Sofishticated

A Smarter Fish Tank for Your Finned Friend

EECS 473

Fall 2019

Team members: Robert Cecil, Madeline DeVore, Ray Puyat, Celine Schlueter, David Waier

I. Introduction and overview of project

Fish are common household pets, both for beginning pet owners and long-time enthusiasts. Obtaining fish is simple, however, keeping them alive can prove challenging. Different breeds of fish require different environments, so it is crucial to monitor water temperature, acidity, and poisonous chemical levels. Monitoring these levels can be tedious, and confusing, so the task is easily ignored, especially by beginner fish owners. As a result, fish oftentimes live in improper environments, experiencing sickness and in some cases even death.

To make fish keeping easier, we created a product to monitor a fish tank ecosystem. The function of this product is to report temperature, pH levels, and concentrations of ammonia, nitrite, and nitrate to an iPhone app. The product is useful to beginners as well as experienced fish owners to monitor their fish tanks and improve the quality of life for the fish.

We met all of our milestone goals, which included reading sensor data and transmitting it to the iPhone app, handling AC to DC power conversion throughout the product, creating a custom PCB, and housing the product in a waterproof casing. However, we did not reach our stretch goals of creating a back-up power supply and making the casing resistant to saltwater.

At the design expo, we demonstrated our product working with a complete fish tank. The temperature and pH readings were timely and accurate, and updated on the app. The app provided feedback to the user on current tank health and allowed for customization for the desired ecosystem. The only issue that was encountered was a parsing error between the product and the app, during the color sensor data transmission. While the values were scanned and analyzed properly, the transmission was incomplete and sometimes incorrect.

II. Description of the Project

Goal

Sofishticated aims to provide users an easy-to-use and simple product to improve the care of their fish. Various fish require different water temperatures, pH levels, and habitats, and fish that live in improper environments usually suffer from poor health, and a reduced lifespan. Monitoring these values can be a chore, and can also be overwhelming to first-time fish owners. The aim of Sofishticated is to reduce the stress and confusion of aquarium care. Additionally, to make this product accessible to all fish owners, the price of the product must be kept as low as possible.

System Concept

Sofishticated keeps aquarium ecosystems at optimal levels for the enclosed fish, by closely and accurately monitoring temperature, pH, ammonia, nitrite, and nitrate levels. This data is transmitted to the user's iPhone app to provide live statistics of the fish tank environment, push a notification should a problem occur, and suggest appropriate actions on how to maintain the water environment close to its optimal level. A high-level design of the product is shown below.

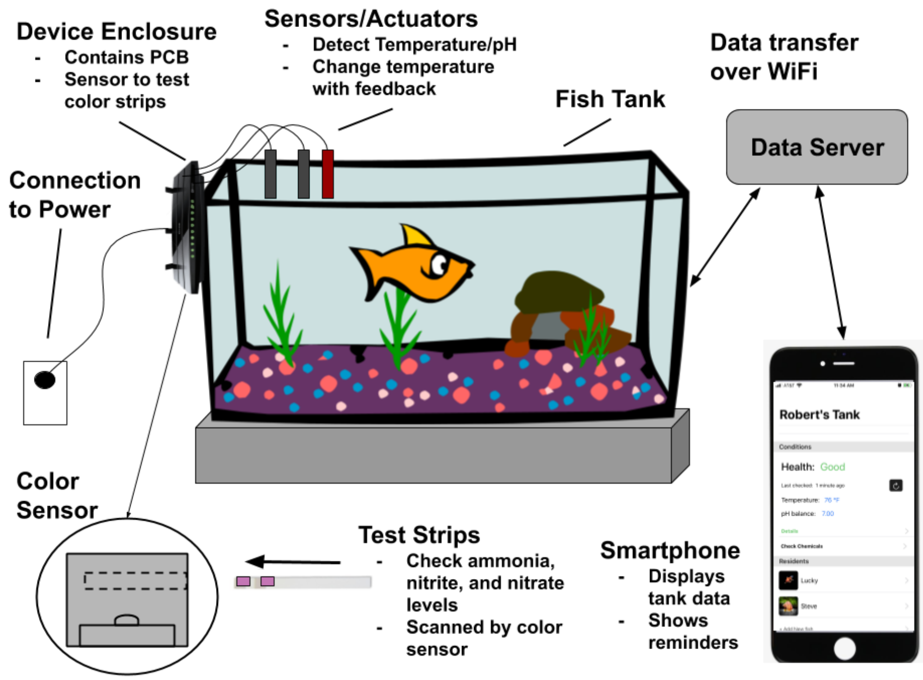

Figure 1: The Sofishticated product attaches to a fish tank and monitors temperature, pH, and chemical levels using various sensors. The temperature is also regulated through a water heater. This ecosystem data is transferred over a data server to a mobile app which displays current data for the fish tank.

