

Monitoring and Control of a Marine Ecosystem in an Aquarium

Bachelor of Engineering Honours

Industrial Computer Systems

Joshua John McCann

2015



Murdoch
UNIVERSITY

This page has been intentionally left unmarked.

Declaration

I, Joshua John McCann, declare that all works presented in this thesis aside from cited works and appendices are the result of my own work and findings.

Acknowledgements

The first acknowledgment I would like to make is to my supervisor Associate Professor Graeme Cole, whose guidance not only throughout this thesis but also through the stresses of third and fourth year university helped me develop my engineering skills to a point at which I feel confident to take on the work force as an engineer. Additionally Graeme's wealth of knowledge and willingness to share it has been an inspiration. This has led to my decision to undertake further studies upon gaining some hands on experience.

Secondly I would like to thank my dad, Eric McCann, who heavily assisted me in the construction of both the stand and the lighting hood. Without his assistance the project would not have progressed as well as it did.

I would also like to thank Mathew Wilson for reading through my entire thesis for spelling, grammar and punctuation errors. It is thanks to his proof reading skills that this thesis is at the quality it is.

Finally, and most importantly, I would like to thank my family and partner who have continuously supported me through my university degree. The patience and understanding in which they have given me is greatly appreciated.

Abstract

Research has shown that marine ecosystems are not only complex but also extremely difficult to replicate. Reef aquarists aim to replicate these marine ecosystems within aquariums in their homes, for the aesthetic beauty of such coral reef tanks. To assist aquarists in the task of maintaining environmental parameters in their tanks, a number of control and monitoring systems have been created through the advancement of embedded systems.

The project, that this thesis is based on, involves the development of a control and monitoring system for a marine ecosystem sustained within an aquarium. The aim for the system was to control and monitor water parameters not yet considered by existing systems, in turn improving the stability of the ecosystem and reducing maintenance of the aquarium.

As a part of this report, significant research was undertaken to assess existing systems and further understand the important water parameters necessary for marine ecosystems.

The main focus of this thesis was to outline the approach, design and implementation of the control and monitoring system. This includes details on the systems selected features including:

- Control and monitoring of pH, Dissolved Oxygen, Temperature and Salinity.
- Control of realistic lighting.
- Control and monitoring of system parameters, water levels, pumps and other equipment.
- HMI interface for easy monitoring.

To facilitate these features a PiLeven microcontroller has been programmed as a master device and performs all control and monitoring functions along with the communication to slave sensors. Additionally a Raspberry Pi microprocessor was utilised as the computer interface enabling the programming of the microcontroller and HMI for monitoring. At the completion of this thesis all aspects of the control and monitoring system are considered complete aside from

the control of the pH and Salinity. Despite this the control and monitoring system was effectively implemented and is ready to house a marine ecosystem.

Contents

Declaration	II
Acknowledgements.....	III
Abstract	IV
Contents	VI
List of Figures	X
List of Tables.....	XII
List of Abbreviations.....	XIII
Chapter 1 : Introduction.....	1
1.1 Project Introduction	1
1.2 Document Structure.....	2
Chapter 2 : Project Overview	3
2.1 Overview	3
2.2 Existing Systems	3
2.3 Project Objectives	4
2.4 Project Layout	5
Chapter 3 : Hardware.....	8
3.1 Overview	8
3.2 Components	8
3.2.1 PiLeven Microcontroller	8
3.2.2 Raspberry Pi Microprocessor	10
3.2.3 pH sensor and embedded circuit	11
3.2.4 Dissolved oxygen sensor and embedded circuit	13
3.2.5 Electrical Conductivity sensor and embedded circuit	14
3.2.5 Additional Minor Components.....	15
3.3 Power Supplies and Converters	23
3.3.1 240 V AC to 12V DC Switch Mode Power Supply	24
3.3.2 12V DC to 5V DC Buck Converter	24
3.3.3 12V DC Boost Converters	25
Chapter 4 : Software Applications, Programming Languages and Protocols	27
4.1 Overview	27
4.2 Software Applications	27
4.2.1 Arduino IDE	27

4.2.2	Raspberry Pi Terminal	28
4.2.3	Notetab	29
4.3	Programming Languages	29
4.3.1	Arduino	30
4.3.2	C/C++	30
4.3.3	HTML	30
4.3.4	CSS	30
4.4	Protocols	31
4.4.1	I2C	31
4.4.2	One-Wire	32
Chapter 5	: Lighting	33
5.1	Overview	33
5.2	LED Selection	33
5.3	LED Circuit Design	39
5.4	Control	41
5.4.1	Control Hardware	41
5.4.2	Program Overview	42
5.5	Testing	45
Chapter 6	: Water Parameter Monitoring and Control	47
6.1	Overview	47
6.2	Temperature	47
6.2.1	Monitoring	47
6.2.2	Control	48
6.3	pH	50
6.3.1	Monitoring	50
6.3.2	Control	53
6.4	Dissolved Oxygen	54
6.4.1	Monitoring	54
6.4.2	Control	56
6.5	Salinity	58
6.5.1	Monitoring	58
6.5.2	Control	60
Chapter 7	: System Monitoring and Control	61
7.1	Overview	61

7.2	Water Levels and Pumps.....	61
7.2.1	Main Tank Monitoring.....	61
7.2.2	Sump Tank Monitoring.....	62
7.2.3	Programming and Testing	63
7.3	Cooling.....	63
Chapter 8	: Network and Communications.....	65
8.1	Overview	65
8.2	Network Overview	65
8.3	I2C Network.....	65
8.4	Serial Communications.....	66
8.5	Raspberry Pi Constituents	66
8.5.1	Server/Client	66
8.5.2	Webpages.....	67
Chapter 9	: Future Works.....	69
9.1	Overview	69
9.2	Uncompleted Works	69
9.2.1	pH and Salinity Control.....	69
9.2.2	Microcontroller to Server Communication	69
9.3	Suggested Improvements and Future Development	70
Chapter 10	: Conclusion	72
Chapter 11	: References.....	73
Appendix 1	: Marine Aquariums and Reef Keeping	78
A1.1	Importance of pH	78
A1.2	Importance of Dissolved Oxygen	78
A1.2	Importance of Salinity.....	78
A1.3	Importance of Water Temperature	79
A1.4	Flow Rates and Pumps and Sumps	79
A1.4.1	Sumps.....	79
A1.4.2	Flow Rates and Pumps	80
A1.5	Importance of Lighting for Coral.....	80
Appendix 2	: Project Build	81
A2.1	Aquarium Stand	81
A2.2	Lighting Hood	82
A2.3	Sump	83

A2.4 Electrical Draw	84
A2.5 Plumbing	85
Appendix 3 : Calculations	89
A3.1 Aquarium Heaters	89
A3.2 Power and Current Draw	89
A3.2.1 12V System.....	89
A3.2.2 5V Devices	90
A3.3 LED Number Calculator for Luminous Flux	90
Appendix 4 : Tests	91
A4.1 pH Home Test	91
A4.2 Salinity Home Test	91
A4.3 I2C Test Results	93

List of Figures

Figure 1-1 – Document Structure.....	2
Figure 2-1 – General Component Layout of Aquarium.....	6
Figure 2-2 - Main Tank and Sump Tank Water Cycle.	7
Figure 2-3 – Control and Monitoring Overview.	7
Figure 3-1 – PiLeven Microcontroller.....	8
Figure 3-2 – Raspberry Pi Model 2 B (Multicherry, 2015).....	11
Figure 3-3 – pH Probe.....	12
Figure 3-4 - pH EZO Circuit.	12
Figure 3-5 – DO Probe.	14
Figure 3-6- DO EZO Circuit.....	14
Figure 3-7 – EC Probe.	15
Figure 3-8 – EC EZO Circuit.....	15
Figure 3-9 – 8 Channel Relay Module.	16
Figure 3-10 – Ultrasonic Ranging Module HC-SR04 (Dilmen, 2014).	17
Figure 3-11 – Peristaltic Pump Operation (Njmcca, 2013).....	19
Figure 3-12 – Submersible Power Head.....	21
Figure 3-13 – IRF540N MOSFETS.....	22
Figure 3-14 – Example MOSFET Circuit to Control a Lamp.	22
Figure 3-15 – CREE XPE LED.....	23
Figure 3-16 – Buck Converter.....	25
Figure 3-17- Boost Converter.	26
Figure 4-1 – Arduino IDE (Messer, 2012).	28
Figure 5-1 – Chlorophyll ‘a’ & ‘b’ Spectra, Wavelengths vs. Absorbance (Aushulz, 2012).	34
Figure 5-2 – Spectral analysis of the sun to show the highest intensity wavelengths found at sea level (Danmichaelo, 2012).....	35
Figure 5-3 - Penetration of Light in Seawater, adapted from (NASA, 2012).....	35
Figure 5-4 – LED Arrangement.	37
Figure 5-5 - Mounted and Wired LEDs.....	38
Figure 5-6 – Boost Converter, Highlighted Are The Potentiometers for Current and Voltage...	40
Figure 5-7 – LED Circuit Wiring Diagram.	41
Figure 5-8 - Microcontroller to MOSFET Circuit.....	42
Figure 6-1 – DS18B20 Temperature Sensor Circuit Diagram For One-Wire Bus.....	48
Figure 6-2 – Circuit Diagram for Controlled Heater Outlets.	49
Figure 6-3 Pseudocode for Heater Control.	49
Figure 6-4 – pH Reading Function.	52
Figure 6-5 – Overview of pH Control.....	53
Figure 6-6 – pH Dosage Pump (left), Fresh Water Dosage Pump (right).	54
Figure 6-7 – String Pars Function, Separates CSV String.	56
Figure 7-1 – Ultrasonic Sensor Over Main Tank.....	61
Figure 7-2 – Return Pump Controlled 240V Outlet.....	62
Figure 7-3 – Ultrasonic Sensor Over Sump Tank.....	63
Figure 7-4 – Controlled Exhaust Fans at the Top of Lighting Hood.....	64
Figure 7-5 – Exhaust Fans for Cooling the Electrical Draw.....	64
Figure 8-1 – Network Overview.	65

Figure 8-2 - Home Page.....	68
Figure 8-3 - System Overview Page.....	68
Figure 8-4 - Water Parameters Page.....	68
Figure 8-5 - Lighting Page.....	68

List of Tables

Table 2.1 - Checklist of Water Parameter Control and Monitoring for some Existing Systems. ...	3
Table 2.2 - Checklist of System Control and Monitoring for Some Existing Systems.....	4
Table 3.1 – Microcontroller Checklist PiLeven Comparison.....	9
Table 3.2 – PiLeven Specifications (freetronics, 2015).....	9
Table 3.3 – Raspberry Pi Model 2 B Specifications (Raspberry Pi, 2015).....	11
Table 3.4 – pH and DO EZO Circuit Specifications (Atlas Scientific).	13
Table 3.5 – EC EZO Circuit Specifications (Atlas Scientific).	15
Table 3.6 – 8 Channel Relay Module Specifications (SainSMART).	16
Table 3.7 – Ultrasonic Ranging Module HC-SR04 Specifications (ELEC Freaks).	17
Table 3.8 – DS18B20 Specifications (Maxim Integrated, 2008).	18
Table 3.9 – RTC Specifications (Maxim/Dallas Semiconductor, 2008).....	18
Table 3.10 – Peristaltic pump specifications.	19
Table 3.11 – Submersible Pump Specifications.....	20
Table 3.12 – LED Specifications (CREE, 2015).	23
Table 3.13 – Inverter Specifications.	24
Table 3.14 – Buck Converter Specifications.	25
Table 3.15 – Boost Converter Specifications.....	26
Table 4.1 – Linux Commands (Raspberry Pi).	29
Table 5.1 – LED specifications and Wavelength Match Ups.	36
Table 5.2 – Numbers of Each LED Type.....	37
Table 5.3 – LED Circuit Specifications.....	40
Table 5.4 – Duty Cycle Values for each LED Circuit Associated With Day Light Recorded Every 10 Minutes to Show Sunrise (Note date was set to January in RTC until later changed).....	44
Table 5.5 – Duty Cycle Values for Different Moon Phases.....	45
Table 6.1 – I2C Mode Commands Utilised.	51
Table 6.2 – pH Test Results, Comparison Between Home Tester and pH Probe.....	53
Table 6.3 – Specific Gravity Test Results, Comparison Between Home Tester and EC Probe.	59
Table 9.1 – Suggested Improvements and Future Developments.....	70

List of Abbreviations

AC	Alternating Current
ADC	Analogue to Digital Converter
DC	Direct Current
DO	Dissolved Oxygen
EC	Electrical Conductivity
HDMI	High-Definition Multimedia Interface
HMI	Human Machine Interface
HTML	Hypertext Mark-up Language
Hz	Hertz
IC	Integrated Circuit
I/O	Input or Output
LAMP	Linux, Apache, MySQL, PHP (Server Architecture)
LED	Light Emitting Diode
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
pH	Power of Hydrogen
PWM	Pulse Width Modulation
RDBMS	Rational Database Management System
RPi	Raspberry Pi

This page has been intentionally left unmarked.

Chapter 1 : Introduction

1.1 Project Introduction

Marine coral reef ecosystems are one of the most visually pleasing ecosystems on the planet, with their diversity in marine life, abundance of colour and sheer beauty it is no surprise that many people desire to recreate these ecosystems within their homes. More commonly known as reef aquariums, these setups require a large amount of attention in order to maintain the environment needed to sustain marine life (Cliff, 2013). With the advancement of technology, maintaining the necessary environmental parameters of a reef aquarium has become easier and more accurate. Using embedded systems, engineers and aquarists have begun developing a range of control and monitoring systems, which are capable of providing different levels of monitoring and control over various parameters. Based on the research undertaken of numerous existing commercial aquarium controllers it was found that the monitoring and control of certain water parameters was lacking. Moreover, none of these commercial systems provided the ability to replicate a reef ecosystem, but rather aimed to control parameters purely on fish and coral requirements. A number of water parameters are important for a healthy reef ecosystem and hence reef aquarium, these include: temperature, lighting, pH, salinity, dissolved oxygen, calcium, alkalinity and magnesium (Ulrich III, 2013). Currently existing systems only provide monitoring of temperature, pH and salinity and control of lighting and temperature. This thesis details the design and implementation of a monitoring and control system that aims to provide increased control and monitoring of water parameters not yet considered by current commercial models. A system that provides increased control and monitoring over the water parameters necessary for reef ecosystems would not only improve the living conditions of marine life but would also aid in marine research as specific water parameters could be altered and monitored accurately to investigate the effects on marine life.

1.2 Document Structure

The structure of this document can be seen in Figure 1-1, with the document being separated into five major categories. The first category, containing Chapter 1, is the Introduction which introduces the main ideas of the thesis, provides a brief background along with a validation of the project's aims. This is followed by Project Objectives, contained in Chapter 2, which compares existing systems, defines the aims and objectives of the thesis. The Hardware and Software category, Chapter 3 and Chapter 4, provide details and information on the components and software utilised throughout the thesis. The main category of this document is the Project Implementation which details the approach, implementation and testing for all major elements of the development work; this is spread across Chapters 5 to 8. Chapter 5 describes the lighting, Chapter 6 details all water parameters, Chapter 7 is concerned with system monitoring and control and Chapter 8 details the network and communications. Following the Project Implementation is the Future Works, in Chapter 9, which indicates to the future development and work suggested for the thesis. Finally the Conclusion category, within Chapter 10, evaluates the thesis based on the results and objectives set.

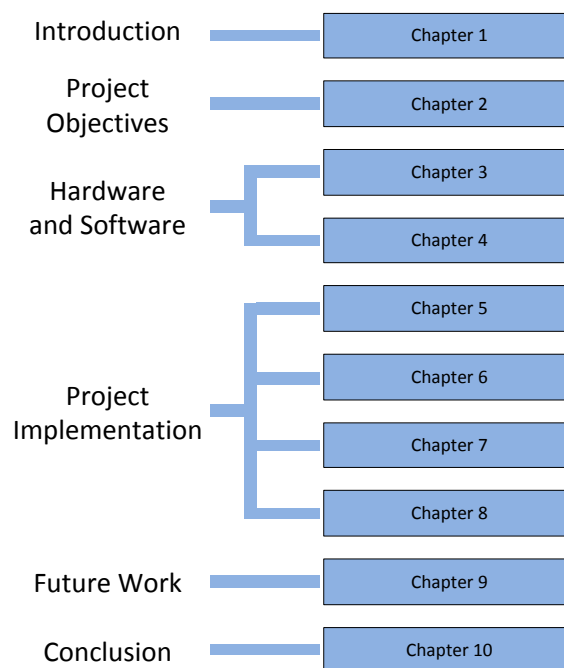


Figure 1-1 – Document Structure.

Chapter 2 : Project Overview

2.1 Overview

This chapter provides an in-depth breakdown of the project objectives for this thesis in Section 2.3 and design layout and overview for the project in Section 2.4 to aid in visualisation.

2.2 Existing Systems

Before defining the objectives of the project, research was undertaken to thoroughly compare a range of existing commercial systems available for monitoring and controlling reef aquariums. In comparing these systems a checklist was created for three main areas: the monitoring (M) of selected water parameters, the control (C) of selected water parameters and the system monitoring and control. The checklist in Table 2.1 shows the control and monitoring aspects available in regards to water parameters for each of the researched commercial systems. This information was used in deciding the additional water parameter monitoring and control required in this project and enabled the creation of the project objectives discussed in Section 2.3. Table 2.2 lists the system monitoring and control available in each of these systems to ensure the system created in this project exceed that of existing models.

Table 2.1 - Checklist of Water Parameter Control and Monitoring for some Existing Systems.

Model	Temperature		pH		DO		Salinity		Reference
	M	C	M	C	M	C	M	C	
Reef Ai Prime	✓	✓	✓	X	X	X	✓	X	(Reef Ai, 2014)
Reef Angel Plus	✓	✓	✓	X	X	X	X	X	(Reef Angel, 2012)
APEX	✓	✓	✓	X	X	X	✓	X	(Neptune Systems, 2012)
Digital Aquatics ReefKeeper Elite	✓	✓	✓	✓	X	X	✓	X	(Digital Aquatics, 2015)
Seneye REEF	✓	✓	✓		✓		✓	X	(Seneye)

Table 2.2 - Checklist of System Control and Monitoring for Some Existing Systems.

Model	Lighting	Water Levels	Water Pumps	Reference
Reef Ai Prime	✓	✓	✓	(Reef Ai, 2014)
Reef Angel Plus	✓	✓	✓	(Reef Angel, 2012)
APEX	✓	✓	✓	(Neptune Systems, 2012)
Digital Aquatics ReefKeeper Elite	✓	✓	✓	(Digital Aquatics, 2015)
Seneye REEF	✓	✓	✓	(Seneye)

2.3 Project Objectives

This thesis aims to not only create a comprehensive control and monitoring system through the use of a microcontroller, microprocessor and necessary sensors but also recreate the real-life environmental cycles that a marine coral reef ecosystem would experience, making the setup suitable for marine research. Based on the features available in existing systems it has been decided that the system will provide monitoring and control for the following water parameters: temperature, pH, salinity and dissolved oxygen. In order to incorporate realistic environmental cycles the lighting will be controlled to simulate day and lunar cycles via the time and date. In addition to these parameters the setup will also provide monitoring of less crucial elements such as water levels and device temperatures to further reduce the maintenance time required. Finally all monitoring and control will be possible through a web server which can be accessed via a HMI and any device with a web browser. To further aid the breakdown of the project objectives, the criteria introduced above are broken into the following list of aims of the system:

- Precisely control and monitor the following water parameters:
 - Temperature within $\pm 2\%$.
 - pH within $\pm 0.5\%$.
 - Salinity within $\pm 2\%$.
 - Dissolved Oxygen within $\pm 10\%$.
- Simulate realistic lighting with respect to timing, intensity and spectral distribution for:

- Day cycle
- Lunar cycle
- Control and monitor the following system parameters:
 - Water levels
 - Pump Operation
 - Lighting temperatures
- Control and monitor the system via a HMI and webpage (future works)
 - Client/server setup

2.4 Project Layout

To aid in the visualization of the project a number of overviews and layouts have been created. The first, seen in Figure 2-1, is a general component layout of the entire system including input components, output components and tanks. This layout was designed and used in the construction of the aquarium which is detailed in Appendix 2. Throughout this report reference will be made to the main tank and sump tank, which cycles water continuously, this layout, can be seen in Figure 2-2 and the plumbing setup for this is detailed in Appendix 2. The setup of the two tanks allows the control and monitoring to occur in the sump tank and increases water volume of the system which in turn increases the marine life that can be sustained within the tank (Reefkeeping Magazine, 2008). The final overview, seen in Figure 2-3, gives a brief look at the control and monitoring aspects of the system and how they will link together. With a large number of electrical components necessary for the project it was decided to locate all of these components in an electrical drawer which can be seen in general component layout in Figure 2-1. The layout to this electric drawer is shown in Appendix 2 along with the build details.

