Water Flow Sensor

October 10th, 2022

Project Specification Document

Carlos Beleno, Christian Ovchinikov, Mohammed Al-Khalifa

Abstract

The Smart Water Flow Sensor project aims to monitor and isolate leaks in household water pipes, helping families reduce water wastage. The system uses a water flow sensor to collect data on water flow through pipes, which is then processed by a microcontroller and transmitted to an HTML page via Wi-Fi. The water consumption data is also displayed in near real-time on an LCD screen attached to the product. A smartphone application is designed for easy daily consumption monitoring.

The project is divided into multiple blocks, including the water flow sensor, Arduino Uno microcontroller, Wi-Fi module, and power supply, all housed in a robust enclosure. The product design is being critically assessed by the team members to ensure a high-quality outcome. The electronic components are tested through a breadboard prototype before incorporating them into the final PCB design. The team simultaneously develops the software, including microcontroller code for data processing, calculations, LCD output, and user-controlled buttons, as well as a computer program that provides an interface for users to monitor their household's water consumption.

The Smart Water Flow Sensor project provides a comprehensive solution for reducing water wastage in households by detecting leaks and monitoring water consumption, promoting sustainable water usage, and providing real-time feedback to users.

Table of Contents

1. Overview

1.1. Executive Summary

According to the United States Environmental Protection Agency, the average family wastes 9,400 gallons of water annually due to leaky pipes [1]. A smart water flow sensor was proposed to monitor and help isolate leaks in household water pipes. A water flow sensor will be used to gather data on the desired line and then stored in the system. The microcontroller will transmit that data using a Wi-Fi signal to an HTML page. The output data will also be given in near real-time on an LCD that will be located on the product itself. The smartphone application will be designed to check daily consumption and monitoring. Since household water lines are pressurized, the only time flow should be detected is when a water fixture is on, or there is a leak.

A water flow sensor is needed to calculate the amount of water flowing through the pipe and measure the flow of water. We need to make sure that we achieve our project without having any issues with testing each block. We will have to use 8 or 9 blocks for our project design. We need to develop our skills in engineering such as coding and designing circuits. We need to have knowledge of programming languages to adapt with technology. We will use Arduino uno for our project in which we will connect the water sensor directly to the Arduino. Also, we need to have a good enclosure to contain all the components, such as HMI, PCB, microcontroller, and power supply.

The design of the product is being critically assessed by each of our team members. This will give us a better understanding of the design and a better finalized product. Each step in the project will be reviewed and revised as needed to ensure that the final product is robust. The electronic portion of the project will be built and tested through a breadboard prototype. This will allow the team to also design and test circuits to incorporate into the PCB design. The circuit will include the integration of the Wi-Fi module, microcontroller, AC-DC converter, DC-DC converter, LCD display and user buttons.

The team will also be working on the software of the project simultaneously to ensure that flashing and testing can begin once a prototype is built. The microcontroller code will be in testing stages first to start the on-board logic that needs to occur. This includes reading data from the sensor, storing a limited amount of data, parsing that data, doing calculations, output to the LCD display, send data over network, and allow for user-controlled buttons that change how the system works. The computer program will begin testing once the microcontroller firmware has been built to a point where data can be parsed and sent out to the network that it is connected to. This data will be displayed in a computer program that gives the user an interface to view this data and monitor their household's water consumption.

1.2. Team Contacts and Protocols

Carlos Beleno
belenosc@oregonstate.edu
541-740-8667
Roles: 3D Modeling, Power Supply
Christian Ovchinikov
ovchini@oregonstate.edu
503-979-9174
Roles: Microcontroller Programming, HMI Module
Mohammed Al-Khalifa
alkhalm4@oregonstate.edu
541-224-2453
Roles: PCB Design, Water Flow Sensor

TABLE I
TEAM PROTOCOLS

Topic	Protocol	Standard
Document Management and Collaboration	The team will use OneDrive for document management and collaboration.	Documents and assignments for the project will be created and stored in the shared OneDrive. This will allow all the team members to collaborate on the documents together.
Deadlines	The team will use a shared Google Calendar for storing deadlines for the project.	All project deadlines have been inputted into a shared Google Calendar that is also stored in the shared OneDrive. Calendar notifications will be turned on to notify the team of upcoming deadlines.
In-Person Meetings	The team will meet at Dearborn Hall – RM211 @ Noon for an hour, weekly outside of regular meeting times.	During these meetings, the team will discuss progress, and questions, and work together on portions of the project with access to the campus lab.
Online Meetings	The team will meet as necessary online via Zoom or Discord.	During these meetings, the team will discuss progress, and questions, and work together on portions of the project.
Online Check-Ins	The team will meet weekly via Zoom or Discord on Sundays @ Noon for a half-hour check-in.	During these meetings, the team will be able to discuss progress, questions, and upcoming deadlines.
Task Management	The team will check in via Discord on the day of the deadline to verify the completion of group assignments.	The team will be in communication with each other on group assignments to ensure that they are completed by the corresponding deadlines.
Treasurer	Allison will oversee contacting the professor for any billing for materials bought for the project.	Be in communication with the group to know when to buy the materials within an acceptable period.

The project is based on a product competition, thus there is no project partner. Instead, our project partner is the "customers." This means that our team will have to do research through family, peers, and online resources to get a perspective of what a "customer" might need in our product.

1.3. Gap Analysis

The reason our project exists is to provide an affordable method of determining the flow of water through a pipe. The project would detect leaks in water lines to homes which could help reduce the 900 billion gallons of water that are wasted annually. A key issue the product will address is not only the amount of water leaked but the time required to detect the issue [3]. This system is intended to be for a singular pipe but could be scaled up into a fully centralized smart water manifold system. Our project is assuming that there is a desire to reduce wasted water in houses, irrigation lines, and any other water line that matches the project's size constraint. Another assumption is that there is a need to see averaged data from the water flow sensor directly on the user's smartphone.

From discussions with peers, we have gathered that there is some need for a simple and cost-effective solution in determining leaks in piped water systems. Some homeowners cannot afford to pay a plumber for a diagnosis. Some leaks are not easily detected such as drippy water fixtures, constantly running toilets, and hairline fractures embedded in the house's walls. One interviewee experimented for one month when they were on vacation. They stated that their water consumption provided by the water company was still in the thousands of gallons. This is an indication of a leak that is wasting water and the interviewee's money.

While researching, there were a few other products that we could learn from for our own project. One of them being the Moen Flow Smart Water Leak Detector and Automatic Water Shut off Valve. While this item can be bought at Home Depot, this is one example of a product that is not cost effective. You would have to spend almost $500 for this sensor that connects to smart app through Wi-Fi. Another project we found that could be helpful was an "Arduino Water Flow Sensor to Measure Flow Rate & Volume". The project uses some of the same components.

Our product could provide a cost-effective solution that homeowners could use to forestall any future or hidden leaks. There are some products readily available online that provide most of our projects' features. However, the available product costs the same if not more than what a diagnosis by a certified plumber would cost. The end user of the product will be someone that has a possible water leak to determine if there is flow in a pressurized system that flow should be absent in

1.4. Timeline/Proposed Timeline

Project Timeline

Project Title: Smart Water Flow Sensor

Project Managers: Allison Galdamez, Carlos Beleno, Christian Ovchinikov, Mohammed Al-Khalifa

Phase I: Design (Fall Term, weeks 1–10, Break) | Phase II: Build & Test (Winter Term, weeks 11–22, Break) | Phase III: Polish (Spring Term, weeks 20–31)

Task #	Task Title	Task Owner	Deliverable	Start Date	End Date
1.0	Intro to Shareholder	[All]	E-mail	10/3/2022	10/7/2022
1.1	Market Research	[All]	Product Proposals	9/30/2022	10/10/2022
1.2	Product Selection	[All]	Product Selection	10/10/2022	10/10/2022
1.3	Project Document Draft	[All]	Section 1 Submission	10/10/2022	10/14/2022
1.4	Module & Block Definitions	[All]	Definition of Individuals' Workload	10/14/2022	10/21/2022
1.5	Component Research	[All]		10/16/2022	10/28/2022
1.5.1	Microcontroller Research	Christian	Microcontroller Proposal	10/16/2022	10/28/2022
1.5.2	Sensor Module Research	Mohammed	Sensor Proposal	10/16/2022	10/28/2022
1.5.3	Wi-Fi Module Research	Carlos	Wi-Fi Module Proposal	10/16/2022	10/28/2022
1.5.4	HMI Module Research	Christian	LCD Display Proposal	10/16/2022	10/28/2022
1.5.5	Power Supply Research	Allison	Power Supply Proposal	10/16/2022	10/28/2022
1.5.6	Computer Software Research	Carlos Christian		10/16/2022	12/1/2022
1.6	Circuit Simulations	Mohammed		10/30/2022	11/11/2022
1.7	Component Selection	[All]	Bill of Materials for Ordering	11/6/2022	11/11/2022
1.8	Engineering Requirement Definitions	[All]	Engineering Requirement Submission	11/13/2022	11/25/2022
1.9	Ordering of Components	Allison	Order Confirmation	11/25/2022	12/1/2022
2.0	PCB Prototype	[All]	Testable Design of Product	1/9/2023	1/27/2023
2.1	Computer Software Development	Carlos Christian	Computer Software [Beta]	1/9/2023	4/21/2023
2.2	Power Supply Prototype	Mohammed	Testable Design of Power Supply	1/22/2023	2/3/2023
2.3	PCB Testing	[All]		1/27/2023	2/3/2023
2.3.1	PCB Design	Allison Mohammed	Design Files for Manufacturing	2/3/2023	2/10/2023
2.4	3D Modeling of Enclosure	Carlos	Design Files for Printing	2/6/2023	2/17/2023
2.5	Microcontroller Programming	Carlos Christian		2/12/2023	3/24/2023
2.6	System Testing	[All]		2/19/2023	3/24/2023
2.7	PCB Ordering	Allison	Order Confirmation	3/19/2023	3/24/2023
2.8	3D Printing of Enclosure	Carlos	3D Printed Enclosure	4/3/2023	4/7/2023
2.9	Mobile Application Testing	[All]		4/3/2023	4/21/2023
3.0	Software Debugging	Carlos Christian		4/3/2023	5/19/2023
3.1	System Assembly	[All]		5/21/2023	5/30/2023
3.2	System Verification	[All]		5/30/2023	6/9/2023
3.3	Product Reflection	[All]	Possible Improvements on Future Revisions of Product	6/11/2023	6/16/2023

Fig. 1. Project Gantt chart [3].

1.5. References and File Links

1.5.1. References

[1] "Statistics and Facts | US EPA," US EPA. (2018). [online] Available at: https://www.epa.gov/watersense/statistics-and-facts [Accessed 12 Oct. 2022].

[2] "Arduino Water Flow Sensor to Measure Flow Rate & Volume," How to Electronics, Jan. 22, 2019. [online] Available at: https://how2electronics.com/arduino-water-flow-sensor-measure-flow-rate-volume/ [Accessed 16 Nov. 2022].

[3] J. S. Tina, *et al. Water Leakage Detection System Using Arduino*. (2022). [online] Available at: https://www.ej-compute.org/index.php/compute/article/download/43/15 [Accessed 16 Nov 2022]

1.5.2. File Links

[4] "Project Timeline," SharePoint Site. (2022). [online] Available at: https://oregonstateuniversity.sharepoint.com/:x:/s/ECE441/EePZYS_cq-5Lk9L46vfaZ1sBaLFnHrGQsCbU7v2aPR0VIA?e=Z80qmy [Accessed 14 Oct. 2022].

1.6. Revision Table

11/17/2022	Mohammed: adding on executive summary
11/16/2022	Mohammed: adding on abstract
11/16/2022	Allison: Revised and added to summary and gap analysis
11/14/2022	Christian: Revised figures, tables and references to be IEEE format.
11/3/2022	Christian: Revised Project Timeline to be on landscape page.
10/14/2022	Christian: Revised Executive Summary and Project Timeline. Added Gap Analysis.
10/14/2022	Mohammed: Added Abstract.
10/13/2022	Carlos: Added Project Timeline Table.
10/13/2022	Allison: Added treasurer role and description.
10/12/2022	Christian: Revised document layout with table of contents and title page.
10/12/2022	Christian: Minimally drafted Executive Summary and added some Team Protocols.
10/12/2022	Carlos & Mohammed: Added Team Contacts and role specifications.
10/10/2022	Christian: Initial document creation.

2. Impacts and Risks

2.1. Design Impact Statement

2.1.1. Introduction

This document presents an assessment of the potential impacts of the Smart Water Flow Sensor project. During the design phase, engineering is done to evaluate these impacts and adjust where they need to be. The engineers are responsible for the design of the project and how it may impact other aspects of life outside of the technical scope. This entails creating a system that not just works efficiently and effectively but is overall ethically correct. This assessment will go into more detail about how the project could cause public health, safety, and welfare impacts, cultural and social impacts, environmental impacts, and economic factors.

The Smart Water Flow Sensor was proposed to help determine water waste in homes which could evidently eliminate some of the national water waste. The system is designed to collect environmental data from water pipes to calculate the water flow through the pipe and relay that information to the user via a computer program or on the product display screen. This would allow the user to see if there is a leak in a pipe that should not have any flow.

2.1.2. Public Health, Safety, and Welfare Impacts

Water damage caused by leaks in pipes can lead to growth and spread of mold [6]. This entails a biohazard if not properly detected and handled. To address this, the use of a water sensor that can accurately identify real-time water flow can communicate to the user if water flow is detected when no water is being used. In addition to the risk of water damage caused by leaks in pipes, there is also a risk of water contamination due to backflow, which occurs when water flows in the reverse direction from the intended flow. Backflow can occur when there is a drop in water pressure, such as during a water main break or a firefighting operation. To prevent backflow, the Smart Water Flow Sensor should be installed with a backflow prevention device [9].

Due to the nature of our product, it is possible that faulty installation or product could lead to leaks and water damage. To avoid this, proper knowledge regarding installation guides will be provided. Making sure there is a tight seal for the sensor will be crucial in the design process. Using items such as O-rings and gaskets can help with this [8]. Since the product utilizes data from an environmental input of water, the electronics will need to be protected and protect the user from electrocution. The system will utilize an integrated fuse in the circuit design to eliminate any shorts that could harm the product. The design of the circuit is important in creating circuit integrity, longevity, and safety for the users. The enclosures for the system will also need to be rated for withstanding external elements such as water. A watertight enclosure will reduce the probability of the user being electrocuted.

2.1.3. Cultural and Social Impacts

The product will output the data it receives to an onboard display and over a network to a computer program. An ideal output to the user would be one that can be universally understood. Since different parts of the world use different measurement systems, the product will have a button that will allow the user to change the units on the LCD Display and the output data to the computer program. This will allow the user to view the data that the product measures in a unit that they can understand. The design of the project considers the concept, “Engineering for Everyone”. This movement is progressive in the sense that it encourages open-sourced engineering and inclusive and diverse engineering ethics when it comes to producing products. [7] In addition to designing the product to be universally understood, it is important to consider the cultural and social context in which the product will be used. For example, in areas where there is a limited water supply, the product could be marketed as a tool to help conserve water and promote sustainability. In areas where water is abundant, the product could be marketed as a tool to save money on water bills.

2.1.4. Environmental Impacts

Water restrictor is one of the most common tools that can be used to maintain the flow of water through the pipe. This can be done by decreasing the area for the flow to move through the pipe or hose. It is important to have water restrictors in order to know how much water it consumes. The benefit of using water restrictors is to save money on your water bill each month. It also can be used to stop leaks and water damage since it can measure the flow and water pressure [2]. This is an important tool that is needed for the water flow sensor to avoid any leaks that might happen. Monitoring water helps researchers to learn about the natural possessions in the environment for the quality of water [3].

2.1.5. Economic Impacts

The primary economic factor for this product is the detection of water waste and possible elimination. The annual water waste due to household leaks is estimated to be about a trillion gallons. With an estimated average cost of $0.003 per gallon, that equates to over $27 million in water waste [1][5]. That number alone could be reduced by finding out if a water pipe has a leak and making potential repairs, because of how long it often takes to detect water waste [4]. The product has the monitoring capability to be a preventive measure for leaks which would help bring that annual water waste down. This impacts individual homeowners and the overall community because it is wasted water that could be utilized elsewhere. Additionally, in addition to the potential cost savings for homeowners, the Smart Water Flow Sensor could also have economic benefits for water utilities and municipalities. By detecting and preventing leaks, the product could help reduce the need for costly repairs and infrastructure upgrades, as well as conserve water resources for future generations.

After the initial proposal, research was done on the project to determine if there was already technology that allowed a user to monitor their water consumption in real-time. There is a new product that does exactly that, however, it is expensive and could be out of the budget of many homeowners. It is important to note that the cost of the product can alter the reach of the consumers. This project was designed not only to be efficient and effective but also affordable. The microeconomics of the product is an important factor in creating an inclusive and obtainable product for many.

2.1.6. Conclusion

In conclusion, the design of the Smart Water Flow Sensor should consider the broader impacts of the product, beyond just its technical functionality. By considering the public health, safety, cultural, environmental, and economic impacts of the product, engineers can create a more ethical and sustainable design that benefits both individuals and society. The Smart Water Flow Sensor is a product that can be built affordably and safely if proper precautions are taken when designing and installing the product. Taking the proper precautions by considering electrical safety standards, safety precautions, in the event of a short circuit, will protect both the user and the product's circuit integrity. Making sure that the product can be universally used by a wide range of ages and in different units.

2.2. Risks

A risk management table was created showing risks that were evaluated during the design assessment. This table includes information detailing the probability, impact, action plan and performance indicator. These sections are utilized for establishing awareness and possible solutions for the potential risks of the project.

TABLE II
RISK MANAGEMENT

Risk ID	Risk Description	Risk Category	Risk Probability	Risk Impact	Performance Indicator	Action Plan
1	Water damage	Safety	Low	High	Mold	Make sure all fittings are watertight to the water sensor to avoid any possible leaks due to the product.
2	Electrocution	Safety	Med	High	Shock	Watertight enclosure for the electronic hardware to avoid any possible damage to the system or electrocution to the user [12].
3	Water flow restriction	Environmental	Med	Med	Low water pressure	Verify the compatibility of the water flow sensor with the average water flow and pressure of households.
4	Low-cost	Economic	Low	Low	Lower cost compared to competition	Designing the product such that it is affordable to majority of the public and outweighs the cost of a water leak [9].
5	Mineral Concentration	Environmental	Med	Med	Faulty or damaged sensor	Verify that the sensor being used is rated for regions with harsher water mineral concentrations [11].
6	Manufacturing Waste	Environmental	Med	Med	Low waste levels from 3D printing	Design an efficient enclosure that will have only necessary supports, and low print time to eliminate excess energy consumption from the 3D printer.

7	Thermal Event	Safety	Low	High	Fire or fuse break	Implement a fuse that can prevent a fire from happening.
8	Electrical inefficiency	Environmental	Med	Low	High power draw	Use low-power draw components and an efficient step-down transformer.

2.3. References and File Links

2.3.1. References

[1] US EPA. (2022). *Understanding Your Water Bill.* [online] Available at: https://www.epa.gov/watersense/understanding-your-water-bill [Accessed Nov. 2, 2022].

[2] Consumer Reports. (2018). *Smart-Home Devices That Stop Leaks and Water Damage*. [online] Available at: https://www.consumerreports.org/home-maintenance-repairs/smart-home-devices-that-stop-leaks-and-water-damage/ [Accessed Nov. 3, 2022].

[3] Fondriest. (2013). *Water Quality*. [online] Available at: https://www.fondriest.com/environmental-measurements/parameters/water-quality/ [Accessed Nov. 3, 2022].

[4] Fondriest. (2013). *Water Quality*. [online] Available at: https://www.fondriest.com/environmental-measurements/parameters/water-quality/ [Accessed Nov. 3, 2022].

[5] "Statistics and Facts | US EPA," US EPA. (2018). [online] Available at: https://www.epa.gov/watersense/statistics-and-facts [Accessed Oct. 12, 2022].

[6] "Mold After a Disaster" CDC (2020). [online] Available at: https://www.cdc.gov/disasters/mold/index.html [Accessed Nov. 30 2022].

[7] "Engineering For Everyone | using your head and your heart to make a difference." (2012). [online] Available at: http://engineeringforeveryone.com/ [Accessed Dec. 03, 2022].

[8] J. Gelston, "Tips for Waterproofing an Electronic Enclosure." (2016). [online] Available at: https://www.aimprocessing.com/blog/tips-for-waterproofing-an-electronic-enclosure [Accessed Dec. 03, 2022].

[9] "Understanding Your Water Bill | US EPA," US EPA. (2017). [online] Available at: https://www.epa.gov/watersense/understanding-your-water-bill [Accessed 31 Oct. 2022]

[10] D. Wroclawski, "Smart-Home Devices That Stop Leaks and Water Damage," Consumer Reports. (2018). [online] Available at: https://www.consumerreports.org/home-maintenance-repairs/smart-home-devices-that-stop-leaks-and-water-damage/ [Accessed 3 Nov. 2022]

[11] "Water Quality - Environmental Measurement Systems," Environmental Measurement Systems. 2013. [online] Available at: https://www.fondriest.com/environmental-measurements/parameters/water-quality/ [Accessed 3 Nov. 2022]

[12] OSHA. *Standard Interpretations, Guarding requirements for 50 volts or more DC*. (2015). [online] Available at: https://www.osha.gov/laws-regs/standardinterpretations/2015-09-04 [Accessed 16 Nov. 2022]

2.4. Revision Table

3/1/2023	Christian: Copied design impact assessment into project document and reformatted it.
12/2/2022	Allison: Added to conclusion and 2.2. Revised for syntax and flow within the document
12/2/2022	Christian: Added Table of Contents and added to 3.1.
12/1/2022	Carlos: Added citations [4] through [6]
12/1/2022	Mohammed: added on conclusion
11/16/2022	Mohammed: modified Risk 3

11/16/2022	Carlos: Added Risk ID 8 and updated gap analysis
11/16/2022	Christian: Added Risk ID 6 to Risk Management Table.
11/14/2022	Christian: Revised table and references to be IEEE format.
11/12/2022	Christian: Removed Design Impact Assessment from Statement section.
11/4/2022	Allison: Added to 2.1.2.2 and 2.2 and risk table
11/3/2022	Carlos: Added Public Health, Safety, and Welfare Impacts
11/3/2022	Mohammed: Finalizing the environmental impact
11/2/2022	Christian: Added Economic Impacts and Cultural and Social Impacts.
10/31/2022	Mohammed: Added Environmental Impacts
10/31/2022	Christian: Added Introduction and Economic Impacts.
10/31/2022	Carlos: Initial document creation and formatting.

3. Top-Level Architecture

3.1. Black Box Diagram

The black box diagram in Fig. 1. represents the system from a simplistic viewpoint. It shows the inputs into the system as a whole and the outputs from the system. This is a good generalization of how the project will interact with the outside world. The inputs include the water flowing through the sensor as an environmental input, the outside power supply from the outlet providing AC power which will be converted and utilized by the system as 5 VDC, and the user inputs on the buttons that will cause some changes to the output display. The only output of the system is to the LCD display which the user will be able to read and manipulate with the input buttons.

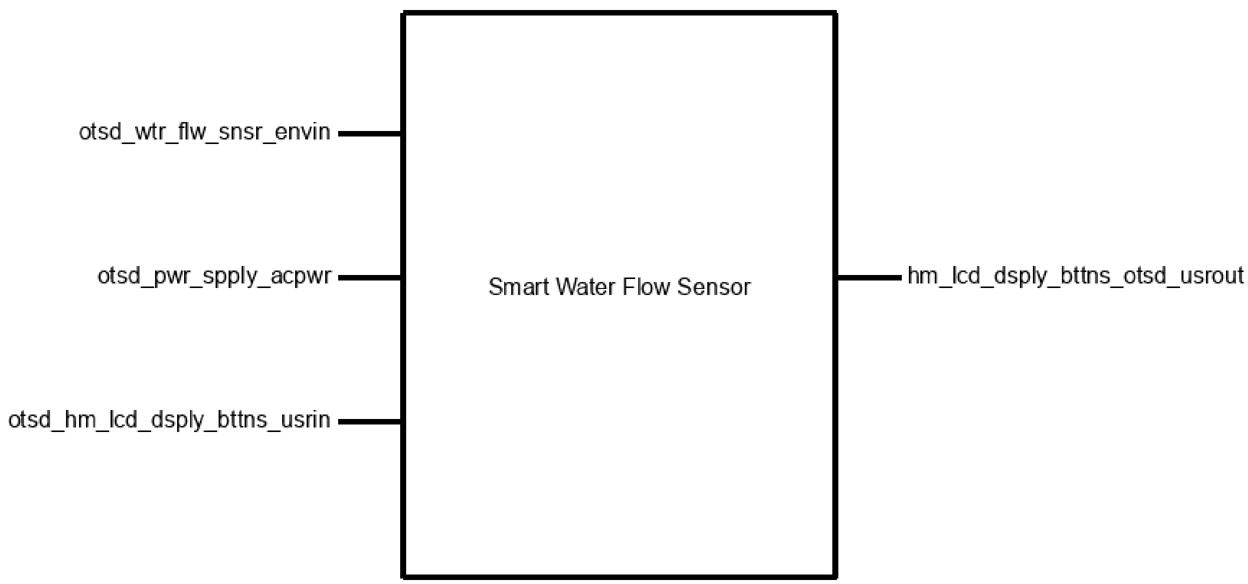

Fig. 1. Black Box Diagram showing the system inputs and outputs [1].

