Automated Greenhouse

Samuel Erickson
Mengtian Ke
Chandler Kramer

FINAL REPORT

REVISION – 2
5 December 2021

Table of Contents

Automated Greenhouse

Samuel Erickson
Mengtian Ke
Chandler Kramer

CONCEPT OF OPERATIONS

REVISION – 2
5 December 2021

Concept of Operations
for
Automated Greenhouse

Team <29>

Approved by:

____
Project Leader Date

____
Dr.Nowka Date

____
Eric Lloyd Robles Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	2/8/2021	Samuel Erickson		Draft Release
2	12/5/2021	Team 29		Final Release

Table of Contents

List of Tables

List of Figures

1. Executive Summary

This project aims to provide an automatic greenhouse solution to let plants grow with as little supervision as possible. This system will be powered by a standard wall outlet, and will monitor the temperature, humidity, soil water content. A microcontroller connected to these sensors will adjust each parameter as needed through the use of fans, misting, and irrigation lines. This microcontroller will interface with a wireless controller to facilitate remote operation as well as status updates and operation logs.

Figure 1: Greenhouse for fruits and vegetables

2. Introduction

Greenhouses are one of the most important innovations in agriculture, as they allow plants and crops to thrive in areas that would normally be hostile to them. They let people settle in extreme areas with confidence in their food supplies, help us prepare for a changing environment with the advent of global climate change, and ensure that plants we rely on for food and medicine will always be able to grow. However, many greenhouses still require manual adjustments in order to maintain a favorable environment, which is imprecise and takes people's time. With an estimated 9 million acres("Solar greenhouses generate electricity and grow crops at the same time, UC Santa Cruz study reveals") of greenhouses around the world, one human error can mean the deaths of many acres of crops. Our goal is to help alleviate both of these problems by creating a system that will automatically regulate the greenhouse environment and will let people monitor plants and make changes to the environment remotely should they see fit.

2.1. Background

Compared to a regular greenhouse, an automated greenhouse will include systems which can adjust the soil water content, humidity, and temperature to maintain a constant environment to grow plants. However, modern automated greenhouses are extremely expensive, so those without access to the money are unable to grow the plants they need in extreme areas. This forces them to use regular manual greenhouses, which will take time out of their day as well as increase the risk of crop failure due to human error. Our automated greenhouse aims to fill this hole in the market by being as easily operable and convenient as possible, while still having all of the key features that growers care about.

2.2. Overview

This project consists of an array of sensors inside an enclosure that will monitor environmental parameters such as temperature, humidity, and the water content of the soil. These sensors will be connected to a microcontroller that will use the data to adjust environmental variables to their desired values using a network of fans and irrigation solenoids. This microcontroller will be connected to a wireless transmitter/receiver, which will send the state of the device and receive commands from a remote user. The user will have access to an application that is able to monitor and communicate with the device in a user-friendly fashion. The device will be powered by a standard American power outlet. Further details are contained later in this document.

2.3. Referenced Documents and Standards

Investments in Armenia's Greenhouses bear fruit. (2019, September). Retrieved February 08, 2021, from https://www.ifc.org/wps/wcm/connect/news_ext_content/ifc_external_corporate_site/news+and+events/news/impact-stories/armenia-investments-greenhouses-bear-fruit

"Solar greenhouses generate electricity and grow crops at the same time, UC Santa Cruz study reveals" (2017, November). Retrieved February 08, 2021, from https://news.ucsc.edu/2017/11/loik-greenhouse.html

3. Operating Concept

3.1. Scope

For the scope of this project, concept systems will be designed to acknowledge all of the relevant operating parameters that the automated greenhouse would need to be implemented. The deliverables for the scope of the automated greenhouse this are as follows:

- Climate Sensors
- Current State Viewing
- Remote Access

Design, construction, and programming will be documented for each part of the project.

3.2. Operational Description and Constraints

The automated greenhouse is intended for use in areas with access to standard power and water distribution and climates with adequate sunlight to assist specific crop production. Constraints associated with the greenhouse include:

- Constant updates of current environment state
- Emergency temperature state to cool down environment quickly

The greenhouse is intended to be used to assist specific crops that fall in range of the greenhouse's temperature.

A smaller sized model of the greenhouse will be created and scaled to meet the dimensions of an actual greenhouse to provide an example on how it will be implemented on a much larger scale.

3.3. System Description

Power Distribution

This system will manage power delivery and control how much power each component receives. This project will be powered from the wall outlets coupled with AC/DC adapters and two buck converter circuits.

Sensor Array

This system will consist of a temperature, humidity, and soil water content sensor, and is used by the microcontroller to determine what the climate actors should do.

Microcontroller

This is the brain of the whole project, it will control the sensor array, the climate actors, and will send and interpret data from the wireless controller.

Climate Actors

This system will consist of fans, misters, and a dripper irrigation system.

Wireless Communication

This system will consist of a simple WiFi enabled microcontroller that will connect the microcontroller to a wireless network to send and receive data.

User Interface/Database

This system will work as a web app and store data sent from the wireless controller and serve as the user’s control system for the whole project.

3.4. Modes of Operations

The greenhouse’s primary operation will use sensors that gather information about the temperature, soil water content, and humidity. This data is sent to the microcontroller, which will compare that data with the user specified parameters and adjust the climate as needed and generate a log for the user.

Error notification routines will be in place to alert the user of problems in the systems operation, such as an extreme drop in temperature.

3.5. Users

Agricultural ventures and independent individuals can use a scaled-up version of our project to grow crops that would normally be unavailable in their environment. Users will need to be computer literate to operate the greenhouse remotely, and should have some experience with plumbing, construction, and electronics should something go wrong with the structure or a system component.

3.6. Support

Support will be offered in the form of a comprehensive user manual as well as construction schematics and sensor specifications so the user can replace them if necessary.

4. Scenario(s)

4.1. Assisting with agricultural diffusion

The automated greenhouse will be used to assist areas that do not have suitable climate conditions to grow crops that would not normally be produced in a specific region. This can be used to provide an abundance of crops to places that would not normally acquire such plants.

4.2. Non-industrial usage

Hobbyists and gardening enthusiasts can use an automated greenhouse as a cheap, reliable means of growing food. It can reduce an individual's carbon footprint while enabling them to produce food reliably.

5. Analysis

5.1. Summary of Proposed Improvements

Summary of improvements compared to a normal greenhouse

- Requires little to no manual labor from the user
- Automatic irrigation
- Precise control over environmental factors so plants can grow under ideal conditions
- Automatic record of environment

5.2. Disadvantages and Limitations

- To properly regulate humidity and soil water content, the greenhouse requires access to a source of water. This in and of itself can be a challenge in some environments, however, if the greenhouse is too big, it could potentially need more than one external hose for irrigation. If this is the case, more solenoids would be needed.
- To facilitate remote operation, we are using a wireless connection to the user's computer, however, this type of connection may be unreliable, especially in inclement weather. If the greenhouse is too far from the user or a signal repeater, it may not be able to properly send and receive commands.

5.3. Alternatives

- An un-automated greenhouse would be more reliable in that it would not require power or wireless capabilities, however, it would not be able to automatically regulate the internal environment. Furthermore, it would require much more manual input from the user to operate compared to an automated one.

5.4. Impact

- This system provides users with a record of their previous plant's performance so that they can adjust parameters accordingly.
- Producing food more efficiently will lead to less greenhouse gas emissions and will lower the world's overall carbon footprint.

5.5. Ethical Implications

- The automated greenhouse will use more water than is normally used in growing crops in order to maintain humidity levels. This can lead to diverted resources that could be used for other purposes.
- Due to the closed off nature of the greenhouse, natural pollinators will not have much access to the plants. This can lead to stagnant populations of plants that cannot naturally reproduce, as well as declining bee and other insect populations, which are a fragile but very important part of our ecosystem.
- Despite the increase in water usage, the greater crop yield would also help with food production, and due to the rising world population this is becoming increasingly important.

Automated Greenhouse

Samuel Erickson
Mengtian Ke
Chandler Kramer

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 2
5 December 2021

Functional System Requirements
for
Automated Greenhouse

Prepared by:

Samuel Erickson, Mengtian Ke, Chandler Kramer
3/1/2021

Approved by:

Project Leader	Date
Dr. Nowka	Date
Eric Lloyd Roble	Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	3/1/2021	Samuel Erickson		Draft Release
2	12/5/2021	Team 29		Final Release

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

The Automated Greenhouse provides a method for automatically caring for plants through the manipulation of environmental factors such as, heat, humidity, and irrigation. It is designed to be as minimal and simple as possible to allow for easy maintenance with very little training.