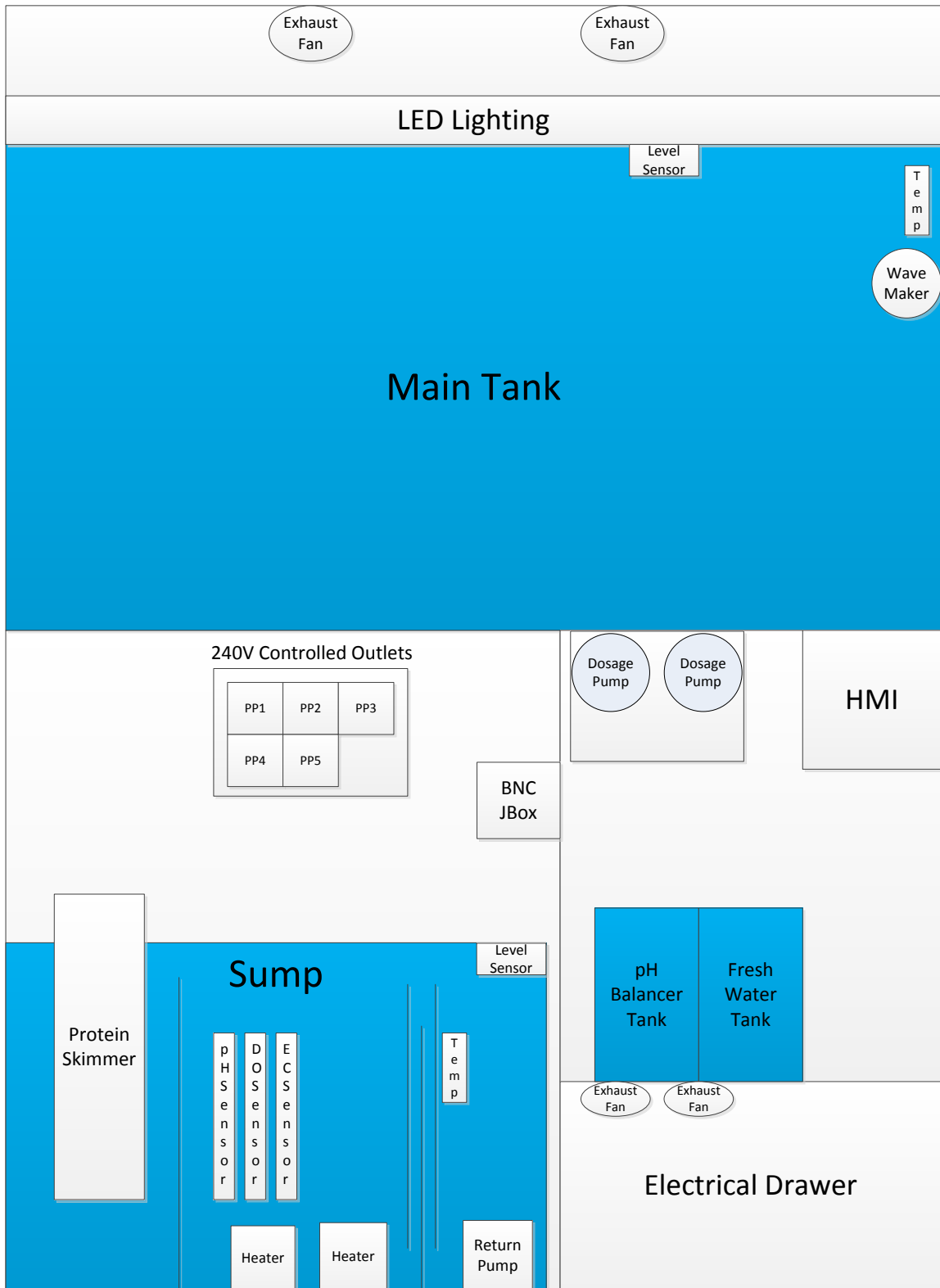


Figure 2-1 – General Component Layout of Aquarium.

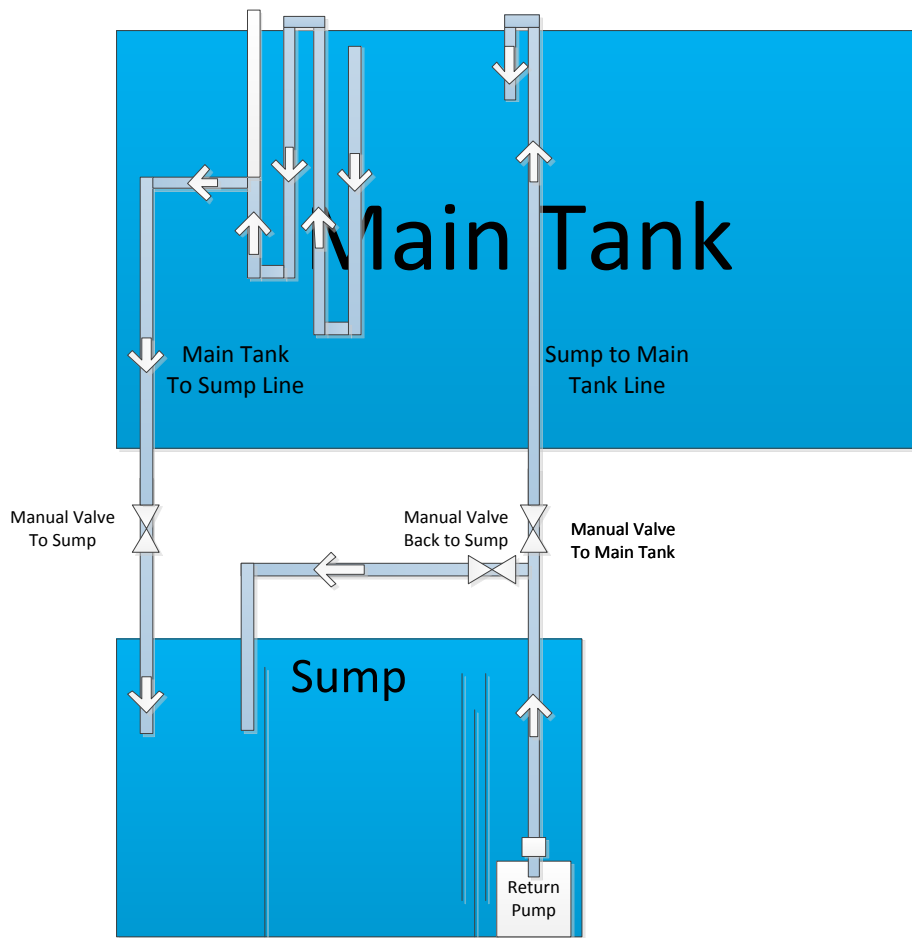


Figure 2-2 - Main Tank and Sump Tank Water Cycle.

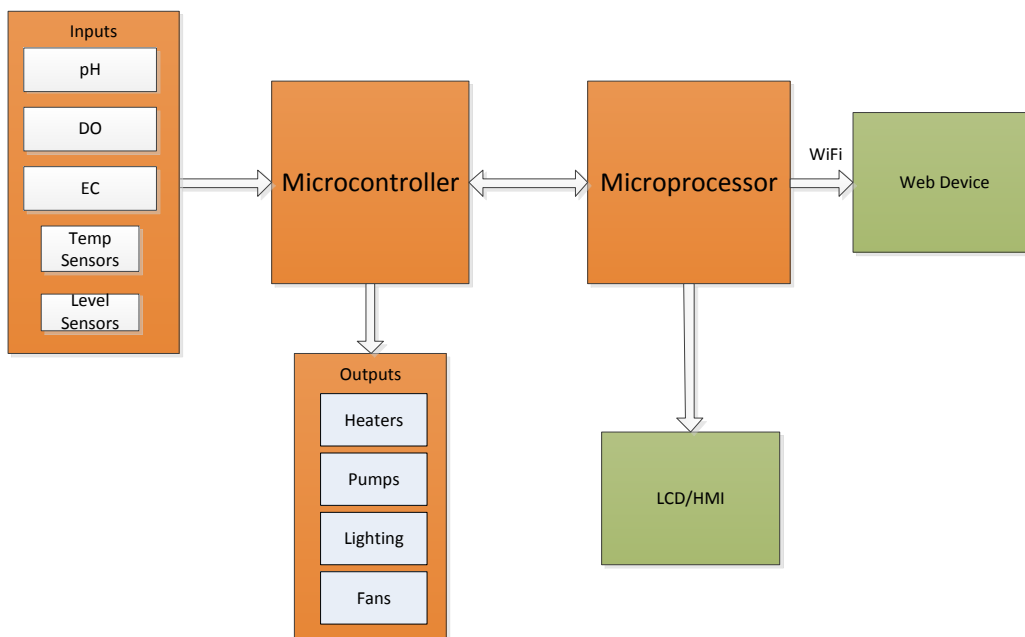


Figure 2-3 – Control and Monitoring Overview.

Chapter 3 : Hardware

3.1 Overview

The purpose of this chapter is to introduce the hardware components that make up the control and monitoring system of the reef aquarium. The Chapter is broken into 3 sections: Section 3.2 covers the individual hardware components used in the build and Section 3.3 details the power supplies and components.

3.2 Components

This section introduces the hardware components utilised in the build of the project along with their specifications, basic operation and design selection criteria for each major device.

3.2.1 PiLeven Microcontroller

The PiLeven (freetronics, 2015), seen in Figure 3-1, is a specific microcontroller board developed by freetronics that allows easy connection to the Raspberry Pi microprocessor. This model of microcontroller has the same features of an Arduino Uno with 6 analog inputs, 14 digital I/O and a 16MHz clock speed. An Arduino Uno is an open source microcontroller developed by Arduino for hobbyists (Arduino, 2015). The specific features of the PiLeven include a DC Power Jack that is capable of providing power to both the PiLeven and the Raspberry Pi and a serial connection for the communication between the two devices (freetronics, 2015).

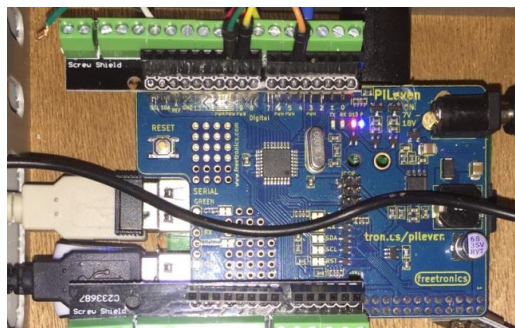


Figure 3-1 – PiLeven Microcontroller.

In selecting the microcontroller for the project a number of factors were considered to ensure the selected device would be suitable. The factors that were deemed of importance included performance, functionality and cost. Regarding performance, the PiLeven was more than capable of running the required tasks which include PWM, I2C communications and general I/O. The functionality of the PiLeven is what made it the most suitable microcontroller for this project. With the easy connection to the Raspberry Pi via the headers, as seen in Figure 3-2, the PiLeven provided power and communication to the Raspberry Pi without the need of additional wires or power supplies. When verifying the functionality of the microcontroller a checklist was made to ensure the PiLeven met the project's needs, this checklist can be seen in Table 3.1 along with a list of all specifications in Table 3.2. The additional features did come at a higher cost of \$39AUS in comparison to the Arduino Uno at \$8AUS from eBay but it was decided the benefits outweighed the additional cost (freetronics, 2015).

Table 3.1 – Microcontroller Checklist PiLeven Comparison.

Necessary Features	Required	PiLeven
Digital I/O	10	14
PWM	3	6
I2C	1 bus	1 bus (A4 SDA and A5 SCL)
One-wire	1	Supported on I/O pins

Table 3.2 – PiLeven Specifications (freetronics, 2015).

Microcontroller	
MCU Type	Atmel AtMega328p
Operating Voltage	5V(3.3V converter interface for Raspberry Pi)
MCU Clock Speed	16 MHz
PiLeven	
Input Voltage	7-18V DC. Recommended 12-18V DC, 1A current capacity
Digital I/O pins	14, 6 provide PWM output(at 8 bit 0 to 255)
Analog Input pins	6 (analog input pins double as digital I/O, giving 20 digital I/O if required)
Analog Resolution	10 bits, 0-1023 at 5V AREF; 4.88mV per step
Current Per I/O pin	40mA max
Total current for I/O pins	200mA max
Total Current for 5V outputs	3A max
Memory	

Flash memory	32KB
SRAM, EEPROM	2 KB SRAM, 1KB EEPROM
Communications	
Raspberry Pi Comms	Hardware serial and I2C connected from Raspberry Pi to microcontroller.
Serial	1 x hardware USART, SPI (Serial Peripheral Interface), 1 x I2C
Others	One-wire, multiwire, LCD etc

For the purpose of this project the PiLeven will allow the control and monitoring of all sensors and devices while relaying the collected data to the microprocessor.

3.2.2 Raspberry Pi Microprocessor

The Raspberry Pi (R Pi) as seen in Figure 3-2, is a microprocessor developed by the Raspberry Pi foundation with the intention of providing a small computer for educating computer science students (Raspberry Pi, 2015). The R Pi has a HDMI connection for a screen, USB ports for peripheral devices such as keyboard and mouse, along with a micro-SD slot which acts as the hard drive and holds all data along with the operating system. The R Pi performs all tasks capable of a typical desktop computer including web browsing, watching or listening to a variety of media content and word-processing. The R Pi exceeds the requirements of the project but due to its user friendly nature it was still considered a suitable option. The specifications for the Raspberry Pi Model 2 B used in this project can be seen in Table 3.3.

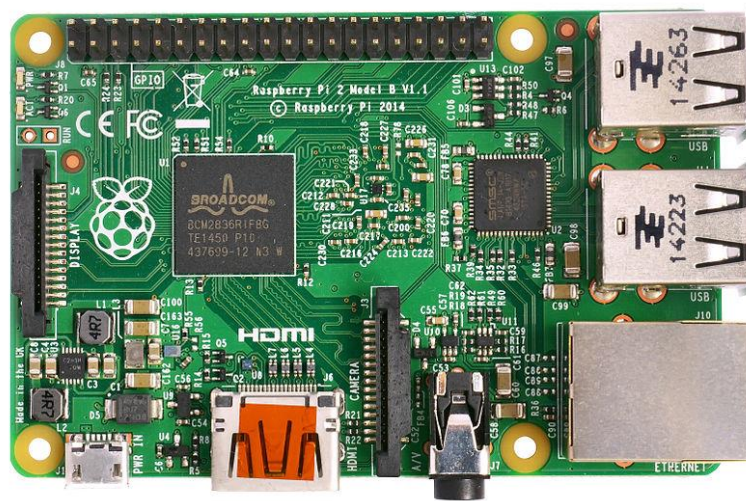


Figure 3-2 – Raspberry Pi Model 2 B (Multicherry, 2015).

In relation to this project the R Pi will take on the role of a server and master. To allow the R Pi to act as a server, a constant internet connection was required. This was achieved through the use of a Wi-Fi Dongle which connected to the R Pi in the same manner as the keyboard and mouse. For more details on the R Pi as a server and master see Chapter 8, Section 8.5.

Table 3.3 – Raspberry Pi Model 2 B Specifications (Raspberry Pi, 2015).

Microprocessor	
Processor	Broadcom BCM2836 ARMv7 Quad Core 32bit
Clock Speed	900MHz
Memory	1GB RAM
Connections	
USB ports	4
Ethernet	1
Audio	3.5mm jack
Video	HDMI
Communications	
SPI	2
I2C	2
UART	1

3.2.3 pH sensor and embedded circuit

The pH sensor utilized in this project was developed by Atlas Scientific who are known for designing a variety of scientific grade environmental and electrochemical sensors for embedded systems (Atlas Scientific, 2015). The pH kit provided by Atlas Scientific includes the pH probe

seen in Figure 3-3, the EZO pH circuit seen in Figure 3-4, 1 BNC connector and 3 calibration solutions. The probe is used to measure the pH of the water and then relays this information to the EZO circuit. The EZO circuit is the embedded circuit that allows communication to the microcontroller via UART or I2C modes. For this project, I2C mode is selected and a unique address given to the EZO circuit. For additional details on I2C mode see Chapter 4, Section 4.4.1 and for information into the importance of pH for reef aquariums see Appendix 1. Specifications for the EZO pH circuit are displayed in Table 3.4.

The pH sensor and EZO circuit calculates the pH by converting a current generated by hydrogen ion activity into pH. The current generated by the hydrogen ion activity is the reciprocal of that activity and can be predicted using the equation:

$$E = E^0 + \frac{RT}{F} \ln(\alpha_{H^+}) = E^0 - \frac{2.303RT}{F} pH \quad (\text{Equation 3.1})$$

Where E^0 is the cell potential in volts, R is the ideal gas constant, T is the temperature in Kelvin and F is the Faraday constant (Atlas Scientific).



Figure 3-3 – pH Probe.



Figure 3-4 - pH EZO Circuit.

Table 3.4 – pH and DO EZO Circuit Specifications (Atlas Scientific).

Power consumption				
	LED	MAX current	STANDBY current	SLEEP current
5V	On	18.3mA	16mA	1.16mA
	Off	13.8mA	13.8mA	1.16mA
3.3V	On	14.5mA	13.9mA	0.995mA
	Off	13.3mA	13.3mA	0.995mA
Absolute max ratings				
Parameter	Min	Typical	Max	
Storage temp (of EZO)	-65°C			125°C
Operational temp (of EZO)	-40°C	25°C		85°C
VCC	3.3V	3.3V		5.5V

3.2.4 Dissolved oxygen sensor and embedded circuit

The dissolved oxygen (D.O) sensor used in this project was also designed by Atlas Scientific. The dissolved oxygen sensor selected comes in a kit containing a D.O probe seen in Figure 3-5, an EZO D.O circuit seen in Figure 3-6, 1 BNC connector and 1 calibration solution. The D.O sensor communicates with the microcontroller in the same manner as the pH sensor detailed above and is also operating in I2C mode for this project. The specifications for the EZO D.O sensor can be seen in Table 3.4. For more details on the important of dissolved oxygen for reef aquariums see Appendix 1.

The D.O sensor by Atlas Scientific uses a Galvanic High-Density Polyethylene (HDPE) probe which gives a signal of 0mV for no oxygen and from 36mV to 54mV for 100% oxygen. This allows a basic calculation for the percentage saturation of oxygen in the water using the equation:

$$\%saturation = \frac{mV \text{ in water}}{mV \text{ in air}} \times 100 \quad (\text{Equation 3.2})$$

This percentage is very important but does not give any information on the water's ability to hold oxygen (Atlas Scientific). To get more accurate readings the temperature, pressure and salinity of the water are necessary (Atlas Scientific). These parameters are used to convert the

percentage saturation into mg/L. For more details on dissolved oxygen calculations see Appendix 1.



Figure 3-5 – DO Probe.

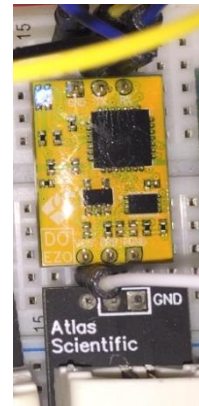


Figure 3-6- DO EZO Circuit.

3.2.5 Electrical Conductivity sensor and embedded circuit

In selecting the conductivity sensor it was important to ensure that it was capable of measuring the necessary range that the electrical conductivity (E.C) would fall into for sea water. The conductivity sensor selected was the K1.0 conductivity sensor by Atlas Scientific which provided not only the correct range but also converted the E.C measurement into Total Dissolved Solids (TDS), Salinity and Specific Gravity (Atlas Scientific). This proves to be a very useful feature of this sensor, since this project requires the salinity to be maintained; this is most commonly controlled through measuring the specific gravity or salinity in parts per thousand (ppt) (Algone, 2012). The K1.0 conductivity sensor kit provided by Atlas Scientific consists of an E.C probe seen in Figure 3-7, an EZO E.C circuit seen in Figure 3-8, 1 BNC connector and 2 calibration solutions. The E.C sensor communicates using the I2C bus in the same fashion as the pH and D.O sensors. The specifications for the E.C sensor can be seen in Table 3.5. Additional information on the importance of salinity for marine aquariums can be found in Appendix 1.

The E.C sensor by Atlas Scientific is a complex device consisting of multiple layers. The details of these layers can be found in the E.C EZO data sheet (Atlas Scientific).



Figure 3-7 – EC Probe.



Figure 3-8 – EC EZO Circuit.

Table 3.5 – EC EZO Circuit Specifications (Atlas Scientific).

Power Consumption					
	LED	MIN	MAX	STANDBY	SLEEP
5V	On	22.5mA	50mA	18.14mA	0.7mA
5V	Off	17.5mA	45mA	15.64mA	0.7mA
3.3V	On	20mA	35mA	16.85mA	0.4mA
3.3V	Off	19mA	34mA	15.85mA	0.4mA
Absolute max ratings					
Parameter	MIN	TYPICAL		MAX	
Storage temp (of EZO)	-60°C			150°C	
Operational temp (of EZO)	-40°C	25°C		125°C	
VCC	3.3V	5.0		5.5V	

3.2.5 Additional Minor Components

8 Channel Relay Module

The 8 channel relay module used in this project enables the microcontroller to control a variety of devices that have different voltages and higher currents while still providing isolation between the PiLeven and relay module. For this project the relays provide control of the 240V outlets (heaters and pumps), 12V fans, peristaltic pumps and 12V fans. The relay module can be seen in Figure 3-9 and the specifications in Table 3.6. This particular relay module was designed specifically for use with Arduino microcontrollers making it the most suitable option for relay control. The control action of this relay module operates by turning on a relay upon receiving a

logical LOW from the microcontroller. The isolation between the microcontroller and relays is provided through optocouplers which are built into the relay module.

