Final Project Report

Underwater Algae Bloom Detection
sddec20-23
Client: Santosh Pandey
Faculty Advisor: Santosh Pandey

Team:

Anastasia Golter -Housing Team
Nicholas Stasi - Sensor Team
Emily Kinne - Sensor Team
Zachary DeMaris - Housing Team
Jack Seiter - Communication Team
Andrew Koenen - Sensor Team

Executive Summary

Development Standards & Practices Used

- Software version control (Git)
- Printed circuit board design
- Weekly virtual standup meetings
- I2C Protocol

Summary of Requirements

- Device must observe temperature, movement, and light levels
- Device must be capable of transmitting and storing underwater sensor data
- Device must be waterproof
- Device must have low power usage
- Device must be able to operate for at least 24 hours
- The total cost of one sensor platform should be under $500
- At least 3 sensor arrays per depth to get an accurate set of data

Applicable Courses from Iowa State University Curriculum

- EE 201 - Electric Circuits
- EE 230 - Electronic Circuits and Systems
- EE 333 - Electronic Systems Design
- CPRE 288/388 - Embedded Systems I/II
- COMS 309 - Software Development Practices
- CPRE 489 - Computer Networking and Data Communications

New Skills/Knowledge acquired that was not taught in courses

- Printed Circuit Board Design
- I2C Communication Protocol
- Soldering

Table of Contents

1. Introduction

1.1 Acknowledgement

We would like to thank the client for immensely generous financial support and the students Christopher Legner and Vishal Patel for their continued technical support throughout this project.

1.2 Problem and Project Statement

Algal blooms release harmful toxins into the water which can negatively affect public health, ecosystems, and fisheries. The main concerns are that the toxins released can get into the food we consume, our drinking water, and damage the balance of resources in aquatic ecosystems. For example, many organisms suffer due to the decrease of oxygen used up by the algae. By monitoring the factors that cause algal blooms, scientists can better prepare and take action to lessen the harmful effects [1].

Our approach to solve this problem is to create a device, called a sensor platform, to collect various types of underwater data. We will use a variety of sensors and eventually analyze and interpret the data to find the conditions that cause an algae bloom.

The purpose of this project is to be able to predict when an algal bloom will occur and use this data to prepare and mitigate or neutralize some of the harmful effects. This data would be beneficial for biologists, environmentalists, and the people who depend on the affected bodies of water for basic resources. We aim to create a device that can record all of this data and be able to predict algae blooms from conditions in the environment.

1.3 Operational Environment

Our sensor platform will operate in an aquatic environment. Because the device will be placed outdoors, and part of it will be submerged underwater at all times, it will need to be completely waterproof. There are also many outside factors like the weather and organisms in the environment that could disrupt the accuracy of our data or damage the platform.

1.4 Requirements

Functional Requirements:

- Device shall observe environmental factors such as temperature, movement, and chemical composition of water
- Device shall be capable of transmitting and storing sensor data
- Device shall be capable of floating on the surface of the water
- Device shall be waterproof
- Device shall have a low power usage
- Device must be able to operate for at least 24 hours
- The total cost of one sensor platform should be under $500
- Device shall use 3 sensor arrays for any desired measurement at a specific depth
- The sensor platform shall be capable of performing readings at a maximum of 3 feet below the waterline

1.4.1 Engineering Constraints and Non-Functional Requirements

Because our device operates underwater we have to handle the constraint that wireless communication through water is different than through the air. This limits the options for transferring data below the water. Secondly, communication from water to air provides complexities in itself that are still being solved on research and industry levels. This provides another barrier that we must workaround to complete our project. Lastly, we have a device in a remote area of water. This limits our communication methods in the air and requires us to think of long-range communication or a way to store the data for retrieval at a later point.

Beyond communication, we also have the issue of collecting data from a hazardous environment. We must protect the sensors enough to keep them from being destroyed, while maintaining their integrity and ability to record accurate information from the surrounding environment.

