Turbidity Sensor Product Launch Document

University of Minnesota
EE 4951W
May 2, 2018

Group Members:
Corbin Buechler
Amanda Hayden
Aaron Nightingale
Sean Rogers
Erik Sundberg
Jacob Wolf

Advisors:
Professor Joseph Talghader - Electrical and Computer Engineering
Professor Andrew Wickert - Department of Earth Sciences
Anthony Aufdenkampe - Senior Environmental Scientist, LimnoTech

Abstract/Summary

Turbidity, the cloudiness or haziness of a fluid, is a major symptom of water pollution in all bodies of water, and it is only getting worse. Detecting water turbidity levels in remote locations is challenging and expensive, and it is often not feasible. Our team has developed a portable and inexpensive turbidity sensor that will work with multiple open source data loggers. The crucial components of the sensor include its waterproof housing, the light source, light detecting circuitry, and the software responsible for relaying data to the open source data logger. After these individual parts were designed, they were integrated into a final product that can be left in remote locations for data acquisition. Researchers hope that this sensor will be able to be reproduced on a larger scale to help monitor and mitigate turbidity levels in rivers, lakes, streams, and perhaps even drinking water in developing countries.

Introduction

This paper addresses the following areas of the project; motivation, objectives and customer requirements, technical background, design and description, team organization, overall goals, future improvements, timeline, and budget. The motivation for the project is simple, to develop a low cost and portable turbidity sensor that can be used with multiple open source data loggers and be deployed for lengthy periods of time. The technical background gives the reader an understanding of what a turbidity sensor is and how it operates. The design and description includes details of development of the main parts of the design. The first major component being the LED and sensor circuit that contains the light to frequency converters and software that correlates the frequency to some turbidity measurement. The second major component being the I2C communications interface between our sensor and the data logger. The third major component is the circuit design and layout. In particular, the LEDs and sensors had to be accurately positioned in order to optimize the light passing through the acrylic tube. The last major component is the waterproof housing and cable, as well as the overall integration of each component. In regards to team organization, we chose to delegate different components of the project to each person. Although each person was responsible for a different part of the project, there was a lot of collaboration in order to integrate the final product. The goal of the project very closely followed the initial specifications and requirements of the client. Although the project was successful, there are some changes that could be implemented if the design were to be reproduced, mostly having to do with time and monetary constraints. As for the timeline, the integration of all parts began the last week in March to allow significant time for testing and calibration that were to be completed by May 1st. The development costs came in under budget, at around $750. In addition, the final sensor design costs less than $100, as requested.

Motivation

Given that Minnesota has thousands of lakes and streams, water turbidity is a serious problem locally. However, measuring it is difficult. There is not enough time and resources to test all bodies of water. Sensors that are available are expensive, often costing over $1,000. To make matters worse, most commercial sensors are designed to be used in a lab setting, meaning researchers and scientists cannot take measurements in the field. Lab sensors also require calibration using special solutions, which significantly increases the cost. The sensors that can give real-time readings are even more expensive, so much so that it is not fiscally viable for a company to purchase more than one or two of these units. This project proposed the development of a low-cost turbidity sensor that can be field deployed in remote locations. In turn, this sensor will help quantify and mitigate the effect of practices that lead to contamination and soil loss. Our turbidity sensor uses low amounts of power, and can be powered from a solar cell in field testing situations. The turbidity sensor also communicates with an open-source data logger, which saves the turbidity readings onto an SD card. This allows researchers to have access to real-time turbidity readings without having to bring samples or the sensor back to the lab. All of the designs for this turbidity sensor are open-source, allowing other researchers to build their own turbidity sensor and continue the development of the design. When the design and documentation is made available, this project could lead to massive changes in environmental research and protection.

Objectives and Customer Requirements

The main goal of this project is to create a working turbidity meter, loosely following a design provided by the client. The sensor shall use light scattering and transmission to measure the turbidity of water. The clients currently use two different data-loggers for field measurements. Another goal of this project was to write code that interfaced our sensor with these open source data loggers. The client required that all sensor and communications code must be open-source for future changes and use.

The client provided several overall project specifications as well. First, a working turbidity sensor must be developed, prototyped, and delivered to the client by the end of the semester. In addition, the fully operational sensor must be shown at the senior design show. The sensor should be able to correctly read the turbidity of various liquids in a blind test. The clients would provide us with a budget of $1,000. We were expected to get approval and place all purchases through their lab, with the exception of minor circuit components, which could be obtained from the ECE Depot if possible.

