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

UCF

UNIVERSITY OF CENTRAL FLORIDA

2. Project Narrative

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.

Hydroponic is a growing method that only uses water, which saves 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.

2.1 Goals and Objectives

Our main goal is to free the user of 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.

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 image 2.1a).
The rapid growth DWC offers will be useful when testing in the limited time we have to finish and test our design.

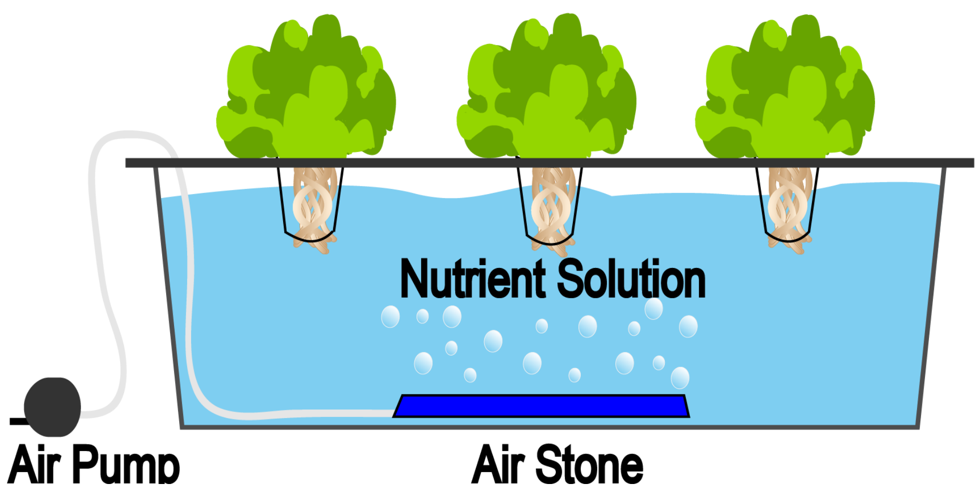


Image 2.1a

How Autonomous?

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 Competition

After analyzing the current market on hydroponic solutions, we found that the main products are **standard hydroponic gardens** (Image 2.2a). 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.