The system will be powered by a standard wall outlet, with a delivery system that diverts the necessary amount of power to each subsystem. There are 3 main categories of subsystems: data collection, control, and environmental manipulation. The data collection subsystems collect data on environmental factors such as soil water content, humidity, air temperature, and the amount of sunlight coming into the system. These values are then used by the controller subsystem to tell the environmental actors, (fans, misters, irrigation pumps, etc.) what to do. The controller bases its judgement off of a 4th subsystem set, which is a remote access point for the user to enter desired environmental values. Furthermore, the system may store previous and current environmental data as a report to the user, who can then use that data to make further adjustments as necessary.

1.2. Responsibility and Change Authority

The team leader, Samuel Erickson, is charged with ensuring that all specifications for the automated greenhouse are met. Any changes that wish to be made to the project must be submitted for approval to either the team leader, Samuel Erickson, or the project sponsor, Kevin Nowka.

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
IEEE 802.11	2012	IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific Requirements

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
Version 1.0	First Draft	ICD

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

3.1. System Definition

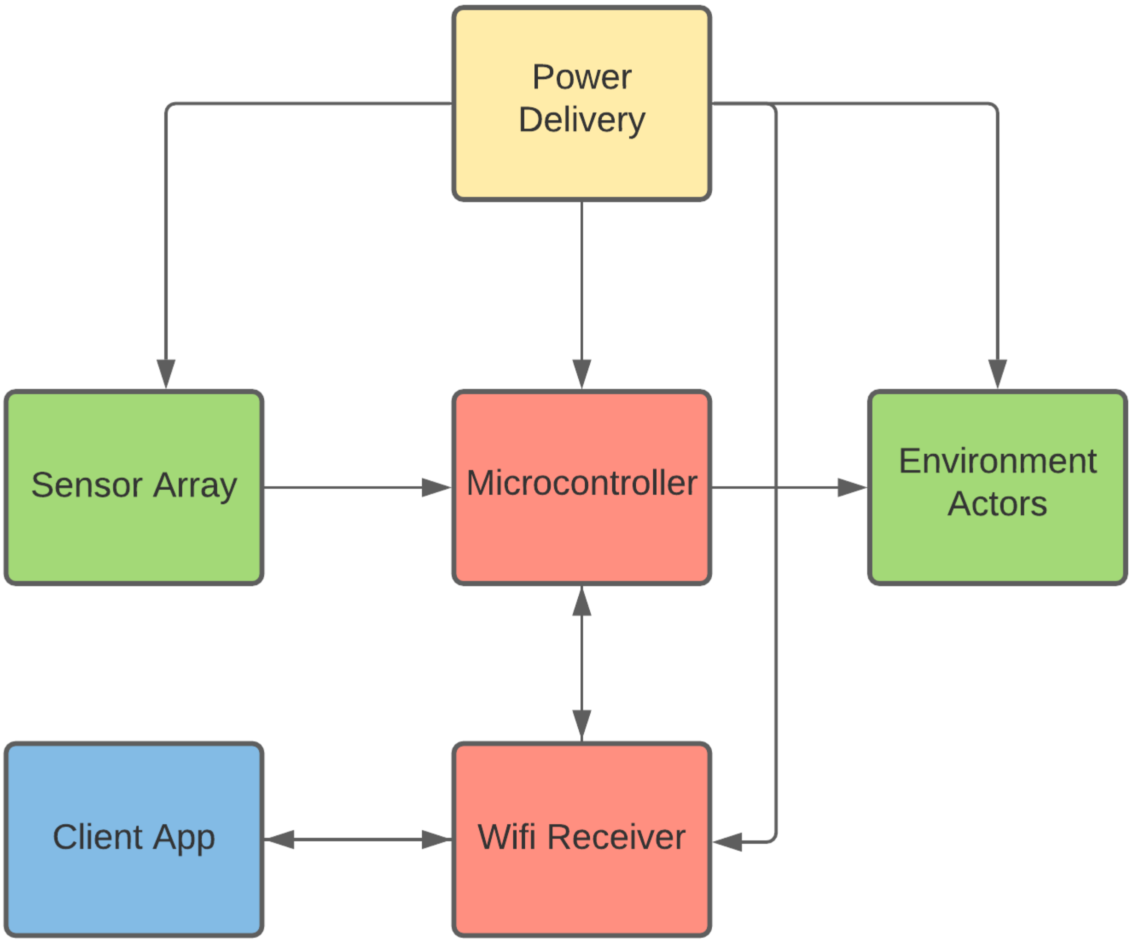


Figure 1.0: High level block diagram.
Legend: Blue - remote subsystem, Yellow - power subsystem, Green - greenhouse subsystem, Red - control subsystem.

This simple block diagram describes a high level overview of the project. The microcontroller is the center of the prototype, it will receive data from the sensor array and will change the environment actors (misters, fans, etc.) to make the specified values match those set by the client app, which is transmitted by the wifi receiver. The Microcontroller is powered by a 12V to 5V buck converter circuit which is connected to an AC/DC adapter that is connected to the wall outlet. The Wifi Receiver is powered by a 12V to 3.3V buck converter circuit which is connected to an AC/DC adapter that is connected to the wall outlet.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Frequency of Measurements

The microcontroller will continuously take measurements as often as it is able without negatively affecting performance of the microcontroller so much that it is unable to perform other functions.

Rationale: Conditions may change suddenly due to weather conditions, so it would be prudent to prepare for such inevitabilities so the system can adjust in time.

3.2.1.2. Accuracy of Measurements

The automatic greenhouse sensors will detect the temperature, humidity, light intake, and soil water content. The specification will be the following:

Sensor	**Accuracy**
Air Temperature	+/- 3 degrees celsius
Humidity	+/- 5% RH
Soil moisture	+/- 10% delta

Rationale: While general accuracy is important with what we are trying to measure, it is not so important that we need lab quality equipment and levels of precision. A general idea of the environment should suffice for most variables.

3.2.2. Physical Characteristics

3.2.2.1. Mass

The mass of the automated greenhouse shall be around the range of 40-45 kg (90-100 lbs).

Rationale: The automated greenhouse is going to be scaled down. The scaled model should be lightweight but also large enough so that it is accessible to most people to enter and exit.

3.2.2.2. **Volume**

The volume envelope of the automated greenhouse shall be scaled at the normal and accepted minimum LxWxH volume (6ft x 8ft x 6ft).

Rationale: This volume specification will allow for adequate space for plant growth while maintaining a smaller scale operation.

Electrical Characteristics

3.2.2.2.1 Input Voltage Level

The input voltage level for the automated greenhouse shall be from +12V to +16V.

Rationale: The input will be heavily reliant on the voltage supplied from a standard wall outlet.

3.2.2.2.2 External Commands

The automated greenhouse shall follow a set of external commands from the user such as:

- Documenting temperature
- Measuring humidity levels
- Checking soil moisture

Rationale: The user will be able to read data from the sensors and initiate commands from the user interface.

3.2.2.3. Outputs

3.2.2.3.1 Data Output

The automated greenhouse shall include an interface compatible with the data system. This system should include storing previous and current environmental data

3.2.2.3.2 Diagnostic Output

The automated greenhouse shall include a diagnostic interface for control and data logging.

3.2.2.4. Connectors

The automated greenhouse shall use external connectors in accordance with MIL-DTL-38999.

3.2.2.5. Wiring

The Automated Greenhouse shall follow the guidelines outlined in MIL-HDBK-5400 paragraphs 3.8 and 4.2.20

> *Rationale: We should conform to military standards that are overkill to ensure functionality.*

3.2.3. Environmental Requirements

The automated greenhosue shall be designed to withstand and operate in the environments and laboratory tests specified in the following section.

Rationale: This is a requirement specified by our customer due to constraints of their system in which the automated greenhouse is integrating.

3.2.3.1. Thermal

The automated greenhouse will be able to withstand temperatures ranging from 0 to 30° C.

3.2.3.2. Rain

The automated greenhouse will be able to operate in range that does not exceed 3” of rain and is not accompanied by flooding.

3.2.3.3. Humidity

The automated greenhouse will be able to operate with an external humidity range of 0-100%.

3.2.4. Failure Propagation

3.2.4.1. Failure Detection

The automated greenhouse will utilize failure checking algorithms to identify extreme high or extreme low levels of humidity, temperature, and water.