1.5 Intended Users and Uses

Biologists: Can use the data to prepare and mitigate the harmful effects of an algae bloom

Environmentalists: Can use the data to decrease the chances of algae blooms and protect the plants and animals that are negatively affected by the toxins

Dependent people: Can use the data to determine when the water is unsafe to drink or when aquatic organisms may be contaminated

1.6 Assumptions and Limitations

Assumptions:

- The device need not be resistant to wildlife interference

Limitations:

- Power is limited to battery capacity
- Wireless communication is limited underwater
- All sensors must be waterproofed

When designing a waterproof system it comes with many rules and extra necessities. Looking at different means of transmitting data, we found that Bluetooth has a very limited range underwater. Because of this, we decided to look into cellular as well as other means of communication. Along with this, we assume that the devices will not need to withstand wildlife trying to break the device. Some limitations of the design include the power consumption of the device. The device needs to function on its own and efficiently manage its power in order to read data for as long as possible. In addition, we must focus on sensors that are capable of being submerged into water or have a way to transmit their data through a medium that touches the water. Water severely weakens wireless communication traveling through it, so we are limited by how we transport data from the sensors to the user.

1.7 Expected End Product and Deliverables

The goal of this project is to create a floatation device that both records and transmits data for a user. This device must provide a floating surface to communicate above the water. Next, there will be a flexible array of sensor devices to collect data about the water at different depths. To ensure the integrity of the data a single platform must support at least three of these sensor arrays in order to provide statistical significance. There should also be a system set up to transfer the data back to the user. The second phase of this project which we are not responsible for would be to create a software program to analyze the data retrieved to help predict algae blooms. It would most likely use graphs and machine-learning algorithms to analyze and interpret the collected data.

2. Specifications and Analysis

2.1 Proposed Approach

The team began by developing a better understanding of the end goal: to gather data on algae blooms. With this goal in mind, we researched and found sensors that could be used underwater. Some of the sensors we will be using include a temperature sensor, accelerometer, and light sensors. We will begin by getting these sensors to work and send data. After a few sensors are working, we will add more and continue testing. Concurrently, we are working with our data transfer module to find ways to transmit our gathered sensor data.

2.2 Design Analysis

There are a few challenging design considerations we encountered thus far. The most challenging has been figuring out how our sensor platform will communicate with a hub on land. The sensor platform would be floating in a lake with potential miles of water on each side of it. Using a Bluetooth module was the first communication idea proposed. Bluetooth has severe limitations of the distance the signal can travel, which doesn't make it feasible for our project. We have transitioned to try and use cellular modules for communication. Cellular data could potentially provide us with internet access almost anywhere in the world. A challenge we have faced with cellular is the lack of modern cellular modules available for the Raspberry Pi which will be used in our head module.

The brain of our sensor platform is the Raspberry Pi Zero. The Raspberry Pi Zero provides an easy to use interface and also supports many useful libraries for communication protocols such as one wire and I2C. Sensors are wired to the Raspberry Pi, which is responsible for reading the sensor data and transmitting it back to a hub for processing. However, the Raspberry Pi has a significantly larger power draw compared to other platforms such as Arduino.

For the structural design, we plan to use a flotation device at the water line and acrylic like tubing that goes underneath the water. The tubing would contain the various sensors and wiring in a waterproof casing. Some of the sensors need to be in contact with the water. For those sensors, we plan to cut holes in the tubing so the sensors can make contact with the water and then seal the holes around that contact point.

2.3 Development Process

The team has created its own development process that is similar to Agile. Each weekly meeting mirrors a scrum meeting and communication among the team is frequent. Individuals share the progress they've made, as well as upcoming goals and any potential obstacles. Work is done in two different ways: self-guided work which may consist of multiple team members doing development on the project, and guided work where team members walk others through their development processes, such as a demo, in order to keep members up to date. These meetings keep members up to date with information and skills. The team has used this combined with frequent communication to create an effective development process.

2.4 Conceptual Sketch

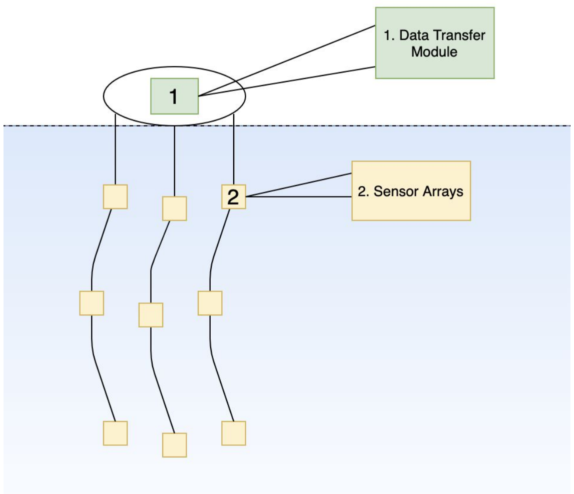


Figure 1: Top-level design of sensor platform.

