester 2 Schedule

2.6. Testing and Validation

The testing and validation criteria of the project for trimester 1 has been defined early on during the project life cycle. The testing and validation criteria and scenarios are as listed in the respective sections below.

2.6.1. Trimester 1 Testing Criteria

Testing of Hull Design

- Testing in a pool - ability to float in water
- Movement of surface vessel under remote control (Port, Starboard)

Testing of Lidar

- Testing on land - changes in the readings
- Ability to detect obstacles from various distances and directions
- From 50cm to 100cm
- Increments of 45° (E.g., 0°, 45°, 90°, etc)

Camera Functionality

- Ability to show video feed via Raspberry Pi Cam

2.6.2. Trimester 1 Testing Scenarios

Testing of Hull Design

- Demonstrate forward Movement of ASV
- Demonstrate movement of ASV starboard side
- Demonstrate movement of ASV port side
- Demonstrate U-turn movement of ASV

Testing Lidar

- Object placed 50cm away. Positioned in increments of 45° from heading
- Object placed 100cm away. Positioned in increments of 45° from heading
- Multiple objects placed within 50-100cm positioned in various angles

Testing of Camera

- Generating of live feed on laptop

2.6.3. Trimester 2 Testing Criteria

Testing of Real-Sense Camera

- Testing on land and water
- Ability to generate visual and data stream
- Ability to detect obstacle based on predefined criteria (Red Obstacle)

Testing of Terminal Control

- Testing of the basic directional control
- Testing of the command switch between RC and Terminal Control
- Ability to display data sent from terminal

Testing of IMU

- Able to display roll, pitch and yaw values

Testing of GPS

- Able to retrieve current latitude, longitude, date, time, speed, heading from GPS module

Simulated Waypoint Navigation

- Ability to Move to targeted latitude and longitude

Integration Testing

- Able to demonstrate Simulated Waypoint Navigation
- Able to demonstrate terminal control and redundancy control
- Able to display IMU data, Camera Stream and obstacle detection, and Lidar Data simultaneously

2.6.4. Trimester 2 Testing Scenarios

Testing of Real-Sense Camera

- Generate live feed on laptop
- Recognize red obstacles

Testing of Terminal Control

- Demonstrate ASV basic directional control
- Demonstrate RC and Terminal Control mode switch
- Generate data sent from terminal

Testing of IMU

- Accurate roll, pitch and yaw values displayed on the terminal

Testing of GPS

- GPS module shows accurate current latitude, longitude, date, time, speed, heading

Simulated Waypoint Navigation

- Ability to Move to targeted latitude and longitude

Integration Testing

- ASV moves to Waypoint
- Demonstrate terminal control and redundancy control
- IMU data, Camera Stream and obstacle detection, and Lidar Data simultaneously generated on the terminal

2.7. Responsibility Matrix

The roles and tasks are distributed equally between the members and are done so according to the strengths of each member. This is to ensure efficiency, reliability, and quality in our work. The roles undertaken by each member is as shown in the Responsibility matrix below:

Member \ Tasks	Abrisam	Claudia	Haseem	Raymond
Team Leader	X			
Programming	X		X	
Design	X			X
Research	X	X		
Commercial		X	X	
Mechanical Engineering		X		X
Electrical Engineering			X	X

Table 2 Responsibility Matrix

2.7.1. Roles and Responsibilities

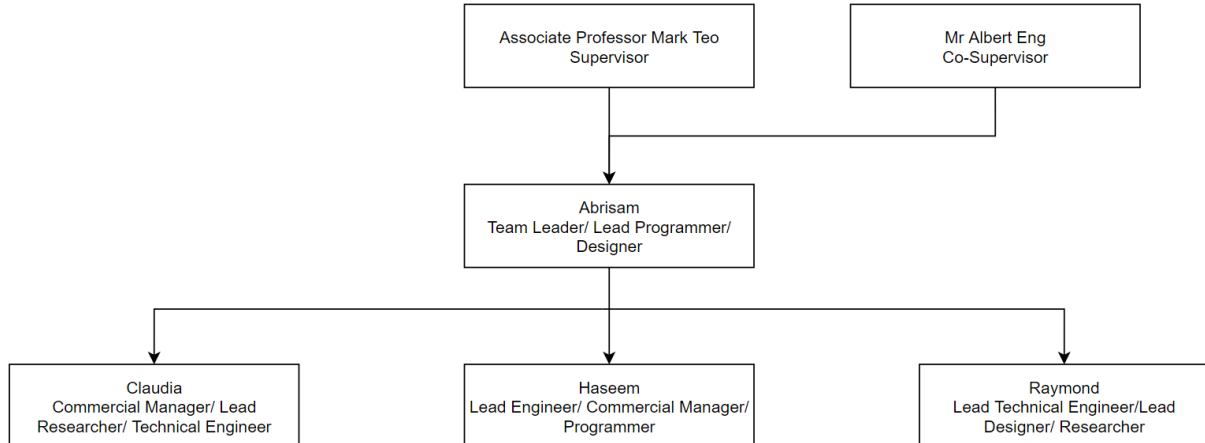


Figure 4: Organisation Tree

Role	Description
Team Leader	Leads the team during the Project and assigns task accordingly
Commercial Manager	In charge of procurement and budget management. The commercial Manager will also assist the Team Leader as a Point of Contact for communication between the Team and the Supervisor
Programmer	In charge of system architecture design and development of algorithm

Researcher	In charge of researching and selection of equipment to be used for the Project
Designer	In charge of overall design and layout of the ASV. Include designing of PDB and PCB boards.
Engineer / Technical Engineer	In charge of all electrical and mechanical aspects such as layout of wiring of equipment and construction of hull. This role will also entails providing input into the design of various equipment such as PDB and PCB boards. This role will also lead the testing and troubleshooting of the ASV

Table 3: Description of Roles

2.8. Budget

The budget for this project is SGD\$1000 spread out over two trimesters. The budget will be funded by SIT.

3. Overall Design

This section will provide the architecture and design for our ASV. The architecture and design will be described via system architecture and wiring diagrams.

3.1. Architecture

The design of our ASV is to have a simple hull installed with 2D Lidar sensor for environment monitoring and obstacle detection. The ASV will also be equipped with Wi-Fi communication module and a GPS positioning system to enable autonomous movement via waypoints. These hardware components will be connected to an operating system and a microcontroller for the processing of data.

To describe the architecture of our ASV, we have divided the project into four areas: the hull, the operating system and the microcontroller, sensor/obstacle detection and navigation. The architecture for each of these areas are detailed later below.

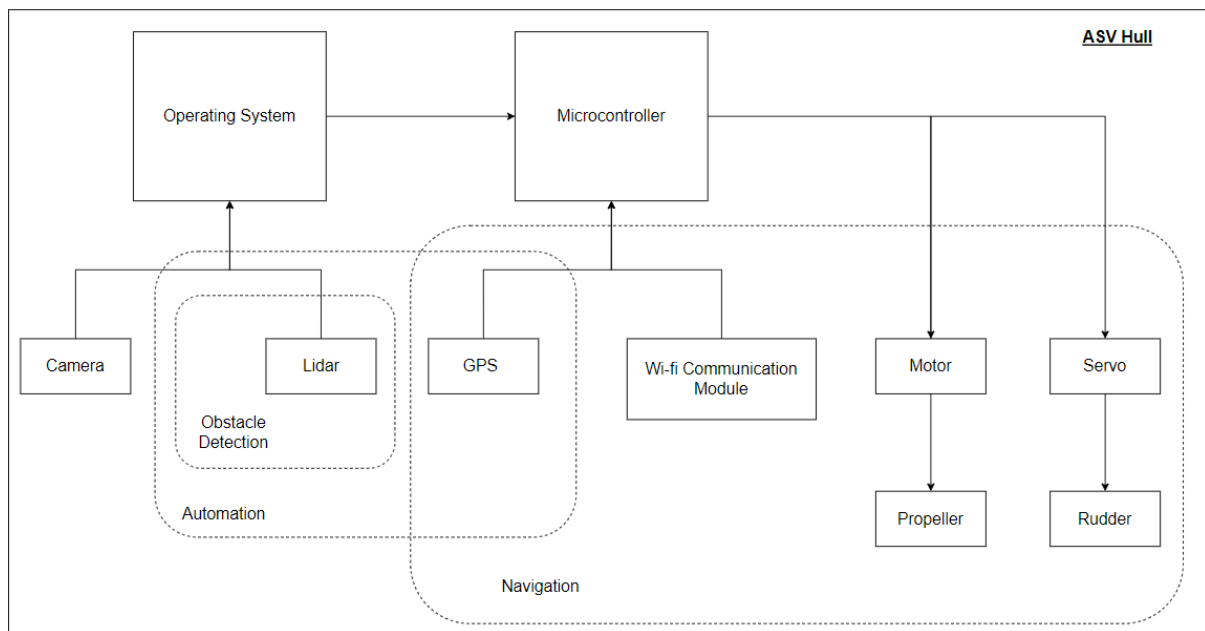


Figure 5: System Architecture Diagram for Trimester 1

Figure 5 above shows the various key components of our ASV with relevance to its functionalities.

3.1.1. Hull

The preliminary hull will be designed to be a simple hull designed to house the hardware equipment required. The hull will also provide protection of the hardware equipment from the water body.

The hull design and fabrication will be highlighted later on in the report

3.1.2. Operating System and Microcontroller

The operating system will be the main driver for the camera and Lidar sensor. The chosen OS is a Raspberry Pi 4 board, and will be connected to the microcontroller, where the data will be processed and used for autonomous capabilities.

The microcontroller will be the main processing hub, deciding the operations of the ASV via the developed algorithm. The microcontroller will be connected to the various subsystems such as the operating system and positioning system. The chosen micro controller for our project is the STM32 STM32H723ZG board.

3.1.3. Sensor/Obstacle Detection

A 2D Lidar sensors capable of 360° range detection will be installed on the ASV to facilitate the monitoring of the environment as well as obstacle detection. The Lidar sensor selected is the RPLIDAR A1-M8 sensor.

3.1.4. Navigation

For navigation control of the ASV, the ASV will be fully autonomously. This attribute can be achieved by using modules such as IMU, Wi-Fi module for communication, and a GPS module.

For Trimester 1, the ASV will be controlled via remote control. This can be achieved by the use of RC controllers or Wi-Fi communication modules. The end goal of the ASV project is to ensure the ASV is fully autonomous and is able to navigate to specified waypoints defined by the user through a control terminal.

The GPS module selected is the X-NUCLEO-GNSS1A1 GPS module from STM.

The IMU module chosen is the Pololu MinIMU-9 v5.

The Wi-Fi module chosen for communication is the XBee Wi-Fi (S6B) XB2B-WFWT-001.

DC and servo motors will also be used to power the propeller and rudder to enable to propulsion and changing directions of the ASV.

3.1.5. Other components

Other components include an LCD screen for troubleshooting and batteries for operation of the ASV. A detailed list and description of all components used for the project will be provided in later sections.

3.2. System Architecture Diagram

3.2.1. System Overall Architecture Diagram

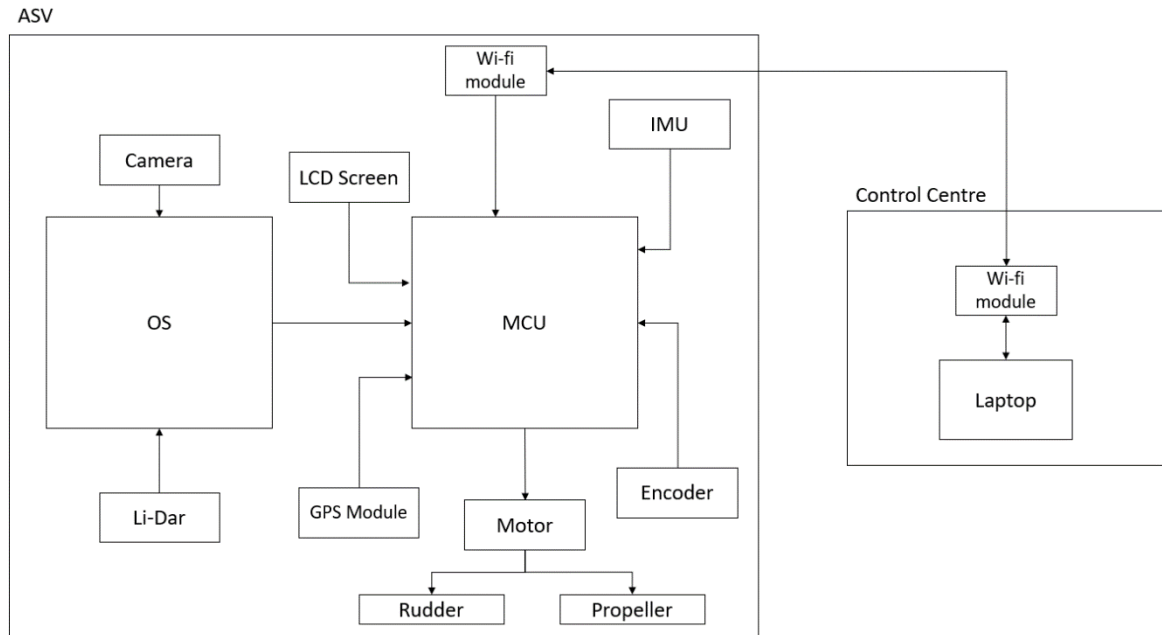


Figure 6: Overall System architecture diagram

Figure 6 above shows the overall system architecture diagram for our ASV. The control centre will be the user, inputting waypoints to which the ASV will autonomously move to.

The ASV will contain the Operating System and the MCU, which will be connected to all key components that will enable the environment monitoring and obstacle avoidance and autonomous capabilities.

3.2.2. System Architecture for Trimester 1

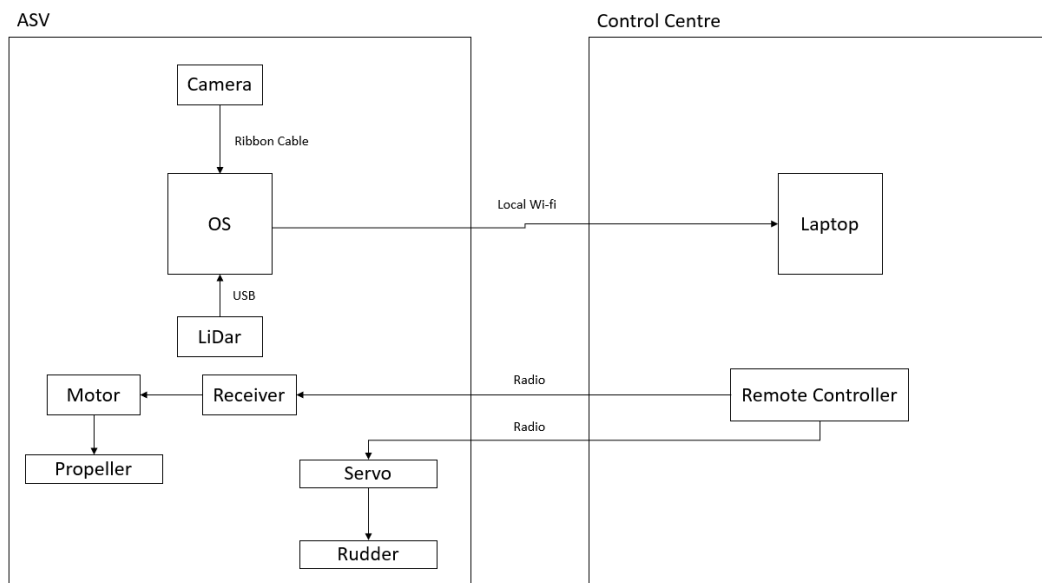


Figure 7: System Architecture for Trimester 1

As the objectives of trimester 1 are progressive milestones towards the main objectives, the system architecture diagram is as shown in Figure 7 above.

The propulsion of the ASV will be controlled by the RC remote controller, thereby making the ASV unmanned.

Only the operating system will be available, driving the camera and Lidar. The streaming of the camera and visualisation of the Lidar data will be streamed to the laptop through a local Wi-Fi network.

3.3. Wiring Diagram

This section will provide a high-level indication of how the ASV will be wired for trimester 1.

