FINAL REPORT

By

Pablo Catalan

Umme Kulsoom

Joseph Rapp

Final Report for ECE 445, Senior Design, Fall 2019

TA: Christopher Horn

11 December 2019

Project No. 04

Abstract

This report discusses implementation of an alternate form of farming, known as aeroponics, which can be adapted to further increase crop yields and mitigate the risk of food shortages. Aeroponics works by growing plants vertically and keeping the roots damp at almost all times without soil to maximize the nutrients received. Aeroponics is also beneficial because the plants no longer are bound to having to grow outside but can be raised in a controlled environment indoors. By taking all these factors into consideration, aeroponic farming is without a doubt a much more optimal choice for plant growth because not only does it take up less land and use less water, but also removes the dependency of certain crops to only be grown in certain regions or certain seasons.

Contents

1. Introduction

1.1 Purpose

By the year 2100, world population is projected to reach 10.9 billion [1]. Based on current world population statistics, this represents a 42% population increase in the next 80 years. In consideration of these statistics, a serious question to consider is whether traditional farming methods will suffice to provide the necessary produce for human flourishing. Many contemporary agriculturalists are beginning to view traditional farming methods incapable of efficiently producing the future world will require. Instead, they are increasingly looking to the concept of vertical farming which is the idea of using enclosed and controlled environment vertical infrastructure to accommodate layers of food production, thus reducing the amount of horizontal space needed.

To support the production of food in vertical structures it is expedient to integrate technologies that will facilitate and optimize the process. A method becoming popular in vertical farming is aeroponics, which uses a food/water system to feed and hydrate plant roots through injectors that disperse the nutrient-laden water as a spray. Using this method considerably reduces the amount of water needed to grow crops; to grow 1 kg of lettuce in traditional farming it would take 250L of water as opposed to 1L of water in aeroponics. The hopeful expectations of aeroponics in its ability to grow crops indoors has led to its commercialization, particularly for home production.

Many commercial aeroponic home systems do not include enclosed and controlled environments for the plant. This leaves the growth of the plant as a precarious exercise as the temperature and humidity preferences of home users vary and may not be optimal for plant growth. Moreover, as far as we have observed, all current commercial aeroponic systems use white-light LEDS which result in inefficiencies. This is because plants only absorb specific wavelengths from the EM spectrum, particularly from the blue and red channels. The rest is reflected or dissipated as heat energy.

These inefficiencies will be resolved by designing PhytoHome, a plant home that provides plants with an enclosed environment, temperature and humidity control, light from LEDs that emit only the specific wavelengths that are absorbed by plants. With these new features and technologies, an optimal environment for plant growth can be achieved and lighting power input per kg of food will decrease, improving the overall efficiency of home aeroponic systems. In additional, the greater reaching extent of PhytoHome to the larger problem of horizontal farming inefficiencies is that PhytoHome will be a scalable model with technological applications relevant to large-scale vertical farming.

1.2 High-Level Project Functionality

- PhytoHome RAS (Regulated Aeroponic System) must be able to provide a temperature assisted environment that performs heating through space heaters and provides minor cooling through a fan ventilation system and water misting.
- PhytoHome must be able to provide approximately 25-30 Watts per square foot to the plants through the LED lighting, as well as monitor ambient sunlight to determine the amount of necessary operation time.
- PhytoHome must be able to provide intermittent or constant water misting directly to the roots of the plants to provide nutrients and sustain plant life.

No modifications were made to the high-level requirements of this project since the *Project Proposal* and the *Design Document*.

1.3 Subsystems Overview

A modular representation of the key entities in the system are represented by the block diagram shown in **FIGURE 1**. The main control unit of the design is the microcontroller, which controls the habitat for the plants based off information it receives from three sensors. The photosensors indicate the brightness of the light in the lighting system, while the temperature and humidity sensors indicate the necessary heating and cooling for best ambience in the HVAC system. The water/feeding system functions based on each plant's unique nutrition schedule, and is independent of sensors save for when humidity is too low. All these subsystems are essential to provide an optimal, controlled environmental conditions in PhytoHome.

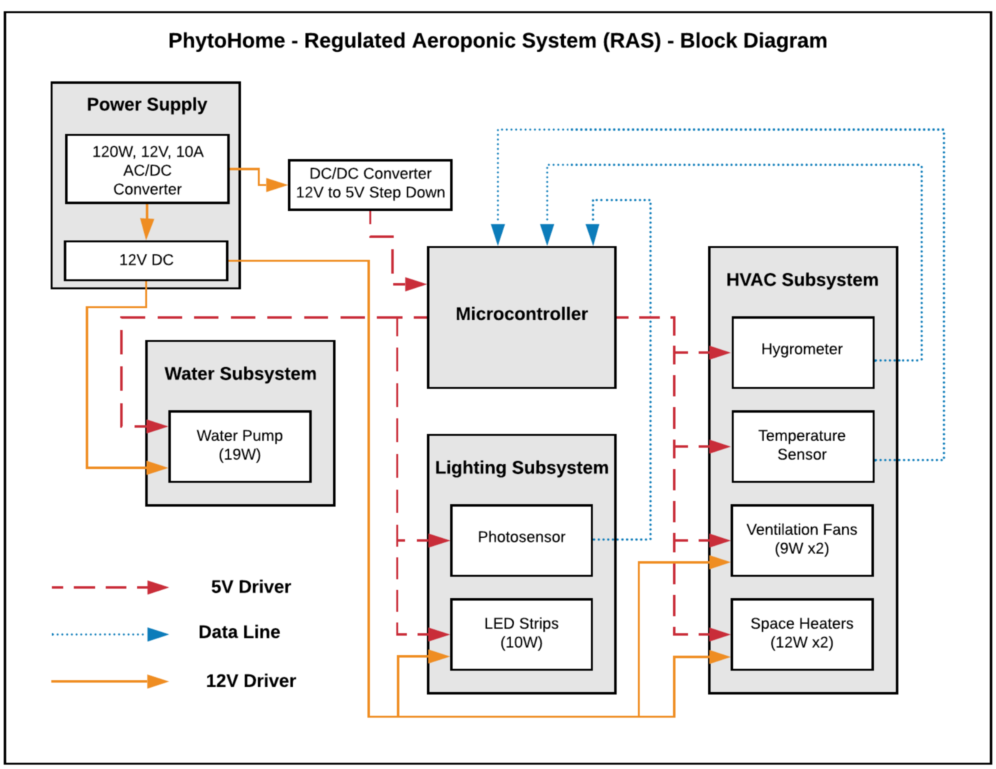

FIGURE 1: BLOCK DIAGRAM

FIGURE 2 shows the projected physical design of PhytoHome, along with where all the components are located. As shown, two circulation fans have been mounted on the top positioned beside the control system and the photosensor. Beneath the ceiling of PhytoHome are 6 LED light strips. Inside PhytoHome mounted on the side legs and opposite the fans will be the temperature sensor, hygrometer, and space heaters. The water tank, water pump, and mist injectors will be below the mesh netting where the plants will rest. The plant roots will grow below the middle mesh netting panel.

