Auto Aquarium

ECE 445 Final Report

Team #51

Caleb Chow (calebyc2)
Sihun Hyun (sihunhh2)
Irfaan Attarwala (iia2)

TA: Pooja Bhangchandani

Spring 2022

Table of Contents

1. Introduction

1.1 Problem

Owning an aquarium requires extensive maintenance, especially for users who own tropical fish that require a very specific range of variables like water temperature or pH levels to survive. This requires the aquarium owner to constantly purchase testing kits and devices to monitor the quality of the aquarium water. Oftentimes, aquariums are put in hard to reach places within rooms, making it difficult to manually test the quality of the tank water.

1.2 Solution

The Auto Aquarium solution allows an automated experience in monitoring temperature, pH, and water quality (through conductivity). Our product is a modular, small, and easy to install design that does not intrude on the aesthetic of the aquarium. Our product consists of a set of sensors that relay the current temperature, pH, and conductivity of the aquarium water to a mobile app via WiFi. This app shows the user the current data, along with acceptable ranges of sensor data which correspond to specific types of aquatic life. In addition, an LED attached to the device lights up when any of the monitored variables is outside the acceptable range.

1.3 Physical Design

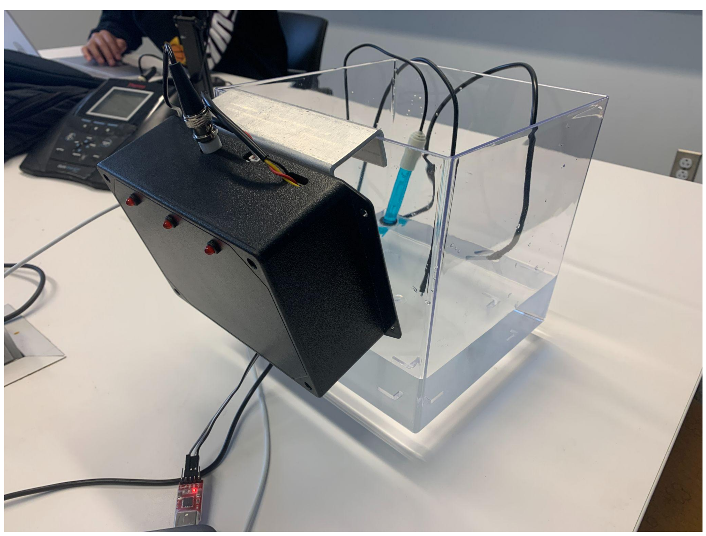

Figure 1: Physical design of the Auto Aquarium

1.4 Objectives

1. Create a cost-effective add-on to any aquarium that helps automate mundane maintenance tasks.
2. Alert owners in real time of any potentially hazardous living conditions in the aquarium via app and LEDs.
3. Combine monitoring of multiple variables (temperature, pH, conductivity) into one small product without any aesthetic intrusiveness.
4. Allow users to set tolerance levels for each of the variables using an app, and show the range of acceptable values for each variable based on the profile that the user sets.

1.5 High Level Requirements

- The device must be able to detect temperature, pH, conductivity within the tolerances listed:
    - Temperature: (±2 °C)
    - pH: (± 0.5)
    - Conductivity: (±20 ppm)
- The mobile app receives temperature, pH, and conductivity values from the wireless transmitter within 30 seconds. It must function to alert the user of dangerous conditions by sending a notification from the app and lighting an LED on the fishtank
- The device must be modular and portable, no larger than 20 inches on any side with a tolerance of 3 inches