Description

1. Float to keep the data transfer module outside of the water. This device will be used to store and transfer data from the sensor arrays
2. The sensor array contains components such as a temperature sensor and accelerometer to collect data from the water
3. The sensors are placed at different depths to create a more robust and accurate set of data (at least 3 sensors to maintain statistical integrity)

3. Statement of Work

3.1 Previous Work and Literature

Currently, there is not much data on algae blooms, and the cause or environmental conditions in which they occur is unknown. Previously, drones have been used to collect data such as the concentration of green color in the water, however, this is not a reliable way to monitor conditions or predict algae bloom because it usually means one has already occurred.

NOAA has various articles that discuss algae blooms and the devastating effects that they can cause to the environment and the economy. It is estimated that algae blooms cause $82 million in economic losses every year. They discuss that they understand factors that contribute to algae blooms, but that they do not understand how they all come together to form algae blooms. Research shows that algae blooms seem to flourish in certain temperature and water currents, which helps reinforce the use of temperature and accelerometer sensors on our sensor arrays [3].

There are already devices available commercially that can help detect if an algae bloom is present. The primary way of determining if an algae bloom is present is by measuring the chlorophyll content in the water. This is helpful to to alert real time if an algae bloom is present. However, our project aims to help better understand what individual factors contribute to the formation of these blooms. The goal is to be able to spot the conditions that will cause a bloom before an algae bloom has formed [2].

A large part of our project consists of developing a system that can monitor water conditions and report that data back out to be analyzed. One product that we researched as a potential alternative for monitoring water conditions is the EXO Platform developed by YSI. It is a modular sensor system that you can plug a variety of different sensors into. They provide sensors for things such as temperature, nitrate levels, pH, light, and many more. It is a fairly expensive system so we did not have the capability to test it as a part of our project. It could be a valuable tool for future research in this area [4].

3.2 Technology Considerations

All of our data is going to be collected underwater. An important consideration would be how to protect our technology and ensure our design is completely waterproof. This is especially important for any exposed wires or sensors which will be in direct contact with the water. They have a higher risk of shorting the circuits and are more susceptible to wear down faster than other components.

3.3 Task Decomposition

Tasks are decomposed into individual units. Each task is labeled with a topic and broken down into smaller more specific subtopics in order to understand the task's hierarchy within our project. An example of this type of label is "platform - sensing - heat" for a task relating to the heat sensors on our sensor array.

The tasks will be divided into two main parts consisting of "Firmware and physical components" and "software and communication". The first piece will deal with all aspects of choosing sensors, getting the sensors hooked up and working, creating PCBs, and designing a waterproof floatation device along with wired connections between the devices. The second piece will focus on taking that data and combining it to be transmitted or received. After the data is back on land it will be analyzed in order to collect data about the algae blooms.

Tasks:

- Firmware and physical components
    - Determine sensors to use
    - Create PCB
        - Design PCB
        - Solder PCB
        - Test PCB
        - Waterproof PCB
    - Setup I2C network
        - Find addresses for all sensors
        - Determine how to combines sensors with the same addresses
            - Integrate sensors into a single I2C network
    - Create code to read from different sensors
        - Read each sensor individually
    - Create code to read from all sensors in an addressed manner
- Software and communication
    - Create software for pi to communicate with cellular network
        - Talk to fona chip on the board
        - Connect to a cellular network
        - Get a cellular account to test this
    - Create software for the computer to receive communication from the cellular network
    - Create a form of data persistence to store the information obtained
        - Decide between file storage and database storage
    - Test cellular communication
- Housing Platform
    - Create float
        - Cut and waterproof wood
        - Seal PVC piping
    - Create waterproof container for Raspberry Pi
    - Waterproof sensor legs
        - Tubing for I2C wiring
        - Waterproof PCBs

3.4 Possible Risks and Risk Management

Some possible risks are damages to the boards and sensors once we attempt to waterproof the device. To mitigate this we will be researching and testing a few different types of enclosures/sealants to make sure we find the right one. Additionally, we risk losing data due to the inability to transmit. We aim to have a high amount of static memory on the sensor platform device.

