

PROJ324 - Project O.R.C.A. - On-water Rubbish Collection robot with Automatic sensing

Luke Waller

Sunday 29th May, 2022

Abstract

Oceans make up over 70% of the world's surface. They are vital to all life on the planet as a large majority of the oxygen on the planet comes from phytoplankton that live near the surface of the water. It is estimated that over 10 million tonnes of litter end up in the ocean each year. It is estimated that by 2050 the amount of plastic in the ocean will outnumber the fish, with about 15% floating on the surface (1.5 million tonnes). This on-surface that is being targeted by the On-water Rubbish Collection robot with Automated sensing (ORCA).

Several solutions to this problem have been created; however, the ORCA based solution aims at a novel application that has previously been left unexplored. ORCA aims to collect rubbish from within in-land bodies of water, plastic chokeholds, using a conveyor-based system to remove the rubbish from the water.

Keywords

Waste Collection, Microcontroller, Conveyor belt, C++

Acknowledgements

The author would like to thank the mentor support provided by Dr Ian Howard, who agreed to undertake the role of mentor during the start of the Autumn Term along with the financial support provided by the University of Plymouth.

The author would also like to thank his family members for their continued support, and his friends for the continued inspiration to push me more and more every year.

Contents

1	Introduction	10
2	Design Requirements	10
3	Aims and Objectives	11
4	Project Management	12
4.1	Phase 1 - Autumn Term 2021/22	13
4.2	Phase 2 - Winter Term 2021/22	13
4.3	Phase 3 - Spring/Summer Term 2022	14
5	Research	15
5.1	Concept Designs - Mechanical Assembly	15
5.1.1	Concept 1 - Dragged Net Based System	15
5.1.2	Concept 2 - Submerged Basket System	16
5.1.3	Concept 3 - Conveyor System with Elevated Basket	16
5.2	Concept Designs - Controller and Receiver	16
5.2.1	Concept 1 - Off the Shelf Solution	16
5.2.2	Concept 2 - Custom RF Based Solution	17
6	Optioneering	18
6.1	Controller	18
6.2	Receiver	19
6.3	Peripherals	20
6.3.1	RF Transciever	20
6.3.2	Brushless DC Motors	21
6.3.3	Conveyor Motors	22
6.3.4	Motor Drivers	23

6.3.5	Batteries	24
6.3.6	IR Distance Sensors	26
6.3.7	Voltage Regulators	26
6.4	Finalised Solution	27
6.4.1	Block Diagram	27
6.4.2	Bill of Materials	30
7	Software Design	31
7.1	Controller	31
7.1.1	Architecture	31
7.1.2	Serial Communications	31
7.1.3	Buzzer API	33
7.1.4	Main Code	34
7.2	Receiver	35
7.2.1	Architecture	35
7.2.2	Serial Communications	36
7.2.3	Buzzer API	36
7.2.4	L298N API	37
7.2.5	ESC API	38
7.2.6	Main Code	39
8	Hardware Design	40
8.1	Prototyping	40
8.1.1	Small Scale Design	40
8.1.2	1:1 Scale Prototype	42
8.1.3	Conveyor Research	44
8.2	CAD Design	45
8.2.1	Main Craft	46

8.2.2	Conveyor Belt	47
8.3	PCB Design	48
8.3.1	Controller PCB	49
8.3.2	Receiver PCB	50
9	Testing	52
9.1	PCB Testing	52
9.2	Software Testing	53
9.3	Hardware Testing	53
10	Assembly	54
10.1	Floats	54
10.2	Superstructure	57
10.3	Conveyor Belt	59
10.4	Belt Drive Motors	61
11	Evaluation	62
12	Further Developments	64
13	Sustainability, Ethics, and Commercial Viability	65
13.1	Commercial Risk Evaluation	65
13.2	Legal Requirements	66
13.3	Intellectual Property	66
13.4	Environmental Impact	66
13.5	Ethical Conduct	67
13.6	Social Context	67
13.7	Product Safety and Liability	67
13.8	Internationalisation	68

14 Conclusion	68
A Design Requirements Table	72
B Software Design Requirements Table	72
C Full Sized Gantt Chart	73
D Gantt Chart Phase 1	74
E Gantt Chart Phase 2	74
F Gantt Chart Phase 3	75
G Transmission Codes	75
H Bill of Materials	77
I Controller PCB Schematic	78
J Receiver PCB Schematic	79
K Back Panel Technical Drawing 1:5 scale	80

List of Figures

1	Concept 1 - Off the shelf RC Controller and Receiver	17
2	Concept 2 - Custom RC Controller and Receiver	18
3	nRF24L01 Discreet Package	20
4	A2212 1400KV Brushless DC Motor	21
5	PLG24 Brushed DC Motor	22
6	Generic no-name 30A ESC	23
7	L298N Driver Board for DC and Stepper Motors	24
8	18650 Li-Ion Batteries	25
9	14.4V 5200mAh Robotic Vaccuum Cleaner Battery Pack	25
10	Generic no-name IR Distance Sensor	26
11	Final System Block Diagram of ORCA	29
12	Milkshake Catamaran Prototype	41
13	Milkshake Catamaran with Radio Controlled (RC) electronics	41
14	1:1 Scale Cardboard Prototype Hull	42
15	1:1 Scale Cardboard Prototype Hull in Water with Additional Load	43
16	Computer Aided Design (CAD) for the Main Craft	47
17	CAD for the Conveyor Belt	48
18	Breadboard for hardware testing for controller PCB	49
19	Top and Bottom sides of Controller PCB design	50
20	Breadboard for hardware testing for receiver PCB	50
21	Top and Bottom sides of Receiver PCB design	51
22	3D Printed 2 part Float Mould	55
23	Foam Float Assembly	55
24	Foam Floats attached to Skis	56
25	Fibreglass Added to the Hull Section	56
26	CAD Model of Support Struts	57

27	RivNut Inserted Into Support Strut	58
28	Fish Tanks Fitted to the Craft	58
29	Removable back panel to access electronics	59
30	3D Printed Hinges and Low Profile Brackets on Conveyor Belts	60
31	Full Conveyor Subsystem Mounted to Craft	60
32	Full Conveyor Subsystem Mounted to Craft	61
33	Full Conveyor Subsystem Mounted to Craft	62

List of Tables

1	Component Selection	27
2	Threads for Controller	31
3	Threads for Receiver	35
4	Functionality of Buttons and Switches in the Main Receiver Code	40
5	Material Considerations for Main Craft hull	44
6	Material Considerations for Conveyor Belt	45

Code

1	nRF24L01 Class Constructor	32
2	nRF24L01 Set Up Transmit Mode	32
3	Sending Data Using nRF24L01	32
4	Buzzer Class Constructor	33
5	Joystick Value to Serial Data to be Sent	34
6	nRF24L01 Set Up Receive Mode	36
7	L298N Class Constructor	37
8	L298N Class Methods	37

9	ESC Class Constructor	38
10	ESC Class Public Methods	38
11	ESC Class Private Methods	38

Glossary

Catamaran A catamaran is a multi-hulled watercraft featuring two parallel hulls of equal size. 41

Dibond An aluminium composite sheet, comprising two 0.3 mm thick aluminium cover layers and a polyethylene or mineral core. 44, 46, 47, 54

ELEC351 Advanced Embedded Programming Final Year Module. 13

Fusion 360 A cloud-based 3D modelling, CAD, CAM, CAE and PCB software platform for product design and manufacturing. 46

KiCad A free software suite for electronic design automation. It facilitates the design and simulation of electronic hardware. It features an integrated environment for schematic capture, PCB layout, manufacturing file viewing, SPICE simulation, and engineering calculation. 48

LiDAR A detection system which works on the principle of radar, but uses light from a laser. 64

Mbed A platform and operating system for internet-connected devices based on 32-bit ARM Cortex-M microcontrollers. 14, 18, 68

Oxo-biodegradable OXO-degradation is the degradation of polymers such as plastic by an oxidative process. Such degradation breaks down plastic into small pieces that

are easier to biodegrade (but not necessarily enough to be called "biodegradable").

43

Plastazote A nitrogen blown, closed cell polyethylene foam which is tough and flexible.

The cells have regular shapes, and polyethylene walls. If the foam is compressed, each nitrogen filled cell acts as a balloon, causing the foam to spring back to its original shape. 45, 59, 63

RivNut A rivet nut, also known as a blind rivet nut, or rivnut, is a one-piece internally threaded and counterbored tubular rivet that can be anchored entirely from one side.

6, 47, 56–58

Acronyms

API Application Programming Interface. 14, 33, 36–38

ARM Advanced RISC Machines Ltd. 13, 18, 19

BLDCM Brushless DC Motor. 13, 68

CAD Computer Aided Design. 5, 42, 46–48, 57, 61, 68

DC Direct Current. 13, 37, 38

ESC Electronic Speed Controller. 23, 38, 39

GPS Global Positioning System. 65

IDE Integrated Development Environment. 18, 19, 68

IP Intellectual Property. 66

IR Infra-Red. 39, 40

ISR Interrupt Service Routine. 34, 39

LED Light Emitting Diode. 52

LLDPE Linear Low-Density Polyethylene. 43

ORCA On-water Rubbish Collection robot with Automated sensing. i, 46, 62, 65–69

PCB Printed Circuit Board. 13, 14, 48–50, 52, 68

PLA Polylactic Acid. 40, 56, 59

PWM Pulse Width Modulation. 33, 38

RC Radio Controlled. 5, 16, 40, 41

RF Radio Frequency. 14, 17, 19, 30–32, 34–36, 53, 63, 68

RTOS Real Time Operating System. 13

SMT Surface Mount Technology. 49, 51

SPI Serial Parallel Interface. 20, 31, 32, 34, 36

1 Introduction

Current technologies for litter collection within in-land bodies of water is limited with few solutions presently available. Litter collection is becoming an increasingly prevalent issue and if not addressed will continue to worsen; the litter damages marine life and the surrounding environment. The amount of litter ending up in in-land bodies of water is increasing at an exponential rate. Capturing plastics inland before they can make their way into the oceans is suggested to be one of the best ways to help reduce the amount of litter pollution in the seas.

Much of the litter ending up in the seas is toxic to marine life, and that which isn't is damaging to marine health. It has been recently discovered that micro-plastics have been found in up to 80% of peoples blood [1]. Although the effects of this are currently only speculative, it is suggested that there may be an affect on immune regulation or predisposition to diseases with an immunological base [2]. This is caused by marine life coming into contact with and ingesting microplastics that are in the oceans. The best way to reduce microplastics ending up in humans is to remove the litter from the water before it has a chance to end up in the food chain.

Project ORCA aims to aid in the solution of this problem by attacking this problem at the source, by collecting litter from in land bodies of water before it has a chance to end up in the ocean. ORCA provides a novel and affordable solution to the collection of litter by utilising a conveyor based collection mechanism. The overall system will be utilised in lakes and reservoirs that lead into the sea with the aim to collect the accumulated litter before it ends up contributing to the ever increasing amounts of litter in the ocean.

2 Design Requirements

The design requirements for this project as outlined in Appendix A and Appendix B were agreed upon by the author and their project supervisor. The table acts as a checklist

for the project to ensure all the requirements were met. This was a dynamic list that was adapted and amended as the project progressed. The core requirements of the project remained the same while the higher level attainable aims were adapted as the project progressed. This was as a result of the critical analysis stages of the project where the feasibility of certain aims was evaluated and were adapted accordingly.

3 Aims and Objectives

The aims and objectives, which have been listed below, are used as an indicator of the project's success. These were created using the design requirements of the project. The overall success of this project was measured against these aims and objectives:

1. Design and create a boat that is capable of floating on water and being driven around by a radio controller and receiver. The craft will need to be an appropriate size such that it can carry all the equipment on board as well as a decent payload of rubbish to be collected. A suitable propulsion system, steering system, and make sure the craft is stable both at top speed and when cornering will need to be created. It is important to make sure the craft is highly controllable and safe to operate.
2. Design and create a suitable litter collection system that can collect litter on the surface of water. The system will need to be able to be turned off, not be damaging to wildlife, and be able to collect all types of litter that build up on the surface.
3. Combine the litter collection system with the main hull of the boat. Check that the functionality of both components is not impeded by the combination of the two parts together. Get the two parts of the boat controlled separately using a radio controller as the currently available radio controller is only 3 channels.
4. Implement a custom microcontroller based remote control and receiver for the boat that would allow broader uses for the controller. The controller would be able to display some rudimentary values such as, battery charge on the boat, the current capacity of

the rubbish collection system, and the status of the litter collection system (operating/not operating).

5. Add sensors, both active and passive to test for current capacity of the litter collection basket as well as current battery charge. The status of these sensors would be outputted to the controller that would alert the user, alongside stopping the boat when the power is low or when the litter collection basket is full.

4 Project Management

The project had a development lifecycle of approximately 36 weeks and began on the 6th September 2021.

The project was split into three phases of development. Phase one was research and development of the project and ideas. Phase two was developing prototypes and testing. The final phase was finalising the design and build.

The project was managed using the following software:

- LaTeX (TeXMaker), to write the documentation for the project.
- GitHub, for version control across the whole project.
- Microsoft Excel, to produce Gantt chart and to produce graphs.
- Microsoft OneNote, to log progress and documentation of research and experimentation.

The management style for this project followed a similar style used by Roke Manor Research Ltd.'s agile framework. This framework follows an agile approach to project development [3], which is evident in the adaptability of the project as it has developed.

The utilised management style was extremely effective and when utilised with software packages above, all tasks were executed in an organised manner. For general awareness

of time management and deadlines, a Gantt chart was utilised. This worked well for this type of project, a solo project undertaken with the supervision of one party, but would not have scaled well to a project with a larger group and organisational supervision. Due to this utilisation method, setbacks were minimised and able to be accounted for within the initial plan. Any unplanned setbacks were easy to coordinate within such a small working group.

The milestones laid out in the project Gantt Chart for phases 1, 2, and 3 are listed below in Appendix C, with further breakdowns in Appendix D, Appendix E, and Appendix F respectively.

4.1 Phase 1 - Autumn Term 2021/22

Phase 1 of the project was ensuring that the planning, research, and development was conducted in line with the agile methodology. The objectives from Phase 1 are outlined in Appendix D. At this stage, the design requirements and optioneering were considered.

