Farming in the Climate Change Era

Draft Project Document - Group One

Adam Farhat, Isaac Sutton, Isaac Guzman, Cameron Lein

October 13, 2022

Project Partner: Gerrad Jones

Contents

1 Overview

1.1 Executive Summary

The Quechan Tribe participates in agriculture of the Imperial Valley of Southern California, the largest year-round irrigated area in the United States. As the effects of climate change increase in intensity, new solutions for the preservation of precious water are coming into demand. Fields are manually irrigated with on-site, meaning a person must go out and close water gates manually. To partially automate this process, a field water flow monitor has been commissioned by Professor Gerrad Jones of the OSU College of Agriculture.

The design will accomplish the following goals:

- Develop a deployable field system capable of recording instantaneous water flow measurements
- Notify managers when a certain volume of water has passed into a field or when fields will exceed their allotment of water
- Provide a user interface that assists in analysis over the course of the season or year
- Utilize a SORACOM SIM card to connect to a 4G network and transmit all information in real time collected by the system.
- Send information to allow irrigation managers to track real time water flow rate through irrigation gates as well as an alert email when water allotment has exceeded the set amount.

The project is currently in the implementation phase. Every subsection has been researched, a full bill of materials has been put together, and all blocks have been prototyped. As the blocks come closer to meeting all of their interface requirements, the system will come together as a whole soon after.

1.2 Team Contacts and Protocols

Adam Farhat farhata@oregonstate.edu

Isaac Guzman guzmange@oregonstate.edu

Isaac Sutton suttoni@oregonstate.edu

Cameron Lein leinca@oregonstate.edu

1.2.1 Table I: Primary Project Roles

Adam	Isaac G	Isaac S	Cameron
Documentation PC Front end/Back end 4-5G Network Communication	Documentation PCB Design Enclosure	Documentation Power Supply & Solar Depth Sensing	Documentation Microcontroller / Firmware Water Flow Sensor

1.2.2 Table II: Team Protocols

Topic	Protocol	Standard
Communication	The primary method of communication will be through our phone messaging system. Secondary will be done through Discord calls prearranged through our primary messaging system.	All team members are expected to be ready at any designated meeting time. Should the meeting be in person, all group members are expected to come prepared and bring any necessary items required for the project.
Time Management	Team will use the project timeline we drafted together to record all major project deliverables. Any newly discovered requirements will be added to the timeline.	All team members are expected to deliver any project items by the time indicated on the timeline. The group will also discuss smaller task goals through our primary/secondary forms of communication.
Documentation	Documentation will be done using the program Overleaf and the LaTeX language.	All team members are expected to participate in documentation efforts.
Universal Standards	Team members are expected to complete any task without breaking any universal standard.	All team members are expected to follow the project universal standards. Each member is expected to heavily test any project item before addition to the main project.

1.3 Gap Analysis

The project exists to aid in the monitoring of water flow for irrigation systems in agriculture. Currently, the cost of commercial irrigation monitoring systems isn't feasible for small agricultural operations. Instead, these farms require employees to monitor and control the irrigation gates 24 hours a day. This leads to wasted water and inefficient irrigation of fields. The Quechan Indian Tribe in the Imperial Valley of Southern California relies on agriculture as its primary source of revenue. New regulations have required the collection of water usage data which is unobtainable when irrigation gates are manually operated. This requires the design of an affordable system to collect information about water flow, provide remote access to the collected data, and send alerts when gates should be adjusted. It is assumed that the system will need to be self-contained with its own source of power because there is no guarantee that power will be available along crucial sections of the irrigation canals. Another assumption is that the system should be able to withstand dust, heat, and water due to its long-term, outdoor application and geographical location.

1.4 Timeline/Proposed Timeline

1.4.1 Table III: Timeline

Timeline				
Objective	Start Week	End Week	Description	Assigned
PCB Design	11	16	Design and print PCB	IG
4-5G Network	11	18	Develop Communication for the prototype	IS, AF
Enclosure	16	20	Build enclosure	CL, IG
Interface (Front End)	5	20	Develop and design front end interface	AF
Interface (Back End)	10	20	Develop and design back end interface	AF
Flow Sensor & Calibration	11	20	Build and configure sensors	CL
Power Supply & Solar	11	20	Develop a 65% efficient power supply	IG, IS
Microcontroller / Firmware	11	20	Design and program a micro controller circuit	CL

All Highlighted text is critical to the project and must be completed to not delay the project

1.5 References and File Links

G. Jones, "EECS Project Portal," *Farming in the Climate Change Era.* [Online]. Available: https://eecs.oregonstate.edu/capstone/submission/pages/viewSingleProject.php?id=TdUqfql0WGC8TdRT

1.6 Revision Table

1.6.1 Revision

Date	Revision made
10/13/2022	ALL: Initial Document Creation
10/13/2022	Adam: Initial completion of section 1.2: Team Contacts and Protocols.
10/13/2022	Isaac G: Completion of timeline for section 1.4: Timeline/Proposed Timeline.
10/13/2022	Cameron: Initial completion of sections 1.1: Executive Summary and 1.5: References and File Links.
10/13/2022	Isaac S: Initial completion of section 1.3: Gap Analysis.
11/3/2022	Cameron: Added to section 1.1: Executive Summary as per feedback.
11/3/2022	Isaac G: Added IEEE headers to tables.
11/17/2022	Adam F: Added critical path highlighting and added additional information to section 1.1. Updated IEEE formating for the tables.
3/12/2022	Cameron: Updated information to match winter term changes in plans.

2 Impacts and Risks

2.1 Design Impact Statement

2.1.1 Introduction

The purpose of this document is to provide an unbiased and honest evaluation of the positive and negative impacts our project could cause. We believe that it is important to acknowledge that biasing can occur even if the team tries to avoid it.

This impact assessment will analyse any concerns that have been recognized for the project. This project is being developed for the Quechan (Kwatsáan) Tribe which reside in the Fort Yuma reservation in Yuma, Arizona. The project will be a flow rate and water usage detection system which will measure and record water usage and the instantaneous flow of the water brought in from the Colorado River. The document will assess any concerns as well as benefits the project brings to the table. This flow rate and water usage detection system will impact multiple facets of the tribe we are helping, which include health, safety and the welfare of the tribe. We also go over any cultural, social, and environmental impact that the system will have with not only the people, but the area it will inhabit. Lastly there are economic factors that have been recognized and addressed to ensure no damage is done to the existing economic ecosystem for the Quechan tribe and the other benefactors of the Colorado River.

2.1.2 Public Health, Safety, and Welfare Impacts

The Quechan Tribe is working in accordance with The Metropolitan Water District of Southern California to preserve usage from the Colorado River, in order to enable Southern Californian cities to have more drinking water. Metropolitan pays Quechan farmers to not grow crops during certain periods of the year, so that the conserved water can boost the record-low water levels in Lake Mead [1]. Typically, in the desert of the Imperial Valley, summer agriculture involves growing low-value and water-demanding crops. In the interest of both parties, the Quechan tribe is given financial incentives to preserve this water, which our project could assist in tracking. Additionally, the Imperial Irrigation District employs more than one hundred zanjeros (irrigation managers), some of whom are Quechan, to control the flow of water throughout nearly 500,000 acres of farmland [2]. They work strenuous jobs, around the clock, and each track water usage across huge swaths of land. An analytic tool, such as the one we are developing, could assist these managers in having a more regular schedule, balanced for sleep and free time. The management of irrigation systems, especially in a time of increased water scarcity, requires a brutally high amount of vigilance and precision. One potential negative impact of our project is that, if not implemented in an intuitive manner, it could introduce even more overhead to this already intensely demanding job by adding another tedious setup and monitoring process on top of the rest of the workload. Ease of use of the system by the operator is therefore essential.

2.1.3 Cultural and Social Impacts

The Quechan Indian Tribe are an aboriginal American Tribe whose reservation borders the states of Arizona, California and Baja California, Mexico[3]. They primarily reside within the Imperial Valley at the tail end of the Colorado river. Because the Tribe resides at the end of the river, the Quechan people face any and all changes that may affect the quality and amount of water retrieved. The Colorado river has a deep importance to the Quechan people as the 2,000 members in the Tribe depend on its waters for agriculture and the economy. The Tribe currently has two irrigation operators who work 24-hour shifts to monitor the water drawn into their fields. Normally, this system would be entirely automated with devices that control the water entering a field automatically. Unfortunately, this system would be too costly for a Tribe of just 2,000 members to be able to feasibly afford. This issue is compounded by the water restrictions plaguing the area requiring the operators to provide flow data to the Bureau of Reclamation.

Our project would serve to remedy some of the addressed issues that the tribe faces. Not only will our project help the Quechan Tribe accurately quantify water use in a given area but also relieve the operators monitoring the entire irrigation system 24/7. The Quechan people have faced many instances of injustice beginning with the formation of the reservation due to the US Army defeating them in the 1850s and establishing the reservation[4]. Just last year the senior design team tasked with solving this issue completed the task but ceased all communication with the Tribe before they received the irrigation system marking another instance of injustice towards the Quechan people. Should we also not deliver our project after completion, we would also be contributing towards the continued injustice towards the Quechan people. We are working to make sure our project helps benefit all 2,000 people within the Tribe by allowing them to not have to manually track water use.

2.1.4 Environmental Impacts

The environmental impacts associated with this project are almost exclusively related to water use. Currently drought is a substantial problem globally, and unfortunately the Imperial Valley of Southern California is no exception. The entirety of the Imperial Valley is listed as being in moderate drought, affecting roughly 180,000 people. 2022 is the 22nd driest year in the past 128 years for the valley [5]. The Quechan Indian tribe of Fort Yuma gets all of its water for irrigation from the Colorado River. Along with the Imperial Irrigation District to the west, the Quechan agricultural area uses enough water from the Colorado River to cover fields with 5 feet of water each year [6]. The monitoring and data collection on water use that is provided by our project will help to reduce the amount of water that is wasted due to over-watering. It will also help to improve efficiency in the distribution of water to different fields. While our project is intended for the Quechan Indian Tribe which obtains its water from the Colorado River, future iterations could be used in agricultural settings that use groundwater. Poor water management and inefficient irrigation in those settings depletes underground water reservoirs. Improving efficiency through water flow monitoring will reduce the impact on these reservoirs and the surrounding ecology reliant upon them [7]. A negative impact of our project is the environmental harm caused during the manufacturing and disposal of electronics. For example, solar panels consist of glass, Cadmium and Lead. Seeing as our system will be submersed in water it could potentially leach toxic chemicals into the water that is being used for watering crops. Lead toxicity negatively impacts the plant germination, growth, and yield [8]. This could harm the local plant and animal life that rely on that water source as well. Often solar panels are disposed of in landfills rather than being recycled, which leaches these chemicals into the ground and potentially into groundwater [9].

2.1.5 Economic Impacts

Agriculture is the tribe's primary economic resource [10]. Growing produce and leasing the thousands of acres of land they own to non-indian/indian farmers. Mentioned prior was the Colorado River which flows through Yuma county. This river alone supports about $1.4 trillion in economic activity not for the Quechan tribe alone but for states like Colorado, Wyoming, Nevada, and California as well. The river provides these states with about 16 million jobs. Any significant drop in the river's water availability can result in even more significant loss of jobs and capital [10]. For scale, if 10% of the river was lost, about $143 million economic activity and 1.6 million jobs would be lost within the year [11].

Our project is designed to help reduce any water waste and ensure that the river and economic balance stays flowing. With the bureau requiring water flow data, it forces the tribe to invest in expensive systems that do more damage financially then they assist in resource control. Designing a more cost effective system can allow the tribe to take part in the effort to conserve water and allow all users of the river to keep their production and crops going. If the project ends up being inaccurate in its volume readings, it could result in the tribe losing money from the bureau and causing more financial damage than the damage they tried to avoid. The tribe stands to gain up to $1.6 million in incentives for tracking their reduced water usage, so maintaining accurate volume readings is crucial [1].

2.1.6 Conclusion

Our project will provide a cost effective solution to an expensive and difficult issue that the Quechan people are currently facing. A professionally made water irrigation device can cost over $10,000 which is much too expensive for the Tribe to be able to afford. We are working to provide the Quechan people with a device that is capable of monitoring and recording instantaneous flow measurements to alleviate irrigation operators from having to manually track this data. Our device will allow irrigation/water operators to remotely track in real time how much water has flowed through an irrigation gate and provide this information though a simple web interface. The device, in combination with our accompanying program, will alert the operators though a phone notification when a certain volume of water has been applied to a field. The device will be simple to operate but also rigid, waterproof, durable, and cost effective.

Although the intended purpose of the project is to improve the efficiency of water use in irrigation for the Quechan people and in turn reduce the environmental impact involved in agricultural water use, we are aware that our project has the potential to impact people and land negatively as well. Unfortunately, no one within group one has any relation to a reservation or Indian Tribe which could negatively impact our deliverable. Because our project is not just building a water irrigation device but building one for a specific customer, we have to keep in mind the Quechan Tribe may live differently than ourselves. Fortunately, our project partner not only is a part of the Quechan Tribe but has also lived in the reservation's area. We make sure to involve him in all deliberations involving parts or design ideas. The negative environmental impacts exist in the production and disposal of electronics, such as the solar panel and rechargeable LiPo batteries. In order to reduce our impact in this regard we intend to source components as ethically as possible while weighing the cost of purchasing them. All components used in the project should be RoHS compliant. Additionally we will encourage the end user of the product to recycle components at the end of their life instead of disposing them in a landfill. Having an inaccurate system is never an objective when designing but it can occur when it's not developed carefully. To ensure that our system is accurate we will prioritize time to calibrate and verify the system throughout development and testing. The goal will be to keep minimal uncertainty with the system's sensors and calculation.

2.2 Risks

2.2.1 Table IV: Risks

Risk Assessment and Action Plans						
Risk ID	Risk Description	Risk Category	Risk probability	Risk impact	Performance indicator	Action Plan
R1	Water enters system	Safety	Medium	High	System Breaks	Test housing before adding electronics.
R2	Application inaccessible	Design	Medium	High	Unable to access software application which monitors system	Thorough testing of software accessibility
R3	Radio system failing to transmit	Design	Medium	High	Data is not updated to current date/time	Verify code, re-calibrate, order new component.
R4	Parts not coming in on time	Organizational	Low	High	Development delays	Watch lead times and order ahead.
R5	Inefficient power supply	Environmental	Low	Mid	System shutdown	Move system into better lighting.
R6	Unstable environment	Environmental	Low	Low	System faces external damage	Enclosure made from durable materials and solar panel shielding
R7	Sensor failure	Design	Medium	High	Incorrect measurements	Thorough testing and software reset implementation
R8	System falls over	Environmental	High	Medium	Sensors not contacting water	Motion sensors for fall detection

2.3 References

[1] "An Urban-Agriculture Partnership with The Quechan Tribe." The Metropolitan Water District of Southern California. https://www.mwdh20.com/media/12072/Quechan%20Fact%20sheet_FINAL%2003.22.22.pdf (accessed Nov. 4, 2022)

[2] M. Anton. "A Way of Life Drying Up." Los Angeles Times. https://www.latimes.com/local/la-me-zanjero14mar14-story.html (accessed Nov. 4, 2022).

[3] "About Us." About Us - Fort Yuma Quechan Indian Tribe. https://www.quechantribe.com/about-us.html. (accessed Nov. 4, 2022)

[4] "Quechan - History and Cultural Relations." Countries and Their Cultures. https://www.everyculture.com/North-America/Quechan-History-and-Cultural-Relations.html. (accessed Nov. 4, 2022)

[5] "Drought Conditions for Imperial County." Drought.gov National Integrated Drought Information System. https://www.drought.gov/states/california/county/imperial (accessed Nov. 4, 2022)

[6] D. Charles. "Meet the California farmers awash in Colorado River water, even in a drought." npr OPB. https://www.npr.org/2022/10/04/1126240060/meet-the-california-farmers-awash-in-colorado-river-water-even-in-a-drought (accessed Nov. 4, 2022)

[7] Water Science School. "Groundwater Decline and Depletion." USGS. https://www.usgs.gov/special-topics/water-science-school/science/groundwater-decline-and-depletion (accessed Nov. 4, 2022)

[8] J Environ Manage, "Lead toxicity in plants: Impacts and remediation." PubMed. https://pubmed.ncbi.nlm.nih.gov/31545179/#:~:text=Pb%20toxicity%20causes%20a%20range,causes%20oxidative%20damages%20to%20plants. (accessed Dec. 2, 2022)

[9] N/A. "Solar Panel Recycling." *EPA*. https://www.epa.gov/hw/solar-panel-recycling#How%20Solar%20Panels%20are%20Recycled (accessed Nov. 4, 2022)

[10] "Economic Impact of the Colorado River" [Online] https://www.nature.org/en-us/about-us/where-we-work/priority-landscapes/colorado-river/economic-importance-of-the-colorado-river/ (accessed Nov. 4, 2022)

[11] "City of Yuma, AZ At A Glance" [Online] https://www.yumaaz.gov/home/showpublisheddocument/686/637315148192930000 (accessed Nov. 4, 2022)

[12] Quechan Tribe, *Fort Yuma Quechan Indian Tribe* [Online]. Available: https://www.quechantribe.com/

[13] L. Rodríguez, "Solar panel orientation: How using East-West structures improves the performance of your project," *Rated Power* [Online]. Available: https://ratedpower.com/blog/solar-panel-orientation/

[14] Fictiv, "Nothing Gets In: Waterproof Enclosure Design 101 (and IP68)" [Online]. Available: https://www.fictiv.com/articles/nothing-gets-in-waterproof-enclosure-design-101-and-ip68

2.4 Revision Table

Date	Revision made
10/13/2022	ALL: Initial Document Creation
11/3/2022	ALL: Completion of Section 2.2 and 2.3
11/17/2022	Isaac S: Changed wording in the Risks table
3/12/2023	Cameron: Ported to Project Document; Cleaned references.

3 Top-Level Architecture

3.1 Block Diagram

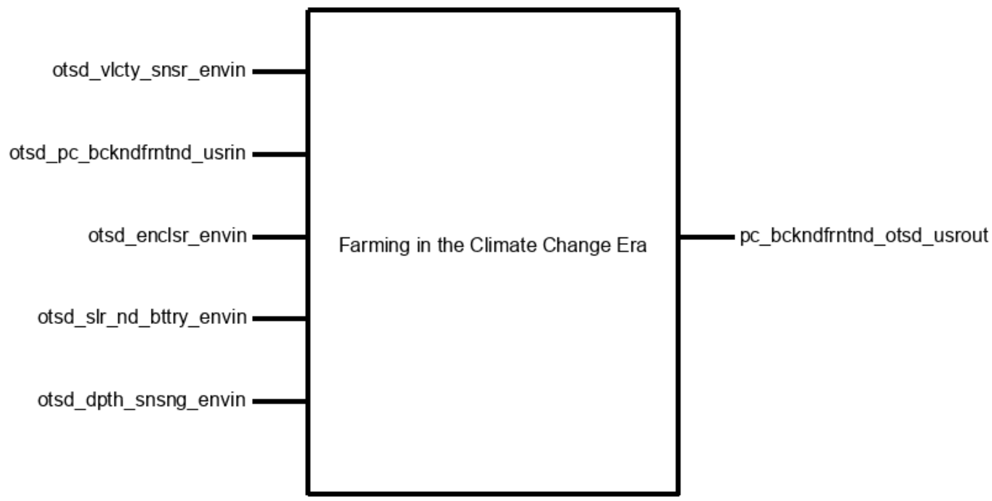


Figure 1: System Black Box Diagram.

