Senior Design Project Report
Trout Lab

Prepared by:

Nicholas Vanni
Jireh Alberto
Brandon Barron

Faculty Advisor: Dr. Ali Kujoory
Industry Advisor: Mr. Tom Greer
Client: Redwood Empire Trout Unlimited

December 12, 2018
https://troutlabsystem.wixsite.com/troutlab

Revised: Dec-2018

Acknowledgment

Dr. Farid Farahmand
Mr. Tom Greer
Dr. Ali Kujoory
Mr. Shahram Marivani
Mr. Eric Leland
Redwood Empire Trout Unlimited
SSU SOURCE

Abstract

The Proposed device the team is building is to improve programs such as Steelhead in the Classroom with smart fish tanks that combine sensor and video data processing with web based data viewing and control to help teacher and students learn about Trout lifecycles and biology. the proposed device will combine light, Temperature, and ORP sensor data with periodically recorded video to assist users and students. The device can monitor the growing of the trout eggs in a tank of water in an environment that is most suitable for the eggs. And alerts users when the danger threshold for these values in order to preserve the eggs and ensuing fry. The water temperature and the Oxidation-Reduction Potential (ORP) level, and light intensity in the tank are controlled from remote. The eggs can be viewed by a camera. The pictures and videos of the eggs together with other data such as the temperature and ORP level are periodically recorded in time and stored in the cloud or a local server for later study. The device can be used in trout growing industry as well as in school to educate the students. It should be mentioned that similar device such as Steelhead in the Classroom does not have the capability to monitor the conditions in the environment, therefore changes in temperature and ORP level may seriously affect the trout egg growth.

Table of Contents

List of Figures 5

List of Tables 6

Introduction 7

Literature Review & Previous Works 7

Problem Statement 9

Methodology 10

Challenges & Risks 10

Project Requirements 12

Marketing Requirements (MR) 12

Engineering Requirements (ER) 13

Customer Surveys 15

Implementation 16

System Architecture 16

Budget/Parts List 17

Project Schedule 18

List of Tests 20

Summary of Tests (FT. = Functional Test) 20

Description of Tests 20

Ethics 35

References 35

List of Figures

Figure 1 – System Diagram……………………………………………………17
Figure 2 – Camera Test Setup…..…………………………………………...21
Figure 3 – Camera Configuration.…………………………………………...21
Figure 4 – Main Raspberry RPi Configuration Menu.…………………………….21
Figure 5 – Interfacing Options Menu.………………………………….…….…22
Figure 6 – Camera Enable Screen…..……………………………………….22
Figure 7 – IR Light Configuration…..………………………………………….22
Figure 8 – How To Take a Single Photo.………………………………………….23
Figure 9 – Fish Egg Size Test.………………………………………...……..23
Figure 10 – Video Test Setup.………………………………..…………..…24
Figure 11 – Relay Design.……………………………………………..…25
Figure 12 - Data Transmission Test……….………………………………....26
Figure 13 – Data on Webpage with Graphs.……………………………………....26
Figure 14 – Twilio Test Setup………….....………………………….………...27
Figure 15 – Twilio Test Messages.………………………………….…………28
Figure 16 – Startup Test Setup…..………………………………….…………28
Figure 17 – ADC RPi Test Setup.………………………………….….…....29
Figure 18 – ADC RPi Wiring Schematic.…………………………….….……....30
Figure 19 – RPi I2C Test.………………………………………..…………30
Figure 20 – ADC Read Voltage Code.……………………………………………31
Figure 21 – ADC Voltage Output……….....…………………………………..31
Figure 22 – Voltage vs Voltage Read.……………………………..……...32
Figure 23 – Light Sensor Test Setup.……………………………………………..33
Figure 24 – Lumens vs Voltage Graphs.……………………………..….......33
Figure 25 – Light Sensor Voltage Data.………………………………………..34

List of Tables

Table 1 - Tests and Risks…………..…………….……………………………..10

Table 2 - Engineering Requirements………….………………………………..13

Table 3 - Customer Survey Responses……...….…………………………………15

Table 4 - Budget…………………………………………………………...17

Table 5 – Gantt Chart……..…………………………………………………..19

Table 6 – Functional Tests……………………………………………………...20

Table 7 – Twilio Response Time………………………………………………….27

Table 8 – Startup to Connection…………………………………………………..29

Table 9 – ADC Voltage Data……………………………………………………...31

Table 10 – Light Sensor Voltage Data…………………………………………….33

Introduction

There are many programs in schools, such as Steelhead in the Classroom, that provide classrooms with fish eggs to grow, hatch into fish, and then release into the wild. The problem that arises is that there are currently no ways to monitor the important conditions of the tank that help the fish eggs grow and develop in.

In the case of Steelhead in the Classroom, the classroom is given steelhead trout eggs to watch grow. Trout eggs require very specific conditions in their tanks in order to develop properly and healthily. These conditions include a dark environment, water temperature to be between 50°F and 60°F, and an ORP (oxygen reduction potential) level between -165 mV and 205 mV. The way that classrooms currently monitor the dark conditions of the tank is by putting styrofoam pieces around the tank and trusting that it is dark enough. They currently monitor the temperature of the tank by placing a mercury thermometer inside of the tank. There is currently no way to monitor the ORP level inside the tank.

Proposed device, Trout Lab, will solve the problems of monitoring the various aforementioned conditions. Using various sensors, the device will electronically gather the temperature and ORP data and send it to a website that will display the conditions of the tank. The device will also use a light sensor to monitor the light intensity inside the tank. If any of these values go outside of the thresholds that they must be under or between, The server will send an alert to the user of the tank informing them that the tank's conditions are not where they should be. A camera will also be placed in the tank to send video data of the progression of the egg growth to the website.

Literature Review & Previous Works

The Device: The device will be one that monitors variables in a fish tank that affect trout egg development. The device needs to be able to monitor water temperature, the ORP level of the water, light intensity, and send the captured data to a website wirelessly via WiFi. The temperature must be between 50 and 60 degrees Fahrenheit, but ideally at 55 degrees. The ORP level must be between -165 mV and 205 mV, as that is the ideal range for trout eggs to grow in. It must also be able to send a warning/alarm notification to the user if the water temperature, ORP, or light intensity levels go out of their specified ranges via SMS message. The device will also have an IR camera that will record the development of the eggs and fish in the dark environment of the fish tank, and it will stream on the website. The device will also have a backup power supply that will last at least 24 hours to compensate for a power loss over the weekend. If the power goes down, the device will use the backup supply and send a message to the user informing them of this via email. The device will be powered by a Raspberry Pi Zero W(referred to as Pi). The user will also have the ability to remotely turn the chiller (the device that regulates the water's temperature) on or off from the website. The similar devices on the market are Seneye, Fishbit, Skybitz, and Neptune systems. These devices will be explained and compared to ours in the following paragraphs.

Seneye: The Seneye tank monitoring system checks temperature, light, ammonia, ph, and water level. It works from one wall mounted sensor inside of the tank that regularly scans the tank for changes in its various parameters. The piece is wired then to a computer running the Seneye applet in the background to analyse the results. This applet can communicate with customer email and phone through an app or through sms messaging. The system is well received but many users do not like the applet integration and connecting it to a computer. Given how electronics and water work, many are scared to wire a sensor inside of their tank directly into one of the most valuable electronic items in their home. Its various capabilities also confound users, given how web integration has been described as half finished by some users, and many complaints have been issued over the frequency of replacing ph sensor parts, and how easily the wall mount falls off the tank wall. [1] [2]

Fishbit: The Fishbit contains a monitor, controller, and app to keep the user connected their aquarium at all times. The two devices, the Fishbit monitor and controller, connect to WiFi which will allow you to connect to the devices from anywhere with the Fishbit app. The features of the Fishbit monitor include, temperature, pH level, and salinity sensing. The monitor also constantly checks each sensor to alert the user if either the temperature is too high or low, if the pH level is too acidic or basic, and if the salinity of the water is too high or low. The Fishbit app allows you to adjust the temperature if a heater is connected, it will also allow you to turn on or off a pump, as well as allow you adjust the brightness of the LED lights. The Fishbit app will send an alert or notification to customers' phone if there is something wrong in customers' tank. It will also keep track of customer tank in real time and will give you an analysis of the temperature, pH level, and salinity throughout the day. There were a few concerns to the Fishbit system that some customers observed. One of the concerns was that the probes are unable to be replaced when damaged and it is installed in the upper-end of the monitor, sealed up by a thick layer of silicone. Using silicone is a big concern because it is prone to failure after a while of being submerged in saltwater. Another issue that customers observed is that the software stops live reporting aquarium conditions to the user's smartphone in real time. For the data to be up-to-the-minute, the Fishbit system needs to be unplugged and fully restarted. This could be a real problem if an user is traveling, at work, or away from their tank. [3] [4]

Skybitz: is an industrial tank monitoring system for shipping fluids and storing fluids. It connects via a 4g or 5g SIM card to the users server to give them real time access to live statistics. These mean having to pay for third party network access and no control over tank conditions. And only offer solutions on the supplier industrial levels. [5]

Neptune Systems: The system comes with a temperature and pH sensor as well as a connected control unit. The system can be improved further with the purchase of extra sensors for all range of variables. The system however, plugs into a router in order to connect remotely. This is dangerous because a router or modem must be near a fish tank. The system is also programmable so users can setup timing for lighting and testing cycles. The casting and sensor rigging seems solid, but maybe a bit too solid. They appear to have issues with mounting easily to tanks and walls. [6] [9]

Trout Research: Trout are the species of fish that the device will be programmed to monitor safely. These fish are very sensitive to many conditions. Of these conditions, they are especially sensitive to light, temperature, pH, and ammonia, especially at the early stages of their life, when classrooms will be interacting with them. The trout's favorite temperature is 55°F with +- 5°F on either end of leniency.[10] For light they prefer almost 18 hours of darkness to live in. For pH, the trout enjoy a range of pH values, specifically around 3.5-9.8 pH. The pH level directly correspond to ORP level in that at a pH level of 7, ORP is 0. for each pH level on either side of the seven, the ORP value goes up or down by 59 mV. Therefore, the safe ORP range is between 205 mV and -165 mV. Trout seem to prefer to be in water with pH levels between 6.5 and 8.0 according to the US Fish and Wildlife Service, so ideally the ORP levels should be between 29.5 mV and -59 mV. [8]

What Makes Trout Lab Different: Proposed device is different from the other devices in that ours will include a camera that will post videos to Trout Unlimited website so that the teachers and students can view the development of the eggs they're growing. Proposed device will also be connected to a network wirelessly unlike all of the other devices, other than Skybitz.

