Remote Water Quality Monitor

Joseph Fitzgerald, CompE, David Gibb, CompE, Tamson Quang, CompE, and Patrick Denert, CompE

***Abstract*- Many water sources in the world are currently contaminated or at risk of contamination. The purpose of this system is to allow for the remote monitoring of water sources in order to preserve the health of many communities. Allowing remote monitoring allows for faster results and constant monitoring of water quality.**

I. INTRODUCTION

Testing water quality is essential for measuring the health of our aquatic ecosystems. However, many aquatic areas where routine maintenance needs to be done can be hazardous to humans. In order to perform these tasks, environmental workers often have to get in the water with dangerous wildlife such as alligators and hazardous objects such as sharp metal submerged in murky water. A small buoy with sensors dangling into the water that could be retrieved from a safe location could help keep humans out of harm's way, reduce travel time and physical labor, and it could also help reduce expenses on scuba gear.

A. Significance

To this day, we witness many environmental issues directly related to the quality of our water. Water is used to support our ecosystem worldwide, yet the testing for water quality is still under review for “why monitor water quality [1]”? To follow up with this question we have to look for reasons to measure water quality, what benefits come from so, and it comes as so “Water-quality monitoring is used to alert us to current, ongoing, and emerging problems; to determine compliance with drinking water standards, and to protect other beneficial uses of water. [1]”. In several studies, we find alarming quantities that raise concern over water quality, “Every day, 2 million tons of sewage and industrial and agricultural waste are discharged into the world’s water (UN WWAP 2003), the equivalent of the weight of the entire human population of 6.8 billion people.” [8], “Lack of adequate sanitation contaminates water courses worldwide and is one of the most significant forms of water pollution. Worldwide, 2.5 billion people live without improved sanitation. (UNICEF WHO 2008)” [8].

B. Context and Survey of Similar Solutions

Last year for SDP22, there was a similar problem in which they solved for testing water quality over a design called riversense. Riversense was able to measure 4 water quality metrics, fit under an affordable budget and had autonomous data collection. A weakness we observed compared to our own design was the number of water quality metrics we plan to measure and the solution in which our design will be implemented. Riversense is designed as a mounted solution that seemed unreliable whereas we plan on demonstrating our design as a buoy.

Another solution was the JXCT Water Quality Monitoring System. The System is very close and similar to our design, it has a floating buoy system, a durable metal frame, dual solar panels, record 4 water quality metrics and automatic remote sensing. The key difference in our design was the number of water quality metrics measured, we planned on having a design very close to this but be able to measure more water quality metrics as well.

C. Societal Impacts

Our system focuses on groups who look for an affordable method in testing water quality for either work or educational purposes. There are many cases in which testing water quality becomes hazardous due to extreme measures of nature, therefore our prototype is designed to measure parameters in terms of water quality and report back to those who are measuring so. These groups deploy a field of what parameters we plan on having our design measure, how long our measurements will progress, and where our design will measure such parameters.

D. Goals, Specifications and Testing Plan

Specification	Corresponding Testing Plan
The system will float on the surface of the water and resist water currents	System was tested in the Connecticut river for buoyancy and stability
The system rises and falls with the tide without issue	Place flotation device in tub and add water to simulate tide
The housing for the electronics will be waterproof	Put the flotation device in water to test for leaks
The device will transmit data when within range of cellular service, otherwise store to sd	Read the HTTP response codes and the data stored on SD card after a certain period of time

card and transmit all readings upon connection	
Record measurements of Temperature (dC and dF), Turbidity (NTU), TDS (ppm), flow (L/h), and Conductivity (us/cm) with <10% error	Check against known sensors or calibration fluids
The battery will last for at least 3 months	Measure average power consumed then calculate how long the battery will last
The solar panel will charge the battery	Isolate the battery charging circuit from the system and measured with a multimeter
Send parameter results to server periodically through cellular communication	Check timestamps of messages sent
The server will be able to handle at least 10 requests per second	Write a program to spam requests at the server, check database after to ensure all data is recorded
Alerts user if parameters reach a certain user defined threshold	Add contaminants to water bucket to change the values and observe that an alert is sent to the user
The device will read sensors in a user specified time interval (15 minutes default)	Observe readings sent using serial to ensure the readings are correct

Table 1: Specifications and Testing Plan

II. DESIGN

A. Overview

In order to solve this problem we ran through multiple scenarios in which we had several different designs. Our initial design was to create a water quality monitoring boat controlled with RF communications. (Appendix Section A). After running through as to if the scenario was realistically possible or not, we went through previous designs in last year's projects and other possible solutions demonstrated by professional engineers. Once we compiled all of these ideas together we came to a conclusion with a buoy system. The technologies we are using are sensors to record data readings, an atmega1284p microcontroller to take the readings and convert the analog readings to digital and prepare a data payload to send to our cellular module, The microcontroller also controls the cellular module using AT commands. Our cellular module is the Sim7000A which is used to connect to a cellular network and opens a TCP connection to send data readings.More information on the technology and standards can be found in Appendix B. Wifi and bluetooth were also considered but cellular allowed for many more applications to be possible as it has a large range and would allow us to leave our device outside without needing an external device to send data. The Atmega328p was considered earlier in our system device but eventually had to be changed due to SRAM limitations. The microcontroller choice was critical for fulfilling the specifications as it had to be able to perform all the necessary functions as well as draw as little power as possible in order to be able to run off solar power and battery for a long period of time.