2. Design

2.1 Block Diagram

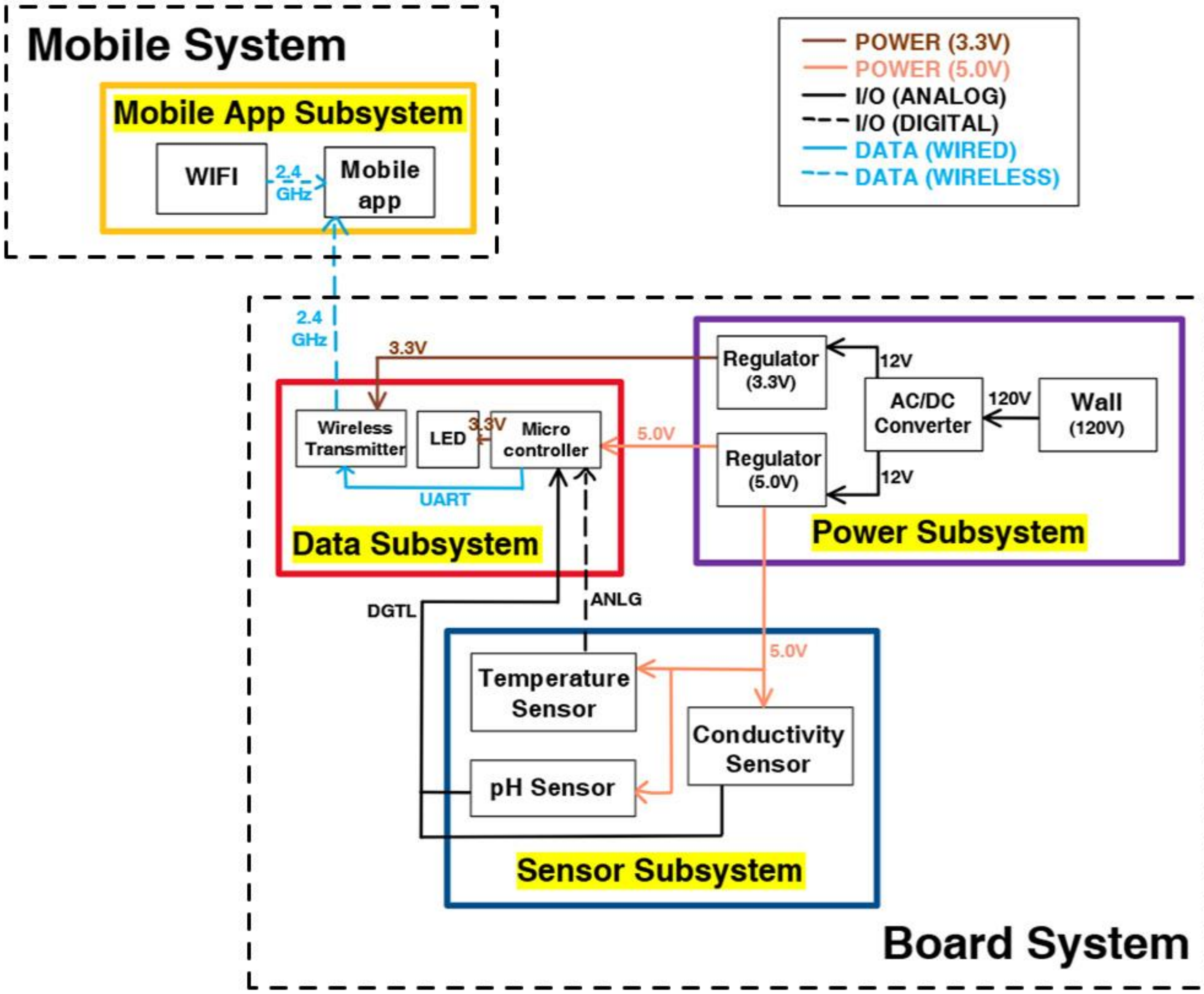

Figure 2: Block diagram of Auto Aquarium design with each subsystem and subsequent components

2.2 Block Descriptions

2.2.1 Sensor Subsystem

The sensor unit receives and manages the information from each of the sensors (temperature, pH, and conductivity). This subsystem communicates with the data module by constantly sending variable data to the microcontroller and receives power for each of the sensors from the power subsystem. Waterproofing every part of the subsystem is necessary because the sensors are submerged in water.

- **Temperature:** The Waterproof DS18B20 Digital temperature sensor constantly measures and reports the aquarium water's current temperature.
- **pH:** The SEN0169 pH sensor relays current pH levels of the aquarium water to be transmitted and displayed on the app. Typically, the acceptable pH ranges for tropical fish is between 6.8 and 7.8.