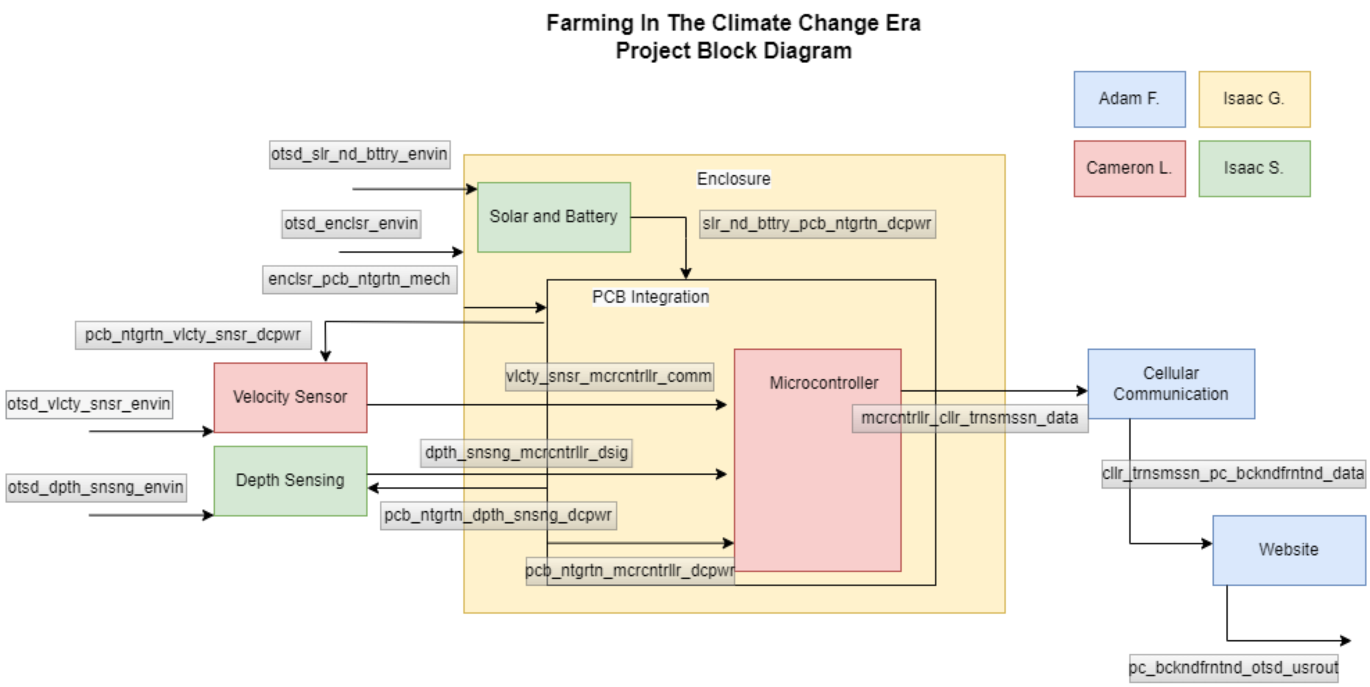


Figure 2: System Block Diagram.

3.2 Block Descriptions

3.2.1 Microcontroller (Cameron)

The microcontroller block operates the two sensors blocks, performs calculations on the fly, and hands them off to the Cellular Communication block for transmission out of the physical system unit. It takes digital waveform readings from the velocity sensor, and performs signal processing on the data set to obtain a water velocity. It also receives readings from the pair of pressure sensors in order to obtain a water channel depth from gauge pressure. These two values are the prepared output of this block.

3.2.2 Velocity Sensor (Cameron)

The velocity sensor is a custom doppler probe, designed for use in low-velocity unidirectional water channels. The block is a functional unit, with its own enclosure and circuit board, using I2C as its output protocol. It can measure water doppler shift from the bottom of a channel by pointing 1 MHz ultrasonic transducers up towards the surface, generating its own environmental output, and performing analog-to-digital conversion on its own derived change in frequency. It interfaces with the Microcontroller block, which is its I2C parent.

3.2.3 Cellular Communication (Adam)

The Cellular Communication device allows for the device to wirelessly connect to cell towers within the device's range and transmit sensor data online to our accompanying website. Connection to the cell towers is done through a SORACOM SIM card plugged into our Arduino MKR NB 1500 which allows us to get 4G connectivity in areas without Wi-fi.

3.2.4 Website (Adam)

The Website allows the user to access system data from anywhere at any time. The website will also allow a user to sign up for the alert system which will email the user once a certain volume of water has exceeded the allotted amount. The website is viewable on a phone and computer which allows the user to be anywhere with internet connectivity and view the system data. The website also hosts a help page that will document how to use the device, information on materials, and how to deploy and activate the system. Lastly, the homepage will display information on our capstone team and chronicle the making of our capstone project.

3.2.5 Solar and Battery (Isaac S.)

The solar and battery block allows the system to remain operational for extended periods of time without user input. A solar panel provides power to the system during sufficient daylight conditions, both powering the microcontroller and charging the Lithium Ion rechargeable battery. The battery powers the system when there isn't enough sunlight.

3.2.6 Depth Sensing (Isaac S.)

The depth-sensing block uses two pressure transducers to measure the depth of the water in the irrigation channel. One pressure transducer is placed at the bottom of the channel to measure the water pressure. The other sensor is placed above water to measure the ambient pressure of the air. The difference between these measurements gives the current water depth. The sensors communicate with the microcontroller through I2C communication.

3.2.7 Enclosure (Isaac G.)

Enclosure must be waterproof to be able to hold electronics inside without them getting wet since it will be in the canals. Gerrad (our project partner) required that the system be at least be able to handle the light rain that Imperial Valley receives every year. The only full resistance the system required was to be completely dust proof. Therefore, we will use IP65 verification to spec out the Enclosure. The reasoning for this is because IP65 rating aligns almost perfectly with what we want to accomplish. Regardless that doesn't mean I won't try to see what limits the seal and enclosure have. Given what we must store in here cautionary measures must be taken to ensure the safety and integrity of the system. Similar to most enclosures, this one will contain and house important objects like electronics. This application specifically will house the SORACOM/Microcontroller module and the battery. A PCB will also be attached to the SORACOM module to allow an easy interfacing solution. This gives me a general idea of what the minimum size of the interior should be, and I can use that to help provide a design early in the project.

3.2.8 PCB Integration (Isaac G.)

This Block is essentially what ties most of the system together. It starts by integrating the solar charging circuit that will receive DC power from the solar panel and then power the system as well as charging the battery. The circuit itself also switches to battery power once it senses lack of input externally power something else. The output of the IC splits into different paths. The first path is optional as the output of the charger IC also connects directly to the micro controller's (Arduino MKR NB 1500) VIN. The second path goes to a female connector that can power and external device. The PCB has a female header pins for the Micro controller to mount on. These headers then break out to the output Rj45 female connectors, which go out to the velocity and depth sensors as well as receive power from the solar charger circuit.

3.3 Interface Definitions

Name	Properties
otsd_vlcty_snsr_envin	• Electromagnetic: 4 Vout Peak-to-Peak • Other: 1000 Hz+- In (Receiver) • Other: 1000 Hz Out (Emitter)
otsd_pc_bckndfrntnd_usrin	• Other: Tab bar at top of website for easy navigation will take the user to any page such as home, data, alerts and help tabs. • Other: Website is viewable through a url on any internet connected device. • Other: User can input their email to sign up for alerts. Email will be stored and error detection will prevent multiple signups under the same email.
otsd_enclsr_envin	• Other: Enclosure can withstand a drop from 6ft • Other: System must be dust proof • Water: IP65 rating (water resistant). Can handle rain or small splashes
otsd_slr_nd_bttry_envin	• Light: Maximum: 450 lux • Light: Minimum: 250 lux • Light: 380 nm - 800nm
otsd_dpth_snsng_envin	• Other: 0 - 25 psi
mcrcntrllr_clllr_trnsmssn_data	• Messages: Floating point water flow rate • Messages: Floating point water depth • Frequency: updated and sent once per five minutes
vlcty_snsr_mcrcntrllr_comm	• Messages: 12 bit (signed) readings • Protocol: I2C • Vmax: 3.3V

Name	Properties
pc_bckndfrntnd_otsd_usrout	• Other: Data can be exported into an excel spreadsheet. • Other: Data is viewable through a data table under the data tab. • Other: Website sends user an email once water use limits have been reached.
pcb_ntgrtn_mcrcntrllr_dcpwr	• Inominal: 100mA • Ipeak: 190mA • Vmax: 7V • Vmin: 5V
pcb_ntgrtn_vlcty_snsr_dcpwr	• Inominal: 4mA • Ipeak: 7mA • Vmax: 5V • Vmin: 3.3V
pcb_ntgrtn_dpth_snsng_dcpwr	• Inominal: 3 mA • Ipeak: 2.8 mA • Vmax: 3.5 V • Vmin: 2 V
clllr_trnsmssn_pc_bckndfrntnd_data	• Datarate: Active connection to LTE broadband tower while main device is running. • Datarate: Data will be transmitted online in 5 minute intervals. • Messages: Can send an HTTP request to the server with an attached JSON file. • Messages: Data can be accessed/stored though both SORARCOM Harvest and the website.
enclsr_pcb_ntgrtn_mech	• Fasteners: 4 standoff screws • Other: dimension restraint of 3.5” x 2.5” x 1.5” • Pulling Force: 20N

Name	**Properties**
slr_nd_bttry_pcb_ntgrtn_dcpwr	• Inominal: 1 A • Ipeak: 1.2 A • Vmax: 5.5 V • Vmin: 4.4 V
dpth_snsng_mcrcntrllr_dsig	• Other: I2C protocol • Vmax: 3.3 V • Vmin: 0 V

3.4 References and File Links

3.4.1 References (IEEE)

3.4.2 File Links

3.5 Revision Table

Date	Revision made
10/13/2022	ALL: Initial Document Creation
3/12/2023	Cameron: Completion of subsections 1-3.
3/14/2023	Isaac S: Updated System diagram and block descriptions 3.2.5 and 3.2.6

4 Block Validations

4.1 Microcontroller

4.1.1 Description

The purpose of this block is to interface with the external sensors of the system, run calculations on readings from them, and make those values available for transmission in the cellular transmission block. Because of how the system's design has changed recently with the team's needs, the cellular transmission block has code running on this same microcontroller, but neither this transmission code nor the successful exportation of data from the microcontroller are considered to be part of the microcontroller block. The microcontroller itself is an Arduino MKR NB 1500. It comes fitted with a SARA-R410M cellular transmission module for use with a 4G/LTE SIM card, supporting the Soracom platform that is being used in the cellular transmission block. It also provides an optional LiPo battery charging circuit for use with a solar panel, but at this point it is unused in favor of a dedicated block for power management throughout the system.

The current system provides two types of sensors for the microcontroller block to interface with: a standalone ultrasonic sensor and a pair of pressure transducers. The ultrasonic sensor is connected to the board's singular Inter-Integrated Circuit (I2C) interface. This is due in to that system's necessary rapid capture of high-frequency waveform data points and its relatively distant proximity from the system introducing the need to avoid analog transmission line effects. The pressure transducers, on the other hand, are operated with the board's Serial Peripheral Interface (SPI), since the specific sensors used in this project use them as their dedicated protocol. If there is time, another sensors block may be implemented for system field maintenance, such as an accelerometer or moisture sensor. These would theoretically trigger external interrupts if the sensors surpass a threshold indicating the system is compromised, and the handler would send out a notice for attention to the unit.

The board periodically takes a "burst" of readings from each of these sensors, and then processes them into meaningful quantities. The array of measurements from the ultrasonic sensor undergo a fast-fourier transform and some additional signal processing in order to attain the frequency of the ultrasonic wave. This is plugged into the doppler effect equation with some other known values to attain the water velocity. The pressure transducers are used to attain absolute water pressure by using the difference of their values to control for the barometric (air) pressure, which causes irregularities in readings with changes in the weather. The absolute water pressure is used with known constants in another linear relationship to obtain depth. These two values are stored in variables, ready for transmission by the cellular block.

4.1.2 Design

This block comprises an Arduino MKR NB 1500 (Manufacturer ID: MKRNB1500WANT) microcontroller, currently priced online at $79.90 directly from Arduino. The microcontroller block is physically connected to three other blocks: the two sensors blocks and the power management system. It also interfaces with the cellular transmission block through its own software. Finally, it uses an antenna, attached by the board's micro UFL connector (coaxial).

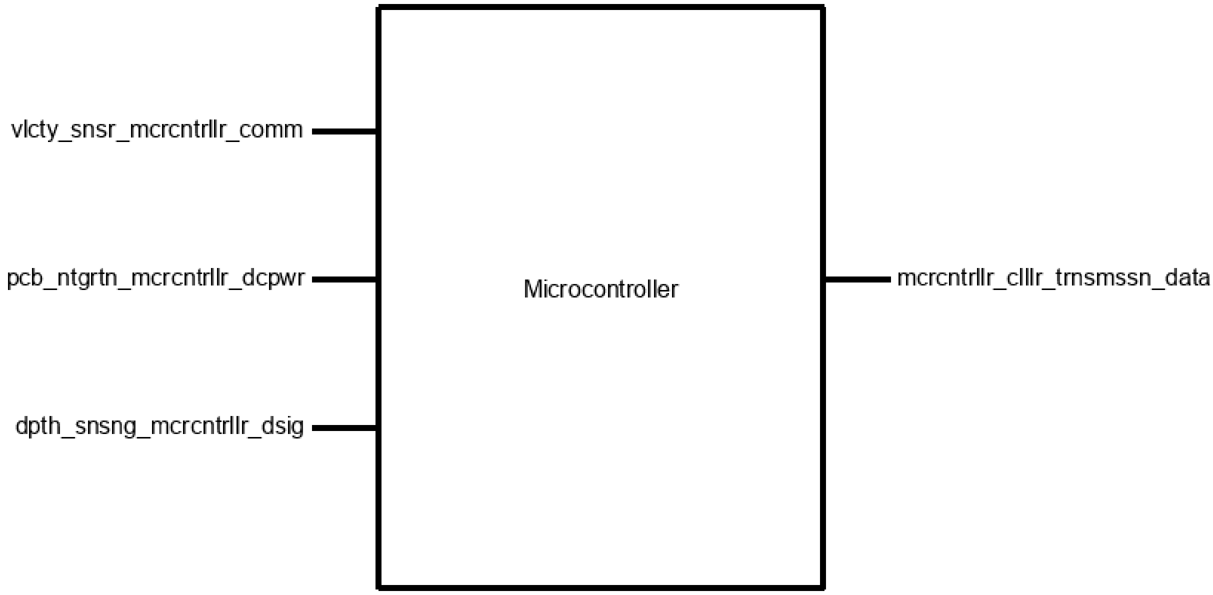


Figure 1: Black Box Diagram.

Figure 1 shows the four interfaces with other blocks with the abstracted microcontroller block.

- pcb_ntgrtn_mcrcntrllr_dcpwr: The 5V power input is received via the Vin pin.
- vlcty_snsr_mcrcntrllr_comm: The ultrasonic sensor connected by I2C uses digital pins 0, 1, 11, and 12 (two analog control pins, SDA, and SCL, respectively).
- dpth_snsng_mcrcntrllr_dsig: The pressure transducers connected by SPI use digital pins 7, 8, 9, and 10 (CS, COPI, SCK, and CIPO, respectively).
- mcrcntrllr_cllr_trnsmssn_data: The output of this block is a pair of variables handed off within the running executable. It is actually exported by the microcontroller in the cellular transmission block.

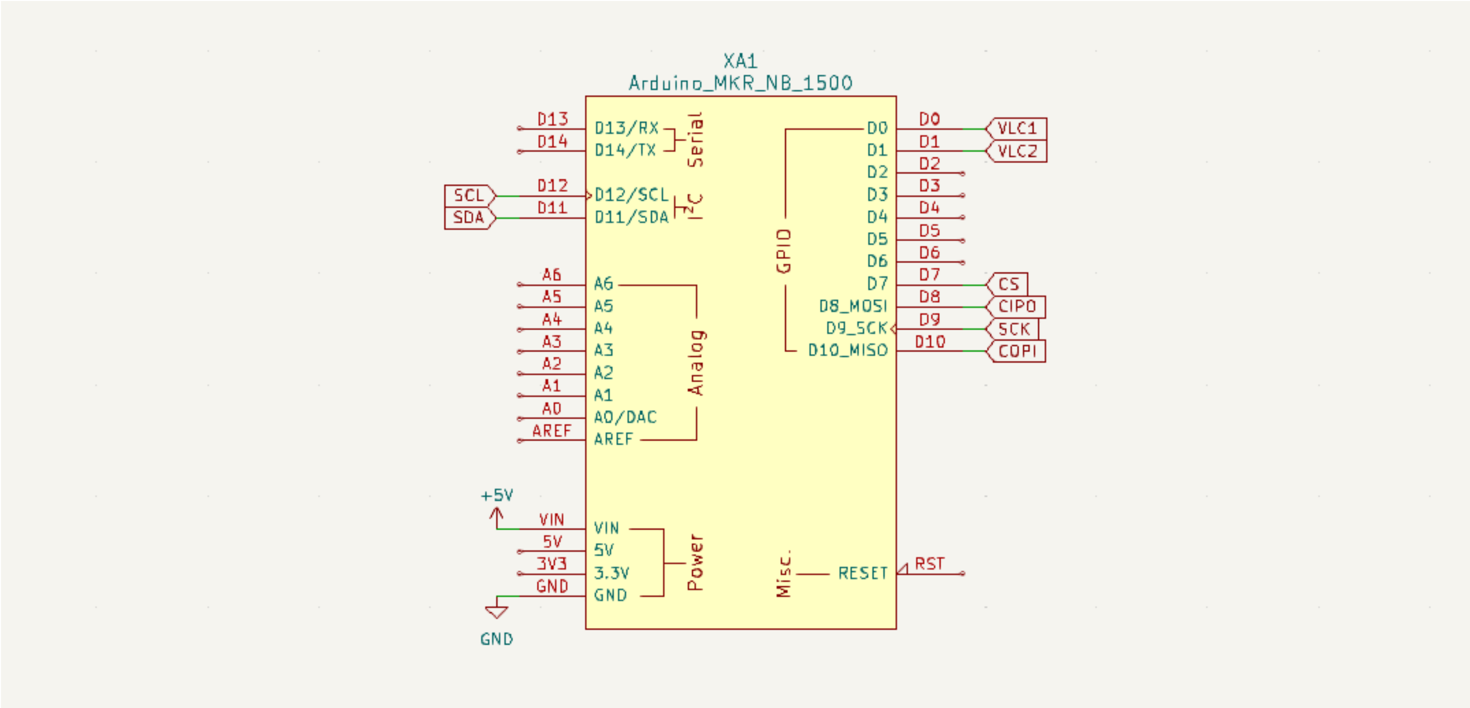


Figure 3: Wiring Diagram.

As it stands now, the microcontroller runs indefinitely, periodically collecting an array of 128 readings from the ultrasonic sensor and grabbing the values from the pressure transducers. Currently, the cellular transmission block has a unidirectional flow of data out of the microcontroller, into the Soracom Harvest interface. Later, there will likely be a reverse flow of data, as this can allow for the server to issue commands to the microcontroller. This is important for implementing a sleep mode on the microcontroller, which can cut power usage significantly, in the case that the system draws too much current to last throughout the night without solar power.