B. *Hardware*

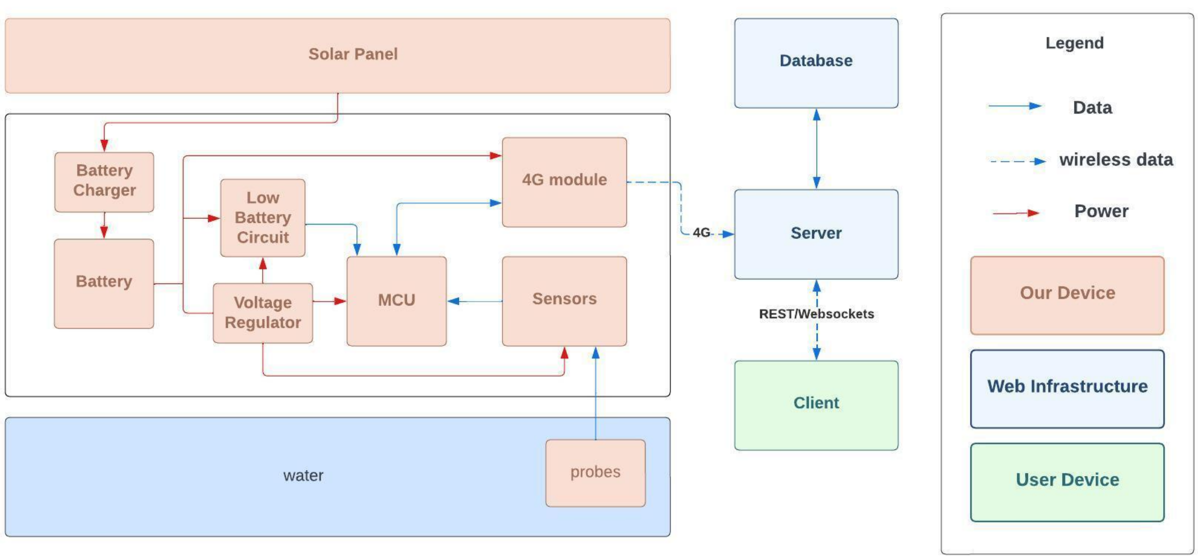


Figure 1: Hardware Block Diagram

This block diagram shows all the hardware components in the system. The microcontroller chosen is the Atmega 1284p due to its low power consumption [2]. This chip was chosen since the processing for the sensors is fairly simple so a powerful processor was not needed. However, we needed the increased RAM the 1284p has to offer to be able to run all of our programs. The p indicates picopower which allows for lower power usage compared to a standard Atmega 1284. The low power also helps increase how long the system can operate remotely by limiting battery usage. The sensors chosen are: the DFR0198 for temperature [3], the SEN0189 for turbidity [6], the SEN0244 for TDS [5],and the SEN0550 for flow [9]. These sensors were chosen for their reliability and documentation. These sensors are connected to the microcontroller in order to have their analog signals get converted with the analog to digital converter before the microcontroller sends the data to the DFR0763 cellular module to send the data to the server [5]. The flow and temperature sensors use digital outputs and the TDS and turbidity sensors use analog outputs. This cellular module was chosen since it is updated to work with current cellular technologies such as LTE. This system was designed with knowledge from the embedded systems class, mainly interfacing sensors with a microcontroller. To build this system the datasheets were studied and the outputs were compared to the atmega328p pins and capabilities in order to ensure that the microcontroller would be able to handle the sensors. To test this system, the sensors will have their probes placed in water and the system will be monitored for the results of the readings and data sent by the cellular module. The hardware block diagram also contains a buck voltage regulator which regulates the voltage from ~7.4-~8.4 V down to 5 V. There is also a low battery detector on the hardware diagram that uses the voltage from the voltage regulator and the voltage from the battery (with a voltage divider) to compare the two voltages and once the battery voltage drops below ~7.2 V the low battery circuit reads high. A solar panel, and two 3.7 V lithium ion batteries are featured in the hardware block diagram. An experiment to test this block was running the system and analyzing the outputs in the serial monitor which would show if the readings are correct and that the ADC is performing a correct translation. The system also outputs the state of the battery.