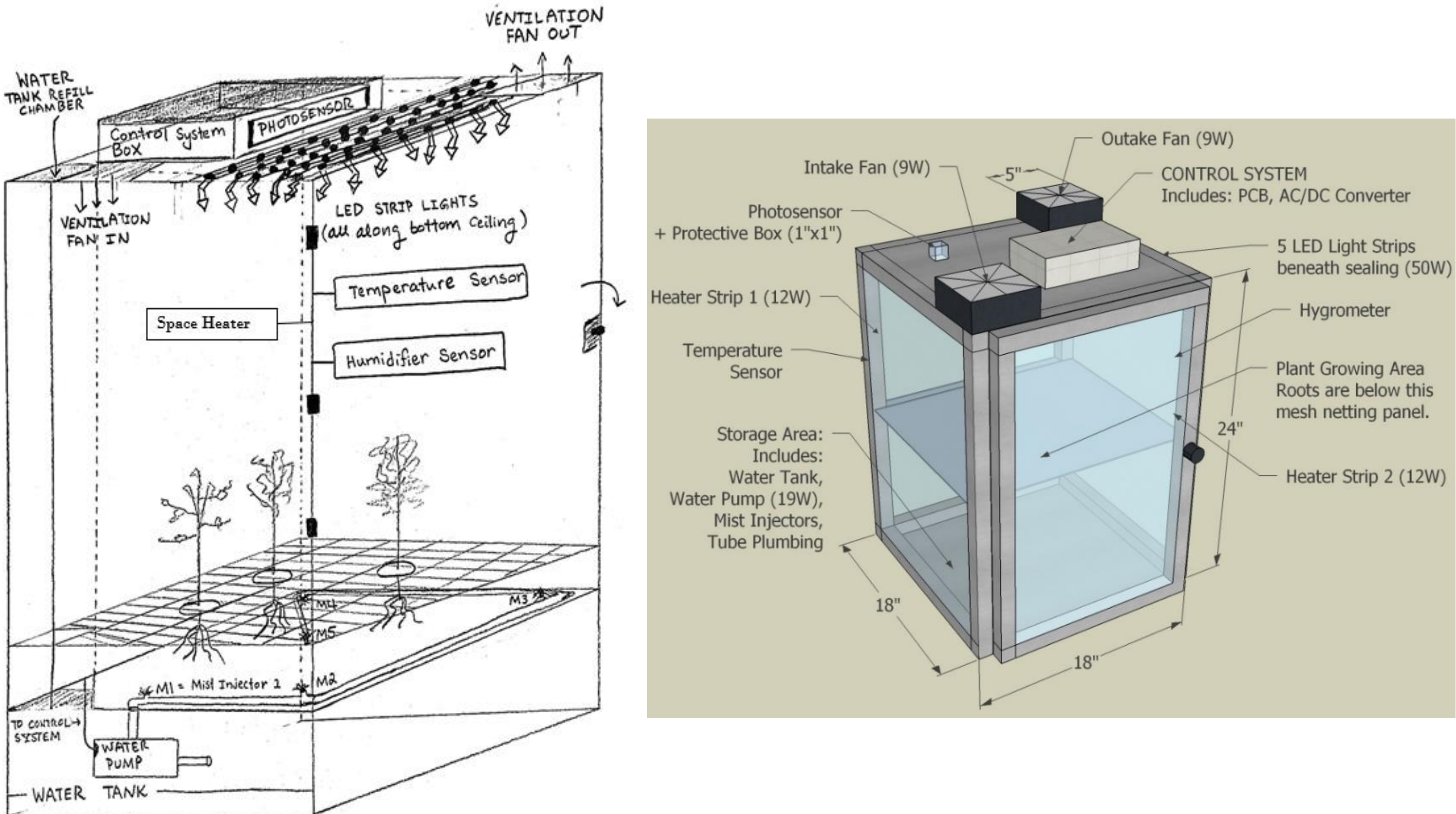

FIGURE 2. PHYSICAL DESIGN

It is important to note that there have been some changes made to the physical design since the *Project Proposal* and the *Design Document*. Originally, it was proposed that PhytoHome be made of glass and steel, whereas the actual physical design is made up of wood everywhere (no steel/aluminum panels) and the screen is made up of plexiglass. This design decision was made to cut down on building expenses. Furthermore, the humidity sensor (hygrometer) and temperature sensor are placed on the same corner inside the chamber of PhytoHome in a self-made plexiglass protection box to prevent water from getting on the sensors. There have also been some more minor changes in the physical design which will be discussed in upcoming sections of the report.

2. Design

2.1 Design Description

As shown in **FIGURE 1**, PhytoHome consists of five main subsystems that each had to be built and tested modularly before the whole system could be brought together. In order to follow an organized and cohesive manner for building and testing, a circuit schematic had to be drawn up to be used as blueprint for the design. Hence, **FIGURE 3** illustrates how each of the subsystems are to be connected for proper functionality of the project.

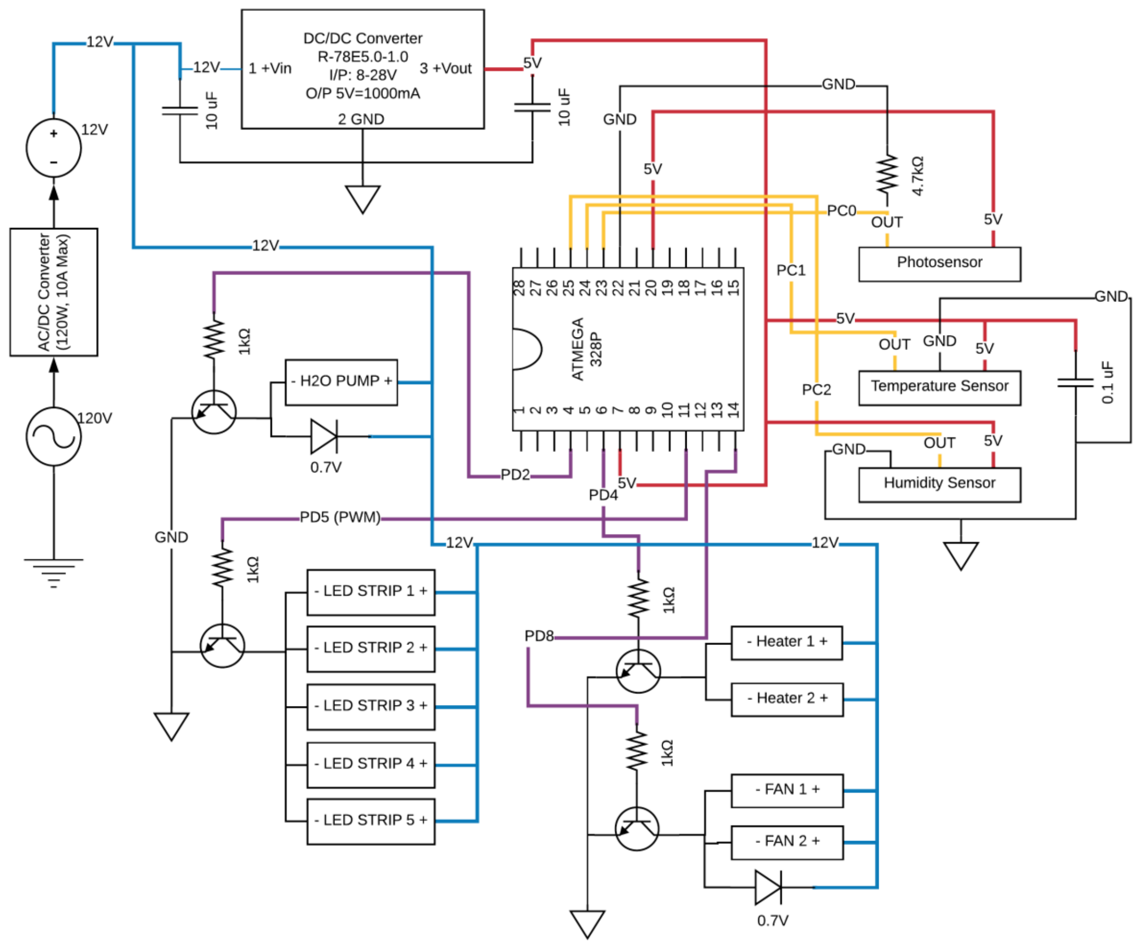

FIGURE 3: PHYTOHOME CIRCUIT SCHEMATIC

From this schematic, it can be seen that the PhytoHome PCB will consist mainly of the microcontroller, capacitors, diodes, transistors, and the DC-DC converter (note that no resistors were used in the final design because they were no longer deemed necessary during the building process). The LEDs, Water Pump, Fans, and sensors will all be separate from the PCB. The sensors are not mounted on the circuit board due to optimal design location requirements. Some modifications through soldering had to have been made on the PCB to prevent shortages of certain connections, which will be further elaborated on in the next sections.

Apart from generating a circuit schematic to aid in the wiring of the hardware, there needed to be a pattern of logic necessary for coding the software. The block diagram in **FIGURE 1** assisted in visualizing what each of the entities are, but there needed to be a methodology for the microcontroller to accurately communicate appropriate commands and responses within the system. For that reason, the following flow chart was produced. The implemented C/C++ code is attached to the *Appendix* of this report.