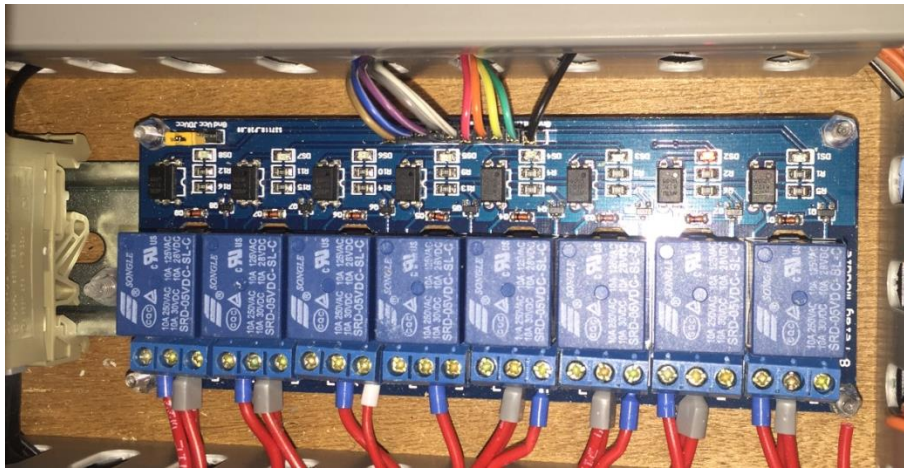


Figure 3-9 – 8 Channel Relay Module.

Table 3.6 – 8 Channel Relay Module Specifications (SainSMART).

Channels	8
Output voltage rating	240VAC
Output current rating	10A (max)
Input voltage rating	3-5V DC
Input driver current	15-20mA
Dimensions	16 x 4.8 x 2 cm
Weight	140g

Ultrasonic Ranging Module HC-SR04

The ultrasonic range module pictured in Figure 3-10, is a device that measures the time it takes for an ultrasonic sound wave to travel to an object and back again. The time it takes for this to occur can be measured and used to calculate the distance from the sensor to the object. The HC-SR04 ranging module has four necessary wires: a 5V supply, Trigger Pulse Input, Echo Output, and ground. The basic operations of the ultrasonic ranging module are as follows (ELEC Freaks):

- A pulse of 10 μ s in length is sent to the signal pin from an output pin of the microcontroller (μ C)

- The module then outputs eight cycles of 40kHz ultrasonic pulses and detects a return echo
- The measured distance is then proportional to the echo pulse width

The specifications for the HC-SR04 can be seen in Table 3.7. The ultrasonic ranging modules are utilised in this project to measure the water levels of both tanks.

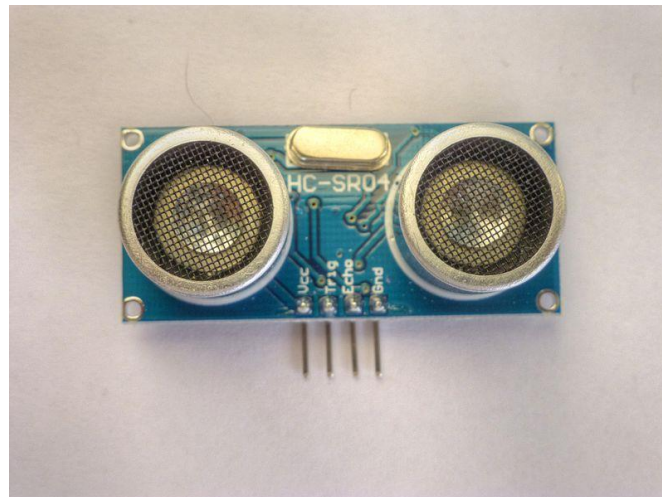


Figure 3-10 – Ultrasonic Ranging Module HC-SR04 (Dilmen, 2014).

Table 3.7 – Ultrasonic Ranging Module HC-SR04 Specifications (ELEC Freaks).

Power Supply	5V
Current	15mA
Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degrees

Water Proof DS18B20 Temperature Sensor

The temperature sensor utilised in this project is a water proof DS18B20 which provides between 9 and 12 bit Degrees Celsius (°C) temperature measurements. The DS18B20 (Maxim Integrated, 2008) communicates on a Dallas 1-wire bus which only requires one port pin making it an excellent sensor for this project as it frees up other pins for general I/O use. Each DS18B20 temperature sensor has a unique 64 bit address which allows multiple sensors to share the 1-

wire bus (Maxim Integrated, 2008). Specifications for the DS18B20 can be found in Table 3.8. For information of the importance of temperature monitoring of a reef aquarium see Appendix 1.

Table 3.8 – DS18B20 Specifications (Maxim Integrated, 2008).

Power supply max	5.5V
Power supply min	3V
Temperature min	-55°C
Temperature max	85°C
Resolution	9 to 12 bit
Protocol	1-wire
Address	Unique 64 bit
Query time	Convert 12 bit in less than 750ms
Pin setup	3 pins (VCC,GND,DATA)
Uncertainty	±0.5%

Real Time Clock DS1307

The DS1307 real time clock (RTC) is a low power, full binary coded decimal (BCD) clock and calendar which transfers data such as time and date via I2C to a microcontroller. Specifically the clock provides information on seconds, minutes, hours, day, date, month and year. The RTC has the ability to automatically detect power failures and switch to the battery backup making this device highly suitable for this project. The specifications for the DS1307 RTC can be found in Table 3.9. The DS1307 has an external 32.768 kHz crystal which is used to provide accurate timing. The RTC is essential for this project as it allows tasks to be executed at specific times of the day and provides time and date information for the lighting system (Maxim/Dallas Semiconductor, 2008). For more information on how the RTC is used for the lighting see Chapter 5.

Table 3.9 – RTC Specifications (Maxim/Dallas Semiconductor, 2008).

Power supply max	5.5V
Power supply min	4.5V

Temperature max (commercial)	70°C
Temperature min (commercial)	0°C
Communications	I2C

Peristaltic Pump (Dosage Pump)

The peristaltic pumps used in the project are essentially dosage pumps used to control different water parameters. A peristaltic pump is a positive displacement pump used for pumping various fluids. As can be seen in Figure 3-11, the pump operates by rotating a number of rollers around and along a flexible tube containing the fluid; as the roller moves around, it pinches closed the flexible tube forcing the liquid to move in the direction of the rotation. When the roller releases the flexible tubing the fluid will then exit the pump. In selecting the peristaltic pumps for this project only one criterion was necessary to meet, this being the flow rate. The peristaltic pumps used in this project operate using an electric motor at 12V and are controlled using relays from the 8 channel relay module. Specifications of the selected peristaltic pumps can be seen in Table 3.10. Further breakdown of the use of the peristaltic pumps in this project can be found in Chapter 6.

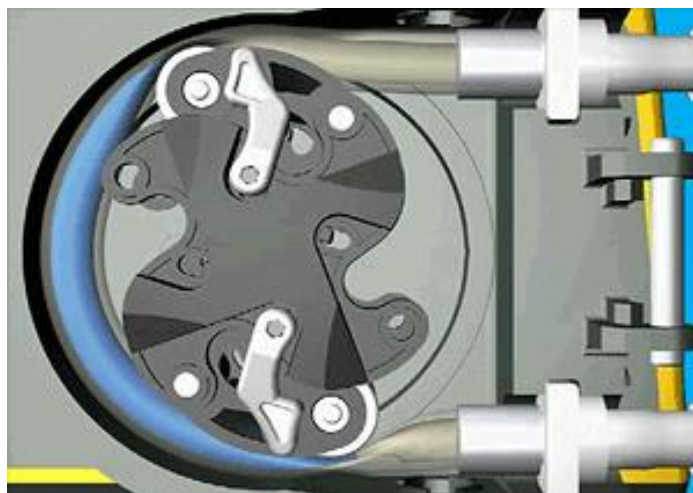


Figure 3-11 – Peristaltic Pump Operation (Njmcca, 2013).

Table 3.10 – Peristaltic pump specifications.

Power Supply	12V DC
Current	80mA

Temperature max	40°C
Temperature min	0°C
Flow rate max	60 mL/min
Flow rate min	20 mL/min

Submersible Pump

As mentioned in Chapter 2, water in the sump and main tank is continually circulated. To achieve this, a submersible pump is used to pump water from the sump back to the main tank. In selecting the submersible pump it was essential that the pump would be capable of operating in salt water, that the required flow rate can be met and that the outlet diameter is correct. Details on the aquarium flow rates and plumbing can be found in Appendix 1. The pump selected for the task has a more than adequate maximum flow rate of 1500 L/hour and a flow control lever to further reduce the flow rate. Specifications for the selected pump can be seen in Table 3.11.

Table 3.11 – Submersible Pump Specifications.

Power supply	240V AC
Power	19W
Max fluid temperature	35°C
Flow rate max	1500 L/H (400 GPH)
Fluids	Fresh water and Salt water
Outlet Diameters	13mm or 20mm
Head max	1.6m

Submersible Heater

A variety of water heating methods were investigated including immersible heaters, filter heaters and submersible heaters. The immersible heaters operate by heating the water from above the water line; this however was not suitable as the heating elements would obstruct the lighting system for the tank. The filter heaters are known to be extremely effective but are also very costly. With this in mind the most suitable heater for this project was the submersible heater as it was reasonably cheap and its only drawback was that it could damage plant life upon

contact, but since the heaters are located in the sump, away from plant life, this was no longer an issue. It was necessary to select the correct power rating for the submersible heater to ensure it would be capable of heating the volume of water of both the sump and main tank combined. Details on the power requirements for aquarium heaters and the importance of aquarium temperature can be found in Appendix 1. For the project, it was calculated that the submersible heater would need to provide 200W of power. Due to the length of submersible pumps it was further decided that this 200W could be split across two shorter 100W submersible glass heaters.

Submersible Power Head

The submersible power head manufactured by Wiltec (Wiltec, 2006) used in this project provides increased circulation and increases dissolved oxygen levels in the water, Figure 3-12. Further details of the use of the submersible pump in this project and the importance of dissolved oxygen for marine aquariums can be found in Chapter 6 and Appendix 1 respectively. The only selection criteria for the submersible power head was the flow rate. The requirement for the flow rate was to ensure that enough water flow would occur without causing disturbance to the environment.

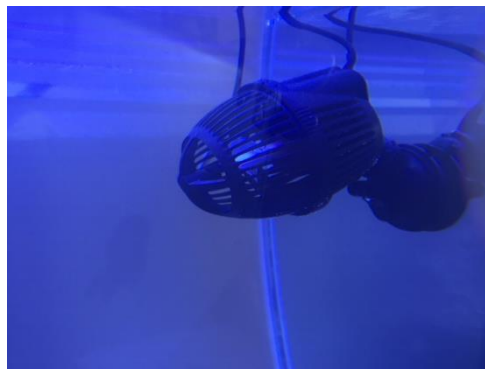


Figure 3-12 – Submersible Power Head.

Power MOSFET IRF540N

A Power MOSFET seen in Figure 3-13 is a semiconductor device which is capable of switching high power DC currents between the Drain and Source of the MOSFET. The MOSFET will switch on and off based upon the duty cycle applied to the Gate of the MOSFET. A basic circuit diagram of this operation can be seen in Figure 3-14 in which a lamp is controlled using a MOSFET. The MOSFET selected for this project is the IRF540N (International Rectifier, 2010) and is used in controlling the LED lighting for the project. Accompanying each MOSFET used in this project is a heat sink to ensure that operating temperatures are not exceeded. Further details on how the IRF540N power MOSFET is utilized in the LED lighting system can be found in Chapter 5.



Figure 3-13 – IRF540N MOSFETS

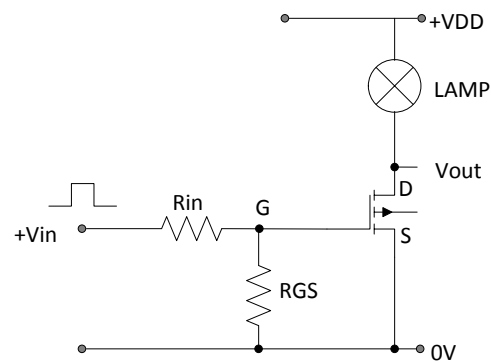


Figure 3-14 – Example MOSFET Circuit to Control a Lamp.

CREE LED

The LEDs selected for this project are manufactured by CREE, a market leading innovator of lighting-class LEDs. Specifically, the LEDs selected from CREE are the XLAMP XP-E LEDs (CREE, 2015), seen in Figure 3-15, which come in a variety of wavelengths and intensity. The criteria for the wavelengths selected for the lighting is detailed in Chapter 5, Section 5.2, with further details on the importance of lighting for corals found in Appendix 1. Specifications for the LEDs selected can be seen in Table 3.12.



Figure 3-15 – CREE XPE LED.

Table 3.12 – LED Specifications (CREE, 2015).

LED Colour	Wavelength (nm)	No. used in project	Forward Voltage(V)	Max Current (mA)	Intensity (lumens)
Royal Blue	450 – 465 nm	15	3.1 @350mA	1000	35
Cool White	450 & 530 – 600 nm	11	3.05 @350mA	1000	NA
Violet	410 - 430 nm	6	3.1 @350mA	1000	35
Red	620 – 630 nm	1	2.3 @ 700mA	700	62
Green	520 – 535 nm	1	3.3 @350mA	1000	100

7 Inch HDMI Touchscreen

Acting as a monitor to the Raspberry Pi and a HMI to the project is a 7 inch HDMI touchscreen (Waveshare, 2015) by Waveshare. With 800 x 480 resolution, easy plug and play HDMI and USB outlets for video, power and touch screen and a low cost of \$60 from eBay the 7 inch LCD touch screen by Waveshare was by far the most suitable monitor for the project.

3.3 Power Supplies and Converters

The project required a variety of different power supplies and voltage manipulation components due to the varying voltages needed for each device and component detailed in Section 3.2. The Power supplies and components used in this project are; a 240V AC to 12V DC inverter, a 12V DC to 5V DC buck converter and multiple 12V boost converters. This section will delve into specifications for these components and the selection process taken for each.

3.3.1 240 V AC to 12V DC Switch Mode Power Supply

The main power supply for the project required a 240V supply and a 12V output due to the large number of components that require 12V. The current requirements for the project were calculated using the specifications for each device at its maximum operating current. Based on the maximum operating current for the project, a 30 A switch mode power supply was selected as the main power supply. The calculations for maximum project current can be found in Appendix 3. The specifications for the inverter selected can be seen in Table 3.13.

Table 3.13 – Inverter Specifications.

Input Voltage	240V AC
Output Voltage	12V DC
Output current max	30 A
Power	360 W
Temperature max	65°C
Temperature min	-20°C
Size	20 x 11 x 5 cm

3.3.2 12V DC to 5V DC Buck Converter

For simple low current microcontroller projects the 5V Vcc is normally supplied from the microcontroller itself, however, due to the high power requirements, caused by a large number of 5V devices, it was necessary to use an alternative supply. For this project the 5V Vcc supply is provided by means of a 12V to 5V buck converter which is supplied from the 12V switch mode power supply. A buck converter is a step down converter which produces a lower average output voltage than its DC input voltage (Mohan, 2003). This buck converter, shown in Figure 3-16, was selected due to its 100 W power rating and its 10 A maximum current rating which exceeds the requirements of all 5V devices. The power and current calculations for all 5V devices can be found in Appendix 3 and the specifications for the buck converter used in Table 3.14.

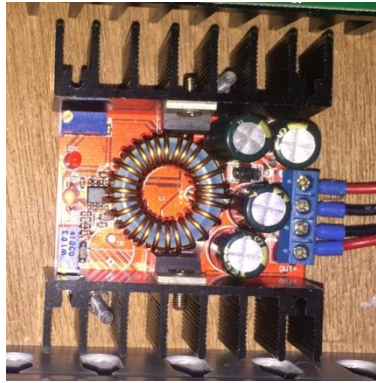


Figure 3-16 – Buck Converter.

Table 3.14 – Buck Converter Specifications.

Input Voltage	7 V – 12 V DC
Output Voltage	1.25 V – 12 V DC
Output Current	10A max, 8A for prolonged periods
Output Power	100W, 150W with heat sink upgrade
Working Temperature	40°C - 85°C
Working Frequency	300 kHz
Efficiency	95%
Short Circuit Protection	Yes (14A limit)
Temperature Protection	Yes

3.3.3 12V DC Boost Converters

The lighting portion of the project is made of multiple series circuits which contain the different CREE LEDs detailed in section 3.2.5. Due to the varying number of LEDs in each circuit and the different forward voltages of each, different voltages are required for each circuit. In order to provide these voltages boost converters have been used to increase the 12V supply to the required voltage for the specific circuit. Boost converters are as their name implies, step up the voltage; that is to say that they produce a higher average output voltage than their DC input voltage (Mohan, 2003). The boost converters selected for the lighting were 600W 15A boost converters which are more than capable of meeting the requirements of each LED circuit. Details on the LED circuits and voltage calculations can be found in Chapter 5, Section 5.3. The boost converter can be seen in Figure 3-17 with its specifications in Table 3.15.



Figure 3-17- Boost Converter.

Table 3.15 – Boost Converter Specifications

Input Voltage	10 V – 60 V DC
Output Voltage	12 V – 80 V DC
Input Current	15A
Output Current	10A adjustable
Output Power	800W
Working Temperature	40°C - 85°C
Efficiency	95%
Short Circuit Protection	Yes (fuse)
Temperature Protection	Yes

Chapter 4 : Software Applications, Programming Languages and Protocols

4.1 Overview

Throughout the project a variety of software applications, programming languages and protocols have been utilised and will be referred to throughout this document. This chapter will give a brief introduction to these with Section 4.2 covering the main applications used, Section 4.3 introducing the important programming languages utilised and Section 4.4 detailing the protocols used and how each operates.

4.2 Software Applications

An application is a program or group of programs designed to perform a variety of functions, tasks and activities to benefit an end user. Typical applications include word processors, spread sheets, games and web browsers (Beal, Applications, 2012). In the context of this project the applications referred to are those which allow programming and system manipulation capabilities.

4.2.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is the software application used to write and upload a program to any Arduino-based microcontroller, including the PiLeven used in this project. The Arduino IDE seen in Figure 4-1 allows for programs (known as sketches) to be written using the specifically created Arduino programming language further detailed in Section 4.3.1 (Arduino, 2015). The sketches written in the arduino IDE can alternatively be written in C/C++.

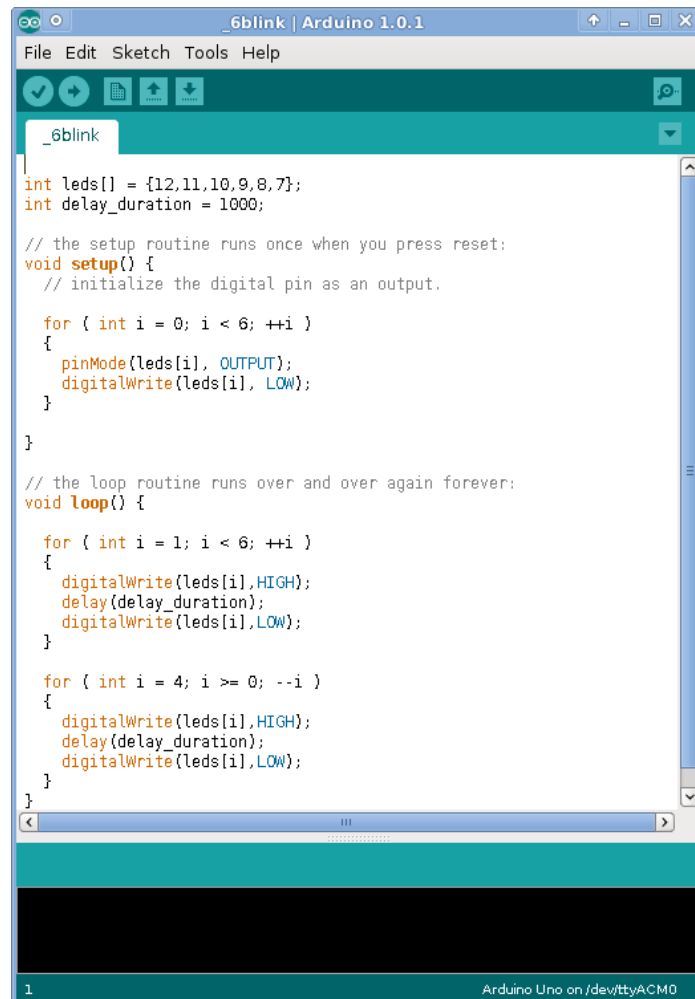


Figure 4-1 – Arduino IDE (Messer, 2012).

The Arduino IDE has a number of features which make it beneficial to this project, the first being the error box which can be seen as the black box at the bottom of the application window seen in Figure 4-1. This window displays all the errors found in the written sketch upon attempting to compile the sketch. Included in the arduino IDE is the ability to access a huge selection of examples useful in starting basic sketches. For this project it was possible to install a version of the arduino IDE onto the raspberry Pi which made uploading and testing programs easier than via a separate computer.

4.2.2 Raspberry Pi Terminal

A terminal or text terminal is a serial interface used for text entry and display (Raspberry Pi). The default terminal for the Raspberry Pi is the LXTerminal which is a terminal emulator. This means

that it is an application that emulates the old-style video terminals before graphical user interfaces were developed. This terminal application allows the user control over the Raspberry Pi. Just like other terminal applications, LXTerminal allows the user to directly manipulate their system using a set of predefined Linux commands (Raspberry Pi). A list of frequently used commands for this project can be found in Table 4.1. These commands allowed manipulation of the Raspberry Pi and facilitated the download of other applications needed such as the Arduino IDE.

Table 4.1 – Linux Commands (Raspberry Pi).

Command	Description
Sudo	Allows a command to be run as a superuser or another user.
Apt-get	Allows the installation and management of software on the Raspberry Pi.
mv	Moves a file and places it in the selected location
ls	Lists the content of a selected directory.
wget	Downloads a file from a website directly.

4.2.3 Notetab

Notetab is a text editor application which has extremely useful HyperText Markup Language (HTML) editing features. For this project Notetab was primarily used as a webpage editor and included a variety of code snippets for both HTML and Cascading Style Sheets (CSS). An additional feature of Notetab is that it provides syntax highlighting, which makes webpage construction far less confusing (NoteTab, 2015).