C. Software

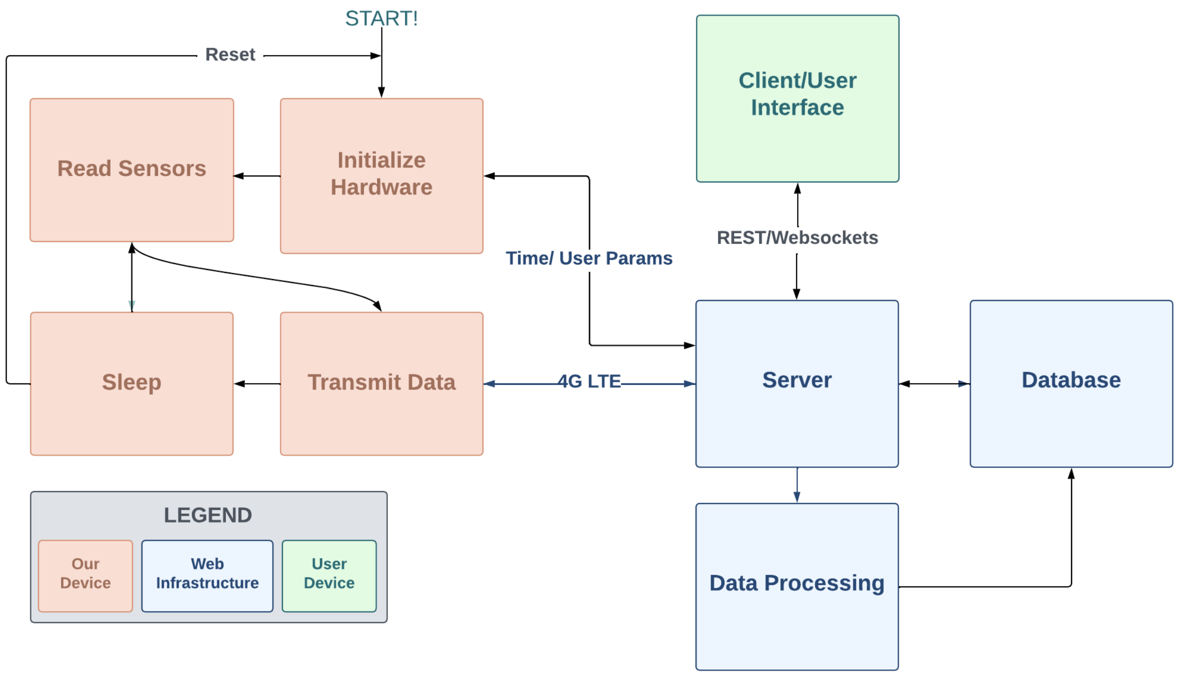


Figure 3: Software Block Diagram

Data received by the sensors are transmitted to a server via 4G/LTE Communication, the MCU we'll be working with to receive this information is an ATmega1284p [2]. The device's goal is to read/send data received from the sensors and go through a sleep cycle to repeat and reduce power consumption. After the MCU gathers the sensor values, the readings are compared to sensor values and if they are outside of a user threshold an alarm is sent to the server. Before any data is sent to the server, the MCU encrypts the data using AES and appends a MAC to the message before it is sent to the server. Communication with the cellular module is done by sending AT commands/

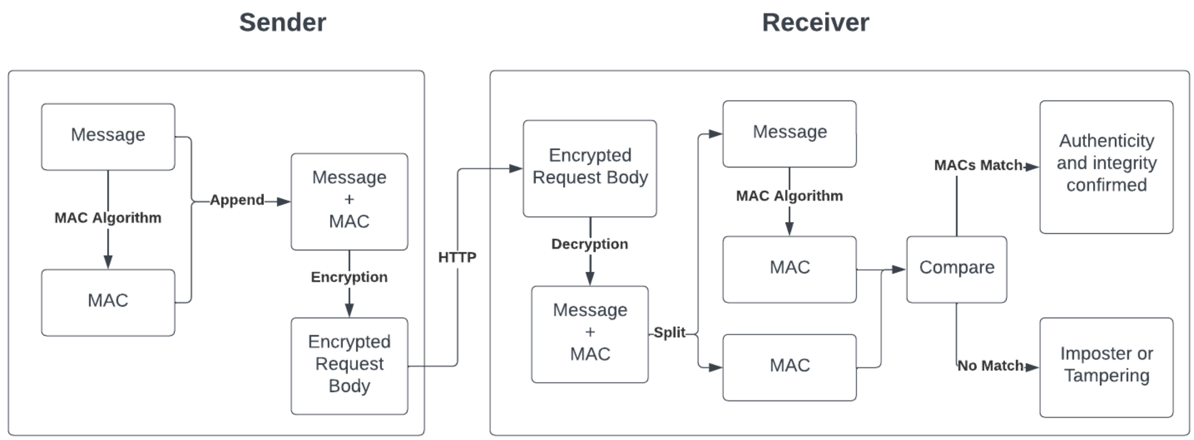


Figure 4: Security Flowchart

If the data is unable to be sent to the server then it is instead saved to an SD card. The server is a Node.js server running on a cloud web hosting platform called fly.io. The primary purpose of the server is to listen for incoming requests and perform the appropriate responses based on the request it receives. Requests will be coming from two sources, the device and the user. The device will send two types of requests, upon device initialization it will need to know the correct time and date in order to provide accurate timestamps for sensor readings and it will also need to get user specified parameters contained on the server. In order to do this it will send an HTTP GET request, and the server will respond with the requested information. The device will also need to send sensor readings to the server which it will do with an HTTP POST request. Upon receiving the sensor readings the server stores them in a database. In order to view the data the user will send an HTTP GET request to the server and the server responds by rendering a page and sending it to the user. This rendering process is done using the Vue.js and Nuxt.js frameworks and for graphing the data we used the ApexCharts library. We have a number of analytics metrics that are computed in the client side application that have various adjustable parameters. Upon receiving the page the user's device initiates a websocket connection with the server so the server can relay updated data to the user as it receives it rather than having the user's device continuously poll the server. The data line in the graph as well as the computed analytics update in real time upon receiving new data over the websocket connection. Knowledge from the Security Engineering class allowed us to properly encrypt our data. Knowledge from computer networking allowed us to understand different communication protocols and how to write code to get and send data, specifically using cellular protocols, TCP, and HTTP. To test this, our entire system was run and the outputs were analyzed. The system would take in readings and then send them to the cellular module which sends the information to the server. Then on the server we can see if the data is sent correctly, if the time was correctly obtained in the initialization loop, and if any warnings were sent.

III. THE PROTOTYPE

- *A. Prototype Overview*

The prototype will have all the sensors connected to the Atmega1284p microcontroller and also have the cellular module connected to the microcontroller. There will also be an SD card module attached to the mcu. The probes for the sensors will be connected to their respective sensors and will have the ends sitting in a bucket of liquid for testing purposes. The sensors will read the probe inputs based on the fluid in the bucket and send analog signals to the microcontroller which will convert them to digital signals to be sent to the cellular module. The cellular module will then send the data using the antenna to the server and database over cellular communication. If the cellular module is unable to communicate with the server then the data is instead saved to the SD Card Module. Part of the prototype also includes our waterproof housing complete with an anchor and counterweight to keep the prototype stable in rough conditions and prevent it from floating away in the current.