Because the client would like to use this sensor in the field, there are several important specifications for the device itself. Foremost, it must be enclosed in a rugged, waterproof enclosure that can withstand up to three months of deployment in typical bodies of water. For that reason, it must be IP68 rated. For example, it must be able to withstand currents pushing it around and any possible impacts from debris churned up during storms or other events. It must have the minimum power consumption possible for prolonged operation. The enclosure must also be able to handle depths of up to three meters. The sensor is being powered from a solar panel mounted on a central box nearby. There are other instruments inside this box that also draw power. It is important that our sensor does not overload this box. Our client uses sensors that draw a maximum of 50 mA, so the current consumption of our sensor should be below that. The sensor must operate on 5 and/or 3.3 volts, as that is the only power provided from the box. The sensor itself must use a light source (LED or laser) to measure scattering at different angles and transmission, in order to better calculate the NTU of the water in question. The sensor code within must be modeled in Arduino in order to be compatible the data loggers. All final code must be open source as well. Calibration, using standard turbidity liquids must be performed, and should be done in a lab setting. The sensors will report readings in Nephelometric Turbidity Units (NTU) with an accuracy of 3%. The target price per sensor should be less than $100. Diana Karwan, the co-lead client for this project, would like to test a unit in the field in April, depending on weather.

The client has also requested several pieces of paperwork along with a working sensor. First, they would like a bill of materials in order to see a breakdown of required parts and costs to manufacture several sensors. Next, they have also requested instructions on how to build the sensor. These must be simple enough so that anyone can follow them. Along with the instructions, the clients would like information regarding important design decisions made throughout the project. This will make it easier for the client to make further changes or improvements, based on difficulties we experienced during our development and prototyping.

Technical Background

The most commonly used unit of measurement for turbidity is the Nephelometric Turbidity Unit (NTU). Turbidity is determined by shining a beam of light through the liquid sample and measuring the amount of light that is scattered at some angle. The most common arrangements measure 90 degree side-scattering, backscattering, which is the light reflected back toward the source, or transmission, which is the light that passes through the sample in a straight line from the source. Many commercial turbidity meters measure only a single direction of light scattering, which can perform well in measuring low turbidity samples but are less accurate past a certain threshold NTU. In more turbid conditions, the light may be scattered multiple times, causing less of it to reach the sensor, which would be interpreted as a decrease in turbidity. By measuring both light scattering and transmission, a sensor can take accurate readings over a wider range of turbidity values. As discussed below, our sensor measures the light transmitted and the light scattered at 90 degrees from two different light sources.

Detailed Design Description

The exterior of the turbidity sensor consists of a waterproof aluminum box with a square acrylic water inlet tube mounted horizontally through the enclosure. Two infrared LEDs and two light intensity sensors are mounted around this square tube so that each LED faces one of the sensors. This arrangement allows the sensors to pick up the transmitted light and the 90 degree side-scattered light from each of the light sources. The LEDs used in our design have a wavelength of 860 nm, which was chosen to comply with the ISO 7027 standard for measuring turbidity. For light detection, we decided to use a pair of integrated circuits that senses light and outputs a square wave what a frequency that is proportional to the light intensity. These signals are collected by an ATmega328P microcontroller, which counts the number of rising edges over a period of time to determine the frequency. In most cases, this measurement is taken over one second, but it can be extended to up to two minutes to ensure that very low frequencies are counted accurately. Since the light sensors can output frequencies up to nearly 600 kHz, while the microcontroller's interrupts can only reach up to 75 kHz, the light sensors' outputs are routed through a pair of binary counters that effectively divide the output frequencies by eight before they reach the microcontroller's inputs. This process introduces a potential error when collecting low frequency signals, but this is minimized by extending the sampling period in these cases.

In addition to capturing the outputs from the light intensity sensors, the ATmega chip also turns the LEDs on and off at the appropriate times. Since the maximum side-scattered light from a source is significantly less than the maximum transmitted light, we have developed a circuit that can light up each LED at a higher and a lower intensity to use as much of the light sensors' range as possible. The sensing routine first measures the amount of ambient light present by measuring the outputs of each sensor with both LEDs off. Next, it turns one LED on with a low intensity, and capture the transmitted light data from the sensor directly across from it. Next, that same LED is turned up to the higher intensity, and side-scattered light data will be read from the other sensor, positioned at a 90 degree angle. This is then repeated for the other LED source. This process repeats five times for each of these four lights measurements followed by a final ambient light measurement. These six collected values are then each averaged and stored until they can be sent to the connected data logger via I^2C.