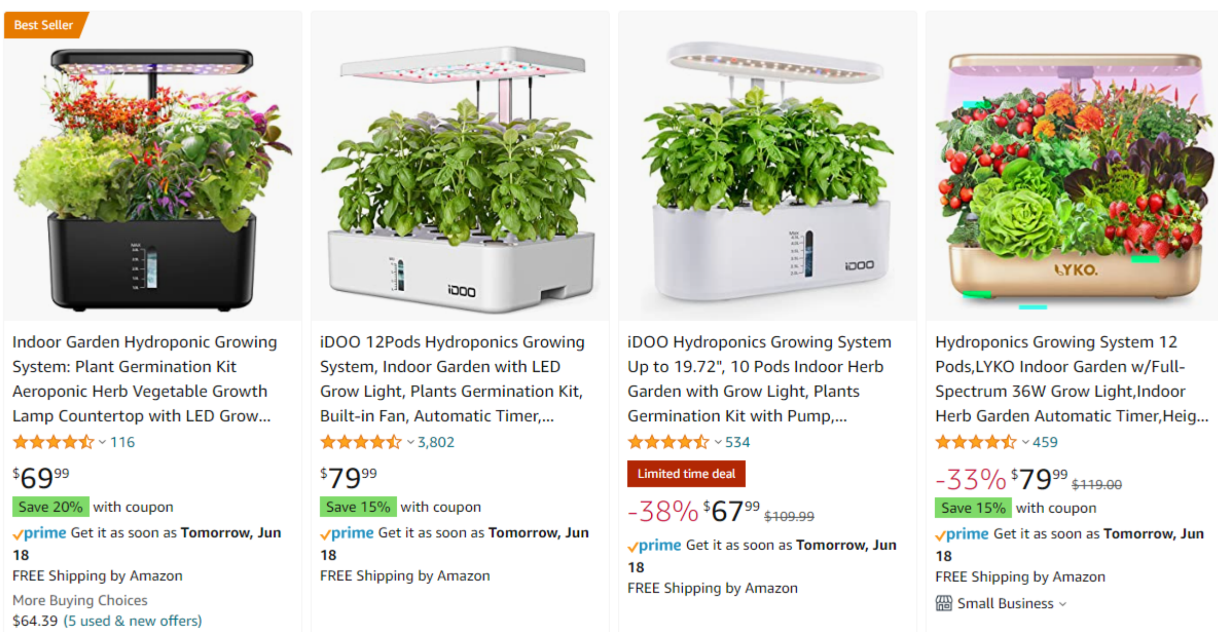


Image 2.2a

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 Image 2.2b), 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.

Image 2.2b

We also found a product called Hybriponic Home Growing System. 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.

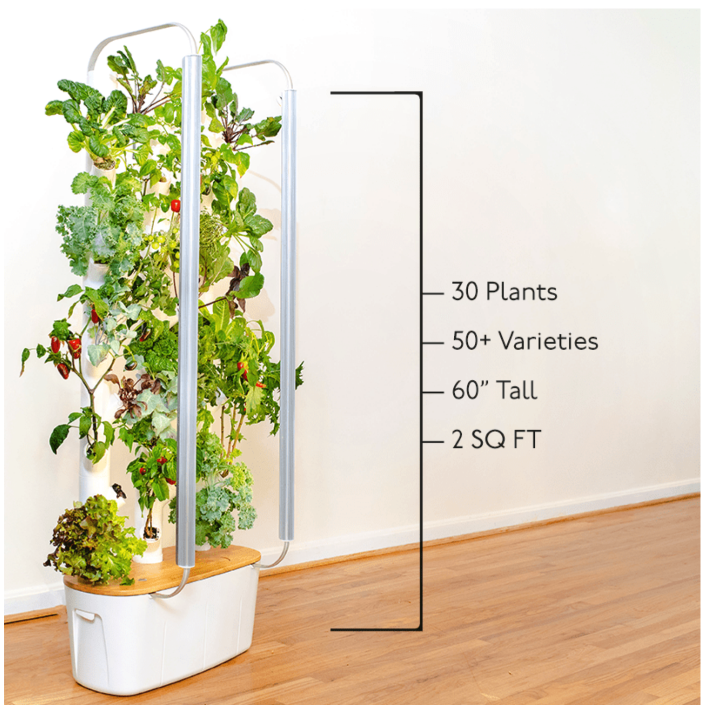


Image 2.2c

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 2.2.

	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 2.2

2.2 Features

Sensors, Pumps and Fan

The system will depend on a combination of sensors that dictate the behavior of different outputs:

Input	Output
pH Sensor	Low Flow Peristaltic Acid Pump
	Low Flow Peristaltic Base Pump
Electric Conductivity (EC) Sensor	Low Flow Peristaltic Nutrient A Pump
Total Dissolved Solids (TDS) Sensor	Low Flow Peristaltic Nutrient B Pump
Water Level Sensor	Water Reservoir Pump
Air Temperature/Humidity Sensor	Tent Fan
Web Cam	Plant Discoloration Monitoring Live Feed / Timelapse

Table 2.2

MCU

A central control unit will run our software and control the different inputs and outputs. Also, it will interface with our user's web app and LCD screen.

Light System with Timer

A set of HPS lights will be automatically activated for 14 to 16 hours per day. Besides a lighting source, HPS is also great for warming up the enclosure, which will be useful to regulate the temperature.

Wi-Fi Capable

All our data will be sent wirelessly to our web full-stack app for both the user and the MCU to operate the plant's maintenance.