From the initial planning phase, a design brief, Gantt Chart, and future steps were created and agreed upon with the project supervisor. A clear set of requirements had been established along with identification of rudimentary concept ideas. The resultant concepts and decision making processes (optioneering) are detailed in §5 and §6.

4.2 Phase 2 - Winter Term 2021/22

During Phase 2, initial designs were created, tested, and evaluated. Phase 2 involved undertaking initial prototyping and light development of the work. This was to further the work conducted at the end of Phase 1 and to supplement the work being conducted within Phase 3. Phase 2 ran alongside three full-time university modules. It was used as a basis for the prototyping for the project.

The second phase helped focus the design requirements for the project, whilst eval-

uating what would be possible within the given time scaled budget. The use of threads, learned from ELEC351, along with the development of classes, drove the development of my project from an Arduino based solution to an Advanced RISC Machines Ltd (ARM) based solution. This enabled the use of the Real Time Operating System (RTOS) functionality.

The initial prototypes produced in Phases 1 and 2 helped evaluate critical dimensions and functionalities of the final craft.

4.3 Phase 3 - Spring/Summer Term 2022

The final phase of the project was the largest and most time consuming. It required the most development work and planning to conduct.

The main objectives from phase three are listed in Appendix F. The project can be split into two main groups: the software and hardware development.

The software consisted of developing a controller and receiver board for the craft. This required a transmission protocol to be designed along with several libraries to control various devices such as, buzzers, Brushless DC Motor (BLDCM)s, and Direct Current (DC) Motors.

Most of the hardware for this project was bespoke including, the Printed Circuit Board (PCB)s, Main Body of the Craft, and the Conveyor belt. The majority of the hardware to interface with the PCBs was off-the shelf as they work effectively.

Not all of the software was able to be developed without the integration of some hardware. An example of this would be the development of the transmission protocol shown in Appendix G. Without the use of the nRF24L01 Radio Frequency (RF) chips, it was ineffective to test and debug. As a result of this, the software and hardware were developed in parallel. This also allowed time for manufacturing of components, such as the PCBs and 3D printed parts.

Setbacks in both the software and hardware development were experienced. As a

result of this, the time accounted for in the Gantt Chart had to be skewed further than previously estimated. Some components ordered in early April of 2022 were still expecting delivery in mid May of 2022. This caused significant setbacks in the project.

Along with massively increased shipping times and lack of availability of some vital components, software setbacks were also encountered. The nRF24L01P Application Programming Interface (API) made for the Mbed board was found to be poorly documented and incomplete in parts. This required time to rectify and understand when ideally it would have been an easy implementation.

The PCBs were another source of delay, once ordered the board arrived within the specified timeframe. Upon inspection, however, these boards, which were supposed to be fully assembled, were totally unpopulated. This required the components to be quickly ordered and the boards to populated by hand. As a result of this, everything was put on hold for a few days until these boards were populated and fully tested.

5 Research

The majority of the coursework for the project was conducted during Phase 1 (§4.1), with some continuation into Phase 2 (§4.2). The following three concepts were critically evaluated to find their strengths and weaknesses.

5.1 Concept Designs - Mechanical Assembly

5.1.1 Concept 1 - Dragged Net Based System

Concept 1 features a mono-hull craft that consists of a net that is dragged. The craft would drive through litter, collecting it with the net as it drives past.

As this design was complex and required a more unstable mono-hull design, it was deemed to be the contingency plan and would only be further explored if all other concepts

had been deemed impractical.

5.1.2 Concept 2 - Submerged Basket System

Concept 2 uses a catamaran style hull with a gap in the middle. This gap would be filled with a removable basket that sits under the surface of the water. The craft would drive over the top of surface litter collecting it in the basket.

As this design was similar to other designs currently on the market, this plan was deemed as a backup. It has been proven to be effective but there is little novelty in this design.

5.1.3 Concept 3 - Conveyor System with Elevated Basket

Concept 3 focuses on a conveyor belt with a raised basket to hold the collected litter. This would be straddled across a catamaran style hull. The excess liquid run off which would be filtered and drained back into the water. This would allow basic samples to be taken from the debris floating in the water.

5.2 Concept Designs - Controller and Receiver

5.2.1 Concept 1 - Off the Shelf Solution

Controller and Receiver concept 1 was to use an off the shelf system for controlling the craft. This system would be reliable, functional, but it would be expensive and not allow for versatility of inputs and outputs that would be required.

The lack of input/output versatility would inhibit the use of smart features on the craft. This does not address any newer technologies or concepts that would be suited to this style of custom RC craft.

Therefore it was decided that this concept would be used for initial testing and as contingency plan. Final deployment would be using this concept, if Concept 2 (§5.2.2)

proved to be impractical.

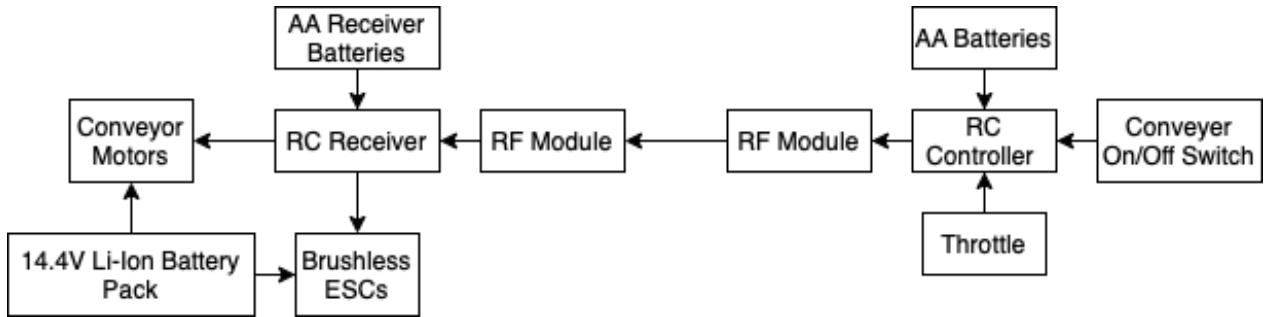


Figure 1: Concept 1 - Off the shelf RC Controller and Receiver

5.2.2 Concept 2 - Custom RF Based Solution

Concept 2 introduces "smart" features such as custom battery management circuitry and litter collection basket capacity sensing. It allowed for a number of inputs and outputs on each system (up to the limit of the board used to control the systems). This would allow for a highly adaptable control system that could be easily adapted an alternative use if required. The main craft could be changed and the controller system would be easily reusable. Using a microcontroller based system allows the ability to add and remove devices as the requirements of the craft change.

One of the downsides of this system is that because it is a custom based system, all of the maintenance is required to be done by the user. This requires a lot of work and effort which would not be the case for the off the shelf system. The other issue with this solution is that the RF chips can be intermittent during their operation something that is also not present with the off the shelf system.

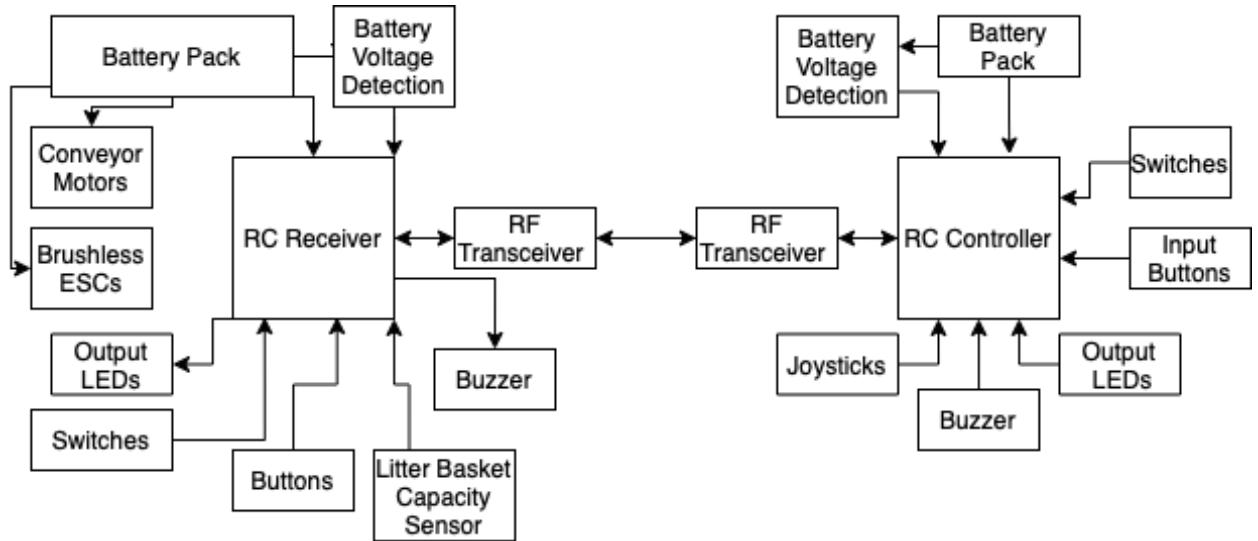


Figure 2: Concept 2 - Custom RC Controller and Receiver

Despite the downsides, this concept had been decided to be the most adaptable and most suited to this application. It is the most complicated of the two concepts; requiring a broad knowledge of embedded systems and battery management technologies. Moreover, the added complexity of the implementation is outweighed by the added functionality and expandability of this system. It was deemed better to design a system that would properly function instead of trying to adapt the correctly available solutions that may not work correctly for this purpose.

6 Optioneering

6.1 Controller

The STM32L432 was selected as the processor used for the controller device for this project. It was decided upon because it was lightweight and powerful enough for its intended use case [4].

Cost was also an important consideration when selecting suitable prototypes. This weighting was important to make sure that the project budget was used in the most ef-

fective way possible. Given that the Mbed OS Integrated Development Environment (IDE) was free to use software the cost of development was not needed to be taken into account.

The two largest competing manufactures within the microcontroller industry, ARM and AVR were included when making a choice about which processor to use. This allowed a full assessment of the given options to take place. The chosen AVR processor was the ATMega328P, found in the *Arduino Nano*. Given that it was an 8-bit processor it would be expected to perform poorly against the 32-bit *STMicroelectronics* boards. One advantage of using the the ATMega328P was its ease of use and setup given the large number of libraries available and popularity within the hobbyist community.

Despite this, the STM chip was chosen, this was primarily due the lack of RTOS functionality on the AVR chip. This would not allow the use of multithreading. The Cortex-L4 chips are lightwieght and do not consumer a lot of power, this would allow the remote to be run for a long time, whilst also enabling features such as RTOS support.

Another factor in the critical decision making process was the ongoing silicon shortage and supply chain issues. This reduced the number of available options for use in the controller.

As a result of the previously mentioned features, the STM32L432 was decided to be the best choice for the controller for use within the controller for Project O.R.C.A.

6.2 Receiver

The STM32F492 was selected for use within the receiver board for this project. This decision making process was conducted in the same manner as the chip selection for the controller. This chip is powerful and includes many inputs and output required to drive the required components for the peripherals [5]

Power consumption for this board was not an important consideration because it was being powered by the same battery that is used by the motors and other peripherals. The power draw of these components is significantly higher than that of the board.

Like with the Controller Optioneering, AVR's ATMega2560 was also considered for use within the receiver. Once again the the RTOS functionality of the STMicroelectronics chip paired with the use of the Mbed studio IDE showed the STM32F492 to be the clear choice for chip selection. Given that it was also an ARM processor like the controller, it would be easy to develop the software for as it would only require the use of one IDE. This meant that code would be easy to write for one of the board and be ported easily for reuse on another controller without having to worry about architecture clashes.

6.3 Peripherals

6.3.1 RF Transciever

In order to deliver the functionality shown in Concept 2 (§5.2.2), an RF module would need to be sourced to allow for control of the craft at range. The nRF24L01 is used widely within embedded systems as a long range transceiver device [6]. Although a basic device it is highly capable of sending limited amounts of data quickly, such as the use case for RC receivers. The nRF24L01 comes as part of a discreet package that contains all of the components that require it to operate correctly. The discreet package is shown in Fig 3.



Figure 3: nRF24L01 Discreet Package

The nRF24L01 package uses an 8 pin interface for communicating using Serial Parallel Interface (SPI) with an added chip enabled pin, Interrupt Request (IRQ) pin, 3.3V and ground. This was the only component in which a pre-written library was used. This library was complicated and worked well. As this was not the main focus of the project it was decided that this library would be used, as it would not detract from the project. This library was written in Mbed OS so it did not require porting over from another architecture.

6.3.2 Brushless DC Motors

The motors for propelling the craft were decided to be BLDCMs. This type of motor was chosen primarily for its water-resistance, given that it contains no brushes unlike a brushed DC motor [7]. BLDCMs are entirely submersible as they do not have any exposed contacts and are wound using enamelled copper wire. For this project the A2212 1400KV BLDS motors were selected.

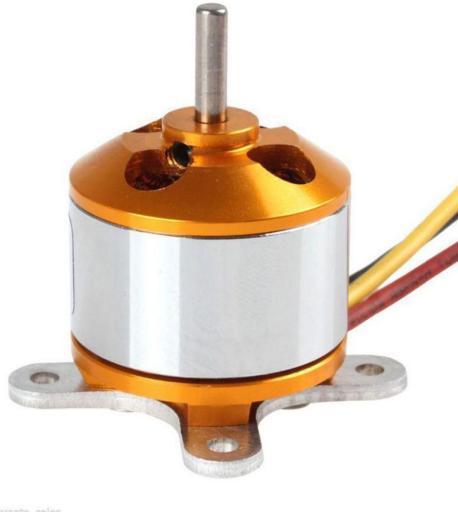


Figure 4: A2212 1400KV Brushless DC Motor

The A2212 motors were chosen for this project because of their small size, ease of availability, and their power output. These motors are capable of 'emptying a sink of water in under nine seconds', so they were deemed to be more than powerful enough for this

project, if paired with the correct propeller.

6.3.3 Conveyor Motors

For the conveyor belt motors, there were two initial choices for motors, either stepper motors, or a brushed motor with an attached gearbox. These two types of motor were the obvious choice for driving a low speed high torque conveyor belt. The brushed DC motors were chosen to drive the conveyor belt for two main reasons, cost and ease of control. Brushed DC motors can be highly geared to have high torque and do not require any specialised motor controller to drive them, unlike stepper motors.