3.3.1. Overall Wiring Diagram

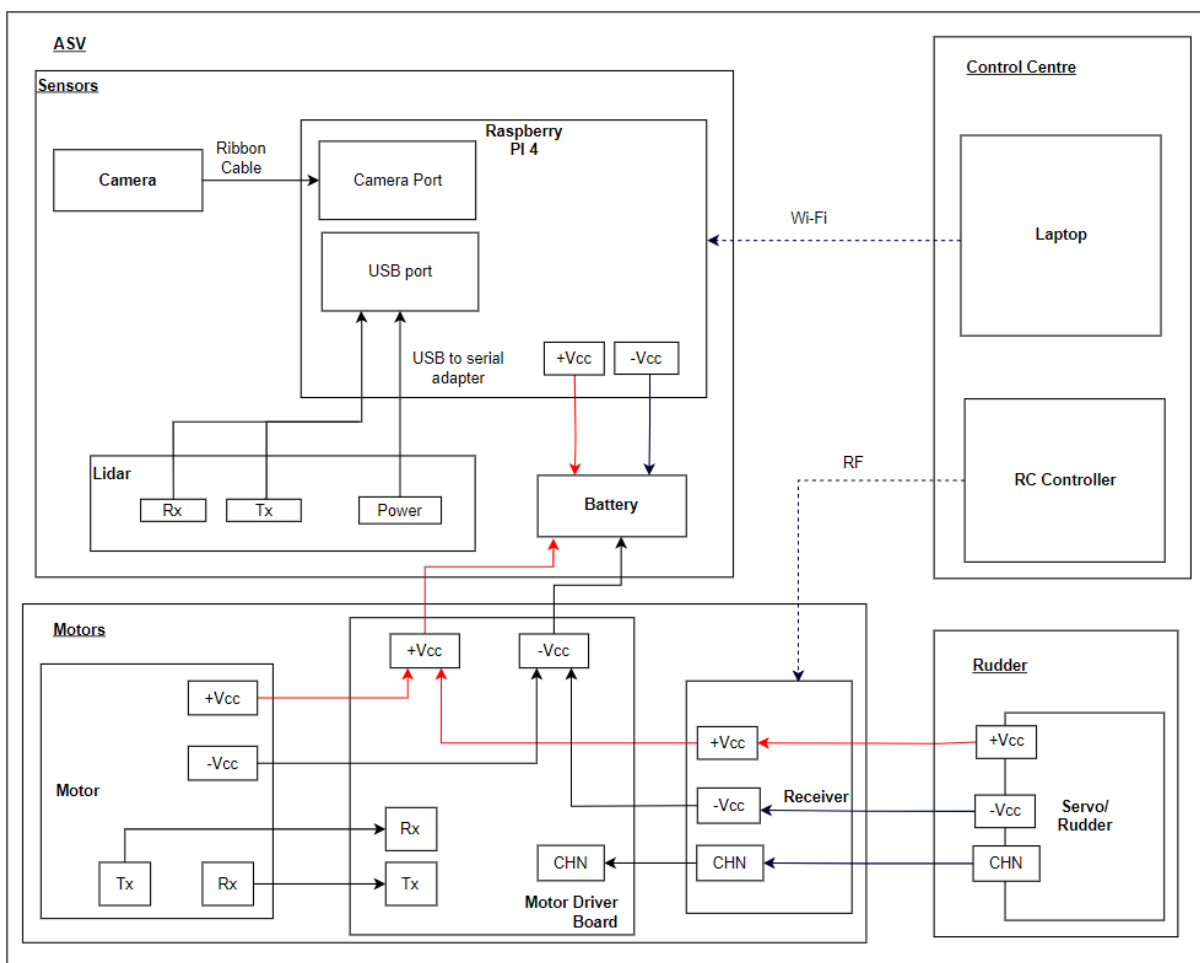


Figure 8: Overall Wiring Diagram of ASV for Trimester 1

Figure 8 above shows the overall wiring Diagram of our ASV for trimester 1. Further subsections below will show the wiring of each individual component

3.3.2. Sensors

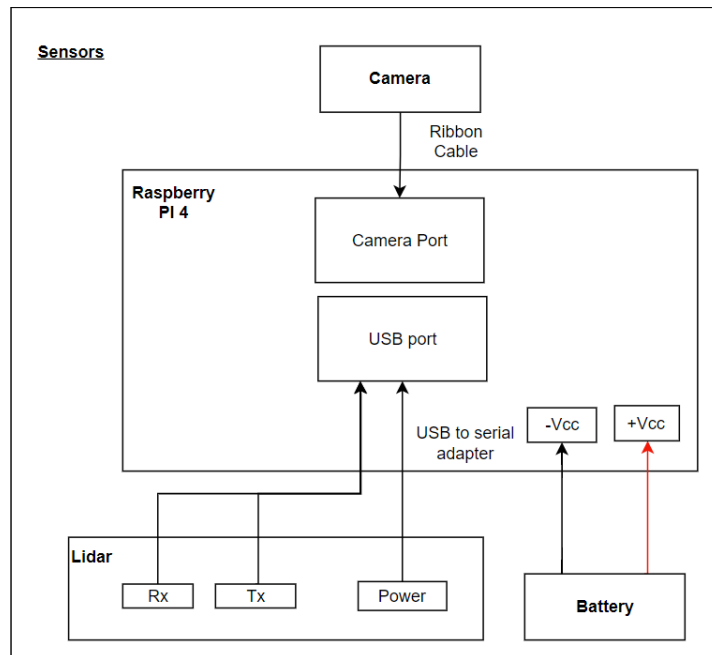


Figure 9: Wiring Diagram of Sensors

Figure 9 above shows how the camera, and the Lidar Sensor will be wired to the raspberry pi 4 operating system.

3.3.3. Motors

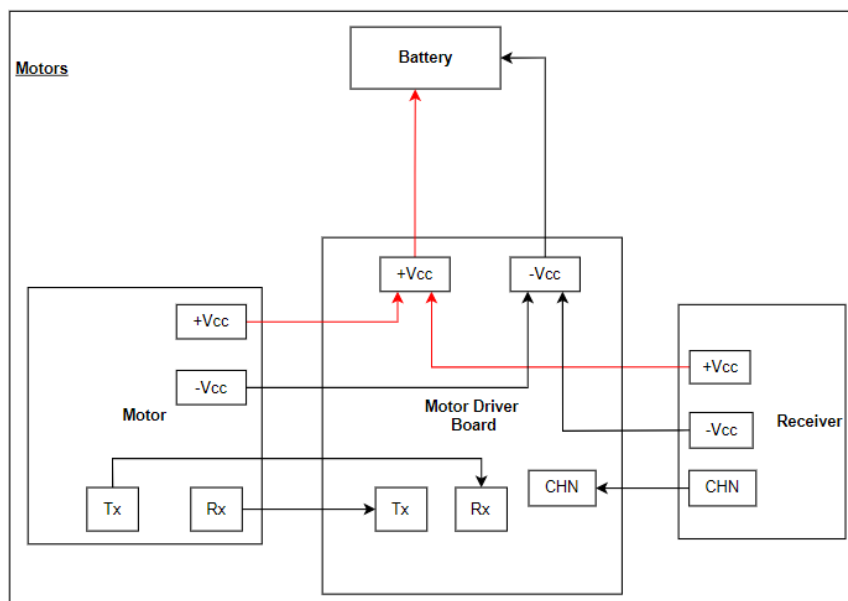


Figure 10: Wiring Diagram for motors

Figure 10 above shows the wiring diagram for the motor, motor driver board and receiver, which would help with the propulsion and movement of the ASV.

3.3.4. Wiring diagram for rudder

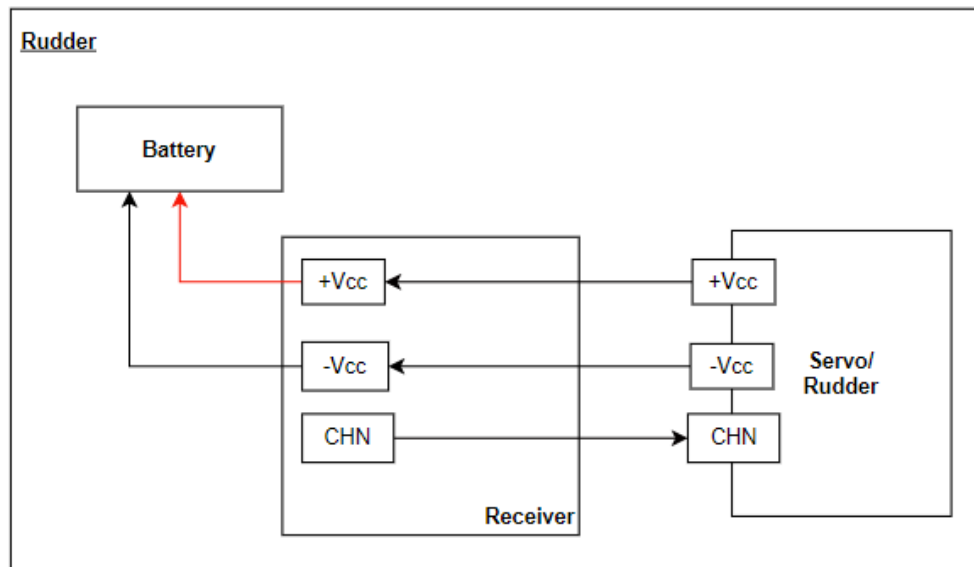


Figure 11: Wiring diagram for rudder

Figure 11 above shows the wiring diagram for the rudder, to enable the changing of direction of the ASV.

3.4. Work Breakdown Structure

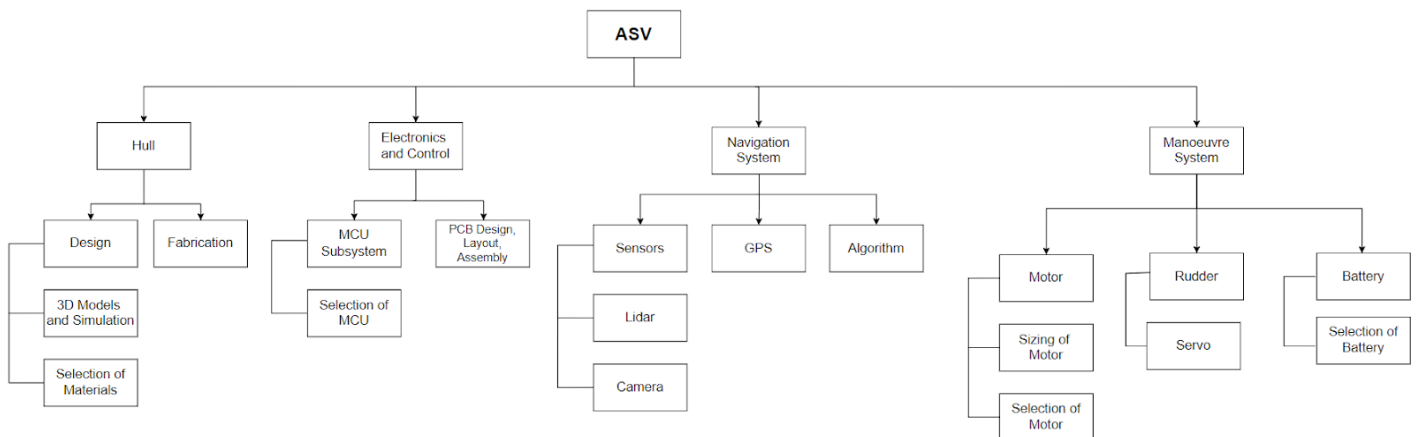


Figure 12: Work Breakdown Structure

Hull:

- Design
 - 3D Models and Simulation
 - Selection of materials
- Fabrication

Electronics and Control:

- MCU Subsystem
- Selection of MCU
- PCB Design, Layout, Assembly

Navigation System:

- Sensors
- GPS
- Lidar/Camera
- Algorithm

Manoeuvre System:

- Motor
 - Sizing of Motor
 - Selection of Motor
- Rudder
- Servo
- Battery
 - Selection of Battery

4. Component Selection

In this section, the selected components of the ASV will be detailed, and the reasons for their selection will also be established. Each component was compared against a few other competitive models, to ensure that the best and most efficient modules for our ASV project were selected.

4.1.1. MCU Selection

The MCU unit is an essential component of the ASV. The MCU will process all the various data received and make decisions for the ASV based on the algorithm that was developed. Therefore, it is important to select an MCU that matches the project requirements. The MCU must be fast enough, have sufficient ports and interrupts and have the required communication channels that would be used for our project.

	STM 32H723ZG	STM F722ZE	STM F411E	STM F429I	NXP 13085	TIVA TM4C123G
Clock speed(3)	Sufficient 6	Fastest 1	Fast 4	Fast 2	Fast 3	Sufficient 5
Power consumption(4)	Low 1	Low 1	Low 1	Low 1	Low 1	Low 1
GPIOs (2)	Sufficient 6	Too many 2	Sufficient 4	Too many 3	Sufficient 5	Little 1
Communication (1)	Sufficient 6	Sufficient 5	Sufficient 4	Sufficient 3	Sufficient 2	Too many 1
Timers(5)	Sufficient 6	Too many 5	Not enough 3	Too many 4	Not enough 2	Not enough 1
Flash memory (6)	Sufficient 6	Many 4	Many 5	Too many 1	Many 3	Many 2

Total Score/126	106	65	73	45	53	39
-----------------	-----	----	----	----	----	----

Table 4: Comparison table for MCU

The table above evaluates each considered MCU, with the features ranking from 1-6, 6 being most suitable 1 being least.

Feature Rating (n) = Features (n) X evaluation (n)

“n” is the number of features that we are evaluating on.

MCU rating = Sum of all feature rating on an MCU

With the above rating, the STM32H723ZG is selected as it is deemed most suitable for the application. This STM board is able to support the various communication protocols that we require, and it also has sufficient ports to be used for the different subsystem modules.

The features of the STM32H723ZG are as shown below:

Board	Communication protocols available	Number of pins	Special Features
STM32H723ZG	UART: 5 SPI: 6 I2C: 5	114	Low power

Table 5: Features of the STM32H723ZG



Figure 13: STM32H723ZG

4.1.2. Operating System Selection

The operating system is another important module as it will be the system that will drive the camera and the Lidar sensor. An operating system is required to enable the running of the driver program for the camera and Lidar sensor, which eliminates the difficult task of writing our own driver file.

As such, the operating system would require a high RAM, and able to support connections to the Lidar and camera.

As such, the Raspberry pi 4 8GB RAM was chosen. After research was carried out, the raspberry pi 4 was found to be the most suitable as it is able to connect to the camera and Lidar sensor directly via a ribbon cable and USB port respectively. The Raspberry Pi 4 is also compatible with the STM32H723ZG. This is essential as the OS needs to be connected to the MCU to transfer the data gathered to the MCU so that the MCU will be able to enable the autonomous movement of the ASV.

The features of the Raspberry Pi 4 are as shown below:

Features	Raspberry Pi 4
Processor	1.5GHz Broadcom BCM2711 SoC
Memory	4GB LPDDR4
USB Ports	Two USB 3.0 ports, two USB 2.0 ports
GPIO	40-pins headers

Table 6: Features of Raspberry Pi 4

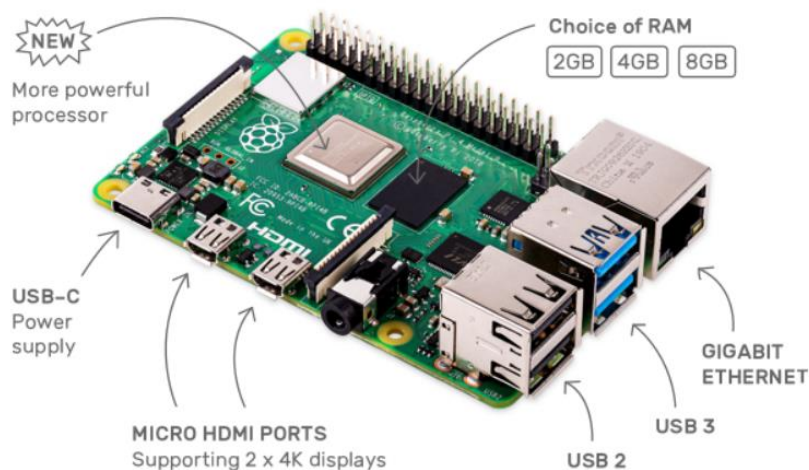


Figure 14: Raspberry Pi 4. Image retrieved from Official Raspberry Pi website

4.1.3. Motor

The motor will allow the vessel to move around and ensure that sufficient torque is provided to move the vessel.

The selected motor for our ASV is the Mabuchi RS-540 Torque Tuned Motor.

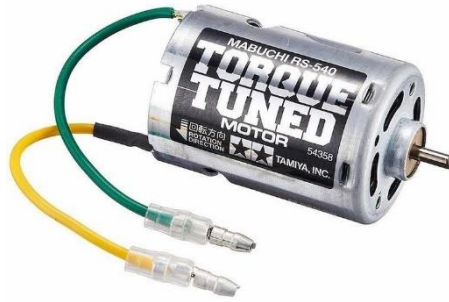


Figure 15: Mabuchi RS-540 Torque Tuned Motor

Features of this motor are as listed below:

- Standard 540-Turns: 27T Top speed: 24.3km/h
- Torque-Tuned Motor: Turns: 25T Top Speed: 26km/h
- Sport-Tuned Motor: Turns: 23T Top Speed: 29.1km/h

4.1.4. Wi-Fi Module Selection

The Wi-Fi module is an important module for the ASV as it acts as the communication channel between the ASV on water and the control centre on land. The ASV will transmit data over Wi-Fi to the control centre, for monitoring of the environment around the ASV through the camera and Lidar sensor.

The control terminal would also send the waypoints for the ASV to travel via the Wi-Fi module.

The selected Wi-Fi module is the XBee Wi-Fi (S6B) XB2B-WFWT-001.

XBee devices communicate with each other over the air, sending and receiving wireless messages. The devices only transfer those wireless messages; they cannot manage the received or sent data. However, they can communicate with intelligent devices via the serial interface.

XBee devices transmit data coming from the serial input over the air, and they send anything received wirelessly to the serial output. Whether for communication purposes or simply for configuring the device, a combination of both processes makes XBee communication possible. In this way, intelligent devices such as microcontrollers or PCs can control what the XBee device sends and manage incoming wireless messages.



Figure 16: XBee Wireless Communication

Features of the XBee are as shown in the table below:

Features	XB2B-WFWT-001
Manufacturer	Digi
Frequency	2.4GHz
Data rate	11 Mb/s, 54 Mb/s, 65 Mb/s
Interface type	SPI, UART
Operating Supply Voltage	3.3V

Table 7: Features of the XB2B-WFWT-001

4.1.4.1. Configuration of XBee

Two XBee devices are used for the project, the first step is to pair the two devices. One device will be the IBSS creator which will act as a master. The IBSS joiner will act as the slave. For the configuration, ensure that the destination IP address must be the same as the module IP address of the other device. The configured the baud rate setting according to the baud rate setting that is being used for the code.



Figure 17: XB2B-WFWT-001

4.1.5. GPS Module

The GPS module will enable the ASV to determine its current location. The GPS module will be vital in enabling the autonomous navigation of the ASV via specified waypoints.

The GPS module must be compatible with the MCU as it will need to communicate and transfer the location of the ASV to the MCU.

The selected GPS module is the X-NUCLEO-GNSS1A1. This module is a GNSS expansion board based on Teseo-LIV3F module for STM32 Nucleo. A key factor in selecting this GPS module is that it is compatible with STM32H723ZG, the MCU that was chosen.