Problem Statement

The problem that the device will solve is the fact that for programs such as Steelhead in the Classroom, there is currently nothing that monitors the parameters that affect trout egg growth and development. These parameters include recording and maintaining a safe temperature for the trout to live in, recording and maintaining the ORP(Oxidation Reduction Potential) level for the quality of the water in the tank, and making sure that light is not exposed in the tank since too much exposure of light will affect the development of the trout eggs. Another problem that proposed device will solve is that, unlike other devices that monitor fish but do not show you videos of what is happening in the users' tank, proposed device will contain an Infrared camera that will record the growth of the fish eggs. The device will be using an Infrared camera because the trout eggs in the tank need to be under dark conditions. The Infrared lights on the camera will allow the user to see images and videos of the trout eggs in the dark. Proposed device will send all of the data of all the sensors and what is recorded from the

Raspberry Pi Zero, throughout the day, to a cloud for historical and analytical use. The user, whether if it is a student, teacher, or any person, will be able to monitor the parameters and safety of the Trout in their tank from anywhere and anytime of the day for a stable development of the trout eggs.

Methodology

The team will solve the proposed problem by creating a device that can measure and monitor the water's temperature, ORP (Oxidation Reduction Potential) level, and light intensity so that teachers and students can be constantly aware of all the useful statistics inside of the tank at any given time. Proposed device will be recording and sending all of the sensor data to a website from a Raspberry Pi Zero. The website can be accessed from any device, such as a smartphone or laptop, with authentication. The device will contain a camera that will post footage to the website as well, so that teachers and students can see what is going on inside the tank. The proposed device will also allow the user to control the temperature of the tank by including a user interface between the website and the Chiller.

Challenges & Risks

Test	**Risk**	**Solution**
1. Streaming Video Over Internet: Begin recording with the camera, and see if the video streams on the website.	**1. The Camera Doesn't stream the video:** This hinders the project because it prevents the users to view the development of the eggs on the website. If this happens, the team may try a different camera and troubleshoot the code that interfaces the camera to the RPi.	**1. Replace camera or change communication protocol:** Depending on the complexity of the programming the camera's digital interface with the micro controller could be an issue so using a different wiring scheme or using different transfer protocol could increase the reliability of the video transmission.
2. Sensor Accuracy: Calibrate sensors with the calibration kit and see if the sensors are accurate to what the calibration kit says.	**2. System sensors are not accurate or fail to read:** Re-calibrate the sensors and see if they function properly. If this doesn't work the team will try different sensors or see if the code to	**2. Replace sensors, check the wires and clean them.** If the sensors are giving false or incorrect readouts, the first measure will be to check the sensors clean and calibrate them. If they continue to give false readings the

	interface the sensors is correctly written.	team will then proceeded to check the wiring and the RPi if the values are correct there. If they are not, then the sensors may be replaced.
3. Database Testing: Test client and server communication - populate the database with false data and log in as a user to check that data. If it shows the proper data from the sensors and the false data that the team inputs, then the database should be successful.	**3. System Database doesn't store and/or communicate the right data:** Use a different database.	**3. Replace the database, realign table, or check transmission.** If the database is giving, storing, or displaying incorrect data the team will have to check the transmission from the database, or the Pi to verify that the data is stable. If it is not stable the team will have to change the transmission method. If it is just the database, the team will have to change the chosen database. If the tables are storing the data wrong, the team will have to have to change the table to better represent how the data is displayed. To do this, new tables will have to be created with different schemas to better show the data.
4. Backup Power Supply Life: Once everything is running correctly, the team can test device backup power supply's life. Ideally it will allow the device to run for at least a full day. The team will try this multiple times to ensure that the backup power supply will be sufficient if there is a power outage over the weekend.	**4. Backup Power Supply Doesn't Last Long Enough:** Observe the power consumption of the device. If it is more than expected based on the specifications of the Raspberry Pi Zero W, a new backup battery with a better power rating may be needed.	**4. Replacing power supply or modifying functionalities for a low power mode.** When the backup power supply fails. The primary measure would be to replace it especially if the device drains battery especially quickly. Other useful measures would be modifying the script running the RPi to run in a low power mode when there is no network to decrease the stress on the

		system, thus enabling the backup battery to last longer.
5. Notification System: The team will feed the system false readings on the system sensors and see if it alerts the users via SMS as planned.	**5. System Notification System Doesn't Alert the user fast enough or at all:** Rewrite the code to use a different alert system such as email, phone calls, etc.	**5. Change notification methods,** If the notifications fail, the team will have to look at the triggers for each notification. If these triggers are not at the right values then verify the transportation method. Then change the method the notification would be received if SMS and Email are not moving fast enough. Checking the triggers would likely solve the problem. And again the problem could originate from the sensors which would lead back to risk 2.

Table 1 - Tests and Risks

Project Requirements

Marketing Requirements (MR)

1. Device must be able to measure water temperature, ORP, and light intensity.
2. User Interface must be easy to use for children.
3. Device must be able to connect to a Web-based application on the Trout Unlimited Website.
4. Application must have user authentication.
5. Device must have a notification/warning system via SMS.
6. Device must be able to be powered by a wall outlet and have a backup power supply.
7. Sensors and camera must be waterproof.
8. Device must have wireless network connection.
9. Device must have a camera that will record the growth of the fish eggs.
10. Sensors must be small enough to go in the tank without taking up too much space.
11. Device must cost less than current similar devices on the market.
12. Have an easy-to-read instruction manual on the website.

13. Device must record and store data of what is being sensed throughout the day, for historical and analytical use.
14. The Device must have easy sensor calibration.
15. Device must have parts that are easy to replace or repair if damaged.
16. Device must be allow user to control temperature remotely through the website.
17. Device must be tamper resistant from children.
18. Device will have a IR camera.

Engineering Requirements (ER)

Marketing Requirement #	Engineering Requirement	Validation
1. Device must be able to measure air temperature, water temperature, ORP and light.	1.1 Temperature must be within +/- 5°F of 55°F and accurate up to 1°F.	Compare system sensors to mercury Thermometers.
1. Device must be able to measure air temperature, water temperature, ORP and light.	1.2 The ORP level of the water must be between -165 mV and 205 mV and accurate within 5mV. [1] (This is the ideal ORP level for trout according to the US Fish and Wildlife Service	Test ORP sensor with things that have set pH levels such as black coffee, distilled water, bleach, etc. since pH and ORP are related to each other.
1. Device must be able to measure air temperature, water temperature, ORP and light.	1.3 Light sensor must be able to differentiate between different light intensities.	Compare readings to that of a lux meter.
1. Device must be able to measure air temperature, water temperature, ORP and light.	1.4 Sensors' analog data must be converted to digital data using an ADC Pi addon.	Compare the voltages applied from ADC pins to the actual voltage applied.
3. Device must be able to connect to a Web-based application	3.1 Use web site creating software to develop the website and connect via wireless Wi-Fi connection.	Connect to a router and see if the data is being transmitted to the website.
4. Application must have user authentication.	4.1 System shall allow the users to register by entering their username and password, in order to get access to the system.	The user data will be stored on a remote server using 64 bit encoding. Verify safety and security through packet capture software like wireshark.