Feasibility
The current version of the Sofishticated product is a fully-functional product that can be used for any freshwater fish tank. However, if this product were to be produced in greater quantities, the main issues to consider are:

- **Safety:**
  - The product is likely to come into contact with water, and the user and fish must be protected from electrocution.
  - The product manages 120V power lines, which must be contained and protected with fuses.
- **Manufacturing**:
  - The casing used for the product is a custom 3D printed component which is not feasible for mass production.
  - Components of the 120V circuitry have been customized by hand to fit within the casing, which is not feasible for mass production.

However, we believe that with further work, this product can be mass-produced. The casing design must be revised and finalized, to allow for simple construction and assembly, while also establishing more room for mass-produced 120V circuitry components. Finally, a greater budget and timeframe will allow for more research into effective methods for waterproofing the product, ensuring user and fish safety.

Design Constraints
The design constraints that were considered for the Sofishticated product are:
- **Budget:** We had a budget of $200 per person, and with a group of 5 people, the total budget was $1000. The Sofishticated product was designed to fit within this budget. In the end, we used $574.32 of the $1000 budget.
- **Time:** Our development and production phases had a deadline of about 4 months (Fall 2019 semester), so the design of the final product depended on the timely feasibility of interfacing with and integrating individual parts. Parts with established documentation were chosen to meet this deadline.
- **Size:** Our final product had to be small and portable to hang along the edge of a standard fish tank. In the end, our product was 22.9 cm by 15.2 cm. This is small enough to fit a 10-gallon fish tank.
- **Power:** The product is powered through a wall outlet, as it is the most common power source that a common household has access to.

- **Safety:** The product enclosure must be waterproof to protect the user and the fish. Additionally, the 120V circuitry must be well contained and protected through fuses.

System Architecture

Below is a high-level overview of the various components that interact throughout the device.

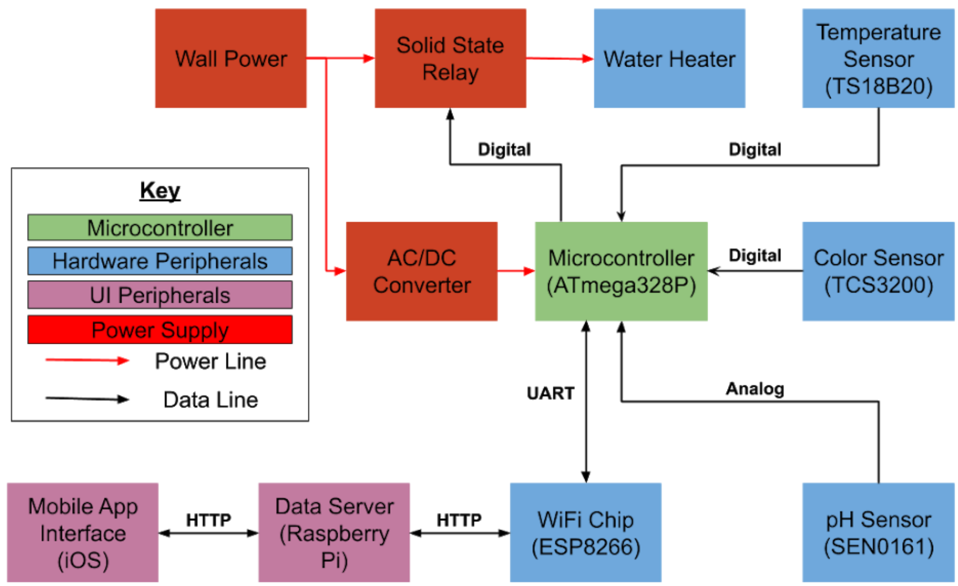

Figure 2: High-level overview of the interactions between various product components. The central "brain" is the microcontroller, which controls the sensors and Wifi module. This wifi module communicates with a mobile app over a data server. The product is powered by a 120V line which splices to power the water heater and the microcontroller at 5V.