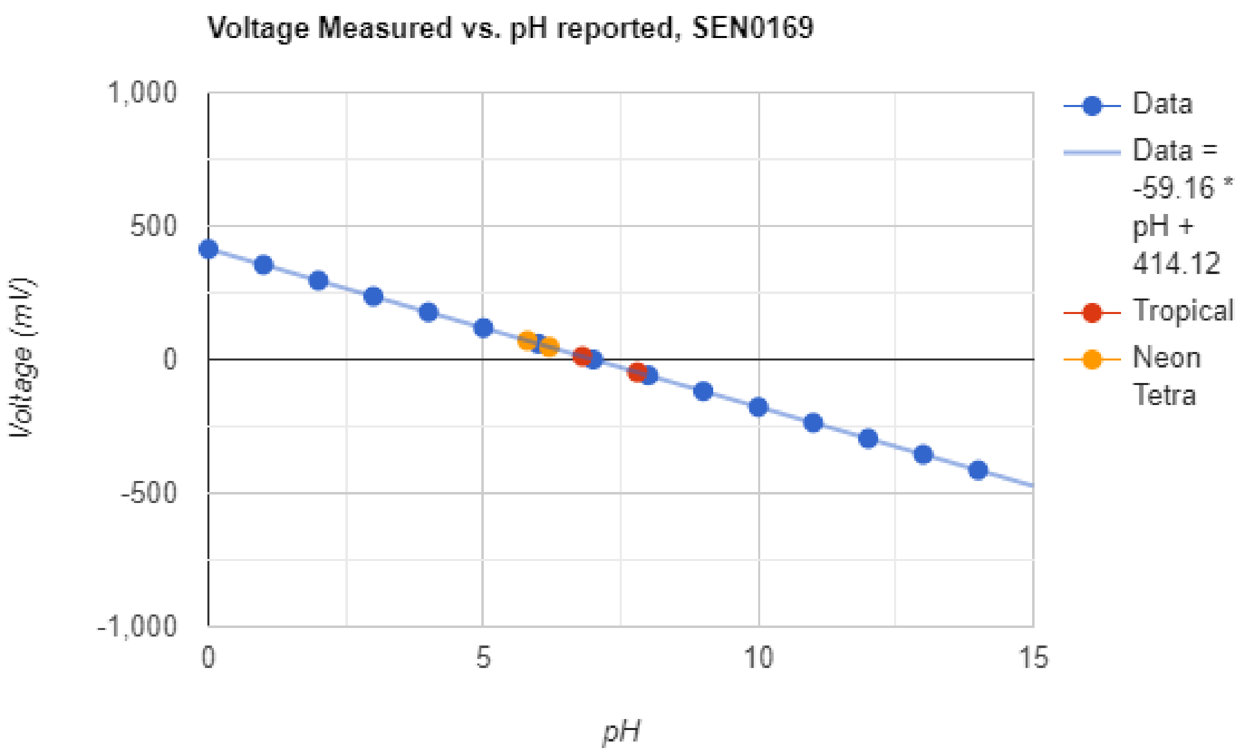

Figure 3: Example pH ranges for fish types

- **Conductivity:** The SKU SEN0244 water conductivity sensor measures the quality of the water based on the total dissolved solids (TDS) dissolved in one liter of the water. Pure distilled water is a poorer conductor of electricity than water with dissolved minerals [2].

2.2.2 Power Subsystem

The power subsystem provides 12V via an AC/DC converter to the components in the sensor and data subsystem. The converter derives 120V of AC power from a wall outlet and output 12V which enters both a 5V and 3.3V voltage regulator in parallel to provide power to the other subsystems. The sensors and microcontroller receive 5V power, while the wireless transmitter receives 3.3V power.

- 5V Regulator: LM7805
- 3.3V Regulator: LM1117-3.3
- AC/DC Converter: YU1201 Wall plug to barrel jack converter

2.2.3 Data Subsystem

The data subsystem manages the constant stream of data from the sensor subsystem and sends it via the wireless transmitter to the mobile app.

- **Microcontroller:** The ATmega328–MMH microcontroller receives 5V power from the power subsystem, and receives data from the sensor subsystem. It is responsible for communication via UART protocol to the wireless transmitter.
- **Wireless Transmitter:** The ESP8266 wireless transmitter receives 3.3V from the power subsystem, and receives processed data from the microcontroller. It then sends this information in an HTTP request via WiFi to the connected mobile app on a 2.4 GHz channel.
- **RGB LED**: The WP154A4SUREQBFZGW LED receives 3.3V from the microcontroller and indicates sensor values exceeding acceptable values (determined by user).

2.2.4 Mobile Subsystem

The mobile subsystem consists of a mobile app that connects to the data subsystem via a 2.4 Ghz signal. In the app, the various data (temperature, pH, and conductivity) are received via the wireless transmitter from the data subsystem. The data is fetched via API and HTTP requests, updating every 30 seconds at the maximum.

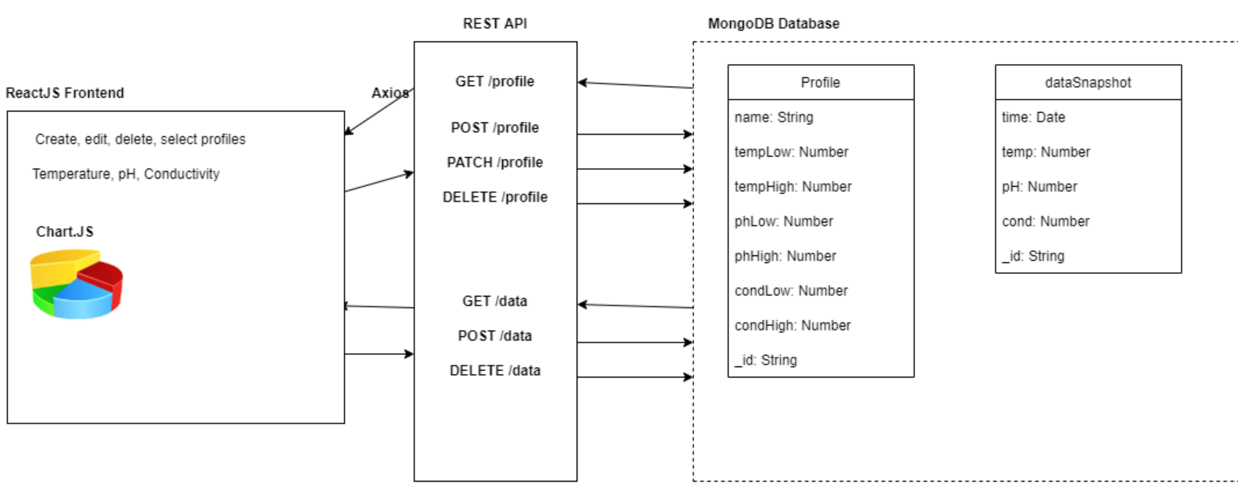

Figure 4: Block diagram of web app and database

- **Dangerous Conditions:** The app displays the received data along with the acceptable ranges based on the profile the user sets. When a variable is larger or smaller than the accepted range, the mobile app sends a notification to alert the user, as well as light up the LED in the data subsystem.
- **User Profiles:** The user can specify a profile, and for each profile can set the custom ranges for each variable (i.e. pH from 6.8 to 7.8). This user profile data is stored in a database.