Figure 5: PLG24 Brushed DC Motor

The motors for the conveyor were decided on being the Dunkermotoren PLG24 DC motor with an attached gearbox with a large reduction, shown in Figure 5. These had an approximate torque of 40Ncm with a gearbox reduction of 353:1. These motors are designed to run at up to 30V [8]. These would lead to the motors being underpowered for this application but as speed was not the defining factor this was not an issue. The only issue with using brushed motors is that they are not waterproof. This would require a waterproof cowling to cover the motor to ensure that it didn't get damaged by the water.

6.3.4 Motor Drivers

Both the Brushless and Brushed DC motors for this project would require motor drivers to allow the microcontroller receiver to control the motors correctly. Each type of motor requires a different type of motor driver. To drive the BLDCM motors, the included 30A 2-3S Electronic Speed Controller (ESC)s were used. These were generic no-name ESC but worked well for the motors and allowed effective control required for this project.

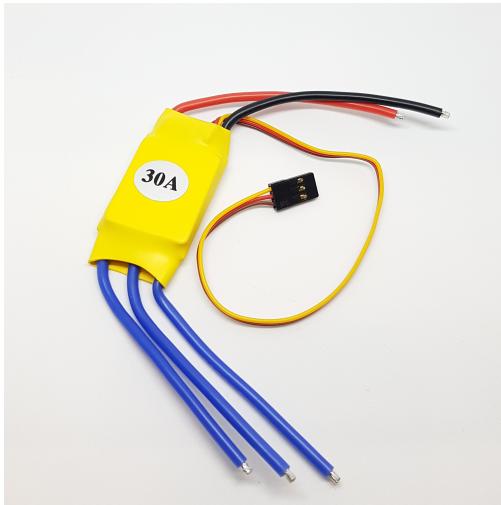


Figure 6: Generic no-name 30A ESC

Unfortunately, these ESCs did not allow reversal of the motors, this meant an additional method of reversing the craft would be required. The easiest solution would be to use two additional motors with reversed propellers to drive backwards.

The brushed DC motors are significantly easier to control, requiring a basic DC current to be put across them, as opposed to a complex 3-phase AC current. This means that just connecting them across the battery would have sufficed to drive these motors forwards. As more control than just on/off was preferable, a motor driver chip would be required to control the movement of these motors. For this, an L298N driver board was chosen.

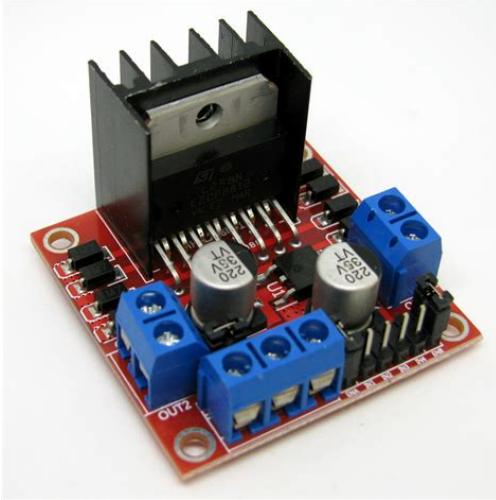


Figure 7: L298N Driver Board for DC and Stepper Motors

The L298N board is able to drive DC motors, reverse their direction, as well as using a PWM signal to control the speed of the motor [9]. These were all preferred features to allow the best control over the conveyor possible.

6.3.5 Batteries

Batteries are one of the most important components for a remote project as they are required to power the entire craft. There also needs to be batteries for the controller, to allow for portability. For the controller Li-Ion batteries, specifically 18650s, were decided upon as they offered good amount of energy storage within a small enough package to fit in the controller. These batteries are widely used and are readily available. They have a capacity of 2200mAh at 3.7V [10]. Two would need to be placed in series to get the required voltage for the circuit to reliably operate.



Figure 8: 18650 Li-Ion Batteries

The ideal batteries for use within the main craft would have been Li-Po batteries but unfortunately due to restrictions within the electronics laboratory in SMB303, Li-Po batteries are not allowed to be used for their instability and fire risk. Because of this, Li-Ion batteries were chosen as a replacement, these were chosen as the next best option for their weight to capacity ratio. Being a prototype the batteries in this craft did not need to be an optimal capacity. The batteries only needed to last just about long enough to prove the concept and be able to perform testing for the proof of concept. With these considerations in mind, a battery from a robotic vacuum cleaner was chosen. This battery was 14.4V and 5200mAh. This would provide approximately 30 minutes of testing with all features for the craft enabled.



Figure 9: 14.4V 5200mAh Robotic Vacuum Cleaner Battery Pack

Upon basic inspection the battery pack was made up of two parallel sets of four in series 18650 Lithium Ion (Li-Ion) batteries, the same type of batteries used to power the

controller. These batteries are easy to charge and have a higher energy density than Lead acid batteries typically used in car batteries.

6.3.6 IR Distance Sensors

Selecting a suitable Infra-Red (IR) sensor was relatively simplistic as they all function in the same way, and have similar advantages and disadvantages. As a result of this, a generic no-name IR distance sensor was used. These were inexpensive and functioned in the exact way specified. The range was adjustable using the small potentiometer on the device. The detection distance was specified to be between 6-30cm, which allowed for ample adjustment to get the right distance.

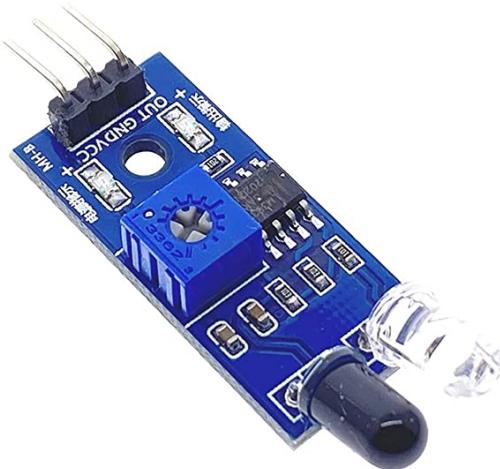


Figure 10: Generic no-name IR Distance Sensor

IR distance sensors were selected as they were simplistic, cheap and required no further circuitry to drive them, meaning they could just be dropped into the schematic without too many operational concerns [11].

6.3.7 Voltage Regulators

Given that many components in this project would need to be powered by the same power input, voltage regulators were used. A 12V, 5V, and 3V were need to get all the

required input voltages for the various components. As the battery voltage fluctuates with varying levels, the linear voltage regulators continue to output the same voltage. This means the components will not be affected by these varying voltages and work reliability.

6.4 Finalised Solution

6.4.1 Block Diagram

The final system level block diagram shown in Figure 11 is the diagram that was decided upon after component sourcing and critical function analysis of the system. The diagram contains the elements proposed in Concept 2 show in §5.2.2, with the addition of battery monitoring circuitry for low voltage detection. A table summarising the block diagram is listed below in Table 1.

Component	Device
Controller Microcontroller	STM32L432KC*
Receiver Microcontroller	STM32F429ZI
Brushless DC Motors	A2212 BLDC Motors
BLDC Motor ESC	Generic no-name 30A ESC
Brushed DC Motor	Dunkermotoren PLG24
Brushed DC Motor Driver	L298N
Controller Batteries	18650 Li-Ion Batteries
Buzzer	Generic no-name buzzer
12V Regulator	12V Fixed LDO*
5V Regulator	12V Fixed LDO*
3V Regulator	12V Fixed LDO*

*selected due to constraints caused by silicon shortage.

Table 1: Component Selection

Details on how the controller microcontroller software and hardware was developed and interfaced within the scope of the project can be found in §?? and §?? respectively.

Details on how the receiver microcontroller software and hardware was developed and interfaced within the scope of the project can be found in §?? and §?? respectively.

Details on how the ESCs were interfaced with can be found in §??.

Details on how the L298N was interfaced with can be found in §??.

Detailed on how the 12V, 5V, and 3V Regulators were interfaced with can be found in §??.

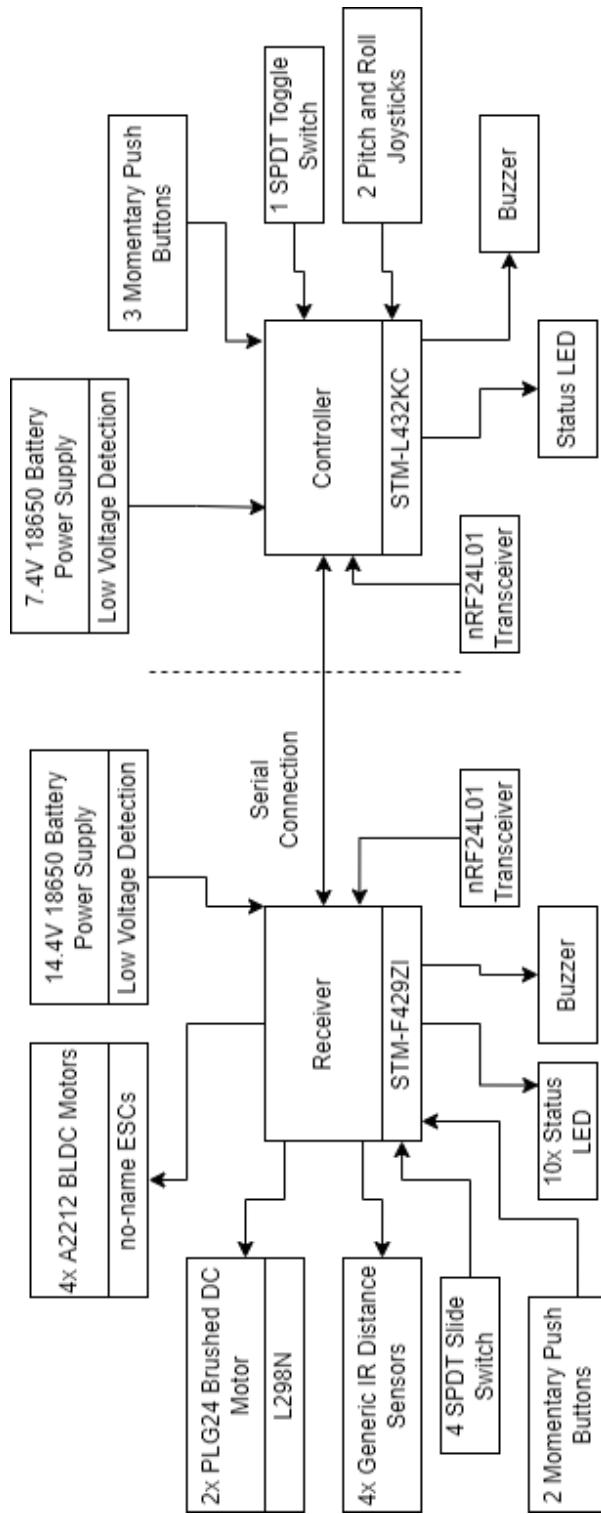


Figure 11: Final System Block Diagram of ORCA

6.4.2 Bill of Materials

A full Bill of Materials (BOM) can be seen in Appendix H. Whilst the budget from the University was set at £100, the project exceeded that amount. Given that most of the materials for this project had been recycled, the actual budget for the project is less than shown in the bill of materials.

The high cost of this project is due in part to this project being a prototype, and in part to the custom parts that needed to be manufactured for this project. Components are always expected to be more expensive at the prototyping scale of production.

Materials were, for the most part, sourced from responsible VAT registered distributors; which was important when selecting components for mass production. The majority of the components were sourced from Farnell, DigiKey US, CPC, or Ooznest. Some of the other components were sourced from Amazon and eBay. These parts could not be verified to be from a responsible distributor, but these components could have been sourced from other locations if the quantity of the components was larger.

The fabrication of custom Surface Mount Technology (SMT) PCBs has been included within the BOM for this project as they were considered to be part of the final prototyped solution. Despite these not being a finalised PCB design for further manufacturing, if applicable, it was a cost that was considered during the prototyping phase of the project. The PCBs were important in the production of this project as they allowed the secure mounting of the RF modules. Given that all of the components needed mounting on a structure that would need to be remotely controlled and manoeuvred, the decision to make PCBs was clear. The PCBs were manufactured in Shenzhen, China by JLC-PCB.

The project was concluded within time scale set but with a slightly expanded budget. Given further time, this project could have been completed closer to the original budget. Due to the expensive nature of some of the components used however, the budget of £100 would likely have not been sufficient for this project without significant cuts to functionality and/or goals.

7 Software Design

7.1 Controller

7.1.1 Architecture

The Controller is entirely multi-threaded. The decision was taken to use multi threaded architecture due to the nature of the system being used; monitoring and controlling sensors with interrupts coming from multiple sources. A single threaded architecture was initially used during initial testing. This was found to be impractical as the state machine used became cumbersome. When communicating using RF the controller could not respond to any other commands from any other devices. This may not have been quick enough to react to any other inputs from the user.

Because of this, the controller architecture was split into threads, listed in Table 2. Each entry in the table is represented by a thread in the program that is initialised during start-up.

Thread Name	Function
Pot Thread	Polls Values of the Joysticks and sends over RF
Input Thread	Used to handle inputs when flags raised

Table 2: Threads for Controller

7.1.2 Serial Communications

The serial communication for this project was done over SPI using nRF24L01 modules to send data. As there was already an nRF24L01 library written for Mbed [12]. This abstracted the serial interface from the user to just writing to the RF devices, and allowing the library to handle the rest.

This library was externally written and whilst it was functional, it was a bit rough around the edges and felt, in parts, to be unfinished. Whilst this functionality was not used, it

would have been nice to have a complete library. Despite this, the library was operational and only required the data to be sent across using char arrays, whilst fiddly this was not too difficult to achieve.

This library could have been re-written to include its missing functionality, but this would have taken up a significant amount of the project given that the serial/RF communication was not the main objective.

Despite the class being functionality incomplete, it was easy to use for sending data using radio. Code 1 shows how to create the object for the nRF24L01 Class. It requires the use of an SPI interface on the board.