4. Support Requirements

4.1 Network access

The Automatic Greenhouse requires users to have the ability to access the web app. This will be the only way for users to communicate with the system though a long distance. Users will also have access to the system and make modifications based on their preference.

Automated Greenhouse

Samuel Erickson
Mengtian Ke
Chandler Kramer

INTERFACE CONTROL DOCUMENT

REVISION – 2
5 December 2021

Interface Control Document
for
Automated Greenhouse

Prepared by:

Samuel Erickson, Mengtian ke, Chandler Kramer
3/1/2021

Approved by:

____
Project Leader Date

____
Dr. Nowka Date

____
Eric Lloyd Robles Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	3/1/2021	Samuel Erickson		Draft Release
2	12/5/2021	Team 29		Final Release

Table of Contents

List of Tables

No table of figures entries found.

List of Figures

No table of figures entries found.

1. Overview

The Interface Control Document (ICD) for the Automated Greenhouse will specify the subsystems outlined in the Concept of Operations, and will be further elaborated upon in the Functional System Requirements document. This ICD will include descriptions of subsystem properties and their interconnections.

2. References and Definitions

2.1. References

IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific Requirements

3. Physical Interface

3.1. Dimensions

- Greenhouse Enclosure
  - The primary enclosure which will house the soil and plants will be no larger than 6ft x 8ft x 6ft in volume.
- Irrigation
  - The water that will be for irrigation and humidity control will come from an external source outside the enclosure such as a water hose.
- Microcontroller
  - The microcontroller and any associated devices such as wifi cards, etc. should not take up more than 1 litre of volume.

3.2. Weight

- Greenhouse Enclosure
  - The weight of the entire device without soil or water should be less than 100 pounds. This will substantially increase when soil and water are included, and the greenhouse itself should be able to hold 20cm of wet soil with regards to depth.
- Irrigation
  - The weight of the water should not drastically affect the weight of the greenhouse enclosure.
- Microcontroller
  - The weight of the microcontroller and associated devices should be no more than 10 pounds combined (MCU, Photon board, power supplies).

4. Thermal Interface

4.1 Environment regulation

The greenhouse temperature will be regulated by a combination of 2 actors: misters to increase humidity and a fan. Each of these actors will be controlled by the central microcontroller.

- Misters
  - A series of 4 misters will be arranged above the plants in such a way that they will not interfere with their growth. They will be pointed down.
- Fans/air circulation
  - A fan on the back wall of the enclosure, will regulate airflow by pushing old air out and new air in.
- Sensors
  - A temperature and humidity sensor will monitor temperature inside the enclosure and will provide the information the microcontroller needs to determine which systems to use to adjust the temperature in the greenhouse.

4.2 Water Distribution

The water distribution system, in addition to providing water to the soil, will also be used to supply a mister which will adjust the humidity level of the enclosure. This irrigation source will be from a garden hose outside of the enclosure. The hose will be connected to two solenoids which will disperse the water to a set of misters and drippers. These drippers and misters will help regulate the humidity and soil moisture within the enclosure.

5. Electrical Interface

5.1. Primary Input Power

The primary input power will be the standard wall power socket coupled with two buck converter circuits which will power all the components in the automatic greenhouse. The main voltage for the MCU will be a buck converter circuit that drops +12V to +5V (Standard USB voltage). The main voltage for the photon board will be a buck converter circuit that drops +12V to +3.3V. Both circuits will get their input voltages from a standard AC/DC 12V adapter.

5.2. Signal Interfaces

The microcontroller will have different pins which connect to sensors in order to control the airflow and water. For example, the temperature/humidity sensor will be connected using I^2C, the wireless board will be connected using UART, and the soil moisture sensor will rely on analog signalling.

- Sensors and environment regulation
  - In order to connect the microcontroller to the sensors and motor controllers, we will use a microcontroller and breakout board with GPIO capabilities.
  - Temperature/humidity sensor will connect using standard I^2C protocol
  - Soil moisture sensor will rely on analog signalling
  - Environment actors (fan, 2 solenoids) will be controlled using a switching relay that is controlled by HI/LO signalling from the microcontroller.
- Wireless card
  - To facilitate wireless communication with a client, we will use a wifi development board with USB connectivity, which we will connect to using the breakout board of the microcontroller using UART.

5.3. User Control Interface

A web app will be the only user interface due to it being convenient and friendly to users. The user will have the ability to control the greenhouse environment as well as receive notifications on a daily basis. Furthermore, the automatic greenhouse allows the user to refresh the page and receive the live time information.

6. Communications / Device Interface Protocols

6.1. *Wireless Communications (WiFi)*

We will use a premade wifi solution that supports the IEEE 802.11b/g/n standard.

6.2. *Host Device*

Users will have to connect to the host before they can have access to control the environment inside the automatic greenhouse.

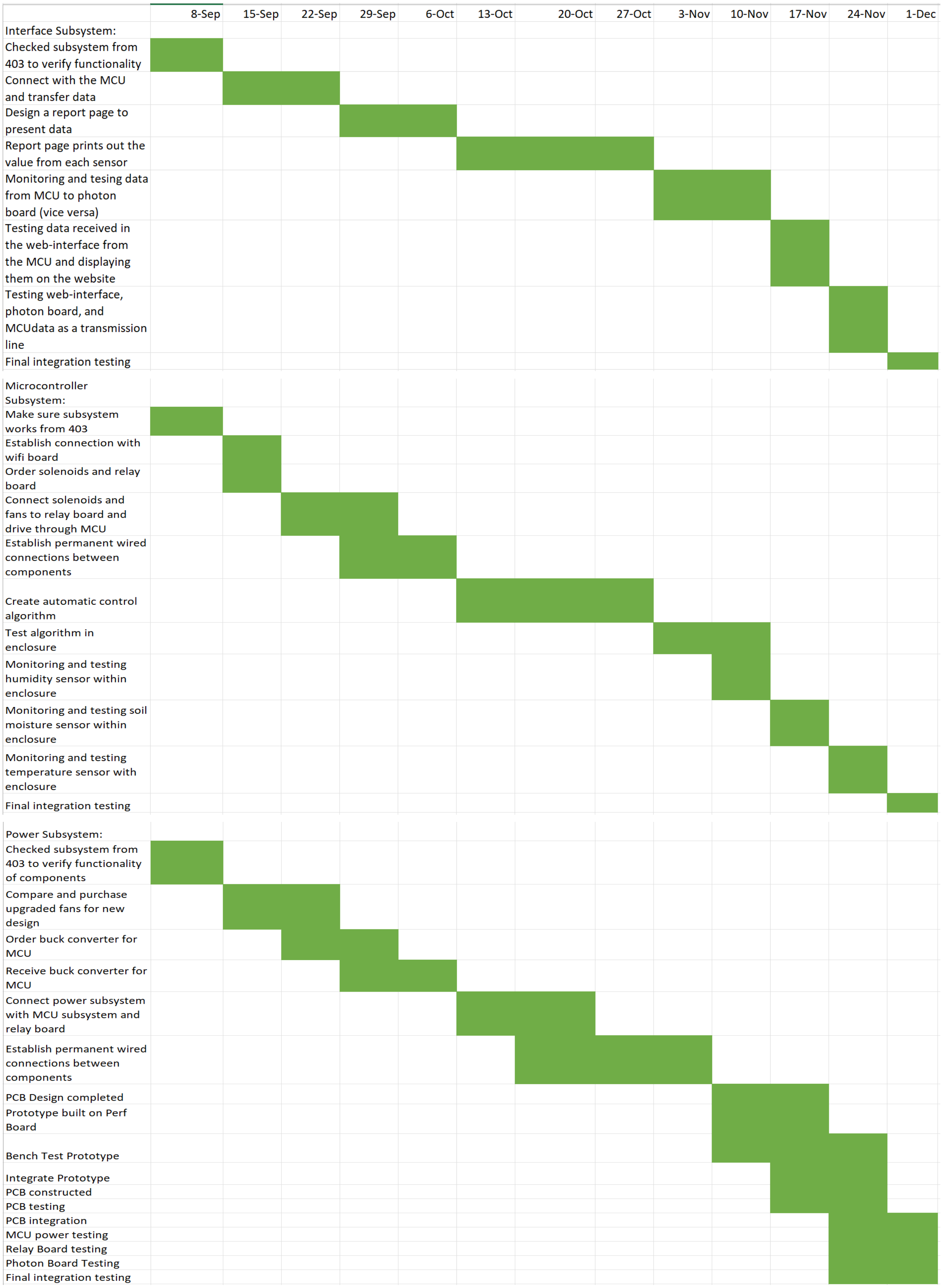
6.3. *Device Peripheral Interface*

The device peripheral interface for the automatic greenhouse will be through the b/g/n wifi standard. The device will connect to an existing wireless network, where a user will be able to control the greenhouse remotely. In a full scale version of this project, cellular service would instead be used to increase the greater effective area the product could be deployed in, however that is out of scope for this incarnation of the project.

