Autonomous Hydroponic Garden

Group 8:

Leandro Alepuz | Computer Engineering
Danny Nguyen | Electrical Engineering
Edwin Rivera | Electrical Engineering
Nathan To | Computer Engineering

Department of Electrical and Computer Engineering
University of Central Florida

Dr. Lei Wei, Dr. Samuel Richie

UNIVERSITY OF
CENTRAL FLORIDA

TABLE OF CONTENTS

1. Executive Summary 6

2. Project Description 7
- 2.1 Project Motivations 7
- 2.2 Goals and Objectives 8
  - 2.2.1 User Independence 8
  - 2.2.2 User Monitoring and Control 8
  - 2.2.3 Why Deep-Water Culture Hydroponics? 8
  - 2.2.4 Autonomy 9
  - 2.2.5 Rapid Growth 9
- 2.3 Requirements Specifications 10
- 2.4 Quality of House Analysis 11

3. Research related to Project Definition 12
- 3.1 Existing Similar Projects and Products 12
- 3.2 Relevant Technologies 15
  - 3.2.1 LAMP Architecture 15
  - 3.2.2 MERN & MEVN Architecture 15
  - 3.2.3 Database 16
  - 3.2.4 Relational Database 16
  - 3.2.5 Non-Relational Database 16
- 3.3 Strategic Components and Part Selections 17
  - 3.3.1 Microcontroller 17
  - 3.3.2 BeagleBone Black 17
  - 3.3.3 Raspberry Pi 4 Model B 18
  - 3.3.4 Deep Water Culture Floater 18
  - 3.3.5 Water Pump 20
  - 3.3.6 Nutrients & pH Pump 21
  - 3.3.7 Air Pump and Diffuser 22
  - 3.3.8 Light System 24
  - 3.3.9 Sensors 24
  - 3.3.10 pH Sensor 24
  - 3.3.11 Electrical Conductivity Sensor 25
  - 3.3.12 Total Dissolved Solids (TDS) Sensor 26

3.3.13 Water Level Sensor 26
3.3.14 Air Temperature/Humidity Sensor 26
3.3.15 Camera System 27
3.3.16 Grow Tent 27
3.3.17 Miscellaneous 28
3.4 Possible Architectures and Related Diagrams 28
3.4.1 Deep Water Culture 29
3.4.2 Wick System 29
3.4.3 Nutrient Film Technique 29
3.4.4 Ebb and Flow 29
3.4.5 Drip System 29
3.4.6 Aeroponics 29
3.4.7 Conclusion 30
3.5 Parts Selection Summary 30

4. Related Standards and Realistic Design Constraints 40
4.1 Design impact of relevant standards 40
4.1.1 Wireless Standard 41
4.1.2 Power Standard 41
4.1.3 Water Quality Standards 42
4.2 Realistic Design Constraints 42
4.2.1 Economic and Time constraints 43
4.2.2 Environmental, Social, and Political constraints 44
4.2.3 Ethical, Health, and Safety constraints 44
4.2.4 Manufacturability and Sustainability Constraints 45

5. Project Hardware and Software Design Details 47
5.1 Initial Design Architectures and Related Diagrams 47
5.2 Controller Subsystem 49
5.2.1 Central Processing Unit 49
5.2.2 Memory 50
5.2.3 Analog to Digital Converter 50
5.2.4 General Purpose I/O 52
5.3 Power Subsystem 53
5.4 Pump & Sensor Subsystem 54

5.5 Software Design 55
    5.5.1 Webstack 55
    5.5.2 Web Server 57
    5.5.3 Database 58
    5.5.4 Website 59
    5.5.5 Websocket 63
    5.5.6 PCB 65
5.6 Summary of Design 67

6. Project Prototype Construction and Coding 68
6.1 Growing System Prototype 68
    6.1.1 Miniature Version Prototype 70
6.2 Integrated Schematics 71
    Protection circuit LTSpice Simulation 85
6.3 PCB Vendor and Assembly 86
6.4 Final Coding Plan 90

7. Project Prototype Testing Plan 97
7.1 Plant Testing Environment 97
    7.1.1 Control Sample 97
    7.1.2 Seed Germination Process 98
    7.1.3 Measuring Plant Height 101
    7.1.4 Measuring Plant Weight 101
    7.1.5 Visual Testing 101
7.2 Hardware Test Environment 102
7.3 Hardware Specific Testing 103
    7.3.1 Water conductivity sensor 103
    7.3.2 pH sensor 104
    7.3.3 Water Level Sensor 105
    7.3.4 Air Temp/Humidity Sensor 106
    7.3.5 TDS Sensor 107
    7.3.6 Air Pump 108
    7.3.7 Nutrient and pH Pump 108
    7.3.8 Water Pump 108
    7.3.9 Water Temperature sensor 109

7.4 Software Test Environment 109
7.5 Software Specific Testing 110

8. Administrative Content 112
8.1 Milestone Discussion 112
8.1.1 Senior Design I Milestones 113
8.1.2 Senior Design II Milestones 115
8.2 Budget and Finance Discussion 115

9. References 118

1. Executive Summary

With the current climate crisis as well as rising food prices and inflation, the risk of food insecurity will rise even in developed countries. Supply chain issues have also been a contributor to the scarcity after COVID 19. Additionally, the average person does not possess the skills and time necessary to successfully grow their own food. This is the context where we thought of a solution to alleviate this existing problem.

Our project would be a fully automatic hydroponic system aiming to relieve the user of many of the necessary tasks needed to grow plants successfully, while also fitting in the average US household. Our team saw the need for a system that can reliably grow food for you while you are away, saving time and money.

The system will be a combination of hardware and software attached to a garden. The garden will be Deep Water Culture Hydroponics (DWCH), which is a growing technique that uses water instead of soil, dipping the plant’s root into oxygen and nutrient rich water. We are also expanding on the concept of DWCH with our own plant platform design that we think will improve results. The hardware will consist of a system of sensors, pumps and lights that are controlled by Microcontroller Unit. Finally, a software will be running on the MCU to control our plant system while also communicating with the Web app software that will act as a Frontend for our user.

The main requisites for a successful plant growth are nutrition, pH, light, water, and oxygen. We plan on having a system of two pumps controlling the pH, two other pumps controlling the nutrients, an additional more powerful pump refilling the water from a secondary reservoir when needed. Also, an oxygen pump will be pumping air inside the water, oxygenating the roots of the plants. The main goal is to minimize the user input needed, and our research concluded that these five components are the most labor-intensive part of growing, so by automating them we would ease a lot of the inconvenience of growing plants.

2. Project Description

This chapter will be an overall description of our senior design project. It starts with our primary motivations behind our decision to create an autonomous hydroponic system, as well the goals and objectives that we set for the project. The specifications for both the hydroponic system as well as the plants we intend on growing are also covered. Finally, a quality of house analysis is provided to show the most relevant qualities of our design project and how they affect one another.

2.1 Project Motivations

The rising popularity of growing your own produce at home has come with a major hurdle for users: the lack of time and botanical knowledge. Most people do not have the time or skills necessary to fully care for a plant and successfully harvest the fruits of their labor. This is why we propose creating a completely Autonomous Hydroponic Garden (AHG) that would monitor all the plant parameters and respond in real-time to its needs.

The event of COVID-19 contributed a lot to the motivation behind our project. With the world stuck indoors people developed new hobbies and interests like baking, cooking, fitness, etc. One of the main hobbies was home agriculture, with states like California increasing the sale of seed of vegetables and herb by 300% **[2.1a]**. Commonly called “lockdown gardeners” have found refuge in being able to sustain themselves during uncertain times. Overall, home growers have been looking after their own personal and health wellbeing, while discovering the benefits of growing their own food for their nutritional and mental health. We believe that with the pace of pre-COVID times coming back we will see a trend of people wanting to have their own produce but will not have enough time to dedicate to it. This is where our product comes in.

We also believe that food insecurity due to COVID ripple effects on the economy is a motivator for our product. We all saw how during the start of the pandemic massive panic buying left empty shelves **[2.1b]**. This led regular people to stress about the future of their access to food, and some of them turned to home agriculture as a response **[2.1c]**. However, the citizens who were able to make this transition mainly live in rural areas and have large space available for planting. We see the possibility of adapting this experience to the everyday household by making a compact growing system that can fit in an average US household.

Another motivator was the rising popularity of hydroponics. Plants grown in hydroponics are not planted in conventional soil, instead the roots are submerged in an inorganic medium to which nutrient rich water is applied. This is the basis for any hydroponic set up, but there is a wide variety of hydroponic systems. Hydroponic systems can save up to 90% of water compared to traditional soil methods. We chose hydroponic because it is the most efficient and cleanest way of growing at home. It saves the hassle of dealing with the soil itself, soil-borne diseases, or weeds infestations. It eliminates the water consumption from evaporation, drainage, and runoff by applying the exact amount of water needed by the plants. Finally, one of the main advantages we see with in-home hydroponics is the control that the user has over the climate, maximizing plant growth even in the off-season.

2.2 Goals and Objectives

This section provides further details on the goals and objectives of the project. It covers the capabilities we would like the customer to have, what type of hydroponic system we decided to design, what aspects we want to automate, and the possible growth rate that we can expect to see from the plants we have decided to grow.

2.2.1 User Independence

Our main goal is to free the user from the most mundane and repetitive tasks that plant growers have to do on a daily basis. This would be achieved by using a set of sensors that communicate with pumps in charge of pH stabilizers, nutrients, and water.

2.2.2 User Monitoring and Control

We want our user to be able to control everything from the web, as well as getting a live feed of the current state of the plant. We will build a web-app where you can see the most recent status update from the sensors in the garden, as well as historic data.

An important part of Monitoring will be the Warning Notifications. To prevent the plants from dying unexpectedly, our web app will notify the user when critical parameters have been reached. These critical parameters are visible at first sight: lack of oxygen, lack of nutrients, and critical pH. Out of all these, oxygen is the most critical because the plants can die in a matter of a day or two if the roots do not receive any oxygen. Also, we will notify the user of any plant discoloration via our integrated camera.

Additionally, the user must be able to change the desired parameters. We plan on making our system able to handle different types of plants which require different levels of nutrients, acidity, or light. We would like to allow the user to change these parameters if needed, making the system act accordingly to calibrate.

2.2.3 Why Deep-Water Culture Hydroponics?

We chose Deep Water Culture Hydroponics for our system. In DWC the plant's roots are directly submerged in water, which maximizes nutrient absorption. Additionally, an Air Pump oxygenates the water from the bottom of the tank (as seen in **Figure 2.2a**).

The rapid growth DWC offers will be useful when testing in the limited time we have to finish and test our design.

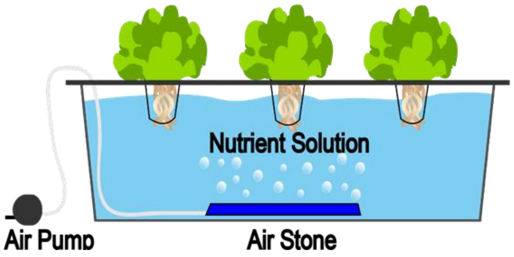


Figure 2.2a

2.2.4 Autonomy

Autonomy is also a crucial part of our project. We want to allow users to monitor the state of their garden, whether directly from home or remotely on any device. This would be accomplished through a live feed of data coming from different sensors surrounding the plants. This data would be stored in a database and be utilized by a back-end that sends it to our user on the front end, whether on a Desktop or Mobile environment. Continuous computer monitoring, combined with control of the plants' needs, will ensure a successful harvest with minimal supervision required from the user.

Our goal is to design a garden that can make the plants survive up to 2 weeks without any user interaction. The user would need to refill the following liquids when the system runs out:

- Water reservoir
- pH base and acid reservoirs
- Nutrients reservoir

2.2.5 Rapid Growth

One of the biggest advantages of hydroponics is rapid growth, up to 50% faster than traditional methods. We want to increase this rate even further, by using our own modification of Deep-Water Culture (Section consisting of a floating platform that will force the roots to be always in contact with our nutrient. Oxygen rich water.

We believe that our modifications will greatly increase the rate of growth significantly and plant. To test this, we plan on comparing a control harvest with a traditional method against regular DWCH and our modified version of DWCH.

2.3 Requirements Specifications

For our requirements we considered both the product requirements (**Table 2.3a**), and the plant's requirements based on a plant's needs to successfully grow in a typical U.S. household (**Table 2.3b**).

We want to make sure that our system fits in a typical room in the US and is transportable. We also want it to be power self-reliant, therefore we decided that it will not be battery powered, and instead it will be directly connected to a 110V outlet.

Requirements	Specifications
Compact size	< 36" x 24" x 20"
Power self-sufficiency	Direct connection to 120V standard outlet
Water Capacity	27 gallons
Sensor monitoring	6 sensors monitor the plants' environment
Affordable cost	~ $400
User Interface	LCD Screen

Table 2.3a

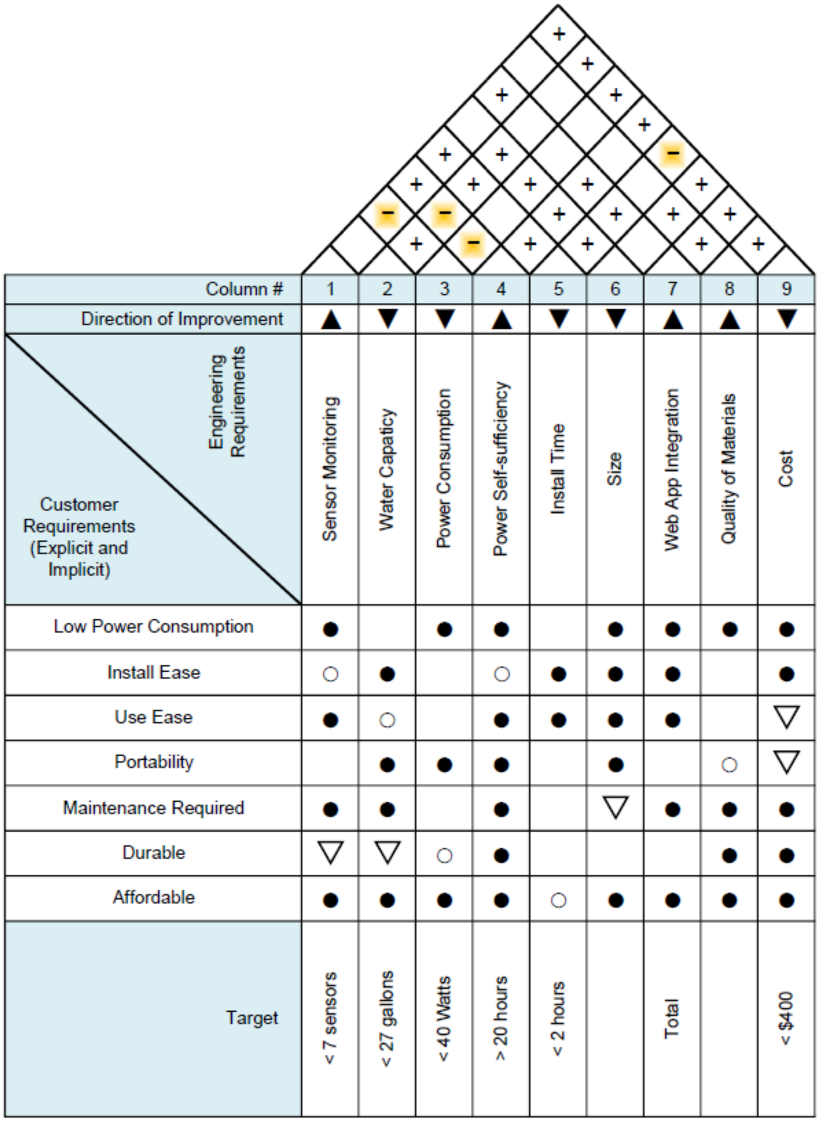
Plant Requirements	Recommended Range
Water Temperature	[65-80]° F
Water Electrical Conductivity (EC)	[0.5-2] mS/cm
Total Dissolved Solids (TDS)	[600-1000] ppm
Water pH level	[5.0 , 7.0]
Air Quality	[800-1000] ppm of CO_2
Air Temperature	[60-90]° F
Humidity	[50-70]%
Light	HPS Lights | 14 to 16 hours | Everyday

Table 2.3b

2.4 Quality of House Analysis

The Quality of House Analysis (shown in **Figure 2.4a**) is a tool that helps us direct the attention to the most relevant qualities of our design/product. Down below we can see our House Analysis, with the Engineering Requirements on the horizontal row and our Customer Requirements on the vertical column. The roof of our house shows how our Engineering Requirements interact with each other, while the dots and triangles show how the Engineering requirements affect the Customer Requirements.

From our House Analysis we concluded that the most affected factor is cost. For instance, we found out that by having more sensors we will not only increase the cost but also affect the installation.



Column #	1	2	3	4	5	6	7	8	9
Direction of Improvement	▲	▼	▼	▲	▼	▼	▲	▲	▼
Engineering Requirements / Customer Requirements (Explicit and Implicit)	Sensor Monitoring	Water Capaticy	Power Consumption	Power Self-sufficiency	Install Time	Size	Web App Integration	Quality of Materials	Cost
Low Power Consumption	●		●	●		●	●	●	●
Install Ease	○	●		○	●	●	●		●
Use Ease	●	○		●	●	●	●		▽
Portability		●	●	●		●		○	▽
Maintenance Required	●	●		●		▽	●	●	●
Durable	▽	▽	○	●				●	●
Affordable	●	●	●	●	○	●	●	●	●
Target	< 7 sensors	< 27 gallons	< 40 Watts	> 20 hours	< 2 hours		Total		< $400

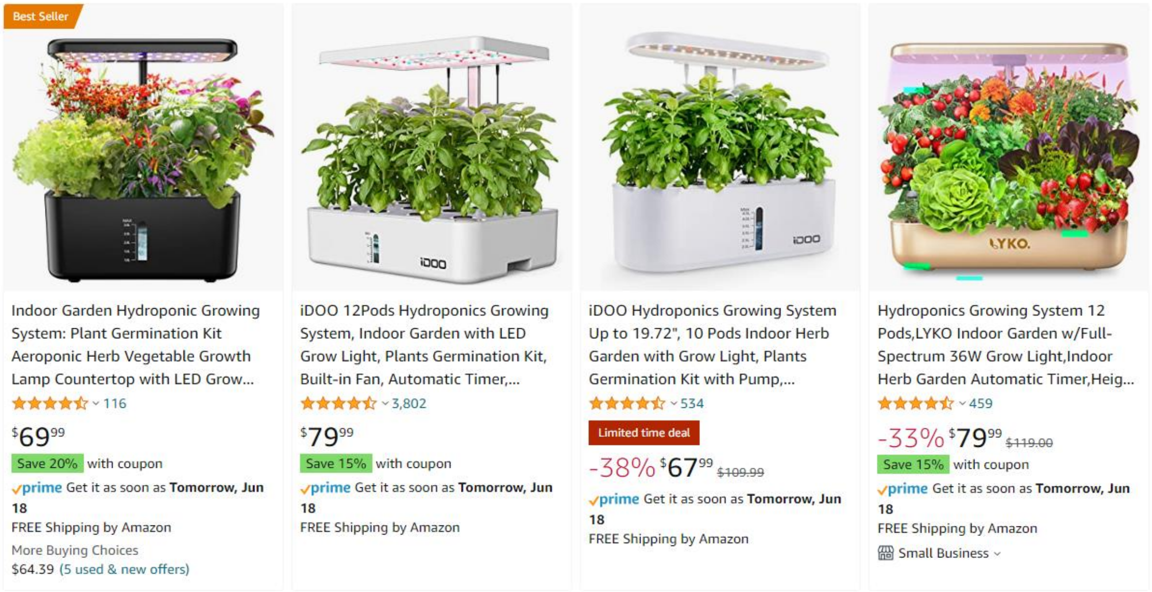
Figure 2.4a

3. Research related to Project Definition

This chapter begins with exploring a few of the products available on the open market that are similar in nature to the project we intend to design. These were fundamental in providing ideas on how our team could create a new rendition of automation in hydroponics. It then continues on to describe the relevant technologies that are required to introduce an interface into the system using a website. The main variations of hydroponic systems are briefly discussed before reviewing the primary strategic components that will be required to build our project. We compare several products for each component to assist in making the final selections that are shown in the last section of this chapter.

3.1 Existing Similar Projects and Products

After analyzing the current market on hydroponic solutions, we found that the main products are **standard hydroponic gardens** (**Figure 3.1a**). These gardens come with a water pump and automatic lighting that cycles throughout the day. They range from $50 to $120. This solution does not offer any automation whatsoever, and is dependent on the user to monitor water and environment.


Best Seller

Indoor Garden Hydroponic Growing System: Plant Germination Kit Aeroponic Herb Vegetable Growth Lamp Countertop with LED Grow...
116
$69.99
Save 20% with coupon
prime Get it as soon as **Tomorrow, Jun 18**
FREE Shipping by Amazon
More Buying Choices
$64.39 (5 used & new offers)

iDOO 12Pods Hydroponics Growing System, Indoor Garden with LED Grow Light, Plants Germination Kit, Built-in Fan, Automatic Timer,...
3,802
$79.99
Save 15% with coupon
prime Get it as soon as **Tomorrow, Jun 18**
FREE Shipping by Amazon

iDOO Hydroponics Growing System Up to 19.72", 10 Pods Indoor Herb Garden with Grow Light, Plants Germination Kit with Pump,...
534
Limited time deal
-38% $67.99 ~~$109.99~~
prime Get it as soon as **Tomorrow, Jun 18**
FREE Shipping by Amazon

Hydroponics Growing System 12 Pods,LYKO Indoor Garden w/Full-Spectrum 36W Grow Light,Indoor Herb Garden Automatic Timer,Heig...
459
-33% $79.99 ~~$119.00~~
Save 15% with coupon
prime Get it as soon as **Tomorrow, Jun 18**
FREE Shipping by Amazon
Small Business

Figure 3.1a

Additionally, we found high-end automated hydroponics that would accomplish our main goals. One that stands out is **Farm.bot**, which is an open-source company that sells a robotic arm system; it is equipped with computer vision and a variety of tools to tend your garden, including water hose (as seen in **Figure 3.1b**), harvester and camera. However, it requires some technical knowledge to use and assemble. Also, their cheapest product starts at $1600, which is not financially feasible for many consumers.

Figure 3.1b

We also found a product called Hybriponic Home Growing System (shown in **Figure 3.1c**). This is a vertical hydroponic garden solution that allows you to stack plants and cascade water with nutrients down the structure. It provides limited Water and Air temperature data through their App and their cheapest option costs around $700.

Figure 3.1c

In contrast, by using readily available technology, our AHG model would offer the best pricing without sacrificing the convenience of automation. We also want to differentiate from the available products by offering more data, customization and control to users, as seen on **Table 3.1**.

	Standard hydroponic gardens	**Farm.bot**	**Hybriponic**	**Our AHG Solution**
Easy to Use	✅	✅	✅	✅
Easy to Install	✅	❌	✅	✅
Scalability	❌	✅	✅	❌
Compact Size	✅	❌	✅	✅
Phone & Desktop App	❌	✅	✅	✅
Water Level Monitoring	✅	✅	✅	✅
Air Temperature/Humidity Monitoring	❌	✅	✅	✅
pH Monitoring	❌	❌	❌	✅
Nutrients Monitoring	❌	❌	❌	✅
Image Monitoring	❌	✅	✅	✅
Affordability	✅	❌	❌	✅

Table 3.1

3.2 Relevant Technologies

In order to create a hydroponics system with an interface that we can connect to remotely, we need a website. This website will be created using multiple different software technologies called a web stack. Our group has decided on three web stacks that we are considering for our website's architecture, the first one is the LAMP Stack which stands for Linux, PHP, MySQL and Apache. This stack is a simple and traditional stack to set up, which provides a website with great performance, cost efficiency and flexibility. Our second and third choices are the MERN stack and the MEVN stack which are alternatives which use MongoDB, Express.js, React.js/Vue.js and Node.js. Which are two very similar stacks that use different front-end applications. These stacks were chosen because of their popularity and the group’s previous experience using these stacks.

3.2.1 LAMP Architecture

The LAMP stack uses Linux as the operating system layer of the stack, considering it’s open-source and flexible and customizable. For the LAMP stack web server layer, it uses Apache HTTP Server, a popular web server that processes requests and transmits the information through the internet. It has many features such as supporting modular protocol handling and its filters. All its content can be encrypted, scanned for viruses, and compressed using these filters. The next application is MySQL, a database management system that supports SQL and relational tables. This database is also cross platform compatible but is inefficient in handling larger databases. Lastly, we use PHP for the API, PHP is the programming language that combines all the elements of the LAMP stack connecting the front-end to the back-end database. This language interacts well with MySQL and is a commonly used language for web development because it can be dynamically embedded into HTML.

3.2.2 MERN & MEVN Architecture

The MERN and MEVN stack refers to four different technologies MongoDB, ExpressJS, ReactJS, and Node.js. The first layer of this stack is MongoDB, a NoSQL document-oriented database, which stores data in JSON-like documents with dynamic schemas. MongoDB offers consistent reads and writes that scale linearly as more nodes are added. This document-oriented database storage system is much more flexible, and scale as opposed to SQL. The next layer in the MERN and MEVN stack is ExpressJS, another open source application framework written in java script. ExpressJS’s primary benefit is its ability to use JavaScript in all parts of the application in both the backend and front end. This next layer is where MERN and MEVN split off. MERN uses the ReactJS library for creating UI made by Facebook, this technology has the benefit of fast load times and great flexibility. MEVN uses VueJS, this technology is fast and easy to learn. It is an open-source framework where the back-end organizes the server-side allowing the back-end development to become faster and more efficient. The last technology that both stacks used is Node.js, which is the JavaScript runtime environment for making networks. The largest benefit of using Node is its ability to be used cross-form, it can easily be used on any system from PC to Mac to a mobile device.

3.2.3 Database

The database is the groundwork of the web stack. The database stores an assortment of interrelated data in a manner that allows efficient insertion, deletion and retrieval of data. The database stores and organizes the data in the form of tables, views, schemas, reports and ect. These databases are commonly modeled in rows and columns in a series of tables to make querying data efficient. In this project we will be picking between the two main types of databases, relational(SQL) and non-Relational databases.

3.2.4 Relational Database

SQL is the programming language that most relational databases use to query, update, manipulate and provide access to the data. Relational databases are a tabular approach to data that allows the data to be reorganized and accessed in many different forms. A relational database consists of tables and the data is placed into predefined sections of those tables. Each of these tables have at least one column with a data category and a row with a particular instance of data. Usually each row in the table will have a unique identifier called a key and each of the columns represent an attribute of that data. These attributes allow the data to be organized in all sorts of different ways making it easy to query via SQL. One of the main technologies that our group is considering if we decide to use a relational database is mySQL.

3.2.5 Non-Relational Database

A non-relational database also known as a NoSQL database is a database that does not use the usual table scheme of rows and columns often found in database systems. Instead, a non-relational database is a database model that is optimized for specific requirements for the types of data being stored. Two such examples are if the data is stored in key/value pairs, or the data is trying to store a graph of edges and vertices. These data types need to be a bit more specific for how it is supported and queried. Some non-relational databases may store data in document-like structures in a range of different formats. This ability of non-relational databases helps make non-relational databases much more flexible than relational databases. Non-relational databases offer many different benefits such as massive dataset organization allowing the database to maintain fast query speed while expanding the size of the database. Another benefit of databases is its ability to use multiple data structures no matter what format the data is in a non-relational database can collect the different information types into the same document database. The main non-relational database we are considering for our project is mongoDB with the only detriment to it is its inability to join together data or rows from two or more tables based on a common field.

3.3 Strategic Components and Part Selections

This section covers the key components that are required to completely build a basic deep water culture hydroponic system and make it fully automated. Items may be subject to change based on price and/or availability.

3.3.1 Microcontroller

Microcontrollers are necessary for many electronics to function, from cars to fridges, from phones to laundry machines. Microcontrollers are used to control many different electronic components such as sensors and screens. These electronics usually do not usually require much computation so a microcontroller can easily be used to control the electrical components. For our project the MCU will be used to control all the sensors and connect to the power systems. There are many specifications we need to consider when purchasing the microcontroller. We must make sure it has the correct number of pins, storage and system memory clock speed, the correct software, and power supply.

The number of pins is very important as the number of GPIO pins determine how many components and sensors we can attach to our microcontroller. Each of these pins allow the microcontroller to communicate and receive signals to the various lights, sensors, and pumps. We also need to be concerned about power consumption and processing power. Faster clock speed may be necessary to ping certain sensors faster for information. Also the various boards come in different types of operating system and built in modules. Some may come in different languages such as java, python, and c/c++. Some boards may come with Bluetooth and Wi-Fi modules built in, and others may come without those which is also something we need to consider. In the project we have two main options for microcontrollers, the BeagleBone Black and the Raspberry Pi 4 Model B.

3.3.2 BeagleBone Black

The BeagleBone Black is the first microcontroller that we have considered by Texas instruments and boasts an affordable price, high-performance, low-power and is open-source. This microcontroller uses a ARM Cortex-A8 processor, an efficient and high performance 32 bit CPU. This processor can scale in speed from 600MHz to greater than 1 GHz, scaling very well for its power usage. This board also supports up to 4GB of 8 bit on board flash storage. It has 69 GPIO pins for a variety of different sensors and it uses Debian as its operating system allowing the use of python and other Linux based languages. The beagle bone can support a camera for color and motion-based detection. The BeagleBone also uses fairly little power only needing 210mA(1.05W) to keep functioning, and an onboard AC to DC adapter making it convenient to add to our power system. The only negative aspect is that it lacks on board WIFI which adds the cost of a usb splitter and a Wi-Fi module.

3.3.3 Raspberry Pi 4 Model B

Our other option is the Raspberry Pi 4 the newest in the line up of raspberry pi's made for hobbyists. The Raspberry Pi acts as a lower priced mini computer with a small size, it can run many different Linux based software with its OS Debian optimized for the Raspberry Pi hardware. For its processor it uses a Quad core Cortex-A72 64 bit processor with speeds of 1.5GHz. The Raspberry board we are looking to purchase has 8GB of RAM and a built in bluetooth and a built in wifi module. This board also supports a camera for color and motion based detection but lacks an onboard AC to DC adapter. The Raspberry Pi has slightly less finds than the BeagleBone with only 40 pin GPIO header other than this it includes 2 micro HDMI ports and a audio port. This is a much more powerful option than the BeagleBone black but at the price of a bit more power draw of 500mA.

3.3.4 Deep Water Culture Floater

One of the main issues that we saw with a traditional DWC system (**Figure 3.3a**) is the necessity to be always pumping water to maintain the water level up to the height of the plant's roots. As can be seen on the diagram, with the roots absorbing water, the water level decreases, making the smaller roots not touch the water, which can cause the plant to stress until the water level is back up.

How traditional DWC fixes this is by constantly pumping water into the tank, so most roots are always in contact with water. However, we believe this is not optimal and there is room for improvement. For instance, a drawback is that the amount of oxygen, nutrients and pH stabilizers has to be proportional to the amount of water. Having unnecessary water means that you will need to use extra oxygen, nutrients, and pH stabilizers to maintain the same optimal ratio the plants need. Another major drawback is the power consumption of having to be constantly pumping water and stabilizers

An easy fix to this problem is seen in **Figure 3.3b**, designing a really small tank where the roots are always touching the water. However, this solution does not work for us because it would contradict our User Independence goal by forcing the user to fill up the reservoir more often than desired.

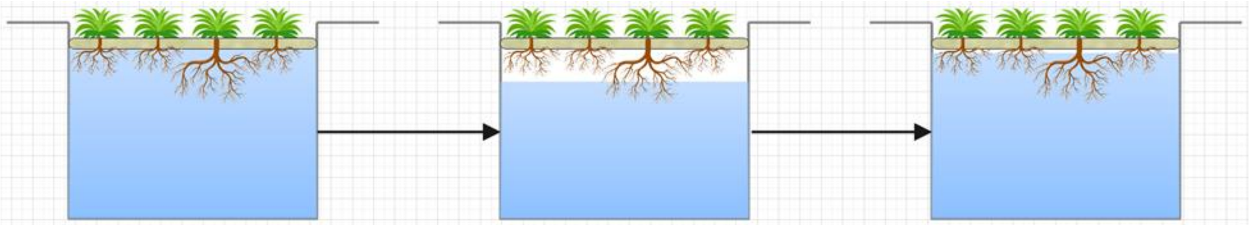


Figure 3.3a

Figure 3.3b

Our solution to the problem was inspired by the Modular Raft System Hydroponics. As seen on **Figure 3.3c**, these systems are floating rafts that carry the plants. These rafts are easily stackable and transportable through the water, making for a really smooth process when harvesting by acting as a moving conveyor belt.

Figure 3.3c

We designed a similar system for our project, where the plant pods are laying on a Styrofoam (or similar) floating platform. As seen on **Figure 3.3d**, as the roots of the plants absorb water the floater sinks down with the plants on it. At some point near 1/3 of the tank we could put our water sensor that can detect the low water level and trigger the water pump to refill from our reservoir tank until the maximum height is reached.

This solution would solve our initial problem perfectly by optimizing the amount of water used inside the tank while minimizing the times that the pumps are functional. Additionally, we will be optimizing the amount of nutrients and pH stabilizers by having a steadily decreasing amount of water.

Most importantly, it forces all the roots to be in contact with the water at all times. We strongly think that a combination of all these factors will also yield an increase in plant growth rate.

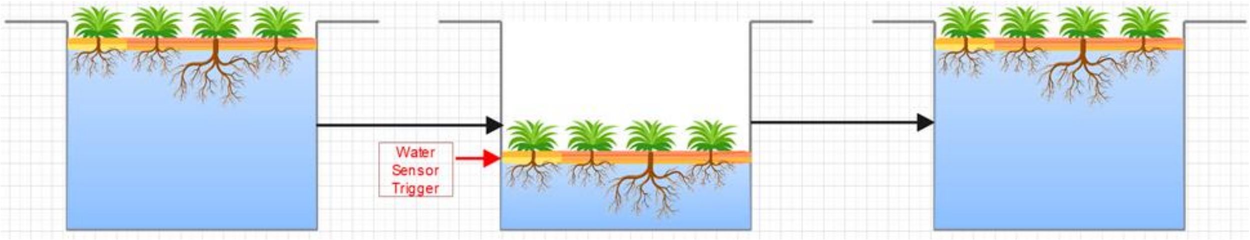


Figure 3.3d

3.3.5 Water Pump

The biggest attraction to hydroponics has primarily been the conservation of water. This is especially vital in areas that experience water shortages on an annual basis. One of the primary benefits to a deep-water culture hydroponic system is that the roots of the plants are continually submerged in oxygenated water that is full of nutrients. For this reason, a water pump is typically not used for this type of hydroponic technique. However, the deep-water culture systems are not normally automated, so it is important for the customer to continually maintain the water supply so that the roots of the plants do not dry out. For the purposes of automation, we want to introduce a water pump that will deliver the appropriate amount of water to the primary reservoir from an external water source once the water level has dropped below the minimum level that is required.