The turbidity sensor can be connected to either the ALog or Mayfly data loggers. Every ten minutes, the data logger tells turns the turbidity sensor on, and requests the current turbidity readings using an I2C interface. The data logger these collected frequency values to a corresponding NTU value.
The algorithm that does this was developed by measuring a selection of liquid samples with known turbidities. This NTU value, along with the six frequency measurements, are saved to an SD card on the

data logger. After this, the data logger turns the turbidity sensor off and waits a certain amount of time for the next data point.

One of the more complicated aspects of the physical design was the mounting of the LEDs and IR light to frequency sensors. We used surface mount light sensors and LEDs, which made it difficult to mount on the four sides of the acrylic tube. In addition, these components had to be aligned in order for the turbidity sensor to work properly. We developed a 3D printed shell that would go around the acrylic tube. This shell consisted of two parts, which allowed the sensors and LED to be mounted on the back side of the part. This left a flat surface facing the side of the tube. This part can hold the LEDs and IR sensors in place, is very low cost, and is easily reproducible given the SolidWorks software model.

To mount the LEDs and IR sensors electrically, we used a small breakout board. It can take the surface mounted LEDs and IR sensors and convert them to 0.1” pins, which we connected to wires that go to the microcontroller. Using two 3D printed parts that attached together around the tube, with each part housing an LED and an IR sensor, turned out to be very easy to mount in place. These 3D printed parts were attached to the tube using an adhesive glue. In addition, that same adhesive glue held the LEDs and IR sensors inset upon the 3D part. To prevent possible short circuits, we put tape over the exposed parts of the sensors and the wires.

The waterproof enclosure and cable assemblies are some of the most crucial elements. The IP68 enclosure is from PolyCase, and is made of cast aluminum. This enclosure will not deteriorate when submerged in water. The square water inlet tube is made of acrylic for the same reason. In addition, it is clear, and will not deflect the light from the LED as to distort any data being measured. Another major benefit of this enclosure is the integrated rubber gasket, which creates a seal around the lid. In order to guarantee that the sensor enclosure sunk, a small square piece of granite was attached to the inside, underneath the circuit board. The plugs connecting the sensor to the cable are IP68 rated, and are designed for continuous submersion. The female panel mount plugs are not permanent when attached, so they can be moved from one enclosure to another without issue. This was ideal for testing and calibration. For power and communications, we used an insulated 6-conductor PVC cable. The insulation also will not degrade in water. The 6 conductors will be for power (2), communications (2), controls(1), and one extra left unconnected. Each cable has a male plug attached at one end, which then connects to its female pair on the box. All joints around the plugs and inlet tube are sealed with marine-grade silicone caulk. Because the aluminum boxes were expensive, several plastic boxes from the ECE Depot were used for initial testing and prototyping. This way, we could make several changes to the design or box construction without exhausting our budget. These boxes were made waterproof by applying silicone caulk around cuts made in the box.

This project met or exceeded a significant amount of its goals. Most significantly, we kept cost well below \$100 per sensor. For one sensor, the price is just under \$89. However, if the client or another customer would like to build as many as 100 at a time, the price would become \$54 per sensor. Any construction methods or other cost limiting strategies available to the customer would drive prices down even further. Next, the sensor had to be deployable for up to 3 months running on solar power alone. Because of this, our target current consumption was 50 mA. Our sensor, while running at maximum power during a measurement routine, consumes 6.6 mA. The client can integrate our sensor into their system without any concerns about power. While we aren’t able to prove the sensor is waterproof for up to three months, all components are rated for such deployments. We are confident the sensor will maintain its integrity at depths of up to 10 feet for three months. All components also have specified operating temperature ratings that easily span our specified temperature range of 0-30°C. Given the structural integrity of the box and low power consumption, it is possible the sensor could be deployed for longer than three months. Also note that the sensor was calibrated with known turbidity solutions. As a result, we have met our accuracy requirements. Further testing and calibration would only increase the accuracy of the sensor.

Overall, this project met nearly all of its goals. There were only two major goals that were not met. First, one of the major specifications was to build a sensor that was field deployable for up to three months. Given the timeline of the project and the school semester, testing this was simply not possible. That being said, all of our enclosure and cable components are rated to be waterproof for extended submersion. However, these ratings are not absolute and do not guarantee to be waterproof for days or months. Secondly, we did not use lab grade turbidity standards when calibrating our sensor. Again, this is because of the limited time and resources available to us during the semester. In addition, the inclement weather that extended into April severely hindered our ability to perform real world tests in various bodies of water nearby. We performed an adequate number of tests to obtain a curve of best fit over the range of turbidity levels that our sensor will subjected to.