2.3 PCB

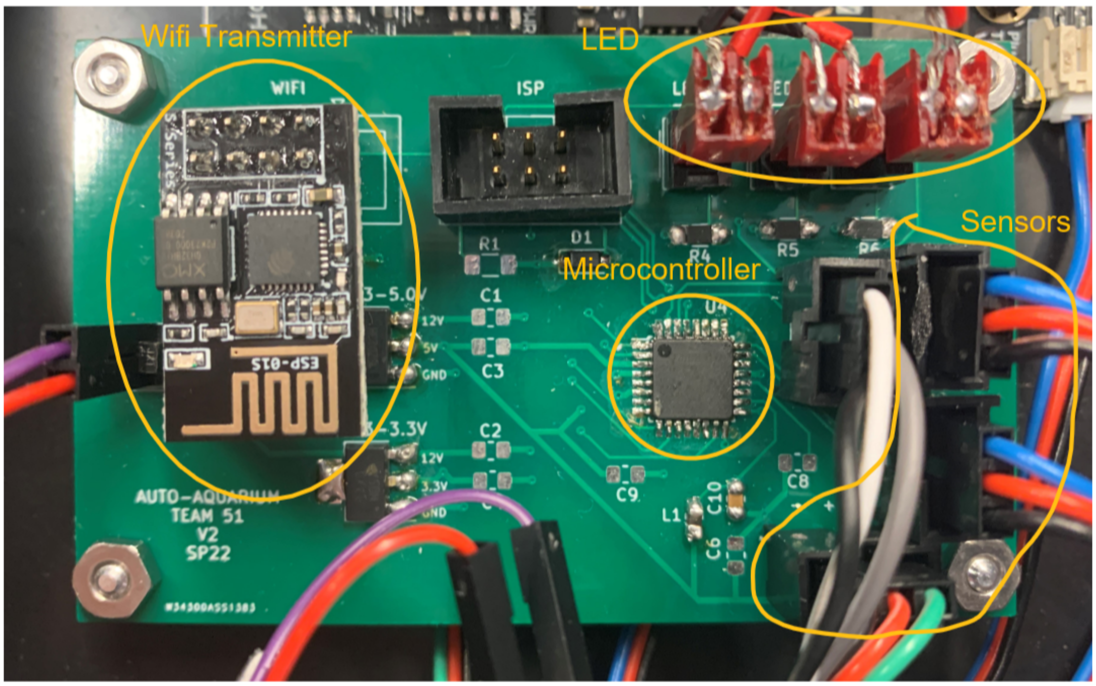

Figure 5: Version 1 PCBs

3. Cost and Schedule

3.1 Labor

Given a semester of 16 weeks, each of the three members worked on 15 hours per week on this project. Wage rates were given based on the average salary of an undergraduate student from UIUC with a Computer Engineering or Electrical Engineering degree. According to annual reports from Illini Success, a part of the University of Illinois, the average salary for a UIUC graduate with a computer engineering bachelor's degree is $99,145, or $47.67/hr. For a graduate with an electrical engineering bachelor's degree, the average salary is $76,129, or $36.60/hr [4].

Table 1: Labor cost analysis

Name	Caleb	Sihun	Irfaan
Wage ($/hr)	$47.67	$47.67	$36.60
Total Hours (in 16 weeks)	15 hrs/week x 16 weeks = 240 hrs	15 hrs/week x 16 weeks = 240 hrs	15 hrs/week x 16 weeks = 240 hrs
Individual Total	$11,440.80	$11,440.80	$8,784.00
GRAND TOTAL			**$31,665.60**

3.2 Cost of Parts

Table 2: Parts costs

Part Name	**Qty**	**Price per unit**	**Total**
Microcontroller (ATMEGA328-MMH)	1	$2.47	$2.47
Wireless Transmitter (ESP-8266)	1	$7.50	$7.50
pH Sensor (SEN-0169 DFRobot)	1	$29.50	$29.50
TDS Conductivity Sensor (SEN-0244)	1	$14.77	$14.77
Temperature Sensor (Waterproof DS18B20)	1	$9.95	$9.95
3.3V Regulator (LM1117-3.3)	1	$1.89	$1.89
5V Regulator (LM7805)	1	$1.50	$1.50
Wall Plug Adapter (YU1201)	1	$12.99	$12.99
RGB LED (WP154A4SUREQBFZGW)	1	$1.89	$1.89
GRAND TOTAL			**$82.46**

