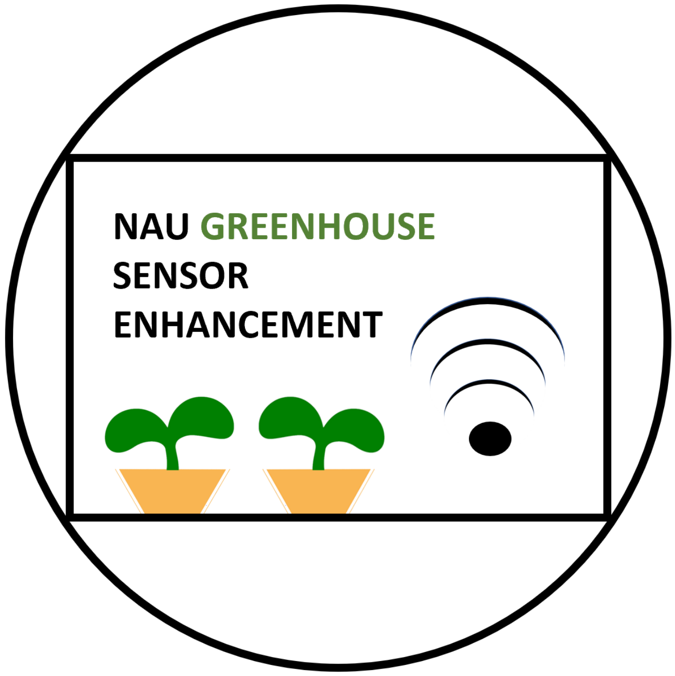
2021-2022 Capstone in Electrical Engineering: NAU Teaching Greenhouse



FINAL REPORT

April 29, 2022

Emilia Connelly - ecgconnelly@gmail.com

Ruopeng Jia - rj495@nau.edu

Jiaxin Liu - jl3479@nau.edu

Alexia Risley - acr324@nau.edu

NAU NORTHERN ARIZONA UNIVERSITY

Introduction

Our client is Doctor Tina Ayers. She is a professor within the Department of Biological Sciences at Northern Arizona University (NAU), which is a part of the College of the Environment, Forestry, and Natural Science. She is the head of the NAU Botany Club, which requires her and her techs to maintain the NAU Biology Teaching Greenhouse. As of the Fall 2021 Semester, she has requested that the greenhouse be equipped with temperature, humidity, and soil moisture sensors that would relay the data on an accessible platform that can be seen from the internet. The sensors were to be mobile so she may place them at any location throughout the greenhouse to gain data from whatever location that she deems necessary.

First, our team was to determine which sensors are ideal for the conditions within the greenhouse. The temperature can range from 40 degrees fahrenheit within the South house where it is cooler to a 120 degrees fahrenheit where the temperature is hotter. We needed temperature sensors that operate within this range. We also needed humidity sensors that can gather data from zero to 100 percent, since the humidity can vary dramatically depending on the day if it is humid or rainy outside, such as during monsoon season here in Flagstaff. The soil sensor must also operate between zero to a 100 percent for soil moisture content for when the plant is dry or has been watered entirely.

Second, our team needed to access what microcontroller to use to connect to our sensors. We needed a sensor that has wifi-capability, is cheap, and small enough so it may easily fit within the case of our choice. We had chosen the ESP8266, since it fitted all of these requirements. Next, we were to connect all of ESP8266's which are connected to our sensors to an internet access point, which would be a router. Once we connect all the sensors from the internet to the router, the router sends over the collected data to our chosen microcontroller, a Raspberry Pi 4B Plus. The Raspberry Pi then accumulates all the data to be sent out to a server on Discord, so that our client receives notifications of the current temperature, humidity, and soil moisture content from the sensors. She will also receive an alert if the temperature and humidity exceed or fall below the values that she set in place. An example would be she sent the maximum temperature to be a hundred degrees in the North house, but one the sensors detect a hundred and five degreed; she will be alerted at this point. Either she or one her techs will come to the greenhouse to make alterations to the thermostat to adjust the temperature.

Third, our team had to create printable circuit boards (PCB) for both the temperature/humidity sensors and soil moisture sensors. If we did not make a PCB, the wires at some point would get pulled out by accident and the whole board would have to be repaired. By using a PCB, we eliminated this risk. We at first created the circuit on a breadboard to see if the circuit would work. Once we determined that the circuit would work, we transferred our design to the Computer Aided Design tool. KiCAD was our choice to make the circuit boards. We recreated the circuit and then printed the circuit out on our PCB mil within the engineering building.

Design Process

We at first had to create a matrix of our skills to determine what each member of the team could contribute to the team. Each one of us had different attributes that would contribute to the team's success. We also had to establish who would fulfill each of the roles. We sat down and listed each role to figure out who wanted to take the role. As the project progressed, we assisted each other to help with each other's parts.

For our design, we had to select the ideal microcontroller to collect data from our sensors. As previously mentioned, we needed a microcontroller that was small, affordable, and had wifi-capability. We chose the ESP8266. We then needed to send this data over to a router with the internet. The router would then connect to our chosen microcontroller which is Raspberry Pi 4B Plus. The Raspberry Pi connects to our Discord server to show all the collected data by the Raspberry Pi.

We had to make printable circuit boards(PCB) to ensure that no wires would get pulled, that is if we were a breadboard. Creating the PCBs, required using KiCAD which is a computer aided design tool that allows the user to make circuit boards. After several revisions, we had the circuit boards up to par and then we printed the rest.

This previous description is the overall process description. A review of each of the major subsystems will be explained. For the soil moisture sensors, they at first had to be calibrated with some code. We had to place them in a cup full of water to observe the value of where it was 100 percent humidity, and then place it in the driest area possible which is the air to observe it at zero percent humidity. Then we took the numerical values presented and adjusted them with code to be at the range from 0 to 100. We then connected a tri-color LED to display as either dry(red), wet(green), or very wet(blue) for each of the values. These values range from zero to 33, 34 to 66, 67 to 100. This step for the tri-color LED required creating three if-loops and setting the values as either high or low for the color that is to be shown. Next, we had to set-up the connection with the arduino IDE with the ESP8266. This requires installing a library to make the arduino IDE talk to the ESP8266 and then installing a driver. There are more minute steps than this, but this explanation provides a basic summary of the process which took up to several hours to complete. Next, we had to make the ESP8266 have wifi capability, which required more code and installing a library. This step too took some time.