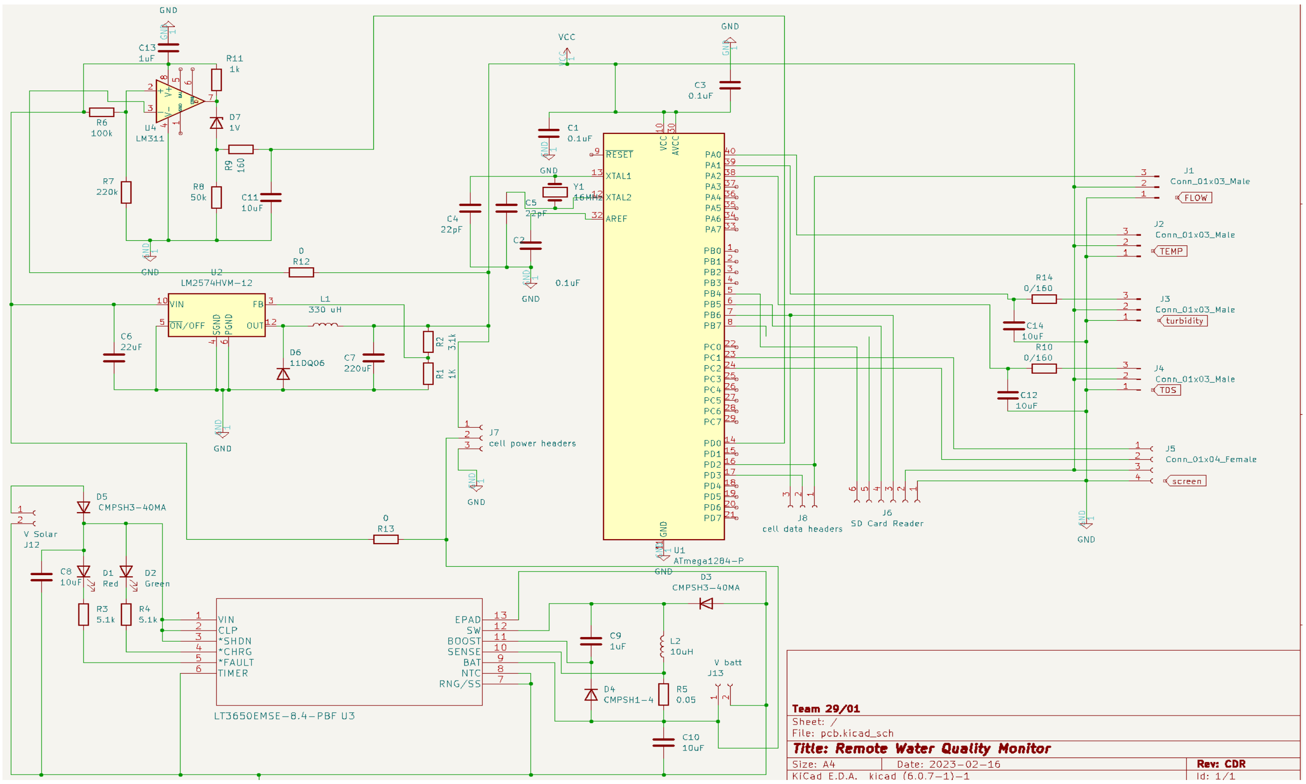
Figure 2: FPR Product in the Connecticut River with sensors and solar panel.