Validation Plan

Task	Deadline	Status
Understand the problem	2/8/2021	Completed
Research and brainstorm	2/8/2021	Completed
Create Conops	2/8/2021	Completed
Identify sensor vendors		Completed
Create ICD	3/1/2021	Completed
Create FSR	3/1/2021	Completed
Order Parts		
Soil moisture/temperature sensor	3/22/2021	Completed
Microcontroller	3/22/2021	Completed
Humidity sensor	3/22/2021	Completed
Light sensor	3/22/2021	Completed
Wifi controller	3/22/2021	Completed
Remote client app	3/22/2021	Completed
Receive Parts		
Soil moisture/temperature sensor	4/3/2021	Completed
Microcontroller	4/3/2021	Completed
Humidity sensor	4/3/2021	Completed
Light sensor	4/3/2021	Completed
Wifi controller	4/3/2021	Completed
Remote client app	4/10/2021	Completed

Full Power Subsystem Designed	4/21/2021	Completed
Packet Protocol Designed	4/21/2021	Completed
Structural Equipment Designed	4/23/2021	Completed
Sensors reporting to microcontroller	4/25/2021	Completed
Online Database created	4/27/2021	Completed
GUI created	4/28/2021	Completed


8-Sep
15-Sep
22-Sep
29-Sep
6-Oct
13-Oct
20-Oct
27-Oct
3-Nov
10-Nov
17-Nov
24-Nov
1-Dec
Interface Subsystem:
Checked subsystem from 403 to verify functionality
Connect with the MCU and transfer data
Design a report page to present data
Report page prints out the value from each sensor
Monitoring and tesing data from MCU to photon board (vice versa)
Testing data received in the web-interface from the MCU and displaying them on the website
Testing web-interface, photon board, and MCUdata as a transmission line
Final integration testing
Microcontroller Subsystem:
Make sure subsystem works from 403
Establish connection with wifi board
Order solenoids and relay board
Connect solenoids and fans to relay board and drive through MCU
Establish permanent wired connections between components
Create automatic control algorithm
Test algorithm in enclosure
Monitoring and testing humidity sensor within enclosure
Monitoring and testing soil moisture sensor within enclosure
Monitoring and testing temperature sensor with enclosure
Final integration testing
Power Subsystem:
Checked subsystem from 403 to verify functionality of components
Compare and purchase upgraded fans for new design
Order buck converter for MCU
Receive buck converter for MCU
Connect power subsystem with MCU subsystem and relay board
Establish permanent wired connections between components
PCB Design completed
Prototype built on Perf Board
Bench Test Prototype
Integrate Prototype
PCB constructed
PCB testing
PCB integration
MCU power testing
Relay Board testing
Photon Board Testing
Final integration testing

Automated Greenhouse

Samuel Erickson
Mengtian Ke
Chandler Kramer

SUBSYSTEMS REPORT

REVISION – 2
5 December 2021

SUBSYSTEMS REPORT
FOR
Automated Greenhouse

PREPARED BY:

Samuel Erickson, Mengtian ke, Chandler Kramer
4/28/2021

APPROVED BY:

____
Project Leader Date

____
Dr. Nowka Date

____
Eric Lloyd Robles Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	4/28/2021	Samuel Erickson		Original Release
2	12/5/2021	Team 29		Final Release

Table of Contents

List of Tables

List of Figures

1. Introduction

Our project consists of a total of 6 subsystems, power delivery, environment actors, the microcontroller, a sensor array, a wifi transceiver, and an interface website. The power delivery subsystem provides power to the entire project except for the interface website. However, most of the power draw will come from the environment actors. The environment actor subsystem consists of fans and pumps which provide irrigation, humidity adjustment, and temperature control, which will be controlled by the microcontroller subsystem. The microcontroller subsystem is the brain of the project, and will control the environment actors and the sensors. It takes data from the sensor array subsystem, and runs it through an algorithm that will change the environmental values as specified by the client in the interface website. The microcontroller will access the website through the wifi transceiver, which hosts the website and will tell the microcontroller what values the client set, as well as send a report of the greenhouse activity to the user through the interface website. This document outlines the details of each subsystem, how it is used in the context of our project, and how each system was validated. Because each system was validated to function properly, there is a clear path towards a final completion of this project as outlined by the ConOps, FSR, and ICD documents.

2. Power Subsystem Report

2.1. Subsystem Introduction

The power subsystem is designed to power all the components that will be placed in the automated greenhouse. The subsystem is powered by a standard wall outlet with a 12V/2A AC/DC adapter. From here, the two adapters will go to two buck converters that will power the MCU (12V to 5V) and the Photon board (12V to 3.3V). The power subsystem was cleared and checked for stability, capacity, and consistency by validating both buck converter circuits using line and load regulation testing.

2.2. Subsystem details

A diagram of the power subsystem is shown below:

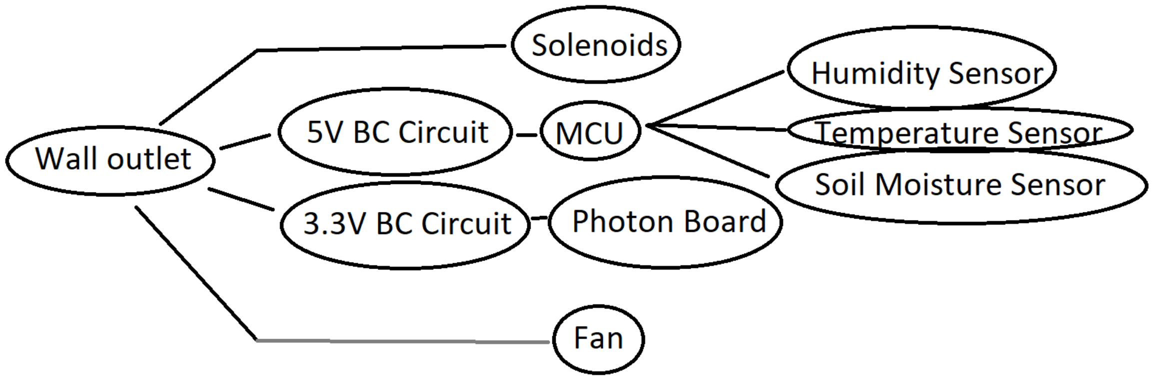


Figure 1. Functional block diagram of the Power Subsystem

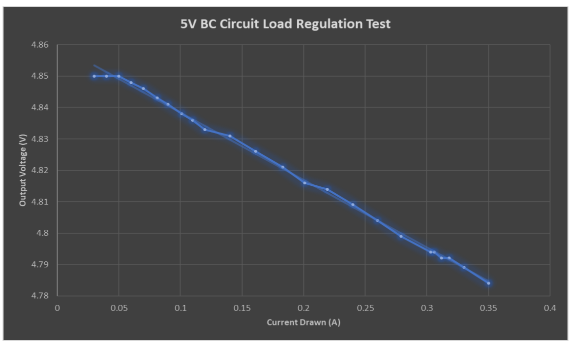
The primary challenge with the power subsystem was constructing and validating each buck converter circuit. The 5V buck converter circuit used for powering the MCU also needed to supply enough power to the relay board. The 3.3V buck converter powered the Photon board by itself since it had issues when it was under the same power source as the MCU.

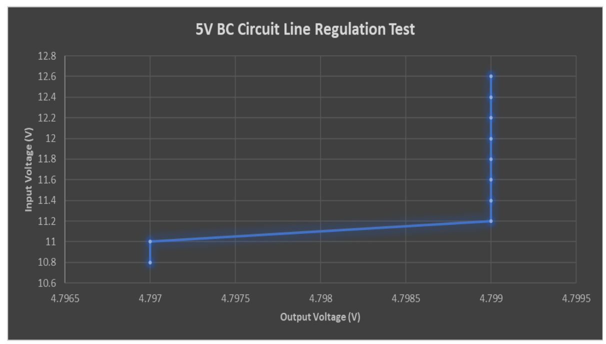
The validation was done prior to integrating the buck converter circuits into the enclosure. Line and load regulation testing were performed prior to integration to ensure functionality. Each circuit was tested to go 10% over or under the desired target.