3.5 Project Proposed Milestones and Evaluation Criteria

- Milestone 1: Determine all sensors needed for the system
- Milestone 2: Create a working sensor board to retrieve data from water in a simple fish tank

- Milestone 3: Create a data analysis tool to make sense of the data
- Milestone 4: Enable communication ability or data storage in a remote environment
- Milestone 5: Create communication between units
- Milestone 6: Waterproof design
- Milestone 7: Full-Scale deployment of multiple units in collaboration

3.6 Project Tracking Procedures

We will use GitLab issues and milestones to create and track individual tasks. It will mark the progress of the tasks so that the overall project process is easy to view. Lastly, we will do regular reviews on current tasks and upcoming tasks.

3.7 Expected Results and Validation

The desired outcome of this project is to create a sensor platform capable of collecting data from multiple depths in the water and use this data to predict aquatic life events, specifically algae blooms. The secondary goal would be to create a network of these devices to communicate across a large lake. To validate the unit we will test it in both lab environments and real-world bodies of water.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

- February: Begin testing sensors
- March: Solder parts to prototype PCBs and begin testing with PCBs
- April: Create new PCBs and continue testing new sensors, work on communication between modules
- August: Have a testable environment and prototype producing data
- September: Final Round of PCBs ordered and assembled, work on waterproofing
- October: Finalize method of communication to home base
- November: Final touches, clean up the design

4.2 Feasibility Assessment

The project will consist of an underwater sensor platform connected to a wireless communication device or onboard storage. This project will undergo many challenges due to both the water-based aspect of the design as well as the communication of data in rural areas. To determine if this project was feasible, we had to determine three main things. First, are there currently sensors available for what we want to measure? With some background from the client, we determined this was realistic. Second, is there a way to safely collect the information without risk to the electronics needed? Through research, we found multiple ways to waterproof the system and make sure we could still acquire the data necessary. Lastly, what type of communication can we use to transfer the data and analyze it? After looking at multiple different methods such as cellular, physical access via a USB, and short-range Wi-Fi communication, we concluded that no matter what we decide, there is a reliable way to transfer the data back making it viable. Since all three parts of the project are feasible the team then decided that the overall project was feasible.

4.3 Personnel Effort Requirements

- This project will require learning assistance from Christopher Legner and Vishal Patel, both students working with Dr. Pandey, for knowledge of pre-existing software, surface mount soldering, and PCB design.
- This project may require help from local wildlife authorities to gain access to a live environment for large scale testing.
- Below is a table breaking down the project into tasks:

Task	Description	Estimated time (Hours)
Selection of sensors	Research and select the sensors necessary to collect the data we need	15
Selection of microcontroller	Research and select the microcontroller(s) needed to fulfill all requirements	15
Designing PCBs	Schematic and board design for prototyping sensors and controllers.	25
Assembling PCBs	Solder parts onto printed circuit boards	15
Testing PCBs	Test printed circuit boards so ensure sensors are working	60
Design prototype housing for data collecting module	Design the first iteration of our waterproof housing for our microcontroller and all peripherals.	120
Test housing structure	Test our housing is waterproof and allows all sensors to collect accurate underwater data.	30
Design Cellular/Communication framework	Decide on a cellular module that works and set up communication between the module and microcontroller.	100
PCB prototype re-design (if necessary)	Redesign a better prototype board following the results from PCB testing	25
Design final housing structure for boards and sensors	Redesign our housing structure from our first prototype to a more permanent solution	20
Build final housing structure based on design	Construct a final housing structure based on the redesign.	40
Documentation	Write up instructions for how to use the device and collect data	15

4.4 Other Resource Requirements

Beyond the financial requirements, we will need a workshop to solder the components together, an environment to test the product in, and help with knowledge of the algae bloom situation, as well as some biological samples to acquire data.

4.5 Financial Requirements

While the total financial requirements to complete the project are unknown at this point due to speculative prototyping costs the goal for the final unit will be no more than $500 in total for a single unit. If the team chooses to use a cellular network for communication of data, there may be a low monthly cost associated with the unit.

