University of British Columbia

Social Ecological Economic Development Studies (SEEDS) Sustainability Program

Student Research Report

Aquaponics Monitoring and Control System

Piponics: An open-source, ‘Internet of Things’ device and mobile application

Prepared by: Carson Berry, Lynes Chan, Mason Duan, Jayden Leong, and Hannah Xu

Prepared for: UBC Mechanical Engineering

Course Code: ELEC 491

University of British Columbia

Date: 27 April 2021

Disclaimer: “UBC SEEDS Sustainability Program provides students with the opportunity to share the findings of their studies, as well as their opinions, conclusions and recommendations with the UBC community. The reader should bear in mind that this is a student research project and is not an official document of UBC. Furthermore, readers should bear in mind that these reports may not reflect the current status of activities at UBC. We urge you to contact the research persons mentioned in a report or the SEEDS Sustainability Program representative about the current status of the subject matter of a report”.

SEEDS Sustainability Program

UBC sustainability

EXECUTIVE SUMMARY

Piponics is a system that monitors and control aspects, such as pH, and water level of a small or medium scale hydroponics/aquaponic system. The Piponics system contains a Raspberry Pi component which would monitor the hydroponics/aquaponic system through various sensors and actuators. The Piponics system also comprises a companion mobile application where the user can access hydroponic/aquaponic systems tied to their account and can monitor and control the specific sensors and actuators on the corresponding hydroponic/aquaponic system. The goal of the Piponics system is to create an affordable hydroponics/aquaponics monitoring system for small scale growers and hobbyists which is in line with SEEDS goal of advancing the wellbeing of the environment and advancing food systems. The Piponics system is intended to be open source so that different sensors and actuators can easily be added to fit the need of any hydroponic/aquaponic system.

The objective of creating this system was to address SEED’s goals of advancing the wellbeing of the environment, food systems, responsible consumption, and community inclusion. Piponics is an affordable and open-source system that was created to address these issues. With the Piponics system, small scale growers and hobbyists are empowered and enabled to grow affordable, local, and organic produce in the communities. The Piponics system would also provide educational and research benefits as the system is configurable and stores long term data of sensor measurements for future analysis. The planned use of the Piponics system is to be a part of any aquaponic installations on UBC campus which would serve to provide affordable, local, and organic produce to the community members on campus.

This project stands out from other aquaponic monitoring and control systems due to its low cost, high quality documentation, expandability for future work, and open-source code and design.

The Piponics system has two main components; the hardware system, which includes the Raspberry Pi, sensors, actuators, and the battery, and the software system, which includes the mobile application. The hardware system includes the Raspberry Pi, sensors, actuators, and the battery. The Raspberry Pi is the computing device that sends the values measured by the sensors to Google Cloud and is what receives commands from the Google Cloud to operate the actuators. The hardware system is encased in a plastic casing that protects the circuitry from outside elements such as water. The battery is included in the system as backup power in case of a power outage. The software system includes the mobile application which is connected to Google Cloud. The software system can query the sensor data from Google cloud as well as send commands to the systems actuators. Each user is authenticated on the mobile application and allows registered users to add, view, and control multiple hydroponic/aquaponic systems to their account provided they know the system's unique identifiers.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 PROJECT CONTEXT

This project is a joint delivery to Bernhard Nimmervoll, our community partner and UBC SEEDS.

This section includes a description of aquaponics and hydroponics. Briefly, aquaponics and hydroponics are food production methods known for being more sustainable than traditional food productions practices.

Finally, a description of the problem this project aims to solve will be given. Overall, this project aims to make aquaponics and hydroponics easier to maintain and monitor. This will, in turn, serve to promote aquaponics and hydroponics as sustainable methods of food production.

1.11 Client Introduction

The primary client of this project is Bernhard Nimmervoll, a Technician for the UBC Mechanical Engineering Department. Bernhard is a passionate advocate of sustainability initiatives, including being a hobbyist aquaponics grower in his spare time. Bernhard introduced this project to promote aquaponics and hydroponics as sustainable food production methods, and to combat food insecurity.

Bernhard has partnered with the Social Ecological Economic Development Studies (SEEDS) Sustainability program. The SEEDS program developed out of UBC's commitment to being a global leader in integrated sustainability [1]. As such, SEEDS is an organization whose aims are to advance the well-being of the environment, local biodiversity, food systems, responsible consumption and community inclusion. They are pursuing these aims by supporting student-led research and development about climate friendly food systems, urban food production for community resilience, biodiverse food systems, zero-waste, and food justice and sovereignty. This project is one of many such UBC SEEDS projects.

Figure 1 : UBC SEEDS big five research priorities [2]

UBC SEEDS has five main research goals, which are shown in Figure 1. The research goal, ‘Enable the great food transformation’, is the primary focus of our project. This goal is typically pursued by researching food security and sustainability, implementing policies and plans, and participating in education, engagement, and demonstration. In our case, our project will contribute to this goal by advancing technology related to sustainable food growth using aquaponics and hydroponics.

1.32 Hydroponics and Aquaponics

Hydroponics is a method of agriculture where plants are grown in a nutrient-rich aqueous solution. Nutrients in hydroponics are inputted by the grower to nourish plants. Hydroponics contrasts traditional forms of agriculture, where plants are grown in soil.

Aquaponics is very similar to hydroponics. In aquaponics, plants are also grown in an aqueous solution, however plants receive nutrients in a different manner. Instead of growers inputting nutrients, plants receive nutrients from fish waste. To make this possible, fish are raised in water tanks that the plants are grown in. Aquaponics receives its name because of this, by being a mixture of aquaculture and hydroponics. A simplified diagram of aquaponics is show in Figure 2.

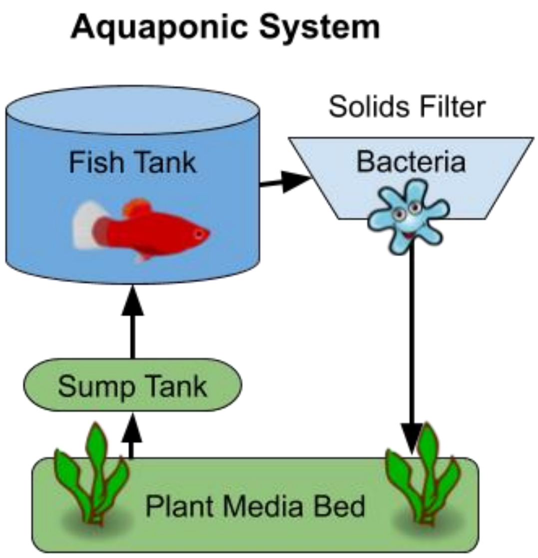


Figure 2: A diagram of a basic aquaponics system.

Both hydroponics and aquaponics are being developed as more sustainable approaches to agriculture than traditional methods for the following reasons [3]:

1. Higher crop production per unit area;
2. Faster plant growth;
3. Lower water usage;
4. Lower susceptibility to pests if grown indoors.

For aquaponics, there are also the following additional benefits [4][5]:

1. Fish and plants form a circular, self-sustaining ecosystem;

2. Fish food is the only input needed;

3. Fish as well as plants can be harvested for food.

For the benefits listed above, our clients, Bernhard Nimmervoll and UBC SEEDS, are interested in improving and advocating for these more sustainable agricultural methods.

Both aquaponic and hydroponic systems need to be carefully monitored. Plants require that the aqueous solutions they are planted in have all necessary nutrients. They also require other environmental variables to be monitored, including solution pH, light intensity, humidity, and more. In addition, aquaponic systems are even more difficult systems to maintain due to their complex multi-species dynamics. Systems need to be balanced such that fish, bacteria, and plants can all thrive.

With this system complexity, hydroponics and aquaponics require frequent supervision and maintenance by a knowledgeable operator. This adds significant cost and labour for aquaponics and hydroponics growers. This can be discouraging for small scale and hobbyist aquaponic or growers who do not currently have access to an affordable, automated monitoring and control system.

1.35 Domain

This project is within the domain of aquaponics and hydroponics monitoring and control systems typically built by hobbyists and small companies. These are small-scale monitoring and control systems usually designed for and installed on a single aquaponics or hydroponics system. They are also usually open-source and non-proprietary. Most of these systems monitor their systems using internet connectivity, making them categorizable as small-scale internet of things (IoT) projects.

Many of the previous implementations of systems of this type are simplistic and designed for single systems without much forethought for future development [7][8]. Some of the systems have been designed with future improvements, ease of setup, cost, and efficiency in mind [9][10]. This system is quite expensive for a monitoring system that is built using off the shelf components, costing around $850. The system also has limited external monitoring capabilities and no licensing for open source although they claim to be an open-source project. As well, the monitoring system doesn't record any data and does not seem to have any response system in place in case of power failure.

1.2 PROJECT PURPOSE, GOALS AND OBJECTIVES

The goal of this project is to develop a user-friendly, no-technical-expertise-needed solution for monitoring and control of hydroponic or aquaponic systems. This system should utilize open-source and low-cost technologies wherever possible to minimize barriers to replication and access. In terms of scope, our project serves to deliver a working prototype of such a system which is installed on an existing hydroponic or aquaponic system.

When completed, this project will help solve the problem of hobbyist and aquaponic growers monitoring and maintaining their systems. It will help reduce cost, time, and effort associated with aquaponics and hydroponic systems; and help reduce system failures resulting in production losses.

In addition to these benefits, this project will also contribute towards larger goals including increasing food systems sustainability, reducing food insecurity, and increasing sustainability awareness. First, by making hydroponics and aquaponics systems easier to develop and maintain, this may encourage more growers to use these methods. This, in turn, can promote food systems sustainability as aquaponics and hydroponics have advantages over traditional growing methods.

In terms of food insecurity, this project may be included in future on-campus aquaponics or hydroponics growing systems. For example, a physical 'Food Hub' is being envisioned UBC Okanagan (UBCO) campus [6]. This 'Food Hub' may provide affordable, local, organic vegetables and fish to UBCO community members as an effort to reduce student food insecurity. This Food Hub is being planned to include an aquaponic system of both functional, research, and educational value. The product of this project could be a helpful addition to monitor such a system and could therefore contribute to broader issues including reducing food insecurity.

2. METHODOLOGY AND METHODS

The process used to design the project is outlined below.

2.1 DESIGN METHODOLOGY

To meet the project purpose, goals, and objectives, our team developed requirements which capture the needs of our project. These requirements were done with consultation to an aquaponics industry expert. As well, requirements were developed with consultation from our client Bernhard Nimmervoll, who is himself a hydroponic grower. Our requirements our are divided into three categories: functional requirements, non-functional requirements, and constraints.

2.11. Functional Requirements

To start, here are the functional requirements we identified for our project:

Functional Requirement	Brief Description	Details
F1	**Display measurements of critical system variables in hydroponic or aquaponic systems.**	For the scope of our project, critical system variables include and are limited to pH, water level, water leakage, and

		water temperature. All displayed measurements must be updated at a minimum of every five minutes.
F2	**Store and display past measurements of critical system variables so hydroponic or aquaponic growers can analyze system trends.**	The system must be capable of storing and displaying at least the past 3 months' worth of measurements.
F3	**Send alerts to growers within 1 minute of a detecting that a critical system variable is outside of a normal range.**	Growers will specify normal ranges for each of the critical system variables.
F4	**Adjust pH level and water level automatically such that they always stay within a pre-defined range.**	This pre-defined range for pH and water level can be configured by growers depending on the needs of their hydroponic or aquaponic system.
F5	**Operate for at least 8 hours in the event of a power supply failure or outage.**	The system must also send an alert to growers indicating that there was a power failure. Alerts should occur a 1 minute of the power failure.
F6	**Give growers the ability re-calibrate sensors and actuators.**	These re-calibrations can be made at any time

2.12. Non-Functional Requirements

Next, see the table below for our project’s non-functional requirements:

Non-Functional Requirement	Brief Description	Details
NF1	**Cost less than $300 CAD per system.**	Additional monthly servicing fees of must also be less than \$5 per month.
NF2	**Be designed to work with any small-scale aquaponics or hydroponics systems**	For this project, small-scale aquaponics or hydroponic systems have under 200 plants and under 3000 gallons of water.
NF3	**Ensure that only permitted growers can monitor and control hydroponic or aquaponic systems**	Future administrators of our project must enable growers to monitor and control a given hydroponic or aquaponic system. Otherwise, access to monitor a given hydroponic or aquaponic system will not be permitted.

NF4	**Be water resistant.**	An IP rating of IP64 (Protected from immersion between 15 centimeters and 1 meter in depth) must be satisfied.
NF5	**Be able to monitor different or additional aquaponics or hydroponics system variables, depending on the grower's preference.**	
NF6	**Measure all critical system variables within an accuracy of 5%**	
NF7	**Maintain all controlled variables within an accuracy of 10%.**	

2.13. Constraints

The following is a table of constraints on our design:

Constraint	Brief Description	Details
C1	**All software developed must be open source, with appropriate licensing to allow for legal modification, duplication, and sharing.**	The license we have chosen is the GNU General Public License v3.0 as it allows the community to make changes and to use the code for personal purposes but prevents distributing closed source versions.

3. DESIGN RESULTS

3.1 SYSTEM OVERVIEW

The figure below provides an overview of our design. As you can see, our design contains the following four subsystems: sensors, control, IoT, and software application.

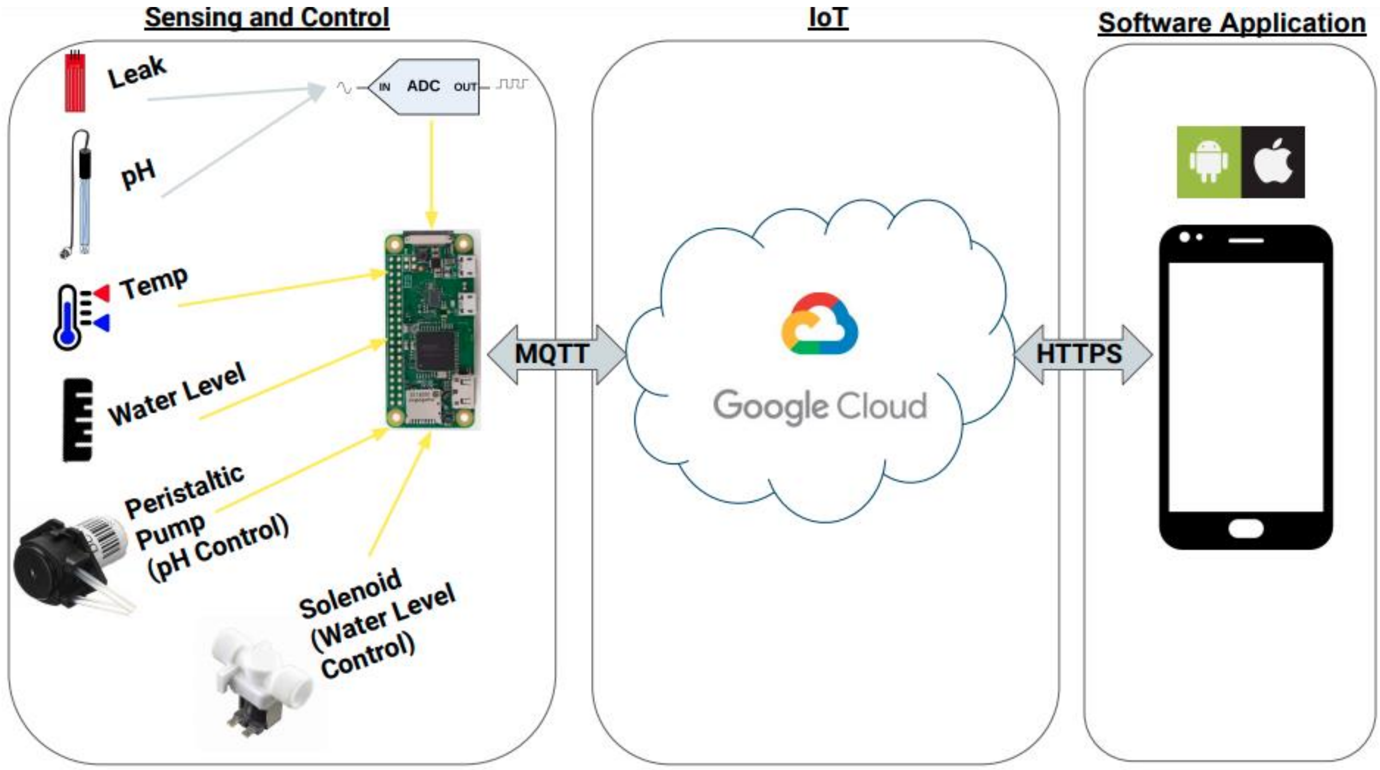


Figure 3: System Overview diagram

Starting from the left-hand side, we can see both the sensing and control subsystems. This consists of a microcomputer attached to multiple sensors and control devices. There are four key sensors in our project shown: the water leakage, pH, temperature, and level sensors. These sensors allow our design to take measurements of the water of aquaponics or hydroponics systems. As such, the pH, temperature, and water level sensors are placed inside water tanks in an aquaponic or hydroponic system. As far as the control systems are concerned, two devices are attached to the microcomputer: a peristaltic pump and a solenoid. These two devices allow our design to automatically adjust pH and water level respectively.

In the middle, we see the IoT subsystem. This system's responsibility is to transfer data between a mobile software application and the sensing and control system. It also stores past sensor measurements for later analysis.

Finally, on the right-hand side, we see the software application. This is where the user primarily interacts with their hydroponic or aquaponic system. This application displays past and current sensor measurements. It also allows the user to receive alerts if urgent issues are detected by the sensing and control system, including issues like power failure and dangerous water pH levels. In the following sections, the design of each of these subsystems will be explored in further detail. The following are some screenshots of the mobile application, to give the reader a further understanding of how our project works:

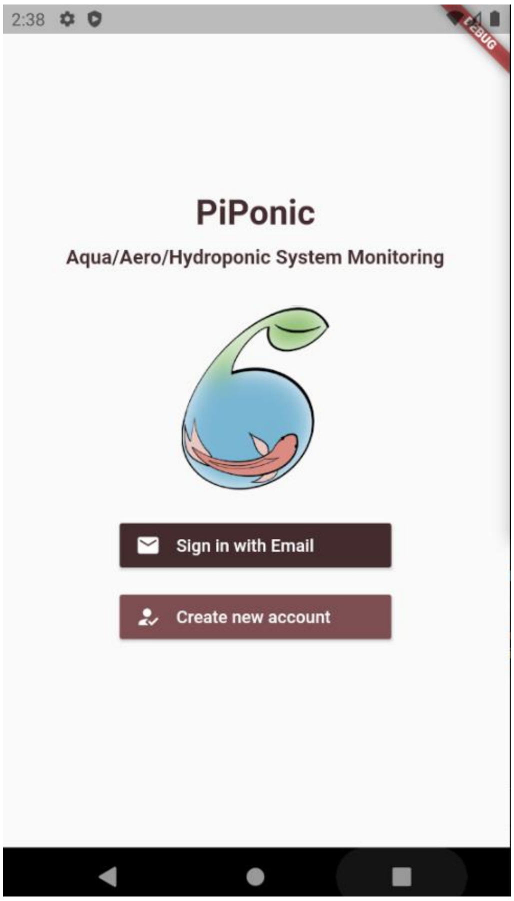


Figure 4: Sign-in or account creation page

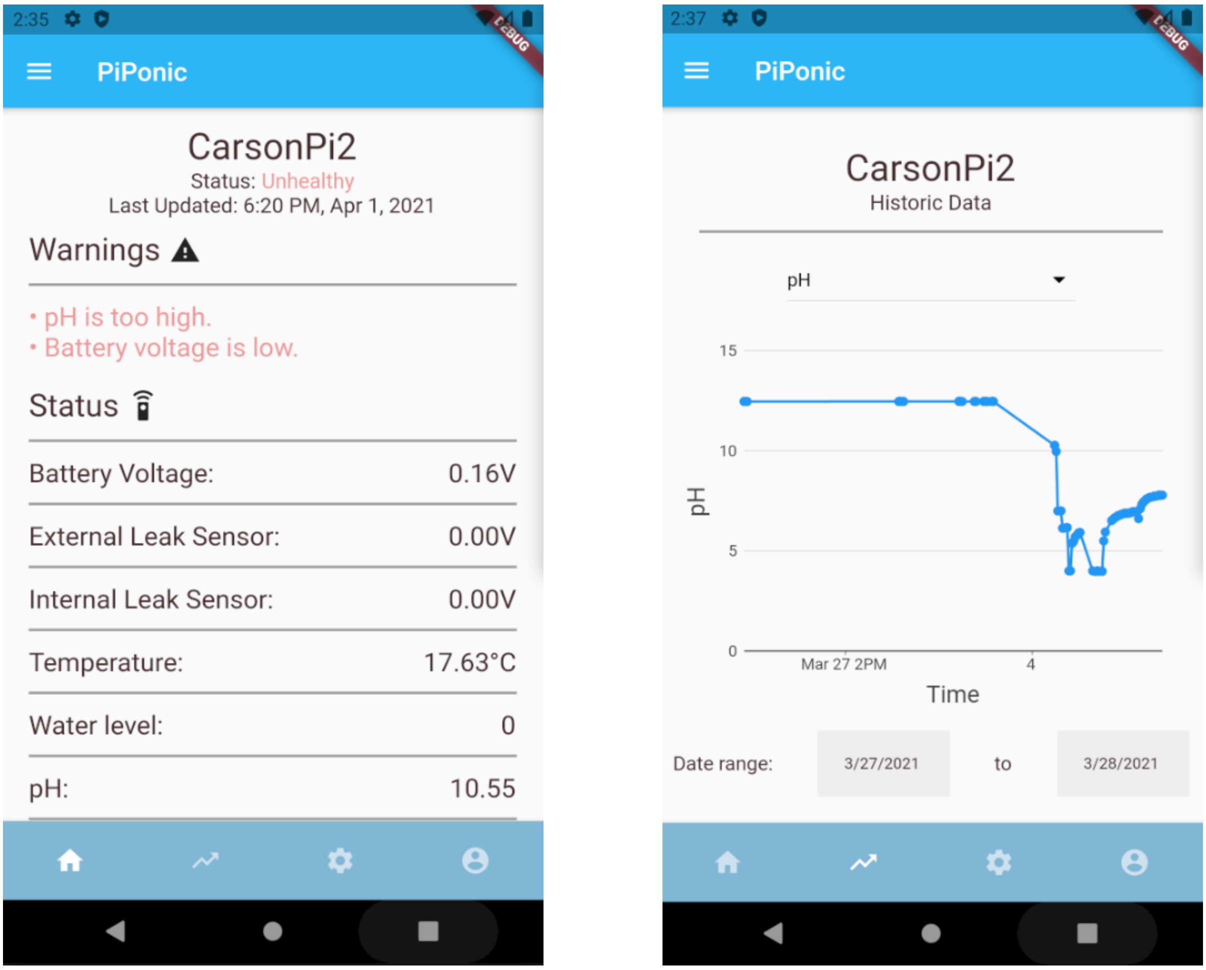


Figure 5: App pages displaying the status and historical trends of an aquaponics system named 'CarsonPi2'.