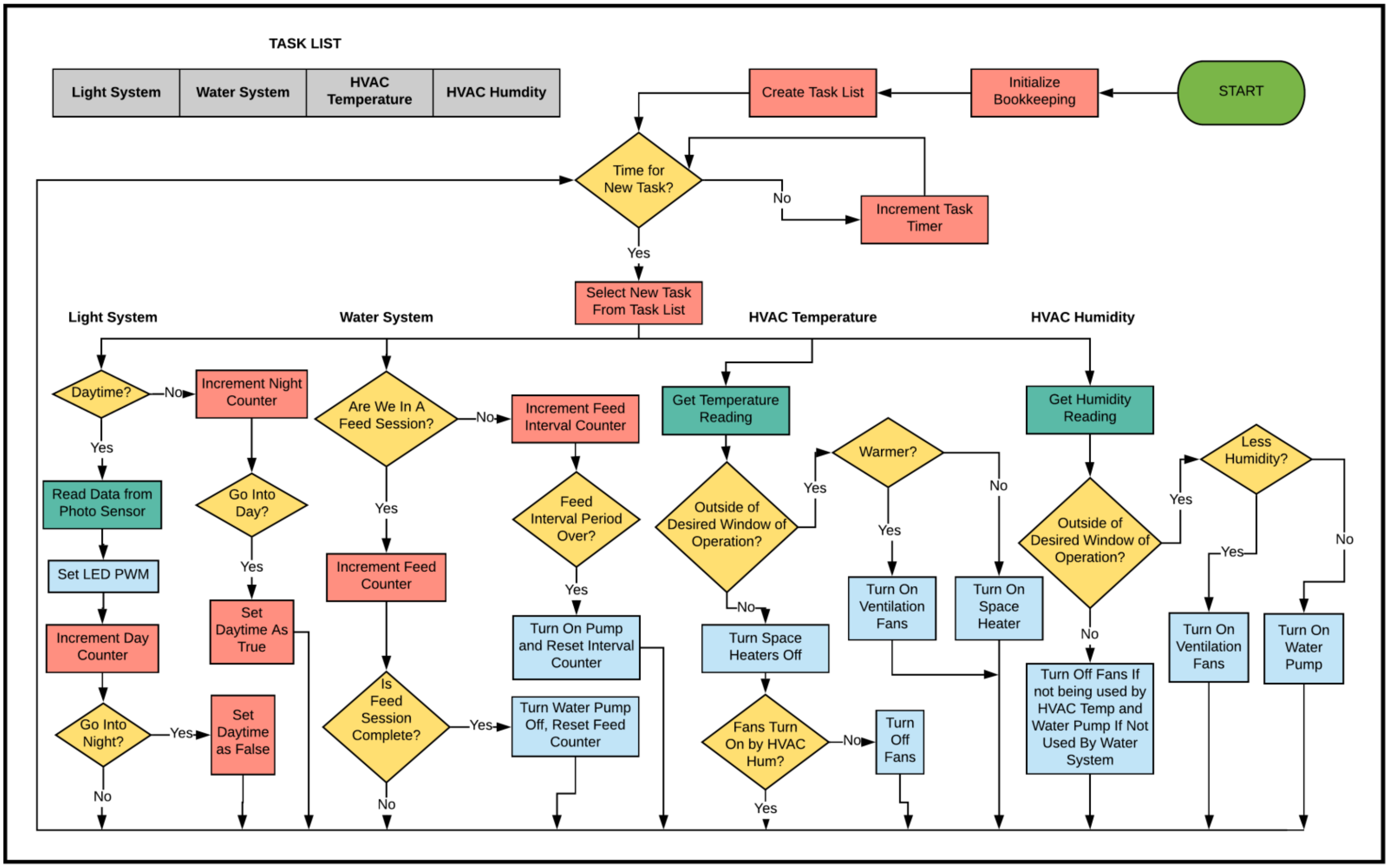

FIGURE 4: PHYTOHOME MICROCONTROLLER SOFTWARE FLOW CHART

The code loaded into the microcontroller basically simulates a night/day setting, where it keeps the lights on for 15 seconds (simulating the sun), and then off for 15 seconds (simulating nighttime), while simultaneously running the water misters every 15 seconds. The sensors also work alongside both the lighting and feeding systems by using temperature and humidity to control the space heaters and fans, and water misters.

2.2 Design Procedure

In constructing all the components of PhytoHome, it was essential to modularly verify and test that each one works independently, especially for debugging purposes. Of course, there were a multitude of approaches that could've been adapted for the designing of this system, so it was optimal to choose methods that best allocate time and resources since this project had to be completed in the course of a semester. In lieu of this, the following sections break down exactly what approach was taken in building each system and why it was chosen so.

Also, please refer to **DIAGRAM 1** in the *Appendix* for additional photographs of the completed design of *PhytoHome*; referring to it may help clarify and visualize exactly what its entities are.

2.2.1 Power Supply

The power supply has two primary tasks: the first is to supply the power to maintain constant data transfer from the temperature, humidity, and photosensor sensors to the microcontroller; the second is to provide power whenever necessary to operate the intermittent, larger loads of each subsystem, namely the ventilation fans and space heater for the HVAC system, the LEDs for the lighting system, and water pump for the water/feeding system. For the water/feeding system, power will be supplied to a motor water pump to pressurize the water tubes that connect with the injectors.

As shown in **FIGURE 3**, there are two different voltages that were necessary for this system to be successful: a 12 V line and a 5 V line. To get the 12 V line, an AC/DC convertor is used; one end is plugged into the wall and the other end is stripped so that its two 12 V and ground wires are exposed. These were soldered carefully into the 12 V rail on the PCB and the ground rail on the PCB, respectively. This was done to ensure that 12 V are always being supplied to the system, so all entities have enough power to stay running in any situation.

The 5 V line was then created by using a DC/DC convertor: one end is connected to the 12 V DC rail and the other end is connected to the 5V pin on the microcontroller. These connections were also soldered on, and any exposed wires not on the PCB were covered with electrical tape to provide insulation and reduce risks of shock.

The **DIAGRAM 2** in the *Appendix* contains the circuit for the PCB; this can be referred to for clarification and understanding of the PCB circuit layout.

2.2.2 Microcontroller

The microcontroller, ATMega328P, is the brains of *PhytoHome*. It receives data from the sensors and performs computations on this data to determine how to manage all the other subsystems, save the power supply subsystem. All sensors belonging to the Lighting and HVAC, and Water subsystems will be connected as inputs to the microcontroller while its outputs will be used to power and control the remaining entities, namely the water system pump, lighting system LEDs, and HVAC space heater, humidifier, and ventilation fans.

The rails of the microcontroller had male-head pins soldered on, and the sensors were connected via male-to-male wires. The MOSFETS for the water system, lights, space heaters, and fans were directly connected to the microcontroller onto the PCB. This was the best way to ensure that the connections would be direct and secure.

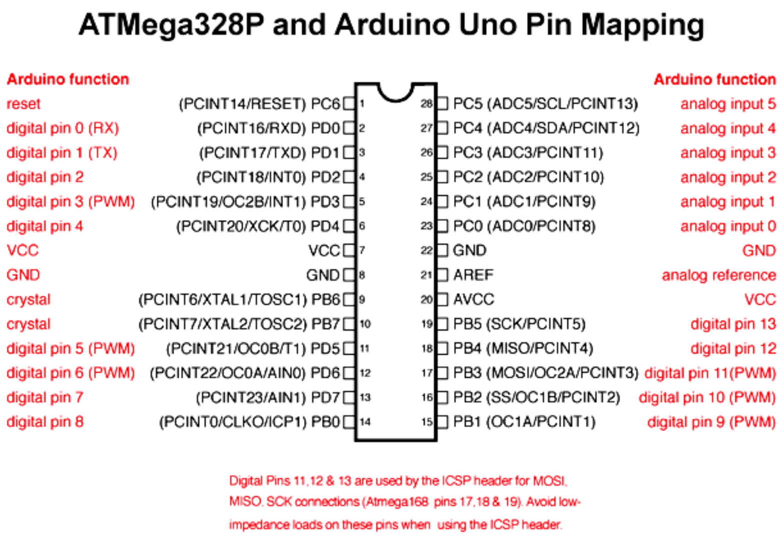

FIGURE 5. MICROCONTROLLER LAYOUT [2]

2.2.3 Water Subsystem

The water subsystem consists of the water tank, pump, and injectors positioned on the bottom of the PhytoHome system. It delivers nutrient-laden water to the roots of the plants in the feeding chamber by pressurizing service tubes with a water pump. This ensures that the amount of water and nutrients necessary for optimal plant growth is controlled efficiently. It is beneficial that this subsystem be physically attached to the overall design because it allows for ease of water transfer via the injectors, which distribute the water resourcefully.

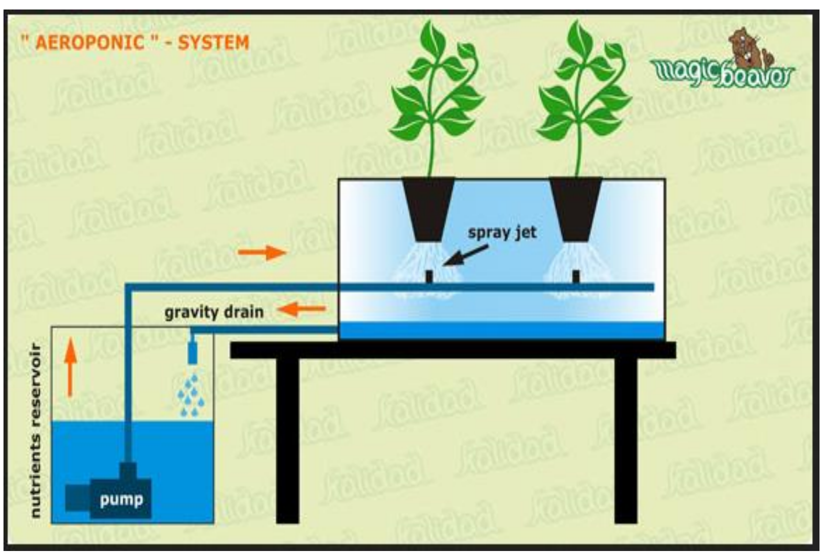

FIGURE 6. WATER PUMP & MISTING SYSTEM [3]

As shown in the above **FIGURE 6**, and in conjunction with the diagram of the system in **FIGURE 2**, the water misters and pump are placed underneath the mesh netting where plants reside. However, the positive and negative wires of the pump are brought alongside the back-corner wall of PhytoHome, through an opening in the top wooden ceiling, and then connected on the PCB via soldering at their appropriate locations. This is to ensure that the connections are secure, supported by the traces of the PCB in **DIAGRAM 2** of the *Appendix*.

2.2.4 HVAC Subsystem

The HVAC subsystem is responsible for monitoring and controlling the temperature and humidity of the plant's environment. This is achieved through sensors, ventilation fans, water misting, and a space heater. It is crucial that all these sensors and devices function accordingly so that they may send the correct signals to the microcontroller to properly adjust the environmental conditions. In order to do this, it is imperative to understand how each sensor and mechanism operates, and what parameters are needed for desired performance.