The ESP8266s are scattered in the greenhouse. We need to gather the sensor data together to do further processing. We need a data pool. So, we have a Raspberry Pi located in the greenhouse equipment room, it is helpful. For communication, we built a NodeJS server by writing JavaScript, which is running well on the Raspberry Pi. ESP8266s send an HTTP post request to a URL on the Raspberry Pi in an interval. Once the Raspberry Pi listens to the request, it will take the post information and then insert the environment data into the Maria database. Then, the scattered data would be gathered into the Maria database neatly. It is easy for us to do further processing like controls or sending alerts.

A very important part of the project is to establish the connection between the control system (raspberry pie) and the greenhouse equipment. Our partner Dataforth plays a very important role in this part and provides us with a very practical and efficient idea. Dataforth is an enterprise focusing on data acquisition and device communication. Coincidentally, many of their devices fit perfectly with our project. MAQ20 is a very reliable communication module developed by them. The MAQ20 family consists of DIN rail mounted, programmable, multi-channel, industrially rugged signal conditioning input and output modules and communications modules. Our project team selected COM4 and dioh modules to complete the communication content of this part of the project.

First, we made a detailed evaluation of the appearance and function of the equipment. We defined the assembly sequence, logic and physical location of the module, and analyzed the internal environment of the greenhouse. Greenhouse is also a very complex structure. After adding this new module, we have put forward a variety of different ideas of equipment separation. Considering the long distance between the greenhouse and the main control module, we decided to use wireless connection to control COM4. After measuring the vegetation distribution of the southern greenhouse in the field, we fixed the module position in the center. After the above items have been formed, we begin to learn how to use MAQ20 in depth. For two weeks, we continued to eliminate all kinds of problems in programming and connection. In the second week, we successfully realized the command of MAQ20 to turn on and off the temperature control fan. Before the end of the second week, we had a similar discussion on switching logic. We found that the previously designed greenhouse control system has deficiencies in continuous data processing. In addition to the rapid identification of temperature, humidity and other information, the greenhouse system should also be able to make corresponding decisions according to human coherent thinking. This includes how to prevent the system from continuously issuing different commands in a short time due to environmental noise or temperature fluctuation. In addition to the execution code of raspberry pie, the team designed a program that can execute the control code at fixed intervals. Using this program, the logic of control is raised to another level.

Final Design

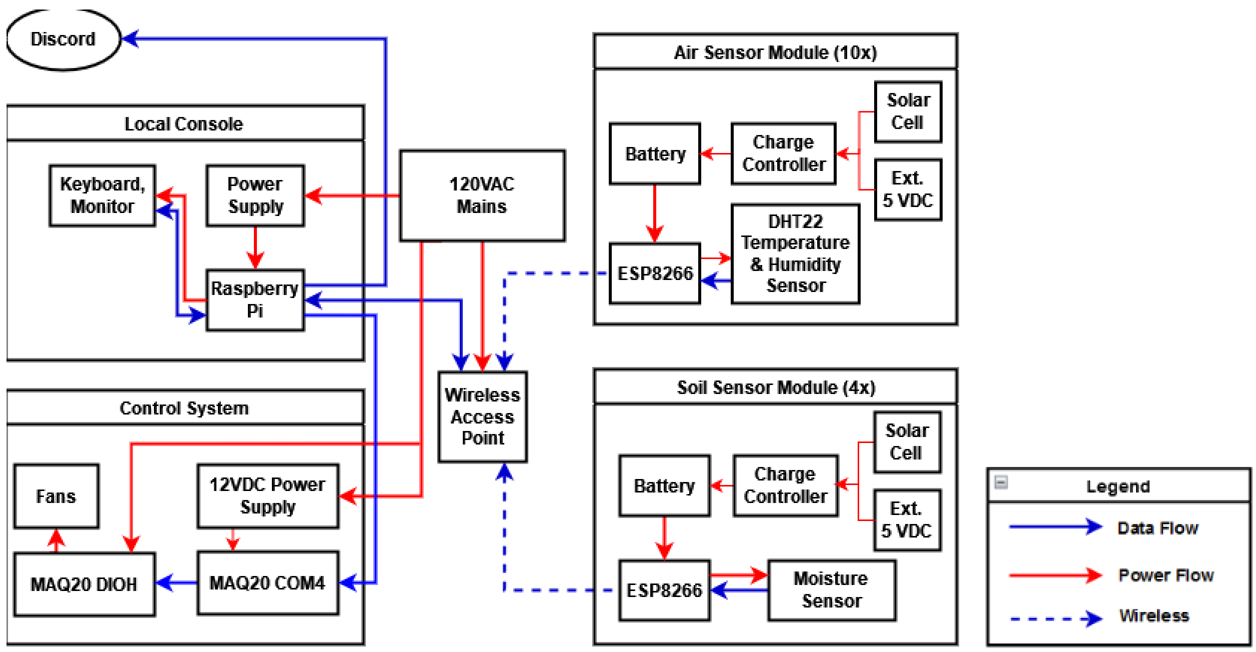


Figure 1: Overall design of the system

The air and soil sensors are connected to the wireless access point, which is our router. From the access point, the router connects to the Raspberry Pi; the Pi then collects all of the sensors' data to then transfer the data over to our Discord server. The server regularly receives data every 5 minutes about the temperature and humidity conditions within the greenhouse.. Also, the mixing fans are wired to the MAQ20 DIOH. From the connection to the MAQ 20, the Raspberry Pi is able to control the mixing fans to turn them off or on depending on the conditions within the greenhouse.