Future Improvements

Since we could not fully test our turbidity sensor in a submerged environment until it was completely assembled, our circuit design had to be developed using a number of assumptions. One of these was the assumption that when the sensor was out of water, the amount of light transmitted from an LED to the light sensor directly across from it would be be the same or greater than the amount transmitted through clean deionized water or any more turbid sample. Following this assumption, the LED brightness for this measurement was scaled with enough headroom to account for additional ambient light leakage, like sunlight, plus a little more to err on the side of caution. However, when we began to calibrate our sensor, the transmission measurements for low to medium turbidities (DI water and 0.4-40 NTU samples) were significantly higher than what we measured when the sensor was out of water, with the sensors saturating at their maximum outputs. This did not prevent us for calibrating over the full range of turbidities, since the side-scatter measurements were reliable in this range, but if the LED intensities were scaled down, the transmission measurements could be used in combination with the side-scatter readings to develop a more robust and accurate NTU conversion algorithm.

Besides just adjusting the resistance in series with the LEDs to control their brightness, which is how our sensor functions, a possible re-design could use a constant current driver controlled by pulse width modulation (PWM) to supply the LEDs. This would come with some increase in the cost and complexity of the circuit, but would allow the LED intensities to be adjusted by simply re-programming the ATmega chip. Since our clients ultimately want to produce and sell these sensors, this would be particularly useful for adjusting for any variations between units in the manufacturing process. This solution would also likely require a custom PCB to fit within the enclosure. This would not be a problem for mass production, but due to our available timeline for our project, we were not able to explore this route for our design.

Although the 3D printed part was able to fit around the acrylic tube well and held the LEDs and IR sensors, some improvements could be made. The easiest improvement would be to print the part in black filament instead of the white filament that we used which needed to be spray painted black to keep the light reflections to a minimum. Simply buying the same filament that we used in black would be ideal. Additionally, the hole made for the acrylic tube would be moved slightly so that there would be more room to fit the 3D printed part around the tube. This would be especially helpful if the protoboard was made into a PCB, presumably making it smaller. A better adhesive could also be used for longer term stability for holding the two 3D printed parts together as well as holding the sensors and LEDs to said parts. These changes would improve the reproducibility of the sensor, however they would not have a direct impact on the measurements.

Although the housing, cabling and waterproofing aspects of the design are adequate for what we had specified, some improvements could be made if future sensors were to be created. First and foremost, additional long term testing could be done on the final enclosure over long periods of time to say definitively how long the product can be submerged. Given the limited timeline, there was not sufficient time for long term testing. Given the short term waterproof testing, it is clear that the enclosure is waterproof. Additionally, all components used in the design are individually rated to meet the specifications of the project. Another thing to consider is the cables and plugs used. Trying to keep the sensor cost low, cheaper plugs and cables were used. If more money were available for each unit ,it could be beneficial to purchase plugs that have a tighter locking mechanism. This would be helpful if the sensor were placed in turbulent waters with risk of the plug becoming detached. As far as the cable used on the project, if more money were allotted it would also be beneficial to use a rubber cable vs the semi-rigid PVC cable that was used simply because the rubber cable is less susceptible to wear from harsher environments. If the sensor were to be reproduced, another thing that could be done differently is to manufacture some mass for sinking the sensor. The piece of granite is not ideal as it is not manufactured to maximize space within the sensor, but was a cheap alternative for the sensor at hand. Again, if the budget allowed, the process could have been improved if the proper tools were purchased. Specifically, for cutting the opening in the enclosure to insert the acrylic tube.

Team Organization

Name	**Role**
Amanda Hayden	Team Lead, Housing
Corbin Buechler	LED and Sensor Code / Hardware
Aaron Nightingale	Data Logger Interface Code
Sean Rogers	Housing, Cabling, Waterproofing
Erik Sundberg	Hardware Layout, Mounting, 3D Modeling
Jacob Wolf	Optics / Hardware Integration

Budget

Our clients had an overall budget of $4,500 for this project, of which they allocated $1,000 for turbidity sensor development. However, they requested that sensor design and production be as inexpensive as possible, allowing for ongoing design and production. Below is a table of the overall costs associated with the development of the sensor over the semester. It is important to note that the working turbidity meter was delivered to the clients both under budget for development as well as under the target "per sensor" price of $100.