4.3 Programming Languages

A programming language is a specific language developed to create applications, scripts and other executable instructions for computers and embedded systems (Computer Hope, 2015). Throughout this project a number of different programming languages have been utilised, this section will introduce a few of the major languages used.

4.3.1 Arduino

The Arduino programming language is an open source language developed by Arduino for use with Arduino based microcontrollers. The language is built around the common computer programming language C/C++ with the major differences being the simplification of the programming functions and syntax (Arduino, 2015).

4.3.2 C/C++

The programming language C was developed in the 1970s and is a high level language originally used for writing Unix programs. Since then C has grown into one of the most commonly used programming languages and is now used to write applications for nearly every available platform. Built off the C programming language C++, pronounced “C plus plus”, uses syntax which is almost identical to that of C with the main difference being the addition of object-orientated features. The object orientated features make C++ an easier and more efficient language and has hence replaced C where possible (Christensson, C/C++ Definition, 2007). This project has incorporated a number of C/C++ functions and syntax within the main program of the microcontroller.

4.3.3 HTML

The computer programming language HyperText Markup Language (HTML) was designed for the creation of web based documents or webpages. HTML utilises what is known as hyperlinks, these hyperlinks allow transitions between webpages located anywhere within the internet. Like other languages HTML has a variety of syntax which define the actions and object orientation of the webpage (Christensson, HTML Definitions, 2015). In this project HTML has been used to create the different webpages for monitoring and controlling the system.

4.3.4 CSS

Cascading Style Sheet (CSS) is programming language utilised to ensure that a webpage is displayed in the correct format. Some of the formatting abilities include; font, colour, spacing,

borders, locations and sizing. The major benefit of CSS is that it allows a separate file that formats the content of the HTML file making the coding structure cleaner and easier to follow or adjust (Christensson, CSS Definition, 2006). CSS was used in this project to aid the formatting of the webpages created.

4.4 Protocols

A protocol is a standard used to define a method of exchanging data between devices on a network. Each protocol is different in some way, each with their own format of data transmission and receipt (Computer Hope, 2015). For this project a number of protocols were utilised and this section will delve into the two major protocols used in the communication between sensors and the microcontroller.

4.4.1 I2C

I2C or Inter-IC (I2C Bus, 2015) is a bidirectional two wire serial bus which provides a means of communication between integrated circuits (ICs). First developed by Phillips in the 1980s for mass produced technologies such as televisions and audio equipment, I2C has grown into one of the most common communications solutions for embedded systems (Rouse, I2C bus (inter-IC bus), 2005). I2C provides three different data speeds for communication: standard at 100Kbps, fast at 400 Kbps and high speed which can provide up to 3.4Mbps. Additionally I2C allows a variety of voltages and supports both 7 and 8 bit addresses (I2C Bus).

The two wires used for I2C are serial data (SDA) and serial clock (SCL). I2C is a master/slave setup in which both device types are connected to only the four wires, SDA and SCL, for communication and power and ground. It is possible for each device to be a transmitter, receiver or both. This is specified by whether the device is set to be a master or slave. A master generates the bus clock and also initiates the communication with all other devices. A slave device receives instructions from the master and sends its transmission upon receiving the command to do so. For correct communication all devices on the one bus must have their own unique address (I2C

Bus, 2015). In this project I2C has been used to communicate from the microcontroller to the RTC, water parameter sensors and back, as detailed in Chapter 6.

4.4.2 One-Wire

The one-wire protocol is a serial communication bus that uses a single data line and a ground reference. One-wire is a master/slave setup in which the master initiates and controls the communication with one or more one-wire slave devices. Each slave device has a 64 bit address which is factory programmed and unalterable. One-wire is unique as a protocol as it is the only-voltage based digital system that works with only two contacts for half duplex bidirectional communication (Maxim Integrated Products, 2008). In this project the temperature sensors detailed in Section 6.2 are one-wire devices and hence this protocol was utilised. The one-wire protocol used in this project is supported through the use of the DallasTemperature.h and the OneWire.h libraries (MilesBurton, 2014).

Chapter 5 : Lighting

5.1 Overview

This chapter details all aspects of the lighting used in the project. The lighting is one of the most significant aspects of the project, providing realistic day and night lighting which is essential for marine life such as corals and algae (Orphek, 2015). The main approach to the lighting for this project was to use LEDs in specifically designed circuits that are capable of being controlled via a microcontroller that is programmed to provide the necessary lighting at the appropriate time. This approach and implementation is detailed in the following sections: the LED selection in Section 5.2, circuit designs in Section 5.3, lighting control in Section 5.4 and testing in Section 5.5.

5.2 LED Selection

Lighting is one of the most important ingredients for successfully growing coral in a reef aquarium and can be broken down into three specific concerns: Luminous flux, Photosynthetic Available Radiation (PAR) and Wavelengths (Orphek, 2015). PAR refers to the spectral range of radiation between the ranges of 400 and 700 nanometres. When looking at PAR it is necessary to understand that Chlorophyll is the pigment present in coral that allows the absorption of light required for photosynthesis (Science Daily, 2015). In particular the chlorophyll types important for coral are chlorophyll 'a' and chlorophyll 'b'. In Figure 5-1 it can be seen that chlorophyll 'a' best absorbs wavelengths in the ranges of 410 to 460 nm and 665 to 675 nm, whereas chlorophyll 'b' best absorbs in the ranges of 460 to 480 nm and 630 to 650 nm (Orphek, 2015). This indicates, that at these wavelengths, chlorophyll a and b absorb the greatest amount of light for photosynthesis. However, it has been found that green and red spectrum light is less effective for coral growth suggesting that the 410 to 460 nm and 460 to 480 nm wavelengths are the most suitable for coral growth (Riddle, Feature Article: Imitating Natural Light Quality, Intensity, and Dosage in a Reef Aquarium, 2013). Therefore, it was decided that the lighting for the project would attempt to emit these wavelengths using LEDs as best as possible.

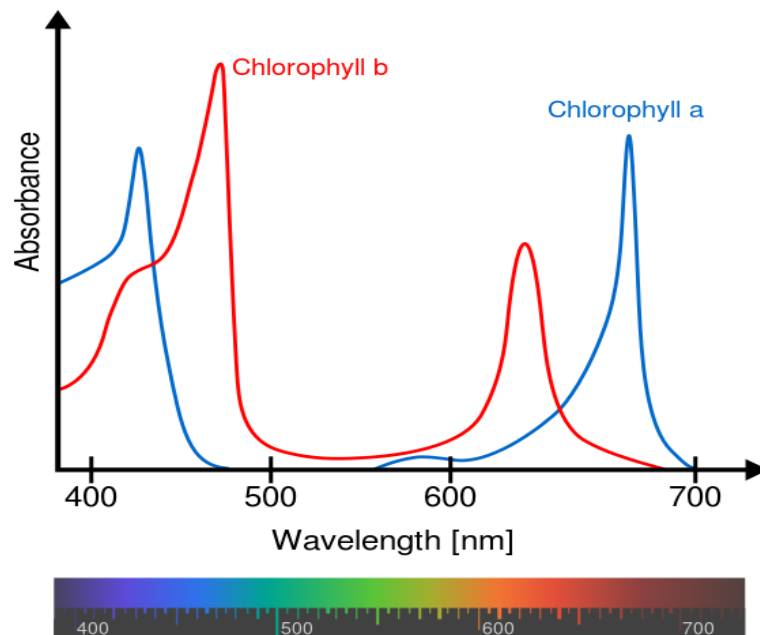


Figure 5-1 – Chlorophyll ‘a’ & ‘b’ Spectra, Wavelengths vs. Absorbance (Aushulz, 2012).

In addition to the lighting required for coral growth it was also decided to add specific wavelengths of light based on the light spectrum of the sun and the spectrum of light penetration into seawater. The spectral distribution of sunlight can be seen in Figure 5-2 and the penetration of light into seawater according to wavelength in Figure 5-3. From Figure 5-3 it can be seen that wavelengths between 400nm and 500nm penetrate the greatest depths and the penetration depth steadily decreases to 700nm with the red spectrum penetrating the least.

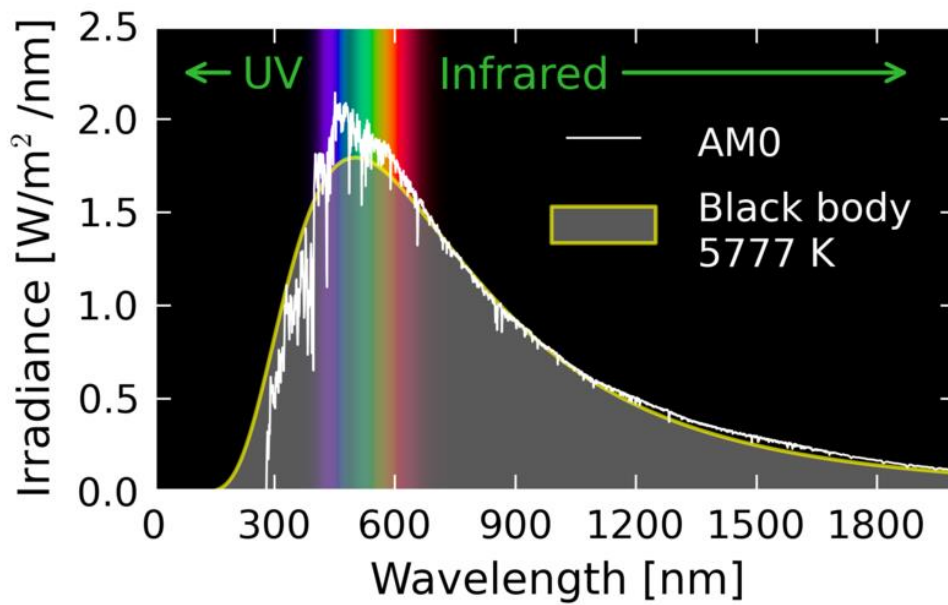


Figure 5-2 – Spectral analysis of the sun to show the highest intensity wavelengths found at sea level (Danmichaelo, 2012).

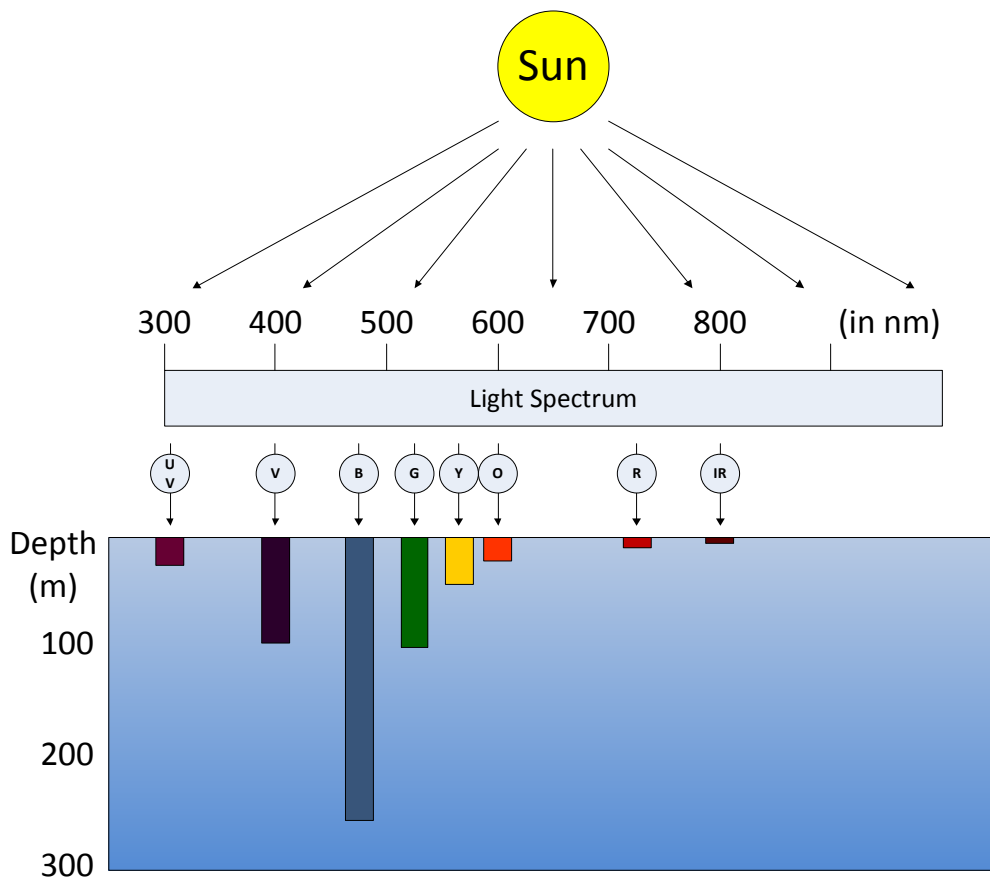


Figure 5-3 - Penetration of Light in Seawater, adapted from (NASA, 2012).

With this information it was decided, that in order to incorporate some of the additional lighting needed to replicate sunlight that would effectively penetrate the water, two additional wavelengths would be incorporated along with cool white LEDs. The cool white LEDs provide a peak wavelength at approximately 450nm and lower intensity wavelengths between 530 and 600 nm. The two individual wavelengths were selected to be in the red and green region of the spectrum but only at low intensities, since they do not penetrate very deep and are not particularly useful for coral growth but do have aesthetic appeal (Riddle, Feature Article: Imitating Natural Light Quality, Intensity, and Dosage in a Reef Aquarium, 2013).

In order to satisfy the lighting requirements detailed above, the specific wavelength LEDs were selected, matching the above mentioned wavelengths as closely as possible. Detailed in Table 5.1 are the selected LED wavelengths and their power, along with the required wavelength.

Table 5.1 – LED specifications and Wavelength Match Ups.

Wavelength of LED	Required Wavelength	Power	Colour
450 – 465 nm	460 – 480 nm	3W	Royal Blue
410 – 430 nm	410 – 460 nm	3W	Violet
520 – 535 nm	Any in Green spectrum	3W	Green
450, 530 – 600 nm	410 – 480 nm	3W	Cool White
620 – 630 nm	Any in Red spectrum	3W	Red

The next concern for the lighting was the luminous flux required for the tank, which was based directly off the size of the tank and the expected content. Luminous Flux is defined as a quantitative expression of the brilliance of a source of visible light and is measured in lumens (lm) (Rouse, luminous flux, 2005). The luminous flux for the tank used in this project is set by the number of each LED type needed. To calculate this, a LED fixture calculator, found on the Nano-Reef forum specified the required number of LEDs based on the tank dimensions entered. This calculator can be found in Appendix 3. The numbers for each LED type can be seen in Table 5.2 with the arrangement in Figure 5-4. The arrangement of the LEDs was based on an even

coverage across the tank of each LED type. As seen in Figure 5-5, the LEDs were mounted onto aluminium heat sinks according to the arrangement, wired in the circuits detailed in Section 5.3 and then mounted on the lid of the lighting hood. More details on the lighting hood and the build can be found in Appendix 1.

Table 5.2 – Numbers of Each LED Type.

LED Type	Number
Royal Blue	15
Violet	6
Green	1
Cool White	11
Red	1

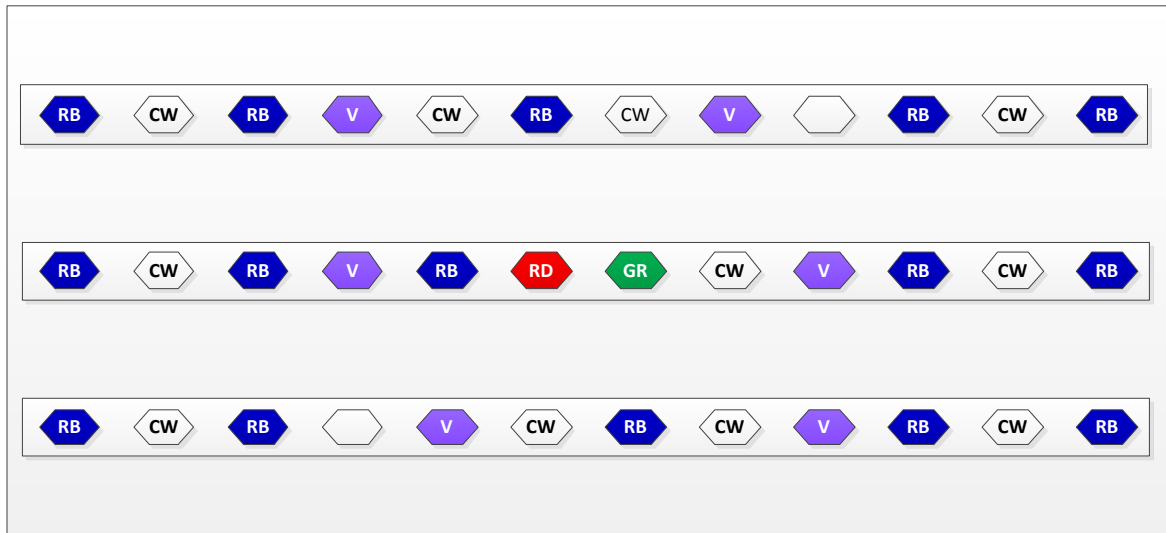


Figure 5-4 – LED Arrangement.

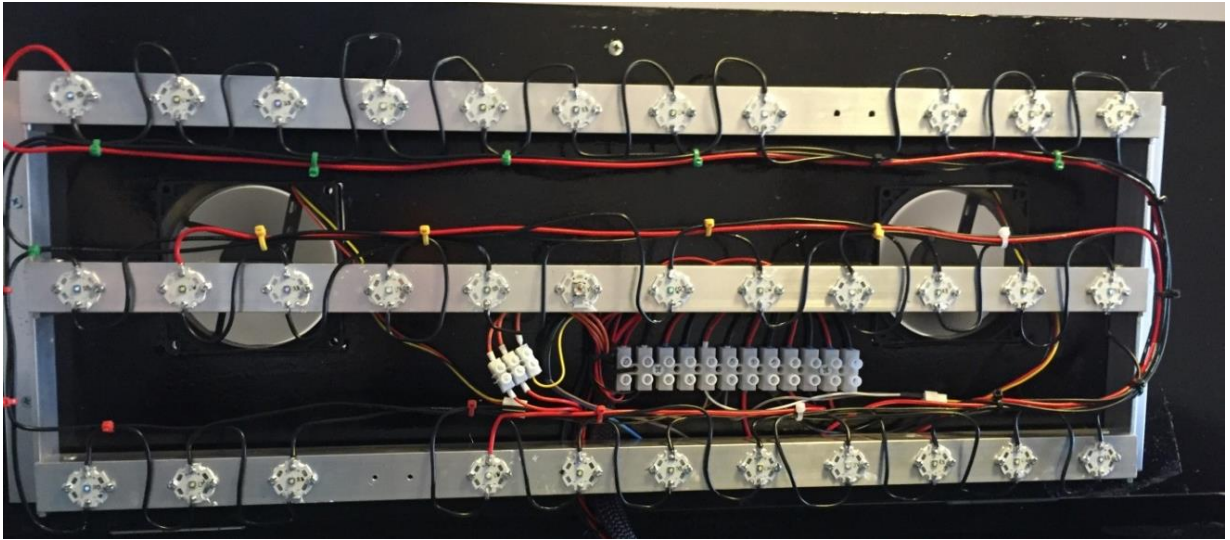


Figure 5-5 - Mounted and Wired LEDs.

5.3 LED Circuit Design

In designing the circuits for the LEDs it was necessary to take a number of factors into account, these included: the forward voltage of each LED type, the planned control over each type and the rated current for each type. Based upon these concerns the following criterion was created;

- The different LED types would all be controlled separately where possible.
- Each circuit of LEDs would be wired in series due to the need for equal current through each LED.
- The supply voltage to each circuit would be different due to the number of LEDs in each and the forward voltage of each LED type.
- Each circuit will be controlled via a MOSFET to ground.

Using this criterion the LEDs were divided into four series circuits with varying currents and voltages for each. Table 5.3 details the specifics for each of the circuits including; voltage supply needed, current through the circuit, total power and number of LEDs. The voltage and power for each circuit was calculated using the forward voltages and currents listed in Table 3.12 from Section 3.2.5 using the following equations:

$$V = No. of LEDs \times V_F + V_D \quad (\text{Equation 5.1})$$

$$P = I \times V$$

Where V_F is the forward voltage of the LED, V_D is the voltage drop across the MOSFET and I is the suggested current for the LED.

Table 5.3 – LED Circuit Specifications.

Circuit No.	Led Types	No. of LEDs	Voltage	Current	Power
Circuit 1	Royal Blue	15	47.7V	350mA	16.695W
Circuit 2	Cool White	11	34.75V	350mA	12.1625W
Circuit 3	Violet & Green	7	22.9V	350mA	8.015W
Circuit 4	Red	1	2.3V	700mA	1.84W

Due the varying voltages for each LED circuit it was necessary to use boost converters to step up the 12V source to each circuits required voltage. The boost convert voltages and circuit currents were set using the potentiometers highlighted in Figure 5-6 and the circuits wired as seen in Figure 5-7. The circuit design used for the lighting was initially selected due to its low cost. Following additional research it was found that controlling the LEDs in this way could lead to changes in forward voltages due to the LEDs rising in temperature. This could in-turn damage the LEDs. To overcome this, the method of using LED drivers to control the lighting was researched and further details of this method are described in Future Works.