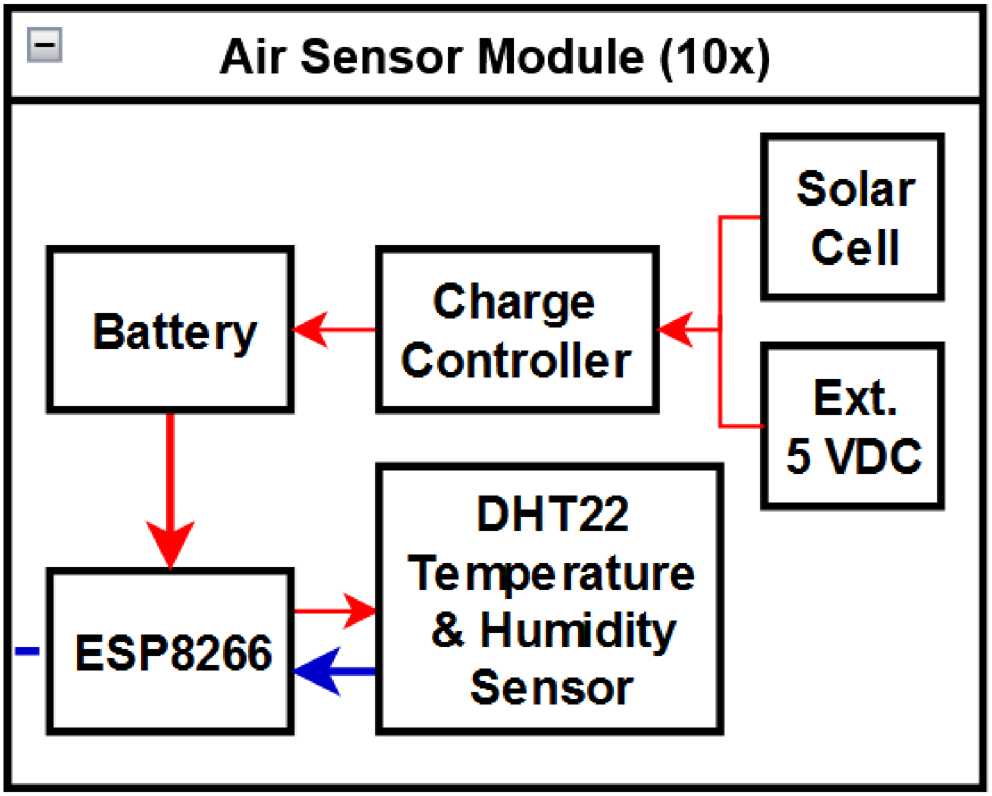


Figure 2: In-depth model of the air sensor

Initially in our design, the air sensor had a charge controller to mediate the power flow between the solar panel and 5V battery. Based on the conditions outside, the solar panel would serve as the main source, but can be compensated by the 5V battery if there was not a high enough irradiance value from the sun. However, now 8 out of 10 air sensors are wired directly to a power source within the greenhouse with the other two dependent on solar power. Besides how the air sensor receives power, the DHT22 (our temperature and humidity sensor) sends data collected every 5 minutes over to the ESP8266, which is our chosen microcontroller due to its reliability, affordable cost, and wifi capability.

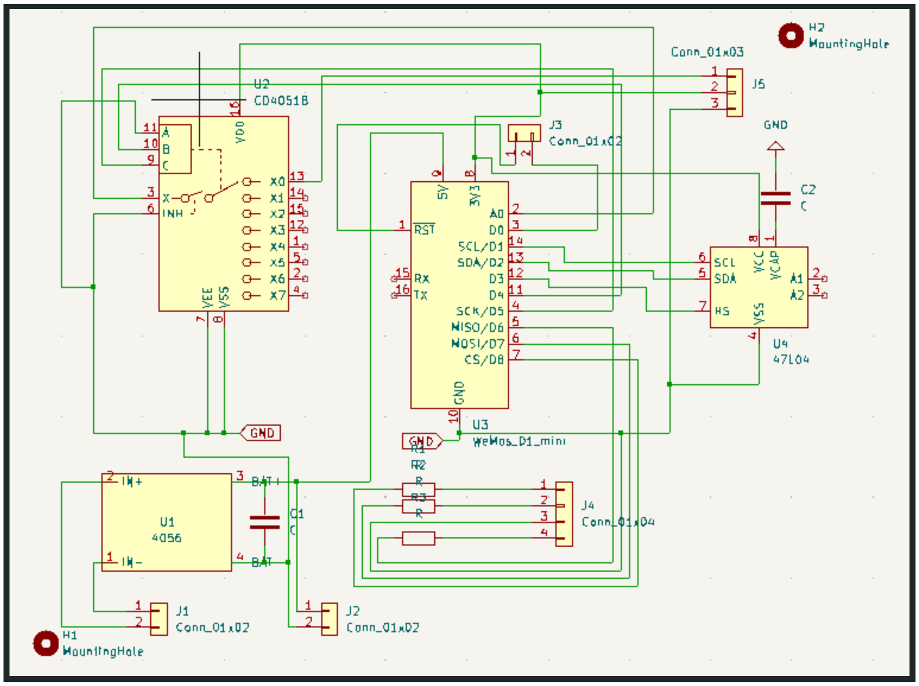


Figure 3: Soil sensor decomposition

The soil sensor also utilizes an ESP8266 to collect the soil moisture content value. The ESP8266 is connected to a charge controller, the TP4056 which mediates between the solar panel and 5V battery. The ESP8266 is also wired to a memory chip to store all the values over an extended period of time and is connected to an analog multiplexer. The multiplexer was added, since the ESP8266 only has one analog pin, so the multiplexer needed to have more analog inputs for our other components. Lastly, the ESP8266 is wired to a tri-color LED, with three resistors, which shows either red, green, or blue depending on the soil moisture value as dry, wet, or very wet.

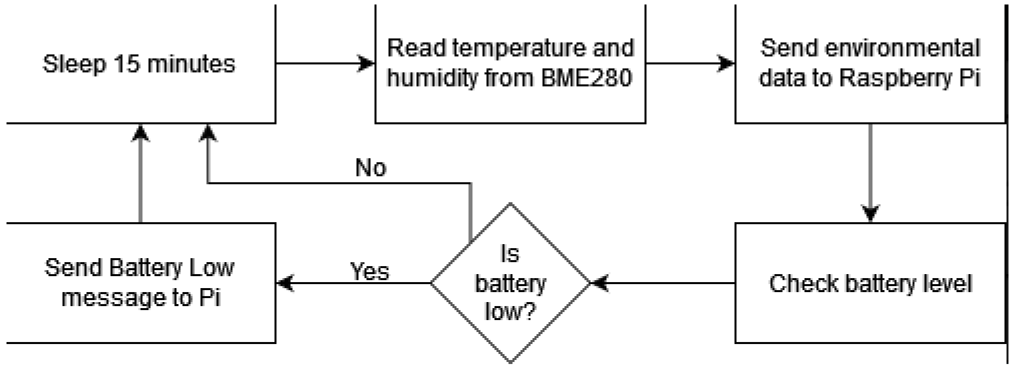


Figure 4: Sleep/wake cycle of BME280

In our initial design, we used a BME280 to record the temperature/humidity values, but this setup did not work, so we switched to the DHT22. However, the sleep/wake is still the same. Depending on the battery level, there is to be an alert sent if the battery is low. Also, when the DHT22 records a value, the ESP8266 and the DHT22 are to go into sleep mode to conserve power.