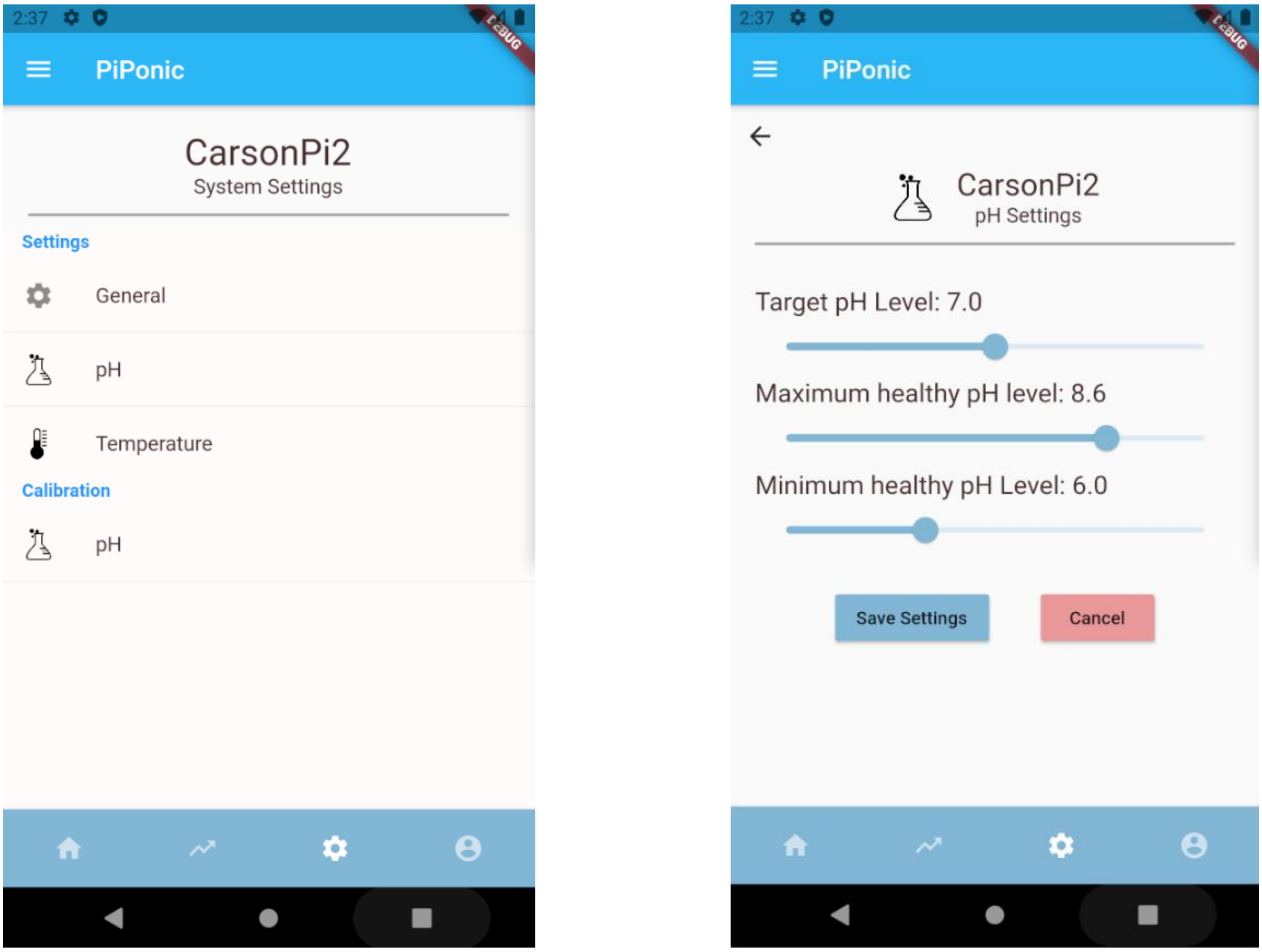
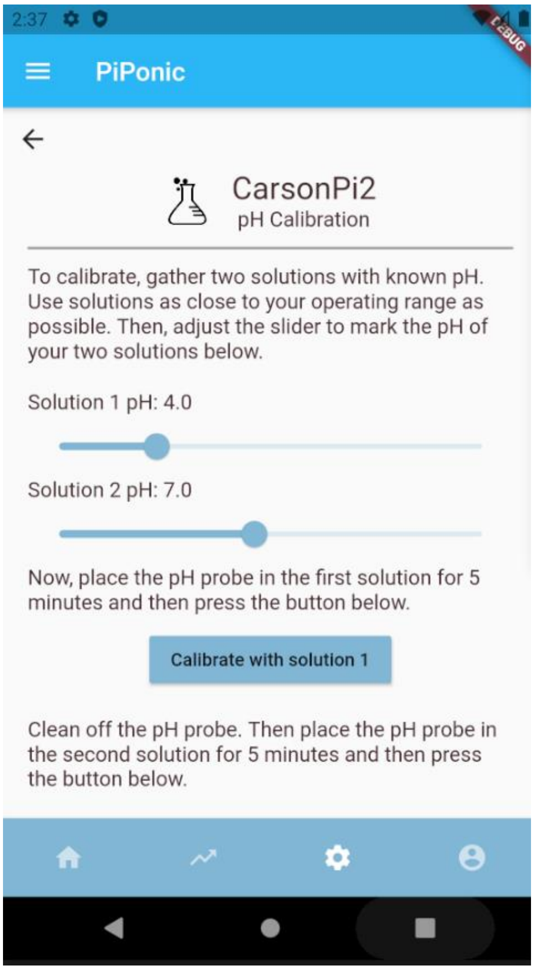


Figure 6: The menu of settings that users can specify for an aquaponics or hydroponics system (left). The menu that users can use to adjust pH settings (right).



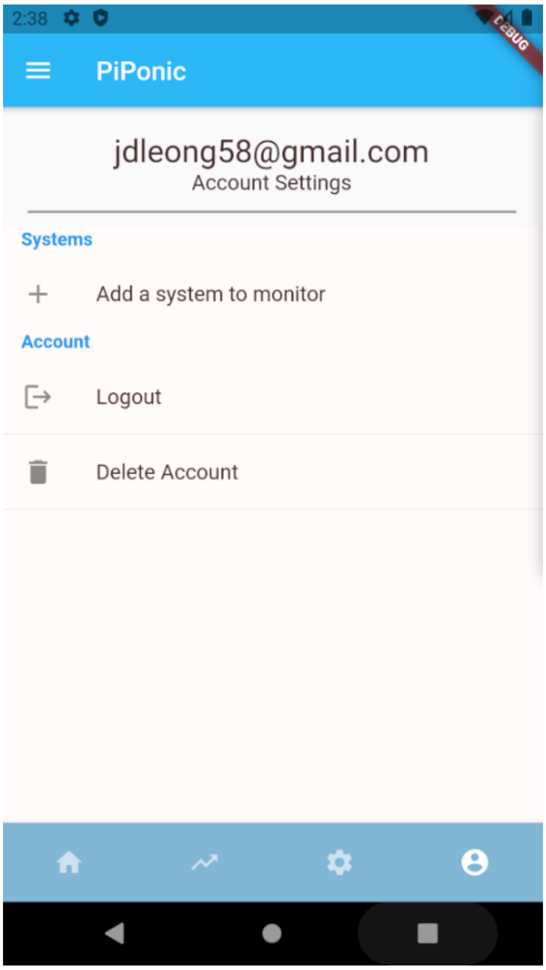


Figure 7: App screen where users can calibrate the pH sensor (left). screen where users can manage their account and add systems to monitor (right).

For a more detailed technical discussion of our design, please see Appendix: Design document.

3.2 SYSTEM VERIFICATION

Each requirement has several verification tests associated with it to ensure that each aspect of the design works as intended to meet the requirement. For a complete list of verification tests performed as part of this project's design, please refer to Appendix C: Verification.

4. DISCUSSION

Our design here meets of all the core requirements identified during communication with the community clients. We recognize that every grower, and every system will be unique and that we are not able to comprehensively account for every feature that future user might need or want. To account for this we strove to make the design for our system as simple as possible so that that future developers can easily pick up where we left off. This was accomplished by creating an expandable hardware system, modular IoT communication protocols, a flexible database, and a custom mobile application.

We were not able to meet non-functional requirement #1 (NF1), having hardware costs of each system less than $300. Instead, our total system cost is closer to $419, with a monthly server cost of $5 CAD. We suggest that the manufacturing cost could be reduced by 10% if parts are to be purchased in bulk (at least orders of 10) from digikey.ca.

5. RECOMMENDATIONS

This system and its intellectual property (IP) has been delivered to UBC staff Bernhard Nimmervol as of April 2021. He is actively working to promote aquaponics and support aquaponic growers and should be contacted for future inquiries about this project.

5.2 RECOMMENDATIONS FOR FUTURE DEVELOPMENT

We anticipate that the future development on this project will occur in several ways.

It will begin with an unmet need, in monitoring or controlling the system. The user will then identify a type of new sensor they wish to add to the system, or a new control device. If the sensor provides an analog voltage, they can simply connect it to the secondary ADC, or if it is another communication protocol, they can connect it directly to the header pins of the raspberry pi. The user will then need to add a few lines of code to the monitoring loop, updating which data is sent to the app.

If they want to add a new control device, they can connect it to the relays present in the system. Then they will add some control logic to the control loop, and it will be ready for operation.

The software application will automatically listen to and show new values that the hardware publishes, so most development can be focused on the raspberry pi python packages. If users wish to add new features to the app, such as calibration routines, or more advanced control metrics they can add new pages by copying several basic templates that we provide.

Our mobile application access rights currently rely on the unique naming of each monitoring system, which could conceivably pose a security risk. We recommend future development is done to ensure that users cannot access the data from any system, even if they know its name. This could be accomplished by integrating the monitoring system public key into the authentication process; however, this comes at the cost of increased complexity for the user.

6. CONCLUSION

This paper presents an overview of an open-source aquaponic and hydroponic monitoring and control system. We outline the design requirements considered and provide reference to comprehensive documentation of how our design meets these requirements. This system will improve the ease of monitoring hydroponic or aquaponic growing operations, allowing for more people to grow their own food, especially in high density urban environments.

REFERENCES

[1] UBC Strategic Plan - Shaping UBC's Next Century. 2018. https://strategicplan.ubc.ca/wp-content/uploads/2019/09/2018_UBC_ Strategic_Plan_Full-20180425.pdf

[2] SEEDS Sustainability Students. 2020. https://sustain.ubc.ca/teaching-applied-learning/ seeds-sustainability-program/seeds-sustainability-students

[3] Hydroponics vs. Aquaponics - A Complete, and Honest Comparison. 2019. https://www.trees.com/gardening-and-landscaping/ hydroponics-vs-aquaponics

[4] What Are The Benefits of Aquaponics? Organic Farming and More Benefits. 2019. https://gogreenaquaponics.com/blogs/news/what-are-the-benefits-of-aquaponics

[5] Why Aquaponics? https://stuppy.com/aquaponics/aquaponiclearn/why-aquaponics/

[6] Envisioning a Physical Food Hub at UBCO. 2020. https://sustain.ubc.ca/about/resources/envisioning-physical-food-hub-ubco

[7] AMACS Aquaponics Monitoring And Control System. 2017. https://www.hackster.io/locogenii/amacs-aquaponics-monitoring-and-control-system-3c0165

[8] IoT NFT Aquaponic System Controller With WebApp https://www.instructables.com/IoT-NFT-Aquaponic-System-Controler-with-WebApp-Int/

[9] Open Aquarium https://www.cooking-hacks.com/documentation/tutorials/ open-aquarium-aquaponics-fish-tank-monitoring-arduino.html

[10] Hydro.Bot. 2020. https://hydro.bot/

APPENDICES

A: Requirements Document

B: Design Document

C: Verification and Validation Document

THE UNIVERSITY OF BRITISH COLUMBIA

IoT Monitoring of Aquaponic and Hydroponic Food Production

Appendix A: Requirements

UBC Electrical and Computer Engineering Capstone 121

Carson Berry
Lynes Chan
Mason Duan
Jayden Leong
Hannah Xu

for

Bernhard Nimmervoll, UBC Mechanical Engineering
UBC SEEDS Sustainability Program

April 27, 2021

Revision History

Date	Author	Sections	Change
2020-10-05	JL	1.0 - 2.0	Creation of document skeleton
2020-10-30	CB	1.0 - 5.0	Added system overview and skeleton of subsystem sections
2020-10-30	CB	1.0 - 5.0	Edited Document with feedback from Dr. Lusina and Clients
2020-11-28	JL	1.0 - 5.0	Major revisions to all sections for design review 1.
2021-01-15	MD	1.0 - 5.0	Major revisions to all sections for design review 2.
2021-02-19	JL	4.1, 4.2	Increase specificity and measurability of functional and non-functional requirements. Included NF_x and F_x labels for clarity, instead of numbered lists

Contents

1 Context

In this section, an introduction of our clients and their goals is given. Our clients are Bernhard Nimmervoll of the University of British Columbia (UBC) Mechanical Engineering Department, and Nelly Leo of the UBC Social Ecological Economic Development Studies (SEEDS) Program.

Secondly, this section includes a description of aquaponics and hydroponics. Briefly, aquaponics and hydroponics are food production methods known for being more sustainable than traditional food productions practices.

Finally, a description of the problem this project aims to solve will be given. Overall, this project aims to make aquaponics and hydroponics easier to maintain and monitor. This will, in turn, serve to promote aquaponics and hydroponics as sustainable methods of food production.

1.1 Client Introduction

The primary client of this project is Bernhard Nimmervoll, a Technician for the UBC Mechanical Engineering Department. Bernhard is a passionate advocate of sustainabililty initiatives, including being a hobbyist aquaponics grower in his spare time. Bernhard introduced this project as a way to promote aquaponics and hydroponics as sustainable food production methods, and as a way to combat food insecurity.

Bernhard has partnered with a UBC organization called the Social Ecological Economic Development Studies (SEEDS) Sustainability program. The SEEDS program developed out of UBC's commitment to being a global leader in integrated sustainability [1]. As such, SEEDS is an organization whose aims are to advance the well-being of the environment, local biodiversity, food systems, responsible consumption and community inclusion. They are pursuing these aims by supporting student-led research and development about climate friendly food systems, urban food production for community resilience, biodiverse food systems, zero-waste, and food justice and sovereignty. This project is one of many such UBC SEEDS projects.

UBC SEEDS' has five main research goals, which are shown in Figure 1. The research goal, 'Enable the great food transformation', is the primary focus of our project. This goal is typically pursued by researching food security and sustainability, implementing policies and plans, and participating in education, engagement, and demonstration. In our case, our project will contribute to this goal by advancing technology related to sustainable food growth using aquaponics and hydroponics.

Figure 1: UBC SEEDS big five research priorities [2]

Nelly Leo is the UBC SEEDS coordinator overseeing this project. She is responsible for ensuring that this project meets overall SEEDS objectives.

1.2 Hydroponics and Aquaponics

Hydroponics is a method of agriculture where plants are grown in a nutrient-rich aqueous solution. Nutrients in hydroponics are inputted by the grower to nourish plants. This contrasts traditional forms of agriculture, where plants are grown in soil.

Aquaponics is very similar to hydroponics. In aquaponics, plants are also grown in an aqueous solution, however plants recieve nutrients in a different manner. Instead of growers inputting nutrients, plants recieve nutrients from fish waste. To make this possible, fish are raised in water tanks that the plants are grown in. Aquaponics recieves its name because of this, by being a mixture of aquaculture and hydroponics. A simplified diagram of aquaponics is show in Figure 2.

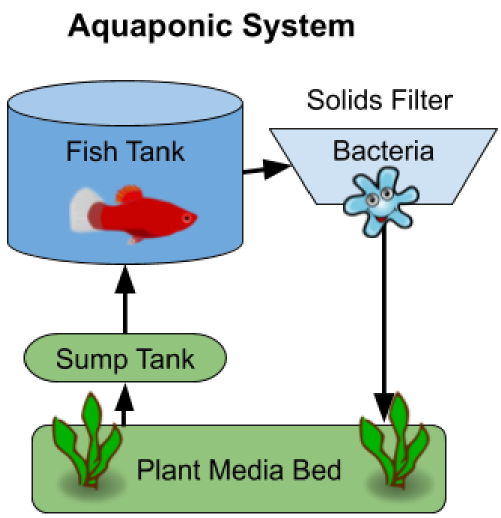


Figure 2: A diagram of a basic aquaponics system

Both hydroponics and aquaponics are being developed as more sustainable approaches to agriculture than traditional methods for the following reasons [3]:

1. Higher crop production per unit area;
2. Faster plant growth;
3. Lower water usage;
4. Lower susceptibility to pests if grown indoors.

For aquaponics, there are also the following additional benefits [4][5]:

1. Fish and plants form a circular, self-sustaining ecosystem;
2. Fish food is the only input needed;
3. Fish as well as plants can be harvested for food.

For the benefits listed above, our clients, Bernhard Nimmervoll and UBC SEEDS, are interested in improving and advocating for these more sustainable agricultural methods.

1.3 Problem

Both aquaponic and hydroponic systems need to be carefully monitored. Plants require that the aqueous solutions they are planted in have all neccesary nutrients. They also require other environmental variables to be monitored, including solution pH, light intensity, humidity, and more. In addition, aquaponic systems are even more difficult systems to maintain due to their complex multi-species dynamics. Systems need to be balanced such that fish, bacteria, and plants can all thrive.

With this system complexity, hydroponics and aquaponics require frequent supervision and maintenance by a knowledgeable operator. This adds significant cost and labor for aquaponics and hydroponics growers. This can be discouraging for small scale and hobbyist aquaponic or growers who do not currently have access to an affordable, automated monitoring and control system.