The main components of the design include:

- **Microcontroller:** The ATmega328P acts as the central controller, as it receives data from the sensors by analog and digital input/output lines. The microcontroller also exchanges data with WiFi chip through UART.
- **WiFi Chip:** The ESP8266 WiFi chip communicates with the microcontroller through UART, relaying HTTP requests from the mobile app and Data Server to the microcontroller for data requests and specific actions.
- **Sensors:** The temperature, color, and pH sensors analyze the ecosystem and report data back to the microcontroller over analog or digital signals.
- **Solid State Relay:** The solid-state relay, which controls the water heater, is controlled through a digital signal from the microcontroller.

- **iOS Mobile App:** The ecosystem data is transferred over HTTP to the data server, which relays this information over HTTP to the mobile app. In the mobile app, the user can view current and historical data, test for chemical levels, and customize their ecosystem.

This assembled product is shown below.

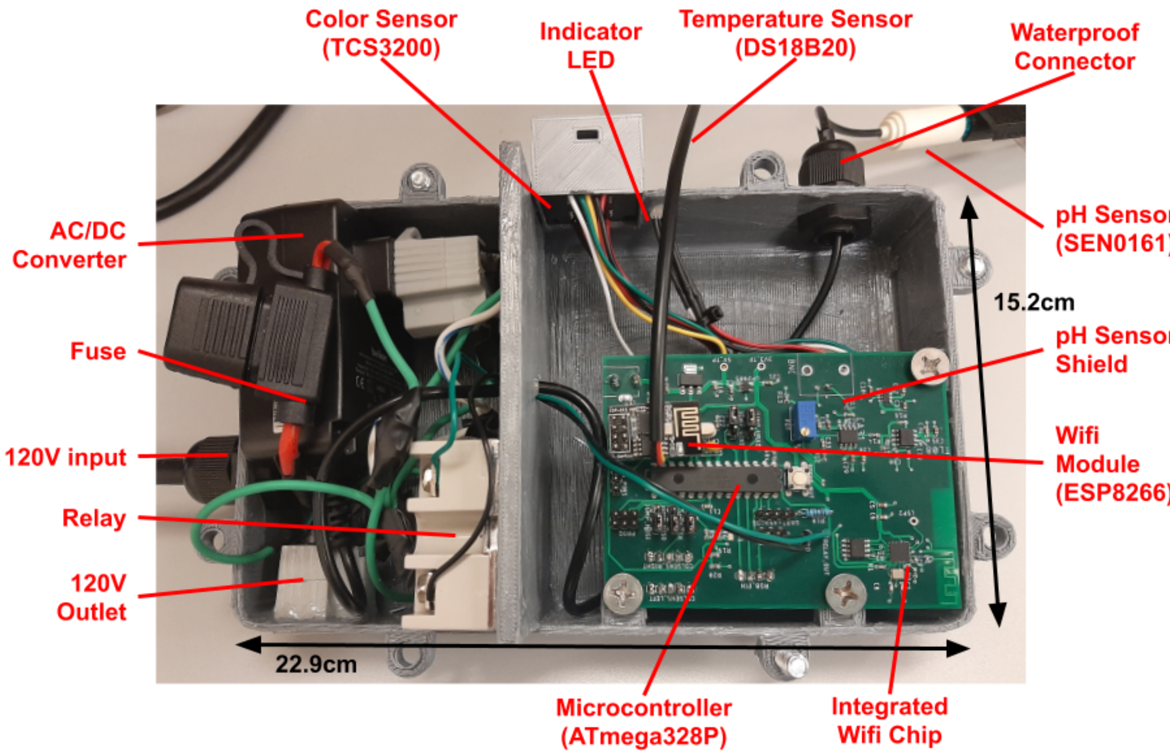

Figure 3: The assembled Sofishticated product. The left side of the casing contains the 120V circuitry and the solid-state relay. The right side contains the custom PCB that has the microcontroller, and that connects to the Wifi module and all sensors.