Subsystem	**Amount spent**
Waterproof Housing and Cabling	$125
Test Equipment	$300
Circuit Design Components	$200
Integrated Circuits and Microprocessors	$125
Total	**$750**

Updated Individual Effort Statement - Corbin Buechler

Since I came to this project without much prior knowledge about water quality and how it is measured, the first work I did was to research the common methods used to measure turbidity and to get an idea of how commercially available turbidimeters are built, hoping to gain some insight about how we could approach the physical form of our turbidity sensor. Since our clients specifically mentioned an interest in measuring light transmission in addition to the side- or back-scattering that is typically measured by commercial turbidimeters, we would need our design to allow for a light source and a sensor to be mounted facing each other with space for the sample in between. After sketching some ideas, I came up with and shared a drawing of a clear square tube passing through a standard rectangular enclosure, which is very close to the form that we ended up moving forward with.

After we developed an overall plan for our project, I volunteered to work on the LED and light sensor circuit and code. I began by identifying LEDs of the appropriate wavelength and sensors that would be easy to integrate without a lot of additional parts. After that, a large portion of my time was spent working on Arduino code that will let the ATmega328P accurately read the highest possible frequency from the light-to-frequency converters. Since the ATmega's interrupts could only read up to about 75 kHz, while the sensors could output up to near 600 kHz, I added a pair binary counters between the sensors and the microcontroller to effectively divide the sensor's output frequency by 8. Since the ATmega was only receiving every 8th pulse, this introduced an additional potential for error, which required me to reconfigure the sensing routine software to extend the sampling time when lower frequency signals were being received.

To maximize the useful range of the light sensors, I wanted to address the fact that the relatively low light produced by side-scattering (less than percent of the maximum transmission intensity) would only use a small portion of the light sensors' frequency range, which could limit the accuracy of our side-scattering measurements. My solution was a scheme that allows each of our LEDs to be illuminated at two levels of intensity. This circuit uses two current limiting resistors in series with each LED. One of these resistors is connected on either end to the drain and source of a MOSFET, with the gate and source each connected to a pin on the microcontroller. The drain is connected in series with the other resistor, which connects on its other side to the LED's cathode. In this configuration, the source pin controls whether the LED is on or off (turning it on when the pin is LOW and vice versa), and the gate pin controls whether a low or high current is allowed to flow through the LED. When the gate pin is LOW, the MOSFET is in the cutoff region and the two resistors in series allow the low current to flow. If the gate is set HIGH, the MOSFET turns on and shorts the second resistor, raising the LED current to produce a brighter light source for measuring side-scattering light.

Since our final design is powered by 3.3V supplied by the data logger, the ATmega328P to be clocked at or below 10 MHz. Our initial code development was being done on an Arduino Uno, which uses the same microcontroller, but is powered with 5V and clocked at 16 MHz. Since the Arduino IDE does not have native support for programming a standalone ATmega chip with an external 8 MHz crystal clock source, I needed to do some research to determine how we could do this.

Besides my main task of developing this code and circuitry, I also planned the layout of our circuit on perfboard and soldered the components together. During the calibration process, I took the frequency data that we collected while measuring samples of know turbidity and produced a set of fit lines to develop a calibration algorithm for determining the turbidity, in NTU, of an unknown sample.

Updated Individual Effort Statement - Amanda Hayden

When I initially came across this project, I was immediately intrigued. I did a quick Google search of ‘turbidity’ and found that it is a measure of cloudiness or haziness of a fluid due to suspended particles. I also knew I needed to further my understanding of the topic. I explored current sensor models and how the sensors worked, and came upon NTU, or Nephelometric Turbidity Units, and the mechanism that measures the turbidity. This is done by measuring the intensity of side- and back-scattered light in a sample of water, taking the frequency of the light and converting it to NTU based on the calibration data saved in the sensor memory. More research led to my understanding of common light sources and light sensors used in the device, which included LED’s, lasers, and light-to-frequency sensors, as well as an understanding of the open source code languages and programs used in these devices. The clients granted access to their Alog and Mayfly code for study and education purposes so I spent a good amount of time parsing through the code to understand the process.

After our initial meeting with the clients and advisors, the group determined that I would be the project lead. I checked in with each team member, submitted assignments, and communicated with the clients and advisors, as well as participating in the housing and waterproofing sub-team with Jake and Sean. Our initial picture of the final product came from Corbin, who had designed for a square tube for water flow to go through the instrument and for circuitry, the sensors, and light sources to surround it. All was going well until I suffered a concussion at the beginning of February, which sidelined me for a solid two weeks and only allowed for limited academic activity for some time after that. I am very thankful that my group was understanding of my situation, and that my limited contribution during the month of February to the housing and waterproofing team was actually useful.