2 Domain

This project is within the domain of aquaponics and hydroponics monitoring and control systems typically built by hobbyists and small companies. These are small-scale monitoring and control systems usually designed for and installed on a single aquaponics or hydroponics system. They are also usually open-source and non-proprietary. The vast majority of these systems monitor their systems using internet connectivity, making them categorizable as small-scale internet of things (IoT) projects.

Many of the previous implementations of systems of this type are simplistic and designed for single systems without much forethought for future development [7][8]. Some of the systems have been designed with future improvements, ease of setup, cost, and efficiency in mind [9][10]. An example of such a system is shown below in Figure 3. This system is quite expensive for an monitoring system that is built using off the self components, costing around $850. The system also seems to have limited external monitoring capabilities and no licensing for open source although they claim to be an open source project. As well, the monitoring system doesn't record any data and does not seem to have any response system in place in case of power failure.

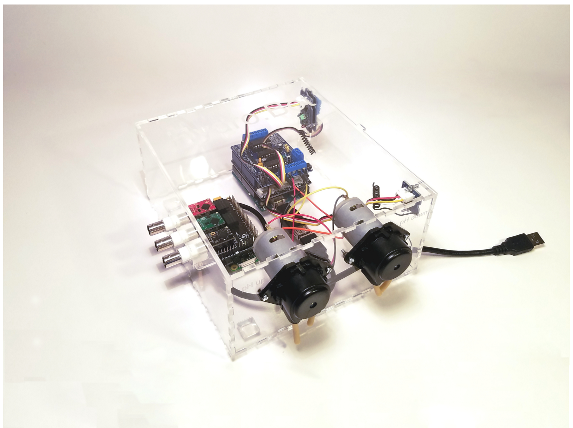


Figure 3: An example of an existing system called HydroBot [10]

3 Goal

The goal of this project is to develop a user-friendly, no-technical-expertise-needed solution for monitoring and control of hydroponic or aquaponic systems. This system should utilize open-source and low cost technologies wherever possible to minimize barriers to replication and access. In terms of scope, our project serves to deliver a working prototype of such a system which is installed on an existing hydroponic or aquaponic system.

When completed, this project will help solve the problem of hobbyist and aquaponic growers monitoring and maintaining their systems. It will help reduce cost, time, and effort associated with aquaponics and hydroponic systems; and help reduce system failures resulting in production losses.

In addition to these benefits, this project will also contribute towards larger goals including increasing food systems sustainability, reducing food insecurity, and increasing sustainability awareness. First of all, by making hydroponics and aquaponics systems easier to develop and maintain, this may encourage more growers to use these methods. This, in turn, can promote food systems sustainabililty as aquaponics and hydroponics have advantages over traditional growing methods.

In terms of food insecurity, this project may be included in future on-campus aquaponics or hydroponics growing systems. For example, a physical 'FoodHub' is being envisioned UBC Okanagan (UBCO) campus [6]. This 'FoodHub' may provide affordable, local, organic vegetables and fish to UBCO community members as an effort to reduce student food insecurity. This Food Hub is being planned to include an aquaponic system of both functional, research, and educational value. The product of this project could be a helpful addition to monitor such a system, and could therefore contribute to broader issues including reducing food insecurity.

4 Requirements

In order to meet the goals in section 3, our team has generated functional and non-functional requirements for this project.

4.1 Functional Requirements

This project must perform the following functions:

- F_1 **Display measurements of critical system variables in hydroponic or aquaponic systems.** For the scope of our project, critical system variables include and are limited to pH, water level, water leakage, and water temperature. All displayed measurements must be updated at a minimum of every five minutes.
- F_2 **Store and display past measurements of critical system variables so hydroponic or aquaponic growers can analyze system trends.** The system must be capable of storing and displaying at least the past 3 months worth of measurements.
- F_3 **Send alerts to growers within 1 minute of a detecting that a critical system variable is outside of a normal range.** Growers will specify normal ranges for each of the critical system variables.
- F_4 **Adjust pH level and water level automatically such that they always stay within a pre-defined range.** This pre-defined range for pH and water level can be configured by growers depending on the needs of their hydroponic or aquaponic system.
- F_5 **Operate for at least 8 hours in the event of a power supply failure or outage.** The system must also send an alert to growers indicating that there was a power failure. Alerts should occur a 1 minute of the power failure.
- F_6 **Give growers the ability re-calibrate sensors and actuators.** These re-calibrations can be made at any time.

4.2 Non-Functional Requirements

The following is a list of non-functional requirements for this project. This project must:

- NF_1 **Cost less than $300 CAD per system.** Additional monthly servicing fees of must also be less than $5 per month.

NF_2 **Be designed to work with any small-scale aquaponics or hydroponics systems**. For this project, small-scale aquaponics or hydroponic systems have under 200 plants and under 3000 gallons of water.

NF_3 **Ensure that only permitted growers can monitor and control hydroponic or aquaponic systems.** Future administrators of our project must enable growers to monitor and control a given hydroponic or aquaponic system. Otherwise, access to monitor a given hydroponic or aquaponic system will not be permitted.

NF_4 **Be water resistant.** An IP rating of IP67 (Protected from immersion between 15 centimeters and 1 meter in depth) must be satisfied.

NF_5 **Be able to monitor different or additional aquaponics or hydroponics system variables, depending on the grower's preference.**

NF_6 **Measure all critical system variables within an accuracy of 5%.**

NF_7 **Maintain all controlled variables within an accuracy of 10%.**

5 Constraints

The following is a list of constraints on our design:

C_1 **All software developed must be open-source, with appropriate licensing to allow for legal modification, duplication, and sharing.** The license we have chosen is the GNU General Public License v3.0 as it allows the community to make changes and to use the code for personal purposes but prevents distributing closed source versions.

References

[1] UBC Strategic Plan - Shaping UBC's Next Century. 2018.
https://strategicplan.ubc.ca/wp-content/uploads/2019/09/2018_UBC_Strategic_Plan_Full-20180425.pdf

[2] SEEDS Sustainability Students. 2020.
https://sustain.ubc.ca/teaching-applied-learning/seeds-sustainability-program/seeds-sustainability-students

[3] Hydroponics vs. Aquaponics - A Complete, and Honest Comparison. 2019.
https://www.trees.com/gardening-and-landscaping/hydroponics-vs-aquaponics

[4] What Are The Benefits of Aquaponics? Organic Farming and More Benefits. 2019.
https://gogreenaquaponics.com/blogs/news/what-are-the-benefits-of-aquaponics

[5] Why Aquaponics?
https://stuppy.com/aquaponics/aquaponiclearn/why-aquaponics/

[6] Envisioning a Physical Food Hub at UBCO. 2020.
https://sustain.ubc.ca/about/resources/envisioning-physical-food-hub-ubco

[7] AMACS Aquaponics Monitoring And Control System. 2017.
https://www.hackster.io/locogenii/amacs-aquaponics-monitoring-and-control-system

[8] IoT NFT Aquaponic System Controler With WebApp
https://www.instructables.com/IoT-NFT-Aquaponic-System-Controler-with-WebApp-Int

[9] Open Aquarium
https://www.cooking-hacks.com/documentation/tutorials/open-aquarium-aquaponics-fish-tank-monitoring-arduino.html

[10] Hydro.Bot. 2020.
https://hydro.bot/

THE UNIVERSITY OF BRITISH COLUMBIA

IoT Monitoring of Aquaponic and Hydroponic Food Production

Appendix B: Design

UBC Electrical and Computer Engineering Capstone 121

Carson Berry
Lynes Chan
Mason Duan
Jayden Leong
Hannah Xu

for

UBC SEEDS Sustainability Program
Bernhard Nimmervoll, UBC Mechanical Engineering

April 27, 2021

Revision History

Date	Author	Sections	Change
2020-10-05	JL	1.0 - 2.0	Creation of document skeleton
2020-10-30	JL	1.0 - 5.0	Added system overview and skeleton of subsystem sections
2020-11-29	LC	3.0 - 4.0	Added the design decisions made for the sensors and the control devices
2021-02-19	CB	3.0 - 4.0	Added details about power architecture, sensors, mounting, hardware, heat management
2021-02-20	JL	1.0 - 2.0	Updated system overview to add brief overview of hydroponics and aquaponics, new system diagram and updated purpose. Updated IoT section with details about the cloud implementation, along with other minor edits to microcomputer section.
2021-02-20	MD	6.0	Updated software section to include use case digagram, screen explanations, and how the mobile application interacts with the back end.
2021-04-11	CB	3.0-4.0, 7.0	Updated power design, sensor firmware, hardware design, added BOM tables, and ordering information.
2021-04-12	JL	1.0-2.0	Added references and further detail to overview and background sections. Updated IoT section with references and implementation details.
2021-04-14	MD	6.0	Updated software section to include final use case diagram design, screen explanations, and how the mobile application interacts with the back end.

Contents

1 System Overview

Since our project is designed to work with both hydroponic and aquaponic systems, this section provides an brief introduction to both hydroponics and aquaponics. Afterward, the purpose of this project is given, and the high-level system design is described.

1.1 Hydroponics and Aquaponics

Hydroponics is a method of agriculture where plants are grown in a nutrient-rich aqueous solution [1]. Nutrients in hydroponics are inputted by the grower into the aqueous solution to nourish plants. This contrasts traditional forms of agriculture, where plants are grown in soil.

Aquaponics is very similar to hydroponics. In aquaponics, plants are also grown in an aqueous solution, however plants recieve nutrients in a different manner. Instead of growers inputting nutrients, plants recieve nutrients from fish waste [2]. To make this possible, fish are grown in separate tanks from the plants, with a filtration tank, and sump tank between them. Aquaponics recieves its name because of this, by being a mixture of aquaculture and hydroponics. A simplified diagram of aquaponics is show in Figure 1. Note that hydroponics systems look nearly the same, minus the fish tank.

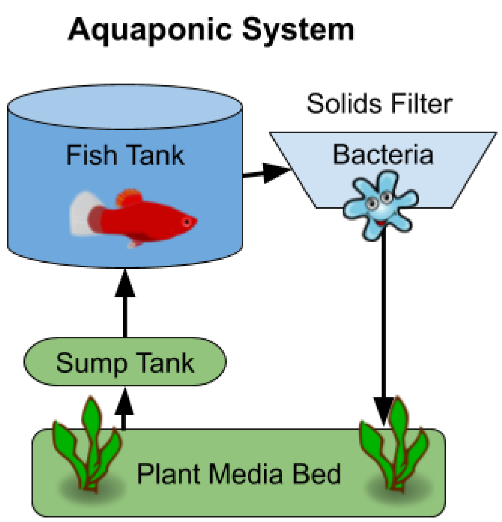


Figure 1: A diagram of a basic aquaponics system

1.2 Project Purpose

The goal of our project is to develop a system to monitor and control critical variables for hydroponic and aquaponic systems. In particular, our project focuses on monitoring water variables as they are most difficult to maintain at healthy levels in aquaponic and hydroponic systems. These variables include water pH, level, leakage, and temperature.

In terms of users, this system will be designed for small scale and hobbyist growers who maintain their aquaponic or hydroponic systems. Note that for this project, small-scale aquaponics or hydroponic systems are defined as systems with under 200 plants and under 3000 gallons of water. Since our users are small-scale growers and hobbyists, the system will utilize open-source and low cost technologies wherever possible to minimize barriers to usage and replication.

Please refer to the included *Requirements* document for further details about system requirements, background, and purpose.

1.3 High-Level System Design

Figure 2 below provides an overview of our design. As you can see, our design contains the following four subsystems: sensors, control, Cloud, and software application.

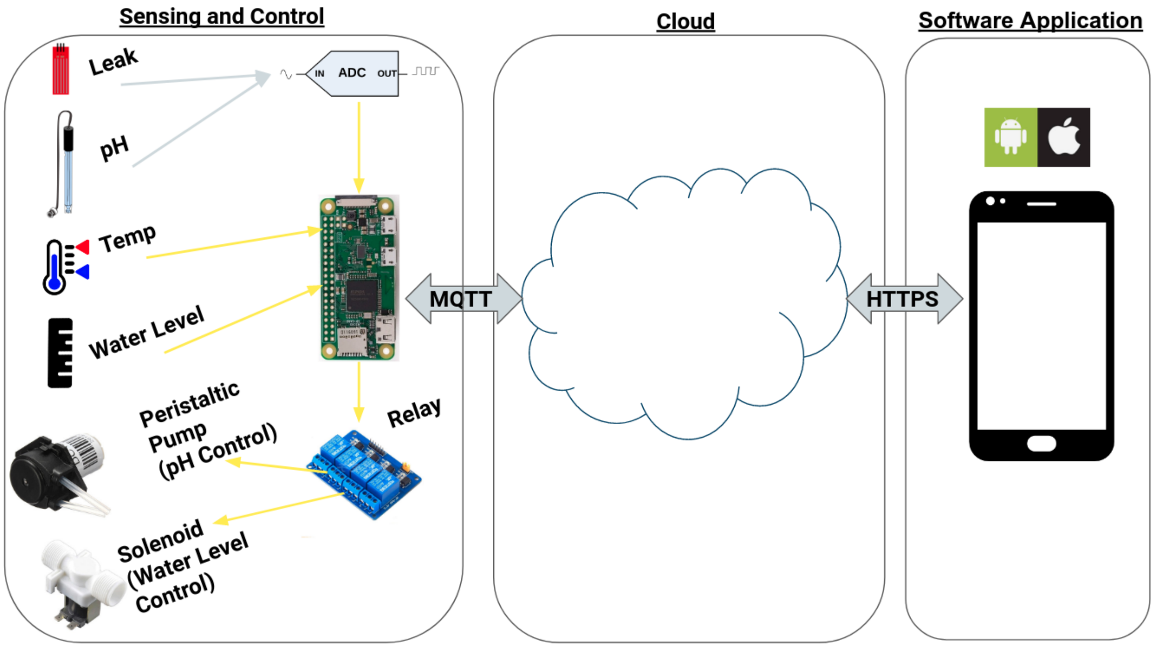


Figure 2: System Level Block Diagram

First, we can see the sensing and control subsystems in the "Sensing and Control" block in Figure 2. This consists of a microcomputer attached to multiple sensors and control devices. As previously mentioned, there are four key sensors in our project shown: the water leakage, pH, temperature, and level sensors. These sensors allow our design to take measurements of the water of aquaponics or hydroponics systems. As such, the pH, temperature, and water level sensors are placed inside water tanks in an aquaponic or hydroponic system. As far as the control systems are concerned, two devices are attached to the microcomputer: a peristaltic pump and a solenoid. These two devices allow our design to automatically adjust pH and water level respectively.

Next, the Cloud subsystem is shown in “Cloud” block in Figure 2. This system’s responsibility is to transfer data between a mobile software application and the sensing and control system. It also stores past sensor measurements for later analysis.

Finally, on the software application is show in the “Software Application” block in Figure 2. This is where the user primarily interacts with their hydroponic or aquaponic system. This application displays past and current sensor measurements. It also allows the user to recieve alerts if urgent issues are detected by the sensing and control system, including issues like power failure and dangerous water pH levels.

In the following sections, the design of each of these subsystems will be explored in further detail.

2 Internet of Things (IoT)

The IoT subsystem recieves its name from the design decision to use internet communication to communicate with sensors and control devices installed on aquaponics or hydroponic systems. This enables our design to leverage advantages of internet communications. One key advantage is that the vast majority of our users have internet connected devices including laptops and mobile phones. Another key advantage is the ability for growers to monitor and control their aquaponics systems from anywhere they have internet connectivity.

The IoT subsystem, in our project, refers to all hardware and software needed to communicate between sensors and output devices and the software application. See Figure 3 for an overview of our IoT subsystem design.

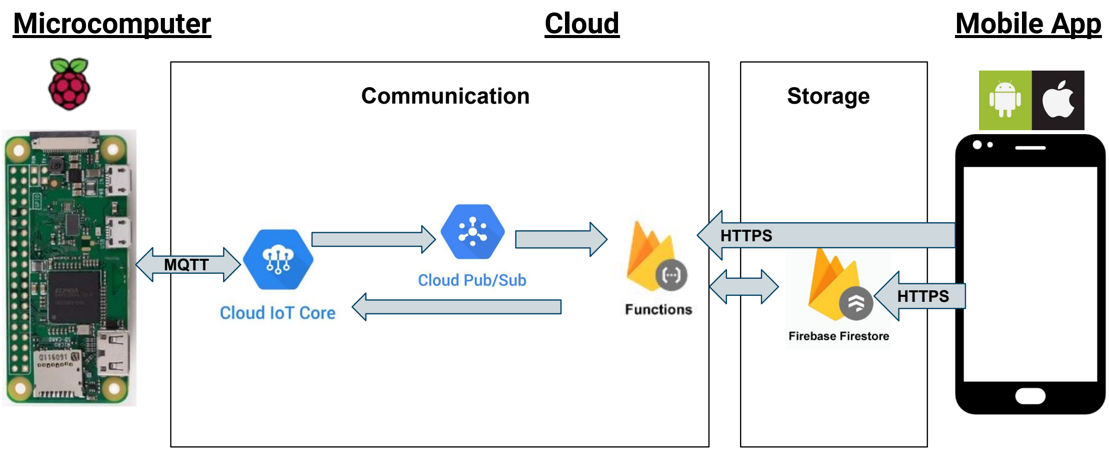


Figure 3: IoT system overview

As seen in Figure 3, our IoT subsystem has 3 major components: the microcomputer, cloud, and mobile devices. The microcomputer and cloud components will be discussed in detail within this section. In the Software Application section, the mobile app component will be discussed further.

2.1 Microcomputer

There are two key functions that the microcomputer serves in our design. The first responsibility is to receive data from sensors, and to control output devices. The second responsibility is to send sensor readings to users, and to recieve commands for output devices from users. For each of these key functions, our team has made chosen design solutions to help address these issues. Consult Table 1 for a description of the key functions and their associated technological strategies.

Table 1: Description of microcomputer key functions and design strategies

Key Function	**Req(s)**	**Design Solution**	**Reason(s)**
Communicate with sensors and output devices	F_1, F_4	Wired connection (e.g. SPI, IIC, GPIO), or wireless connection (e.g Bluetooth) depending on the sensor or control device	• Allows the flexibility to interface with many different sensors and output devices.
Communicate with user devices	F_1, F_2, F_3, F_6	WiFi internet network connection	• Low cost compared other alternatives like LTE • High accesibility as hobbyist and small-scale growers typically have a WiFi network setup within range of their system.

With these key functions in mind, our team selected a capable microcomputer. This microcomputer is the Raspberry Pi Zero W [3], shown in Figure 4.

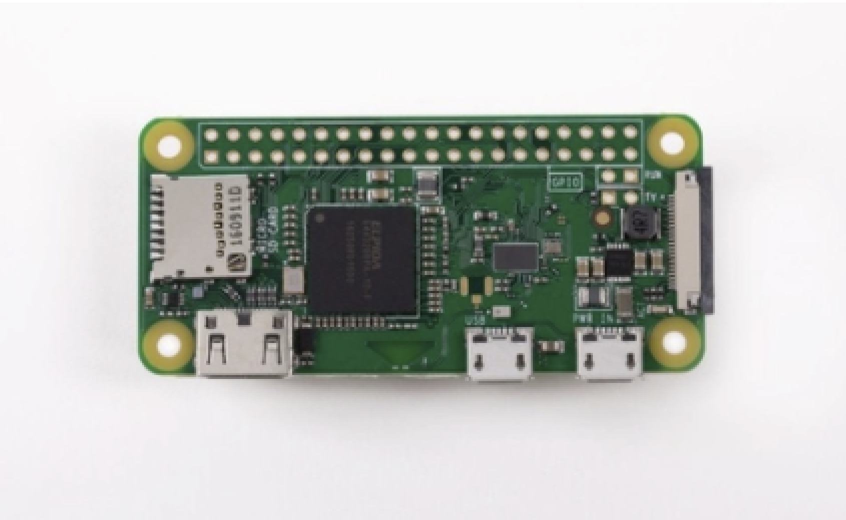


Figure 4: The Raspberry Pi Zero W microcomputer

The Raspberry Pi Zero W was chosen over other microcomputers that could perform the same key functions. This was done for the following reasons:

1. **Low cost:** costs around 40 CAD.
2. **Wireless networking:** supports both WiFi and Bluetooth.
3. **Versatile:** runs a Linux-based operating system called Raspberry Pi OS.
4. **Documentation and support:** Raspberry Pi is a popular platform with extensive tutorials, resources, and forums.
5. **Interfacing:** supports a variety of wired protocols including SPI, I2C. Also has 40 GPIO pins to interface with multiple devices.