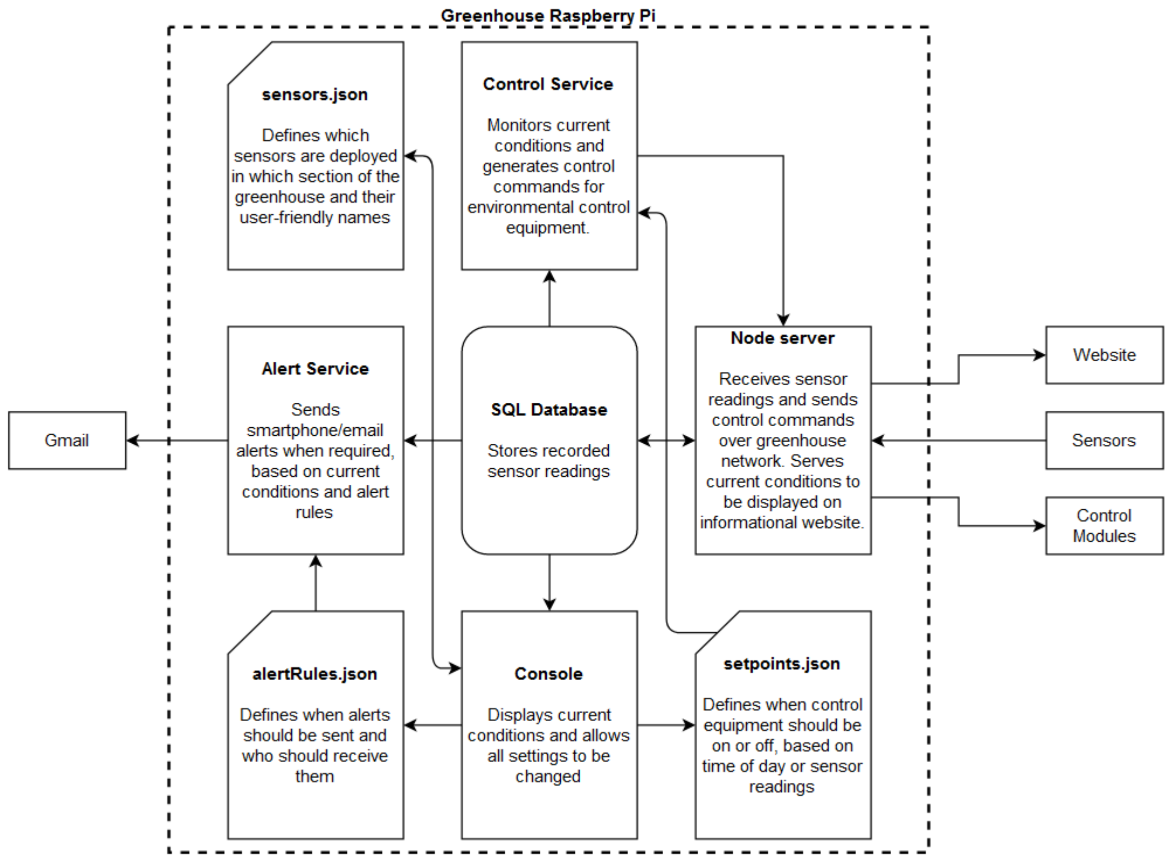


Figure 5: How the system sends over the collected values

The Raspberry Pi receives the temperature/humidity values and sends them over to a Discord server. However, initially in our design we had the Pi send the data over gmail. The Control Service monitors conditions within the greenhouse. If the greenhouse's temperature exceeds or falls below a set value, an alert is sent. AlertRules.json then sets the parameters to determine when an alert should be sent according to the Control Service's collected information. The Sensors.json defines the greenhouse's temperature/humidity sensors and gives them user-friendly names so the techs or our client is able to easily track each sensor according to its location in the greenhouse. All this collected information is then displayed on our website. Alerts are also displayed within the Discord Server, which require an acknowledgement to make the alert go away.