III. Milestones, schedule, and budget

In our original schedule (Figure 4), we planned to complete most deliverable items a few days before each milestone was due to account for debugging time and to move on to other action items. When executing our action items, we managed to complete each deliverable before the corresponding milestones, but after the original deadline we set for ourselves. Our PCB design was completed on October 31st and was delivered the same day as our milestone 2 meeting, so we were unable to start integration of the PCB system until November 13th when we finished soldering the PCB. The integration of the system with the PCB went smoothly. We were able to test different areas of the PCB by having multiple test programs written to test certain sensors and communicating what needed to be accounted for in order to test the full system. This testing required coordination between people working on different sensors so that we had the RTOS set up correctly and did not comment out code necessary to the function of each sensor. We also had to communicate on what debugging output we were using because what worked for debugging the wifi and server communication would change how the wifi and color sensor interact with each other.

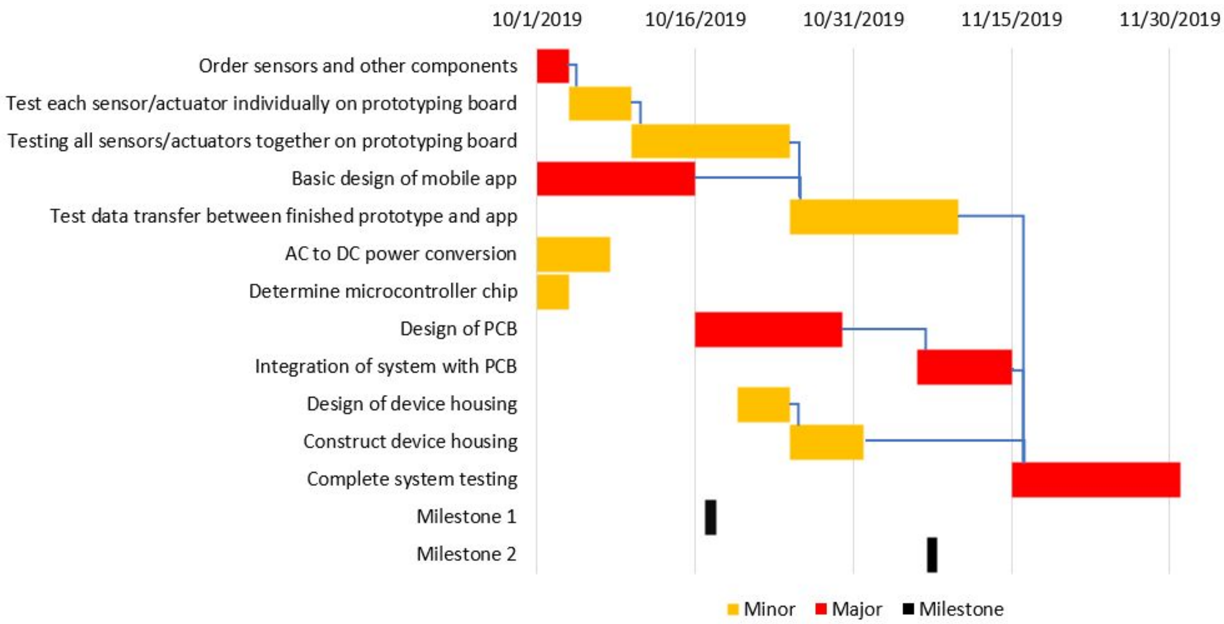

Figure 4: The milestones as outlined in the project proposal.

Milestone 1 Deliverables:

- Read data from each sensor individually
- Mobile app basic interface completed and testable (does not include communication with device)

Milestone 2 Deliverables:

- Device housing constructed
- Data transfer between system and mobile app

We greatly underestimated the time required to create an app, and because of a recent update in Swift, there was a greater learning curve than anticipated. A lack of thorough documentation also slowed down app development. The deadlines for our deliverables were met, but the increased development time decreased the time for testing communication between the product, the server, and the app.

Problems with the Wifi chip arose that took time to resolve and factored into delays with communication between the product and mobile app. We were able to construct a preliminary device housing by milestone 2 but had to make revisions and completed our device housing a week later. These delays were largely failures in printing or basic CAD problems that were easy to resolve.

In order to stay on track, we implemented a system of regular check-ins and work sessions where most team members could attend, which helped with communication and debugging.