One main sensor in the HVAC subsystem is the hygrometer. This is a sensor that is used to continuously record humidity data of the environment and provide it to the microcontroller to maintain optimal humidity levels for plant growth. Most vegetative plants thrive between 50% - 70% humidity levels, while most flowering plants thrive between 50% - 60% [6]. To ensure the humidity levels in PhytoHome remain within this threshold, Honeywell's HIH-4030 humidity sensor breakout board was used.

The following formula is used to get the *true relative humidity*, not just the *relative humidity*, as interpreted from the code:

$$V_{out} = (V_{supply})(0.0062(sensorRH) + 0.16)\ ;typical\ at\ approx.\ 25℃ \qquad (2.2.4.1)$$

$$sensorRH = \frac{\frac{V_{out}}{V_{supply}} - 0.16}{0.0062} \qquad (2.2.4.2)$$

$$trueRH = \frac{sensorRH}{1.0546 - 0.00216T}\ ;T\ in\ ℃ \qquad (2.2.4.3)$$

The temperature sensor is the second sensor used in the HVAC system. It assists with determining the *true relative humidity*, and ensures that the plants are growing in an environment where the temperature maximizes their growth. Research suggests that cool season crops, such as many herbs and salad greens, thrive in temperatures between $16℃ - 22℃$, whereas warm season crops, such as tomatoes and cucumbers, thrive in temperatures between $18℃ - 28℃$ [6]. The LilyPad Temperature Sensor MCP9700 is utilized, and the following formulas will be used to take the analog data from the sensor and use it to interpret and gauge the ambient temperature:

$$V_{out} = rawTemp * \left(\frac{3.3}{1023.0}\right) \qquad (3.1.4.4)$$

$$°C = (V_{out} - 0.5) * 100 \qquad (3.1.4.5)$$

To verify that the values obtained from each sensor are accurate, their supplemental graphs given in their datasheets can be used.

As briefly mentioned before, both the temperature and humidity sensors are placed in the top back corner of the chamber in *PhytoHome* and are suspended there through Velcro and the plexiglass protection box; this box is used to shield the sensors from water and minimize any damage. Both the positives and grounds for each of the sensors are connected to the PCB via male-to-male wires and pulled through the back opening in the wooden ceiling of the home.

2.2.5 Lighting Subsystem

Finally, the lighting subsystem is purposed to provide the necessary radiation for the plants to undergo photosynthesis. The LEDs will be used to deliver roughly the proper blue (420-520nm) and red wavelengths (610-720nm) that are best absorbed by plants due to the wavelengths where peak chlorophyll production occurs, as illustrated by **FIGURE 7** shown below. Unnecessary wavelengths are thereby omitted. To determine how much LED light is needed from the system, a photosensor was used to detect natural light entering the PhytoHome.

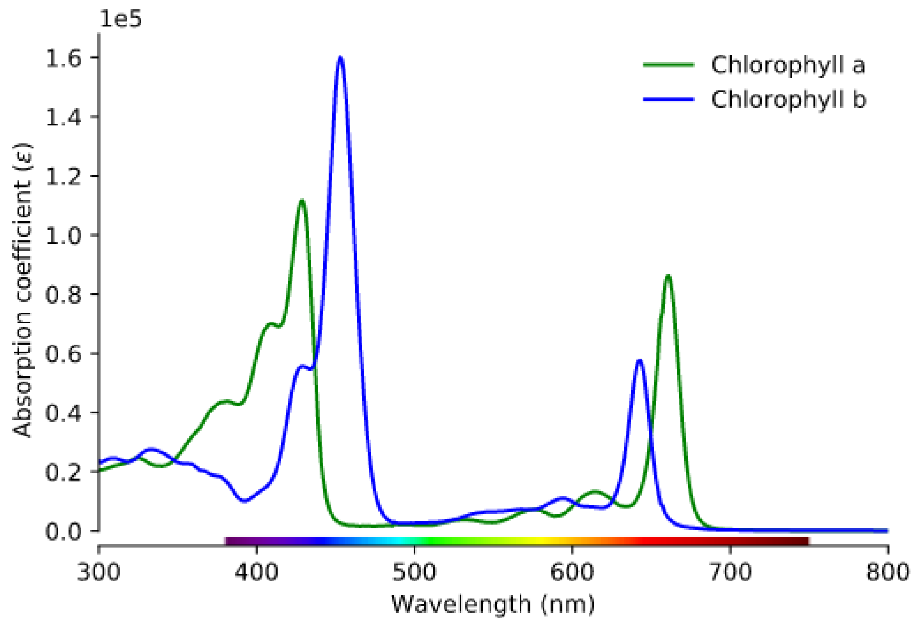

FIGURE 7. CHLOROPHYLL ABSORPTION SPECTRUM [4]

2.3 Design Alternatives

Of course, as with every design process, there comes non-ideal scenarios and problems that forces engineers to improvise a safer, yet efficient method to still achieve desired results and be successful in the implementation. In a perfect world, ideas that are built and tested work on the first try, but unfortunately, this occurs very rarely, hence begging the need for corrective actions in case of inconsistencies and design issues.

2.3.1 Inconsistencies & Justifications

During the semester while working on building PhytoHome, there were many unprecedented problems that occurred. The gravity of some problems was far greater than others, and for that reason, it was decided to compile a list of these concerns, in order of increasing impact on the design.

Challenge	**Corrective Action**
Due to limitations on the AC/DC convertor, it is suspected that microcontroller browning occurred which could have been the cause for system malfunctions when all entities were on	Decided to only connect half of the LEDS (i.e. connect 3 strips instead of all 6) and only one of the two fans. This mitigated the extent to how badly *PhytoHome* malfunctioned and *PhytoHome* was deemed to be working properly after this correct action was shown
The gauge of many of the wires used were too small and therefore cause smoking between many of the connections, especially the 12V power and ground lines	Replaced many wires by striping the black rubber part off at the end (which is where a lot of the smoking was occurring), soldering the exposed metal, crimping with a plug, and then insulating with electrical tape. For the 12V rail, wires were added in parallel to the LEDs to minimize heat power dissipation
Due to many accidental shorts occurring during the debugging process, some of the traces on the PCB were too thin and could not handle the high currents/power. This caused a lot of the traces on the PCB to be blown	Manually had to replace the blown traces on the PCB by soldering on wires in the back (can refer to **DIAGRAM 2** in the *Appendix*). Also had to re-order PCB. Soldering work was done on the new PCB to minimize risk of traces being blown. This work involved placing standard breadboard wires in parallel with several traces.

Although the pressure provided by the initial rail of misters was enough to create the mist that the design originally was intended for, this mist was a very weak spray that would not effectively reach all the roots of the plants	Because the misters were an add-on to the pump, it was decided to disconnect them and design a new misting system. This new system was able to provide a much stronger mist that was originally desired, and the extent of the spray could be controlled so that it could cover almost all the plants' roots.
Due to the manufacturer's calibration of the equipment, the temperature sensor occasional yielded data anomalies	The code, which was written in a schedular/task-list format, had to be re-written to account for these anomalies and discard any of the outlier data
The space heaters that were initially proposed to be used melted upon the first round of the integration process due to high currents/power	Disconnected and discarded of these heaters and did further research to look for more durable ones. Ended up finding more durable space heaters that provided approximately the same heating as the original proposed ones and used this instead.

3. Requirements & Verification

The requirements and verification table is given in the *Appendix* of this document to maintain brevity and conciseness of the report. Therefore, the *Appendix* is where it can be referred to for ease of understanding.

3.1 Requirements & Verification Table

TABLE 2 in the *Appendix* highlights all the necessary requirements that were needed for the successful completion of the project, along with the verification processes involved. The third column in the table confirms whether each of these requirements were met. As can be seen, each requirement was satisfied. There was virtually no changes in the requirements of the R&V table, except for one minor modification in the humidity range due to a design decision.

In the original *Design Document*, it was given that the optimal humidity for the plants to thrive in is approximately between 40%-70% [5]. Although the system could work to maintain this humidity, if the environment is too humid the fans decrease temperature during this process, potentially to an undesirable level. Since temperature is a more important element than humidity, a design choice was made to extend the range of humidity from 20% to 70% in order to ensure a more consistent temperature, while still ensuring that humidity would stay within a reasonable range.

3.2 Quantitative Results

All subsystems were tested to meet quantitative results given in the R&V table. In this section, we will particularly discuss one of these quantitative results, namely the performance of the initial water misting system.

The working pressure of the final 0.4mm diameter orifice nozzles used in the design of PhytoHome is between 2-3kg. The following conversion rates for comparing kg of pressure to PSI [7].