Once my head was fully healed, I began by helping with selecting and creating the turbidity standard solutions for calibration. To save some money, we decided to order Formazin at 4000 NTU as our standard solution, and to practice our chemistry skills of dilution. Our client, Andy Wickert, has a lab in Tate Hall and he granted us access to use the deionized water for our calibrations. We requested a set of graduated cylinders, pipettes, a set of beakers, and a set of Nalgene bottles for storing our solutions. One of the cylinders was a two-milliliter syringe, which allowed for extremely precise measurements. Upon recommendation by Professor Wickert, our calibration solutions are spaced out on a logarithmic scale: 4000, 400, 40, 4, 0.4 NTU. Deionized water is 0 NTU, by definition.

The entire group was present for calibration and testing, and we all contributed to the poster, the final design paper, and the final presentation.

Updated Individual Effort Statement - Aaron Nightingale

After some initial brainstorming for the turbidity sensor project, I chose the role of developing software that allowed our sensor to interface with various data loggers. In the first few weeks of this project, I was focused on choosing a communications protocol. This was an important decision, because it influenced the circuit design and programming that was done later. To make a good decision, I skimmed through the ALog and MayFly data logger specifications to see what interfaces they supported. We had already decided that we would use an Arduino as the “brains” of our turbidity sensor. After all of this research, I decided that I2C would be the best communication protocol to interface our sensor with these data loggers. I2C is supported by Arduino software and has great documentation. In addition, the MayFly and ALog data loggers already have libraries that interface with other sensors over I2C. Using I2C as the communications interface significantly simplified the development of our turbidity sensor.

The next step in my software development plan was to create a working prototype of two devices communicating using the I2C protocol. To do this, I did a lot of research to learn more about how I2C worked. I also looked for example code from various online sources. By the middle of February, I had a firm understanding of I2C, and began writing two programs for two Arduino microcontrollers. The first program included code for an I2C “slave” device. In our case, the slave device is our turbidity sensor. The second program included code for an I2C “master”. In this case, the master is one of the previously mentioned data loggers. In the I2C master program, I included functions that the data logger would use, such as a function that would check if the sensor is present, a function that could send a command to the sensor, and a function that would request a data packet from the sensor. In the I2C slave program, I included functions that the turbidity sensor would use. This includes handling different commands from the master and sending a data packet with the current turbidity readings.

By the end of February, I had begun testing these I2C master / slave programs on two Arduino boards. Shortly thereafter, I had a working prototype of two Arduinos communicating with each other using I2C in a master / slave format. This prototype used a communication method that was almost identical to the data logger / turbidity sensor interface. This made it very easy to adapt this code to run on the ALog and Mayfly data loggers.

In March, the next step was to become familiar with the data logger source code. In the first three weeks of this month, I carefully examined the source code and learned how these data loggers operated. The data logger software was much more complex than I anticipated. The things that I learned in these three weeks allowed us to successfully interface our turbidity sensor with the data loggers later.

In accordance with the project timeline, the prototype code had to be adapted to run on the Mayfly and ALog data loggers in April. I did this by creating a new C++ class for our turbidity sensor. This class includes various different functions that the data logger uses to setup, get data, and send commands to the turbidity sensor. Because we already had a prototype of two microcontrollers communicating over I2C, adapting this code to run on the data loggers was a fairly straightforward process. After the data logger software was complete, I merged my I2C slave program with Corbin’s data acquisition program. I believe that developing these two programs separately was a wise decision, and allowed us to avoid some potential problems earlier in the project. These two programs were merged in late April, and once it was finished, we had a working turbidity sensor that was compatible with the ALog and MayFly data loggers.

Updated Individual Effort Statement - Sean Rogers

The first step for me was to do some basic research and find out exactly what the group was being asked to do. After meeting the group and advisors, we went home to brainstorm design ideas. Corbin brought a very cool design to our first team meeting, that we all immediately agreed on. We broke down what specific areas of the project needed handling, which would be enclosure and cabling, sensor code, data logger interface code, circuit design, and hardware integration. I chose the waterproof housing, cabling, and hardware integration. I was also in charge of most of the parts ordering.

My research of waterproofing led me to Ingress Protection (IP) ratings. They are an international standard used to define levels of sealing effectiveness of electrical enclosures against dirt and moisture intrusion. The rating consists of a two-digit number. Our enclosure, cabling, and outlets would need to be IP68 rated. This means it is dust tight and protected against complete, continuous submersion in water.