3.2. Top-Level Block Diagram

The top-level block diagram below in Fig. 2. is a thorough representation of the interfaces between each block. This was designed to show how each block will work with each other in the system from a more detailed view. The blocks were color-coded to show who owns each block and is responsible for the block itself working. Once all blocks are individually built, tested, and verified, they will be integrated together into the full system. The inputs and outputs from the black box diagram in Fig. 1. can still be seen as interfaces that do not have a termination on both ends. The interfaces shared between each block require coordination between the two block owners to ensure that each block will work together successfully during system verification. The PCB design and enclosure are two blocks that are interface-less and were placed on the exterior of the diagram.

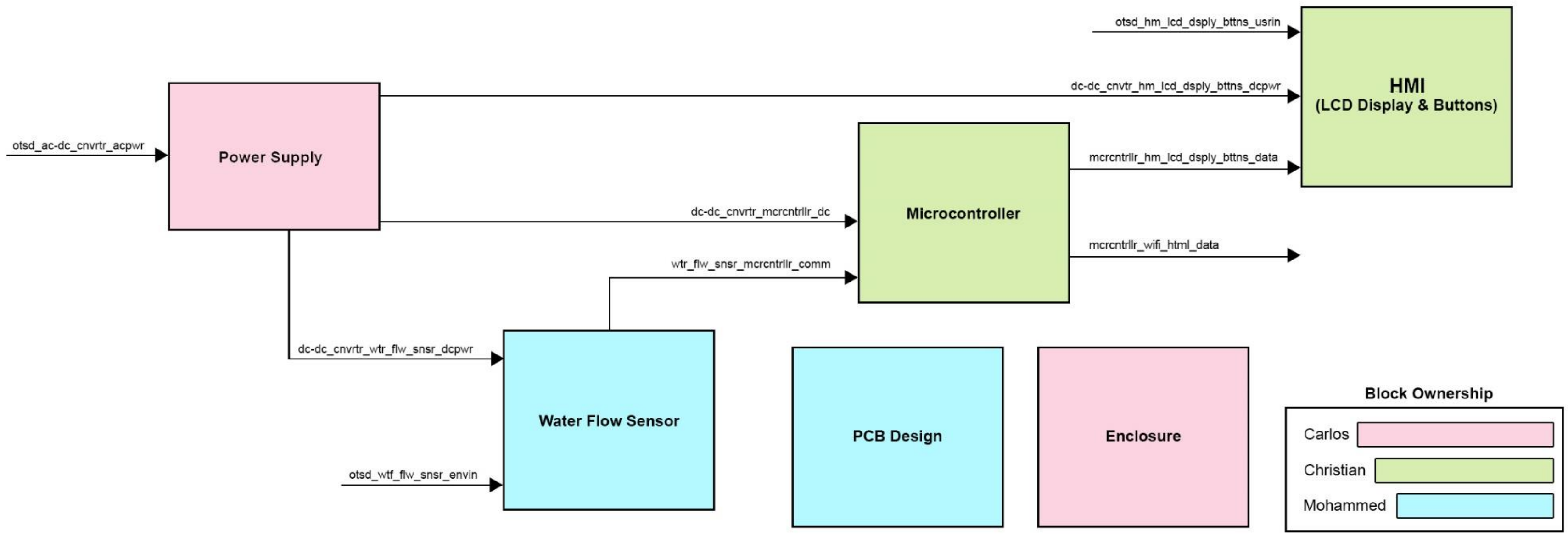

Fig. 2. Top-Level Block Diagram showing each block owner and the interfaces [2].

3.3. Block Descriptions

During the design of the project, each block was brainstormed and described in a way that it would be functional, sustainable and work with the system as a whole. Table I below outlines each block with the block champion, and the description of what the block's purpose in the project will be.

TABLE I
BLOCK DESCRIPTIONS

Name	Description
PCB Design Champion: Mohammed Al-Khalifa	A PCB will be tested, designed, and manufactured to house the power supply, Wi-Fi transmitter, LCD Display, Buttons, Water Flow Sensor, and an Arduino Uno. The design will allow for the major components such as the Arduino Uno, Water Flow Sensor, Wi-Fi transmitter, and LCD Display to be replaced in the product. This will be done using a modular PCB design approach. The PCB will be designed and simulated before creating a prototype system. The prototype system will be probed and tested to ensure that the design is ready for manufacturing.
Water Flow Sensor Champion: Mohammed Al-Khalifa	The purpose of using a water flow sensor block is to calculate the rate of water flow through the pipe, total volume in liters and gallons, and it is reading the temperature of the water sensor internally in Celsius. The water flow sensor will capture the flow of the water using the hall effect. This will send an analog pulse signal to the Arduino Uno which can be captured through serial data communication. The water flow sensor will be connected to the Arduino uno to verify the results by using Arduino software in the serial monitor to receive results of how much water is flowing through the pipe, and it will represent the data by showing the flow rate of water, total volume that the water is poured in the cup to check the accuracy of the water sensor. The internal temperature of the water flow sensor is important to confirm that the sensor is operating while it is displaying the result in the serial monitor of the Arduino uno. Also, we need to use the DC power supply to know the nominal and peak current by using the maximum and minimum voltages. This DC testing can be done by using Triple channel DC power supply which is available in Dearborn building to test.
Power Supply Champion: Carlos Beleno	This block will convert typical wall outlet power from 120VAC to 12-24VDC. The converter will utilize a fuse in the circuit design as a preventative measure. This will ensure that if there is any excess current, short circuit, or fault in the circuit, the rest of the electronics will be safe. This block will then convert the 12-24VDC supply to a 5VDC to supply the water flow sensor, the Arduino Uno, and the LCD display.
Enclosure Champion: Carlos Beleno	The enclosure will be designed in Blender and will coordinate heavily with the PCB design. This includes housing the manufactured PCB board, LCD display, and buttons for the users to interact with. The

	design of the enclosure will have to take the user interaction portion into consideration for an effective product. The enclosure will also incorporate a gasket system to seal the system. This will allow the system to withstand colder temperatures and some minor water exposure.
Microcontroller Champion: Christian Ovchinikov	The Arduino Uno microcontroller will be the backbone of our system, integrating all electronic components and facilitating wireless data transmission. It will receive pulse signals from the water flow sensor, translating them into consumption data which will be recorded on the microcontroller's onboard memory for up to 30 days. The Arduino will be powered by a 5VDC power supply, and the system will be designed to detect high/low digital signals, indicating when one of the four buttons is pressed. This will enable the user to change the output time frame and units of measurement displayed on the LCD screen in real time. The code will utilize I2C serial communication to output the water flow rate data to the LCD display. The two other buttons will be used for turning the display on and off and switching the system into "Wi-Fi Configuration Mode". This will be when the Arduino notifies the ESP32 to turn into a wireless access point, allowing the user to enter their network credentials. The system will wait for a signal from the network, that when triggered, will send data back to the requested computer on the network. This allows the user to access and monitor their water consumption through their home computer. The code for the system will be written in C++ using the Arduino IDE, providing a user-friendly interface for monitoring, and managing water consumption.
HMI (LCD Display & Buttons) Champion: Christian Ovchinikov	LCD Display to output the live readings of the sensor on the enclosure. This will show the user the water flow sensor readings and allow the user to view the overall consumption for up to 30 days. There will be 2 buttons on the enclosure as well that will allow the user to change the display between the current flow rate, and the daily and weekly average. Another button will be used to change the units displayed on the LCD screen. There will also be a power on/off switch that will allow the user to shut down the whole system or turn it on for safety measures.

3.4. Interface Definitions

Each interface of the system was also analyzed during the design phase of the project to ensure compatibility and sustainability. Table II shows each interface listed out with properties that will be important for the compilation of each block. For power supply interfaces, the minimum and maximum voltage ratings are provided as well as the nominal and peak currents. Data communication interface properties show the communication protocols, data rates and other information that might be valuable to the project. All other interface properties that are included in the table below have some sort of importance and can be verified in the specific interface validations from the block sections of this document.

TABLE II
BLOCK INTERFACES

Name	Properties
otsd_wtr_flw_snsr_envin	**Other:** current time: every 3 seconds, it calculates flow rate, total volume, and internal temperature **Other:** working range works between 5-40 L/min **Temperature (Absolute):** Operates at 25 degree C as room temperature
otsd_pwr_spply_acpwr	**Inominal:** 27mA **Ipeak:** 33mA **Vnominal:** 120 VAC
otsd_hm_lcd_dsply_bttns_usrin	**Timing:** 750 millisecond delay between inputs **Type:** 4 User button inputs
wtr_flw_snsr_mcrcntrllr_comm	**Datarate:** 9600 BPS **Other:** Total volume should be in Litter and gallon units on serial monitor **Protocol:** Serial Data Communication
pwr_spply_wtr_flw_snsr_dcpwr	**Inominal:** 5mA **Ipeak:** 15mA **Vmax:** 18V **Vmin:** 5V
pwr_spply_mcrcntrllr_dcpwr	**Inominal:** 29 mA **Ipeak:** 45 mA **Vmax:** 8 V **Vmin:** 5 V
pwr_spply_hm_lcd_dsply_bttns_dcpwr	**Inominal:** 25 mA **Ipeak:** 37 mA **Vmax:** 6 V **Vmin:** 5 V
mcrcntrllr_hm_lcd_dsply_bttns_comm	**Datarate:** 9600 BPS **Other:** Synchronous I2C **Protocol:** I2C Communication
hm_lcd_dsply_bttns_otsd_usrout	**Type:** Numbers (water consumption) and Units **Usability:** Output needs to be understandable for 9 out of 10 people
hm_lcd_dsply_bttns_mcrcntrllr_dsig	**Other:** HIGH/LOW Digital Signal **Vmax:** 5V **Vmin:** 0V

3.5. References and File Links

3.5.1. References

[1] Senior Capstone Website, “Block Diagram Entry,” https://eecs.engineering.oregonstate.edu/capstone/ece/student/, Accessed: 3/6/2023

3.5.2. File Links

[2] Sharepoint, “Block Diagrams & Interfaces.docx,” https://oregonstateuniversity.sharepoint.com/:w:/s/ECE441/EcXKgr6EfkpDjq-7vRlp4Q8BCRVlrVBi0necuXeqLj_E5g?e=iI9Xj1, Accessed: 3/6/2023

3.6. Revision Table

5/3/23	Christian: Updated block diagram and table from student portal, fixed minor formatting.
3/11/23	Christian: Added references and file links.
3/7/23	Christian: Updated figures and tables to incorporate IEEE formatting, added introductions to each figure and table.
3/6/23	Christian: Added Updated Black Box and Top-Level Block Diagrams, Block Descriptions, and Interface Definitions

4. Block Validations

4.1. Power Supply Block

4.1.1. Description

The power supply block provides a fixed 5 V rail from which the entire system draws power. It can be divided into two different parts, an AC/DC voltage rectifier circuit, and a DC/DC voltage regulator circuit. The AC/DC circuit is connected to wall power (120 Vrms @ 60 Hz), which will step down the voltage and reduce the resulting ripple. The DC/DC circuit will receive the rectified voltage and regulate it into a stable 5 V output.

4.1.2. Design

The design of the AC/DC circuit includes the following components: a transformer with an approximate turns ratio of 8:1 (as to step down from 120 Vrms to 15 Vrms), a full wave rectifier, and capacitor. This circuit also includes a male NEMA C14 with a 0.2 Arms fuse for circuit protection. The circuit diagram can be seen in Figure 4.1.

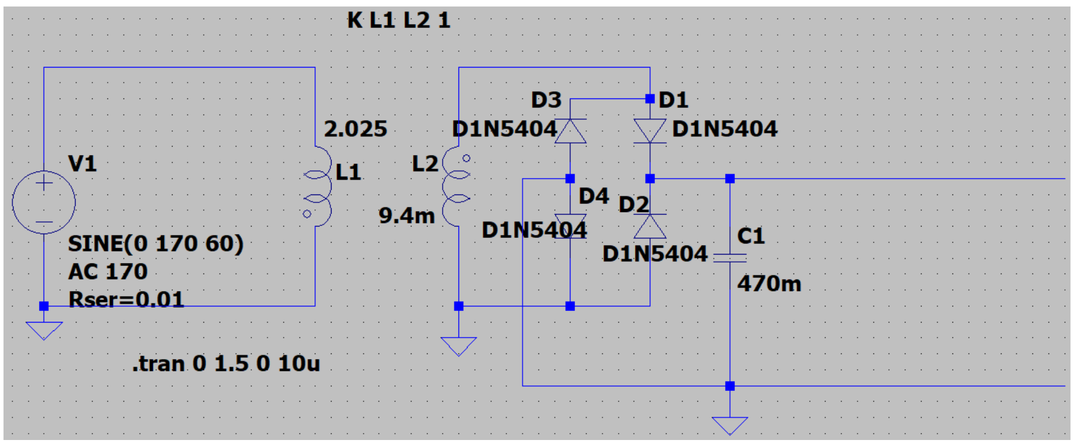

Figure 4.1: AC/DC LTSpice circuit simulation

Once the voltage is stepped down by the transformer, it is rectified by the full bridge rectifier, and lastly, a capacitor s placed in parallel to the bridge's output to filter out some of ripple caused by the rectified voltage. After this process is completed, a DC/DC voltage regulator has its inputs connected in parallel to the rectified and filtered output of the AC/DC converter. The regulator used is based on a LM2596 buck converter circuit. It can take the rectified voltage and step it down to a 5 V output, in addition to holding the output voltage at a less fluctuating rate.

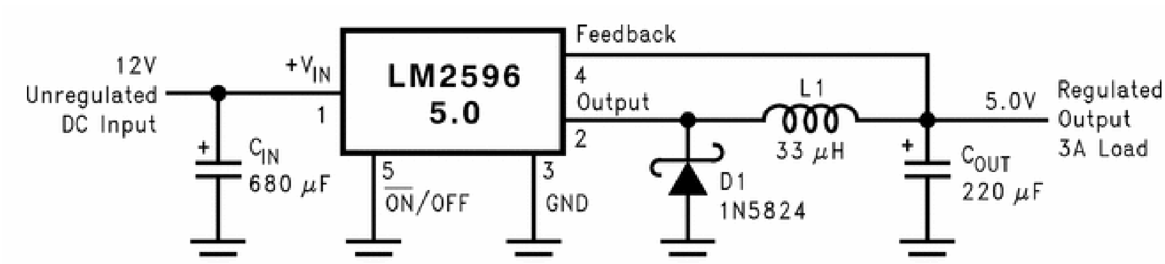

Figure 4.2: TI

4.1.3. General Validation

To validate the power supply block, necessary materials are as follows: a NEMA 5-15 cord, a NEMA 5-15 wall receptacle that is powered (120 Vrms @ 60 Hz), a multimeter, and the power supply unit. First verify that the power supply's rocker switch is turned off, then plug the NEMA 5-15 cord into the power supply, then proceed to connect it to wall power. After the cable is connected to both the power supply and wall power, turn the power supply's rocker switch on. It should be able to power on the rest of the system.

One of the challenges faced during the prototyping of the power supply was the step-down transformer's efficiency. We had to achieve a minimum 65% efficiency on the amps from the primary coil to the secondary coil. Efficiency testing resulted in a lot of small variations to the schematic seen in Figure 4.1., one of the main points of contention being the inclusion of a bleed resistor at the end of the AC/DC converter.

An alternative solution that would have been quicker and probably cheaper to implement would be to buy a 12 V 1 A external wall mount adapter and building the DC/DC regulator circuit to fulfill the made blocks requirement. This would have prevented any safety concerns when it comes to testing and building the high voltage circuit that is the AC/DC.

4.1.4. Interface Validation

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
otsd_pwr_spply_acpwr : Input		
Inominal: 27mA	This current was calculated knowing the load current	Simulation and calculations for the system as a load resistance show that the current through the primary coil should be about 30 mA rms
Ipeak: 33mA	This current was calculated using the lowest possible load resistance for the system	Simulations in LTSpice show an expected of 40 mA of inrush current.
Vnominal: 120 V	Standard rms voltage provided by wall power in the US	The transformer has an 8:1 turns ratio to step down the wall power voltage into a lower and more useful value.
pwr_spply_wtr_flw_snsr_dcpwr : Output		
Inominal: 5mA	This value is calculated knowing the resistance of the water flow sensor and the voltage it is connected to.	The resistance seen by the power supply will require 5 mA to function properly.
Ipeak: 15mA	The lowest resistance the water flow sensor can be seen as will only ever draw 15 mA from a 5 V power supply.	The resistance seen by the power supply will require 5 mA to function properly.

Vmax: 18V	The highest voltage at which the water flow sensor can function properly	Tested the water flow sensor with a power supply providing 18 V
Vmin: 5V	The lowest voltage at which the water flow sensor can function properly	Tested the water flow sensor with a power supply providing 18 V

pwr_spply_mcrcntrllr_dcpwr : Output

Inominal: 29 mA	This value is calculated knowing the resistance of the microcontroller and the voltage it is connected to.	The resistance seen by the power supply will require 30 mA to function properly.
Ipeak: 45 mA	The lowest resistance the water flow sensor can be seen as will only ever draw 45 mA from a 5 V power supply.	The resistance seen by the power supply will require 45 mA to function properly.
Vmax: 8 V	Usual maximum voltage accepted by the Arduino Uno	Max voltage that was tested with the microcontroller.
Vmin: 5 V	Minimum voltage to power the Arduino and it's on-board voltage regulators.	Minimum and nominal voltage for the microcontroller's input voltage.

pwr_spply_hm_lcd_dsply_bttns_dcpwr : Output

Inominal: 25 mA	Expected current when the voltage source is 5 V.	Calculated current for the switch buttons.
Ipeak: 37 mA	Expected max current accounting the minimum resistor value connected to the buttons	This represents the current draw from the buttons using the lowest value resistors.
Vmax: 6 V	User input buttons do not require higher voltages	Tested voltage known to work.
Vmin: 5 V	Nominal voltage at the user input buttons	Voltage that will be nominally provided by the regulator.

4.1.5. Verification Process

To verify the functionality of this block, the following process should be followed:

1. Make sure that the system is turned off.
2. Connect an electronic load or a resistor of 75 Ohms to the power supply's output.
3. Connect an amp-meter in series with the primary coil's hot wire (color coded as black) to measure the AC current.
4. Connect an amp-meter in series with the power supply's positive output (color coded as red) to measure the DC current.
5. Connect the power supply to a 120 Vrms @ 60 Hz wall outlet using a compatible NEMA 5-15 cord.
6. Turn the power supply's rocker switch on.

7. Observe the measure currents from the amp-meters, compare with the table above. DC current should be the sum of all the DC currents in the table above.

4.1.6. References and File Links

4.1.1.1. References

[1] Texas Instruments, “LM2596,” https://www.ti.com/product/LM2596?utm_source=google&utm_medium=cpc&utm_campaign=app-bsr-null-prodfolderdynamic-cpc-pf-google-wwe_int&utm_content=prodfolddynamic&ds_k=DYNAMIC+SEARCH+ADS&DCM=yes&gclid=Cj0KCQiA6rCgBhDVARIsAK1kGPLQPxP18AbUrHuCDBZbS6RtTZPAlbCvJ2XiKRPnh0E4MT53x65jkwgaAiCxEALw_wcB&gclsrc=aw.ds, Accessed: 3/6/2023

4.1.1.2. File Links

[2] AC/DC LTSpice simulations, https://drive.google.com/file/d/1DHlhsdydCfY-pxnkEHthk9QHecr3-g6O/view?usp=sharing, Accessed: 3/6/2023

4.1.7. Revision Table

3/11/2023	Carlos: Merged with project document

4.1. Water Flow Sensor Block

4.1.1. Description

The purpose of using a water flow sensor block is to calculate the rate of water flow through the pipe, total volume in liters and gallons, and it is reading the temperature of the water sensor internally in Celsius. The water flow sensor will capture the flow of the water using the hall effect. This will send an analog pulse signal to the Arduino Uno which can be captured through serial data communication. The water flow sensor will be connected to the Arduino uno in order to verify the results by using Arduino software in the serial monitor to receive results of how much water is flowing through the pipe, and it will represent the data by showing the flow rate of water, total volume that the water is poured in the cup to check the accuracy of the water sensor. The internal temperature of the water flow sensor is important to confirm that the sensor is operating while it is displaying the result in the serial monitor of the Arduino uno. Also, we need to use the DC power supply to know the nominal and peak current by using the maximum and minimum voltages. This DC testing can be done by using Triple channel DC power supply which is available in Dearborn building to test.

4.1.2. Design Schematics:

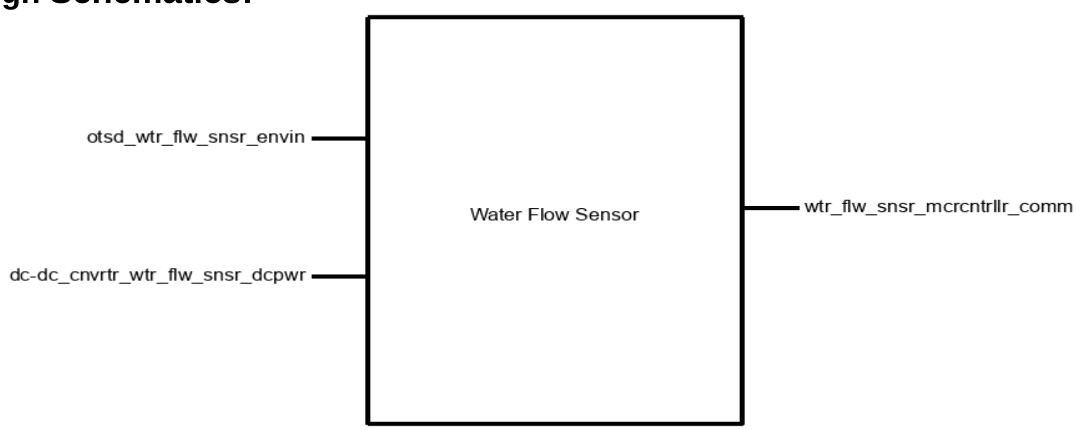

Figure 1: Block diagram

In the block diagram, it is important to have the inputs and output for the block diagram in order to know what we need the water flow sensor to do. Therefore, the two inputs are environment and DC power supply, and the output will be the microcontroller communication for the water flow sensor. For the environmental inputs, it is important to know that we are going to display the rate of water flow to confirm that the working range is between 2-100 L/min. The internal temperature of the water flow sensor will operate between -20 to 80 in Celsius because it might cause error on the measurement while testing it through the serial monitor in Arduino uno software. To ensure that we get accurate readings, we need to confirm that the internal temperature of the water flow sensor is working between -20C and 80C. The total volume of the water flow sensor is important to help us to check the accuracy of the water flow sensor that it is pouring through the pipe. Volume should be in liter and gallon units while it is displayed in serial monitor in arduino uno. The result in the serial monitor calculates Total volume, flow rate, operating temperature every 3 seconds as I include it in the instructions. For the DC power supply, we need to use minimum and maximum of voltage to know the nominal current, so we used 5 volts as the minimum voltage based on operating range is between 5-18 volts for the water flow sensor and the maximum volt is 18 volts because this might cause issue if we use higher volt in DC supply. While I am testing the maximum and minimum voltage, the nominal current gets around 5mA which is good enough because in our project, we are going to use less power. For the peak current, I used 15mA in order to confirm that the current will not go over than this value since I have tested the Vmax and Vmin and it was lower than 15mA, it should be good for the system.

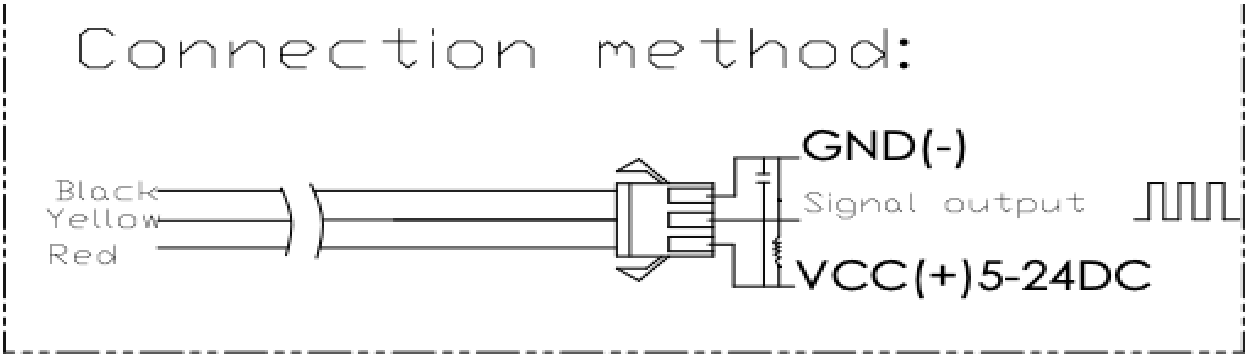

Figure 2: The Wiring Diagram

The purpose of showing the water flow sensor wiring diagram is to know which wire needs to be connected, so there should be three wires for the water flow sensor needed to be connected in the arduino uno and DC power supply. As we know that the block wire is going to be ground if we connect it in arduino uno because it operates as zero voltage and it will be connected to a GND pin. Red wire will be the voltage of the water flow sensor since it will be connected to a 5v pin in arduino uno. The yellow wire will operate to help the sensor to send an analog pulse signal to the Arduino Uno which can be captured through serial data communication.