5. Device must have a notification/warning system via SMS.	5.1 Notifies the user via SMS within 1 minute when the system reaches any threshold value.	Simulate a change in temperature and see if the user received an email in the specified time.
5. Device must have a notification/warning system via SMS.	5.2 Site will be specific about the issue in the text message, informing the user of issue.	Send data with specific flags
6. Device must be able to be powered by a wall outlet and have a backup power supply.	6.1 Device backup power must last up to 24 hours.	Leave unit unplugged and check for continuous updates to server over the period it is unpowered. As well as checking power light whilst not wall powered.
7. Sensors and camera must be waterproof.	7.1 Sensors must be operable and submersible up to 1 feet in water.	Submerge sensors into 12 inches of water and test if the sensors are still operable and accurate.
8. Device must have wireless network connection (Interface to connect to the wireless network).	8.1 The HDMI port of the RPi must be accessible for the user to plug into a monitor in order to connect to their classroom's wireless network.	Connect the RPi to a monitor or small screen and connect to a Wi-Fi network manually.
8. Device must have wireless network connection (Interface to connect to the wireless network).	8.2 Constantly monitoring the system and updating the website every minute.	Create a server log and change the temperature to see if changed on the log.
10. Sensors of device must be small enough to go in the tank without taking up too much space.	10.1 Tank is about 1.5' x 1' x 1' so the sensors must be small enough to fit in this tank without taking up too much space. The sensors that go in the water should not take up more than (length)5" x (width)5" x (height)11" of space	Find dimensions of sensors and make sure they are not too big for the tank.
11. Device must cost less than current similar devices on the market.	11.1 Competitors such as FishBit, Seneye, and Neptune Systems sit in a range of about $279-$800. The product must cost less than about $279 to create.	Add up cost of materials and see if it is less than what is currently on the market.
13. Device will store and record data	13.1 Must store data on each tank to a	Use a packet sniffer on a

of what is being sensed throughout the day, for historical and analytical use.	centralized location, in a web server. Over standardized 802.11a WiFi.	router or through a hardline connection to see if packets are sent and received by the RPi.
13. Device will store and record data of what is being sensed throughout the day, for historical and analytical use.	13.2 System will display a graph of the results for each day on an plot of Measurement vs. Time.	Log into the site to view simulated or actual data sent through the internet to the site. To test the web based application.
14.The Device must have easy sensor calibration.	14.1 Device must have simple sensor calibration that has been precoded. Takes no more than 10 minutes to be calibrated.	Purchase calibration kit, code, and test sensors for accuracy.
16. Device must be able to turn on and off the temperature remotely through the website.	16.1 User will be able to remotely turn the chiller on or off from using a relay that allows the 120 V chiller to be controlled from minimal voltage from the RPi Zero. Device will respond in under 1 minute.	Send commands to the device onsite, offsite, and remotely, and test the time taken for the device to activate the chiller. Send voltage over the relay manually to test its capabilities.
18 device must have an IR camera.	18.1 Camera should have at least 720p resolution.	Order a camera with appropriate resolution
18 device must have an IR camera.	18.2 Camera must be able to see inside the tank inside the dark.	Test camera by taking pictures in low light conditions.

Table 2 - Engineering Requirements

Customer Surveys

Question	Answer	Notes
What are the most important variables to measure?	ORP, Light, and Temperature.	Most of the customers agreed with these are the most dangerous values that the customers would most like to see measured.

What frequency would you like data uploaded?	Every 5-10 minutes for data, and once a day for video.	The customer wanted data at a semi frequent rate, but says that the video is not essential because they can always remove the cover from the tank.
What is the ideal price you would like to see for this product?	Under $250	Customers would like a semi-expensive one time solution to their tank monitoring needs.
How would you like to access the data?	Internet browser or webpage.	Because of the usage for remote viewing and open availability customers prefered internet based displays.
Which notification would you prefer?	SMS	Customers prefer SMS because of its quick response, and because of the accessibility over multiple platforms.

Table 3 - Customer Survey Response

Implementation

The Trout Lab system takes data from a fishtank sends it to users on a website where they can view the data or request certain things like, new video, new data points, or chiller control. the microcontroller unit should operate headless at all times and take minimal user support at startup only.

System Architecture

At the center of the Trout Lab system sits the Raspberry Pi Zero W, which connects to all the sensors and the web server. The sensors will be inside of the tank casing, with temperature and ORP sensors actually inside of the water. the chiller relay coil will be powered by the RPi as well. The RPi then transmits the data from the sensors to the web server where the data is processed and displayed. the server will handle the user interface and the display of data. and depending on user level they can send commands back to the RPi to control the Chiller relay. The data will be cached for California Department of Fish and Wildlife and for Trout Unlimited data processing purposes. and videos can be shared and stored on other sites if needed.

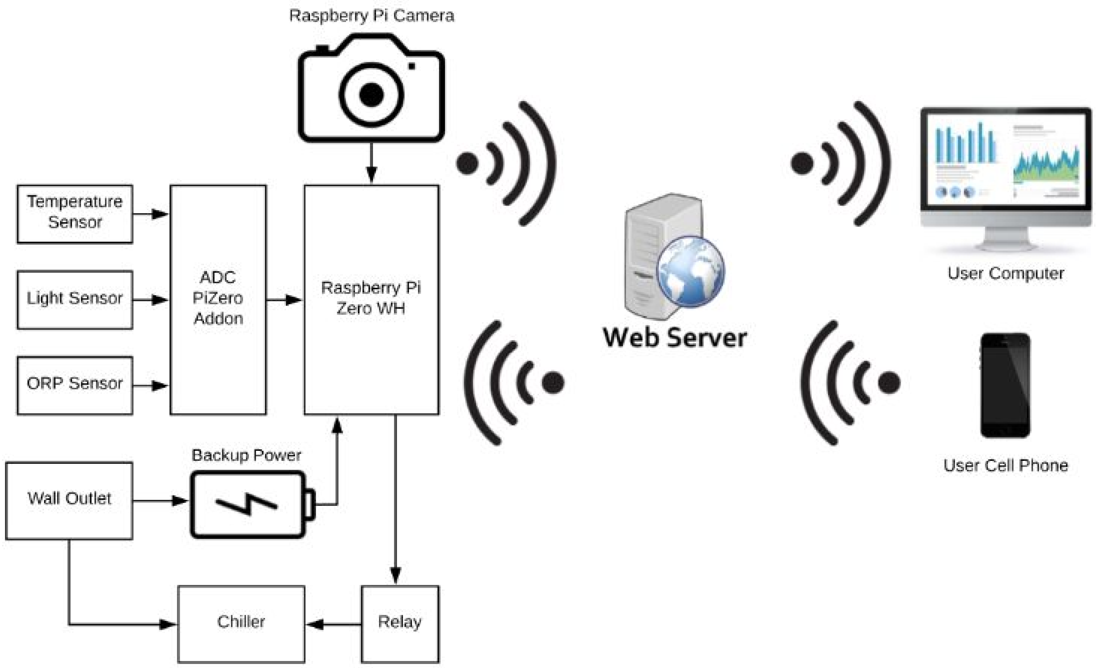


Fig. 1 - High level schematic of Trout Lab systems.

Budget/Parts List

The budget below highlights all the necessary parts for this project. The RPi Zero W is the keystone of Trout Lab, which powers the sensors and camera. the sensors will be attached to the RPi via the AB Electronics RPi ADC which will convert the sensors analogue signals to digital via the I2C interface. From there it will be sent to a web server, which is piggy-backed on the project's sponsor's existing web server, therefore there is no need to pay for it. The most expensive sensor is the ORP sensor which requires waterproofing by default but it is also a highly accurate sensor, capable of measuring the slight changes in the ORP. The ADC comes with several sample code set which enable us to process data quickly.

Part/service	**Qt**	**Part Number**	**Description**	**Price($)**
Raspberry Pi Zero WH	2	RPI-ZERO-WH	Microcontroller used in project.	28.00
Analog to Digital Converter Shield	2	ADC PiZero Addon	An accessory for the microcontroller that will give it the capability to convert analog signals from some sensors to digital signals.	44.40

Waveshare Raspberry Pi IR Camera	3	RPi IR-CUT Camera	A camera that is able to record in low-light conditions.	103.97
Waveshare Raspberry Pi Zero Ribbon Cable Connector	3	Raspberry Pi Zero V1.3 Camera Cable 15cm	A cable that makes the camera compatible with the Raspberry Pi Zero.	10.47
SUNKEE DS18B20 Temperature Sensor	8	SUNKEE DS18B20	Raspberry Pi compatible digital and waterproof temperature sensor.	48.00
Atlas Scientific ORP Sensor	3	EZO-ORP	RPi compatible ORP sensor.	330.00
SunFounder Photoresistor Light Sensor Module for Arduino and Raspberry Pi	10	43237-2	Non-waterproof light sensor for determining light pollution inside of the fish tank.	69.90
5A Relay 5 V trigger	4	G6B-2214P-US DC5	A relay that takes a 5 V input to open and close a 50W unit.	20.00
EasyAcc 26000 mAh power bank	3	PB26000MS	A 26000 mAh battery pack that will be used as the backup power supply.	150.00
USB to Micro USB adapters	5	NA	A adapter for turning a USB power output to micro USB	25.00
Electronic Components	NA	NA	Electrical Components as needed	70.00
			Total	**889.74**

Table 4 - Budget

Project Schedule

In the month of October 2018, the team finalized all of the requirements, design, and scheduling of the Trout Lab project. In November the team started making tests on some of the hardware and software of the Trout Lab system to make sure that they can resolve any risks that might occur. At the end of the month, before the Fall semester is over, the team will finalize the project proposal and present it to the students, faculty, and industries present for approval. Once all of the components and materials are purchased and received, the team will start developing the project. The team has divided the project into separate categories for the group to work on separately. One member will be incharge of the hardware design and testing, another will be in charge of the network interfacing, and the last member will be in charge of the website and server setup. Starting in December, the team will start constructing and testing the circuitry of this project, which includes calibrating the sensors, the relay mount design, and waterproofing the light sensors and camera. Once all of the sensors are calibrated, the team can start interfacing the sensors to the Raspberry Pi Zero. Starting in January, the team needs to start network interfacing the project by wirelessly connecting the router to the RPi, integrating SMS messaging, and web integration. While one of us works on the network interfacing, the other group members will setup the website and server which includes generating the HTML format, generating PHP code to build login and authentication, and verifying the storage and database. The team plans to complete all of the hardware designing, network interfacing, and website development by the end of February. At the beginning of March the team will work on complete the final touches of the project, making sure they have fixed all of the errors they might encounter and by making sure that the Trout Lab system meets all the Engineering and Marketing Requirements. The team will also start making the poster and final presentation for the final and complete product, which will be complete by mid April. Finally in May the team will present the final project to the audience.