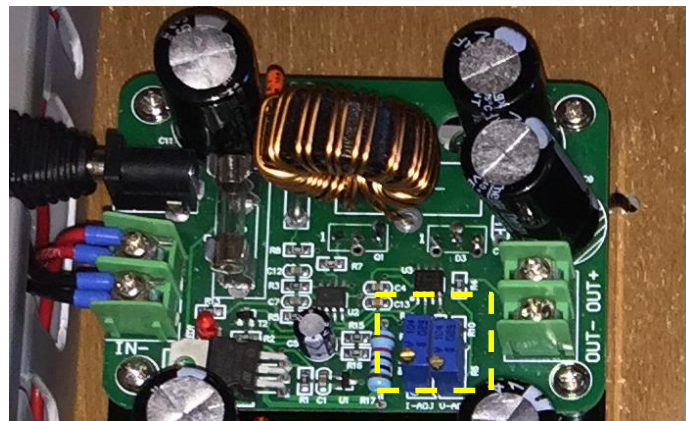


Figure 5-6 – Boost Converter, Highlighted Are The Potentiometers for Current and Voltage.

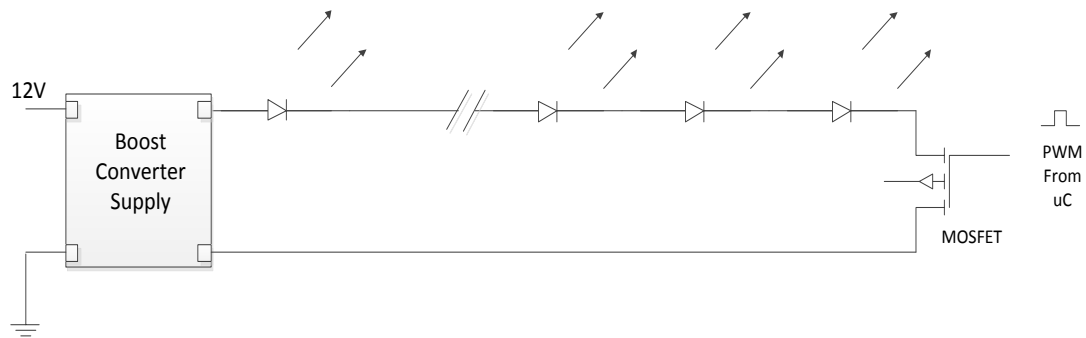


Figure 5-7 – LED Circuit Wiring Diagram.

5.4 Control

Control of the LED lighting circuits is an essential component of the lighting as it enables the lighting to be simulated depending on the time, date and user selections. This section will cover the hardware used for control along with an overview of the programming structure.

5.4.1 Control Hardware

As mentioned in section 5.3 each LED circuit is controlled via a MOSFET which is in turn controlled by a PWM signal from the PiLeven. The PWM signals are provided by four (490Hz) PWM pins on the PiLeven, one for each circuit. The control signals transmitted by the PiLeven are isolated from the MOSFETs by the use of opto-couplers which were deemed necessary as current fluctuations could exceed the max current rating for the PiLeven and in turn damage the microcontroller. A circuit diagram detailing the connection from the microcontroller to the gate of the MOSFET can be seen in Figure 5-8.

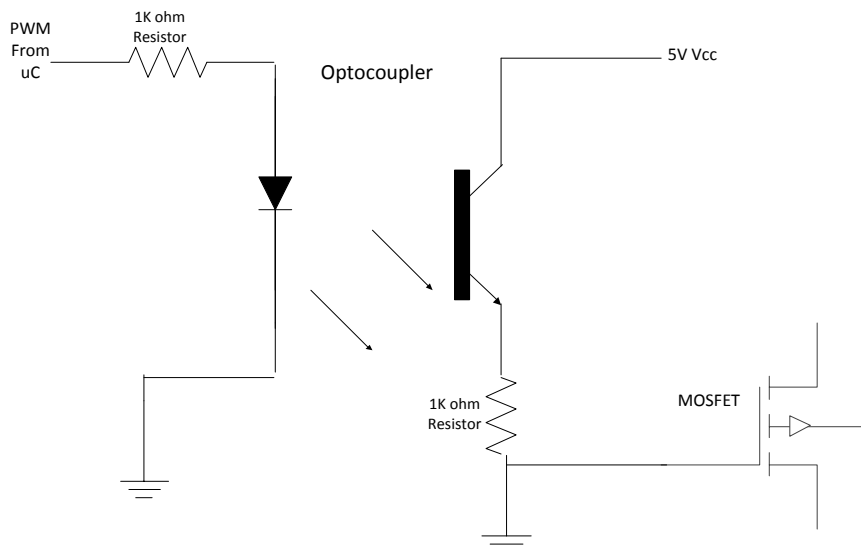


Figure 5-8 - Microcontroller to MOSFET Circuit.

Another essential component for the lighting is the RTC, this enables the microcontroller to have a constantly updating correct time and date. This time and date is then used to calculate the necessary PWM outputs for each LED circuit. The RTC communicates with the microcontroller via I2C which is detailed in Section 8.3.

5.4.2 Program Overview

The control of the lighting is performed using the PiLeven microcontroller with two different operating modes; automatic and manual. In automatic mode the lighting will update every ten minutes. During this update the PWM signal sent to each LED circuit is calculated based on the time and date received from the RTC and the latitude and longitude entered as the location of the reef aimed to be simulated. In order to replicate the sun light during the day and lunar light at night, two separate functions were created in the program these being, Daylight and lunar cycle. As detailed in Table 3.2, the PiLeven specifications, the value sent to any PWM pin must be between 0 and 255 and therefore the value calculated by both of these functions is an integer up to 255 based on the percentage light required.

Automatic - Daylight Function

The daylight function has been adapted from an open source function developed by J. Harp used to replicate sunrise and sunset (Harp, 2013). The function calculates the duty cycle required for the royal blue, cool white and red LED circuits. The sunrise and sunset times are accurate to the location entered in as latitude and longitudes. An example sunrise showing the time, date and duty cycle in percentage of each LED circuit can be seen in Table 5.4 with the location set to the Ningaloo Reef.

Table 5.4 – Duty Cycle Values for each LED Circuit Associated With Day Light Recorded Every 10 Minutes to Show Sunrise (Note date was set to January in RTC until later changed).

Time of Day	Date	Duty Cycle %		
		Royal Blue	Cool White	Red
5.41am	4/1/2015	0	0	0
5.51am	4/1/2015	4	2	2
6.01am	4/1/2015	12	8	8
6.11am	4/1/2015	20	13	13
6.21am	4/1/2015	28	18	18
6.31am	4/1/2015	36	24	24
6.41am	4/1/2015	44	29	29
6.51am	4/1/2015	56	37	37
7.01am	4/1/2015	64	43	43
7.11am	4/1/2015	72	48	48
7.21am	4/1/2015	80	54	54
7.31am	4/1/2015	88	59	59
7.41am	4/1/2015	100	67	67
7.51am	4/1/2015	100	72	72
8.01am	4/1/2015	100	78	78
8.11am	4/1/2015	100	83	83
8.21am	4/1/2015	100	89	89
8.31am	4/1/2015	100	97	97
8.41am	4/1/2015	100	100	100

Automatic - Lunar Cycle Function

The lunar cycle function operates differently to the daylight function as the lunar rise and set has not been taken into consideration at this stage, only the phase of the moon based on the date. This means the moon lighting turns on with the associated duty cycle percentage according to the phase at night and then turns completely off during the day. The lunar cycle function controls the combined violet and green LED circuit, which is now referred to as the moon light circuit. In Table 5.5 the percentage of duty cycle can be seen for each day of a half lunar cycle, from new moon to full moon in October 2015.

Table 5.5 – Duty Cycle Values for Different Moon Phases.

Date	Lunar Phase	Duty Cycle %
12/10/2015	New Moon	0
13/10/2015		1
14/10/2015		1
15/10/2015		4
16/10/2015		8
17/10/2015		14
18/10/2015		22
19/10/2015		31
20/10/2015	First Quarter	50
21/10/2015		52
22/10/2015		62
23/10/2015		73
24/10/2015		83
25/10/2015		91
26/10/2015		97
27/10/2015	Full Moon	100

Manual

In manual mode the Daylight and lunar cycle functions do not run, instead the duty cycle of each LED circuit can be set via the client webpage and uploaded to the server. The duty cycle information will then be transmitted from the server to the microcontroller. This communication will be detailed further in Chapter 8.

5.5 Testing

Throughout the build process and completion of the LED lighting a number of tests were carried out to ensure its correct operation. This section will detail three of these tests targeting the RTC, LED control and 24/7 operation.

RTC test

The RTC test involved initialising the RTC's time and date, then writing a test program that would print the time and date to the terminal of the Arduino IDE every second. This data was then monitored over a 24 hour period to ensure the time remained in sync. Additionally the power was removed from the RTC to confirm that the battery would hold the correct time and date.

LED control test

Upon completion of the wiring for all LEDs, converters and MOSFETs a test program was written to cycle through different duty cycles for each circuit to ensure that the lights operated as expected. This test was initially undertaken before the addition of the isolating opto-couplers and it was found that the program would stop upon high duty cycles of the higher power LED circuits. This fault was accredited to the microcontroller shutting down in order to protect itself from the effects of over current. The opto-couplers were then added and the LED circuits were tested across all duty cycles simultaneously and individually.

24/7 operation test

The final test to be carried out on the lighting system was a test to ensure that the automatic mode worked correctly. This involved checking the sunrise, sunset and lunar lighting. The test was undertaken by printing the percentage duty cycle for each LED circuit to the Arduino IDE every 10 minutes and visually observing both the sunrise and sunset, ensuring that it accurately replicated each. This test was carried out on numerous days to ensure the lighting remained in sync.

Chapter 6 : Water Parameter Monitoring and Control

6.1 Overview

The monitoring and control of water parameters is one aspect of this project that advances it from previously developed domestic reef aquariums. This chapter will detail the methods utilised to monitor and control the selected water parameters. All monitoring and control of the selected water parameters is undertaken in the sump tank as described and illustrated in the overview and layout sections of Chapter 1. These parameters are covered in four sections of this chapter; Temperature in Section 6.2, pH in Section 6.3, Dissolved oxygen in 6.4 and Salinity in Section 6.5.

6.2 Temperature

The correct water temperature is crucial for the survival of marine life and is therefore a necessary parameter to monitor and control. The aims for the temperature monitoring and control system are to be able to select a desired temperature, have the system maintain this temperature continuously without large fluctuations and have the ability to monitor this temperature via a serial monitor or webpage. This section describes the approach and implementation of both the monitoring and control of the water temperature. Further details into the importance of water temperature for aquariums can be found in Appendix 1.

6.2.1 Monitoring

To accurately measure the temperature of the entire system it was decided that, in addition to the temperature sensor located in the sump, another temperature sensor would be needed in the main tank to ensure large discrepancies in temperature can be seen. Therefore, the water temperature monitoring for the system utilises two DS18B20 water proof temperature sensors detailed in Section 3.2.5 of Chapter 3. These two sensors communicate using a 1-wire bus. The

circuit diagram for the temperature sensors can be seen in Figure 6-1. More detail on the 1-wire bus protocol can be found in Chapter 4, Section 4.4.2.

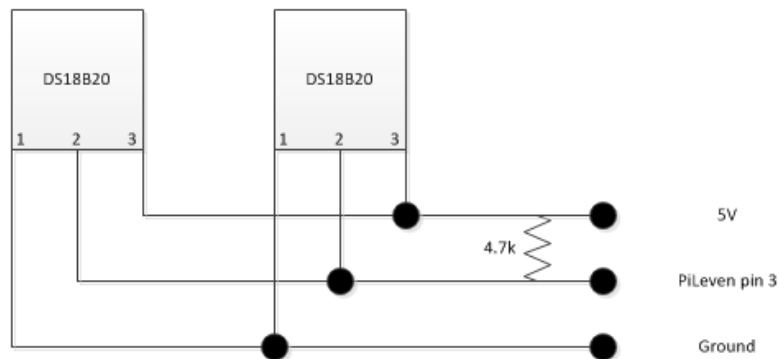


Figure 6-1 – DS18B20 Temperature Sensor Circuit Diagram For One-Wire Bus.

In order to communicate with the temperature sensors on the 1-wire bus, the address of each sensor was needed. To find the addresses an open source sample address finder script (Maxim Integrated, 2008) for Arduino was used. This script was used to great affect and both 64 bit addresses were found. Communication between the sensors and the microcontroller was now possible and an appropriate code was written to obtain the temperature of the sump and the main tank every ten minutes (arbitrarily selected for convenience). The temperatures are then monitored using either a serial monitor or webpage. Further details on the webpage used to monitor the temperatures can be found in Chapter 8, Section 8.5.2.

6.2.2 Control

The approach taken to control the water temperature of the system was to heat the water using two submersible glass heaters. The heaters, detailed in Chapter 3 Section 3.2.5, are plugged into two separate 240V outlets which are controlled via two of the relays on the 8 channel relay module. The associated relays are in turn controlled from two output pins of the microcontroller. Using the monitored temperatures from both the sump and the main tank an average of the two is taken. A program was written using this average and an 'if, then and else' statement, to determine how many heaters should be operating at any time to maintain the

required average temperature. An overview of the heater control from microcontroller to 240V outlets can be seen in Figure 6-2.

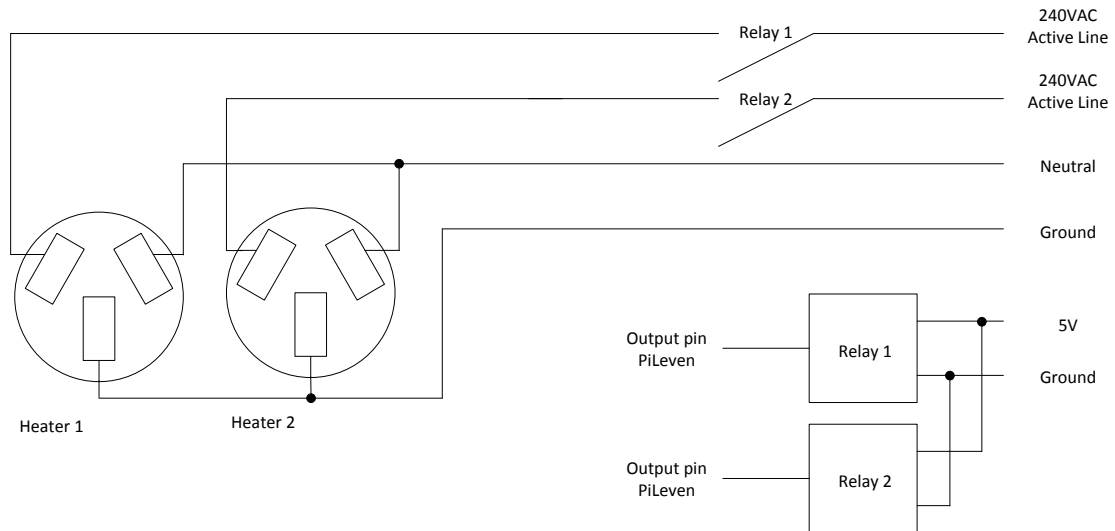


Figure 6-2 – Circuit Diagram for Controlled Heater Outlets.

A snippet of the heater control pseudocode written in Arduino can be seen in Figure 6-3. The program operates by turning the associated heater outputs either HIGH or LOW depending on the result of comparisons between the average temperature and the desired temperature. This pseudocode was then adapted into the main project code with minor changes.

```

//Heaters
//Heater 1
AvTemp = (TempTank + TempSump)/2
if ("heating on from webpage" == on && AvTemp < 27){
digitalWrite(HEATER_1, HIGH)
}
else if(HEATER_1 == on && AvTemp < 24){
digitalWrite(HEATER_2, HIGH)
}
}
Else {
digitalWrite(HEATER_1, LOW)
digitalWrite(HEATER_2, LOW)
}
}

```

Figure 6-3 Pseudocode for Heater Control.

6.3 pH

With the importance of pH for aquariums detailed in Appendix 1, this section will cover the methods for monitoring and control of the pH that have been implemented in this project. The aims for monitoring and control of the pH were to be able to set the desired pH, to have the system maintain the pH without large fluctuation and to have the ability to monitor the pH levels via a serial monitor or webpage.

6.3.1 Monitoring

In order to monitor the pH levels of the system a pH probe and pH EZO circuit, detailed in Chapter 3 Section 3.2.3, was installed with the probe located in the sump tank and the pH EZO located in the EZO junction box.

Setup

The first step in setting up the pH probe and EZO circuit involved downloading specific sample code to communicate with the sensor via the Atlas Scientific website Figure 6-4. As mentioned in Section 3.2.3 the pH EZO circuit can operate in either UART or I2C mode. According to the data sheet of the pH EZO circuit, the default mode was UART and in order to change this, the UART sample code was required and a command was necessary to switch it to I2C. This command was:

I2C,[99]

This command performed the task of setting the device to I2C mode and additionally setting the I2C address which in this case was 99. Following this it was possible to communicate with the device via I2C.

In I2C mode there are a number of specific commands listed in the pH EZO datasheet that were necessary for communicating with the device. The main commands utilised in this project can be seen in Table 6.1 with a description of each.

Table 6.1 – I2C Mode Commands Utilised.

Command	Function
Cal	Performs calibration
R or r	Returns a single reading
Status	Retrieve status information

Calibration

Upon setting the pH EZO circuit to I2C mode the I2C sample code was utilised to carry out the calibration. The pH EZO circuit can be calibrated using ‘single point’, ‘two point’ or ‘three point’ calibration with the only requirement being that pH 7 calibration is performed first. It was decided for this project ‘three point’ calibration would be performed since this would provide the most accuracy in pH measurements. The initial pH 7 calibration was undertaken using the following steps.

- The probe was placed into a pH 7.00 calibration liquid.
- The following command was entered: Cal,mid,7.00
- After waiting the 1.4 seconds calibration time the probe was removed and dried.

These steps were repeated for low and high calibration which used calibration solutions of 4.00 and 10.00 respectively.

Programming

With the calibration complete, the I2C sample code (Atlas Scientific) supplied by Atlas Scientific was now modified to suit the main project code. The sample code was modified into a function, seen in Figure 6-4, and operates using the I2C library (Wire.h) developed by Arduino (Arduino, 2015). Utilising the inbuilt library commands of the Wire library the function begins a transmission with the pH EZO circuit; followed by the transmission of the command ‘r’ to the pH EZO circuit. This transmission is then ended and upon waiting the appropriate time, a request is sent to the pH EZO circuit for 20 bytes. This data is then loaded into an array which is then

printed to the serial monitor and converted to a floating point number for use in the control aspect. This function runs every 10 minutes to retrieve the pH level of the water and can be monitored via the serial monitor and the webpage as discussed in future works.

```

//-----
//-----pH_reading_function-----
//-----
void ph_reading() {

commanddata[0] = 'r';           //character 'r' to be sent as command to ph EZO circuit.
time=1400;                     //wait time is 1400ms for reading data
Wire.beginTransmission(ph_address); //call the circuit by its ID number.
Wire.write(commanddata);      //transmit computerdata through serial port.
Wire.endTransmission();       //end the I2C data transmission.
delay(time);
Wire.requestFrom(ph_address,20,1); //call the circuit and request 20 bytes
  while(Wire.available()){     //are there bytes to receive.
    in_char = Wire.read();     //receive a byte.
    ph_data[i]= in_char;      //load this byte into our array.
    i+=1;                     //incur the counter for the array element.
    if(in_char==0){          //if 0 then null command.
      i=0;                    //reset the counter i to 0.
      Wire.endTransmission(); //end the I2C data transmission.
      break;                  //exit the while loop.
    }
  }
  Serial.print("pH:");
  Serial.println(ph_data);    //print the data.
  ph_float=atof(ph_data);    //converts ph data to floating value
}

```

Figure 6-4 – pH Reading Function.

Testing

Upon incorporating this function into the main program the updated program was tested by running and monitoring the pH levels over a full day. Since the pH reading was printed to the serial monitor every 10 minutes it was possible to ensure communications to the pH EZO circuit were occurring correctly. The function was deemed operational and has been running since. In addition to this, a home pH tester was used at appropriate times and the two readings were compared to ensure the values did not differ over time. The comparison proved the correct pH readings and can be seen in Table 6.2. For more details on the home pH test undertaken, see Appendix 4.

Table 6.2 – pH Test Results, Comparison Between Home Tester and pH Probe.

Date	Time	pH (Probe)	pH(Home Tester)
20/10/2015	7.33am	8.506	8.5
20/10/2015	7.43am	8.517	8.5
20/10/2015	7.53am	8.505	8.5
20/10/2015	8.03am	8.517	8.5
20/10/2015	8.13am	8.516	8.5
20/10/2015	8.23am	8.517	8.5
20/10/2015	8.33am	8.520	8.5

6.3.2 Control

When considering the methods for controlling the pH it was found that, in reef aquariums, the pH is far more likely to drop than to rise (Reef Central, 2008). Taking this into account it was decided that a dosage pump would be used to feed pH balancer/buffer into the tank to increase the pH when necessary. An overview of this procedure can be seen in Figure 6-5. The dosage pump detailed in Chapter 3 Section 3.2.5 is controlled using one of the relays on the 8 channel relay module. The dosage pump has been wired and can be seen in Figure 6-6 next to the dosage pump used for salinity control.