2.3. Subsystem Validation

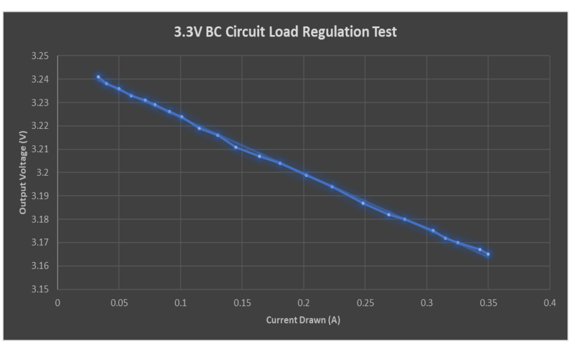
The fan was tested and validated first. All that was required from the fan was functionality since it’s power source was a wall outlet. The solenoids were the next components to be validated. This was done by connecting them to the outlet and listening for a small clicking noise. The noise would indicate that the solenoid was switching back and forth. Both the fan and solenoids were connected to the MCU and were once again validated when they received commands from the MCU to turn on/off.

The first buck converter circuit that was validated was the 5V circuit. A line and load regulation test were performed and passed with a 10% margin. The MSP432 can function in between a range of 3.3V – 5V. If the voltage exceeds the maximum range of the MSP432, the MCU could be damaged making it unreliable to use.





The second buck converter (and last component) to be validated was the 3.3V buck converter. A line and load regulation test were performed and passed with a 10% margin. The Wi-Fi Transceiver needs around 3.3V to operate. The transceiver should not fall or rise 1.5V above or below the required voltage or else the transceiver could corrupt and cause application errors.



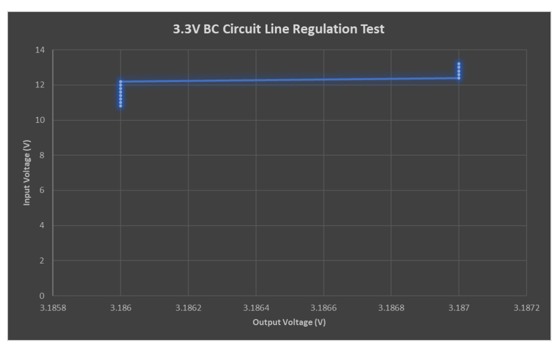


Table 1. Current draw from components

Component	Current Draw
Fan	.52A
Solenoids (each)	1.6A
Wi-Fi Transceiver	< 80 mA
MCU (MSP432)	0.5A (Maximum)
Soil Moisture Sensor	negligible
Temperature and Humidity Sensor	negligible

Both the Solenoids and the fan operate on AC voltage so they can be plugged in directly to the wall outlet. The MCU and Photon board are less than the total power of the AC/DC adapters (12V/2A/24W).

2.4. Subsystem Conclusion

Each component of the subsystem was tested and validated. The power subsystem generates and stores the needed power for all the components in the automated greenhouse. When interfaced with the other subsystems, it will enable the system to continuously collect and read data and transmit it to the database through the network.

3. Microcontroller Subsystem Report

3.1 Introduction

The microcontroller subsystem is the brain of the project. It takes values from the sensor array subsystem, uses that data to adjust the environment values using the environment actor subsystem, and finally reports its activity to the interface website using the wifi transceiver subsystem.

3.2 Details and Operation

We used the MSP432 microcontroller on an MSP-EXP432P401R development board as our microcontroller. We settled on this board specifically due to its lower power consumption, support for up to 4 concurrent I2C connections, 256 KB of flash memory, UART connectivity, and large user base and support. In total, it can support up to 16 external analog data connections and 16 digital connections through a combination of I2C, UART, and SPI. While the microcontroller can be used in a very low power mode we opted to reduce complexity in the validation phase of this project and run it at the default 3.3-5V power mode. It is connected to the sensor subsystem using one instance of I2C communication and one instance of analog communication, totalling 2 power lines, 2 grounds, 1 SDA line, 1 SCL line, and 1 analog voltage line. It is also connected to the wifi transceiver via UART and the relay board through digital HI/LO connectivity, and will be powered by the 5V buck converter.

3.3 Functional Validation

Due to the complexity of validating the functionality of an entire microcontroller, validation consisted of making sure that code pushed to the microcontroller would actually run independent of any other computer and that it could meet our needs of interfacing with the sensors. The ability to push code to the microcontroller was validated through a simple blink code that changed the color of an on-board LED at regular intervals. Furthermore, when the board was disconnected from the programming computers USB port, and instead connected to a battery bank that would supply no code, the code functioned as expected, showing that the pushed code was saved to the memory and the microcontroller could operate independent of any other code source.

To confirm that the microcontroller works with the sensor array subsystem, please see section 4.0 of this document.

3.4 Conclusion

The microcontroller works as expected for our purposes. It is able to receive code and function independently, and it will be able to support our future expansions though the copious amounts of I/O ports. Furthermore, it can also support communication with our sensors by analog and digital means. Please see the following section (4.4) of this document to see the results of the sensor validation.

4. Sensor Array Subsystem Report

4.1 Introduction

The purpose of the sensor array subsystem is to gather data from the environment around it and send it to the microcontroller for meaningful processing. This is done through 3 sensor types across 2 physical sensors. The first is a combination sensor that reads temperature and humidity values, the second is a soil moisture sensor. The microcontroller takes readings from each sensor every 0.5 seconds, so it will always have current data. For this subsystem, it is important to note that the absolute values of the data that the sensors receive are not particularly useful, as (excluding temperature) we do not have a satisfactory way to control the environment so we can calibrate the sensors. Instead, what is more important is that we are able to discern the change in the values they record, so we can construct an algorithm to adjust for those differences.

4.2.1 Temperature & Humidity Sensor Introduction and Operation

The combination temperature and humidity sensor will be affixed to the top of the final enclosure facing downwards, so that growing plants will not affect the data it collects. We chose the Si7021 sensor for its small size, I2C communication protocol, included pull-up resistors, and thorough documentation. It is connected to the microcontroller via a simple bus which facilitates transmission of 3.3V power, ground, SCL and SDA lines for I2C communication.

4.2.2 Temperature & Humidity Sensor Validation

After being connected to the microcontroller with power, ground, SCL and SDA lines through a breadboard, code to interface with the sensor through I2C communication was uploaded to the microcontroller. The microcontroller requests data every 0.5 seconds, which the sensor responds with a total of 4 bytes of information over 2 I2C instances. Each pair or bytes is then concatenated and then a formula is applied to each number to obtain the final humidity and temperature values. The humidity is measured in terms of relative humidity, and the temperature is measured in terms of Celsius.

```
temperature - 24.36 C
humidity - 52.09 %
temperature - 24.41 C
humidity - 51.96 %
temperature - 24.41 C
humidity - 51.75 %
temperature - 24.40 C
humidity - 51.54 %
temperature - 24.42 C
humidity - 51.34 %
temperature - 24.41 C
humidity - 51.12 %
temperature - 24.40 C
humidity - 50.91 %
temperature - 24.36 C
humidity - 50.72 %
```

Figure 2 Sample Temperature and Humidity Sensor Output

After it was established that the microcontroller and the sensor were in fact communicating and sending useful data, testing of the sensor began. To test the sensor, baseline values in a room temperature and humidity environment were recorded to be compared against. To validate that the sensor could discern changes in the environment, it was suspended 20 cm above boiling water so that both the temperature and humidity would change.

Test	Value Average
Baseline Temperature	22.43 C
Baseline Humidity	51.14%
Temperature 20 cm above boiling water	32.57 C

Humidity 20 cm above boiling water	98.37%-105.93%

Table 2 Data recorded from temperature humidity sensor

Each value was averaged over 10 instances of data collection which took a total of 5 seconds, so while there could be fluctuations in temperature and humidity, the overall trend is that the sensor does indeed respond quite strongly to real changes in temperature and humidity. One thing of note is that the value of 105.93% for relative humidity is normal. As specified in the data sheet the sensor does lose accuracy in humidity readings as the humidity approaches 100%, furthermore the steam from boiling water most likely supersaturated the air, so that the relative humidity could actually be greater than 100%. In conclusion, the temperature and humidity sensor returns useful data to the microcontroller and functions as we need it to.

