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

Luke Waller

Wednesday 18th May, 2022

Abstract

Oceans make up over 70% of the worlds 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 (O.R.C.A). Several solutions to this problem have been created; however, the O.R.C.A based solution aims at a novel application that has previously been left unexplored. O.R.C.A 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 to undertake the role of mentor during the start of the Autumn Term.

The author would also like to acknowledge 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.

This is a test bib [1]

Contents

B Software Design Requirements Table **13**

List of Figures

List of Tables

1   Glossary

2   Introduction

3   Design Requirements

The design requirements for this project as outlined in Appendix A and Appendix B were agreed upon by SOMEONE - NOT SURE HOW TO PHRASE IAN AND ME. The table acts a checklist for the project to ensure that the met all of these requirements, this was a dynamic list that was adapted and amended as the project progressed. The core requirements of the project remained the same but some of the higher level attainable aims were adapted as the project progressed.

4   Aims and Objectives

The aims and objectives, which have been listed below, are used as an indicator of success of the project. 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. I will need to create a suitable propulsion system, steering system, and make sure the craft is stable both at top speed and when cornering. 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 collect 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 radio controller that I have 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 would be these combined 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.

5 Project Managment

The project has 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, phase three, 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 Ltds. agile framework. This framework follows and agile approach to project development [1], which is evident in the adaptability of the project as it has developed.

The utilised style used to manage the project 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 **??**, Appendix **??**, and Appendix **??** respectively.

5.1   Phase 1 - Autumn Term 2021/22

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

From this initial planning phase, a design brief, Gantt Chart, and future steps were created and agreed upon with the project supervisior. 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 section **??** and section **??**.

5.2   Phase 2 - Winter Term 2021/22

Phase 2 involved undertaking initial prototyping and light development of the work 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. During Phase 2, initial designs were created, tested, and evaluated.

The second phase helped focus the design requirements for the project, whilst evaluating what would be possible within the given time scaled budget. The use of threads, learned from ELEC351, along with ..., drove the development of my project from an Arduino based solution to an ARM based solution, as this enabled the use of RTOS functionality.

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

5.3   Phase 3 - Spring/Summer Term 2022

6   Research

6.1   Concept Designs - Controller and Receiver

6.1.1   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 we would be required.

The lack 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 concept, unless this proved to be impractical.
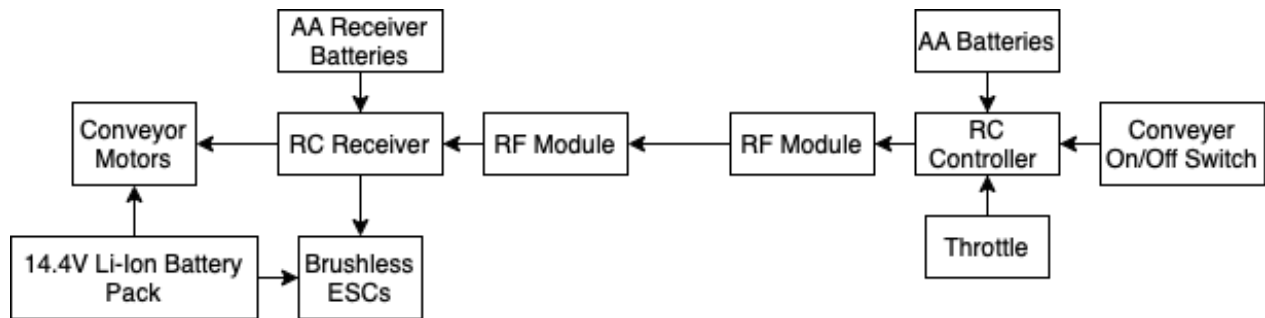

Figure 1: Concept 1 - Off the shelf RC Controller and Receiver

6.1.2   Concept 2 - Custom RF Based Solution

Concept 2 introduces "smart" features such as custom battery management circuity 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 for another use-case. The main craft could be changed and the controller system would be easily reusable. Using a microcontroller based system add the ability to add and remove devices as the requirements of the craft change.