Trout Lab	Start Date	End Date	Days	Team	Nov	Nov	Dec	Dec	Jan	Jan	Feb	Feb
1. Hardware Design/Test	**7-Nov**	**27-Feb**	**111**									
1.1 Design Circuitry	7-Nov	14-Nov	7	Brandon, Jireh, Nick								
1.2 Purchase Components	14-Nov	7-Dec	23	Brandon								
1.3 Construct & Test Circuits	28-Nov	30-Jan	64	Brandon								
1.3.1 Calibrate Sensors	28-Nov	18-Dec	21	Brandon								
1.3.2 Relay Design/Testing	19-Dec	9-Jan	22	Brandon								
1.3.3 Waterproof Sensors/Camera	10-Jan	30-Jan	21	Brandon								
1.4 Design & Create Encasing	31-Jan	27-Feb	28	Jireh, Brandon								
1.4.1 Design Using SolidWorks	31-Jan	9-Feb	10	Jireh, Brandon								
1.4.2 3-D Print Design	10-Feb	27-Feb	18	Jireh, Brandon								
2. Network Interfacing	**7-Nov**	**18-Jan**	**72**									
2.1 Microcontroller and Coding	7-Nov	15-Dec	38	Jireh, Nick, Brandon								
2.1.2 Python coding Sensors	14-Nov	15-Dec	31	Jireh, Brandon								
2.2 Router/ Networking connection	21-Nov	28-Dec	37	Jireh								
2.2.1 Connect a router to RPi	21-Nov	7-Dec	16	Jireh								
2.3 SMS	28-Dec	11-Jan	14	Jireh, Nick								
2.4 Web Integration	7-Dec	18-Jan	42	Jireh, Nick								
3. Website/Server Setup	**14-Nov**	**21-Feb**	**99**									
3.1 Generate psudo Code	14-Nov	15-Dec	30	Nick, Jireh								
3.2 Generate HTML format	16-Dec	15-Jan	32	Nick								
3.2.1 Debug HTML	20-Dec	20-Jan	31	Nick, Jireh								
3.3 Generate PHP Code	30-Dec	20-Feb	21	Nick								
3.3.1 Debug PHP Code	31-Dec	21-Feb	58	Nick, Brandon								
3.3.2 build login and authentication	1/1/2018	2/5/2019	31	Nick, TU								
3.3.3 Test Database access	2/10/2019	2/20/19	10	Nick								
3.4 Verify Storge and Database	2/1/2019	2/20/2019	19	Nick								
3.4.1 verify storage setup	2/3/2019	2/10/2019	7	Nick								
3.4.2 Setup tables	2/10/2019	2/17/2019	7	Nick								
3.4.3 Setup pathways to/from storage	1/10/2019	2/17/2019	37	Nick								
3.4.4 Debug	2/1/2019	2/21/2019	20	Nick, Jireh								

Table 5 - Gantt Chart

To be Done by January 25 2019

The objectives to be completed by the 25th of January are as follows: By December 14 first phase of parts are to be ordered, and then delivered by the next week December 21st. First draft of the webpage shall be designed and initial framework put in place for testing by December 28th. Initial waterproofing will be completed by December 19th, where the first build of the sensor array will then be tested. Final Prototype array will be completed by January 10th. where development will begin moving online. Relay design and prototyping should also be done by January 10th. Website infrastructure will begin December 28th, and user login and basic data display should be completed by January 20th. The remaining time will be spent debugging errors in the site, and documentation review for meeting.

List of Tests

Summary of Tests (FT. = Functional Test)

Below is a summary of tests that have been conducted so far.

Test Number	Objective	ER to address	Notes
FT.1	Camera Test	ER.18.1, ER.18.2	Complete
FT.2	Broadcastable Camera	ER.13.1, ER.18.1, ER.18.2	Complete
FT.3	Working Relay	ER.16.1	Complete
FT.4	RPi Updating and Transmitting Data Properly	ER.13.1, ER13.2	Complete
FT.5	Test SMS Alert System	ER.5.1, ER.5.2	Complete
FT.6	Startup Tests and Connecting to a Network	ER.3.1, ER.8.2	Complete
FT.7	Test the ADC on the Raspberry Pi	ER.1.4	Complete
FT.8	Test that the Light Sensor is Usable and Accurate	ER.1.3	Complete

Table 6 - Functional Tests

Description of Tests

FT.1: Test that the video from the camera is readable and storable by the Raspberry Pi

Test Setup:

In order to see if the device could record any footage from the Waveshare Raspberry Pi IR-Cut Camera, the team needed to first configure the camera settings on a Raspberry Pi. A Raspberry Pi 3 was used for this test, as the Raspberry Pi Zero had not come in yet. The configuration and setup is the same for both of the Raspberry Pi's however.

Figure 2 - Camera Test Setup

Configuration:
The team had to run a command (shown in the figure below) in the terminal on the Raspberry Pi in order to configure the ribbon port to read a camera.

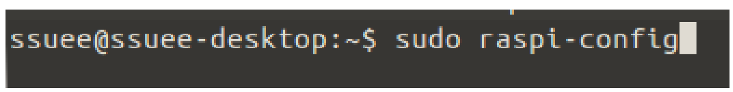


Figure 3 - Camera configuration script.

Then, a configuration menu was displayed. The team followed the menu and enabled the camera interface as shown in the following screenshots.

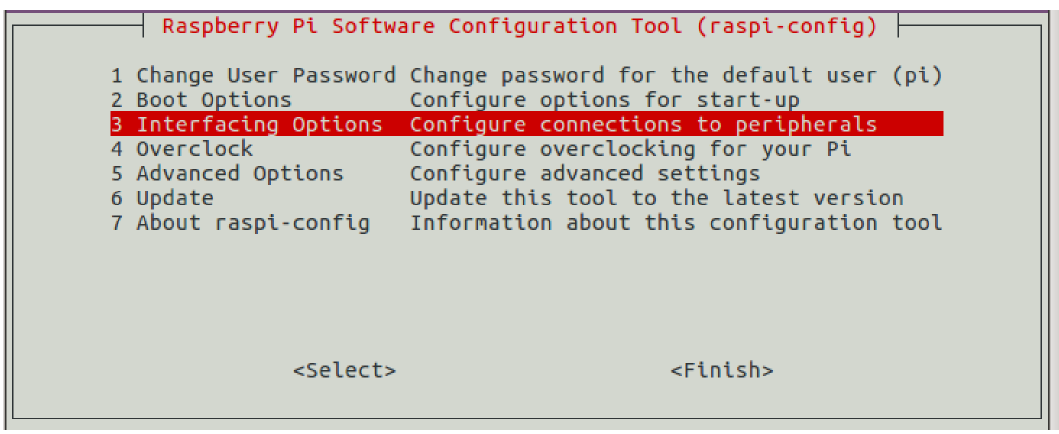


Figure 4 - Main Raspberry Pi configure menu.

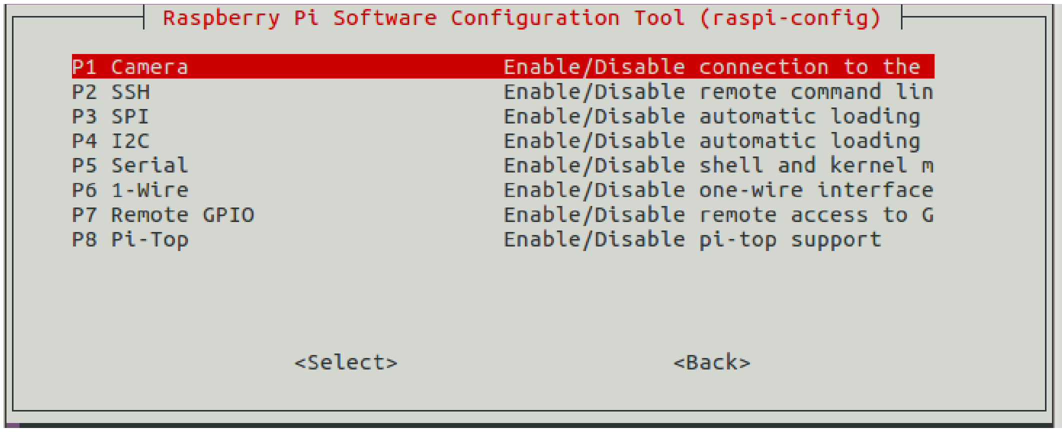


Figure 5 - Interfacing Options menu

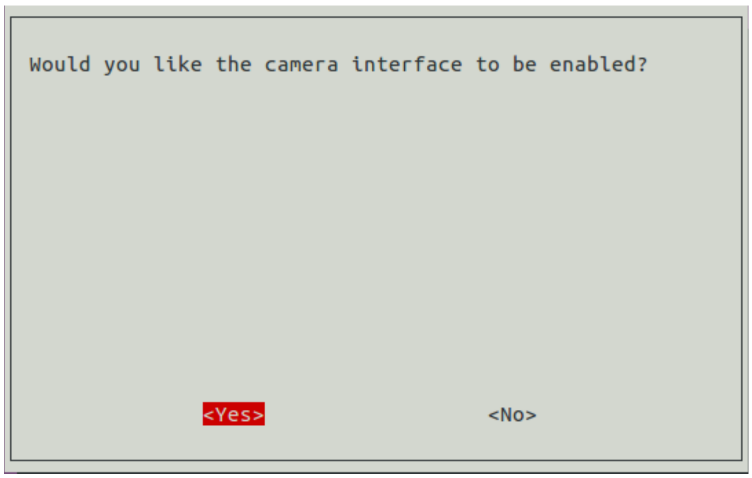


Figure 6 - Camera enable screen

After the camera interface was enabled, enable the infrared lights on the camera. To do that go into the 'config.txt' file and uncomment the line of code that is highlighted in the figure below.