4.3.1 Soil Moisture Sensor Introduction and Operation

The soil moisture sensor will be partially buried in the soil roughly 2.5 inches so that the sensing element is sufficiently deep and so that the exposed electronics are not touching anything which would corrode them. In the controlled environment of validation the electronics were exposed however in the final version of this project any exposed electronics will be protected as a greenhouse is a relatively extreme environment with many possibilities for corrosion and electrical shorts. The sensor itself is a Gikfun Capacitive Soil Moisture Sensor. It was chosen for its capacitive nature, it has no exposed element that would touch the soil in normal conditions so that it will not corrode, and its analog output for easy interfacing and reliability. In this phase of the project only one sensor was used for validation however in the final product there will be multiple soil moisture sensors due to the sensor only picking up differences in soil moisture in a very small area.

The sensor is connected to the microcontroller through 5V power, ground, and an analog data wire. Because the sensor only returns an analog voltage as data, it is up to the microcontroller to interpret the data and transform it into something useful. In this case, the microcontroller we use interprets analog values of 0 - 5V as integer values of 0-1023, so no value will ever exceed that range. For this sensor, lower values correlate to high moisture values and higher values correlate to lower moisture values.

```
700
701
701
695
701
700
700
701
700
700
701
701
701
702
701
```

Figure 3 Sample Soil Moisture Sensor output

4.3.2 Soil Moisture Sensor Validation

To validate the soil moisture sensor, an open air baseline was recorded to compare all other values against. It was then exposed to extreme scenarios, such as being completely submerged in water, and regular scenarios, such as being exposed to freshly watered dirt.

Test	Average Value
Dry Control (inside)	745
Submerged in water (tip only)	563
Submerged in water (to water line)	366
Dry mulch (outside)	707
Freshly watered soil (outside)	345
Submerged in brine solution	296

Submerged in vinegar	303

Table 3 Soil Moisture Sensor validation results.

As the data in the table shows, there are two major factors that change the value that the sensor returns: the depth of the sensor in the material and the water content of the material. However, it is of note that the moisture value of a material which has a high electrical conductivity i.e. brine and vinegar, is substantially lower than that of low conductivity (tap) water. This leads us to conclude that the reason the moist soil is returning a lower moisture value than pure water is because of conductivity. We will have to adjust for this in the microcontroller through testing various soils, because different soils may have different electrical responses when wet. For our purposes however, the sensor returns different and discernable enough values for it to be useful to us in this project.

4.4 Sensor Array Validation Conclusion

Each sensor functions correctly and is validated both in terms of functionality and ability to return useful information, and both styles of sensor will be used in the final product.

5. Wi-Fi Transceiver Subsystem Report

5.1 Introduction

The Wi-Fi transceiver, Particle Photon WiFi Development board, was used in this project to transfer user inputs from the Client interface to the MCU. The Photon board used UART(TX/RX) to communicate and receive 3.3V of power from the MCU. Once MCU receives user inputs, it will pass them to the sensor and that would lead the fan, mist and dripper to start cooling the greenhouse. On the other hand, MCU will collect current data and send them back to display on the client interface through the photon board.

5.2 Wifi Transceiver operation

The Photon board receives client inputs and stores them in Particle cloud. Those numbers will pass to the MCU as well as to the temperature sensor, humidity sensor and soil moisture sensor. Later on, it sends back the current temperature, humidity and soil moisture data values from the greenhouse every two seconds. The Particle cloud will record those values according to the time it arrives. Finally the Photon board will send those current values to the client interface and display them.

5.3 Wifi Transceiver validation

The Wifi transceivers were tested and validated through the local terminal for all three different phases.

Phase1: A loop counter was used in order to continue to have inputs. This number will be sent to MCU and MCU will subtract by one and send it back. (Complete)

Phase2: MCU generates 3 values named as temperature, humidity and soil_moisture while using a random number generator. The range for each value was also shown in Table 4. These values will pass back to the photon board and these random values will either show in the local terminal or particle cloud. (Complete)

Phase3: Use Photon board to generate a random value for temperature, humidity and soil_moisture and send them to MCU. MCU sends back the same value but subtract by one as a representation for current value. (Complete) code.ino

```
#include "Particle.h"

// Constants
const unsigned long SEND_INTERVAL_MS = 2000;
const size_t READ_BUF_SIZE = 256;

// Structures
typedef struct {
  int temperature;
  int humidity;
  int soilmoisture;
} WeatherData;


//WeatherData USERINPUT;
int tempVal(String Temp);
int humindityVal(String humindity);
int soilMoistureOne(String soilMoistureone);

double number = 50;
double numberone = 0;
double numbertwo = 1000;


// Forward declarations
void processBuffer();
void handleWeatherData(const WeatherData &data);

// Global variables
int counter = 0;
unsigned long lastSend = 0;
char sendBuf[256];

char readBuf[READ_BUF_SIZE];
size_t readBufOffset = 0;

// report page
char status[] = "{\"debug\":%s}" ;
char results[] = "{\"temperature\":%.02f,\"humidity\":%.02f,\"soilmoisture\":%.02f}" ;
char msg_st[sizeof(status) + 16];
char msg_res[sizeof(results) + 32 ];

unsigned long interval = 0;

void setup() {
    Serial.begin(9600);
    Serial1.begin(19200);

    Particle.function("Temperature", tempVal);
    Particle.function("Humindity", humindityVal);
    Particle.function("SoilMoistureOne", soilMoistureOne);

    Particle.variable("number", number);
    Particle.variable("numberone", numberone);
    Particle.variable("numbertwo", numbertwo);

    Particle.variable("data", msg_res);
}

```

code.ino

```
void loop() {
    if (millis() - lastSend >= SEND_INTERVAL_MS) {
        lastSend = millis();

        WeatherData send;
          send.temperature = number;
          send.humidity = numberone;
          send.soilmoisture = numbertwo;

    snprintf(sendBuf, sizeof(sendBuf), "%d,%d,%d\n",
      send.temperature, send.humidity, send.soilmoisture);
            Serial1.print(sendBuf);
            Serial.printlnf("Sent to MCU: temperature=%d humidity=%d soilmoisture=%d",
            send.temperature, send.humidity, send.soilmoisture);
    }

    // Read data from serial
    while(Serial1.available()) {
        if (readBufOffset < READ_BUF_SIZE) {
            char c = Serial1.read();
            if (c != '\n') {
                // Add character to buffer
                readBuf[readBufOffset++] = c;
            }
            else {
                // End of line character found, process line
                readBuf[readBufOffset] = 0;
                processBuffer();
                readBufOffset = 0;
            }
        }
        else {
            Serial.println("readBuf overflow, emptying buffer");
            readBufOffset = 0;
        }
    }
}

int tempVal(String Temp) {
    number = Temp.toInt();
    return number;
}
int humindityVal(String humindity) {
    numberone = humindity.toInt();
    return numberone;
}
int soilMoistureOne(String soilMoistureone) {
    numbertwo = soilMoistureone.toInt();
    return numbertwo;
}

void processBuffer() {
    WeatherData data;

  if (sscanf(readBuf, "%d,%d,%d", &data.temperature, &data.humidity, &data.soilmoisture) == 3) {

    handleWeatherData(data);

  }
```

```
  else {

    Serial.printlnf("invalid data %s", readBuf);

  }
}

void handleWeatherData(const WeatherData &data) {
  Serial.printlnf("FROM MCU temperature=%d humidity=%d soilmoisture=%d",
      data.temperature, data.humidity, data.soilmoisture);

    //Particle.publish("temperature", String(data.temperature), 60, PRIVATE);
    //Particle.publish("Humindity", String(data.humidity), 60, PRIVATE);
    //Particle.publish("SoilMois", String(data.soilmoisture), 60, PRIVATE);

    double temperature;
    double humidity;
    double soilmoisture;

    if (millis() - interval > 1000) {
        interval = millis();
        temperature = data.temperature;
        humidity = data.humidity;
        soilmoisture = data.soilmoisture;
        snprintf(msg_res, sizeof(msg_res), results,
            temperature, humidity, soilmoisture);
        Particle.publish("ALL_DATA", msg_res, 60, PRIVATE);
    }
}
```

Figure 4: Particle Photon code

6. Interface Website Subsystem Report

6.1 Introduction

The client interface was designed to help users to modify the physical environment inside the greenhouse as well as monitor it in a comfortable location. While using the client interface, users will be able to adjust the temperature, humidity and the soil moisture level as well as maintain the environment as needed. On the other hand, users will be able to view the data once they open the client interface. In other words, it's not necessary to make a change in order to view the data value.