- 1 PSI = 6,894.76 Pa
- 1 kg/cm^2 = 98066.5 Pa

Using an orifice nozzle of 0.4mm and a pressure of 2kg, the corresponding PSI can be calculated as follows:

$$PSI\ Lower\ Bound = 2\ {}^{kg}/{}_{cm^2}\left(\frac{98066.5\ Pa}{1\ {}^{kg}/{}_{cm^2}}\right) * \left(\frac{1\ PSI}{6{,}894.76\ Pa}\right) = \sim 28.45\ PSI$$

$$PSI\ Upper\ Bound = 3\ {}^{kg}/{}_{cm^2}\left(\frac{98066.5\ Pa}{1\ {}^{kg}/{}_{cm^2}}\right) * \left(\frac{1\ PSI}{6{,}894.76\ Pa}\right) = \sim 42.67\ PSI$$

The max working pressure of the misters is around 450 PSI [6]. Using our 0.4mm orifice nozzle service system in combination with our 12V brushless submersible water pump with flow rate capacity of 800 L/H (210 GPH) and max height lift of 5m (16ft), a fine spray out of 9 mist injectors was produced, thus helping to ensure operation in the advertised PSI range, though likely far from the max working pressure of 450 PSI since the mist was not super fine nor extremely conical. This quantitative test resulted in fulfilling one of the main R&V table in the design document, namely to produce an estimated PSI of 35.

4. Costs

4.1 Parts

Below the estimated cost for the implementation of PhytoHome is calculated. **TABLE 1** represents the material costs for one *PhytoHome* unit. Note that Table 1 does not include design or R&D costs. It is only a raw material estimate. The additional design labor and R&D costs are estimated as follows.

Three team members work 10 hours per week for 16 weeks, and we estimate our time to design, implement, and develop content is worth $35/hour. Say the Machine Shop worked 60 hours to implement the physical design of *PhytoHome* and that their time is worth $40/hr. Thus, the total cost of labor for is estimated to be as follows:

$$\text{Total Student Labor Cost} = 3 * \frac{\$35}{hr} * \frac{10hr}{week} * 16\ week = \$16{,}800$$

$$\text{Machine Shop Labor Cost Estimate} = \frac{\$40}{hr} * 60hr = \$2{,}400$$

$$\text{Total Cost of Labor} = \$16{,}800 + \$2{,}400 = \$19{,}200$$

Finally, additional R&D costs are estimated to be 1x the *PhytoHome* Cost Per Unit due to testing various parts to see which ones perform best and due to potential mishaps during the design process.

Thus, the total cost of *PhytoHome* is estimated to be as follows:

$$\textbf{PhytoHome Total Cost} = \mathbf{2} * \frac{\mathbf{\$408.47}}{\boldsymbol{unit}} + \mathbf{\$19,200} = \mathbf{\$20,016.94}$$

TABLE 1: PHYTOHOME COST PER UNIT (ESTIMATE)

PART	**COST**
ATMega328P	$4.30
LilyPad Light Sensor	$3.50
LilyPad Temperature Sensor	$4.50
SparkFun Humidity Sensor Breakout	$18.95
Polycarbonate Plastic Sheet	$8.99
Plant Mesh Netting	$2.00 (estimate)
Water Tank	$7.00
Submersible Water Pump	$22.99
Misting Service System Nozzles 1	$19.99
Misting Service System Nozzles 2	$11.85 (x4)
Hose Splitter 1 (x2)	$19.88
Hose Splitter 2	$11.99

LED Grow Lights	\$8.95 (x6)
Ventilation Fans	\$17.95 (x2)
Ceramic Heating Plate	\$3.89
AC to DC Converter	\$21.99
Protective Fan Covers	\$3.50 (x2)
Acrylic Plexiglass Sheets	\$9.25 (x4)
Plexiglass Laser Cut Sheets	\$0.31 (x7)
Plastic Waterproof Junction Box	\$10.99
N-Channel MOSFETs	\$0.95 (x4)
DC/DC Converter	\$4.95
Water Pump to Hose Splitter Adapter	\$5.59
Miscellaneous (Resistors, Diode, Capacitors, Sealant/Mounting Tape, Jumper Wires, etc.)	\$20.00 (estimate)
Machine Shop Wood and materials	\$20.00 (estimate)
PCB Creation	\$10.00 (estimate)
TOTAL COST (Estimate)	**\$408.47**

4.2 Schedule & Labor

TABLE 2: DESIGN IMPLEMENTATION SCHEDULE

<table>
<tr><th>Week</th><th>Goal</th><th>Pablo Catalan</th><th>Joseph Rapp</th><th>Umme Kulsoom</th></tr>
<tr><td>10/07 - 10/13</td><td rowspan="2">Implement Lighting, HVAC, and Water Subsystems</td><td rowspan="2">Write microcontroller code based on software flow chart in figure 7 and build and build full breadboard circuit based on figure 6</td><td rowspan="2">Write microcontroller code based on software flow chart in figure 7 and build and build full breadboard circuit based on figure 6</td><td rowspan="2">Write microcontroller code based on software flow chart in figure 7 and build and build full breadboard test circuit based on figure 6</td></tr>
<tr><td>10/14 - 10/20</td></tr>
<tr><td>10/21 - 10/27</td><td>Test Design, PCB Design, Complete Machine Shop Deliverables</td><td>Test code and circuit and design PCB</td><td>Test code and circuit and work on physical design drawing for Machine Shop</td><td>Test code and circuit and design PCB</td></tr>
<tr><td>10/28 - 11/03</td><td>Machine Shop Meeting + Drawing Deliverables</td><td>Test code and circuit and design PCB</td><td>Test code and circuit and design PCB</td><td>Test code and circuit and design PCB</td></tr>
</table>

	Deadline + Test Design			
11/04 - 11/10	Test Design and Prepare for Incorporating Design into Physical PhytoHome	Test and Prep for Incorporating Design	Test and Prep for Incorporating Design	Test and Prep for Incorporating Design
11/11 - 11/17	Incorporate Physical Design into PhytoHome and Test	Incorporate Design and Test – Prepare for Demo	Incorporate Design and Test – Prepare for Demo	Incorporate Design and Test – Prepare for Demo
11/18 - 12/01	**11/18 - 12/01**	Write Final Report	Write Final Report	Write Final Report

5. Conclusion

5.1 Accomplishments

To reflect over the entire design process of this system, it is important to point out and understand what *PhytoHome's* strengths and weaknesses were. For the accomplishments, the main thing to note is that *PhytoHome* was indeed able to achieve all its intended requirements (mentioned in the high-level requirements portion of this report). It delivered, at the minimum, what it promised to do. Through the design process we learned that even though *PhytoHome* only had one layer where plants can grow, it does harbor the ability to include various layers/shelving for more large-scale vertical productions. It could even have each layer be unique to the type of plant grown by altering the temperature, humidity, and amount of water delivered. If *PhytoHome* were to be implemented in a production plant, it would be able to, in theory, grow crops of all seasons under one roof simply by making minor adjustments to its environmental settings. The prototype of *PhytoHome* built in this class was targeted for in-home use specifically, and because the system itself isn't all that large or heavy, it makes it portable for the user. Wheels could be added to it to make it even simpler for transport.

5.2 Uncertainties

To discuss some of the weaknesses and uncertainties of *PhytoHome*, it is important to start out with the fact that tests were never performed to see if a plant could survive in this system from seed to maturity. The reason for this is that this project needed to be completed within the scope of the semester, and thus due to the limited time, not enough trials could be performed to verify this notion. Secondly, there needs to be more diagnostics performed on the AC/DC convertor to figure out exactly why the system malfunctions when all its entities are on. For the purpose of the class, it was sufficient to demonstrate that *PhytoHome* works with just one ventilation fan, one space heater, and three out of the six strips of LEDs. However, it would be optimal to try to incorporate both fans, all the LEDs, and even multiple space heaters so that the system can run as efficiently as possible. Furthermore, for this current prototype, *PhytoHome* was made from wood, which was a somewhat poor long-term decision because the water from the misting system will eventually cause the wood to rot. The reason wood was chosen was because it is cheap and easier to work with than using steel or aluminum. Using wood, however, sacrifices *PhytoHome's* durability. For future designs, it would be best to make *PhytoHome* with a sturdier material, even if it raises the costs a bit, so that the system lasts longer.

5.3 Ethical considerations

A legitimate safety issue in creating PhytoHome is that it was undertaken by engineering students with limited professional experience. In IEEE Code of Ethics statement #7 mentions undertaking "technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations" [7]. This means it was ethically necessary to fully disclose pertinent limitations, namely lack of experience that increases the risk of accidents and warrants additional measures of technical review and criticism.

There are two safety issues relevant to the project. The first is of potentially mis-designing the circuit to cause electrical failure in such a way that it increases the risk of electrical shock, burns, and fire hazards. The second is the danger of having electronics operating in and/or near water sources. While considering these two safety issues have arose during the development of the project, it is important to consider the ethical obligation to avoid harm that is set forth in section 1.2 of the ACM Code of Ethics and Professional Conduct, part of which says, "Well-intended actions, including those that accomplish assigned duties, may lead to harm" [8]. Thus, even though the team did embark on a process of well-intended actions in order to solve a problem considered serious to the bettering of the world, these actions could have unintentionally lead to harm, and the team was, as the ACM code goes on, "obliged to undo or mitigate the harm as much as possible" [8].