3.3 Schedule

Table 3: Design schedule

Week	Goals	Members Assigned
2/20 - 2/26	1. Finalize Design Document 2. Rough Draft for PCB	All
2/27 - 3/5	1. Design Review 2. Finalize PCB Design 3. PCB Board Approval	All
3/6 - 3/12	1. Solder PCB board, Power Subsystem 2. Develop mobile app frontend/database	1. Irfaan, Sihun 2. Caleb
3/13 - 3/19	Spring Break	All
3/20 - 3/26	1. Solder validation and PCB debugging 2. Connect frontend/database, finalize frontend design	1. Irfaan, Sihun 2. Caleb
3/27 - 4/2	1. Finalize PCB, begin sensor subsystem 2. Finalize mobile subsystem with sample data 3. Individual Progress Reports	1. Irfaan, Sihun 2. Caleb 3. All
4/3 - 4/9	1. Sensor subsystem 2. Waterproof sensor subsystem enclosure	All
4/10 - 4/16	1. Debug full subsystems (power, data, mobile, sensors) 2. Finalize demo procedure	All
4/17 - 4/23	1. Mock Demo 2. Final Presentation Rough Draft	All
4/24 - 4/30	1. Demonstration 2. Mock Presentation	All
5/1 - 5/7	1. Final Presentation 2. Finish Final Paper	All

4. Design Verification

4.1 Sensors

The verification process for the sensor subsystem consisted of checking sensor data against external sensors. For pH and temperature, we used a pH probe and a digital thermometer to ensure sensor readings were accurate. As shown in the images below, the sensors were able to return measurements well within initially set tolerances. Additionally, we verified the sensors were returning data within the maximum query time as shown in the second row of each reading in Figure 8.

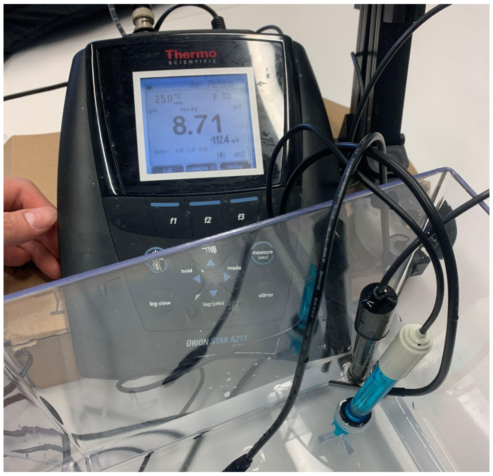

Figure 6: External pH measurement

Figure 7: External temperature measurement

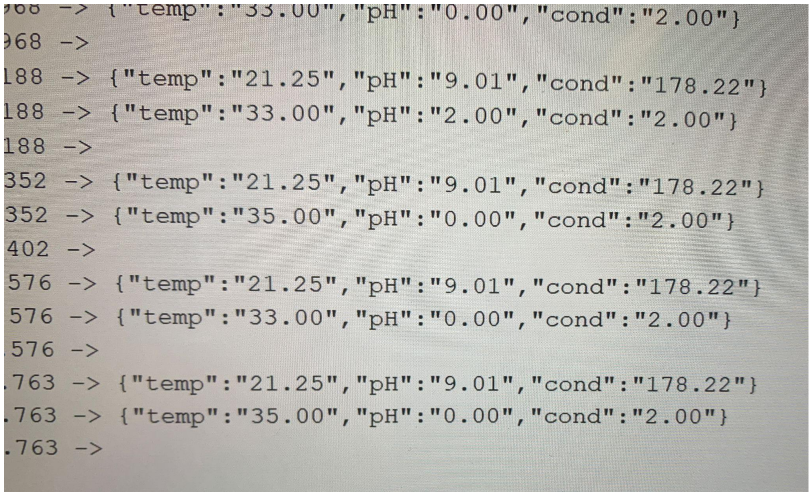

Figure 8: Sensor data

4.2 Power

Verifications for the power subsystem included ensuring that each part component on the PCB was receiving the proper voltage. Our design included an AC/DC converter, and two voltage regulators to achieve this. In order to verify this, we used a multimeter to measure the voltage across different portions of the PCB. From left to right, Figure 9 shows the correct voltage readings at the AC/DC converter, the 5.0V regulator, and the 3.3V regulator

Figure 9: Design schedule

4.3 Mobile

Verifications for the mobile subsystem included ensuring that dangerous condition notifications would be sent when the live data was outside of the ranges set. Figure 10 below shows the notification sent when the temperature and conductivity are greater than the acceptable range.