Web App

A full-stack web app will handle the data and be able to change the trigger in our sensors, depending on the user's needs. This app would display multiple parameters coming live from the sensors and act as a user interface. Additionally, we want to run a computer vision algorithm that can recognize plant discoloration from the plants and alert the user.

Physical User Interface

An LCD screen will be attached to the outside of the tank to allow the user to see the data in real time without the need of a phone or computer. An expansion of said interface to allow the user to change sensor parameters on the spot can be discussed in the future.

Webcam

A webcam will be attached to the top of the tent to provide an overhead angle of the plants. The main use for our webcam will be the implementation of a Python computer vision algorithm that can detect plant discoloration and alert the user. Also ,a picture will be taken every 2 hours to generate a timelapse that allows us to track plant growth.

2.3 Fail-Safe Mechanisms

MCU Independence from Web Server

Our program will run inside the MCU independently from our Web app, this way if there is a failure to connect or a server shutdown then our plant's health is not compromised. We want the Web App to mostly act as a receiver, except for when the user sends requests to modify the plant's parameters.

Backup and Surge Protector System

We plan to have our entire system hooked up to a battery backup that will kick in when there is a power outage. Additionally, we would like to detect the power change in the system and notify the user of said outage.

Air Flow Monitoring

One key component is the Air Pump. If the roots do not receive any oxygen, they will likely die in a matter of days. We want to reassure the user that they will receive feedback if air is not flowing inside the tank and oxygenating the water. We plan to achieve this by using a pressure sensor that can detect the current of air flowing upwards from the Air Stone/Mat.

Waterproof Electronic Components

Working with electronics in a water environment can be dangerous. This is why we want to build an enclosure where all the electronic parts will go, that is water-sealed and placed outside the water tank.

2.4 Functions

The UI should display water/air temp, water ph, water EC, water level, TDS sensor, and humidity sensor. A page has settings where you can change the ranges for the preferred ph, EC, TDS, and air temperature. Once those are set, the pumps should automatically dispense nutrients into the water until those measurements are set to the appropriate ranges. There should be an additional lcd screen on the outside of the grow tent displaying all the plant statistics. There should be a timer on the lights to create a proper day night cycle for the plant. A fill and remove valve for the water and the ability to pause the system for the refill.

The MCU we are using should be able to connect to the database and send updates to the database regarding new sensor data.

Regarding the webcam the plan for it is to use python and the library open cv to run a color mask on the plants that searches for any yellow or brown discoloration over a certain percentage which then sends a warning to the webpage monitoring the plant.

2.5 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.

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.

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. Requirements and Specifications

For our requirements we considered both the product requirements (Table 3.0a), and the plant's requirements based on a plant's needs to successfully grow in a typical U.S. household (Table 3.0b).

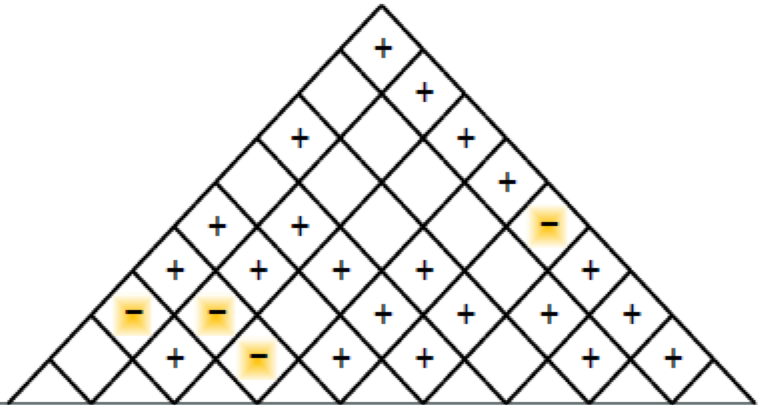
Requirements	Specifications
Compact size	< 36" x 24" x 20"
Power self-sufficiency	Direct connection to 120V standard outlet
Water Capacity	27 gallons
Sensor monitoring	6 sensors monitor the plants' environment
Affordable cost	~ $400
User Interface	LCD Screen