5. Testing and Implementation

5.1 Interface Specifications

- For our sensor interface, I2C was chosen as an interface with the sensors because it allows flexibility and compatibility with most of our sensors.
- The controller to communication module interface will occur in a tree topology over the UART Data Link Layer implementation. When networking is required a binary application protocol ontop of TCP/IP which uses go-back-n as a scheme for sending sensor frames treats each sensor report as a frame with ordering based on the time the sensor data was read and the ordering of the sensors.

5.2 Hardware and Software

Our reporting system, ReportReceiver, is the system by which data is collected on the server and is transmitted to the remote server. ReportReceiver's testing is in JUnit. JUnit is a framework for writing and running unit tests.

The hardware to be tested is the head unit containing our controller and the accompanying legs that include our custom sensor arrays. The head unit will be a single board computer, in this case a pi, since this keeps power low while still providing a wealth of interfaces.

5.3 Functional Testing

Functional testing includes our unit tests, integration tests, and load tests. Our unit testing is done through JUnit and is run during and after development. Much of the complex code has dedicated unit tests covering it. Demonstrations to the client of the networking have already taken place with encouraging results. The major final test of our device will be a deployment when we put the device into a lake and attempt to receive data for an extended period of time. Tests have shown that the device is mostly waterproof, the sensors integrate correctly, and that the reporting software properly transmits data.

5.4 Non-Functional Testing

Non-functional testing will include things such as checking the energy efficiency of the device, passing the 24-hour functional requirement, testing if sensors can be replaced easily, and testing the speed of data transmission. Testing the efficiency will be simple as we can change the sampling rate and see what kind of battery life we have. Testing for replaceability of sensors will require us to look into our design to make sure the device is easily maintainable. We found that the actual sensors were easily replaceable with the most difficult part being replacing the shielding after the sensor would be replaced. Lastly, we checked how fast the cellular network transmits the data to determine the real-time of our information, cellular communication had a couple seconds lag, which is acceptable for TCP and our initial usage of TCP/IP is helpful for this.

5.5 Process

For the data transfer tool, testing occurs through JUnit tests. The hardware we create is run through separate testing that consists of flashing the device with the software and testing the results in a controlled environment. If the results remain consistent with regular sensors (such as a thermometer) that are not attached to the board then we deem the device good. For waterproof testing, we submerged the device into a lake and made sure it stayed functional. The unit maintained functionality for a short period of time before part of it failed and needed to be replaced.

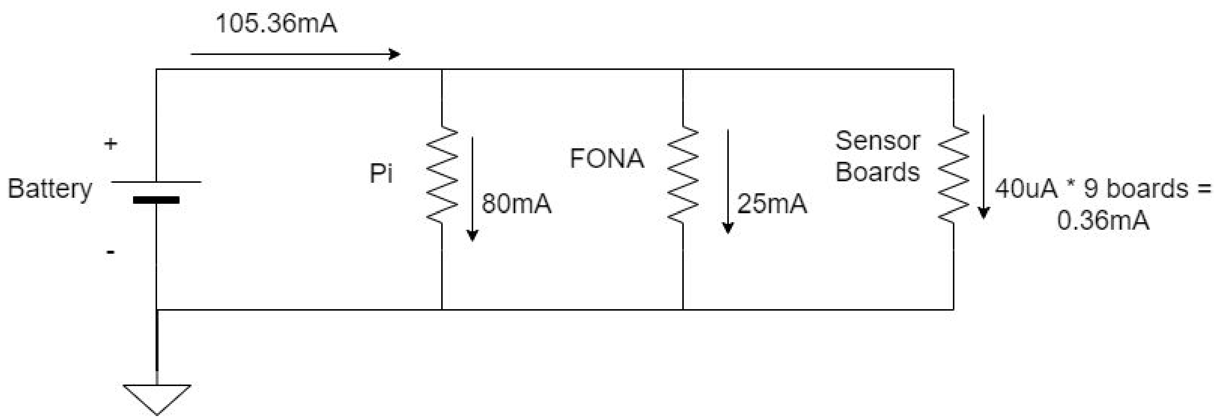
5.6 Results

General Results

For the data transfer tool, all tests are passing, with all important code covered in tests. All code required for the project has been written and tested. For the hardware, we have an entire leg constructed and have submerged it in water. Unfortunately, the seal was weak and water slipped in. However, the pi itself and all other devices were safe and did not short. This is great for safety as even if one leg is destroyed the others are still operable. However we did get data from the leg for a short time before the waterproofing was punctured, so we know how the system will perform when waterproofing doesn't fail.