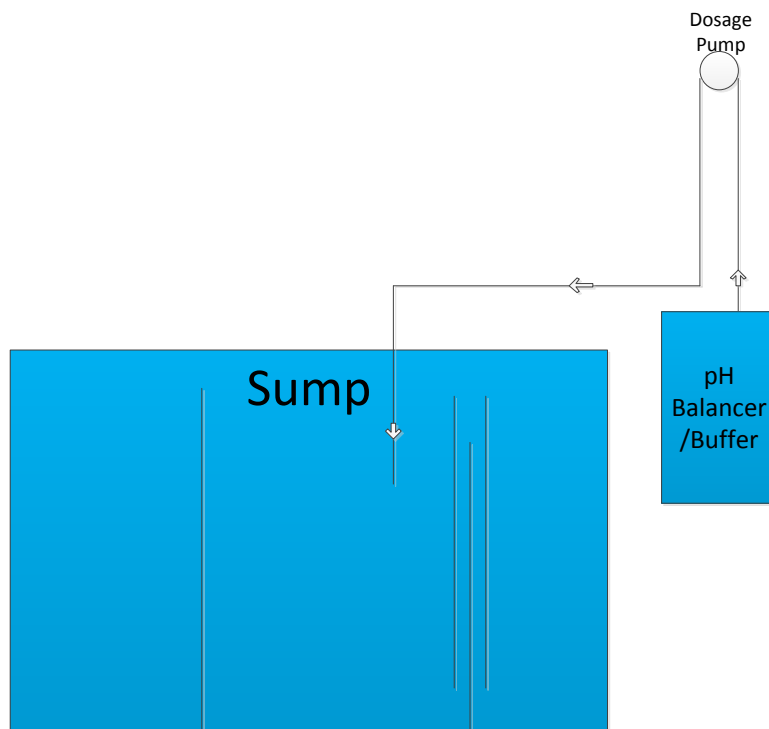


Figure 6-5 – Overview of pH Control.



Figure 6-6 – pH Dosage Pump (left), Fresh Water Dosage Pump (right).

Programming

The concept for controlling the dosage pump is to compare the pH reading to the pH set point and when the two differ by 0.3 or more the dosage pump will run for the appropriate time, dosing the water with the necessary amount of pH balancer/buffer. The appropriate time for the dosage pump to run will be calculated from the volume of the tank, the increase in pH needed (in this case 0.3) and the instructions of the buffer solution. At the completion of this thesis this program has only been written in pseudocode and will be further detailed in the future works in Chapter 9.

6.4 Dissolved Oxygen

With the importance of dissolved oxygen for marine life detailed in Appendix 1 this section will cover the methods for monitoring and control of the dissolved oxygen that have been implemented in this project. The aims in monitoring and controlling the dissolved oxygen were, to have a set dissolved oxygen level that the system can maintain, to provide two operation modes (Automatic and Waves) and finally, to have the ability to monitor the dissolved oxygen levels via a serial monitor or webpage.

6.4.1 Monitoring

The monitoring of the dissolved oxygen is performed using the dissolved oxygen sensor and DO EZO circuit detailed in Section 3.2.4. In order to communicate with the DO EZO circuit the same process performed for the pH EZO was undertaken, which was to first communicate via UART

mode, send the appropriate command to switch to I2C mode and then calibrate the probe and DO EZO using the I2C sample code (Atlas Scientific) for the DO sensor from the Atlas Scientific website and the same I2C commands listed in Table 6.1.

Calibration

To perform the calibration of the dissolved oxygen probe and DO EZO circuit the probe required two point calibration. This was performed using the following steps:

- The probe was placed in the air for atmospheric calibration
- The following command was entered: Cal
- The probe was then placed into 0 dissolved oxygen calibration solution
- The following command was entered: Cal,0

At the completion of these steps the dissolved oxygen probe and DO EZO circuit were accurately calibrated.

Programming

The program written to monitor the dissolved oxygen uses two functions, the first function has a structure identical to that of the pH function except that the data received has both dissolved oxygen and oxygen saturation in a CSV string. To deal with this a second function was created, as seen in Figure 6-7, which breaks up the CSV string into its separate readings allowing them to be individually printed and converted to floating point values for use in the control portion of the program. The two functions run every 10 minutes and the dissolved oxygen measurement can be read via the serial monitor or webpage.

```

void string_pars(){ //function breaks up CSV strin into DO & Sat
                    //this is done using the C command "strtok".

    sat=strtok(DO_data, ","); //seperates the string at each comma.
    DO=strtok(NULL, ","); //seperates the string at each comma.

    Serial.print("DO:");
    Serial.println(DO); //this is the D.O. value.

    Serial.print("Sat:");
    Serial.println(sat); //this is the % saturation value.

    DO_float=atof(DO);
    sat_float=atof(sat);

} |

```

Figure 6-7 – String Pars Function, Separates CSV String.

Testing

Upon incorporating these two functions into the main program the updated program was tested by running and monitoring the dissolved oxygen levels over a full day. Since the dissolved oxygen reading was printed to the serial monitor every 10 minutes it was possible to ensure communications to the DO EZO circuit were occurring correctly. The function was deemed operational and has been running since.

6.4.2 Control

To increase dissolved oxygen levels in an aquarium a number of approaches exist, the most effective methods include, correct water plants, an air pump delivering oxygen into the water and increasing the surface area of the water via wavemaker power heads. The approach to controlling the dissolved oxygen in this project was by means of the wavemaker power heads since this option was easily controllable and would provide an additional means of mixing the water to achieve uniform water parameters throughout the entire system. It was decided that only one wavemaker power head would be necessary for the tank used in this project and this power head would be controlled using one of the relays from the 8 channel relay module detailed in Section 3.2.5.

Program

The program to control the dissolved oxygen and hence the wavemaker power head operates in two modes, Automatic and Waves. The Automatic mode is responsible for ensuring the dissolved oxygen levels of the system remain at the level set. This is performed by pulsing the wavemaker power head when dissolved oxygen levels begin to drop. Upon the dissolved oxygen decreasing below the set value the associated output for the powerhead is toggled HIGH for one second and LOW for two seconds (selected as it provided the highest water turbulence possible on the surface). This continues until the dissolved oxygen levels return to the set value. The second mode of operation, Waves, is an aesthetic operation mode as it does not operate based on the dissolved oxygen levels of the system but rather pulses at set intervals to create visually pleasing water movement in the tank.

Testing

To test that this method of dissolved oxygen control would operate as expected a test solution was created. This solution was made using the zero oxygen calibration solution mixed with ordinary tap water to create a solution with a dissolved oxygen level just below the set value which in this case was 6 parts per million (ppm) as suggested by (Atlas Scientific). The probe was placed in this solution to ensure that the wavemaker power head would begin to pulse. The probe was then removed from the solution and since it was now in air the oxygen level increase beyond the set value. The wavemaker power head was then monitored to ensure it switched back off. This test confirmed the correct operation and so the probe was then moved to the sump and the system set to operate in Automatic mode. This has been monitored by reading the dissolved oxygen levels printed to the serial monitor over a number of days. The dissolved oxygen level did not drop significantly during daily operation and so the dissolved oxygen control was deemed to be working correctly.

6.5 Salinity

Since the entire system is based on a marine ecosystem, salinity is a necessary parameter to monitor and control. Greater detail on the importance of salinity for marine aquariums can be found in Appendix 1. The aims of monitoring and controlling the salinity of the system were to be able to set the desired salinity level, to have the system maintain the salinity without large fluctuations and to have the ability to monitor the salinity levels via a serial monitor or webpage.

6.5.1 Monitoring

The monitoring of salinity level for the system is performed using the electric conductivity EC probe and EC EZO circuit detailed in Section 3.2.5. Similar to the pH and dissolved oxygen probes and circuits it was necessary to first calibrate the EC probe and EZO circuit.

Calibration

In the same manner as the two sensors detailed in Section 6.3 and 6.4 the EC probe and EZO circuit required being set to I2C mode and then the calibration performed. This calibration was performed using the appropriate calibration solutions and the I2C commands listed in Table 6.1.

The steps performed are as followed:

- The probe left dry for a dry calibration
- The following command was entered: Cal,dry
- The appropriate 1.3 seconds calibration time was waited
- The probe was then placed into the low electrical conductivity calibration solution
- The following command was entered: Cal,low,12880

The last three steps were then repeated for the high calibration using the calibration solution containing 80000 μ S.

Programming

The program used to receive the salinity from the EC probe and EC EZO circuit is structured in the exact same way as the pH and DO sensors with one small difference, the bytes request is now 48. Similar to the dissolved oxygen sensor the EC sensor required a second function to split up a CSV string returned by the EC EZO circuit and it is the size of this CSV string that causes the bytes requested to increase. The CSV string produced by the EC EZO circuit is larger than that of the previous sensors as it contains the readings of Electric Conductivity, Total Dissolved Solids, Salinity and Specific Gravity. These are all different parameters that electrical conductivity can provide and this sensor performs the calculations within the EC EZO circuit. It was decided that for this project both the salinity and specific gravity would be monitored since both measurements are regularly used by aquarists. Using the same method detailed for the dissolved oxygen sensor a second function was written to split this CSV string and the necessary measurements are sent to the serial monitor and webpage every 10 minutes.

Testing

The testing of the salinity monitoring was undertaken by incorporating the salinity functions into the main program and running the system continuously. The Specific Gravity was monitored via the serial monitor over a period of two days and it was found that the levels matched those taken using the home tester kit at appropriate times. The test comparisons can be seen in Table 6.3. For more details on the home salinity test, see Appendix 4.

Table 6.3 – Specific Gravity Test Results, Comparison Between Home Tester and EC Probe.

Date	Time	(Home Tester) Specific Gravity	(EC Probe) Specific Gravity
20/10/2015	6.37am	1.014	1.013
20/10/2015	6.47am	1.014	1.014
20/10/2015	6.57am	1.014	1.014
20/10/2015	7.37am	1.011	1.012
20/10/2015	7.47am	1.011	1.011
20/10/2015	7.57am	1.011	1.011
20/10/2015	8.07am	1.011	1.011

6.5.2 Control

The approach taken to control the salinity of the project was similar to that of the pH. Due to the fact that the salinity levels would most likely rise as fresh water evaporates from the tank, it was decided that a dosage pump would be utilised to supplement the tank with fresh water when necessary. The dosage pump is setup in the same manner as the dosage pump used to control the pH, using a relay from the 8 channel relay module which is controlled via the microcontroller. This pump has been wired and can be seen in Figure 6-6 of Section 6.3.2. Of the two measurements monitored using the EC sensor it was decided that, for controlling the salinity, the units of 'parts per thousand' (PPT), in relation to salinity, would be used rather than the units of 'kg/m³' relating to specific gravity.

Programming

The concept for controlling the dosage pump was to compare the salinity level in PPT of the system to the desired salinity level. When the salinity in the tank rises above this set point by 2 PPT the dosage pump runs for the appropriate amount of time (which is to be determined during testing) to supplement the tank with fresh water. At the completion of this thesis this program has only been written in pseudocode and will be further detailed in the future works in Chapter 9.

Chapter 7 : System Monitoring and Control

7.1 Overview

In order to continuously operate the aquarium in a safe and reliable manner the need for system monitoring and control is absolutely essential. This chapter will detail the different components that have been necessary to monitor and control with Section 7.2 covering the water levels and pumps and Section 7.3 detailing system temperature and cooling.

7.2 Water Levels and Pumps

Monitoring the water levels of the system is of utmost importance as a number of issues could arise that could cause the following problems: main tank overflow, return pump running dry or the water circulation stopping. To accurately measure the water levels of both the main tank and the sump tank, two ultrasonic ranging sensors detailed in Section 3.2.5 have been utilised.

7.2.1 Main Tank Monitoring

The ultrasonic sensor used to monitor the water level of the main tank can be seen in Figure 7-1. The ultrasonic sensor is set to check the distance to the water as detailed in Section 3.2.5 every second. Based on the distance, two possible conditions are monitored, these being high and low.



Figure 7-1 – Ultrasonic Sensor Over Main Tank.

High Water Level

The first condition is a high water level in the main tank. If the distance between the ultrasonic sensor and the water is below the chosen threshold of 5cm for 10 consecutive distance checks the system will consider this as a high water level in the main tank. When the high water level condition occurs the return pump will switch off to ensure the water does not overflow the tank. The control over the return pump is provided by use of one of the relays on the 8 channel relay module which is controlled via the microcontroller. Since the return pump is 240V AC the relay controls one of the 240V outlets highlighted in Figure 7-2.



Figure 7-2 – Return Pump Controlled 240V Outlet.

Low Water Level

The second condition is a low water level which would be due to either a leak or water evaporation. This condition is met when the distance to the water is greater than 10cm for 10 consecutive distance checks and will also turn off the return pump. This condition turns off the return pump because the water siphon from the main tank to sump tank does not operate when the water level is below the siphon inlet.

7.2.2 Sump Tank Monitoring

The second ultrasonic sensor is used to monitor the water level of the sump tank. The sensor, seen in Figure 7-3, measures the distance to the water of the final section of the sump tank,

where the return pump sits. If the distance to the water in this section falls below 20 cm for more than 10 seconds then the return pump will be switched off to ensure it does not run dry. The program for the sump tank level is incorporated in the main tank level control program detailed above.



Figure 7-3 – Ultrasonic Sensor Over Sump Tank.

7.2.3 Programming and Testing

The water level monitoring and pump control for both the main tank and sump tank detailed above was performed by creating a function that operates every one second, checking for the conditions and implementing the appropriate actions. Upon including the function into the main program, the conditions were simulated by adding or siphoning water. It was found that all conditions performed the correct actions when necessary, indicating that the system could operate safely.

7.3 Cooling

There are two areas of the project that required cooling, these being the lighting hood and the electrical draw. As mentioned in Chapter 5 the lighting hood is the compartment in which the LEDs are contained. Due to the heat produced by the LEDs while operating it was necessary to incorporate an additional form of heat extraction aside from the heat sinks detailed in Appendix

1. To achieve this, exhaust fans, as seen in Figure 7-4, were utilised and are controlled via one of the relays in the 8 channel relay model and are in turn controlled from the microcontroller. The exhausts fans are programmed to operate when both the blue and cool white LEDs are operating above 50% duty cycle as this when the LEDs produce the most noticeable heat.



Figure 7-4 – Controlled Exhaust Fans at the Top of Lighting Hood.

The second area requiring exhaust fans was the electrical draw, which as detailed in Section 2.4, contains the majority of the power and control devices. Since the heat produced in this electrical draw is relatively constant it was decided that the exhaust fans for this area would be on permanently rather than controlled. The exhaust fans can be seen in Figure 7-5.



Figure 7-5 – Exhaust Fans for Cooling the Electrical Draw.

Chapter 8 : Network and Communications

8.1 Overview

Device communication is a fundamental aspect of this project with a number of different device types, protocols and programming platforms needed to communicate. This chapter will give a brief introduction to how the controlling devices communicate within the project. Section 8.2 gives a network overview of the entire system, Section 8.3 covers the use of I2C between the microcontroller and sensors, Section 8.4 covers the use of serial communications between the microcontroller and microprocessor and Section 8.5 will introduce the Raspberry Pi constituents including the LAMP server and webpages.

8.2 Network Overview

The network overview of the system, seen in Figure 8-1, illustrates the communications and protocols utilised, along with the topology of the network. It can be seen that the I2C bus topology is a bus whereas the serial communication is from point to point.

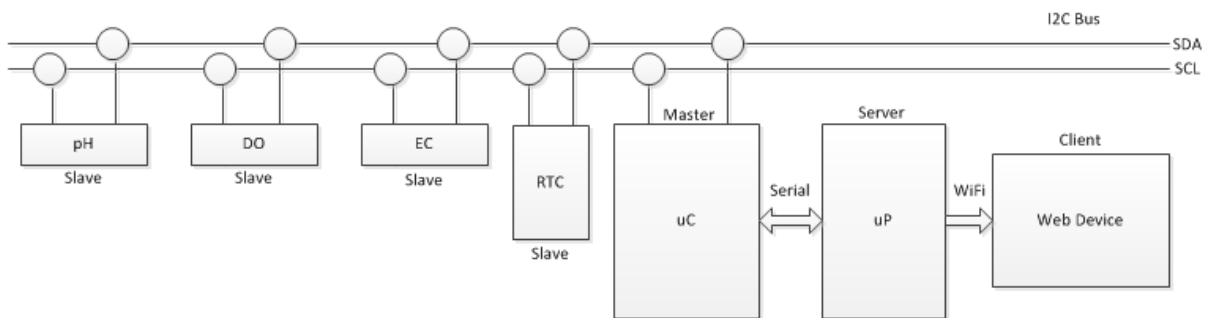


Figure 8-1 – Network Overview.

8.3 I2C Network

The I2C network in this project consists of one master and 4 slaves. The master is the PiLeven microcontroller and the slaves are; the pH EZO circuit, DO EZO circuit, EC EZO circuit and RTC. The communication between the master and the slave is facilitated by a pre written library by Arduino, known as the wire.h library (Arduino, 2015). This library contains a number of pre-built functions enabling the transmission and request to I2C devices from the master. Using these

functions the master polls the RTC at the clock speed of the PiLeven to maintain correct timing and the other three slaves are polled every 10 minutes at separate times. The I2C network has been tested thoroughly by printing the readings returned by each slave device at their appropriate poll times along with the date and time polled. The results to two hours of this test can be seen in Appendix 4 and from these results it was concluded that the I2C network was operating as required.

8.4 Serial Communications

Communication between the Raspberry Pi and the PiLeven is necessary for monitoring the data of all measurements and controlling any set points. At this point in the project the serial communication has been used to send all measurement data to the Raspberry Pi, which is then read using a serial monitor. Since the aim is for the final product to be monitored and controlled via a server/client setup, with the client being a webpage, the program will need to be adapted to incorporate this. Details on the serial to server/client communication will be given in Chapter 9, Future Works.

8.5 Raspberry Pi Constituents

As described in the project objectives, the entire system will eventually be monitored and controlled from a server/client setup. This was deemed to be beyond the time frame of this project but since it is still a necessary objective of the work, this section will detail the approach that has been researched and partially implemented including the server/client and the webpages.

8.5.1 Server/Client

After considerable research it was decided that a LAMP server (Beal, LAMP, 2012) would be a suitable option for the project. Short for Linux, Apache, MySQL and PHP, LAMP is an open source web development platform or web stack which constitutes the elements necessary for a server. These elements are; Linux as the operating system, Apache as the web server, MySQL as the

relational database management system and PHP, Python or PERL as the object-orientated scripting language (Beal, LAMP, 2012). The Raspberry Pi's operating system is Raspbian which is Linux-based so the next step in setting up the LAMP was to install Apache, MySQL and PHP. This was performed using the LXTerminal and appropriate commands. With these elements installed it was now possible to create the necessary webpages for the server and setup the communication between the server and the PiLeven. As mentioned earlier this communication has not yet been completed and will be discussed in Chapter 9.

8.5.2 Webpages

The webpages for monitoring and controlling the system were created using HTML for the body and CSS for the formatting with the assistance of w3school, a tutorial website (w3schools, 2015). Since the requirements for controlling and monitoring the system were known, it was possible to design the webpages around these requirements. There are four webpages in total, these are: Home, System Overview, Water Parameters and Lighting. The Home page, as seen in Figure 8-2, is used as an information page and does not provide any control or monitoring. The System Overview page, seen in Figure 8-3, will provide monitoring of the tank levels and control over the pumps. The Water Parameters page, seen in Figure 8-4, allows the selection of manual or automatic water parameter control, monitoring of all water parameters and slide bars to select the set points of each water parameter. The final page is the lighting control and monitoring page, seen in Figure 8-5. This page allows the selection of either manual or automatic mode for the lighting, monitoring of the current light intensities and slider bars for each LED circuit when in manual mode. At this stage these webpages are only templates and will require modifying when the PiLeven to server communication is complete, see Chapter 9.

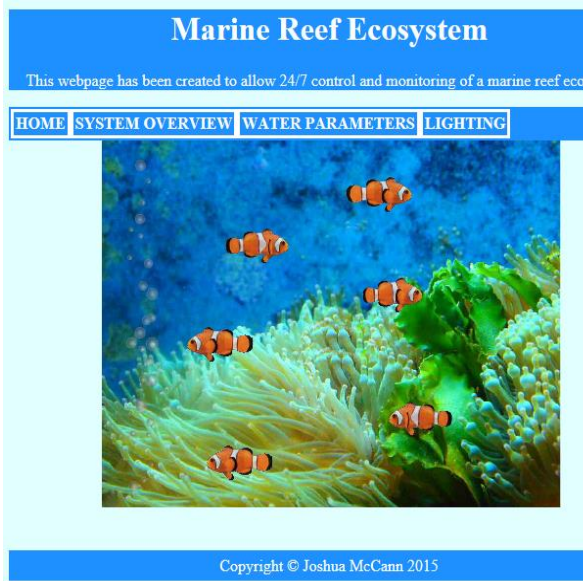


Figure 8-2 - Home Page.



Figure 8-3 - System Overview Page.

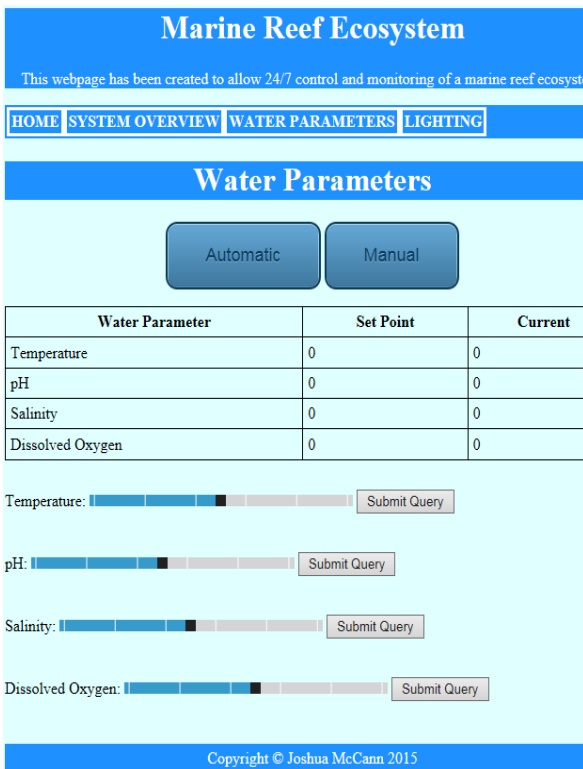


Figure 8-4 - Water Parameters Page.