We reviewed three different water pumps that could potentially be used for this project. The first is a 12V DC dosing pump manufactured by Gikfun, that has a flow rate of 0.1 liters per minute. It has a specified current of 80 mA and a horizontal head of 25 meters. Although the specifications for this pump are mostly appropriate, the low flow rate made us decide not to choose it. It would take an hour to fill 1.585 gallons of water and the primary concern is overexerting the water pump.

The second option is a 12V DC submersible water pump manufactured by Aideepen, that has a flow rate of 600 liters per hour and uses 17 watts of power. It has a brushed motor and cannot be run for more than 24 hours at a time, but continuous water flow is not required for a deep-water culture system. Some consideration was given to choosing this water pump, but a lack of online reviews kept us from doing so. The third option is a 12V DC mini submersible water pump manufactured by LEDGLE, that has a flow rate of 240 liters per hour. It has a specified current of 300 mA, uses 3.6 watts, and the motor is brushless. With both the cost and power usage less than that of the water pump made by Aideepen, we decided that this would be the most appropriate water pump that can fulfill our needs.

3.3.6 Nutrients & pH Pump

The most vital part to successful plant growth in a hydroponic system is maintaining the water chemistry at the required levels for the plants being grown. Normally, this would require that the customer measure the pH and electrical conductivity levels of the water before administering trace amounts of acid (to lower the pH levels), base (to raise the pH levels), and two different nutrients (to raise electrical conductivity levels). It was decided to purchase the pH control kit from General Hydroponics (shown in **Figure 3.3e**) because of its low cost and reputation for providing reliable results. It is an all-in-one kit that not only contains the acid/base liquids, but a test indicator with a test tube, eyedropper, and pH level chart. These additional items will prove useful in ensuring the accuracy of our pH sensor.

Figure 3.3e

It was also decided to use the base A and base B plant growing nutrients by TPS Nutrients (shown in **Figure 3.3f**). This two-part solution contains the necessary minerals required to provide a balanced nutrition and can be used to assist in the growth of all plants. TPS Nutrients also provided a recommended feed chart to help give our team a better understanding of how to use the nutrients and can be seen in **Figure 3.3g** (courtesy of their online store at tpsnutrients.com) shown below.

Figure 3.3f

For our system, these solutions will be dispensed using peristaltic pumps that are ideal in delivering specified amounts of fluids. The first peristaltic pump we reviewed is a 12 V DC pump manufactured by INTLLAB with a flow rate of approximately 0.04 liters per minute and is capable of operating within a temperature range of 0°C - 40°C and a relative humidity < 80%. The second peristaltic pump is also a 12 V DC pump, but is manufactured by Kamoer and has a flow rate of approximately 0.07 liters per minute. The operating conditions also match that of the pump manufactured by INTLLAB. Lastly, we reviewed the 12V DC peristaltic pump made by Yanmis. Its flow rate is approximately 0.065 liters per minute and also has the same operating conditions as the two previous pumps. Since all three peristaltic pumps are very similar regarding their specifications, our choice will be dictated by price and availability. For the four solutions, we chose a pH control kit by General Hydroponics due to its relatively low cost and the excellent reviews from the online hydroponics community. We also decided to choose Base AB solutions by TPS Nutrients.

3.3.7 Air Pump and Diffuser

Providing sufficient oxygen to the water in the reservoir prevents the roots of the plants from suffocating and requires the use of an air pump in conjunction with an air diffuser. As the air is injected into the reservoir's water, agitation is caused on the surface level of the water by displacing it with air bubbles. This agitation allows the carbon dioxide to exit the water while simultaneously allowing more oxygen to enter the water.

Three different air pumps were reviewed that could potentially be used for our project. The first is an aquarium air pump manufactured by Uniclife that can sustain a reservoir up to 60 gallons. It has dual outlets to allow for better disbursement of oxygen and has a max air flow rate of 64 gallons per hour. Additionally, the air flow can be adjusted using a rotary knob to provide the customer with more control over the desired air flow. The second is an aquarium air pump manufactured by HITOP that can sustain a reservoir up to 100 gallons. It also has dual outlets like the Uniclife air pump and has a max air flow rate of approximately 79 gallons per hour. The final air pump that we reviewed is manufactured by AQQA and can sustain a reservoir up to 110 gallons. Compared to the other pumps, it only has a single air tube outlet and has a max air flow rate of 19 gallons an hour.

When air pumps are used in hydroponic systems or aquariums, they are always connected directly to an air diffuser. Air diffuser was chosen over Air stone because it increases the area the oxygen is delivered to. As seen in **Figure 3.3.5a**, the Air Diffuser covers a large area around the tank, it is flexible and malleable to our requirements. When the air diffuser is on, as seen in **Figure 3.3.5b**, we can see that there is an even distribution of oxygen around the tank. We considered this was necessary because otherwise we would have plants around the edges starved from oxygen, affecting their growth.

Air stones (**Figure 3.3.5c**) are typically made of either a porous rock containing many small holes or limewood, while air diffusers are made of rubber materials with orifices where small bubbles can form and release. An air diffuser's main purpose is to receive large bubbles from the air pump and release them back into the tank/reservoir as smaller bubbles. Decreasing the size of the air bubbles provides better aeration to the water by widening the oxygen dispersed throughout the system and also minimizes the turbulence that is caused at the surface level.

Figure 3.3.5a

Figure 3.3.5b

Figure 3.3.5b

3.3.8 Light System

Another important factor in maintaining the health of the plants is providing the proper amount of light. This can vary based on the size of the deep-water culture hydroponic system as well as the plant growth phase. The lighting system will run off a timer so that the power is turned off for 8 hours at night to give the plants a resting period. Additional lighting is not required for an outdoor system, but it is vital for an indoor system. According to the article "*Deep Water Culture (DWC) Ultimate Guide"*, by smallscalegardener.com, in commercial hydroponic systems, high-intensity discharge (HID) lamps are used during the growth phase and high-pressure sodium (HPS) lamps are used during the bloom phase. As of recently however, multi-spectrum LED grow lights are becoming more popular since less heat is generated by the lights, and they also consume less power. Since we already own a lighting system that is manufactured by Daystar AC Hydrofarm, we only need to purchase the necessary bulbs. The wattage levels will be chosen once a final selection has been determined for which plants will be grown.

3.3.9 Sensors

In order to satisfy our own requirements for a fully automated hydroponic system, it is crucial that the necessary sensors are incorporated to periodically measure the chemistry levels of the water in the reservoir as well as other factors that will minimize the amount of manual work required by the customer.

3.3.10 pH Sensor

Maintaining the pH levels of a hydroponic system is important because it is what affects the availability of nutrients to the growing plants. Such deficiencies with the intake of nutrients can cause pale and/or yellowing of the leaves as well as burnt leaf tips. As such, choosing the right type of pH sensor that can handle continuous measuring on a long-term basis is required for our project. The required pH level can vary based on the type of plants being grown, but the pH level that we are attempting to maintain is between 5pH - 7pH.

The first pH sensor that was considered was manufactured by Atlas Scientific and has a measurement range between 2pH - 13pH with an accuracy of +/- 0.1pH and can operate in temperatures ranging from 1°C - 60°C. Although it's the most recommended pH sensor in the online hydroponic community, it's also the most expensive compared to the others that were considered. The second pH sensor that was considered was manufactured by OCESTORE and has a measurement range between 0pH - 14pH with an accuracy of +/- 0.25pH and can operate in temperatures ranging from 0°C - 60°C. The third pH sensor that was considered was manufactured by GAOHOU and similar to the sensor made by OCESTORE, it also has a measurement range between 0pH - 14pH with an accuracy of +/- 0.25pH and can operate in temperatures ranging from 0°C - 60°C. The cost of the sensor is slightly higher, but has much more positive reviews in the online community, so we decided that it would be the most appropriate choice for our hydroponic system.

3.3.11 Electrical Conductivity Sensor

The electrical conductivity measurement of a solution is used to indicate the amount of dissolved salts that are contained within. This is an important factor because if the concentration of salts within the nutrient solution is higher than the appropriate levels for the plant, the roots will be unable to retain the water. This is similar to "fertilizer burn" where the water is completely depleted from the plants and causes dehydration as well as leaf burns. The required electrical conductivity level can also vary based on the type of plants being grown, but the electrical conductivity level that we are attempting to maintain is between 500µS/cm - 2,000µS/cm. The first electrical conductivity sensor that was considered was manufactured by Atlas Scientific and has a measurement range between 5µS/cm - 200,000µS/cm with a response time of 90% in one second. It's also the most recommended electrical conductivity sensor in the online hydroponic community, and also one of the more expensive products on the market.

The second electrical conductivity sensor that we reviewed was manufactured by HoneForest and has a measurement range between 0µS/cm - 9,990µS/cm. Although a response time was not provided online, the biggest benefit of this sensor is that it's multipurpose and relatively cheap. It can measure not only the electrical conductivity of the solution, but also the total dissolved solids (TDS) and solution temperature. It was decided not to choose this particular sensor because it does not come with the capability to communicate with our MCU via Arduino, so much more work would be required to customize the sensor to fit our required needs. The third electrical conductivity sensor that we considered was manufactured by Apera Instruments and has a measurement range between 5µS/cm - 200,000µS/cm. Further research has actually shown that the results of a TDS meter can be mathematically converted to show the results of electrical conductivity. For this reason, it was decided not to purchase this additional sensor.

3.3.12 Total Dissolved Solids (TDS) Sensor

The total dissolved solids in a solution is similar to the electrical conductivity as they both are vital to the plants ability to take in the water and nutrients being provided. Where electrical conductivity shows how well a plant is able to absorb the nutrients in the system, the total dissolved solids show the amount of nutrients in the solution. Once again, the required total dissolved solids that should be present in a solution varies based on the type of plants being grown. The level that we will be attempting to maintain is between 600ppm - 1000ppm (parts per million).

The first total dissolved solids sensor that was considered was manufactured by ASHATA with a measurement range of 0ppm - 1000ppm and measurement accuracy of +/- 10%.

The second total dissolved solids sensor that was considered was manufactured by KUIDAMOS and has a measurement range of 0ppm - 1000ppm with a measurement accuracy of +/- 12%. The third total dissolved solids sensor that was considered was manufactured by PUSOKEI and has a measurement range of 0ppm - 1000ppm with a measurement accuracy of +/- 9%. Since all three total dissolved solids sensors are closely similar to one another, our final choice will be determined by the price and availability.

3.3.13 Water Level Sensor

The next important factor for our automated hydroponic system is maintaining an appropriate water level for the reservoir. In a deep-water culture system, the roots of the plants are constantly submerged in the nutrient and oxygen-rich solution. As such, the water level will continually drop as the water is gradually absorbed by the roots. The recommended water level is to keep approximately 1.5 inches of the plant roots above the water, allowing the dry parts to absorb oxygen from its surrounding environment more efficiently. There are many affordable water level sensors available on the market, but we already own an electronics kit by ELEGOO (MEGA 2560 Project), that comes with a good quality water level sensor. According to the data sheet, the sensor was manufactured by China Harbin Okumatsu Robot Technology Co. and its product item is RB-02S048. It operates at a working voltage of 5V and a current less than 20 mA. The sensor also has a working temperature between the range of 10°C - 30°C and has an output voltage signal of 0V - 4.2V.

3.3.14 Air Temperature/Humidity Sensor

The last sensor that is vital to maintaining the health of our growing plants is the air temperature and humidity sensor. When plants are in the process of germinating and in the beginning stages of growing their roots, a certain level of moisture in the air is required. However, with higher temperatures and humidity comes an increased risk of mold developing or attracting certain pests. There are many types of temperature/humidity sensors available on the market with varying levels of accuracy, but since our hydroponic system will mainly be within a climate-controlled environment, an expensive sensor with high levels of accuracy is not a high priority. For this reason, we chose to use the DHT11 air temperature/humidity sensor that also came with the ELEGOO electronics kit. The resolution of the relative humidity being measured is 16 bit and the accuracy is +/- 5% RH with a surrounding temperature of 25°C. The resolution of the temperature being measured is also 16 bit and has a range of +/- 2°C with a surrounding temperature of 25°C. The DC power supply for this sensor should be approximately 3.5V - 5.5V.

3.3.15 Camera System

For the camera system we have three possible features in mind for the usage of the camera. The first feature would be a live broadcast of the plant at all times that can be accessed remotely from the web server. This would allow a user to observe the condition of the plant at all times for any abnormalities. The second feature we are thinking of adding is python code to issue a warning to the web server if there is any discoloring from the natural color of the plant. We plan to cut the camera frame to only show the plant and take a color range of yellow and brown. Based on this color detection we can issue a warning once the discoloring takes over a certain percentage of pixels in the frame. The last feature we were thinking of is using the webcam to monitor plant height and grown growth using openCV. With this technology we can even send alerts to users to harvest based on the plant growth.

3.3.16 Grow Tent

Grow tents are another component that play an important role in the growth of plants within a hydroponic system. They allow the user to have a higher level of control over the plant's environment while simultaneously providing improved levels of energy efficiency. All grow tents have a reflective surface on the inner walls to allow for circulation of lighting from the grow lights. This wider concentration of light provides a higher level of efficiency for the photosynthesis of the plants. There were a couple of key components to consider when choosing a grow tent with the first being the overall size. We are anticipating using a heavy-duty tote with a capacity of 27 gallons and approximate dimensions of 30.6" (D) x 20.6" (W) x 14.3" (H). To accommodate this reservoir along with the other necessary components, we believe that a grow tent that is at least 32" x 32" x 63" or slightly larger should be sufficient. The next component to consider is the material being used in the grow tent. Most modern grow tents are created using heavy-duty canvases such as nylon or polyester. However, the thickness of the material being used varies with a "D" rating that ranges anywhere from 200D (thinnest material) to 1680D (thickest material).

The first grow tent that was considered for our project is manufactured by VIVOSUN and comes in two different sizes (32" x 32" x 63" and 36" x 36" x 72") and the thickness of the material is rated at 600D. The second grow tent that we reviewed is manufactured by MELONFARM and comes in several more sizes compared to that of VIVOSUN and the thickness of the material is also rated at 600D. Since the two manufacturers make grow tents with similar specifications and price, the decision to choose between the two would rest solely on availability. The third grow tent that we are considering is manufactured by YITAHOME and also comes in many different sizes. The key differences between this grow tent and the others are that the thickness of the material is rated at 1680D at the observation window is larger by comparison.

3.3.17 Miscellaneous

Besides the primary strategic components listed above, there are several other components that are necessary for the completion of our project. There are many different kinds of containers that can be used to act as the reservoir for our deep water culture hydroponic system. It was decided to choose a heavy-duty plastic tote for strength and durability. We also wanted a reservoir big enough to grow several plants so we settled on a 27 gallon 30.6” x 20.6” x 14.3” container that comes with a standard snap lid. We should be able to drill several holes through the lid to be the resting place for all the plants. Net pots are plastic containers with several small openings along the sides and bottom to allow for drainage of excess water, better air circulation, and space for the plant’s roots to grow through so that they may fully reach into the water of the reservoir.

These net pots would sit right into the holes that are drilled into the lid of the heavy-duty tote. Grow cubes are a hydroponic plant media that assist in the initial germination of a plant’s seeds. We will first allow the seeds to germinate within the grow cubes using a seedling heat mat before transitioning them to the net pots. A plastic mounting panel is also necessary and will serve as the location where the sensors and other electronics will be placed. This mounting panel should be close enough to the reservoir to avoid unnecessarily long wires, but separated enough that water damage will not be a concern.

3.4 Possible Architectures and Related Diagrams

Once our team had decided that we wanted to create an automated hydroponic system, we began researching the various hydroponic techniques that are widely used and found that there are six main types of hydroponic systems.

3.4.1 Deep Water Culture

The first is a deep water culture system that simply suspends the plants in aerated water. We went with this architecture because it is really simple and efficient. It needs the least amount of moving parts, while requiring a very small amount of energy to maintain the flow of water.

3.4.2 Wick System

In a wick system plants sit within a growing media that rests atop a reservoir. The nutrient rich water flows up a wick and waters the growing media around the roots of the plant. While Wick seems very similar to Deep Water Culture in its simplicity, we found that having two different reservoirs was not as compact.

3.4.3 Nutrient Film Technique

In nutrient film technique the plants are placed above a stream of nutrient rich water that is constantly running through the root systems of the plants. This system is ideal for large spaces because it allows for a lot of plants to be planted along the stream of water. We decided that this was not optimal because it required a more complex assembly process with more moving parts and a constant pumping of water against gravity.

3.4.4 Ebb and Flow

Ebb and flow involves flooding a grow bed with nutrient rich water for a certain time limit using a water pump. Once the time has elapsed, the water pump is turned off and the water drains from the grow bed via gravity. This technique proposes similar challenges as the Nutrient Film and Wick System.

3.4.5 Drip System

Drip system that aerates nutrient rich water from the reservoir using water pumps through a network of tubes that individually waters each plant. As previous systems, this is an active system that requires constant water pumping.

3.4.6 Aeroponics

The last architecture we considered is aeroponics, which releases a fine mist of nutrient rich water from a reservoir over the plants. The reason we really considered this system is because it is extremely water efficient (up to 90% water savings compared to traditional). However, it requires moving parts that can easily malfunction, requiring constant supervision.

3.4.7 Conclusion

Each of the six techniques have both pros and cons, but ultimately we decided on developing a deep water culture system because of its simplistic approach. The main benefits to using a deep water culture system is its low maintenance requirement. Once the system has been completely put together, the customer only needs to worry about replenishing the water supply every 2-3 weeks, ensuring that the air pump is continually providing oxygen to the reservoir, and maintaining the appropriate chemistry levels of the water in the reservoir. The primary disadvantages to using a deep water culture system is the limitation on what kind of vegetation can be grown and maintaining a water temperature approximately between 60°F - 68°F. The temperature control can be difficult to maintain due to the stagnant water of a deep water culture system.

The primary goal of our project was to create a fully automated deep water culture hydroponic system, but we also want to make sure it is as user-friendly as possible. This can be done by creating a system design that is intuitive as well as developing a step by step manual on how to perform certain operations with the hydroponic system. Further instructions will also have to be provided that instructs the customer how to properly maintain the hydroponic system so that the lifespan can be extended as far as possible.

We intend on placing the heavy-duty tote within the grow tent and aligning all electronic equipment nearby but in a water proof container to avoid water damage and possible electrocution. We would also like to incorporate a floating apparatus that can support the plants instead of using the lid of the heavy-duty tote. This provides the added benefit to the customer of not having to replace the water in the reservoir as often.

3.5 Parts Selection Summary

This section will cover the final selections that our team has made for all the primary parts of our hydroponic system. Much deliberation was given towards each component that is required for our project and the primary characteristics that were compared against the alternative options involved the cost, durability, accuracy, and availability.

<u>Grow Tent:</u>

A 36”x36”x72” Yitahome grow tent (shown in **Figure 3.5a**) was chosen to help control the growing environment around the hydroponic system. Its key highlights include: An Oxford canvas with a thickness of 1680D (thickest grown tent canvas on the market) prevents light leaking from the tent to ensure the plants absorb as much light as possible. The interior part of the canvas is made of a reflective diamond Mylar material to also allow for more efficient light reflection. The frame of the tent consists of metal poles and clasps with corner connectors that do not require tools to assemble. The hanging bar of the frame can also support a maximum load of 120 pounds. It also contains an easily foldable observation window to allow the customer to view the plants without having to fully open the grow tent. Besides the ventilation port near the top of the tent to insert an exhaust fan, it also contains ventilation vents at the bottom of the tent to provide more efficient dissipation of accumulating heat.

Figure 3.5a

Microcontroller:

It was decided that we would use the Raspberry Pi 4 (shown in **Figure 3.5b**) over the BeagleBone due to the fact that we have more experience using this particular hardware. The key specification of this product include:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 4GB LPDDR4-2400 SDRAM
- 2.4GHz and 5.0 GHz IEEE 802.11ac Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports and 2 USB 2.0 ports

- Raspberry Pi standard 40 pin GPIO header
- 2 micro-HDMI ports supporting up to 4Kp60 video resolution
- 2-lane MIPI DSI/CSI ports (display/camera)
- 4-pole stereo audio and composite video port
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A)
- 5V DC via GPIO header (minimum 3A)
- Power over Ethernet (PoE) enabled
- Operating temperature: 0°C - 50°C

Figure 3.5b

Temperature/Humidity Sensor:

The DHT11 sensor (shown in **Figure 3.5c**) was chosen since we already own one from the Elegoo MEGA2560 electronics kit and it has a digital signal output with a calibrated temperature and humidity combined into one sensor. The sensor's parameters are listed in **Table 3.5a**:

Relative Humidity	**Temperature**	**Electrical Characteristics**

Resolution: 16-bit	Resolution: 16-bit	Power Supply: DC 3.5 ~ 5.5V
Repeatability: +/- 1% RH	Repeatability: +/- 0.2°C	Supply Current: Measured 0.3mA and Standby 60µA
Accuracy: At 25°C +/- 5% RH	Range: At 25°C, +/- 2°C	Sampling Period: > 2 sec
Response Time: 1/e (63%) of 25°C 6s	Response Time: 1/e (63%) 10S	
Hysteresis: < +/- 0.3% RH		
Long-term Stability: < +/- 0.5% RH/yr in		

Table 3.5a

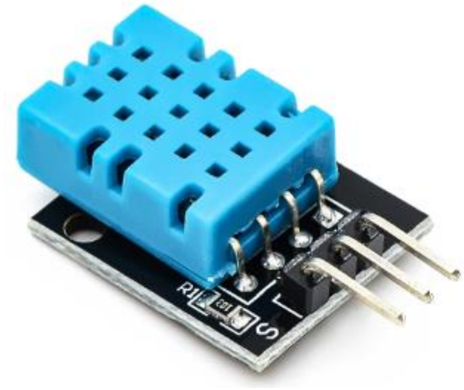


Figure 3.5c

pH Sensor:

A Gaohou pH sensor (shown in **Figure 3.5d**) was selected because of its relatively low cost, moderate reviews, and it already comes with an Arduino for ease of use. The key specifications of this sensor module include:

- Heating voltage: 5 +/- 0.2V (AC DC)
- Working current: 5mA - 10mA
- Detectable concentration range: 0pH - 14pH

- Detection liquid temperature range: 0°C - 80°C
- Response time ≤ 5 seconds; Settling time ≤ 60 seconds; Component power ≤ 0.5 watts
- Working air temperature: -10°C - 50°C (nominal temperature 20°C)
- Module size: 42mm x 32mm x 20mm

Figure 3.5d

TDS Sensor:

The Pusokei TDS sensor (shown in **Figure 3.5e**) was also chosen because of its low cost and its reported high level of accuracy. The specifications of the signal adapter board are the following:

- Input voltage: 3.3V - 5.5V; Output signal: 0V - 2.3V; Working current: 3mA - 6mA
- TDS measurement range: 0ppm - 1000ppm (parts per million)
- TDS measurement accuracy: +/- 10% F.S. (25°C)
- Size: Approximately 42mm x 32mm
- Module interface: XH2.54-3P; Electrode interface: XH2.54-2P

Figure 3.5e

Water Level Sensor:

The CQRobot water level sensor (shown in **Figure 3.5f**) was chosen over the Elegoo sensor because it is more waterproof and already comes with an Arduino component for ease of use. The Elegoo water level sensor would have required further modifications from our team to ensure that all of the electrical components would not be exposed to any water. The key specifications of the CQRobot water level sensor include:

- Power Supply Voltage: DC 5V
- Output Current: 12mA
- Working Temperature Range: -25°C - 105°C
- Low Level Output: < 0.1V and High Level Output: 3.3V - 5V
- Liquid Level Detection Accuracy: +/- 0.5mm
- Approximate Lifespan: 50,000 hours

Figure 3.5f

Water Temperature Sensor:

The Gaohou DS18B20 water temperature sensor (shown in **Figure 3.5g**) was also incorporated because research has shown that maintaining an appropriate water temperature not only ensures a better level of calibration for the pH sensor, but it also allows for more accurate measurement readings of the water's pH level. The key specifications of the Gaohou water temperature sensor include:

- Working Voltage: 3.2V ~ 5.25V DC
- Working Current: 2mA
- Resolution: 9-12 bit programmable
- Measuring Range: -55°C - 110°C
- Measuring Accuracy: +/- 0.5°C @ -10°C - 80°C and +/- 2°C @ < -10°C or > 80°C

Figure 3.5g

Air Pump:

The aquarium air pump by Uniclife (shown in **Figure 3.5h**) was chosen for our hydroponic system due to its low cost, excellent online reviews, and relatively low ambient noise. Since the air pump will have to run continuously to ensure that an appropriate amount of oxygen is provided to the plants, we wanted to ensure that the noise level of the air pump would be low enough as to not cause a disturbance. Additionally, the air pump is ideal for aeration in fresh-water tanks that vary in size from 10 - 60 gallons and has dual outlet ports. The key specifications of the Uniclife air pump include:

- Maximum Air Flow Rate: 64 gallons per hour with a pressure of 0.016 MPa
- Noise Emission: 25db - 45db (minimum to maximum air flow)
- Power Consumption: Approximately 4W

Figure 3.5h

Water Pump:

The mini submersible water pump by Ledgle (shown in **Figure 3.5i**) was chosen because of its low cost, its ability to be operated underwater, and its relatively low ambient noise. Additionally, its stellar online reviews have shown that this particular water pump is heavily used in both aquariums and hydroponic systems. The key specifications of the Ledgle water pump include:

- Input Voltage: 12V DC; Power Consumption: 3.6W; Current: 300mA
- Max Lift Height: 3 meters; Max Flow: 240 liters per hour
- Noise Emission: ≤ 40db; Max Water Temperature: 60°C

Figure 3.5i

Peristaltic Pump:

A peristaltic pump by Gikfun (shown in **Figure 3.5j**) was chosen to be the dispensing unit for the pH chemicals (base and acid) as well as the nutrients for the plants. The primary reasons for choosing this particular peristaltic pump are its low cost and its accurate liquid disbursement. It's also known for its "snap-in" type design that allows for easy removal of the pump head when the pump tubes require cleaning or replacement. The direction of the flow can also be controlled by the positive and negative connections. It is also used in many different applications from supporting medical equipment to liquid dispensing in the food and beverage industry. The key specifications of the Gikfun peristaltic pump include:

- Input Voltage: 12V DC; Current: 80mA
- Relative Humidity: < 80%; Operating Temperature: 0°C - 40°C
- Flow Rate: 0 - 100 mL/min; Horizontal Head: 25 meters
- Motor RPM: 5000 rpm; Rotate Speed: 0.1 - 100 rpm

Figure 3.5j

Water Temperature Sensor:

A Waterproof digital temperature sensor with an adapter module for an DS18B20 adapter Module for the arduino. This temperature module helps determine the health of the roots of the plants. This sensor ensures the water is within livable temperatures for the plant. This module also helps in calibrating the our tds sensor which determines the amount of nutrients in the water. In the calculations for TDS the water temperature sensor is necessary for accurate readings and will help in determining the amount of nutrients to add to the water. This water temperature sensor made by GAOHOU (shown in Figure 3.5k) has many other features shown below.

- Working voltage: 3.2 ~ 5.25VDC
- Working current: 2mA (max)
- Resolution: 9-12 bit programmable
- Measuring range: -55 ~ 110 ℃
- Output leads: yellow (DATA), red (VCC), black (GND)
- Measuring accuracy: ±0.5'C@-10~+80'C; ± 2'C @ -55 ~ + 110'C

Webcam:

For this project the webcam we will be using is the Logitech C270 HD Webcam (shown in Figure 3.5l). The reason we chose this webcam is because it is directly compatible with linux OS. Therefore when we add this webcam into the raspberry pi we can directly access its feed without needing to install any new drivers. The main reasons we will be using this webcam is in order to maintain a live feed of the plant 24/7 in order to observe any accidents that could happen with the plant remotely. There are also various other scripts we can run to monitor the discoloration and growth of the plant.

Figure 3.5k

- HD 720p video calling and HD video recording, 2.4 GigaHertz
- Intel Core2 Duo, 2 GB RAM,200 MB hard drive space
- Video capture: Upto 1280 x 720 pixels, Logitech fluid crystal. Focus type: Fixed focus
- Crisp 3 MP photos technology, Hi speed USB 2.0
- Compatible with: Windows 10 or later, Windows 8, Windows 7, Works in USB video device class (UVC) mode with supported video calling clients: MacOS 10.10 or later, Chrome OS, Android v 5.0 or above

Figure 3.5l

4. Related Standards and Realistic Design Constraints

For this chapter, we cover the main industry standards that we will be adhering to since they will be directly influencing our entire design process. Additionally, we will be reviewing some of the design constraints that can and/or will affect our project.

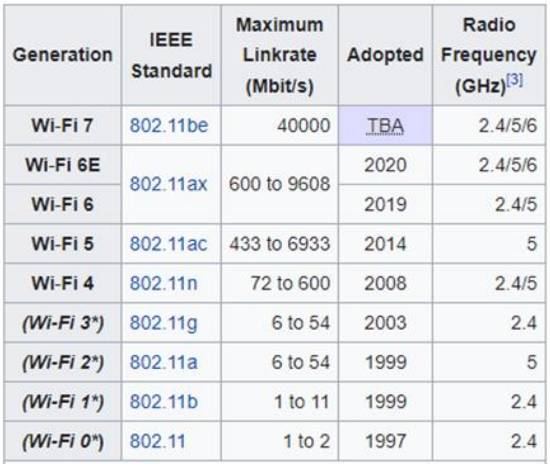
4.1 Design impact of relevant standards

The primary reason why industry standards were established in the first place is to ensure that various products will be compatible with one another and allow customers to safely mix and match similar products from different brands. Although there is a plethora of different industry standards, the ones we will be primarily focusing on is the wireless standard for WiFi, power standards that have been established by various safety commissions, and water quality standards as they are vital to ensuring the proper growth of the plants.

4.1.1 Wireless Standard

Our system will fully communicate to our Web app through WiFi. WiFi, or Wireless Fidelity is a protocol for cableless connection to the Internet. We decided that WiFi would be the most reliable and safe way to connect to our MCU because it avoids the need of cable and having an Ethernet compatible board.

We will follow the Wireless Standard dictated by the Institute of Electrical and Electronics Engineers (IEEE), which are shown below in **Table 4.1a.** The current and most recently updated standard is 802.11ax.



Generation	IEEE Standard	Maximum Linkrate (Mbit/s)	Adopted	Radio Frequency (GHz)[3]
Wi-Fi 7	802.11be	40000	TBA	2.4/5/6
Wi-Fi 6E	802.11ax	600 to 9608	2020	2.4/5/6
Wi-Fi 6			2019	2.4/5
Wi-Fi 5	802.11ac	433 to 6933	2014	5
Wi-Fi 4	802.11n	72 to 600	2008	2.4/5
(Wi-Fi 3*)	802.11g	6 to 54	2003	2.4
(Wi-Fi 2*)	802.11a	6 to 54	1999	5
(Wi-Fi 1*)	802.11b	1 to 11	1999	2.4
(Wi-Fi 0*)	802.11	1 to 2	1997	2.4

Table 4.1a

Within the used Standard some signal modulation is allowed to better suit our needs. Our requirements call for a low bandwidth connection due to the small amount of information we are exchanging with our Web server. These small packages would allow us to maximize our range if needed.

4.1.2 Power Standard

We will adhere to the safety and regulations of the American National Standard for Electric Power Systems and Equipment dictated by the American National Standards Institute, which currently operates on the 120V range.

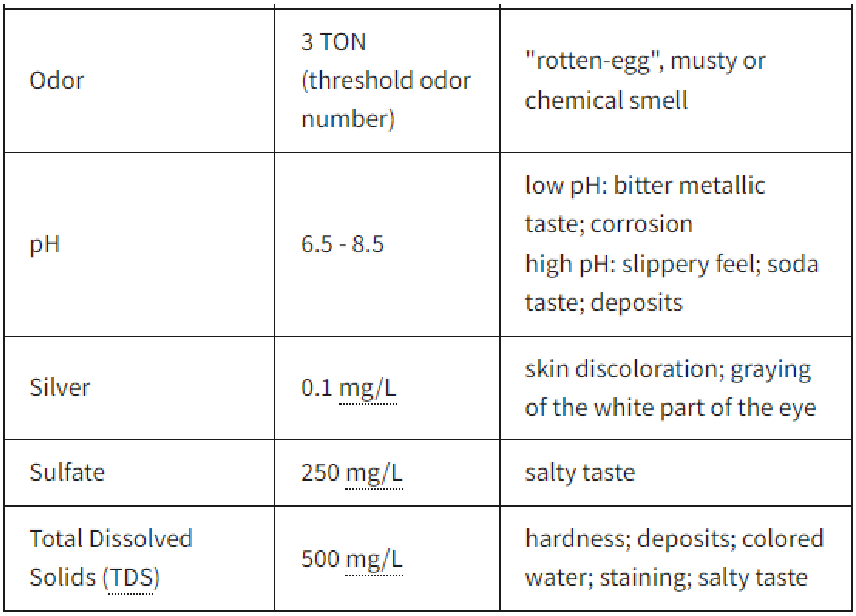
Also, we would be using Type B 5–15 U.S. 3 pin connector that is standardized by NEMA, the National Electrical Manufacturers Association. This connector is designed to operate at 15

Amps/125 Volts. We feel safer with this choice because by having a longer ground pin the user is grounded before the power is connected.

This system would need to be modified if it was to adhere to international standards if the product were to be used outside North America.

4.1.3 Water Quality Standards

Our system will be using regular tap water, which is why it is critical that we adhere to the Water Quality Standards enforced by the EPA. A system working with poor quality water for an extended period of time could damage our electronic components in the long term. Additionally, we would be harming the quality of the food plants produce and in extreme cases killing our harvests. Certain EPA standards as seen on **Table 4.1b** can be tested using already available equipment, visually or olfactory.