Table 1: Detailed original budget for creation and testing Sofishticated

Item	Vendor	Quantity	Cost/Item ($)	Shipping Cost ($)	Total Cost ($)
Wall Adapter Power Supply	Sparkfun	1	5.95	0.00	5.95
Temperature Sensor	Adafruit	1	9.95	0.00	9.95
Color Sensor	DFRobot	3	7.90	10.00	33.70
Ammonia Test Strips	Amazon	1	7.15	0.00	7.15
Nitrate/Nitrite Test Strips	Amazon	1	15.99	0.00	15.99
Ammonia	Target	1	1.19	0.00	1.19
Nitrate	Amazon	1	15.09	0.00	15.09
Nitrite	Grainger	1	26.37	10.00	36.37
Sulfur	Amazon	1	14.99	0.00	14.99
pH Sensor	DFRobot	1	30	2.13	32.13
RGB LED (pack of 100)	Amazon	1	8.99	0.00	8.99
Wifi Module	Sparkfun	1	6.95	0.00	6.95
Tap Water Conditioner	Amazon	1	6.64	0.00	6.64
Temperature Sensor	Adafruit	1	9.95	0.00	9.95
PCB	Unknown	3	40.00	15.00	135.00
Casing	EECS Dept.	1	75.00	0.00	75.00
ATmega328P	Amazon	1	24.99	0.00	24.99
Fish	Petco	10	3.00	0	30.00
Fish Tank Supplies	Petco	1	50.00	0	50.00
Fish Tanks	Petco	2	15.00	5.00	35.00
Total					**554.94**

Our original budget for this project was about $550. This budget includes shipping cost for some of our items we anticipated having to order online as well as extra parts we thought would be useful for testing. After purchasing, all of the supplies for our project our total is $574.32 including all of the shipping required for parts ordered online. The difference in our budget from our final total results from needing to purchase hardware to supplement and waterproof our enclosure, parts needed to handle high AC voltages, and additional shipping costs that we had not accounted for. Overall, we were close to our original budget and went over because of a decision to add a high voltage system after turning in our original proposal.

IV. Lessons learned

What went well

Throughout the project, our greatest strength was the ability to cooperate to meet deadlines that we set for ourselves. Everyone collaborated to complete research, debug code, integrate the project, and finalize the end product. As a result, we were able to stay on schedule, besides the issues explained earlier, and complete our product within the given timeframe. We were also able to balance the workload between the project and the associated documentation.

All of us experienced learning and growth from this project. Robert took the lead on the iPhone app and setting up the data server, developing his skills in Swift and general app development. David and Madeline managed the PCB design and assembly, gaining experience in how to create various components, manage power, solder, and test a PCB. Ray worked extensively with the pH sensor, allowing him to create custom code for our final product that allows us to incorporate the V1 shield with the V2 sensor. Celine managed the color sensor and learned about controlling ambient light. She also learned how to use Blender to create 3D models for the product casing with help from David. Everyone on the team worked on integration and documentation, allowing all of us to grow these skills.

What went poorly

The major drawback of this project is the distribution of tasks. Based on how we had completed our EECS 373 project, we split up the tasks by general components. However, this division proved to be a rather uneven split, as the sensors varied in complexity, and app development was greatly underestimated. Robert is the only team member familiar with app design, and we failed to train more team members in this skill, resulting in Robert completing all of the iPhone app work. The greatest improvement our team could have made to our organization is the division of tasks. Some team members were overloaded, so taking the time to train more people in various areas would have been highly beneficial.

V. Contributions of each member of team

Team member	Contribution	Effort
Robert Cecil:	I was tasked with configuring the User Interface and wireless communication portion of our project. This involved creating and developing an iPhone app from scratch, implementing a wireless server application, and programming an ESP8266 W-Fi chip with the appropriate functions. I then configured a data pipeline for all these components to communicate with each other via Wi-Fi and HTTP responses. In addition, I maintained communication with team members working on other components, to integrate the wireless system with the PCB and ensure that my iOS interface would communicate with the system seamlessly. Finally, I engaged in user interface research and found ways that my app would be most effective in aiding a fish owner in raising their fish.	20%
Madeline DeVore:	At the beginning of the project, it was decided that I would be in charge of team scheduling and managing work sessions. This included coordinating work sessions for people working on codependent parts of the projects, learning enough about each system to help out at work sessions and keeping up team morale. I worked with Celine to manage timelines and make sure we worked on what we said we would do when we said we would do it. This helped keep us on track for milestone reviews and project integration. For the actual project, I worked on the wifi module, designing the PCB, soldering, and system integration. Specifically, programming and debugging the wifi module hardware and integrating the code for communication between the wifi module and the server. Working on research for communication between the esp8266 and our server. I also collaborated with David to design the schematic, layout and debug the board, make the parts list and solder the PCB. For software, I worked on writing software for the wifi module communication with the PCB and the server and helped debug software for the pH sensor.	20%
Ray Puyat:	I was tasked with the research, development, and integration of the pH sensor. This included choosing a pH sensor module that not only fits our given budget but also one that we can easily interfaced with and has established documentation written for it. In addition, I assisted during the development of the PCB by providing the schematic for the pH sensor as well as keeping the multitude of parts organized for the ones soldering. Software-wise, I helped in the code integration for both the pH sensor as well as the temperature sensor as the pH sensor relies on the data that the temperature sensor gathers. Also, I helped resolve and debug RTOS-specific issues in our code. As for our documentation deliverables, I completed my assigned parts in the initial proposal, final proposal, user manual, and final report diligently. Lastly, I assisted in the physical integration and assembly of the fish tank that we	20%