Data

Power consumption calculations:



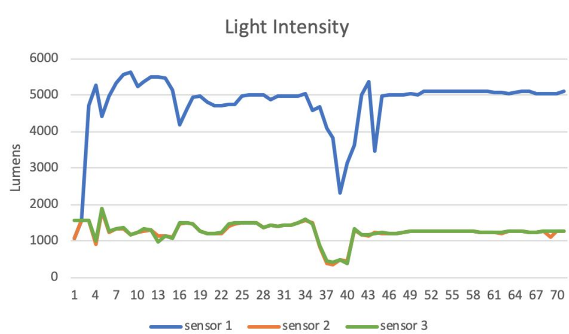
The numbers shown in the figure above are what we measured for each of our components that are running in parallel off of the battery. These numbers are typical averages and could change if we change how software is accessing the hardware resources.

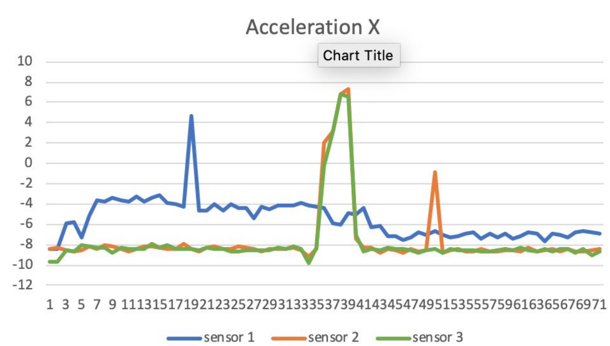
We are using two 3.7 6000mAh batteries wired in series to give us 7.4V. We then use a voltage regulator to drop the 7.4V down to 5V. Wiring the batteries in series helps us attain the higher voltage required by the pi, but it does not increase our capacity so our total battery capacity is 6000mAh. The following calculation shows how long our system will last under normal operation.

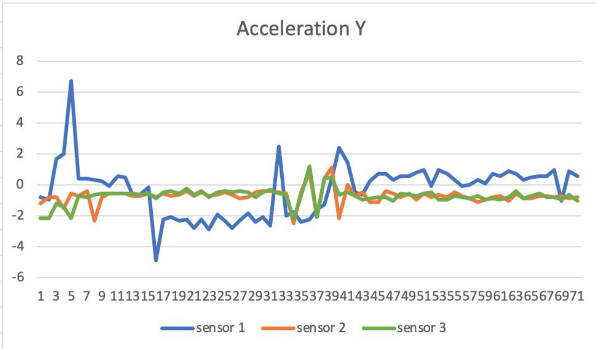
$$\frac{6000mAh}{110mA} = 54.5\ hours$$

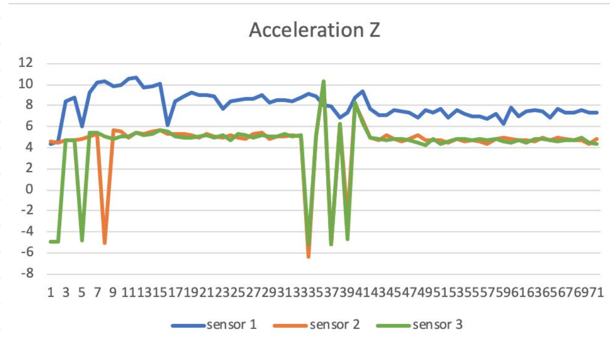
As you can see by our estimations, our system will last more than double the 24 hour requirement.

For the sensors we connected one entire leg and leg it run for a while varying the light level and angle throughout to show what a standard simulation may look like and we get the results below:









As you can see the data is relatively steady. The light sensors show how bright the environment is and have small dips as it gets darker. The three acceleration sensors provide the orientation of the board which would be used to show water speed as they change. The large spike would reference a strong gust or movement. The main thing we obtain from these results is that the sensors are stable and can record data over an extended period of time sampling at regular intervals. This trial was approximately 20 minutes worth of data at an increased sampling rate. As the final test after each individual stage of testing mentioned above this shows the entire device working together. This means without waterproofing the project is a success.