4.1.3. General Validation

Water flow sensor is the sensor that needs to be used in order to achieve the project of water flow sensor. It is supposed to calculate the water that flows through the pipe every minute in order to help us to control and monitor the rate of water within applications and ensure control processes are running safely. Water sensors are already available anywhere such as Amazon, which sells any size and range of water works you want to complete the project successfully. Also, the cost of a water flow sensor only costs about 18$ in amazon, and this product is light and flexible, small in size and easy to use to install. The size of the water flow sensor is 1 gallon and it needs to be fit with the enclosure because the water sensor will be outside of the enclosure. Based on product information is that the item weight is 5.6 ounces.

Water flow sensor has straightforward instructions because code is easy to find anywhere in google, especially in arduino uno. For the engineering time, this block does not take a long time because it is easy to deal with the water sensor. The only time you are going to spend more time is writing code for the water sensor because you need to know the specification of the water sensor such as operating voltage, operating temperature, flow rate, water pressure, and other specifications. Therefore, each water sensor will have different values from others, so you need to be sure about the values. Water sensor will meet all the requirements that are needed for the partner project and project, and our group agreed with me to use a water flow sensor in the project because it is one of the main components that we need to accomplish our project successfully.

4.1.4. Interface Validation

There are three interfaces for the water flow sensor block. The first interface would be otsd_wtr_flw_snsr_envin : Input interface, the purpose of this interface is to test the code in arduino uno software to get the result in serial monitor. There are three properties to test the water flow sensor such as working range of sensor, Total volume of water, and Temperature that is read internally in the sensor.

The second interface would be wtr_flw_snsr_mcrcntrllr_comm : Output, which would be for the serial communication to display the result for rate baud, serial data, and result time. The baud rate is 9600 to send analog signals.

The final interface would be dc-dc_cnvrtr_wtr_flw_snsr_dcpwr : Input, which will be tested Vmax and Vmin to get the nominal current to make sure that the nominal current is in the range of specified value. For the peak current, as long as the nominal current is not higher than peak current, the value is appropriate. But I used 15mA for the peak current for the maximum current that can operate in a water sensor.

TABLE I
INTERFACE VALIDATIONS

Interface Property	Why is this interface of this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
otsd_wtr_flw_snsr_envin : Input		
Other: working range is between 2-100 L/min	The water flow sensor is working between these two values because of the capacity of the sensor. Based on the video, I followed how to calculate the rate of water but in a different way from units, range value.	For the YF-G1: based on the datasheet of the water flow sensor is that the value range is working between 2-100 L/min because of the capacity of the pipe of water flow.
Other: Total volume should be in Litter and gallon units on serial monitor	The volume should be in the unit of liter or gallon based on the interface. One of the units is good but I included these two to show how they convert in code while printing the result.	For the YF-G1: Based on volume, the units are litter or gallon based on the requirement of the project, and this is what I decided to use these two to display in the output of the serial monitor.

Temperature (Absolute): Operates between -20C to 80C	The value is chosen to show the internal temperature of the water flow sensor in serial data. Therefore, it might give error measurement if internal temperature is higher or lower than expected value	For the YF-G1: The range of temperature of the water flow sensor should be between -20 to 80 in celsius based on the datasheet. Therefore, it might give error measurement if internal temperature is higher or lower than expected value

[2] "Datasheet water flow sensor."

[4] How to connect water flow sensor with Arduino, Accessed 1/12/2023

wtr_flw_snsr_mcrcntrllr_comm : Output

Data Rate: 9600 BPS	This value of baud rate can help arduino to send out commands through the USB connection. This value is appropriate because of water size to analog signal	It will meet the expectation because this is good enough bits per second for the water flow sensor, and it is capable of transferring data to the arduino serial monitor.
Protocol: Serial Data Communication	It is important to use a serial monitor in order to display data from testing the water in the sensor. The purpose of this block is to pour the water through the pipe of the sensor to ensure that the system works between these values.	It will meet the expectation because it will work successfully by using arduino software with appropriate code for the flow rate. Also, it is going to display the total volume in liters and gallons and internal temperature that is operating in water flow sensor
Messages: Results prints every 3 seconds	This value was chosen in order to make the sensing period 3 seconds for every reading.	Based on the requirement, it is important that to set up enough time for the water flow sensor to give a chance to measure how much water flows every 3 seconds

dc-dc_cnvrtr_wtr_flw_snsr_dcpwr : Input

Inominal: 5mA	The nominal current is based on the DC power supply device when we set up Vmax and Vmin as it is shown in the property.	For the YF-G1: The DC power of water flow sensor gives the nominal somewhere around 5mA
Ipeak: 15mA	This value was chosen because it is a maximum current of the water flow sensor based on the specification from amazon in operating current.	For the YF-G1: This value is good since the DC power supply does not exceed 15 mA, it supposed to be good because the component might get burn if we use high value of voltages but if we used reasonable values, it should be fine
Vmax: 18V	This value was chosen because this is the maximum voltage that the DC power supply run based on the specification from 5-18 v that the water flow sensor operates	For the YF-G1: The voltage operates on as follow: Minimum voltage: 5v Maximum voltage: 18v based on specification
Vmin: 5V	This value was chosen because the arduino uno has a 5 volt pin which the sensor would run.	For the YF-G1: The voltage operates on as follow: Minimum voltage: 5v Maximum voltage: 18v based on specification

[1] ALLPARTZ-2-100L. [Accessed: 20-January-2023].

4.1.5. Verification Process

In this section, we will need to test the input and output of the water flow sensor based on the interfaces with their properties. We are going to use an arduino uno, water flow sensor, and DC power supply that is available in dearborn 211. In the serial monitor, it is supposed to display the total volume, flow rate, and internal temperature in arduino uno with the output of baud rate and serial data. For the DC power testing, you need to calculate the Vmax and Vmin in order to get the nominal current. For the nominal current it is supposed to be a value in the range because sometimes the dc power device does not give an accurate reading. For the Ipeak, since the nominal did not go over the peak current, it will be fine to pass the verification. The steps of doing the environment testing and DC power testing:

Environment Testing steps:

1. Obtain sensor and arduino
- Buying water flow sensor and arduino uno from Amazon

2. Connect the sensor to Arduino uno
- The right leg of the water flow sensor will be connected to the 5v pin in the arduino uno which is the red wire.
- The middle leg of the sensor will be connected to the D2 pin in arduino uno which represents yellow wire.
- The left leg of the sensor or black wire will be connected to ground (GND) pin which is zero volt.

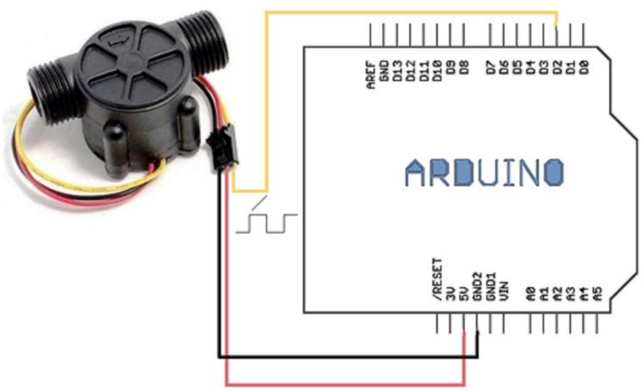

Figure 3: connected to Arduino Uno

3. Connecting the arduino uno to laptop
- Connect the arduino uno to the laptop by using USB
- Upload the code to operate the water flow sensor and start testing water in sensor
- Open serial monitor to display the flow rate of water, total volume in gallons, internal temperature of the sensor

4. Verify the data
- Pour the water through the pipe of the water sensor to check to display how much water flows through the pipe to confirm it is working on flow range.
- To display the internal temperature while I pour the water through the pipe.
- Checking the total volume conversion units

Test code:

[3] OpenAI test code, Accessed 1/19/2023

DC power supply testing:

Step 1: connect red wire to the red channel of power supply DC, which is the Vcc for the arduino, you will also need to connect the ground or black wire into black channel of DC power. For the middle wire, you will connect it into the arduino pin 2 to send the signal from the arduino in order to be connected through the computer USB for arduino software.

Step 2: Turn on the DC power device, and set up the value of Vmax and Vmin in order to get the nominal current for both

Step 3: verify that the nominal current does not exceed higher than the peak current that you have included in the properties.

These two testing can be done once but the difference is you will need to connect the wires based on the DC power steps.

4.1.6. References and File Links

4.1.6.1. References

[1] "Allpartz 2-100L/min 1 inch water flow sensor flowmeter hall flow sensor ..." [Online]. Available: https://www.amazon.com/ALLPARTZ-2-100L-Sensor-Flowmeter-Counter/dp/B07L97DLGF. [Accessed: 20-January-2023].

[2] "Datasheet water flow sensor." [Online]. Available: https://rajguruelectronics.com/Product/1665/YF-G1%20DN25%201inch%20Water%20Flow%20Sensor.pdf. [Accessed: 08-Feb-2023].

[3] "Chat.openai.com." [online]. Available: https://chat.openai.com/. [Accessed: 19-Jan-2023].

[4] "How to connect water flow sensor with Arduino," *YouTube*, 13-May-2019. [Online]. Available: https://www.youtube.com/watch?v=-uoTlJc5hGI&t=5s. [Accessed: 12-Jan-2023].

4.1.1.1. File Links

https://www.electroschematics.com/working-with-water-flow-sensors-arduino/

4.1.2. Revision Table

Date	Revision Description	Done By
1/18/2023	Did Block 1 validation draft	Mohammed Al-khalifa
1/19/2023	Did interfaces validation	Mohammed Al-khalifa
2/8/2023	Modified in design and validation	Mohammed Al-khalifa
2/8/2023	Adding titles and paragraphs in same font	Mohammed Al-khalifa
2/9/2023	Modified in verification and interfaces	Mohammed Al-khalifa

2/9/2023	Write revision statement	Mohammed Al-khalifa
2/10/2023	Link the water flow sensor code	Mohammed Al-khalifa
2/10/2023	Finalized Block 1 validation draft	Mohammed Al-khalifa

4.2. Microcontroller Block

4.2.1. Description

The Arduino Uno microcontroller will be the backbone of our system, integrating all electronic components and facilitating wireless data transmission. It will receive pulse signals from the water flow sensor, translating them into consumption data which will be recorded on the microcontroller's onboard memory for up to 30 days. The Arduino will be powered by a 5 VDC power supply and the system will be designed to detect high/low digital signals, indicating when one of the four buttons is pressed. This will enable the user to change the output time frame and units of measurement displayed on the LCD screen in real time. The code will utilize I2C serial communication to output the water flow rate data to the LCD display.

The two other buttons will be used for turning the display on and off and switching the system into "Network Configuration Mode". This will be when the Arduino notifies the ESP8266 to turn into a wireless access point, allowing the user to enter their network credentials. The system will wait for a signal from the network, that when triggered, will send data back to the requested computer on the network. This allows the user to access and monitor their water consumption through their home computer. The code for the system will be written in C++ using the Arduino IDE, providing a user-friendly interface for monitoring, and managing water consumption.

4.2.2. Design

The Smart Water Flow Sensor will provide the user with the water flow rate and consumption visually on the system and through a computer program on their home network. This will be done using system code to measure the flow rate from the sensor, store that data, and output that data to an LCD display and over Wi-Fi. The code will be flashed to the ATmega328p once before the production of the system. This will be done through the interfaces visible in Table 1 that show the different inputs and outputs of the microcontroller block. The water flow sensor, LCD display, and Wi-Fi module communicate over serial data communication. The buttons will be circuited to where LOW will be default and when pressed, the Arduino will receive a HIGH input back as a digital signal.

The different inputs and outputs can be visualized using the black box diagram in Fig. 1. The inputs are on the left side of the include the water flow sensor communication, the 5 VDC power supply, and the digital signal from the user buttons. The right side of the black box diagram shows the outputs being the I2C communication for the LCD display and the ESP32 communication. Below Fig. 1 is Table I with the interface names and properties.

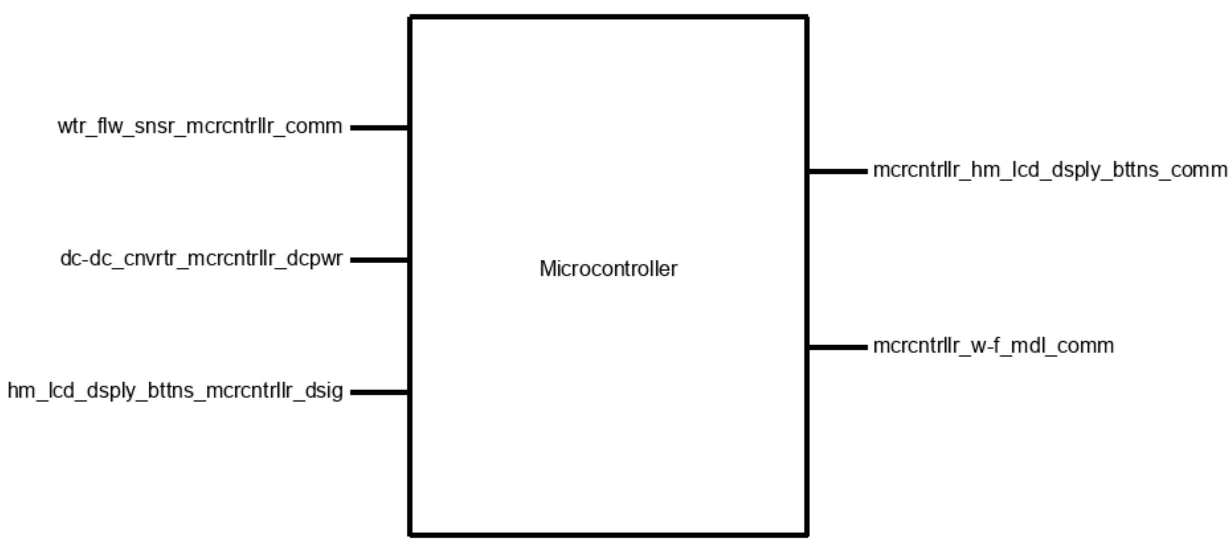

Fig. 1. Black Box Diagram of Microcontroller Code Block

TABLE I
MICROCONTROLLER BLOCK INTERFACES AND PROPERTIES

Interface	Properties
wtr_flw_snsr_mcrcntrllr_comm	Baud Rate: 9600 BPS Other: Pulse Signal
dc-dc_cnvrtr_mcrcntrllr_dcpwr	V_{MAX}: 20 V V_{MIN}: 6 V $I_{NOMINAL}$: 50 mA I_{PEAK}: 200 mA
mcrcntrllr_hm_lcd_dsply_bttns_comm	Baud Rate: 9600 BPS Protocol: I2C Communication
mcrcntrllr_w-f_mdl_comm	Baud Rate: 9600 BPS Messages: Sending Water Flow Data Protocol: Serial Data Communication
hm_lcd_dsply_bttns_mcrcntrllr_dsig	V_{MAX}: 5 V V_{MIN}: 0 V Other: HIGH/LOW Digital Signal

Since the project will be using a 5V DC power supply, the microcontroller will be primarily wired for data communication. Fig. 2 shows the wiring diagram used for testing and building the code for the Arduino Uno. The water flow sensor, buttons and the ESP8266 will be wired through digital inputs with the ESP32 being connected to TX/RX respectively. The LCD display with the integrated I2C communication interface will be wired to the analog pins with serial clock and data capabilities. Table II below Fig. 2 has each of the inputs and outputs coordinated with the pin of the Arduino Uno and the functionality.

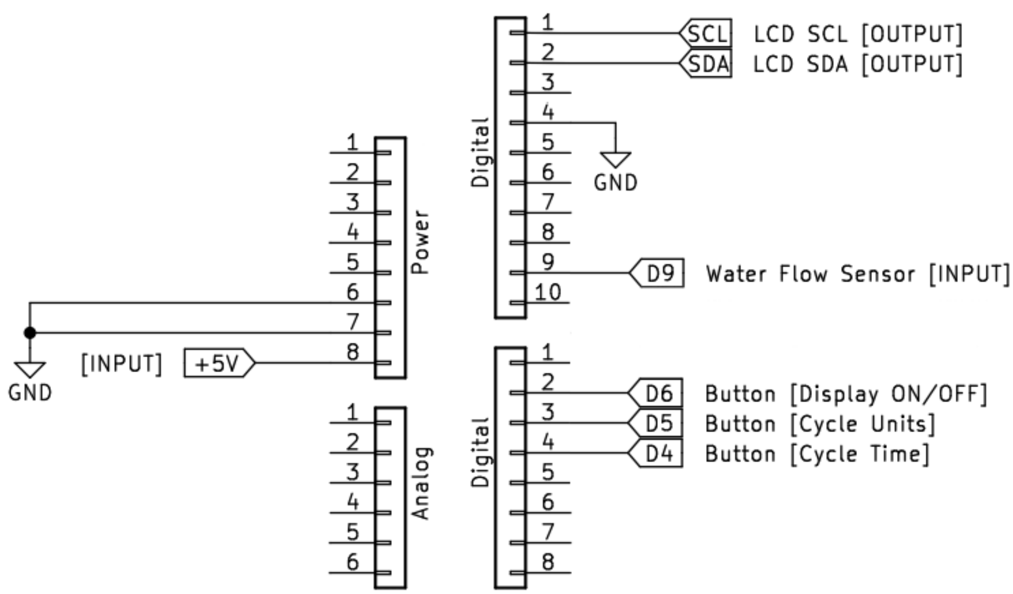

Fig. 2. Example schematic of connections for the microcontroller block.

TABLE II
PIN DESCRIPTIONS

Pin Name	Direction	Usage
VIN	Input	+5V DC power supply for the Arduino.
SCL	Output	Serial Data Pin
SDA	Output	Serial Clock Pin
D2	Input	Button – Cycles through output times on LCD Display (second, minute, hour, day, week, month)
D3	Input	Button – Changes output units on LCD Display (gallons, liters)
D5	Input	Button – LCD Display ON/OFF
D6	Input	Pulse signal from Water Flow Sensor which is used to calculate water flow rate.

Code Description

The design of the code was created based around the flow diagram in Fig. 3 which presents the different stages of execution. The code was designed knowing that there would be user and environmental inputs, network configuration, data storage and output. This output would need to be able to be transmitted over a wireless network which is where the network configuration comes into effect for new users.

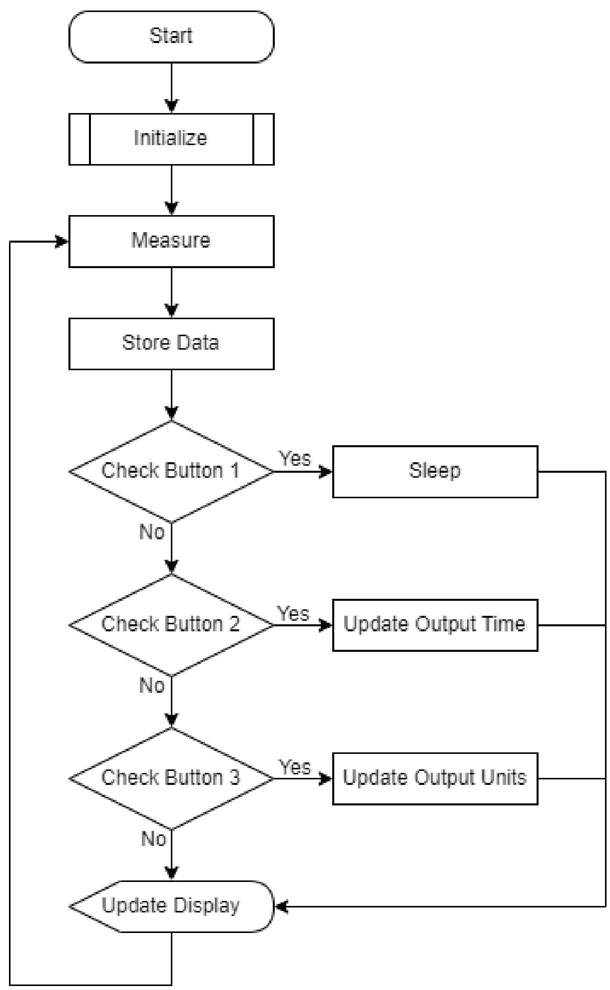

Fig. 3. Code Flow Diagram

As outlined in the code flow chart in Fig. 3, the first step after thrcode starts is the initialization phase. During this phase, the code will first configure the system's input and output pins accordingly. Serial communication will also be initialized in this phase to 115200 BPS to ensure swift communication between all the onboard components. Table II lists the pins used for this design using the Arduino Uno pin names, direction of data flow and the usage.

After the system is initialized, the code enters an infinite loop where it first performs a reading of the water flow sensor and stores that data in memory. The system then checks to see if any buttons were pressed by the user that would indicate either a change in the output display or sending the system into "Network Configuration Mode". The first button check will cycle the LCD display from being on and off. This will act as a sleep mode for the system to minimize any wasted current being drawn by the device. This will update the display, then continue the loop checking for data transmission requests and then back to the beginning of the loop. If the second or third button is pressed, the output display time or units will be updated respectively.

If the fourth button is pressed, the system will send a message over serial to tell the ESP8266 to produce a wireless access point. The ESP8266 will then prompt the user to provide their credentials to their Wi-Fi network which will allow the system to then be connected to the network. If the connection is established, the ESP8266 will prompt the Arduino with a reply over serial stating either a success or failure. The system then updates the display and continues forward through the loop. The last step of the loop is checking whether a message over serial requests the water flow data that is stored on the Arduino. If the Arduino gets that message, the data will be transmitted over serial data.

4.2.3. General Validation

The selection of the Arduino Uno microcontroller with the ATmega328 was carefully analyzed due to the vast number of variable inputs and outputs of the system. Firstly, the ATmega328 boasts exceptional efficiency, providing ample computational power for the system's requirements with a processing speeds capable of running several different I/O peripherals.

Additionally, the Arduino Uno is a highly accessible and versatile microcontroller platform, benefiting from a vast community of users and developers who provide support and share resources. This enables users to quickly and easily find relevant code examples, libraries, and tutorials, which greatly aids in the development process. The ATmega328 can be picked up for around $8, and by using a modular design, we have the capability to swap out the microcontroller if anything were to fail.

The ease of use of the Arduino Uno is also a key factor, as it provides a streamlined interface for programming and development through the Arduino Integrated Development Environment (IDE). The system code will be written in C++, a widely used and well-documented programming language, which allows for easy modification and extension by users.

Finally, the integration of the Arduino Uno with other components, such as the water flow sensor, LCD display, and Wi-Fi module, is straightforward and thoroughly documented. The utilization of I2C communication and serial data transmission between components ensures fast and reliable communication, which is essential for the system's functionality.

The microcontroller code block will ensure that the system can accurately provide the water flow data and stores that data accordingly. The block will also ensure that the system is easy to use with inclusive usability by the users. This will include integrating a wireless access point for the user to connect the system to their network and being able to change the output display from imperial to metric units. The microcontroller block is not defined by a specific system requirement but plays a key role in completing many of them.

4.2.4. Interface Validation

The interface properties for each input and output of the microcontroller block are validated in Table III below. This table describes what and why of the interface properties along with how the design of the block ensures that each interface meets or exceeds the design property.

TABLE III
INTERFACE VALIDATIONS

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
wtr_flw_snsr_mcrcntrllr_comm : Input		
Baud Rate: 9600	The default baud rate for the ESP8266 is 9600 BPS, thus that rate is used for consistency [4].	In the initialization step of the code, the baud rate of the serial communication is set to 9600 BPS.
Other: Pulse Signal	The pulse signal provided by the water flow sensor is used to measure the water flow rate by the microcontroller.	The water flow sensor is connected to a digital input pin on the Arduino. The pulse produced by the hall effect sensor will be read as a HIGH/LOW for detecting water flow.
dc-dc_cnvrtr_mcrcntrllr_dcpwr : Input		
Inominal: 50 mA	The ATmega328 typically draws about 50 mA of current with a 5 V DC power supply [2].	The microcontroller will not be powering anything from itself; thus, the only current being drawn will be by the ATmega328.
Ipeak: 200 mA	The absolute maximum current rating of the ATmega328 is provided in the datasheet as 200 mA [2].	The Arduino provides onboard power regulation which will limit current provided to the ATmega328 and has an onboard 500mA fuse to protect against current peaks.
Vmax: 20 V	The absolute maximum voltage rating of the Arduino Uno is 20 V [1].	The DC power supply will only supply 5 V, thus will not exceed the absolute maximum allowed by the Arduino Uno's voltage regulator.
Vmin: 6 V	The minimum voltage rating of the Arduino Uno is 6 V [1].	The ATmega328 has a typical voltage rating of 6 V, which will be achieved by stepping up the DC power supply form 5 V to 6 V [2].
mcrcntrllr_hm_lcd_dsply_bttns_comm : Output		
Datarate: 9600 BPS	The default baud rate for the ESP8266 is 9600 BPS, thus that rate is used for consistency [4].	In the initialization step of the code, the baud rate of the

		serial communication is set to 9600 BPS.
Protocol: I2C Communication	The LCD display utilizes an interface that communicates using I2C which simplifies the connections to just 2 pins, Serial Data (SDA) and Serial Clock (SCL) [3].	I2C simplifies data transfers by transferring messages instead of specific bits. The messages have an address frame and one or more data frames that contains the information.
mcrcntrllr_w-f_mdl_comm : Output		
Datarate: 115200 BPS	The default baud rate for the ESP8266 is 9600 BPS [4].	In the initialization step of the code, the baud rate of the serial communication is set to 9600 BPS.
Messages: Sending Water Flow Data	A serial message containing “SEND” from the ESP8266, will indicate to the Arduino that data needs to be transferred to the ESP8266 for data transfer to the network.	The Serial Monitor can verify that these messages are executing functions that will initiate the network connection with the ESP8266, and to send data through serial communication.
Protocol: Serial	The communication between the microcontrollers is done through serial communication using the TX/RX pins respectively on the Arduino Uno and the ESP8266.	The flow of messages are over serial communication and can be verified using the Serial Monitor.
hm_lcd_dsply_bttns_mcrcntrllr_dsig : Input		
Other: HIGH/LOW Digital Signal	The buttons are configured to transmit a low signal to the digital pins and a high signal when pressed.	The digital inputs from the buttons are checked when a HIGH is seen in the loop, the corresponding function related to that button is executed.
Vmax: 5 V	When the button is pressed, the circuit is completed sending 5 V to the digital pin which indicates to the microcontroller that the button was pressed.	The buttons are powered with a 5 V circuit, once pressed, the 5 V will reach the digital pin indicating a HIGH signal.
Vmin: 0 V	When the button is not pressed, the circuit is not completed, and the digital pin will not receive any voltage.	When the button is not pressed, the output is grounded, thus having 0 V input indicating a LOW signal.