The downsides of the system are... Fiddly AF, Hard to GET RC working, Required Knowledge of Microcontrollers to utilise full functionality.
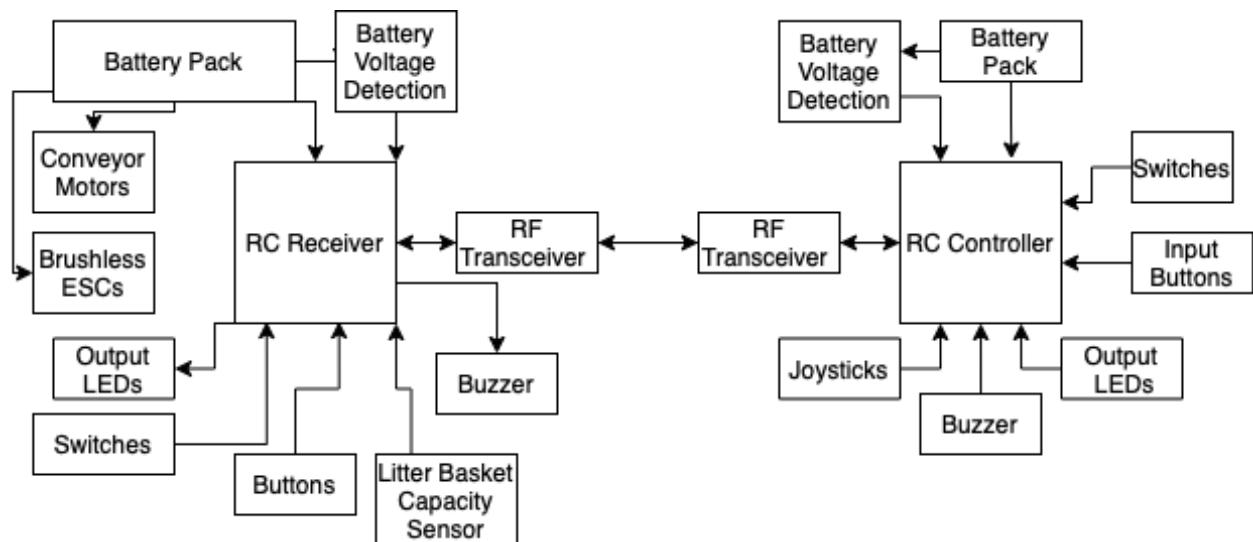

Figure 2: Concept 2 - Custom RC Controller and Receiver

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 correct for this purpose.

7 Optioneering

7.1 Controller Optioneering

The STM32L432 was selected as the processed used for the controller device for this project. It was decided upon because it was lightweight and powerful enough for it's indented use case.

Cost was also important consideration when selecting suitable prototypes. This weighting was important to make sure that the project budget was used in the most effecitve way possible. Given that the Mbed OS IDE was free to use software the cost of development was not needed to be taken into account.

The 2 largest competing manufactures within the microcontroller industry, ARM and AVR were included when making a choice about which proccessor to use. This allowed a full assemenet 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 *STMicroelectrionics* boards. One advantage of using the the ATMega328P was its ease of use and setup given the large number of libraries avaliable and popularity within the hobbiest 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 desicion making process was the ongoing silicon shortage and supply chain issues. This reduced the number of avaliable 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.

7.2 Receiver Optioneering

The STM32F492 was selected for use within the reciever 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.

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 signicifantly higher than that of the board.

Like with the Controller Optioneering, AVR's ATMega2560 was also considered for use within the reciever. Once again the the RTOS functionality of the STMicroelection 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.

7.3 Peripheral Optioneering

7.3.1 RF Transciever

In order to deliver the functionality shown in Concept 2 [6.1.2] , an RF module would need to be sourced to allow for control of the craft at range. The nRF24L01 is used widley within embedded systems as a long range transciever device. 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 packge 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 SPI with an added chip enabled pin, 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 libray would be used, as it would not detreact from the project. This library was written in Mbed OS so it did not require porting over from another architecture.

7.3.2   Brushless DC Motors

7.3.3   Conveyor Motors

7.3.4   Batteries

7.3.5   IR Distance Sensors

7.4   Finalised Solution

7.4.1   Block Diagram

7.4.2   Bill of Materials

References

[1] 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.

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 nuetrality of the collected litter, how much litter would need to be collected to make the craft environmentally viable	No - nice to have