Some features of the X-NUCLEO-GNSS1A1 are listed in the table below:

Features	X-NUCLEO-GNSS1A1
Interfaces	UART, I2C
GNSS protocol	NMEA
Assisted GNSS	Predictive autonomous Predictive server-based Real-time server based

Table 8: Features of X-NUCLEO-GNSS1A1



Figure 18: X-NUCLEO-GNSS1A1

4.1.6. Lidar Sensor

The Lidar sensor will be the main obstacle detection for the ASV. Several considerations were made for the type of sensor to be used for ASV, such as ultrasound, sonar and Lidar. However, after extensive research, the Lidar was selected as it will enable a full omnidirectional range of detection (360°). Lidar would also be less susceptible to reflection from water which will affect the detection of obstacles.

The selected Lidar sensor for our ASV is the RPLIDAR A1. The APLIDAR A1 is a 2D Lidar sensor based on laser triangulation ranging principal and uses high-speed vision acquisition and processing hardware developed by Slamtech. Some of the features of the chosen Lidar sensor is as shown in the table below:

Features	RPLIDAR A1
Range	12 metres
Range in degrees	360°
Sample rate	Up to 8000 times
Scanning rate	2-10Hz
Angular resolution	1°

Table 9: Features of Lidar Sensor



Figure 19: RPLIDAR A1

4.1.7. Finalised BOM List

The finalised BOM list can be found in Appendix A.

5. Trimester 1 – Development

5.1. Hull Design

The team first started with the design of the hull, which would house all the electrical components and serve as the test platform.

5.1.1. Preliminary Draft

Taking into consideration the weight of the electric components which the vessel would have to bear, a trimaran design was selected as it offered better stability as compared to a monohull.

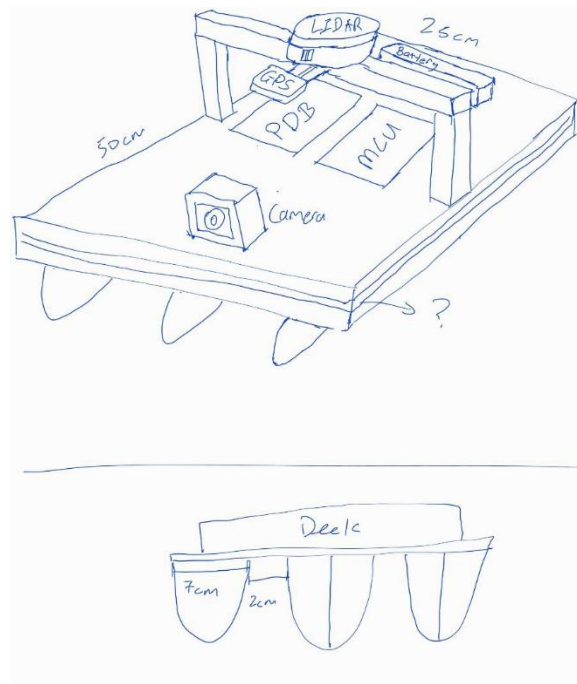


Figure 20: First Preliminary Draft Design

The electrical components would be housed on the deck while a raised structure would house the Lidar and GPS module, with the camera situated in front.

5.1.2. First Iteration

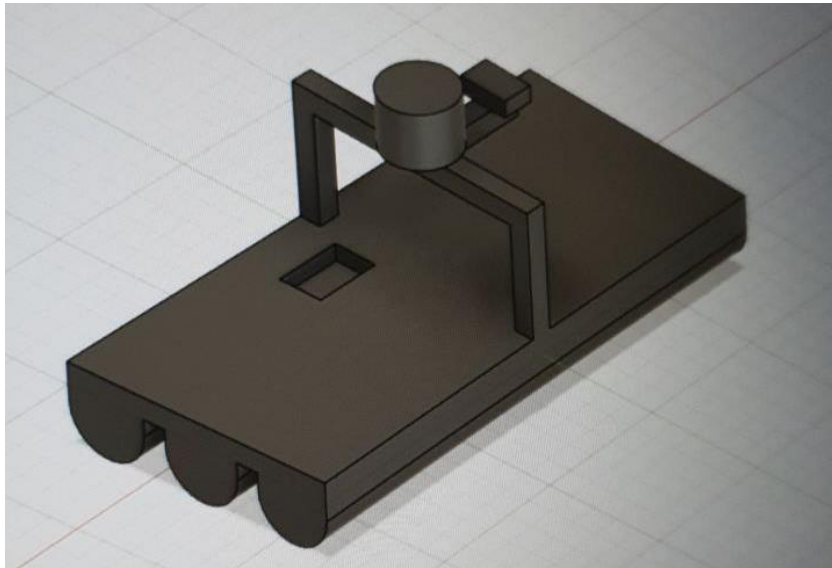


Figure 21: CAD model of first Iteration

Continuing the idea of a trimaran, the CAD model in Figure 21 was developed.

Using this model, the calculations for buoyancy and stability of the design were done. However, this design was complex and would require a significant amount of material to 3D print. As such, the design was revised.

5.1.3. Second Iteration

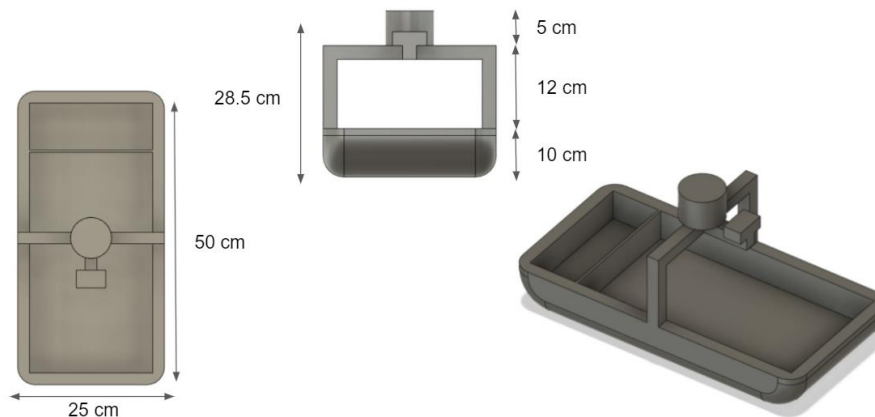


Figure 22: Second Iteration of ASV

After getting inputs on how to improve our hull design from our stakeholders, we continued the idea of a trimaran, and the second iteration of the ASV was developed.

The second iteration of the design was based on the idea of a rectangular box with rounded edges and a structure to house the Lidar, GPS module and camera as shown in Figure 22 above.

This change in shape was mainly due to the buoyancy and stability calculations conducted for the first iteration.

5.1.4. Challenges Faced During Hull Fabrication

The team was looking to 3D print the hull, however after intensive consultation 3D printing in house was ruled out due to the time required for 3D printing.

The team then researched and evaluated alternative methods to fabricate the hull. The following 4 methods were investigated: 3D Printing (External Vendor), Plastic, Acrylic, and Wood.

The main areas considered were Source, Cost, whether assembly was required, and who would be the one to assemble it.

	3D Printing	Plastic	Acrylic	Wood
Source	External	External	In House	External
Cost (SGD)	>\$1000	>\$300	>\$150	>\$200
Assembly required?	No	Yes	Yes	Yes
Assembly done by	NA	Assembled by Vendor	Self-assembly	Self-assembly

Table 10: Hull Fabrication methods comparison

After evaluating all the options available, the team was leaning towards either the plastic or acrylic method. 3D printing was out of the budget while there were waterproofing concerns with wood.

5.1.5. Solution and Final Hull Design

Thankfully, our group was fortunate that our supervisor, Mr Mark Teo, had an unused 3D printed hull that fits our required dimension. We were allowed to take over the hull, which greatly reduced our potential project costs.

However, modifications were required in order to ensure the hull is able to support the equipment and functionalities of our ASV.

Certain areas of the hull required waterproofing, and the hull required the addition of 2 decks platform where all the electronics will be mounted. The bottom of the hull has 2 stabilizers 'fins' to allow the ASV to turn in ease during starboard or port-side manoeuvres as shown in Figure 23 below.



Figure 23: The provided 3D printed Hull design

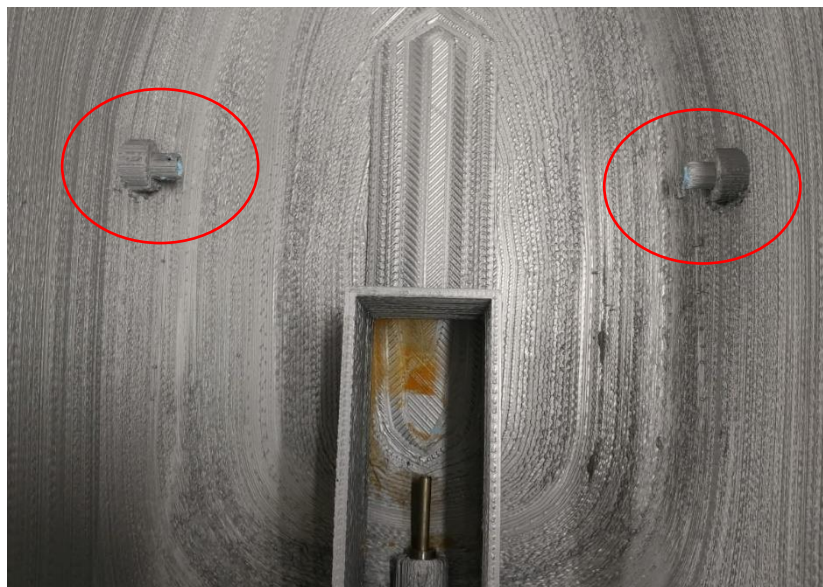


Figure 24: Holes for water pump

The holes on the bottom of the hull as shown in Figure 24 above were meant for the initial design of the 3D printed hull to serve the purpose of the water pumps.

However, since the ASV does not have any requirements to utilise water pumps, the team planned to seal the holes permanently using silicon glue fills to prevent any water from entering, which could potentially sink or damage our electronic circuits on-board.



Figure 25: Hole for propeller shaft

Holes for a water pump and the hole for the propeller shaft were previously sealed with Blu-Tack as a temporary sealing measure while the 3D printed hull was tested for its buoyancy.

The blutack was removed and a more secure sealant such as silicon glue is used to ensure proper waterproofing to prevent any water from entering which could potentially sink or damage our electronic circuits on-board.

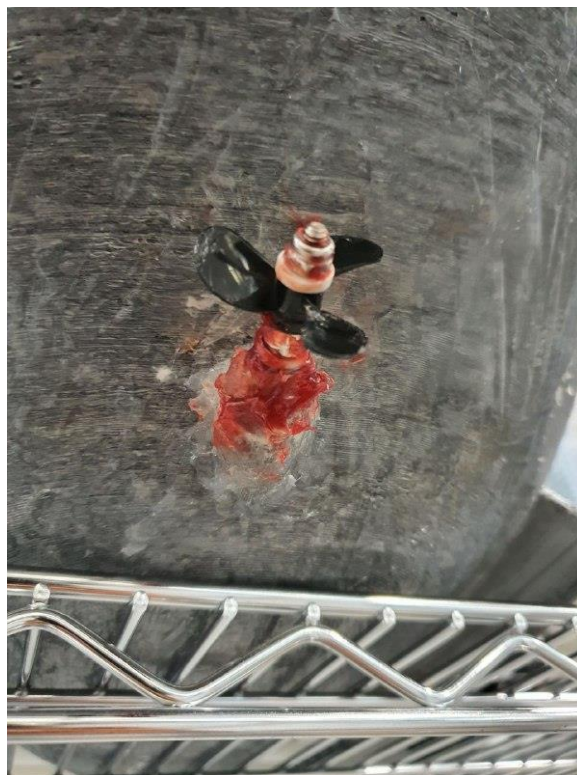


Figure 26: Waterproofing the propeller shaft hole

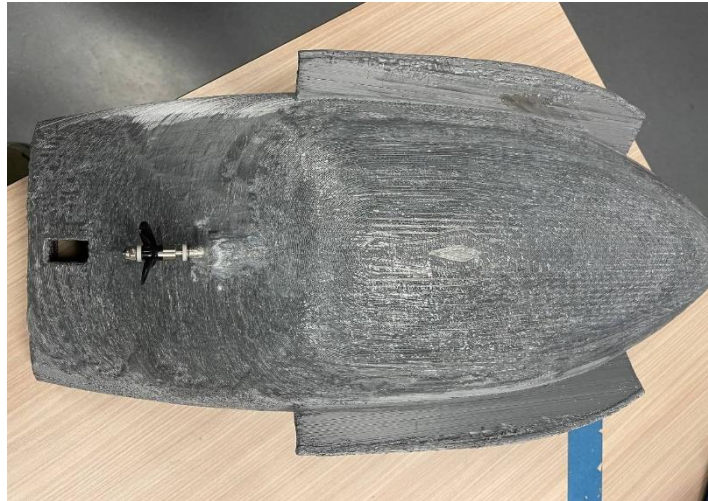


Figure 27: Underside of hull with silicon applied

Additionally, silicon was applied to the underside of the hull as an additional waterproofing measure.

5.2. Equipment Housing Deck

The aim of the design is to keep the electronic components at an elevated height above the motor. Should water leak in via the propeller shaft, the electrical components would be safe.

The housing deck would also be required to support all the electrical components and ensure that they are sheltered from any water. The housing deck would also need to have mounts for the camera and the Lidar sensor.

5.2.1. Design of Housing Deck

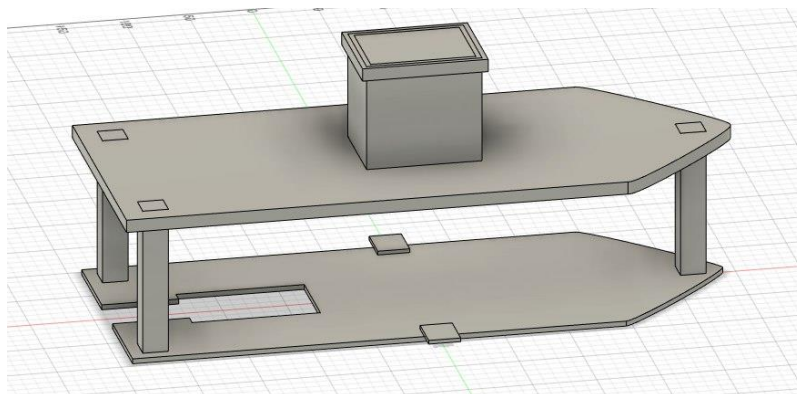


Figure 28: CAD model of Housing Deck

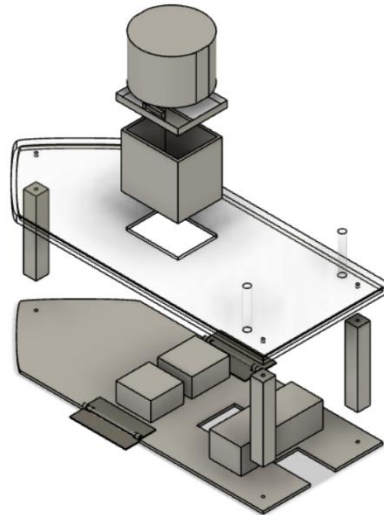


Figure 29: Exploded View of Housing deck with components

The design has two decks, with the lower deck supporting the components and the upper deck acting as the lid that sits on top of the hull. Three pillars will connect the lower lid to the upper lid to ensure modularity and ease of removal, as well as protection against conditions like rain.

5.2.2. Fabrication of Housing Deck

To house the ASV electronic components, we had to design our very own 2 platform deck with camera and Lidar mount.

We utilised acrylic plates joined with T-slots and fabricate to fit our design. The bottom platform deck is designed to house main electronics such as Battery, Raspberry pi, STM microcontroller and IMU.

The 3 pillars would support an outer lid to protect the electrical components against conditions like rain.

The aim of the design is to keep the electronic components at an elevated height above the motor as shown in the figure below:



Figure 30: Equipment Housing Deck



Figure 31: Housing Deck attached to the Hull

Figure 30 and 31 shows the finished product and how the housing deck would look like when it's attached to the hull.

In addition to this, the housing deck design was modified to fit the camera mount and Lidar sensor.

This was achieved by using acrylic and creating a camera and sensor mount that is to be attached is to the upper lid.