4.2.5. Verification Process

For this code block, the minimum verification of the interface is required. The interface properties however include specific properties about communication protocols and digital signal inputs. The verification below will include some tests that will verify the interface communications and digital signals.

1. Connect the microcontroller to a PC using a USB cord with data transmission capabilities. Flash the Arduino Uno with the compiled code [6].

2. The system will power on and start looping through collecting, storing, and outputting data to the display using serial data communication.
3. Press the first button, putting the device into sleep mode. The output display will shut off. Pushing it again will turn the output display back on.
4. Press the second button, cycling the water consumption time frame on the output display.
5. Press the third button, changing the units of the output display.
6. Press the fourth button, putting the device into network configuration mode.

a. Enter “CONNECTED” into the serial monitor to imitate the ESP32 telling the Arduino that it has established a connection.

b. When “Connecting…” appears on the output display, enter “SSID” into the serial monitor and press enter to imitate the ESP32 sending a serial message with the network SSID.

PASS: To pass this test:

- The system must change the output display according to the button pressed by the user. This will prove that the block has satisfied portions of the engineering requirements.
- The system must enter “Network Configuration Mode” by sending an indicator message to the ESP32 and receiving serial data back over the serial monitor. This can be verified in the serial monitor.
- The system must still loop through the collection and storage of data while in sleep mode. This can be verified in the serial monitor while the output display is in the OFF mode.

4.2.6. References and File Links

4.2.6.1. References

[1] Arduino, “Arduino Uno R3 Datasheet,” https://docs.arduino.cc/static/c5af4b1dbd0260fb58643409d2ada778/A000066-datasheet.pdf, Accessed: 1/10/2023

[2] Atmel, “ATmega328 Datasheet,” https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf, Accessed: 1/18/2023

[3] Handson Technology, “I2C Serial Interface 1602 LCD Module,” https://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf, Accessed: 12/5/2022

[4] Microchip. “ESP-01 WiFi Module Datasheet,” https://www.microchip.ua/wireless/esp01.pdf, Accessed: 2/16/2023

4.2.6.2. File Links

[5] SharePoint. “water_flow_arduino.ino,” https://oregonstateuniversity.sharepoint.com/:u:/s/ECE441/ESAR0L0x3DtPuHLXHF5DnioBvyn9XTFQbnVXfaHilMR8Fw?e=HZj8ki, Accessed: 2/8/2023

4.2.7. Revision Table

2/17/2023	Christian Ovchinikov: Added to Project Document and made some minor changes.
2/11/2023	Christian Ovchinikov: Revised general validation contents and verification plans.

2/10/2023	Christian Ovchinikov: Finished adding to interface validations, updated general validation contents and verification plans.
2/9/2023	Christian Ovchinikov: Added interface validation table and populated majority of it.
2/8/2023	Christian Ovchinikov: Added references and file links, added wiring diagram figure and description, updated design and overall organization of document for better flow.
2/7/2023	Christian Ovchinikov: Updated formatting of fonts, figures, etc. to IEEE, updated flow diagram per peer review.
2/5/2023	Christian Ovchinikov: Updated formatting of document headers.
2/3/2023	Christian Ovchinikov: Updated interface properties.
1/17/2023	Christian Ovchinikov: Populated every section.
1/14/2023	Christian Ovchinikov: Initial document creation.

4.3. HMI (LCD Display & Buttons)

4.3.1. Description

The system will use an LCD Display to output the live readings of the sensor on the enclosure. This will show the user the water flow sensor readings and allow the user to view the overall consumption for up to 30 days. There will be 2 buttons on the enclosure as well that will allow the user to change the display between the current flow rate, and the daily and weekly average. Another button will be used to change the units displayed on the LCD screen. There will also be a power on/off switch that will allow the user to shut down the whole system or turn it on for safety measures.

4.3.2. Design

Our Water Flow Sensor will enable users to monitor water flow rate and consumption visually on the system and through a computer program on their home network. To achieve this, the system code will measure the flow rate from the sensor, store the data, and display it on an LCD screen. The buttons will be circuited so that when pressed, the Arduino will receive a HIGH input back as a digital signal.

The different inputs and outputs of the microcontroller block can be visualized using the black box diagram in Fig. 1. The inputs on the left side include the water flow sensor communication and the 5 VDC power supply, along with the digital signal from the user buttons. The right side of the diagram shows the outputs being the I2C communication for the LCD display. The code will be flashed to the ATmega328p once before the production of the system, and the water flow sensor, LCD display, and buttons with the help of the microcontroller. For reference, Table 1 lists the interface names and properties to assist with flashing the code to the ATmega328p.

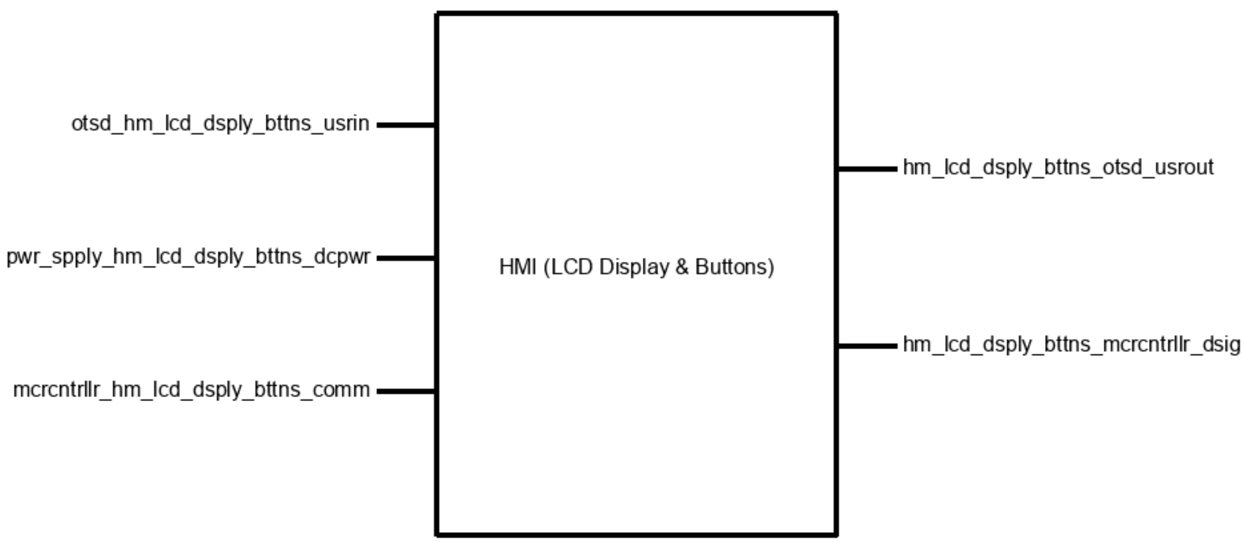

Fig. 1. Black Box Diagram of Display and User Input Block.

TABLE I
MICROCONTROLLER BLOCK INTERFACES AND PROPERTIES

Interface	Properties
otsd_hm_lcd_dsply_bttns_usrin	Timing: 750 millisecond delay between inputs Type: 4 User button inputs
pwr_spply_hm_lcd_dsply_bttns_dcpwr	V_{MAX}: 6V V_{MIN}: 5 V $I_{NOMINAL}$: 25 mA I_{PEAK}: 37 mA
mcrcntrllr_hm_lcd_dsply_bttns_comm	Baud Rate: 9600 BPS Protocol: I2C Communication Other: Synchronous I2C
hm_lcd_dsply_bttns_otsd_usrout	Type: Numbers (water consumption) and Units Usability: Output needs to be understandable for 9 out of 10 people.
hm_lcd_dsply_bttns_mcrcntrllr_dsig	Other: HIGH/LOW Digital Signal V_{MAX}: 5 V V_{MIN}: 0 V

Since the project will be using a 5V DC power supply, the LCD display and buttons will be primarily wired to the Arduino for data communication. Fig. 2 shows the wiring diagram used for testing and building the code for the Arduino Uno. The buttons will be wired through digital inputs and will initiate certain tasks when the microcontroller receives a high input. The LCD display with the integrated I2C communication interface will be wired to the serial clock and data pins. Table II below Fig. 2 has each of the inputs and outputs coordinated with the pin of the Arduino Uno and the functionality.

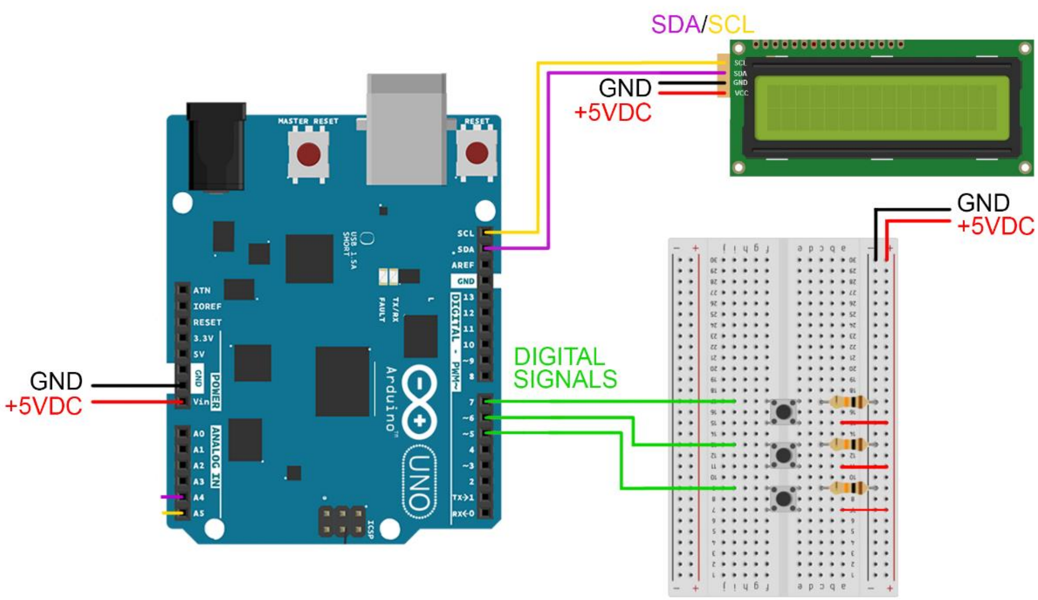

Fig. 2. Example wiring diagram for the Display and User Input Block.

TABLE II
PIN DESCRIPTIONS

Pin Name	Direction	Usage
SDA	Output	Serial Data Pin
SCL	Output	Serial Clock Pin
D5	Input	Button – LCD Display ON/OFF
D6	Input	Button – Cycles through output times on LCD Display (second, minute, hour, day, week, month)
D7	Input	Button – Changes output units on LCD Display (gallons, liters)

The design of the code was created based around the flow diagram in Fig. 3 which presents the different stages of execution. The code was designed knowing that there would be user and environmental inputs, network configuration, data storage and output. This output would need to be able to be transmitted over a wireless network which is where the network configuration comes into effect for new users.

4.3.3. General Validation

The selection for the LCD display with the I2C interface was primarily for ease of accessibility and removability. The project was proposed with a modular design for the PCB which will allow the user to replace components as needed for the device. The interface makes communication between the microcontroller and the LCD output simple with just having to connect the serial clock and data pins. Connecting the serial clock pins will align the clock of the microcontroller and the interface enabling synchronous data transmission over the serial data pin.

The design of the button layout was tested and selected due to the high/low output signal that would be captured by the digital pins on the Arduino. By connecting one side to the +5V power supply and the other side of the buttons to a pull-down resistor, the corresponding digital pin and ground. When the button is not pressed, the digital pin is connected to the ground resulting in a low signal. Once a button is pressed, the circuit is completed and the pull-down resistor pulls the +5V through the digital wire to ground. This will result in a high signal which will complete the corresponding task to that button.

4.3.4. Interface Validation

The interface properties for each input and output of the display and user input block are validated in Table III below. This table describes the what and why of the interface properties along with how the design of the block ensures that each interface meets or exceeds the design property.

TABLE III
INTERFACE VALIDATIONS

Interface Property	Why is this interface this value?	Why do you know that your design details <u>for this block</u> above meet or exceed each property?
otsd_hm_lcd_dsply_bttns_usrin : Input		
Timing: 750 millisecond delay between inputs	The delay was added to remove debouncing and to allow the user to precisely choose their selection.	In the code, there is a debouncing delay incorporated [4].
Other: 3 User button inputs	Three buttons were selected to be used for the design which enable the user to change various things about the device.	The three buttons were incorporated in the PCB design and through the Arduino code [4].
pwr_spply_hm_lcd_dsply_bttns_dcpwr : Input		

Inominal: 25 mA	When testing with an external +5 VDC power supply, the LCD output had a nominal current draw of around 25 mA.	The power supply for the project will be around +5 VDC, thus the nominal current will remain the same.
Ipeak: 37 mA	When testing with an external +6 VDC power supply, the LCD output had a peak current of around 37 mA. This testing was done to consider any fluctuates that may occur in the power supply.	The max current that the LCD output will require is 200 mA, but with the power supply that our project provides the LCD will not require that much current.
Vmax: 6 V	The absolute maximum voltage rating of the LCD display is 12 V [1], but the project will not supply more than 6 V.	The DC power supply will only supply 5 V, thus will not exceed the absolute maximum allowed by the LCD display
Vmin: 5 V	The minimum voltage rating of the LCD display is 3.3 V [1], but the power supply will provide a minimum of 5 V.	The LCD display has a typical voltage rating of 5 V, which will be provided by the power supply.
mcrcntrllr_hm_lcd_dsply_bttns_comm : Output		
Datarate: 9600 BPS	The default baud rate for the microcontroller is 9600 BPS, thus that rate is used for consistency [2].	In the initialization step of the code, the baud rate of the serial communication is set to 9600 BPS [4].
Protocol: I2C Communication	The LCD display utilizes an interface that communicates using I2C which simplifies the connections to just 2 pins, Serial Data (SDA) and Serial Clock (SCL) [3].	I2C simplifies data transfers by transferring messages instead of specific bits. The messages have an address frame and one or more data frames that contains the information.
Other: Synchronous I2C	The I2C interface utilizes synchronous serial data transmission.	The LCD display is wired to the Arduino through the SDA and SCL pins aligning the clocks of the two devices.
hm_lcd_dsply_bttns_otsd_usrout : Output		
Type: Numbers (water consumption) and Units	The LCD display outputs the water flow rate and consumption to the user.	The LCD display is connected to the Arduino hardware-wise, and the code sends the information through serial communication to the output [4].
Usability: Output needs to be understandable for 9 out of 10 people.	The information provided on the output should be understandable by 9 out of 10 people.	The output display can be changed to include other units of water flow rate and displays the information in a simple way through the Arduino code.
hm_lcd_dsply_bttns_mcrcntrllr_dsig : Output		

Other: HIGH/LOW Digital Signal	The buttons are configured to transmit a low signal to the digital pins and a high signal when pressed.	The digital inputs from the buttons are checked when a HIGH is seen in the loop, the corresponding function related to that button is executed.
Vmax: 5 V	When the button is pressed, the circuit is completed sending 5 V to the digital pin which indicates to the microcontroller that the button was pressed.	The buttons are powered with a 5 V circuit, once pressed, the 5 V will reach the digital pin indicating a HIGH signal.
Vmin: 0 V	When the button is not pressed, the circuit is not completed, and the digital pin will not receive any voltage.	When the button is not pressed, the output is grounded, thus having 0 V input indicating a LOW signal.

4.3.5. Verification Process

For this display and user input block, the minimum verification of the interface is required. The interface properties however include specific properties about communication protocols and digital signal inputs. The verification below will include some tests that will verify the interface communications and digital signals.

1. Connect the microcontroller to a PC using a USB cord with data transmission capabilities. Flash the Arduino Uno with the compiled code [6].
2. The system will power on and start looping through collecting, storing, and outputting data to the display using serial data communication.
3. Press the first button, putting the device into sleep mode. The output display will shut off. Pushing it again will turn the output display back on.
4. Press the second button, cycling the water consumption time frame on the output display.
5. Press the third button, changing the units of the output display.

PASS: To pass this test:

- The system must change the output display according to the button pressed by the user. This will prove that the block has satisfied portions of the engineering requirements.
- The system must still loop through the collection and storage of data while in sleep mode. This can be verified in the serial monitor while the output display is in the OFF mode.

The power supply will also be tested to ensure the sustainability and efficiency of the system. Each component is tested to see how much load they will require and then that is accounted for in the design of the power supply.

1. Connect the LCD display I2C interface VCC and GND to a DC power supply.
2. Reutilize the Arduino with the USB connection, wire the SDA and SCL to the respective pins from the I2C interface to the Arduino.
3. Set the power supply to 5V and start it.
4. Read the current either on the power supply or with a multimeter to see nominal current draw.
5. Turn off the power supply output and turn it up to 6V. Start it again.
6. Read the current either on the power supply or with a multimeter to see peak current draw.

4.3.6. References and File Links

4.3.6.1. References

[1] Handson Technology, “I2C Serial Interface 1602 LCD Module,” https://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf, Accessed: 3/4/2022

[2] Arduino, “Arduino Uno R3 Datasheet,” https://docs.arduino.cc/static/c5af4b1dbd0260fb58643409d2ada778/A000066-datasheet.pdf, Accessed: 1/10/2023

[3] Hackster, “Learn To Use LCD 1602 (I2C & Parallel) With Arduino UNO,” https://www.hackster.io/Hack-star-Arduino/learn-to-use-lcd-1602-i2c-parallel-with-arduino-uno-f73f07, Accessed: 2/2/2023

4.3.6.2. File Links

[4] SharePoint. “water_flow_arduino.ino” https://oregonstateuniversity.sharepoint.com/:u:/s/ECE441/ESAR0L0x3DtPuHLXHF5DnioBvyn9XTFQbnVXfaHilMR8Fw?e=HZj8ki, Accessed: 2/8/2023

4.3.7. Revision Table

3/10/2023	Christian Ovchinikov: Revised general validation and figures.
3/4/2023	Christian Ovchinikov: Added to general validation and references.
3/1/2023	Christian Ovchinikov: Added to interface validation.
3/2/2023	Christian Ovchinikov: Added to design including black box diagram and table.
2/28/2023	Christian Ovchinikov: Wrote description and added verification plan.
2/24/2023	Christian Ovchinikov: Initial document creation.

4.4. Enclosure

4.4.1. Description

The enclosure will be designed in Blender and will coordinate heavily with the PCB design. This includes housing the manufactured PCB board, LCD display, and buttons for the users to interact with. The design of the enclosure will have to take the user interaction portion into consideration for an effective product. The enclosure will also incorporate a gasket system to seal the system. This will allow the system to withstand colder temperatures and some minor water exposure.

4.4.2. Design

The enclosure has gone through numerous design revisions that eventually reach our final draft. There were many variables and other constraints that played roles in the changes of the enclosure design. To begin with, the team wanted to ensure that the final design was at least water resistant and could house the entirety of the system. This includes the power supply which would be a decent size due to the AC-AC converter, the PCB with all the modules, the buttons and the input and output cables. The input cables would be an outlet to dongle cord that would allow our system to receive 120 VAC. The design includes a power switch at the dongle that is used to shut down all power to the system.

The final design includes a removable lid to allow for easy access to the electronics. This ties into the PCB design being modular as the team wanted the system to be easily interchangeable to someone with enough electrical knowledge. Some safety precautions would need to be addressed since there is 120 VAC inside the device.

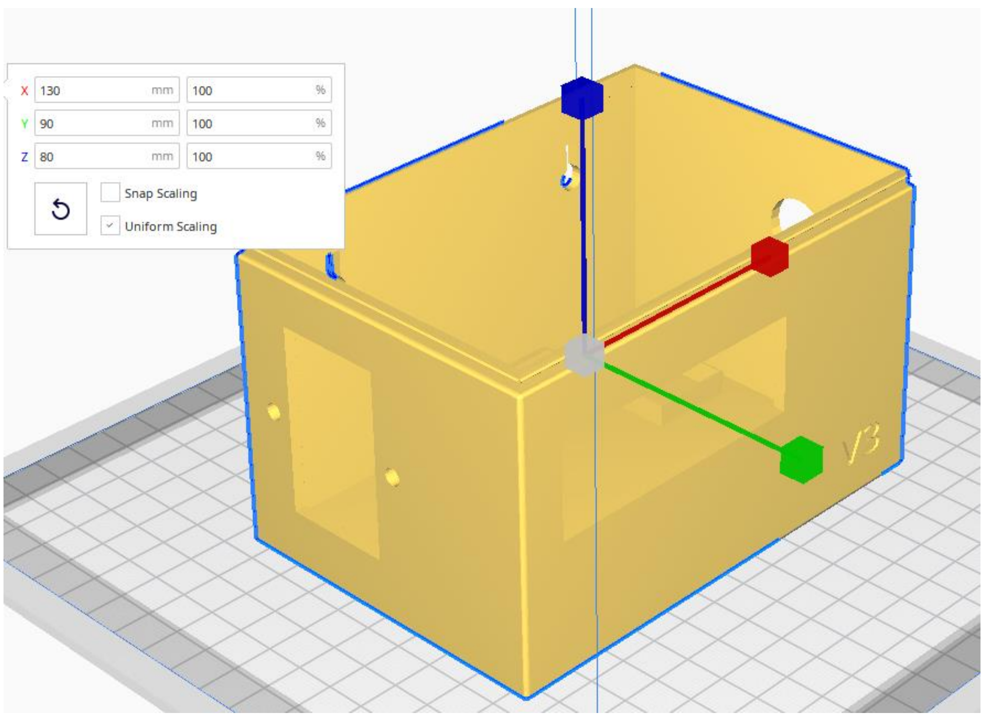

Figure 4.5.2: Enclosure Design

4.4.3. General Validation

The design of the enclosure has continuously validated itself with the constant revisions to improve the final design. The first revision was simple with a hinge-style lid which was later changed to a lip-style that would eliminate water from easily getting into the enclosure. The team anticipates the device being inside a garage or momentarily outside and could be exposed to various environmental and weather conditions. This is why the lip-style lid was incorporated as it would lessen the chances of rain or moisture getting into the device. Through our design impact assessment, one of the concerns that arose was that the device could cause bodily harm since there will be 120 VAC in the device with a removable lid. A warning of electrical shock will be placed near the AC-AC converter.

The enclosure will be 3D printed using UL rated filament to minimize any concerns of fire-related accidents due to the electronics. UL certified materials can prevent the spread of fires due to electrical overload, component failure, or loose connections. The plastic material can also prevent the user from being shocked if there is a short that could be touching the enclosure [1]. This is a big reason why the use of UL certified material is important and is incorporated into the design of the enclosure for the system.

4.4.4. Interface Validation

For this enclosure block, there are no interfaces to have validations for. Since there are no interface properties as this is a standalone block, the enclosure will be printed and present to validate the block.

4.4.5. Verification Process

For this enclosure block, the minimum verification of the interface is required. There are no interface properties as this is a standalone block. The enclosure will be printed and present at the time of verification.

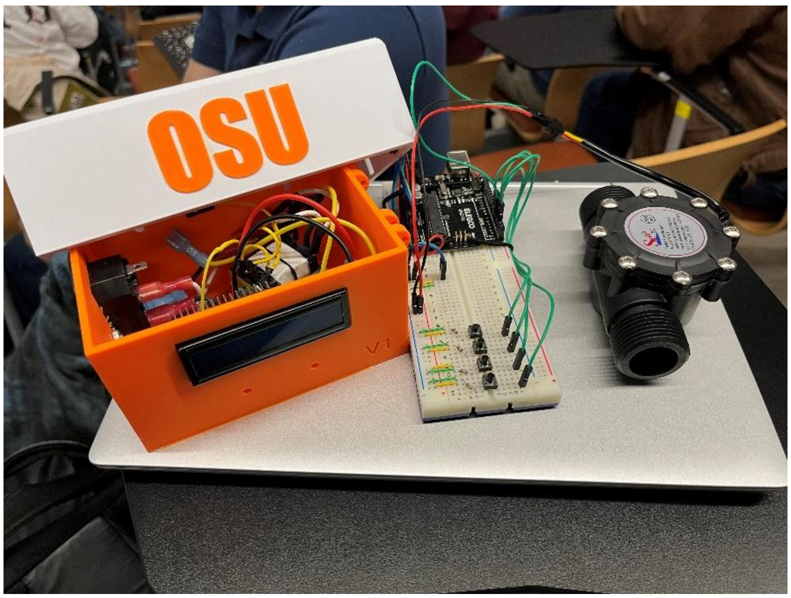

Fig. 1. The first version of the enclosure with the prototype electrical system.