Table 3.0a

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 3.0b

3.1 House of Quality



Column #	1	2	3	4	5	6	7	8	9
Direction of Improvement	▲	▼	▼	▲	▼	▼	▲	▲	▼
Customer Requirements (Explicit and Implicit) / Engineering Requirements	Sensor Monitoring	Water Capacity	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

4. Block Diagrams

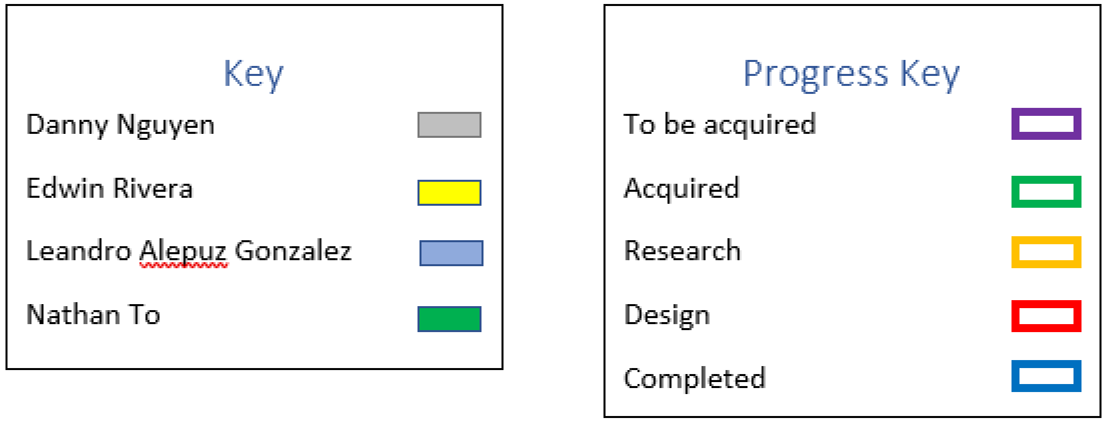
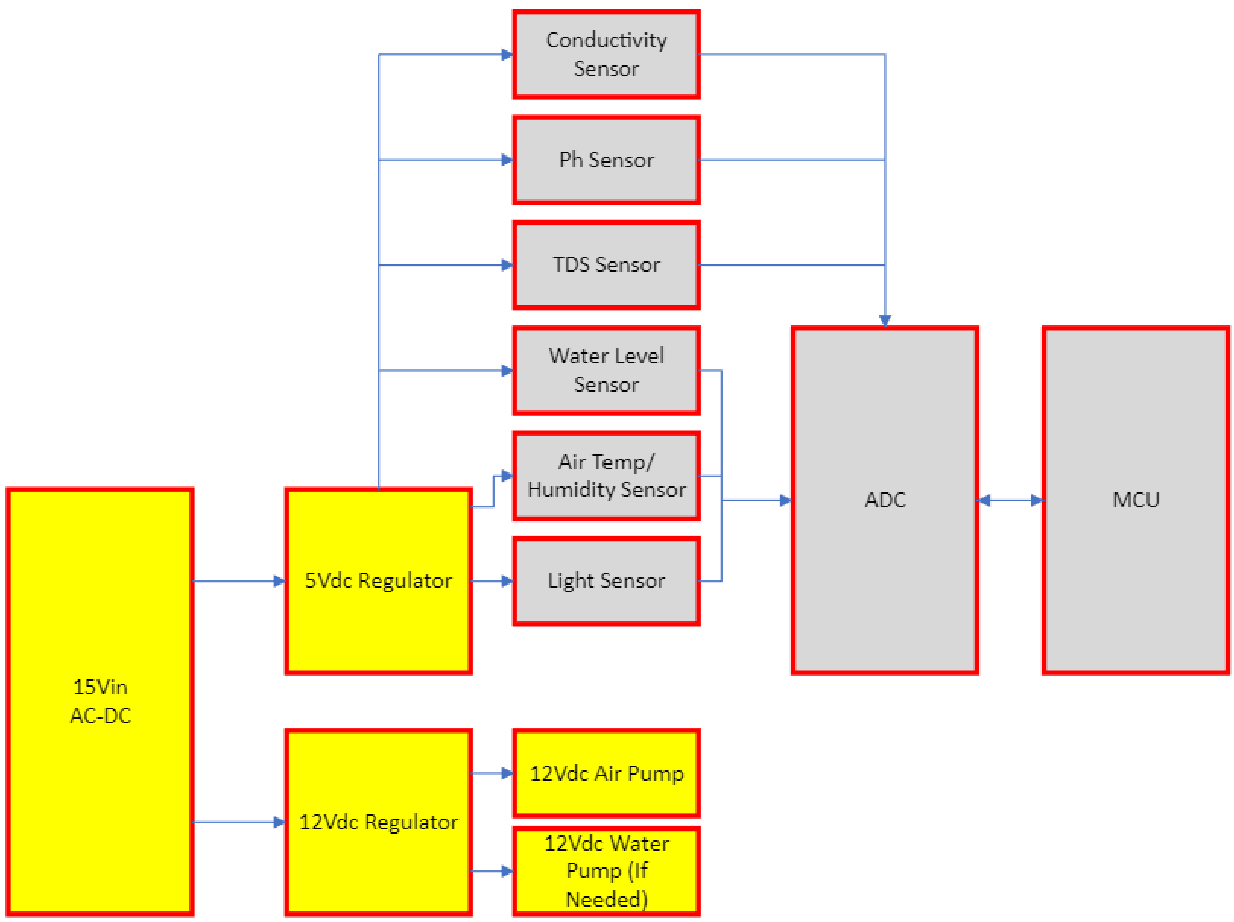
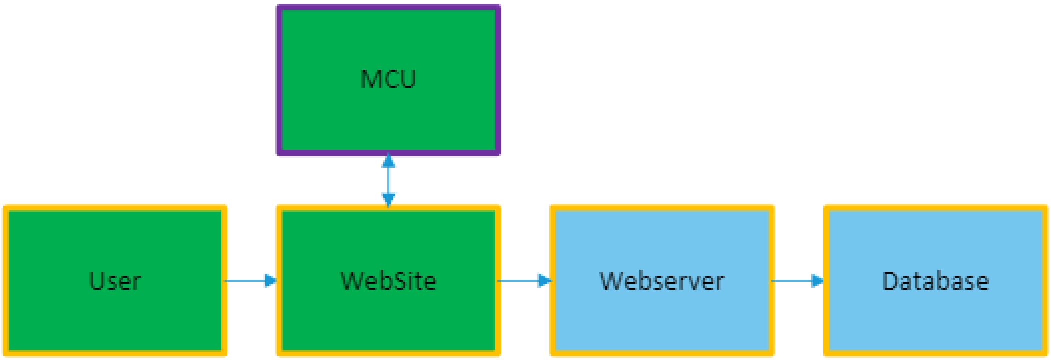