Results

5.1 Requirement Spreadsheet

TEACHING GREENHOUSE - MASTER TESTING LIST - UPDATED 1 APRIL 2022				
Type of Test	Status	Req #	Requirement	Remarks
Inspect		1.1.1	The north house shall be equipped with at least two temperature and humidity sensor modules.	The sensors exist and have been functionally tested, but have not yet been delivered to the customer.
Inspect		1.1.2	The south house shall be equipped with at least eight temperature and humidity sensor modules.	
UTM		1.1.3	Temperature sensors shall be accurate to +/- 2°F or better.	
UTM		1.1.4	Humidity sensors shall be accurate to +/- 5% or better.	
UTS		1.1.5	All sensors shall operate properly in temperatures from 40 to 120 °F.	Tests were passed but the tests did not cover the entire specified ranges.
UTS		1.1.6	All sensors shall operate properly from 0 to 100% humidity.	
Inspect		1.2.1	The greenhouse shall have at least 3 mobile sensors that are available to monitor the moisture content of the soil in any container.	
UTM		1.2.2	Soil moisture content readings shall be accurate to +/- 5% or better.	
UTS		1.2.3	Soil moisture sensors shall comply with the operating conditions specified for temperature and humidity sensors.	
UTS	*	2.1.1	Temperature and humidity data shall be retrieved from sensors and stored at intervals of 15 minutes or less.	
UTS		2.1.2	Environmental logs shall include, at a minimum, the temperature and humidity reported by each sensor and the time and date of the reading.	
Inspect		2.1.3	The system shall have adequate storage to retain logged data for at least two years.	
Integrate	*	3.1.1	A web interface shall numerically show temperatures and humidities from the latest reading.	
UTS		3.1.2	The web interface shall be accessible from off campus without use of the VPN.	
Inspect		3.2.1	The web interface shall be able to show graphs of recent temperature and humidity data, either individually per sensor, or using an average for each section of the greenhouse.	
~~UTS~~		~~3.2.2~~	~~The time range of the graphs shall be user selectable (allow the user to view data from the last 24 hours, 7 days, month, etc).~~	Client prefers all such graphs shown simultaneously.
UTS		3.2.3	The system shall provide a method for users to retrieve a CSV file containing all logged data, without requiring user knowledge of linux tools such as scp or sftp.	
UTS & Integrate	*	4.1.1	A smartphone alert will notify Tina Ayers or other greenhouse personnel when user-defined safe temperature or humidity ranges are exceeded.	
UTS		4.1.2	A method shall be provided for users to add/remove alert recipients.	
UTS		4.1.3	A method shall be provided to adjust temperature setpoints and any control conditions.	
(not testable)		4.1.4	The alert will be in a concise, numerical listing of data.	Ambiguous
UTS		4.1.5	The system will notify all alert recipients within 5 minutes of an unsafe condition being detected.	
UTS		4.2.1	All alert recipients will be notified within five minutes of a sensor failure being detected.	
Inspect		4.2.2	At least two extra sensor modules will be left behind to allow for user replacement of failed modules.	
Inspect		4.2.3	Documentation will be provided so that new modules can be constructed if necessary.	Documentation to be produced in the coming weeks
Inspect		5.1.1	At least two mixing fans shall be installed in the south house.	Fans on hand but not installed
Inspect		5.1.2	Mixing fans shall be rated for wet locations.	
Inspect		5.1.3	Mixing house fans shall not be battery-powered.	
UTS		5.1.4	The system shall be able to turn the fans on and off. At minimum, the following control modes will be provided:	
UTS		5.1.4.1	Always on	
UTS		5.1.4.2	Always off	
UTS		5.1.4.3	On during user-specified hours	
UTS & Integrate		5.1.4.4	On when excessive temperature differentials are detected within the greenhouse.	
~~Integrate~~		~~5.2.1.1~~	~~The system shall control the south house glycol heater to maintain a user-selected temperature in the south house.~~	Client no longer needs this functionality.
~~Inspect~~		~~5.2.1.2~~	~~The temperature selection method shall be calibrated in degrees Fahrenheit.~~	
~~Integrate~~		~~5.2.2.1~~	~~The system shall control the wet wall and south house exhaust fan to maintain a user-selected temperature in the south house.~~	
~~Inspect~~		~~5.2.2.2~~	~~The temperature selection method shall be calibrated in degrees Fahrenheit.~~	
Inspect		6.1.1	Equipment left in the greenhouse at the conclusion of the project shall not use solderless breadboards or other temporary circuit construction techniques.	
Inspect		6.2.1	All equipment shall adhere to relevant safety standards.	We are not qualified to test this.
Inspect		6.2.2	Equipment operating above 12V shall comply with the National Electrical Code.	

Important Test Results

Although all project requirements exist for a reason, some requirements are more pressing than others. These key requirements are: taking sensor readings no less often than every 15 minutes, providing remote display of sensor data, and sending smartphone notifications when unsafe temperatures are detected.

The latter two requirements are the primary motivation for the project. If notifications of unsafe temperatures were not provided, hundreds or thousands of dollars worth of plants would likely die in the event of freezing or excessively hot temperatures developing in the greenhouse. Without the continuous data display, the client could not trust that the system was functioning properly at any given point in time, and therefore the client would likely feel the need to physically travel to the greenhouse on particularly cold nights to check on conditions. Finally, the third critical requirement allows the first two to be effective. Greenhouse glass can break or a heater can fail at any time. Allowing more than 15 minutes to elapse between measurements presents an unacceptable risk of dangerous conditions going undetected for long enough to cause serious injury or death to the plants the system is meant to protect. Were this a commercial project, failing any one of the three critical requirements would likely cause the client to reject the entire project and refuse to pay.

When testing a product against a set of requirements, several types of test can be used. The test type is not arbitrary; rather, an appropriate test method is selected based on the nature of the requirement to be tested. For example, a requirement involving sequential logic would most likely be tested using the step-by-step method. In contrast, a requirement that a device operate properly over a range of inputs would be tested using the matrix method. Other options include an integration test, which checks that multiple subsystems interact properly, and an inspection test, which is used for simple requirements that can be tested without actually operating the system. On this project, all four types of test were performed, as discussed below.

The simplest test type, Inspection, was performed to verify the project's compliance with requirement 5.1.3 "Mixing fans shall not be battery powered." An inspection test was appropriate here, since the requirement can be fully tested simply by looking at the fans and observing that there is no place to insert a battery, and that the fans are equipped with standard Edison plugs for connecting to a 120 VAC supply.

The first two formal tests performed for this project were of the Matrix type. These tests verified compliance with requirements 1.1.3 "Temperature sensors shall be accurate to +/- 2℉ or better" and 1.1.4 "Humidity sensors shall be accurate to +/- 5% or better". Tests 1 and 2 were designed, performed, and reported by RJ and JL, respectively. The matrix test type was appropriate for these tests because the requirements specify that the sensors should be accurate to the specified tolerance *over their entire operating range*. In order to test compliance, the sensors had to be exposed to a range of temperature and humidity conditions. A step by step test was inappropriate because there was no reason to believe that the reported data would be affected by anything other than the ambient temperature and humidity.

The third formal test was conducted using the step by step method. This test verified the project's compliance with requirement 2.1.1: "Temperature and humidity data shall be retrieved from sensors and stored at intervals of 15 minutes or less." As discussed previously, compliance with this requirement is critical to the success of the project. This test was not designed to verify the system's response to any particular set of conditions; rather, it was designed to test the system's stability over a somewhat extended period of time. Customer acceptance testing may include an equivalent test performed over a longer period, such as the 72 hour burn-in period commonly specified in the construction field. The step by step method was appropriate because a guaranteed stable starting condition was required.

The last formal test was an integration test. This test verified compliance with requirement 4.1.1 "A smartphone alert will notify Tina Ayers or other greenhouse personnel when user-defined safe temperature or humidity ranges are exceeded." Because the test was intended to verify that multiple subsystems (sensors, Node server, database, and Discord bot) interact as intended, the test is automatically considered an integration test. Again, the requirement to notify greenhouse personnel of unsafe temperatures is critical to the success of the project and the team cannot afford to have this functionality fail during the customer acceptance test.

<u>Test 1: Temperature Accuracy</u>

This test was performed using the matrix method. The device under test was brought into a climate–controlled room with the room thermostat pre-set to 70 degrees Fahrenheit. The temperature readout from the device under test was compared to the indication from the commercial thermometer-hygrometer used for Test 2 (shown in Figure 2A). The thermostat setting was then increased by two degrees Fahrenheit, the room temperature was allowed to stabilize, and the temperature indications were compared again. The process was repeated up to a thermostat setting of 90 degrees Fahrenheit. At each step, the DUT and commercial thermometer indications were reported to match within the 2-degree requirement.