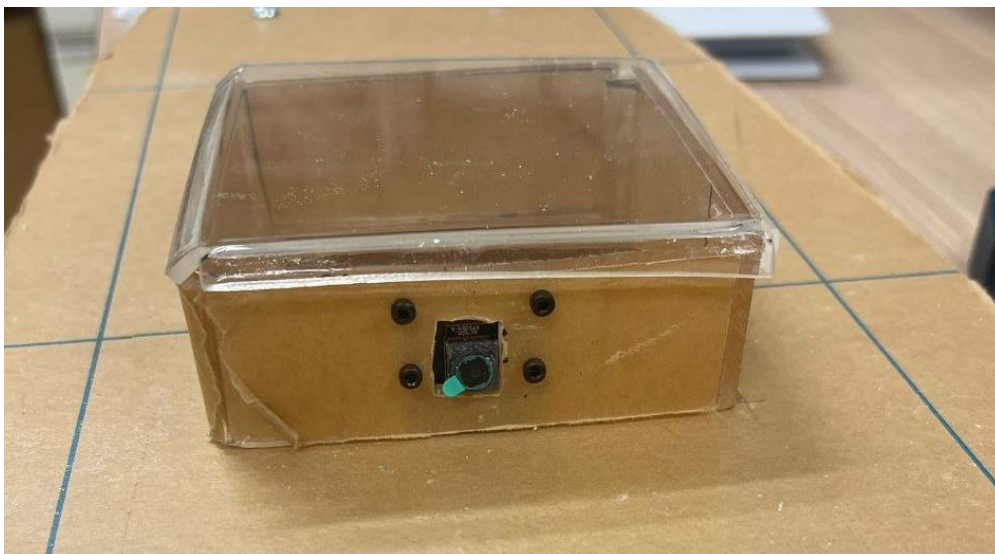


Figure 32: Fabricated camera mount

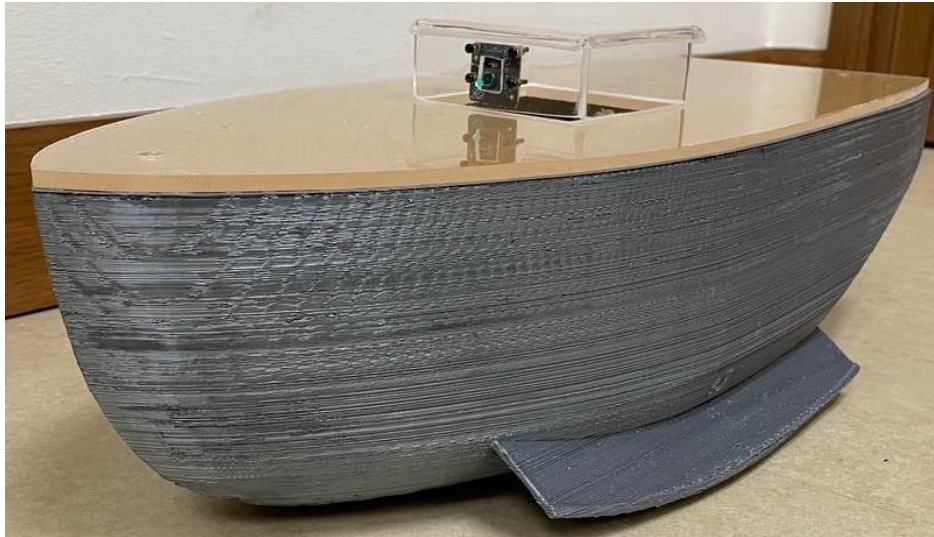


Figure 33: Final ASV Design

Figure 33 above shows the final ASV design. The top platform deck will be mounted with GPS modules, Wi-Fi module and radar antenna where they can be exposed to receive optimum connectivity signals.

The Lidar sensor will be mounted on top of the camera housing giving it a height advantage where it will be able to scan for oncoming/approaching obstacles.

5.3. Camera connection to OS

The Raspberry Pi had to be set up first before the camera could be connected. Both the laptop and the Raspberry Pi had to be on the same network as the Pi streams the output of the camera module over the web via Flask.

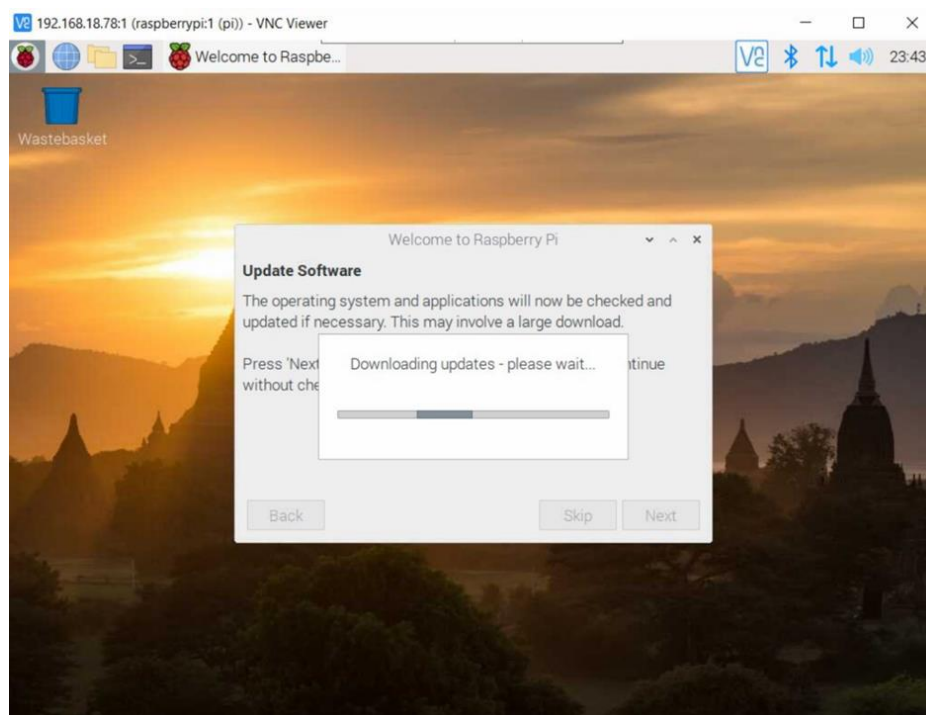


Figure 34: Setting up connection to Raspberry Pi

After connecting the Raspberry Pi to the laptop, the PiCam was then connected to the Raspberry Pi. The code for the camera was written in Python, as it is the language most commonly used for the Raspberry Pi and there is a large and active Python community to consult on issues.

```
11 class VideoCamera(object):
12     def __init__(self, flip = False):
13         self.vs = PiVideoStream().start()
14         self.flip = flip
15         time.sleep(2.0)
16
17     def __del__(self):
18         self.vs.stop()
19
20     def flip_if_needed(self, frame):
21         if self.flip:
22             return np.flip(frame, 0)
23         return frame
24
25     def get_frame(self):
26         frame = self.flip_if_needed(self.vs.read())
27         ret, jpeg = cv2.imencode('.jpg', frame)
28         return jpeg.tobytes()
```

Figure 35: Snippet of the code for the connection of the PiCam

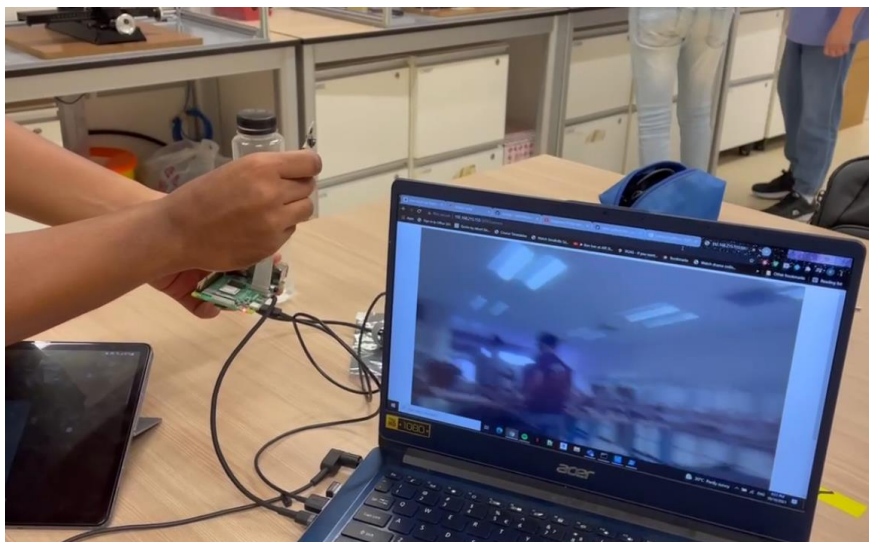


Figure 36: Live Feed generated from the PiCam

From Figure 36 above, we were able to establish a connection to the PiCam via a local network and was able to get live feed.

However, further refinements are still required as the project progresses.

5.4. Lidar

The A1-M8 RPLidar is powered and connected to the OS via USB Cable. The data gathered from the Lidar is sent to the raspberry pi for visualisation in Trimester 1.



Figure 37: A1-M8 Lidar

The Lidar is driven by ROS Noetic, a software released by Robot Operating System (ROS), developed by Open Robotics. ROS provides libraries and tools that help in creating robot applications.

ROS noetic was installed on the raspberry pi to also help visualise the data gathered from the Lidar via the RVIZ software.

The first initial step was to download the required packages to run ROS onto the Raspberry Pi.

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main"
> /etc/apt/sources.list.d/ros-latest.list'
```

```
cat /etc/apt/sources.list.d/ros-latest.list
```

```
deb http://packages.ros.org/ros/ubuntu buster main
```

```
sudo apt-get install -y python-rosdep python-rosinstall-generator python-wstool p
ython-rosinstall build-essential cmake
```

```
rosinstall_generator desktop --rostdistro noetic --deps --wet-only --tar > noetic-
desktop-wet.rosinstall
```

```
sudo src/catkin/bin/catkin_make_isolated --install -DCMAKE_BUILD_TYPE=Release --i
ninstall-space /opt/ros/noetic -j2 -DPYTHON_EXECUTABLE=/usr/bin/python3
```

Figure 38: Example lines of code used to download ROS packages

After downloading the ROS packages, the RPLIDAR packages were installed next, in order to drive the Lidar.

```
sudo git clone https://github.com/Slamtec/rplidar_ros.git
```

```
source devel/setup.bash
```

Figure 39: Example lines of code used to download RPLidar Packages

Once all the required dependencies have been downloaded, and the RPLidar is connected, we can run a simple command to start the visualisation of the RPLidar:

```
roslaunch rplidar_ros rplidar.launch
```

Figure 40: Command used to launch visualisation

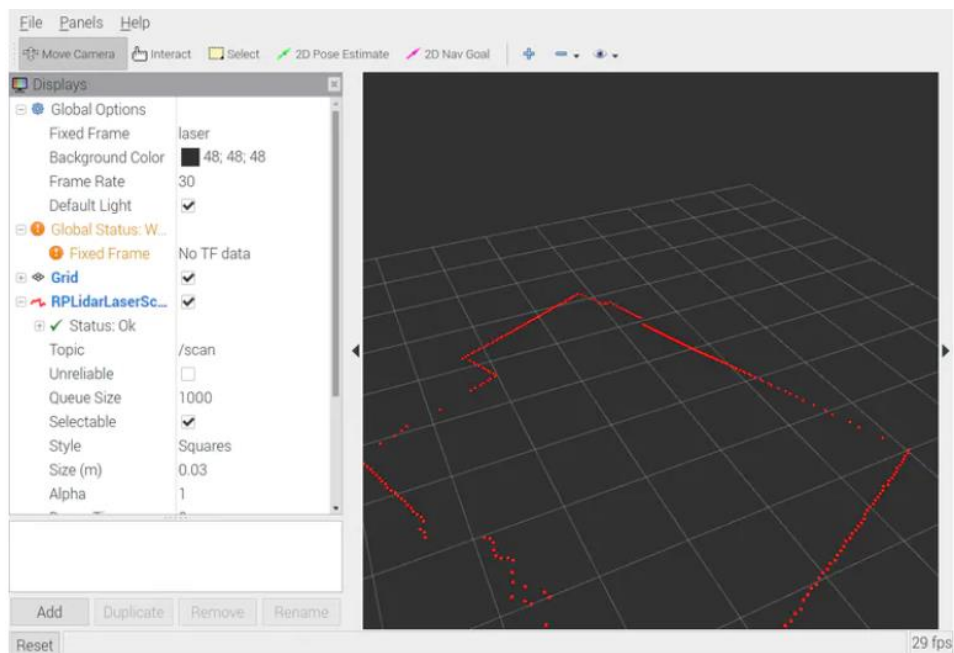


Figure 41: Visualisation of the data from the Lidar

AS we are running the raspberry pi heedlessly, the display of the raspberry pi is done via a local network through a remote desktop.

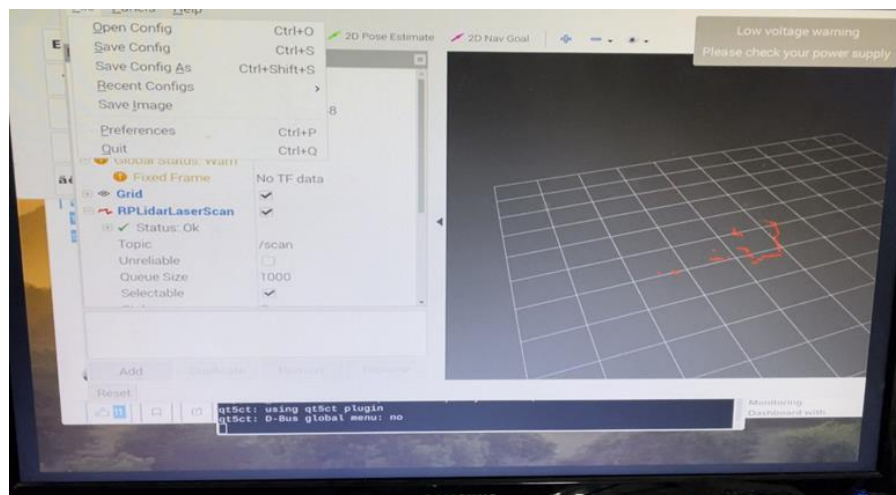


Figure 42: Displaying the raspberry pi to the laptop via remote desktop

Figure 42 above shows the displaying of the raspberry pi and the visualisation of the data onto a laptop via remote desktop, making use of a local network to enable a headless raspberry pi set up.

6. Trimester 1 – Testing

There were 2 major tests conducted to demonstrate the functionality of the system, the Float Test and the Pool Test.

6.1. Float Test and Pool Test

In the Float Test, the motor and battery were fixed onto the Hull itself and weights substituted for the weight of electric components and the Equipment Housing Deck.

The Hull was then placed into a tank of water to test if it is able to float.

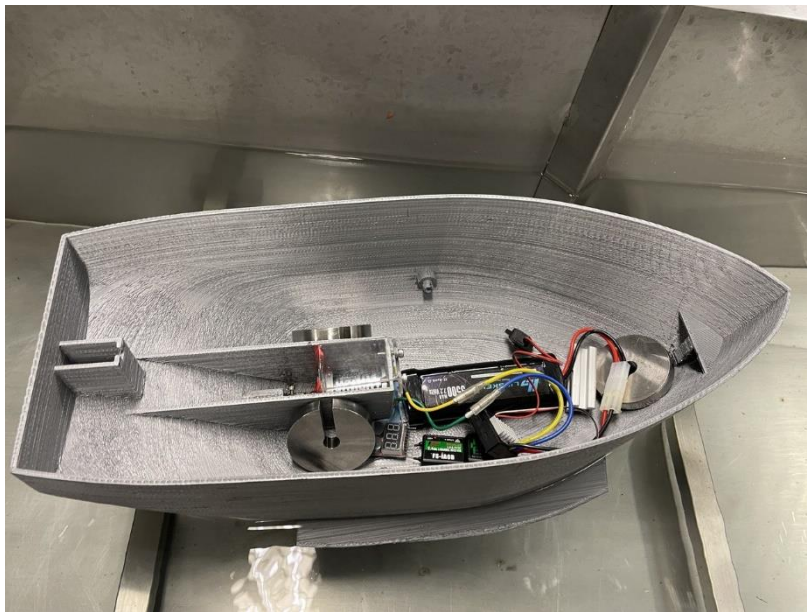


Figure 43: Conducting Float Test

The Float Test was essential as it allowed us to check for any potential areas of water leakage, as well as to confirm our calculations. Calculations such as for the weight and buoyancy of the hull were correct and that it could support the weight of all the components, for the motor so that it could power the movement with all the components.

6.2. Lidar Testing

As part of the requirements of trimester 1, we carried out testing of our Lidar as laid out in the Trimester 1 testing Scenarios:

- Object placed 50cm away. Positioned in increments of 45° from heading
- Object placed 100cm away. Positioned in increments of 45° from heading
- Multiple objects placed within 50-100cm positioned in various angles

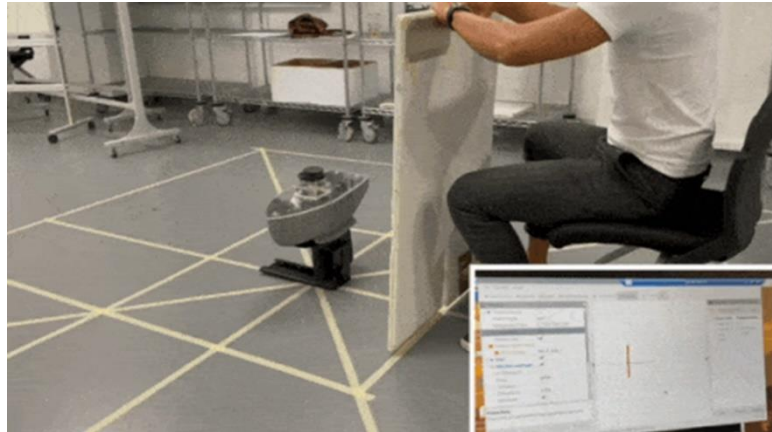


Figure 44: Testing of Lidar from a distance of 50cm



Figure 45: Testing of the Lidar from a distance of 100cm

Figure 43 and 44 above shows the testing of Lidar on land, from a distance of 50cm to 100cm. The figures above also shows that the Lidar was able to detect the object, as shown in the terminal.

Through this testing, we were able to show that our Lidar was working, and that we were able to visualise the data gathered from the Lidar onto our terminal.

6.3. Pool Tests

Multiple Pool Tests were conducted during the duration of the testing phrase.

During the test, the Hull was placed into a pool to test the movement under remote control. The testing location was the swimming pool in Singapore Polytechnic.



Figure 46: SP Swimming pool used for the Pool Test

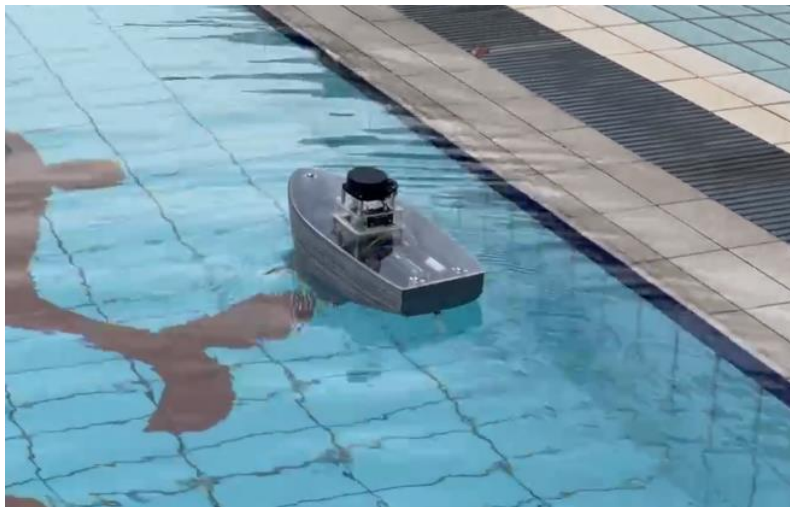


Figure 47: Pool Test

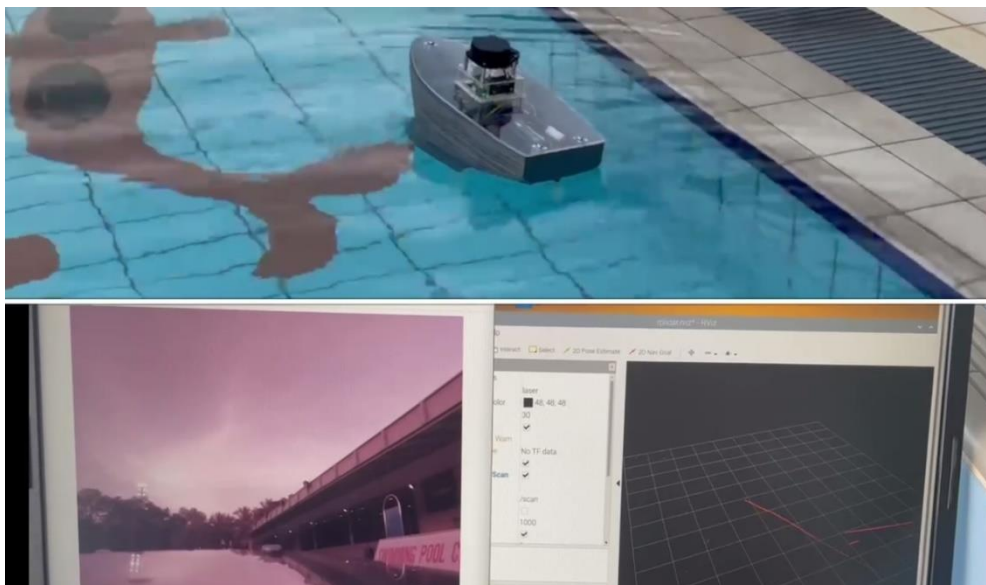


Figure 48: Pool Test with Camera and Lidar

As shown from Figure 47 to Figure 48, the ASV was able to manoeuvre in the pool to a satisfactory level.

From Figure 46, it can also be seen that concurrent with the operation of the ASV, we were able to gather Lidar information as well as the camera live stream.

Through these tests, we were able to demonstrate the basic movement capabilities of our ASV, and also demonstrate the functioning of our environment monitoring sensors, thereby showing that we were able to meet our Trimester 1 objectives.

7. Trimester 2 – Development

7.1. Physical Hardware

7.1.1. Rudder

As the 3D printed rudder used in Trimester 1 kept falling off, the team decided to analyse between using a store-bought rudder or to 3D print another one.

STORE BOUGHT RUDDER

On the functionality standpoint, the store-bought rudder was preferred as it provided a more reliable operation over the 3D printed design.

3D PRINTED DESIGN

The 3D printed design relied solely on the means of friction as a form of connection between servo and rudder, which is not a reliable and permanent solution, having a high risk of failure during operation/transportation would not be ideal.

After weighing the pros and cons of each option, the team decided to go with a store-bought rudder.

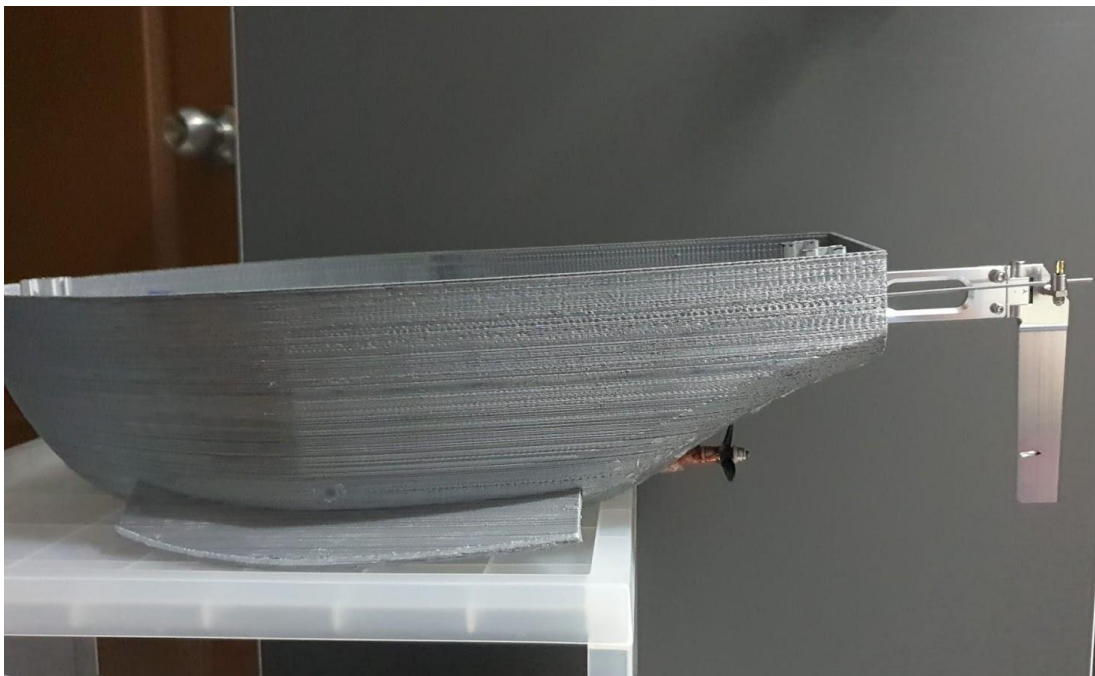


Figure 49: Store bought rudder

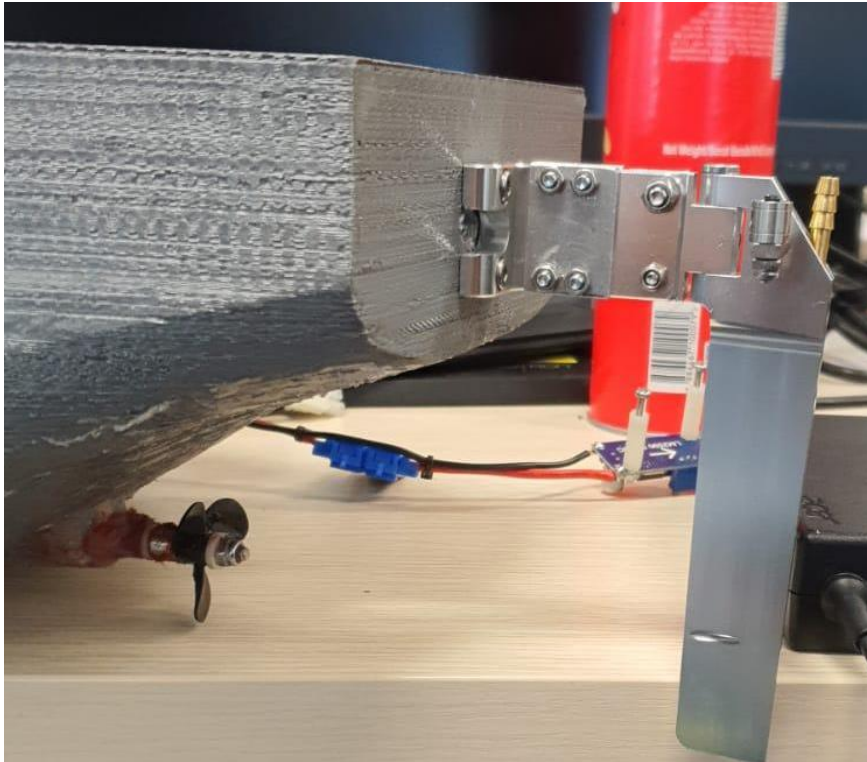
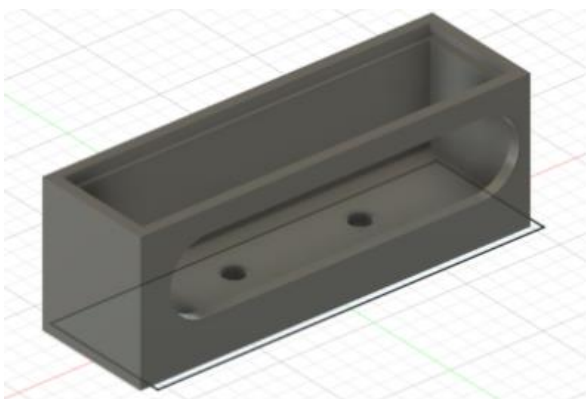


Figure 50: Store Bought Rudder after modification

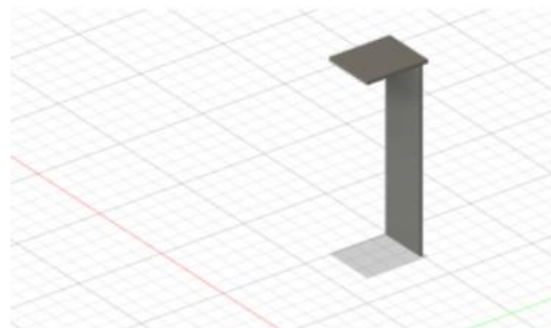
The store-bought rudder had to be modified as it was too long, which would affect the turning of the hull.

7.1.2. Camera Mount

Due to the camera change from the Raspberry Pi Cam to Intel Realsense, a new camera mount had to be designed and printed. After printing it was then mounted to the equipment housing deck.



Camera Holder



Camera Slit (cover)

Figure 51: CAD model of camera mount design

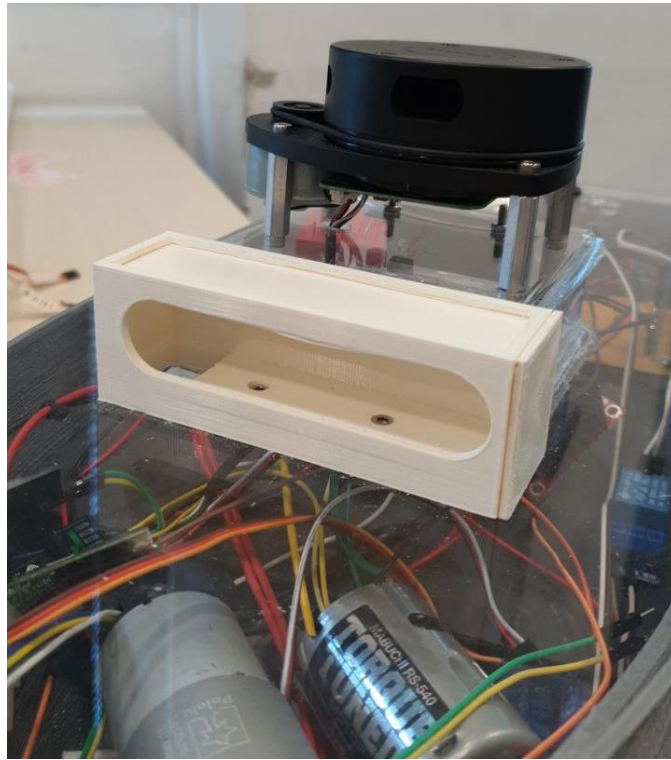


Figure 52: 3D printed camera mount

7.1.3. RC Relay Circuit

The purpose of the RC Relay Circuit is to provide a level of redundancy during autonomous operation, the ability to switch over to a human operated vessel allows us to bring back the vessel safely back to shore any technical troubleshooting.

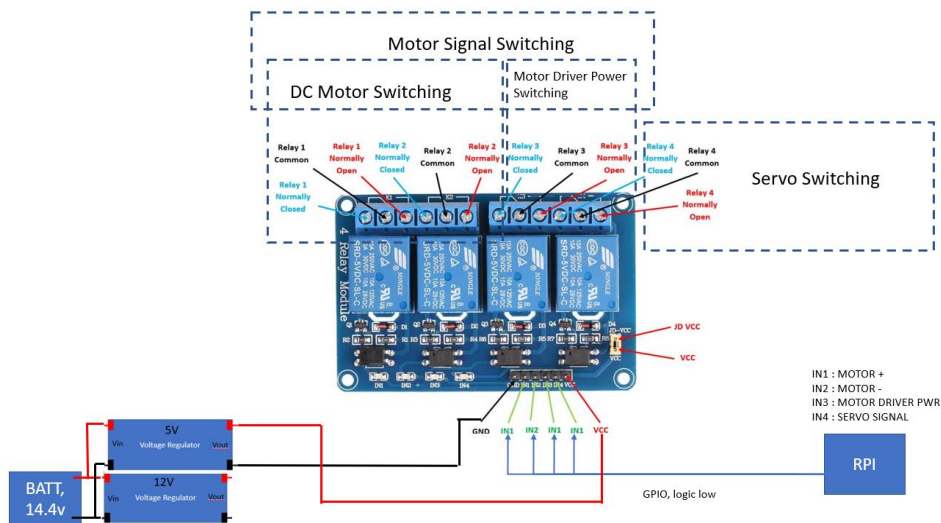


Figure 53: RC Relay Circuit

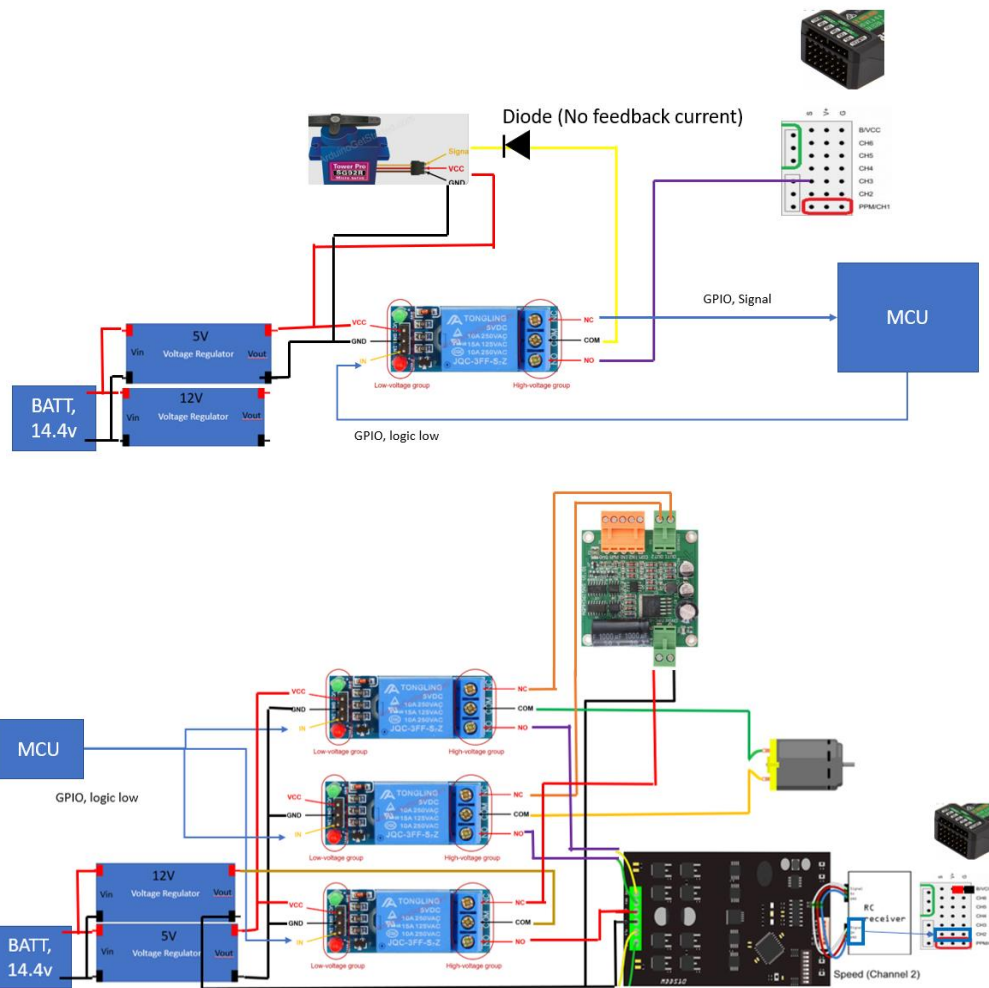


Figure 54: Detailed RC Relay Circuit

The Detailed RC Relay Circuit, provides a visual representation of the physical connection as well as the signal flow between the different modes e.g. Autonomous/Terminal & RC.

7.2. Software

7.2.1. Wi-Fi-Communication

To enable terminal control, we first have to establish the communication between the Laptop and the ASV. This can be done by using two Wi-Fi modules, the XBee S68. The XCTU software was used to configure the modules. The Wi-Fi module connected to the laptop will be configured as the creator (Master) and the Wi-Fi module connected to the ASV will be configured as the joiner.



Figure 55: Xbee S68

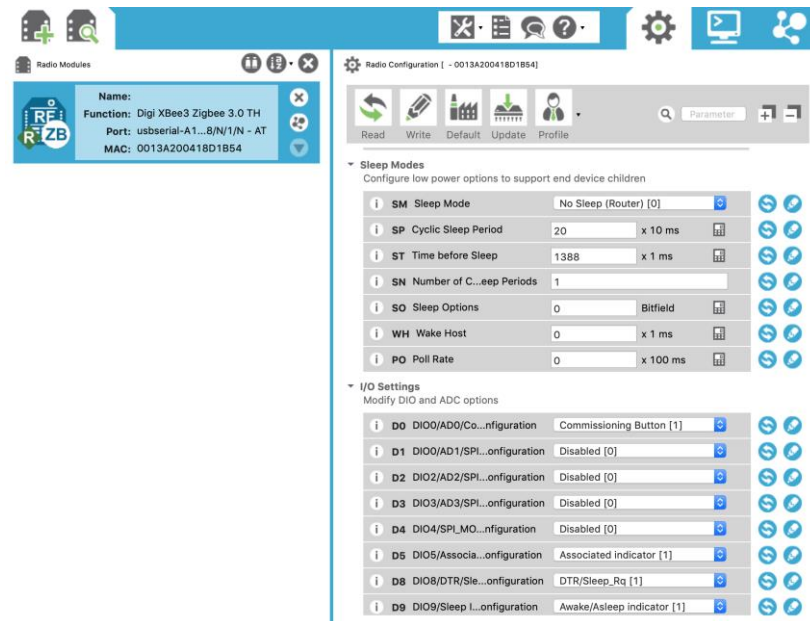


Figure 56: XCTU-Software

The configurations of the Wi-Fi modules are as shown in the following figures below:

▼ **Network**
Change Network Settings

i	AH Network type	IBSS Creator [1]	⚙️ 🔒
i	CE Infrastructure Mode	STA mode [2]	⚙️ 🔒
i	ID SSID	CapstoneASV	⚙️ 🔒
i	EE Encryption Enable	No security [0]	⚙️ 🔒
i	PK Passphrase		⚙️ 🔒
i	IP IP Protocol	UDP [0]	⚙️ 🔒
i	MA IP Addressing Mode	Static [1]	⚙️ 🔒
i	TM TCP Client Connection Timeout	64 * 100 ms	⚙️ 🔒
i	TS TCP Server Connection Timeout	258 * 100 ms	⚙️ 🔒
i	DO Device Options	3	⚙️ 🔒
i	EQ Device Cloud FQDN	devicecloud.digi.com	⚙️ 🔒

▼ **Addressing**
Change Addressing Settings

i	SH Serial Number High	40	⚙️ 🔒
i	SL Serial Number Low	9D66452F	⚙️ 🔒
i	NS DNS Address	208.67.222.222	⚙️ 🔒
i	DL Destination IP Address	192.168.1.11	⚙️ 🔒
i	NI Node Identifier		⚙️ 🔒
i	KP Device Cloud Description		⚙️ 🔒
i	KC Device Cloud Contact		⚙️ 🔒
i	KL Device Cloud Location		⚙️ 🔒
i	CO Source Port	2616	⚙️ 🔒
i	DE Destination Port	2616	⚙️ 🔒
i	GW IP Address of gateway	192.168.1.1	⚙️ 🔒
i	MK IP Address Mask	255.255.255.0	⚙️ 🔒
i	MY Module IP Address	192.168.1.10	⚙️ 🔒

▼ **Serial Interfacing**
Change module interfacing options

i	BD Baud Rate	9600 [3]	⚙️ 🔒
i	NB Parity	No Parity [0]	⚙️ 🔒
i	SB Stop Bits	One stop bit [0]	⚙️ 🔒
i	RO Packetization Timeout	3 * character times	⚙️ 🔒
i	FT Flow Control Threshold	7F3 Bytes	⚙️ 🔒
i	AP API Enable	Transparent Mode [0]	⚙️ 🔒
i	AO API Options	XBee 802.15.4 API Rx Indicator - 0x80 [2]	⚙️ 🔒

Figure 57: Configuration for the Creator (Laptop)

▼ **Network**
Change Network Settings

i	AH Network type	IBSS Joiner [0]	⚙️ 🔒
i	CE Infrastructure Mode	STA mode [2]	⚙️ 🔒
i	ID SSID	CapstoneASV	⚙️ 🔒
i	EE Encryption Enable	No security [0]	⚙️ 🔒
i	PK Passphrase		⚙️ 🔒
i	IP IP Protocol	UDP [0]	⚙️ 🔒
i	MA IP Addressing Mode	Static [1]	⚙️ 🔒
i	TM TCP Client Connection Timeout	64 * 100 ms	⚙️ 🔒
i	TS TCP Server Connection Timeout	258 * 100 ms	⚙️ 🔒
i	DO Device Options	3	⚙️ 🔒
i	EQ Device Cloud FQDN	devicecloud.digi.com	⚙️ 🔒

▼ **Addressing**
Change Addressing Settings

i	SH Serial Number High	4	⚙️
i	SL Serial Number Low	F313D15E	⚙️
i	NS DNS Address	208.67.222.222	⚙️ 🔒
i	DL Destination IP Address	192.168.1.10	⚙️ 🔒
i	NI Node Identifier		⚙️ 🔒
i	KP Device Cloud Description		⚙️ 🔒
i	KC Device Cloud Contact		⚙️ 🔒
i	KL Device Cloud Location		⚙️ 🔒
i	CO Source Port	2616	⚙️ 🔒
i	DE Destination Port	2616	⚙️ 🔒
i	GW IP Address of gateway	192.168.1.1	⚙️ 🔒
i	MK IP Address Mask	255.255.255.0	⚙️ 🔒
i	MY Module IP Address	192.168.1.11	⚙️ 🔒

▼ **Serial Interfacing**
Change module interfacing options

i	BD Baud Rate	9600 [3]	⚙️ 🔒
i	NB Parity	No Parity [0]	⚙️ 🔒
i	SB Stop Bits	One stop bit [0]	⚙️ 🔒
i	RO Packetization Timeout	3 * character times	⚙️ 🔒
i	FT Flow Control Threshold	7F3 Bytes	⚙️ 🔒
i	AP API Enable	Transparent Mode [0]	⚙️ 🔒
i	AO API Options	XBee 802.15.4 API Rx Indicator - 0x80 [2]	⚙️ 🔒

Figure 58: Configuration of Joiner (ASV)

Once the Wi-Fi modules are configured, we can test its functionality by using the XCTU software and TeraTerm, in order to transmit messages in between the Wi-Fi modules.

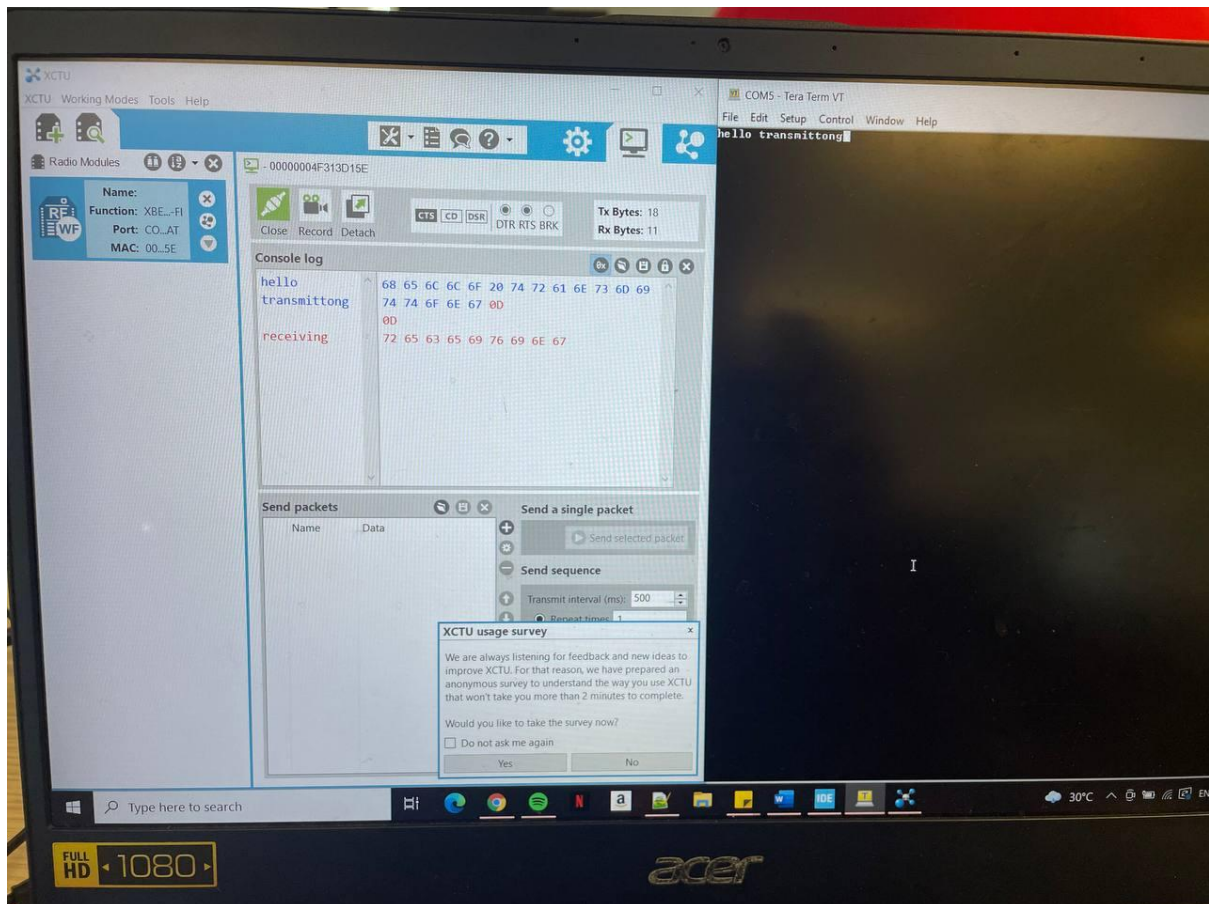


Figure 59: Testing the Wi-Fi modules by transmitting and receiving data

The figure above shows that we were able to transmit data from one Wi-Fi module and the other Wi-Fi module was able to receive the data. From this testing, we are able to ensure that the Wi-Fi modules have been configured correctly.

The next step will then be to use the microcontroller to be able to transmit and receive commands from the laptop.

7.2.2. Camera

The Intel Realsense D435 depth camera is a solution integrating vision processor, stereo depth module, RGB sensor with colour signal processing and Inertial Measurement Unit (IMU). The features of the camera are listed below.

Features of Intel RealSense D435

- Up to 1280 x 720 active stereo depth resolution
- Up to 1920 x 1080 RGB resolution
- Compact dimension of 90mm x 25mm x 25mm
- Depth diagonal field view over 90 degrees
- Dual global shutter sensors for up to 90 fps depth streaming
- Range between 0.2m - 10m

- IMU of 6 Degree of freedom data
- Accuracy of <2% at 2 m

The SDK package and wrapper files were downloaded into the Raspberry Pi 4. The detailed steps will be shown in the technical report. After the intel libraries and dependencies are successfully installed, the Intel RealSense Viewer application is launched and the following output can be seen as shown below. This application is used to view, calibrate and monitor the camera's output only. Ensure you are equipped with an USB3.0 type A to type C to connect the camera to Raspberry Pi.

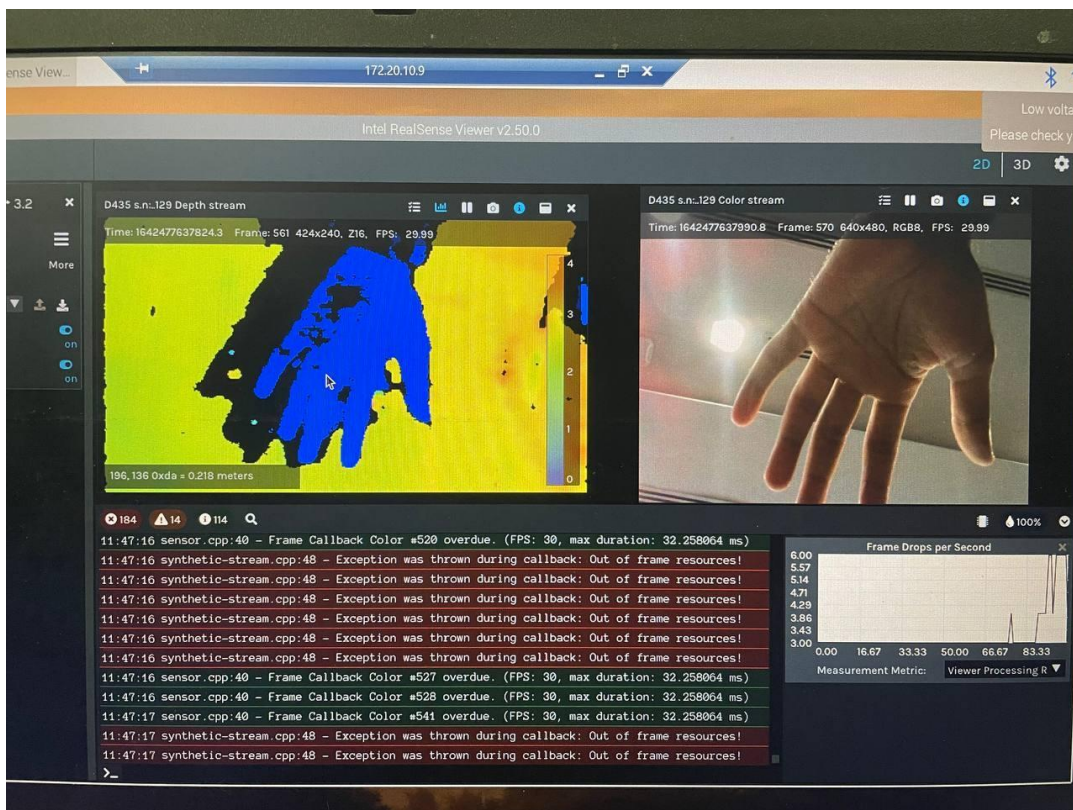


Figure 60: Testing Intel RealSense Viewer application in Raspi

A python script was written to output the detection and colour frame to achieve the obstacle detection objective of this project. After connecting the camera to the Raspberry Pi, the python script is executed where the violation distance can be set. There will be 2 frames which will pop up, 1 showing the visual of the D435 camera with the RGB camera with the distance between the object and the ASV, another which shows contours.area frame where the filtered image after eliminating background noises of the object being detected.

The snippet below shows the output of the 2 frames from executing the python script where the left frame shows the filtered image using the contour.area function in python, and the right frame shows the output of the RGB camera with depth frame. The detailed description of the code will be submitted in the technical document.

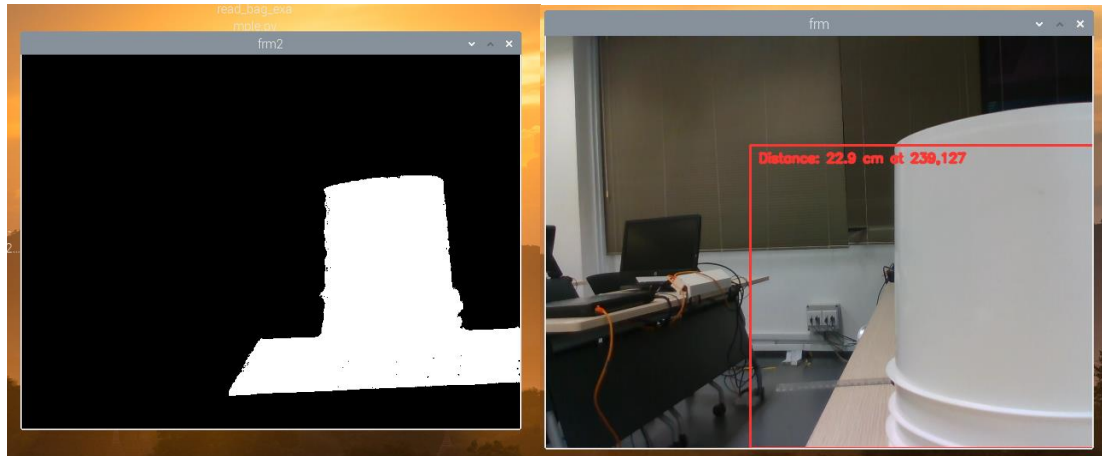


Figure 61: Testing of the python script's output