Other notable microcomputers that were considered ESP32-based microcontrollers, and the Raspberry Pi 3B. However, the ESP32 was not chosen due to its more limited documentation and inability to run a linux-based operating system (making development slower). The Raspberry Pi 3B was not selected because of its similar functionality to the Raspberry Pi Zero W, but its higher cost of about 80 CAD.

Nevertheless, a final benefit of chosing a Raspberry Pi Zero W is that its software is compatible with other Raspberry Pi models, like the 3B. This gives design flexibility to optionally use more powerful or improved Raspberry Pi models other than the Raspberry Pi Zero W.

2.2 Cloud

The Cloud component of our IoT subsystem sits in between the microcomputer. Cloud in our project refers to cloud computing. This constitutes all the internet connected servers and associated IT infrastructure needed to communicate between the microcomputer (MCU) and the user's mobile devices. As such, the cloud is responsible for the following:

- Relaying information between the MCU and user's mobile devices
- Storing past sensor measurements
- Providing a software interface to the user's mobile devices so they can access past sensor data and communicate with the MCU.
- Scaling to allow monitoring and control of multiple aquaponics or hydroponics systems from a single mobile device.

As such, the cloud contributes to meeting the functional requirements F_1, F_2, F_3, and F_6. Please refer to the *Requirements* document for details about these requirements.

As far as design solutions are concerned, we considered 2 key alternatives based on the above specifications. The first was using IT infrastructure as a cloud proxy to communicate between the microcomputer and the mobile application. The second choice was to communicate directly with the MCU, without any infrastructure in between. These options are summarized in Figure 5.

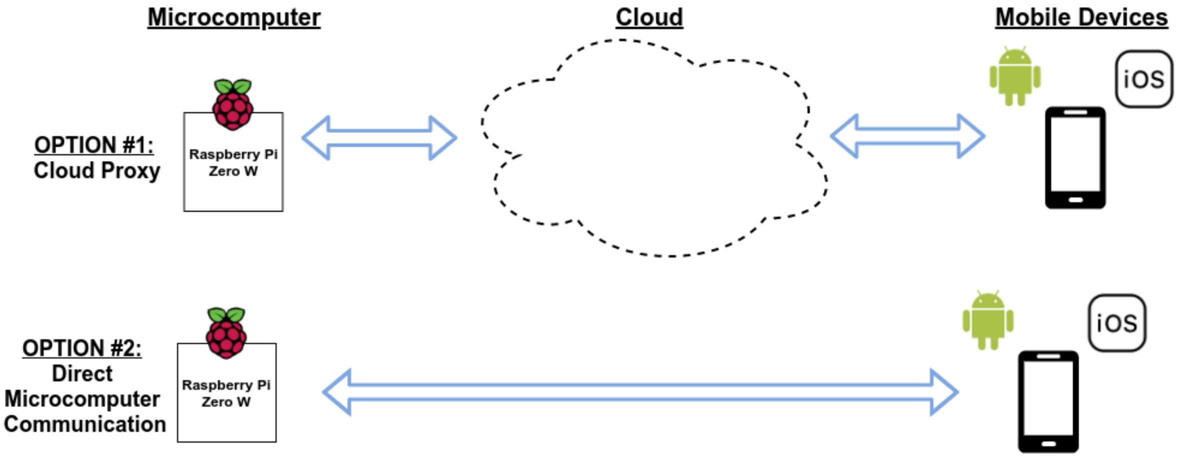


Figure 5: Two MCU to mobile application communication options.

To decide between these two main options, we weighed their strengths and weaknesses relative to their key functions. This comparison is summarized in Table 2.

Table 2: Comparison of MCU to mobile application communication options.

Function	Cloud Proxy	Direct MCU communication
Relaying information between the MCU and user's mobile devices	**Advantage:** User devices can communicate the MCU anywhere with internet connectivity.	**Advantage:** No cost for proxy service. **Disadvantage:** User devices must be on the same local network as the microcomputer.
Storing past sensor measurements	**Advantage:** Cloud proxy services can automatically provision data storage to meet unlimited demand inexpensively.	**Disadvantage:** Users must be on the same local network as the MCU to access data.
Providing a software interface to the user's mobile devices	**Advantage:** Cloud proxy services can offer APIs and pre-built tools to connect with mobile devices.	**Advantage:** communicating directly with an MCU can reduce software interface complexity and maintenance.
Scaling to allow monitoring and control of multiple aquaponics or hydroponics systems.	**Advantage:** Cloud proxy services offer automatic provisioning of resources based on demand	**Disadvantage:** limited to monitoring only systems on the same local network.

As seen in Table 2, the advantages of the cloud proxy option outweigh the direct MCU communication option. Most notably, the cloud proxy option offers more flexibility as it is not limited to being on the same private network as the MCU. It also is more capable of storing large amounts of past sensor data.

After choosing to use the cloud proxy option, we had a choice between 3rd party service providers, or managing our own cloud computing resources. We ultimately chose to use 3rd party IoT service providers over managing our own cloud computing resources for the following reasons:

1. **Low cost:** The estimated cost is less than 5 CAD/month.
2. **High implementation speed:** Service providers manage cloud computing resources, instead of developers needed to create and manage these from scratch.
3. **Easy maintainance:** Service providers typically provide web consoles for easy maintainability of resources.
4. **Automatic scalability:** Service providers automatically scale resources to match the amount of usage. For instance, more memory in a database is automatically provisioned.

Some popular 3rd party IoT service providers include Microsoft Azure, Google Cloud IoT, and Amazon Web Services (AWS) IoT. These service providers offer very similar services at comparable prices. So, our team decided to develop using Google Cloud IoT because it offered a better software interface to our mobile application.

This software interface will be discussed in further detail in the following sections. In addition, the architecture of our cloud system will be described in terms of two aspects: cloud communication and cloud storage.

2.2.1 Cloud Communication

Referring back to Figure 3, we see that the cloud communication component transfers data between the MCU and cloud storage. These components therefore help satisfy the functional requirements F_1, F_2, and F_3, which require communication between the mobile application and the sensing and control system.

From Figure 3, we also see that the cloud commmunication part is made of 2 key components which are Google Cloud services. These are Google IoT Core, Google Cloud Pub/Sub, and Google Firebase Cloud Functions. See Table 3 below to see the role of each of these services in our design.

Table 3: Roles of Google Cloud communication services used.

Google Cloud Service	**Role**
IoT Core	• Provides an API for MCUs to communicate with Google Cloud using the MQTT protocol. • Provides MCU authentication and data encryption using public/private RSA keys
Pub/Sub	• Automatically recieves incoming messages from Google Cloud IoT core. • Allows Google Firebase Cloud Functions to respond to each message that is published by an MCU.
Firebase Functions	• Provides the ability to write functions that run on Google managed servers. • Writes MCU messages (with sensor data) in databases. These messages can then be retrieved by the mobile application. • Sends commands to MCUs through Google IoT Core.

Lastly, note that the communication protocol used between the MCU and Google Cloud IoT Core is MQTT. This was chosen over other possible protocols like HTTPs as it uses lower bandwidth and has less software overhead [4]. These two factors decrease the power consumption and data usage of our design.

2.2.2 Cloud Storage

Referring back to Figure 3, we see that the cloud storage components saves messages from both the MCU and the mobile application.

The key requirement satisfied by this section is F_2, to be able to store past sensor measurements for analysis. However, like the previous section, the satisfy the functional requirements F_1, and F_3, which require communication between the mobile application and the sensing and control system.

Table 4 also shows another Google services that are used in our design: Google

Firebase Firestore.

Table 4: Roles of Google data storage services used.

Google Service	Role
Firebase Firestore	• Stores sensor measurements in a NoSQL format. • Stores MCU configurations.

Google Firestore is used for storage of past sensor measurements. This is because Google Firestore charges a cheaper rate for storage than other Google databases, making it preferable for storing large amounts of sensor measurement data [5]. It allows features a subscription model where new sensor readings are updated within our app as soon as they are stored in Firestore [6]. This allows us to display most recent sensor updates in near real-time.

Google Firestore also stores MCU configurations. When these configurations are updated, Google Cloud Functions automatically updates a device configuration. This is used to send commands to the sensing and control system from the mobile application. This design was chosen as it allows the mobile application to both read the current MCU configuration, and change it. That way, the mobile application can make configuration updates based on a MCU's current status. For instance, if the pH level of a given MCU configuration is below the target value, it could be changed to a higher value by updating that MCU's configuration.

2.2.3 Cloud System Implementation

As mentioned in section 2.2.1 above, Google Firebase Cloud Functions are used to save sensor messages, and connect IoT devices to the mobile application. So, just as depicted in Figure 3, Google Firebase Cloud Functions are central to the cloud system and define much of its functionality.

The implementation of this project's Google Firebase Cloud Functions can be found in the piponic_cloud repository (`https://github.com/jaydenleong/piponic_cloud`). The README of this repository also provides further documentation about how the overall cloud system can be setup and maintained.

3 Power

Our system is comprised of various electric components, including the Raspberry Pi microcomputer, sensors, and output devices. Power for the system will be available from two sources, a main, and a backup. The main power source is 12V DC. This is supplied by a commercially available power supply that rectifies and steps-down voltage coming from a 120V wall outlet. This will provide power to the sensor hub during normal operation. That is, when the device is plugged in, and the line-voltage is available.

3.1 Back-up Power

The backup power will be provided by a battery pack with 2 AA batteries, with voltage boosted to 5 V from a DC-DC boost converter. The arrangement of these two inputs can be seen in figure 6. 3 AA batteries could also be used in series to produce a voltage of 4.5V, which will increase the lifespan of the back-up power system and the efficiency of the boost-converter chip. If this arrangement of batteries is used, a voltage divider must be added to the battery-voltage monitoring circuit, so that the maximum voltage of 4.5V is read at max 3.3V. We recommend using 100k and 36.9k resistors for this voltage divider to minimize power loss.

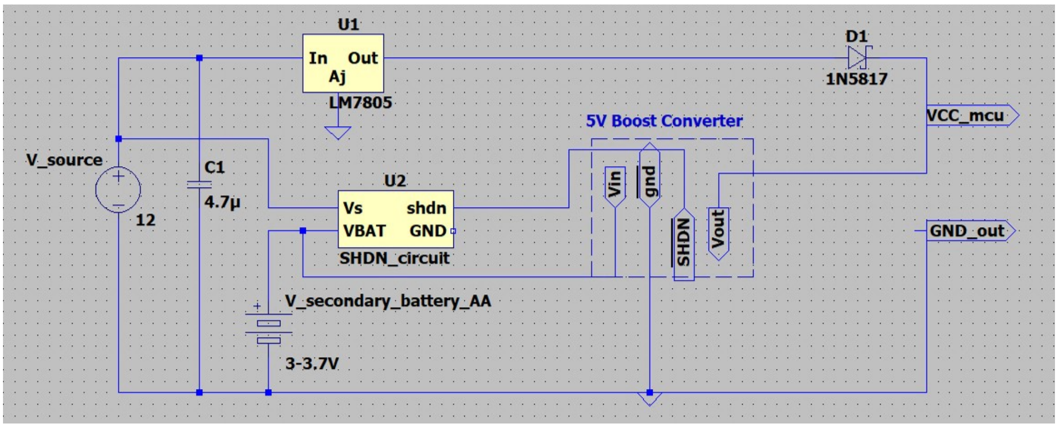


Figure 6: Uninterruptible power supply architecture.

The LM7805 voltage regulators are responsible for delivering all of the current used by the 5V devices during normal operation. It has been observed that all of the 5V devices consume around 200mA during normal operation, and the voltage regulators are rated to 1A. To increase the safety margin associated with heat-burnout, a 15g aluminum TO-220 heat sink was added to the LM7805 chip. Two LM7805 regulators are used in parallel to ensure adequate power is always available. This was also added to the Schottky diode at the output of the LM7805 chip.

To conserve the battery-lifespan, the boost converter is turned off automatically when the 12V line voltage is present. The circuit that accomplishes this can be seen in figure 7 - a simple BJT switch application. In the event of a power failure, the system will be able to run for 8 hours on backup power. This functionality is intended as a way to keep monitoring the condition of the aquaponic or hydroponic systems in the event of a power failure.

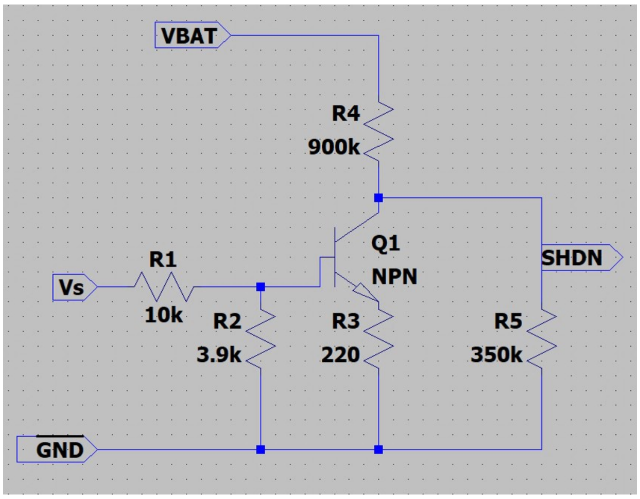


Figure 7: Boost converter shutdown circuit. This circuit disables the AA batteries when 12V is present.

For boosting 2 AA batteries up to 5 V, the Pololu 5V Step-Up Voltage Regulator U1V11F5 is a high quality, low cost, easily accessible option that can be implemented by hobbyists with minimal difficulty. It can take anywhere between 0.5 and 5 V input and will automatically boost it to 5.02 V output.

3.2 Power Budget

Our monitoring system consumes on average 810 mW when on back-up power. Most of this comes from the operation of the raspberry pi. A breakdown of the average power use can be seen in figure 8.

The control devices and relays operate only when line power is available (that is, when not on back-up power), and when all control devices are active the system consumes an additional 5W.

3.3 Current Protection

The output voltage will be connected via Schottky diode to the line voltage in, before being connected to the Raspberry Pi. This will effectively OR the voltage source seen by the Raspberry Pi. Several other circuit topologies were simulated and tested, and the Schottky diode topology represents the best functionality in both

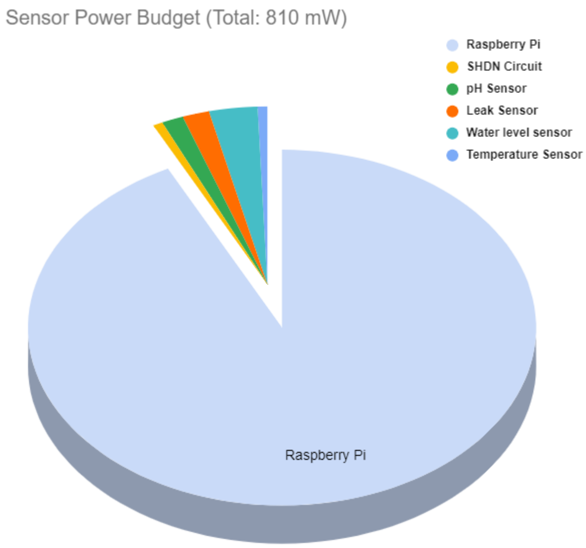


Figure 8: Average monitoring system power breakdown.

Table 5: Power current protection device comparison.

Device Type	Current Protection	Power Loss	Line Voltage	Backup Voltage
N Mosfet	No	2 mW	5 V	5 V
P Mosfet	Yes	2 mW	5 V	4 V
Schottky Diode	Yes	60 mW	4.7 V	5 V

normal operation and backup power mode. The other topologies that were modelled prior to making the design decision were a P-channel mosfet or an N-channel mosfet.

The N-channel mosfet does not achieve the desired functionality of 1-way current protection, so this solution is disregarded. The P-Channel mosfet does an excellent job minimizing power loss, however in the back-up power scenario, drains an unacceptable amount of current such that the line voltage will be 4V, which is too low for our applications. Therefore, the best solution is the schottky diode, despite its relatively high power consumption, because it keeps the line voltage at an acceptable level in both operating conditions.

This system is intended to be used with a LiPo battery pack as a backup battery source, however due to safety considerations, the implementation of this will be limited to a recommendation and design only. If one is using LiPo batteries to provide backup power, please familiarize yourself with the appropriate safety procedures associated with using LiPo technology.

For keeping a LiPo battery fully charged while plugged in, and boosting the voltage a 3.7V LiPo cell to 5V when using, the PowerBoost 1000 Charger by Adafruit is our recommended breakout board. It supports load-sharing which means the board can supply power to the hub at the same time as charge the battery. It accomplishes this with 2 special ASICs. The input line voltage can be directly input into the PowerBoost 1000, which will automatically manage OR-ing the power for the Raspberry Pi.

4 Sensors

Our design is centered around 4 types of sensors: Temperature, pH, water-leak, and water level. These sensors focus on the the status of the water in the system because this will effectively address the needs of fish, plants, and microbes, as water is the media through which these species interact.

4.1 Sensor Type Selection

There are many distinct indicators of ecosystem health when monitoring a hydroponic or aquaponic system [7]. There's aqueous concentration of carbon dioxide and oxygen. There's accumulation of bio-solids from fish waste in the water. There are three different important types of dissolved nitrogen: nitrate, nitrite, and ammonium. The alkalinity and pH of the water are both very important as well. The water level, flow-rate, and temperature are not to be overlooked either.

Implementing a system that monitors all of these important summary variables is possible, however would require significant investment of both time and money. Limited as we are in both time and money, we decided to focus on the most important, the lowest cost, and the easiest to implement sensors.

By building our project with simple sensors, we make our design accessible to as many people as possible. Our sensor-framework can easily be expanded upon by future developers or users, who can choose to purchase more expensive sensors if they wish to.

We identified 4 types of sensors that fit the above criteria

1. pH
2. Temperature
3. Water-level
4. Water-leak

The other sensor types were either too expensive or did not add enough valuable information to the monitoring system for them to be considered.

4.2 Analog to Digital Conversion

The Raspberry Pi does not have an analog to digital converter (ADC) built in. Several of the sensors that are being used, and that future designers could use will require the use of an ADC, so we added one to our project. The ADS1115 is a 16 bit, 4 channel, programmable gain, I2C ready ADC, which can operate between 3.3 and 5 volts. These specifications meet the needs of this project by maximizing resolution, minimizing pin use, minimizing power consumption, and having sufficient channels. The ADS1115 was designed to be a very power-efficient board by

Currently 4 of the ADC channels are being used, leaving 4 free for new features. The current channels in use are:

- External Leak Sensors
- Enclosure Internal Leak Sensors
- pH Sensor
- Battery voltage

The pins of the Raspberry Pi are designed to operate at 3.3 V, however have large enough pull-up and pull-down resistors that they can handle 5V as well. To maximize longevity, and minimize chance of pin burnout, the ADS1115 ADC is operated at 3.3V.

By operating the ADC at 3.3V, the maximum voltage it can read from one of its input channels is now 3.3V also. If any component cannot run at 3.3V, then it must be reduced to 3.3V from it's higher voltage (such as 5V) so as to not damage the ADC. This can be accomplished with a voltage divider.

4.3 Temperature

Temperature is important in aquaponic or hydroponic growing systems because different species all have their own preferred temperature range. If a user is growing multiple species in a single system, they may wish to control the temperature so that all of the species they are growing are within the functional temperature limits.

We considered three types of temperature sensor. We chose DS18B21 ($10) temperature sensor over the DS18B20($5) and I2C BME280($20) temperature sensor. This decision was made because:

- The DS18B21 provides temperature values every minute and allows the Implementation of a system that can display values in real time. Requirement: F_1

- The DS18B21 is waterproof and the other alternatives are not. Choosing this sensor leads to a system that is water resistant. Requirement: NF_4
- The DS18B21 is in the mid price range and contributes to an affordable system. Requirement: NF_1

The temperature sensor communicates over 1-wire interface with the Raspberry Pi. The Raspberry pi operating system has convenient protocols for 1-wire interfaces that automatically receive the data from these devices and store them in a given location in memory. The script that reads the temperature, quite simply, checks this location in the memory for the temperature, and performs some simple cleanup.

4.4 pH

In hydroponic and aqauponic systems, the pH has been described by experts as the *Master Variable* [7]. This means that it controls or influences many metabolic processes. It is essential that the pH of the system does not go outside of the maximum or minimum tolerable limits of the organisms present in the growers system.