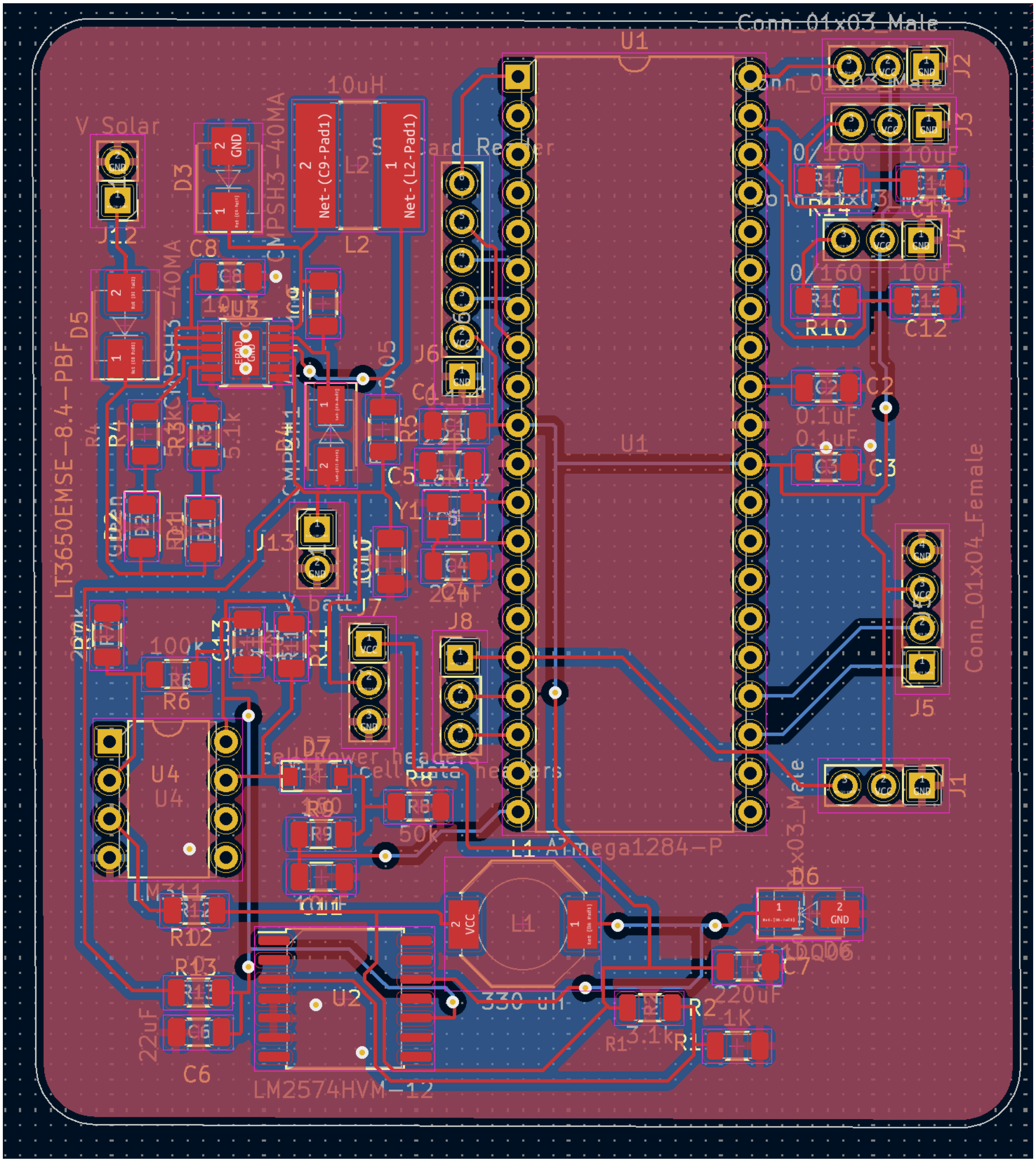
- *B. List of Hardware and Software*

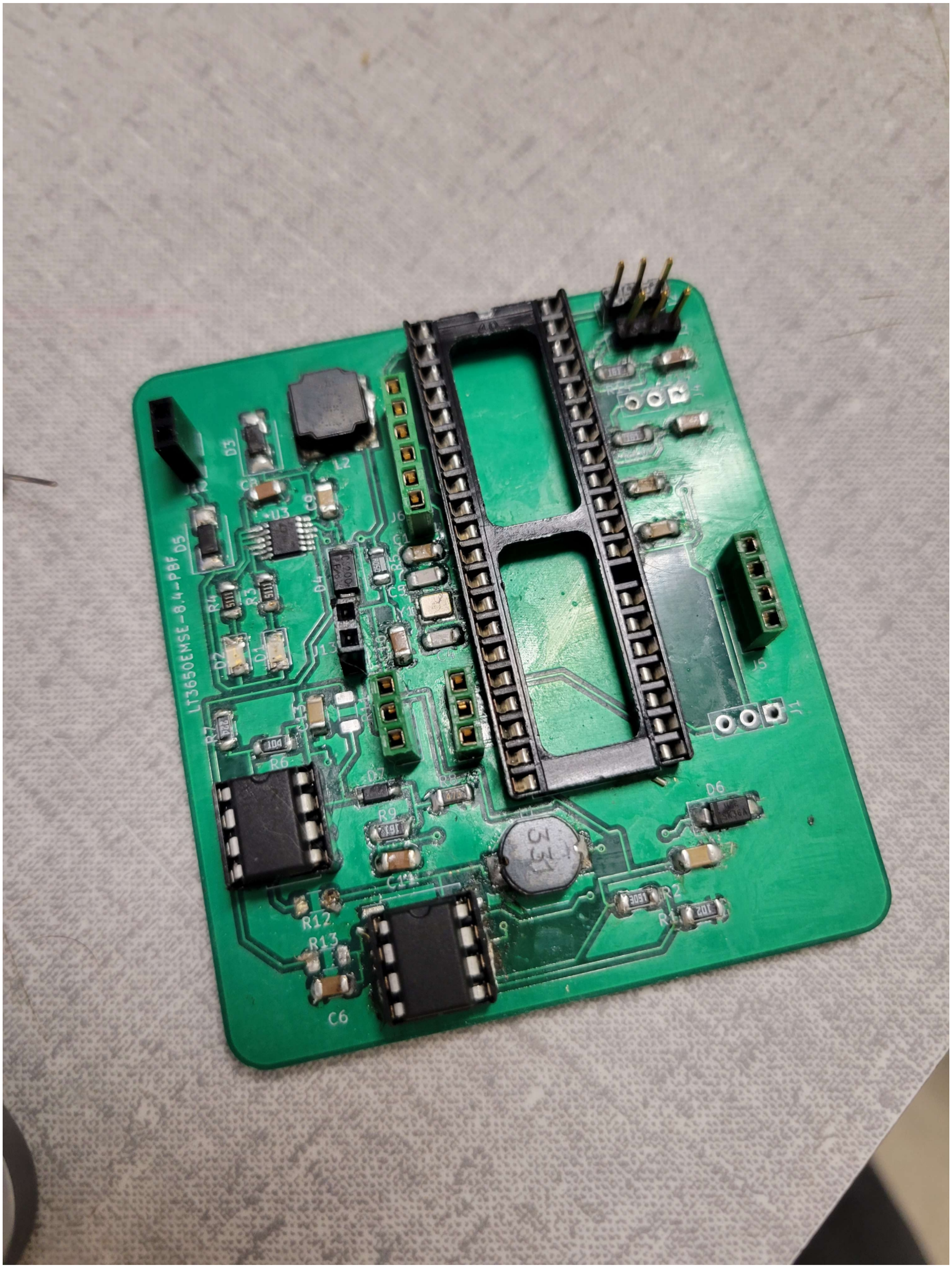
Hardware	Software
Atmega 1284p (MCU)	Arduino bootloader
SEN0189 (turbidity sensor)	PuTTY(serial monitor)
DFR0198 (temperature sensor)	Fly.io
SEN0244 (TDS sensor)	Node.Js
SEN0550 (Flow sensor)	Vue.js
ANT-5GWWS4-SMA (antenna)	Nuxt.js
DFR0763 (cellular module)	simple-statistics
Micro SD Card Module	zemlyansky / arima