6.2 Interface Website operation

When the user opens the client interface, it directs the user to the home page. There are eight different greenhouses listed above but for project purposes only Automated Greenhouse 1 will be available. When the user accesses the link named as Automated Greenhouse 1,it will direct the user to the main page of the greenhouse. Users will have the option to control temperature, humidity and soil moisture level. For the test purpose, out of range values are also available to pass in but there is a warning message that will be displayed and inform the user that the value is out of range. After the user adjusts the values and hits the “run” button, these data values will pass to the Photon board and the MCU. On the other hand, the client interface will also present the current environment values every few seconds.

	User valid inputs
Temperature	0C-50C
Humidity	1%-100%
Soil Moisture	300-700

Table 4: Valid user inputs for the client interface Particle Photon code

6.3 Interface Website Validation

The client interface was tested and validated to transceive data to the Photon board.
Phase 1: As users input the target value in the client interface, those targets will be sent to the particle cloud and should store as a number in the Photon board. (Complete)
Phase 2: Target value should sent one by one, in other words when sending the target value for temperature that humidity and soil moisture value should remain as default value (Complete)
Phase 3: All three environment factors should be updated at different times. For example, temperature should be updating every 15-30mins and soil moisture level should be measured every 6 hours. For the test purpose, all three values are receiving data from the sensor together every 2 seconds. (Complete)

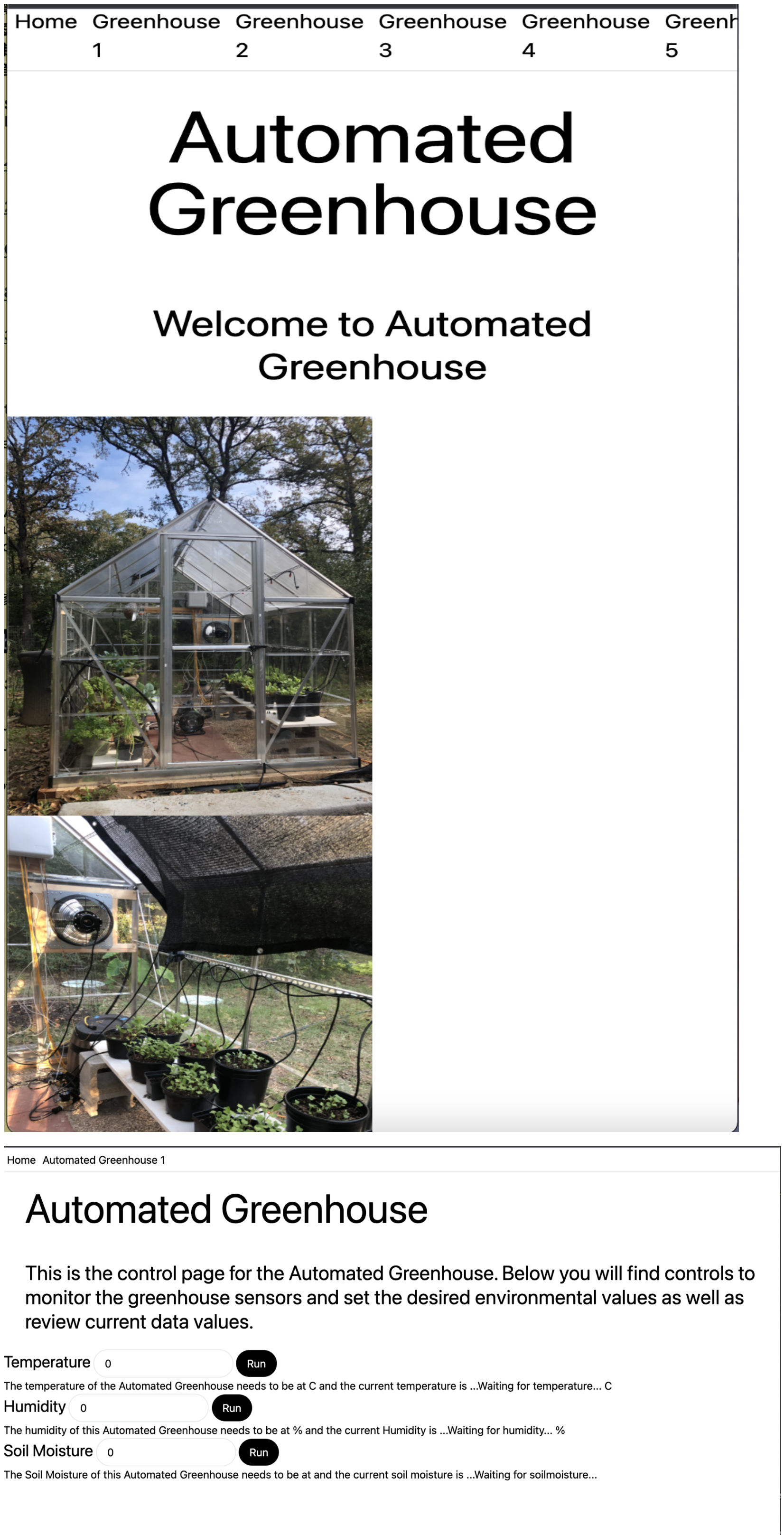


Figure 5: A simple demonstration for Client interface

7. System Validation Report

7.1 Introduction

The greenhouse was fully assembled and tested prior to validation test except for the power subsystem, as it was not complete at the time of testing. Standard USB power was used to supply the power to the MCU in lieu of the 5V buck converter. Due to the nature of the project this is still a fair test of the control algorithm as well as the capabilities of the environment actors to change the greenhouse environment.

7.2 Validation Setup

In order to properly test the greenhouse, a heater was placed in the inside of the greenhouse while the door was closed to maximise the interior temperature and to lower the humidity. The heater was on throughout the entirety of the test. After temperature could not go any higher, target values of temperature, humidity, and soil moisture content were remotely loaded into the MCU from the client website. These values were intentionally chosen to trigger the emergency mode of the algorithm, where it would run both the fan and the misters simultaneously in order to cool the greenhouse down to safe levels. Furthermore, the test also had the watering interval reduced due to tester time constraints as well as to showcase the ability of the irrigation system to maintain a constant water content level. The test ran for 7 minutes and 30 seconds.

7.3 Summarized Results

Data from Greenhouse Sensors	Temperature (°C)	Relative Humidity (%)	Soil Moisture Content
Initial State	28	63	433
Target State	20	80	390
Final State	20	82	391

Figure 6: Summarized test results, showing initial, target, and final states of the environment conditions.

7.4 Detailed Results

The graphs below show how the relevant environmental conditions changed over the course of the test. The blue error bars show the acceptable tolerance levels as outlined in the FSR.

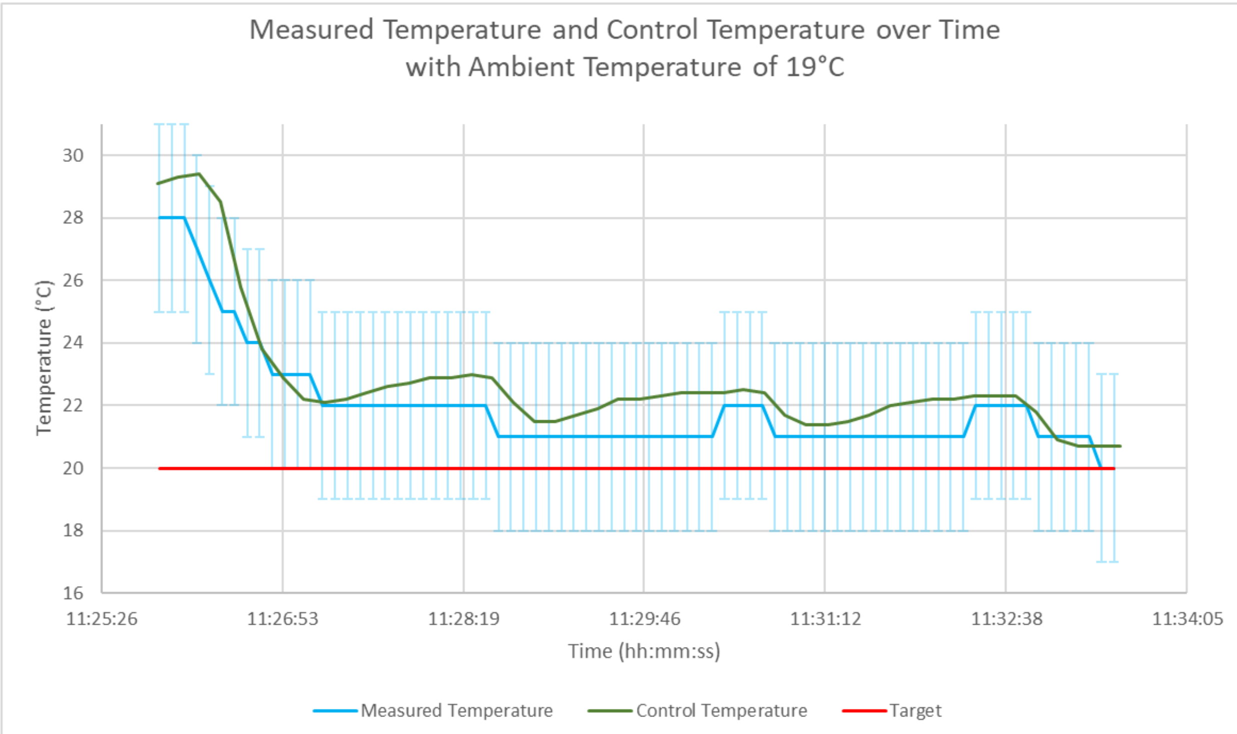


Figure 7: Result of the temperature change of the validation test.