My first research step was to find an enclosure and plugs that would suit our needs. I discovered PolyCase, a company that makes several different models of waterproof enclosures out of ABS or aluminum. They had a box that was the perfect size and affordable for our budget. We decided to buy several similar sized boxes from the Depot for testing and design purposes.Switchcraft has a line of plastic, waterproof plugs that were perfect for the project. We needed a male cable plug and a female panel mount plug. Each plug was specifically IP68 rated and did not require special installation. The next item was the actual cable running from the sensor to the central hub. We decided to use six-conductor cable – two for power, two for communications, and two extra. We agreed on a fifteen-foot length, but weren't sure how the cable should be insulated. After some research, I discovered PVC would be okay. We found several cable options on DigiKey that were in budget. The first purchase had 28 AWG conductors, however, they were too small to properly solder. A few days later, Switchcraft launched a new line of waterproof plugs that were slightly bigger, had 24 AWG conductors, and were more sturdy. Finally, I had to consider the tube running through our enclosure. I found some square acrylic tube I could cut to length. It would not degrade or shrink in water. I decided we would use marine-grade silicone caulk to seal the area around the tube.

Jake and I tested the enclosures to make sure they were waterproof. We both constructed a box with a tube in place and no plug, and placed them in buckets over break. There were no leaks, therefore the tube and waterproofing was sound. I cut holes for the tube in the aluminum box using a Dremel tool. After securing the tube and sealing the edges, I tested it for leaks. It was completely dry after two days. Jake then drilled a hole for the plug. Again, we tested it and found no leaks. Next, I worked with Jake and Amanda to create our turbidity standard solutions. I worked with Bobby to order one liter of 4000 NTU solution that we would dilute to 400, 40, 4, and 0.4 NTU for testing. We did this by working in the lab with proper measuring equipment. We used deionized water to dilute the solutions to the necessary volume and stored them in nalgene bottles to be used when testing.

I began working with Erik to design and test a 3D part that would be a mount for our LEDs and sensors, which would sit on the tube. Erik printed an initial design that didn't quite fit. With some experimentation, we found a design that could easily be glued to the tube.

Then, working with Bobby, I soldered the LED's and sensors onto breakout boards. With his help and equipment, I completed the task without issue. As a group, we met in the design lab to attach the sensors to the 3D mount and attach those to the tube. I used hot glue to secure all the in place around the tube and make sure there wouldn't be any shifting during use. Corbin then built the circuit and placed it in the box. Working with Jake, we soldered wires in place between the plug and circuit board. Our design was complete. The next phase was testing using the solutions I helped create earlier. This testing went off without issue. Corbin used the data to construct best fit curves that would be the basis of our sensors code.

The final steps will be creating the project documentation for the client, designing our presentation poster, and verifying all code works with the data loggers. Most of this will be done as a group.

Updated Individual Effort Statement - Erik Sundberg

In the initial assignment of roles, I, as an electrical engineering major with little programming experience and a focus in power, thought it would be best to take on a hardware role within the project. As most of our team are electrical engineers with a power focus, I was happy to take on the more detailed role of interfacing the electrical hardware with the physical hardware. Initially, I helped design and provide feedback to the various preliminary designs of the sensor. Specifically, the layout of the box with respect to where the water would enter the acrylic tube, how the water would then be read from inside the box, and a general layout of the turbidity sensor's insides.

After we had some general layout of the whole project, I wanted to have a visual model that we would be able to use for prototyping. Since the University offers Solidworks for free, I installed it and started learning how to navigate and create what we needed. After a few hours, I was able to make a simple box that looked vaguely like the box we had ordered. After a few more hours, I had the correct dimensions and was able to have both the tube inside the box and a top.

Once we had received parts, I helped and offered feedback to those working on the specific electrical hardware. This was, in one form, deciding the best way to mount the LEDs and IR sensors that would be surrounding the acrylic tube and taking measurements of the water. Initially, we decided on a four PCBs or perfboards solution where they would interlock at 90 degree angles around the tube. Though this was not ideal, we could not see a better alternative. In addition to the four PCBs or perfboards, we thought it could be reasonable to use a custom PCB for the main circuit. After exploring this option further by looking at the software Eagle, which only one of our group members had ever had exposure to, I thought it would be best to use four breakout boards for the surface mount components that our advisors wanted us to use. After trying one type of board somewhat successfully, I found a much better breakout board that was smaller and didn't have additional, unnecessary terminals. Using the model I had created, I found the largest size perfboard that we would use for the main processor and circuit.