4.4.6. References and File Links

4.4.6.1. References

[1] HCL DFMPro, “UL Certification and Why It Is Important to Plastics Part Designers!,” https://dfmpro.com/blog/ul-certification-important-plastics-part-designers/, Accessed: 3/4/2022

4.4.6.2. File Links

4.4.7. Revision Table

3/12/2023	Carlos: Added enclosure in Slicer view
3/11/2023	Christian: Merged with Enclosure document

4.5. PCB Design

4.5.1. Description

A PCB will be tested, designed, and manufactured to house the power supply, Wi-Fi transmitter, LCD Display, Buttons, Water Flow Sensor, and an Arduino Uno. The design will allow for the major components such as the Arduino Uno, Water Flow Sensor, Wi-Fi transmitter, and LCD Display can be replaced in the product. This will be done using a modular PCB design approach. The PCB will be designed and simulated before creating a prototype system. The prototype system will be probed and tested to ensure that the design is ready for manufacturing.

4.5.2. Design

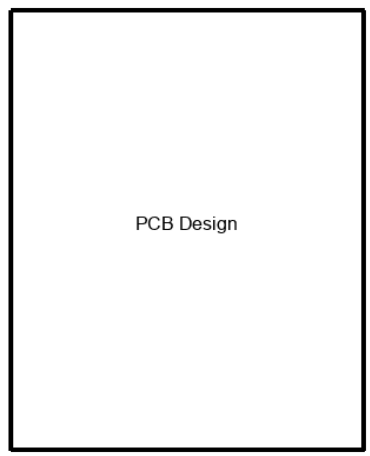

Figure 1: Block Diagram

This diagram is on the PCB design which does not include the interfaces because all connections should be in the PCB which will be inside of the enclosure. This needs to be designed, tested and manufactured to house the power supply, Wi-Fi module, LCD display, Buttons, water flow sensor, and Microcontroller.

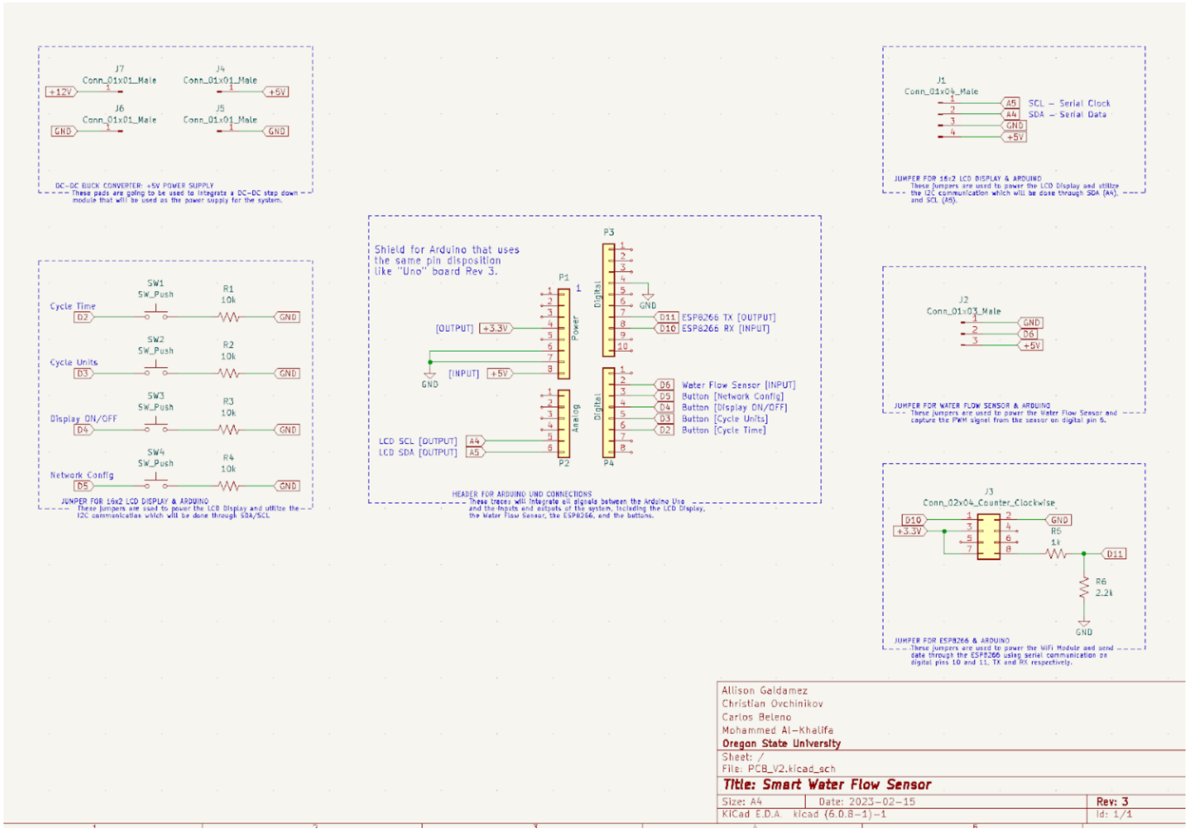

Figure 2: Kicad schematic

In this schematic explains each block where are needed to be connected into the PCB board, so for the water flow sensor is going to be jumper for the water flow sensor and Arduino uno, so these jumpers are used to power the water flow sensor and send a signal from sensor to pin 6. Also, headers for Arduino uno are used for the traces that will integrate all signals between Arduino uno, input and output of the system including LCD display, water flow sensor, ESP8266, and buttons. There are also switches that are used in order to control the LCD output, so they are functioning on cycle time, cycle units, display on/off, and network configuration.

The power supply will be considered as pads to be used to integrate a dc-dc step down module that will be used for the power supply. Also, Jumpers for the ESP8266 and arduino are for using serial communication on digital pins 10 and 11, Tx and Rx.

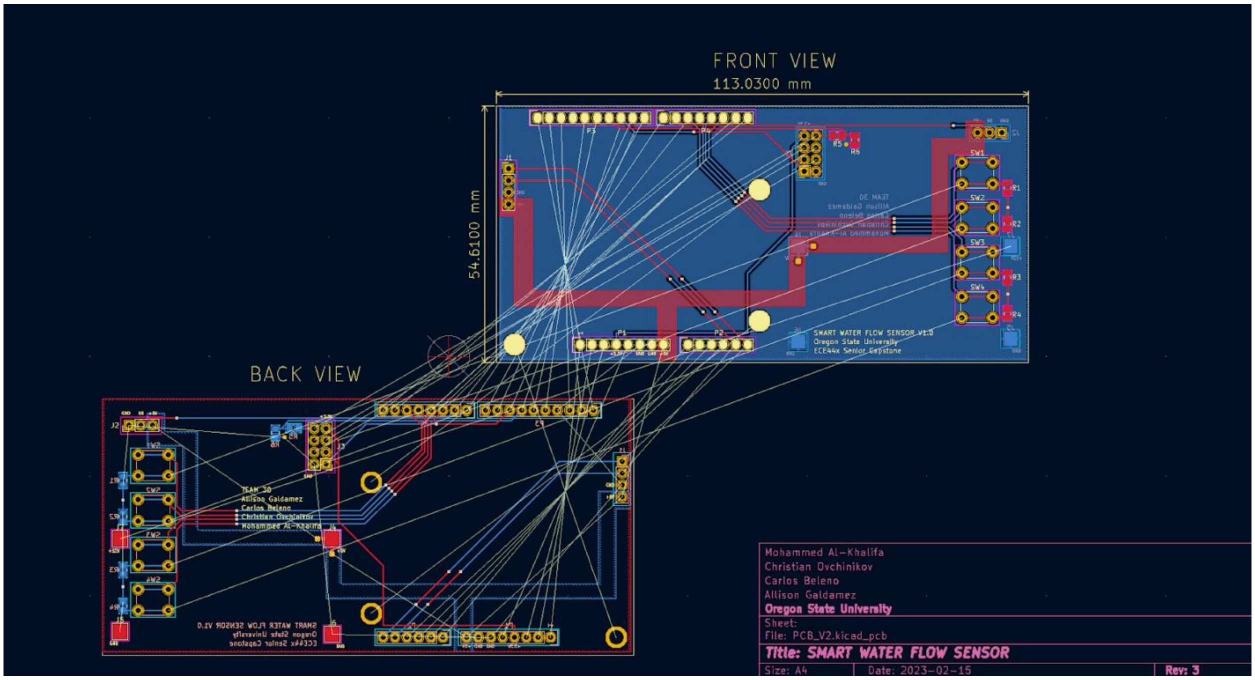

Figure 3: Kicad layout

I have already converted the schematic to layout to look like the real PCB board. In the PCB it has a front and back view in order to represent the components that are used. I added traces from the push buttons to be connected to the headers of Arduino uno in order to send the signal between Arduino uno and input and output. Also, there are holes on the PCB board that are shown in the front and back side.

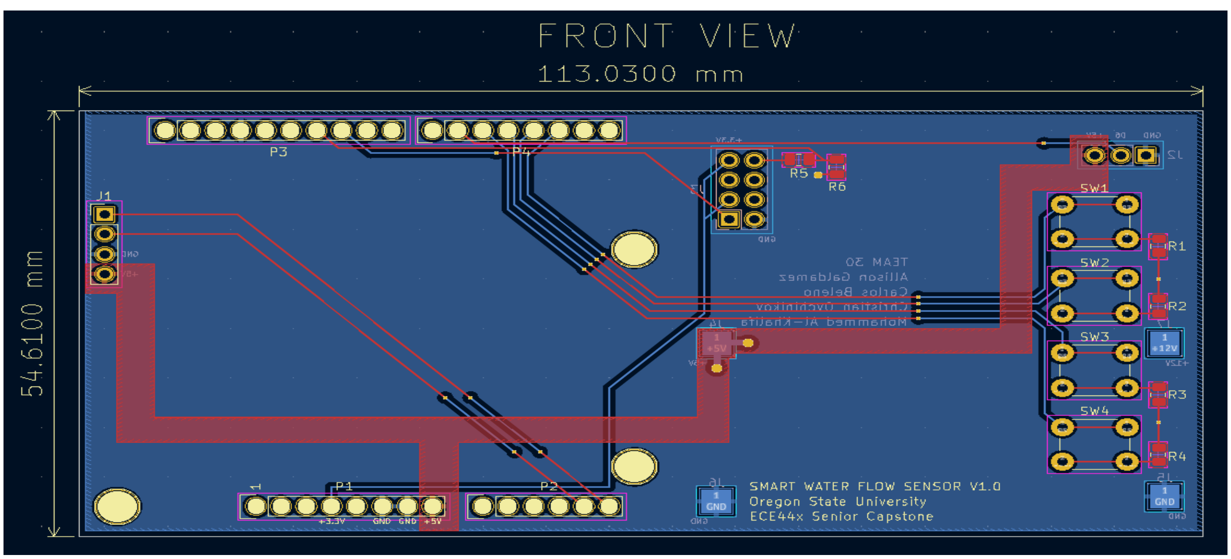

Figure 4: Front schematic

This represents the final layout including all components in the PCB board. It includes 4 buttons in front of the PCB and LCD screen in order to display the results of total volume in different units. Also, headers for all components that are needed to be connected to Arduino and traces that are integrating all signals between Arduino uno and inputs/outputs of the system.

4.5.3. General Validation

The block design section of the PCB design project includes the major components of the system, such as the power supply, Wi-Fi transmitter, LCD Display, Buttons, Water Flow Sensor, and an Arduino Uno. These components have been carefully chosen to meet the needs of the system and provide the required functionality.

The modular PCB design approach that has been adopted allows for easy replacement of the major components, such as the Arduino Uno, Water Flow Sensor, Wi-Fi transmitter, and LCD Display, if required. This approach offers the advantage of reducing downtime and repair costs, as well as allowing for quick and easy upgrades or customization of the product.

To ensure that the PCB design meets the needs of the system, the design will be tested, simulated, and validated before manufacturing. The simulation and testing process will ensure that the PCB design is feasible, reliable, and functional, and meets the expected performance requirements.

The simulation process will be done using specialized software, such as KiCAD, to model the behavior of the system under different operating conditions. This will allow for the identification and correction of any potential issues, such as timing or signal integrity problems, before the hardware is built.

The prototype system will be probed and tested to ensure that the design is ready for manufacturing. During the validation process, the major components of the system will be tested independently, and the system will be tested to ensure that it functions as expected under different operating conditions.

Overall, the block design section of the PCB design project has been carefully planned to meet the needs of the system and provide the required functionality. The modular design approach allows for easy replacement of major components, while the simulation and testing process ensures that the design is feasible, reliable, and functional.

4.5.4. Interface Validation

This PCB block does not include any interfaces since all connections will be connected into it, and this will be for the entire system. PCB should be connected to the power supply in order to control the amount of current and voltage getting into PCB. The rest of the components should be headers of the PCB such as water flow sensor, LCD screen, three buttons, microcontroller. The enclosure will include the rest of the blocks except the water flow sensor.

4.5.5. Verification Process

Steps for designing PCB:

1. Using the Kicad to draw schematic diagrams for the blocks
2. Once the schematic diagram is done, you will need to convert it to layout and add traces to the headers such as switch push to the headers of the Arduino uno.
3. Viewing the 3D model of designed PCB in Kicad
4. Ordering the PCB from PCB way in order to have the headers of the blocks connected
5. Once the PCB delivered, the components need to be soldered on the PCB

6. The power supply should be connected to the PCB
7. Verifying that the PCB is working properly based on the schematic that you designed in order to show how it works successfully.

4.5.6. References and File Links

4.5.6.1. References

“Affordable prototype PCB manufacturer in China,” *Custom PCB Prototype Manufacturer*. [Online]. Available: https://www.pcbway.com/. [Accessed: 19-Feb-2023].

4.5.6.2. File Links

1- **PCB Design V3**

2- **Schematic V3**

4.5.7. Revision Table

Date	Revision Description	Done
02/14/2023	Done with schematic	Mohammed Al-khalifa
2/16/2023	Added traces in schematic and layout	Mohammed Al-khalifa
2/19/2023	Ordered PCB	Mohammed Al-khalifa
2/26/2023	Done with validation	Mohammed Al-khalifa

4.6. Computer Program

4.6.1. Description

The computer program block, also referred to as UI, is an application that connects to the project through a local internet connection. This program will communicate with the user's router, which will act as the common node for interaction between the program and the system's microcontroller. Once a connection is established between the program and the system, the microcontroller will start sending water flow data through the system's Wi-Fi transceiver. These data packets will be received by the router and sent to the computer program using TCP as the main communication protocol. The computer program will store and timestamp the received data. It will then calculate the water volume (default being gallons) used per time unit (default being hourly). The program will then display all of the stored data in an array fashion, from oldest to newest. Displaying of stored data occurs regardless of Wi-Fi status, if a connection was not stablished at program start-up, it will display the current stored values.

The computer program will allow the user to view the water consumption data from the system. The code will be written in Python and will communicate with the system over the wireless network connection that the computer is hosted on. The python code will have to acquire the data from the system and then translate that into the user interface provided by the executable. The user interface should be intuitive and easy to navigate. There will also be an option to translate that data into a different unit of measurement to include international users.

4.6.2. Design

The computer program (UI) block includes a main program that handles internet connectivity, data collection, and data display on the computer that is running the program. The program has a single source of input, this being data received from the microcontroller (w-f_mdl_cmptr_prgrm_rf). The block's outputs are the displayed data stored (cmptr_prgrm_otsd_usrout), and a constant ping request being sent out from the computer to the router (cmptr_prgm_w-f_mdl_rf).

The program consists of two distinct sections, the Python based connectivity and calculations segment, and the HTML output segment. The Python code segment is responsible of handling the TCP/IP communication happening with the microcontroller's Wi-Fi transceiver through the local network's router. In addition, it is also responsible of calculating and converting the data received into different units. The HTML part of the code is responsible of formatting the data processed by the Python segment and compartmentalizing it in a language easily displayed by a web browser. The HTML code is solely responsible of formatting and displaying a readable output to the user.

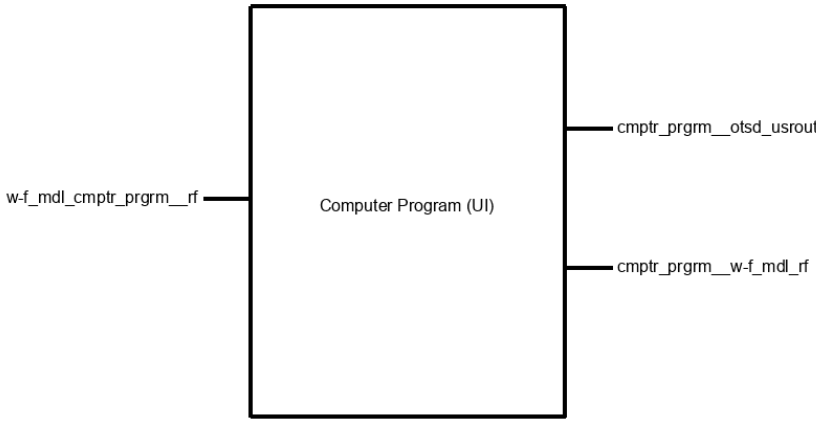

Fig. 1 Black Box for Computer Program

As seen in Fig. 2, the program's pseudocode/flowchart can be seen. The only step that is done with HTLM is "Display Stored Data", as all the rest are done in the python section of the code. To access the HTML webpage, the user will have to enter a specific IP address given by the program into a web browser, as they will only be able to see limited amounts of data in the computer program's direct output.

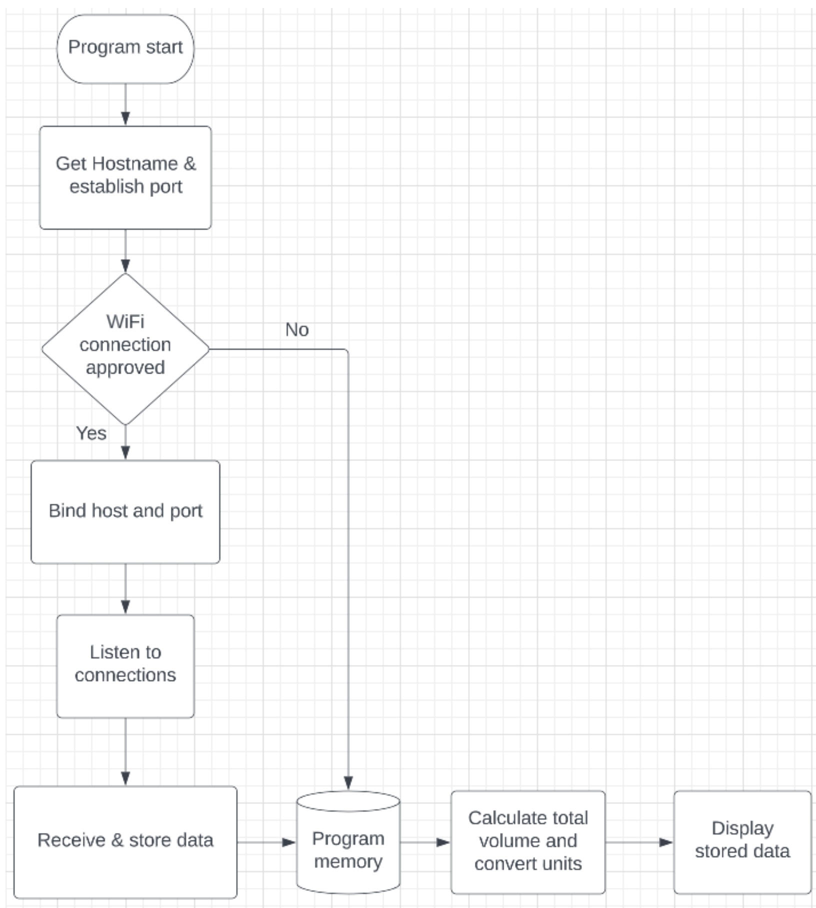

Fig. 2 Computer Program Flow Chart

4.6.3. General Validation

The computer program block was designed to achieve the goal of data transmission and long-term storage, with the added benefit of displaying and transforming the data into a more digestible format for the user. Other blocks that interact with the program are the microcontroller and the Wi-Fi module; however, it is important to note that the interaction with the microcontroller is only by proxy, as all communication is done through the Wi-Fi module. The Wi-Fi functionality does imply the need for a router that the user has the credentials to access, in addition to a computer that is able to run the program. There is no preference between having the router itself connected directly to the computer running the program (ethernet) or wirelessly through Wi-Fi.

Some of the challenges faced during the completion of this block are technical in nature. For example, setting up a connection from the local computer running the program and the router is a manageable task; however, communicating and storing information from that is being sent from the microcontroller to the computer program is a steeper challenge.

An alternative solution to this block could have been setting up a web page to which the system's Wi-Fi transceiver would send all information to. This website would require a system to store information in on the back end, while the front end would display the data in a similar manner to the current solution. Further steps can be taken as to setting up an external server that could allow the user to access the stored data from any network (not just the one the system is connected to). This would change some of the current interface definitions and thus, not considered for the purposes of this block validation. This modification would shift the server portion from the current program, into a different server all together, requiring a connection to be established between the local router and a different router that is connected to the different server. However, the overall concept remains the same.

The computer program block is designed to function in clearly distinct stages that initialize and manage wireless connectivity and data management. The program starts independently from any other component in the system by declaring a local host and port in the router that the computer being used is connected to. It then creates a socket in the host network, which is bound to the specific host and port mentioned previously. If the socket binding process were to fail, the computer program will display an error code representing what went wrong. An important detail to mention is that a predetermined, fixed port number is chosen by the program. The port number is 5678, the value chosen holds no significant importance other than being greater than 1023, since ports lower than that are reserved in most networks for other purposes. It is completely possible that this port (5678) is already being used by another unrelated subsystem in the user's network, this will result in a failure to establish a bound socket and preventing the program from receiving data from the microcontroller. A possible solution to this is having the computer program generate a large random number (from 0 to 4096) and add that to 1023 to ensure the program is not using any of the first 1023 reserved ports. The only issue with this is that the microcontroller would also require the number generated at random, which could easily be communicated by the user after the socket has been bound by the program.

If the program successfully binds a port in the local network's router, the Wi-Fi module integrated to the microcontroller is able to communicate directly and exclusively to that specific port in the local network's router. With a connection established both the Wi-Fi module and computer program are set up to communicate using the TCP/IP communication protocol. With the router acting as the midpoint of communication between the system and computer program. It is relevant to note that the speed between the system and the program are wildly different. The Wi-

Fi module is communicating with the router at 9600 baud (or ~960 Bps) which for the amount of data being sent is more than quick enough. While the communication speeds between the computer program and the router depend wholly on the users internet speed. For the purposes of testing, the computer on which the program is running on ~56 MBps, but the actual speed between the program and router is variable from user to user.

If working correctly, this block enables easy access to the information stored in the microcontroller wirelessly, as long as the user is connected to the same Wi-Fi network as the system. It also does not make a difference whether the computer running the program is connected to the local network's router wirelessly or through an ethernet cable. While a computer is required to run the program, any device in the network can access the IP provided by the program to view the data collected.

4.6.4. Interface Validation

Interface property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property
w-f_mdl_cmptr_prgrm_rf		
Wireless Communications: The program will have input sent by the system's transceiver through the local network.	This method lets the program receive data from the system wirelessly.	By wirelessly connecting to the router's local network, both the system and program are able to communicate by sending packets of data.
Data rate	111 Mbps is the average internet speed in Oregon, 150 Mbps is a reasonable top speed.	this represents the connection speed between the computer and the router.
Messages	Messages: Protocol uses 8 bits for the body of the message, 2 for start and stop bits. 1 Byte per message	standard USART message size (ignoring start and stop bits)
Protocol	Protocol: Wi-Fi, TPC/IP Wireless data transfer	Wi-Fi (wireless communication)
cmptr_prgrm_otsd_usrout		
Type: Numeric value (display volume of water consumption sent by the microcontroller)	The values represented in the graph and table will be determined by data received	The computer program uses HTML that displays the

	from the microcontroller (variable)	stored data on a webpage in the local network
Usability:	Browser Based (output will be displayed in commonly used webbrowser)	Output is provided in one of multiple common browsers (Chrome, Firefox, Edge, etc.)
Usability:	Intuitive to 9 out of 10 users	Clear output that is easy to interpret

cmptr_prgm_w-f_mdl_rf

Wireless communication: Program will establish Wi-Fi communication to a predetermined port in the local network	This method only requires the computer the program is being ran on to have access to the local network.	The computer program will attempt to establish a connection to the local network, but it will present any available data regardless of connection status
Protocol: Information outgoing from the computer program to the router uses TCP as its protocol	TCP/IP This is the method of communication between the program and the microcontroller	TCP/IP is one of the possible protocols that can be used, it is compatible with the chosen Wi-Fi transceiver on the system
Protocol	Wi-Fi	Communication
Datarate:	9600 Baud	Usual data rate for Arduino Uno

4.6.5. Verification Process

To verify this plan, the following components are required: a microcontroller, a power supply for the microcontroller (3.3 V), a Wi-Fi transceiver (ESP8266 or ESP32), a method of creating data (water flow sensor or analog button), a computer capable of running Python3.11, and a Wi-Fi router with known credentials.