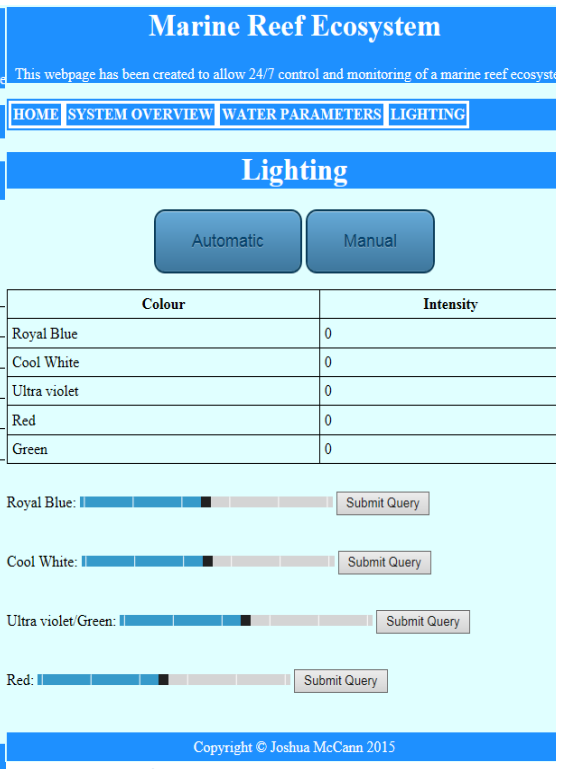


Figure 8-5 - Lighting Page.

Chapter 9 : Future Works

9.1 Overview

Given the size and complexity of this project a number of tasks were unable to be completed. This chapter will detail the major works necessary to complete the project in Section 9.2 and run through of suggested improvements and future developments for the project in Section 9.3.

9.2 Uncompleted Works

This section details the major works necessary for the project to be considered fully complete. Not detailed in this section are the exact steps required to carry out these works.

9.2.1 pH and Salinity Control

As mentioned in Sections 6.3.2 and 6.5.2, at the completion of this thesis the control for both the pH and salinity was not yet complete. For both the pH and salinity the pseudocode for controlling the dosage pumps needs to be incorporated into the main program. Additionally the dosage pumps must be tested to find the exact flow rate of each and from this, the time each pump should run to deliver the necessary liquid. Upon the completion of this the control for all water parameters will be complete.

9.2.2 Microcontroller to Server Communication

The communications between the Pileven and the server is yet to be implemented. At the completion of this thesis, the LAMP server components have been installed as detailed in Section 8.5. Therefore, the next step is to write a script using Python which will read the serial communications and store them in the MySQL database. Then, using another PHP script, the values stored in the database can be retrieved and displayed on the server and hence on the webpages. Additionally these scripts would take the control actions performed on the webpages and store them into the database. This control information would then be taken from the database using another Python script and sent to the Pileven via the serial communication.

9.3 Suggested Improvements and Future Development

This section lists, in Table 9.1, the suggested improvements and future developments to the project with a description of each improvement and the benefits each would provide.

Table 9.1 – Suggested Improvements and Future Developments.

Improvement/Addition	Description	Benefits
Additional pH control	At the completion of this project, including the unimplemented works detailed in section 9.2.1, the pH control is only capable of increasing the pH. In some instances the pH could become too high. Therefore an additional dosage pump could be installed to dose the water with a suitable acid solution.	This would account for both increase and decreases of pH in the system, making the system more robust.
Additional salinity control	At the completion of this project, including the unimplemented works detailed in section 9.2.1, the salinity control is only capable of decreasing the salinity. Therefore an additional dosage pump could be installed to dose the water with salt water.	This would account for both increase and decreases of salinity in the system, making the system more robust.
Additional dissolved oxygen control	To increase the level of dissolved oxygen an air pump could be added to pump air into the sump tank as required.	This would allow higher levels of dissolved oxygen to be reached which also increases the number of fish the tank can hold.
Weather conditions/seasons	Within the program the lighting could be adapted to account for the average light seasonally, based upon the weather of each season. This would be done using the date received from the RTC.	This addition would make the system lighting more realistic and therefore more suitable for research.
Water Cooling	Since this project was undertaken during winter and spring it is not certain if the water temperature will rise enough to require a water cooler. A water cooler or chiller is a self-contained unit which could be purchased and controlled via a relay when necessary.	This addition would insure temperatures would not become too high.
Logging	Not detailed in this project is the ability to log any of the water parameters. This could be achieved through the use of the MySQL database introduced in Section 8.5.1.	This feature would aid in research, allowing history and records of water parameters over time.
Additional water parameter monitoring	A number of additional water parameters not monitored in this	Increased water parameter monitoring

	project could be included in further developments. Possible parameters include; alkalinity, calcium and/or phosphorus.	would further increase the accuracy of the system in replicating a reef ecosystem.
Automatic top off	In addition to the salinity control tanks, many systems have what is known as an automatic top off. An automatic top off is a tank that feeds directly into the system when the water level becomes too low due to evaporation. This would use the level sensors and an additional tank and pump. The tank would contain seawater to ensure the salinity of the main system is not affected too much.	This addition would save maintenance time and ensure the system is able to run continuously without interruption as the main tank to sump line operates via overflow which would stop operating if the water level became too low.
Fish camera	With the use of the Raspberry Pi a camera could be installed to visually monitor the marine life over time.	This would be particularly useful for research as it would allow time lapse of coral growth under different set conditions.
Further Testing	To ensure the system is sufficiently replicating the environmental parameters expected in a typical reef ecosystem, further tests could be conducted.	Further testing would ensure accurate parameters for marine research.
LED Drivers	The current circuit design may eventually damage the LEDs. To rectify this the MOSFETs can be swapped out for LED drivers.	The LED drivers will regulate the current in-turn ensuring the LEDs operate at expected currents and therefore last longer.

Chapter 10 : Conclusion

The overall objective of the project that made up this thesis was to create a monitoring and control system for a marine ecosystem based within an aquarium, through the use of current technologies in embedded systems. Research into existing monitoring and control systems was undertaken and it was found that many systems lacked the monitoring and control of several important factors for marine ecosystems. Therefore it was decided that the project would aim to monitor and control the following water parameters: pH, dissolved oxygen, temperature and salinity. In addition to this the system would include the system control and monitoring features available in existing models including: lighting, water levels and pumps.

With the objectives and aims clarified it was then possible to select the appropriate hardware. A PiLeven microcontroller was selected to oversee all control and monitoring: a Raspberry Pi as a computer for interfacing with the microcontroller; numerous sensors for water parameters; LEDs for lighting; various pumps for control and a range of power converters for power supplies. The necessary circuits were then designed, based on the approaches for monitoring and controlling each water parameter and system parameters. These designs were then implemented and the system built. Upon completion of the build, numerous software functions were written to control the components via the microcontroller. These functions were first tested individually to ensure correct operation of each component, followed by a test of the combined functions to ensure correct operation of the system as a whole.

At the time of writing this document the control of the pH and Salinity are yet to be fully completed. Despite this the project was considered a success, with the accurate monitoring of all specified water parameters, control of temperature and dissolved oxygen, fully functioning realistic lighting, and monitoring and control of all necessary system parameters. With this success, it is now possible for the system to be used to house a marine ecosystem with additional monitoring and control in comparison to currently existing systems.

Chapter 11 : References

Algone. (2012, August 11th). *Salinity and specific gravity guidelines for the salt water and reef keeper*. Retrieved August 20th, 2015, from Algone for healthy aquariums: <http://www.algone.com/salinity-and-specific-gravity-guidelines-for-the-saltwater-and-reef-keeper>

Aquatic Community. (2004). *Marine Aquariums: pH and Alkalinity*. Retrieved November 3rd, 2015, from Aquatic Community: <http://www.aquaticcommunity.com/marineaquarium/ph.php>

Arduino. (2015). *Arduino Introduction*. Retrieved July 10th, 2015, from <https://www.arduino.cc/en/Guide/Introduction>

Arduino. (2015). *Arduino Time Library*. Retrieved September 9th, 2015, from <http://playground.arduino.cc/code/time>

Arduino. (2015). *Language Reference*. Retrieved October 12th, 2015, from Arduino: <https://www.arduino.cc/en/Reference/HomePage>

Arduino. (2015). *Wire Library*. Retrieved from Arduino.

Atlas Scientific. (2015). *About*. Retrieved October 12th, 2015, from Atlas Scientific: <https://www.atlas-scientific.com/about.html?>

Atlas Scientific. (n.d.). EZO class embedded DO circuit. Atlas Scientific.

Atlas Scientific. (n.d.). EZO class embedded EC circuit. Atlas Scientific.

Atlas Scientific. (n.d.). EZO class embedded pH circuit. Atlas Scientific.

Atlas Scientific. (2015). *pH kit*. Retrieved August 10th, 2015, from Atlas Scientific: http://www.atlas-scientific.com/product_pages/kits/ph-kit.html

Aushulz. (2012, August 2nd). *Chlorophyll ab spectra*. Retrieved October 10th, 2015, from Wikimedia Commons: https://commons.wikimedia.org/wiki/File:Chlorophyll_ab_spectra-en.svg

Beal, V. (2012). *Applications*. Retrieved October 14th, 2015, from Webopedia: <http://www.webopedia.com/TERM/A/application.html>

Beal, V. (2012). *LAMP*. Retrieved August 12th, 2015, from webopedia: <http://www.webopedia.com/TERM/L/LAMP.html>

Christensson, P. (2007, October 11th). *C/C++ Definition*. Retrieved October 29th, 2015, from TechTerms: <http://techterms.com/definition/cplusplus>

Christensson, P. (2006). *CSS Definition*. Retrieved October 29th, 2015, from TechTerms: <http://techterms.com/definition/css>

Christensson, P. (2015, May 23rd). *HTML Definitions*. Retrieved October 29th, 2015, from TechTerms: <http://techterms.com/definition/html>

Cliff. (2013, March 28th). *Common Beginner Mistakes With Marine Aquariums*. Retrieved July 14th, 2015, from Reef Aquarium: <http://www.reefaquarium.com/2013/10-most-common-beginner-mistakes/>

Computer Hope. (2015). *Programming Languages*. Retrieved October 15th, 2015, from Computer Hope: <http://www.computerhope.com/jargon/p/proglang.htm>

Computer Hope. (2015). *Protocol*. Retrieved October 29th, 2015, from Computer Hope: <http://www.computerhope.com/jargon/p/protocol.htm>

CREE. (2015). Cree XLamp XP-E LEDs.

Danmichaelo. (2012, May 16th). *Solar AMO spectrum with visible spectrum background*. Retrieved October 10th, 2015, from Wikimedia Commons: [https://commons.wikimedia.org/wiki/File:Solar_AMO_spectrum_with_visible_spectrum_background_\(en\).png](https://commons.wikimedia.org/wiki/File:Solar_AMO_spectrum_with_visible_spectrum_background_(en).png)

Digital Aquatics. (2015). *Systems*. Retrieved November 2nd, 2015, from Digital Aquatics: <http://www.digitaquatics.com/saltwater/systems/reefkeeper-systems/>

Dilmen, N. (2014, January 8th). *HC SR04 Ultrasonic sensor 1480322 3 4 HDR Enhancer*. Retrieved October 14th, 2015, from Wikimedia Commons: https://commons.wikimedia.org/wiki/File%3AHC_SR04_Ultrasonic_sensor_1480322_3_4_HDR_Enhancer.jpg

ELEC Freaks. (n.d.). Ultrasonic Ranging Module HC-SR04. ELEC Freaks.

freetronics. (2015). *pileven arduino compatible expansion for raspberry pi*. Retrieved July 12th, 2015, from http://www.freetronics.com.au/products/pileven-arduino-compatible-expansion-for-raspberry-pi#.VcP_UfmqpBc

hacktronics. (2012). *Arduino 1-Wire Address Finder*. Retrieved August 15th, 2015, from <http://www.hacktronics.com/Tutorials/arduino-1-wire-address-finder.html>

Harp, J. (2013). *Aquarium Lighting*. Retrieved August 13th, 2015, from Reef Central: <http://www.reefcentral.com/forums/showthread.php?t=2304434>

Hauter, S., & Hauter, D. (n.d.). *Importance of Dissolved Oxygen Levels in Your Aquarium*. Retrieved August 15th, 2015, from saltaquarium: <http://saltaquarium.about.com/od/aquariummaintenancecare/a/Importance-Of-Dissolved-Oxygen-Levels-In-Your-Marine-Aquarium.htm>

Hunt, P. (2013, August). *Marine Aquarium Basics, Part 1:Water and Salt*. Retrieved November 3rd, 2015, from Tropical Fish Magazine: <http://www.tfhmagazine.com/details/articles/marine-aquarium-basics-part-1-water-and-salt-full-article.htm>

I2C Bus. (2015). *I2C Bus Specifications*. Retrieved October 15th, 2015, from I2C: <http://i2c.info/i2c-bus-specification>

I2C Bus. (n.d.). *Speed*. Retrieved October 22nd, 2015, from I2C Bus: <http://www.i2c-bus.org/speed/>

International Rectifier. (2010). IRF540N. *HEXFET Power MOSFET* .

Maxim Integrated. (2008). DS18B20 Programmable Resolution 1-Wire Digital Thermometer. California, USA.

Maxim Integrated Products. (2008, June 19th). *Overview of 1-wire Technology and Its Use*. Retrieved August 12th, 2015, from Maxim Integrated: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1796>

Maxim/Dallas Semiconductor. (2008). *DS1307*. Retrieved September 9th, 2015, from http://smart-prototyping.com/image/data/2_components/Arduino/100845%20Arduino%20Tiny%20RTC%20I2C%20real%20time%20clock%20module%2024C32%20storage%20%20DS1307/DS1307.pdf

Messer, D. (2012, August 16th). *Arduino*. Retrieved October 18th, 2015, from Wikimedia Commons: <https://upload.wikimedia.org/wikipedia/commons/6/63/Arduino.png>

MilesBurton. (2014). *Dallas Temperature Control Library*.

Mohan, N. (2003). *Power Electronics: Converters, Applications, and Design*. Hoboken: John Wiley & Sons.

Multicherry. (2015, February 21st). *Raspberry Pi 2 Model B v1.1 top*. Retrieved November 6th, 2015, from Wikimedia Commons: [https://commons.wikimedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_top_new_\(bg_cut_out\).jpg](https://commons.wikimedia.org/wiki/File:Raspberry_Pi_2_Model_B_v1.1_top_new_(bg_cut_out).jpg)

NASA. (2012, May 8th). *The Blue, the Bluer, and the Bluest Ocean*. Retrieved October 16th, 2015, from GES DISC: <http://disc.sci.gsfc.nasa.gov/education-and-outreach/additional/science-focus/locus/ocean-color/oceanblue.shtml>

Neptune Systems. (2012). *Apex Controllers*. Retrieved November 2nd, 2015, from Neptune Systems: <https://www.neptunesystems.com/products/apex-controllers/apex-controller-system/>

Njmcca. (2013, June 21st). *Peristaltic pump*. Retrieved October 15th, 2015, from Wikimedia Commons: https://commons.wikimedia.org/wiki/File:Peristaltic_pump.gif

NoteTab. (2015). *NoteTab 7*. Retrieved October 15th, 2015, from NoteTab: <http://www.notetab.com/>

Orphek. (2015). *About Corals* . Retrieved July 24th, 2015, from <https://orphek.com/about/about-corals/>

Orphek. (2015). *About Corals*. Retrieved March 24th, 2015, from <https://orphek.com/about/about-corals/>

Raspberry Pi. (2015, February). *raspberrypi 2 model b*. Retrieved July 12th, 2015, from <https://www.raspberrypi.org/products/raspberrypi-2-model-b/>

Raspberry Pi. (n.d.). *Terminal*. Retrieved October 15th, 2015, from Raspberry Pi: <https://www.raspberrypi.org/documentation/usage/terminal/>

Reef Ai. (2014). *Package Systems*. Retrieved November 2nd, 2015, from Reef Ai: <https://www.reefai.com/product-category/package-systems/>

Reef Angel. (2012). *Controllers*. Retrieved November 2nd, 2015, from Reef Angel: <http://www.reefangel.com/products.aspx>

Reef Central. (2008). *Low pH: Causes and Cures*. Retrieved August 2nd, 2015, from Reefkeeping: <http://www.reefkeeping.com/issues/2004-09/rhf/>

Reefkeeping Magazine. (2008). *Reefkeeping 101 - Sumps*. Retrieved August 15th, 2015, from Reef Keeping: <http://www.reefkeeping.com/issues/2008-01/newbie/>

Riddle, D. (2013). *Feature Article: Imitating Natural Light Quality, Intensity, and Dosage in a Reef Aquarium*. Retrieved August 25th, 2015, from Advanced Aquarists: <http://www.advancedaquarist.com/2013/12/lighting>

Riddle, D. (2006). *Feature Article: Temperature and the Reef Aquarium*. Retrieved November 3rd, 2015, from Advanced Aquarist: <http://www.advancedaquarist.com/2006/2/afeature2>

Rouse, M. (2005, September). *I2C bus (inter-IC bus)*. Retrieved October 20th, 2015, from WhatIs: <http://whatis.techtarget.com/definition/I2C-bus-Inter-IC-bus>

Rouse, M. (2005, September). *luminous flux*. Retrieved October 29th, 2015, from What is: <http://whatis.techtarget.com/definition/luminous-flux>

SainSMART. (n.d.). *8 Channel DC 5V Relay Module for Arduino*. Retrieved August 15th, 2015, from SainSMART: <http://www.sainsmart.com/8-channel-dc-5v-relay-module-for-arduino-pic-arm-dsp-avr-msp430-ttl-logic.html>

Science Daily. (2015). *Chlorophyll*. Retrieved from Science Daily.

Seneye. (n.d.). *Compare*. Retrieved November 2nd, 2015, from Seneye: <http://www.seneye.com/devices/compare/>

Ulrich III, A. B. (2013, February 7th). *9 Most Important Reef Tank Aquarium Parameters*. Retrieved July 22nd, 2015, from <http://saltwateraquariumblog.com/9-most-important-reef-tank-aquarium-water-parameters>

w3schools. (2015). *HTML Tutorial*. Retrieved August 17th, 2015, from <http://www.w3schools.com/html/>

Waveshare. (2015). 7 inch HDMI LCD (B).

Wiltec. (2006). *Wave Maker JVP-101A*. Retrieved from Wiltec:
http://shop.wiltec.info/product_info.php/language/EN/info/p10153_SunSun-Stroemungspumpe-Wavemaker-fuer-Sue---und-Salzwasser-3000-l-h-6W.html

Appendix 1 : Marine Aquariums and Reef Keeping

A1.1 Importance of pH

pH or 'Power of Hydrogen' levels in marine aquariums are a matter of constant concern to even the most seasoned of aquarists. In a marine tank the marine life is heavily dependent on a constant range of pH for their survival. Basically, pH is the measure of alkalinity or acidity in water. A pH of 7 is considered to be neutral since pH ranges on a scale from 0 to 14, below 7 being acidic and above 7 being alkaline. For a marine tank the pH should be between the ranges of 8.1 and 8.4, however, reef tanks are more sensitive and require the pH to be at the upper end of this range. The pH of a marine tank is affected by a number of factors but the most prevalent is the natural by products of fish which cause the build-up of nitric acid in the water and can lead to a reduction in pH (Aquatic Community, 2004).

A1.2 Importance of Dissolved Oxygen

In some way or another all marine life reacts with oxygen. Marine life such as fish, invertebrates, corals and bacteria all ingest oxygen O₂ and expel carbon dioxide CO₂ 24 hours a day. For this marine life a dissolved oxygen content of between 5 and 7 PPM is required in the saltwater of the tank. It has been found that marine occupants will show their first signs of stress at 4 PPM dissolved oxygen (Hauter & Hauter).

A1.2 Importance of Salinity

Since the life that is held in marine aquarium is from the ocean it is important to ensure that the salinity levels of the tank effectively mimic the salinity level of the marine life's original habitat. The salinity levels necessary for marine life range between 27 and 35 PPT, or 1.020 and 1.026 for specific gravity. Salinity is the measure of dissolved sodium ions in water whereas specific gravity is the ratio of salt water to pure water. When the salinity levels are not correct for the inhabitants of the tank, depending on their tolerance, some inhabitants may die (Hunt, 2013).

A1.3 Importance of Water Temperature

Coral reefs most commonly grow in tropical regions. This tropical climate means the water temperatures for coral reefs is between 24 and 29 degrees Celsius. When the temperature begins to rise above 29 degrees Celsius many marine life will experience stress and many corals can become bleached or may even die. In addition to this it is important to ensure that the temperature does not fluctuate too rapidly. Although coral reefs do experience changes in temperature of up to 10 degrees Celsius they do not suffer any damage as this temperature change has taken place over weeks or even months through seasonal changes (Riddle, Feature Article: Temperature and the Reef Aquarium, 2006).