Code 1: nRF24L01 Class Constructor

```
1 #include <nRF24L01P.h>
2 nRF24L01P radio (MOSI, MISO, SCK, CSN, CE, IRQ) ;
```

Once the object is declared, the code in Code 2 is then used to set up the nRF24L01 so that it can be used correctly.

Code 2: nRF24L01 Set Up Transmit Mode

```
1 radio.powerUp();
2 radio.setTransferSize(TRANSFER_SIZE);
3 radio.setAirDataRate(NRF24L01P_DATARATE_2_MBPS);
4 radio.setRxAddress(NRF24L01_RX_ADDRESS);
5 radio.setTxAddress(NRF24L01_TX_ADDRESS);
6 radio.setTransmitMode();
7 radio.enable();
```

Code 3 shows how the data is written to the nRF24L01 module. This requires the data to be in a char array format, ideally the module would allow strings and integers to be passed to it, but the use of char arrays was acceptable.

Code 3: Sending Data Using nRF24L01

```
1 radio.write(char *data, int pipe, int count);
```

The serial communication was conducted using custom transmission protocols, outlined in Appendix G. These were designed to be short, 5-bits, and easy to extract the data from on the other end. This system was used to make the transfer of data to faster and efficient with hopefully minimal packet loss.

7.1.3 Buzzer API

The buzzer used for this project is a standard piezo buzzer that required a Pulse Width Modulation (PWM) pin to drive at a given frequency. To make the buzzer audible and fit within the western standard of music notation, a class was written to encapsulate the buzzer functionality, allowing the buzzer to be easily used.

This buzzer class was inspired by the University of Plymouth's buzzer class written for the Module Support Board.

This class has been written such that it is thread safe. The object-oriented nature of the class has allowed the class and code to be utilised on both the controller and receiver, both of which utilise buzzers as a method of alerting the user.

Code 4: Buzzer Class Constructor

```
1 #include "Buzzer.h"  
2 Buzzer buzzer(BUZZER);
```

Using this API, the buzzer could be turned on or off, the note can be changed, and the octave the note is in can also be changed. This allowed multiple different alerts to the user to be utilised. This increased the versatility of the buzzer.

This class includes operator overloads such that using the buzzer is easy and intuitive.

7.1.4 Main Code

The main code on the controller board runs in two separate threads, one for monitoring the button and switch states, and one for monitoring the joysticks and extracting their relative positional data.

They three buttons and switch are all driven using interrupts. When the interrupts are triggered, the Interrupt Service Routine (ISR) raises a thread flag which wakes up the thread. The thread then handles the flag then goes back into the waiting state where the thread is put to sleep and not scheduled until another event flag is triggered.

Being an [analogue input](#), the joysticks cannot be run using the same interrupt based methodology. The joysticks were run using a [ticker](#). The ticker is set to create an interrupt every 200ms and runs the joystick ticker ISR. This wakes up the joystick thread, polls the joystick values, if required it sends the values of SPI to the nRF24L01 then puts the thread back to sleep until the next ticker interrupt.

If the joystick values have not changed, the new joystick values get stored and then the thread goes back to sleep. This helps reduce power usage and RF usage when unnecessary.

Code 5: Joystick Value to Serial Data to be Sent

```
1 newLeftPitchVal = abs(newLeftPitchVal);  
2 sprintf(tempThrottleChar, "%d", newLeftPitchVal);  
3 fwdLeftPitch[3] = tempThrottleChar[0];  
4 radio.write(fwdLeftPitch, DEFAULT_PIPE, TRANSFER_SIZE);
```

Code 5 shows the method of transferring the data from integer form into the char array that is required to send data over the nRF24L01 device. This is an example of how this library could have been improved for this project. If the nRF24L01 library was able to accept integers to be sent, the code on the controller could have been simpler and easier to understand.

Despite its complexity, the code for extracting the joystick values worked repeatable and did not cause any detectable false readings.

7.2 Receiver

7.2.1 Architecture

The Receiver is entirely multi-threaded. The decision was taken to use a multi threaded architecture due to the nature of the system being used; controlling multiple real time devices that, many of which do not need rapid periodic updating. A single threaded architecture was initially utilised during testing. This was found to be impractical as the number of inputs and outputs became unmanageable using rapid polling. Reading from the RF module is a blocking operation. This would cause the program to halt when fetching and reading the data from the RF module. This could cause updates to the motor speeds and other sensors to be missed, potentially causing delays to the real time system.

Because of this, the architecture was split into threads, listed in Table 3. Each entry in the table is represented by a thread in the program that is initialised during start-up.

Thread Name	Function
Left Motor Thread	Controls throttle Values of Left Motor
Right Motor Thread	Controls throttle Values of Right Motor
Radio Thread	Used to handle incoming RF Data
LED Thread	Used to change the LED status'
IR Thread	Used to handle interrupt from when litter collection basket is full
Input Thread	Used to handle inputs when flags raised
Battery 30 Percent Thread	Used to raise warning when battery is at 30 percent
Battery 15 Percent Thread	Used to raise warning when battery is at 15 percent

Table 3: Threads for Receiver

7.2.2 Serial Communications

The serial communication for this project was done over SPI using nRF24L01 modules to send data. An overview of the serial communication can be seen in §7.1.2.

The main difference in serial communication on the receiver, is that it primarily uses receive mode. Therefore the set-up for the RF module is almost identical, except that it is initialised into receive mode.

Code 6: nRF24L01 Set Up Receive Mode

```
1 radio.powerUp();  
2 radio.setTransferSize(TRANSFER_SIZE);  
3 radio.setAirDataRate(NRF24L01P_DATARATE_2_MBPS);  
4 radio.setRxAddress(NRF24L01_RX_ADDRESS);  
5 radio.setTxAddress(NRF24L01_TX_ADDRESS);  
6 radio.setReceiveMode();  
7 radio.enable();
```

The serial communication on the receiver board also used the transmission protocols, outlined in Appendix G. This allows the boards to communicate effectively.

To handle the incoming transmission codes, a separate radio thread is used, as shown in Table 3. This thread does not cause any external blocking behaviour to any other threads. This allows the system to remain real time.

7.2.3 Buzzer API

The buzzer API used for the receiver is the same as used within the controller. This can be found at §7.1.3.

7.2.4 L298N API

The DC motors were driven using an off the shelf L298N motor controller. This required a simple API to be written. This was to make the motors easier to control by abstracting functionality away from the user.

This class has been written such that it is thread safe. The object-orientated nature of the class would allow the class to be used on any Mbed project that utilises an L298N to drive DC motors.

This API uses four arguments, a motor A, a motor B, an enable, and a motor enabled. The motor enabled argument has a default value so it does not need to be used. If unused, the value will default to 0. This can be seen in Code 7.

Code 7: L298N Class Constructor

```
1 #include "L298N.h"  
2 L298N DCMotor1(PinName motorA, PinName motorB, PinName enable, int  
motorEnabled = 0);
```

Using this API, the motor can be enabled, disabled, and the direction can be set and read. These methods can be seen in Code 8.

Code 8: L298N Class Methods

```
1 void setDirection(int direction);  
2 void enableMotor();  
3 void disableMotor();  
4 int direction();
```

Given the nature of the motors being controller, basic DC motors, the control was trivial. Despite this, abstracting this control made the motors easier to control within the main body of the code.

7.2.5 ESC API

Unlike the brushed DC motors, brushless motors require an ESC to operate. This requires more complex control to make the motor move. The ESCs require a PWM signal that at 50Hz varies between 1ms-2ms. This is similar to a servo control signal using a PWM output [13]. This class was inspired by the control system for servos.

This class has been written such that it is thread safe. The object-orientated nature of the class allows the class to be used on any Mbed project that utilises an ESC that requires a 50Hz signal with a duty cycle varying between 1ms-2ms.

The ESC API uses two arguments, the PWM pin, and calibrate. The calibrate argument has a default value so this does not need to be used. If unused, the value will default to 0. This can be seen in Code 9.

Code 9: ESC Class Constructor

```
1 #include "ESC.h"  
2 ESC brushlessMotor(PinName pin, int calibrate = 0);
```

Using this API the motors can be set a certain speed, have the PWM value altered, and the value of speed can be read. This can be seen in Code 10.

Code 10: ESC Class Public Methods

```
1 void write(float speed);  
2 void setPWM(int PWMValue = DEFAULT_PWM_VALUE);  
3 float speed();
```

This API also contains private members, these are used to connect the ESC, calibrate the ESC, and normalise the speed values. These can be seen in Code 11.

Code 11: ESC Class Private Methods

```
1 float normalise(float speed)  
2 void connect();
```

3 | `void calibrate();`

The normalise method is used to ensure that a value between 0-1 is written to the ESC. If the value was not between these values, the ESC may not behave as expected.

To connect the ESC, the throttle value is set to its minimum value then held there until the ESC beeps. This can be for varying amounts of time. Therefore 5s was chosen as this would be long enough for the ESC to connect. The calibration of the ESC is more complex. It requires the throttle value to be at its maximum value for 4s, then at its minimum value for 4s. This allows the ESC to 'learn' the range of values that it is expecting as an input. This calibration should only be required for the first time the ESC and board are connected together, after this, the board remembers the previous values.

7.2.6 Main Code

The main code on the receiver board runs in eight threads. These have been outlined in Table 3. The left and right motor threads are for controlling the speed of the motors; the radio thread is for reading the serial data and either raising the appropriate flag for that action of performing an action dictated by the transmission codes. The LED thread is used for turning the LEDs on and off in a thread safe way that can be handled when the scheduler is available to, as these LEDs are less important than the functionality of the motors. The other remaining four threads are all controlled by flags controlled by interrupts. This allows the inputs to be responsive whilst allowing the system to remain real time.

When the IR sensors get triggered, the ISR is triggered raising a flag to alert the board that the collection basket is full. This activates the IR thread which disables the IR ISRs and triggers the buzzer periodically.

When the battery level is running low, the battery low input pins are pulled high. This triggers an ISR which periodically sounds the buzzer,. This alerts the user that the battery

is running low.

Table 4 shows the functionality of the buttons and switches on the receiver board.

Input Name	Function
Button 1	Disable Alarms and deactivate IR sensors
Button 2	Litter basket emptied, Resets Sensors and Enables Buzzer
Switch 1	Disable Motors
Switch 2	Enable Motors
Switch 3	Not in use
Switch 4	Not in use
Switch 5	Not in use
Switch 6	Not in use

Table 4: Functionality of Buttons and Switches in the Main Receiver Code

8 Hardware Design

8.1 Prototyping

The initial prototyping started during Phase 2 (§4.2) of the project. The initial prototyping goals were discussed and clearly laid out before any work began. The primary object of the prototyping stage was to create a small scale model craft that fits into testing tank eg. a bath. After that, it would need to be controlled remotely using an RC controller. The third aim of prototyping was to investigate how to make a working conveyor system and what the advantages and disadvantages of this system may be. The final aim was to create an approximately 1:1 scale cardboard prototype to do a variety of tests on.

8.1.1 Small Scale Design

The small scale prototype design was made using milkshake bottles and 3D printed Polylactic Acid (PLA) parts. This design was christened the Milkshake Catamaran. The design for this can be seen in Figure 12.

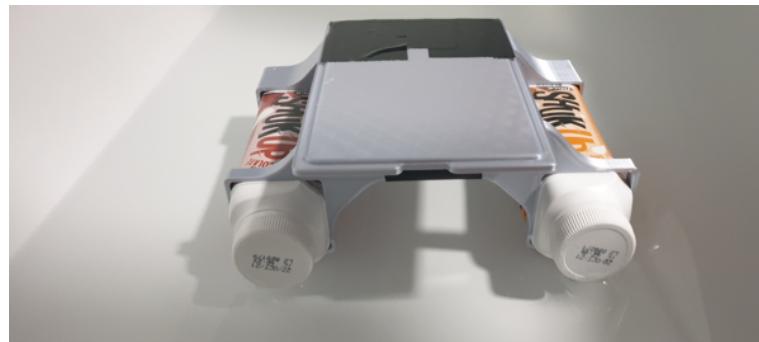


Figure 12: Milkshake Catamaran Prototype

This design was tested for buoyancy and stability in the bath. Being a Catamaran design, the craft was extremely stable and well balanced. This was encouraging for the future developments of this project.

Once the base catamaran had been tested, RC electronics were added. This enabled the craft to be controlled. This was used to test propulsion systems. The fully loaded Catamaran can be seen in Figure 13.



Figure 13: Milkshake Catamaran with RC electronics

From this testing it was discovered that the maximum capacity of this craft was about

670g. This was not surprising as each of the milkshake bottles was 335ml, so the total mass of displaced water was 670g. Given that the weight of the craft including the basic electronics system was 627g, there was very additional capacity for the craft to carry any litter.

From this discovery, it was decided that the scale of this craft would need to be larger to accommodate a heavier electronics system, batteries, and the additional overhead to carry any litter that it may collect.

8.1.2 1:1 Scale Prototype

Once the small scale prototype was completed, it was clear that a larger prototype was required. This would allow for an approximate capacity to be calculated as well as to gain a sense of scale for the craft.

The prototype was made primarily of cardboard and was held together with duct tape. This helped get a rough estimation of size and scale to be used during the CAD stage of the project. The 1:1 scale prototype can be seen in Figure 14.



Figure 14: 1:1 Scale Cardboard Prototype Hull

Once the prototype was constructed, it was then tested in the water at Plymbridge Woods. This area was used as it was a calm, inland body of freshwater that was easy to

get to. Although not very heavy, the craft was large and so getting somewhere that didn't require it to be carried large distances was preferable.

The hull of the craft was waterproofed by attaching waterproof Oxo-biodegradable plastic bags to the outside of the hull. Although not perfectly bio-degradable, they are much better than the standard Linear Low-Density Polyethylene (LLDPE) bags that are typically used. After this testing was completed, the bags were disposed of in the appropriate manner, as to not increase pollution rates. Figure 15 shows the craft in the water.



Figure 15: 1:1 Scale Cardboard Prototype Hull in Water with Additional Load

Upon testing, the craft was able to hold at least 15Kg of additional load. The craft was not tested to its limits because it was made of cardboard and held together with duct tape. Having calculated the volumetric displacement of the hulls, using the cross sectional area multiplied by the length of each of the hull, it was estimated that the approximate buoyancy of this craft was about 40Kg. This was much larger than the tested load that was used.