Odor	3 TON (threshold odor number)	"rotten-egg", musty or chemical smell
pH	6.5 - 8.5	low pH: bitter metallic taste; corrosion high pH: slippery feel; soda taste; deposits
Silver	0.1 mg/L	skin discoloration; graying of the white part of the eye
Sulfate	250 mg/L	salty taste
Total Dissolved Solids (TDS)	500 mg/L	hardness; deposits; colored water; staining; salty taste

Table 4.1b

4.2 Realistic Design Constraints

As with any project that is being designed and created with the intention to be introduced to the open market, it is imperative to review any and all constraints that will affect the project's design process. The principal constraints that are of concern to our team's focus are the economic and time constraints. This is due to the fact that our team did not secure a sponsor to help assist with the budget of our project and there is a hard deadline for the project's completion. Environmental, social, and political constraints were also reviewed to ensure that our final product will not have to be concerned with having a negative public perception. We finally address the ethical, health, and safety concerns associated with building our design since these factors can have a major impact on the population at large.

4.2.1 Economic and Time constraints

As Engineers we must constantly move forward flexibly working around many different constraints. One major constraint that we will be working through in this project is time and economic constraints. We have a strict amount of time to complete projects for senior design as students. For our hydroponic system we need to complete our design and start putting together our system even earlier than other projects as we need much more time to test our product and make sure it works. We need to account for the growing time of the plant and test all the features of our hydroponic growth as they happen naturally. Individually we also have other time constraints such as work, classes, family and other such responsibilities that we will need to work around.

In total this project must be a complete product in two semesters according to ABET requirements. Within this time, we must make sure all design considerations are scaled and completed appropriately during the first semester and the build and testing possible to complete in the second semester. If we are unable to complete our design in the first semester it will lead to many complications as we move towards our second building semester. We will need our custom PCB designed and shipped late first semester or early second semester as there are many time and economic constraints around it. Focusing on the time constraints currently the world is going through a semiconductor shortage making the current PCB constructions a lot slower during this time period. If this custom PCB is not ordered early, it may take far too long to arrive and implement in our project, not to mention shipping times which would make the process even longer.

In order to avoid the impact of shipping delays we must focus on ordering all parts early allowing us plenty of time to assemble the product. We must also schedule in person and online meetings allowing us to organize the workflow but also leading to additional cost in fuel and time driving. There will also be a large testing period where we need to test all of our sensors and pumps individually and together. Sadly, many of us work so we will have to carefully plan our meeting and make sure we are still working on the project individually according to the plan. We need to make sure we are not being too overambitious with each feature and make sure we have allotted time to implement each feature.

Since we are still students, we meet a lot of economic constraints as well. One of our goals for the project is to make sure that this automated hydroponic system includes many of the features that can be found on the market while keeping the budget as low as possible. Prices on the parts may fluctuate for each component depending on the supplier so we must carefully decide as a group our price range for each component. Since one of us owns a 3D printer it greatly helps with the building of some structural components. We also may own some parts that we can reuse from other projects such as lights, microcontrollers, webcams, timers, and sensors that can greatly relieve the burden on our budget. Some of us also work engineering jobs that allow us to get an estimate for some of the items on the project. For this project we have filed to find a sponsor for the project so all of the budget for the project will be coming out of the group's pockets split between 4 different people. This helps ease the cost of materials for a single person.

4.2.2 Environmental, Social, and Political constraints

As hydroponics continues to revamp the way society grows its food, it will only become more popular amongst the general populations of every country. This is especially true in areas that experience water shortages on an annual basis since hydroponics utilizes minimal amounts of water when compared to traditional growing methods using soil. We believe that a very minimal amount of society would be upset with the increased usage of this revolutionary technique as most would be impressed with the fact that the environmental impact of hydroponics is essentially the same when compared to the cultivation of plants within the soil of an open field. With the major difference being that hydroponics tends to provide much higher yields when compared to traditional methods. Some may argue that traditional farmers may lose work due to an increased use of hydroponic systems but this argument is somewhat exaggerated. There are several different hydroponic techniques that can be incorporated on a higher commercial scale. Farmers would simply have to restructure their own infrastructure to accommodate this kind of change. Additionally, with the human population that continues to increase year over year, the ever-growing need for food will not slow down.

In terms of environmental constraints, the only negative impact that we can consider would be the additional consumption in energy. Further research would have to be done to calculate the average power consumption of commercial hydroponic systems. Otherwise, hydroponics appears to be environment-friendly. If anything, it's the current state of our own environment that can ruin the crops of many farmers. As global warming continues to worsen, farmers will be unable to produce the yields necessary to maintain their own livelihoods. Traditional farming relies heavily on the right weather and environmental conditions that are conducive to plant growth. Especially since the nutrients that are normally provided directly to a plant in a hydroponic system must be found by the plant's roots by extending out in every direction of the soil. We also do not believe that there would be many political constraints against hydroponics. Most politicians care most about how they are perceived in the public eye. So as long as the public is willing to stand behind hydroponics for the betterment of society, the politicians will fall in line. The only constraints that would be applied to hydroponics is some form of taxation.

4.2.3 Ethical, Health, and Safety constraints

For our project, since we are trying to make an automated hydroponic system that creates produce for people to consume, we must keep many ethical, health, and safety measures in mind. As engineers we need to design our hydroponics plant to mitigate as many safety risks as possible. One such risk is fire hazards as we are working with many electronics alongside water there are many electrical hookups wired alongside the water tank. So, we must properly organize and design our power system around making sure there is no leakage in any tank. We also needed to have precautions against power outages as well, since a loss of power and light can lead to oxygen deprivation in the plant. What we plan to do is have a separate APC for back up power and protection from power surges.

Another constraint we may need to worry about is food safety. We must make sure there are no pathogens or harmful bacteria growing in the water and after harvest the herbs must be properly washed. The water in the tub must also be treated with a bio fungicide in order to prevent the pathogen. Another precaution we will take is preventing light from reaching the water, and without light bacteria will decrease in growth rate. This precaution doubles as a plant disease prevention technique too. If one plant gets infected with plant disease it can risk affecting the whole system. Since the plants are planted in a water media rather than soil or other hard media the water will quickly spread disease directly to all the plants in the media leading to a loss of all the crops. To prevent the loss of our crop we will take precautions by routinely cleaning, sanitizing and checking on the system. A big deterrent against plant disease will be our webcam, with 24/7 monitoring the system will be able to quickly pick up any discoloration and send an alert to the owner of the unit. Our system will also have the ability to pick up any pH or nutrient anomalies in the water, allowing the user to be alerted to any sign of root rot. In regards to ethical constraints this plant will be grown in a grow tent isolated from the outside world while being completely contained. The hydroponics system should only be able to affect the local environment of the plant rather than any consequential impacts to the outside world. The only pollution this system will give off would be noise pollution of the fans but if placed somewhere isolated is of no significance.

4.2.4 Manufacturability and Sustainability Constraints

Manufacturing Automated Hydroponic systems from the start to the end of production is a complex and time-consuming process. Especially at this time with COVID it makes it more complicated with the shortage of semiconductor, disruptions to the supply chain can cause some crucial components to have long lead time. This will cause delays for on-time deliveries, this will not satisfy your customer.

For parts with long lead time there are website resources to find similar parts with same specification, this will help manufacturing keep up and ship products out on time. In engineering there are engineers that specialize in the obsolescence part, the part that is no longer available they will look at the data sheet and find exact or similar to the part that is not available.

Create processes and plans for manufacturing the hydroponic system. Plans for circuit card assembly and process for flying probe test after circuit has been assembled to ensure analog signature analysis and short/open, this is classified as an in circuit. If there 100 CCA is assembled, fly probe test 5% of the 100 CCA to ensure no short/open are present in the circuit card.

Safety guideline for the consumers. Handling of the nutrient and pH product, these products can cause skin irritation, use gloves while handling the product. Creating a process plan for consumers on installation and setup of the Hydroponic System, the process written should be simple so any consumer can follow.

All the sensors, such as pH, TDS, Conductivity, humidity, water level for the Hydroponic System are readily available on the market. The vendors are very competitive, this can help drive down the cost of the hydroponic system, in return the customer will get a reliable product at less cost. This creates a competitive edge and an advantage over our competitor, in return new customers will want to use our product. As production/business of the system grows, it will need to account for supply cost. The sensors can be designed and produced, this will cost. And also can sell the sensor on the market to other consumers. This will give us an advantage/edge over our competitors.

Due to some large parts such a water container structure and the various sensors to be handled as fragile parts. This will require instruction for shipping on how to package and handle all the material for the Hydroponic System

Sustainability will be through all the process plans and training.

5. Project Hardware and Software Design Details

This chapter covers the hardware and software designs that were implemented in our project. The first portion of this will cover our early block diagrams used from the “Divide and Conquer” paper to better illustrate our initial design ideas as well as the distribution of responsibilities for each project member. It will then further discuss the various subsystems that are required to make our automated hydroponic system functional and fully automated. These subsystems include the microcontroller unit, the main power system, as well as the pumps and sensors. Finally, we will discuss the primary design of our software.

5.1 Initial Design Architectures and Related Diagrams

The power system diagram in **Figure 5.1a** below represents the general idea of how we plan on organizing each of the individual components. A 15V AC to DC converter is first needed before proceeding to either the 12V or 5V DC regulators. The 12V DC regulator is meant to deliver the appropriate power to either the air and/or water pump. The 5V regulator is meant to deliver the appropriate power to the sensors being used to monitor the surrounding environment as well as the nutrient and pH levels of the water solution in the reservoir. Those sensors then send their output signal to an analog to digital converter that in turn sends the final results of the sensors to the microcontroller unit.

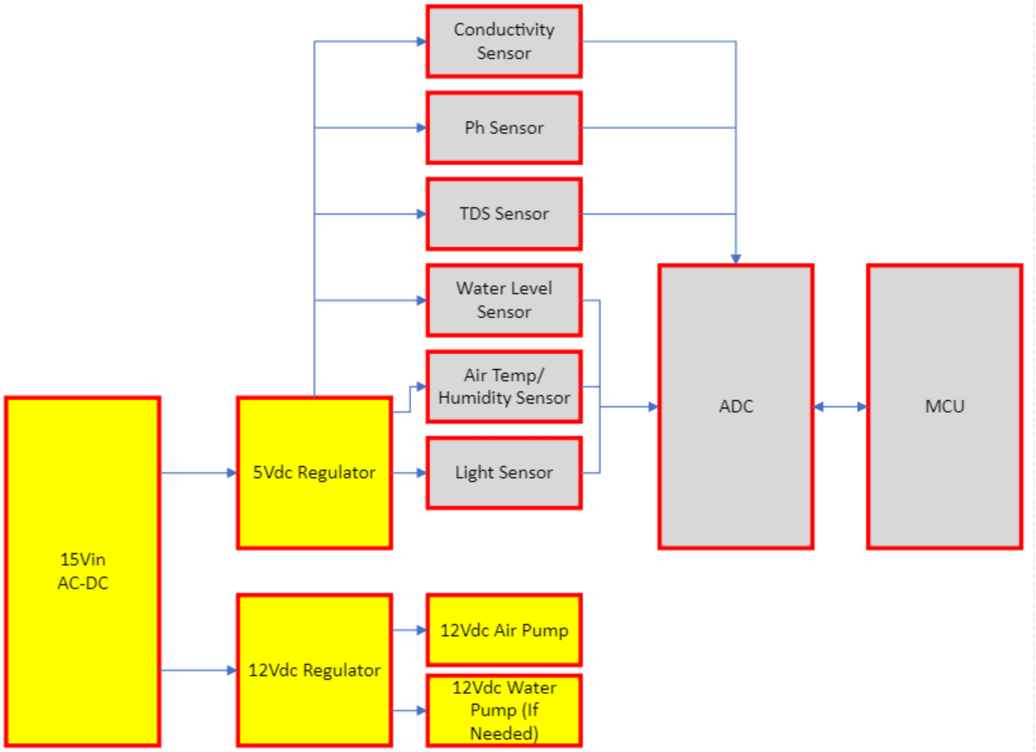
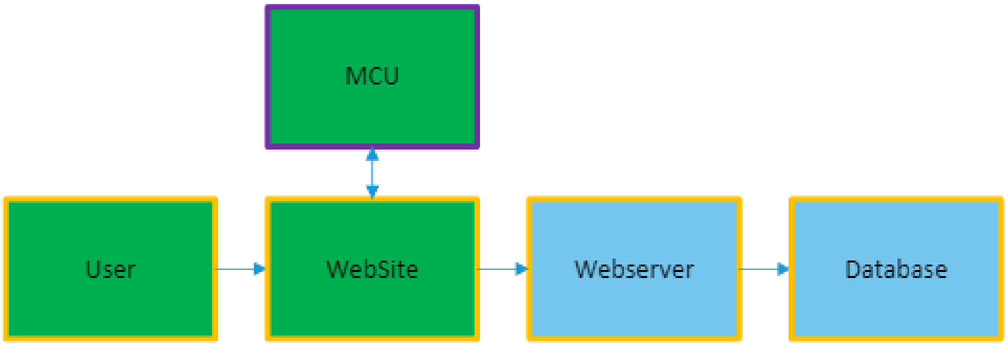


Figure 5.1a

The software and hardware diagrams in **Figure 5.1b** below represent the remaining "flow" of the other components. Notice that the hardware diagram picks up where the power system diagram left off and shows where the signals from the microcontroller unit are sent to.



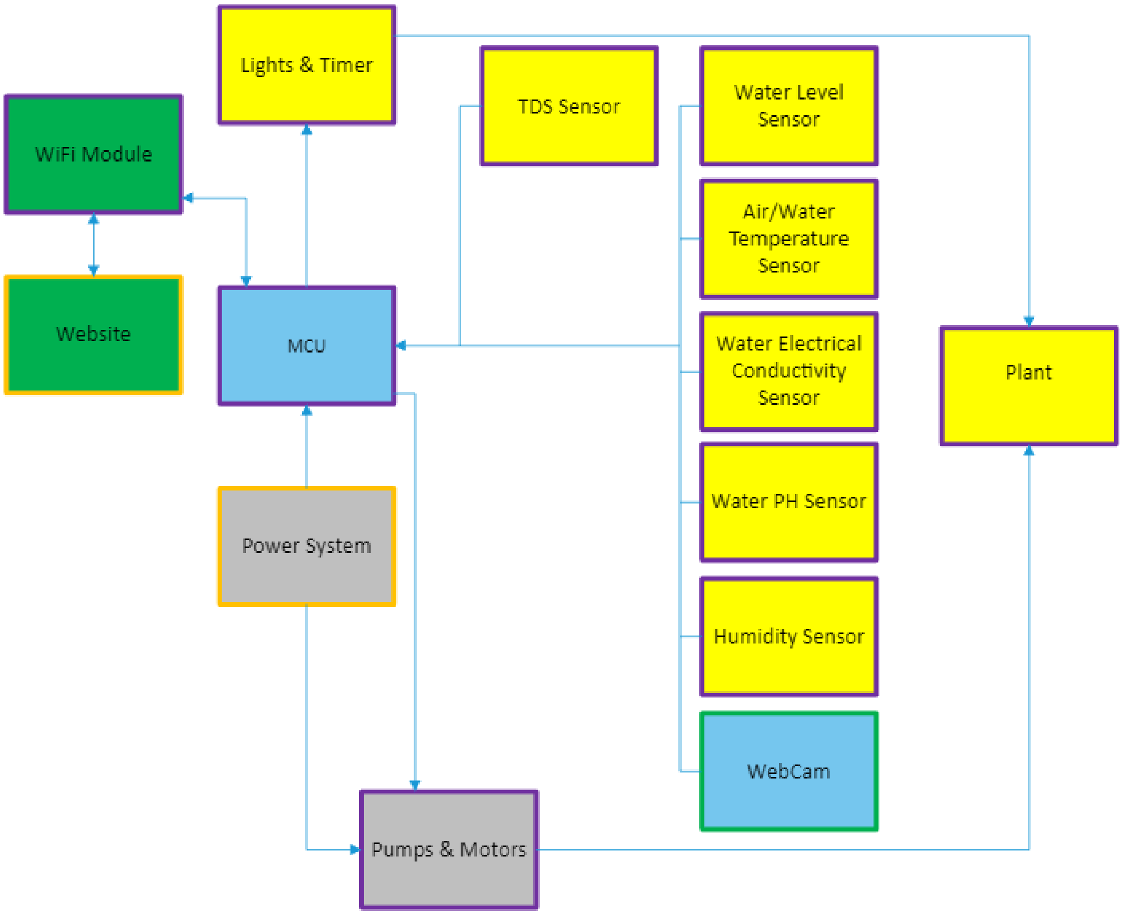


Figure 5.1b

5.2 Controller Subsystem

This section covers the primary components that makeup our controller subsystem and is essentially the heart of the automation for the deep water culture hydroponic system. It covers the central processing unit, main memory, the analog to digital converter, and the general purpose input/output pins that are being used.

5.2.1 Central Processing Unit

Our board comes equipped with a Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, which is enough to handle the workload. We believe that the most resource intensive task will be the image processing coming from the Camera that will be used for discoloration/plant growth. This processor follows the ARM architecture. Additionally, it allows for Core Related interrupts, allowing each of the 4 cores to handle interrupts individually, which will be quite useful when trying to lower the power consumption.

5.2.2 Memory

Our MCU comes equipped with 1GB of SDRAM that is Low-Power Double Data Rate, which consumes less power than DDR memories.

This memory will store our code and also our collected data before being sent to the Web Server. We want to ensure complete self-reliance on our system in the case of the Web server failing. To accomplish this the data will be stored locally inside our memory and work in redundancy with the Web server as backup.

The ATMEGA328 Nanos memory for Nano is 32Kb. There is also a pre-installed bootloader on the device, which takes a flash memory of 2kb. SRAM memory is 2kb and is summarized in **Table 5.2a** shown below.

Microcontroller:	ATMEGA328
Architecture:	AVR
Flash Memory:	32 KB of which 2 KB used by bootloader
SRAM:	2 KB
EEPROM:	1 KB

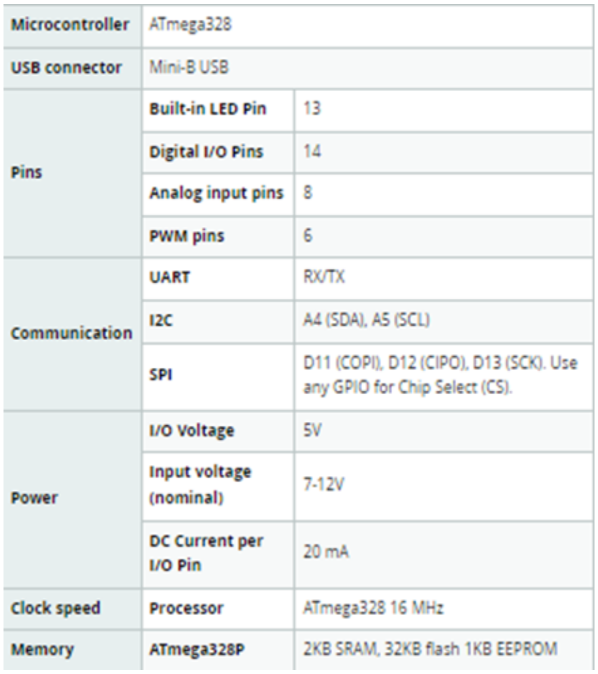
Table 5.2a

5.2.3 Analog to Digital Converter

An AD converter is used to convert an analog signal like voltage to digital signal so it can be read and processed by a microcontroller. Some microcontrollers have built-in AD converters. The Raspberry Pi 4 B does not have an AD built into the microcontroller. The ADS1015 is analog to digital with a 12 bit 4 channel. If precision is needed, the ADS1115 is a higher precision 16-bit ADC with 4 channels. Using the AD converter to get the analog signal from the sensors and route the digital output to the Raspberry Pi 4 microcontroller to process the AD data.

The ATMEGA328P microcontroller has an eight channel analog-to-digital conversion (ADC) built into the microcontroller and it is capable of converting an analog voltage into a 10-bit number from 0 to 1023. The inputs to the ADC on the microcontroller pins have connections A0 through A7. The ADC can convert signals at an approximate rate of 15 kSPS (samples per second), the

ATMEGA328 can only convert one channel at a time. The ADC circuit additionally needs to have a clock signal. The clock is generated internally from the same clock that is used to run the microcontroller. The CPU clock is too fast so the microcontroller includes an adjustable "prescaler" to divide the CPU clock down to a more appropriate rate of speed. Interfacing the ADC, the software uses a group of registers. **Figure 5.2a** and **Figure 5.2b** summarize this.



Microcontroller	ATmega328	
USB connector	Mini-B USB	
Pins	**Built-in LED Pin**	13
	Digital I/O Pins	14
	Analog input pins	8
	PWM pins	6
Communication	**UART**	RX/TX
	I2C	A4 (SDA), A5 (SCL)
	SPI	D11 (COPI), D12 (CIPO), D13 (SCK). Use any GPIO for Chip Select (CS).
Power	**I/O Voltage**	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
Clock speed	**Processor**	ATmega328 16 MHz
Memory	**ATmega328P**	2KB SRAM, 32KB flash 1KB EEPROM

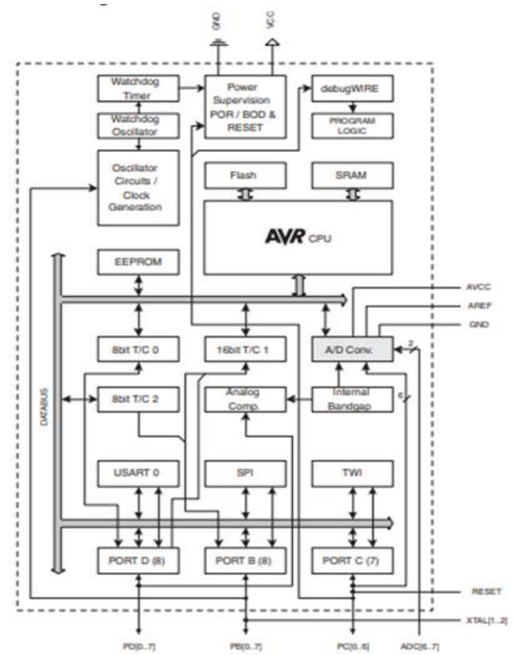


Figure 5.2a

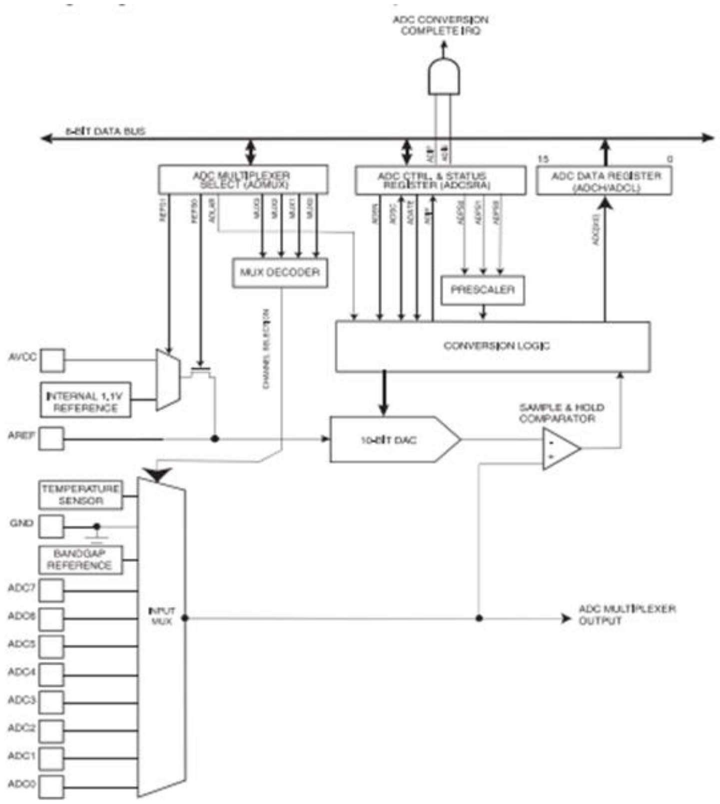


Figure 5.2b ATMEGA328 Internal ADC Flowchart

5.2.4 General Purpose I/O

Our CPU has 40 GPIO pins that will be used to receive the input from the sensors and transmit the output to the pump, lights, etc. There are multiple GPIO pins on the microcontroller to be used for multiple interaction with simultaneous application. The pins can be programmed as input, where data is routed from an external source such as sensor signal to the microcontroller and Output GPIOs, can be transmitted to outside devices. The GPIOs can be configured as interrupt lines for a CPU to signal immediate processing of input lines. A summary is provided in **Table 5.2b** shown below.

Digital I/O Pins:	22 (6 of which are PWM)
PWM Output:	6
Analog-in Pins (ADC):	8
DC Current Per I/O Pins:	40 mA (I/O Pins)

Table 5.2b

5.3 Power Subsystem

Total Power Consumption of all Subsystem:

AC to DC converter will be 120 Vac converted to 12Vdc with 4.1 Amp rating. The 12 Vdc will be used to power up the project design circuit CCA. Add all the device power consumption added up from the design CCA , it will be close to 2Amps. This power consumption is an approximate, the operating current will go up as devices are added to meet the Hydroponic system design.

Sensor Power Consumption

The pH detector sensor: operating current 5-10mA input voltage 5Vdc. Water liquid level: operating current 12mA input voltage 5Vdc. Water conductivity sensor: Operating current 3-6mA input 5Vdc, Water Sensor (TDS) Operating current 3-6mA input 5Vdc.

Pumps Power Consumption

The water pump: operating current 300mA input voltage 12Vdc. The pH and Nutrient pump operating current 80mA each input voltage 12Vdc. The pumps will utilize the AC to DC converter 12Vdc to run the pumps. The power consumption for the pumps added up to be around 460mA, the 12Vdc is rated for 4.1Amps.

DC To DC Converter

The dc to dc converter takes 12Vdc input and 5Vdc output, operating current 200mA. There ceramic capacitor added at the input and output of Dc to DC converter for filtering. This 5Vdc rail will supply all the sensors Water Conductivity, pH, TDS, Water Level, and humidity. All the sensor power consumption added to be around 100mA, this well in range of the 200mA of the 5Vdc converter. Looking ahead we might have to use two 5V dc to dc converters to supply enough current to all the devices or find a 5Vdc regulator with a higher current rating.

120Vac Wall Outlet

The 120 AC wall outlet will be used for the lighting system for the plants, we will be utilizing the 120 AC wall outlet, also will be utilizing a timer to keep the light on for the duration it needs to be on and off for the duration when the plants does not require light. The Air pump, and Raspberry pi will also require the 120Vac wall outlet.

120Vac Back up Supply

Uninterruptible power supply is an electrical device that provides emergency power to all the devices that keep all the equipment running for a short time, near instantaneous protection from energy that are in the batteries, supercapacitors. This will be used for backup power, also incorporated into the hydroponic system, if there should be a power outage. The hydroponic system will have a built-in circuit that will detect this failure and warn or prompt the user via web server or email of a power outage.

5.4 Pump & Sensor Subsystem

The peristaltic pumps and sensors are a crucial subsystem that is meant to periodically test the chemistry of the water solution. If either the pH, electrical conductivity, or total dissolved solids values do not fall within the ranges established by the user, a signal should automatically be sent from the microcontroller to the peristaltic pumps to administer the appropriate type and amount of solution to rebalance the water's chemistry. Maintaining the correct water chemistry entails dispensing minute amounts of acid to lower the pH level, base to raise the pH level, as well as two different nutrient solutions for raising the electrical conductivity of the water. For this reason, a total of four peristaltic pumps will be required in order to ensure that each of the solutions are added independently from one another. We feel that when the chemistry levels are within range, it may only be necessary to take a sample reading every hour as the likelihood of a drastic change occurring in less time is remote. However, once it has become necessary to rebalance the water's chemistry by adding the appropriate solution, sampling should occur every few minutes until the readings have returned to their normal levels.

Even though many of the recorded sensor readings will be available via a smart device, an LCD screen will still be used in conjunction with the peristaltic pumps and sensors to review the results of the measurements being recorded by the sensors. This LCD screen along with the sensors and peristaltic pumps will have to be connected to the same pins of the microcontroller in order to provide communication amongst the devices. Additionally, it will be important to thoroughly test the peristaltic pumps to ensure that we are able to administer the desired amount of solution to our main water reservoir. The peristaltic pumps that we are considering incorporating into our project have flow rates that range from 0.04 liters per minute to 0.07 liters per minute. Additionally, thorough testing must be done to each of the sensors to confirm that each one is in working order. Since every sensor has not been ordered and delivered as of yet, we were only able to complete preliminary testing on the two sensors that are already owned. The water level sensor and the air temperature/humidity sensor.

Since both sensors were part of the ELEGOO electronics kit, a free Arduino IDE by Genuino was available as a free download online and a simple code in C language was provided as well that would allow for a brief test to ensure that both the sensors would operate correctly and provide the measured results to the system. The water level sensor has several parallel wires across the board that sense the liquid level. The sensor will return analog values read by the Arduino, where a value of 0 represents the lowest measurement and 1023 represents the maximum water level. A quick initial test showed that the sensor is in working order and provides results almost instantaneously.

For a deep water culture hydroponic system, it’s recommended that approximately 1.5 inches of root are exposed above the water. Since we have yet to make an attempt at setting up the main components of our project, we have not yet determined what analog values of the water level sensor will represent the moment that the water level has reached its minimum height of the reservoir. Our goal is to have the microcontroller unit automatically turn on the water pump before the water level has reached its critical point and then turn off the water pump once it has regained the appropriate water level.

For the air temperature and humidity sensor, the temperature readings are provided in degree Celsius and the humidity is shown as a percentage. A quick initial test shows that the sensor is in working order. The temperature results closely matched the readings of the thermostat in the room but I could not compare the humidity readings to anything else. To ensure that the temperature and humidity readings experienced no issues increasing/decreasing, I put a closed fist around the sensor for a few moments and watched the values steadily increase. After removing my hand, the values then started to gradually decrease. Our goal is to have the microcontroller unit automatically turn on the exhaust fan within the grow tent before the surrounding temperature has reached the maximum allowed temperature for the plants and then turn off the fan once it has returned to the desired temperature.

5.5 Software Design

For this section we will split the design of our system by separating it into a web stack, web server, database, UI, and website. On the software side Nathan To and Leandro Alepuz will be the main members working on this side. For microcontroller and power supply integration the main members will be Danny Nguyen and Edwin Rivera. These members may swap, and their roles may overlap as necessary since all these systems must be integrated flawlessly. Our initial flow diagram shows the flow and features of the system from the user to the control unit. This diagram shows the custom PCB intaking all the sensor data and relaying the data to the MCU which then sends that data over the website for display. The code in the MCU should also relay pump and light commands over to the Custom PCB based on that data, adjusting the various attributes of the hydroponic system. The code with the various thresholds can then be edited by the user from the website.

5.5.1 Webstack

For our application we want to focus on a dynamic website that can be accessed from both a desktop and mobile phone. The website should be able to modify itself to fit mobile screens. This website will allow each user to create its own account for security reasons and will be to modify threshold levels for the pump and access the sensor data and its history from the database. For this project we will need to determine which stack will best suit our application and give us the best framework to construct our website. We had many concerns given the application of our project of which web stack would be optimal while being familiar and streamlined to implement.

In our group we had to research the LAMP stack vs the MERN, MEVN and determine which one would be the best for our usage. Comparing the two stacks we then were able to narrow it down to the MERN and MEVN stacks. First of all the LAMP stack required the usage of multiple different languages and scripts to implement but was fairly easy to learn and pick up. Some of our members have had experience with this stack before. The MERN and MEVN stacks are even more streamlined though only using JavaScript throughout all parts of the stack. The MERN and MEVN stack also use a non-relational database. While the mySQL the relational database that the LAMP stack uses is viable, the NoSQL non-relational database that MERN and MEVN use is much better suited for our project. MongoDB has much better scalability to speed comparisons as it intakes more data and document storage is much better suited for storing sensor data than the tabular storage method mySQL uses. Regarding the webserver the LAMP Apache web servers and pHP require a decent amount of initial setup and management while the Node.js web servers are very quick to pick up and intuitive. Lastly, comparing the libraries pHP and reack.js we felt they have a near equal value in the quantity of library modules and large communities, but we lean toward react.js because of our experience with it and considering it is written in JavaScript.

Once we've determined which of the two main stacks we plan to use, now we need to determine which of the stacks we were going to use between MERN and MEVN. Although some of our members have previous experience with Reach.js libraries we are leaning towards the MEVN stack using Vue.js in our project. Vue has the benefit of being very easy to learn and uses HTML templates apart from JSX. Vue also has a very balanced combination of both third party and prebuilt tools by its core team. Therefore, we will be using the MEVN stack of MongoDB, Express, React and Node.js as our web stack. Vue.js was chosen since it is very optimal for small projects and has an easier learning curve for picking it up. Vue also has many mobile templates for cross platforming making it easy to adapt to a mobile device. The documentation is also very good and written in a very detailed manner giving a good allowing us to easily look up and troubleshoot problems. Vue.js is a bit more niche of a framework making it very lucrative to learn, and we feel that this project is a great opportunity to explore different frameworks. Below the webstack we will use is shown.