Intentional or accidental misuse of the product is foreseen if the mechanical design is altered in such a way the safety precautions which involve separation of circuitry from water and tampering with the location of the heat source to place it in an area of PhytoHome that increases user risk of burns. Additional misuse that that current design would not have control over could come from use of PhytoHome for a process that facilitates the growth of plants used in the production of illegal drugs.

Thus, it is important to best avoid these potential ethical breaches and safety issues by disclosing any obligated ethical information, consulting superior technical support when deemed necessary, and through careful design to limit device alterability and testing of *PhytoHome* in order to ensure safety measures.

5.4 Future work

Any future work would be intended to improve the efficiency and efficacy of *PhytoHome's* systems themselves and secondly to improve *PhytoHome's* user interface, which was deliberately overlooked for this first design and implementation of PhytoHome. Regarding system improvement, an exciting undertaking is the development of a multi-frequency emitting LED system coupled with complementary software that controls the LED system driver to emit any visible wavelength desired. Additionally, the software would have access to a database of plant absorption spectrums and would, through time-slicing, replicate the absorption pattern's emission. In theory, this would ensure that the plant would receive every wavelength its needs at maximum efficiency. Furthermore, in the data processing of the information received by the light sensor, it would be expedient to assess the wavelengths being picked up as the ambient light to eliminate cases in which wavelengths not useful for plant growth make up the majority thereof and bamboozle PhytoHome into thinking LEDs should be set low.

For user-interface improvement, future *PhytoHome* designs would consist of a display unit to display the temperature and humidity of the system to a user. In addition to this would be an interface in which the user can select the type of plant that will be grown in order to initialize the optimal values for plant-health parameters. One can envision navigating this interface and selecting Aloe Vera or a rarely observed exotic plant indigenous to the Amazon forest. Lastly, an alert system could be integrated to inform the user when water levels of the water tank are low, and that refilling is necessary.

References

[1] “Population of the World Today,” Our World in Data. [Online]. Available: https://ourworldindata.org/world-population-growth [Accessed: 09-17-2019]

[2] “ATMega328P Microcontroller,” Components 101. [Online]. Available: https://components101.com/microcontrollers/atmega328p-pinout-features-datasheet [Accessed: 10-27-2019]

[3] MagicBeaver. [Online] Available: https://commons.wikimedia.org/wiki/File:Systeme_AEROPONIC_573px.jpg [Accessed: 12/0/2019] No changes were made.

[4] “File:Chlorophyll Absorption Spectrum.svg” – Wikimedia Commons [Online]. Available: https://commons.wikimedia.org/wiki/File:Chlorophyll_Absorption_Spectrum.svg [Accessed: 10/08/2019]

[5] “TEMPERATURE AND HUMIDITY,” HealingCanna. [Online]. Available: https://www.thehealingcanna.com/growroom-temperature-humidity#targetText=Ideal%20humidity%20levels%20in%20a,to%2060%25%20for%20flowering%20plants. [Accessed: 10/02/2019]

[6] “1/4" Slip-Lock, Outdoor Mist Tee Nozzle, Water Brass Misting Mister Nozzle, Misting Nozzles Kit, with Thread 10/24 UNC Tees 10pcs and 0.4mm Orifice Nozzle 12pcs, for Outdoor Cooling System,” – Hylaea on Amazon.com. [Online]. Available: https://smile.amazon.com/gp/product/B07BFVGZGN/ref=ppx_yo_dt_b_asin_title_o08_s02?ie=UTF8&psc=1 [Accessed: 10-28-2019]

[7] “IEEE Code of Ethics,” IEEE - Advancing Technology for Humanity. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 09-15-2019]

[8] “ACM Code of Ethics and Professional Conduct,” ACM - Association for Computing Machinery. [Online]. Available: https://www.acm.org/code-of-ethics [Accessed: 09-15-2019]

Appendix

DIAGRAM 1

DIAGRAM 2

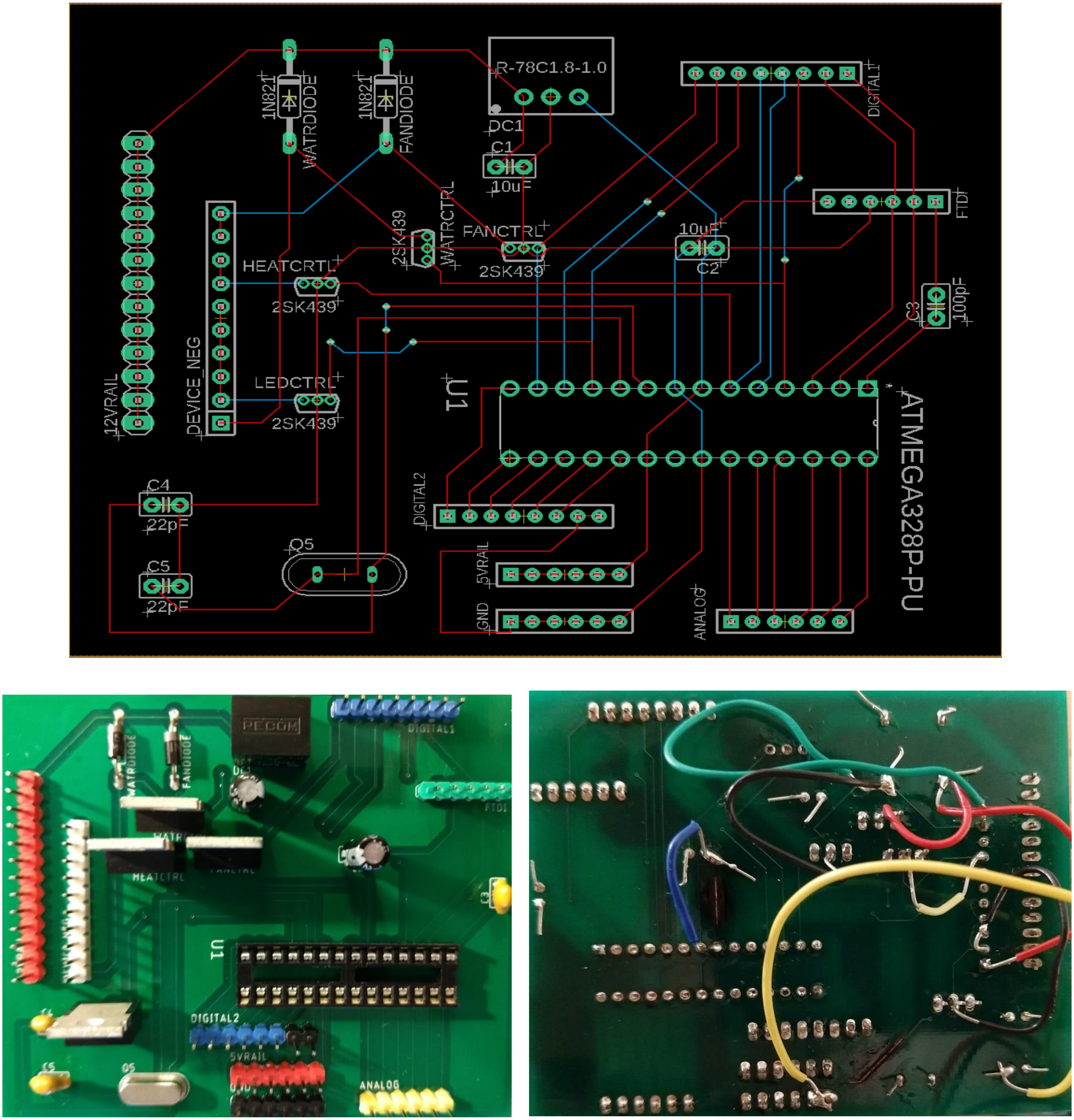
R-78C1.8-1.0
DC1
C1
10uF
1N821
WATRDIODE
1N821
FANDIODE
DIGITAL1
FTDI
12VRAIL
DEVICE_NEG
HEATCRTL
2SK439
LEDCTRL
2SK439
2SK439
WATRCTRL
FANCTRL
2SK439
10uF
C2
C3
100pF
U1
ATMEGA328P-PU
C4
22pF
C5
22pF
Q5
DIGITAL2
5VRAIL
GND
ANALOG
PECOM
WATRDIODE
FANDIODE
DIGITAL1
FTDI
HEATCTRL
U1
DIGITAL2
5VRAIL
ANALOG

TABLE 2: REQUIREMENTS AND VERIFICATION TABLE

Requirements	Verification	Verification Status (Y/N)
<u>Power System</u> 1. Outputs 12V ± 5%, 6.92A ± 5% to the all PhytoHome Subsystems (~102 W): 46 W for HVAC system, 30 W for lighting system, 19 W for water pump, 4 W for all sensors, 3 W for microcontroller. 2. Outputs 5V ± 5%, 1A ± 5% to the microcontroller which powers the photosensor, temperature sensor, and humidity sensor.	1. The AC/DC converter rated voltage and current are 12V and 10A, respectively. This means that we are within the limits of the AC/DC converter expected performance. To quantitatively ensure the necessary power is being delivered is to measure the voltage across the power lines. This will also be verified by evaluating the performance of the individual subsystems as described in the other requirements. 2. Connect the DC to DC converter's Vin to 12V coming from AC to DC adapter. Also connect to GND Then connect the DC to DC's Vout to the 5V power line on the microcontroller PIN 7 (PD7) Use a voltmeter to ensure that 5V lines are being maintained.	**Y**