```

t1 = time.time()
try:#if True:
    t1 = time.time()
    frames = pipeline.wait_for_frames()
    depth_frame = frames.get_depth_frame()
    color_frame = frames.get_color_frame()
    depth_image = np.asanyarray(depth_frame.get_data())
    frame = np.asanyarray(color_frame.get_data())
    if depth_frame:
        mx = max(depth_image[0])+1
        nhist,bins = np.histogram(depth_image.ravel(),mx,[0,mx])
        rel = np.where(nhist[100:] > 500)[0]
        if len(rel):
            n = rel[0]
            m = n+200
            res_l = np.uint8(np.where((depth_image > n), 255, 0))
            res_u = np.uint8(np.where((depth_image < m), 255, 0))
            res = cv2.bitwise_and(res_u,res_l)
            #res = cv2.inRange(depth_image,[n],[m])
            _,contours, hierarchy = cv2.findContours(res,cv2.RETR_LIST,cv2.CHAIN_AP
            for cnt in contours:
                cv2.contourArea(cnt)

```

Figure 62: Snippet of the code

The following code above shows the functions where it obtains the histogram with its number of pixels that has a particular value.

After which, the data is filtered by removing the first 100 values as they output noise and maybe unwanted data.

The function gets values that are more than 500, in other words covering at least 500 pixels. res_1 stores the depth image where it is greater than the lower limit. res_u stores the image where it is smaller than the upper limit.

Using contours functions in opencv, we will be able to get connected regions in an image.

7.2.3. GPS Set up and Testing

As touched on previously in Section 4.1.5, the GPS module used is the X-NUCLEO-GNSS1A1.

The main function of the GPS is to locate the current position of the ASV via both the longitudinal and latitudinal coordinates. With this information we could accurately plot any desired location and translate this information into MCU to be processed.

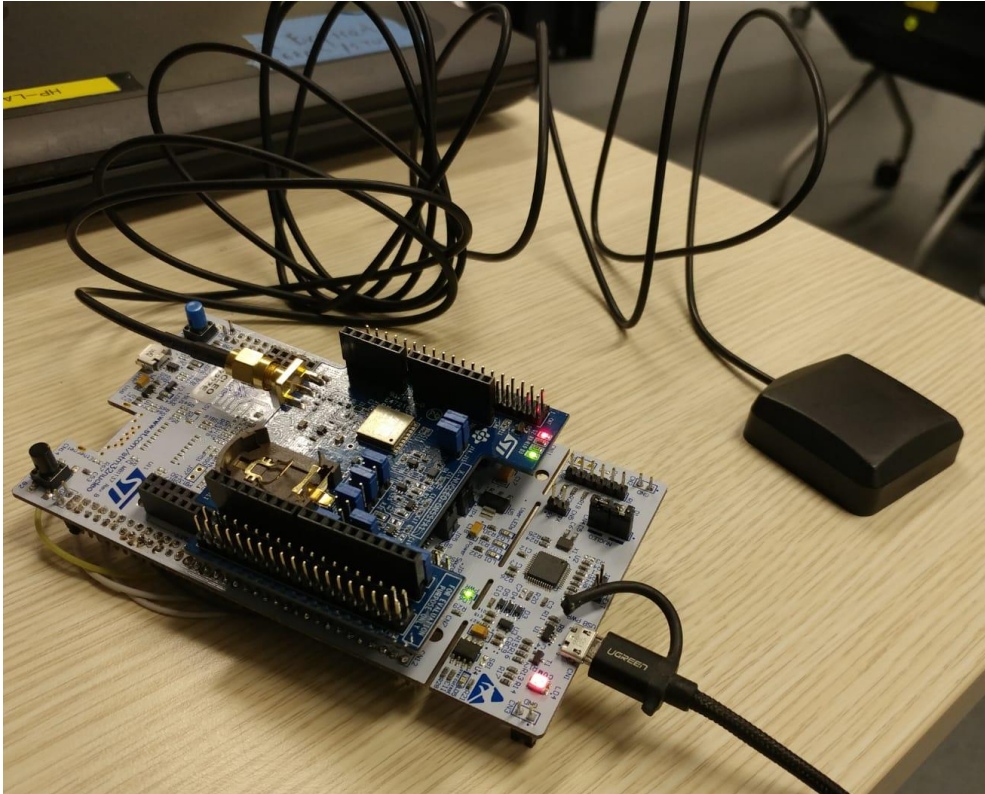


Figure 63: GPS Module connected to MCU