For these reasons, we deemed the pH to be an essential part of our monitoring solution.

We considered many types of pH sensors among many and initially chose the PH-4502C($39) over the LeoEc($60) because:

- The PH-4502C is a low power sensor (less than 0.5W) and will therefore provide better lifespan in the event of a power failure. Requirement: F_5
- The PH-4502C provides pH values with a delay of 0.5s. Requirement: F_1
- The PH-4502C is lower priced compared to the LeoEc and middle priced compared to other pH sensors that don't meet our requirements. This choice contributes to an affordable system. Requirement: NF_1

The PH-4502C is intended to be used by mapping pH 7 to 2.5V, 14 to 0V, and pH 0 to 5V. These values rarely seem to be present in reality, necessitating significant calibration.

This pH sensor performed poorly in testing, and so another type of pH sensor was purchased; the SEN0161-V2 from DFRobot. This electrolytic pH probe works on the same principle of the one above but has a different amplification board that does not have any mechanical calibration requirements. It also meets all of the other requirements outlined above.

Figure 9: pH sensor Kit

4.5 Water Level

Measuring the water level enables growers to ensure that they have adequate water in their system for their plants and fish. If the water level is too low the system can heat-up quickly or change in temperature or chemical concentration. The water level is measured with the intention of automating the water refilling process.

There are different ways to measure the water level. We chose the Binary Water Level Detector, SEN0204-ND ($13). The other alternatives are the Ultrasonic Sensor, HC-SR04 ($4) and the E-tape Liquid Level Sensor ($60).

We made this decision because:

- The SEN0204-ND provides the minimum data to maintain a desired water-level F_1
- The SEN0204-ND is small and can scale very well to differently sized systems. This makes the sensor easier to install on many more types of aquaponics systems compared to alternatives. NF_5
- The SEN0204-ND is low cost. The HC-SR04 and other Ultrasonic Sensors have a very short lifespan and break very easily, especially in environments

where water is involved. So in the long term, the SEN0204-ND contributes the most to an affordable system as it is IP67 rated. NF_1, NF_4

4.6 Water Leak Sensor

For indoor aquaponic or hydroponic systems a leak may be a catastrophic failure that could cause significant property damage or could result in eviction of the grower(s). Detecting leaks, is therefore of paramount importance to ensuring long-term social sustainability of aquaponic or hydroponic systems in urban environments.

Leak sensors could be implemented in many ways. We chose a very low cost hobbyist leak sensor available online as seen in figure 10.

In our design, this sensor is also placed inside the enclosure to detect if any of the system electronics may be compromised.

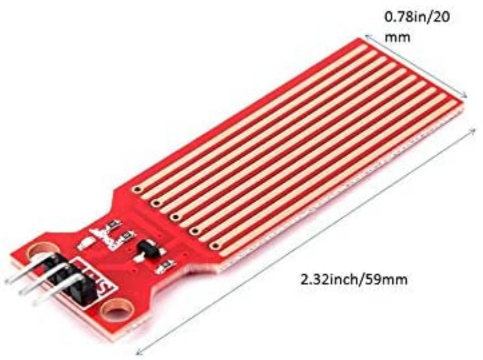


Figure 10: Water leak sensor

We made this decision because:

- The leak sensor provides the minimum data to detect a leak, and can be read continuously by the ADC. F_1, F_3
- This sensor is low cost. NF_1, NF_4

5 Control Systems

We considered 3 types of output devices: Air pump to oxygenate the water, Water pump to control the water levels, and the pH adjuster to control the pH levels

5.1 Air Pump

Introducing oxygen to a system can be done by creating movement in the water. This is normally done by having an air pump create movement within the tank. The problem arises due to the scope of our project. The air pump needs to work for systems of up to 3000 gallons of water. There is too much variability in the size of the system and introducing an air pump complicates this control system greatly. We have decided not to include this within the scope of our project.

5.2 Water Pump

We considered having a simple control system that would automatically keep the desired water level with the use of water pumps. Once we focused on the scope of our project, a water pump is not ideal being controlled through the RaspberryPi due to 2 reasons. The first reason is that the scope of our project is to design for systems up to 3000 gallons of water. Installing a water pump system suitable for a RaspberryPi would be unreasonable due to the relatively small pumps available through the RaspberryPi. The second reason is if we were to install a reasonably size pump, this would take far too much power for our entire system to work in parallel with this pump.

5.3 pH Adjuster

There are two ways to change and maintain the pH levels of the system: with a liquid solution or a solid object usually peat moss or driftwood. We have chosen to use the liquid solution and have decided to use a Peristaltic Pump ($12) to control this liquid solution due to the following reasons:

- The solids can produce discolouration in the water. This will not only make physical observation harder, a feature we might implement is a waste detecting system, and discolouration affects that. The liquid solution will not interfere with the appearance of the system and provide accurate values in real time. (F_4)
- The liquid solution is a lot easier to measure. The volume of a liquid is much easier to measure and transport compared to something solid. This results in a much easier installation. (NF_2, NF_5, NF_7)
- The Peristaltic Pump is cheaper and easier to design compared to something that needs to transport a solid. This leads to an affordable system (NF_1, NF_2)
- There are many types of liquid solutions that can be used while the solids needed are very specific and usually hard to find. This allows users to customize more easily compared to the solids. (NF_5, NF_7)

The peristaltic pump will pump a basic solution into the aquaponic or hydroponic system to raise its pH, because in most aquaponic system, the bacteria acidifies the water. We recommend that the solution to be pumped into the system is potassium hydroxide and calcium hydroxide in an alternating manner to provide some micronutrients as well as pH buffering.

6 Software Application

To implement the software application to display the aquaponic's data to the user, we considered various aspects, like the technology we were going to use, how we were going to implement said technology, and the trade-offs we made for each decision.

6.1 Technology

The two main technologies we considered to implement the software application were web and mobile. We ultimately choose to use a mobile based approach to implement our software application than a web based approach. The main factor that that determined this decision was ease of use. In a mobile based approach users would be able to quickly access the application, as user preferences are saved and tied in with the mobile application. This means that the user doesn't have to re-authenticate time after time when trying to access their aquaponics data. Another big factor we considered when choosing a software approach was accessibility. Almost everyone has their mobile devices on them at any given time during the day, which would provide greater accessibility to the application. A final consideration that we explored was push notifications. With a mobile based approach, we are able to send push notifications to users when extremely urgent matter arises, whereas in a web based approach this isn't even an option.

6.2 Implementation

We considered many different options we could take to create a mobile application. We debated on whether or not we would want to create a native application or build one with a cross platform language. Given the time and nature of this project, we ultimately decided on a cross platform approach. With a cross platform approach, we would be able to develop for both Android and IOS with half the effort and half the code base. One of the draw backs however is that we might lose niche native functionality specific to either Android or IOS. Since our software application is simple, we believe that the niche native service loss wouldn't be detrimental to the project. We choose Flutter as the our cross-platform language to implement this mobile application over other cross platform languages such as React Nastive, or Xamarin since Flutter allows for quick implementation of native feeling applications.

6.3 Use Case Overview

Upon entering the application, the application checks if the user has previously been authenticated. Here users can authenticate themselves by either signing up or signing in. Upon successful sign in or creation of an account, the user would be taken to the application's home screen. The ability to authenticate users to access to their account and aquaponioc system meets our non-functional requirement NF_2.

The home screen displays a summary of one of the existing systems that the user has registered with the application. The summary includes the name of the system, the status of the system, last time the information was updated, the warnings for the system, and a list of all the sensors and it's measured values. The summary of the aquaponic system on the home screen meets our function requirement F_1. From the home screen, the user is able to navigate to the sensor chart screen, the aquaponic system settings screen, and the user setting screen.

The sensor chart screen displays historic measurements made by the sensors on a chart. The user is able to select which sensor to view and the date range in which the measurements were taken. The ability to view historic sensor measurement within a given date range meets our function requirement F_2.

The system settings screen allows the user to set the range of acceptable values for the aquaponic sensors and well as calibrate the pH sensor. The user can also change the sensor measurement interval on this screen. The ability to adjust and calibrate sensors meets our functional requirement F_3.

The user setting screen is where the user is able to logout or delete their account if they so choose. In the user setting screen, the user is also able to add additional aquaponic systems to their account provided they know the aquaponic systems identification. The ability to add and monitor additional aquaponic systems meets our non-functional requirement NF_5.

If the user is already logged in the application, the user may receive notifications if one or more of their aquaponic system is in a critical state. Upon tapping the application, the user will be brought to the home page with the summary of the system in the critical state. The ability to send alerts to growers when a aquaponic system enters a critical state fulfills our function requirement F_3.

An overview of how the user can interact with the software application is highlighted and illustrated in the use case diagram below in figure 9.

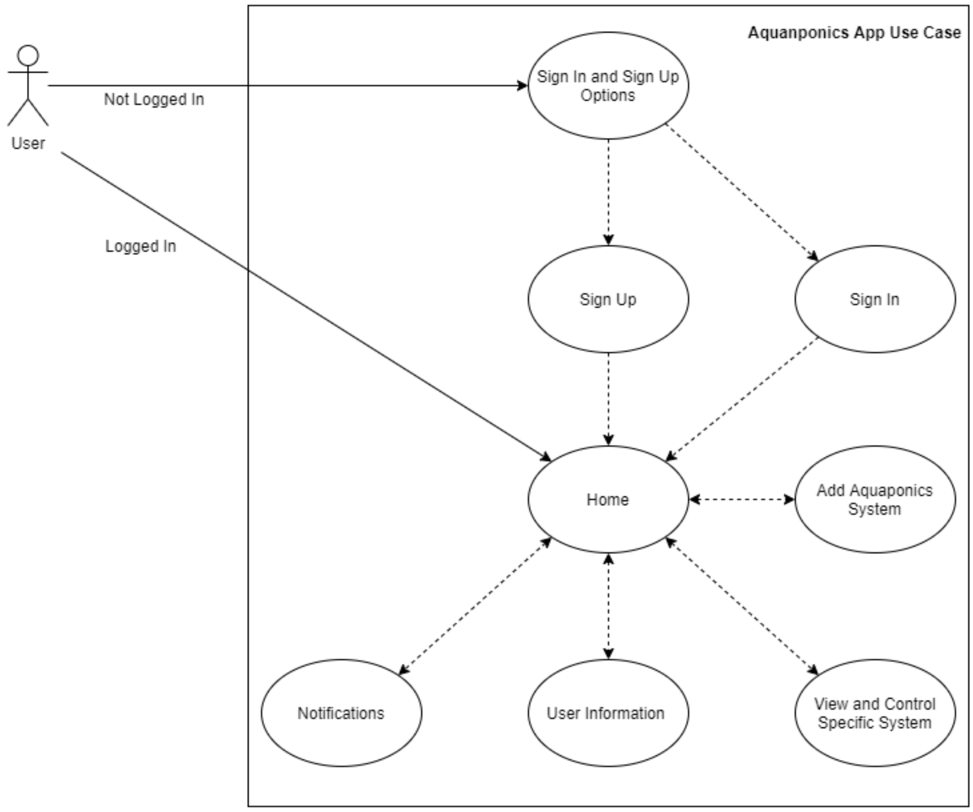


Figure 11: Use Case Diagram

6.4 Screens

Below are the different screens in the software application and an explanation of their intended user interactions as well as how they interact with the overall IOT infrastructure.

6.4.1 Sign In and Sign Up Options Screen

The sign in and sign up options screen will be presented to users who aren't previously authenticated on the application. The sign in and sign up options screen will give the user the option to either Sign Up an account or login to an existing account. The user can Sign Up with the application by providing an email and password or by Signing in using a valid Google account. An image of the application's sign in and sign up options screen can be seen below in figure 12.

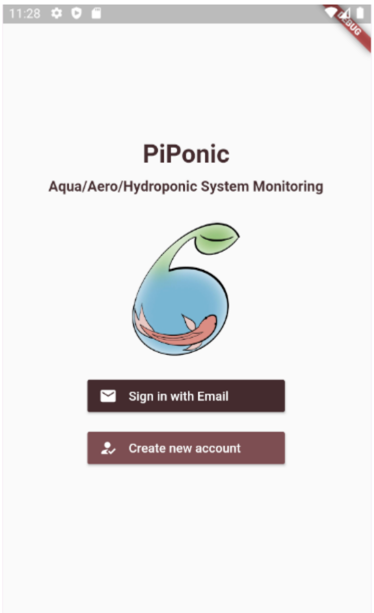


Figure 12: Sign In and Sign Up Options Screen

6.4.2 Sign In Screen

The sign in screen is presented when the user taps on the "Sign in using Email" option in the sign in and sign up options screen. This page contains two input fields for the email address and password. Unsuccessful authentication submissions won't let the user into the application whereas successful authentication submission will direct users to the application's home screen. Each login attempt will trigger a request to Firebase to confirm the users credentials and to get the user's unique User ID. The email input box verifies that it's a valid email address by checking if the '@' character is present within the email submission. Similarly password input box verifies that the password is sufficient by the password length is at least 6 characters. An image of the sign in screen can be seen below in figure 13.

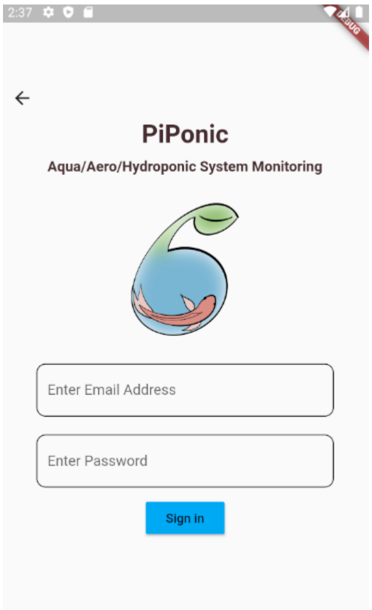


Figure 13: Sign In Screen
-b

6.4.3 Sign Up Screen

The sign up screen is presented when the user taps on the "Sign up with Email" option in the sign in and sign up options screen. This page contains two input fields for the email address and password. Unsuccessful sign up submissions won't let the user into the application whereas successful sign up submission will direct users to the application's home screen. Each login attempt will trigger a request to Firebase to create the user account by using the submitted user credentials and to get the user's unique User ID. The email input box verifies that it's a valid email address by checking if the '@' character is present within the email submission. Similarly password input box verifies that the password is sufficient by the password length is at least 6 characters. An image of the sign up screen can be seen below in figure 14.

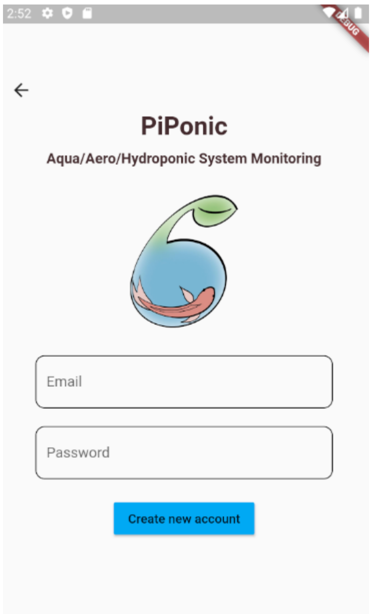


Figure 14: Sign Up Screen

6.4.4 Home Screen

The home screen is presented when the user successfully authenticates with the application. If there are existing aquaponic system(s) associated with the users account, the home screen will show a summary of one of the existing systems. The home page would show a summary of the chosen system including the name of the system, the status of the system, last time the information was updated, the warnings for the system, and a list of all the sensors and it's measured values. The values are of the aquaponic system is directly tied to the Firebase backend; this means that the values on the home page will update in real time whenever a there is a new reading in the Firebase backed. Am image of the home page can be seen below in figure 16.

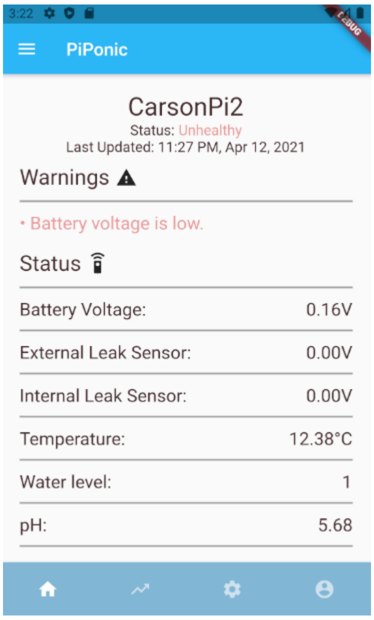


Figure 15: PiPonic home screen

6.4.5 Aquaponic/Hydroponic List

Tapping on the hamburger menu button on the top left of the application, the list of the user's aquaponic systems will slide out. Here the user is able to switch between the different aquaponic systems associated with their account. The list of the aquaponic systems is directly tied to the Firebase backend; this means that the list of aquaponic systems will change in real time whenever a there is a change to the users access to different aquapnic systems.An image with the location of the hamburger menu button and the list of the user's aquaponic systems is shown below in figure 14.

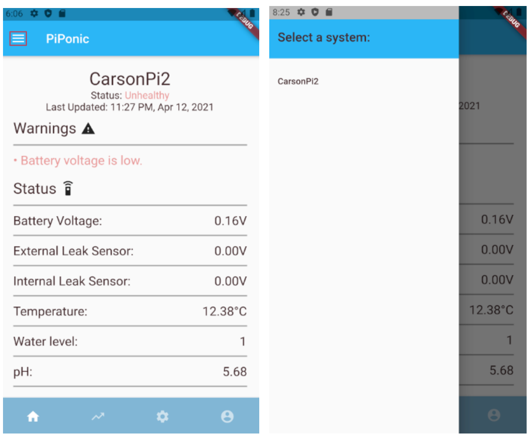


Figure 16: Location of the hamburger menu button and the list of the user's aquaponic/hydroponic systems

6.4.6 Chart Screen

Tapping on the rising trend icon in the bottom navigation bar brings the user to the chart screen where the user is able to see a chart of their sensor values and the time at which they were measured. Here the user is able to change the different sensors that they want to see and are also able to change the range of time for which the measurements were taken. New measurements are queried from the Firebase backend and the graph is changed whenever a user changes a sensor to view or changes a time range. A screenshot of the charts screen can be seen below in figure 15.

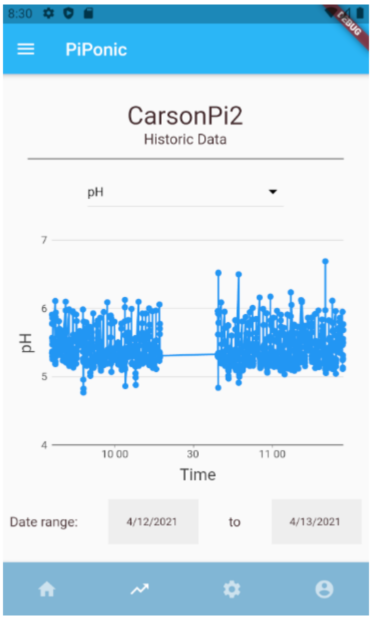


Figure 17: PiPonic Chart Screen

6.4.7 System Settings Screen

Tapping on the gear icon in the bottom navigation bar brings the user to the aquaponic system setting page. Here the user can set the range of acceptable values for the aquaponic sensors and well as calibrate the pH sensor. The user can also change the sensor measurement interval in the "General" tab. When the user enters a new acceptable range or calibrates the pH sensor or changes the sensor measurement interval, the application will update the backend which would then update the system of the new changes. A screenshot of the aquaponic settings screen can be seen below in figure 16.

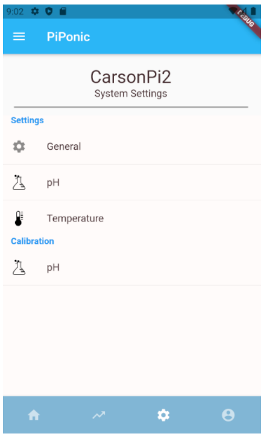


Figure 18: PiPonic system settings screen

6.5 User Settings Screen

Tapping on the user icon in the bottom navigation bar brings the user to the user settings page. Here the user can add additional aquaponic systems to their account to monitor and control. When the user adds additional aquaponic systems to their account, the application would then notify the backend which would then associate the new aquaponic system with the users identification. The application would then automatically add the new aquaponic system in the users application. The user can also logout of their account and delete their account. A screenshot of the user settings screen can be seen below in figure 17.