After testing the cardboard prototype, the material for the construction of the shell of the main craft. Table 5 shows the pros and cons of each of the considered materials.

Material	Pros	Cons
Aluminium	<ul style="list-style-type: none"> • Strong • Non-corrosive 	<ul style="list-style-type: none"> • Expensive • Difficult to machine by hand • Heavy
Acrylic	<ul style="list-style-type: none"> • Light • Easy to machine by hand • Non-corrosive 	<ul style="list-style-type: none"> • Brittle • Not hard wearing • Very Flexible
Dibond	<ul style="list-style-type: none"> • Strong • Lightweight • Easy to machine by hand • Non-corrosive 	<ul style="list-style-type: none"> • Relatively Expensive

Table 5: Material Considerations for Main Craft hull

Ultimately Dibond [14] was chosen for this project as it had all of the positive attributes of the other two materials, without any of the perceived negatives. The main drawback to using the material was the cost. Given that the Dibond was reused material, cost was not an important factor in the decision making process.

8.1.3 Conveyor Research

Early in the project it was decided that a conveyor would be the preferred method of collecting litter. A conveyor belt would provide a suitable level of challenge whilst adding some additional novelty to the design. A conveyor belt would also allow the collected litter to be removed from the water sooner. Additionally, removing the litter from the craft will not require the whole craft to be removed from the body of water it is operating in.

One of the biggest issues when looking at the conveyor was the material of the belt. This was vital to get correct in order to create an effective conveyor belt. Table 6 shows the pros and cons of each of the considered materials.

Material	Pros	Cons
Linoleum	<ul style="list-style-type: none"> • Waterproof • Cheap • Moderately Flexible • Non-reactive 	<ul style="list-style-type: none"> • Heavy • Coefficient of friction decreases when wet • Expensive
Cargo Netting	<ul style="list-style-type: none"> • Strong • Waterproof • Light • Cheap • Non-reactive 	<ul style="list-style-type: none"> • Litter may get caught • Hard to get into a strip • Easy to get caught in mechanism
Plastazote	<ul style="list-style-type: none"> • Waterproof • Cheap • Lightweight • Easy to source • Non-reactive • Very Flexible • Coefficient of friction increases when wet 	<ul style="list-style-type: none"> • Stretches and Deforms

Table 6: Material Considerations for Conveyor Belt

It was ultimately decided that Plastazote [15] would be used for the conveyor belt. It was overall the best material for the job. It had all the best aspects of the other two evaluated materials but without their negatives.

Plastazote is easy to machine. Adding parts to the conveyor belt was as easy as poking a hole in the material with a round file and pushing through a bolt. It was a mixture of this along with the points highlighted in Table 6 that made the decision to use this material such an obvious one.

8.2 CAD Design

The majority of this project was designed in Fusion 360. The decision to use Fusion 360 was made primarily as it was the CAD package that was used by the university. This allowed access to features that would not be available to non-premium or non-students

members. As it was the CAD package of choice from the university, it was already well known and any design level issues could be sorted quickly and easily.

The design for ORCA was split into two sections, the main craft assembly and the conveyor assembly. This have been outlined in §8.2.1 and §8.2.2 respectively.

8.2.1 Main Craft

The main craft was designed using the measurements from the prototype craft. The prototype craft was a good size and shape, it was stable, and it could carry a large capacity of litter without sinking. These were vitally important when evaluating the size that the craft had to be.

One of the major considerations when designing hardware of this size is how it will be constructed, especially with non-ideal working conditions such as using the pavement outside. This was one of the driving thoughts when designing this craft. The other was how to maximise the buoyancy of the craft and how to keep the overall weight down so that it would float.

It was quickly decided in the design process that the exterior panels would be made out of Dibond and that the floats would be made out of fibre glass and expanding foam. This was decided so that the floats of the craft would be waterproof and tough. The Dibond was chosen because it is strong, slightly flexible, and weatherproof.

Accuracy of measurements for this design were important to be kept sympathetic and not overly precise. This is because all of the exterior panels, base panels, and floats were created by hand. If the measurements were too precise ($< 1\text{mm}$) which is possible in Fusion 360, the craft would quickly become very difficult to construct as these slight lack of tolerances add up across the design.

The supporting struts, filter holders, and conveyor holders were decided to be 3D printed. This made sense as they were complex shapes that needed to line up accurately. To attach the panels to the struts, it was decided that RivNuts would be used.

Overall, the CAD file contained more than 50 different components with about half of these being planned to be 3D printed. The finalised design for the craft can be seen in Figure 16.

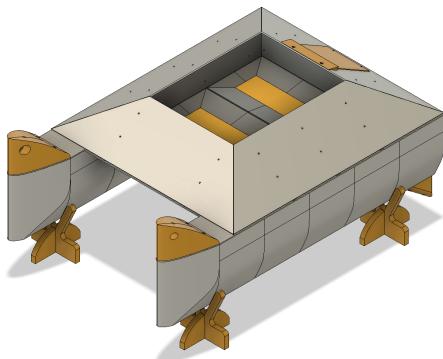


Figure 16: CAD for the Main Craft

8.2.2 Conveyor Belt

Once the design of the main craft was completed, the design for the conveyor belt could begin. The conveyor was more complex to design than the main craft. The majority of the parts on the conveyor were planned to be 3D printed. It needed to fit within the space left on the front of the craft, this limited the width the conveyor could be.

As the majority of the parts for the conveyor were going to be 3D printed, the tolerances on the parts could be tighter than those on the main craft. This precision was important to maintain as it is notoriously difficult to keep conveyor belts tracking straight.

The important parts of the design were to keep it lightweight and simplistic. The top plate of the conveyor is made from Dibond, the same material as the body panels for this craft.

For the main struts of the conveyor, 2020 aluminium extrusion was used, that allowed for minor adjustments to tighten the belt and keep it running true. Designing in these tolerances and features was tricky.

Overall the conveyor belt was designed to be light and strong and able to function

effectively without getting caught and snagging. This was the most difficult sub assembly to design and thus took the most amount of time to complete.

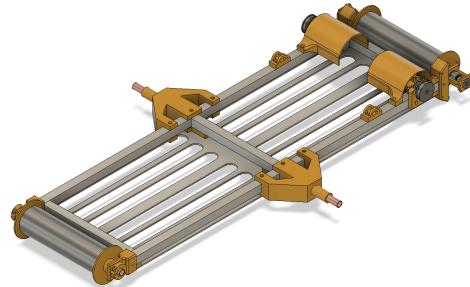


Figure 17: CAD for the Conveyor Belt

8.3 PCB Design

For this project, two PCBs were designed and developed using KiCad. The decision to use KiCad was based primarily on the fact that it is open-source, user-friendly, and allowed custom components to be designed for use on the PCBs. Because KiCad is free-to-use software, it required no membership or sign-up processes.

Another advantage of using this software was the community support. Any issues that were not able to be directly resolved could be investigated and solved quickly with help from the supportive KiCad community.

Once both PCBs had arrived they were soldered by hand. These PCBs were supposed to be fully assembled by JLC-PCB but a misunderstanding during the manufacturing process meant that the board arrived unassembled. Whilst not devastating to this project, it did require a couple of days to fully assemble and test these PCBs, something that should have been done at the factory.

8.3.1 Controller PCB

Before a PCB was designed, a circuit schematic first needed to be developed. This started by evaluating the hardware that was going to be used. This hardware can be seen on the controller side of the finalised block diagram in Figure 11.

The first stage of was to mock up all of the hardware on a breadboard with jumper wires. This would allow the schematic to be drawn up from this, once the hardware is known to be working. This can be seen in Figure 18.

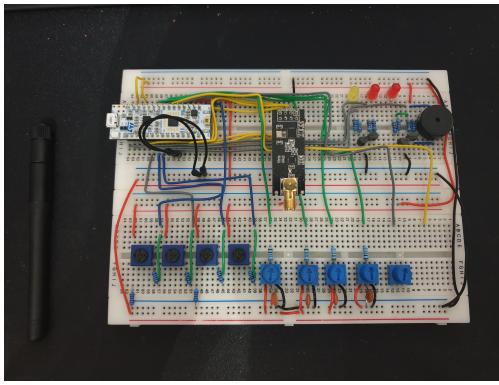


Figure 18: Breadboard for hardware testing for controller PCB

Once the hardware and schematic were verified, a schematic was created for use with the PCB. This can be seen in Appendix I.

The hardware on this PCB was decided to be SMT wherever possible. This was to cut down the size of the PCB but also the placing of all of the components. There are components on the front and back of the board and SMT allows these components to be place on adjacent sides right on top of each other.

Once this had all be decided upon, the final PCB could be designed. This is shown in Figure 19.

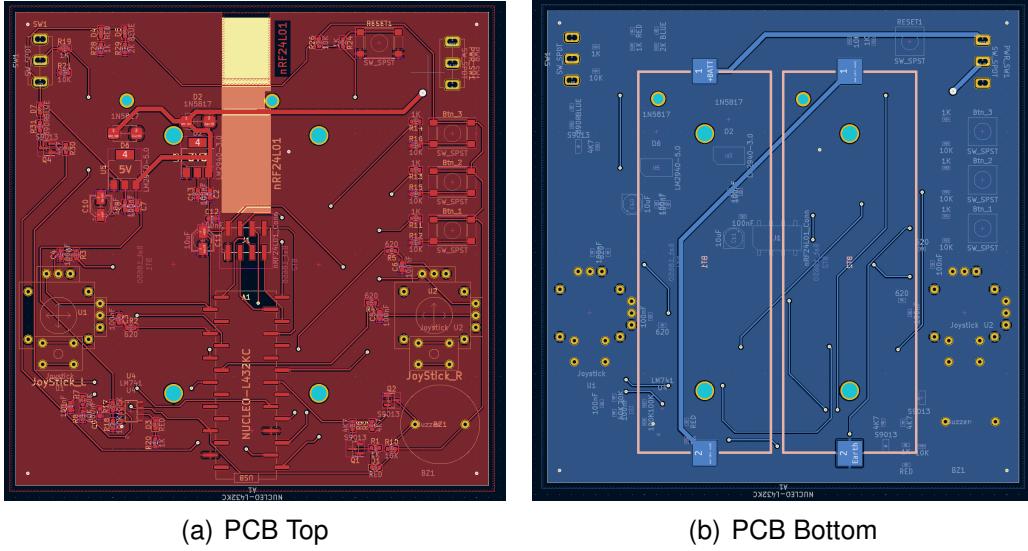


Figure 19: Top and Bottom sides of Controller PCB design

8.3.2 Receiver PCB

Much like with the controller, before a PCB was created for the receiver, a schematic firstly needed to be created. The first step of this was to evaluate the hardware that needed to be used for this process. This hardware can be seen on the receiver side of the finalised block diagram in Figure 11.

The first step in producing the schematic was to mock all the hardware on breadboards using jumper wires. From this, the hardware can be tested before the finalised version of the schematic is drawn up. The breadboard and testing can be seen in Figure 20.

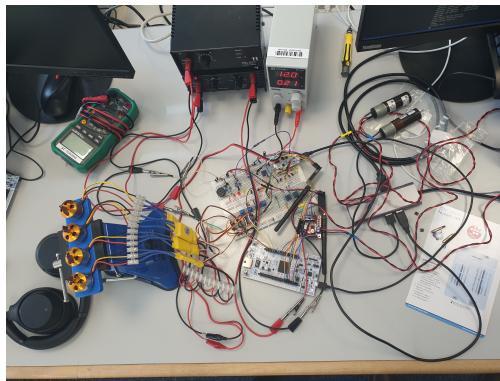


Figure 20: Breadboard for hardware testing for receiver PCB

Once the schematic and hardware was all verified to be working, a schematic was created for use with the PCB. This can be seen in Appendix J.

Much like with the controller board SMT was chosen to be used wherever possible on this board. This was primarily to keep the board as small as possible. As this was going to be placed on top of the Nucleo-F429ZI board, it needed to fit within the foot print of the board. This required the use of SMT as this reduces the footprint of components significantly.

After the schematic had been finalised, the final PCB could be designed. This is shown in Figure 21.

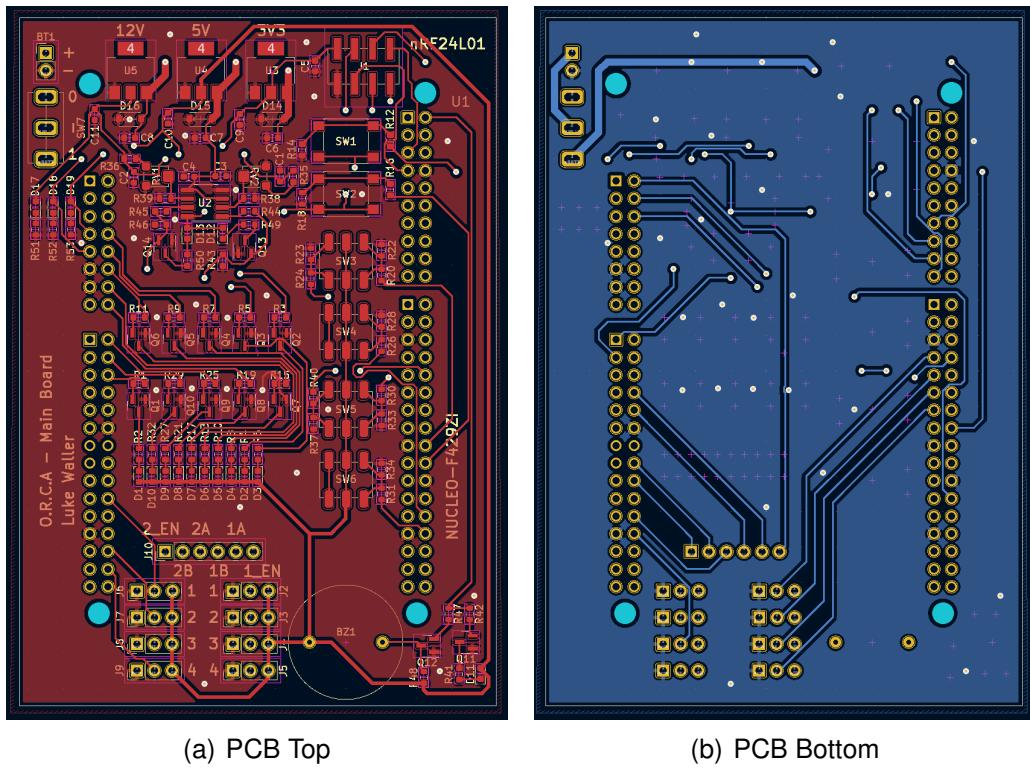


Figure 21: Top and Bottom sides of Receiver PCB design

9 Testing

The testing for this project was split into three main parts, the PCB testing, the software testing, and the hardware testing.