```

/* This function creates the task reading the messages coming from Teseo */
static void Teseo_Consumer_Task_Init(void)
{
    #if (osCMSIS < 0x20000U)
        osThreadDef(teseoConsumerTask, TeseoConsumerTask, osPriorityNormal, 0, configMINIMAL_STACK_SIZE);
        teseoConsumerTaskHandle = osThreadCreate(osThread(teseoConsumerTask), NULL);
    #else
        task_attributes.name = "teseoConsumerTask";
        teseoConsumerTaskHandle = osThreadNew(TeseoConsumerTask, NULL, (const osThreadAttr_t *)&task_attributes);
    #endif /* osCMSIS */
}

#if (USE_I2C == 1)
/* This function creates a background task for I2C FSM */
static void Background_Task_Init(void)
{
    #if (osCMSIS < 0x20000U)
        osThreadDef(backgroundTask, BackgroundTask, osPriorityNormal, 0, configMINIMAL_STACK_SIZE);
        backgroundTaskHandle = osThreadCreate(osThread(backgroundTask), NULL);
    #else
        task_attributes.name = "backgroundTask";
        backgroundTaskHandle = osThreadNew(BackgroundTask, NULL, (const osThreadAttr_t *)&task_attributes);
    #endif /* osCMSIS */
}
#endif /* USE_I2C */

/* This function creates the task reading input from the cocsole */
static void Console_Parse_Task_Init(void)
{
    #if (osCMSIS < 0x20000U)
        osThreadDef(consoleParseTask, ConsoleParseTask, osPriorityNormal, 0, CONSOLE_STACK_SIZE);
        consoleParseTaskHandle = osThreadCreate(osThread(consoleParseTask), NULL);
    #else
        task_attributes.name = "consoleParseTask";
        task_attributes.stack_size = CONSOLE_STACK_SIZE;
        consoleParseTaskHandle = osThreadNew(ConsoleParseTask, NULL, (const osThreadAttr_t *)&task_attributes);
    #endif /* osCMSIS */
}

```

Figure 64: Snippet of GPS code

The figures below show the testing of the GPS module at SIT@Dover.

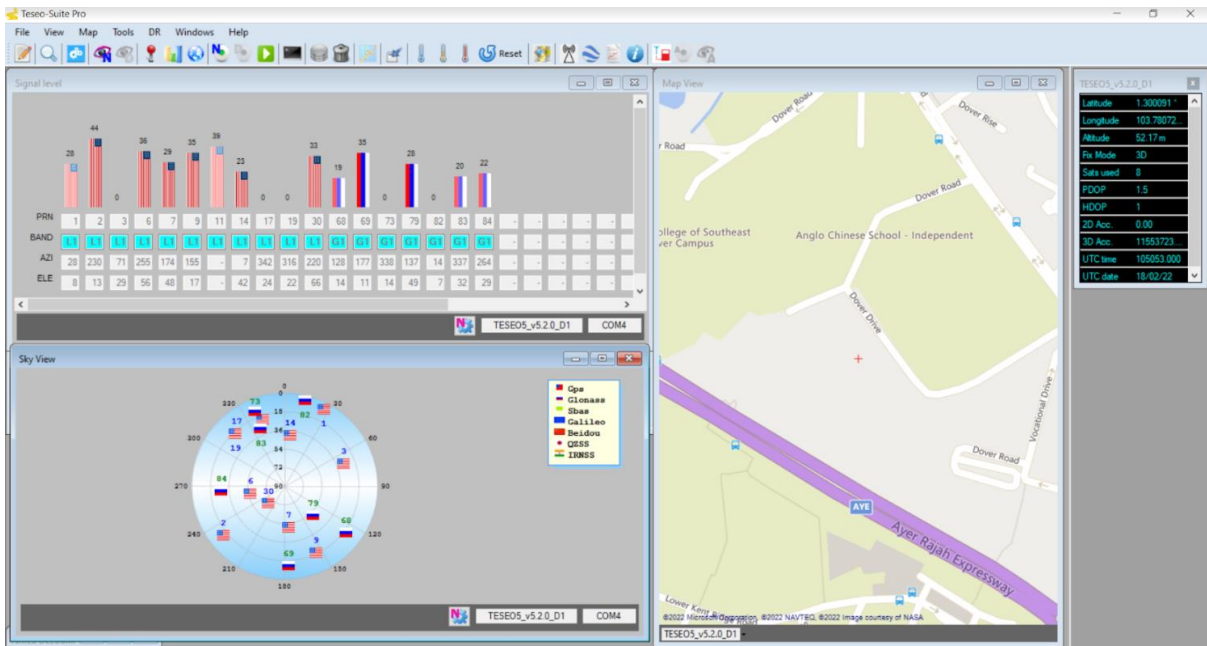


Figure 65: Results retrieved from GPS

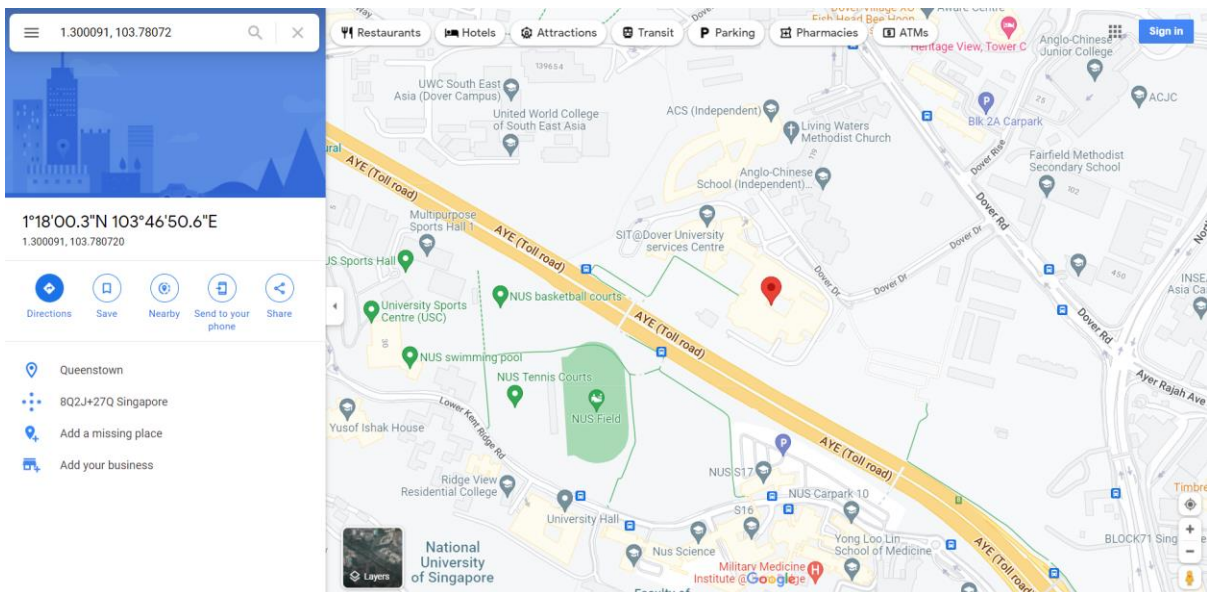


Figure 66: Coordinates from GPS in Google Maps

As can be seen in the figure above, the coordinates retrieved from the GPS were 1.300091, 103.78072. Entering the coordinates into Google Maps, it shows the location as SIT@Dover, which is correct.

The next step will then be to set up the data exchange between the GPS module and the terminal.

7.2.4. IMU

The IMU provides roll, pitch and yaw data, which is used to determine heading of the ASV with respect to the starting position.

```
interrupt receive: 0
*****break*****
interrupt receive: 0
MU [Gyro]: 348 -775 450 [Accell]: 143 316 16665 rpy: -0.81 -0.29 4.93 -155.74
*****break*****
interrupt receive: 0
MU [Gyro]: 448 -798 191 [Accell]: 145 385 16646 rpy: 5.98 -1.16 7.24 -156.83
*****break*****
interrupt receive: 0
*****break*****
interrupt receive: 0
*****break*****
interrupt receive: 0
MU [Gyro]: 428 -731 282 [Accell]: 160 348 16641 rpy: 5.62 -1.72 11.43 -157.02
*****break*****
interrupt receive: 0
*****break*****
interrupt receive: 0
MU [Gyro]: 438 -741 115 [Accell]: 132 382 16648 rpy: 5.63 -0.81 16.04 -157.28
*****break*****
interrupt receive: 0
MU [Gyro]: 489 -1213 -557 [Accell]: 377 379 16646 rpy: -1.14 -3.58 14.57 -157.62
*****break*****
interrupt receive: 0
*****break*****
```

Figure 67: Sample of Output from IMU displayed on terminal

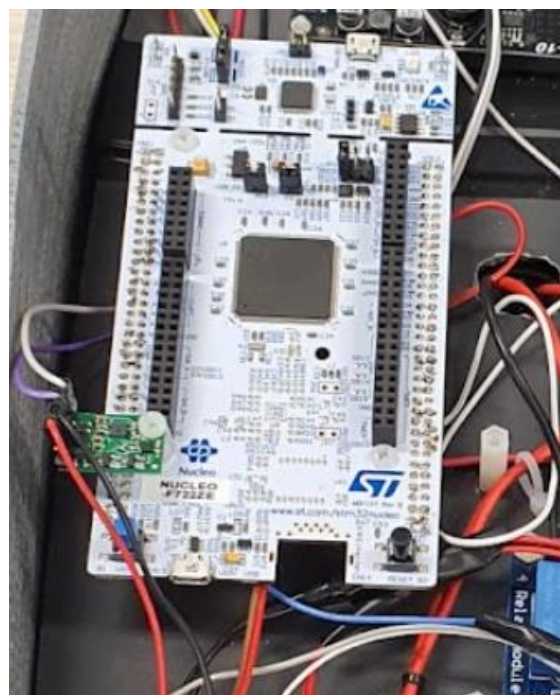


Figure 68: IMU attached to STM32 mounted on ASV

The readings from the IMU is displayed on the terminal, as can be seen in Figure 67.

7.2.5. Alternative GPS

After integration testing, it was found that the X-NUCLEO GNSS1A1 was incompatible with

the current setup as it ran on Real Time Operating System (RTOS) while the main program was not.

The team then evaluated all available options (presented in the table below).

	Option	Pros	Cons
1.	Using existing setup	<ul style="list-style-type: none"> No additional components required 	<ul style="list-style-type: none"> Requires re-writing of the main program to run on RTOS, which would take time
2.	Using the X-NUCLEO GNSS1A1 on another STM32 board	<ul style="list-style-type: none"> No additional coding required to setup the GPS No additional purchase required as the team had an extra STM32 board 	<ul style="list-style-type: none"> There would be an additional board on the ASV which would affect the component layout
3.	Buying another STM32 compatible GPS module to use on the current board	<ul style="list-style-type: none"> GPS module compatible with the current setup and able to run 	<ul style="list-style-type: none"> Requires purchase of another GPS module Requires writing of GPS drivers, which would take time
4.	Buying another GPS module and using Arduino	<ul style="list-style-type: none"> Able to utilize existing Arduino libraries for quicker coding No additional board purchase required as the team is able to loan Arduino boards from supervisor 	<ul style="list-style-type: none"> Requires purchase of another GPS module There would be an additional board on the ASV which would affect the component layout

Table 11: Evaluation of GPS Options

After consultations and taking into consideration the amount of time left, it was then decided go ahead with Option 4.

7.2.5.1. Alternative GPS: Adafruit Ultimate GPS Breakout Board

After researching online, the GPS chosen was the Adafruit Ultimate GPS Breakout Board with Arduino Uno.

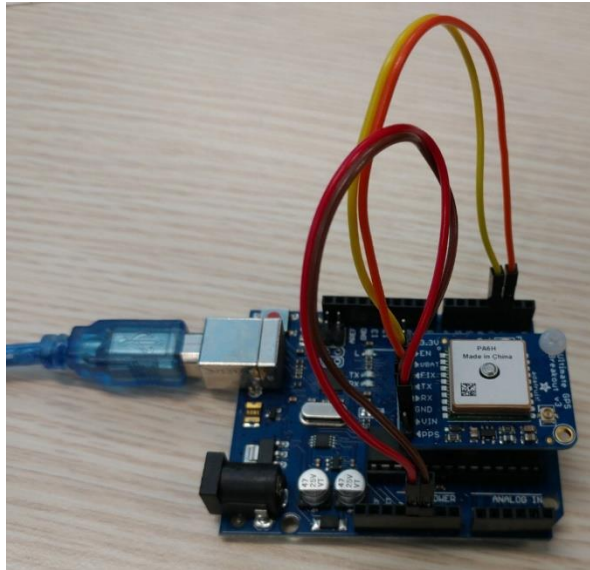


Figure 69: Arduino Uno and Adafruit Ultimate GPS Breakout Board