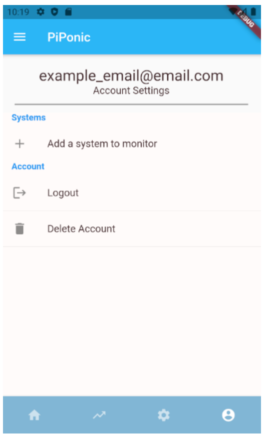


Figure 19: PiPonic user settings screen

6.5.1 User Information Screen

The notification screen is presented when the user clicks on the user icon in the bottom navigation bar. The user screen shows the user information and allows for the user to log out of the application or delete their account.

6.5.2 Add Aquaponics Screen

The add aquaponics screen is presented when the user clicks the profile icon in the bottom navigation bar. The add aquaponics screen contains input fields that pertains to the aquqponic system they are trying to add. Upon successful submission of results, the aquaponic system will be created in the Firebase back-end and linked with the users User ID.

6.5.3 View and Control Specific System Screen

The view and control specific system screen is presented when the user taps on a specific aquaponic system in the home screen. This screen contains real time accurate data of the different measurements taken by the sensors of the aquaponic system as well as buttons to request value readings. The color of the buttons represents whether or not the specific data is within an acceptable range; green means that the data is within an acceptable range and read means that the data is within an unacceptable range. This screen also contain buttons to control the different actuators that affect the system such as the water pump. An image of the view and control specific system screen can be seen below in figure 12.

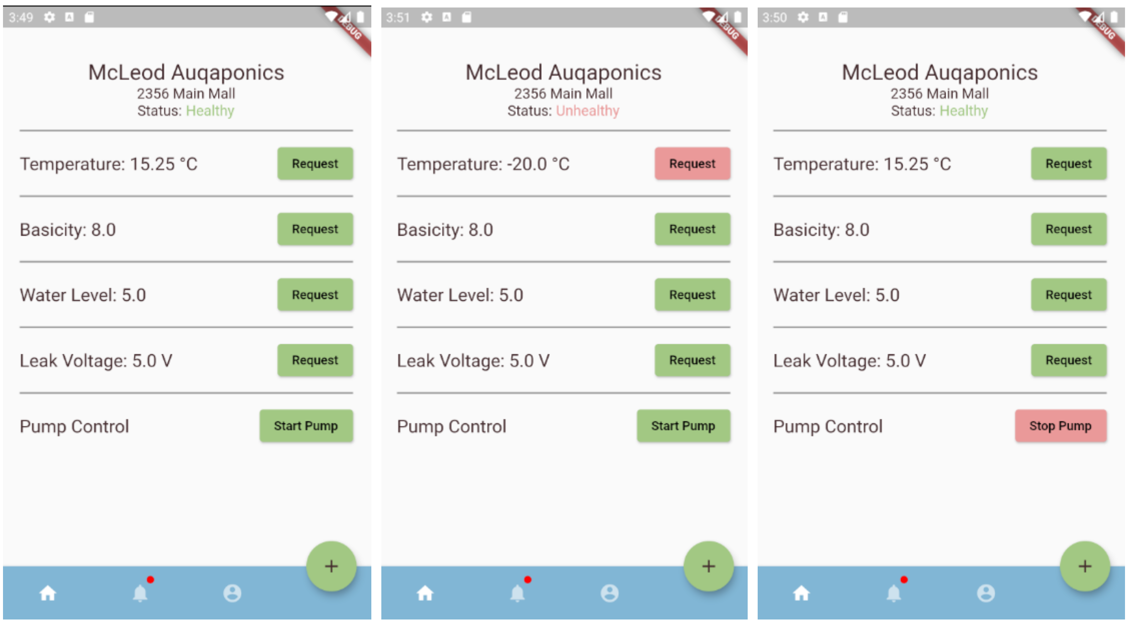


Figure 20: View and Control Specific System Screen

7 Circuit Integration - PCB

In order to maximize simplicity and repeatability of construction, robustness, and lower production costs, a custom printed circuit board was designed. This element integrates several circuit elements, including the power supply architecture, reading from sensors, sending control signals to the relay module, and interfacing with the Raspberry Pi. The PCB design can be seen in figure 21.

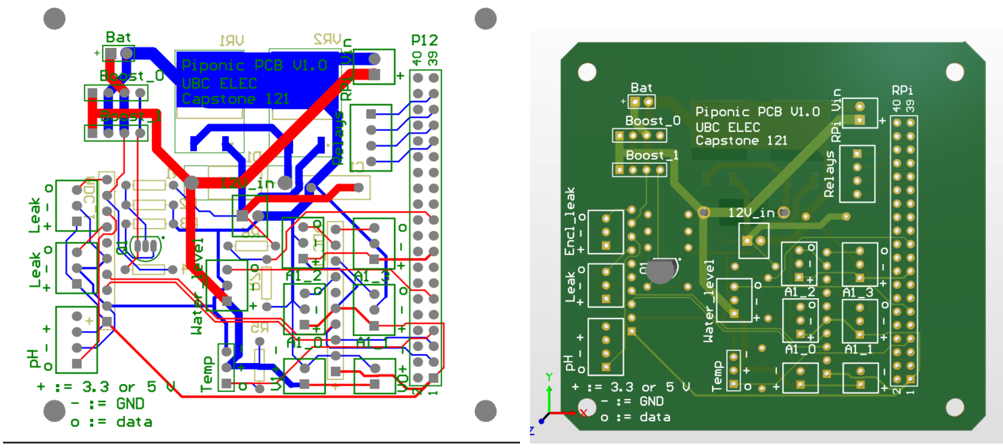


Figure 21: Custom PCB Layout

The integration with the raspberry pi is accomplished by having this PCB take the format of a 'hat'. This means that the board will directly mount onto the Raspberry Pi's male header pins, so that no mistakes can by made by the user in wiring.

All sensors will integrate with the PCB through simple board-mounted screw-down terminal blocks. Although this adds to the size of the final design, it is the simplest for users.

7.1 Component Choices

The components for this PCB were chosen to make manufacturing without a solder-reflow station easy - so almost all of the components are through-hole. The only exception to this are the voltage regulators, which are chosen as SMT components for the built-in heat-dissipation, and for the low profile allowing them to be mounted to the underside of PCB, which will allow for the PCB to have a much smaller footprint.

7.2 Future Development Considerations

This PCB was designed with future development in mind. It features 8 total ADC channels, 4 of which are unused. These ADC channels can be used for additional sensors. It has 2 unused relay-channels for additional control devices. All of the Raspberry Pi header pins are still accessible with this design, so that any type of device or communication channel may be added through jumper wires by future users.

8 Assembly

In consideration of the construction of the system, the NF_4, NF_1 requirements were chiefly considered. That is, that the design is waterproof and low-cost.

All components were roughly modelled in Solidworks 2019 to gauge how large the enclosure needed to be. Models of all necessary components can be seen in figure 22 and 23.

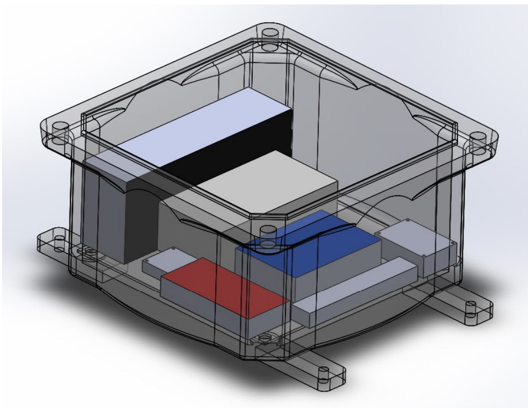


Figure 22: Enclosure elements inside enclosure: Power Supply, Breadboard, Relay Block, Terminal Block, Raspberry Pi, Water-level board, pH sensor board.

There are 5 independent wire ports that must penetrate the enclosure in a waterproof manner. To accomplish this, waterproof nylon cable glands were used. An example of one of these can be seen in figure 24. These devices have a rubber seal which is compressed around a wire inside it, forming a seal.

For ease of maintenance and durability, all of the electronic components are mounted to a board that are mounted to the corners of the main enclosure with double-sided tape. This board is constructed from 1/2” plywood.

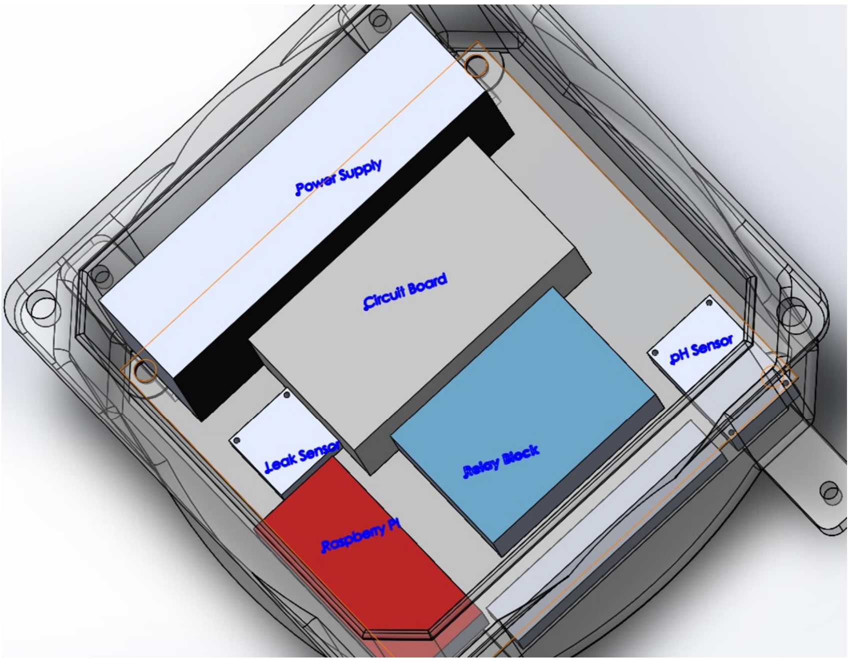


Figure 23: Labelled Enclosure elements: Power Supply, Breadboard, Relay Block, Terminal Block, Raspberry Pi, Water-level board, pH sensor board.

Figure 24: Waterproof Nylon Cable Gland example.

9 Bill of Materials

In order to reconstruct this project, one will need to purchase many components. The sources for these are outlined in this section and in the Installation Instructions document.

9.1 Hardware

The hardware BOM can be seen in the table below.

Designator	Description	Quantity	Link
Enclosure	8x8x4' junction box	1	https://www.homedepot.ca/product/carlon-thermoplastic-junction-box-8x8x4-in/1000403713
Heatshrink	Variety Pack	1	https://leeselectronic.com/en/product/15370-kit-heatshrink-variety-pack-170pcs.html
Cable Gland	7mm	3	https://leeselectronic.com/en/product/61270-fitting-water-proof-pg-7-light-grey-csa-bi html
Cable Gland	9mm	6	https://leeselectronic.com/en/product/6125-fitting-water-proof-pg-9-black.html
Screws/Bolts	4x1/2", 6x1/2"	Box	

Table 6: Hardware BOM required for building project

9.2 Electronics

9.2.1 Sensor Devices

Designator	Description	Quantity	Link
Temperature	DS18B20	1	https://www.digikey.ca/en/products/detail/dfrobot/DFR0198/7597054
Water Level	SEN0204	1	https://www.digikey.ca/en/products/detail/dfrobot/SEN0204/6579443?s=N4IgTCBcDaIMoFEByAGMKAsIC6BfIA
pH Sensor	SEN0161-V2	1	https://www.digikey.ca/en/products/detail/dfrobot/SEN0161-V2/9608223
Leak Sensor	Analog Leak Sensor	1	https://www.amazon.ca/gp/product/B07PV9SYLV/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1

Table 7: Sensor Bill of Materials

9.2.2 Control Devices

Some devices will be necessary to set up the control system aspect. This includes what is seen below in the table BOM. The design currently has 2 unused relay channels that may be used in future expansion.

Designator	Description	Quantity	Link

Relay Module	4 Channel 5V controlled Relay Module	1	https://leeselectronic.com/en/product/31302-4-relay-digital-module-5v.html
Solenoid Valve	12V 3/4" solenoid for connecting with hoses	1	https://leeselectronic.com/en/product/440-44012VSOLENOIDVALVE34SMALLROB104.html
Peristaltic Pump	12V Peristaltic Pump with surgical tubing	1-2	https://www.amazon.ca/Peristaltic-Liquid-Dosing-Silicone-Tubing/dp/B075VN1QZM/ref=asc_df_B075VN1QZM/?tag=googleshopc0c-20&linkCode=df0&hvadid=292901727312&hvpos=&hvnetw=g&hvrand=16446818779147696161&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9001561&hvtargid=pla-376008502070&psc=1
Power Supply	12V 2A Power Supply	1	https://www.amazon.ca/Adapter-Supply-Switching-100-240V-Transformer/dp/B0711SZCMQ/ref=sxin_9_ac_d_rm?ac_md=2-2-MTJ2IDJhIHBvd2VyIHN1cHBseQ%3D%3D-ac_d_rm&cv_ct_cx=12V%2Bpower%2Bsupply&dchild=1&keywords=12V%2Bpower%2Bsupply&pd_rd_i=B0711SZCMQ&pd_rd_r=1eff5954-32d8-4e80-8994-f3f9e1651a94&pd_rd_w=vjLDB&pd_rd_wg=riHjN&pf_rd_p=cf70c38a-5522-42af-aeb2-119882942fe1&pf_rd_r=Z0P28K1ZC0Q2NJ9QT85Q&qid=1618245824&sr=1-3-12d4272d-8adb-4121-8624-135149aa9081&th=1 or https://www.amazon.ca/ALITOVE-Universal-Regulated-Switching-Transforme dp/B078RY6YY3/ref=sr_1_15?dchild=1&keywords=12V+power+supply&qid=1618245824&sr=8-15

Table 8: Control Devices Bill of Materials

9.2.3 Electronic Parts for PCB

Designator	Description	Quantity	Link

12V_in	Power 12V in	1	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN02A150/1588862
A1_0, A1_1, A1_2, A1_3	Header, 3-pin	4	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN03A150/1588863
ADC_0, ADC_1	Analog to digital converter	2	https://www.digikey.ca/en/products/detail/adafruit-industries-llc/1085/5761229
Bat	2AA Battery Holder	1	https://www.digikey.ca/en/products/detail/hammond-manufacturing/BH2AAW/3869832
Boost_0, Boost_1	5V Boost converter	2	https://www.pololu.com/product/2562
C1 (4.7uF)	Capacitor	1	https://www.digikey.ca/en/products/detail/tdk-corporation/FA20X7S2A475KRU00/7384281
D1	Power Schottky rectifier, 60 V, 5 A	1	https://www.digikey.ca/en/products/detail/smc-diode-solutions/95SQ015/8021507
Leak Sensor Header	Header, 3-Pin	2	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN03A150/1588863
Header 20X2	GPIO shield for RPi	1	https://www.digikey.ca/en/products/detail/adafruit-industries-llc/2223/5629433
pH Sensor Header	Header, 4-Pin	1	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN04A150/1588864
Q1	2N3904 NPN General Purpose Amplifier	1	https://www.digikey.ca/en/products/detail/stmicroelectronics/2N3904/603420
R1(10k), R2(1k), R3(220), R4(900k), R5(4.7k), R6(10k), R7(3.9k)	Resistor	6	
Relay header	Relay Control	1	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN04A150/1588864

RPi_Vin	Header, 2-pin	1	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN02A150/1588862
Temp Sensor header	Header, 3-Pin	1	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN03A150/1588863
V1, V2	Header, 2-Pin	2	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN02A150/1588862
VR1, VR2	Voltage Regulator	2	https://www.digikey.ca/en/products/detail/texas-instruments/LM7805SX-NOPB/6110585
WL Sensor Header	Header, 3-Pin	1	https://www.digikey.ca/en/products/detail/on-shore-technology-inc/OSTVN03A150/1588863

Table 9: PCB Bill of Materials

References

[1] J. B. Jones, "Hydroponics: a practical guide for the soilless grower." Boca Raton: CRC Press, 2005.

[2] W. Lennard and S. Goddek, "Aquaponics: The Basics," Aquaponics Food Production Systems, pp. 113–143, 2019.

[3] Raspberry Pi Zero W. `https://www.raspberrypi.org/products/raspberry-pi-zero-w/?resellerType=home`

[4] "HTTP vs. MQTT: A tale of two IoT protocols," Google. [Online]. Available: https://cloud.google.com/blog/products/iot-devices/http-vs-mqtt-a-tale-of-two-iot-protocols. [Accessed: 13-Apr-2021].

[5] "Choose a Database: Cloud Firestore or Realtime Database — Firebase," Google. [Online]. Available: https://firebase.google.com/docs/database/rtdb-vs-firestore. [Accessed: 13-Apr-2021].

[6] "Get realtime updates with Cloud Firestore — Firebase," Google. [Online]. Available: https://firebase.google.com/docs/firestore/query-data/listen. [Accessed: 13-Apr-2021].

[7] David Sayson. Aquaponics Expert. Personal Communications. [Date: 25-Sep-2020]

THE UNIVERSITY OF BRITISH COLUMBIA

IoT Monitoring of Aquaponic and Hydroponic Food Production

Appendix C: Verification and Validation

UBC Electrical and Computer Engineering Capstone 121

Carson Berry
Lynes Chan
Mason Duan
Jayden Leong
Hannah Xu

for

UBC SEEDS Sustainability Program
Bernhard Nimmervoll, UBC Mechanical Engineering

April 27, 2021

Contents

Revision History

Date	Author	Sections	Change
2020-10-05	JL	1.0 - 4.0	Creation of document skeleton
2020-10-16	MD	1.0 - 4.0	First draft
2020-11-28	CB	2.0 - 4.0	Added subsystem specifics test outlines
2021-01-15	CB	3.0	Restructuring document to reflect design and requirements
2021-02-19	CL	1.0 - 6.0	Added section 6.0 Validation and minor changes to earlier sections to be consistent with the requirements documentation
2021-02-19	JL	2.0 - 3.0	Reworded overview section to be more concise. Updated testing section to be more concise, and match updated functional requirements. Removed some outdated tests due to changed requirements.
2021-04-11	JL	1.0 - 3.0	Major revisions to all sections for Milestone 4.

1 Overview

This document outlines the verification and validation plans for our project, an IoT hydroponic or aquaponic monitoring system. The verification tests and results included in this document seek to prove that our project meets its requirements. For more information about these requirements, please see the included *Requirements* document.

The design of our project informs the structure of the verification testing and results below. Our hydroponic and aquaponic monitoring system contains three main subsystems seen in Figure 1: a sensing and control system, cloud infrastructure, and a mobile software application. As such, our verification tests and results will be organized under each of these subsystems.

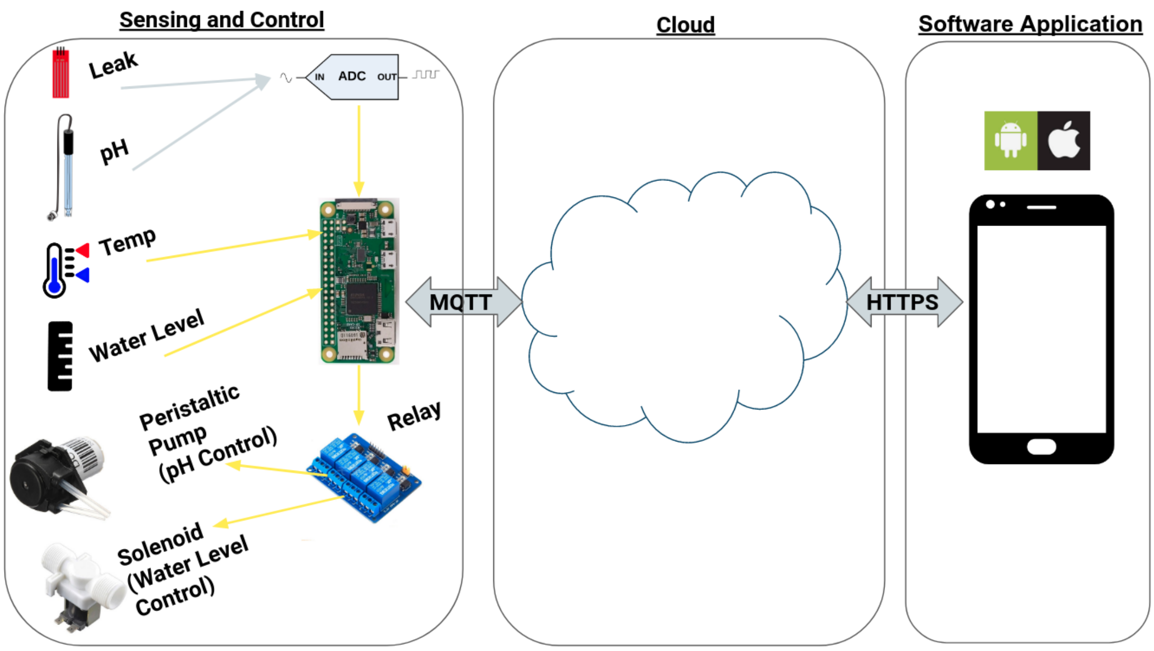


Figure 1: System Level Block Diagram

Additionally, this document outlines our project's validation plan, to ensure the project meets our client and user's needs. To see our validation tests and results, please see section 3.