The two transducers provide a linear 0.5-3.3 V input to the microcontroller based upon pressure, which go through analog-to-digital conversion and are multiplied to get pressure values. The values are subtracted based upon the (not yet) known constant distance between the sensors to obtain differential pressure, and this is used in a the hydrostatic pressure linear equation: $h = p/(\rho * g)$ [1], where p is the gauge pressure, ρ is the density of water at a known temperature, g is the acceleration of gravity at water level, and h is the desired water level. This is added to the known height of the bottom transducer, and the measurement is afterwards ready for transmission.

The ultrasonic sensor's readings go through a line of signal processing on the microcontroller:

- DC removal, subtracting the mean of the readings.
- Fast Fourier Transform, to obtain an array of frequency domain data.
- Accumulation, to reduce the effects of noise.
- Whitening, to normalize the signal for convolution.
- Convolution with an array of constants, to take advantage of the previous two steps. This requires calibration, which will be left to after the completion of the sensors blocks.

- Edge detection, meant to isolate a single modal frequency, which is the doppler frequency.

This final frequency is used in the doppler shift equation among known constants:

$$v = \frac{c\Delta f}{2f\cos\theta} \text{ [2]}$$

where c is the speed of sound in water, f is the frequency transmitted by the sensor, Δf is the received frequency normalized by f, θ is the angle of the sensor with respect to the current, and v is the desired water velocity. These two values are finally loaded into floating point variables, ready to be sent by the cellular transmission block.

4.1.3 General Validation

As the team's designs have materialized, nearly all of the problems we ran into happened to hinge upon the initial choice of microcontroller. The use of the Arduino MKR NB 1500 was a recent decision made to alleviate all of these issues at once. Originally, before I began working on it, the generally accepted direction for this block was to use a single-board computer (SBC) like a Raspberry Pi. Over time, it became clear that this introduced complications with several other blocks:

- The power draw was comparatively large, requiring a larger LiPo battery for overnight use, which is a danger for a unit that is meant to be left out in the sun.
- The Pi itself had a large footprint, requiring a larger enclosure with cooling solutions.
- an SBC needs to run an entire operating system, which is quite a lot of overhead and complication for a system that remotely takes measurements and relays them to a server.
- The previous Soracom interface, a separate third party USB, was not working correctly with the Pi drivers.

Therefore, each team member was having troubles with their own blocks because of this early assumption that we ran with without searching for a simpler solution. The MKR board has a fraction of the power draw and footprint, it runs a single executable as opposed to an entire OS, and it integrates the Soracom interface into the unit, which at this point has been tested to be working. This solution reduces the complications of the block's interfaces, as well as removes a design bottleneck on one of every other team members' blocks.

The microcontroller block's interfaces are made to accommodate its neighboring blocks, as opposed to the other way around. The water flow sensor, for example, is connected to the microcontroller via the board's Inter-Integrated Circuit (I2C) interface. The sensor, located potentially a few meters away from the microcontroller, turns high-frequency readings into digital encodings before sending them via this relatively faster protocol. This was chosen in order to negate the potential distortion of readings by transmission line effects, and to give the microcontroller block full control of the clock speed of the sensor. On the other hand, the pressure transducers require the use of the Serial Peripheral Interface, so all of the peripheral sensors by necessity cannot use the same protocol. This is no issue, since the microcontroller can take the "parent" role of both protocols at the same time. The output interface is a strange one, as it is an artifact of the cellular transmission block and this block being entirely separate entities only a couple weeks ago. Back when this block was intended to be an SBC, there was to be one script that would handle capturing and processing data, and a separate one for handling GSM procedures and dispersing updates to the external server.

Due to the aforementioned microcontroller block design recourse, this interface has been reduced to simply storing two floating point values into variables at runtime. In the future, these may be consolidated into one block, as it currently makes sense to do so.

The power requirements of the MKR board are simple: a nominal 5 to 7 volts DC, a minimum current of 100mA, and a maximum current of 190mA [3]. The input pins are only rated for a maximum of 3.3 volts in, so the pressure transducers specifically must meet this requirement. The board is capable of providing the sensors block power, but at this point the plan is for the power management and PCB blocks to handle this task.

4.1.4 Interface Validation

pcb_ntgrtn_mcrcntrllr_dcpwr

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Inominal: 100mA	This is defined in the technical specification for the MKR board.	This is the idle current used by the board while not sleeping. This is primarily what the battery capacity for the system is modelled around.
Ipeak: 190mA	This is defined in the technical specification for the MKR board.	This is the highest draw the MKR board is capable of, and it will typically reach this threshold during a GSM handshake (establishing cellular connection) on startup.
Vmax: 7V	This is defined in the technical specification for the MKR board.	This voltage is the maximum rating of the voltage regulator on the board. It should never receive spikes this high.
Vmin: 5V	This is defined in the technical specification for the MKR board.	This is the minumum operating voltage for the CPU on the board.

vlcty_snsr_mcrcntrllr_comm

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Protocol: I2C	This is known to work in cables past 2 meters in length at 100kHz.	This block will be connected to the ultrasonic sensors from a distance of less than two meters.
Messages: 12-bit (signed) readings	This is the output of the ADC I am using in the ultrasonic sensor.	MAX11613 Specification [4].
Vmax: 3.3V	This is defined in the technical specification for the MKR board.	The sensors blocks will have their outputs limited to this same voltage, to avoid circuit burnout.

dpth_snsng_mcrcntrllr_dsig

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Protocol: SPI	The current sensors used require the use of SPI.	The microcontroller supports SPI and I2C simultaneously.
Vmax: 3.3V	This is defined in the technical specification for the MKR board.	The selected peripherals will be the determining factor of this property.
Vmin: 0V	This is the voltage of common ground between the sensors and the microcontroller	both blocks will be grounded at the same electrical node.

mcrcntrllr_cllr_trnsmssn_data

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Messages: Floating point water flow rate	Float used for greater precision.	This is verified to be able to be sent to the system's website.
Messages: Floating point water depth	Float used for greater precision.	This is verified to be able to be sent to the system's website.
Frequency: updated and sent once per five minutes.	This value was specified by the stakeholder as a reasonable update frequency.	The board is capable of running all signal processing on 256 data points within the span of seconds (tested).

4.1.5 Verification Process

1. Supply 5V DC to the Vin pin of the microcontroller. Observe the state of the power indicator LED, and the current draw that shoud be below nominal.
2. Repeat step one, but supply 7V DC.
3. Flash the main executable to the microcontroller, with it hooked up as described in the Design Section to the two sensors blocks. Open the serial monitor in the Arduino IDE, and see how it drives both sensors and calculates the two desired values.
4. Hook the oscilloscope to the SDA pin, and observe the logic level voltages of the I2C protocol.
5. repeat the previous step, but with the COPI pin, to observe the logic level voltages of the SPI protocol.

4.1.6 References and File Links

[1] W. R. Brody and J. D. Meindl, "Theoretical Analysis of the CW Doppler Ultrasonic Flowmeter," in *IEEE Transactions on Biomedical Engineering*, vol. BME-21, no. 3, pp. 183-192, May 1974. https://ieeexplore.ieee.org/abstract/document/4120758?casa_token=Fy0NIgOb7boAAAAA:o_4Lz6Yn6SuZX-2wrrQfLKvleqE68GgYcht_vBv453eMCqF841xkxfmDl4bZcb27P1jmLHlL. (accessed Feb. 11, 2023)

[2] E. Bossart, "Hydrostatic level measurement in open geometries and vessels — calculation of the filling height," in *Wika Blog*. https://blog.wika.com/products/level-products/hydrostatic-level-measurement-open-geometries-vessels-calculation-filling-height/?doing_wp_cron=1676172602.2239339351654052734375#. (accessed Feb. 11, 2023)

[3] "MKR NB 1500," in *Arduino Docs*. https://docs.arduino.cc/hardware/mkr-nb-1500. (accessed Feb. 11, 2023)

[4] "MAX11612–MAX11617: Low-Power, 4-/-/12-Channel, I2C, 12-Bit ADCs in Ultra-Small Packages," in *Maxim Integrated*, 2012. https://www.analog.com/media/en/technical-documentation/data-sheets/max11612-max11617.pdf . (accessed Feb. 11, 2023)

4.1.7 Revision Table

Date	Revision made
2/11/2023	Cameron Lein: Initial subsection creation.
3/12/2023	Cameron Lein: Revisions for Project Verification.

4.2 Water Velocity Sensor

4.2.1 Description

The goal of this block is to periodically obtain water velocity from under the surface of a unidirectional channel. This sensor will be at an uncertain depth, so it will be a doppler unit angled up at the surface. This is a custom unit that generates a 1MHz sinusoidal waveform with a square wave oscillator, a pair of low-pass filters, and an op-amp. This signal is amplified to maximize readable range, and then emitted from and received by a pair of ultrasonic sound sensors, oriented at the floor of the canal and angled up towards the water surface. The received signal is channeled into a double-balanced mixer to increase the gain, and finally an Analog-to-Digital Converter to create a digital log of the voltage waveform. This will be sent off to the microcontroller block over I2C, where it can be digitally processed to find doppler frequency shift and therefore water velocity. The final result of the processed readings from this sensor and the depth sensor block can be multiplied together to attain the desired metric of water volume over time passing through the canal.

4.2.2 Design

This is a custom-built sensor, designed particularly to be a standalone, low-cost unit that can function as a child in an I2C environment over the distance of at least two meters from a parent (control) unit, such as a microcontroller. Within the ecosystem of this project, it is currently connected via an RJ45 port to its two interfacing blocks: the microcontroller and the PCB, which distributes power.

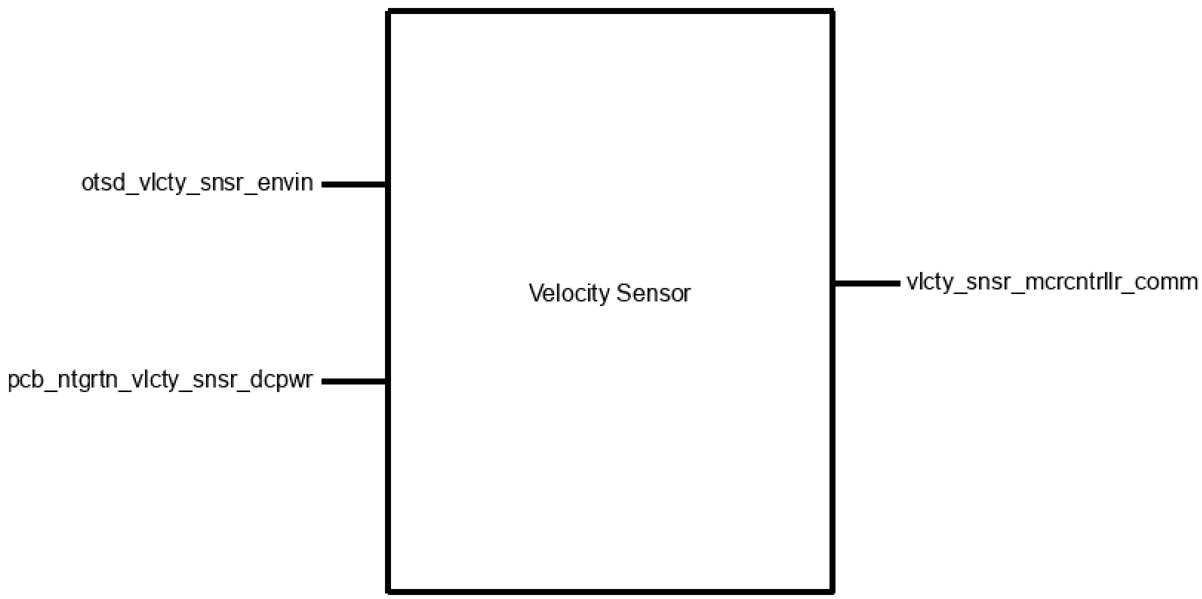


Figure 4: Black Box Diagram.

Figure 4 shows the external interface and two block interfaces of the abstracted velocity sensor block.

- otsd_vlcty_snsr_envin: This is the environmental input of the sensor; it collects a reflected ultrasonic frequency through water.
- vlcty_snsr_mcrcntrllr_comm: The microcontroller clocks the unit's data collection, receives exported data, and toggles power to the oscillator and amplifier.
- pcb_ntgrtn_vlcty_snsr_dcpwr: The 5V power input and common ground is distributed to the unit via the PCB.

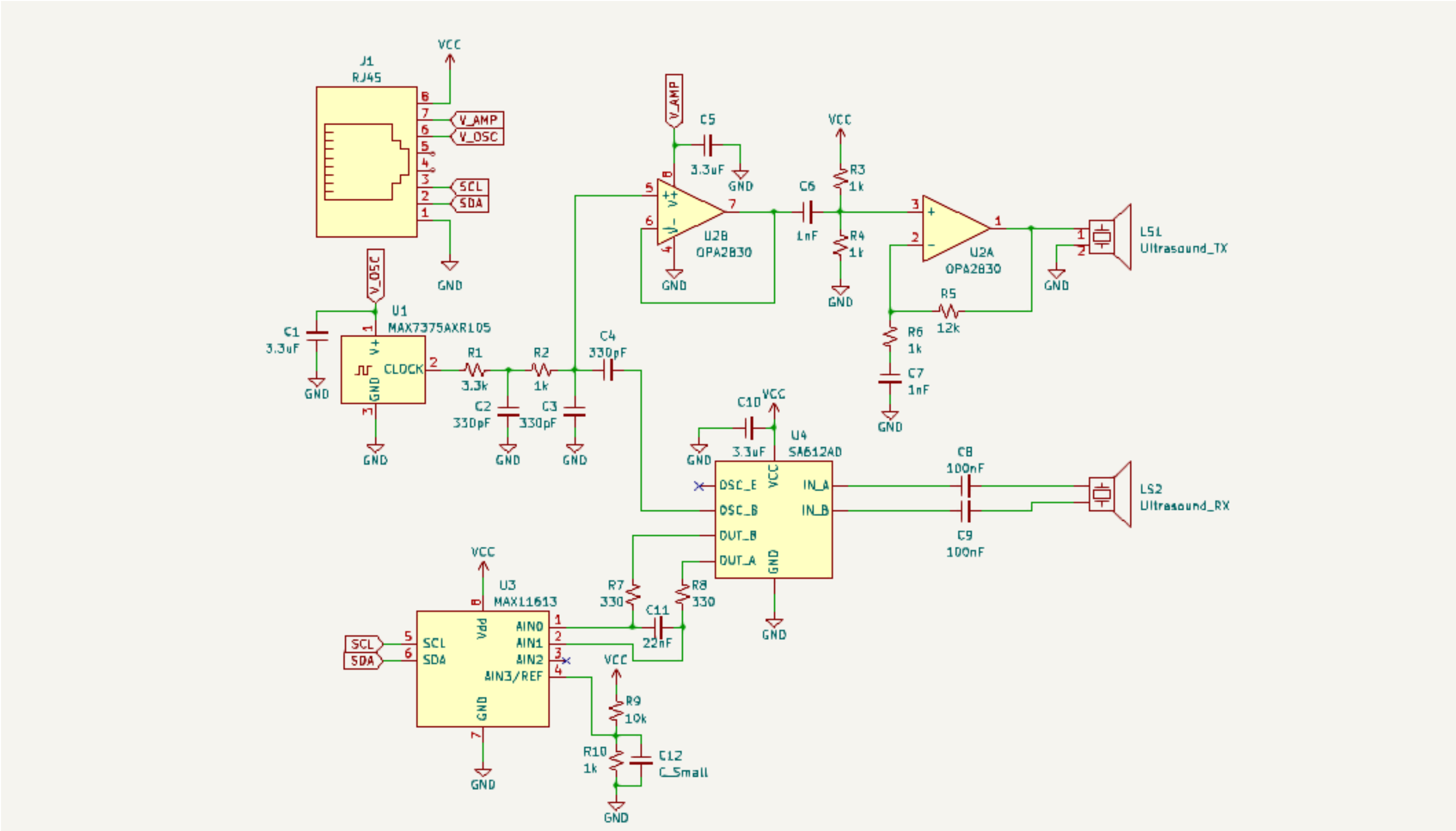


Figure 5: Wiring Diagram.

The unit has a linear flow, as follows:

- A 1MHz, 5 Vpp square wave is generated by the MAX7375, when toggled on by the V_OSC signal.
- The square wave passes through two passive low-pass filters, to better approximate a sine wave.
- The now attenuated sine wave passes through the OPA2830 AC-coupled operational amplifier, bringing it back up to 4 Vpp, when toggled on by the V_AMP signal.
- The sinusoidal wave is transmitted by an ultrasonic transducer, and received by another identical transducer.
- The received waveform, now shifted by the velocity of the water, is passed into the input of a dual-band gilbert cell mixer, SA612AD. The original, pre-transmission waveform is referenced, and the output of the IC results with the same frequency, but plus or minus the original 1MHz frequency, resulting in the isolated doppler shift frequency waveform.
- The isolated doppler frequency waveform is routed to an ADC, which converts the analog readings to 12-bit values. When clocked by the parental I2C unit, in this case the microcontroller block, the value is sent out over the SDA line.

The 1MHz frequency matches the resonance frequency specification of the SMFM21F1000 ultrasonic transducer pair. The low-pass filters have cutoff frequencies of 150KHz and 480KHz, which specifically target the higher harmonics

of a 1MHz square wave. This setup is preferred over a dedicated sine-wave generator, which is generally much more expensive than the MAX7375 and passive components, both in terms of price and power.
The limit of amplification to 4Vpp by the amplifier is to make it more compatible with microcontroller applications, such as one this project is aiming for, while also maximizing the usage of the capabilities of the SMFM21F1000 transmitter. Similar industrial-level sensors will drive their sensors with 12V waves, but these are typically used in rivers or inside of pipes. The application of this project is for low-velocity irrigation canals, typically capping at less than a meter in depth, meaning that range can be sacrificed for a lower driving voltage.
The mixer is used primarily because the ADC is unable to capture waveforms in the ultrasonic range in a high enough resolution to tell what their frequencies are. The received frequency, shifted by the forward flow of the water, is compared with the original driver frequency to get a much slower change in frequency, which readable by the ADC and still provides the information needed to resolve the flow rate of the water.

4.2.3 General Validation

The system, as designed for lower cost, has some limitations, but none that interfere with the functionality of the overall system. The use of the mixer removes the change in frequency from the received frequency, effectively removing the ability to detect a backwards flow (it would be represented as a forwards flow). This is not a problem for this application, as the system will be exclusively deployed in unidirectional flow environments, like irrigation canals, and not in open bodies of water, where something like a boat's doppler radar might be employed.
When the enclosure for the two sensors blocks is completed, the angle of the sensors will be oriented 30 degrees up from horizontal, and the unit will be placed at the floor of the channel, underwater. This orientation is proven to maximize the accuracy of readings doppler velocity readings in most cases, and optimizes the range of depths where the sensor remains accurate [1].
The sensor block interfaces with the microcontroller via I2C, which has a limited effective range, depending upon frequency and voltage. At 0-3.3V and 100KHz collection, I2C can become unreliable past a length of two meters. Currently, the system's frame, which stands in the channel and suspends the microcontroller and transmitter above water, is one meter tall, which means the used protocol will suffice.