	presented in Design expo.	
Celine Schlueter:	At the beginning of the project, it was decided that I would be in charge of team documentation. As a result, I prepared the Milestone reports, and took the lead on the project proposal, final report, user manual, and poster. I laid out timelines for these documents, worked to assign sections to the team, and organized the team to meet these deadlines. For the physical project, I began by working on the color sensor. I learned how to program it, test it, and integrate it into the final system. I also created the overarching FreeRTOS code that runs on the PCB. I learned how to set up the Atmel Studio environment for our team to develop and test on the final PCB. Next, I learned how to use Blender, with the generous help from David, to design and 3D print the casing. This task included creating a separate section for the 120V components, a mount for the PCB, and a special light-insensitive enclosure for the color sensor. I also researched and implemented methods to make the case water-resistant. Finally, I worked on the overall integration and testing on the PCB.	20%
David Waier:	I was tasked with researching the temperature sensor, PCB design, soldering, AC power research, and full system integration. This included choosing a waterproof temperature sensor for our fish tank, choosing the correct AC power components to handle 120V safely, and assembling the AC power wiring and solid-state relay to control the tank heater. I helped to design parts of the schematic with Madeline and laid out and ordered the PCB. I was assisted by Madeline and Ray to assemble and solder the PCB and have helped debug the PCB hardware. On the software side, I created the temperature sensor library and have helped debug the FreeRTOS parts of the software. For the 3D case design, I have helped resolve any Blender issues and tutored Celine on how to do basic CAD design.	20%

VI. Cost of Manufacture

It is estimated that about 7.2 million households in the US own freshwater aquariums (prnewswire). If Sofishticated were to continue as a business it would be reasonable that we would aim to produce at least ten thousand units. Based on the quote we received from circuithub.com, this could cost $435,948.73 which breaks down to a cost of $43.59/board. Of this quote, $28.72 or 66% of the total cost comes from assembly and the remaining third from the cost of the board and parts itself. For producing large quantities this quote does not feel reasonable. If we were to really try and make ten thousand or more boards then we would need to be able to reduce the cost of assembly per board so that it will not be the main determinant of production costs for the total system.

VII. Parts

Table 2: The final parts list and budget.

Item	Vendor	Quantity	Cost/Item ($)	Total Cost ($)
AC/DC Converter	Amazon	1	12.90	12.90
Temperature Sensor	Adafruit	1	9.95	9.95
Color Sensor	DFRobot	3	7.90	23.70
Ammonia Test Strips	Amazon	1	7.15	7.15
Nitrate/Nitrite Test Strips	Amazon	1	15.99	15.99
Ammonia	Target	1	1.19	1.19
Nitrate	Amazon	1	15.09	15.09
Nitrite	Amazon	1	9.99	9.99
Sulfur	Sulfur	1	14.99	14.99
pH Sensor and Module	DFRobot	1	54.50	54.50
RGB LED (100 count)	Amazon	1	8.99	8.99
Wifi Module	Sparkfun	2	6.95	13.90
Tap Water Conditioner	Amazon	1	6.64	6.64
PCB Parts	Mouser	1	84.58	84.58
PCB/Stencil	JLCPCB	5	7.00	35.00
5A Fuses (25 count)	Amazon	1	12.90	12.90
Fuse Holders (10 count)	Amazon	1	8.99	8.99
Solid State Relay	Amazon	2	9.78	19.56
ATmega328P	Amazon	1	24.99	24.99
pH Buffer Solution	Amazon	1	19.99	19.99
Cable Glands	Amazon	1	9.49	9.49
Lexan	HomeDepot	1	3.28	3.28
Lexan Cutter	Home Depot	1	5.21	5.21
Zip ties	Home Depot	1	2.90	2.90
Grounding Adapter	Home Depot	4	0.79	3.16
6mm lock nut	Home Depot	2	0.60	1.20
#6-32 lock nut	Home Depot	1	1.18	1.18
6mm screw	Home Depot	3	0.90	2.70
#6-32 screw	Home Depot	1	1.18	1.18
6M screw	Home Depot	1	0.85	0.85
¼” lock nut	Home Depot	1	6.57	6.57
8’ power cord	Home Depot	1	10.97	10.97