- *C. Custom Hardware*

We started with the schematic which was an iterative process in which research was done, design was created/revised, and problems or areas that could be improved were identified. We would repeat this process until we had a design that was ready to move on to the next phase. The FPR PCB was designed to contain 4 isolated modules that could be tested independently and connected with 0 ohm resistors as opposed to jumper cables which is what we had used in previous iterations. These modules are the microcontroller and sensor module, the battery charger, the low battery detector and the voltage regulator. We also included headers for connecting additional hardware including an OLED screen, an SD card, low pass filters, and both power and data headers for connecting the Cell module. One major change to the PCB for the FPR iteration is that we had to make significant modifications to the layout to support the larger footprints for our inductors and for our new microcontroller. Design began Immediately after CDR and took about 3 weeks. There was a problem with our gerber files which held up the ordering process for about another week. Eventually we did get the PCB ordered and populated, it was not ready in time for FPR, however it was ready and functioning for Demo Day on Friday although we did not move the prototype to the PCB until the demo on Saturday.


GND
C13
1uF
R11
1k
R6
100k
U4
LM311
D7
1V
R9
160
R7
220k
R8
50k
C11
10uF
GND
0
R12
U2
LM2574HVM-12
VIN
FB
ON/OFF
SGND
PGND
OUT
L1
330 uH
C6
22uF
D6
11DQ06
C7
220uF
R2
3.1k
R1
1k
GND
J7
cell power headers
GND
VCC
GND
C3
0.1uF
C1
0.1uF
GND
Y1
16MHz
C4
22pF
C5
22pF
C2
0.1uF
GND
RESET
XTAL1
XTAL2
AREF
VCC
AVCC
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
GND
U1
ATmega1284-P
GND
J8
cell data headers
J6
SD Card Reader
J1
Conn_01x03_Male
FLOW
J2
Conn_01x03_Male
TEMP
R14
0/160
C14
10uF
J3
Conn_01x03_Male
turbidity
R10
0/160
C12
10uF
J4
Conn_01x03_Male
TDS
J5
Conn_01x04_Female
screen
GND
D5
CMPSH3-40MA
V Solar
J12
0
R13
C8
10uF
D1
Red
D2
Green
R3
5.1k
R4
5.1k
VIN
CLP
*SHDN
*CHRG
*FAULT
TIMER
EPAD
SW
BOOST
SENSE
BAT
NTC
RNG/SS
LT3650EMSE-8.4-PBF U3
D3
CMPSH3-40MA
C9
1uF
L2
10uH
V batt
J13
D4
CMPSH1-4
R5
0.05
C10
10uF
Team 29/01
Sheet: /
File: pcb.kicad_sch
Title: Remote Water Quality Monitor
Size: A4
Date: 2023-02-16
Rev: CDR
KiCad E.D.A. kicad (6.0.7-1)-1
Id: 1/1


Conn_01x03_Male
U1
J2
J3
J4
J1
J5
Conn_01x04_Female
V Solar
J12
10uH
L2
C8
D3
D5
LT3650EMSE-8.4-PBF
J6
J13
J7
J8
U4
LM311
R6
100k
R8
50k
R9
R12
R13
C6
22uF
L1
330 uH
U2
LM2574HVM-12
Atmega1284-P
D6
C7
220uF
R2
3.1k
R1
1K
C2
C3
0.1uF
R10
C12
10uF
0/160
C14
R14


LT3650EMSE-8.4-PBF
D3
L2
J12
U3
D5
J6
R4
R3
D2
D1
D4
J13
R7
R6
D7
R9
R12
R13
C6
D6
R2
J5
J1

D. Prototype Functionality

The sensors in the prototype were working properly. The sensor probes were placed in water and were then read using PuTTY. The values read were correct for what measurements should be for tap water. The temperature sensor was compared with a thermometer and was correct up to 0.01 C. The other sensor values indicated that the testing liquid was clean water based on their readings but will need to be compared to other testing equipment in future iterations to ensure their accuracy. 1 issue that came up with the sensors was that the tds sensor was incorrectly wired by the manufacturer and to fix this the datasheets were read and then the wiring was corrected and the issue was resolved. Another issue was that the Turbidity sensor had to be recalibrated. This was done by turning the potentiometer on the sensor until there was a reading of 4.2V in distilled water. There were no other major issues regarding the sensors for this system.

The web application is functioning correctly, there is a fully built out API for basic CRUD functionality of sensor data, warning data, and user parameters. Page is served and data from the sensors is viewable and all proposed metrics of analytics are correctly computed. The analytics are viewable in a line graph which updates in real time as new data is received or as algorithm parameters are updated.