4.2.4 Interface Validation

otsd_vlcty_snsr_envin

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Output: 4Vpp	This is the maximum capability of the SMFM21F1000.	The amplifier brings the 0.4Vpp output of the filters back up to this amplitude.
Output: 1000 Hz out	This is the resonant frequency of the SMFM21F1000 pair.	The MAX7375 generates a square wave of this frequency.
Input: 1000 Hz - Shift	1000 Hz is the resonant frequency of the SMFM21F1000 receiver.	The mixer compares the received frequency with the driver frequency to obtain the shift.

vlcty_snsr_mcrcntrllr_comm

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Messages: 12 bit readings	This provides ample resolution for the determination of the frequency of a 4 Vpp wave.	The ADC creates 12-bit signed values from its input.
Protocol: I2C	This is a fast enough protocol for capturing frequencies without noise distortion.	The ADC uses I2C as its transmission protocol.
Vmax: 3.3V.	The selected microcontroller has a logic high of 3.3V on its digital pins.	The ADC matches this specification.

pcb_ntgrtn_vlcty_snsr_dcpwr

Interface Property	Why is this interface property this value?	Why do the design details for this block meet or exceed each property?
Inominal: 4mA	This is the ideal current usage of the unit at idle.	The system's high power-draw ICs have their power supplies as a toggle.
Ipeak: 7mA	This is the maximum output of a digital pin on the microcontroller.	Each IC on-board uses less than this, but this property may need to change soon.
Vmax: 5V.	The oscillator cannot handle more than 5V in.	The maximum voltage output of the microcontroller is 5V.
Vmin: 3V.	Arbitrary, but going lower severely reduces the range of the transducer pair.	The ICs on-board all operate on low voltages down to 2.5V.

4.2.5 Verification Process

1. Connect the unit to a microcontroller capable of running the project's Arduino code and dispersing the maximum specified voltage and current.
2. Disconnect the power pins, and use a controlled supply. See that it still transmits data at Vmax and Vmin.
3. Probe the output of the amplifier. Observe that the wave is 4 Vpp and that the frequency is 1000Hz.
4. Probe the output of the mixer. Observe that the frequency has been normalized by 1000Hz.

4.2.6 References and File Links

[1] H. Bonakdari and A. A. Zinatizadeh, "Influence of position and type of Doppler flow meters on flow-rate measurement in sewers using computational fluid dynamic," in *Flow Measurement and Instrumentation*, vol. 22, no. 3, pp. 225-234, 2011. https://www.sciencedirect.com/science/article/abs/pii/S0955598611000288. (accessed Feb.

16, 2023)

4.2.7 Revision Table

Date	Revision made
2/16/2023	Cameron Lein: Initial subsection creation.
3/12/2023	Cameron Lein: Revisions for Project Verification.

4.3 Cellular Communication

4.3.1 Description

A SORACOM sim card will be inserted into the primary system and allow the unit to connect to the internet even when placed in the middle of a farming area. Soracom Air SIM modules do not have any limitations with cellular networks around the world. A subscription for LTE connectivity will cost around two dollars a month and will allow the device to be easily used without a nearby WiFi connection. With the support of the backend code, we will be able to process and store the information online for easy access.

The device will send an HTTP request to our Python website containing a JSON file. The JSON file will contain all the sensor data (modular which allows for easy addition of any extra data consumers could request) organized and with the correct values. The JSON file makes it easy for the website to grab specific data and understand exactly which sensors that specific data came from. This also lets us avoid having to transmit each sensors data separately as the JSON will tell the website exactly where each value came from.

4.3.2 Design

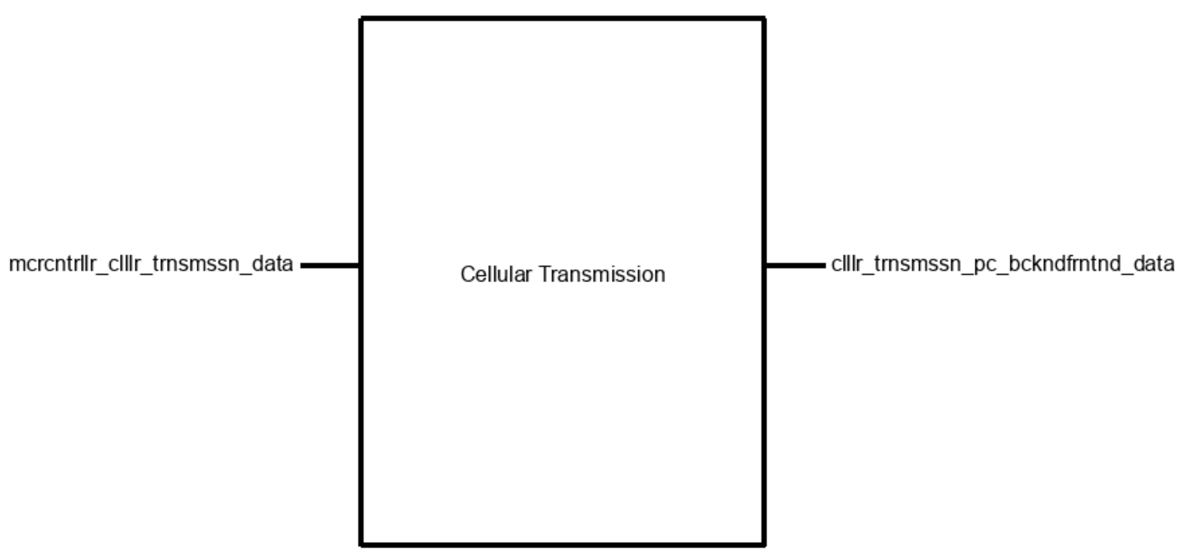


Figure 6: Black Box Diagram.

The center of the block represents our cellular transmission system which is responsible for transmitting our data to our website. The cellular transmission system is connected to the **PC Backend/Frontend Block** and to the **Microcontroller** block. The microcontroller will input the collected sensor data and the cellular transmission system will output the data to the website.

The Arduino we purchased for the project is available here and includes the following items:

- MKR NB 1500 Arduino
- Dipole Pentaband Waterproof Antenna
- Soracom Global IoT SIM

Figure 7: Physical picture of Microcontroller, Antenna, and SORACOM SIM.

The microcontroller is equipped with an integrated Long-Term Evolution (LTE) modem that enables seamless interfacing with any cellular tower located within the United States. The accompanying Subscriber Identity Module (SIM) card offers cost-effective internet connectivity at a nominal monthly fee of approximately $2, which can be further reduced since it solely facilitates data transmission and not internet browsing. The Arduino board boasts a built-in SIM slot, rendering the entire connectivity system almost entirely pre-assembled. Most of the design process revolves around the programming of code to ensure the optimal transmission of data.

The code is divided into three main parts: the initial connection to the cell tower, gathering the data into a JSON, and transmitting the data. This all happens very quickly and loops every five minutes ensuring the continuous transmission of data.

- Cell Tower Connection: The microcontroller uses an APN (Access Point Name settings), USERNAME, and PASSWORD in order to set up a connection to the gateway between the carrier's cellular network and the Internet. The APN can be thought of as a password to be able to access the cell tower and make sure that the SIM we are using is a real SORACOM SIM. The username and password are provided to the cell tower which is sent to SORACOM to make sure that the SIM we are using is a real SORACOM SIM card. Lastly, the code needs the PIN number of the SIM to be able to use the SIM card as the SIM will reject access unless the PIN is provided first. Once all this information is provided, the code will start up the modem with

  ```
  nbAccess.begin(PINNUMBER, apn, user, password)
  ```

  and the nearest cell tower should respond by accepting the connection and printing that it is connected within the serial plotter.
- Gathering Data: The microcontroller takes in the sensor readings as float values and organizes the information within a JSON file. JSON stands for JavaScript Object Notation. It is a lightweight, text-based format used for storing and exchanging data between web servers and applications. JSON files are easy for both humans and machines to read and write. They are primarily used to transmit data between a server and web application, as well as for storing data in a file. The JSON is attached to the POST request which is then sent to our website.
- Data Transmission: The microcontroller connects to our website by using a URL, PATH, and a PORT to find the location of the website. The microcontroller sends a POST request asking the website if it can send it data and the website contains code to accept the request from the microcontroller.

4.3.3 General Validation

The cellular connectivity system as of yet is fully functional and is capable of connecting to our website and transmitting information. The microcontroller gathers the required data from the connected sensors and stores the data into two separate float variables. Within the cellular connectivity code, the data is organized into a JSON file and is placed into a specially built POST request so that the data may be sent to the server and that the server may recognize that our microcontroller is requesting to interface with the website. The code is mostly optimized as it solely needs to gather information and send it to our website.

Concerns as of now are mostly centered around issues that may occur such as a loss of connection or data transmission issues. Our solution to a loss of connection issue is that the microcontroller code will shut off the connection to the cell tower at the end of transmission and reconnect every 5 minutes before it sends the data again. This is so that if it ever loses connection, it will automatically attempt to reconnect every five minutes. Shutting down cell tower connections also has the benefit of saving us money since it will not be using data 24/7 even when the device is not transmitting data. If there are transmission issues, we can use SORACOM harvest which is SORACOMS proprietary way of viewing Arduino data. This allows us two ways of checking data transmission so that if there are any issues with our website, we can use SORACOMs website to check to see if the data is transmitting properly. Using this service does cost a bit extra ($0.50 a month) so it is only necessary to use should the website not function.

4.3.4 Interface Validation

Table I: mcrcntrllr_clllr_trnsmssn_data

Interface Property	Explanation	Verification
Messages: Floating point water flow rate	The water flow rate value will be collected as a floating point number on the microcontroller for calculating how much water has entered a given area.	This value can be checked on the serial plotter. We can make sure that the value is correct by using a controlled environment such as a bathtub.
Messages: Floating point water depth	The water depth value will be collected as a floating point number on the microcontroller for calculating how much water has entered a given area.	This value can be checked on the serial plotter. We can make sure that the value is correct by using a controlled environment such as a bathtub.
Other: Frequency: updated and sent once per five minutes	As for the customer request, it is not necessary to send data every second as the device will be deployed for long periods of time. This will save power and allow the device to operate for long periods of time especially when there is no sunlight.	This can be tested by checking the serial monitor and timing how long it takes the device to loop before taking a data point.

Table II: clllr_trnsmssn_pc_bckndfrntnd_data

Interface Property	Explanation	Verification
Datarate: Active connection to LTE broadband tower while main device is running.	The main device will be out in a field for extended periods of time and needs a way to communicate with the users over long distances. This will be accomplished using a SORACOM SIM card and a Arduino with capabilities for accepting a SIM card.	There are two ways of checking for an active connection to a cell tower. The serial plotter on the microcontroller has been coded to reply when it is connected to a cell tower but this does not provide further information. SORACOM's website allows us to see exactly which tower our SIM is currently connected to.
Datarate: Data will be transmitted online in 5 minute intervals.	It is unnecessary and a waste of cell data to send sensor information every second. Code within the Arduino will limit the transmission to every 5 minutes and sleep in between to save power. The website will just need to wait for an HTTP POST request.	If new data appears and is properly displayed within a table on the website after every 5 minutes then we can be sure that the device is transmitting every 5 minutes and the website is picking up that request every 5 minutes. The serial plotter within the Arduino will display exactly when data is transmitted giving us two ways of checking that data transmission only occurs every 5 minutes.

Messages: Can send an HTTP request to the server with an attached JSON file.	A JSON file allows us to organize our sensor data in a quick and easy way for our website. The JSON sections off the sensor data so that the website can tell exactly where each value came from. For example:
	`String postData =`
	`"{\"FlowRate\":"+`
	`String(flowRate) +`
	`",\"Volume\":"+`
	`String(volume) +"}";`
	sends two values as a JSON: the flowrate and the volume. The website gets the JSON and understands the difference between the values received.
		This can simply be verified through the data tab on the website as if the transmission is successful the exact values seen on the serial plotter will match the data table on the website. We know the JSON file system works if the data is in the proper column on the website. Harvest can also be used here to verify the values.

Messages: Data can be accessed/stored through both SORARCOM Harvest and the website.	We wanted to give our customers two different ways to check the data that is being received. Our website will display the data in a table format on a dedicated data page for the customer to see how much water is entering a field at a given moment. SORACOM also has a proprietary way to see data being transmitted using their SIM cards which gives the customer two ways of accessing the same data.	This can be verified by loading the "flow rate" and "volume" variables within the microcontroller code with preset values and checking if the same values appear on both SORACOMS website and our own website.

4.3.5 Verification Process

1. Check the serial plotter to see if connected to a cell tower and cross-check with the SORACOM web console to see the exact tower and data speeds.
2. Check the serial plotter to see if the data is being properly organized into a JSON file. We can crosscheck this using SORACOM harvest to make sure that the same data is being sent to our website.
3. Check the serial plotter to see if POST request is formatted correctly and check the data tab on our website to see if data is received and placed into data table. We can check this without the sensors by placing preset values into the variables to make sure those same values appear on the website.
4. Time transmission times by checking/timing serial plotter and refreshing the website after 5 minutes to make sure data is being sent every 5 minutes.

4.3.6 References and File Links

[1] T. A. Team, "MKR NB 1500: Arduino documentation," Arduino Documentation — Arduino Documentation. [Online]. Available: https://docs.arduino.cc/hardware/mkr-nb-1500. [Accessed: 11-Feb-2023].

[2] T. A. Team, "Scanning available networks with MKR NB 1500: Arduino documentation," Arduino Documentation — Arduino Documentation. [Online]. Available: https://docs.arduino.cc/tutorials/mkr-nb-1500/nb-scan-network. [Accessed: 11-Feb-2023].

4.3.7 Revision Table

Date	Revision made
2/16/2023	Adam Farhat: Initial subsection creation.
3/12/2023	Adam Farhat: Revisions for Project Verification.

4.4 Website (PC Backend/Frontend)

4.4.1 Description

A program written in Python will assist in the collection of data from the main system for storage and remote access. The program will be hosted on a website that specializes in hosting Python programs which will make it easier once it comes to handing off our design to our customers. The code will interface with the SORACOM module within the main system and transfer all necessary collected data. The website will show the data in real-time and with the assistance of the frontend, will display the data collected in a simple and readable way for the irrigation officers. Tabs will allow the user to access data stored within an excel type table that show the date, time, and collected information to allow the irrigation officers to provide the correct water use data to their local counties.

The frontend will make the website look presentable, readable, and easy to follow. Fields will direct the user to specific areas such as the alert page where a user can enter their email to receive email alerts once a specific amount of water has entered the fields. Other tabs include a home page which will chronicle the making of our entire project. A help tab will allow a user to access a group-made guide that will discuss how the project works and anything that may be useful for anyone utilizing the device in an actual irrigation canal. With the addition of an SQL database, the website can store information gathered by the main device and with the inclusion of the SORACOM module on the main device, the system can transmit the information to the website. The website will then organize the information into a .CSV file (excel) and allow the user to access all information gathered from when the device was activated to the current date that a user is viewing the website.

4.4.2 Design

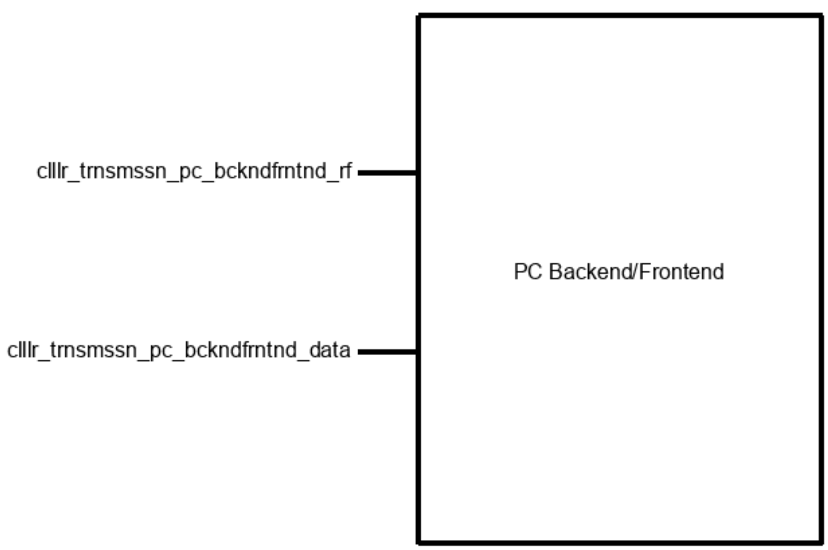


Figure 8: Black Box Diagram.

The center of the block represents the written code within the system. The only interface which needs to be connected to the main block is the **Cellular Transmission** block. This is because the code will be hosted off the system which means that the power supply as well as the data-collecting hardware does not directly communicate with the website. Instead, the cellular transmission system within the main device will gather the data from the hardware and simply transmit the gathered information. This means that the only block that needs to connect with our code is the cellular transmission block as the rest of the system interfaces with the microcontroller/transmission system.

The backend code will handle the following:

- Retrieval of transmitted data.
- Storage of data over a long period of time.
- Alerting user after "X" amount of water has passed through the system.
- General website features such as tabs, navigation, website information, etc.

The frontend code will handle the following:

- Displaying the collected data to the user.
- Displaying project information/help to the user.
- Button to allow the user to download data into a CSV.
- General website design such as font, pictures, page design, etc.

Generally, as a rule of thumb, if a file ends in .py it is written in Python code and would be considered as **Backend** code. Conversely, a .html file written in HyperText Markup Language (with a bit of CSS) would be considered as **Frontend** code.

Link to GitHub page included in the File Links section. Included below is an outline of each file within the source code:

- The file "*main.py*" will handle starting up the Flask web server. Flask is a web application framework written in Python that allows a developer to easily create and scale web apps.
- The file "*views.py*" will store all the information that a visitor to the website can access.
- The file "*models.py*" organizes the data received for storage within an SQL database. An SQL database is a way to easily store and organize information. As of 1/20/2023, "*models.py*" only handles organizing user data such as email along with a username/password and a date/time for a login system (as of now has been commented out within the code as I am unsure if it is necessary). The modular nature of the file allows us to easily create any organizational structure required which once the main system is further in development, can receive system data and store them properly.
- The file "*auth.py*" handles a sign-up and login/out system. As of 1/20/2023, I have commented out the sign-up tab as well as the login/out system as we wait for a response from our project partner on how secretive the collected information needs to be. This file also handles all HTTP requests such as ones transmitted by our main device. It also handles placing the data into the SQL database.

- The file "*init.py*" handles the initialization of anything critical for the website to properly function. This includes the creation of a SQL database (only if the database does not already exist), information on where to store files, and URL storage.
- The file "*base.html*" handles the general look of each page. Unlike the specific .html files below, any changes in this file will change the look of every tab within the code. The file also handles the look of flashed messages (like a popup) which are used for situations like a user's email not including a "@" while they are registering for Alerts.
- The file "*data.html*" handles the look of our data table where the long-term stored data can be viewed by the user. This file contains SQL code which will parse through the database for specific sensor data and display it properly within the website.
- The files "*alerts.html*" and "*home.html*" handle the look of the specific tabs. If we wanted to change the look of a button only on the alerts tab, the change would go into "*alerts.html*". This file is also where we would include written information such as instructions on how to use our project or pictures of the project. Anytime a tab is created, it is good practice to create the corresponding .html file should the need arise to change how something appears on specific pages.
- The file "*database.db*" is the database that stores any relevant information for long-term use/viewing.

4.4.3 General Validation

As of 1/20/2023, the website is ready for features to be added. The website is functional and the tabs work as intended. The home page is slowly being built by myself currently including the team's information along with emails should the Quechan tribe ever need to reach us. I believe that all design choices fit the system as I am already far into the process of production and the website is functional and properly designed. Because this block is all about code, there is almost no cost to myself and my team. This block has required a lot of time to get to the point of functionality which has been the biggest cost of the block as of yet.