To hold these small boards in place, it was initially proposed we use liquid rubber. However, because this would require manual alignment with a large margin of error, I spent some time researching a better alternative. I eventually came up with the idea of using a 3D printed part that would house the breakout boards in a very specific orientation for much better accuracy and mounting ability. I went about creating something in Solidworks as it is the preferred method for printing 3D parts according to our supervisors. Because there was no datasheet that detailed the exact size of the small breakout boards, I estimated the size and created a model to put inside my existing model of the box and tube combination. After receiving the part, I was able to make an exact model of both the board and the 3D part that I designed. After conferring with my teammates, I refined my design from being one part that would slide around the tube, to being two parts that would join together around the tube. After printing this design, the other breakout board was found and chosen so I had to further redesign the 3D printed part. It was at this point that a total redesign was made. Instead of having the sensors mounted to the inside of the 3D printed part, I thought it would be best for easy mounting to have them attached on the back. Once printed, this design was used as it also provided a better fit against the acrylic tube.

After designing and refining the 3D printed part, I helped with the overall completion of the project. I helped with the calibration of the sensor to the known turbidity solutions and with the documentation of our project. I also helped by troubleshooting the few problems we had with the full sensor which included determining the issue of why some readings were off which was due to some resistors not being correctly sized for measuring very low turbidities. I helped with the poster and the final presentation as well as getting all the documentation in order for our clients to use when they build their own.

Updated Individual Effort Statement - Jacob Wolf

At the start of the project, I did a significant amount of research on existing turbidity sensor designs. I found that there are some commercial units available as well as small units that don't contain microprocessors. I studied the circuit layout for the different types of sensors and researched different ways to detect transmission and scattering. I found that although there are different ways of implementing a turbidity sensor, and many different circuit layouts, they all essentially do the exact same function to perform a common task. The research gave me an understanding of how the overall design of the turbidity sensor should be and what we types of parts we should expect to have to make our sensor operational. I then focused more on learning how the scattering and forward transmission in infrared emitters and detectors worked and how we can use the light detection to actually measure turbidity.

The next task I focused on was determining how we could make the unit waterproof. Since we had decided on an approximate enclosure shape and type, I worked with Sean to determine an appropriate size. We decided on the size of the enclosure by looking at the approximate circuit size and layout with the sensor's viewing window and cable termination inside the box. The goal was to then find a box of the approximate size we wanted that contained a watertight seal on the cover. Knowing that we will have to cut a hole in the enclosure for the acrylic tube, which will be the 'viewing window' for the sensor, we want everything else to be as watertight as possible. Sean and I found and decided to use an aluminum box that was the appropriate size and had a watertight seal (IP68 rated) for the lid. From there I worked again with Sean to determine the best method for waterproofing the enclosure. We decided to use marine grade silicone around any opening that we cut into the box. Additionally, after discussing with the other team members about the number of conductors we shall require in the cable to the data logger (for power and ground, communication, etc), I did some research to find cables and plugs that are waterproof. Additionally, I wanted to find a plug that we could disconnect from the unit but yet was secure enough that it would not come disconnected if the unit were put into fast moving waters and were jolted around. After acquiring the plugs and cables, I drilled the appropriate holes in the boxes to assemble the plugs and then proceeded to solder the cables to the plugs as well as conductors on the inside of the plug to connect to the circuit board. For additional waterproofing, I used silicone to install the watertight plug. I then painted the acrylic viewing window with a black epoxy paint to reduce light interference with our measurements. I then placed the acrylic tube in the enclosure and used the silicone to seal around all surfaces both inside and outside the enclosure. I cut a piece of granite to fit within the enclosure while still allowing room for the circuitry and wires. The granite had to be heavy enough to cause the sensor unit to sink but no so heavy that it would strain the cable. To test the waterproofing of the unit I then cut a hole in the ice on a pond and submerged the unit for several days at a depth of 10 feet, which is the max depth specified for the unit. The inside of the unit was dry when removed from the water, indicating that it was indeed waterproof.

The next action I took with Sean and Amanda was creating the testing liquids. Using the 4000 NTU solution that we ordered form HACH, we made serial dilutions following a log scale to the values of 400, 40, 4, and 0.4 NTU. These would be the solutions we use to take data, plot the data points, and ultimately create our line of best fit curves for converting the output frequency from our sensors to an NTU value. We then spent an afternoon as a group running the testing sequence with the various liquids we created to get a set of data points for our sensor calibration.

The last stages of the project include putting together all the documentation on the parts and methods used so that our clients can re-create the sensor. In addition, we put together our final design poster in preparation for the design show.