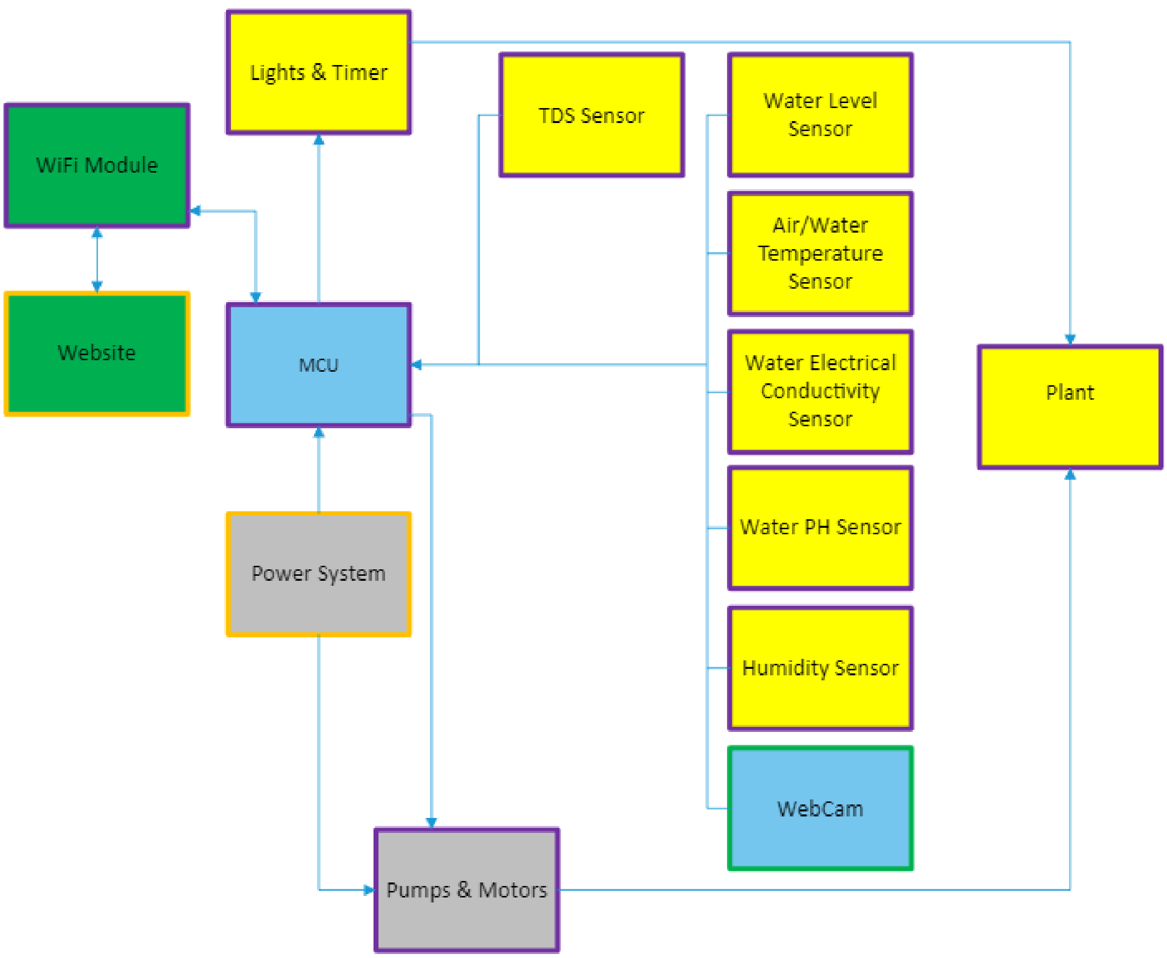
Diagram Keys



Power System Diagram



Software Diagram



Hardware Diagram

5. Budget and Financing

Item	Quantity	Estimated Cost
Grow Tent	1	$60
Grow Lights	2	Owned
Large Bin	1	$16
Plant Growth Nutrients	1	$39
pH Solutions	1	$21
Mycostop Biocontrol	2 Grams	$32
Seeds	N/A	Owned
Grow Cubes	1	$11
Seedling Heat Mat	1	Owned
BeagleBone Black	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	TBD	TBD
Air Temperature/Humidity Sensor	1	Owned
pH Sensor	1	$20
Electrical Conductivity Sensor	1	$21
TDS Sensor	1	$16
Water Level Sensor	1	Owned
Solution Peristaltic Pump	4	$10 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		**$350 - $400**
Cost Per Member		**$88 - $100**

6. Project 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

6.1 Senior Design II Milestone

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