<u>Test 2: Humidity Accuracy</u>

This matrix test was performed by operating a team-produced air sensor in a small enclosed bathroom. At the beginning of the test, the room was dry. A humidifier was started, which slowly increased the humidity in the enclosed space. A commercial thermometer-hygrometer (Figure 2A) was used for comparison. Throughout the test, the commercial sensor was observed. When the commercial sensor indicated a designated test point (20 % relative humidity to 60 %RH in 5% steps), the next serial output from the team-developed air sensor (Figure 2B) was compared to the commercial reading. All test points were reported as meeting the 5 %RH accuracy requirement.

Figure 2A. Commercial thermometer-hygrometer for test comparison.

```
16:27:01.177 -> Attempting to connect to DHT22...
16:27:01.177 -> --------------------
16:27:01.177 -> 23.40%
16:27:01.177 -> --------------------
```

Figure 2B. Serial output from the device under test.

Test 3: Data Rate

This test confirms the system's stability over a period of several hours.

Per the test instructions, five air sensors were set up. For convenience, each of the five test sensors was set to be in the "north house" although the test physically took place in the engineering building. The ControlDesk application was used to confirm that data was being received from each of the test sensors, as shown below in Figure 3. Note that the "data age" readout for each of the "RATE TEST" sensors shows a recent report, easily complying with the 15-minute maximum reporting period requirement.

Throughout the testing process, the ControlDesk application was observed periodically. This was not a required part of the test. However, if a sensor had stopped working during the test, observing the readout would have likely alerted the tester to the failure and obviated the need to continue the test for the full three hours. Such a condition would have been indicated by the "Data age" readout for one or more sensors showing greater than 15 minutes.

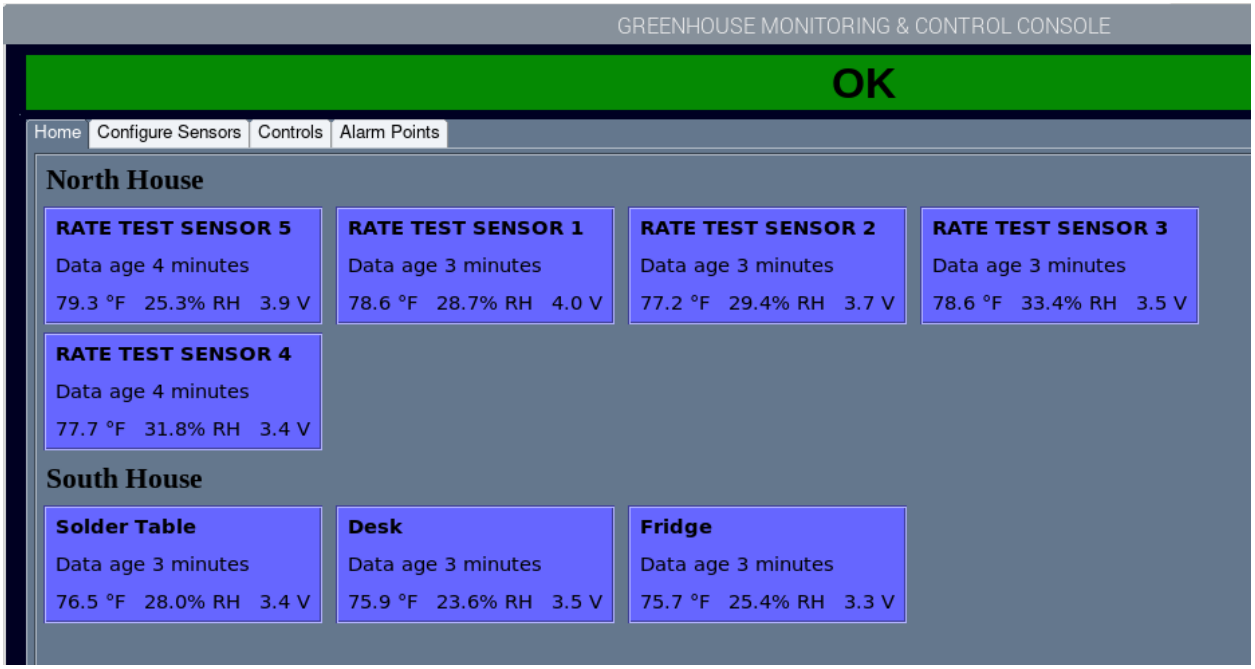


Figure 3. ControlDesk window during the data rate test.

5.3 Analysis of Results

All our test results show that several important parts of the project meet or greatly exceed the required design performance.

At the beginning of the design, we discussed several important components of the project. After dividing the whole project into several modules, we believe that the test in data collection, transmission and analysis is the most helpful to the whole project. Since we have tested the two fans ordered by the greenhouse and their control systems with simulated data at IEEE, we believe that there is no need to conduct officially recorded tests before installation. The MAQ20 system is a mature product verified by dataforce. After studying and installing it, we believe that the only factor that can affect its stability is the condition of the wireless network. For wireless network architecture, we have conducted several greenhouse experiments in advance. The coverage and stability of the network meet the initial requirements of the project, so there is no need to test again. On the contrary, data acquisition is not only a prerequisite for the operation of the system, but also has never been completely simulated and tested. In the previous data reading, we often only compared the temperature and humidity values in one or two cases. In order to make it work effectively in the greenhouse, we need to regulate the temperature in a wide range and record the performance of the sensor for a period of time. In addition, the data rate test not only comprehensively started our system, but also further repeated verification of the greenhouse internal network. Several tests will review our system in an all-round and non blind way to ensure that it will stably provide services to the greenhouse and biology teaching group in the next few years.

During the test, each of us strictly implemented every requirement in the test task form. In the temperature and humidity test, we found that the test results were basically consistent with the prediction. The performance of the sensor we selected at the beginning of the project (accuracy of 0.2 degrees Fahrenheit and 0.1% humidity) is much better than the required value, so the test results meet the experimental requirements. In the data rate test, the data transmission speed is consistent with the initial setting in IEEE and perfectly meets the requirements.

After performing relevant tests, we summarized the test results and completed the test report and display. Based on these works, we thought we could arrange to assemble and deliver the whole system to the customer.