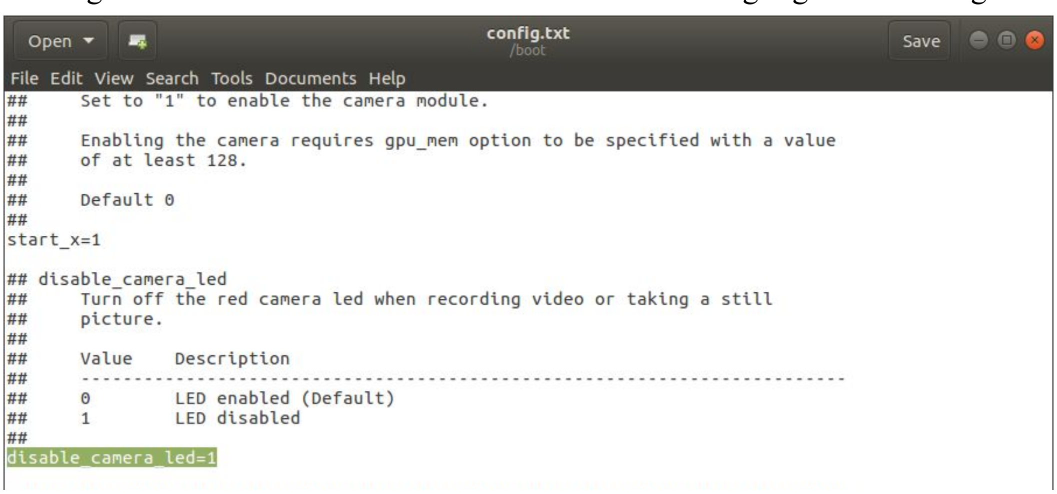


Figure 7 - IR light configuration

Once the camera interface and infrared lights were enabled, take test pictures and videos to see if the camera was actually readable and that the pictures and videos were storable on the Raspberry Pi as well.

Results:

Run the command shown below to take a picture.

```
ssuee@ssuee-desktop:~$ raspistill -t 2000 -o image.jpg
```

Figure 8 - How to take a single photo

The -t option provides a delay before the picture is taken, and the -o option allows the user to name the picture and to save it as whatever type of image file they please.

The image is automatically stored in the home folder of the Raspberry Pi.

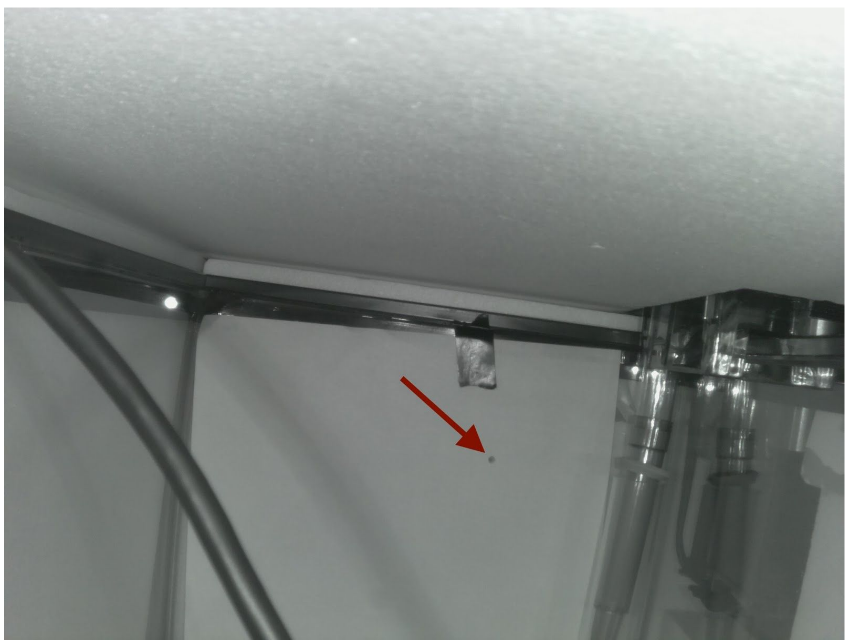


Figure 9 - Fish egg size test

Conclusion: The image shown above is a picture that was taken from within the dark conditions of the tank. The Infrared lights on the camera allowed the picture to be clear in the dark conditions. The dot that the arrow is pointing to is a representation of a trout egg. Trout eggs range in size from 3.5 mm to 4.5 mm. A 4 mm dot was drawn on a piece of paper and placed on the opposite side of the tank in comparison to where the camera was placed. This distance is about 1.5 ft. Even in the dark conditions of the tank and with the distance between the mock trout egg and camera, the camera took a clear picture for users to observe.

Requirements Satisfied: ER#18.1,18.2

FT. 2: Test that the video from the camera is viewable online.

Test Setup: To test that the video from the raspberry pi camera is broadcastable, an FTP server, called aitislab.com, was used. The video file and the html file were both uploaded and written to the server.

Figure 10 - Video Test Setup

Results:
http://aitislab.com/engineering/Brandon_Barron/plotchart/webtest.html

Conclusion: The Camera is Broadcastable
Requirements Satisfied: ER#13.1, 18.1, 18.2

FT. 3: Test which relay will work.

Test Setup: The Omron G6B-2214P-US 5DC Relay was tested. According to the datasheet, the rated current is 40 mA and the rated voltage is 5 volts for the coil inside of the relay. To find the operating point of the relay, a power supply was hooked up to the relay and voltage was applied to it until an LED connected to the other side of the switch turned on. The minimum voltage and current for the relay to switch was 3.0 volts and 40 mA.

A GPIO port from the Raspberry Pi was tested to see if it could supply enough power to activate the switch in the relay. Using a multimeter, while a GPIO pin was set to high, the GPIO port was measured with an output of 3.148 V and 12.89 mA across a 220 ohm resistor. Although the output voltage is sufficient to drive the switch, the GPIO port does not supply enough current. Instead of using a transistor as a current driver to supply the relay with enough current in order to activate the switch, a more cost effective way was to set three GPIO ports to high and

connect them in parallel as the input of the relay. This would allow 38.68 mA to pass through the relay and activate the switch.
Shown in the figure below is the relay design tested:

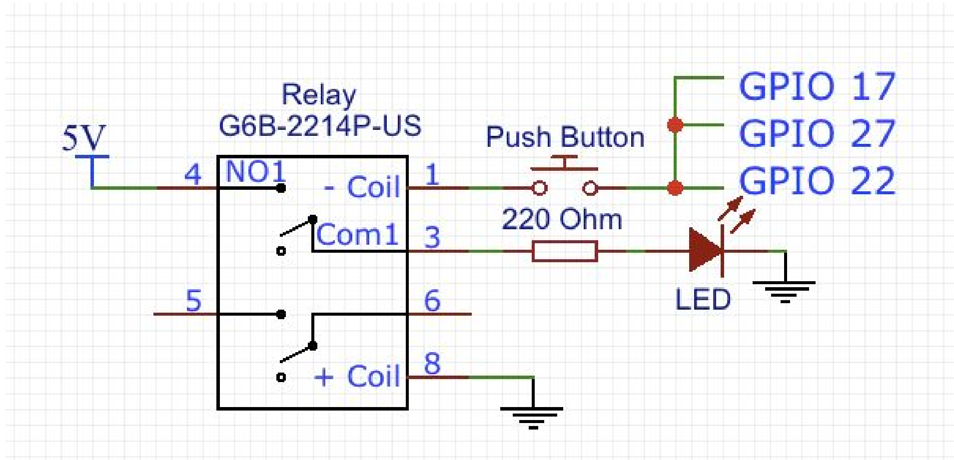


Figure 11 - Relay Design

Results: When GPIO 17, GPIO 22, and GPIO 27 were set to high and the button was pressed, the switch was activated and resulted in the LED to turn on.

Conclusion: This concludes that the relay works for what was needed.
Requirements Satisfied: ER#16.1

FT 4: Test that the RPi will update data and transmit it properly.

Test Setup: The most important piece of this project is transmitting data from the RPi to the Trout Lab web server. At this time however the team does not have a web server so extra space on a teacher web server was used to test the transmission of data over a wifi network to a remote server. Because of this the RPi is able to run a test script simulating several fake sensor values. The script, given below, creates random values at the push of a button and then uses the a basic linux command to send those values. Request library mentioned below sends a get request to the website with the variables between the ? and the & symbols. The PHP code on the site given by that URL the site takes the url and pulls the variables and inserts the to a CSV document for now. Then on another web page the values stored inside the CSV file are able to be read and graphed. The web page displays the values and a graph of their outputs for selected variables.

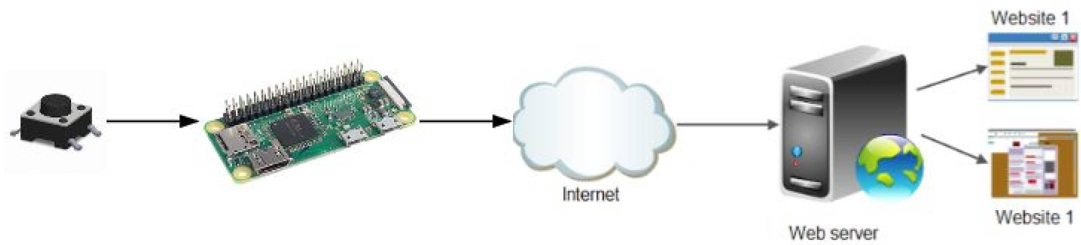


Figure 12 - Data Transmission Test

Results: The code successfully sends the fake data from the RPi to the web server. Where the PHP script places it into the proper areas of the CSV and the other PHP script, shown below, pulls and displays the values appropriately.