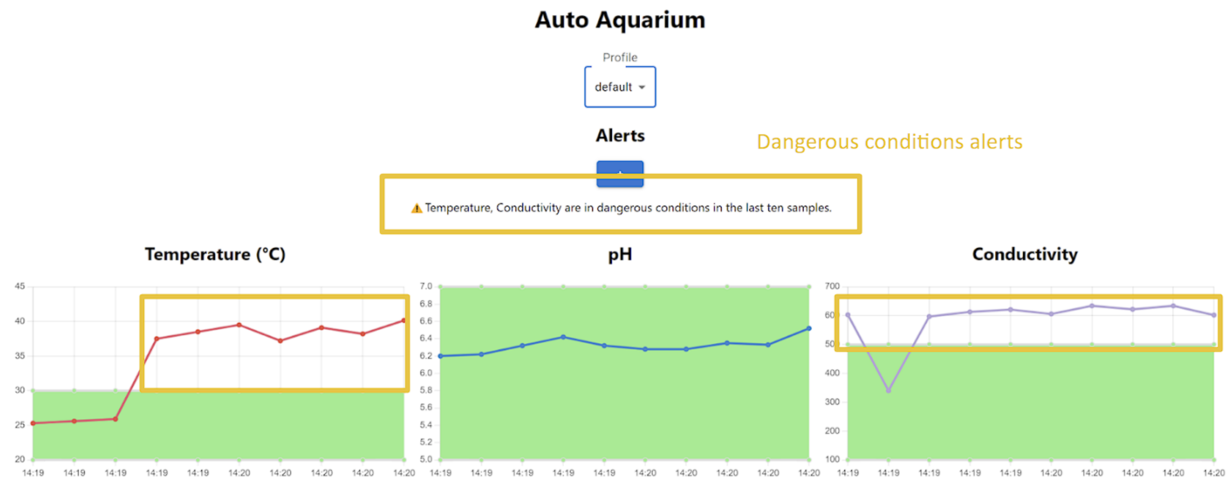

Figure 10: Design schedule

5. Conclusion

5.1 Accomplishments

Our product, the Auto Aquarium, was able to meet all three of the high level requirements listed in section 1.5. Our sensors were able to accurately detect temperature and pH within the margin of error that we specified. Unfortunately, because we were unable to obtain a TDS sensor, we were unable to properly calibrate the conductivity sensor. Our app, however, was able to detect dangerous conditions based on the live data received, and notify the user. Finally, our design was portable within the size restriction listed in our requirements.

5.2 Uncertainties

The data subsystem was one of the biggest challenges we faced because we were unable to connect the microcontroller with the wireless transmitter. The reason for this was the incompatible power connection between the microcontroller and transmitter. While the ATMEGA328 microcontroller outputs 5V on its serial UART connection, the transmitter requires a maximum of 3.3V. While a higher resistance dropped the voltage to an appropriate value, the current then became too low for the transmitter to function.

Additionally, we were unable to properly calibrate the conductivity sensor due to lack of an external conductivity probe to verify our sensor readings against.

5.3 Future Work/Alternatives

Although our project was a big success, there are some things that we would like to fix and improve. One of the things we would like to fix is the data transfer between the microcontroller and the wireless transmitter. Because of this issue, we were not able to establish a connection between the device and the mobile and web application, however, the device and the mobile and web application worked individually as intended. If we were to fix this issue, there would be no other problems with our project.

To further help our device and the user, we would create a smaller product. We intentionally left extra space in our PCB enclosure when designing our prototype to account for any errors or extra components that we may have needed. Reducing this extra space would reduce costs of the device and also provide a more compact and less intrusive product to the fishtank. We can also add additional stickers or icons under the LEDs to indicate which LEDs represent which sensor. This would help the user easily detect which sensor’s values are out of range without having the user needing to look at their mobile or web application.

5.4 Ethics and Safety

This project follows proper ethical and safety protocol by creating a device that proves to be non-toxic and harmless to both humans and animals. Section I.1 of the IEEE Code of Ethics [3] states “to hold paramount the safety, health, and welfare of the public” (IEEE p.1 I.1). This means that in our practices, we must uphold the necessary requirements to create a safe product for anyone to use. Since our sensors are suspended in water, it should not leak any material into the water that could be harmful to the fish inside. All circuitry components should have an airtight design to ensure no water damage can occur to the device. No wire should be exposed underwater to ensure that a current cannot enter the aquarium. In the event where a component malfunctions, the sensors are not able to change any values, such as temperature, pH level, and conductivity, in the aquarium itself. If someone were to misuse the product, they would only be able to damage the sensors or circuitry of our device. This is not able to affect the water and did not harm any of the fish inside.

To ensure proper safety when creating our device, we must follow the proper guidelines for electrical, mechanical, and lab safety. We should never work alone in the lab, bring food or drinks, and should always clean up after ourselves and clear our lab stations. We must report any broken equipment and properly dispose of any materials not needed. When building our device, potential hazards and things that we must be careful of are glass, soldering equipment, electrical components, and batteries. The user only needs to make sure that no components are broken when placing the device into their aquarium. If something were to be broken, the possible dangers are exposed electrical wires, broken glass, or sharp plastic.

References

[1] DFRobot, “SEN0161 SEN0169 DFRobot Datasheet”, SEN0169 datasheet, 2017. https://www.application-datasheet.com/pdf/dfrobot/sen0161.pdf (accessed Feb. 9, 2022).

[2] T. Scherer and M. Meehan, “Using Electrical Conductivity and Total Dissolved Solids Meters to Field Test Water Quality”, North Dakota State University, July 2019, [Online]. https://www.ag.ndsu.edu/publications/environment-natural-resources/using-electrical-conductivity-and-total-dissolved-solids-meters-to-field-test-water-quality. (accessed Feb. 10, 2022).

[3] IEEE. “IEEE Code of Ethics.” IEEE Code of Policies, Section 7 - Professional Activities (Part A - IEEE Policies). June 2020. https://www.ieee.org/about/corporate/governance/p7-8.html. (accessed Feb. 10, 2022).