Conclusion

Important Requirements and Results

(1) Req 1.2.1 The greenhouse shall have at least 3 mobile sensors that are available to monitor the moisture content of the soil in any container.
Result: The request was successfully completed. We planned and installed a total of 10 sensors in two greenhouses in the north and south.

(2) Req 3.2.1 The web interface shall be able to show graphs of recent temperature and humidity data, either individually per sensor, or using an average for each section of the greenhouse.
Result: The system can present all the above contents well. The results meet the expectations of this requirement.

(3) Req 4.2.2 At least two extra sensor modules will be left behind to allow for user replacement of failed modules.
Result: In addition to the sensors installed in the greenhouse, we have also reserved two as possible substitutes and supplements in the future. Customers can add new modules to the system or change the location of sensors at any time according to their needs.

(4) Req 4.2.3 Documentation will be provided so that new modules can be constructed if necessary.
Result: All of our underlying code is organized intact and saved in Raspberry Pi for subsequent maintenance and update. Customers can easily find them. Subsequent project groups can also quickly modify it.

(5) Req 5.1.1 At least two mixing fans shall be installed in the south house.
Result: According to the customer's requirements, we installed two mixing fans in the South greenhouse. They are controlled and powered by MAQ20 and can well execute the signals of the control module.

We installed a large number of humidity, temperature and soil moisture sensors in the NAU teaching greenhouse to monitor the vegetation in each area inside the greenhouse. Sensors can accurately record all data and label the data to be identified. After being monitored, the data will be recorded in the database immediately and orderly and retained indefinitely. Through the database, Our customer can check the past parameters at any time for reference. Then, the console, Raspberry Pi, will call the latest stored data in real time. The program in the controller will run logic statements to judge the greenhouse conditions in real time and upload execution instructions. Finally, the instructions are read by the MAQ20 module and used to turn the air mixing fans on or off as appropriate.