The SD Card module is working properly, the module is able to store and retrieve all of the data that was sent to the module. There was a small issue with the module that would cause the mcu to crash if the mcu used too much of the SRAM. This was fixed by swapping out our microcontroller for one with more memory.

The security in the system is working properly as well, the MCU is able to encrypt the data in a reasonable amount of time and the server is able to correctly decrypt the data that was sent. Authenticity and integrity of the data are also able to to be verified with a MAC generated by both sides.

The cellular communication is currently working properly. The module receives a signal from the cell tower and attaches to a service to a signal which allows the data to be transmitted to a cloud service and there since sent to the web server. This was tested by using AT commands to have the module send our sensor readings to the server and we found that the data was sent correctly without any loss or corruption. This module was a large issue for a majority of the system and through research we found that our device was possibly being blacklisted by the network for being an unrecognized device. To fix this we changed our provider to Hologram and their support helped set up our module and got it to work.

The housing, anchor, and counterweights functionality was also tested. The housing did prove to be waterproof and the anchor was successful in keeping the prototype from floating away in the current and wind of the Connecticut river.

The custom hardware is functioning as expected, although on demo day Saturday we encountered a problem with our microcontroller and we did not have a replacement so the microcontroller subsystem was not working on that day but all the other subsystems were working.

E. Prototype Performance

Our prototype has been able to conduct reads on all 4 of our metrics: TDS, Temperature, Flow, and Turbidity. Conductivity was able to be calculated from the readings. This can be observed by reading the serial output of the mcu. The prototype is able to alert users if any of the readings is outside of a certain range which is also checked via the serial monitor at the moment. The prototype is able to save data to an SD card if the cellular communication module is malfunctioning. This can be observed through reading the SD card on a computer. The device is also able to take readings based on a user set time, typically 15 minutes is the industry standard. This can also be observed through the Serial monitor.

We also were able to set up our web server, so far we’ve been able to add visuals with key analytics. The web application behaves according to the specification we've defined and even greatly exceeds the number of requests we are able to process per second.

We were able to establish cellular communication with our server. We know this is functioning properly because the cell carrier records any data we send from our cellular module, and we can also see it show up in server logs as well as in our user interface.

IV. CONCLUSION

Currently the project is functioning but is not as polished as we would have hoped to get it at this point in the year. Our project has a full end to end communication pipeline from the sensors to web application that goes through a cellular network. The server and web interface are also working and are able to display the sensor data and warnings as well as analytics. The housing also floats, is waterproof, and the anchor and counterweight systems work effectively. There is still some difficulty working with the Cell modem and some behavior we still do not quite understand such as certain global variables changing or getting overwritten after making an HTTP request, and our program mysteriously crashing or restarting after a certain number of loops if we don't turn the cell module off after every time it opens and closes a TCP connection. We have found working solutions for these problems however the underlying cause is still unknown and I all team members would like to understand better. Another feature that was not promised but we would have liked to implement is using a programmable switch to cut power to certain subsystems such as the sensors and cell module when they are not in use.

ACKNOWLEDGMENT

We would like to thank our advisor Professor Yadi Eslami for providing us with lots of feedback and advice throughout our entire Senior Design Project development.

We would also like to thank our course evaluators, Professor William Leonard and Professor Sandip Kundu for their time in viewing our presentations and providing us with great feedback on our project.

We would also like to thank Professor Jeremy Gummeson for helping us solve an issue with our cellular communication.

References

[1] Myers, D. N. (n.d.). *Why monitor water quality? - USGS*. Why Monitor Water Quality. Retrieved October 30, 2022, from https://water.usgs.gov/owq/WhyMonitorWaterQuality.pdf

[2] "Atmega328p datasheet - microchip technology." [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.

[3] "DFR0198 datasheet by DFRobot," *Digi*. [Online]. Available: https://wiki.dfrobot.com/Waterproof_DS18B20_Digital_Temperature_Sensor__SKU_DFR0198_

[4] DFRobot SIM7000_Arduino_NB-IoT_LTE_GPRS_Expansion_Shield_SKU__DFR0505_DFR0572. [Online]. Available: https://wiki.dfrobot.com/SIM7000_Arduino_NB-IoT_LTE_GPRS_Expansion_Shield_SKU__DFR0505_DFR0572. [Accessed: 30-Oct-2022].

[5] Gravity Analog TDS Sensor Meter For Arduino.[Online].Available: https://wiki.dfrobot.com/Gravity__Analog_TDS_Sensor___Meter_For_Arduino_SKU__SEN0244

[6] "Turbidity Sensor Sku: Sen0189 - digi-key." [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/SEN0189_Web.pdf.

[7] PIR Sensor Module, SE10, Data Sheet*, Sparkfun*, https://www.sparkfun.com/datasheets/Sensors/Proximity/SE-10.pdf .

[8] *Advancing Water Resilience*. Pacific Institute. (2020, April 17). Retrieved October 30, 2022, from https://pacinst.org/

[9] SKU SEN0550 Gravity Water Flow Sensor G1/2.[Online]. Available:https://wiki.dfrobot.com/SKU_SEN0550_Gravity_Water_Flow_Sensor_G1_2

Appendix

A. Design Alternatives

The first design alternative was to create a water quality monitoring boat controlled with RF communications. This idea was discarded since it would have required having a person at the location to control the boat and collect water samples which would not have been as beneficial as having a remote water monitoring system that collects data without having a person present. This alternative helped shape the system by making the group question what is actually important for the client and what design would work best for the client, which ended up being a remote water quality monitoring system.