Figure 5.5a

5.5.2 Web Server

Continuing the comparison between the MERN and LAMP stacks. Our group has experience with both Node.js and Apache HTTP server, but feel more comfortable with Node.js. We have worked with Node.js more recently in other courses as well so there will be less refreshing to do once we start on the project. Node.js also uses standard JavaScript for development which is one of the most prevalent languages used for web development across the entire software stack in the world. Node.js offers a large amount of scalability in both vertical and horizontal scaling allowing the allocation of more resources to single nodes as it expands. This technology is also known for its high performance, using Google's V8 JavaScript engine. The engine allows the JavaScript code to be compiled directly into machine code. Making it faster and easier to implement. The speed of the code is enhanced even further by the runtime environment because it supports the non-blocking I/O operations. With this runtime environment Node.js can process several requests concurrently. This system can handle concurrency more efficiently and faster than Ruby or Python. Any incoming requests are queued up and implemented quickly and systematically. In conclusion, Node.js is a flexible, fast and efficient technology that works great for our application. Below are some of the other benefits of using Node.js

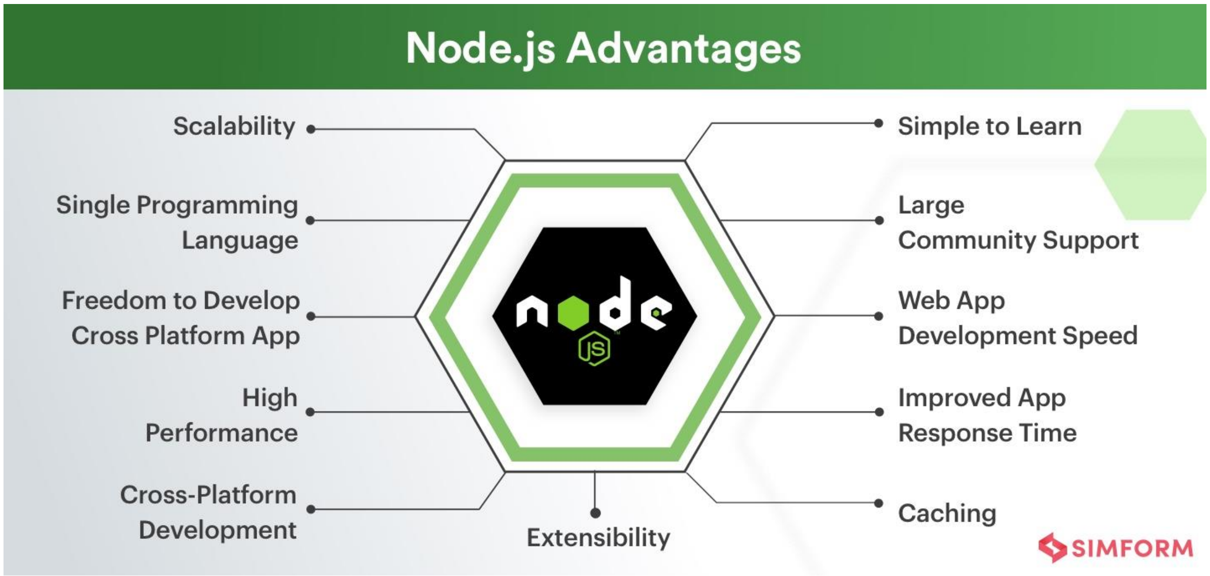


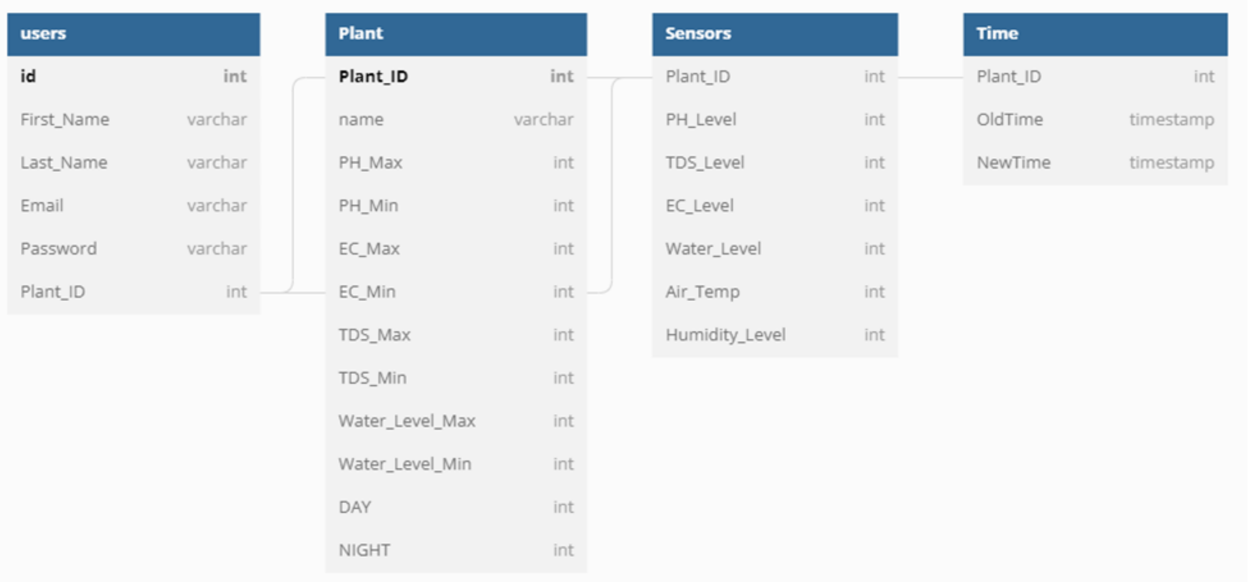
Figure 5.5b

In order to connect our microcontroller to our web server we will need to interact with the system though a mobile device or computer. The plan is to use our microcontroller and a web server and host it on whatever network our microcontroller is connected to. Once the Microcontroller is producing the web server it will allow other devices to connect to it on the network and access its various sensor data and other features. Connecting the wifi module to the

network is only the first step, once this is complete and the webserver is set up the hydroponics system can then send and receive information from the website to the microcontroller. The user will then have access to all the features of the hydroponics system allowing them to check up on the plant at any time.

5.5.3 Database

After deep consideration between the two databases mySQL a relational database and MongoDB a non-relational database our group decided it would be more beneficial to go with MongoDB. MongoDB was chosen because we are fairly familiar with it from previous projects so we can quickly and easily set up its API. MongoDB has a small learning curve for our group, is highly scalable and able to take in a lot of data. With its fast update speeds and schema less document database we found it ideal for storing our data. The main reason why we decided on MongoDB was its compatibility with Node.js and the fact that it can be accessed through JavaScript helper nodes. Below is a basic database entity relationship diagram that we will be using for our hydroponic system.



users	
id	**int**
First_Name	varchar
Last_Name	varchar
Email	varchar
Password	varchar
Plant_ID	int

Plant	
Plant_ID	**int**
name	varchar
PH_Max	int
PH_Min	int
EC_Max	int
EC_Min	int
TDS_Max	int
TDS_Min	int
Water_Level_Max	int
Water_Level_Min	int
DAY	int
NIGHT	int

Sensors	
Plant_ID	int
PH_Level	int
TDS_Level	int
EC_Level	int
Water_Level	int
Air_Temp	int
Humidity_Level	int

Time	
Plant_ID	int
OldTime	timestamp
NewTime	timestamp

Figure 5.5c

This is a simple diagram that can be applied to multiple different plants. We can create different plant profiles and scale to however many plants as we would like. Though for now and in future testing we will be sticking to one plant type. With mongoDB we should be able to quickly search and query through the different documents to update the plant data. Plant data can also be quickly collected and stored in our sensors documents where it can be quickly provided for the user to view.

As the diagram shows we will be creating users with different access levels so lower end users cannot do anything that will break the system. The document will include email, name and passwords as well as the plant ID. The Plant ID will be used to find all the sensor data and thresholding for that specific plant. There is also a category to specify the user's plant as well as its survival information.

A new document will need to be created for each sensor entry from our hydroponic system. Each time we create a new sensor document a Time document will be created as well logging each and every update of the sensors in order to create a graph in the UI. The oldTime represents the first sensor data received and the newTime represents the most recent timestamp stored to determine when the most recent testing was done. We can use this document to determine if the system has lost power or connection to the internet. The user can then be notified if sensor data has not been recorded for a long period of time.

5.5.4 Website

The URL of our website will be autohydroponic.com subject to change. The website will consist of several different pages such as the login, register, home, sensor data, webcam, profile and plant settings. There will be a navigation pane across the top of the website with links to all the different pages. The mobile browser and the web browser should be as similar as possible containing all the same information and access. When the webpage is initially accessed the user will be led to the front page with a welcome screen and login area. Here the user will input their email and password, if the user does not have an account or has forgotten their password there will be a link to a forget password page and registration page.

Once the user has filled in their data, they will be led to the user dashboard. This dashboard will display all the current sensor data of the plant and have a live feed of the plant. The dashboard should also have links where the user can access his account settings, plant settings, about us, and plant history. There will also be alerts on the main dashboard if there are any problems with the plant or if the sensors are not in the required range. In the plant history page, you can view and search old sensor data in different forms such as in a graph or list. In the calibration page there will be buttons to manually activate pumps, stop pumps, and refill the water reservoir. The MCU will transmit sensor data from the custom PCB to the website. Connection with the website's data will update all the old sensor information in the hydroponic system. Below is a table that gives us an overview of each page.

Pages	Function
Login	Users can input their email and password into their intended fields in order to log into the hydroponics website.
Registration	Users will need to input their email, password, first name, and last name to be stored in the database and make an account. The user will then receive a confirmation email to their email address.
Forgot Password	If the user does not remember their password the user can input their email address, and a reset password email will be sent to their inbox.
Dashboard	This will be the welcome screen seen after a user logs in. The dashboard will be filled with the current sensor data of the plant as well as a live feed of the plant. There will also be links to the other essential pages.

User Settings	In this page the user will be able to change their email address, and password. The user will also be able to logout or delete their account from here.
Plant Settings	In this page the user will be able to change the threshold values of the pH, TDS, EC, water level and the day night values. The user will be able to adjust when the automatic water balancing happens and when alerts go off.
Plant History	Enables the user to observe past sensor data of the plant. The user can view the sensor data in a graph, list view, and filter content by attribute.
Calibration	This page is for when the user needs to manually operate on the plant. There will be options to pause different operations of the plant for refilling and harvest. There will also be manually adjusted pumps and lights.
About Us	This page is a pure informational page about the project and will offer troubleshooting information and documentation

Table 5.5a

The code of these parts will be split into two different pages: the backend and the front end. The back end will transfer data to the second front end page which will show data on the web browser. We will also have to have precautions in place in order to prevent others from accessing and modifying these two pages. The backend for the raspberry pi will have code to transmit data to the database using a post call. MongoDB our NoSQL library will connect to the database and update all sensor information. The backend for the website will then pull the data from the database using a get call that will verify and format the data in JavaScript. The back end of the website will also include the post function for changing the values of the sensor ranges for the MCU. We considered different methods and languages for our database and API language but decided on JavaScript to keep things consistent throughout the backend and frontend. In the next figure we will show how the backend of the website will function.

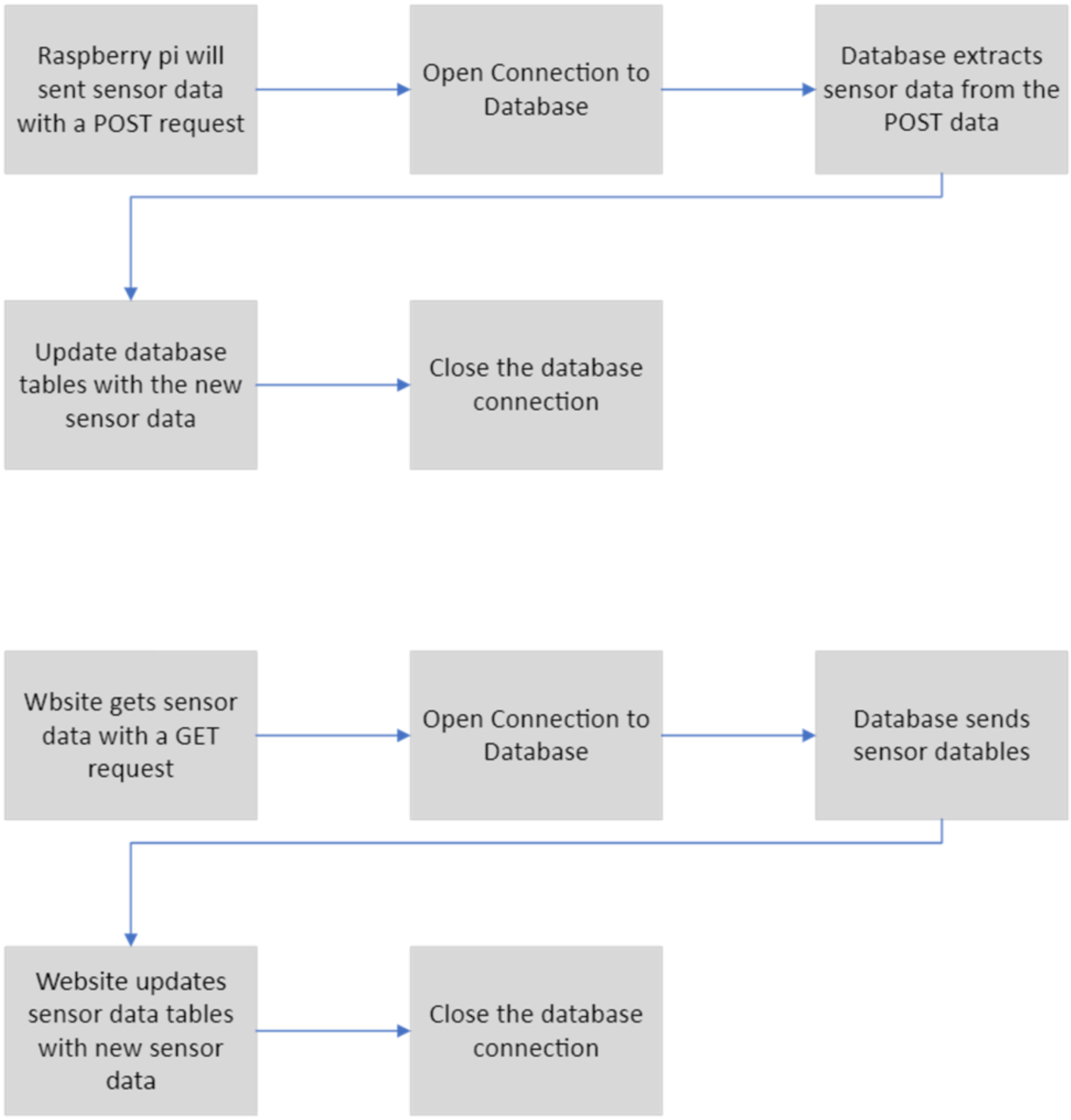


Figure 5.5d

As shown the back end for this project will be fairly simple in structure although more functions will possibly be added in the future as we wish to add more functionality to the project. As seen the main purpose of the code is to populate the database and error handling on the MCU. From the website side the main code will receive data to populate the tables in the front end. There will be quite a lot of processing of this information into graphs as the main function of the website is for the user to monitor the plant and its current state. This website will have to have an intuitive UI that will be easy to use, and report possible problems to the plant. There will also be an administration page that will be used for testing and to determine if the data received is valid. As long as the database connection is functioning properly, we can troubleshoot any other problems with error handling.

On the security side of the website we will consider that there will be multiple users wanting to use the website so we will make two levels of clearance. The first level will be the administrative level with the ability to modify the sensor ranges of each sensor. Another function of the administrative user is the ability to manually start and stop pumps for testing purposes and shut down the system. The administrative user will also be allowed to change the access level of other users. The lower end user has basic access to all the sensor data and the live feed of the grow without the ability to change anything. the lower end users can have its access level changed to access other features by the administrative user. A flow chart of our login and webpage will be shown below in figure 5.5e

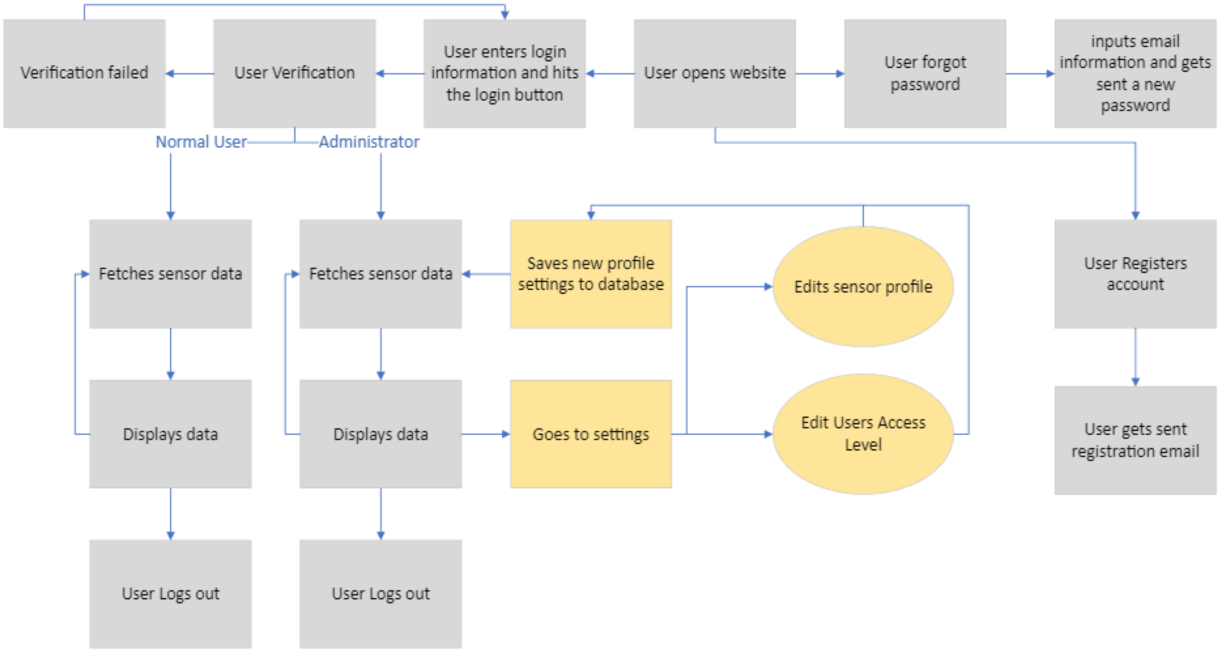


Figure 5.5e

5.5.5 Websocket

Our system would transmit data on an intermittent basis that will be determined by the circumstances. For instance, if the tank is actively balancing nutrients and pH, we want to capture data from the sensors more frequently than when the pumps are inactive.

The user must then be able to look at the fluctuating parameters in the web client. This could be done by having multiple requests and responses between server and client, however, this can cause many problems from the backend side.

Whenever there is a need for constant communication between server and client, a **Websocket** is the best solution. As seen on **Figure 5.5.5a**, this websocket would help us maintain a constant stream of data to the server client by only having one request (handshake). Once this request is accepted, the client's connection is established.

Being a full-duplex persistent connection has many advantages. For instance, after the Client sees the data coming from the sensors, they might want to change the parameters. Thanks to our two way connection we could also transmit data in the form of JSON for the microcontroller to pick it up and readjust the pumps.

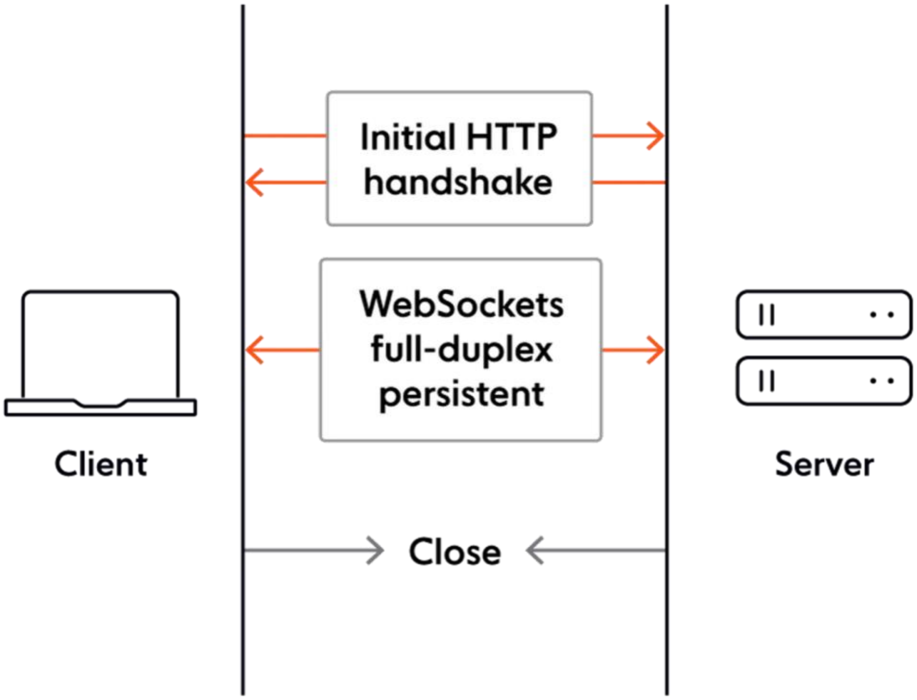


Figure 5.5.5a

Websockets are used in real-time web services like Chats or Online Gaming. We believe this is the best solution for our Website.

5.5.6 PCB

For our automated hydroponic system there will be a lot of complex subsystems that will take a lot of code. We will need to control all the hardware and transmit information in between each of them. We will need many contingencies to be able to react to as many different situations as possible. We will have many sensors and pumps we will need to control so in order to control all these systems we will split the code into three main parts. The first part will control the microcontroller on the pcb that has all the pumps and sensors connected to it. The second part would control the raspberry pi that will receive the data from pcb and transmit it. The third part of the code will receive data from raspberry pi and move it over the website and database backend.

In the microcontroller section the code for the pH and TDS sensor will be simple taking in an analog input to the Arduino. Once the connection is established, we will create a serial connection with our raspberry pi. Once the serial connection with the raspberry pi is established the raspberry pi will send a signal over to the Arduino. The Arduino should then respond with the sensor readings. The pH sensor should transmit pH reading back with two decimal spaces of digits and the TDS sensor should respond with the total dissolved solids written in ppm. When not transmitting data the sensors should run in low power mode. But there are other commands that should be able to put the sensors into continuous reading mode where it can continuously transmit data. When we read the pH or TDS sensors and notice that their values are not within the specified values the pumps will be activated to adjust the level accordingly.

The pumps will be controlled via an IC controller. When the pumps are needed a digital 12v signal. A timer will be written into the code so that the peristaltic pumps will be activated on a strict timer for a set amount of time. These pumps work on a rotary motion and pump a set amount of liquid per rotation. Once we find out how long the pump needs to be active for one rotation, we can create code to leave the pumps on for a set amount of time on a timer. From this point we can keep track of the number of rotations of the pump and accurately know the amount of liquid we are pumping. Once we know how much liquid we are pumping per a rotation we can derive when the nutrient solutions are running low based on the size of the container. Based on the nature of hydroponics and their sensors we will have to wait a set amount of time for the nutrients and pH to permeate the water and stabilize. We will have to create another timer built into our microcontroller code to keep track of the previous times the pump has gone off.

The water level is another function that the microcontroller will have to regulate in order to automatically refill the reservoir as necessary. The water level sensor will send a digital signal over to the PCB whenever it is too low to dynamically increase the water level until it is at an acceptable level. We will have a sizable reservoir outside our main tank for the hydroponic growth and whenever the system is below the required amount of water for growth the pumps will fill the main tank for a set amount of time until the main tank is completely refilled. Though we will have to run tests to find the time until the main tank is filled in order to not overflow the system.

All these processes should run on and should run in multiple loops sending information to and from the raspberry pi. At the start of the loop the raspberry pi should check the database for any flags that would indicate changes to the ranges of the acid base solution, or the tds solution. If there are any changes the raspberry pi would get the new values from the database. Then I2C connection would allow the raspberry pi to make changes to the loop code in the Arduino to make changes in the ph and tds range. Once this check is complete the Arduino starts its loop starting its serial connection with all the sensors and pumps. The first thing the Arduino will check is the water level once the water level sensor is triggered the water pump will run for a set amount of time repeating on a loop until the water level is within appropriate levels. At the beginning of each loop the arduino will send the new sensor data to the database to be recorded. Once the water level is within appropriate levels the loop will end. The next loop would be the ph sensor loop at the start of the loop the ph sensor will check sensor data and send the data to the database, if the ph level is within the ph range the loop will end if it is not depending on whether the ph of the water if too acidic or basic the pumps will activate for a set amount of time and the sensor data will be checked, and the data will be sent to the database again. This loop will repeat until the ph is within the ph range and the loop will be exited. The last sensor to check is the tds sensor, the tds loop follows the same idea as the ph loop. Once the tds loop starts the tds sensor data is checked and sent to the database and if it is within the tds range the loop will stop. If the tds value is not within the tds range the loop will continue until the tds value is within the tds range. Depending on whether the tds value is too high or too low the nutrients will be pumped into the water for a set amount of time and the sensor will be checked again. After the tds loop is completed we return to the start of the loop and once again check for any changes to the range values. Below is a flow chart of this loop in figure 5.5e.

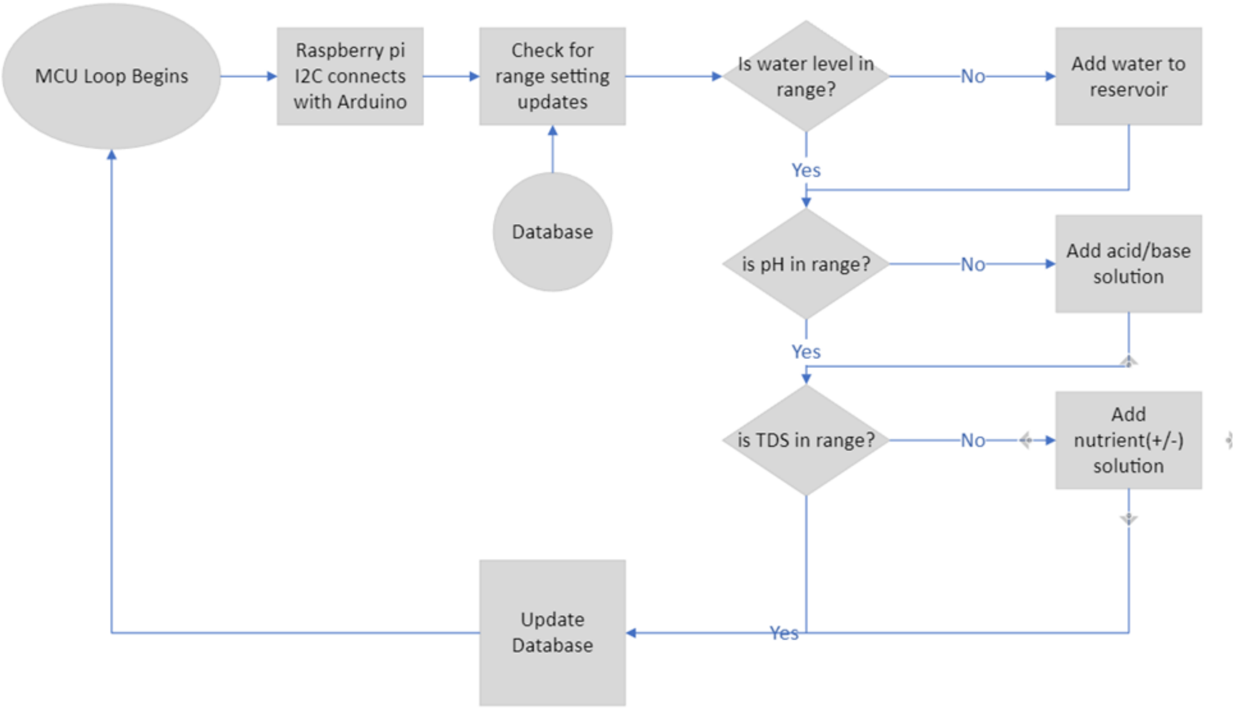


Figure 5.5f

In a setup loop we will establish serial connections to all the sensors and pumps as well as the I2C connection from the raspberry pi to the Arduino. We will use a call and return type connection between these sensors and pumps so that the pumps and sensors can sit in low power mode when not in use. We also set up the Wi-Fi connection through the raspberry pi so that we can interface with the database accordingly. This diagram for the microcontroller should be the general format of the code although subject to change. Many functions may be altered during development but the result should be the same.

5.6 Summary of Design

The initial design architecture provides the layout of how we are organizing the work to successfully complete our project. The various subsystems and software design also give a clear understanding of the direction we are trying to take to accomplish our goal. The website will provide full control of the hydroponic system to the customer from the comfort of their own home or while they are “on the go” via a smartphone. We will be using the MEVN stack of MongoDB, Express, React and Node.js as our web stack. The web server we will be using is Node.js, and the database will be MongoDB.

6. Project Prototype Construction and Coding

This chapter focuses on our team's plans for constructing and coding the prototype of our project before we make an attempt at building the final design of our hydroponic system during the second half of senior design. It will cover the beginning stages of the growing system's prototype, the integrated schematic designs, the creation and assembly of our custom printed circuit board, and the final coding plan.

6.1 Growing System Prototype

Our prototype for the growing system is featured in **Figure 6.1a** below. This design is not to scale and serves as a visual guide about the connections between different components. We also had in mind the spacing between different parts.

All of our components will be inside our growing tent. This would isolate our plants from the outside environment, making a more humid and warm environment. As seen in the design, the main tank houses the plants, pH, water level, TDS and water temperature sensors. These sensors will be connected to the sensor interfacing microcontroller (in our case an Arduino), which will then send the data to our main Raspberry Pi MCU. We plan on housing the microcontrollers outside the tent to avoid any water and humidity control.

Inside the water tank, one of the issues we foresaw is the roots wrapping around our sensors and altering the data. To ensure that our water related sensors are always in contact with the water while going up and down with the floater, we plan on attaching them to the underside of our floater but inside an acrylic or 3D printed enclosure as seen on **Figure 6.1b.** This enclosure would have entrances that allow water to come in but not any other entity.

Figure 6.1b

One sensor would need to be separate from the others: the water level sensor has to be specifically fixed in position, waiting for the low level to start a main tank refill. To do this, we plan on using a conduit that acts as a rail for the cable and sensor to go through, this way we avoid drilling any holes in our tank while also allowing for our free floater system to move vertically.

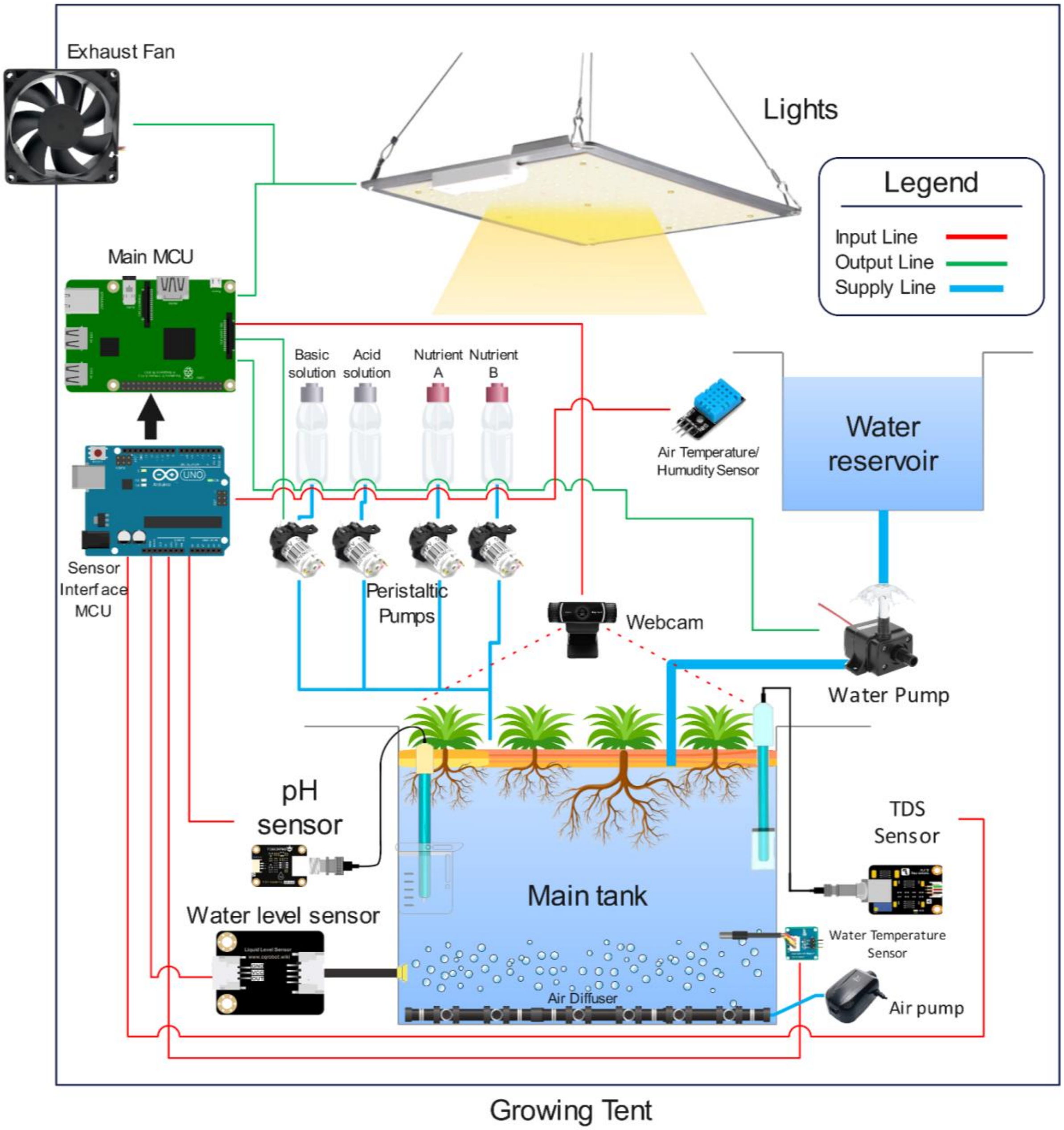


Figure 6.1a

Connected to the tank, we would have our reservoir tank. This tank would allow for easy refilling of the main tank via a water pump triggered by our water level sensor. The water level sensor is connected to our sensor interface MCU, while the water pump is connected to our main MCU as output.

Additionally, we plan on attaching the exhaust fan to the side of the tent, controlled by the Air/Humidity sensor which will be attached to a mid-high point on the inside of the tent. Our tent comes equipped with a special exhaust hole for ventilation. This fan would be used to regulate the temperature and humidity inside the tent and would be connected to the same output as our lights. The lights would be on for 16 hours per day, directly timed by the MCU. We think that the fan would need to stay on if the lights are on due to the heat they dissipate, however, we will test the environment and reassess if our assumption is wrong.

The pH and Nutrient regulator subsystem is outside the tank, and it works by connecting the four peristaltic pumps to the four bottles containing each specific fluid. Then, plastic tubes will run from the pumps to the inside of the tank to deliver the liquids. We will use transparent tubing to be able to visualize the system working while testing and deploying.

Another important part will be our Webcam that will feed visual data to our MCU. We plan on mounting this camera on the front side of our garden, giving a lateral view of the plants as they grow. The camera would need to have a white background to accurately calibrate the computer vision software that we plan on using. All of this will allow us to monitor plants' growth and have a live feed of our garden.

6.1.1 Miniature Version Prototype

We decided to construct a mini prototype to test our idea before we can implement a large size final product. We used materials that were readily available to us. The main tank consists of an 8 Liter cooler (**Figure 6.1.1a**) that perfectly fits our requirements. Additionally, we used an aquarium water pump to oxygenate the water (black device on the bottom right).

Figure 6.1.1a

The pump delivers air to the bottom of the water tank via its twin outlets that are connected to two air stones through rubber tubes. As seen on **Figure 6.1.1a**, we considered that the two air stones deliver more than enough oxygen spread evenly through the tank. This is a temporary substitute of the larger air diffuser that we plan on implementing on the final prototype.