Concerns as of now are mostly centered around how to host the website. The best and current solution is to utilize a service called "Python Anywhere" which is a website that specializes in hosting Python code and has information on how to host a Flask web server. The problem is that this service costs $5 a month which is well within the amount of our budget but may be difficult for our project partner to deal with after we finish the project. Alternative solutions include having our project partner host the server on his computer which would be free but may be troubling to set up. Another solution would be to host the website on the main our own devices such as our desktops but many issues could arise after the project is delivered and we graduate from Oregon State University but this solution would be completely free.

An alternative solution for the block itself would be to completely disregard the website entirely and just have the internal transmission system along with the Arduino MKR NB 1500 run code to directly transmit collected data to a user at a specified interval. While this removes the ability to directly see current information, it would still be autonomous and a simple fix should some unfixable issue arise in the future.

4.4.4 Interface Validation

Table I: otsd_pc_bckndfrntnd_usrin

Interface Property	Explanation	Verification
Other: Tab bar at top of website for easy navigation will take the user to any page such as home, data, alerts and help tabs.	The website will have different pages for different items such as a page that only displays the device data or a help page that explains how to use the device.	This can be tested by navigating to any page on the website and checking to see if a navigation bar GUI appears with links back to the homepage, along with several menu items that correspond to different sections of the website (Data, Alerts, Help, etc).
Other: Website is viewable through a URL on any internet-connected device.	The website needs to be accessible from anywhere and to get to the website, the user would need to type the URL on an internet-connected device.	The website can be accessed from this url https://quechanirrigation.pythonanywhere.com/home when typed on an internet-connected device.
Other: User can input their email to sign up for alerts. Email will be stored and error detection will prevent multiple signups under the same email.	The website needs a way to communicate to the user once water use limits have been reached and this is accomplished by sending the user an email automatically.	Users can input their email under the "alerts" tab and will receive an email shortly after explaining how the alerts system functions and confirming that they have signed up for alerts. A popup will tell the user if the email has already been registered for alerts to prevent email spam.

Table II: pc_bckndfrntnd_otsd_usrout

Interface Property	Explanation	Verification
Other: Data can be exported into an excel spreadsheet.	Data needs to be ported to excel as requested by the customer for better readability and workability.	A button on the data page will export the data shown on the data page to an excel spreadsheet once clicked.
Other: Data is viewable through a data table under the data tab.	A table will organize the transmitted data from the machine and store it online.	When the user clicks the data page, it will display a table that shows "flowrate" and "volume" as well as each value separated into a new row once a new data set is received.
Other: Website sends user an email once water use limits have been reached.	User needs to be alerted once water limits have been reached so that they can stop watering a field.	An email will be sent to the user alerting them that they have exceeded a specified amount of water within a field.

4.4.5 Verification Process

1. Verify that the website is accessible by typing the URL into a web browser.
2. Verify that the tabs are intractable and the user can view the instruction text or the homepage text.
3. Verify data transmission by loading a false value into the main system and seeing if it matches the given value. Another way of putting this is that we tell the system to transmit the number "1" and check if the website receives and displays the number "1".
4. Verify data is properly stored by closing the web page and reopening it to see if all prior received values are still accessible.
5. Verify data is properly organized by analyzing what values are placed where when viewing the database on the website.
6. Verify email alert system functionality by loading an extremely high value into the main system after we use our personal emails to sign up for alerts.
7. Verify website stability by accessing the website periodically throughout the month. *Only once everything is deemed functional and website is ready for deployment/hosting.

4.4.6 References and File Links

[1] Joe Supan May 20, J. Supan, and Columbia University Working from Home guide, "How many mbps do I need: 2022 internet speed guide," Allconnect, 11-Nov-2022. [Online]. Available: https://www.allconnect.com/blog/faqs-internet-speeds-what-speed-do-you-need. [Accessed: 20-Jan-2023].

[2] "Python website full tutorial - FLASK, authentication, Databases & More," YouTube, 01-Feb-2021. [Online]. Available: https://www.youtube.com/watch?v=dam0GPOAvVI. [Accessed: 20-Jan-2023].

[3] "Welcome to Flask," Welcome to Flask - Flask Documentation (2.2.x). [Online]. Available: https://Flask.palletsprojects.com/en/2.2.x/. [Accessed: 20-Jan-2023].

[4] "Flask HTTP methods, Handle GET & Post requests," Flask HTTP methods, handle GET & POST requests - Python Tutorial. [Online]. Available: https://pythonbasics.org/flask-http-methods/. [Accessed: 11-Feb-2023].

[5] "Getting to know the SORACOM user console," User Console — SORACOM Developers. [Online]. Available: https://developers.soracom.io/en/start/user-console/. [Accessed: 11-Feb-2023].

[6] "Overview SORACOM Harvest," Soracom Harvest Overview — SORACOM Developers. [Online]. Available: https://developers.soracom.io/en/docs/harvest/. [Accessed: 11-Feb-2023].

[7] T. A. Team, "MKR NB 1500: Arduino documentation," Arduino Documentation — Arduino Documentation. [Online]. Available: https://docs.arduino.cc/hardware/mkr-nb-1500. [Accessed: 11-Feb-2023].

[8] T. A. Team, "Scanning available networks with MKR NB 1500: Arduino documentation," Arduino Documentation — Arduino Documentation. [Online]. Available: https://docs.arduino.cc/tutorials/mkr-nb-1500/nb-scan-network. [Accessed: 11-Feb-2023].

4.4.7 Link to GitHub Page

[9] Adamhat, "Adamhat/IrrigationDevice: Capstone Project working with the Quechan Indian Tribe in South California.," GitHub. [Online]. Available: https://github.com/Adamhat/IrrigationDevice. [Accessed: 20-Jan-2023].

4.4.8 Revision Table

Date	Revision made
2/16/2023	Adam Farhat: Initial subsection creation.
3/12/2023	Adam Farhat: Revisions for Project Verification.

4.5 Solar and Battery

4.5.1 Description

The purpose of this block is to provide power to a water flow rate monitoring device that will be used to track the quantity of water passing through an irrigation channel in agricultural applications. The application of the project requires that the end user can easily set it up at a desired point along an irrigation channel and leave it for an extended period of time. Crops will be watered during the day and the night, which requires that the system can monitor the flow of water at all hours of the day. The system will be used in farmland without access to power from the grid to supply the system or recharge the battery. These overall system requirements mean that this block needs to be able to maintain power to the system without any user interaction. Therefore, in order to maintain operation during the day and night, a solar panel will be used to charge a rechargeable battery. The input to the block is sunlight during the day, which is received by the solar panel and used to provide power to both the system and a rechargeable battery. During the night, and times without sufficient sunlight, the battery will provide the necessary power to the system. Output by the block is 5V with a nominal current of 1A supplied to the rest of the system. The rechargeable battery is charged by the solar panel when direct sunlight is available. Recharging is handled by a solar panel controller that both maintains the charge of the battery and provides power to the system. A boost converter is used to step up the output voltage from 4.4V from the solar controller to 5V in order to meet the demands of the microcontroller.

4.5.2 Design

Seeing as the system needs to remain operational for extended periods of time without input from the end user, a rechargeable battery and solar panel will be used. The project is intended for agriculture in Southern California, which should allow for ample sunlight during the daytime. Using a sufficiently large rechargeable battery, solar panel, and efficient charge controller will allow for the project to operate throughout the day and night without shutting down or needing the user to intervene. The top level diagram for this block is shown in figure 7. The only input to the block is sunlight, and output by the block is regulated power which is supplied to the rest of the system. The input and output interface properties are listed in Table 1 and Table 2 respectively. The amount of sunlight received as an input depends on the time of day, season, and weather. The output of the block is 5V and a nominal current of 1A. A Voltaic solar panel with specifications of 6V and 2W is used to supply power for charging the battery as well as powering the system while daylight conditions are sufficient. The dimensions of the solar panel are roughly

5.25” x 4.4” and it meets the IP67 waterproof requirements. This is slightly smaller than the dimensions of the lid of the system enclosure, which will make it easy to secure safely to the enclosure. The IP67 rating will help ensure that the longevity of the project will meet the needs of the end user. The panel is connected to the bq24074 solar LiPoly charger. It is an efficient solar charging controller board ideal for relatively small solar projects compared to commercial or household solar panels. The controller board is designed to efficiently draw the maximum power from the solar panel based on varying sunlight conditions, similar to a maximum power point tracking (MPPT) device. This means that it adjusts current draw based on the voltage of the solar panel to supply maximum power for charging the battery. The controller board also supplies power to the load when there is sufficient sunlight so that the battery remains charged to power the system when there isn’t enough light. The controller board has two outputs, one being reserved for the rechargeable battery and the other for the load output. The chosen rechargeable battery is a 3.7V 6600mAh Lithium Ion battery. A LiIon battery was chosen over alternative rechargeable battery types for its relatively fast rate of charging, high cycle life, and large capacity for its physical dimensions. 3.7V is a common operating voltage for a LiIon battery, and 6600mAh is enough capacity to maintain operation of the system for at least 12 hours. The output voltage of the controller board is regulated to 4.4V. The system requires 5V so a boost converter will be connected to the load output of the charger to convert this to 5V as the output of the block to the system. This will in turn decrease the current provided by the board. The board is rated to provide a maximum load current of 1.5A. The chosen boost converter is the Dorhea MT3608 DC-DC Step Up Boost Power Converter which has a maximum power conversion efficiency of 93%. Calculating with a conversion efficiency of 90% reduces the current to roughly 1.2A which is sufficient for the nominal current of 1A required by the system.

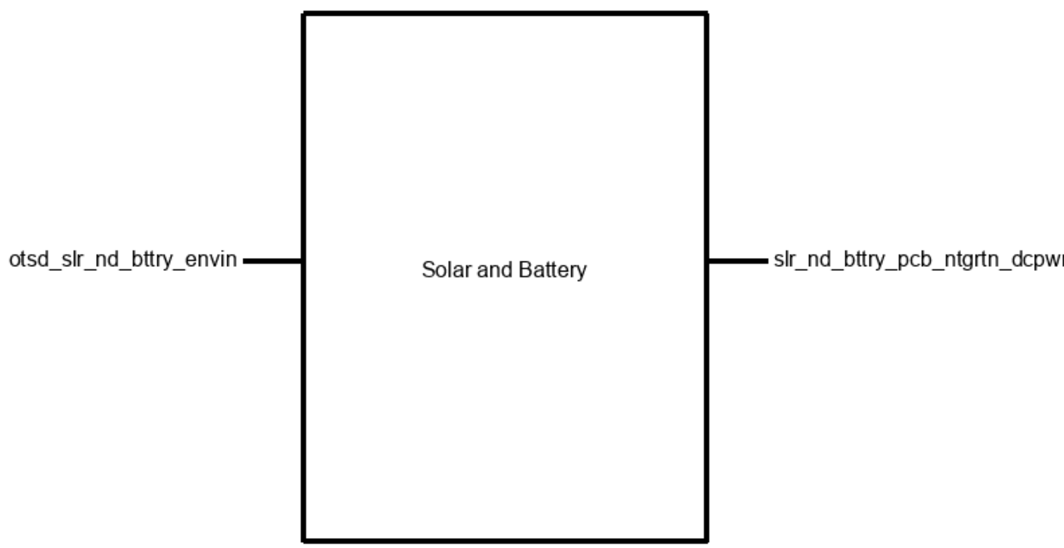


Figure 9: Solar and Battery black box diagram.

4.5.3 General Validation

The power requirements of the system are adequately met by the design of this block. Operation of the system requires 5V and a nominal current of 1A. The output of the boost converter ensures a stable output of 5V without impeding the 1A current draw from the rest of the system. During times of day or adverse weather conditions that lead to a lack of sunlight to effectively power the system, the battery will have sufficient capacity to maintain operation of the system for extended periods of time. During the preliminary design phase of the project it was determined that the system would need to remain operational without any sunlight input to the system for a minimum of 12 hours. This time was chosen so that the system could effectively monitor water flow during intermittent adverse weather during the day, such as cloud cover, and throughout the night. The system will use a low power microcontroller that will be programmed to cycle between an idle state and a wireless transmission state. The low power demands of the system coupled with the power cycling of the microcontroller mean that the 6600mAh LiIon battery will have more than enough capacity to power the system for at least 12 hours. A larger capacity battery may be chosen and easily swapped with the current battery to increase this operation time as needed. The chosen solar charger will adjust current draw to maintain power efficiency based on the changing input voltage from the solar panel during varying sunlight conditions. This feature should ensure that the universal design constraint regarding power supply efficiency is met. The chosen boost converter has a power efficiency specification of "up to" 93%. This value depends on the power conversion of the system, and the boost converter is rated to step 2-24V input to 5V-28V output. This block only requires stepping up 4.4V to 5V, so the efficiency will likely be closer to the maximum specified by the boost converter. This more than meets the demands of the system and the universal design constraint requiring 65

4.5.4 Interface Validation

The interface properties for the input and output interfaces of this block are displayed in the following tables. The input specifications are measurements of light. These values are all determined based on the specifications of the LEOTER Grow Light that will be used for testing this block during verification. While these values are not exactly the same as those of sunlight, they fall within the range of their respective values compared to sunlight and are sufficient for powering the block through the solar panel. The output values are determined by the specifications of the system and the power it requires.

Table I: otsd_slr_nd_bttry_envin: Input

Interface Property	Explanation	Verification
Light: 380 nm - 800 nm	This is the wavelength of the UV grow light that will be used for testing.	The solar panel can receive light in this band as an input.
Light: Maximum: 450 lux	This is the lux of the UV grow light measured an inch from the light.	This value falls within the range of sunlight and the solar panel can accept it as an input.
Light: Minimum: 250 lux	This is the lux of the UV grow light measured a foot from the light.	This value falls within the range of sunlight and the solar panel can accept it as an input.

Table II: slr_nd_bttry_pcb_ntgrtn_dcpwr: Output

Interface Property	Explanation	Verification
Inominal: 1 A	This is the nominal current draw of the system given the microcontroller, sensors and peripherals.	The output of the block is regulated to 1A.
Ipeak: 1.2 A	This is the maximum current provided by the boost converter.	The output of the block is regulated to 1A.
Vmax: 5.5 V	This is the maximum voltage that will be output by the boost converter.	The output of the block is regulated to 5V.
Vmin: 4.4 V	This is the regulated voltage output by the controller board.	The output of the block is regulated to 5V

4.5.5 Verification Process

The following are steps to verify that the block meets the power requirements of the system as described in table 2 above. The verification process will test that the block maintains a stable 5V output at the nominal current draw when a UV grow light is used to simulate sunlight as an input to the solar panel.

1. Make sure the components of the block are properly assembled.
2. Plug in the specified grow light to an outlet.
3. Connect a multimeter to measure the output voltage and/or current from the output of the boost converter.
4. Place the grow light 4 inches above the solar panel.
5. Turn on the light.
6. Measure the output voltage and current after 2 minutes have passed.

The test passes if the output voltage is between 4.4V and 5.5V and the nominal current is 1A.

4.5.6 References and File Links

[1] B. Siepert, "Adafruit Universal USB / DC / solar lithium ion/polymer charger - BQ24074," Adafruit Learning System, 16-Sep-2020. [Online]. Available: `https://learn.adafruit.com/adafruit-bq24074-universal-usb-dc-solar-charger-breakout/design-notes.[Accessed:20-Jan-2023]`.

4.5.7 Revision Table

Date	Revision made
3/12/2023	Isaac Sutton: Initial subsection creation.

4.6 Depth Sensing

4.6.1 Description

The depth sensing will be part of a frame that is placed in the irrigation channel, and will take measurements to calculate the depth of the water in the channel. It consists of two pressure transducers, one placed in the water near the bottom of the channel and the other above the water level. The lower pressure transducer is a known distance from the bottom of the irrigation channel based on the design of the frame that the sensing system is built within. Data is communicated using the I2C protocol between the sensor and the microcontroller located in the primary system enclosure which is placed out of the water. The difference between the measurements of the two pressure transducers is used to calculate the depth of the water. The submerged sensor is used to calculate the depth of the water by measuring the pressure relative to the atmospheric pressure reading which is gathered by the other sensor. Combined with user input irrigation channel geometry and dimensions the system can calculate the depth of the water while considering changes in the weather.

4.6.2 Design

The sensors being used for this block are Honeywell MPRLS0025PA00001AB pressure transducers. They are rated to measure pressures ranging from 0 to 25 PSI with a supply voltage of 3.3V and using the I2C communication protocol. This specific sensor comes already assembled on a PCB breakout board and male header pins to interface with. Two of these sensors are used in order to gather accurate depth readings. One sensor is placed on the submersible frame that is placed in the irrigation channel. The sensor is fixed to the base of the frame so that it is close to the bottom of the channel. Pressure readings from the submerged sensor are used to measure the depth of the water in the channel given the relative atmospheric pressure and the known distance between the sensor and the bottom of the channel. The other sensor is housed by the enclosure containing the microcontroller and power supply on the bank of the channel. This second sensor measures the current atmospheric pressure which will vary depending on the weather conditions. There are three interfaces for this block making consisting of the environmental input and two way communication between the sensors and microcontroller. The interface otsddpthsnsngenvin is the pressure of the water or air on each sensor. Pcbntgrtndpthsnsngdcpwr is the DC power input from the microcontroller to power the sensor. It provides 3.3V and a nominal current of 3mA to the sensor. Finally, the interface dpthsnsngmcrcntrllrdsig represents the I2C communication between the sensor and microcontroller.

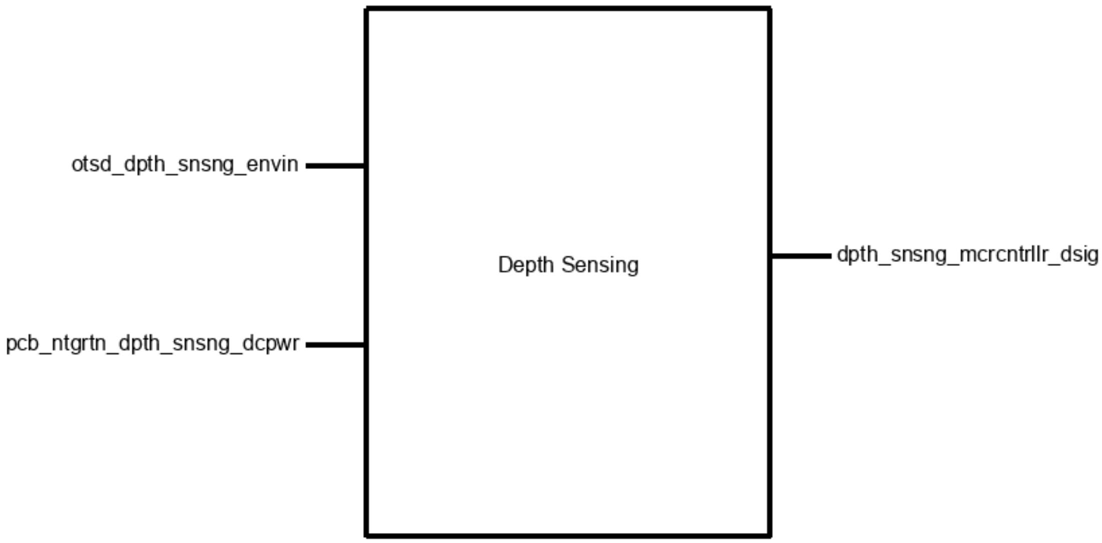


Figure 10: Depth Sensing black box diagram.