To start block verification, accounting for all above components working, the user must go through the following steps:

1. Turn on the system and launch its internet connection functionality (w-f_mdl_cmptr_prgrm_rf).
2. Search for the system's network on your computer or smartphone and connect to it.
3. Input the local network information (name & password) in the corresponding fields.
4. Use a computer that is connected to the same local network and launch the program.
5. The program must state if a connection to the local network was established (cmptr_prgm_w-f_mdl_rf).
6. Use the system to measure water flow at the maximum rate allowed by the spigot for 10 seconds.

7. The computer program must have the updated data received after step 5 and display it (cmptr_prgrm_otsd_usrout).
8. The user must change the current unit being displayed in the computer program.

If all compentents are working as intended, the computer program output should state that a successful connecition to the local network was made, followed by confirmation of socket binding. After that, the program will constantly "listen" to connections from the Wi-Fi transceiver to the specific port the socket was bound to. As the system is powered on and ready to send data, the data stored will be sent to the router, and from the router to the program. The program will store the received data and will display it through HTML into a simple webpage hosted on the local network. If all of the above occurs, then the program is considered to have successfully accomplished its goal.

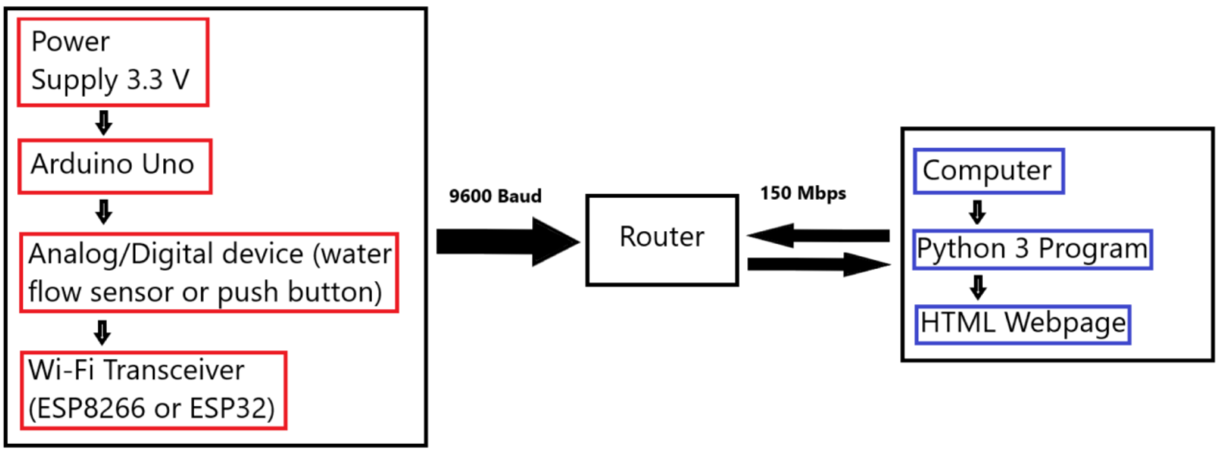

Fig. 3: System to Program diagram

4.6.6. References and File Links

4.6.6.1. References

[1] Espressif Systems, "ESP8266EX," Dec. 2015, [Revised Nov. 2022]. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf, Accessed: 3/4/2023

[2] E Python Basics, "Create a Python Web Server," Jan. 2021 https://pythonbasics.org/webserver/#Web-server, Accessed: 11/4/2022

4.6.6.2. File Links

4.6.7. Revision Table

5. System Verification Evidence

5.1. Universal Constraints

5.1.1. The system may not include a breadboard.

The system only utilized breadboards for testing of the components before designing and manufacturing a PCB for the system. The image seen below in Fig. 1. shows that the system will not use a breadboard in the final product.

Verification of Satisfied Constraint:

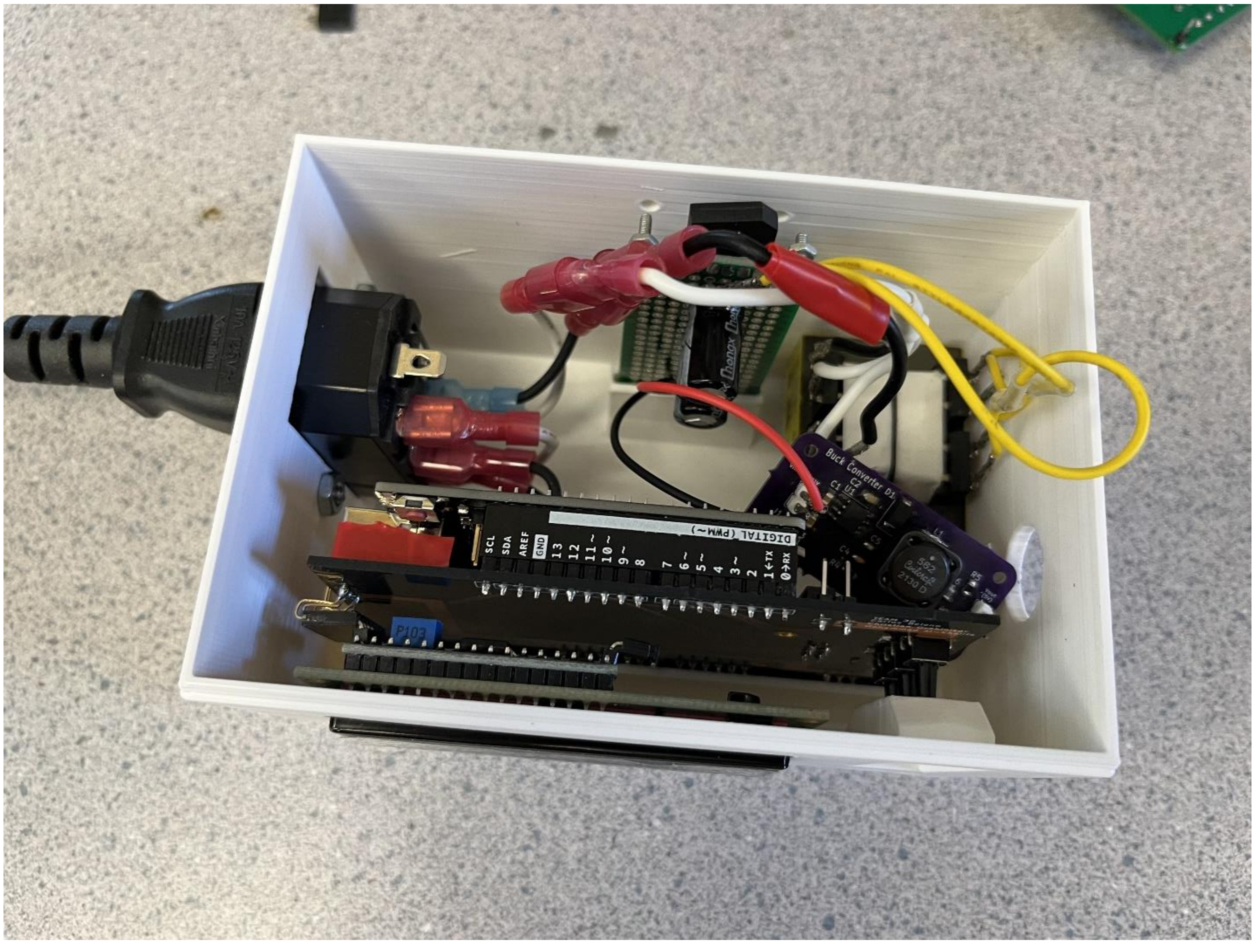

Fig. 1. The project PCB proving that no breadboard is used during system verification.

5.1.2. The final system must contain a student designed PCB.

The system contains a student designed PCB that has one flaw with the LCD output connector pins being swapped around. The final product utilizes wire jumpers that will allow the four connections to the I2C interface. Fig. 2. and 3. below show the front and back view of the manufactured PCB. The original designs had enough components to satisfy the quantity of surface mounted pads constraint. However, we had lost the team member that was responsible for the DC-DC step down block, so a module was used instead.

Verification of Satisfied Constraint:

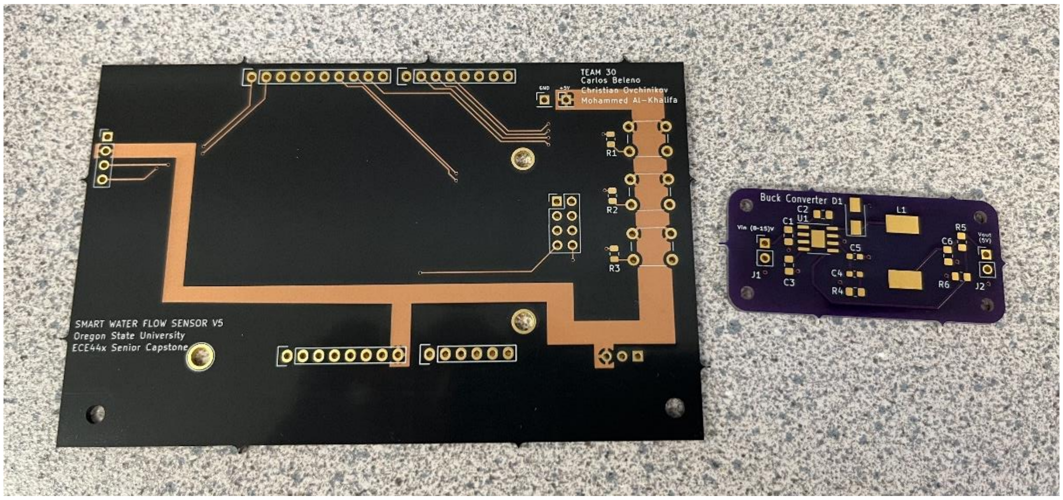

Fig. 2. Front view of the PCB design for the project.

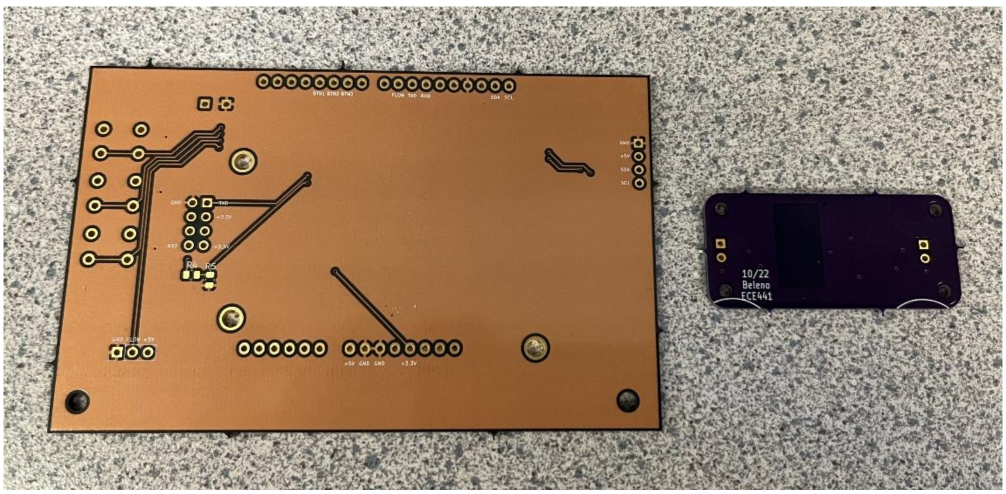

Fig. 3. Back view of the PCB design for the project.

5.1.3. All connections to PCBs must use connectors.

The front of the PCB seen in Fig. 4. has one set of connectors that are for the output display. The back of the PCB seen in Fig. 5. Has three sets of connectors, one set running parallel to each other for the Arduino Uno, one set of three at the bottom left for the water flow sensor, and one set of four for the DC-DC step down seen at the top left. Referring back to Figure 1, the system assembled uses removable connectors for each connection.

Verification of Satisfied Constraint:

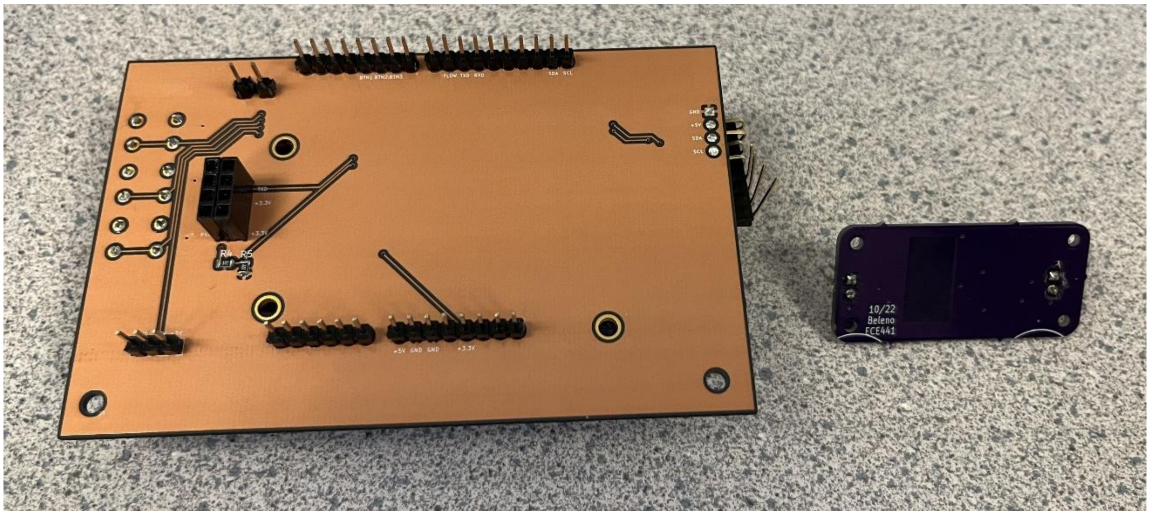

Fig. 4. Front view of the PCB connections for the project.

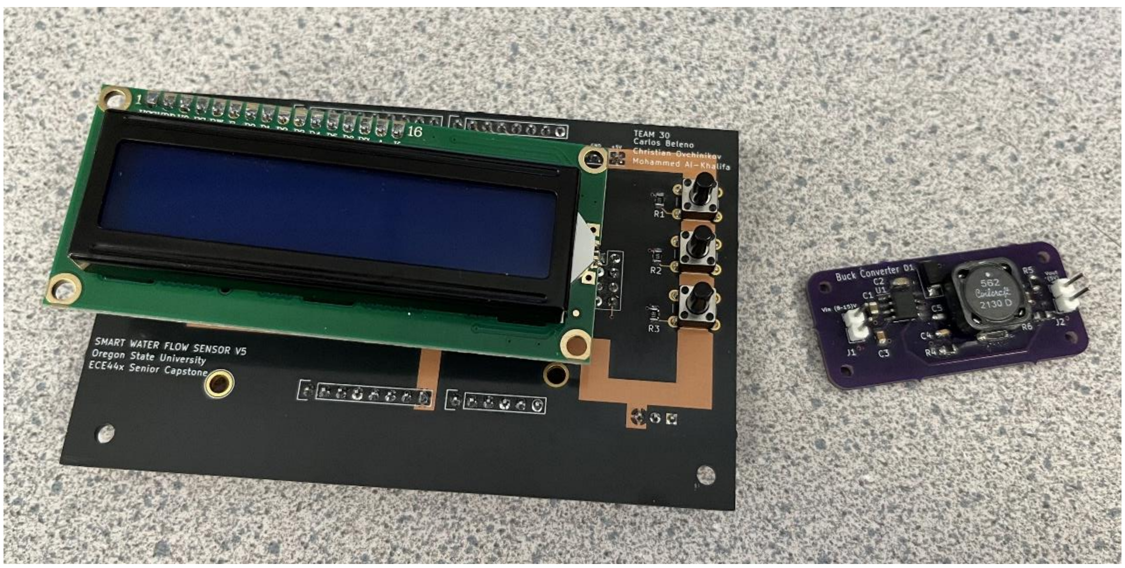

Fig. 5. Back view of the PCB connections for the project.

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

To prove the efficiency of the power supply, measurements were taken before and after the buck converter. Figures 6 and 7 show the multimeter voltage reading from probing before and after the buck converter. The voltage produced by the AC-DC rectifier is 12.54 VDC. This is then stepped down to 4.98 VDC to power the rest of the components in the system.

Current readings were then taken by completing the circuit using the multimeter. The circuit was completed before and after the buck converter to show the current different across the step-down. The current reading in Figure 8 shows 40.7 mA before the buck converter. After the module, the multimeter reads 80.3 mA. According to Ohm’s law, to maintain the same power usage across the step-down, as the voltage is reduced, the current must increase. To prove the efficiency, the output power is divided by the input power. The values used in the following calculations were obtained from multimeter readings found in Figures 6, 7, 8, and 9.

Verification of Satisfied Constraint:

$$\frac{\text{output power}}{\text{input power}} = \frac{4.98\ \text{VDC} * 80.3\ \text{mA}}{12.54\ \text{VDC} * 40.7\ \text{mA}} = \frac{0.399\ \text{W}}{0.508\ \text{W}} = 78.5\%\ \text{efficiency}$$

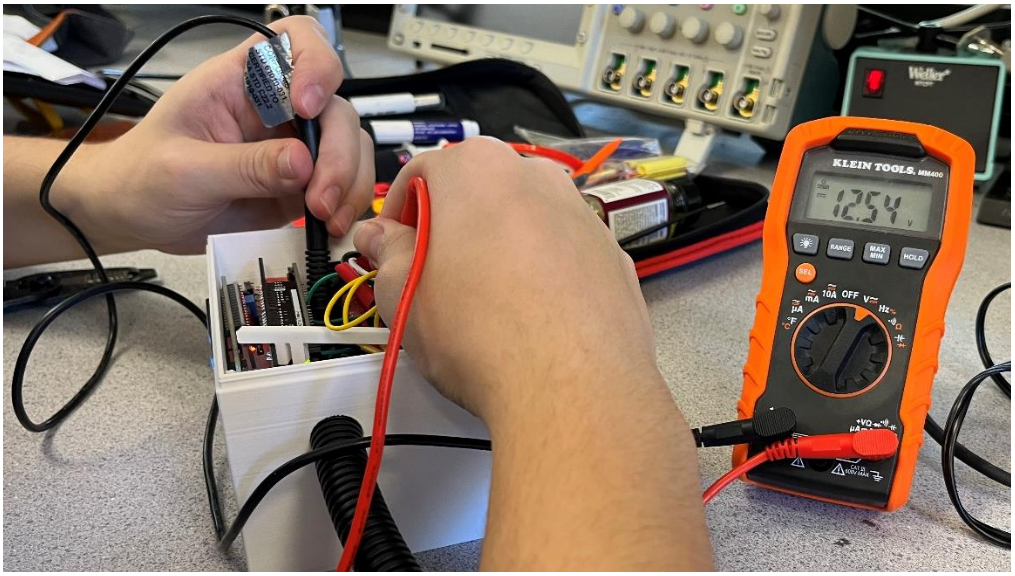

Fig. 6. Probing for voltage before DC-DC step-down in the power supply.

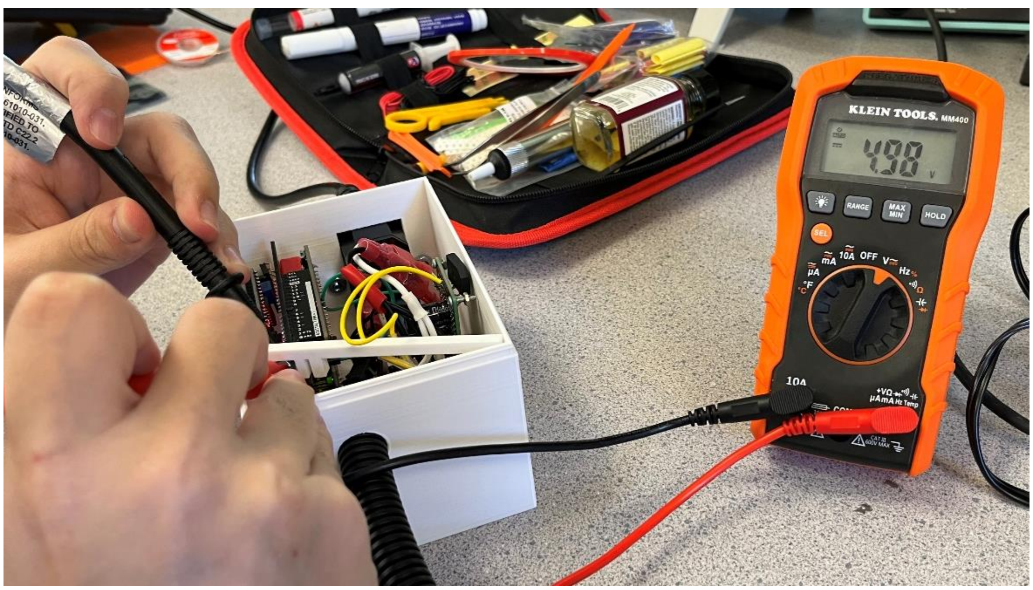

Fig. 7. Probing for voltage after DC-DC step-down in the power supply.

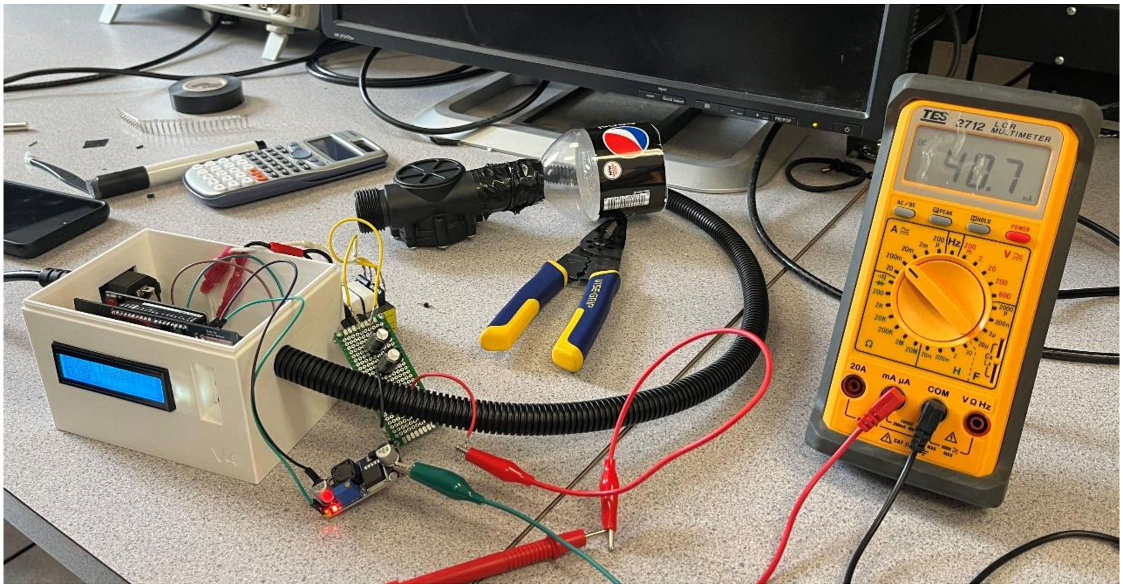

Fig. 8. Measuring current with multimeter before DC-DC step-down in the power supply.

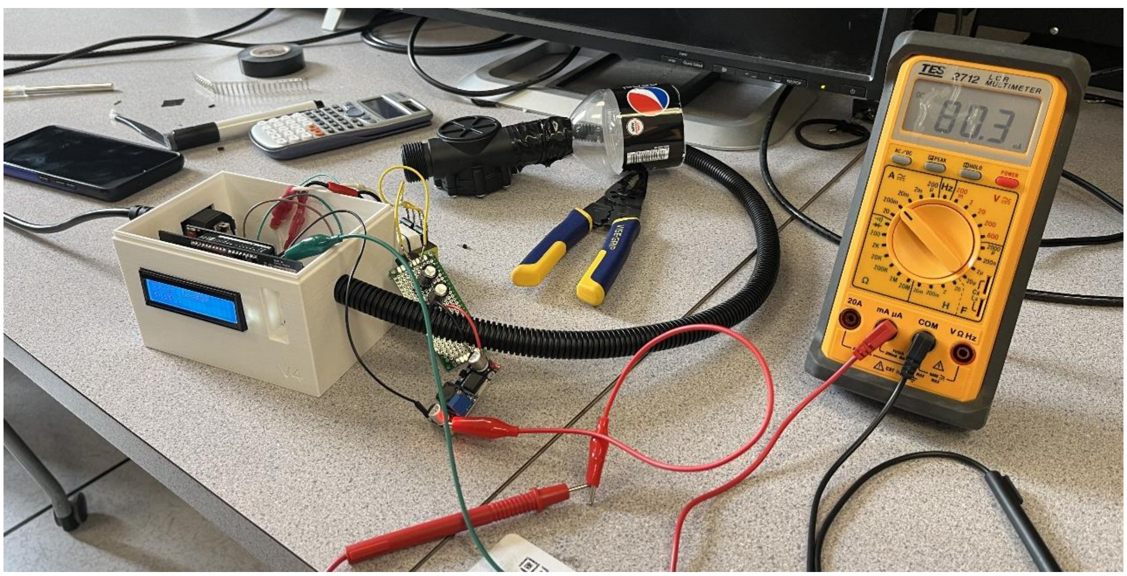

Fig. 9. Measuring current with multimeter after DC-DC step-down in the power supply.

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

The system uses a mix of built and bought modules for simplicity and ease of use. The PCB design was centered around a modular design that would allow the user to change out components when needed to some degree. Table I on the next page shows the blocks and whether they were built or bought. A validation is also included for each block describing why the verdict was given.