```
Location: 118.5636N, 10346.5878E
Google Maps location: 1.3094, 103.7765
Speed (knots): 0.29
Heading: 230.17
-----
Time in Singapore: 16:14:35.0
Date: 1/4/2022
Fix: 1 Quality: 2
Satellites: 5
Location: 118.5634N, 10346.5878E
Google Maps location: 1.3094, 103.7765
Speed (knots): 0.44
Heading: 186.91
-----
Time in Singapore: 16:14:37.0
Date: 1/4/2022
Fix: 1 Quality: 2
Satellites: 5
Location: 118.5632N, 10346.5878E
Google Maps location: 1.3094, 103.7765
Speed (knots): 0.35
Heading: 206.36
-----
□
```

Figure 70: Sample of output from the GPS

The GPS is coded to retrieve six data readings: Latitude, Longitude, Date, Time, Speed, and Heading.

8. ASV Operation

The ASV is capable of 3 levels of Operation as defined in the table below:

Mode of Operation	Description
Level 1	<ul style="list-style-type: none"> Control of ASV via Remote Control (RC) Redundancy measure
Level 2	<ul style="list-style-type: none"> Control of ASV via Terminal Control Use of C# program on laptop to control ASV
Level 3	<ul style="list-style-type: none"> Autonomous operation of the ASV via Waypoints

Table 12: ASV Modes of Operation

8.1. Level 1 Operation

In Level 1, the control of ASV is via Remote Control. This is also a redundancy measure to allow control of ASV in the case of emergencies.

8.2. Level 2 Operation

In Level 2, the control of the ASV is via a User Window on the terminal (Laptop) with the commands being transmitted via Wi-Fi. The terminal control consists of buttons that control the directions and one button that toggles between Remote control and Terminal Control.

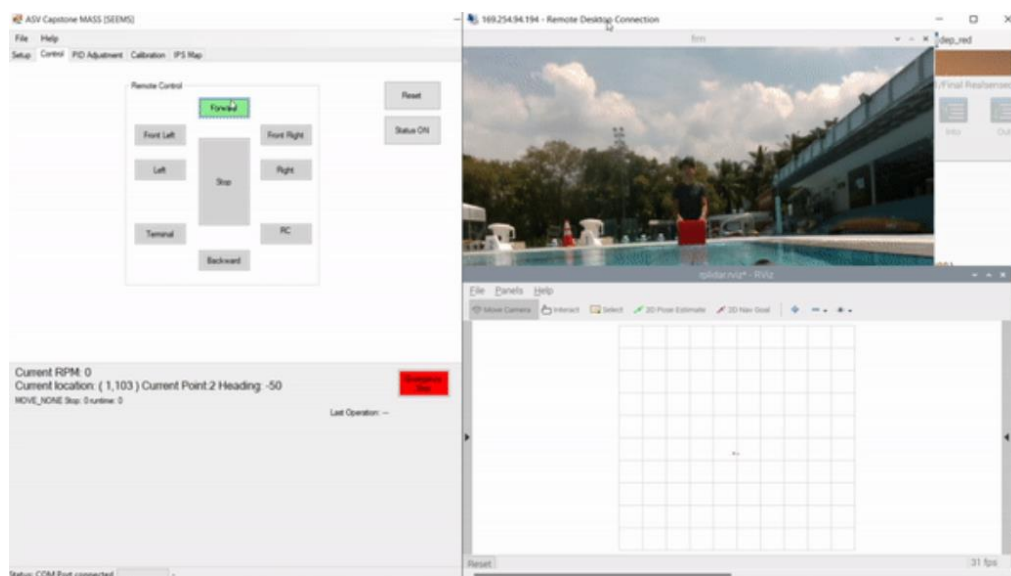


Figure 71: Screenshot of Terminal Control view on laptop

The terminal control user window also displays the current latitude, longitude and heading of the ASV. When the ASV is turned on, it will await commands from terminal control.

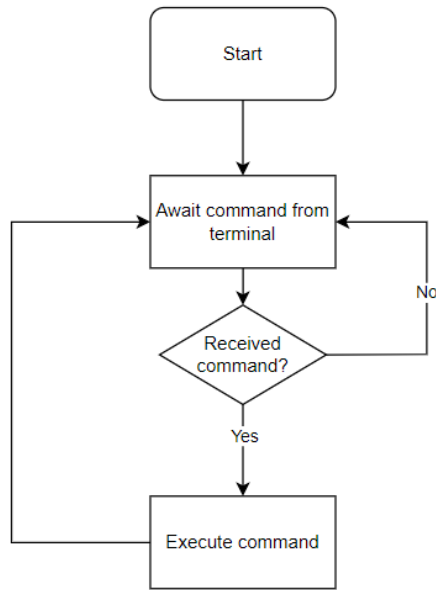


Figure 72: Flowchart illustrating logic for Level 2 Operation

8.3. Level 3 Operation

In Level 3, once a waypoint is provided on the terminal, the ASV moves autonomously towards the waypoint entered. The bearing between the waypoint and the current location of the ASV is calculated (bearing offset). The bearing offset is then compared against the heading of the ASV provided by the IMU.

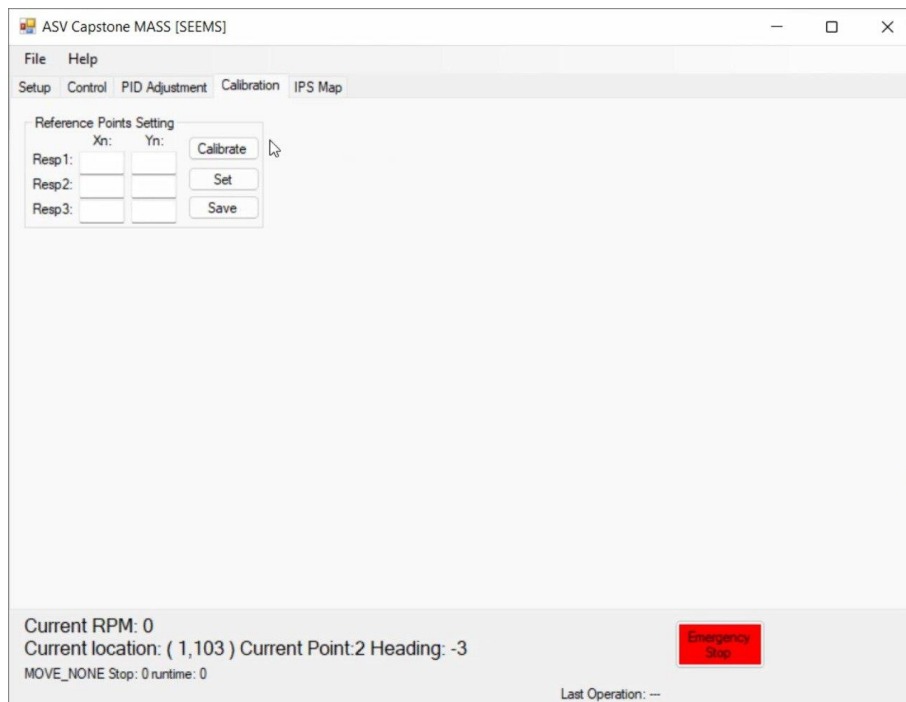


Figure 73: Window to input Waypoint

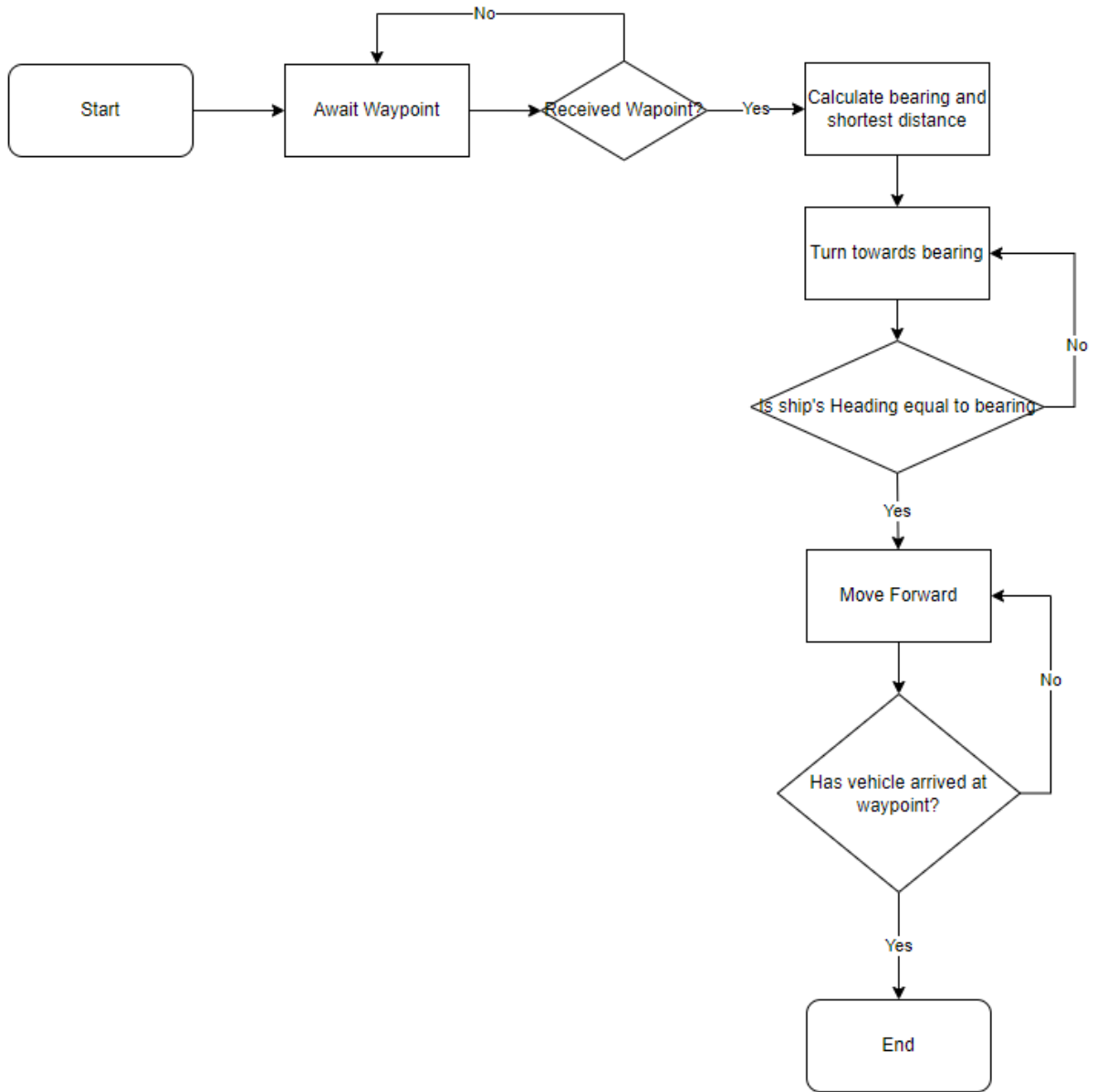


Figure 74: Flowing illustrating logic for Level 3 Operation

9. Trimester 2 – Testing

Testing in Trimester 2 was mainly done at the swimming pool in SP.

The testing scenarios defined in Section 2.6.4 were carried out and checked against the testing criteria defined in Section 2.6.3.



Figure 75: Testing at SP Pool

10. Overcoming Challenges

No project is without its challenges, and naturally we encountered quite a few challenges during the course of the trimester.

10.1. Hull Design

The initial stages of the execution phase were challenging, as we were required to design a suitable hull capable of carrying all of our electronic components. Although the design of the hull isn't the main focus of the ASV project, it was important to have a thorough and fool proof design as the hull will serve as the foundation of the system, encapsulating all of our components and ensuring that the movements of the ASV can be carried out correctly in the water.

As the team did not have any maritime background, the team struggled with the overall design and dimensions, and had difficulties in understanding the major requirements of ensuring that the hull remains afloat. The team faced difficulties in the initial calculations, and struggled to find a suitable shape and design for our ASV.

Fortunately, through the help of Professional Officers, we were able to carry out the necessary tasks. By consulting with the professional officers who were experienced in the maritime industry, we were able to determine the required calculations and were successful in coming up with a suitable design for our ASV.

10.2. Running ROS

Another challenge that we overcame was familiarizing ourselves with ROS. As mentioned in the earlier sections, ROS was used to drive and visualise the data from the Lidar. As the team were new to the ROS and Raspberry pi, we initially hit a major obstacle when downloading the required ROS packages.

The Packages were incorrectly downloaded during setup and there were vital files that were missing. Certain errors that were displayed during installation were foreign as it was the first time the team faced such errors. As such, debugging and finding the error in the download was difficult and troublesome.

However, through research and online communities, several steps to rectify the incorrect download were found, and the missing packages were able to be downloaded, enabling the team to run ROS correctly on our Raspberry Pi.

One other challenge faced when running ROS was displaying the RVIZ software through the headless VNC server. As we were running the raspberry pi headless, we required an external viewer server to display the screen of the raspberry pi. The challenge that we faced was that the RVIZ could not be displayed through the headless server due to resolution issues. This was overcome by switching from VNC server to using remote desktop. By running a remote desktop, we were able to achieve better resolution and were able to display the raspberry pi screen and the RVIZ GUI.

10.3. Camera Live Feed

The objective of the camera live feed was to provide an on-board visual of the ASV allowing the operator to observe any incident at bow direction of the ASV. Through researching the RPI forum we were able to adopt one of the many solutions found which fulfils the criteria, In order to operate the live feed camera, scripts and software setup was needed in the aspects of the python scripting which allows the feed to be streamed over a wireless medium.

During the trial test we faced some latency issues, investigation was done, and we deem that it was a congestion in the network and to mitigate these issues we avoid connecting to the network when it is not necessary during any further testing.

10.4. Multiple Commitments

Throughout our capstone curriculum, It was a challenge for us to balance school workload and our capstone deliverables due to other module's commitments. We had multiple submissions, assignments and quizzes which were all scheduled in various weeks which made it challenging for us.

We improved our time management by using an excel sheet to document the progress and the need for action items (Impediment list). By doing so we learnt on how to manage our time and focus on our priorities.

11. Lessons Learnt

11.1. Software Competencies

One of the main things learned this trimester was learning how to operate the Debian OS. As this is the first time the team is working with raspberry pi, we were required to familiarise ourselves with the Debian OS as that was used to run the raspberry Pi. Through this capstone, we were able to learn a new operating system, and familiarise ourselves with the required language needed in order to run the programs on the raspberry pi.

Another learning point was learning how to code in python. As the common language for running programs on the raspberry pi is python, we were required to pick up and learn the python language. Python scripts were also used in order to enable our camera livestream. As we were required to program our camera in python, it helped the team developed a competency in a new programming language.

11.2. Robot Operating System

Another major learning point is learning the functionalities and steps required to develop a robotic operating system. We were needed to learn how to work with specific hardware such as camera and Lidar. These hardware components are vital in building a robot operating system and require special integration and drivers in order to ensure that the full Robotic operating system is functioning.

In addition to this, we also learned how to use existing libraries, such as ROS, in order to help build our ASV. Libraries provided by ROS were readily available and were vital in ensuring the functionality of our camera and Lidar sensors. Through this ASV project, we were able to develop competencies in developing a robotic operating system, and learn how to integrate vital hardware components to enable simultaneous localisation and mapping.

11.3. Naval Knowledge

With little to no background knowledge on naval architecture and design, we had to research and familiarise ourselves with, properties of hull design, designing rudder mechanism, rules & regulation that govern the movement of vessels and the degree of automation to achieve the goal of this project.

12. Future Works

As this Capstone Project is a multi-year project, some potential improvements to the ASV can be done such as:

12.1. Sensor Integration

The data retrieved from the Camera and Lidar can be fused to provide a more holistic approach to obstacle detection. An example would be the data from the Lidar showing that an obstacle has been detected and confirming the presence of that obstacle using the data from the camera, thereby lowering the chances of a false detection.

12.2. Development of Obstacle Avoidance Algorithm

In order to meet the objectives of autonomous operations, the ASV requires an Obstacle Avoidance algorithm to guide its movements. This algorithm is a major objective of the project, as this will enable the ASV to move autonomously, and carry out obstacle avoidance movements without requiring any human inputs. The algorithm should also be coded to navigate via the maritime rules to better

12.3. Usage of data retrieved from GPS and RPI

Further work can be done to incorporate the data from the GPS and RPI into the main program, so as to improve the navigation capabilities of the ASV. The data from the GPS will help the ASV navigate to any given waypoint while the data from the RPI will ensure that obstacles in the way are avoided.

13. Conclusion

The Capstone project has provided us with the invaluable opportunity to work on industry related projects. Such projects allow us as students to contribute towards solving ongoing problems in the working world. Being able to work collaboratively with ST Engineering was a privilege as working with such an esteemed company has enabled us to have a better understanding and appreciation of how real-world problems are being solved.

The ASV project has been a welcomed opportunity as it has enabled us to join in the efforts to better the maritime industry through the development of an ASV. Being the first batch of students to take on such a project in SIT was daunting, but it posed a challenge that the group was willing to take on.

The ASV project has already posed a lot of challenges, which required application of past knowledge taught in school to overcome them. Project management knowledge was required in order to plan and create the project schedule, to ensure that the project scope did not creep. Requirements engineering knowledge was also required to fully understand the aspects of this project, and to ensure that the project meets the needs of SIT, ST Engineering, and the marine industry.









Working on such a big industry project was challenging, however, seeing how much the project has progressed since the start of the Capstone has made it worthwhile.

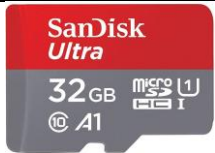






We have learnt a lot and hope that the work done on this project will help lay a solid foundation towards a development of a proof of concept for an ASV by SIT.

References

- [1] Z. Abdullah, "Singapore navigating shipping squeeze, container congestion amid surge in cargo demand," Channel News Asia, 26 March 2021. [Online]. Available: <https://www.channelnewsasia.com/news/singapore/shipping-congestion-delays-singapore-port-cargo-demand-14480112>. [Accessed July 2021].
- [2] K. Yufeng, "Port of Singapore regains momentum despite disruptions caused by Covid-19 pandemic," The Straits Times, 18 June 2021. [Online]. Available: <https://www.straitstimes.com/singapore/singapore-port-handled-more-cargo-this-year-despite-disruptions-caused-by-covid-19>. [Accessed July 2021].
- [3] "IMO MSC Identifies 4 Degrees of Ship Automation," Offshore Energy, December 2018. [Online]. Available: <https://www.offshore-energy.biz/imo-msc-identifies-4-degrees-of-ship-automation/>. [Accessed July 2021].
- [4] Adaptive, "System Classification and Glossary," Adaptive, 2014. [Online]. Available: <https://www.adaptive-ip.eu/index.php/Adaptive-SP2-v12-DL-D2.1-System%20Classification-file=files-adaptive-content-downloads-Deliverables%20&%20papers-Adaptive-SP2-v12-DL-D2.1-System%20Classification.pdf>. [Accessed July 2021].
- [5] International Maritime Organization, "IMO takes first steps to address autonomous ships," International Maritime Organization, 2018. [Online]. Available: <https://www.imo.org/en/MediaCentre/PressBriefings/Pages/08-MSC-99-MASS-scoping.aspx>. [Accessed July 2021].
- [6] Nippon Express, "Maritime Autonomous Surface Ships," Nippon Express, December 2019. [Online]. Available: <https://www.nipponexpress.com/press/report/10-Dec-19.html>. [Accessed July 2021].
- [7] R. Gordon, "Autonomous boats could be your next ride," Massachusetts Institute of Technology, 26 October 2020. [Online]. Available: <https://news.mit.edu/2020/autonomous-boats-could-be-your-next-ride-1026>. [Accessed July 2021].

Appendix A

Item Number	Item Name	Manufacturer/ Manufacturer Part No.	Quantity	Image	Amount
1	PiCam 8 MP	RPI 8MP CAMERA BOARD	1		41.01
2	Raspberry Pi 4	RPI4-MODBP-4GB	1		82.25
3	RPLIDAR A1M8-R6	DFR0315	1		170.00
4	Pololu MinIMU-9 v5	2738	1		33.46
5	XBee Wi-Fi (S6B)	XB2B-WFWT-001	1		55.87
6	X-NUCLEO-GNSS1A1	3A991A2	1		34.30
7	Arduino UNO	A000066	1		23.00
8	NUCLEO-H723ZG	STM32H723ZG	1		28.42

9	Sandisk Ultra MicroSD 32GB	Sandisk Ultra MicroSD 32GB	1		11.00
10	Mabuchi RS-540 Torque Tuned Motor	54358	1		28.00
11	LRP HYPER PACK 5000 NIMH STICK 7.2V	LRP71145U	1		83.54
12	Analog Micro Servo 9g (3V-6V)	TS90A	1		4.29
13	Sparkfun XBee Explorer USB	WRL-11697	1		16.21
14	Intel Real-sense D435 Depth camera	D435	1		299
15	Adafruit Ultimate GPS Breakout Board	746	1		29.95