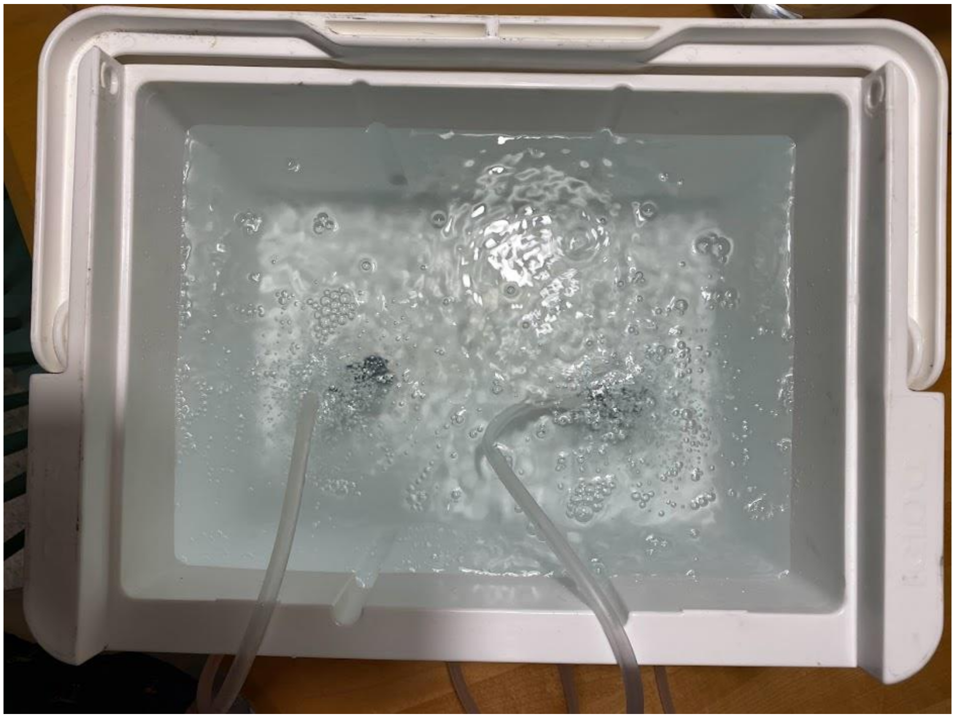


Figure 6.1.1b

The nutrients will be inserted into the tank through an additional tube. The concentration of nutrients will be the same as supplied to our control hydroponic system.

To achieve a cleaner, more organized look, the rails on the inside of the tank will be used to pass the tubes for air and nutrients supply.

6.2 Integrated Schematics

This Hydroponic System will utilize 1 PCB with maybe 2 layers. The PCB will get its power from the 15Vdc supply by the AC to DC converter output 15Vdc with a rating of 4.1 Amps. This board will take sensors of the analog signal and will be routed to each channel of the ADC (4 channel), as the design progressive I will look into an 6 or 8 channel ADC to support all current and future sensors . The digital output of the ADC will be routed to a connector on the design PCB.

The connector from the design PCB will mate with the Raspberry Pi GPIO connector. Some of the sensors will need a conditioning circuit to smooth out the small signal voltage, due to impedance and noise in the long cable of each sensor. Amplifiers will be utilized to bring gain to the small analog signal, because the ADC analog input signal ranges from 0-5Vdc. I will research the data sheet on each of these individual sensors to see what the output range for the specific sensor is to choose the right amplifier for the gain.

The 5V dc to dc converter will power all the sensor devices. The 12Vdc linear regulator (**Figure 6.2a)** will power on all the pumps, with the use of an IC controller that will turn on the water the pump when the main tank water is low, shuts off the pumps via the IC controller. The water level sensor will send a signal to the IC controller when the water is detected below the minimum, above maximum, and at the desired water level. The Nutrient and pH pumps will also use the 12Vdc regulator, the Nutrient and pH sensor will be controlled via the IC controller. When the nutrient and pH sensor detects low nutrient and pH level is low. The IC controller will turn the pumps on and pump the correct amount of nutrient or pH liquid into the main tank.

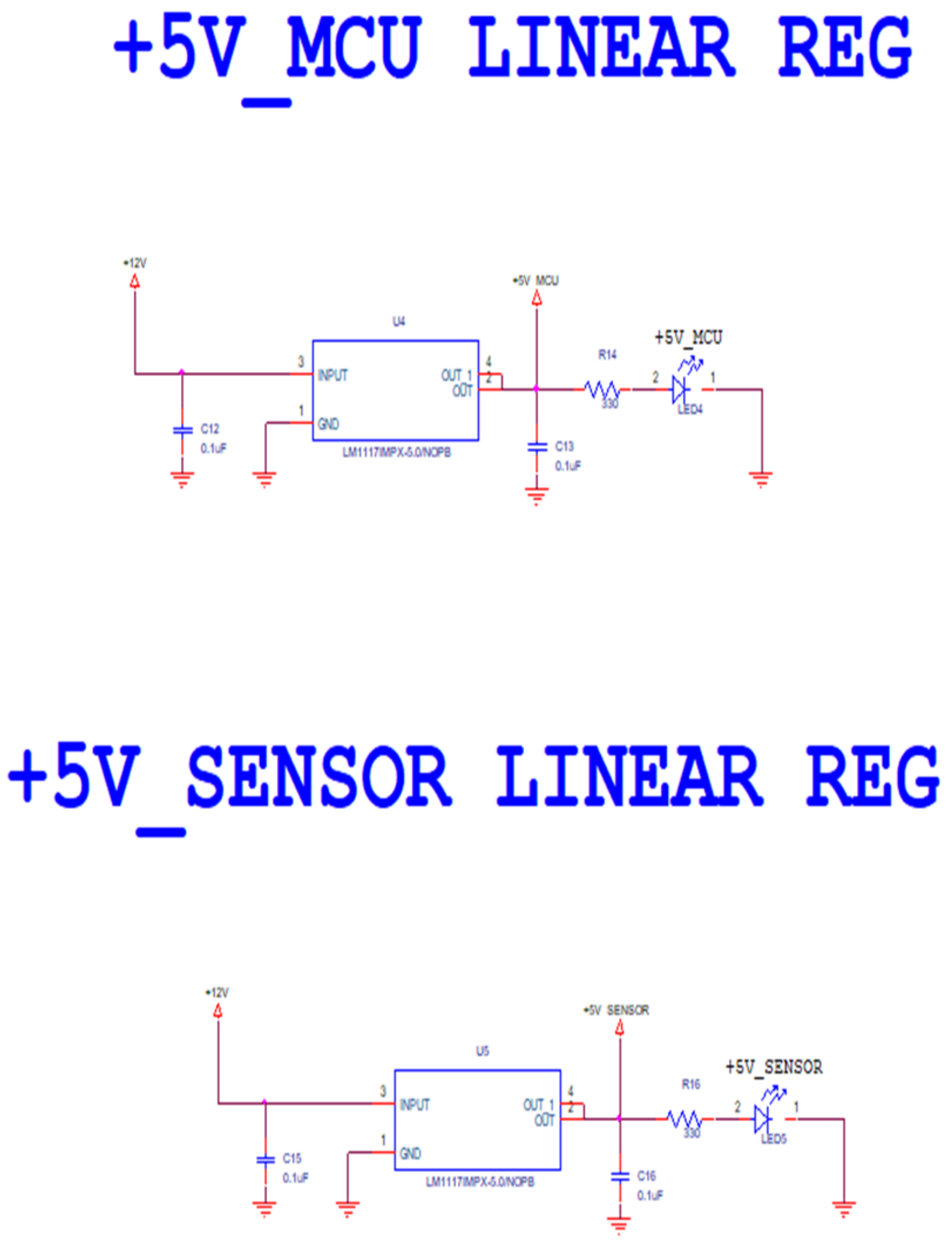


Figure 6.2a

All the devices on the Design CCA will have onboard circuit protection. The fuses will be incorporated on the CCA, this will protect the circuit from an unlikely event of power surge, short circuit, or overvoltage event.

The Raspberry pi microcontroller will collect all the data from the ADC. The data will be stored on a web server. The data will analyze the nutrient, pH, and purity of the water. This data will help to keep the plants healthy. The webcam is incorporated to monitor the health of the plants by looking at the color of the plants.

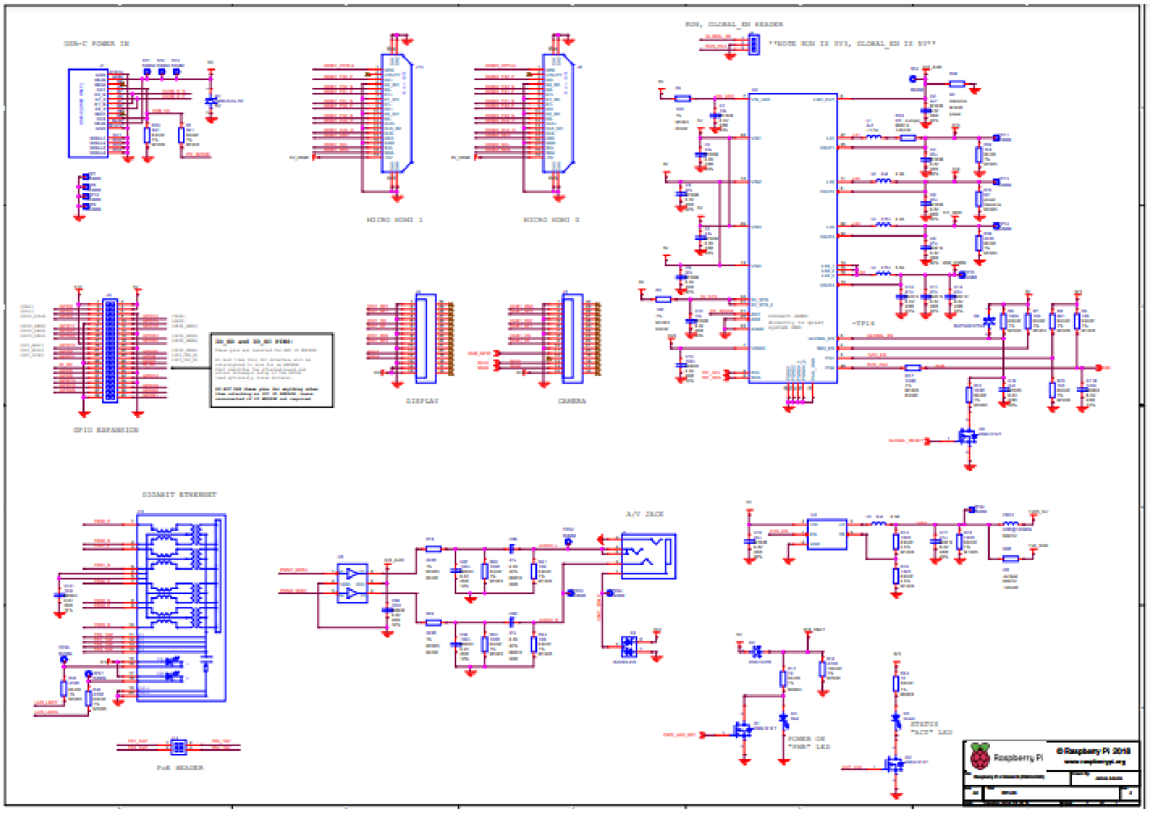


Figure 6.2b Raspberry Pi 4 B Schematic

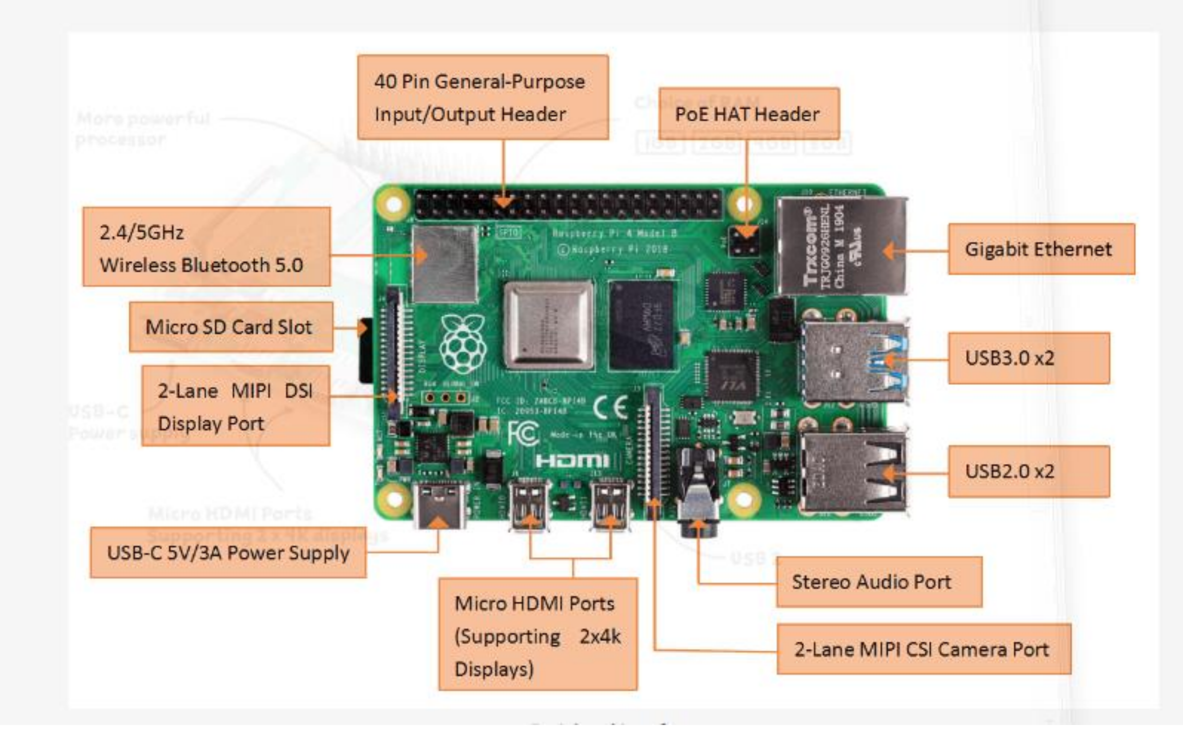


Figure 6.2c Raspberry Pi 4 B CCA

This part ATMEGA328 was chosen because it was readily available from a starter kit and the microcontroller has an 8 channel Analog to Digital Converter (ADC) built internally in the microcontroller. There were other microcontrollers available on the market, but those microcontrollers did not have an internal ADC built into the microcontroller. ATMEGA328 will optimize the PCB, by using fewer components, which will save space on the PCB. The group is currently using a breadboard built circuit with an ATMEGA328 microcontroller to test the individual sensors. That was the other reason why we picked this microcontroller, because the software team is currently testing their code with these sensors with ATMEGA328. These individual sensor testing is the low level of integrating the hardware with the software. This integration will ensure that the sensors are compatible with the microcontroller.

ATMEGA328 has a built 8 channel to A/D converter with a 10Bit precision. This part was perfect for our project. This design will incorporate five sensors, these sensors provide an analog voltage signal output, which is routed to the microcontroller analog input (A0-A7). Each individual sensor analog output will run to each input of the ADC of the microcontroller. The DC current per I/O pin is 20mA, which will meet the specification of operating current for all the sensors and the digital output signal for all the pumps.

The microcontroller also provides digital output pins (**Figure 6.2d**). We will utilize these digital output pins to control the water pump and the four parasitic pumps. When the water level is low, the water level sensor will know, because the software team has incorporated the code to let the sensor know that the water level has dropped below the minimum. The digital pin for that sensor will produce a 5V signal to the N channel MOSFET this will close the Drain to Source connecting the 12Vdc supply to run the pump, the code written by the software will control the pump, when the water level has reached its desire level, the digital signal will be 0Vdc the Vgs will see the 0Vdc the N channel MOSFET will be an open switch.

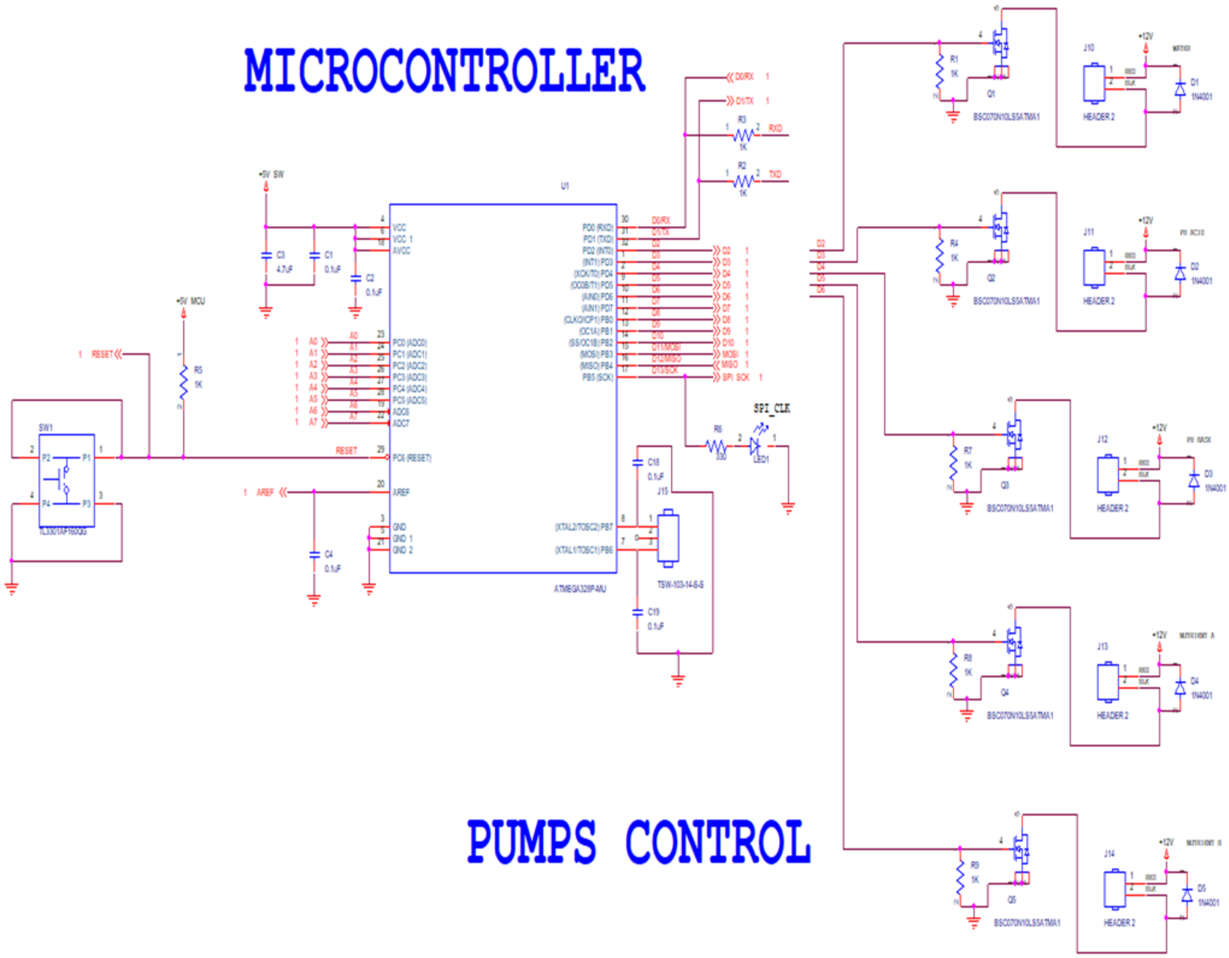


Figure 6.2d

The FT232RL (USB UART IC) shown in **Figure 6.2e** is a ICs that is commonly used to convert USB signals to UART signals, also this IC can convert USB to RS232/RS422/RS485. This IC is a commonly used and a popular way to convert USB signals to TTL (Transistor-Transistor Logic) signals. It is used to interface devices like Arduino IDE with the laptop through a USB cable. In a simple word, FT232RL makes it really easy to convert a USB signal to the UART signal to communicate with the microcontrollers (ATMEGA328).

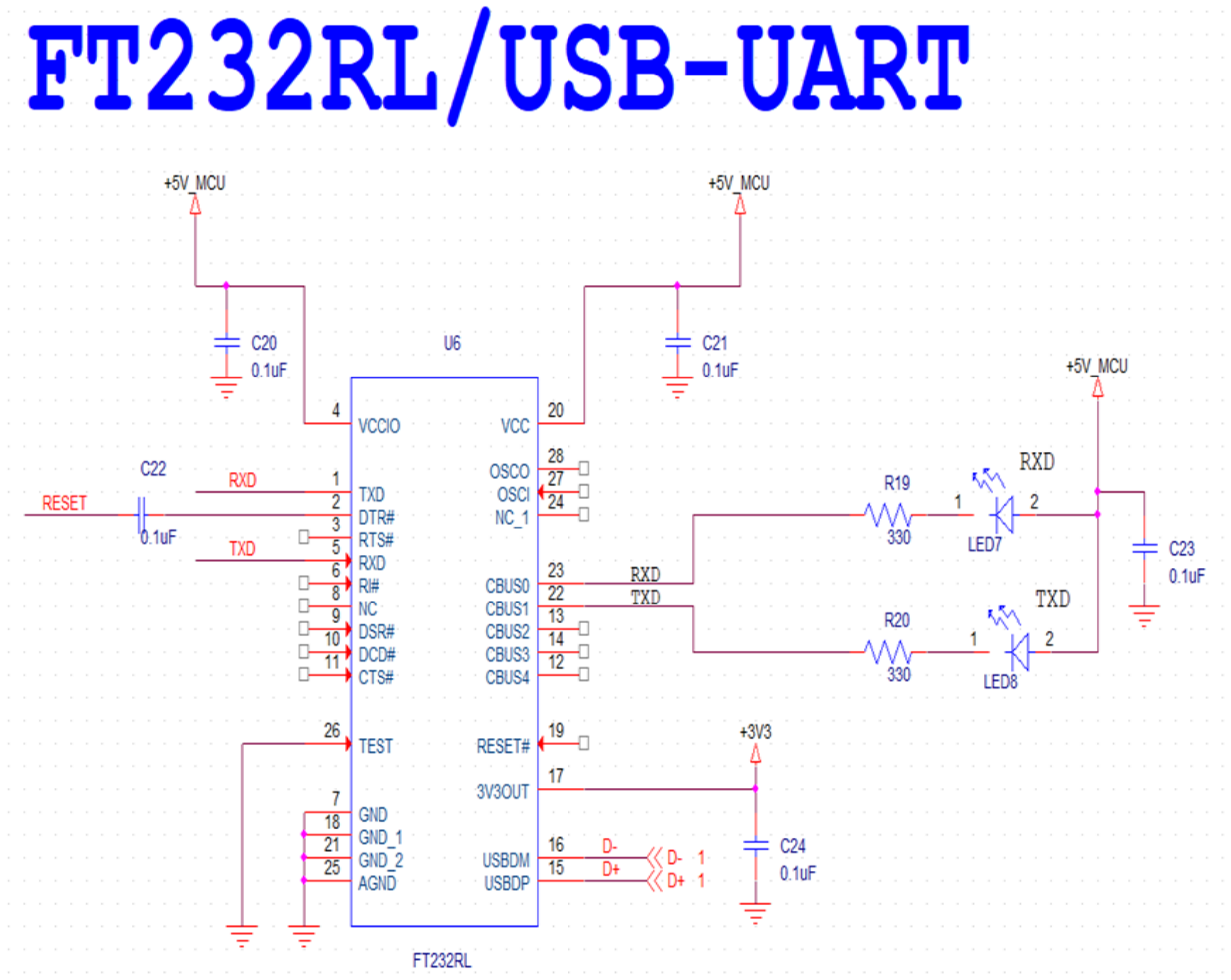


Figure 6.2e

Using Arduino IDE on the laptop, the user can write and compile the code to the microcontroller by going through the FT232RL, this device has a Tx(Transmit) pin which will transmit the user data to the ATMEGA328 Rx (Receive)pin, the Rx will receive all the data the user is transmitting. Then from the microcontroller Tx pin, the data will sent serially to the Rx pin on the FT232RL device, then this Rx signal will be converted to usb signal then the user will receive the data to the Arduino IDE window screen.

The complexity of this integration will be taking the Tx data signal from the microcontroller and sending that data to the USB UART or the Raspberry Pi 4 CCA. This will require a two state 5Vdc buffer to split the Tx signal to two devices. Using a three pin header to divide which device the Tx pin of the ATMEGA328 will send the data to, it will require a jumper to connect to FT232RL or to the Raspberry Pi 4. This will manual moving of the jumper on the three pin header to choose which device the data will be sent to.

Another way to split the Tx signal data is by using a 2 to 1 MUX (Multiplexer). With the 2 to 1 MUX, the user can command the multiplexer to select which device to transmit the data to either the USB UART or the Raspberry Pi. This will be time consuming and more parts will be required to accomplish this task, a simple 3 pin header and a jumper is much cheaper.

The Integration of the Tx signal (5Vdc) from the ATMEGA328 to the Raspberry PI 4 CCA, must take into consideration and ensure that both Tx signal from the microcontroller to the Raspberry Pi has the same voltage level signal, make sure that the Tx signal on the Pi is at 5Vdc level. The most important part is to make sure both the ATMEGA328 and the Raspberry Pi GND (ground) are tied together, to make it a common ground for the both CCA.

This Micro USB connector (P/N 0473460001) in **Figure 6.2f** was chosen because of its availability from a scrap CCA. I use a hot air blower to remove the part of the scrap board. The micro usb is used to interface from the laptop to the microcontroller. The signal from the laptop goes to the USB UART IC. The USB signal is converted to serial (UART) Tx and Rx. The user transmits (Tx) the USB signal through the FT232RL, then the signal will be received (Rx pin) on the microcontroller. Then the Tx pin on the microcontroller will send the data to the Rx pin of the FT232RL device, then through the micro usb to the laptop, the data will appear on ARDUINO IDE GUI.

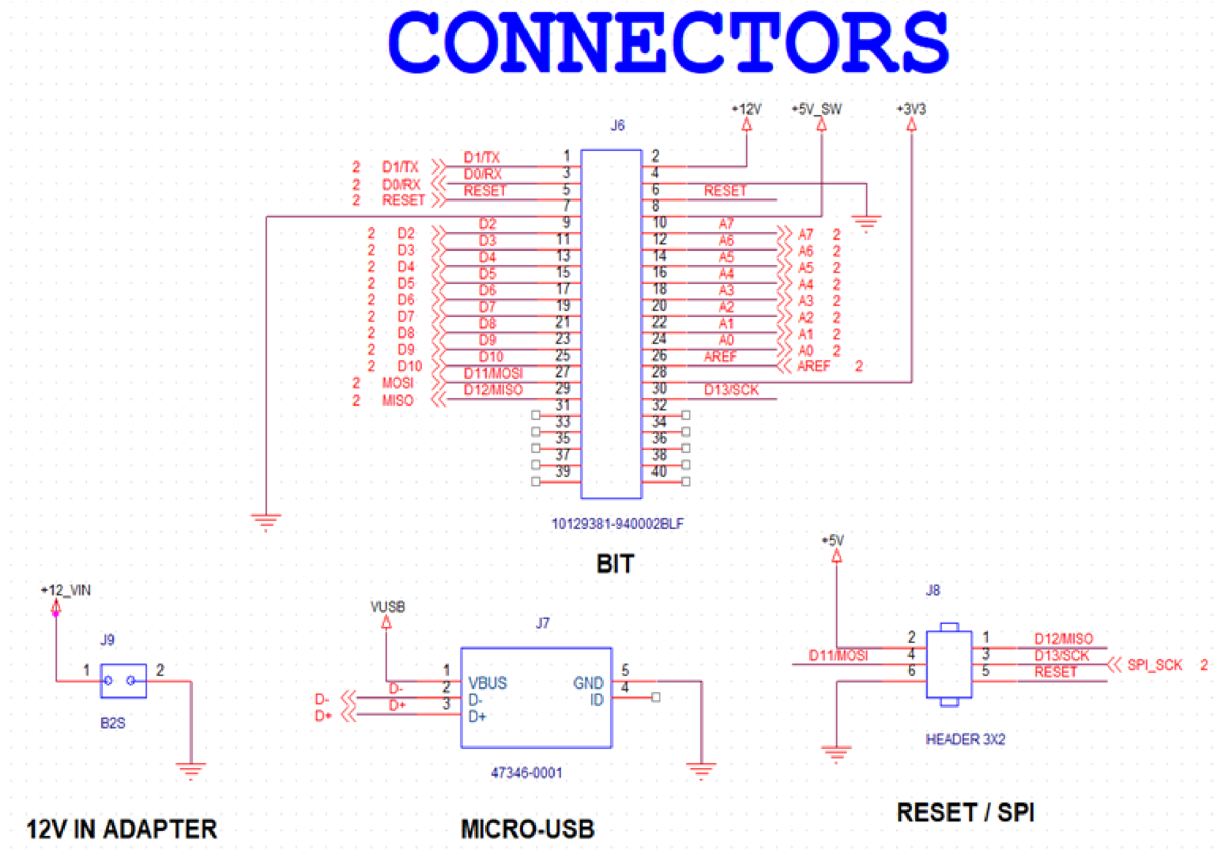


Figure 6.2f

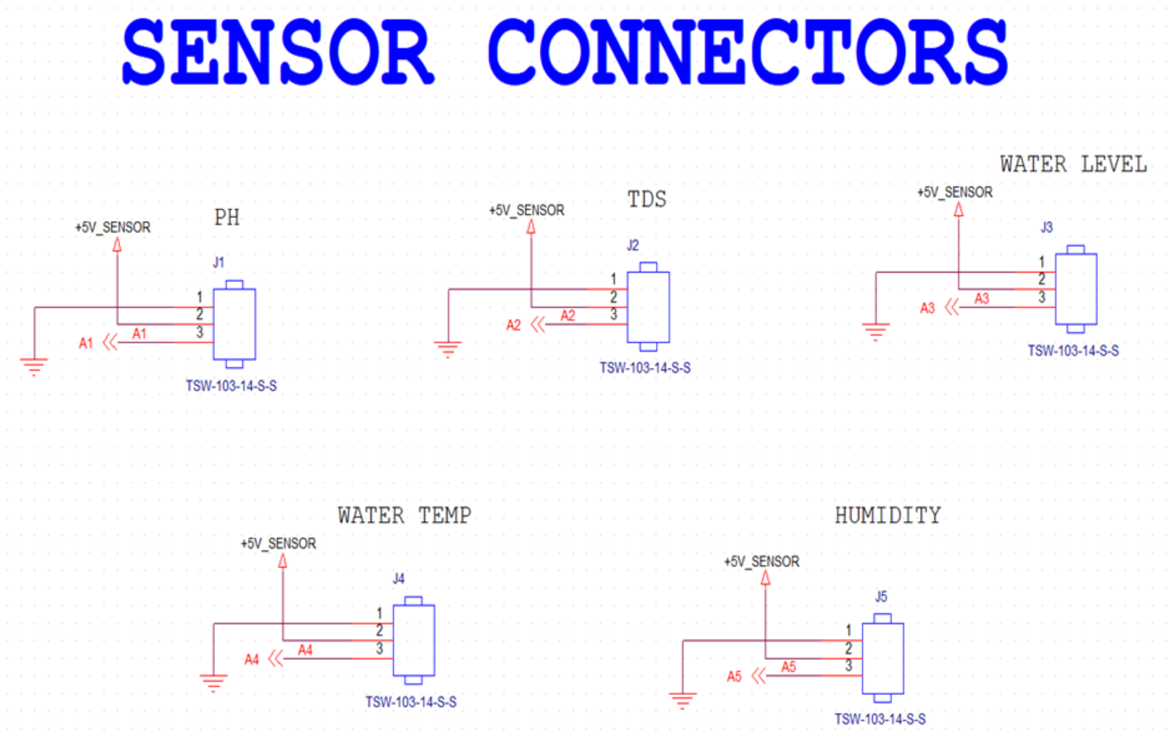
The 330 ohm surface mount resistor 0603 package was picked because it was available at work, from the 0603 resistor kit. The 330 ohm resistor is to dim the Green LED (Light Emitting Diode), the 330 ohm resistor (I = V/R) is used to limit the current going to the LED, this dim the brightness of the LED.

The Green LED in the circuit is used as indicators to let the user know the signal is working properly. The Tx signal is working properly, when data is being transmitted the LED will be lit to let the user know data is transmitting. The LED for Rx line will act the same as the Tx line, the LED will be lit when data is being received. The LED is connected to a linear regulator. The 5Vdc output is to let the user know the 5Vdc from the linear regulator is working properly.

The 1k ohm surface mount resistor 0603 package was picked because it was available at work from the 0603 resistor kit. The 1k ohm is used as a pull-up or pull-down resistor. Pull-up and Pull-down resistors are used to correct the bias inputs of the digital gates to stop it from floating about randomly when there is no input condition. The input is now effectively unconnected from either a defined HIGH or LOW condition, it has the potential to "float" about between 0V and +5V allowing the input to be self-bias at any voltage level. The pull-up or pull-down resistor is to ensure that either the signal is High or Low, not in the floating state; this could damage the device or cause problems in the circuit.

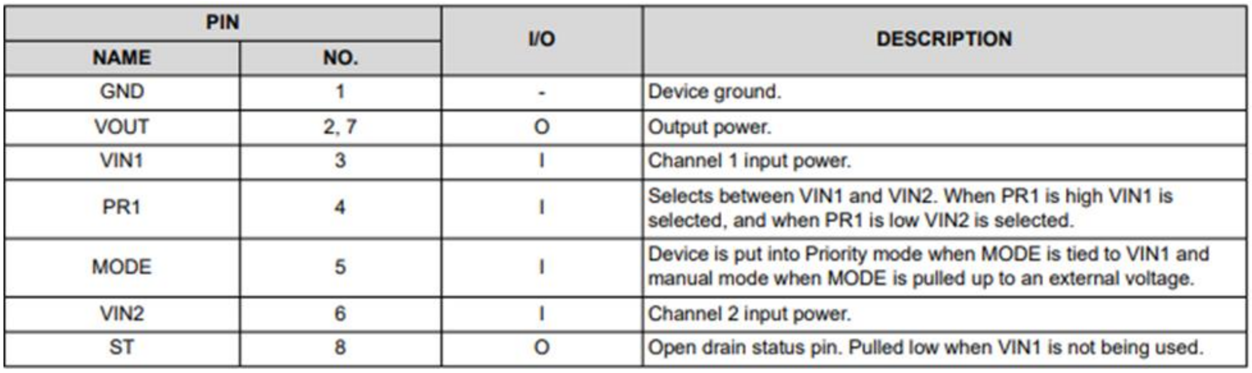
The 1N4001 was available on the Digi key; this diode only allowed electricity to flow in one direction. In the circuit the diode is connected in parallel to the motor. When the power is turned off to the motor, the negative spike of voltage can damage the microcontroller or the mosfet. The diode protects against the voltage spike. Without the diode, the voltage across the motor would be a lot to keep the current flowing, which would fry the mosfet.