The testing was conducted in this order as this was the most optimal well of testing. For the software to be tested, the PCBs needed to be fully assembled. This was because this system was a hardware based embedded system which would not work effectively without the hardware being in place. Whilst it was possible to emulate hardware in software, this was deemed to be a poor use in time when the hardware was already available.

9.1 PCB Testing

The PCB was an unexpected task to be completed during the testing phase. As mentioned in §8.3, the boards were supposed to arrive fully assembled. This added additional time to the testing phase, although this was a manageable setback.

The boards were soldered by hand using a fine tipped soldering iron. Whilst a challenge, it only took a couple of hours to complete both the receiver and controller board. After the assembly of the boards, they needed to be checked and tested. This took the longest time in testing.

Inspecting the soldering joints on the board was the most time consuming part of the testing. The joints were first inspected by eye, then using a continuity tester. Light Emitting Diode (LED)s, and other polarise components need to be checked that they were in the right orientation before the board was powered on. This was done by inspecting the schematic, PCB layout and inspecting the board to make sure they matched.

Once the soldering and component placement inspection had been tested, a simplistic dummy software was written to test each of the components on the board.

9.2 Software Testing

The software for both the controller and receiver is hardware based. This required the boards to be fully assembled, verified and tested before the main software could be tested as discussed in §9.1.

To test the software, the two boards were connected up as they would be in the final design. Initially the two boards were tested separately. These boards were connected to a computer via a serial interface. This allows the internal states of the boards to be shown on the computer. This was helpful in debugging the boards separately.

Once this initial testing had been conducted, both boards were tested simultaneously. This testing was done primarily to test the RF interface. This interface was responsible for sending across the transmission codes, seen in Appendix G. The code to be printed was outputted through serial, and the code that was received was also outputted through serial. Comparing these two values next to one another identified that the RF communication was successful.

Once it was established that the RF communication was successful, a range test was required. This was done by connecting the two boards up to portable laptops, and walking apart until the transmissions stopped being received. Using this method of testing, the transceivers managed to communicate at a range of about 500m whilst being outside. This was a good distance that would be more than suitable for operating the craft using a remote control.

9.3 Hardware Testing

In order to achieve a stable platform, with enough buoyancy to support the system, at the scale suitable to be manufactured and transported, a catamaran design with two side floats connected to a central deck plate was created. The superstructure was designed to be waterproof, with a central, removable collection tray in the middle. The following elec-

tronics are mounted within the superstructure and comprise of: battery pack, specialised bespoke PCBs for remote control, motor control for both motion and conveyor belt drive.

10 Assembly

The main body and frame of the deck and superstructure was made from a sheet of Dibond. The total thickness was 3.2mm. This material is used extensively for advertising signs. The entire project was cut from a single sheet 2m x 2m which was salvaged from a skip. This material was ideal because it was strong, light and not affected by sea water. It is easy to cut with a jigsaw or handsaw and finished with sandpaper. When used in the structure it provided a rigid, lightweight framework that was waterproof and needed no additional surface finish to make it waterproof.

The collection tray was a plastic mushroom box which was found on the beach, washed up after a storm.

10.1 Floats

The side floats were made using two types of mould. The design required a buoyancy of 35Kg to ensure a stable platform. Tests and calculations determined the required float size. Each type of mould comprised two sections made by 3D printing using PLA – itself a biodegradable material. The two parts were bolted together and used as a mould to prefabricate sections that were cast in the mould using expanding builders foam. This is cheap and readily available waterproof material.

The first mould was for the bow section which was shaped to give a streamlined front to traverse through the water. One was used on each side at the front. The second mould was used to produce the other sections – five of each per side. The size of mould was restricted by the maximum size that could be printed using the 3D Printer. The two part mould can be seen in Figure 22.



Figure 22: 3D Printed 2 part Float Mould

In order to ensure the cast section was removable from the mould, a biodegradable food bag was used to line the mould. The bag was then filled using the builders foam and care was taken to ensure the material was squeezed into all the corners. A key characteristic of the builders foam is that it expands and the mould only needed an 80% fill to ensure that the shape was achieved. Once the foam had set, it was able to be removed from the mould. The bag could then be removed from the foam. The top of the moulding was then levelled by cutting the foam with a hand saw. These activities were undertaken outside in fresh air, using masks and suitable PPE to ensure personal safety measures were observed at all times. Once completed, the sections were placed together in line on a Ski made from the Dibond and the gaps filled with builders foam. This can be seen in Figure 23.



Figure 23: Foam Float Assembly

Inserted into the Ski were 6mm RivNuts so that it could be bolted through from the top of the deck. The float attached to the skis can be seen in Figure 24.



Figure 24: Foam Floats attached to Skis

Once dried, the sections were smoothed using sandpaper and a surform file to give a smooth surface. The floats, one for each side, were then covered with fibreglass and gel coat to give a hard waterproof surface. The technique for adding fibreglass to the hulls can be seen in Figure 25.



Figure 25: Fibreglass Added to the Hull Section

At the rear of each float, a motor mounting plate was integrated into the structure to provide a secure mounting for the drive motor supports. There were two motors on each side, one for forwards and one for reverse. The holders that were being bolted to the motor supports were printed in PLA.

The floats were initially glued to the underside of the deck using builders silicon and

then bolted to the deck. Once dried the builders silicon was used to seal the outside edge of the float where it met the deck. This stops the ingress of water which may lead to corrosion.

10.2 Superstructure

All parts were cut by hand, using a power jigsaw and hand saw, from the polycarbonate sheet. Full size drawings, printed from the 3D CAD models were used to mark out the piece parts. An example of the technical drawings can be seen in Appendix K.

The external panels were supported and fixed to triangular support frames, shown in Figure 26, printed on a 3D printer.

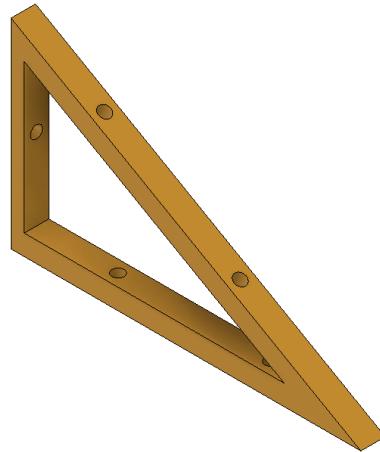


Figure 26: CAD Model of Support Struts

These were designed to support all the panels and fix them to the deck underneath. The Triangular support frames had RivNuts inserted using a soldering iron so that the panels could be bolted onto the frame.

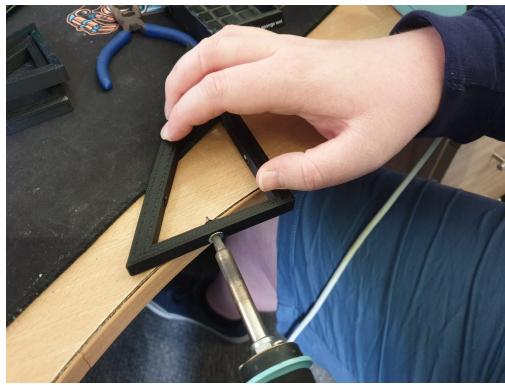


Figure 27: RivNut Inserted Into Support Strut

Fish tank filters were used to filter the water that drained through the collection basket. The specially designed support frames, for the two filters, were 3D printed using PLA, these filters can be seen in Figure 28.



Figure 28: Fish Tanks Fitted to the Craft

The filter supports and the external panels were sealed using builders silicon. This provided a waterproof seal and glued the edges together to strengthen the overall structure. At the rear of the superstructure a removable was mounted that gave access to the electronics and battery mounted inside. This is shown in Figure 29.

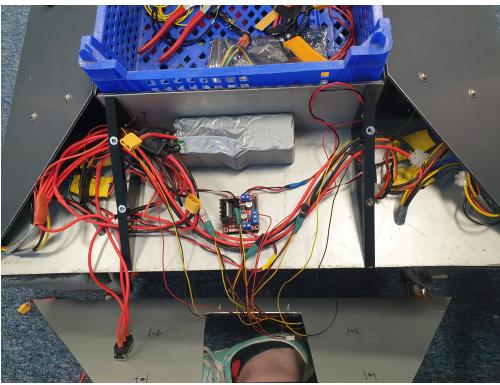


Figure 29: Removable back panel to access electronics

10.3 Conveyor Belt

The conveyor design was such that it could pivot to allow for varying angles of attack to the water along with ease of transport. The pivots, mounted centrally, were fixed to the bows using 15mm diameter copper pipe with sealed roller bearings supported in housings. One was mounted either side on the front of the deck. A central cut-out allowed the conveyor to tilt. This was to safeguard the craft in the event that it got stuck or knocked. The conveyor was held in place by elastic cord, which allowed some movement in the event of any collision, while still maintaining the conveyor in the correct place during normal operation. This cord also allowed the litter collection baskets to be easily removed and replaced during operation.

The belt was made from a Plastazote camping mat. This was cheap, readily available and gave the right amount of stretchiness, was flexible and easily cut. The belt has a surface that allowed the litter to be transported up the conveyor and released it into the hopper as the belt turned.

The belt was joined in two places using specially designed hinges, printed using PLA.

In order to ensure that bulkier waste was traversed up the conveyor, low profile brackets were installed. These brackets were nearly the same width as the belt and were bolted through the material. The brackets gave some rigidity to the belt and gave a helping hand

to moving waste up the conveyor.

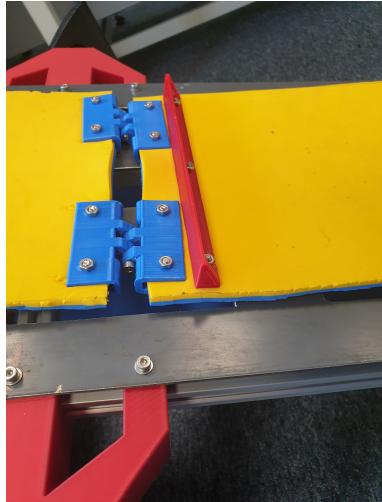


Figure 30: 3D Printed Hinges and Low Profile Brackets on Conveyor Belts

The full conveyor subsystem mounted to the craft can be seen in Figure 31.



Figure 31: Full Conveyor Subsystem Mounted to Craft

10.4 Belt Drive Motors

Drive axles were made from 15mm diameter copper pipe. The copper would not corrode in sea water, was cheap, readily available and easily cut. Standard plumbing fittings were used to connect the dual motor drive pulleys which had an 8mm inside diameter. This allowed the use of copper fittings to reduce the diameter from 15mm to 8mm. These were standard plumbing parts, cheap, readily available and easy to use. The plumbing fittings can be seen on the right hand side of Figure 32.

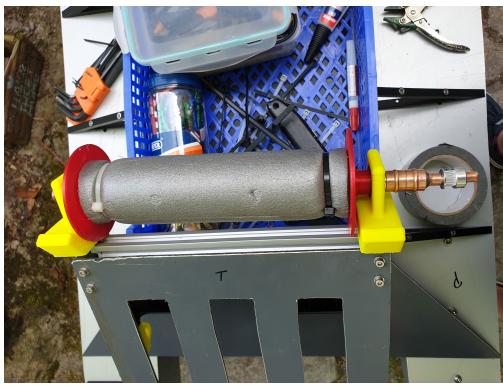


Figure 32: Full Conveyor Subsystem Mounted to Craft

The design uses two rollers, one at each end of the conveyor, with the motors driving the top roller. The rollers consisted of 15mm diameter copper pipe with 60mm outside diameter pipe insulation foam acting as the roller. The insulation was held in place using cable ties, which were located on each side and when squeezed down tight helped form the foam into a convex roller. This acted to help align the drive belt centrally. The rollers held down using cable ties can be seen in Figure 32.

The cheeks on the end of each shaft stopped the conveyor belt from wandering off the rollers. The CAD model for these parts can be seen in Figure 33.

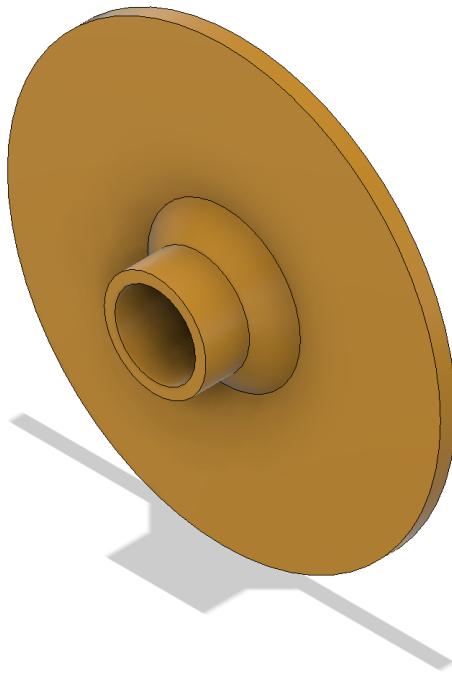


Figure 33: Full Conveyor Subsystem Mounted to Craft

The use of Aluminium section allowed the motion system to be fitted easily and adjusted to give a parallel track for the belt. Standard sealed ball bearing races were used with the 15mm copper pipe. The bearings were housed in 3D printed mounts bolted to the aluminium frame.

11 Evaluation

The finished prototype of ORCA is a successful and capable device. It has been shown to collect and store litter, which was the main objective of this project. This can be seen at this [Link](#). Upon reflection, the project ran to time with all hard deadlines being met. However, there were some setbacks and design changes, which if given more time could have led to an improved finalised execution.

The conveyor worked well; it collected litter as expected and deposited it into the collection basket. With a few tweaks to the conveyor design, made apparent upon testing,

the system as a whole could have worked almost flawlessly. One of the best ideas would be to add a brush to the underside of the conveyor belt. As the Plastazote had a high coefficient of friction when wet, litter clung to it better than anticipated. This caused issues when trying to remove the litter. A simple brush would have snagged the litter, depositing it into the litter collection basket.