Finally, please note that this document assumes that the reader is familiar with the setup of our design described in *System Installation* document. As well, we assume that the reader is familiar with the design described in the *Design* document. If this is not the case, we strongly recommend reading those documents before reading the tests described here.

2 Verification

This section provides verification tests and results for the following main subsystems in our design:

- Sensing and Control System
- Cloud Infrastructure
- Software Application

2.1 Sensing and Control System Verification

The sensing and control system has two functions. First, it measures important system variables including water pH, level, temperature, and leakage. It then sends these measurements to the Cloud Infrastructure so users can monitor their aquaponic or hydroponic system. Second, the sensing and control system automatically controls pH and water level to ensure user's aquaponics or hydroponics systems remain healthy. Please consult Figure 2 below and the *Design* document for more details about the sensing and control system.

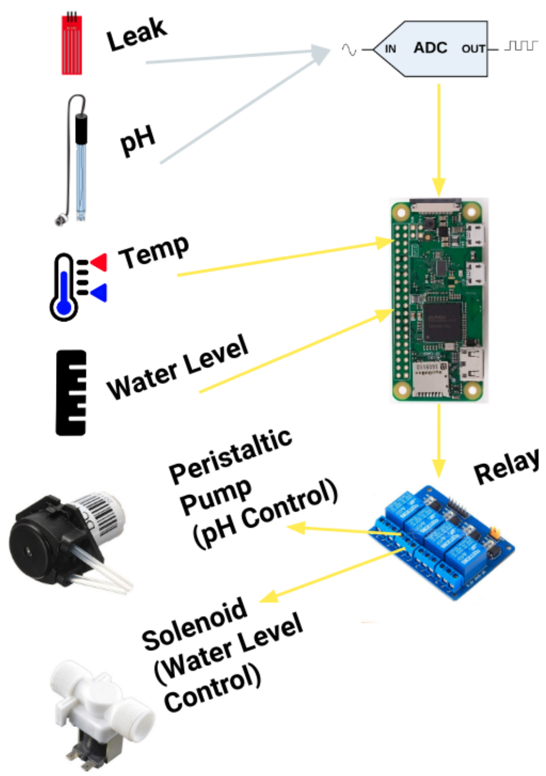


Figure 2: Overview of the sensing and control subsystem.

For an overview of all the verification tests performed on the sensing and control subsystem, please consult Table 1 below. Each of these tests in described in further detail in the proceeding subsections.

Table 1: Sensing and Control Verification Test Overview.

Test	**Purpose**	**Requirement(s)**	**Result**
SCT_1	Sensors measurements can be made and send to the Cloud Infrastructure	F_2	PASSED
SCT_2	pH and water level are maintained in healthy ranges automatically	F_4	PASSED
SCT_3	Operates on backup power	F_5	PASSED
SCT_4	The sensing and control system is water resistant	NF_4	PASSED

SCT_1: Sensors measurements can be made and send to the Cloud Infrastructure

Purpose

Be able to send sensor measurements to users. Also, to test whether sensors can be read.

Procedure

1. Run the sensing and control system
2. Follow the testing procedure in CT_1 to ensure messages are recieved by the cloud correctly. Otherwise, the test FAILED.

Result

The test **PASSED**.

SCT_2: pH and water level are maintained in healthy ranges automatically

Purpose

To make sure that pH and water level are controlled to be within healthy levels.

Procedure

1. Place the pH probe in an acidic solution.

2. Place the one end of the peristaltic pump in a basic solution and the other in the same solution as the pH probe.
3. Run the sensing and control system, and ensure that the pH stabilizes at the target pH set in the mobile application.
4. Place the water level sensor outside of water.
5. Ensure that the solenoid turns on periodically to simulate adding more water to the hydro/aquaponics system.

Result

The test **PASSED**.

SCT_3: Operates on backup power

Purpose

To ensure that our system can remain operational in the case of a power failure.

Procedure

1. Connect 2AA batteries into the 5V boost converter board.
2. Verify that the boost converter is able to output 5V DC with a multimeter. If not, test FAILED.
3. Simulate the 5V boost converter shutdown circuit's functionality, as seen in Figure 3.

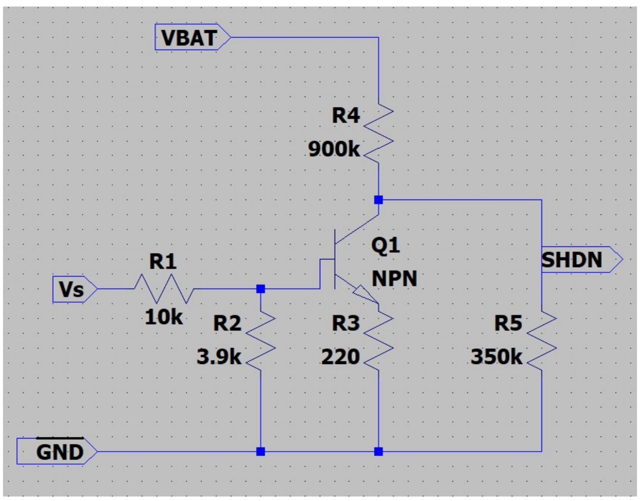


Figure 3: Boost converter shutdown circuit. This circuit disables the AA batteries when 12V is present.

4. Ensure that the circuit simulation outputs a low voltage when the 12V power supply is present and a ‘high’ voltage (above 0.8V) when the 12V power supply is absent. A plot of this behaviour can be seen in figure 4. If this behavior is not seen, the test FAILED.

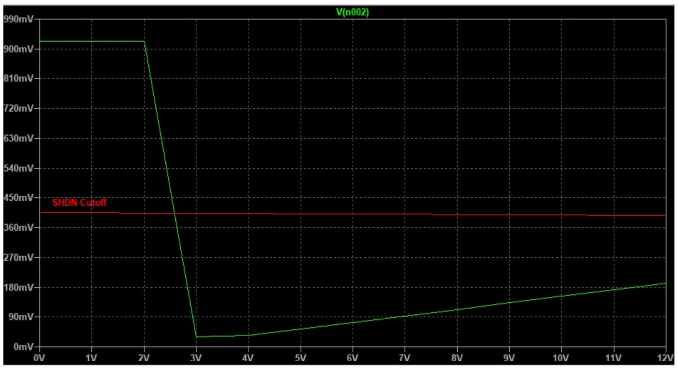


Figure 4: 5V boost converter shutdown circuit output vs source voltage. We see that once the source voltage (normally 12V) drops below 3 V, the SHDN signal goes high.

5. Construct the physical circuit.
6. Attach a multimeter voltage probe to the 5V rail and GND.
7. Plug in the 12 V power source
8. Connect 2AA batteries to the 5V boost converter.
9. Ensure that when 12V is present, the voltage rail read 4.7V. Otherwise, the test FAILED.
10. Ensure that when the 12V power supply is disconnected, or unplugged, the voltage rail should now read 5V. Otherwise, the test FAILED.

Result

The test **PASSED**.

SCT_4: The sensing and control system is water resistant

Purpose

The purpose of this is to ensure that the sensing and control system is water resistant. This is important for aqua/hydroponics systems that are around water, or are perhaps placed outside.

Our system must comply with a rating of IP64. This means that the enclosure must be protected from low pressure water jets in any direction.

Procedure

1. Spray the enclosure with a shower head for 15 minutes from all directions
2. Ensure not liquid was able to enter the enclosure. If not, the test FAILED

Result

The test **PASSED**.

2.2 Cloud Infrastructure Verification

The cloud infrastructure connects the software application with the sensing and control system. This means that it provides the following functionality:

- Passing data beztween the software application and the sensing and control system.
- Sending notifications to the software application if there are issues detected by the sensing and control system.
- Saving sensor measurements in a database.

For an overview of the Cloud Infrastructure, please see Figure 5 below:

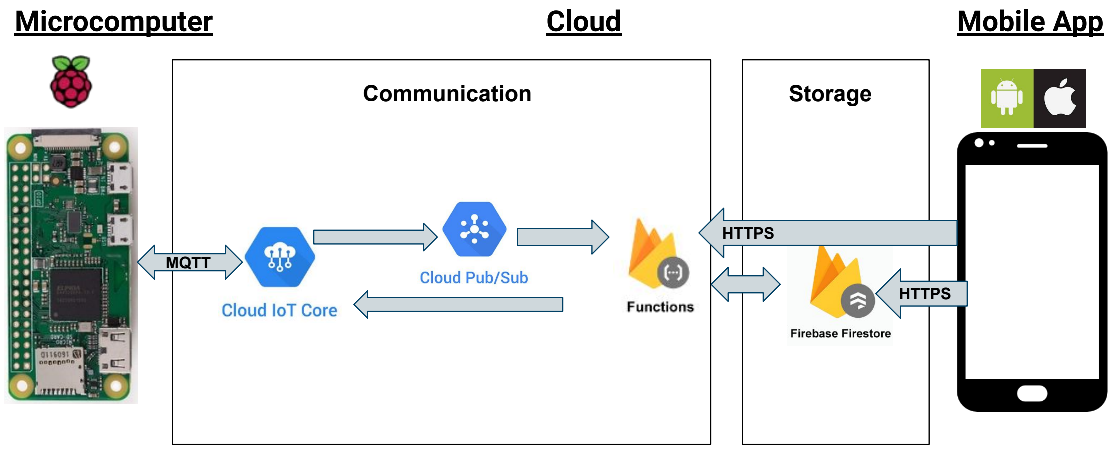


Figure 5: Overview of the cloud infrastructure.

For an overview of the tests performed on the cloud infrastructure system, please see Table 2 below.

Table 2: Cloud Infrastructure Verification Test Overview.

Test	Purpose	Requirement(s)	Result
CT_1	Messages from sensing and control system are saved in a database	F_1, F_2	PASSED
CT_2	Detects when sensor readings are outside the normal range	F_3	PASSED
CT_3	Send configuration messages to the sensing and control system	F_3	PASSED

CT_1: Messages from sensing and control system are saved in a database

Purpose

Ensures that sensor measurements from an aqua/hydroponic system are saved properly. This allows the mobile application to display the status and history of an aqua/hydroponic system.

Procedure

1. Set the sensor update interval to be 1 minute using the mobile application.
2. Run the sensing and control system so sensor measurements being made
3. Go to the Firebase web console for the project.
4. Select the Firestore Database tab.
5. Select the Status collection to display the most recent sensor readings stored in the database.
6. Verify that sensor measurement are updating every 1 minute. If not, test FAILED.
7. Select the History collection to display past measurements of an aqua/hydroponic system.
8. Verify that new sensor measurements are being written to the database every 1 minute. If not, test FAILED.
9. Verify that all sensor measurements are saved in the database. If not test FAILED.
10. Test PASSED if no failures detected above.

Result

In Firestore, updates can be seen as new data flashes as orange when recieved. This can be used to check whether messages have been updated in the Status and History collections. See Figure 6 for an example of what this looks like.

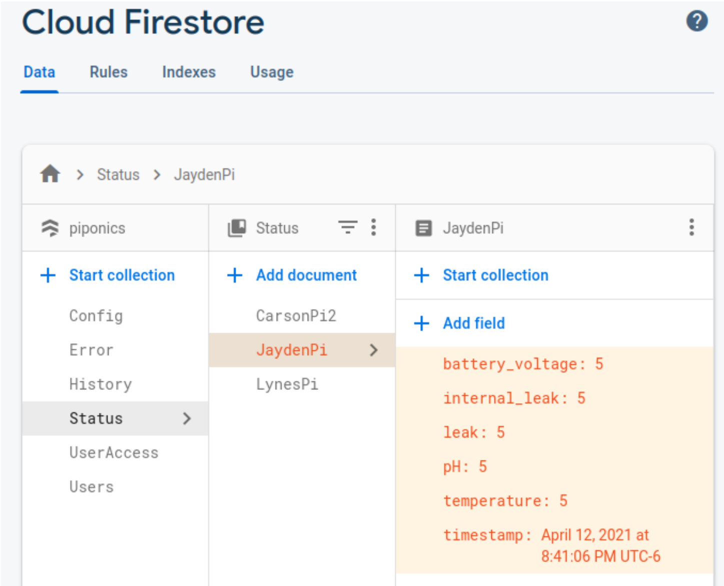


Figure 6: Firestore database updating during CT_1

After running this test, it **PASSED**.

$\mathbf{CT_2}$: Detects when sensor readings are outside the normal range

Purpose

Ensures abnormal sensor readings are detected. For example, the system should detect if there is a leak or if the temperature is too low.

Procedure

1. Go to the Firebase web console for the project.
2. Select the Firestore Database tab.
3. Select the Error collection to display whether an error is detected on a given aqua/hydroponic system.
4. Place the pH probe in an acidic solution.
5. Ensure that the PH_LOW field is true. If not, test FAILED.
6. Place the water temperature sensor in ice water. Ensure that the the TEMP_LOW field is true. If not, test FAILED.
7. Repeat the same process for all sensors to generate errors. Ensure that the appropriate errors are detected. If not, test FAILED.

8. If no failures detected above, test PASSED.

Result

Please see Figure 7 below for a demonstration of the test. As you can see, the system has detected that the water temperature is low, and there are multiple leaks.

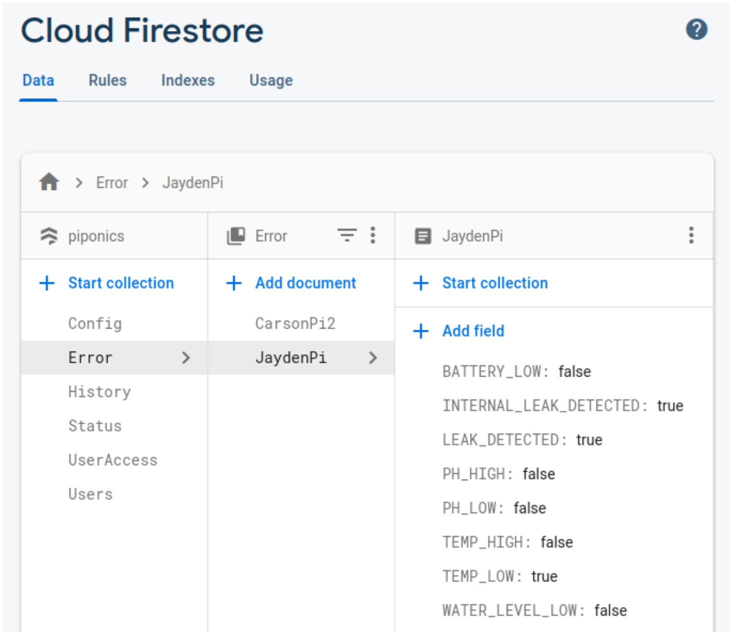


Figure 7: Firestore database Error collection updating during CT_2

All tests **PASSED**.

$\mathbf{CT_3}$: Send configuration messages to the sensing and control system

Purpose

Allows the mobile application to configure settings like the sensor sampling rate and the target pH. This test verifies this functionality.

Procedure

1. Go to the Firebase web console for the project.
2. Select the Firestore Database tab.
3. Select the Config collection to display the configuration settings of an aqua/hydroponic system.

4. Run the sensing and control system and view the logs being outputted in terminal
5. Change the values of each of the configuration values in the Firestore database.
6. Ensure the sensing and control system prints a message with the new configuration values. If values are incorrect or not recieved, test FAILED.

Result

The test **PASSED**.

2.3 Software Application Verification

The software application allows users to monitor their aquaponic or hydroponic system. This includes viewing sensor readings, adjusting system settings, and recieving notification if errors are detected. Please see Figure 8 for demo screenshots of the software application. For more details about the design of the mobile application, please see the included *Design* document.

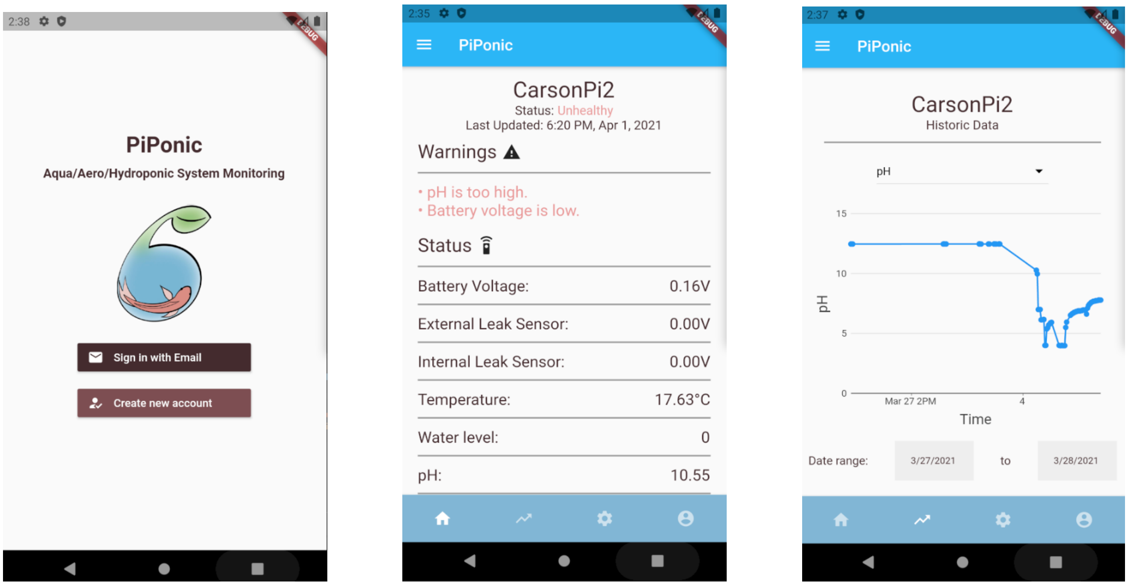


Figure 8: Screenshots demonstrating the software application.

For an overview of all the verification tests performed on the software application, please see the Table 3 below.

Table 3: Software Verification Test Overview.

Test	**Purpose**	**Requirement(s)**	**Result**
ST_1	Software application displays most recent sensor readings stored in the cloud	F_1	PASSED
ST_2	Software application displays charts of previous sensor readings	F_2	PASSED
ST_3	Software application can set normal range of aquaponics or hydroponics system variables	F_3	PASSED
ST_4	Software application recieves notifications if sensor readings are outside of a normal range.	F_3	PASSED
ST_5	Software application can be used to calibrate the pH sensor	F_6	PASSED
ST_6	Software application has an appropriate login system	NF_2	PASSED

In the following subsections, we will describe each software application test in detail.

ST_1: Software application displays most recent sensor readings stored in the cloud

Purpose

To ensure aquaponic and hydroponic growers can monitor most recent updates from their system. Therefore, helps verify requirement F_1.

Procedure

1. View the most recent data stored in the database. To do this, open the Firebase web console for the project.
2. Select the Firestore Database tab.
3. Select the Status collection to display the most recent sensor readings stored in the database.
4. Check that the values displayed on the mobile application status page match the values in the database. If not, test FAILED.
5. Check that the mobile application status page displays all of the values stored in the database. If not, test FAILED.
6. Check that the "Last Updated" timestamp is correctly updated on the app's status page. If not, test FAILED.
7. Check that all systems in the database pass the above three tests. If not, test FAILED.
8. If no failures detected above, test PASSED.