3. In case of a short or unusually large surgency of voltage being drawn through the AC to DC adapter from the wall outlet, the device will automatically shut down and halt all power/voltage signals to rest of the system	3. The AC to DC converter has a built-in fuse that detects for any abnormally high levels of voltage	
<u>Microcontroller</u> 1. The microcontroller serves 5V ± 5%, 1A ± 5% to photo/temperature/humidity sensors and also biases MOSFET driver circuits into saturation to serve as switching circuits.	1. Connect microcontroller PIN 20 (AVCC) to the 5V power lines for the photo/temperature/humidity Sensors. Use a voltmeter to ensure that 5 V are in fact being sourced from the microcontroller PIN 20 (AVCC) If voltage being delivered from PIN 20 (AVCC) is above the maximum voltage rating, insert a resistor to step voltage down to desired value Connect microcontroller PIN 22 (GND) to ground, and connect the grounds of photo/temperature/humidity sensors to this pin too Use a voltmeter to ensure that 0 V is in fact the voltage reading at microcontroller PIN 22 (GND) 2. Connect microcontroller PIN 23 (PC0/ADC0) to the	**Y**

2. The microcontroller collects data from the photosensor, the temperature sensor, and the humidity sensor as an analog read, and then biases the MOSFET driver circuits accordingly.	output signal “S” line of the photosensor (light sensor) Connect microcontroller PIN 24 (PC1/ADC1) to the output signal “S” line of the temperature sensor Connect microcontroller PIN 25 (PC2/ADC2) to the output signal “OUT” line of the humidity sensor Connect microcontroller PIN 4 (PD2) to the MOSFET of water pump Connect microcontroller PIN 6 (PD4) to the MOSFET of the space heater Connect microcontroller PIN 11 (PD5) to the MOSFET of the LED strips Connect microcontroller PIN 13 (PD7) to the MOSFET of the ventilation fan	
<u>Water Subsystem</u> 1. According to the algorithm of the control unit, the switch circuit of the water pump will be switched on to supply 12V to the water pump, ensuring that the proper PSI (estimated 35 PSI) is provided to the water service tubes to obtain a mist injection to the root feed chamber.	1. Make sure that the microcontroller PIN 4 (PD2) is to be connected to the gate of a MOSFET. Double check that the drain of this MOSFET will be connected to the negative terminal of the water pump. Qualitatively, this can be checked by visually observing the spraying of the injectors.	**Y**

	Measure the collector current and ensure that it's in saturation with the provided voltage of the microcontroller at the gate of the MOSFET	
<u>HVAC Subsystem</u> 1. The HVAC subsystem will maintain a humidity level between 20-70% for optimal plant growth	1. Evaluate the readings from the humidity sensor using the microcontroller. Based on these readings turn on/off the necessary systems. During testing, use **FIGURE 10** to ensure that the output voltage readings displayed on a serial monitor (when first uploading Arduino code) are in accordance with the *relative humidity* readings in the figure The output voltage readings should yield values between 2.2 V and 2.9 V for a 20% - 70% *relative humidity* If humidity does not fall in these ranges and is too <u>*high*,</u> send signal from microcontroller PIN 13 (PD7) to turn on ventilation fans by sending a high signal to the MOSFET until the humidity range is sufficiently back into the desired range	**Y**

	If humidity does not fall in these ranges and is too <u>*low*</u>, send signal from microcontroller PIN 4 (PD2) to turn on misters by sending a high to the gate of the water pump MOSFET	
2. The temperature sensor needs a minimum input voltage of 2.3 V DC and a maximum input voltage of 5.5 V DC, with the recommended operating voltage as 5.0 V	2. Use a voltmeter to check that the voltage maintained in the + terminal of the temperature sensor from PIN 20 (AVCC) of microcontroller is approximately 5V DC. Visually ensure that this voltage reading is not below 2.3 V DC, nor above 5.5 V DC	
3. The HVAC subsystem must maintain a temperature of above 21 degrees Celsius The HVAC subsystem will work to cool PhytoHome when the ambient is 32 degrees Celsius or above	3. During testing, use **FIGURE 11 MCP9700** data line to ensure that the output voltage readings displayed on a serial monitor are in accordance with the *ambient temperature* readings in the figure. If temperature does not fall in these ranges and is too <u>*low*</u>, send signal from microcontroller PIN 6 (PD4) to turn on space heater by sending a high signal to the gate of the heater MOSFET until the sensor detects temperature has increased up to the desired range If temperature does not fall in these ranges and is too	

	high, send signal from microcontroller PIN 13 (PD7) to turn on ventilation sending a high signal to the gate of the fan MOSFET. This signal should stay high (i.e. ventilation fans should remain ON) until sensor detects temperature has decreased down to the desired range	
Lighting Subsystem 1. Based on the photosensor lux detection of up to 10k lux, the LEDs will vary their output lux to work in conjunction with ambient light to ensure the plants are always receiving at least 25-30W/square foot during the daily light cycle.	1. During testing, use **FIGURE 15** to ensure that the output voltage readings displayed on a serial monitor are in accordance with the *luminescence* readings in the figure, approximately 1.5 V. If the photosensor detects that lux levels fall below 10,000 lux, send a signal from microcontroller PIN 11 (PD5) to turn on LEDs to supply the deficit of the ambient light to maintain lux levels within PhytoHome at approximately 10,000 lux	**Y**