Kwik Seal Plus	Home Depot	2	4.98	9.96
¼” hex bolt	Home Depot	10	0.15	1.50
Fish	Pet Supplies Plus	10	1.98-4.98	12.63
Fish Tank Supplies	Petco	1	42.37	42.37
Fish Tank	Petco	1	10.90	10.90
LP3986-33DBVR	Mouser	10	0.48	4.80
Trimmer Resistors	Mouser	3	1.43	4.29
pH Sensor	Bangood	1	26.46	26.46
Mini Jumpers	Amazon	1	5.82	5.82
Caulk Cord Seal	Lowe’s	1	5.91	5.91
Total				**574.32**

VIII. References and citations

We would like to thank Stryker for sponsoring this project. Additionally we would like to thank Professor Mark Brehob, Matthew Smith, Rahul Gangwani, Owen Winship, and Steve Rogacki for their assistance throughout this project. Finally, we would like to thank Jeb Hamel for providing support and knowledge about fish care and maintenance.

Final PCB:

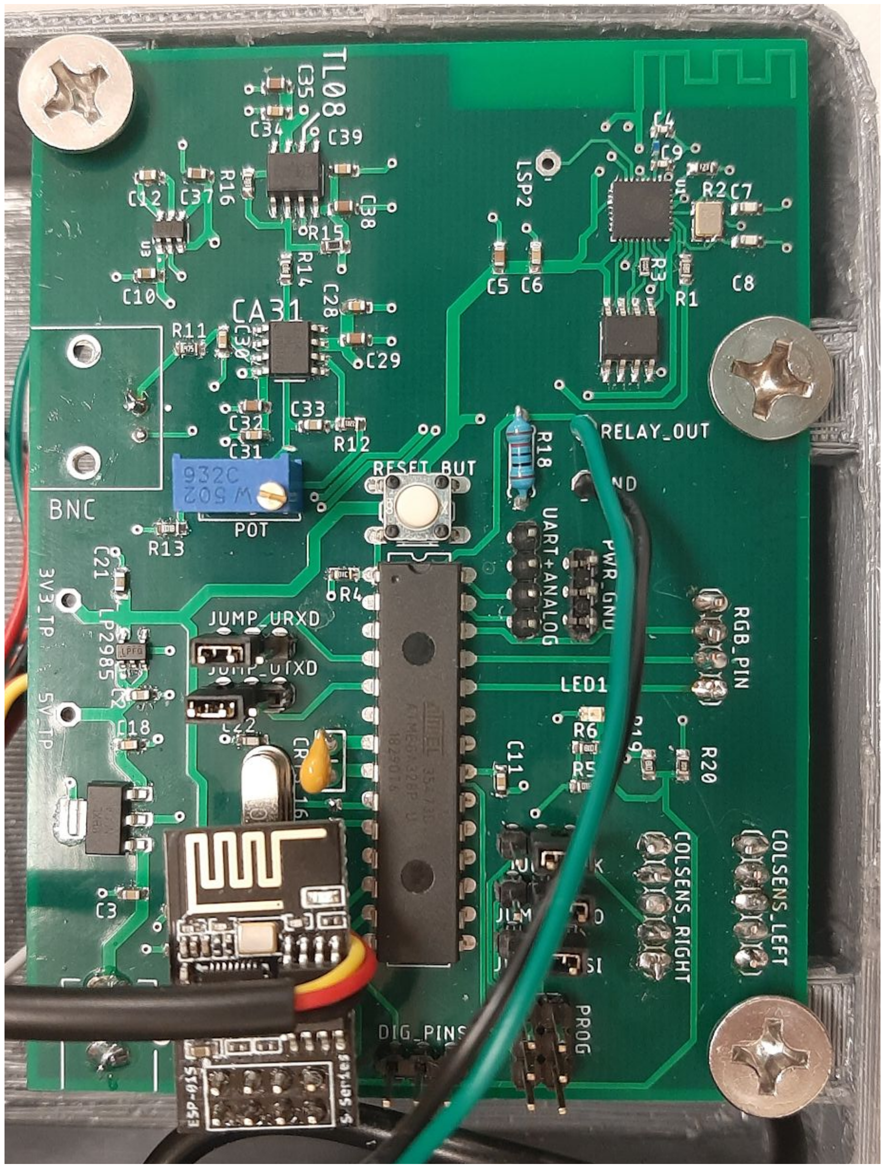

Figure 5: The final, assembled version of the PCB.

Completed interface code:

```
color_sensor.h
```

```
#include <MD_TCS230.h>    // color sensor library
#include <FreqCount.h>    // used by above library

// Pin definitions
#define  S2  12
#define  S3  13
#define  OE   8                  // LOW = ENABLED
```

```
#define MAX_AMMONIA_COLORS 5
#define MAX_NITRITE_NITRATE_COLORS 7
#define TOLERANCE 5              // How far out the red,green or blue can
be to match

// store RGB values of a Pixel
typedef struct
{
      int R, G, B;
}Pixel;

// Each color associated a Pixel to a ppm value
typedef struct
{
      double ppm;
      Pixel p;
}Color;

// types of test strips to analyze
enum ReadType {
      AMMONIA,
      NITRATE,
      NITRITE,
      FIND_TEST_STRIP
};

// add possible color strip values to specific arrays
void setupCS();

// check if scanned color matches a specific stored color
bool SameColor();

// go through all stored colors of a certain chemical to find a match
double FindMatch();

// get current color value from sensor
long ScanColor(ReadType r);