The craft was a suitable size which gave it plenty of buoyancy. It floated at a good height out of the water, it was not sat too high in the water to collect water. As the litter collection basket filled up the craft would sink, but there was plenty of overhead in the design to stop the craft from sinking. Overall the craft was very well balanced in the water both in pitch and roll. This allowed the craft to sit flat in the water. The craft was stable and did not appear to be easy to capsize.

Despite the craft working well and the two boards acting as expected on their own, the RF communication was intermittent and sometimes refused to work entirely. Unfortunately, the reasoning behind this was never found. This meant that the craft was not able to be driven and tested at any range. In theory, the nRF24L01 modules should have been operating and were being controlled as specified within their datasheets. They seemed to be extremely sensitive to fluctuations in input voltage and current draws, this could have caused the reliability issues with the boards.

If this project was to be redone, more focus would be placed on making the craft and conveyor system more robust. Although it worked as expected, there are a few issues with the boards that could be improved. Time on the project would have been better spent finalising the novel part of the project instead of spending so much time on re-creating an RF system which could have been bought. This type of time management and focusing on the important aspects of the project is one of the biggest take aways from this project.

Although it worked well and the novel part of the project worked, the overall execution could have been better.

12 Further Developments

The scope of this project is wide reaching, leaving plenty of scope for further development with additional development time and funding. These features are being considered for further development during a Master's level project. Further developments can be broken down into hardware and software. The hardware sections would require more effort than the software elements. The main software elements would be establishing two way communication protocols to allow the user to better understand the current state of the craft. Hardware developments would require more investment in both time and money, but this craft is a good base line to work off to achieve this. Listed below are some obtainable developments with the biggest restrictions being time and money constraints.

Firstly, autonomous driving and mapping would allow the craft to traverse it's environment autonomously. This would be a big step in having driverless operation and patrolling. This would allow for long periods of operation without significant cost or surveillance. Autonomous mapping could be achieved using a LiDAR. The craft could firstly map out it's environment then either drive a predetermined path or in random paths. This would lead to whole coverage of the space and if this was left to run for a long period of time, would theoretically clean the surface of the water.

Additionally, a computer vision system may be implemented to track and identify rubbish in the water. The vision system could be used to asses the surface of the water to identified any potential spots of litter. The craft could then navigate to that location and use the vision system to identify what is on the surface of the litter and act accordingly. Whilst doing this, the craft could also take photos and log the most common types of litter that are being collected by the craft. This could be useful in preventing litter build up in these types of areas in the future.

Machine Learning could be implemented to learn where litter tends to collect in a given body of water. This could allow the craft to be more efficient in litter collection. The craft could travel to the locations where litter collects in larger amounts and clean these areas

first. This could be especially effective if ORCA is placed in a body of water on a semi-permanent basis.

Solar charging would allow the craft to left in the body of water for longer periods of time. The craft could run on battery power during it's typical operation and once it has finished it could be left to drift around in a low power state and use a solar panel to charge the batteries back up. This would allow the craft the be left to autonomously operate for longer periods of time. This would reduce the down time of the craft to when the collection basket needs emptying during maintenance.

Furthermore Global Positioning System (GPS) tracking could be added to allow the craft to map the paths it has taken and store location data about areas with higher amounts of litter. A GPS could also be used to add a homing feature to the craft for when the litter collection basket is full. It could return to a predetermined location that is easy to access for someone on the side of the body of water then released again once the craft has been emptied. The GPS could be used in conjunction with the other sensors to allow the real time litter status in the body of water to be drawn onto an interactive map, thus allowing the amounts of litter to be tracked in real time.

Lastly, a better litter storage container that stops the collected litter from ending back into the body of water that it has been removed from. This is important so that litter actually gets collected by the craft instead of just being pushed around. Although a simple fix, it is very important in creating a more reliable collection and storage solution.

13 Sustainability, Ethics, and Commercial Viability

13.1 Commercial Risk Evaluation

Commercially speaking, this project is not too big of a risk. A fully operation version of this product could conceivably cost approximately £1000. One of the closest commercially available products to ORCA is the wasteshark, costing about £19500 for the manual ver-

sion. Whilst the wasteshark is a more capable machine, as expected from a commercially available product, it also costs about twenty times the expected price of an ORCA.

13.2 Legal Requirements

The Laws surrounding powered craft in waterways and other similar bodies of water in the UK can be complex and unclear. Although research was done into the requirements of licenses, nothing conclusive was found. For ORCA to be deployed in an area, it would required to gain permission from the local council/body in charge of the water systems and ask for their permission. This may be subject to change depending on where in the country you are trying to operate the craft.

13.3 Intellectual Property

As far as the author is aware, there is no breach in Intellectual Property (IP). All of the software used is open source/allowed to be used within this context. The litter collection system is the biggest potential violation of IP, but as far as the author is aware, there is no conflict within this area.

13.4 Environmental Impact

ORCA aims to have a net positive environmental impact. As the primary objective of this project is to clean up water systems, the impact that this craft has is positive. Although considerations were made when undertaking this project, such as using recycled and reusable materials, inevitably the construction process will increase the carbon footprint of the project. With a long enough life span of a working product, this is hoped to be offset making the craft carbon neutral and even have a positive environmental impact. Given a more thorough analysis, it could be calculated how much litter the craft would need to collect to make the project have a net carbon impact of zero.

13.5 Ethical Conduct

Throughout this project, care was taken at every step to ensure it was being managed in the most ethical way possible. Materials for this project were re-used and recycled where possible. Scrap materials were disposed of in the most responsible way possible. Components were sourced from sustainable and slavery-free environments, where possible. Even when this was unknown, it was still a consideration when obtaining parts for this project.

13.6 Social Context

At the time of writing this report, water clean-up is one of the biggest environmental operations. It is a multinational aim, that has been gaining a lot of attention. ORCA aims to help tackle these issues head on. With the limited exposure during testing, the general public seemed to be enthusiastic about the project and what it stands for. Cleaning the oceans is a topic that a lot of people are passionate about and this was the authors way of adding to this global effort.

13.7 Product Safety and Liability

Given that ORCA would be used in open public spaces and amongst wildlife in these environments, product safety and liability is an important consideration. In the real world, ORCA will hopefully not cause any damage to the environment or anyone that it comes into contact with. If any local wildlife does happen to get caught on the conveyor and deposited into the basket, it should be able to escape out of the top. Given that it only collects debris on top of the water, it should not pick up any fish or other marine life. The propellers would be covered up so that they do not get caught in anything suspended in the water or damage local wildlife in the water. This should also make the craft safe to handle when not in the water.

13.8 Internationalisation

This biggest challenge to internationalisation would be potentially restrictive and unknown waterway laws and restrictions. Differing countries may have differing laws and regulations about what is allowed in waterways; this may effect the ability to obtain ORCA licensing and usage ORCA.

14 Conclusion

This project has been an attempt at helping in an ever increasing environmental issue that affects all life on earth. At the start of the project, the effectiveness of a conveyor based system to collect litter from the surface of water of a small scale was not thoroughly explored.

Project ORCA has shown that it is possible to collect litter using a conveyor belt in a small unmanned craft. This has also been shown to be a highly effective method of collecting litter.

Additionally, this project has been good to show how far IDEs and CAD software has progressed. The Mbed IDE was exclusively used for this project. Setting up a workspace was easy and it took minutes to get into programming for the Mbed family of devices.

This project has provided a great experience and a new method of litter collection at a small scale using an adaptable platform. It has allowed exploration of PCB creation, RF communications, BLDCM control, whilst also allowing hands off manufacturing. The decision to use custom electronics took over the project, which was a shame given the main aim of the project; to create a craft that was capable of collecting litter.

The scope of ORCA was broad and covered many engineering disciplines. For a larger project with a bigger team this would not have been an issue. As a final year university project, the scope could have been narrowed to focus more on either the hardware side or software side. Given the novelty of this project came from the use of a conveyor belt, it

would have been a better use of time to focus on perfecting the design and operation of this before trying to create a custom control solution.

Despite this, the final prototype of ORCA was successful in collecting litter. The proof of concept has been shown and the viability of this solution has been made apparent. This report contributes to the first milestones in the development in low cost litter collection craft for use in inland water reserves and reservoirs.

References

- [1] H. A. Leslie, M. J. van Velzen, S. H. Brandsma, A. D. Vethaak, J. J. Garcia-Vallejo, and M. H. Lamoree, “Discovery and quantification of plastic particle pollution in human blood,” *Environment International*, vol. 163, p. 107199, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0160412022001258>
- [2] K. Blackburn and D. Green, “The potential effects of microplastics on human health: What is known and what is unknown,” *Ambio*, vol. 51, pp. 1–13, 06 2021.
- [3] S. Sharma, D. Sarkar, and D. Gupta, “Agile processes and methodologies: A conceptual study,” *International Journal on Computer Science and Engineering*, vol. 4, 05 2012.
- [4] STMicroelectronics. (2018, May). [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32l432kc.pdf>
- [5] ——. (2019, Nov). [Online]. Available: <https://html.alldatasheet.com/html-pdf/1211922/STMICROELECTRONICS/NUCLEO-F429ZI/2273/1/NUCLEO-F429ZI.html>
- [6] N. Semiconductor. (2006, Mar). [Online]. Available: https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf
- [7] P. Millett. (2021) Brushless vs brushed dc motors: When and why to choose one over the other. [Online]. Available: <https://www.monolithicpower.com/en/brushless-vs-brushed-dc-motors>
- [8] Dunkermotoren. (2010, Feb). [Online]. Available: <http://www.nrc.com.tw/CAT/Dunkermotoren/PLG.SG/PLG24.pdf>
- [9] H. Technology. (2013, Jan). [Online]. Available: https://components101.com/sites/default/files/component_datasheet/L298N-Motor-Driver-Datasheet.pdf

- [10] S. E. Group. (2015, Jun). [Online]. Available: https://voltaplex.com/media/whitepapers/specification-sheet/Sony_VTC6_Specification_Sheet2.pdf
- [11] S. TechnoLabs. (2008, Apr). [Online]. Available: https://components101.com/sites/default/files/component_datasheet/Datasheet%20of%20IR%20Sensor.pdf
- [12] O. Edwards. (2011, Jan) Brushless vs brushed dc motors: When and why to choose one over the other. [Online]. Available: <https://os.mbed.com/users/Owen/code/nRF24L01P/>
- [13] W. Zhang, W. Wang, N. Ding, and C. Li, "The main-loop and servo-loop structure control strategy of esc," *Chongqing Daxue Xuebao/Journal of Chongqing University*, vol. 35, pp. 19–24, 07 2012.
- [14] A. Composites. (2010, Nov). [Online]. Available: <https://www.tecnoplastsas.eu/img/cms/PDF%20PANNELLI%20COMPOSITI/Scheda-tecnica-dibond.pdf>
- [15] ZoteFoams. (2004, Jan). [Online]. Available: <https://www.interpak.ltd.uk/Downloads/Plastazote%20LD45%20Spec%20Sheet.pdf>

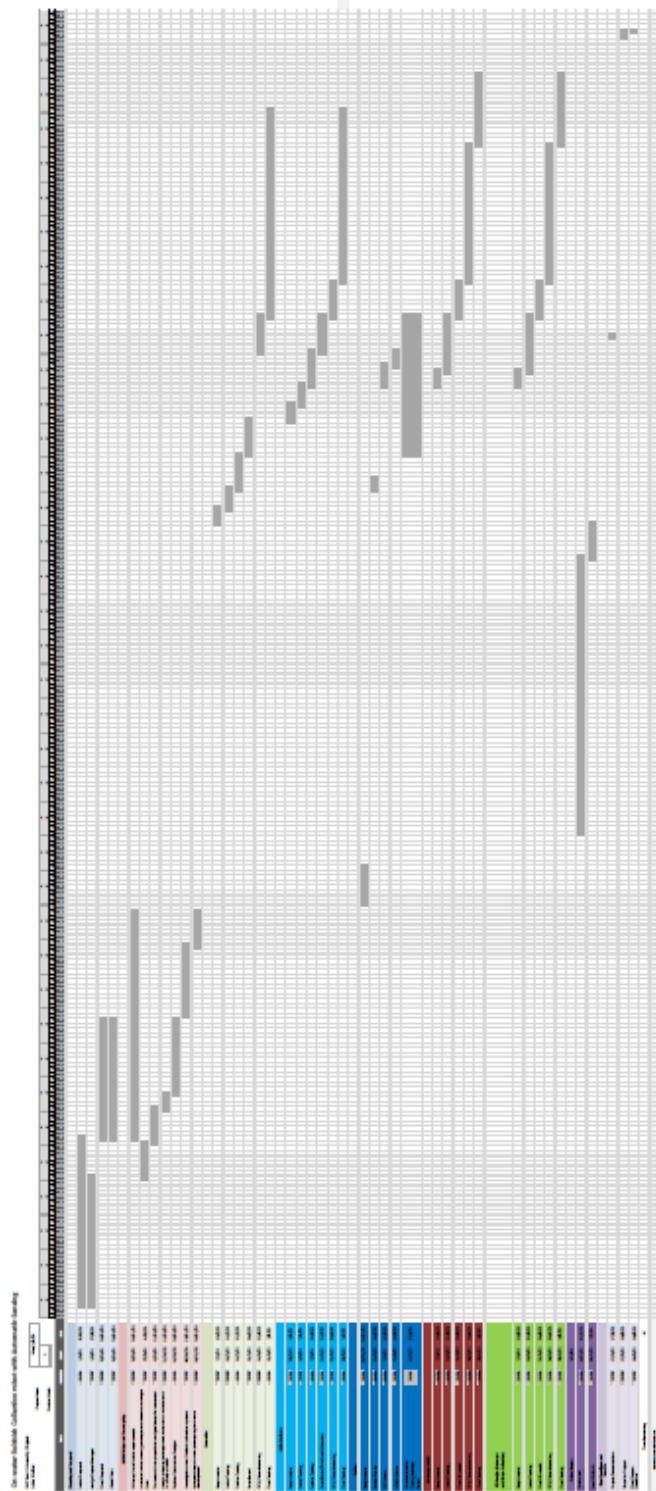
A Design Requirements Table

Ref	Requirement	Required?
1	Main body of the craft	Yes
2	Suitable Propulsion Method	Yes
3	Collection System for the litter	Yes
4	Custom remote controller and receiver	No - highly desirable
5	Battery powered	Yes
6	Suitably Waterproofed	Yes