The LM1117IMPX-5.0 surface mount is a 5v linear voltage regulator, and its output current is 800mA surface mount. This part is available on a scrap CCA, and I will salvage the part by using a heat blower to remove it from the scrap PCB. The linear 5V regulator will supply power to all the sensors, USB to UART IC, and the ATMEGA328 microcontroller.



Sensor Connectors Figure 6.2g

This device TPS2116 has a power mux that has voltage rating from 1.6 V to 5.5 V and the maximum current rating of 2.5 A. The power mux device uses an N-channel MOSFET to switch between supplies while providing a controlled slew rate when the voltage is first applied. This device can be configured two ways. The 1st configuration is the automatic priority mode which prioritizes VN1 as the main supply and switches over secondary supply VN2 when VN1 drops below the threshold voltage. The 2nd configuration is the manual mode which will let the user toggle a GPIO or enable signal to a switch. The pin VIN1 is connected to PR1 through a resistor divider. When the MODE is connected to high, PR1 will determine the channel to be selected. To configure VIN1 as the priority supply, connect MODE to VIN1 and set the proper threshold through a resistor divider from VIN1 to PR1, to configure the resistor divider refer to datasheet for correct resistor values.. When VIN1 takes priority over VIN2, a resistor divider can be used to set the switchover voltage threshold. When VIN1 is first applied, PR1 is high and VOUT is powered by input voltage. When VIN1 begins to drop, the voltage on PR1 is lowered until it crosses the VREF threshold. When PR1 goes below VREF threshold the device will switch over to VIN2.

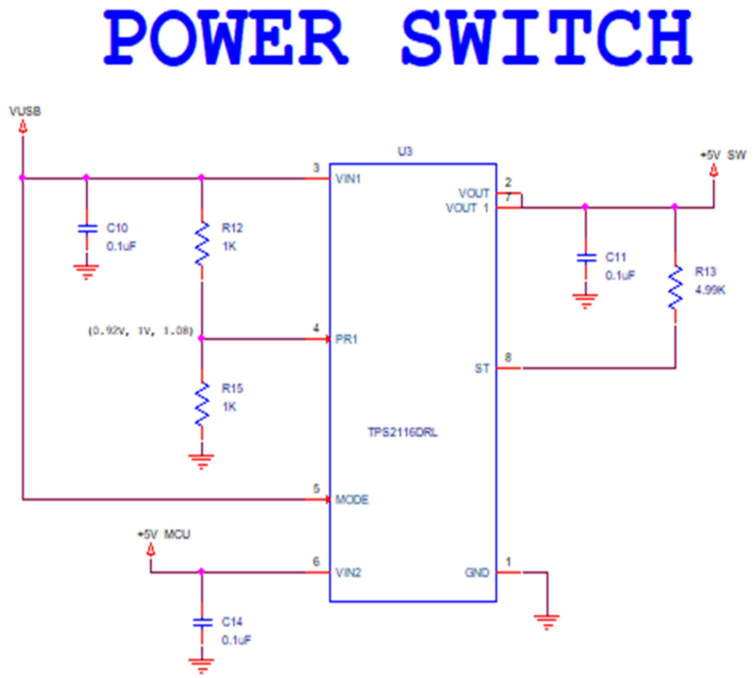


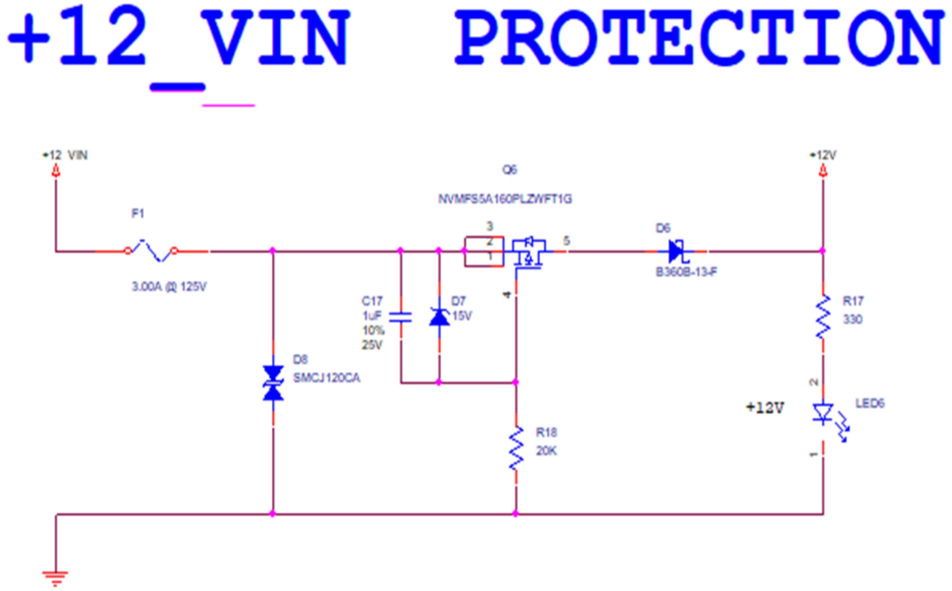
PIN		I/O	DESCRIPTION
NAME	NO.		
GND	1	-	Device ground.
VOUT	2, 7	O	Output power.
VIN1	3	I	Channel 1 input power.
PR1	4	I	Selects between VIN1 and VIN2. When PR1 is high VIN1 is selected, and when PR1 is low VIN2 is selected.
MODE	5	I	Device is put into Priority mode when MODE is tied to VIN1 and manual mode when MODE is pulled up to an external voltage.
VIN2	6	I	Channel 2 input power.
ST	8	O	Open drain status pin. Pulled low when VIN1 is not being used.

A 12V linear voltage regulator will be used to power all the pump and motors when the N channel mosfet is turned on by the digital signal from the microcontroller. As of the moment I am still deciding on which 12V regulator I will be utilizing with my design.

I have decided to go with an AC to DC converter output 12V 5A power supply adapter. This will eliminate the 12V linear regulator that will turn the pumps on, this will be more efficient. This 12V power supply adapter is equipped with an automatic overload cut-off, over Voltage cut-off, automatic thermal cut-off, short circuit protection. The 12V from the power supply adapter will now be the voltage source for the 12V pumps, also it will be voltage source for the two 5V linear regulators.

In order to protect the circuit downstream from an accidental reverse connection of the battery terminals, a 3.0a surface mount Schottky barrier rectifier (part #B3608-13-F) was chosen to serve as a polarity protection diode. This is because when a reverse voltage is applied to a Schottky diode, it becomes an open circuit to protect the other components. A few of its key features and benefits include a guard ring die construction for transient protection, its ideally suited for automated assembly, low power loss with high efficiency, surge overload rating to 125A peak, with a lead-free finish it is additionally RoHS compliant. This device is typically for use in low voltage, high frequency inverters, free wheeling, and polarity protection applications. At test conditions of $I_F = 3.0A$ & $T_A = +25°C$, the max forward voltage drop $V_F = 0.50V - 0.70V$, and a max rated leakage current $I_R = 0.1mA$.





A surface mount zener diode (part #BZT52C15LP-7) was additionally incorporated into the circuit design. This ultra-small leadless surface mount package is ideally suited for automated assembly processes. It is also lead-free by design and RoHS compliant. Its maximum forward voltage is 0.9V at an $I_F = 10mA$. At a radiant air temperature of $T_A = +25°C$, the power dissipation $P_D = 250$ mW.

A TVS diode (part #SMCJ12CA) was also chosen to serve as the ideal protection for all the I/O devices that are being utilized. It also comes with an extensive life of key features such as a peak pulse power dissipation P_{PPM} = 1500 W, a peak forward surge current I_{FSM} = 200A, and a maximum instantaneous forward voltage V_F = 3.5/5.0 V.

A P-Channel MOSFET (part #NVMFS5A160PLZWFT1G) was chosen and has a low $R_{DS(on)}$ to minimize conduction losses. It has a wettable flank option for enhanced optical inspection, its AEC-Q101 qualified and PPAP capable. Its drain to source voltage V_{DSS} = -60V, gate to source voltage V_{GS} = +/-20V, a continuous drain current I_D= -100A, and a power dissipation P_D = 200W.

We chose the Cadence OrCAD Capture tool for the design circuit schematics because it is already available at one of our places of work and is easily available for our team to utilize. Additionally, it would be beneficial for both electrical engineering students to better familiarize ourselves with all the functions and tools available with the software as it will strengthen our knowledge for use in the field after graduation. Cadence is one of the most widely used schematic design tools for creating and documenting electrical circuits. Together with the OrCAD CIS (component information system) product that is used for component data management and its highly integrated flows that support the engineering process, OrCAD Capture is one of the most powerful design tool for taking today's product creation from concept to production.

We are also utilizing Ultra Librarian for OrCAD. It provides a unified library that automatically links elements of the part as a single component with multiple views for schematic symbols and PCB footprints. This library is very helpful for schematic symbols and PCB footprints. If the symbols do not exist within the software, we can also create our own library for individual components. We would have to manually draw the part with all the signal and pin descriptions by using the part's data sheet.

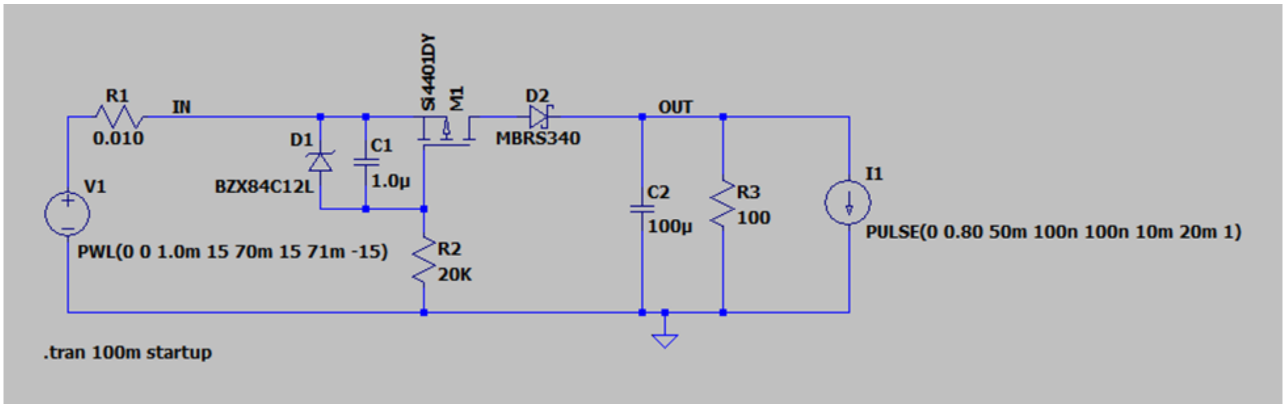
Additionally, we obtained our BOM from the previous schematics. The BOM is shown below on **Table 6.2.**

BILL OF MATERIALS					
QTY	REFERENCE DESIGNATOR	VALUE	DESCRIPTION	MANUFACTURER	MFR PART NUMBER
15	C1, C2, C4-C16, C18, C19	0.1uF	CAP CER 0.1UF 16V X7R 0603	KEMET	C0603C104K4RACAUTO
1	C3	4.7uF	CAP CER 4.7UF 50V X7R 1210	KEMET	C1210C475K5RACAUTO
1	C17	1uF	CAP CER 1UF 10V X7R 0603	AVX Corp	0603ZC105KAT2A
5	D1, D2, D3, D4, D5	1N4001	RECTIFIER Diode, D0-41 50 V 1A	Diotec Semi	1N4001
1	D6	B360B-13-F	DIODE SCHOTTKY 60V 3A SMB	Diodes Incorporated	B360B-13-F
1	D7	BZT52C15LP-7	DIODE, ZENER, Vz 15V, 250mW, SMT	Diodes Incorporated	BZT52C15LP-7
1	D8	SMCJ12CA	TVS DIODE 12VWM 19.9VC SMC	Diodes Incorporated	SMCJ12CA-13-F
1	F1	0451003.MRL	FUSE BRD MNT 3A 125VAC/VDC 2SMD	Littelfuse Inc.	0451003.MRL
5	J1, J2, J3, J4, J5	TSW-103-14-S-S	CONN HEADER VERT 3POS 2.54MM	Samtec Inc.	TSW-103-14-S-S
1	J6	10129381-940002BLF	CONN HEADER VERT 40POS 2.54MM	Amphenol ICC (FCI)	10129381-940002BLF
1	J7	47346-0001	CONN RCPT USB2.0 MICRO B SMD R/A	Molex	47346-0001
1	J8	HEADER 3X2	CONN HEADER VERT 6POS 2.54MM	Samtec Inc.	HEADER 3X2
6	J9, J10, J11, J12, J13, J14	TSW-102-14-S-S	CONN HEADER VERT 2POS 2.54MM	Samtec Inc.	TSW-102-14-S-S
6	LED1, LED2, LED3, LED4, LED5, LED6	SML-LX0805SUGC-TR	Green LED Indication - Discrete 2.2V	Lumex Opto	SML-LX0805SUGC-TR
5	Q1, Q2, Q3, Q4, Q5	BSC070N10LS5ATMA1	MOSFET N-CH 100V 14A/79A TDSON	Infineon Technologies	BSC070N10LS5ATMA1

1	Q6	NVMFS5A160PLZWFT1G	MOSFET P-CH 60V 15A/100A 5DFN	onsemi	NVMFS5A160PLZWFT1G
10	R1,R2,R3,R4,R5,R7, R8,R9,R12,R15	1K	RES SMD 1K OHM 0.5% 1/10W 0603	Vishay Dale	CRCW06031K00DHEAP
6	R6,R10,R11,R14,R16,R17	330	RES SMD 330 OHM 1% 1/10W 0603	Vishay Dale	CRCW0603330RFKTA
1	R13	4.99K	RES SMD 4.99K OHM 1% 1/3W 0603	Vishay Dale	CRCW06034K99FKEAHP
1	R18	20K	RES SMD 20K OHM 5% 1/10W 0603	Vishay Dale	CRCW060320K0JNEB
1	SW1	TL3301AF160QG	SWITCH TACTILE SPST-NO 0.05A 12V	E-Switch	TL3301AF160QG
1	U1	ATMEGA328P-MU	IC MCU 8BIT 32KB FLASH 32VQFN	Microchip Technology	ATMEGA328P-MU
1	U2	FT232RL	IC USB FS SERIAL UART 28-SSOP	FTDI	FT232RL
1	U3	TPS2116DRL	Power Switch/Driver 1:1 N-Channel 2.5A	Texas Instruments	TPS2116DRL
2	U4, U5	LM1117IMPX-5.0/NOPB	IC REG LINEAR 5V 800MA SOT223-4	Texas Instruments	LM1117IMPX-5.0/NOPB

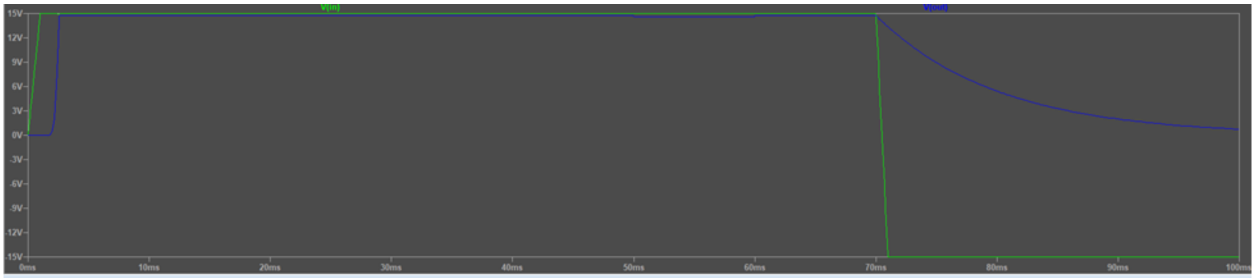
Table 6.2

Protection circuit LTSpice Simulation

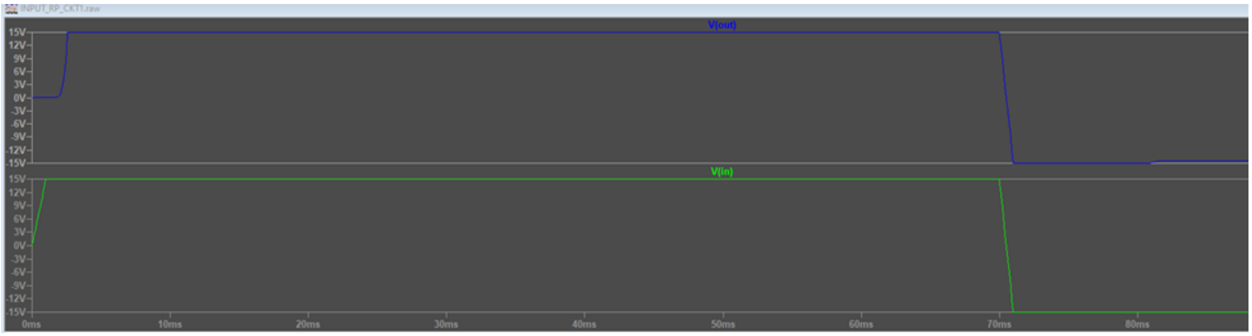


In this simulation the D2 works to block the MOSFET diode. The internal parasitic P-Channel mosfet Q6 diode doesn't allow turnoff in the above circuit. D2 (SCHOTTKY DIODE) is to allow Q6 (P-CHannel MOSFET) to turn off in the negative direction when the input is negative because the substrate of the mosfet diode is on even though Q6 is off. In simulation if D2 was not in the circuit, Q6 substrate diode is conducting even though mosfet Q6 was off. Also in this simulation it shows that at -20V, the current in D8 is 6 AMPS, this more than enough to blow the Fuse F1. Simulating at -18V the current of D8 is 3 AMPS which slowly burns the element in F1 to open the open 12VIN circuit. The TVS D8 is protecting overvoltage in both directions.

If the input voltage exceeds the breakdown voltage rating of TVS Diode bidirectional SMCJ12CA, it will start drawing a lot of current because it goes to low impedance. This excessive current will blow fuse F1. If it has sufficient duration to exceed I^2T rating. The fuse I^2T rating is what determine the melting of the fuse element.



Blue is the Output, which decays to zero volts.



Blue is the Input, Green is the Output

6.3 PCB Vendor and Assembly

Designing and printing a custom made printed circuit board (PCB) through a third-party vendor is necessary to connect all of the electronic components together. Being able to lay out the traces and determine the locations of certain components such as resistors, inductors, and capacitors is done using a PCB design software. There are many different PCB design software available on the market but we chose to use Autodesk Eagle because of our familiarity with it during our junior design course. Additionally, the software can be used free of charge when utilizing UCF's access through their apps page. Although it should be noted that the free version limits the user to only an 80cm^2 board area, two schematic sheets, and two signal layers.

The software allows the user to easily search for the necessary parts of their printed circuit board and add them to a bill of materials (BOM) list. One of the most popular tools used in current PCB design software is automatic routing. This feature is also available in Autodesk Eagle, but one of the most common mistakes when designing and building a custom PCB is an over-reliance on the automatic routing tool. It was advised during our junior design class that automatic routing has a tendency to take up more space on the board than what is necessary and the via holes that are created are typically larger than what is needed. For this reason, it is recommended to only use the automatic routing tool as a guide when organizing the wires on the board.

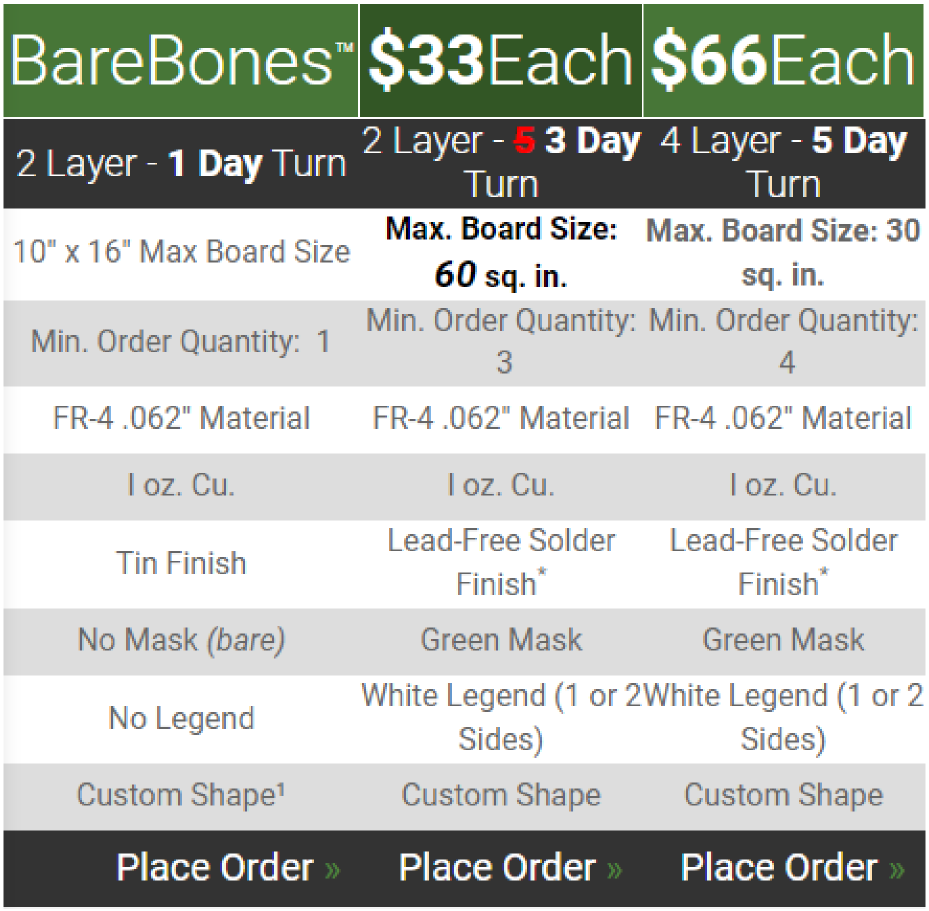
There are also a multitude of different vendors available that can fulfill the needs for creating the printed circuit board to our specifications, such as Advanced Circuits/4pcb, Sunstone Circuits, Allpcb, JLCPCB, PCBgogo, PCBWay, and others. It's important to note however that the cost of the completed work is not the only thing that varies between vendors. Many vendors will require a minimum amount of products to be purchased, so it is important to check for this criteria before beginning an order with a vendor. Another major factor that we must consider is the length of time required for the vendor to actually manufacture our PCB design. Since the start of COVID-19, many different electronic components are not readily available and are on "back-order". This can cause significant delays in receiving our PCB and may cause us to miss the milestones we set for our project in section 8.1 below.

After researching several of the various PCB manufacturing companies in the United States via the internet and speaking with their customer service representatives, we were able to compile a brief comparison between our top choices of vendor and is shown in **Table 6.3a.** After much deliberation, it was decided that we woud work directly with Advanced Circuits at https://www.4pcb.com/pcb-prototype-2-4-layer-boards-specials.html. This was chosen since the manufacturer offers special prices for college students as is summarized in **Table 6.3b** shown below and is courtesy of Advanced Circuits' website.

Vendor	Website	Min. Order Qty.	Cost Estimate
Advanced Circuits	https://www.4pcb.com/	1	$33
PCBWay	https://www.pcbway.com/	5	$30
AllPCB	https://www.allpcb.com/	5	$36

PCBgogo	https://www.pcbgogo.com/	5	$31

Table 6.3a



BareBones™	$33Each	$66Each
2 Layer - **1 Day** Turn	2 Layer - **5 3 Day** Turn	4 Layer - **5 Day** Turn
10" x 16" Max Board Size	**Max. Board Size: 60 sq. in.**	**Max. Board Size: 30 sq. in.**
Min. Order Quantity: 1	Min. Order Quantity: 3	Min. Order Quantity: 4
FR-4 .062" Material	FR-4 .062" Material	FR-4 .062" Material
I oz. Cu.	I oz. Cu.	I oz. Cu.
Tin Finish	Lead-Free Solder Finish*	Lead-Free Solder Finish*
No Mask *(bare)*	Green Mask	Green Mask
No Legend	White Legend (1 or 2 Sides)	White Legend (1 or 2 Sides)
Custom Shape[1]	Custom Shape	Custom Shape
Place Order »	Place Order »	Place Order »

Table 6.3b (Courtesy of 4pcb.com)

It is important to note that even though the table shows a minimum order quantity of three PCB boards, it is not enforced if ordered through their student program. Regardless, it may still be more appropriate to order two boards in order to have a backup in case something were to occur. There are however certain specifications/requirements that our PCB design must adhere to if we are to participate in Advanced Circuits' student program:

Specification/Requirements for Special Pricing Options:

- Minimum 0.005" line/space
- Multi-part or step-and-repeat *$50 additional charge per order
- No internal routing (cut-outs). No slots or overlapping drill hits
- No scoring, tab rout, or drilled hole board separations

- No controlled impedance/dielectric
- No countersinks/counterbores/castellated holes
- No cavities/controlled depth
- Minimum 0.010” hole size. Maximum 50 drilled holes per sq. inch
- Plated or non-plated holes (if specifications not provided, all holes will be plated)

Another important factor to consider before fully assembling our project is to thoroughly test the printed circuit board. In most cases, it will not function correctly and may require either further debugging or a complete redesign and reprint of a new board. For this reason, it may be necessary to pre-plan two different PCB designs ahead of time in order to prepare for a worst-case scenario. It is also the reason we scheduled a hard deadline to submit an order for a custom printed circuit board before the end of senior design 1. This should provide enough of a buffer in case the first PCB design is incapable of operating correctly and our team is forced to redesign and order a new printed circuit board.

PCB vendors make the raw board, these boards are procured by the CCA manufacturer. The CCA manufacturer will purchase all the components on the BOM. The PCB will then be assembled. The stencil will be a template placed over the PCB and solder paste will be applied and the squeezed will sweep the solder paste across the stencil leaving solder paste onto all of the PCB. The next process would be a machine to place the part on the PCB, then the next process the PCB will go through solder flow ovens. Optional flying lead probe, this test will check for short/open in the circuit. The flying lead probe process would be recommended for a mass quantity of CCA to be assembled. From the lot of CCA, the customer can request that 10% (as customer desire of how many CCA should be tested) of the lot of CCA be tested with flying lead test. After the assembly and testing of the CCA is completed, the CCA will go through inspection, optical inspection, and final accepting inspection to follow ISO 9000 standard for Quality Assurance.

Putting together the initial schematic design via the Eagle software requires finding the appropriate symbols and footprints for the specific components we have chosen to build our design. The Eagle software already has an extensive library of various components along with their symbols and footprints. One of the convenient functions of Eagle is that when taking the steps to add a new component to the schematic, it will automatically open its list of available libraries to allow the user to search for their intended component. The majority of components that we selected were not readily available within Eagle’s libraries, but thankfully the software allows the user to import libraries from other sources and saves it for future use. There are many CAD library sites available online that can be utilized to import our missing component footprints but we primarily used Ultra Librarian and SnapEDA.

6.4 Final Coding Plan

For the final coding plan we will give an overview of all the sub systems and how they interact to create an overall autonomous hydroponic system. In this section we will go in depth about the web applications and how the database, microcontroller, and sensors connect. The microcontroller sends data directly to the website which will then communicate with the database. In this system databases can only be interacted with from the website, leading to an extra layer of security and making it necessary to use the website. In the next flow chart (shown in **Figure 6.4a**) we will display how a user will access the website using their computer and influence the MCU.

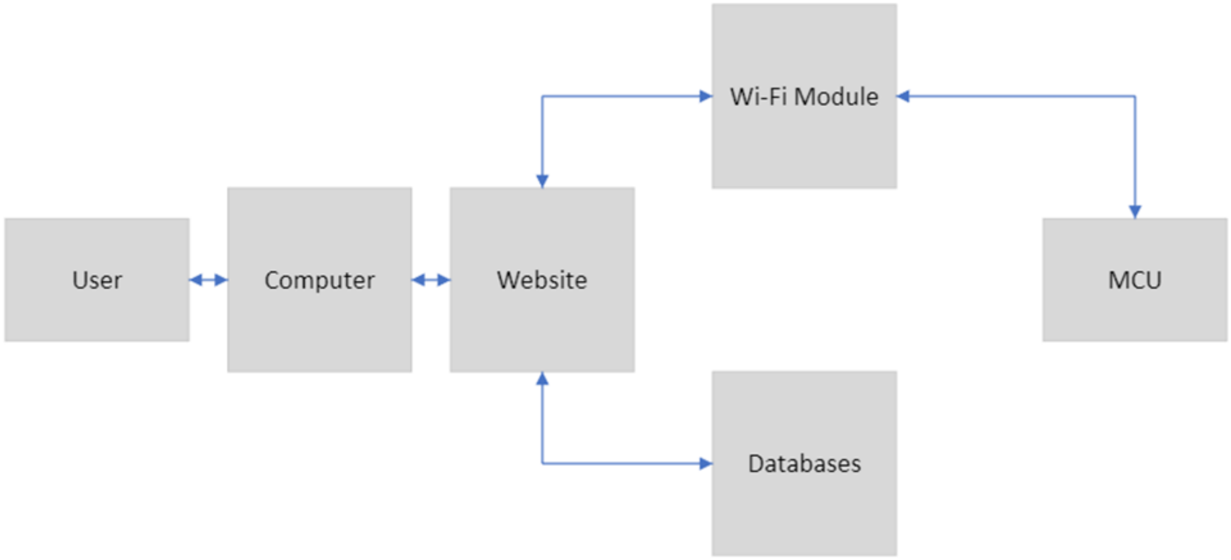


Figure 6.4a

User access can be obtained as they go to the website by connecting with a computer, from that point they can influence the MCU. Once the user logs on and sets the settings for the hydroponic plant the website will update the database and send information to the microcontroller. Once the microcontroller receives the information the microcontroller will begin to make the system function with a series of actions in a loop. The next flowchart in **Figure 6.4b** shows the microcontroller loop.

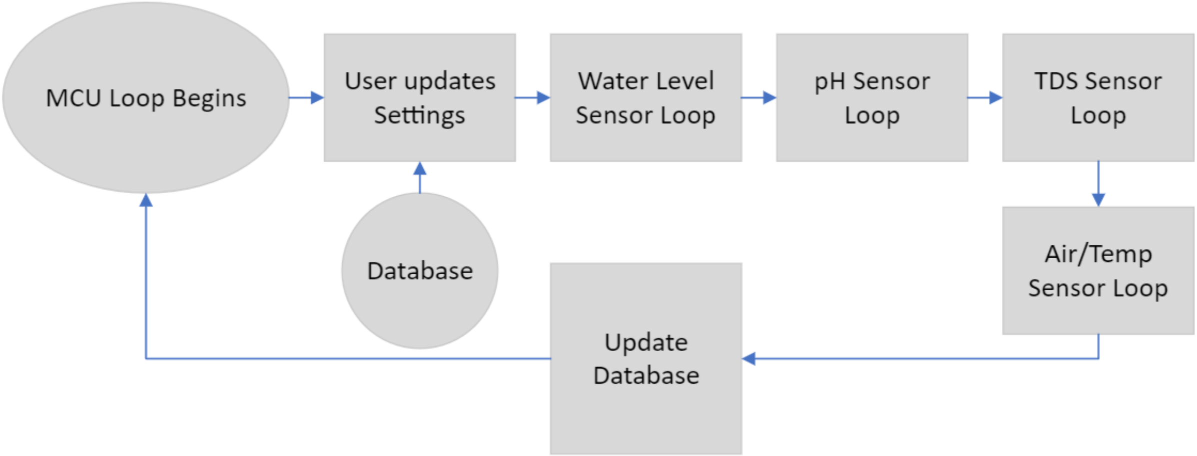


Figure 6.4b

The MCU loop is then split into smaller loops that checks each of the sensors. The first thing the MCU does is check the user settings for any updates or change in values. Once this is complete the MCU moves on to check the first sensor loop, the water level sensor. The water sensor loop first checks the last time the sensor has been checked and whether 1 hour has passed. If 1 hour has not passed since that loop was checked the loop will be exited. If 1 hour has passed, then the MCU will check the water sensor and see whether the water level is within threshold values. If the MCU is within the threshold value, the MCU will then exit the loop. Otherwise depending on the user setting the MCU will update the database and alert the user or automatically pump water to fill the tank from a water reservoir. The water level sensor loop was placed first because this is the main safety feature of the hydroponic system to ensure the plant is getting enough water. If the water level gets too low not only can the pump be damaged the plants may also be injured. A diagram of this loop is shown below in **Figure 6.4c**.

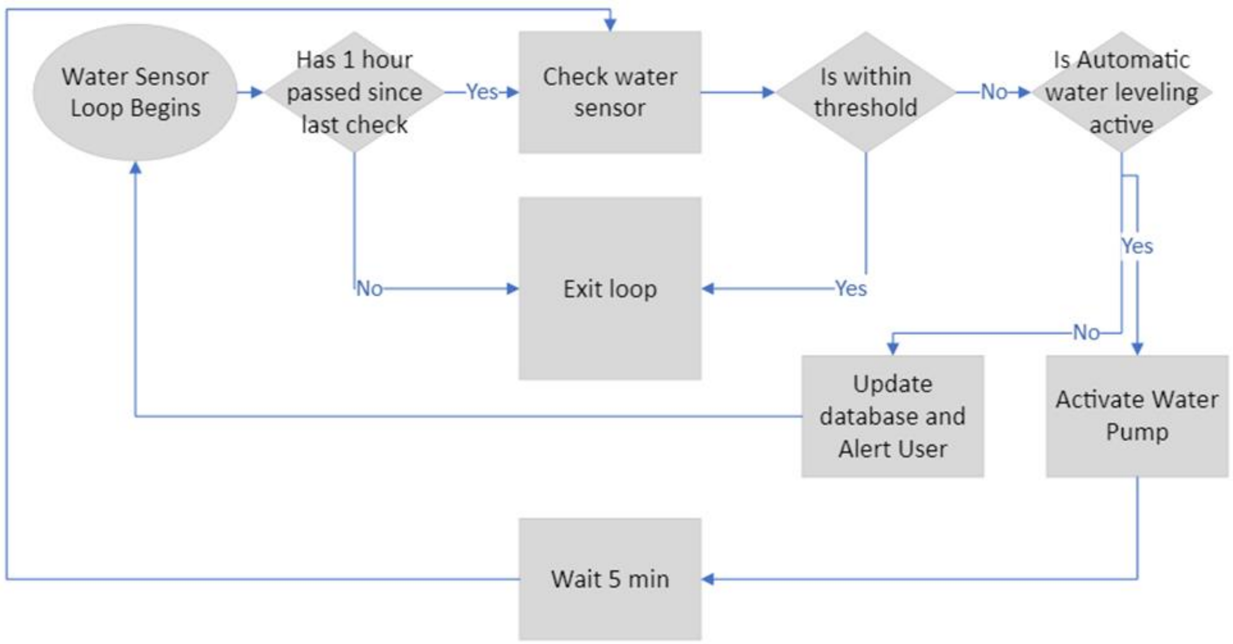


Figure 6.4c

The next loop that occurs after the MCU exits the water sensor loop is the pH sensor loop. The first thing the MCU will do is again check whether an hour has passed since the pH sensor was checked. If it has been checked recently the loop will immediately exit and move on the next one. If it has been more than one hour since the sensor has been checked the MCU will check the sensor and determine if it is within the pH threshold. If the pH sensor is within threshold the MCU will exit the loop. If the pH level is not within the threshold we need to determine if the pH is on the acidic side or base side. If the pH is too acidic(low) we will add the base until it is within the threshold. If the pH is too basic(high) we will add an acid solution. The solution pumps have a set amount of time that it runs, so once the pumps have run its course the loop will wait 5 min then check the pH sensor. This loop will repeat until the pH is within the limits. The pH level is essential to the health of the plant if the pH is not in its proper threshold the plant can quickly rot and die. The flow chart is shown below in **Figure 6.4d**.