4.6.3 General Validation

The purpose of the system is to measure the amount of water flowing through an irrigation channel over time. This task requires knowing the dimensions of the channel, the velocity of the water and the water depth. The depth sensing block measures the depth of the water in the channel while the flow rate sensing block tracks the water velocity and the channel dimensions are provided through user input on the accompanying website. Using two pressure transducers allows for accurate depth measurements because the relative atmospheric pressure is being measured as well as the pressure of the water at the bottom of the channel. A crucial requirement of the system is to have 90% accuracy with water flow measurements, so the sensor measurements need to be as accurate as possible to make up for errors due to signal noise and channel dimension estimation. The system relies on solar power to allow for long term operation and ease of physical relocation. This means that the power demand of sensors needs to be limited so that flow rate measurements can be read for a minimum of 12 hours of operation at a time. The Honeywell sensors require 3.3V and a nominal current of 3mA. Given the solar panel and 6600mAh battery the demand of the sensors is minimal. The irrigation channels are only a few feet deep and the sensors can measure pressures up to 25 PSI. The average air pressure is 14.7 PSI. This allows for measurements up to 20 feet deep which is more than enough for the irrigation channels that the system will be installed in.

4.6.4 Interface Validation

Table I: otsd_dpth_snsng_envin: Input

Interface Property	Explanation	Verification
Other: 0 - 25 psi	The pressure of the water will lie in this range.	This range is the limit of the sensor.

Table II: pcb_ntgrtn_dpth_snsng_dcpwr: Input

Interface Property	Explanation	Verification
Inominal: 2.8 mA	This is the nominal current draw of the sensor.	The microcontroller can supply up to 7 mA per pin.
Ipeak: 3 mA	This is the peak current drawn by the sensor.	The microcontroller can supply up to 7 mA per pin.
Vmax: 3.5 V	This is the maximum supply voltage limitation of the sensor.	The microcontroller supplies 3.3V per pin.
Vmin: 2.5 V	This is the minimum supply voltage limitation of the sensor.	The microcontroller supplies 3.3V per pin.

Table III: dpth_snsng_mcrcntrllr_dsig: Output

Interface Property	Explanation	Verification
Other: I2C protocol	This is the communication protocol used by the sensor.	The microcontroller supports I2C communication.
Vmax: 3.3 V	This voltage represents a digital "high" value.	The microcontroller accepts 3.3V digital signals.
Vmin: 0V	This voltage represents a digital "low" value.	The microcontroller accepts 3.3V digital signals.

4.6.5 Verification Process

The following are steps to verify that the block meets the depth sensing requirements of the system as described in the interface tables above. The verification process will test that the block can communicate readings given the DC power interface properties listed in table 2.

1. Connect arduino to computer.
2. Connect the data pins of the sensor to the arduino using jumper wires.
3. Connect the bench power supply to the sensor's Vdd and GND terminals.
4. Turn on the power supply to supply Vmin of 2.5V to the sensor.
5. Flash the arduino code.
6. Measure the sensor readings on the serial monitor for Inominal and Ipeak currents.
7. Switch the power supply to Vmax of 3.5V and repeat steps 5 and 6.

The test passes if the psi printed to the serial monitor is accurate over the range of DC power inputs.

4.6.6 References and File Links

[1] "USE OF SUBMERSIBLE PRESSURE TRANSDUCERS IN WATER-RESOURCES INVESTIGATIONS," pubs.usgs.gov. [Online]. Available: https://pubs.usgs.gov/twri/twri8a3/#N10009(accessedMar.15,2023).

[2] "MPR Series MicroPressure Board Mount Pressure Sensors," Honeywell. [Online]. Available: https://prod-edam.honeywell.com/content/dam/honeywell-edam/sps/siot/en-us/products/sensors/pressure-sensors/board-mount-pressure-sensors/micropressure-mpr-series/documents/sps-siot-mpr-series-datasheet-32332628-ciid-172626.pdf?download=false. [Accessed: 14-Mar-2023].

4.6.7 Revision Table

Date	Revision made
3/12/2023	Isaac Sutton: Initial subsection creation.

4.7 Enclosure

4.7.1 Description

Enclosure must be waterproof to be able to hold electronics inside without them getting wet since it will be in the canals. Gerrad (our project partner) required that the system be at least be able to handle the light rain that Imperial Valley receives every year. The only full resistance the system required was to be completely dust proof. Therefore, we will use IP65 verification to spec out the Enclosure. The reasoning for this is because IP65 rating aligns almost perfectly with what we want to accomplish. Regardless that doesn't mean I won't try to see what limits the seal and enclosure have. Given what we must store in here cautionary measures must be taken to ensure the safety and integrity of the system.

Similar to most enclosures, this one will contain and house important objects like electronics. This application specifically will house the SORACOM/Microcontroller module and the battery. A PCB will also be attached to the SORACOM module to allow an easy interfacing solution. This gives me a general idea of what the minimum size of the interior should be, and I can use that to help provide a design early in the project.

4.7.2 Design

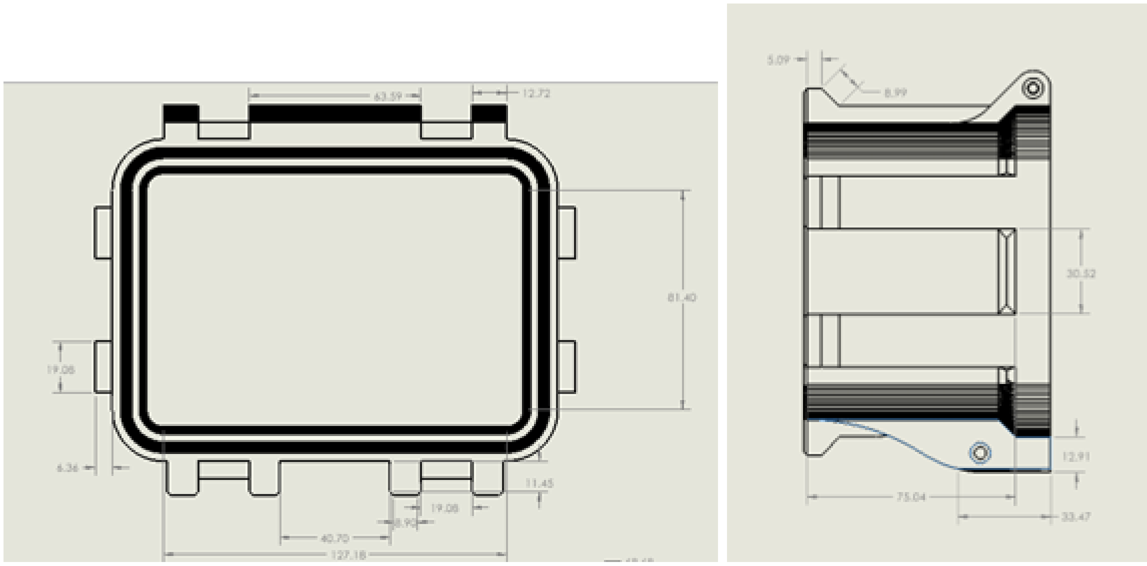


Figure 9: Left: Top View of enclosure base (mm) Right: Left View of base (mm).

Since the block will be exposed to the elements a rugged design path was taken. The design choice was then pretty straight forward. We started with a small utility box found on Thingiverse and scaled it to our desired size. Our method for manufacturing this Enclosure was through 3D printing using an Ender 3 S1 that I own, because of this the enclosure was to be printed in parts and then assembled. There are four total pieces for this enclosure. Those pieces the bottom, lid, latches, and the seal. Because of the rugged approach to this enclosure, we required to use a more durable material then the standard Polylactic Acid (commonly known as PLA). PLA is usually a good prototyping material, but it doesn't do so well when exposed to sunlight for a prolonged period of time [3] as well as not being very good in the water resistivity department[4]. Therefore, working within the constraints of my printer and with the durability we require the material it comes down to is Polyethylene Terephthalate Glycol (or known as PETG). PETG is essentially the improved version of PLA. With a higher melting point [5] and much more of a wall on the micro level then PLA. PETG gives us the more water-resistant and less prone to temperature deformation then PLA does. PETG is also recyclable therefore it's a much more sustainable material to use then PLA is.

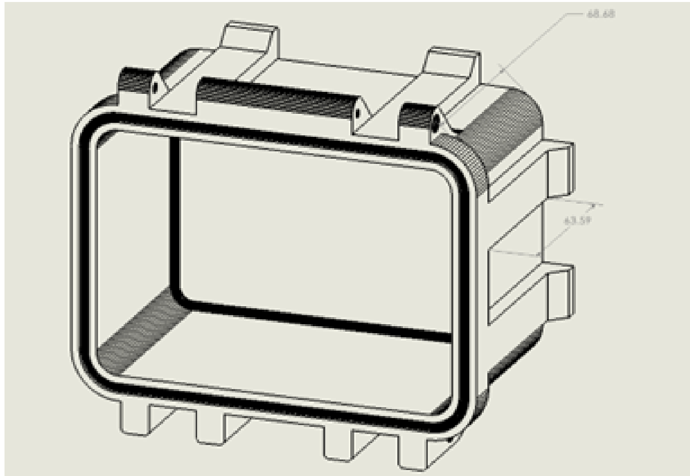


Figure 10: Diametric View of Enclosure Base (mm)

Looking at figure 9 you can see that our internal dimensions of the enclosure is 127.18 mm x 81.40 mm (5.01" x 3.21"). This should provide sufficient space to hold not just our battery but our microcontroller circuit as well. Height, wise you can see in figure 10 above, that it reaches 68.68 mm (2.7") this allows us to have enough vertical height for any wire slack and open space for heat

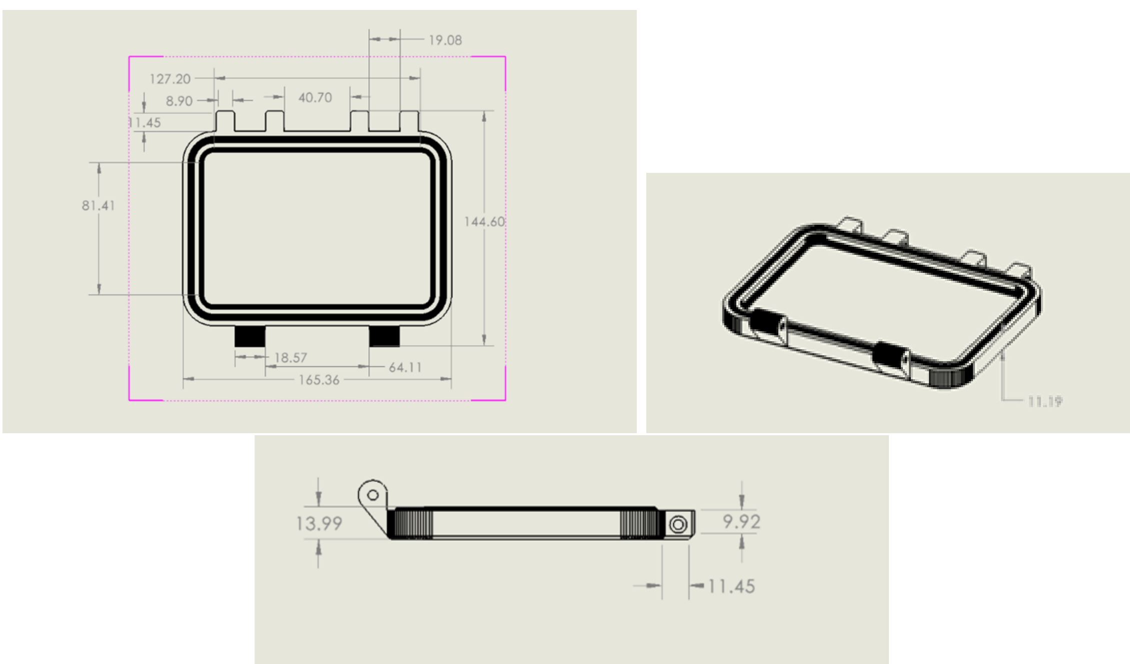


Figure 11: Enclosure Lid(mm)

Next figure is our lid for the enclosure this part was included in the original STL but just scaled to our needed spec. The lid will be what clamps down on to the bottom of the enclosure. Initially the lid was going to hold the Solar

Panel but that design was changed. Regardless the flat lid stayed simply because, there really isn't much else we can do with a lid. The lid does how ever contain the joint to connect with the bottom of the enclose to allow a hinge for the clamping mechanism. As well as some two prongs to clamp the latches in. The total dimension of the lid comes to 165.36 mm X 144.60 mm

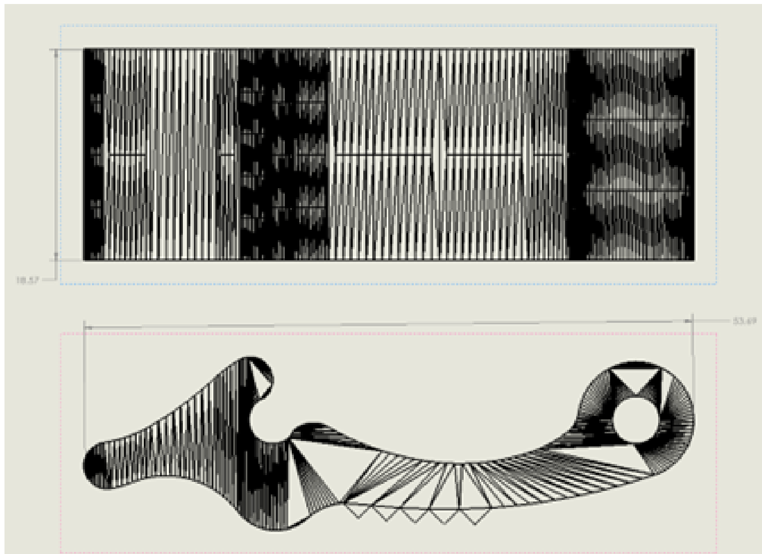


Figure 12: Latch for Enclosure Lid(mm)

The latch is a design that also came included in the original file. There are a total of two latches on this enclosure. Each being an equal dimension of 53.69 mm x 18.57mm. Both mount on the lid and then latch to a prong on the bottom part of the enclosure. The clamp design helps to apply tight pressure on the seal.

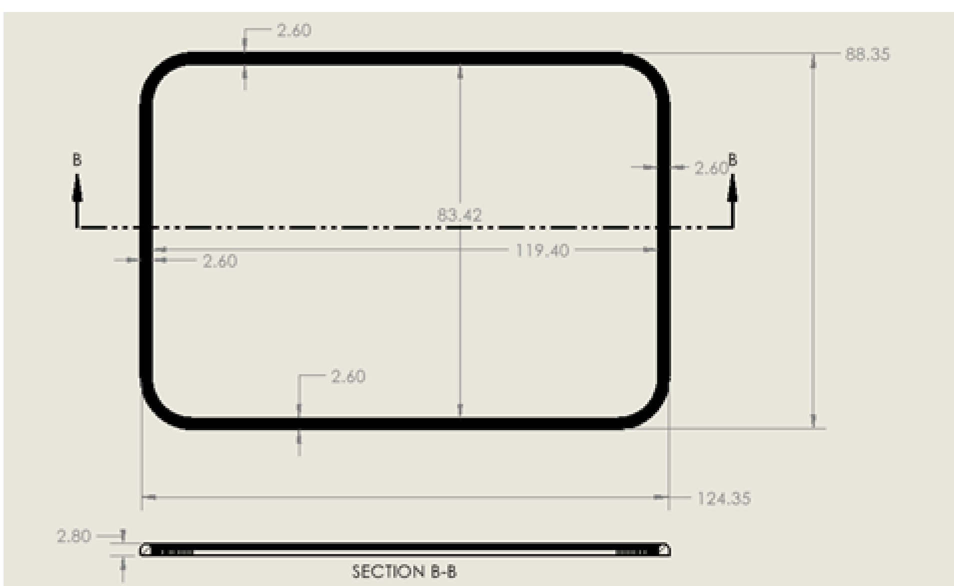


Figure 13: Enclosure Seal(mm)

Last piece of the design is the water seal, figure 13. This piece is made out of a different material then the PETG. Because the seal requires a more flexible property it was instead printed in a more rubber like filament called TPU (Thermoplastic Urethane[6]). In terms of physical properties the material is definitely a lot similar to PLA [6] but much more flexible and water resistant. The choice of material was pretty trivial when it came to picking it and it was compatible with my printer. The seal sits on the canal made on the prim of the bottom and is clamped down by the lid using the latches. The seal should be enough to hold back any water but not be able to handle complete submersion. With that concludes the summary of all the pieces of the enclosure of our projects. At the moment this seems to be all that we require in terms of functionality. Modifications will likely be made coming closer to our final looking design such as better cable routing solutions and possibly a slight redesign not make it as bulky and reduce material cost when it comes to material of the enclosure. Over all happy with what we have put together for this design and what it has accomplished.

4.7.3 General Validation

Validation for this design definitely focuses heavily on more environmental effects. Using Gerrad's requirements the enclosure should be able to handle the little rain Imperial Valley gets around the year, can be exposed to heat for long periods of time, and be dust proof. Therefore an IP65 rating was essentially the goal when it came to the water and dust, as for the heat the PETG we used will be more then effective. Aside from the environmental inputs the system does have some static connections to mount the PCB and strap the battery.

For environmental testing I will be following closely the industry standard for testing as I possibly can. When testing water resistance usually quality has a system that shoots a water jet from all angles and then inspection shortly follows. Due to the fact that the closest thing I have to a water jet is a high PSI pressure washer a hose with a simple stream should suffice. As for dust proof testing it will be a lot more less practical then what manufacturers use.

For the static connections I plan to use standoffs for the PCB and cable ties for the battery. These should keep our main components secured in the enclosure for when it's being moved. Pulling force will be calculated to get a reasonable spec.

4.7.4 Interface Validation

Table I: otsd_enclsr_envin

Interface Property	Explanation	Verification
Water: IP65 Standard	The interface was given this value because it states that the system is water resistant and dust proof. This standard aligns with our project partner's requirement.	The design was given a rivet around the edge for a rubber seal to sit that will repel any water and dust that interacts externally with the system. The enclosure is also a clam shell design so it can provide a solid pressure around the seam to maintain proper water resistance.
Other: System must be dustproof	Reasoning for this value is similar to the water resistance. IP65 rating states the systems will be dust proof. Therefore, the target will be IP65	As mentioned before, the design has a rivet around the brim of the bottom of the enclosure where a TPU seal will be clamped down with the lid to seal off any dust from the outside.
Other: Enclosure can withstand a drop from 6ft	This value was given incase of a drop from any person. Due to human error. The enclosure must be built for rugged application therefore must be resistant to the elements.	The enclosure has a good surface area and most of its weight in the columns on the outside. PETG should handle the shock well enough if it were to be dropped from 6ft.

Table II: enclsr_pcb_ntgrtn_mech

Interface Property	Explanation	Verification
Fasteners: 4 Standoff screws	The PCB will be essentially a breakout board where the battery will connect and send power to the microcontroller/SORACOM module which then goes to the sensors. Due to this situation the board will likely get hot and require airflow on the bottom. Therefore 4 standoffs are required.	4 hex nut inserts are added into the floor of the enclosure where they will be mended on to allow standoffs to be added.
Other: Dimension Restraint	The space in this enclosure will be shared between the PCB and the battery. Therefore a dimension restrain must be applied to allow both pieces to fit. 3.5" x 2.5" x 1.5" inches should be plenty of space.	When scaling the enclosure. The dimensions of the battery and PCB were taken into account and therefore the space for the two will fit.
Pulling Force: 20N	This value was calculated using this equation: $F_pull = mass * acceleration$ For mass we took into the account just the mass the SORACOM module and the enclosure itself (totals up to 2.03kg) . For acceleration I just used the gravity constant $9.8m/s^2$ and came out to roughly 19.6N	I know this value will work because the object will be secured enough to the enclosure and therefore be able to support a pull like that.