Verification of Satisfied Constraint:

TABLE I
BUILT OR BOUGHT MODULES

Block	Built or Bought?	Validation
Power Supply	Built	The power supply block was half and half, where the AC-DC converter was built, but the DC-DC converter was bought due to some last-minute team issues.
Microcontroller (Code)	Built	The microcontroller block was built as well because this block was a code block.
HMI (LCD Display & User Buttons)	Bought	The HMI block was bought because the I2C LCD display and buttons were bought and only utilized the microcontroller code to work.
Water Flow Sensor	Bought	The water flow sensor was bought as well since it is also a sensor that uses the microcontroller code to work.
Enclosure	Built	The enclosure was designed by our team and 3D printed which results in it being built.
PCB	Built	The PCB was designed by our team as well which is also built.

5.2. Requirements

5.2.1. Accuracy

5.2.1.1. The system should be accurate.

5.2.1.2. The system shall report values within 7% of 5 gallons, filled at a rate of at least 2 gallons per minute reading.

5.2.1.3. Verification Process

Verification Method: Analysis

Testing Process:

1. Using 1 gallon of water in the system, open the water Jug.
2. Verify on the LCD output that at least 0.3 gallons per minute is achieved.
3. Shut the Jug once the 1 gallon of water runs through the water flow sensor.

4. Compare the released amount of water, to what the system captured from the water flow sensor.

5. Calculate the assuming the target of 5 gallons.

6. Calculate the percent error of the system and verify if it was within 7%.

Testing Pass Condition:

The water flow reading is within 7% of the final output of water.

5.2.1.4. Testing Evidence

The video evidence showing the passing condition is met is linked in the File Links portion of this section of the document [1]. Fig. 6. below shows the system recording 0.99 gallons of water through the system when pouring a 1 gallon water jug through it. The video also shows the water flow rate being 0.04 gallons per second which is above the required 2 gallons per minute requirement (0.03 gallons per second). When starting the second clip in this video, the system is shutdown and turned back on showing that the stored data was retained. The final display reading was 1.97 gallons while pouring an overall 2 gallons through the system.

$$\frac{0.99\text{ gal} - 1\text{ gal}}{1\text{ gal}} = 0.01 \approx 1\%\text{ error}$$

Fig. 6. Snippet of the video evidence showing 0.99 gallons on the output.

Evidence Link:

https://www.youtube.com/watch?v=LYrH9WusKRI

5.2.2. Data Storage

5.2.2.1. The system will store water consumption data.

5.2.2.2. The system will store water consumption data in memory for 30 days.

5.2.2.3. Verification Process

Verification Method: Analysis

Testing Process:

1. Let the system run for 5 minutes
2. Using the stored size of data for the 5 minutes test, calculate the theoretical amount of data required for 30 days.
3. Validate that the system has the storage space to accommodate 30 days' worth of data.

Testing Pass Condition:

Ensure the storage has the memory capability to store the calculated size of 30 days' worth of data.

5.2.2.4. Testing Evidence

As the system is intended to reduce the waste of water, the test was shortened to eliminate any waste caused by us. Calculations were made based off the time it took to pour 1 gallon of water and compared that to the storage capabilities of the Arduino Uno.

The video evidence shows that it took 23.8 seconds to pour 1 gallon of water into a bucket. Assuming the device were running for 30 days straight, with the same average flow rate of 0.04 gallons per second, the estimated total consumption can be calculated as follows:

$$\frac{1 \text{ gallon}}{23.8 \text{ seconds}} = \frac{0.04 \text{ gallon}}{\text{second}}$$

$$30 \text{ days} * \frac{24 \text{ hours}}{\text{day}} * \frac{60 \text{ minutes}}{\text{hour}} * \frac{60 \text{ seconds}}{\text{minute}} = 2{,}592{,}000 \text{ seconds}$$

$$\frac{0.04 \text{ gallon}}{\text{second}} * 2{,}592{,}000 \text{ seconds} = \mathbf{103{,}680 \text{ gallons}}$$

The average flow rate of a home is 6 to 12 gallons per minute which would give an approximate total consumption of:

$$\frac{12 \text{ gal}}{\text{minute}} * \frac{\text{minute}}{60 \text{ seconds}} = \frac{0.2 \text{ gal}}{\text{second}} * 2{,}592{,}000 \text{ seconds} = \mathbf{518{,}400 \text{ gallons}}$$

The variable used to store the total volume is a C++ float and can be reviewed in the code section of the project artifacts. The maximum size of a float variable is 4 bytes or 32 bits. The approximate range of the float variable with a precision of 6 to 7 digits is:

$$-3.4028235^{38} \text{ to } 3.4028235^{38}$$

When comparing the storage capacity of a float variable on the Arduino Uno with the theoretical total consumption, it can be concluded that there is more than enough space available to store the data. The video below shows the time recording of pouring 1 gallon through the system and that powering the device off and back on does not affect the stored total consumption [2].

Evidence Link:

https://www.youtube.com/watch?v=E016q-jX1hs

5.2.3. Easy to Use

5.2.3.1. The system should be intuitive.

5.2.3.2. The system will be reported as "usable with no outside instruction" by 9 out of 10 people who use the system.

5.2.3.3. Verification Process

Verification Method: Test

Testing Process:

1. Power on the device.
2. User can see current water flow data and stored daily consumption.
3. User can cycle through different units.

Testing Pass Condition:

9 out of 10 people should test the system and understand how to use it based on the system alone.

5.2.3.4. Testing Evidence

The video evidence below and in the File Links section shows a user easily being able to turn the device on and off, change the units of the output, and read the flow of the water using the sensor [3]. Table II below shows a list of electronic signatures of the users that have tested and verified that the system is easy to use. The sheets page that was outputted from a Google Forms document can be found in the File Links of this section [4]. Figure 7 shows a pie chart of the Google Forms document as well.

TABLE II
EASY TO USE FORM

Timestamp	Name	Email	Does the user able to see current water flow rate data, cycle through different units, and power display on and off?
5/6/2023 13:55:08	Ahmad alsaad	Ahmadalsaad.5500@gmail.com	Yes
5/6/2023 14:15:03	Mohammed	alhajajm@oregonstate.edu	Yes
5/6/2023 15:06:47	Khaled	Almutak3@oregonstate.edu	Yes
5/6/2023 15:42:38	Omar Kartam	Kartamo@oregonstate.edu	Yes
5/6/2023 15:42:52	RASHID AL-MARRI	almarrir@oregonstate.edu	Yes

5/6/2023 19:13:09	Mansour Al-Khaldi	Alkhaman@oregonstate.edu	Yes
5/6/2023 21:31:32	Abdulrahman	ahkalkhalifa7@gmail.com	Yes
5/6/2023 21:33:16	Sultan	Sultan.alkhalifa1701@gmail.com	Yes
5/6/2023 21:39:47	Salem	Salemalkhalifa@gmail.com	Yes
5/6/2023 23:00:20	Nawaf Al-khalifa	Nawafalkhalifa99@gmail.com	Yes
5/7/2023 0:56:25	Ahmad	Alkhaahm@oregonstate.edu	Yes
5/7/2023 11:34:20	Ian Ovchinikov	ovchinii@oregonstate.edu	Yes
5/7/2023 18:08:42	Bader Mohammad	mohammba@oregonstate.edu	Yes

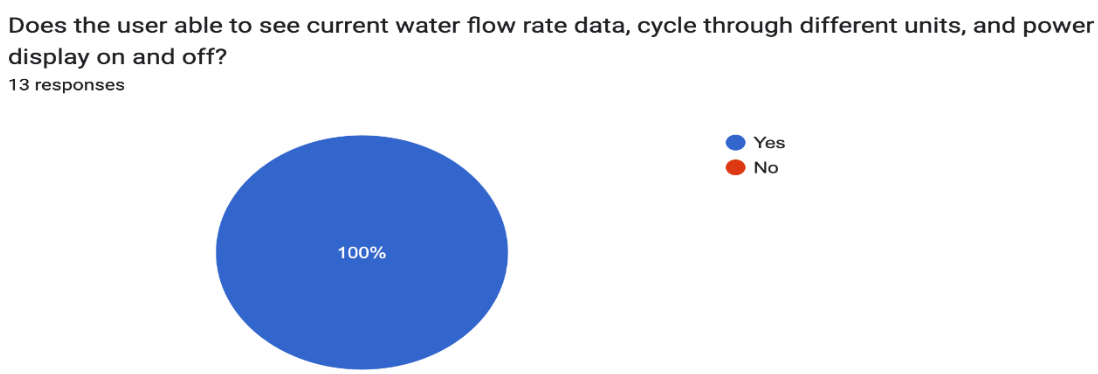

Fig. 7. Showing the Google Forms data output.

Evidence Link:

https://www.youtube.com/watch?v=HATJBxuTjos

https://docs.google.com/spreadsheets/d/1CiNE8pA2y96f1p3MOb2HnQGyx5QwE5dLiFiB07fi9nc/edit?resourcekey#gid=1032277778

5.2.4. Inclusive Readability

5.2.4.1. The product should be able to display data in different units.

5.2.4.2. The system will display water units in both imperial and metric.

5.2.4.3. Verification Process

Verification Method: Demonstration

Testing Process:

1. Power on the system

2. Press the unit change button

3. Check if units being displayed changed

By doing a volume measuring test, the recorded data must be able to be converted back and forth over to a different unit without loss of accuracy (prevent loss of significant digits due to rounding).

Volume/Time units:

L/min <-> gal/min

L/day <-> gal/day

L/week <-> gal/week

L/month <-> gal/month

Volume:

L <-> gal

Testing Pass Condition:

User can switch to and from units without loss of accuracy.

5.2.4.4. Testing Evidence

The video evidence below shows the output units of the water flow rate and total usage being changed from liters to gallons. The video link can also be found in the File links at the end of this section [5]. In Figure 8, the device can be seen with both units of measurement for total usage which confirms the correct calculation when changing units.

$$1 \text{ gal} \approx 3.785 \text{ liters}$$
$$1 \text{ gal} * 1.97 \approx 3.785 \text{ liters} * 1.97$$
$$1.97 \text{ gal} \approx 7.45 \text{ liters}$$

Fig. 8. Showing the device with different unit outputs.

Evidence Link:

https://www.youtube.com/watch?v=icjbeyxF-8U

5.2.5. Non-restrictive Water Flow Sensing

5.2.5.1. The system must not restrict the standard water flow of the home.

5.2.5.2. The system will not cause more than 10% reduction of water flow rate.

5.2.5.3. Verification Process

Verification Method: Test

Testing Process:

Have a 1 Gal jug of water, a 5 Gal bucket, and a funnel.

1. Fill the jug of water.

2. Pour 1 Gal of water inside the jug into the funnel, which is placed directly over the bucket until the 1 Gal jug is empty. Time this step, this will be the control.

3. Repeat the same step with the water flow sensor measuring the amount of water being poured. Time this step.

Testing Pass Condition:

There is less than a 10% volume difference between the tests.

5.2.5.4. Testing Evidence

Two side-by-side tests were done comparing pouring 1 gallon of water into a bucket with the sensor and without the sensor. The screenshot seen in Figure 9 below shows a clip of the video evidence [6]. With the no sensor test, the time it took to pour 1 gallon of water into a bucket was 15.04 seconds while it took 15.55 seconds while pouring through the sensor. This test was important because the design of the system was aimed towards not restricting the standard flow of water through a home or any water system.

Fig. 9. Showing the device plugged in to a standard 120VAC receptacle.

Using the times above, the calculation can be performed to check that there is less than a 10% error in the output.

$$\frac{\text{No sensor time} - \text{With sensor time}}{\text{With sensor time}} = \frac{15.04 \text{ seconds} - 15.55 \text{ seconds}}{15.55 \text{ seconds}} = \mathbf{3.3\% \ error}$$

Evidence Link:

https://www.youtube.com/watch?v=3qVWQLc0js0

5.2.6. Power Supply

5.2.6.1. The system will use 120VAC NEMA 12R outlet for power.

5.2.6.2. The system will utilize outlet power of 120VAC NEMA 12R.

5.2.6.3. Verification Process

Verification Method: Test

Testing Process:

1. System is turned on.
2. With a multimeter, probe the positive and negative output of the AC/DC's bridge rectifier, reading should be 13 V.
3. With a multimeter, probe the output of the DC/DC, reading should be 5 V.

The PCB design will utilize an AC-DC converter and then a DC-DC converter to convert and then step down the 120VAC to 5VDC. This should be done with minimal error due to the potential of overloading onboard components. With most components allowing for a maximum voltage of 5VDC, it is crucial to stay relatively close to that output. The team can probe the system to verify that there is 5V +/- 0.2V with the allowed 4% error.

Testing Pass Condition:

When probing, the system should have 5V at the output of the DC-DC converter and the input of the components with 4% error

5.2.6.4. Testing Evidence

The power supply was probed and tested to prove the efficiency of the supply. The system draws about 76 mA of current nominal. The input voltage is from the wall and can be seen on the oscilloscope as 120 VAC. This is then converted and can be verified with the multimeter in the video as about 4.98 VDC. Figure 10 below shows the device powered on using standard wall outlet power. The evidence link is posted below and is provided in the File Links at the end of this section [7].

$$\frac{4.98 \text{ V (actual)} - 5 \text{ V (expected)}}{5 \text{ V (expected)}} = 0.004 = \mathbf{0.4\% \ error}$$

Fig. 10. Showing the device plugged in to a standard 120VAC receptacle.

Evidence Link:

https://www.youtube.com/watch?v=LyWKFExDrnA

5.2.7. Water-Resistant System

5.2.7.1. The system will be water-resistant.

5.2.7.2. The system will continue to function after being sprayed with at least 2 gallons per minute for 30 seconds.

5.2.7.3. Verification Process

Verification Method: Test

Testing Process:

1. Using a 16oz spray bottle to simulate rain, spray the active system with water for 30 seconds.
2. After the time period, check to see if the device still operates.
3. Check inside enclosure to see if any water got into the system.

Testing Pass Condition:

The system still functions properly, and no water is present in the enclosure.

5.2.7.4. Testing Evidence

This section is evidence that the requirement has been met. Be sure that all needed evidence is here. If a link to a video is supplied there must also be some text description/images in this section. Don't forget to include the date that the requirement was verified.

Evidence Link:

[Link]

5.2.8. Wireless Data Communication

5.2.8.1. The system should communicate data wirelessly to the home network.

5.2.8.2. The system will operate normally when the flow sensor sub-system is at least 30 feet away from the external home router.

5.2.8.3. Verification Process

Verification Method: Test

Testing Process:

1. Place the system 30 feet away from the WiFi router.
2. Press network configuration button to turn the system into Wireless Access Point.
3. Enter network credentials using a WiFi capable device
4. Notification should appear on the LCD output display indicating connection result.
5. The computer program will receive data from the system through wireless communication.

Testing Pass Condition:

The data from the water flow sensor is outputted to the computer program.

5.2.8.4. Testing Evidence

Figure 11 below shows the device outputting 1.10 gallons of total usage and the corresponding webpage on the network displaying the same data. The figure is a screenshot of the evidence video showing the page being refreshed and the new data amount being uploaded to the page [8]. The device is connected to the home network and transmits the data to an open port on the network. This can be accessed via any computer connected to the same network by visiting that page and port on any web browser.

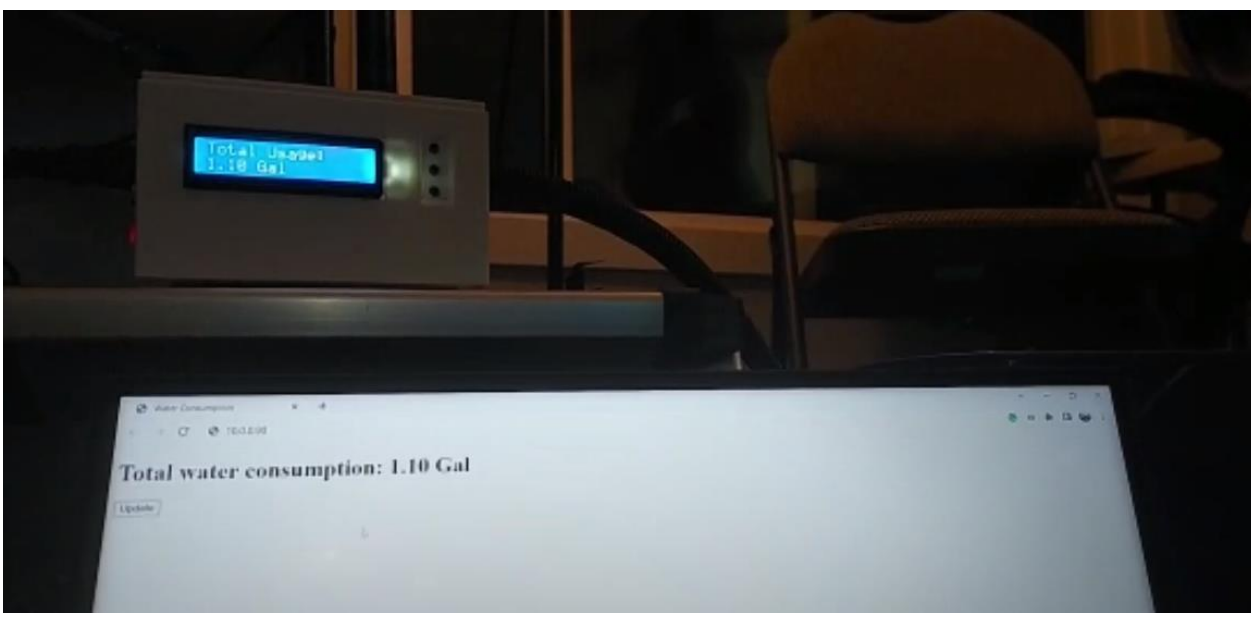

Fig. 11. Showing the device transmitting data to a server page wirelessly over a home network.

Evidence Link:

https://www.youtube.com/watch?v=44Kizpka9Uk

5.3. References and File Links

5.3.1. References

5.3.2. File Links

[1] YouTube "Accuracy Requirement," https://www.youtube.com/watch?v=LYrH9WusKRI, Accessed: 5/4/2022

[2] YouTube "Data storage Requirement," https://www.youtube.com/watch?v=E016q-jX1hs, Accessed: 5/5/2022

[3] YouTube "Easy to use Requirement," https://www.youtube.com/watch?v=HATJBxuTjos, Accessed: 5/4/2022

[4] Google Spreadsheets "Easy to use Form Data," https://docs.google.com/spreadsheets/d/1CiNE8pA2y96f1p3MOb2HnQGyx5QwE5dLiFiB07fi9nc/edit?resourcekey#gid=1032277778, Accessed: 5/7/2022

[5] YouTube "Inclusive readability Requirement," https://www.youtube.com/watch?v=icjbeyxF-8U, Accessed: 5/4/2022

[6] YouTube "Non-restrictive water flow sensing Requirement," https://www.youtube.com/watch?v=3qVWQLc0js0, Accessed: 5/7/2022

[7] YouTube "Power supply Requirment," https://www.youtube.com/watch?v=LyWKFExDrnA, Accessed: 5/4/2022

[8] YouTube "WFS WiFi," https://www.youtube.com/watch?v=44Kizpka9Uk, Accessed: 5/7/2022

5.4. Revision Table

5/7/2023	Carlos: Completed Wi-Fi requirement fulfillment video.
5/7/2023	Mohammed: Added Easy to Use form figure and evidence.
5/7/2023	Christian: Added non-restrictive water flow sensing requirement section and video evidence.
5/5/2023	Mohammed: matching the testing process based on the student portal for each engineering requirement
5/5/2023	Christian: Added data storage requirement section and updated universal constraints.
5/4/2023	Christian: Added accuracy and easy to use video evidence and updated each of the corresponding sections. Added power supply and inclusive readability requirement video and explanation.
3/16/2023	Christian: Added video evidence of power supply requirement fulfillment.
3/12/2023	Mohammed: fixing the verification process on power supply and non-resistive water flow sensing.
3/11/2023	Christian: Added all Project Partner and Engineering Requirements, added Verification Processes and Testing Evidence for the water flow sensor accuracy.

3/10/2023	Christian: Added all Universal Constraints and added verifications for each excluding the power supply efficiency.

6. Project Closing

6.1. Future Recommendations

6.1.1. Technical Recommendations

In our smart water flow sensor project, we utilized the YF-G1 water flow sensor, which served its purpose effectively. However, a potential technical improvement could be to use a more advanced sensor, such as the FS300A G3/4 Water Flow Sensor. This sensor offers enhanced sensitivity, accuracy, and a wider flow rate measurement range compared to the YF-G1 sensor [1]. By implementing the FS300A G3/4 sensor, we could improve the precision of water consumption measurements, enabling users to better understand their water usage patterns and detect leaks with greater accuracy. Ultimately, this upgrade would contribute to more effective water conservation efforts and provide a more reliable user experience.

Another potential improvement for our smart water flow sensor project is to replace the Arduino Uno with a more cost-effective and power-efficient embedded microcontroller, such as the STM32 series from STMicroelectronics. These microcontrollers are based on Arm Cortex-M cores and offer a wide range of performance, power, and peripheral options [2]. By implementing an STM32 microcontroller, we could not only reduce the overall cost of the system but also optimize power consumption, resulting in a more energy-efficient solution for users. Additionally, the STM32 series offers advanced features, such as real-time clock, timers, and communication interfaces, which can further enhance the functionality and flexibility of our smart water flow sensor [3].

A possible upgrade to the ESP8266 ESP-01 Wi-Fi module for our smart water flow sensor project could be the ESP32 module by Espressif Systems. The ESP32 is a powerful, dual-core microcontroller with integrated Wi-Fi and Bluetooth capabilities, offering more features and better performance than its predecessor, the ESP8266 [4]. The ESP32 provides a higher processing power, which enables more efficient data handling and faster response times for user applications. Additionally, the integrated Bluetooth Low Energy (BLE) support allows for seamless connectivity with various devices, further expanding the range of use cases for our system [5]. By implementing the ESP32 module, our smart water flow sensor can benefit from improved performance, enhanced connectivity, and better energy efficiency.

Implementing a dedicated phone application for our smart water flow sensor project, rather than relying on an HTML page, offers several advantages in terms of user experience, functionality, and seamless integration with the recommended ESP32 module. A mobile app can provide a more responsive and tailored user interface, optimized for various screen sizes and mobile operating systems. This ensures a consistent and intuitive experience for users when accessing real-time data, insights, and notifications about their water usage.

Furthermore, a mobile app can take full advantage of the ESP32's integrated BLE capabilities, enabling direct, low-latency communication between the smart water flow sensor and the user's device without the need for an intermediary Wi-Fi network [5]. This can result in improved security, as well as simplified setup and configuration processes for the user. Additionally, a dedicated app can leverage device-specific features such as push notifications, location services, and integration with other smart home ecosystems, providing a more comprehensive and convenient solution for monitoring and managing water consumption.

Finally, an app can offer better offline functionality, storing data locally on the user's device when an internet connection is unavailable and synchronizing the information when connectivity is restored, ensuring uninterrupted access to water usage data.

6.1.2. Global Impact Recommendations

Future iterations of the smart water flow sensor could benefit from integration with local and national water management systems, which could help optimize water distribution and consumption on a larger scale. By collecting and analyzing data from numerous households and businesses, utility companies and governmental agencies can make data-driven decisions to improve water resource allocation and infrastructure planning [6]. Furthermore, this aggregated data can contribute to identifying patterns of water consumption, enabling the development of targeted water conservation policies and public awareness campaigns. Such integration could have a global impact by contributing to more efficient and sustainable water management practices, ultimately helping to address water scarcity challenges faced by many regions around the world.

Another global impact recommendation for future iterations of the smart water flow sensor is to adapt and deploy the technology in developing countries, where water infrastructure is often less reliable and more susceptible to leaks and water loss. According to a World Bank report, an estimated 45 million cubic meters of water are lost daily in developing countries due to leaks in water supply systems [7]. By implementing cost-effective and easy-to-install smart water flow sensors in these regions, communities can identify and address leaks more efficiently, reducing water waste and improving access to clean water for residents. The adoption of this technology in developing countries can contribute to more equitable water distribution, enhance public health, and promote sustainable development.

6.1.3. Teamwork Recommendations

In our project smart water flow sensor, we worked together last two term to get the project done successfully with having good teammates. Teamwork is important for projects because electrical engineers need to work on advanced projects in their jobs. For example, everyone should work on their blocks to create a smart water flow sensor, and we have several blocks that we worked on it such as water flow sensor, PCB, Microcontroller, Power supply, Wi-Fi-module, Enclosure, HMI and buttons. There are two future recommendations that are helpful for working on the team.

It is important that the members on the group be productive in group assignments because it is teamwork, and everyone will get the same grade. In this situation, members will need to cooperate with the group to keep track of the assignments because it will affect everyone on the group. Teamwork also needs to have phases of management to make sure that the group are doing their jobs without having any issues. Phases of management includes process of the project based on the figure [6]. In this case, the teamwork can include phases of management to have every member a responsibility of the task on the project.

Teamwork must unify the version of the project because we don’t want to have different versions of the project. Some group having an issue of getting the project unified because they don’t have a google drive file shared or other websites. So, sharing a file for the project is important to avoid having issues with documents. In this case, everyone can edit from the same page, and they can have all their projects done on same page as what our group did for the project. Our group used a unified project showcase to illustrate the smart water flow sensor [8].

6.2. Project Artifact Summary with Links

The schematic in Figure 1 shows the design of the main system circuitry. This includes the connections between the Arduino Uno, Wi-Fi module, water flow sensor, LCD display, user inputs and the input from the power supply PCB. The KiCad schematic file can be found in the File Links of this section [1].