6. Closing Remarks

6.1 Conclusion

From the beginning of our project, we have completed a fully functional prototype of our *Underwater Algae Sensor*. We accomplished creating a PCB for our sensor boards that included three sensors: accelerometer, temperature, and light sensor. The PCBs were manufactured and printed, soldered and we wrote a script to collect data from up to 16 boards. We used I2C communication and a 16:1 multiplexor to address each board separately. In addition, we finished our head module that is responsible for transferring our data via 2G cellular communication and built waterproof housing for our components to function in the water.

With the limits on power and communication due to the device's environment, we had limited options, but made the best design choices we could based on our research. The cellular choice was made as it would create a simpler implementation and require the least amount of added hardware. Dial up is done by running pppd on startup using appropriate settings for the module which will establish a connection with the modum like a normal phone and then pppd establishes a public ip.

We created a perfboard to combine the hardware in our head module. This included the Raspberry Pi, FONA board, and 16:1 multiplexor. This was an additional design decision that helped with the organization of wiring and made our hardware look more professional. We also decided on using two 4V li-poly rechargeable batteries to power the Pi and sensors.

We worked on the hardware and software simultaneously and brought everything together in the last few months, including waterproofing and field testing the final design after all components were working individually. Testing our design in water had some disappointing, but not unexpected results. We knew the risks of using sensors underwater and due to some waterproofing leaks, our design got wet and stopped working. The Raspberry Pi and all components of the head module were unaffected however the sensor boards failed due to water exposure .

We would have liked to add more sensors to our PCBs to detect phosphorus or nitrogen, however due to a shortened semester and delays in production and shipping we did not add any more. This could easily be implemented in future designs. With help from faculty in the area of biological research (Dr. Pandey and Christopher Legner), and with our combined knowledge and research we were able to successfully create a possible solution for the remote analysis and collection of data on algae blooms.

6.2 References

[1] Andersen, Enevoldsen, and Anderson (2003). Harmful algae monitoring programme and action plan design. *Manual on Harmful Marine Microalgae*, *33*, 627-645.

[2] Laser Diagnostics Instruments. “Algal: Bloom Sensor.” LDI. https://ldi.ee/products/algal-bloom-detector/#:~:text=The%20Algal%20Bloom%20Sensor%20is,when%20a%20threshold%20is%20reached (accessed November 11, 2020).

[3] NOAA. “Why do harmful algal blooms occur?.” NOAA. https://oceanservice.noaa.gov/facts/why_habs.html#:~:text=Harmful%20algal%20blooms%20(HABs)%20occur,%2C%20marine%20mammals%2C%20and%20birds.&text=Studies%20indicate%20that%20many%20algal,and%20water%20currents%20are%20favorable (accessed November 11, 2020).

[4] YSI. “The EXO Platform.” YSI. https://www.ysi.com/exo (accessed November 11, 2020).

6.3 Appendix

Appendix I: Operations Manual

Sensor Operations:

Prerequisites:

1. Obtain a raspberry pi
2. Install raspbian: https://www.raspberrypi.org/documentation/installation/installing-images/

Installation:

1. Configure I2C on the pi: https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c
2. Make sure python3 is installed and is the default on the pi: “python --version”
    a. If it is not set to python3 and install git and pip
        i. sudo apt-get install -y python3 git python3-pip
        ii. sudo update-alternatives --install /usr/bin/python python $(which python2) 1
        iii. sudo update-alternatives --install /usr/bin/python python $(which python3) 2
        iv. sudo update-alternatives --config python
3. Setup the environment
    a. Get the GPIO library
        i. sudo pip3 install RPI.GPIO
    b. Get the adafruit libraries(board, digitalio, etc)
        i. sudo pip3 install adafruit-blinka

c. Install the libraries for the sensors we use:
    i. Light
        1. sudo pip3 install adafruit-circuitpython-veml7700
    ii. Temperature
        1. sudo pip3 install adafruit-circuitpython-mlx90614
    iii. Acceleration
        1. sudo pip3 install adafruit-circuitpython-adxl34x
4. Grab the code from the repository
5. The device is now setup