// check if we are seeing a valid test strip, not just an empty box or
the blank area of the test strip
bool findTestStrip(ReadType r);
```

---

indicatedLED.h

```
// Pin definitions
#define  R_LED  A1
#define  G_LED  A2
#define  B_LED  A3

// RGB values for each color the LED can display
```

```
typedef struct
{
	int r, g, b;
} LEDColor;

// colors for LED
LEDColor Red = {255, 0, 0};
LEDColor Green = {0, 255, 0};
LEDColor Blue = {0, 0, 255};
LEDColor Raspberry = {255, 255, 255};
LEDColor Cyan = {0, 255, 255};
LEDColor Magenta = {255, 0, 255};
LEDColor Yellow = {255, 255, 0};
LEDColor White = {255, 255, 255};
LEDColor Off = {0, 0, 0};

// set LED to a specified RGB color value
void setLED(LEDColor c)
{
	analogWrite(R_LED, c.r);
	analogWrite(G_LED, c.g);
	analogWrite(B_LED, c.b);
}

// initialize LED pins and turn off LED
void setupLED()
{
	pinMode(R_LED, OUTPUT);
	pinMode(G_LED, OUTPUT);
	pinMode(B_LED, OUTPUT);
	setLED(Off);
}
```

---

phSensor.h

```
#ifndef phSensor_h
#define phSensor_h

#include <Arduino.h>

#define PH_PIN A0

// return the pH value calculated from the analog voltage
float calcPH();

#endif
```

```
tempSensor.h
```

```
#include "OneWire.h"

// measureTemp() returns the measured temperature in degrees Celsius
float measureTemp();
```

Online references used to test/build our product:

https://www.xtronical.com/coloursensing/

https://wiki.dfrobot.com/Gravity__Analog_pH_Sensor_Meter_Kit_V2_SKU_SEN0161-V2

https://wiki.dfrobot.com/PH_meter_SKU__SEN0161_

https://www.espressif.com/sites/default/files/documentation/esp8266_hardware_design_guidelines_en.pdf

https://www.hackster.io/harshmangukiya/how-to-program-esp8266-with-arduino-uno-efb05f

https://cdn.sparkfun.com/datasheets/Wireless/WiFi/Command%20Doc.pdf

https://cdn-shop.adafruit.com/datasheets/DS18B20.pdf

https://playground.arduino.cc/Learning/OneWire/

https://github.com/AppPear/ChartView