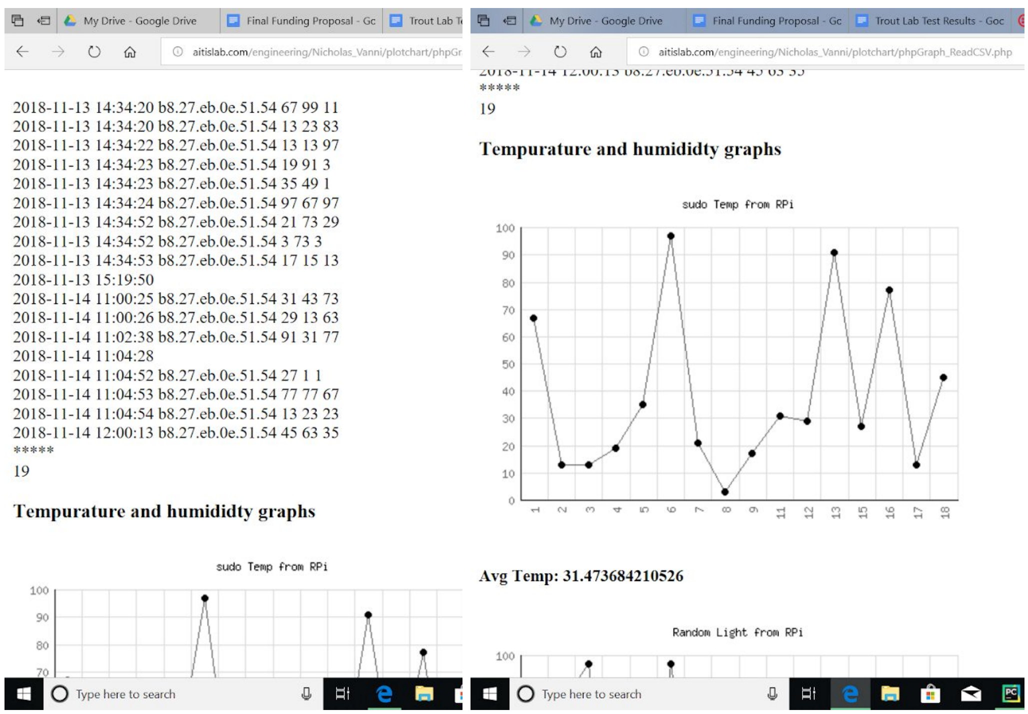


Figure 13 - Data on a webpage with graphs

Conclusion: These values are stored with total accuracy and then graphed appropriately underneath the displayed data.

Requirements Satisfied: ER# 13.1 & 13.2

FT 5: Test SMS alert system

Test Setup: Under a set of given values or RPi or web site will alert the primary user when the tank reaches critical values or does not transmit for a selected period of time. To do this employ the Twilio API and libraries to send messages over this network as long as there is a connection to a strong enough wireless system. Inserting the given script to a python enabled device allows it to transmit a text message to the appropriate user in a timely fashion via the Twilio API and SMS network. The most basic forms of the utilization scripts are given by the company to any user who seeks to test or purchase their services. Below is the most basic form of the twilio python script used in this test. Where the twilio library is imported and the account information is given under the account sid and the authentication token. And stored as the client information. Then the message.create command is run using the users phone number under the to section, the company number under the from section and the doby section delineated the message.

Figure 14 - Twilio Test Setup

Results: Using the sample code they provided for Python, SMS texts were able to be sent from an RPi to the user's phone. The messages arrived in under a minute from the time the message was sent from the RPi over the internet and cellular network.

Network reliability	Average	Poor	Low	Low	Poor	Good	Good	Average
Time so receive	2.39 s	90.10 s	5.02 s	6.78 s	55.46 s	1.95 s	1.35 s	2.75 s

Table 7 - Twilio Response Time

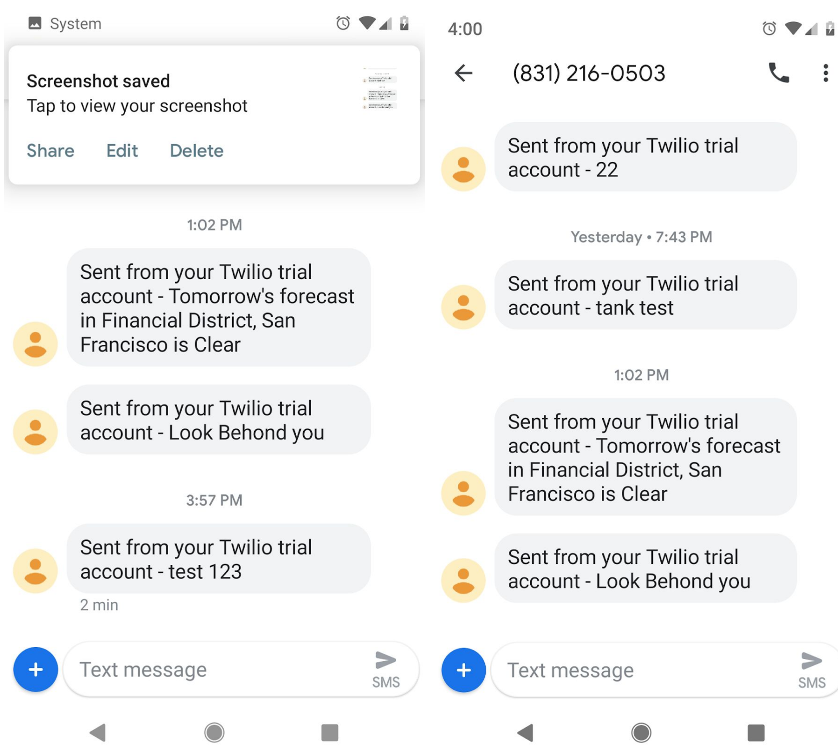


Figure 15 - Twilio Test Messages

<u>Conclusion:</u> Depending on the equality of the network in an area the team able to get pretty quick responses but as expected. When the network quality drops the response time of the message does drastically.

<u>Requirements Satisfied:</u> ER#5.1 & 5.2

<u>FT 6:</u> Startup tests and connecting to a network:

<u>Test Setup:</u> A cornerstone in this project is using wireless connectivity to broadcast the data to the Trout Lab web server where the user can easily access and view the data. Without this Trout Lab does not have the same capabilities as the competitors. First enable the RPi Zero W with WiFi, to do this, simply activate the WiFi module inside of the RPI to accept connections. Then add credentials to the system and it will validate those credentials and connect to the WiFi.

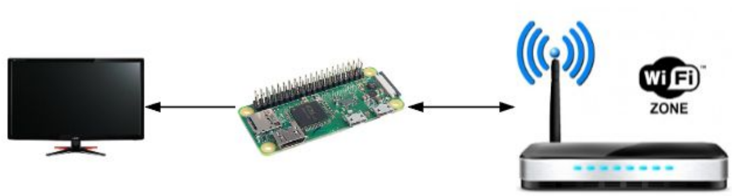


Figure 16 - Startup Test Setup

Results: The RPi was powered off and on at various times under various circumstances to check its abilities to establish and maintain a connection. The test will be performed at various times when traffic varies to help us ensure that the RPi will get connected to the wireless network.

Date and time	11/18/2018 2:15 PM	11/18/2018 5:00PM	11/19/2018 1:13PM
Time to connect to the network	15 Seconds	18 Seconds	13 Seconds
Expected network traffic	None	Little	Medium

Table 8 - Startup to Connection

Conclusion: Regardless of network traffic it is part of the RPi's startup sequence as long as you have credentials you can connect to the local area network so that the device operates safely.
Requirements Satisfied: ER# 3.1 & 8.2

FT 7: Test Analog to Digital Converter on Raspberry Pi

Test Setup: In order for the sensors to read data from the Raspberry Pi Zero, the analog data transmitted from the sensors needs to be converted into digital data. The ADC Pi addon was necessary for this since the Raspberry Pi Zero did not have the capability of converting analog data to digital data. The ADC Pi is an 8 channel 17 bit analog to digital converter designed to work with the Raspberry Pi. The ADC Pi is based on two Microchip MCP3424 A/D converters each containing 4 analog inputs. ABElectronics provided a library of code to help configure the ADC Pi addon to the RPi Zero. The library also provided code to test the voltage readings from each analog pin. This code was used with some minor adjustments to test the ADC Pi. To set this up, the ADC Pi addon was connected to the RPi Zero, and the RPi ran the code to read the voltage of each pin of the ADC Pi. Voltage was applied to each pin using a power supply and recorded the voltage through each pin from the RPi Zero and a multimeter for accuracy.