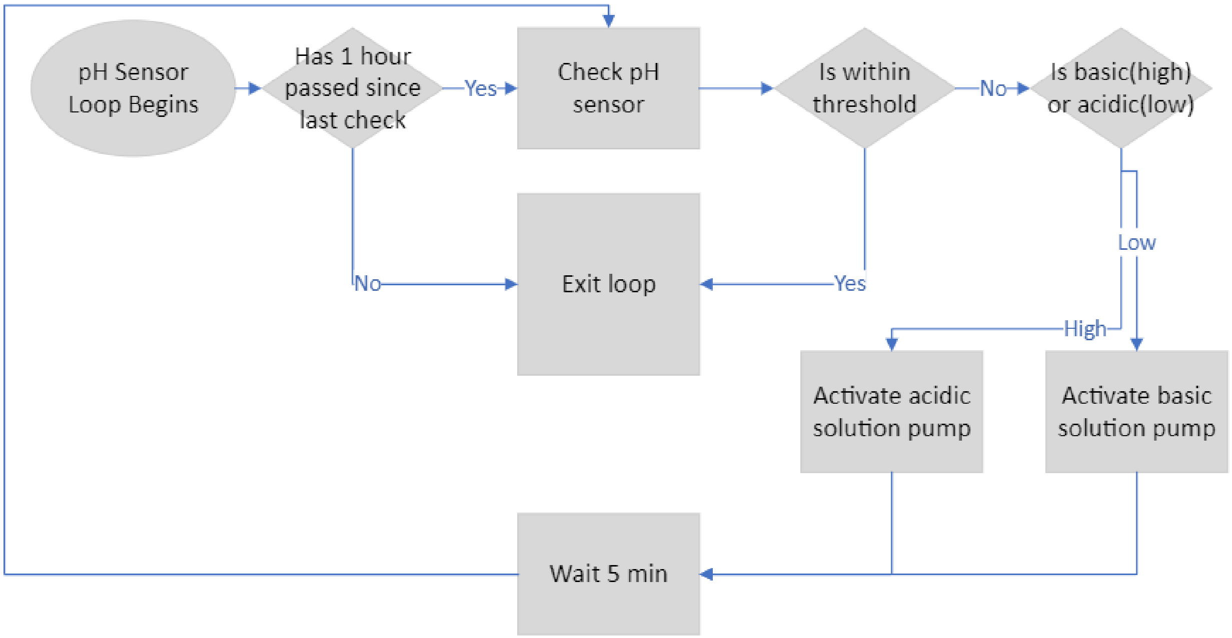


Figure 6.4d

Once the pH loop is complete the MCU will move on to the final loop, the TDS sensor loop. This loop runs the same course as the water and pH sensor. The MCU will first determine if the sensor has been checked recently, and if the sensor has been checked within an hour the MCU will exit the loop. If the sensor has not been checked within the hour the MCU will then check the TDS sensor, and if the TDS is within the threshold the MCU will terminate the loop. If the TDS levels are not within the threshold the MCU will activate the nutrient pump and run for a set amount of time. Once the pump has run its course the loop will wait for 5 min and check the TDS sensor. This loop will repeat until the TDS level is within the threshold. The TDS level being within range is just as important as the pH levels. Without enough nutrients within the water the plants will quickly die. This loop is set and the end since the plant can survive for longer without the proper nutrient level than pH. A flow chart for the TDS loop is shown below.

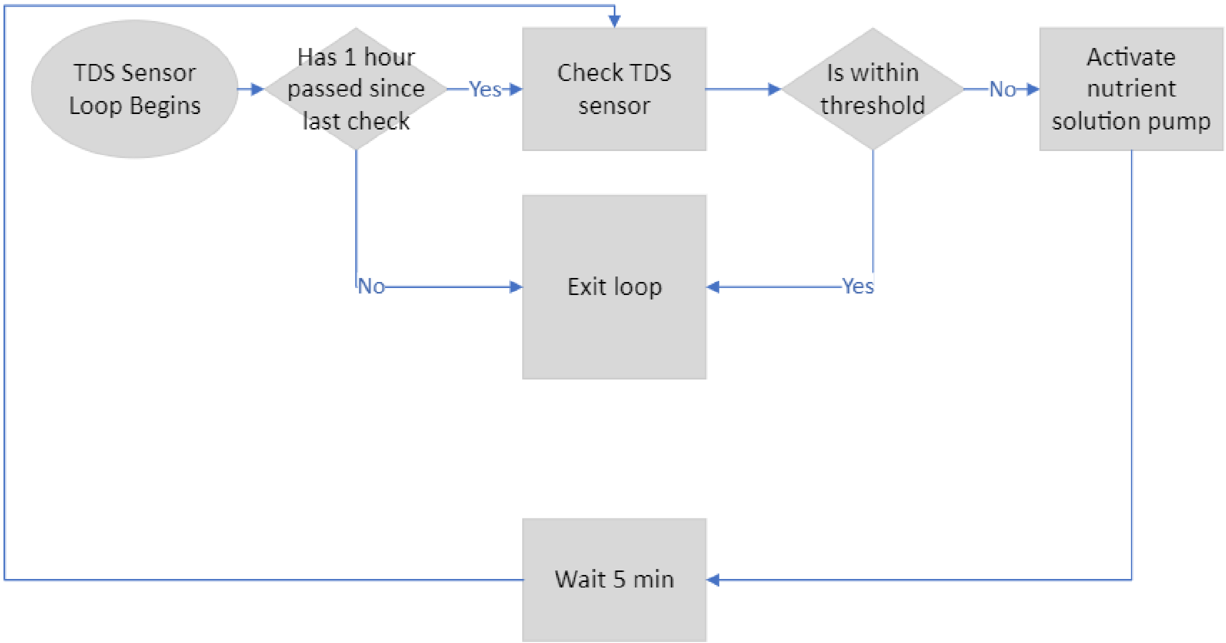


Figure 6.4e

There is actually one last sensor to check and update the database with its information but the microcontroller cannot actually influence the data for this sensor. The sensor is the Air/Humidity sensor. The following figure will show the loop used to check the sensor readings from the air humidity sensor and relay that information to the database though the microcontroller. There will be no adjustments of resources.

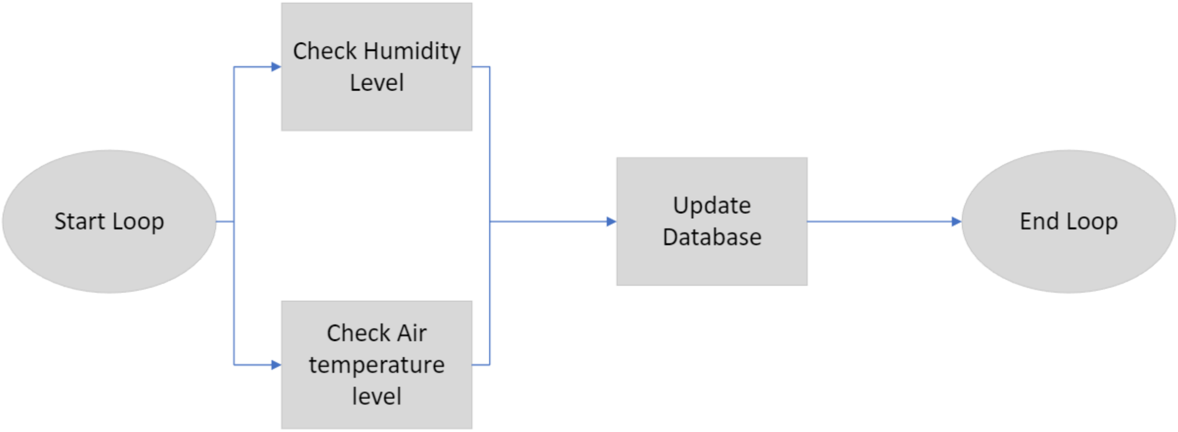


Figure 6.4f

The next part of the system design that we will be discussing is the database system. In this database we will be using foreign keys and primary keys for data verification within our program. Below is a relationship diagram for our database in figure 6.4g.

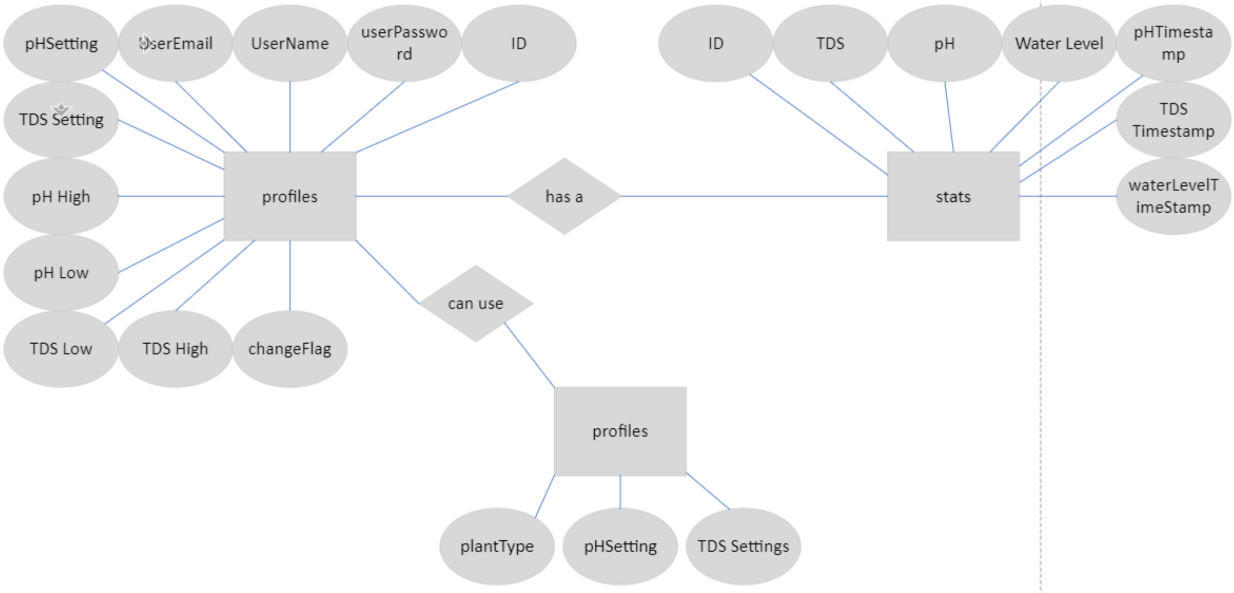


Figure 6.4g

As seen in the previous figure our database is fairly simple in its current state it only has three tables. Each user profile has multiple stat tables holding sensor data from the MCU. Multiple stat data tables are stored so we can create a graph that shows the transition of sensor data over a period of time and past data can be shown. Each time new stats are inserted the ID will increment. We also have a flag in the profiles section that is modified whenever the TDS or pH ranges are modified. This flag will allow us to check if these ranges are changed on the website's side and make sure we can update these ranges on the Arduino. This relationship diagram may be changed in the future as we add more features and will possibly need more sections for storage of webcam data for a timelapse.

In the next part of this section, we will be going over the design of our website. The website should be simple in design as to create an ease in flow and to not confuse the user. The goal of this website is to make it easy to monitor the plant while maintaining its simplicity. The most important part of this website would be the database as the database is what is communicating between the raspberry pi and the website. The database will need to return information to the MCU while also receiving information to display on the website. The figure below shows the flow of the website from the user to the database.

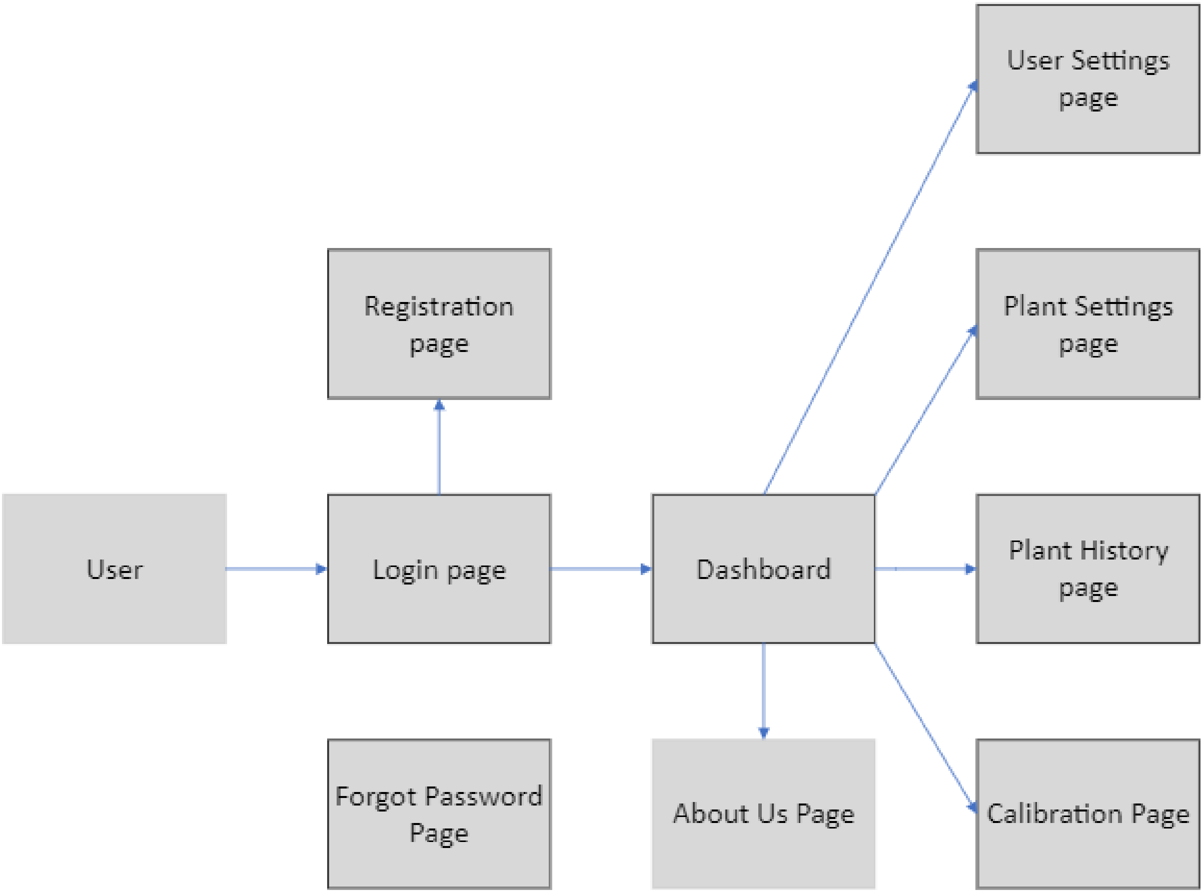

User Settings page
Registration page
Plant Settings page
User
Login page
Dashboard
Plant History page
Forgot Password Page
About Us Page
Calibration Page

Figure 6.4h

7. Project Prototype Testing Plan

Chapter 7 will cover our team's plan for testing the prototype of our project. Since our project heavily relies on showing that our hydroponic system can efficiently grow plants with a higher yield at a faster rate than traditional growing methods, it's important to ensure that all testing and assembly is finalized earlier than what would be required for a typical senior design project since the plants will require time to grow. We'll discuss both the plant and hardware testing environment, the testing of all the specific hardware components that make up the automation of our hydroponic system, and finally testing of the software that is required.

7.1 Plant Testing Environment

To have a real understanding of the efficiency of our system, we need to make sure that we can test and assess the plants health by qualitative and quantitative measures. This is why we plan on testing the plants and compare them with a control hydroponic system that is not automated.

7.1.1 Control Sample

We want to compare our system to a regular hydroponic system to be able to compare it to our results. Our expectation is that the Automated system will be more efficient while yielding better quality plants.

Thankfully, one of our members already owns a Deep Water Culture Hydroponic system that will be used as a control sample (**Figure 7.1.1a**). This system works by turning on an LED light for 14 hours/day, while a water pump on the tank cycles the water to oxygenate the roots. For this system however, the user needs to maintain the nutrients and water level themselves. We will follow the instructions as given by the manufacturer, refilling the 3 Liter tank once a week, with 15 mL of both the A and B nutrient solutions.

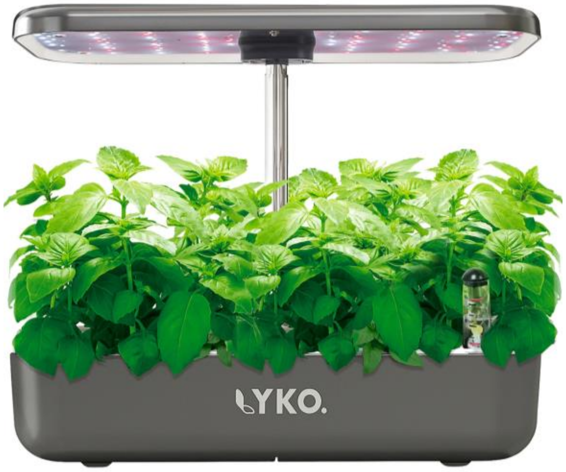


Figure 7.1.1a

7.1.2 Seed Germination Process

Due to the fact that we are working with plants, we need to have into consideration the time sensitivity of our project. That is why every day counts, and one of the slowest processes in our timeline is germinating the seeds that will later become the plants we will use.

We will lay our Rockwool cubes (**Figure 7.1.2a**) inside the starter tray with a humidity dome(**Figure 7.1.2b**). Rockwool cubes will be used because they are really good at absorbing humidity while providing a really good structure for the plants to develop. By laying the seeds in a really humid environment, we speed up the process and can sprout the first leaves in 5 to 7 days. Additionally, the tray will be warmed up by a heat mat.

Figure 7.1.2a **Figure 7.1.2b**

Our plan is to have a wide variety of herbs and leafs vegetables already started so we can use them if needed: basil, cilantro, lettuce and spinach.

As seen on **Figure 7.1.2c,** we have been able to start seed by provisioning an extremely humid environment while maintaining a temperature between 75-85 degrees Fahrenheit. These temperatures are monitored by a temperature sensor directly inserted into the plants Rockwool, which is the closest to the seeds. (**Figure 7.1.2d)**

Under these conditions, seed can become seedlings in 5 to 7 days, saving us time and money during the duration of the project.

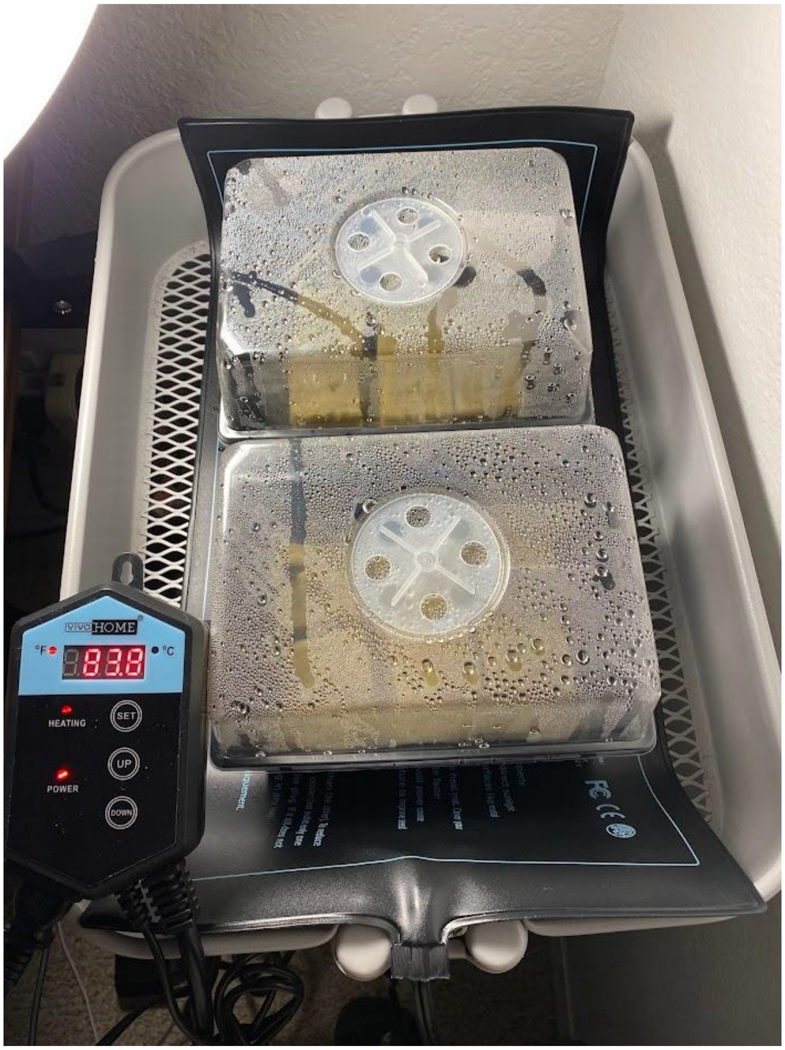


Figure 7.1.2c

The sprouted seedlings would be ready to be transported to our main growing system. The rockwool is extremely useful for the transfer because it can be easily opened up without having to damage the roots. As seen in **Figure 7.1.2d and 7.1.2e ,** after just 3 days, the lettuce and basil seeds are almost ready to be transferred.

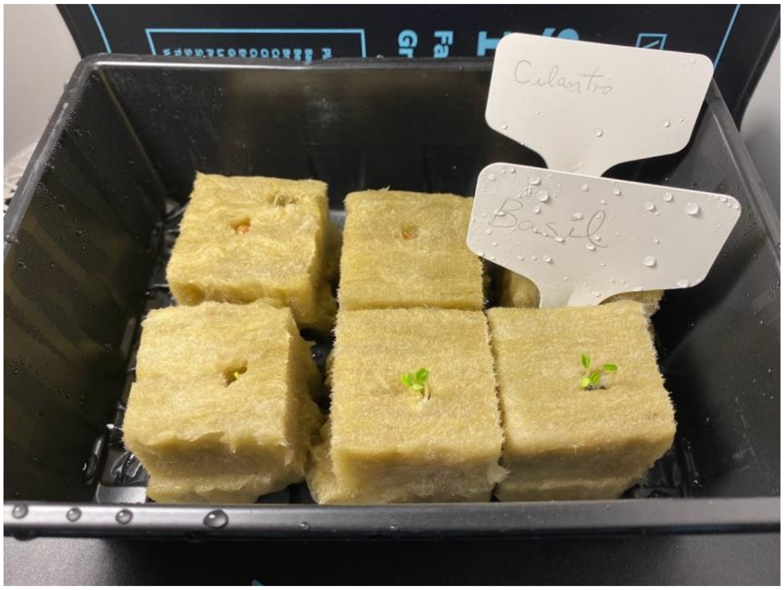


Figure 7.1.2c

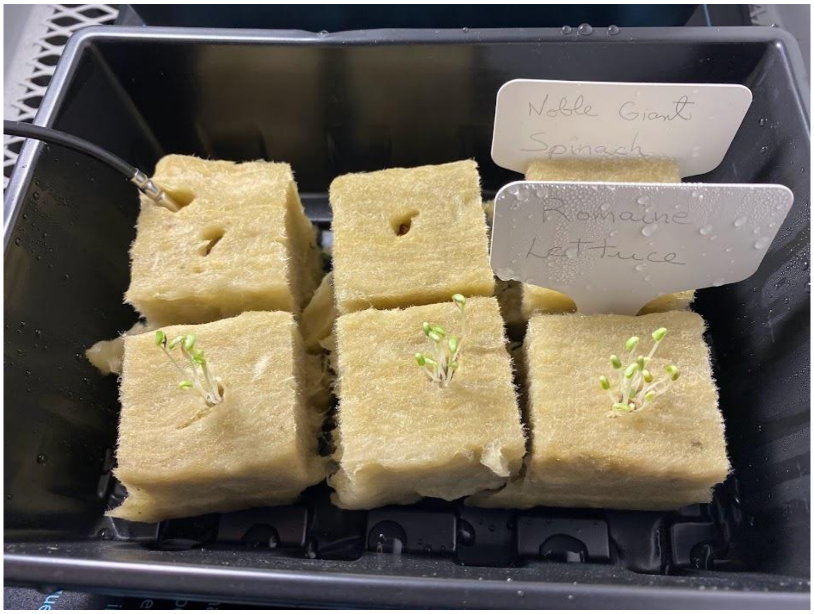


Figure 7.1.2d

After germination has concluded, we do not want to start the normal concentration of nutrients because it could damage the plants. We plan on slowly introducing nutrients into the plant until it reaches the recommended concentration. **Table 7.1.2** describes this process in milliliters per gallon. These measurements will be applied to both our hydroponic system and the control system.

Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**

BASE A	2 mL/G	2 mL/G	3 mL/G	3 mL/G	7 mL/G	7 mL/G	5 mL/G	4 mL/G
BASE B	2 mL/G	2 mL/G	3 mL/G	3 mL/G	7 mL/G	7 mL/G	5 mL/G	4 mL/G

Table 7.1.2

7.1.3 Measuring Plant Height

First, we will make sure to measure the plant height progress in both the control and Automated hydroponic. This method would only work for plants that grow vertically. We will use a measuring rod starting from the base of the plant, following the main stem up to the top. After, we can input the data to our sheet and calculate growth date with this basic formula:

$$Height\ growth\ rate = \frac{Height_N - Height_{N-1}}{days}$$

7.1.4 Measuring Plant Weight

Measuring the height is not viable for plants that grow around their center or horizontally. This is why we will measure weight for both herbs and leaves. This method is quite dangerous for regular soil plants because it involves tanking the plant out of its pot, exposing its roots. However, hydroponics is quite easy because the roots are not attached to a soil.

We will remove the plant from its hydroponic pod and weigh it on a small digital scale. The same process for collecting data as for height will be used.

$$Weight\ growth\ rate = \frac{Weight_N - Weight_{(N-1)}}{days}$$

7.1.5 Visual Testing

Our webcam will track the plant’s growth by taking a picture every 30 minutes and creating a timelapse. We will replicate this process in our control system with a camera set up. We concluded that 30 minutes is the most optimal time in terms of capturing significant changes while saving software resources.

Additionally, on the software side, we are planning on implementing a computer vision algorithm that can track leaf discoloration (**Figure 7.1.5)** so we can send a warning to the user. The main challenge will be having an evenly white background that allows for accurate color masking.

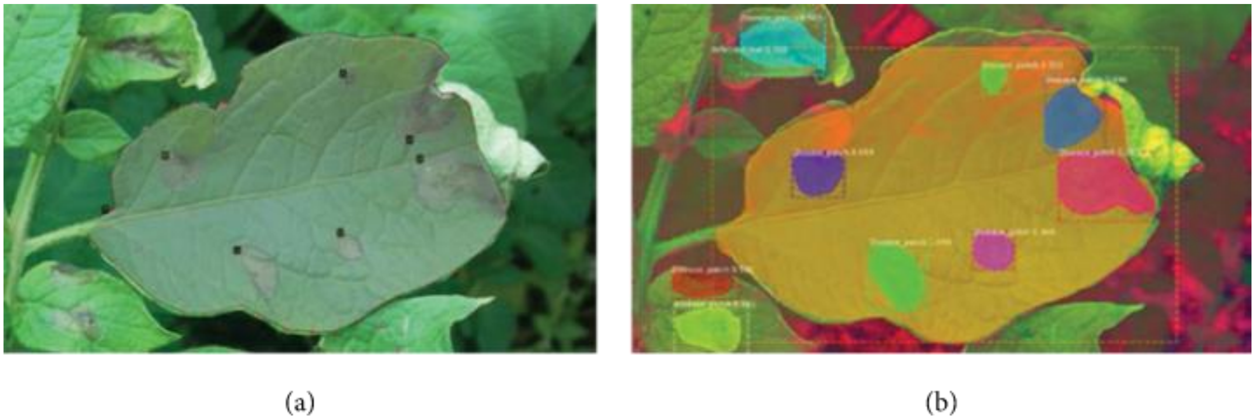

(a) (b)

Figure 7.1.5

7.2 Hardware Test Environment

The hardware will be located most of the time in a control environment, the setup will be inside the house in a dedicated room to have a better control of temperature and humidity. We will monitor temperature and humidity via the sensor, but we will incorporate a home temperature and humidity measuring device to ensure that the sensor is correct, by using the webcam to see the home device temperature and humidity reading.

The main tank will be sitting on the wagon flat surface for easy transportation, wagon dimension is 41” (L) x 20” (W) x 20”(H). The only time it will be moved is if the group needs to test the system to UCF to Analyze. Located on the floor the 10-20 gallon reservoir tank will be right next to the main tank. There will be four rows of floating Styrofoam with plants. The plants will be floating freely on the surface of the main tank with the nutrient and pH mix.

This room will be dedicated as the “ The Integration Room ", it will contain equipment for troubleshooting, such as Oscilloscope (Oscope), multimeter, function generator and the solder station. This room will have a surge protector for 120 AC users. The Oscope will be used to analyze circuits on the CCA, looking at signals to verify the correct signals are the correct output. Also for troubleshooting individual(S) components when the circuit is not functioning correctly, and analyzing faulty components. Solder station will be utilized for quick changes or modification to the circuit so we can have real time testing.

Our group will be utilizing a UCF solder oven to assemble the Design CCA or another option. I can utilize my company solder flow station because I am trained on this flow station and at my job I have all the equipment to complete the assembly of the Design CCA. This solder flow station is capable of flowing 500 pin BGA parts. Also I debugged my circuit for the start up of the Design CCA.

All the circuit CCA will be mounted and enclosed in a test Bud box. The box will have cut outs for interfacing with USB, Ethernet, HDMI, Micro USB etc. This Bud box will be portable, and easy to connect and disconnect connectors to the Hydroponic system. This will give us a portable transportation Hydroponic system on rugged wheels.

The Integration Room will be used to test all individual sensors and pumps because this is where the environment for the plant to grow. This would be the perfect condition to test individual components of the hydroponic system. The water nutrient, pH, would be ideal to test in this environment. The individual sensor has specifications for operating temperature, going below the minimum and maximum temperature will give false data.

We will do our integrating of the hardware and software in the Integration Room. Do an initial bring up check on the Design CCA to make sure the CCA powers up and all the voltage rail comes up. Then software engineers can begin to run the code and control the hardware of the hydroponic system. This will be the time for Electrical and Software engineers to work together to resolve any problem during bring up. Software can begin to debug their code.

The final step after electrical and software have completed their work. We will perform the Acceptance Test Plan (ATP), this will test individual functions, such as sensors, and pumps to make sure they are functioning. Test for nutrient level, pH sensor to make it fail, the pumps should begin to fill the desired amount of nutrient and pH, also it should warn the user via web server and the. For the water sensor remove water below the minimum and above the maximum and observe to see the reservoir water getting pumped into the main tank and back to the desired water level. All the testing is done in the Testing Environment.

7.3 Hardware Specific Testing

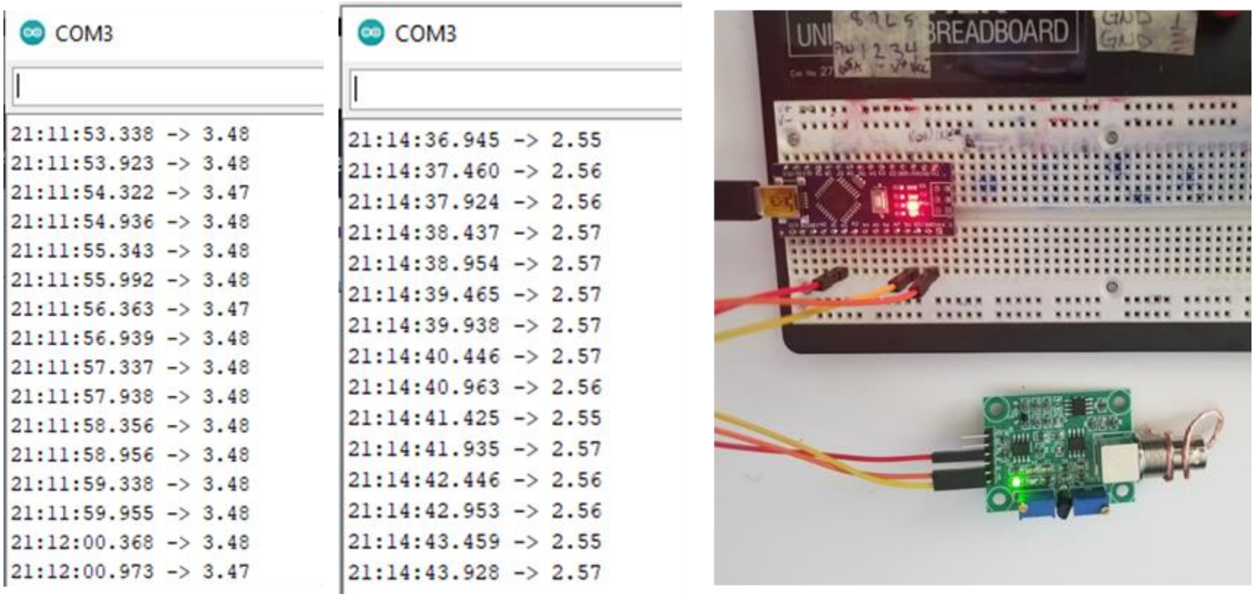
The specific hardware test will require testing individual components to ensure that it will function properly and operate at the minimum and maximum temperature for each component specification. This will ensure when assembling the Hydroponic system that each component was functioning properly. This will require a test plan on how to test each component. We will begin to test these sensors with the raspberry pi when it arrives from the vendor.