PhytoHome Code

```
int light_sensorPin =  A0;

int temp_sensorPin  =  A3;

int humid_sensorPin =  A5; //A2;


/*///// ACTUATOR PIN MACROS////*/

#define WATERPUMP     6

#define LEDS          5

#define SPACEHEATERS  4

#define FANS          7



#define NUMSYS        4

#define LIGHTSYS      0

#define WATERSYS      1

#define HVACSYS_T     2

#define HVACSYS_H     3



/*///// Temperature System Macros///////*/

#define TEMP_LOWER_BOUND    10

#define TEMP_UPPER_BOUND    35

#define TARGET_TEMP         25

#define TARGET_TEMP_OFFSET  2

#define DELTA_TEMP          6


/*///// Humidity System Macros/////////*/

#define HUM_LOWER_BOUND    20

#define HUM_UPPER_BOUND    70


#define TARGET_HUM         60

#define TARGET_HUM_OFFSET  3

#define DELTA_HUM          5

/*/////// LIGHT system ////////////*/

#define MAXPWM      100

#define LIGHTQUOTA  20000000

#define NIGHTTIME   20000000

#define FEEDPERIOD  15000000/* period in milliseconds*/

#define FEEDINTERVALPERIOD 10000000


/* Structure that will serve as bookkeeping for PhytoHome*/


typedef struct water_sys

{

 unsigned long period;

 unsigned long p_baseTime;

 unsigned long session;

 unsigned long s_baseTime;

 bool pumpON;

}water_sys_t;


typedef struct light_sys
```

```
{
  unsigned long lightCounter;
  unsigned long lc_baseTime;
}light_sys_t;

typedef struct phytoHome
{
 unsigned long nightTimeCounter;
 unsigned long ntc_baseTime;
 bool daytime;
 int (*system)();

 water_sys_t water_sys;

 light_sys_t light_sys;

 bool hvacTemON;
 bool hvacHumON;
 bool hvacFansON;
} phytoHome_t;
phytoHome_t phytoHome;
int task_list[NUMSYS];
int current_task = -1;
unsigned long time0 = 0;

void setup()
{
  // set the water pump pin as OUTPUT
pinMode(WATERPUMP, OUTPUT);

// set LEDs pin as OUTPUT
pinMode(LEDS, OUTPUT);

// set space heater pin as OUTPUT
pinMode(SPACEHEATERS, OUTPUT);

// set HVAC ventilation fans
pinMode(FANS, OUTPUT);

// Set the light sensor pin as an INPUT:
pinMode(light_sensorPin, INPUT);

// Set the temperature sensor pin as an INPUT:
pinMode(temp_sensorPin, INPUT);

// Set the humidity sensor pin as an INPUT:
pinMode(humid_sensorPin, INPUT);

// Set the light sensors as an INPUT
pinMode(light_sensorPin, INPUT);
// initializing Bookkeeping
phytoHome.water_sys.period = 0;
phytoHome.water_sys.session = 0;
phytoHome.water_sys.p_baseTime = 0;
phytoHome.water_sys.s_baseTime = 0;
phytoHome.water_sys.pumpON = false;
```

```
  phytoHome.light_sys.lightCounter = 0;
  phytoHome.light_sys.lc_baseTime = 0;

  phytoHome.daytime = true;
  phytoHome.nightTimeCounter = 0;
  phytoHome.ntc_baseTime = 0;
  phytoHome.hvacTemON = false;
  phytoHome.hvacFansON = false;
  phytoHome.hvacHumON = false;
  task_list[LIGHTSYS] = &light_system;
  task_list[WATERSYS] = &water_system;
  task_list[HVACSYS_T] = &temperature_system;
  task_list[HVACSYS_H] = &humidity_system;
  // Initialize Serial, set the baud rate to 9600 bps.
  Serial.begin(9600);
}
void loop()
{
  delay(100);
  time0 = micros();
  scheduler();
}
void scheduler()
{
  current_task = (current_task + 1)%NUMSYS;
  phytoHome.system = task_list[current_task];

//  Serial.println("hum");
if(current_task != -1){
  phytoHome.system();
}
}
//////////////////////////////////////////////////////// function to feed/mist plants
void water_system()
{
  if(!phytoHome.water_sys.pumpON){

     phytoHome.water_sys.period += time0 - phytoHome.water_sys.p_baseTime;
     phytoHome.water_sys.p_baseTime = time0;
     if(phytoHome.water_sys.period >= FEEDINTERVALPERIOD){
      Serial.print("WATER PUMP");
        analogWrite(WATERPUMP, 255);
        phytoHome.water_sys.pumpON = true;
        phytoHome.water_sys.period = 0;
        phytoHome.water_sys.s_baseTime = time0;
      }
  }
  else{

      phytoHome.water_sys.session += time0 - phytoHome.water_sys.s_baseTime;
      phytoHome.water_sys.s_baseTime = time0;
      if(phytoHome.water_sys.session >= FEEDPERIOD){
```

```
        Serial.print("Water Pump OFF");
        analogWrite(WATERPUMP , 0);
        phytoHome.water_sys.pumpON = false;
        phytoHome.water_sys.session = 0;
        phytoHome.water_sys.p_baseTime = time0;
        }
    }
  }
///////////// Light System
int light_system()
{
  if(phytoHome.daytime){
//    Serial.println("day");
      phytoHome.light_sys.lightCounter += time0 - phytoHome.light_sys.lc_baseTime;
      phytoHome.light_sys.lc_baseTime = time0;

      if(phytoHome.light_sys.lightCounter <= LIGHTQUOTA){
        long lightValue;
        float emulatedPWM;
        lightValue = analogRead(light_sensorPin);
//        Serial.print(lightValue);
      //  Serial.println();
        emulatedPWM = lightValue*(MAXPWM/1023.0);
        analogWrite(LEDS,MAXPWM - emulatedPWM);
      //  Serial.println();
      }
      else{
        phytoHome.daytime = false;
        phytoHome.light_sys.lightCounter = 0;
        phytoHome.nightTimeCounter = 0;
        phytoHome.ntc_baseTime = time0;
        analogWrite(LEDS, 0);
        }
  }
  else{
    // nighttime
     phytoHome.nightTimeCounter += time0 - phytoHome.ntc_baseTime;
     phytoHome.ntc_baseTime = time0;
     if(phytoHome.nightTimeCounter >= NIGHTTIME){
          phytoHome.daytime = true;
          phytoHome.nightTimeCounter = 0;
          phytoHome.light_sys.lc_baseTime = time0;
    }
  }
}
////// Temperature System Function
void temperature_system()
{
  float current_celsius; // variable to store new tempp data readings
  float static prev_celsius = -1;// previous reading

  current_celsius = readTemp(); // read the current temperature data
```

```
  isTempValid(&current_celsius, prev_celsius);   // verify its a good reading

  Serial.println("Current Temp");

  Serial.println(current_celsius);


  if(current_celsius >= (TARGET_TEMP - TARGET_TEMP_OFFSET) && current_celsius <= (TARGET_TEMP + TARGET_TEMP_OFFSET))

  {

      digitalWrite(FANS, LOW);

      digitalWrite(SPACEHEATERS, LOW);

      phytoHome.hvacTemON = false;

      phytoHome.hvacFansON = false;

      return;

  }


  // check if we are outside desired temperature window

  if((current_celsius < TEMP_LOWER_BOUND || current_celsius > TEMP_UPPER_BOUND) || phytoHome.hvacTemON)

  {

     phytoHome.hvacTemON = true;

    if(current_celsius < TARGET_TEMP - TARGET_TEMP_OFFSET)

      {

        Serial.println("spaceheaters");

        digitalWrite(FANS, LOW);

        digitalWrite(SPACEHEATERS, HIGH);

        phytoHome.hvacFansON = false;

      }

      else

      {

        Serial.println("fans temp");

        digitalWrite(SPACEHEATERS, LOW);

        digitalWrite(FANS, HIGH);

        phytoHome.hvacFansON = true;

      }


  }

}

///////// Temperature Helper //Functions

* readTemp

* input: none

* output: temperature in Celsius

*

*/

float readTemp()

{

      float current_celsius; // Variable to store Celsius value

      float rawTemp;

      float voltage; // Variable to store voltage calculation

      rawTemp = analogRead(temp_sensorPin); // Read the raw 0-1023 value of temperature into a variable.

      voltage = rawTemp * (5.0 / 1023.0);

      current_celsius = (voltage - 0.5) * 100;

//      printTempTest(rawTemp, voltage);
```

```
    return current_celsius;

}


/*

* isTempValid

* input: current and prev data readings; current is passed by reference.

* output: NONE

* Effect: the current data reading will be change accordingly as described in the following sentence. If

* the current data reading is an anomaly, then its value will be changed to that of the previous; otherwise, it remains the same.

*/

void isTempValid(float* current_celsius, float prev_celsius)

{

   if(prev_celsius != -1)

  {

   if(abs(prev_celsius - *current_celsius) >= DELTA_TEMP)

   {

     *current_celsius = prev_celsius;  // the current value was an anomaly

   }

  }

}


/*

* printTempTest

* effects: prints voltage and celsius readings

*

*/

void printTempTest(float celsius, float voltage)

{

   // use these prints for debugging

  Serial.print("Voltage: "); // Print voltage reading to serial monitor

  Serial.println(voltage);

  Serial.print("Celsius: "); // Print celcius temp to serial monitor

  Serial.println(celsius);

 //   Print a blank line

  Serial.println();

}

////      Humidity System Function

void humidity_system()

{

  float current_humidity; // variable to store new tempp data readings

  float static prev_humidity = -1;// previous reading


  current_humidity = readHumidity(); // read the current temperature data

//  Serial.println(current_humidity);

  isHumidityValid(&current_humidity, prev_humidity);   // verify its a good reading

  Serial.println("Humidity" );

  Serial.println(current_humidity);
```

```
  if(current_humidity >= (TARGET_HUM - TARGET_HUM_OFFSET) && current_humidity <= (TARGET_HUM + TARGET_HUM_OFFSET))

  {

    if(!phytoHome.hvacFansON){

     digitalWrite(FANS, LOW);}



     if(!phytoHome.water_sys.pumpON){

     analogWrite(WATERPUMP, 0);}



     phytoHome.hvacHumON= false;

     return;

  }



  // check if we are outside desired humidity window

  if((current_humidity < HUM_LOWER_BOUND || current_humidity > HUM_UPPER_BOUND) || phytoHome.hvacHumON)

  {

       phytoHome.hvacHumON = true;

       if(current_humidity < TARGET_HUM - TARGET_HUM_OFFSET)

      {

//        Serial.println("Watermpump ON- humdity");

        if(!phytoHome.hvacFansON){

          digitalWrite(FANS, LOW);

        }

        analogWrite(WATERPUMP, 255);

      }

      else

      {

        Serial.println("Fans High Humdidity");



      if(!phytoHome.water_sys.pumpON){

       analogWrite(WATERPUMP, 0);}

        digitalWrite(FANS, HIGH);

      }

  }



}



//Humidity Helper Functions

/*

* readHumidity

* inputs: NONE

* outputs: humidity reading (trueRH)

*/

float readHumidity()

{

  // Variable to store raw temperature

  long rawHumid;

  // Variable to store voltage calculation

  float voltage;

  float v_out;

  float sensorRH;

  float trueRH;

  float temp;
```

```
  // Read the raw 0-1023 value of temperature into a variable.

  rawHumid = analogRead(humid_sensorPin);

  temp     = analogRead(temp_sensorPin);


  // figure out how to add actual temp for

v_out = rawHumid * (5 / 1023.0);

sensorRH = (v_out/(.0062 * 5)) - 25.81;

trueRH = sensorRH/(1.0546 - (0.00216 * 8.89));

//printHum(sensorRH, trueRH, v_out);

return trueRH;

}

/*

* isTempValid

* input: current and prev data readings; current is passed by reference.

* output: NONE

* Effect: the current data reading will be change accordingly as described in the following sentence. If

* the current data reading is an anomaly, then its value will be changed to that of the previous; otherwise, it remain the same.

*/

void isHumidityValid(float* current, float previous)

{

    if(previous != -1)

  {

    if(abs(previous - *current) >= DELTA_HUM)

    {

      *current = previous;  // the current value was an anomaly

    }

  }

}

void printHum(float sensorRH, float trueRH, float v_out){

Serial.print("Relative Humidity: "); // Print voltage reading to serial monitor

Serial.println(sensorRH);

Serial.print("True Humidity: "); // Print voltage reading to serial monitor

Serial.println(trueRH);

Serial.println(v_out);}
```