Hardware Setup:

1. Connect the pi to the piHat that we created then slot in the analog mux and the cellular card
    a. Make sure the Fona chip has a sim card in it and it is activated
2. Next to the red multiplexer are the data lines. Connect these to the legs in order. The 0-2 output should go to leg 1, 3-5 goes to leg two, and 6-8 goes to leg three
    a. Within the leg the sensor should be ordered top to bottom to keep data consistency easy
3. Once this is in place move on to the next section on software

Usage:

The sensor code is called "sensor.py". As shown above to make sure the installation is fully completed before running. The usage of the code is:

sensor.py <leg number> <sensor number>

The sensor numbers start at 0 for both. The sensor number increments along the leg. This means that 0-2 is the first board (highest) in a leg and 3-5 is the second board and 6-8 is the last or lowest board. Getting 0 or 0.0 means that the sensor is not connected properly. This return value allows us to avoid errors while still giving a distinct way to tell if something is wrong. This code is called by the networking code in the next section, but can be used with or without it.

The sensors return three different types of data. The light sensor returns an int. The accelerometer returns three signed doubles in the form "x.xxxxxx x.xxxxxx x.xxxxxx" in the order of x y and z. The temperature returns a float in the form "xx.xxxxxxxxxxxxxxx" and is in celsius. This specific temperature sensor uses 15 points of precision.

Cellular Operations:

To run the ReportRecever on the platform run the compiled jar with the argument "client". This will immediately run the client and start collecting data on regular intervals and attempt to report it to the server. The server specification is in the code and will need to be recompiled if the ip changes. If the platform configuration needs to change the in code configuration for the platform's sensors must be updated as well. To run this on startup a systemd unit or timer may be used.

To run the ReportRecever run the compiled jar with the argument “server”. This will immediately start collecting any data sent to the server. Running the jar with the argument “export” will export all saved data in a plain text format. An export may be run while the server is running.

Appendix II: Alternative/Initial Designs

When we were first introduced to this project our client had been doing research on sensor boards with a Bluetooth chip, so we had considered using Bluetooth communication. However, after doing some research we found that Bluetooth does not perform well underwater and has a very short range, so getting data from the middle of the lake to on land would not be feasible.

We considered using an Arduino for the head communication device as well 1-wire interface for the sensors that similar to I2C share the same bus. However I2C offered higher data rates and more control when trying to communicate with more sensors. We had some issues interfacing with the FONA board from the Arduino so we switched to the Pi to make the high-level communication programming easier.

Some other versions of our project that resulted in failure were trying to control the sensor boards via power. We pursued this idea because our sensors are not addressable therefore we cannot differentiate between which sensor board is sending data. We tried to use individual GPIO pins on the Raspberry Pi as the power to turn on an individual sensor board when we wanted to collect data. This was done in software by our python script. However, this design failed due to a power leak on our PCBs. When we turned the GPIO pins off there was a small amount of power running through the boards and therefore our software would get confused by which I2C device was connected.

Lastly, for different waterproofing ideas we thought of using PVC pipe underwater but some sensors like the temperature and light needed to see the water to get accurate data. We had also thought of putting each sensor board in a watertight plastic box. All of these waterproofing designs including our current prototype have issues with the sensor array “legs” being too buoyant. Because our tubing is filled with air we need something to weigh down the legs to get data underwater.

Appendix III: Other considerations

Environmental:

One important aspect of this project is that it is working to protect the environment. That being said the materials used for this project are important and while creating a prototype to validate the concept is important you need to eventually look more into the materials. How can the pipes for the legs be made to not introduce micro plastics into the water? Does the metal used on the base have a lot of issues with rust and corrosion? Perhaps some less oxidizing metals could be used.

Sensing:

There are a ton of full system units that try to do in situ (on site) analysis especially when it comes to chemical makeup. However most of these systems are very large and would require more power than some of the I2C sensors we included here. However the great part of this platform is that any sensor you want can be added and can utilize the same layout to extend it. The only limitations we look at are when sensors become full modules themself and therefore adding them to the small sensor array would add complexity when trying to get the device to stay underwater.