4.7.5 Verification Process

[1] IP65 Standard

1. Suspend Enclosure 3ft off the ground using fishing line to allow for water jet and dust access from any angle.
2. Spray enclosure with a hose that has a nozzle diameter of 6.3mm for 3 minutes
3. Splash enclosure 1 minute
4. Check interior of enclosure for any signs of moisture

[2] Dust Proof

1. Enclosure will remain suspended again 3ft off the ground.
2. Use a leaf blower to blow flirt in the direction of the enclosure to simulate dusty winds.

3. Open up enclosure and check for signs of dust exposure inside.

[3] Drop Test

1. Close enclosure
2. Suspend 6' off the ground
3. Drop Enclosure
4. Inspect for any damage

[4] Standoff Screws

1. Place standoffs on the 4 installed thread inserts
2. Place mock PCB on the standoffs
3. Secure PCB onto standoffs with thumbscrews
4. Flip enclosure upside down to test if it's secure or not

[5] Dimension Restraint

1. Place mock to prove we hit the measurement constraint
2. Showcase both components fitting properly together

[6] Pulling Force

1. Keeping the Mock PCB in the standoffs
2. Pull mock PCB with hand
3. Check if it's secure
4. Unit should still remain secured

4.7.6 References and File Links

[1] Filippo De Lucia - www.filippodelucia.com - Consulenza e Sviluppo Siti Web, "IP 65 enclosures rating IP IEC/EN 60529 (ip6x IPX5)," Test IP 65 Ingress Protection Water Dust IEC 60529 EMCTEST. [Online]. Available:https://www.emctest.it/index.php?l=prove_ip65&country=uk. [Accessed: 20-Jan-2023].

[2]H. H. A. K. A. www.crazyhamster.co.uk, "Ip rated enclosures explained," Find an electrical enclosure. [Online]. Available: https://www.enclosurecompany.com/ip-ratings-explained.php. [Accessed: 20-Jan-2023].

[3]Keene Village Plastics. [Online]. Available: https://www.villageplastics.com/3d-printing-materials/. [Accessed: 11-Feb-2023].

[4] "How to 3D print waterproof parts," https://ultimaker.com. [Online]. Available: https://ultimaker.com/learn/3d-print-waterproof-parts. [Accessed: 11-Feb-2023].

[5] "PETG vs PLA," Petg vs pla. [Online]. Available: https://www.flashforge.com/product-detail/petg-vs-pla. [Accessed: 11-Feb-2023].

[6] "TPU filament 3D printing material – The Complete Guide - AllThat3D," AllThat3D. [Online]. Available: https://www.allthat3d.com/tpu-filament/. [Accessed: 12-Feb-2023].

4.8 PCB Integration

4.8.1 Description

This Block is essentially what ties most of the system together. It starts by integrating the solar charging circuit that will receive DC power from the solar panel and then power the system as well as charging the battery. The circuit itself also switches to battery power once it senses lack of input externally power something else. The output of the IC splits into different paths. The first path is optional as the output of the charger IC also connects directly to the micro controller's (Arduino MKR NB 1500) VIN. The second path goes to a female connector that can power and external device. The PCB has a female header pins for the Micro controller to mount on. These headers then break out to the output Rj45 female connectors, which go out to the velocity and depth sensors as well as receive power from the solar charger circuit.

4.8.2 Design

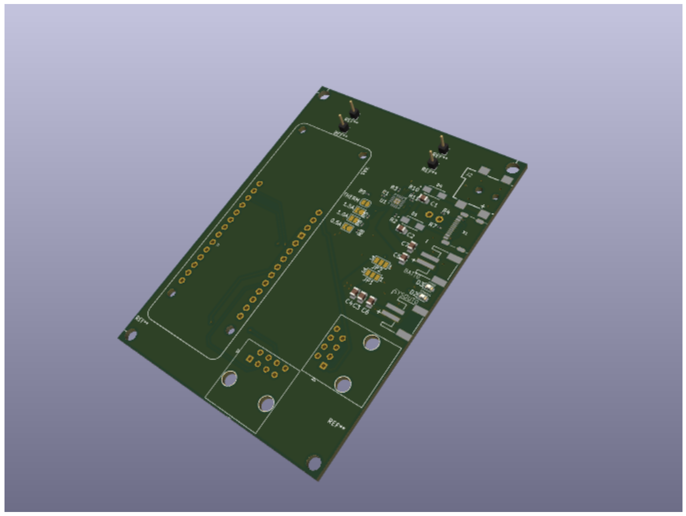


Figure 15: 3D render of Project PCB

The design for the PCB had an initial parameter that had to be followed which was a dimension constraint. With a horizontal limit of 3.5" x 2.5" this was done to allow space for a battery in the enclosure. With this initial parameter the size of the PCB was determined. The purpose of the PCB as mentioned in the description was designed to take the power generated from the solar panel and power the circuit while also charging the LIPO battery. To find out more about how the power circuit works reading section 4.6 is recommended. With the purpose of distributing power from the solar panel to the Arduino trace size became one of the first design choices to consider aside from the dimension's itself. Thanks to the existence of industry standards some of these trace widths can be approximated based off of the expected current[1]. I also had the assistance of Adafruit who was happily able to upload their solar power charger PCB schematic for creators. the files are certified as open source[2], but in a way we still supported them since we bought the solar power charger circuit to prototype the power block[3].

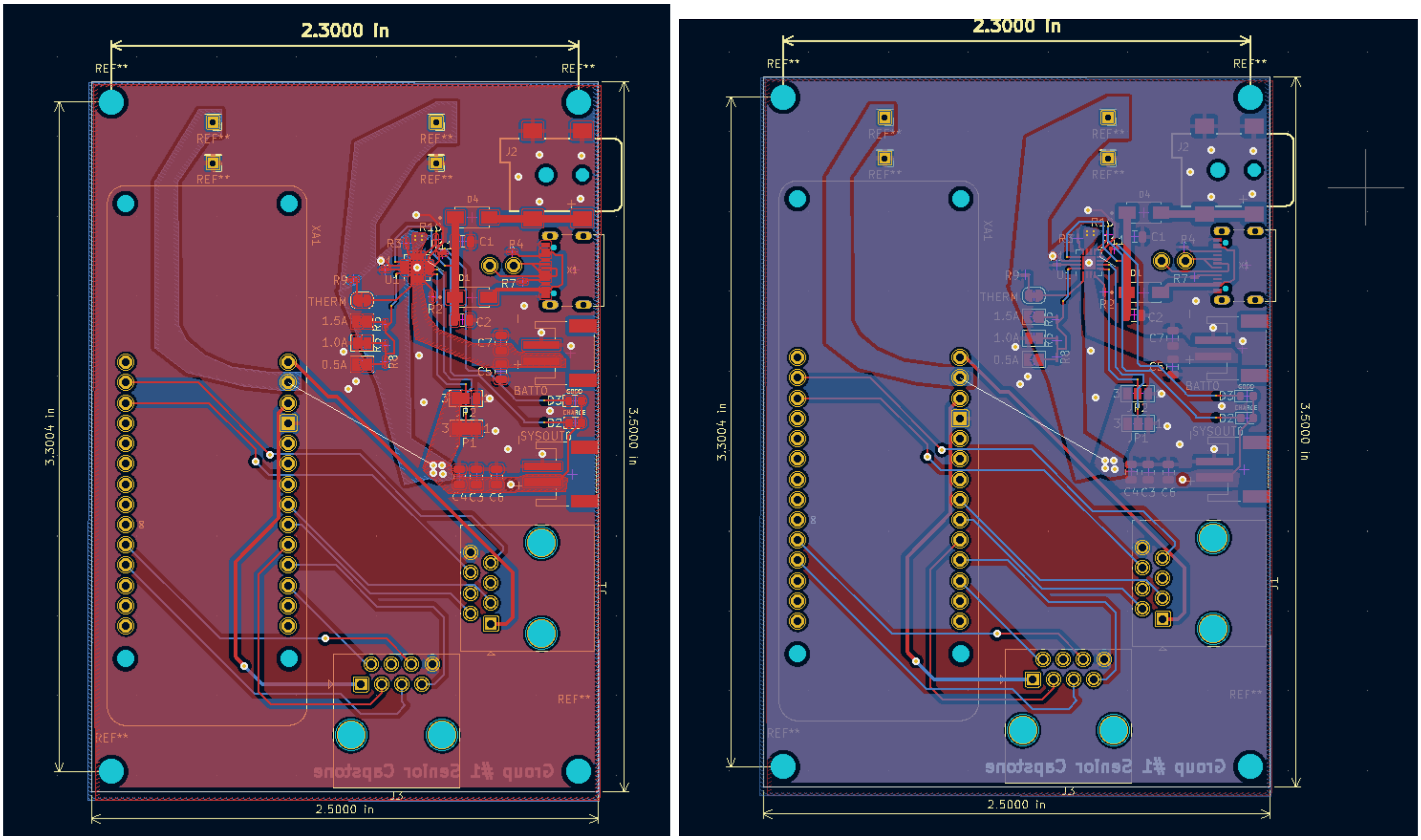


Figure 15: (Left)Top PCB Layer (Right) Bottom PCB Layer w/Dimensions (in)

With the implementation of the solar power circuit and the dimension constraint the design was pretty straightforward. Most of the routing was also straightforward. the path of the charging circuit's output was for the power to go through the solar circuit to the mounted boost converter that raised the voltage to 5V where it then went to the Arduino's Vin pin. You'll notice in the top layer in figure 15 that it shows a missing connection between and Arduino pin and one of the output thermal vias. This does get connected when the external boost converter is attached which completes that circuit.

4.8.3 General Validation

For validation of this block it will primarily be safety checks for the assembled board to make sure there are no shorts. Pre-assembly the board will be inspected to make sure all traces are correct with continuity checks as well. Once board is assembled and checked for shorts we will connect it to the power supply and check for voltages to make sure we have the correct voltages in the areas we need it to be. The circuit should also be charging an external battery which we will test after checking initial voltages to ensure spontaneous combustion does not occur.

Once voltages and battery is safe the micro controller Will be mounted in the form similar to Arduino shields. Where it has mating pins for the Arduino's female headers. This design allows for an easy tear down of the system in case of hardware malfunction or retiring the system all together.

With the PCB fully assembled including battery, Arduino, and solar panel, the system should show expected behavior. Which will be verified through metering the board while it runs. if our voltages and current are all were it needs to be then it the board is finished and ready for the final build, thus concluding verification.

4.8.4 Interface Validation

Table I: pcb_ntgrtn_mcrcntrllr_dcpwr

Interface Property	Explanation	Verification
Inominal: 100mA	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the idle current used by the board when it's not sleeping. this is primarily what the battery capacity for the system is modelled around.
Ipeak: 190mA	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the highest current the board will draw. We will likely get this number when the board transmits over cellular communication.
Vmax: 7V	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This voltage rating is the maximum the Arduino's voltage regulator can handle. The system should never receive spikes this high.
Vmin: 5V	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the minimum operating voltage for the Arduino's CPU

Table II: pcb_ntgrtn_vlcty_snsr_dcpwr

Interface Property	Explanation	Verification
Inominal: 4mA	This is the ideal current usage when unit is in idle	The system's high power-draw ICs have their power supplies as a switch toggle.
Ipeak: 7mA	Max output of the Arduino's digital I/O pins	The sensor's ICs use less than this therefore the value may need to be adjusted soon
Vmax: 5V	the sensor's oscillator cannot handle more then 5V	Max voltage that the Arduino outputs is 5V
Vmin: 3V	This value is arbitrary, but any lower would limit the range of the transducer pair	All ICs on the sensor have a Vmin of 2.5V

Table III: pcb_ntgrtn_dpth_snsng_dcpwr

Interface Property	Explanation	Verification
Inominal: 100mA	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the idle current used by the board when it's not sleeping. this is primarily what the battery capacity for the system is modelled around.
Ipeak: 190mA	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the highest current the board will draw. We will likely get this number when the board transmits over cellular communication.
Vmax: 7V	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This voltage rating is the maximum the Arduino's voltage regulator can handle. The system should never receive spikes this high.
Vmin: 5V	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the minimum operating voltage for the Arduino's CPU

Table IV: enclsr_pcb_ntgrtn_mech

Interface Property	Explanation	Verification
Fasteners: 4 Standoff screws	The PCB will be essentially a breakout board where the battery will connect and send power to the microcontroller/SORACOM module which then goes to the sensors. Due to this situation the board will likely get hot and require airflow on the bottom. Therefore 4 standoffs are required.	4 hex nut inserts are added into the floor of the enclosure where they will be mended on to allow standoffs to be added.
Other: Dimension Restraint	The space in this enclosure will be shared between the PCB and the battery. Therefore a dimension restrain must be applied to allow both pieces to fit. 3.5" x 2.5" x 1.5" inches should be plenty of space.	When scaling the enclosure. The dimensions of the battery and PCB were taken into account and therefore the space for the two will fit.
Pulling Force: 20N	This value was calculated using this equation: $F_pull = mass * acceleration$ For mass we took into the account just the mass the SORACOM module and the enclosure itself (totals up to 2.03kg) . For acceleration I just used the gravity constant $9.8m/s^2$ and came out to roughly 19.6N	I know this value will work because the object will be secured enough to the enclosure and therefore be able to support a pull like that.

Table V: slr_nd_bttry_pcb_ntgrtn_dcpwr

Interface Property	Explanation	Verification
Inominal:	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the idle current used by the board when it's not sleeping. this is primarily what the battery capacity for the system is modelled around.
Ipeak: 190mA	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the highest current the board will draw. We will likely get this number when the board transmits over cellular communication.
Vmax: 7V	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This voltage rating is the maximum the Arduino's voltage regulator can handle. The system should never receive spikes this high.
Vmin: 5V	defined in the electrical characteristics section of the Arduino MRK 1500 datasheet	This is the minimum operating voltage for the Arduino's CPU

4.8.5 Verification Process

4.8.6 References and File Links

[1]Mcl, "PCB trace width calculator: Trace width vs. current table," mcl, 30-Mar-2022. [Online]. Available: https://www.mclpcb.com/blog/pcb-trace-width-vs-current-table/. [Accessed: 14-Mar-2023].

[2]C. Nelson, "Accessing and using Adafruit PCB design files," Adafruit Learning System. [Online]. Available: https://learn.adafruit.com/accessing-and-using-adafruit-pcb-design-files?view=all. [Accessed: 14-Mar-2023].

[3]B. Siepert, "Adafruit Universal USB / DC / solar lithium ion/polymer charger - BQ24074," Adafruit Learning System, Sep-2020. [Online]. Available: https://learn.adafruit.com/adafruit-bq24074-universal-usb-dc-solar-charger-breakout/downloads. [Accessed: 14-Mar-2023].

5 System Verification Evidence

5.1 Universal Constraints

5.1.1 The system may not include a breadboard

The system does not use any breadboards. The electronics housed in the enclosure are integrated on a PCB and the sensors submerged in the water are also installed on their own PCBs. All wired connections and interfaces use connectors.

5.1.2 The final system must contain a student designed PCB

Isaac G. champions a PCB integration block, for dispersing power from the Solar and Battery block to the microcontroller, and for handling connectors between the microcontroller and the two sensors blocks. It also surpasses the requirement on the number of pads.

5.1.3 All connections to PCBs must use connectors

The PCB is designed with this requirement in mind. It has no wires directly soldered to it. It includes RJ45, mini USB, male/female headers, 3.5mm DC jack and 2-pin JST PH connectors.

5.1.4 All power supplies in the system must be at least 65% efficient

This is taken into account *after* the Solar panel, since solar cells are typically only around 12% efficient. A bench power supply will be used to test input voltages to the power supply block that are representative of what the solar panel would provide in daylight conditions. The charging and supply portion of this block is tested to be greater than 65% efficient.

5.1.5 The system may be no more than 50% built from purchased 'modules.'

There is one block in this project that consists of pre-purchased modules: the Depth Sensors. Everything else is custom-made:

- The Microcontroller is a firmware block.
- The Website is a code block.
- The Cellular Connection block is a code/API block.
- The Solar and Battery block has been subsumed into the PCB and customized.
- The Velocity Sensor is a custom hardware block.
- The PCB block is evident.

- The Enclosure is custom-made as well.

5.2 Requirements

5.2.1 Budget

5.2.1.1 Project Partner Requirement: "The unit should be somewhat cheap to reproduce."

5.2.1.2 Engineering Requirement: The system will cost less than five hundred dollars.

5.2.1.3 Verification Process: The entire Bill of Materials will be provided, alongside the MSRP of each component. These retail prices will be added up, and should be less than or equal to $500.

5.2.1.4 Testing Evidence:

Table 1: Generated by Spread-LaTeX

Part Number	Description	Quantity	Unit Price	Cost
N/A	Digikey PCB components	1	12	12
5366	6V 2W Solar Panel - ETFE - Voltaic P126	1	19.95	19.95
353	Lithium Ion Battery Pack - 3.7V 4400mAh	1	19.95	19.95
2788	3.8 / 1.3mm or 3.5 / 1.1mm to 5.5 / 2.1mm DC Jack Adapter Cable	1	1.5	1.5
N/A	PVC pipes	1	20	20
N/A	Plate Weight	1	12.5	12.5
MPRLS0025PA00001AB	MICROPRESSURE EVAL BOARD 25 PSI	1	40.17	40.17
N/A	Overture TPU Filament 1.75mm Flexible TPU Roll Soft 3D Printer	1	2.79	2.79
N/A	Overture PETG 3D Printer Filament 1.75mm, 1kg Filament	1	7.99	7.99
N/A	MKR NB 1500 SORACOM LTE	1	89.9	89.9
SMFM21F1000	Ultrasonic Flow Sensors, 1MHz	2	9.98	19.96
OPA2830IDR	IC Voltage Feedback 2 Circ 8SOIC	1	3.7	3.7
MAX7375AXR105+T	MEMS OSC XO 1.0MHz CMOS SMD	1	1.64	1.64
MAX11613EUA+T	IC ADC 12BIT SAR 8UMAX	1	4.11	4.11
NX5032GA-8.0M-STD-CSU-1	CRYSTAL 8.0MHz 8PF SMD	1	0.69	0.69
SA612AD/01,112	NXP Mixer	1	3.125	3.125
N/A	Custom System PCB	1	29.17	29.17
CL21A335KPFNNNG	3.3uF Capacitor	3	0.21	0.63
885012207083	330pF Capacitor	3	0.1	0.3
CL21B104KBCNNNC	100nF Capacitor	3	0.1	0.3
CL21B223KBANNNC	22nF Capacitor	1	0.1	0.1
SR1-0805-233	3.3k Ohm Resistor	1	0.02	0.02
RNCP0805FTD1K00	1k Ohm Resistor	5	0.1	0.5
RC0805JR-0712KL	12k Ohm Resistor	1	0.1	0.1
RC0805JR-07330RL	330 Ohm Resistor	2	0.1	0.2
RC0805JR-0710KL	10k Ohm Resistor	1	0.1	0.1
Total BOM Cost				291.40

5.2.2 Data Transmission

5.2.2.1 Project Partner Requirement: "Must use 4G-5G communication technology."

5.2.2.2 Engineering Requirement: The system will wirelessly transmit data in an area without Wi-Fi to a range of at least 1km displaying it to the user.