Result

See the following screenshot of the database and the mobile application status page for an example of how this test is conducted. This example demonstrates the test using the system called *CarsonPi2*:

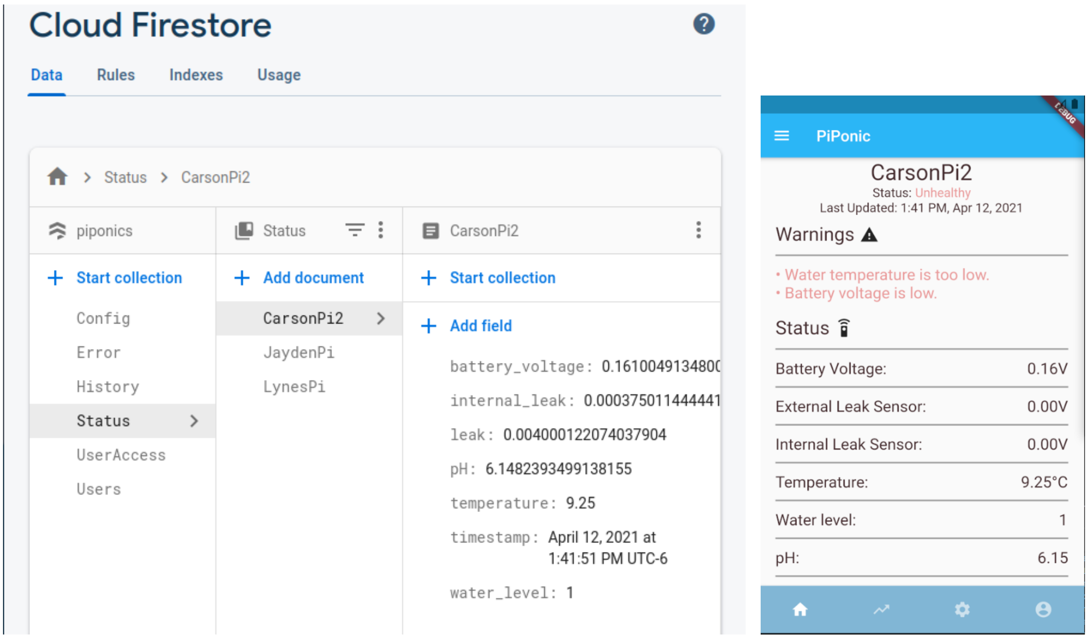


Figure 9: Screenshots taken during test ST_1

These screenshots demonstrate that the test **PASSED**.

$\mathbf{ST_2}$: Software application displays charts of previous sensor readings

Purpose

To allow aquaponic and hydroponic growers to analyze system trends over time. Therefore, helps verify requirement F_2.

Procedure

1. Turn the sensing and control system on for 24 hours, allowing the sensor measurements to be recorded.
2. Using the "Status Page" note down the sensor measurements every hour for some 8 hours during that 24 hour period.
3. Open the "Charts Page" on the mobile application by selecting the arrow icon on the bottom panel.
4. For each sensor reading, verify that the sensor values match your recorded values during the 8 hour period. If not, test FAILED.
5. If sensor measurements for each sensor is not displayed for the previous 24 hour period, test FAILED.
6. If no failures detected above, test PASSED.

Result

Below you can see an example of the test running, with the screenshot of the temperature plot in Figure 10. This test also PASSED for all other sensors.

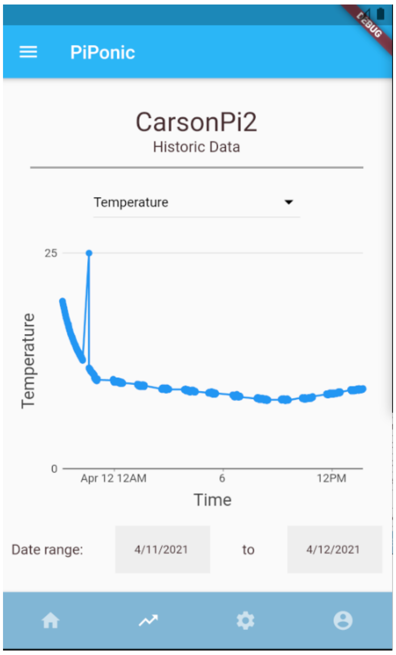


Figure 10: Software application "Charts Page" displaying temperature during test ST_2.

Note that the single spike in temperature was expected as the temperature sensor was temporarily moved to hot water. The final result is that the test **PASSED**.

ST_3: Software application can set normal range of aquaponics or hydroponics system variables

Purpose

This test ensures that users can configure their system's healthy range. This is used to detect when a system may be unhealthy and report that to a grower.

Procedure

1. Open the Firebase web console for the project.
2. Select the Firestore Database tab to see database values.
3. Select the "Config" collection to see system configuration values

4. Select the gear icon on the bottom tab of the mobile application to open system settings.

5. Click on "General Settings" and adjust the sensor measurement interval slider. Click Submit. Ensure that the *update_interval_minutes* field in the database is updated accordingly. Otherwise, test FAILED.

6. Click on pH Settings and adjust the sliders and click submit. Like the point above, ensure the database is updated accordingly. Otherwise, test FAILED.

7. Adjust a slider in the "pH Settings" page, then click cancel. Ensure that slider values return to their previous values. Otherwise, test FAILED.

8. Repeat the two tests above, only for the "Temperature Settings" page.

9. The test PASSED if no failures were detected above.

Result

See the images below for samples of this test, where database values are verified with app configuration settings:

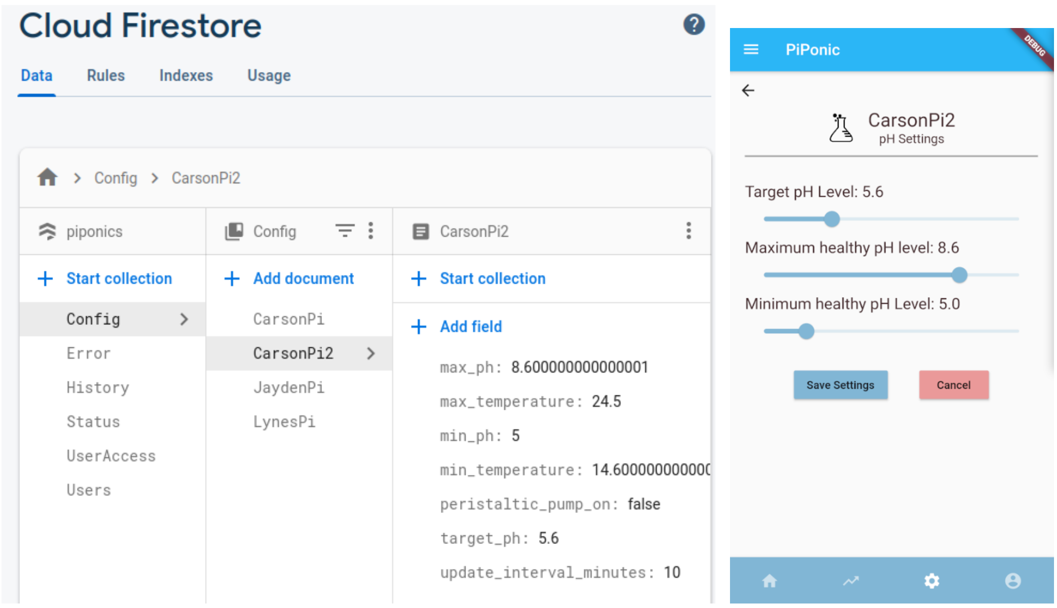


Figure 11: Screenshots taken during test ST_3

When ran, the above test **PASSED**.

ST_4: Software application recieves notifications if sensor readings are outside of a normal range.

Purpose

This test checks whether notifications/alerts are sent to users if sensor measurements from their aqua/hydroponic systems are outside of a normal range. For instance, if the pH is detected to be very acidic (i.e pH = 4), then the user should be alerted with a warning that the pH is too low.

Procedure

1. Spill water on the water leakage sensor. If no push notification recieved by the mobile application, test FAILED.
2. Set the minimum pH to 5. Place the pH probe in pH = 4 solution. If no pH too low notification recieved by the mobile application, test FAILED.
3. Remove backup batteries from the sensing and control system. If no battery low notification recieved by the mobile application, test FAILED.
4. Remove the water level sensor from water. If no water level low notification, test FAILED.
5. Set the minimum water temperature to 10 C. Place the temperature sensor in ice water. If no water temperature low notification, test FAILED.
6. If no failures detected above, test PASSED.

Result

Please see below for a sample image of notifications being received by the mobile application.

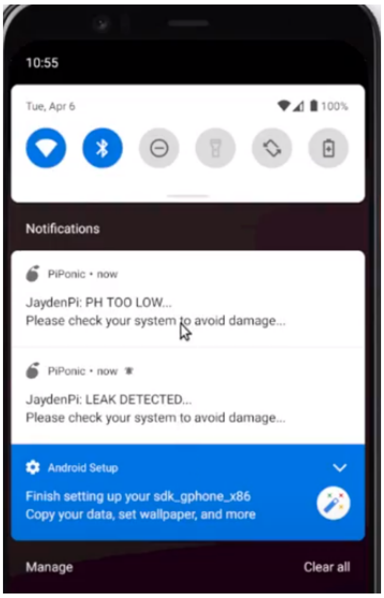


Figure 12: App notification screenshot taken during test ST_4

This test **PASSED** when ran.

ST_5: Software application can be used to calibrate the pH sensor

Purpose

This test verifies that the pH sensor can be calibrated from the mobile application.

Procedure

1. Open the mobile application and select the settings gear icon on the bottom tab
2. Select the pH calibration setting
3. Follow the instructions on the mobile application page.
4. Place the pH probe in a solution with known pH and ensure the value read on the mobile application matches.

Result

Below in Figure 13, you can see the pH calibration page. We used two solutions to calibrate the pH sensor, with pH = 4 and pH = 7.

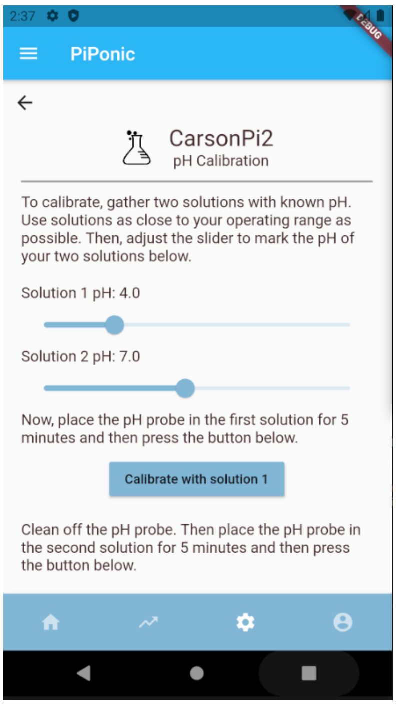


Figure 13: pH calibration page during test ST_5

Afterward, we placed the pH probe in a solution with pH = 7, and found that the pH reading was accurate. Therefore, this test **PASSED**.

ST_6: Software application has an appropriate login system

Purpose

To verify that users can create their own accounts and login in properly.

Procedure

1. Open the mobile application.
2. Try to login using a random email and password. This should not allow you to login. Otherwise, test FAILED.
3. Create a new account using an email and password.
4. Try to login to the system, but misspell your username. If the app crashes, or allows access to your account, test FAILED.
5. Login to the system using your email and password. If you are not able to login to your account using email and password, then test FAILED.
6. Test PASSED if no failures detected above.

Result

Below in Figure 14, you can see the login pages that were tested using the above procedure.

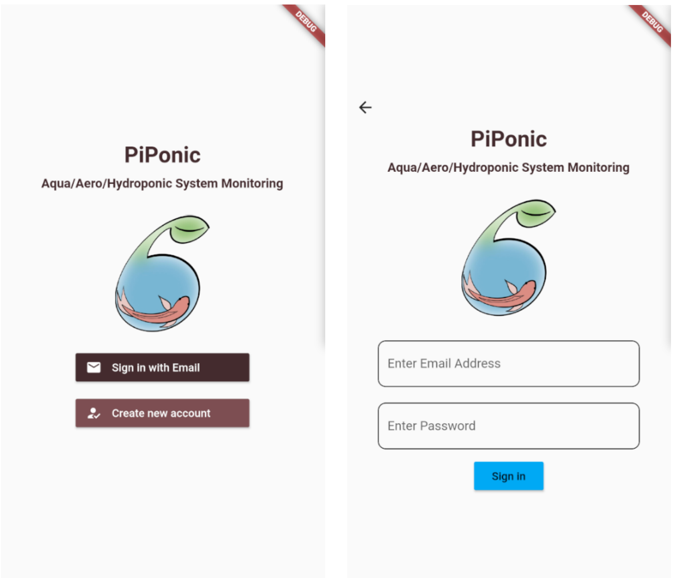


Figure 14: Login pages during test ST_6

All the tests for ST_6 **PASSED**.

3 Validation

This section states validation tests and their corresponding results for our project. The purpose of this section is to ensure that our design meets the needs of our users, who are aquaponics and hydroponic growers.

Our validation tests consist of questions asked to users, who provide feedback about the design. Our design has been used by our client over the past two months at his home hydroponics system. The feedback provided in this section will be from his comments. In future, similar questions can be asked to new users too.

Table 4 below provides an overview of our validation tests and their corresponding results.

Table 4: Validation Test Overview.

Test	**Req(s)**	**Purpose**	**Key User Feedback**
V_1	F_1	Viewing system variables	• Add configurable time periods between sensor measurements for each sensor.
V_1	F_2	Analyzing system variable trends	• Time-averaging data points in sensor measurement charts. This helps the user visualize trends more clearly.
V_3	F_3	Recieving system alerts	• Verify that notifications are recieved by phones even when the app is not running. • pH and water temperature notifications can be changed to daily notifications instead of immediate notifications.
V_4	F_4	Recalibrating sensors	PENDING
V_5	NF_3	Installation and setup procedures	• Create a short video explaining how to install the system
V_6	All	General feedback	• Adjust the pH only if multiple measurements detect that the pH is low

Please note that this section assumes the reader is familar with the design of our project as outlined in the attached *Design* document.

V_1: Viewing system variables

Purpose

System variables like water pH, level, temperature and more must be displayed to growers in a clear manner.

Results

In Table 5, we provided validation questions to help us understand whether our design meets our user's needs. Note that for each question, we asked our client to mention any difficulties he had with our system, or if he had any suggestions for improvements.

Table 5: Questions to validate user experience of viewing system variables.

	Question	User Feedback
Q_1	Were you able to create an account and login?	Yes
Q_2	Were you able to locate all sensor measurements with the app?	Yes
Q_3	Were the sensor measurements easy to interpret and understand?	Yes
Q_4	Were warning messages displayed by the app easy to interpret and understand?	Yes
Q_5	Do you have any recommendations about how we might improve how sensor readings are displayed?	Read interval should be controllable by sensor. Eg: - spill sensor needs updates frequently. Water level sensor may not show low reading for a long time - it only will go low for a short part of the fill/drain cycle.

Analysis

The key recommendation from our client is to add configurable time periods between sensor measurements for each sensor. This allows some sensors (like the water leakage sensor) to be read faster than slower changing variables (like water pH). This improvement can be made in the next iteration of the design.

V_2: Analyzing System Variable Trends

Purpose

The user should be able to effectively analyse trends within their aquaponic/hydroponic system. This means that a user should be able to analyze variables including water pH, level, battery level over time using our design. Currently, this is implemented by showing time charts of the various system variables we monitor.

Results

Please see Table 6 for the questions we asked our client and his feedback about analyzing system trends.

Table 6: Questions to validate user experience when analyzing system trends.

	Question	User Feedback
Q_1	Were you able to locate where to view system trends within the app?	Yes
Q_2	Were you able to view system trends within the timeframe that you like?	Yes
Q_3	Were charts displayed of system variables easy to understand and navigate?	Yes
Q_4	Do you have any recommendations about how we might improve how system trends are displayed?	The graphs are sufficient for short period trends. Looks like data points could be averaged out to improve the looks of the charts.

Analysis

As seen above, one area of improvement is to improve the look of the charts. Currently, many data points can appear clustered together if sensor measurements are taken at a fast rate. See Figure 15 below for an example of this phenomenon.

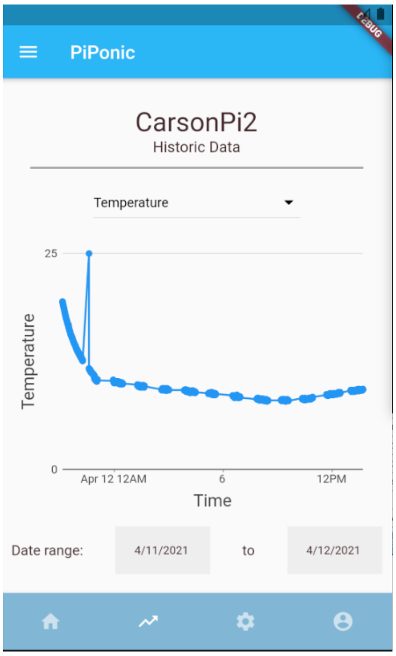


Figure 15: "Charts Page" with clustered temperature readings.

This can be improved as our client suggests by time-averaging data points in the charts.

V_3: Recieving system alerts

Purpose

The user should be able to recieve alerts if there are any issues detected with their aquaponic or hydroponic system. Currently push notifications are send to users phones if there are any issues detected. This section seeks to validate this functionality.

Results

Please see Table 7 for the questions we asked our client and his feedback about recieving system alerts.

Table 7: Questions to validate user experience when recieving system alerts.

	Question	**User Feedback**
Q_1	Were you able to configure your system's normal operating range for water temperature and pH?	Yes
Q_2	Were you able to understand warning and alert messages?	Yes
Q_3	Were you able to recieve a notification if your system was outside of its healthy operating range?	Yes - I am receiving notifications. It may be that I only get notifications when the app is running on my phone?
Q_4	Do you have any recommendations about how we might improve how alerts are sent and displayed?	Some alerts are more important than others. Leak - very important and needs to go out immediately. Water temp - pH level, could be handled as a summary once per day.

Analysis

There are two points of improvement that can be made here, as our client suggests:

1. Verify that notifications are recieved by phones even when the app is not running.
2. pH and water temperature notifications can be changed to daily notifications instead of immediate notifications.

The first point has previously verified in simulation. However, it should be tested with different phones including the client's to detect any issues.

V_4: Recalibrating sensors

Purpose

After reading the user manual, the user must be able to calibrate the each of the sensors to ensure their readings are accurate.

Results

Please see Table 8 for the questions we asked our client and his feedback about recieving system alerts.

Table 8: Questions to validate user experience when recalibrating sensors

	Question	**User Feedback**
Q_1	Were you able to successfully calibrate the pH sensor?	PENDING
Q_2	Are there sensors that you could not calibrate but would like to?	PENDING
Q_3	Was the calibration process clear and easy to follow for all sensors?	It appears okay. I did not go through the steps.
Q_4	Do you have any recommendations about how we might improve how sensors are calibrated?	I like how the pH level calibration is set up.

Analysis

This section needs to be repeated once our client has tested the calibration procedure. However, our client gave positive feedback about the calibration procedure as shown on the app.

V_5: Installation and setup procedures

Purpose

Users need to install our sensing and control system onto their hydroponic or aquaponics system. This section seeks to assess how straightforward it is for users to install our system.

Results

Please see Table 9 for the questions we asked our client and his feedback about recieving system alerts.

Table 9: Questions to validate user experience when recalibrating sensors

	Question	**User Feedback**
Q_1	Was it clear how to install all sensors and control devices onto your hydro/aquaponic system?	I was familiar with it already. Without that familiarity I may need some more instructions.
Q_2	Do you have any recommendations about how we might improve how our system can be installed?	A short instructional video - possibly with URL in form of a barcode.

Analysis

Our client was involved in the testing process throughout the development of our design. Therefore, this test should be repeated for new users.

Nevertheless, the recommendation of creating a short video explaining how to install the system would be helpful and can be done as a future step.

V_6: General feedback

Purpose

This section seeks to gather any other feedback that users may have that is not included in the above sections.

Results

Please see Table 10 for the questions we asked our client and his feedback about recieving system alerts.

Table 10: General user experience questions.

	Question	User Feedback
Q_1	Do you have any recommendations about how we might improve how our system in general?	I notice that the pump is coming on as soon as the pH level registers low. I would make adjustments slowly - when reading goes low, start a counter. Only if reading is confirmed by several readings, make an adjustment. Once adjustment is made, wait the same interval to make further adjustment. This process should override the apps interval settings for the pH sensor.

Analysis

The key piece of feedback here is to adjust the pH only if multiple measurements detect that the pH is low. This could improve the reliability of our control system despite fluctuating pH levels.