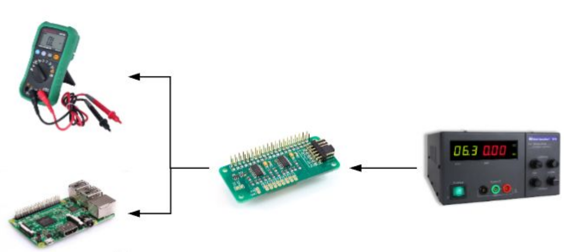


Figure 17 - ADC RPi Test Setup

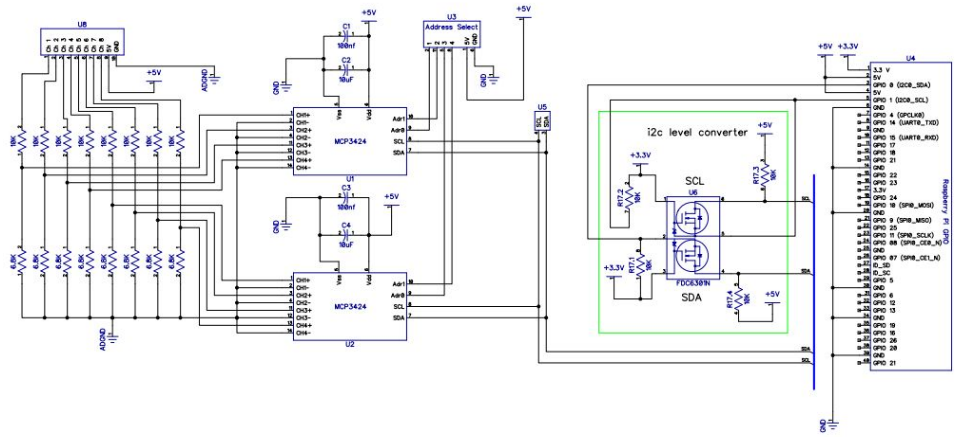
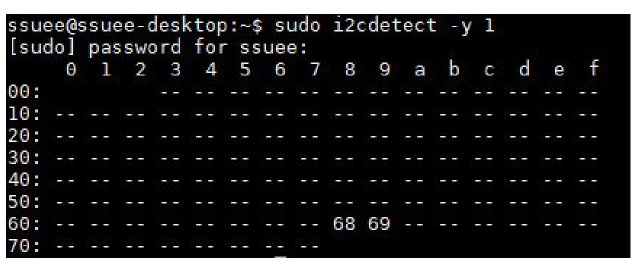


Figure 18 - ADC RPi Wiring Schematic

The ABElectronics Python library was downloaded and installed on the RPi in the terminal using the command:
git clone https://github.com/abelectronicsuk/ABElectronics_Python_Libraries.git

After that, the python library was installed by navigating into the ABElectronics library and running the command: sudo python3 setup.py install . The library also required python-smbus to be installed using the command: sudo apt-get install python3-smbus. Then the ADC Pi demo code was used to read the voltage to test an Analog to Digital converter. The code was modified so that when a button is pushed the data of the voltage from each pin will be updated. There was an error at first while running the code because the I2C function needed to be enabled on the RPi. To install it, the following command was used: sudo apt-get install -y i2c-tools. After testing the ADC Pi was connected through I2C on an RPi. The following shown is the test for the I2C.



```
ssuee@ssuee-desktop:~$ sudo i2cdetect -y 1
[sudo] password for ssuee:
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- 68 69 -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Figure 19 - RPi I2C Test

The code to test the voltage going through each analog pin is: demo_readvoltage.py

```
#!/usr/bin/env python
from __future__ import absolute_import, division, print_function, \
                                                    unicode_literals
import time
import os
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
# Set pin 17 as the INPUT
GPIO.setup(17, GPIO.IN)
try:
    from ABElectronics_Python_Libraries import ADCPi
except ImportError:
    print("Failed to import ADCPi from python system path")
    print("Importing from parent folder instead")
    try:
        import sys
        sys.path.append('..')
        from ADCPi import ADCPi
    except ImportError:
        raise ImportError(
            "Failed to import library from parent folder")
```

```
def main():
    '''
    Main program function
    '''

    adc = ADCPi(0x68, 0x69, 12)

    while True:
        if (GPIO.input(17)):
            # clear the console
            os.system('clear')
            # read from adc channels and print to screen
            print("Channel 1: %02f" % adc.read_voltage(1))
            print("Channel 2: %02f" % adc.read_voltage(2))
            print("Channel 3: %02f" % adc.read_voltage(3))
            print("Channel 4: %02f" % adc.read_voltage(4))
            print("Channel 5: %02f" % adc.read_voltage(5))
            print("Channel 6: %02f" % adc.read_voltage(6))
            print("Channel 7: %02f" % adc.read_voltage(7))
            print("Channel 8: %02f" % adc.read_voltage(8))

            # wait 0.2 seconds before reading the pins again
            time.sleep(0.2)

if __name__ == "__main__":
    main()
```

Figure 20 - ADC Read Voltage Code

Results: After running the code, the voltage of each pin was recorded. Five different voltages were tested for accuracy. The voltage of each pin was tested ten times and recorded to find the average of every pin at each voltage applied.