A1.4 Flow Rates and Pumps and Sumps

A1.4.1 Sumps

A Sump is a secondary tank separate to the main tank; the water cycles from the sump tank into the main tank and back again. The sump tank performs a number of necessary roles in a reef aquarium setup, these include:

- Mechanical Filtration
- Chemical processing
- Creates flow
- Healthy algae growth

A typical sump layout can be seen in Figure A1, this layout was used as a template for the design of the sump used in this project.

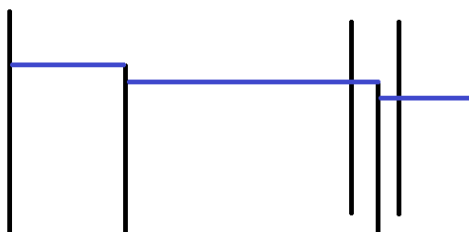


Figure A1 – Sump Design Template.

A1.4.2 Flow Rates and Pumps

A return pump is the most important piece of equipment for a saltwater tank with a sump in regards to life support. The return pump creates the flow through the tank and sump, cycling the water to provide the benefits listed for a sump. When selecting a return pump it is necessary to select a pump with a flow rate suitable to the tank size and tank occupants. The general rule of thumb for flow rate is 10 times your tank volume per hour. Having said that it is often found that this value is too high and a pump of this magnitude will create too much heat and noise. Therefore a flow rate of between 5 and 7 times the tank volume per hour is typically used.

A1.5 Importance of Lighting for Coral

Corals are colonial animals which build calcareous frameworks to live in. They feed in two ways; the first is by filtering plankton from the water and the second is via symbiotic algae which live inside their bodies. As the algae are plants they require light to photosynthesize and grow. It is this reason that correct lighting is so important in reef aquariums as these algae require certain wavelengths of light rather than just any light.

Appendix 2 : Project Build

A2.1 Aquarium Stand

The stand used to house the aquarium was created from an old TV cabinet as can be seen in Figure B1. For better viewing the cabinet was raised and further supported to withstand the weight of the full tank. The cabinet was the painted for water proofing and aesthetics. The modified stand can be seen in Figure B2.



Figure B1 – TV Cabinet.



Figure B2 – Modified Stand.

A2.2 Lighting Hood

The lighting hood was constructed to sit on top of the tank with a lid on the top for access. This lid was also the mounting position for the LED lighting and therefore two exhaust fans were fitted into the lid to keep the LED lighting cool. The lighting hood was constructed as seen in Figure B3. The hood was then painted for water proofing and fitted to the top of the tank as can be seen in Figure B4.



Figure B3 – Lighting Hood, Construction.



Figure B4 – Completed Lighting Hood on Tank.

Upon completion of the lighting hood, the LED heatsink was designed and built from aluminium angle. As can be seen in Figure B5, the LEDs were then positioned onto the heatsink according to the arrangement. This LED heatsink was then mounted to the underside of the lid.



Figure B5 – LED Heatsink and LEDs.

A2.3 Sump

The sump was built from Perspex/acrylic sheets and glued together with acrylic cement. Seen in Figure B6 is the sump prior gluing, it can be seen that the design for the sump is the same as that shown in Figure A1. The completed sump can be seen in Figure B7 in which it is positioned in the stand and holding water.



Figure B6 – Partially Complete Sump.

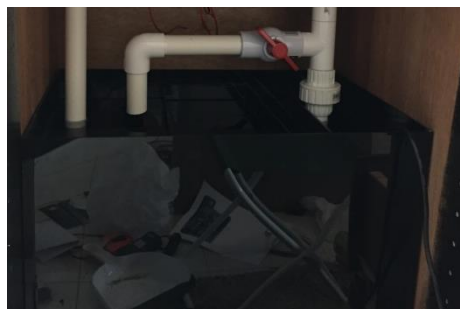


Figure B7 - Completed Sump in Position and Holding Water.

A2.4 Electrical Draw

The electrical draw design can be seen in Figure B8. The electric draw was designed to hold all control components and power converters. This provided easy wiring, maintenance, upgrades and a water free zone for safety. The original electrical draw can be seen in Figure B9 and the completed electrical draw in Figure B10. With varying voltages and crucial components being located in the draw a lock was installed to ensure unauthorized access can not occur.

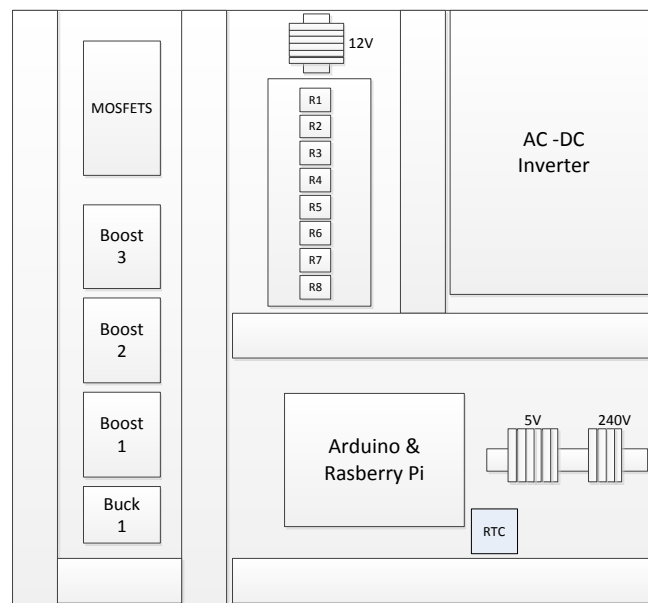


Figure B8 – Electrical Draw Design.



Figure B9 – Original Electrical Draw.

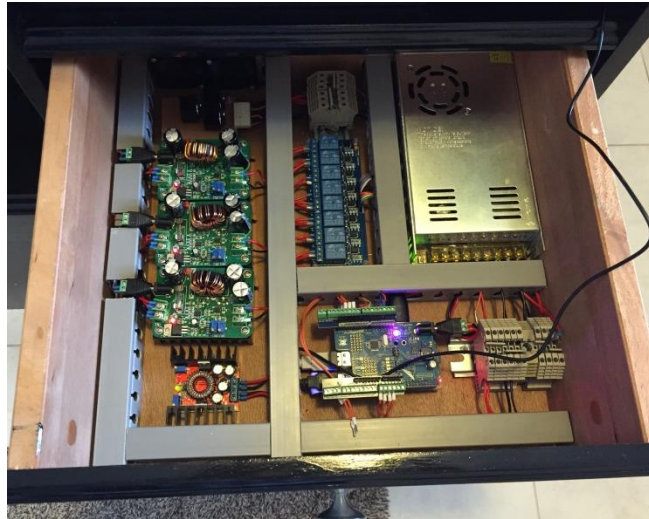


Figure B10 – Completed Electrical Draw.

A2.5 Plumbing

As mention in the thesis the water is cycled from the main tank to the sump and back continuously. To facilitate this, plumbing was needed to connect the two tanks. The front plumbing layout can be seen in Figure B11 and the back plumbing layout in Figure B12. For the plumbing $\frac{3}{4}$ inch PVC pipe was utilised along with three manual valves to manually stop or redirect water flow. As mentioned previously the flow from the sump to the main tank operates via a return pump. This is significantly different to the flow from the main tank to the sump which uses a siphon drain setup, as seen in Figure B13. Figure B14 shows the siphon drain setup for this project. To ensure there was no risk of water/electricity hazards, all electrical areas were sealed with rubber seals. Additionally the smaller tanks above the drawer were sealed to ensure these could not leak or spill.

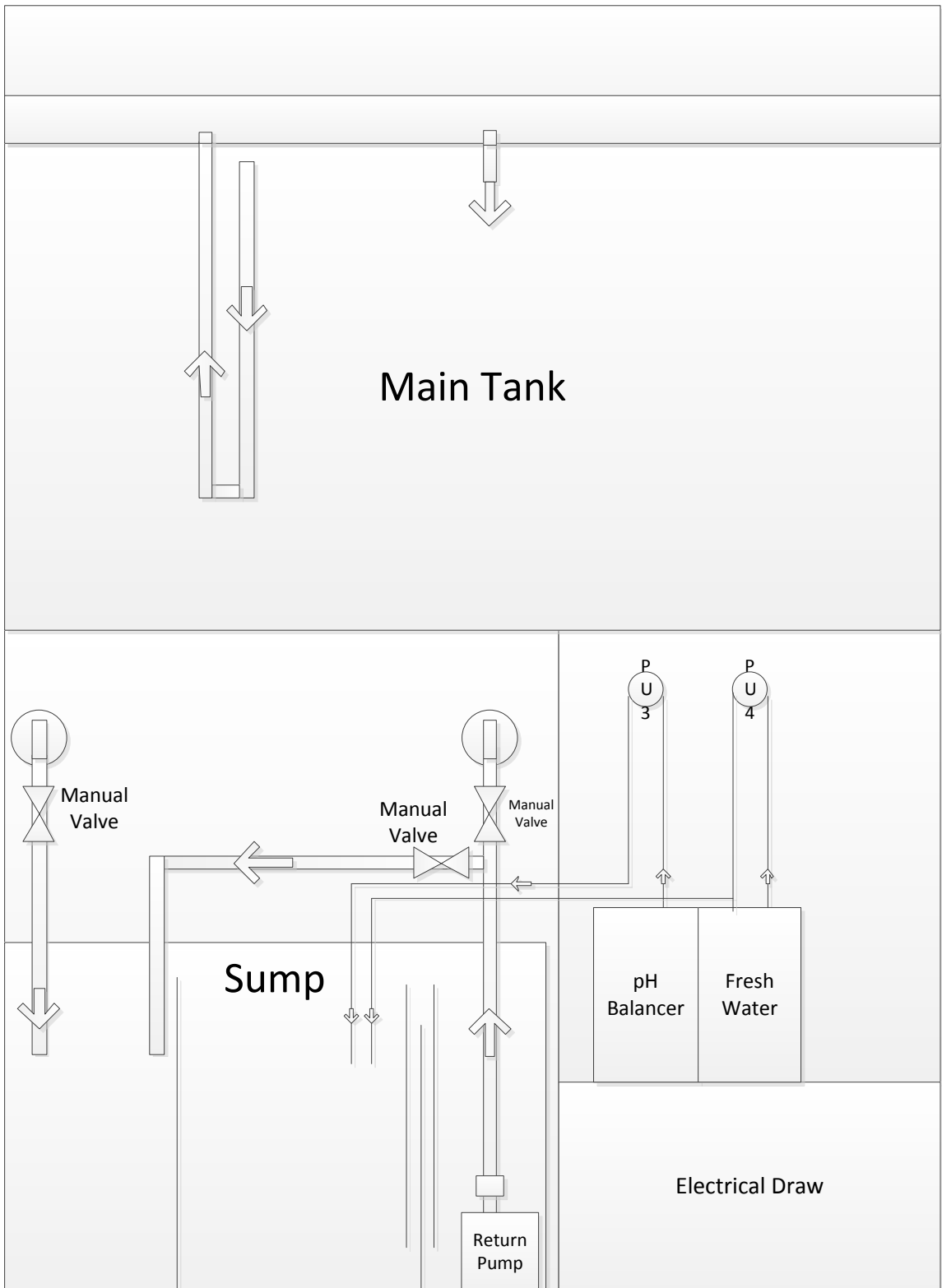


Figure B11 – Plumbing Layout, Front.

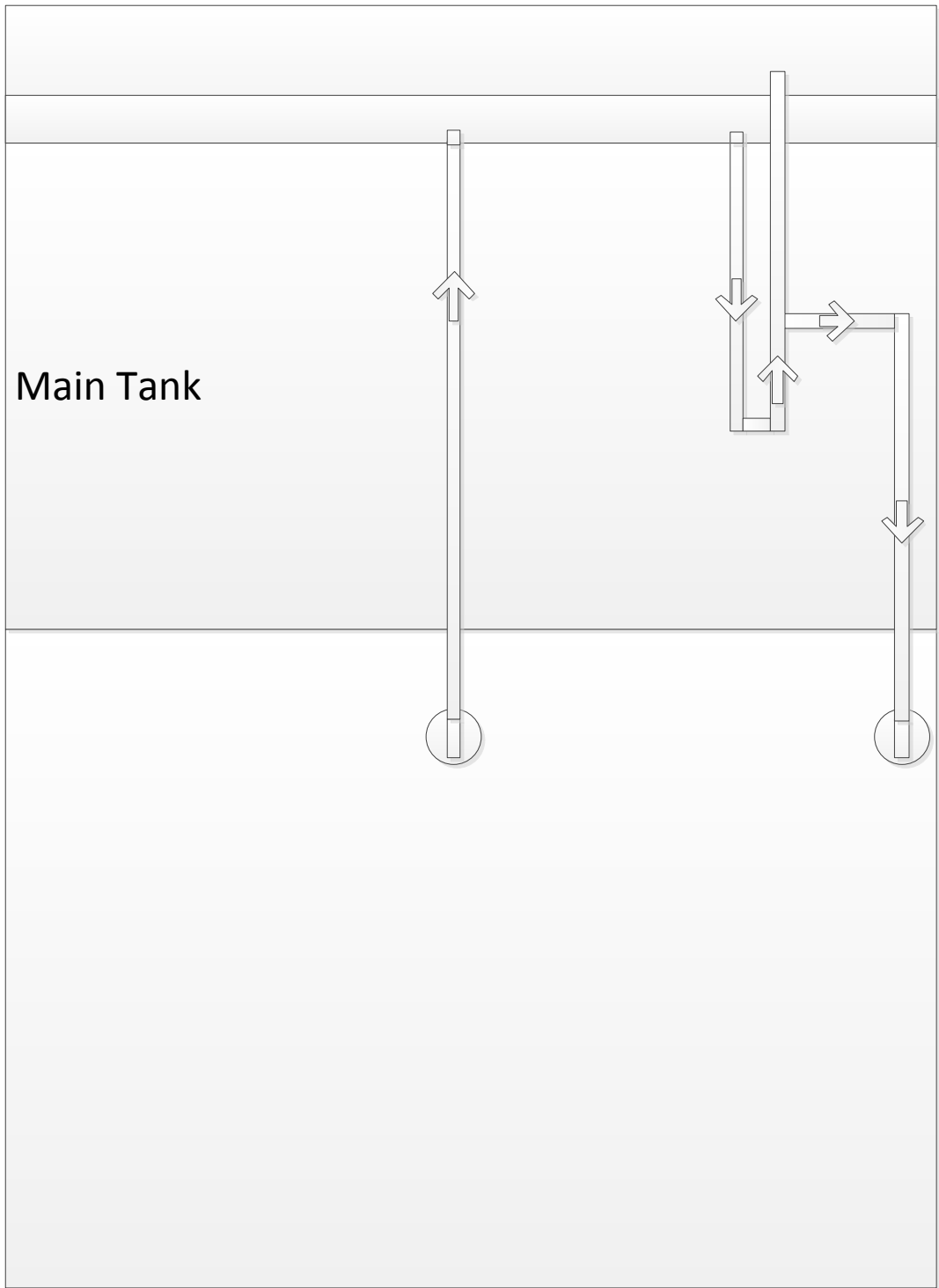


Figure B12 – Plumbing Layout, Back.

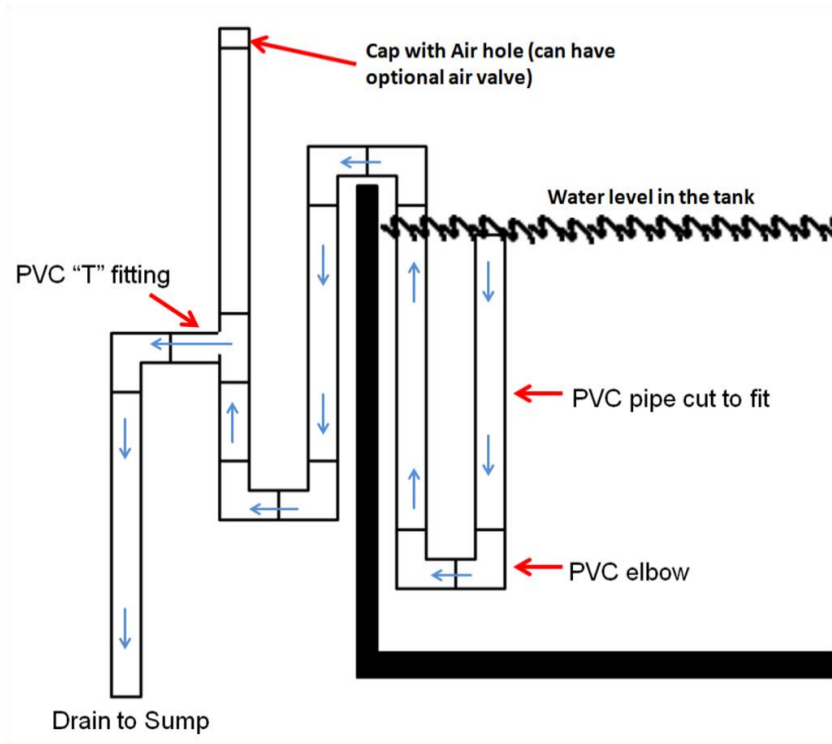


Figure B13 – Siphon Drain Setup.



Figure B14 – Completed Siphon Drain Setup for the Project.

Appendix 3 : Calculations

A3.1 Aquarium Heaters

In calculating the wattage of heaters required the volume of water was first calculated using the dimensions of the tank and sump in centimetres.

$$\text{Main Tank} = 91.44 \times 43.18 \times 40.64 = 160462\text{cm}^3$$

$$\text{Sump Tank} = 43.18 \times 25.4 \times 30.48 = 33429.6\text{cm}^3$$

The volume of each tank was then converted to litres and summed to give approximately 194 litres. Using Table C1, the wattage for the heaters were selected based on a necessary temperature increase of between 5 and 12 degrees Celsius.

Table C1 (Aquarium Monsters, 2011)

Litres/Gallons (std tank size)	5°C/9°F	10°C/18°F	15°C/27°F
25 L/5 gal (1ft)	25 watt	50 watt	75 watt
50 L/10 gal (2ft)	50 watt	75 watt	75 watt
75 L/20 gal (3ft)	50 watt	75 watt	150 watt
100 L/30 gal (4ft)	75 watt	100 watt	200 watt
150 L/40 gal (6ft)	100 watt	150 watt	300 watt
200 L/50 gal(8ft)	150 watt	200 watt	two 200 watt
250 L/65 gal (10ft)	200 watt	250 watt	two 250 watt
300 L/75 gal	250 watt	300 watt	two 300 watt

A3.2 Power and Current Draw

The total power and current calculations for the 12 V devices can be seen in Table C2, and the 5 V devices in Table C3.

A3.2.1 12V System

Table C2 Total Power and Current for all 12V Devices.

Component	Voltage (V)	Current Draw (A)	Power Draw (W)
PiLeven	12	1	12
R Pi (Fed from PiLeven)	0	0	0
RB LED Boost Converter	12	3.975	47.7

CW LED Boost Converter	12	2.89	34.68
UV & G LED Boost Converter	12	1.9	22.8
Dosage Pump	12	0.08	0.96
Dosage Pump	12	0.08	0.96
5 V VCC Buck Converter	12	0.202	2.424
Exhaust Fans(total)	12	0.05	0.6
Total		9.177	110.124

A3.2.2 5V Devices

Table C3 Total Power and Current for all 5V Devices.

Component	Voltage (v)	Current Draw (A)	Power Draw (W)
pH EZO	5	0.018	0.09
DO EZO	5	0.018	0.09
EC EZO	5	0.05	0.25
Relay Module	5	0.16	0.8
Ultrasonic sensors	5	0.03	0.15
Optocoupler & Resistors	5	0.11	0.55
All other 5 V components approx	5	0.1	0.5
Total		0.486	2.43

A3.3 LED Number Calculator for Luminous Flux

The LED number calculator for luminous flux can be found as a excel file in folder A3.3 submitted as part of this thesis.

Appendix 4 : Tests

A4.1 pH Home Test

A handheld digital pH tester, was loaned to confirm the pH readings given by the pH probe by Atlas Scientific. The loaned sensor gave a reading down to one decimal of accuracy and was left in the water to be read at the same times as the atlas scientific probe.

A4.2 Salinity Home Test

A salinity tester, seen in Figure D1, was purchased to test the accuracy of the salinity probe. In this case the salinity tester measured the salinity via specific gravity as can be seen in Figure D3. Although not particularly accurate the sensor did provide some validation of the atlas scientific EC or salinity probe.



Figure D1 – Salinity Tester.

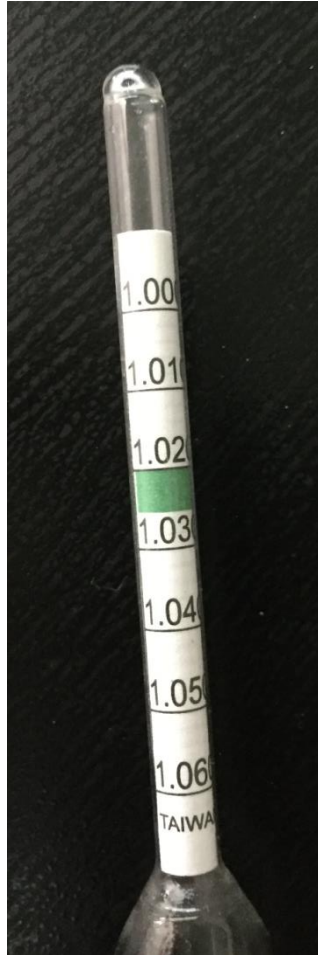


Figure D3 – Salinity Measurements in Specific Gravity.

A4.3 I2C Test Results

A test was conducted on all I2C devices by printing the readings of each sensor every ten minutes to the serial monitor. The serial print outs were saved and can be found in file A4.3 as a text file.