[4] Illini Success. “Annual Report 2019-2020”, https://uofi.app.box.com/s/1t8xj69117lrsqm7753ujnrg8yyrtcwn (accessed Feb. 24, 2022).

ECE 445 Lab. ECE 445 - Senior Design Laboratory, https://courses.engr.illinois.edu/ece445/. (accessed Feb. 10, 2022).

Espressif Systems, “Espressif Smart Connectivity Platform: ESP8266”, ESP8266 datasheet, Aug. 03, 2017, [Online]. https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf (accessed Feb. 21, 2022).

Appendix A: Requirement and Verification Tables

Table 4: Sensors requirements and verifications

Requirements	Verification
1. Temperature sensor must constantly measure temperatures in the range 23°C to 27°C (±2°C) at an operating voltage of 5.0V (±0.5V) 2. pH sensor must measure pH in the range of 6.8 to 7.8 (±0.5) at an operating voltage 5.0V (±0.5V) in temperatures ranging from 23°C to 27°C (±2°C) 3. TDS sensor must measure conductivity in the range 70ppm - 120ppm (±20ppm) at an operating voltage of 5.0V (±0.5V) in temperatures ranging from 23°C to 27°C (±2°C) 4. Sensors must have a query time of < 1000ms (±250ms)	Verifying Requirement 1: ● Place additional thermometer in water ● Use multimeter to ensure sensor is receiving 5.0V (±0.5V) ● Place sensor in water and compare results shown in software to that of the additional thermometer ● Ensure the measurements match with a tolerance of ±2°C Verifying Requirement 2: ● Ensure water temperature falls within threshold ● Use multimeter to ensure pH sensor is receiving 5.0V (±0.5V) ● Measure the water with a handheld pH meter and ensure that the value it produces is the same as that of the sensor in software (±0.5) Verifying Requirement 3: ● Ensure water temperature falls within threshold ● Use multimeter to ensure TDS sensor is receiving 5.0V (±0.5V) ● Measure water with handheld TDS meter and ensure the value it produces is the same as that of the sensor in software (±100pm) Verifying Requirement 4: ● For each sensor in the water, log the time it takes the microcontroller to receive a temperature reading using timing libraries in Arduino IDE ● Ensure query times fall within stated thresholds

Table 5: Power requirements and verifications

Requirements	Verification
1. AC/DC converter must convert 120VAC (±100VAC) from a wall outlet into 12VDC (±0.5VDC) 2. LM1117-3.3 must output 3.3V (±0.2V) and 200mA (±50mA) with an input voltage of of 12V (±0.5V) and input current of 500 mA (±50mA) 3. LM7805 must output 5V (±0.2V) and 300mA (±50mA) with an input voltage of 12V (±0.5V) and input current of 500 mA(±50mA)	Verifying Requirement 1: ● Plug AC/DC converter into a wall outlet and plug the barrel jack into a female port linked to a breadboard ● Use a multimeter to measure the output voltage and ensure it reads within the threshold listed Verifying Requirement 2: ● Feed 12V into the LM1117-3.3 using an external power supply at 500mA of current ● Using a multimeter, ensure the output of the LM1117-3.3 is within the threshold listed Verifying Requirement 3: ● Feed 12V into the LM7805 using an external power supply at 500mA of current ● Using a multimeter, ensure the output of the LM7805 is within the threshold listed

Table 6: Data requirements and verifications

Requirements	Verification
1. Wireless transmitter must be able to interface between the microcontroller and mobile app and send updated data at a rate of at most 30 seconds per cycle 2. Data subsystem must be able to communicate with the mobile subsystem via WiFi 3. LED must light up when data measured is outside of acceptable range	Verifying Requirement 1: ● Send a test message (001 in binary) from microcontroller to the wireless transmitter, and verify it displays on the mobile app Verifying Requirement 2: ● Ensure that both the Auto Aquarium and mobile app are on the same WiFi ● Display log statements in Arduino IDE console from the microcontroller code that shows the data to be sent via the wireless transmitter Verifying Requirement 3:

	- Use the mobile app to set a range that is smaller than the current temperature measured (ex. 0-10℃ for room temperature water) - LED must light up within 30 seconds of this change

Table 7: Mobile app requirements and verifications

<u>Requirements</u>	<u>Verification</u>
1. The mobile app must update the displayed information within 30 seconds after it receives it from the wireless transmitter 2. The mobile app should allow the user to adjust the ranges of tolerance of variables	Verifying Requirement 1: - Execute microcontroller code function that sends test data for temperature to 999 - Check the visualized data ranges in the mobile app and see if the displayed data has been updated Verifying Requirement 2: - Adjust temperature range on the mobile app to 0-10℃, when actual aquarium water is room temperature - Verify that the mobile app sends dangerous condition notification