B. Technical Standards

IEEE 260-1978: IEEE Standard Letter Symbols for Units of Measurement (SI Units, Customary Inch-Pound Units, and Certain Other Units)

Each of the units we use in the project are the standard units and symbols set by IEEE

IEEE 1619.1-2007

IEEE Standard for Authenticated Encryption with Length Expansion for Storage Devices

AES was used for encryption of data

ITU-T V.250 SERIES V: DATA COMMUNICATION OVER THE TELEPHONE NETWORK Control procedures

International Telecommunication Union - Telecommunication standard for AT (attention) command language used to control modems. This was used by the mcu to send commands to the cellular module to perform commands such as turning on/off, establishing a connection, and sending data to an endpoint.

RFC 9293: Transmission Control Protocol (TCP)

One of the standardized Transport layer services set by the Internet Engineering Task Force used by the module to connect to an endpoint.

RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1

One of the standardized Application layer services set by the Internet Engineering Task Force used by the module to communicate with the endpoint to send and receive data using HTTP commands such as GET.

ISO/IEC 21778:2017

Information technology — The JSON data interchange syntax

The data is sent in JSON format to ensure the server is able to correctly identify the information,

SD Association SDHC standard

Creates standards for SD cards which use a FAT32 file system such as the one used in this project. This was used with the SD module to store data on the card.

EPA Water Quality Metrics were followed as default values for the Alert System

C. Testing Methods

To test that the sensors are working properly, the system will have the sensors be placed in a bucket filled with water. The readings will be recorded and compared to known working sensors or instruments such as a thermometer. The water will then have contaminants added to it in order to see if the sensors will notice the change and update the readings accordingly, such as by adding hot water to increase the temperature. This experiment was used to test the sensor hardware to see if it was calibrated correctly. The experiment also tests software for the sensors by making sure the code is able to read the values and output them correctly.

To test the SD card module, the sensors are left for some time to have them upload the info to the module and then the SD card is read after some time and the readings are compared to the readings in the Serial monitor to make sure they were saved correctly.

To test that the device is waterproof, floats, and can resist the water currents, the device will be placed in the floatation device and placed in the Connecticut river along with the anchor, and left there for 15 minutes.

To test the battery life, the system will be measured at peak power consumption by having all the devices working at once and measuring the power consumption and doing calculations to see how long the battery will last when the system is at peak power consumption.

To test the solar system the panel will be placed outside on differing weather conditions and the power generated will be measured.

To test the server, a program will send 10 requests a second to test for reliability and to test that the server detects measurements outside of a range a false reading will be sent.

To test that the cellular communication system sends data every 15 minutes, the server will check the timestamps of the data sent. To check that the data is sent to the server the http response code will be checked.

D. Project Expenditures

Part	Cost
Sensors and PC	$147.00
MDR PCB	$10.00
Battery and Sim	$39.44
Solar Panels	$30.98
Floatation Devic	$38.46
CDR PCB comp	$82.00
CDR PCB:	$25.00
PCB MDR:	$25.00
FPR PCB Parts	$60.23
FPR PCB	$30.00
Total Spent	**$488.11**

Development Expenditures

Part	Cost
Battery	$60.00
Sensors	$40.00
Cellular Module	$40.00
PCB	$30.00
PCB Components	$30.00
Solar Panels	$30.00
Floatation Device	$40.00
Total	$270.00

Production for 1 unit

E. Project Management

Our team is organized in a manner where each member is a lead for a specific subsystem of the project. Dave is responsible for Web programming and PCB. Tamson is responsible for establishing cellular communication. Patrick is in charge of sensor programming. Joe is in charge of the hardware design.

F. Beyond the Classroom

Tamson Quang - Having a hand in this project, I've witnessed moments in which there were skills that I required to develop in order to be participating in this design. In this project, I stand in charge of the wireless communication between our design and the user. Learning how to process a working communication from hardware to software and then from software to user is a skill I've polished for this project. Working with peers and alumni play as a powerful resource for me in this process, once in the professional world I do see a connection between now and then.

Patrick Denert - I had to learn about how the different sensors operate, how they produce an output and how to display that output. I also learned a lot about how SD cards work and how they operate with the mcu. I also learned a little about web development as I had to work with Dave to make sure the mcu is able to produce data in JSON format so the server can understand it. I also learned a lot about analog to digital converters which are used quite often since most of the sensors use analog outputs for data. I also had to polish the embedded programming skills I learned in the Junior Design project and apply them to the Senior Design Project. I also learned a lot about cellular communication in order to help with writing code to control the cellular module. I also used my knowledge from computer networking to help connect the cellular module to an endpoint to get and send data. The most useful resources have been the datasheets since they have lots of extensive information. DFRobot especially has lots of information on their products, including sample code and how to calibrate sensors. This experience will be very useful in the professional world since I have gained a lot of experience working in a group, the project development lifecycle, and programming experience. I also gained experience in many different areas of software development which would help me become a more rounded programmer.

Joseph Fitzgerald - I have had to brush up on basic information about various hardware parts of our project. Have also learned the proper way to read data sheets and look for information that is needed for assembling a system. I have also learned to solder because we are going to need to solder to create our system. This project has taught me a lot about working in a group setting and communicating as a group.

David Gibb - For this project I've made great use of skills that often don't get tested in the classroom. Some of these skills include creating a specification to solve a problem and working through an iterative process to create a design that satisfies the specification. There are many project management skills that I have developed as well such as making design decisions while taking into account our budget as well as time. Sometimes The ideal solution is outside of our budget or would take too long to develop. One last skill that I have been working on is teamwork, we don't do many team based projects so this has been a unique challenge learning how to divide the workload evenly, but still work together on a number of problems we encounter.