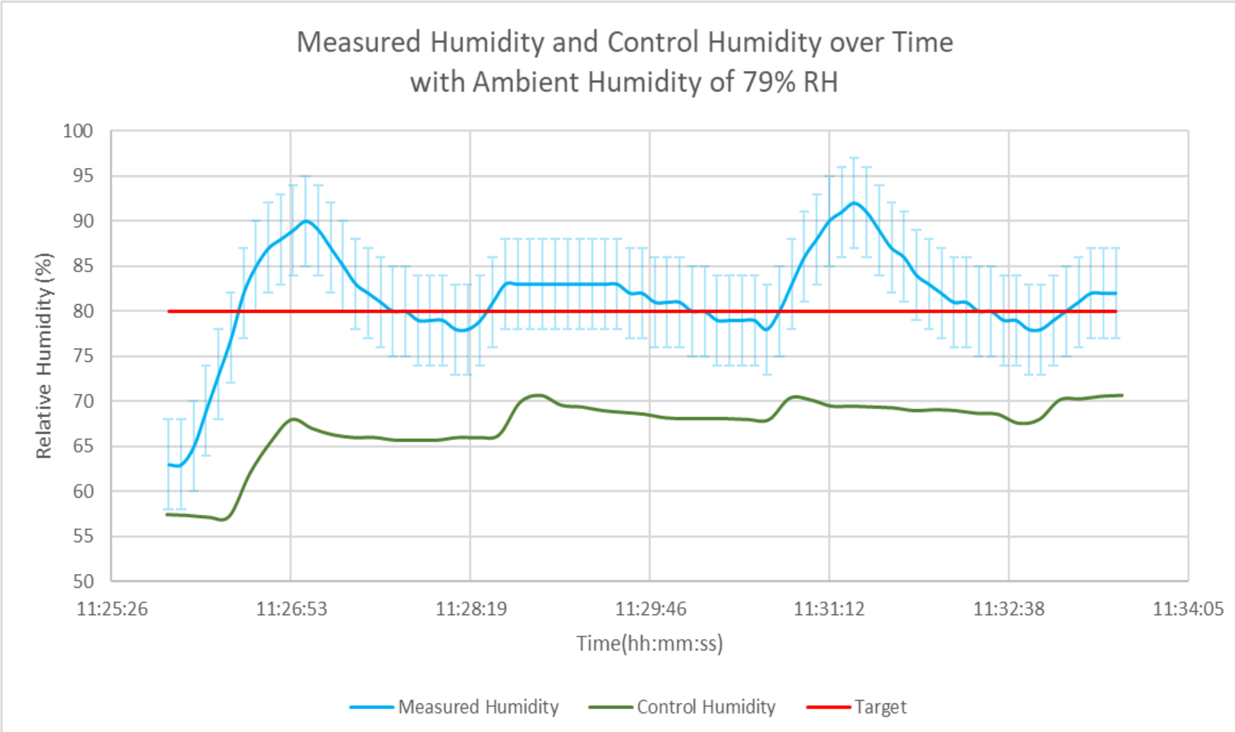


Figure 8: Result of the humidity change of the validation test.

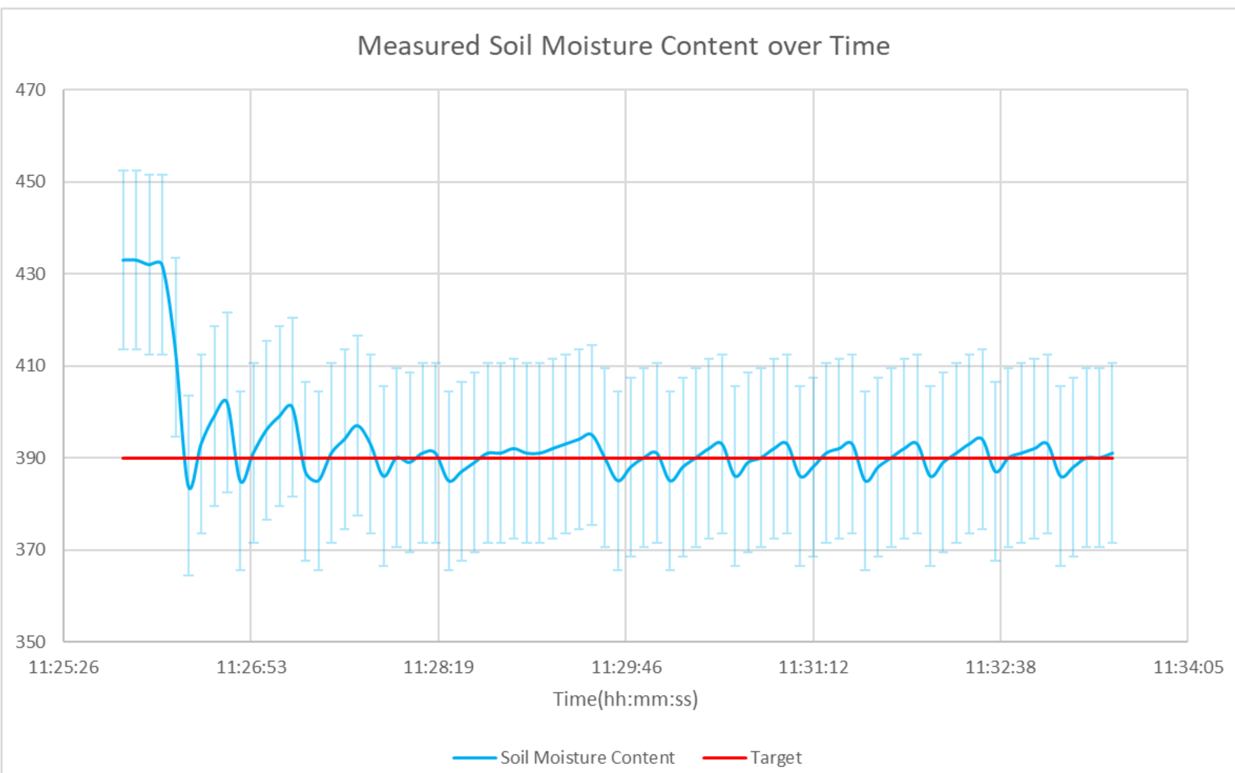


Figure 9: Result of the soil moisture change of the validation test.

7.5 Conclusion

In conclusion, the system generally performed well. Temperature was dramatically lowered within the first minute of the test, showing the ability of the emergency mode to reduce the temperature in the case of an extreme event. The temperature was then steadily lowered to the target value, with no erroneous spikes outside of tolerances outlined in the FSR. The humidity levels did show 2 erroneous spikes outside of tolerance levels, the first of which can be attributed to the emergency mode, where the misters were turned on in order to combat the high temperatures. However, the 2nd spike cannot be accounted for in this manner. Furthermore the extreme difference between the humidity measured using the greenhouse sensor vs the control sensor is most likely explained by the difference in sensor quality. Experimentally derived offsets for the humidity sensor were used in the algorithm in order to bridge the gap in data quality, however it is evident that there is more work needed in this area. The soil moisture content was able to be successfully maintained once it reached the specified value, to greater precision than expected. The error bars for that test illustrate 5% deviance rather than the planned 10% tolerable deviance.