```
Channel 1: 1.010639
Channel 2: 0.000000
Channel 3: 0.000000
Channel 4: 0.000000
Channel 5: 0.000000
Channel 6: 0.000000
Channel 7: 0.000000
Channel 8: 0.000000
```

Figure 21 - ADC Voltage Output

Voltage	Pin 1(V)	Pin 2(V)	Pin 3(V)	Pin 4(V)	Pin 5(V)	Pin 6(V)	Pin 7(V)	Pin 8(V)
1V	0.98	0.993	0.99	0.993	0.991	0.993	0.993	0.995
2V	1.92	1.92	1.91	1.91	1.91	1.91	1.91	1.91
3V	2.98	2.89	2.87	2.89	2.88	2.883	2.883	2.881
4V	3.82	3.815	3.81	3.815	3.805	3.813	3.815	3.82
5V	4.75	4.752	4.74	4.75	4.739	4.747	4.749	4.749

Table 9 - ADC Voltage Data

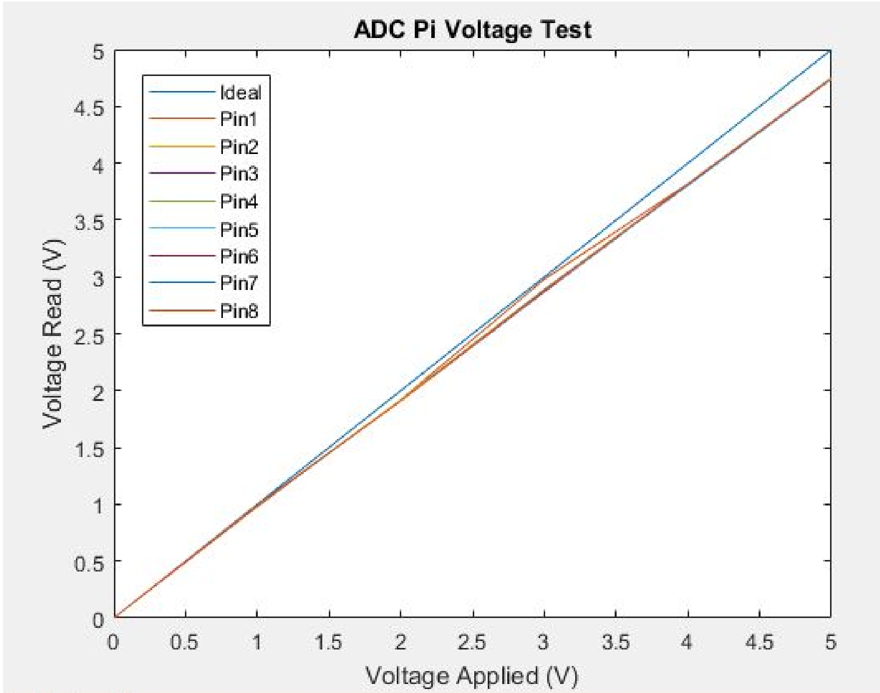


Figure 22 - Voltage Applied vs Voltage Read

Conclusion: As shown in the graph, as the voltage increases when applied to each pin, the less accurate the reading is. There is a -0.9% error of the average voltage of every pin at 1V and a -5% error of the average voltage of every pin at 5V.
Requirements Satisfied: ER# 1.4

FT 8: Test that an light sensor is usable and accurate

Test Setup: The chosen light sensor for the device is one made by Sunfounder. The brightness of the light detected determines what voltage is being transmitted. Using the ADC Pi demo_readvoltage.py code, the light sensor will be tested by shining different types of light towards the light sensor which will change the amount of voltage being transmitted through each pin. Attach the ADC Pi to the RPi Zero and assemble the light sensor and the button circuit to the board. Using the same steps as the test for ADC Pi, access the ADC Pi library and run the demo_readvoltage.py code. Record the luminous flux by using a Lux meter to record the lux, use the Luminous Flux equation by multiplying the lux by the surface area of the tank.

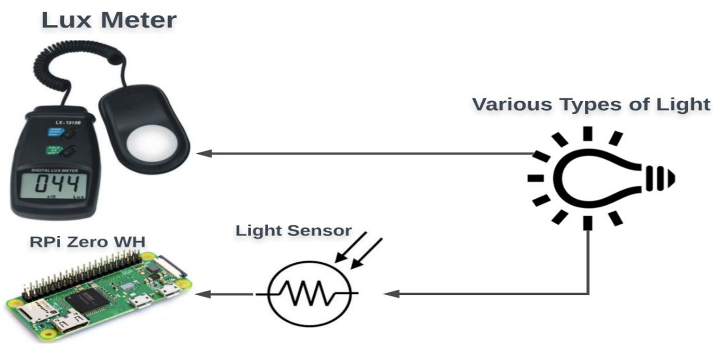
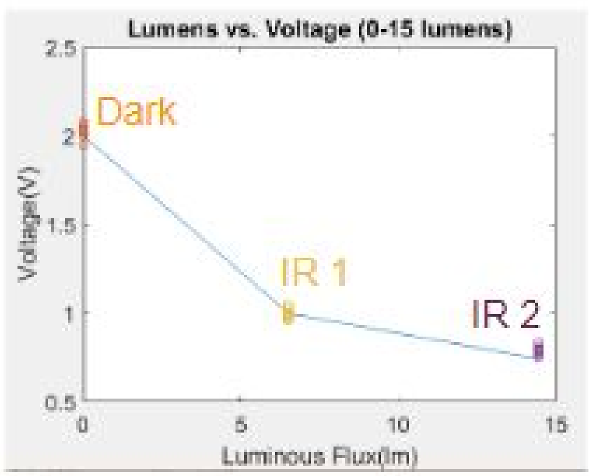


Figure 23 - Light Sensor Test Setup

Results:



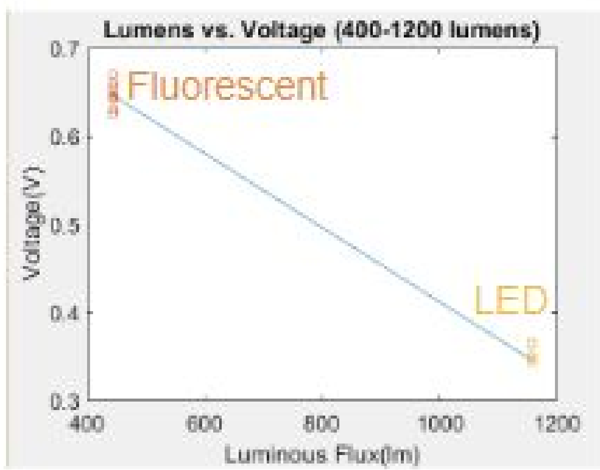


Figure 24 - Lumens vs. Voltage Graphs
This is a graph of the lumens vs the voltage of each condition.

Light/Lumens	Pin 1(V)	Pin 2(V)	Pin 3(V)	Pin 4(V)	Pin 5(V)	Pin 6(V)	Pin 7(V)	Pin 8(V)
Dark: 0 lm	2.02	1.947	2.13	2.15	2.07	2.00	2.273	2.125
Fluorescent: 444 lm	0.642	0.647	0.63	0.67	0.655	0.625	0.644	0.66
LED: 1157 lm	0.294	0.35	0.338	0.343	0.343	0.343	0.343	0.365

Infrared (1mm): 14.5 lm	0.714	0.602	0.657	0.696	0.775	0.753	0.835	0.86
Infrared (8cm): 6.5 lm	1.1	0.98	1.2	1.0	1.1	1.0	1.0	0.99

Table 10 - Light Sensor Voltage Data

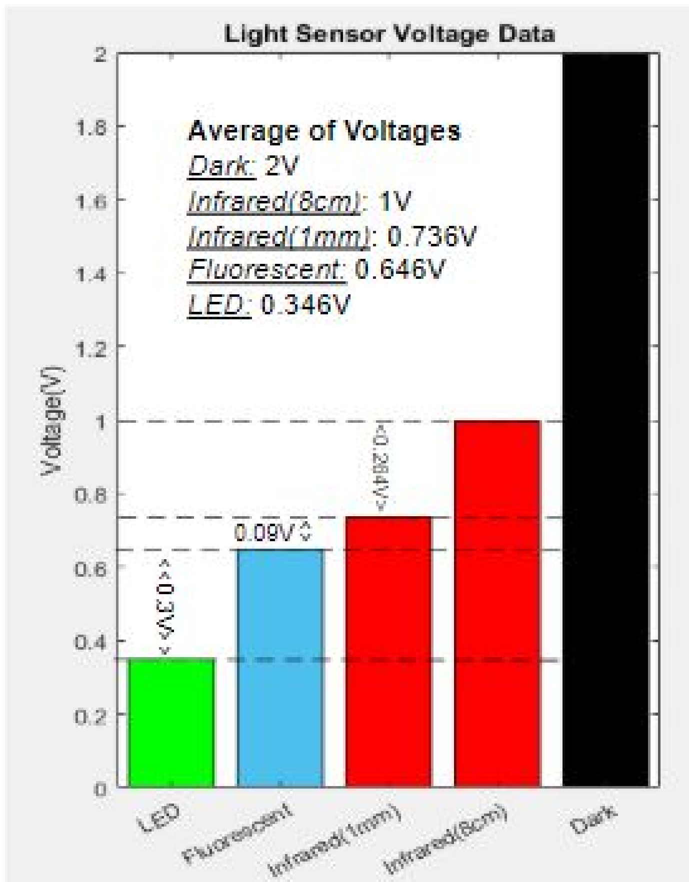


Figure 25- Light Sensor Voltage Data

Conclusion: It is clear that when the light sensor is exposed to darkness (lower lumens) the voltage applied through the pins is at an average of about 2V. The brighter the light and the more lumens, the less voltage is being applied through the sensor. When the light sensor is exposed to an LED, the voltage applied through the pins is at average of about 0.346V. The fluorescent and the Infrared exposure had a close average voltage with the fluorescent being at 0.646V and Infrared at 1mm away from the sensor being at 0.736V. The Infrared light about 8 cm away from the light sensor has an average voltage of about 1V. In conclusion, this data will be used to set a threshold of voltage applied determined by the amount of lumens absorbed by the light sensor.

Requirements Satisfied: ER#1.3 & 1.4

Ethics

An Ethical design is a design that has been worked hard through an establishment of requirements that regulate the morals of others [11]. The team will develop an ethical design of the project by fulfilling and meeting all of the IEEE Code of Ethics. For example, they will fulfill and meet the requirement of accepting responsibility in making consistent engineering decisions in regards to safety, health and welfare of the public, by making a safe system and child proofing it for the elementary school children who will use the device in their classrooms. One challenge they have encountered is that trout are affected by the three attributes of light intensity, spectral composition, and polarization [12]. Their main concern with trout being sensitive to light was about how they were going to record the fish in the dark with the camera [13]. The team researched and found out that a low intensity infrared light will be safe to expose on the trout while providing a dark enough environment for the trout. The device will also be sustainable and will have a modular design, so that if one part is damaged and cannot be repaired, the part that was damaged can be simply replaced and recycled without replacing the whole device. The last main ethical concern the team has towards the project is that they are not experienced in most of the applications that are being dealt with. In the event that a problem in which expertise is lacking, the team promises to seek help from someone with experience in the area. The design team has thoroughly reviewed all ten of the IEEE Code of Ethics and addressed all the concerns that could potentially break one of the requirements.

References

[1] - "Seneye Seneye." Aquarium Monitor System - Fish Tank Water Sensor - Seneye. Accessed September 24, 2018. https://www.seneye.com/.

[2] - "Seneye Reef" Aquarium Monitoring System - Seneye. Accessed September 24 2018. SENEYE Reef Aquarium Monitor and Par Meter https://www.amazon.com/dp/B005HR11W2/ref=cm_sw_r_cp_apa_mewQBbQV4Y7JH.

[3] - "Welcome to Fishbit." Fishbit - Smart Tools For Beautiful Aquariums. Accessed September 24, 2018. https://getfishbit.com/.

[4] - Adams, Jake, and Brian Blank. "Fishbit Controller and Monitor Come out of Beta, Ready for Primetime Gear, Marquee Reef Builders | The Reef and Marine Aquarium Blog." Reef Builders | The Reef and Marine Aquarium Blog. February 26, 2016. Accessed September 26, 2018. https://reefbuilders.com/2016/02/24/fishbit-controller-and-monitor-come-out-of-beta-ready-for-primetime/#.

[5] - "Tank Monitoring." SkyBitz. Accessed September 24, 2018. https://info.skybitz.com/lp-tank-monitoring/?gclid=EAIaIQobChMI54nQjarK3QIVk9dkCh3ZwwKqEAAYASAAEgIVYvD_BwE.

[6] - "Apex." Neptune Systems. Accessed September 24, 2018. https://www.neptunesystems.com/products/apex-controllers/apex-controller-system.

[7] - "High Ammonia Tolerance in Fishes of the Family Batrachoididae (Toadfish and Midshipmen)." NeuroImage. August 21, 2000. Accessed September 25, 2018. https://www.sciencedirect.com/science/article/pii/S0166445X99001010.

[8] - Schowalter-Hay, Ethan. "The Optimum PH Level of Water That Trout Can Live In." Accessed September 24, 2018. https://www.trails.com/facts_41600_water-that-trout-can-live.html.

[9] - Saltwater Aquarium. Apex "Classic" Lab Grade System - AquaController Base Package - Neptune Systems. https://www.saltwateraquarium.com/apex-classic-lab-grade-system-aquacontroller-base-package-neptune-systems/?gclid=Cj0KCQjw3KzdBRDWARIsAIJ8TMSwkdYuR4Lfz5XgqOAOxtrjmUksgYrAef4xgib8nhmZg0nOYT73pQIaAvNcEALw_wcB

[10] - "Trout in the Classroom: Activity Guide for Teachers." New Jersey. Accessed August 22, 2018. https://www.nj.gov/dep/fgw/pdf/tic_guide.pdf.

[11] - D. B. Estreich's Powerpoint slides on, "Professional Ethics," October 24, 2018

[12] - https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=2103&context=wnan

[13] - https://www.troutprostore.com/class/color_vision_trout_eyes