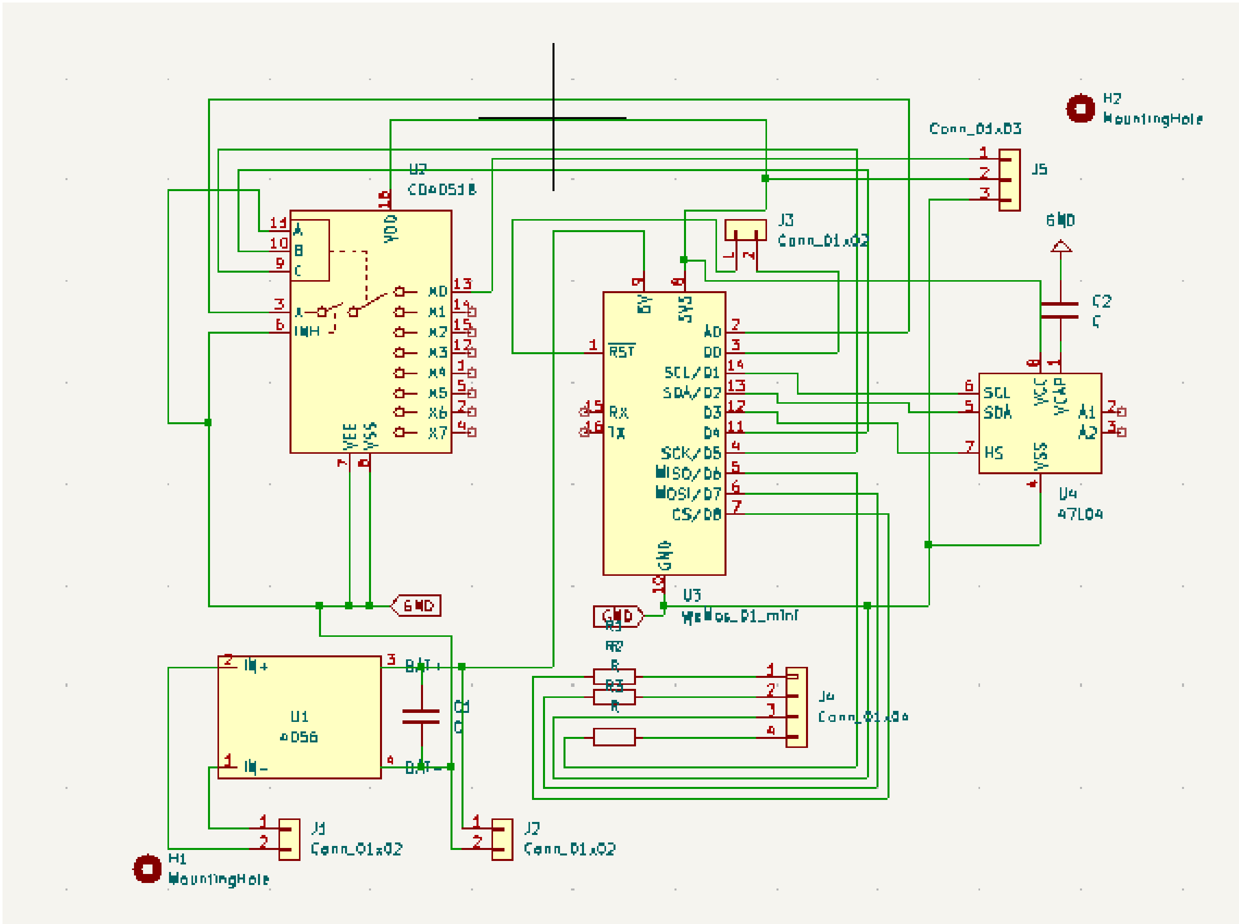
<u>Lessons Learned</u>

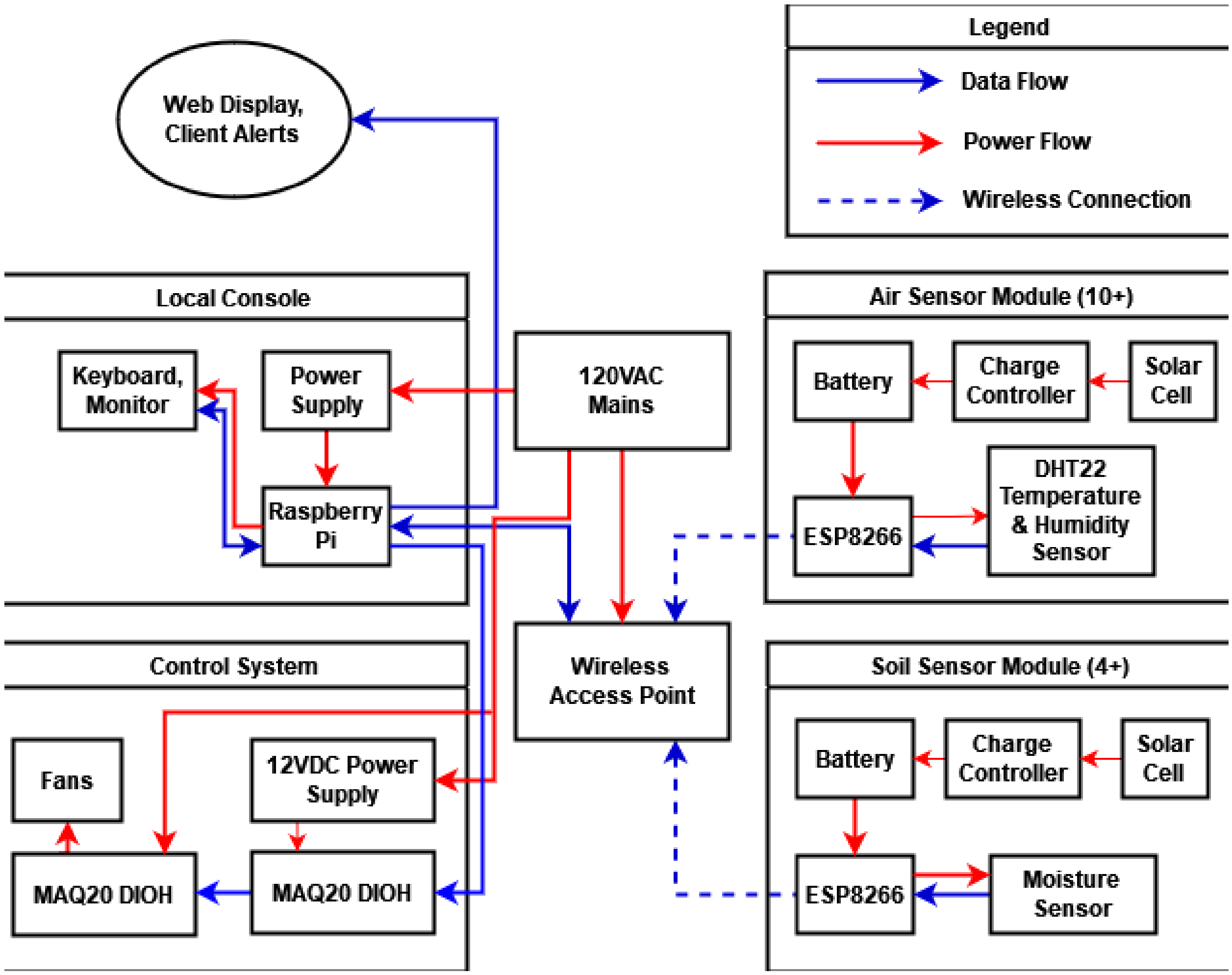
First of all, we learned how to plan the whole project reasonably. From last September to now, we have experienced many stages in the long process of the project. The whole project is not just a simple jigsaw puzzle. From determining requirements, assigning tasks, stage acceptance to final report, each part is closely connected with other parts. In the initial stage, we need to have clear team rules and medium-term goals, and have a clear mind to roughly allocate the goals that should be achieved in each stage. In the process of implementation, sharp judgment and decision-making are very key. We need to understand how the project is progressing and whether the previous planning is suitable for the current situation. When the project progress deviates from expectations, quickly judge whether the previous planning should be continued. In this regard, we have several successful experiences. When the customer had no intention of the soil sensor, we decided to give up this part and focus on the improvement of the control system. Finally, the function of the controller was further enhanced, and the customer was very satisfied with this change. We have also discussed the logic of the control system many times. With each discussion constantly improving the underlying documents, our logic module includes many functions that have not been considered in the initial stage of the project, which greatly improves its operability.

Secondly, we learned communication and cooperation within the group. In daily work, it is difficult for one person to complete a project. We often need to communicate and cooperate with many colleagues to complete the task perfectly, and communication is the main way of cooperation. In learning, we also need to communicate constantly. The main way to spread knowledge is language, and communication can help us learn from each other and acquire new knowledge and skills. Communication is also a very important way for relatives to communicate with each other. Only by communicating with their elders or younger generations can they enhance their feelings. In all periods of human beings, we need to use the way of communication to obtain friends. Only when we are good at communication can we make friends and have our own "circle". Human reproduction requires communication between both sides. Only communication can attract each other and reproduce. In this process, appropriate communication is very necessary. Human beings tend to "live in groups". We will have dinner and play together.

On some public occasions, it is very necessary to communicate with the people around us. In work and study, we will encounter many setbacks and pressures, which need friends to help solve and communicate with others before we can get material or spiritual help. No matter at any age, a person's strength is very limited. We often need the help of others. At this time, it is very important to communicate with others.

Finally, the project has also improved our engineering ability and literacy. Engineering is ubiquitous in life. No matter how big or small, we should face them with a rigorous attitude and the most outstanding treatment. In the project, we constantly brainstorm, adhere to the inherent requirements, and look for better possibilities. This provides a good atmosphere for us to constantly challenge our goals in life. We obtain resources from various channels and learn from each other, which enables us to improve beyond professional courses. With the dual wealth of knowledge and quality, we will successfully complete every engineering task in the future.


H2
MountingHole
Conn_01x03
J5
U2
CD4051B
J3
Conn_01x02
GND
C2
C
U4
47L04
U3
WeMos_D1_mini
GND
U1
4056
C1
C
J4
Conn_01x04
J1
Conn_01x02
J2
Conn_01x02
H1
MountingHole


Legend
Data Flow
Power Flow
Wireless Connection
Web Display,
Client Alerts
Local Console
Keyboard,
Monitor
Power
Supply
Raspberry
Pi
120VAC
Mains
Air Sensor Module (10+)
Battery
Charge
Controller
Solar
Cell
ESP8266
DHT22
Temperature
& Humidity
Sensor
Control System
Fans
12VDC Power
Supply
MAQ20 DIOH
MAQ20 DIOH
Wireless
Access Point
Soil Sensor Module (4+)
Battery
Charge
Controller
Solar
Cell
ESP8266
Moisture
Sensor