7.3.1 Water conductivity sensor

This conductivity sensor is used to measure conductivity in the water solution and determine the purity or impurity of the water. The nutrient solution carries salt that will conduct an electrical current. The sensor included two nutrient packages for testing the accuracy of the conductivity sensor. Also I tested the sensor to see the output change in the analog voltage signal to see how accurate the two sample nutrients were. This sensor is crucial for measuring the nutrient in the water, it will be critical for the plants health to have the right amount of nutrient in the water. The sensor operating temperature range is +50C to +86C, the room is temperature controlled so the conductivity sensor will be within its function range. We will begin to test this sensor with the raspberry pi when it arrives from the vendor.

7.3.2 pH sensor

The pH sensor plays a critical role in measuring the water nutrient, it measures the acidity of the water. The sensor must be calibrated with use of a liquid substance called buffers to get the correct reading. For our sensor we used the GAOHOU E-201-C PH sensor with a pair of trimmers which we could use to adjust the analog reading offset of the ph sensor which will be necessary when calibrating this sensor. This sensor has an on board voltage regulator which can be powered using a 3.3 to a 5.5 voltage input. Initially we want to calibrate our sensor by shorting the external part of the BNC connector by connecting the center of the BNC connector with a running calibration code. We can then read the analog output and display the voltage on the serial monitor. The voltage is found by multiplying the analog value by 5/1023. We then want to adjust the trimmer until the value is set to 2.5 volts because a pH of 7 means 2.5 volts. In our test we failed to get the trimmer to a value of 2.5 so we will have to add an offset in the code as seen in the **Figure 7.3a** below.



```
21:11:53.338 -> 3.48
21:11:53.923 -> 3.48
21:11:54.322 -> 3.47
21:11:54.936 -> 3.48
21:11:55.343 -> 3.48
21:11:55.992 -> 3.48
21:11:56.363 -> 3.47
21:11:56.939 -> 3.48
21:11:57.337 -> 3.48
21:11:57.938 -> 3.48
21:11:58.356 -> 3.48
21:11:58.956 -> 3.48
21:11:59.338 -> 3.48
21:11:59.955 -> 3.48
21:12:00.368 -> 3.48
21:12:00.973 -> 3.47
```

```
21:14:36.945 -> 2.55
21:14:37.460 -> 2.56
21:14:37.924 -> 2.56
21:14:38.437 -> 2.57
21:14:38.954 -> 2.57
21:14:39.465 -> 2.57
21:14:39.938 -> 2.57
21:14:40.446 -> 2.57
21:14:40.963 -> 2.56
21:14:41.425 -> 2.55
21:14:41.935 -> 2.57
21:14:42.446 -> 2.56
21:14:42.953 -> 2.56
21:14:43.459 -> 2.55
21:14:43.928 -> 2.57
```

Figure 7.3a pH sensor calibration

The figures show the before and after the trimming of the ph sensor after shorting the BNC connector, and shows the connection of the Arduino nano to the BNC board after trimming is completed.

After we are finished trimming the BNC board further calibrations still need to be completed. In the next step we will need to create buffer solutions with a pH of 4.00, 6.86, and 9.18 in three separate containers. For this test we used purified water and mixed each buffer solution in 250ml of water and waited until the mixture was completely dissolved. Once the mixture was complete, we would connect the probe and rinse it in purified water and test it in each of the three solutions. We would then tweak the code until it is as close to the buffer solution values as possible. The testing is shown in the **Figure 7.3b** below.

COM3

```
17:53:55.318 -> 3.99
17:53:56.625 -> pH Val:
17:53:56.625 -> 4.00
17:53:57.925 -> pH Val:
17:53:57.925 -> 4.01
17:53:59.226 -> pH Val:
17:53:59.226 -> 4.03
17:54:00.532 -> pH Val:
17:54:00.532 -> 3.98
17:54:01.836 -> pH Val:
17:54:01.836 -> 4.03
17:54:03.097 -> pH Val:
17:54:03.097 -> 4.00
17:54:04.403 -> pH Val:
17:54:04.403 -> 4.00
```

COM3

```
17:55:49.844 -> 6.84
17:55:51.147 -> pH Val:
17:55:51.147 -> 6.88
17:55:52.442 -> pH Val:
17:55:52.442 -> 6.89
17:55:53.743 -> pH Val:
17:55:53.743 -> 6.87
17:55:55.046 -> pH Val:
17:55:55.046 -> 6.90
17:55:56.346 -> pH Val:
17:55:56.346 -> 6.87
17:55:57.645 -> pH Val:
17:55:57.645 -> 6.88
17:55:58.942 -> pH Val:
17:55:58.942 -> 6.87
```

COM3

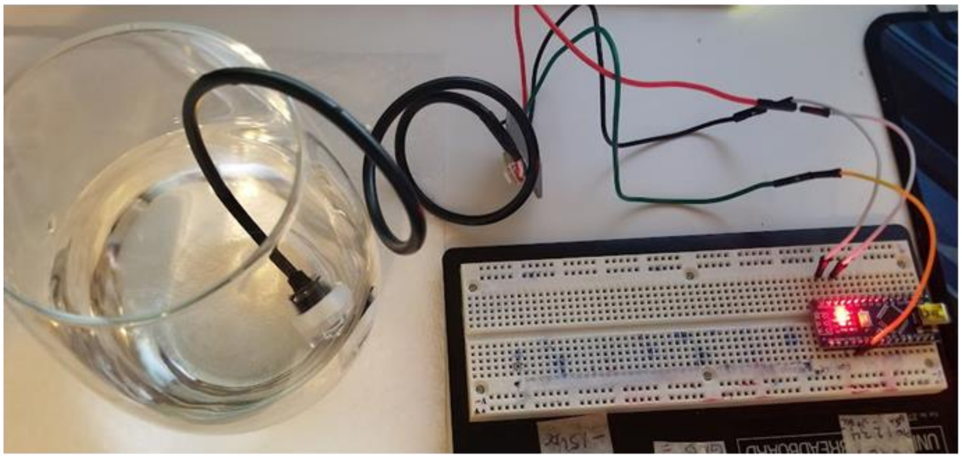
```
18:03:24.567 -> 9.20
18:03:25.867 -> pH Val:
18:03:25.867 -> 9.18
18:03:27.164 -> pH Val:
18:03:27.164 -> 9.21
18:03:28.507 -> pH Val:
18:03:28.507 -> 9.22
18:03:29.800 -> pH Val:
18:03:29.800 -> 9.21
18:03:31.108 -> pH Val:
18:03:31.108 -> 9.21
18:03:32.414 -> pH Val:
18:03:32.414 -> 9.24
18:03:33.710 -> pH Val:
18:03:33.710 -> 9.26
```

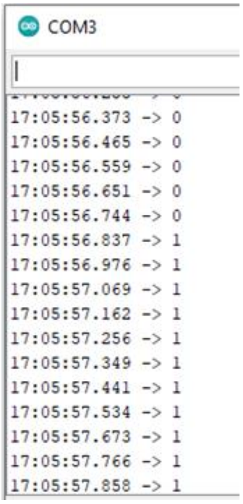
Figure 7.3b ph sensor calibration

7.3.3 Water Level Sensor

The water level sensor uses a photoelectric water liquid level sensor. The sensor DIP switch controls the output voltage of the signal terminal. When the DIP switch is set at 5V, the high level is 5V. When the DIP switch is set at 3V, the high level is 3.3V. This sensor came in a kit, so we were able to test it with a Mega 2560 microcontroller. By putting the water level sensor in half a bucket of water then removing half of the water, the result from this test proves the sensor is working properly. Initially we were planning on using this sensor, but we found the sensor would not be appropriate for this project. The sensor module would need to be completely submerged at a depth the sensor could not handle and would not reach lengthwise.

Instead, we decided to go with a different sensor: the CQRobot Ocean contact water level sensor. This sensor also uses a DIP switch to control the output voltage of the signal in a similar method to the other sensor. This sensor has no mechanical parts so needs no calibration and has high resistance to temperature and corrosion. This product was tested on two different microcontrollers: the Mega 2560 microcontroller and the Arduino nano that will be placed in our pcb. In both cases the boards are connected through a serial connection to a computer and displays whether the probe is submerged or not though a display on our computer. If the probe is not submerged the monitor will return a zero and if it is submerged the probe will return a one. In both test cases the sensor was able to properly identify whether the probe was submerged or not although there is some delay as water may need to dry off the probe. The testing setup for this sensor is shown in **Figure 7.3c** below.



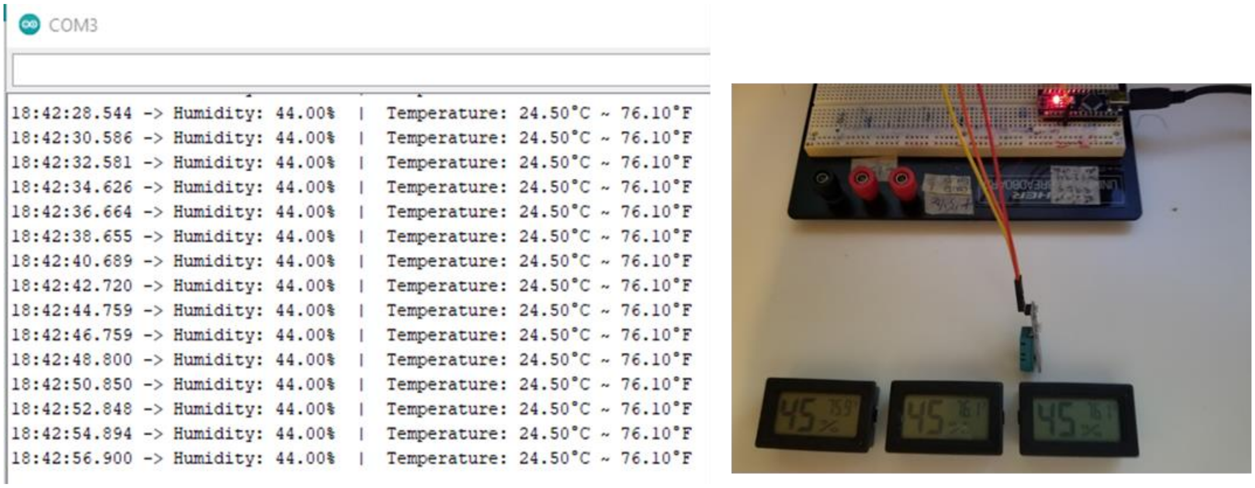


```
17:05:56.373 -> 0
17:05:56.465 -> 0
17:05:56.559 -> 0
17:05:56.651 -> 0
17:05:56.744 -> 0
17:05:56.837 -> 1
17:05:56.976 -> 1
17:05:57.069 -> 1
17:05:57.162 -> 1
17:05:57.256 -> 1
17:05:57.349 -> 1
17:05:57.441 -> 1
17:05:57.534 -> 1
17:05:57.673 -> 1
17:05:57.766 -> 1
17:05:57.858 -> 1
```

Figure 7.3c Water Level sensor test

7.3.4 Air Temp/Humidity Sensor

We are using a DH11 temperature humidity sensor connected to the Arduino nano. It works by detecting changes that alter electrical currents or temperature in the air. The sensor operating temperature is -25C to +105C, these temperatures well in range of the room the plant will grow in. We tested the sensor inside the room where the plant will be to see what the humidity was in that room. We have external humidity temperature sensor devices that we will be using to test the room and compare the data to the information we are getting from the Arduino. Currently we have only tested the DH11 in an indoor setting under fixed temperature and humidity, but more tests will be performed in other locations to ensure the temperature humidity range of the sensor. Figures of the test will be shown in **Figure 7.3d** below.According to these figures the DH11 and external sensors match up quite nicely proving the sensor to be very accurate.



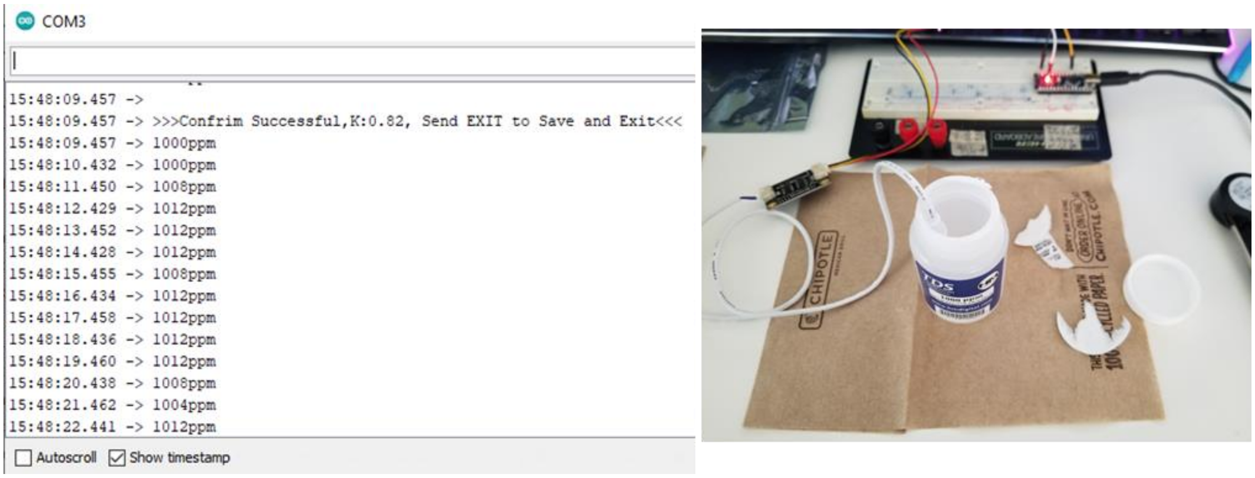
```
18:42:28.544 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:30.586 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:32.581 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:34.626 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:36.664 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:38.655 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:40.689 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:42.720 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:44.759 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:46.759 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:48.800 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:50.850 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:52.848 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:54.894 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
18:42:56.900 -> Humidity: 44.00%  |  Temperature: 24.50°C ~ 76.10°F
```

Figure 7.3d Temperature/Humidity sensor test

7.3.5 TDS Sensor

The TDS Sensor detects the Total Dissolved Solids (TDS) levels in the water which can be used to measure the water quality, it is measured in Parts Per Million (PPM). TDS Measurement Range: 0~1000ppm, the sensor operating temperature range is +50C to +86C, the room is temperature controlled so the conductivity sensor will be within its function range. To test the tds meter calibration will be required before measurement. It is also recommended to have a water temperature sensor for temperature compensation in order to improve accuracy. Normally TDS would be half of the electrical conductivity value. So for our calibration we would need a solution of known electrical conductivity or TDS value.

For our use case we have a standard buffer solution of 1000 ppm for our TDS value if converted to electrical conductivity value will be converted to 2000 us/cm. Currently we do not have the water temperature sensor on hand, but the k value for the solution will be fixed after it is found and can be adjusted to a change in temperature. For this first calibration we will be using a meat thermometer to measure the temperature of the solution. Then we can insert a clean probe in our 1000 ppm solution and run the calibration command and enter the solution to find the K value. A figure of this experiment is shown in **Figure 7.3e** below.



```
15:48:09.457 ->
15:48:09.457 -> >>>Confrim Successful,K:0.82, Send EXIT to Save and Exit<<<
15:48:09.457 -> 1000ppm
15:48:10.432 -> 1000ppm
15:48:11.450 -> 1008ppm
15:48:12.429 -> 1012ppm
15:48:13.452 -> 1012ppm
15:48:14.428 -> 1012ppm
15:48:15.455 -> 1008ppm
15:48:16.434 -> 1012ppm
15:48:17.458 -> 1012ppm
15:48:18.436 -> 1012ppm
15:48:19.460 -> 1012ppm
15:48:20.438 -> 1008ppm
15:48:21.462 -> 1004ppm
15:48:22.441 -> 1012ppm
```

Figure 7.3e ppm calibration test

7.3.6 Air Pump

The oxygen air pump creates oxygen in the water for the plants roots, it also prevents bacteria and algae from forming in the tank. The bubbles passing through the water create a current, keeping the water moving. The current helps to keep the nutrients dissolved in the water. Testing the pump, I submerged the air pump to the bottom of the bucket, then applied 12Vdc to the red(+) and black(-) wire , this powers up the air pump. I see an air bubble coming from the pump, so the pump is working properly.

7.3.7 Nutrient and pH Pump

The nutrient and pH utilize the same type of pump, these two pumps are to replenish the nutrient and pH via the Water conductivity sensor and pH sensor. The sensor will control when the nutrient and pH are released from the pumps. These brushless motors utilize 12Vdc to power the up, I hook the pumps red and black lead to a 12Vdc power supply, both pumps were working properly.

7.3.8 Water Pump

The water pump replenishes the main tank by pumping the water from the reservoir back to the main tank. This is controlled by the water level sensor, when the water level gets low in the main tank the sensor will turn the water pump on to refill the main tank. When the level is back to the desired position the pump will turn off. The water pump utilizes 12Vdc to power the up, I hook the pump's red and black lead to a 12Vdc power supply and the pump is pushing water through, the pump is working properly.

7.3.9 Water Temperature sensor

The water sensor we will be using is the GAOHOU DS18B20 waterproof temperature sensor. This water temperature sensor is mainly used to calibrate the TDS sensor and ensure its accurate readings. This sensor can also be used to ensure the health of the roots and make sure it is sitting in appropriate temperatures. Whenever the TDS sensor needs to update its readings the water temp sensor will run and give its results to the TDS sensor. To test this sensor we will be comparing it against a thermometer and checking if the results from the sensor are accurate.

7.4 Software Test Environment

An automated hydroponics system is a very complex system and people often misjudge the system and think of it as simple. Many small features and requirements can easily be overlooked in projects like these. Both our software and hardware systems need to work together in perfect tandem or parts can break and the plant will die. Like we said there are many small features that are overlooked and need to be tested. Therefore, we must go over and incrementally test all parts of the software system before we test the overall system. The time it takes for this testing can vary depending on the severity of the problem that we run into.

We have multiple options when it comes to the location one location is the senior design lab which all senior design students have access to as long as we notify the professors. In this room we can work on testing the software without restriction and in safety. The lab also offers testing equipment, a lot of space and a controlled environment to test and implement new features. The room comes with a locker to put valuables aways safely and limited groups can work there because of covid precautions. We can also leave the project there as there are secure lockboxes to keep our project safe. The other location we have an option of using is one of our teammate’s houses. This will provide complete isolation from any other groups so we can work without the interference of any other teams and professors. Our group also owns or was given by the school the main tools needed to test and troubleshoot any problems that may come our way. Testing at our teammates' place has the main benefit of being able to easily connect to the WI-FI and not having to deal with the UCF WI-FI.

The two most important pieces of hardware that controls and transmits data from the rest of the system is the MCU and its attached Wi-Fi module. Our main microcontroller will be the Raspberry pi 4 that runs a Linux based OS and can support a variety of languages from Scratch to Java, but in our case, we will mainly be coding in JavaScript or Python. The Raspberry Pi has a large strongly driven community of hobbyists and coders who work with this board. This allows us to quickly find resources for all types of different projects. There are many different libraries and examples to help us test new features that we want to include. The Raspberry Pi makes many features possible from computer vision to web development, the Raspberry Pi has the resources and processing power to make it happen.

The testing will be done through a direct serial connection to a computer allowing the microcontroller to print messages directly to the computer through a serial connection. We will be able to run code on an interface that looks like a console and any message will be outputted directly to our console on the computer. The serial connection allows intuitive programming so we can view any hiccups directly on our computer that we would need to troubleshoot. Once the LCD screen is tested and applied to our board, we can also display messages on there as our project advances.

For this testing we will not be doing anything complex all we will be doing is testing each individual part to determine how they communicate with the board. We will simply need to detect, gather and store information from our components and slowly try different combinations to see the components communicate. Then we will slowly need to develop our code to test more components at once until we have a completed product. One main dilemma that will need to be thoroughly tested is the set amount of time each pump should pump for. This time may vary from pH to the nutrient pump. We will also need to do through testing on all the different subsystems and how they communicate with one another. Such as, making sure the sensor values are correctly received by the microcontroller, Website, and sent into the database.

7.5 Software Specific Testing

In our hydroponic system we will need to have to commit many software tests for all parts of the web stack. We will describe all the different debug tools used in this testing and the scenarios that they will be used in. We will need to test all pages of our website piece by piece from the login screen to the sensor data. The system will need to be set up in a way that functions even without connection to the network. We will need to create contingencies in case network connection is lost. The system needs to skip over any functions that send data over the network to prevent the system from crashing and stalling. Several tests will be run where the sensors will test for data when the network is lost mid-way. Once the system can constantly run without connection that test will be passed. We can also use the data sent to the database as a debug to see what type of values can be sent and retrieved from the database.

One test case will be creating a new user and adding the information to the database. This will test the connectivity of our microcontroller and whether our API queries are properly sending data to the database. We will expect the user's information to be added to the database and troubleshoot otherwise. When we first set up the system, we needed to test whether our hard coded Wi-Fi settings correctly connected the board to the internet. Once our IP and port settings are configured in the Wi-Fi, we can ping the server. If the ping is successful, we know the Wi-Fi is set up correctly and that we can access our web server. We will need to run this test several times.

Another test we will have to run will be on our TDS and pH threshold values. This test will be done so we know that proper values are used when we analyze the sensors. The default threshold values will be hard coded into the microcontroller, and user values for a particular plant will be stored in different variables on the system. Each time the test is executed the default and user values will be compared. If they aren't the same the default values will be updated to the user variable values. After this comparison a post request would be sent to the server for verification.

Another test we can try is testing if the user can feed and access data from the database. To test this, we will write a query to the user profile that will modify the sensor data. This query will be sent without the hardware and database being merged so we can simulate the sensor data. If this is complete the data will be added to the database, and we will know the MCU can feed sensor data. In the second part of this test, we will be testing whether we can access this data. In this test we will write a query to retrieve the sensor data stored in the database. If this test is completed properly, we will be able to access the data we want to retrieve the query.

The next test we can run is a simple test of the login page. We will need to log in to a user using their login information, and test whether our information will authenticate correctly. The other test we can do on the login page is registration. We can try to register a new user and view if the new user information was sent to the database. We will also see if a confirmation email was sent out by the register page. The last test we can run is on the login page is the forgot password button. The user will type in an email to an account that exists on that database so that the api can run a query to search for users with that email. Once the user is found, the database should update the user password with a randomized password and send out that email to the user. If the user can log in with the new password.

8. Administrative Content

This final chapter is a simple review of the milestones our team has set for ourselves throughout both the senior design 1 and senior design 2 semesters as well as our budget and financing plans. It may be subject to change depending on a multitude of different variables but is a sufficient assessment of what we expect.

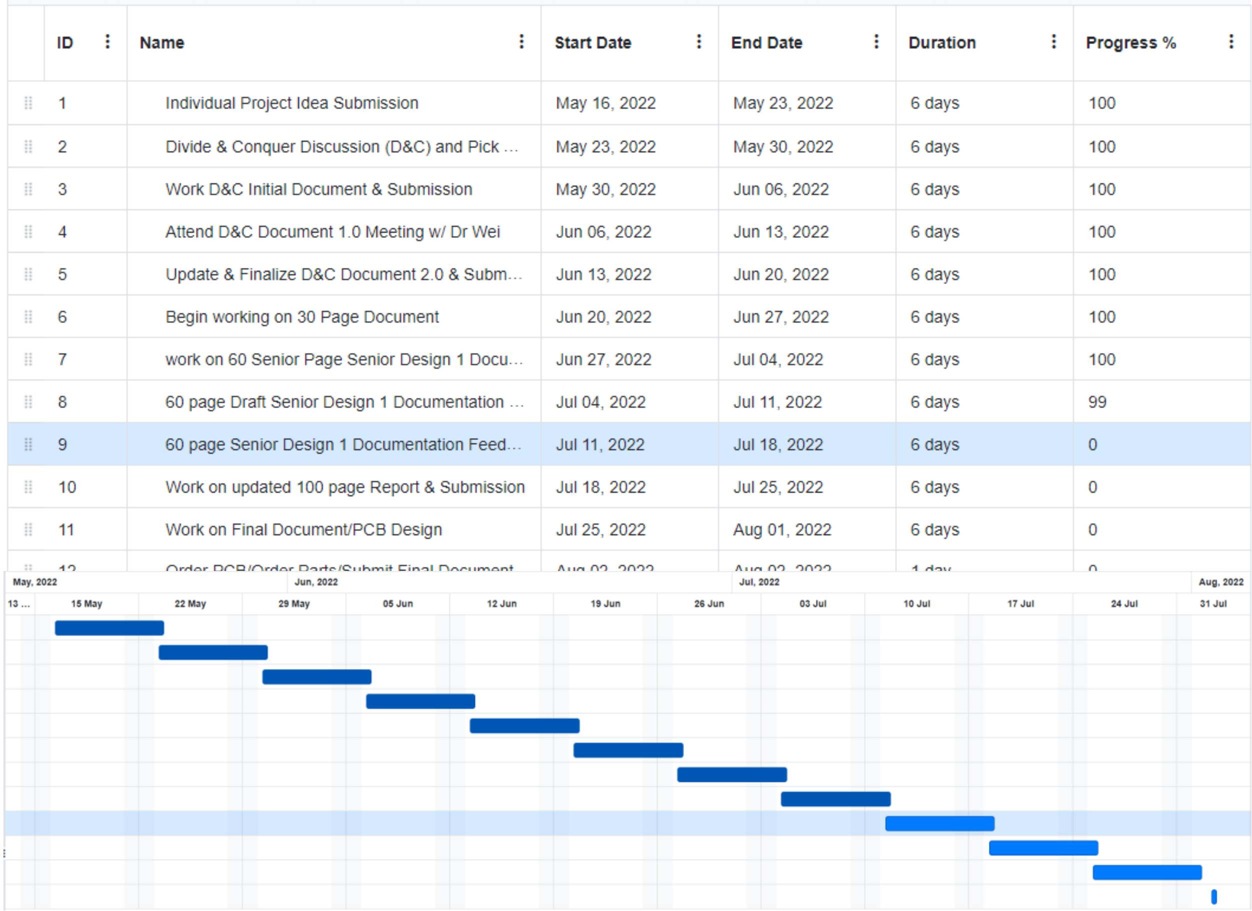
8.1 Milestone Discussion

The senior design project consists of two consecutive semesters in its entirety and requires full concentration and commitment in order to meet the project deadlines by the end of the course. As such, a carefully planned schedule of milestones should be incorporated into the report of our project to better help with planning a weekly list of tasks that should be completed and helps keep accountability to each member of the team. This list of milestones is used throughout each phase of the project from our initial project idea, to the actual submission and presentation and should be referenced periodically to measure our teams' progress. As can be seen in the senior design 1 milestone table shown below, the first twelve weeks consist of planning out our team's ideas for the project and making vital design decisions. Everything must be thoroughly documented and all research materials properly referenced.

The first three weeks consisted of discussing our ideas as a team to decide what kind of project we wanted to create and writing out the project idea in our first "Divide and Conquer" document. Week four was dedicated to the senior design bootcamp and meeting with our professors to discuss the "Divide and Conquer" document that we submitted on 6/5/2022, where we were given further guidance on how to improve upon the second and final version of our "Divide and Conquer" document, which was finished and submitted during week five on 6/19/2022. As of today's date, we have accomplished every milestone that we have set for ourselves from week one to week eight. Weeks nine through twelve are the last of the first semester and entail finishing the final 120 page document that is to be submitted during the final week on 8/2/2022. It is also imperative that we create our custom PCB design and order all necessary parts before the end of the semester.

8.1.1 Senior Design I Milestones

Week #	Date	Activity
1	05/16/2022-05/22/2022	❖ Group 8 was assigned. ❖ Individual Project Idea Submission
2	05/23/2022-05/29/2022	❖ Group 8 Divide & Conquer Discussion (D&C) and Pick Project
3	05/30/2022-06/05/2022	❖ Group 8 Work D&C Initial Document & Submission
4	06/06/2022-06/122022	❖ Group 8 Attend D&C Document 1.0 Meeting w/ Dr Wei ❖ Senior Design Boot Camp
5	06/13/2022-06/19/2022	❖ Group 8 Update & Finalize D&C Document 2.0 & Submission
6	06/20/2022-06/26/2022	❖ Group 8 Begin working on 30 Page Document
7	06/27/2022-07/03/2022	❖ Group 8 work on 60 Senior Page Senior Design 1 Documentation
8	07/04/2022- 07/10/2022	❖ Group 8 60 page Draft Senior Design 1 Documentation Submission
9	07/11/2022-07/17/2022	❖ Group 8 60 page Senior Design 1 Documentation Feedback meeting ❖ Work on Schematic Capture.
10	07/18/2022-07/24/2022	❖ Group 8 Work on updated 100 page Report & Submission
11	07/25/2022-07/31/2022	❖ Group 8 work on Final Document. ❖ PCB Design, Check material and & component to make sure it is available
12	08/02/2022	❖ Order PCB ❖ Order Parts ❖ Submit Final Document



ID	Name	Start Date	End Date	Duration	Progress %
1	Individual Project Idea Submission	May 16, 2022	May 23, 2022	6 days	100
2	Divide & Conquer Discussion (D&C) and Pick ...	May 23, 2022	May 30, 2022	6 days	100
3	Work D&C Initial Document & Submission	May 30, 2022	Jun 06, 2022	6 days	100
4	Attend D&C Document 1.0 Meeting w/ Dr Wei	Jun 06, 2022	Jun 13, 2022	6 days	100
5	Update & Finalize D&C Document 2.0 & Subm...	Jun 13, 2022	Jun 20, 2022	6 days	100
6	Begin working on 30 Page Document	Jun 20, 2022	Jun 27, 2022	6 days	100
7	work on 60 Senior Page Senior Design 1 Docu...	Jun 27, 2022	Jul 04, 2022	6 days	100
8	60 page Draft Senior Design 1 Documentation ...	Jul 04, 2022	Jul 11, 2022	6 days	99
9	60 page Senior Design 1 Documentation Feed...	Jul 11, 2022	Jul 18, 2022	6 days	0
10	Work on updated 100 page Report & Submission	Jul 18, 2022	Jul 25, 2022	6 days	0
11	Work on Final Document/PCB Design	Jul 25, 2022	Aug 01, 2022	6 days	0
12	Order PCB/Order Parts/Submit Final Document	Aug 02, 2022	Aug 02, 2022	1 day	0

The senior design II milestones involve the assembly and testing of every component for our hydroponic system and debugging any issues that arise prior to the final presentation date. Since the plants also need time to grow, it was recommended that we make our best attempt to finish the project earlier than usual. For this reason, we intend on finishing the full assembly and testing of the system by week eleven.

8.1.2 Senior Design II Milestones

Week #	Date	Activity
1-2	TBD	❖ Prototype of Hydroponic Structure and Electronic are finalized and ready for Assembly ❖ Complete Order parts
3-6	TBD	❖ Solder Flow component onto PCB board. Test circuit and software to ensure it is working properly.
7-10	TBD	❖ Hydronic Assembly and PCB board completed, Ready for ❖ Testing
11-14	TBD	❖ Last Minute Corrections ❖ Prepare Assembly for presentation
15	TBD	❖ Final Project Presentation

8.2 Budget and Finance Discussion

Many different techniques and methods have been used for building hydroponic systems but the deep water culture hydroponic system is perhaps the simplest of all and would normally require a limited amount of money to fund the build. A generic deep water culture system would only require a reservoir to contain the water, net pots to hold the plants, an air pump connected to an air stone, the solutions needed to maintain the proper chemistry levels of the water, and the testers to measure pH, total dissolved solids, and electrical conductivity levels. Since our project introduces automation, further complications must be addressed since there are much more components involved. Additionally, the overall cost increases substantially since we have incorporated a grow tent and multiple sensors that are meant to automatically monitor the water level of the reservoir, the water's chemistry, and the surrounding environment's temperature/humidity.

Research has shown that the most expensive components of our project would have been the grow lights and our desired microcontrollers. We were able to save money on the budget since we already own those components, as well as a few others. The budget and finance portion of our project is still a work in progress as we have not completely finalized our choices for certain components. As such, the list of items and their estimated cost in the budget shown in **Table 8.2** shown below are subject to change as the project progresses. The project we are developing does not have a sponsor, so we have decided to split the cost evenly amongst the four of us. The current estimate for the total cost of our project is approximately up to $550.

Item	Quantity	Estimated Cost
Grow Tent	1	$90 - $120
Grow Lights	2	Owned
Heavy-Duty Tote (Reservoir)	1	$16
Plant Growth Nutrients	1	$39
pH Solutions	1	$21
Seeds	N/A	Owned
Grow Cubes	1	$11
Net Pots	Pack of 12	$8
Seedling Heat Mat	1	Owned
BeagleBone Black	1	Owned
Raspberry Pi 4 Model B	1	Owned
Micro SD Card (32 GB)	1	$10
Raspberry Pi v2 Camera Module	1	Owned
HDPE Plastic Mounting Panel	1	$22
LCD	1	Owned
Custom PCB	1	$10 - $50
Power Control Box Components	1	$30 - $40
Air Temperature/Humidity Sensor	1	Owned
pH Sensor	1	$37
TDS Sensor	1	$16
Water Level Sensor	1	Owned
Solution Peristaltic Pump	4	$10 - $18 each

Analog to Digital Converter	1	Owned
AC Current Sensor w/split transformer	1	$12
3D Printed Parts	1 kg materials	$20
Total Est. Cost		**$410 - $530**
Cost Per Member		**$103 - $133**

Table 8.2

9. References

2.1a Helmer, J. How the Coronavirus Pandemic Has Led to a Boom in Crisis Gardening. HuffPost Canada. 3 April 2020. Available online: https://www.huffpost.com/entry/seeds-crisis-gardening-coronavirus-food_n_5e85eca0c5b6f55ebf492212 (accessed on 20 June 2022).

2.1b Brewster, M. Canadians are Being Asked to Stop Panic-Shopping. Are They Listening? CBC News. 19 March 2020. Available online: https://www.cbc.ca/news/politics/covid-19-pandemic-coronavirus-border-panic-1.5502058 (accessed on 20 June 2022).

2.1c Helmore, E. Americans Turn to Home-Farming as They Fear for Their Food Supply. The Guardian. 6 June 2020. Available online: https://www.theguardian.com/world/2020/jun/06/food-insecure-americans-are-home-farming-as-covid-wreakshavoc-on-agricultural-chain (accessed on 20 June 2022).

2.1d Small Scale Gardner, Deep Water Culture (DWC) Ultimate Guide. Date of publication is not shown. Available online: https://smallscalegardener.com/hydroponics-farming-guide/hydroponic-growing-systems/deep-water-culture-hydroponic-system/ (accessed on 30 June 2022).