5.2.2.3 Verification Process:

1. Turn on the system.
2. Collect data with the main system at a nearby stream such as oak creak.
3. Have another team member at a set distance away from the main system attempt to access the data online.

5.2.2.4 Testing Evidence:

5.2.3 Enclosure

5.2.3.1 Project Partner Requirement: "Must be able to stay waterproof/water resistant."

5.2.3.2 Engineering Requirement: The system must stay water resistant and follow IP65 guidelines.

5.2.3.3 Verification Process:

1. Suspend enclosure 3ft off the ground using fishing line to allow for water jet and dust access from any angle.
2. Begin with throwing sand from every angle and surface of the enclosure.
3. Spray enclosure with a hose that has a nozzle diameter of 6.3mm for 3 minutes.
4. Splash enclosure for 1 minute.
5. Check interior of enclosure for any signs of moisture or sand.

5.2.3.4 Testing Evidence:

5.2.4 Measurements

5.2.4.1 Project Partner Requirement: "Measure passing water over time."

5.2.4.2 Engineering Requirement: The system will calculate instantaneous volume flow over time through the entirety of an open water channel and record it with a instantaneous accuracy of 90%.

5.2.4.3 Verification Process: The video will show obtained readings from both sensor types in a debug window of the microcontroller, as well as show calculation results for water volume. The reading frequency will be sped up for the purpose of the demonstration.

5.2.4.4 Testing Evidence:

5.2.5 PC Backend

5.2.5.1 Project Partner Requirement: "Collected system data must be assessable online."

5.2.5.2 Engineering Requirement: The system will have a program written and hosted online that will communicate with the main system using wireless communication to store/sort all collected data up to six prior months.

5.2.5.3 Verification Process:

1. Turn on the system.
2. Collect data with the main system at a nearby stream such as oak creek.
3. Have another team member at a set distance away from the main system attempt to access the data online.
4. Verify data is collected properly and stored properly by shutting the main system down and refreshing the website to make sure data is saved.
5. Verify if the system collects multiple data points so it will sort data properly.

5.2.5.4 Testing Evidence:

5.2.6 PCB

5.2.6.1 Project Partner Requirement: N/A

5.2.6.2 Engineering Requirement: PCB must be able to handle up to 12V in power and regulate proper amount of power to the certain components (Microcontroller, sensors, etc.)

5.2.6.3 Verification Process: Using some test points in the board we will meter the outputs of regulators and inputs of components to ensure nothing will get fried.

5.2.6.4 Testing Evidence:

5.2.7 User Interface Alerts and Analytics

5.2.7.1 Project Partner Requirement: "The system will display analytics for water flow rate and quantity through the user interface."

5.2.7.2 Engineering Requirement: The system will display updated water flow analytics every five minutes through the smartphone-accessible user interface.

5.2.7.3 Verification Process:

1. Wirelessly send alerts from the system.

2. Use a stopwatch to time how long it takes for the UI to display the alert.
3. Wirelessly send the updated flow rate data from the system.
4. Use a stopwatch to time how long it takes for the UI to display the updated analytics.

5.2.7.4 Testing Evidence:

5.2.8 Wireless Power Source

5.2.8.1 Project Partner Requirement: "The system should be able to stay powered on without a wired connection to a power source."

5.2.8.2 Engineering Requirement: The system must have sufficient battery capacity to operate continuously for up to 12 hours without sunlight.

5.2.8.3 Verification Process:

1. Charge the battery entirely.
2. Check every 30 minutes for a total of 12 hours to see if the system is still operating.

5.2.8.4 Testing Evidence:

5.3 References and File Links

5.3.1 References (IEEE)

5.3.2 File Links

5.4 Revision Table

Date	Revision made
10/13/2022	ALL: Initial Document Creation
3/12/2023	Cameron: Rushed and uneducated completion!
3/14/2023	Isaac S: Updated Constraints and added Bill of Materials

6 Project Closing

6.1 Future Recommendations

The continuation of this project in future years should focus on the refinement of the current design as well as expanding the scope for both larger scale data collection and automation of the irrigation gates. Maintaining contact with the farmers who will be using the product is crucial to better understand their specific needs. The goal is to provide data collection and potentially automation to aid in more efficient water use without burdening the user financially or requiring too much of their time when deployed.

6.1.1 Technical recommendations

A large recommendation carried by all team members is a change in environment sensing methods. Theoretically, both velocity and depth sensing could be performed by the same ultrasonic sensor, which could in turn heavily cut back communication protocol overhead and reduce exposure of fragile components to wet conditions.

The team went down the line with the stakeholder's desires for the types of sensors implemented, but something of a similar purpose could possibly be made with a pre-made mechanical fin velocity sensor and some software tweaks for low velocity conditions. This would further reduce the cost of each unit, but as a trade-off it would be less durable and require more depth-velocity calibration.

As a further extension of this project, a network of these systems could be created, each fit with output signals for mechanical gates to open and close remotely. Individual units could either relay to one another with long range radio or all interface with the website under unique identifiers. This would in theory let irrigation managers forgo the entire process of riding out to gate sites, provided it is reliable enough. Additionally, the units could be changed from deployable units to static ones attached to irrigation gates: mechanical flow readings at this point are likely to be more accurate, while an ultrasonic reading may decrease in accuracy due to high turbulence. This would eliminate errors stemming from improper stream orientation as well.

A more practical recommendation would be to survey the type of channel the system is to be used on before the design process. Since this system was tailored for a specific group in a different state, the irrigation channels were never actually witnessed by any of the members of this team, which led to some design ambiguity. Knowing the precise application range could prove to yield opportunities for optimization.

6.1.2 Global impact recommendations

A system such as this could be used for a variety of other purposes, and its price and replicable design would make for an easy time in re-purposing. For example, the ultrasonic sensor was designed after similar sensors used in sewer systems, and thus could be scaled down for use inside of pressurized pipes or other construction contexts.

Furthermore, the enclosure is a rugged design tested for IP65 standards and routine physical shock. Any number of sensors could interface with this unit, making it a powerful Internet of Things relay for outdoor analytical field work of any kind.

6.1.3 Teamwork recommendations

An improvement on team communication for a project such as this would be continual use of collaborative version control software, as opposed to this group's approach, which was a post hoc compilation of project information. This would lead to a more unified design, more asynchronous collaboration time, and more transparent documentation for those who wish to reproduce the project.

Another suggestion would be to have a more organized approach to task delegation and project management. By regularly checking in on progress towards milestones can help to ensure that the project is on track and any potential issues get identified early on. Additionally setting up regular team meetings can facilitate the exchange of ideas and keep everyone informed and up to date on changes to the project. With these strategies the team can works more efficiently and effectively towards their project goals.

6.2 Project Artifact Summary with Links

Part of this project's purpose is to be easily replicable, and as such this team has made the design open source. Section 6.4.2 of this document contains two links: one being a repository for all website code, firmware, PCB .gbr files, and enclosure .stl files, and the other being the project website, which is to contain more detailed steps and a copy of the bill of materials. This information is held on these platforms, as opposed to on this document, because the changing nature of the project demands an environment that is friendly towards persistent updates.

On top of this information, the schematics for the two PCBs follow:

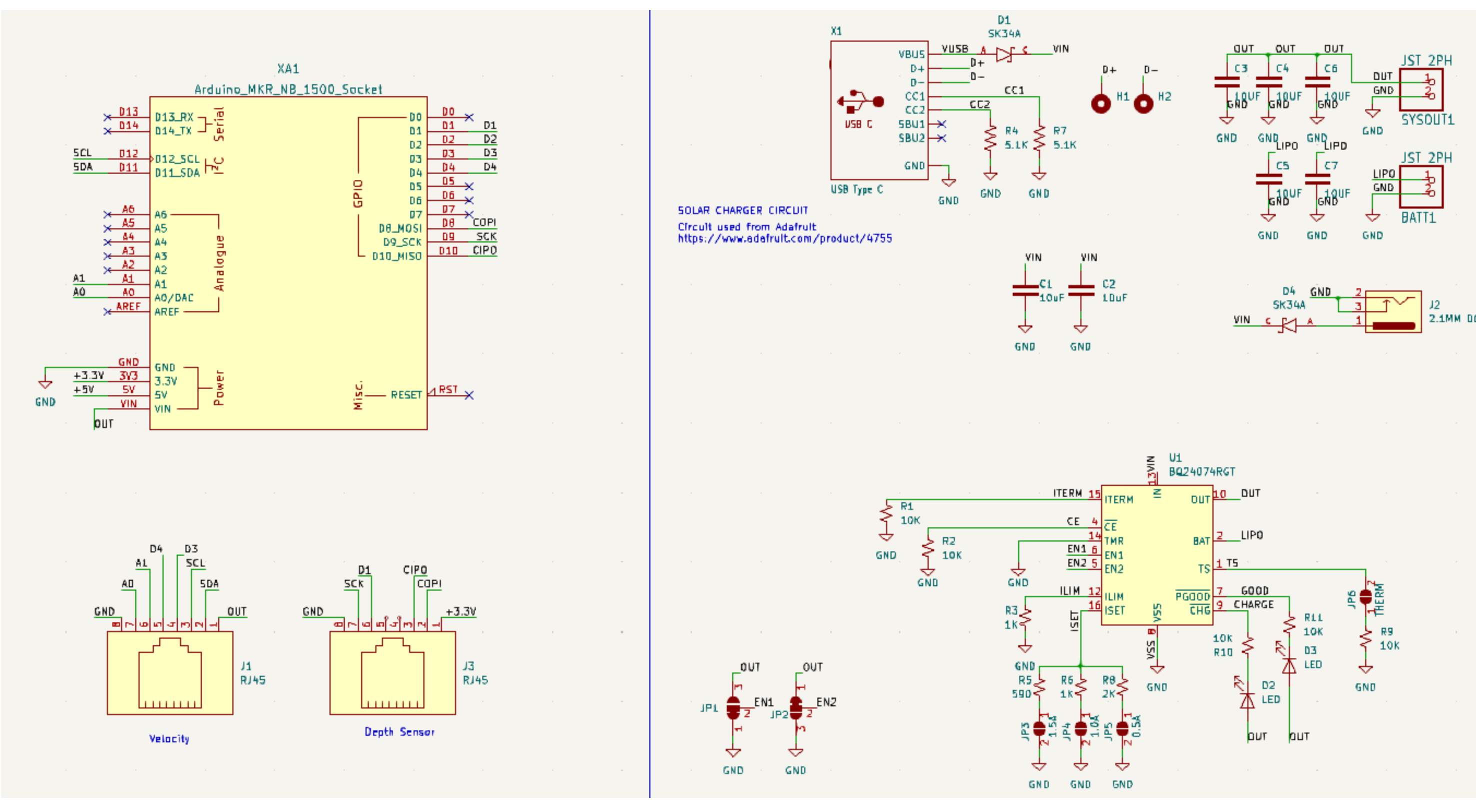


Figure 6.1: Central PCB for power and microcontroller integration.

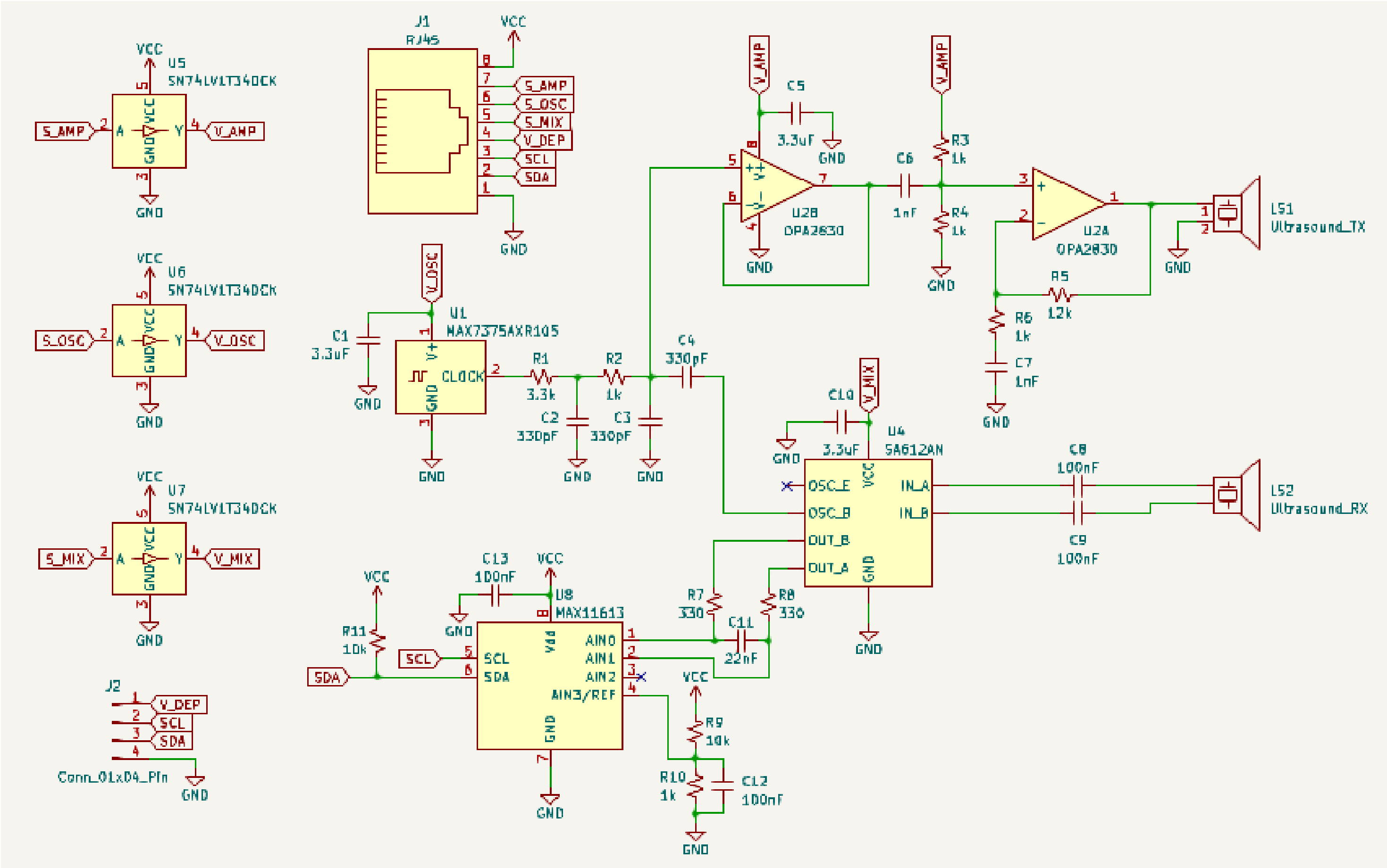
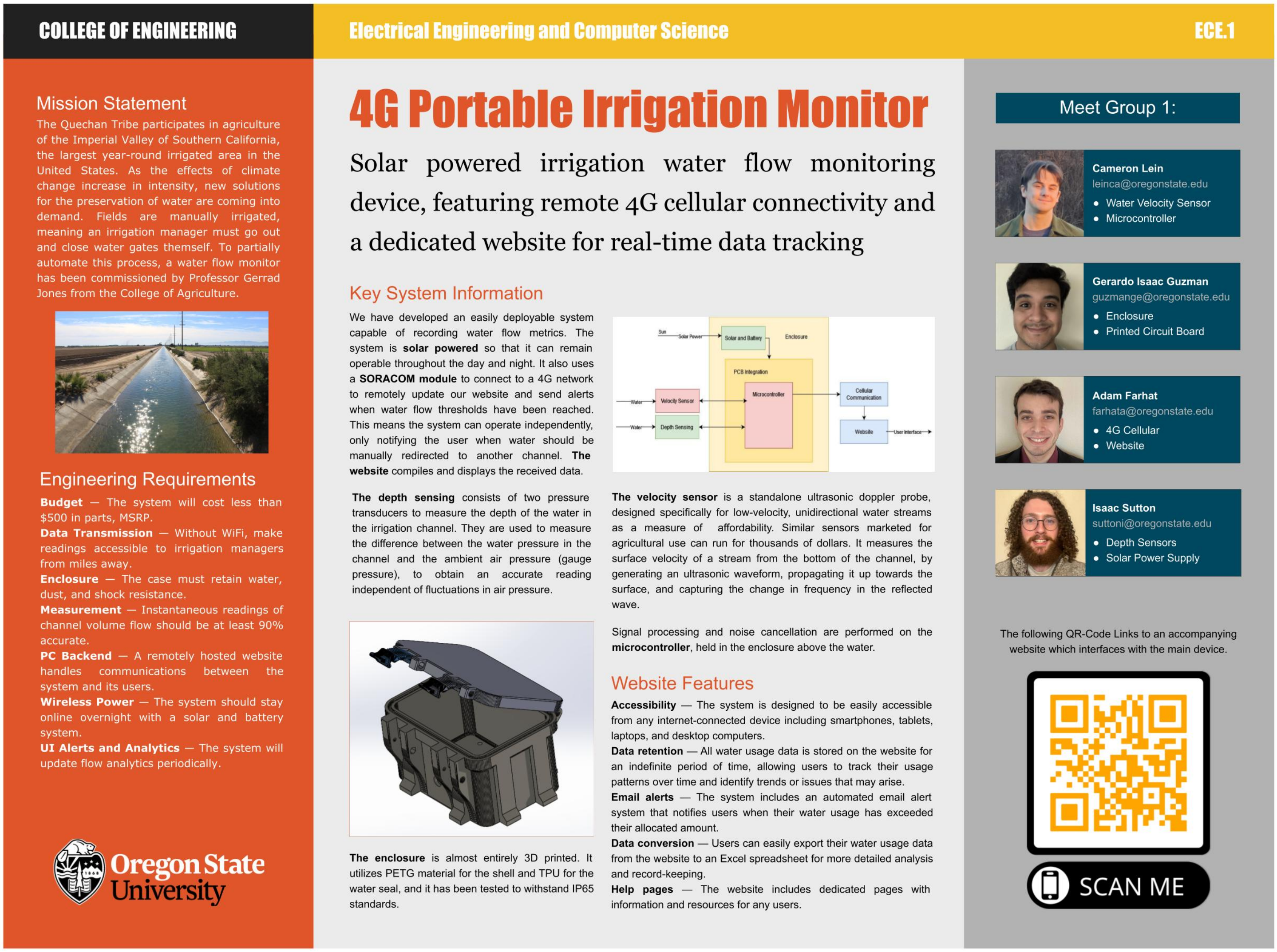


Figure 6.2: Ultrasonic velocity sensor PCB.

6.3 Presentation Materials

Below is the poster the team is to use for the expo.



The project showcase for this year's iteration can be found here [insert link later when created].

6.4 References and File Links

6.4.1 References (IEEE)

[1] A. York, "10 Project Management Best Practices You Must know," Project and Team Management Software, 08-Nov-2021. [Online]. Available: https://www.teamwork.com/blog/project-management-best-practices/. [Accessed: 28-Apr-2023]. [2] "Considering social impact when engineering for Global Development." [Online]. Available: https://scholarsarchive.byu.edu/etd/9158/. [Accessed: 29-Apr-2023]. [3] "The perfect poster," American Psychological Association. [Online]. Available: https://www.apa.org/gradpsych/2011/01/poster. [Accessed: 28-Apr-2023]. [4]

6.4.2 File Links

Project Github containing relevant code (and soon PCB files): https://github.com/Adamhat/IrrigationDevice

Website "About" Page with further replication details: http://quechan2023.pythonanywhere.com/

6.5 Revision Table

Date	Revision made
10/13/2022	ALL: Initial Document Creation
4/28/2023	ALL:

7 Appendix