Appendix B: Initial PCB Design

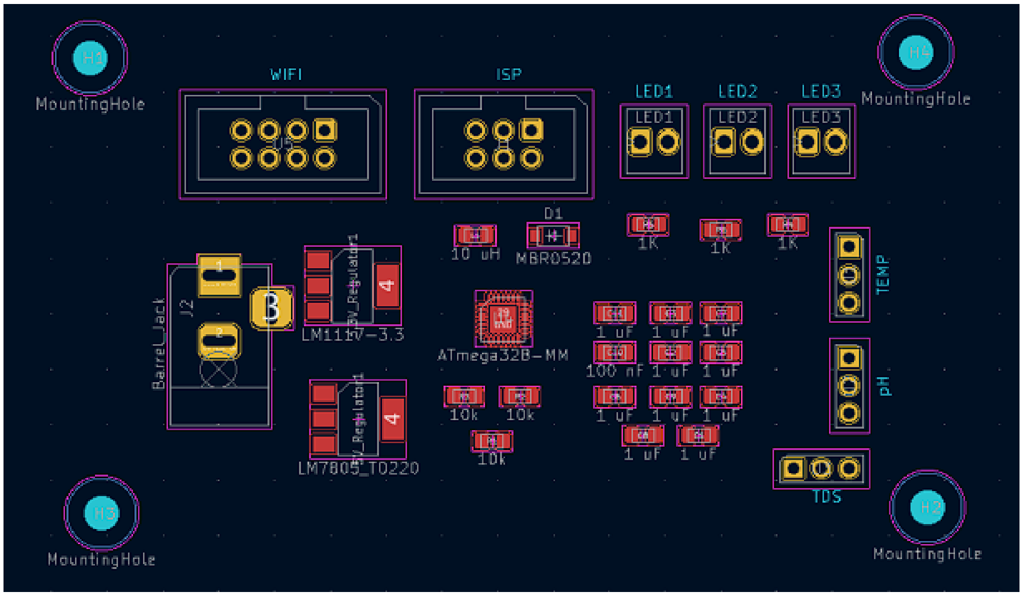

Figure 11: PCB Diagram

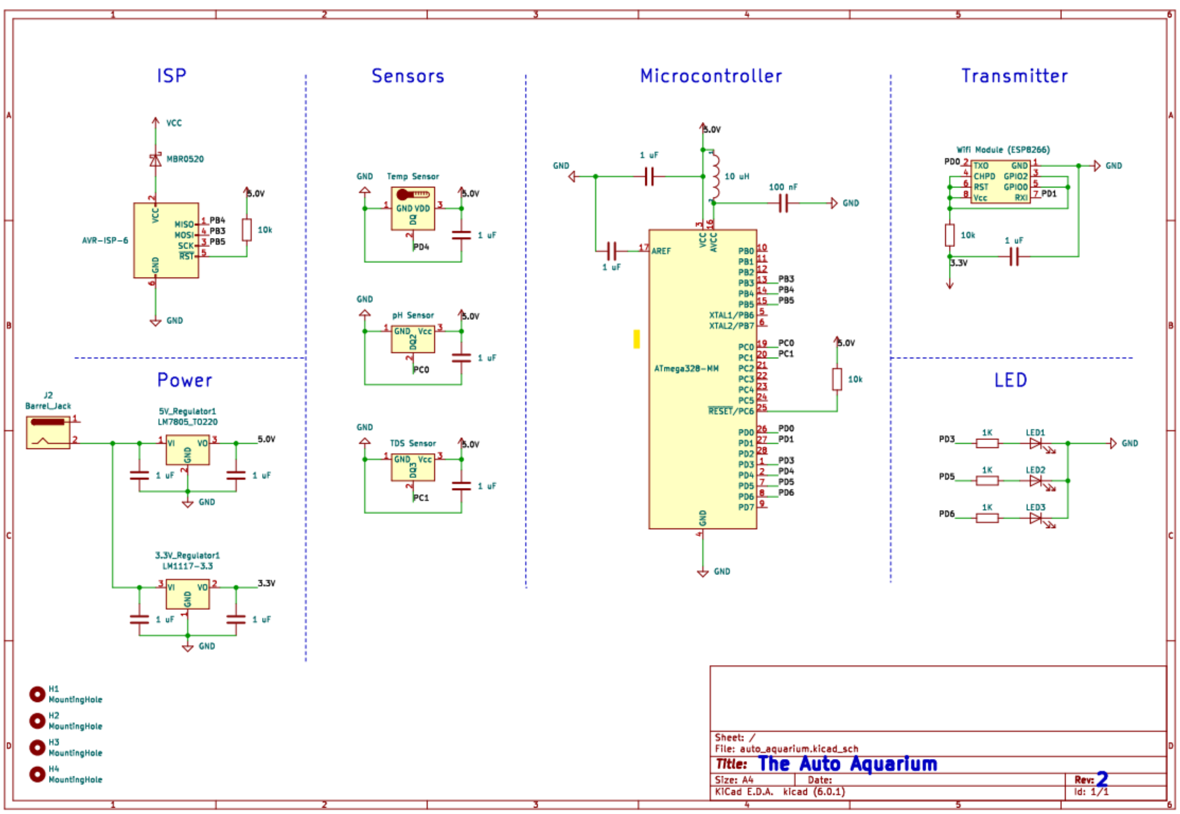

Figure 12: PCB Schematics