B Software Design Requirements Table

Ref	Requirement	Required?
1	Suitable control system for movement of the craft	Yes
2	Suitable control system for collecting litter	Yes
3	Sense when rubbish collection basket is full and alert the user	Yes
4	Sense when the batteries (on either craft or remote) is running low and alert the user	Yes
5	Suitable communication system between the remote and receiver	Yes
6	Relay information from the receiver to the controller about battery voltage and time of current usage, etc.	No - nice to have
7	System to track amount of rubbish collected	No - nice to have
8	Calculations of carbon neutrality of the collected litter, how much litter would need to be collected to make the craft environmentally viable	No - nice to have

C Full Sized Gantt Chart



D Gantt Chart Phase 1

Task	Start Date	End Date
Initial Proposal	1/9/21	5/10/21
Assign Project Supervisor	1/9/21	27/9/21
Final Proposal	5/10/21	29/10/21
Gantt Chart	5/10/21	29/10/21

E Gantt Chart Phase 2

Task	Start Date	End Date
Write up list of initial requirements	5/10/21	20/11/21
Create an initial floating prototype and measure its weight limit	27/9/21	4/10/21
Do research into propulsion and gearboxes for movement	4/10/21	11/10/21
Design a moving prototype that floats and is controllable in a body of water	11/10/21	14/10/21
Explore Catamaran Designs	14/10/21	29/10/21
Investigation into Rubbish collection systems	30/10/21	13/11/21
Evaluate analysis from prototyping and initial development	13/11/21	20/11/21

F Gantt Chart Phase 3

Task	Start Date	End Date
Controller		
Requirements	7/2/22	10/2/22
Initial Testing	10/2/22	14/2/22
Further Testing	14/2/22	21/2/22
Breadboard	21/2/22	28/2/22
PCB / Manufacturing	14/3/22	21/3/22
Final Testing	21/3/22	2/5/22
Main Reciever		
Requirements	28/2/22	3/3/22
Initial Testing	3/3/22	7/3/22
Further Testing	7/3/22	14/3/22
Breadboard	14/3/22	21/3/22
PCB / Manufacturing	21/3/22	28/3/22
Final Testing	28/3/22	2/5/22
Coding		
Requirements	22/11/21	29/11/21
Buzzer Library	14/2/22	16/2/22
ESC Library	7/3/22	11/3/22
L293D Library	11/3/22	14/3/22
Communication Testing Between Boards	21/2/22	21/3/22
3d Design (Hull)		
Requirements	7/3/22	10/3/22
Initial Testing	10/3/22	21/3/22
Final 3D model	21/3/22	28/3/22
PCB / Manufacturing	28/3/22	25/4/22
Final Testing	25/4/22	9/5/22
3d Design (Conveyor and Litter Collection)		
Requirements	7/3/22	10/3/22
Initial Testing	10/3/22	21/3/22
Final 3D model	21/3/22	28/3/22
PCB / Manufacturing	28/3/22	25/4/22
Final Testing	25/4/22	9/5/22

G Transmission Codes

The 1 bit denotes the type of device.

The 2 bit denotes the component in that device type.

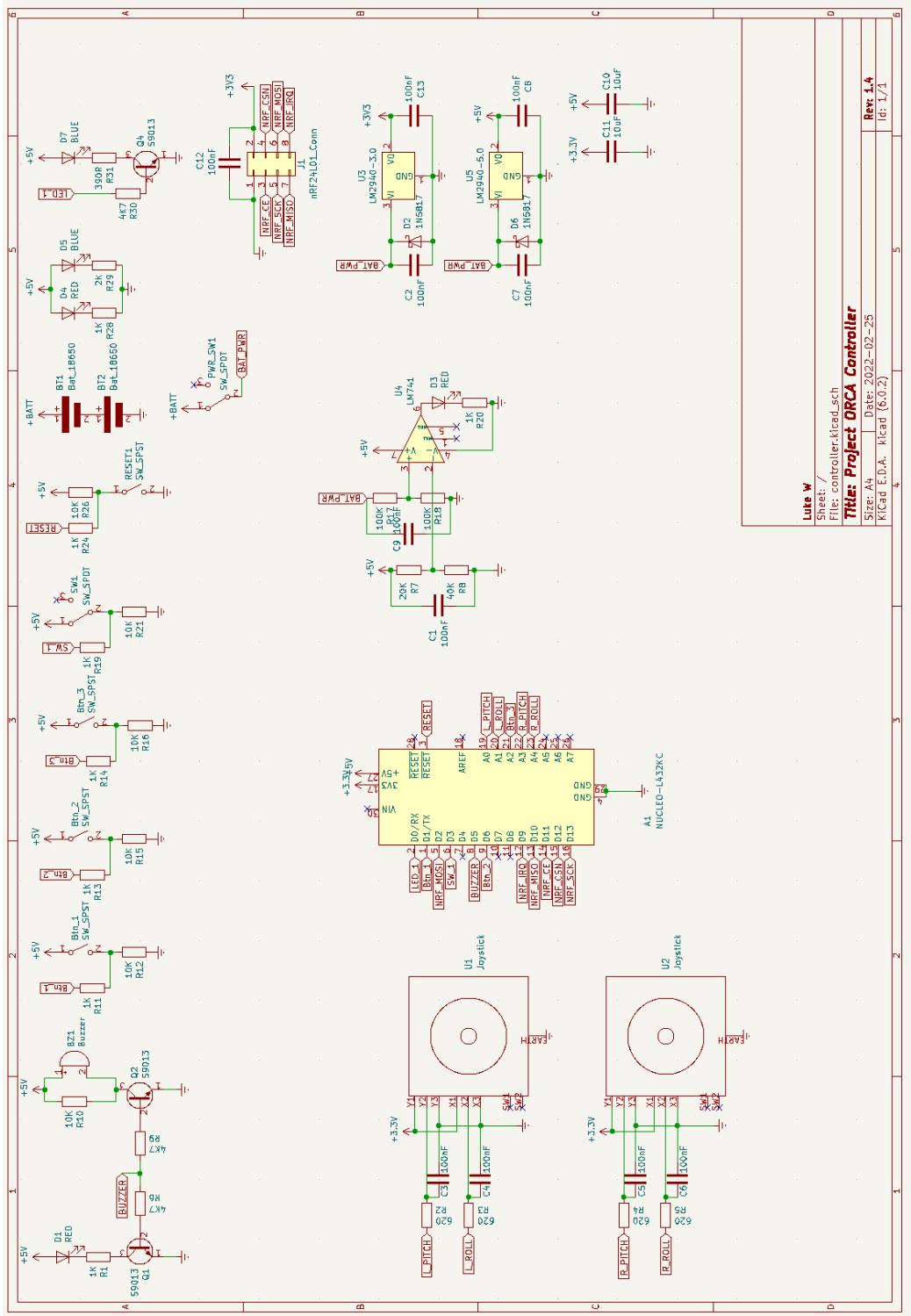
The 3 and 4 bits denote the specific device instruction.

- 1xxx - POTs
 - x1xx - Left Pitch
 - x2xx - Left Roll
 - x3xx - Right Pitch
 - x4xx - Right Roll
 - * xx1x - Forwards
 - * xx0x - Reverse
 - xxx0 - Value of throttle
- 2xxx - Buttons
 - x1xx - Button 1
 - x2xx - Button 2
 - x3xx - Button 3
- 3xxx - Switches
 - xxx1 - Switch On
 - xxx0 - Switch Off

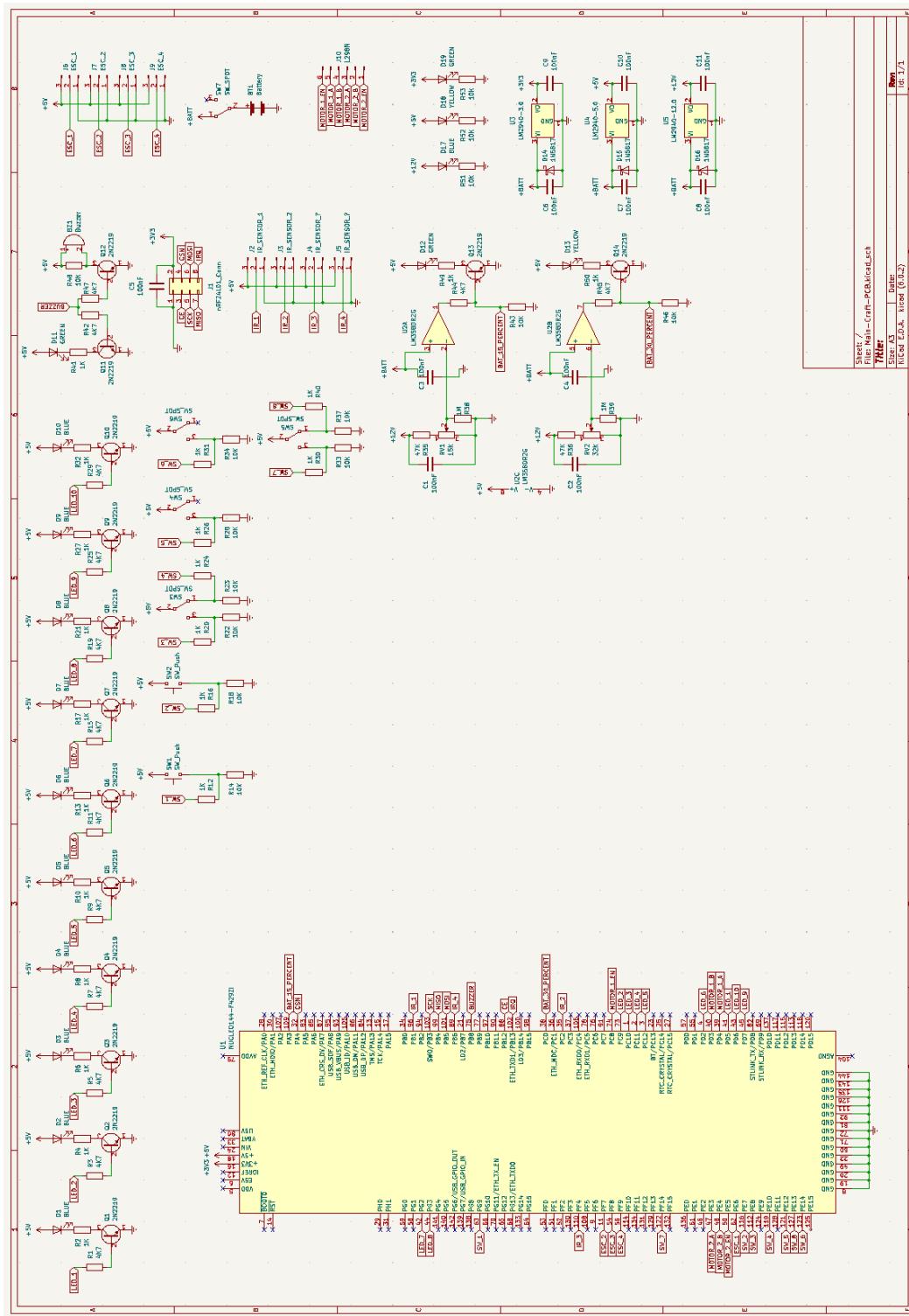
H Bill of Materials

Part no.	Qty	Name	Product Description	Unit Price	Total
Farnell					
3588829	2	5V regulator	5V SMT SOT-223 Linear Regulator	0.34	0.68
2332042	20	1M Resistor	1M 0603 SMD Resistor	0.01	0.2
1759122	20	SMD Capacitor	0.1uF 0603 SMT Capacitor	0.03	0.6
3796304	20	SMT Green LED	0603 SMT Green LED	0.07	1.4
3796307	20	SMT Yellow LED	0603 SMD Yellow LED	0.09	1.8
3796315	20	SMT Blue LED	0603 SMD Blue LED	0.09	1.8
1217269	40	SMD Transistors	SMD NPN Transistors	0.11	4.4
1520617	5	SMD Potentiometers	100K SMD Rotary Potentiometers	0.18	0.9
2861870	30	SMD Resistor	10K 0603 SMD Resistor	0.01	0.3
9238670	30	SMD Resistor	39K 0603 SMD Resistor	0.01	0.3
1692517	30	SMD Resistor	100K 0603 SMD Resistor	0.03	0.9
2078919	30	SMD Resistor	47K 0603 SMD Resistor	0.05	1.5
9233466	30	SMD Resistor	4K7 0603 SMD Resistor	0.01	0.3
9233423	30	SMD Resistor	2K2 0603 SMD Resistor	0.01	0.3
3495891	30	SMD Resistor	620R 0603 SMD Resistor	0.01	0.3
2078907	30	SMD Resistor	1K 0603 SMD Resistor	0.05	1.5
DigiKey					
497-16592-ND	2	Nucleo Board	Nucleo-L432KC Board	10.81	21.68
1568-COM-16273-ND	4	Joysticks	2 Axis Joystick	3.88	15.52
Amazon					
Link	1	DC Motor Driver	L298N DC motor Driver	2.60	2.60
Link	2	IR Sensors	Infra-red Obstacle Avoidance Sensor	0.85	1.70
Link	2	GTS Belt Kit	GT2 Timing Pulley with 60T gear and 20T gear	5.14	10.28
Mouser					
511-LDL1117S30R	2	3V Regulator	5V SMT SOT-223 Linear Regulator	0.34	0.68
511-L79L12ACU-TR	1	12 V Regulator	5V SMT SOT-223 Linear Regulator	0.47	0.47
Ooznest					
VSL0T-2020-CTS-C	2	V-Slot Linear Rail	V-Slot Linear Rail - 900mm	6.48	12.96
VSL0T-2020-CTS-C	3	V-Slot Linear Rail	V-Slot Linear Rail - 400mm	2.40	7.2

I Controller PCB Schematic



J Receiver PCB Schematic



K Back Panel Technical Drawing 1:5 scale