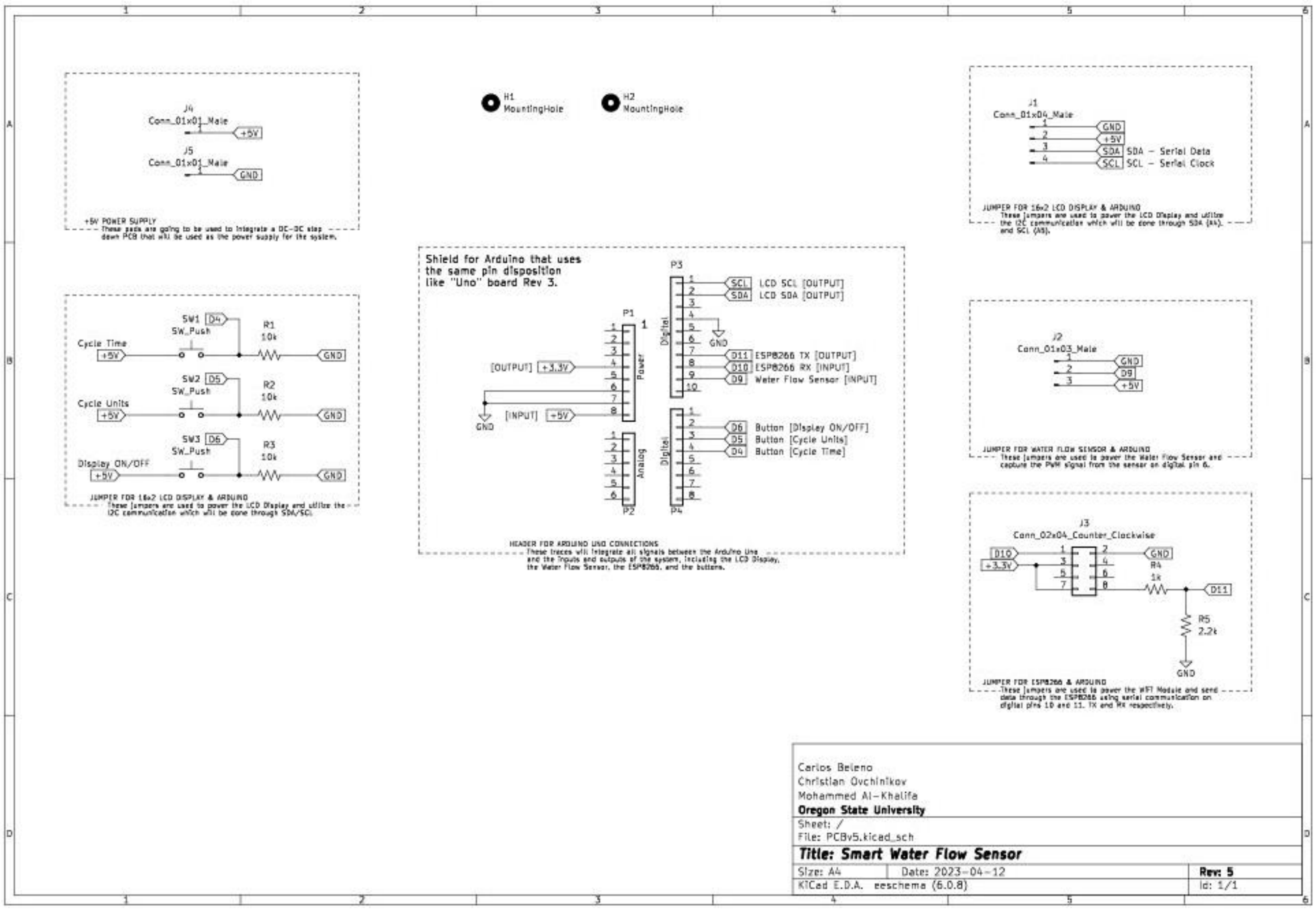

Fig. 1. KiCad Schematic of the main system PCB.

The PCB layout in Figure 2 shows the design of the main board used in the system. This includes the trace connections between the Arduino Uno, Wi-Fi module, water flow sensor, LCD display, user inputs and the input from the power supply PCB. The KiCad file for this PCB can be found in the File Links of this section [2].

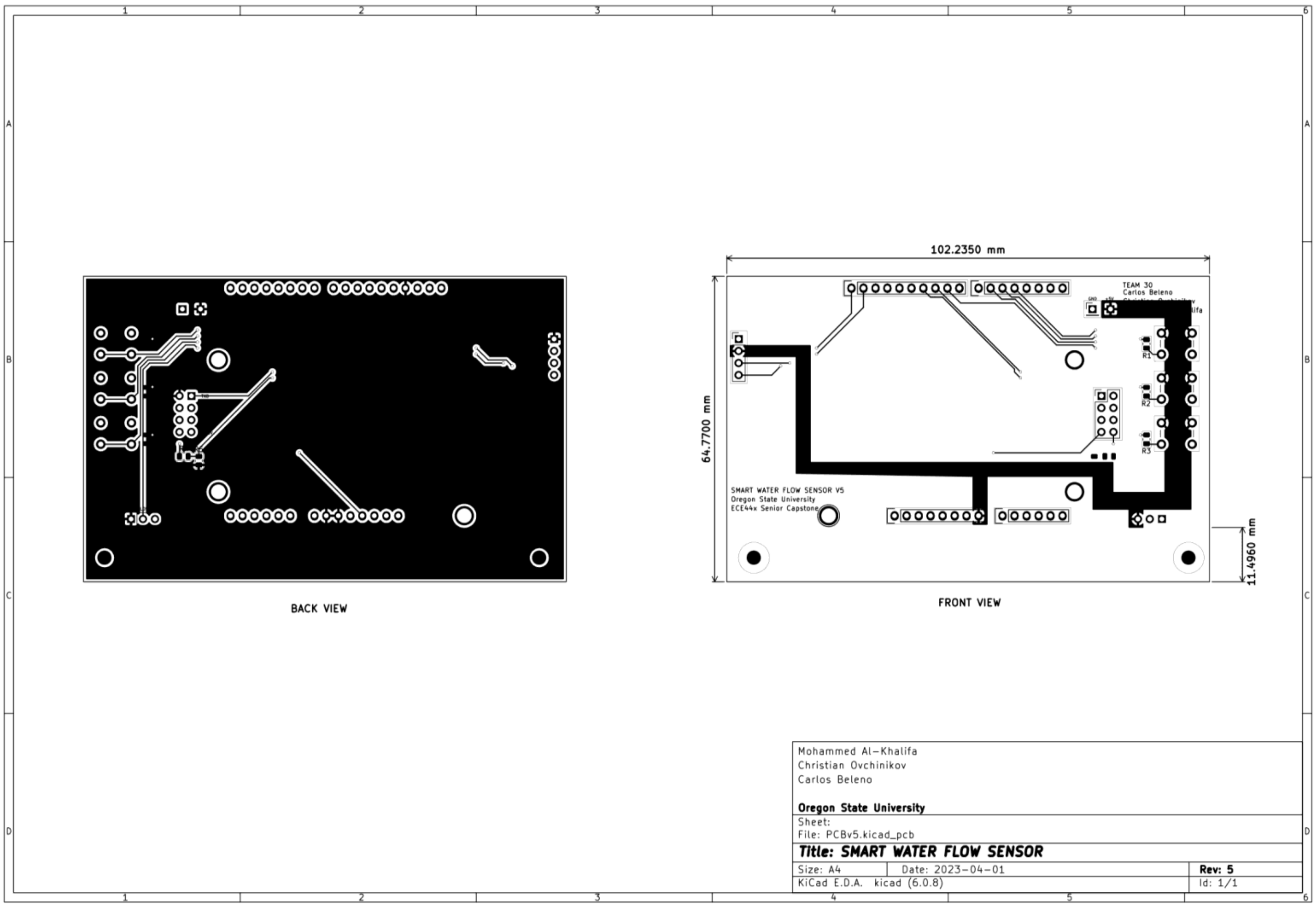

Fig. 2. KiCad PCB layout of the main system.

Figures 3 through 6 below show various views of the enclosure design in Blender. The main points of interest are the front and top view that details out how the user interface would be laid out and how the internal electronic components would be mounted in the system. Slots were designed to hold each PCB vertical to allow for a streamline fit of all the input and output components. All measurements on the following figures are in centimeters.

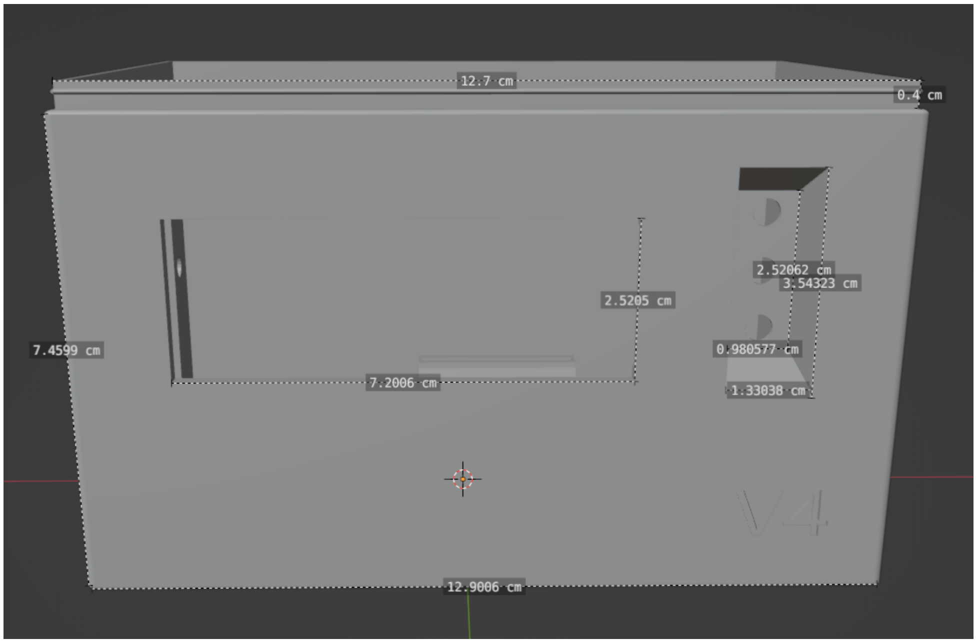

Fig. 3. Front view of the enclosure.

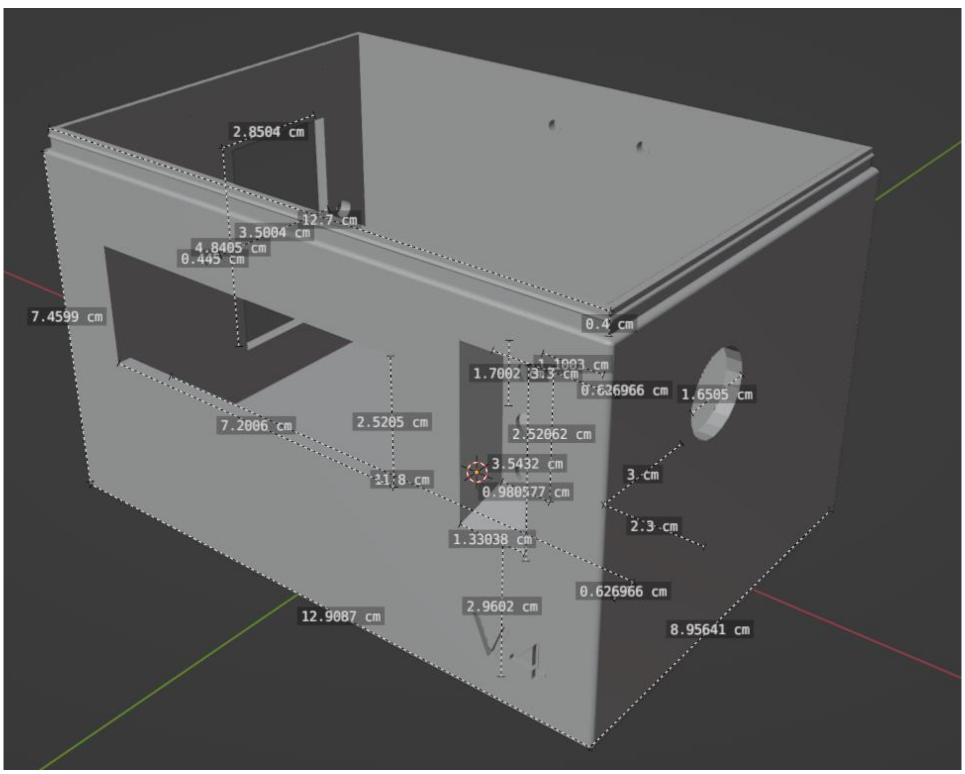

Fig. 4. Isometric view of the enclosure.

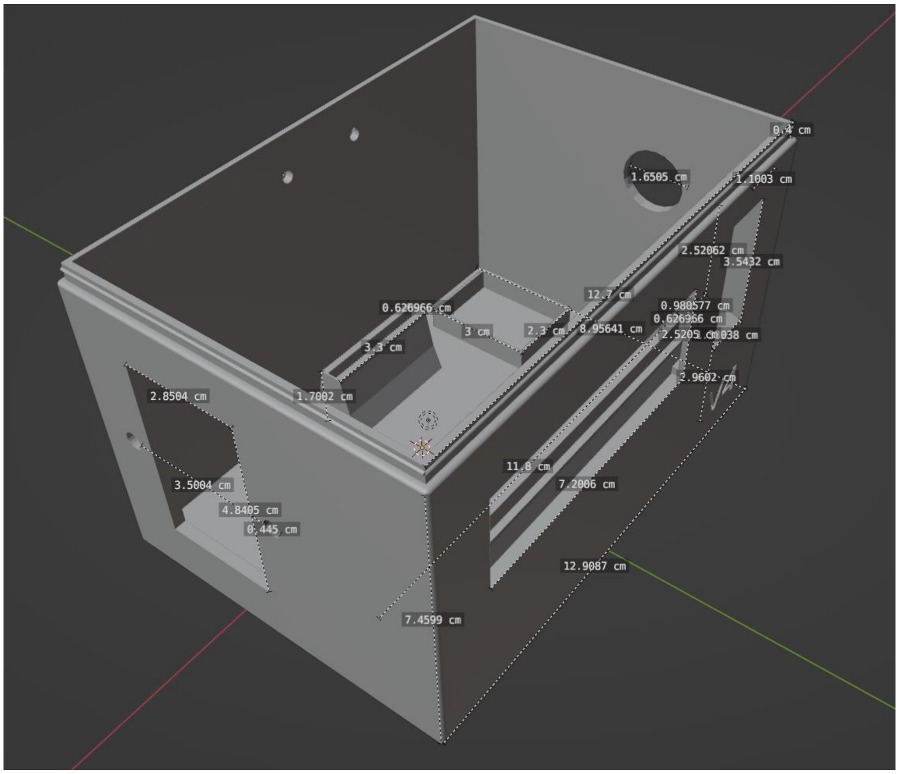

Fig. 5. Different angled isometric view of the enclosure.

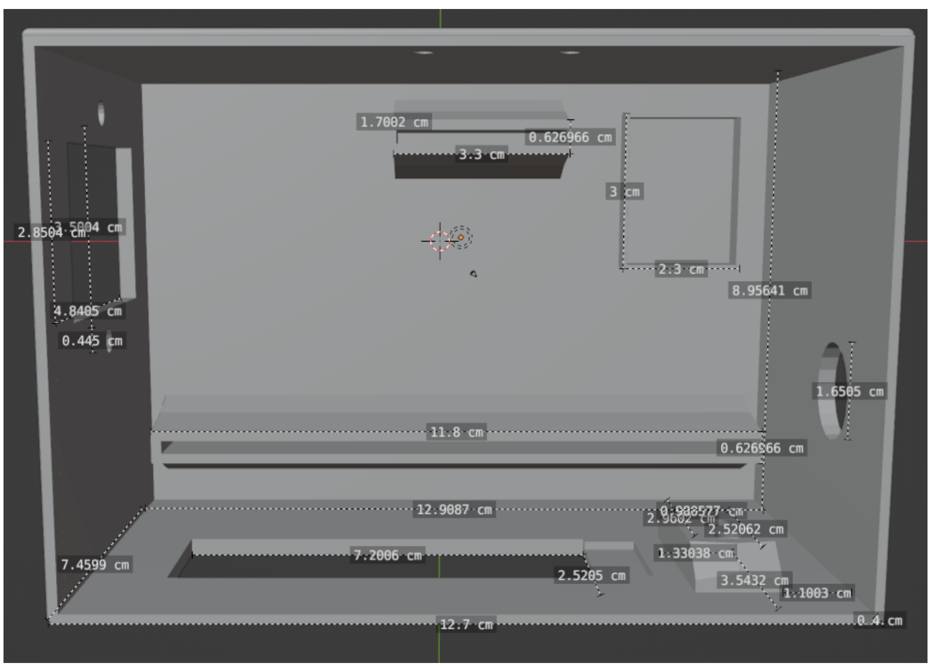

Fig. 6. Top view of the enclosure.

The Arduino Uno's firmware was developed in the Arduino IDE to perform data accumulation from the water flow sensor, execute computations, and subsequently present the processed information on the LCD display. Additionally, the firmware managed user inputs acquired from the button interface and facilitated data transmission via the ESP8266 module to the designated HTML page, ensuring seamless communication between the various components of the system.

Reviewing Figure 3 in Section 4.2.2, the following pseudocode can describe the firmware used to operate the system. The main code file can be found in the File Links of this section [3].

Initialization:

1. Initialize LCD display, buttons, and flow sensor
2. Attach interrupt to flow psensor to count pulses

Main Loop:

1. Calculate and update flow rate every second
2. Update display if turned on
3. Check button states and perform corresponding actions
4. Reset total volume after 30 days

Functions:

1. pulse_counter(): Increment pulse count
2. DisplayUpdate(): Update the LCD display with flow rate or total usage (in gallons or liters)
3. CalculateFlow(): Calculate flow rate and total consumption
4. ButtonCheck(): Check button states and perform actions
    a. DisplaySwitch(): Toggle display on/off
    b. DisplayOutput(): Toggle between flow rate and total consumption display
    c. DisplayUnits(): Toggle between gallons and liters display
5. Reset30Days(): Reset total volume after 30 days

6.3. Presentation Material

The expo poster in Figure 7 was designed to be descriptive visually and technically. The flow diagrams were simplified to show the design of the system and how each block works with each other. There is also a project overview with an introduction to the project, and the process of having a problem and providing a solution.

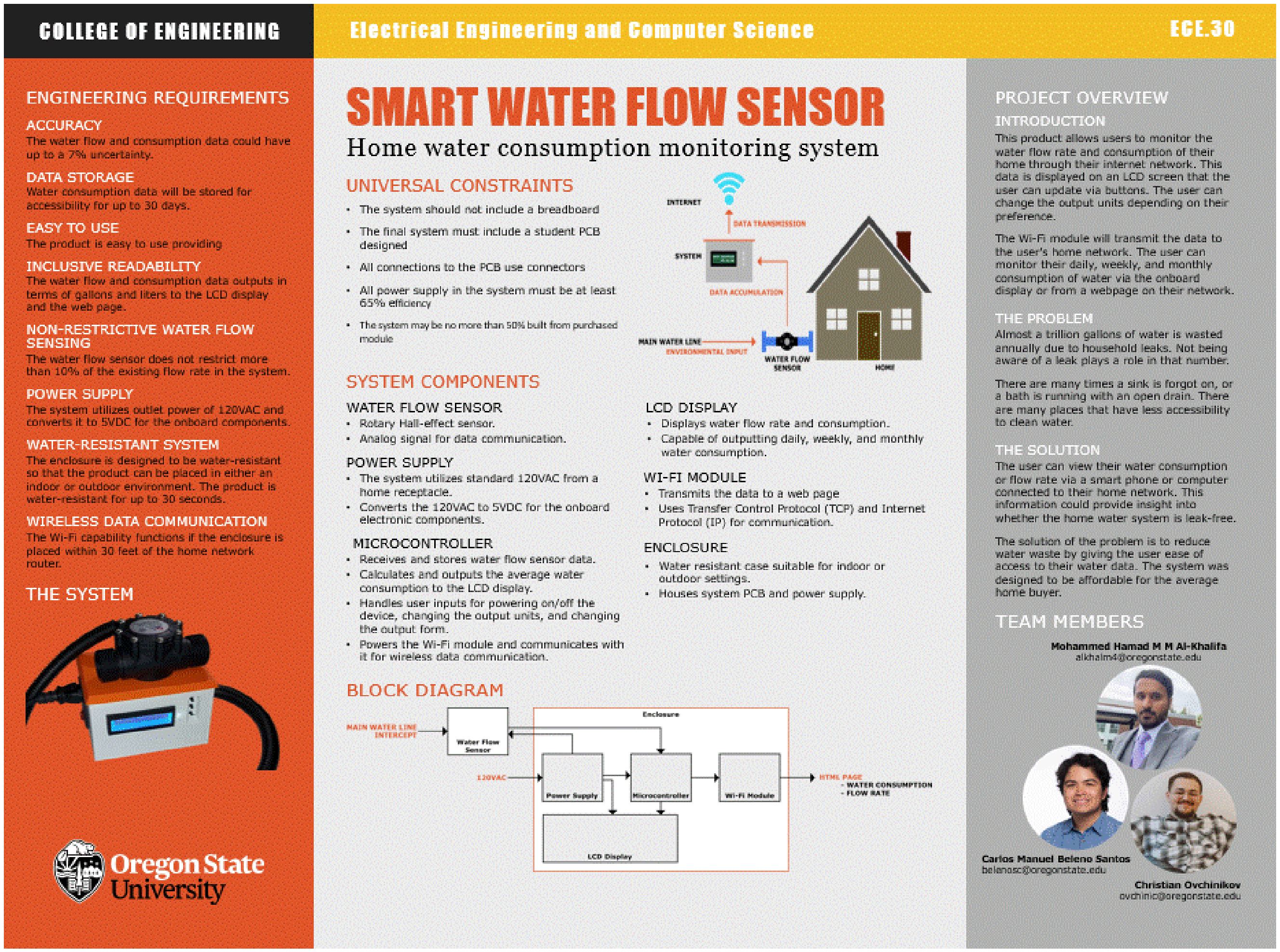

Fig 7. Expo Poster created using the template provided.

The project showcase can be located on the EECS Project Showcase site by searching for keywords: 3D Printing, Arduino, and IoT [8]. This page is the central hub for information regarding the project including majority of the important information from this document in plain view.

The project required an extensive number of resources to build, test and verify that the system would work and be sustainable for consumer use. A bill of materials was produced to show the breakdown of designing and manufacturing a single system [9]. This can be used to propose price points and encourage the future iterations of the system to further lower the cost of the product. With the future recommendations, the system could become significantly cheaper to manufacture and be sold a close price to competitors.

6.4. References and File Links

6.4.1. References

[1] Seeed Studio, "Seeed Studio FS300A G3/4 Datasheet," https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/314150003_Web.pdf, Accessed: 4/25/2023

[2] STMicroelectronics, "STM32 32-bit Arm Cortex MCUs," https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html, Accessed: 4/25/2023

[3] Deep Blue, "Getting Started With STM32 ARM Cortex MCUs," https://deepbluembedded.com/getting-started-with-stm32-arm-cortex-mcus/, Accessed: 4/26/2023

[4] Espressif, "ESP32," https://www.espressif.com/en/products/socs/esp32, Accessed: 4/26/2023

[5] Random Nerd Tutorials, "Getting Started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE," https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/, Accessed: 4/26/2023

[6] Taylor & Francis Online, "Integrated water resources management: balancing views and improving practice," https://www.tandfonline.com/doi/full/10.1080/02508060802272820#:~:text=Integrated%20water%20resources%20management%20is%20a%20conceptual%20framework,definitional%20confusion%20over%20the%20precise%20meaning%20of%20IWRM, Accessed: 4/27/2023

[7] The World Bank, "The Challenge of Reducing Non-Revenue Water (NRW) in Developing Countries," https://documents1.worldbank.org/curated/en/385761468330326484/pdf/394050Reducing1e0water0WSS81PUBLIC1.pdf, Accessed: 4/27/2023

[8] EECS Project Showcase, "Smart Water Flow Sensor: Team 30 ECE 44x," https://eecs.engineering.oregonstate.edu/project-showcase/projects/?id=Gb6b3OD9LIsDf3r3, Accessed: 4/26/2023

6.4.2. File Links

[9] Microsoft SharePoint, "Bill of Materials," https://oregonstateuniversity.sharepoint.com/:b:/s/ECE441/ESCg1PdLuS5FrlhTfCLdyY4B_43ehHRtfXwEME2RwbvgCA?e=07uETR, Accessed: 5/7/2023

6.5. Revision Table

5/7/2023	Christian: Added mechanical drawings and bill of materials to this section.
5/5/2023	Christian: Updated project artifacts, added pseudocode and PCB footprint
4/28/2023	Mohammed: finalized section 6
4/28/2023	Mohammed: added two teamwork recommendations
4/27/2023	Mohammed: Added future recommendations.
4/27/2023	Christian: Added global impact recommendations.
4/26/2023	Christian: Added technical recommendation for Wi-Fi module and phone application. Added main system schematic and Arduino code. Created project showcase and added to document.
4/25/2023	Christian: Added sections to document and added poster to section 6.3. Added technical recommendation for water flow sensor and microcontroller.