Boston University

Electrical & Computer Engineering

EC463 Greenhouse Prototype Test Report I

by

Team # 13

GreenHouse

Olivia Dorencz odorencz@bu.edu

Laura Morgan lamorgan@bu.edu

Hok Yin Shum hyshum@bu.edu

Kiahna Tucker kiahnat@bu.edu

Qian Zhang zhang1@bu.edu

Submitted: November 18, 2018

Table of Contents

Summarization of Equipments and Setup

Our prototype was intended to test the ability to accurately obtain measurements from our various sensors, confirm that we were able to communicate with our Arduino over Wifi, and display data that we had generated in a web application. We had split this functionality over three different Arduinos that will need to be integrated together for our final design. All the Arduinos were connected to the laptop computer with a USB 2.0 cable for power.

The first Arduino obtained readings from a TMP36 temperature sensor, a DHT11 temperature and humidity sensor, and a NSL19M51 photoresistor. Two temperature sensors were used as our final product will be able to measure temperature both inside and outside of the greenhouse. Also connected to the Arduino was an electronic relay that turns on a red LED when the temperature of the DHT11 sensor is above 80°F. For testing, this was triggered by applying a heat gun set to 100°F to the DHT11 sensor.

The second Arduino was connected to a Flying Fish MH Sensor Series module which was then attached to a QLOUNI soil moisture sensor. The connecting module is used propagate a 5V supply from the Arduino to the soil moisture sensor and export the readings from the sensor to the Arduino. For testing, the QLOUNI sensor was placed in pots of soil until the full length of the probes were covered.

The third Arduino powers the ESP8266, for passing all the data acquired from the sensors to web server and database. ESP8266 is a self contained SOC with integrated TCP/IP protocol stack that can give Arduino access to WiFi network. Connected and powered by the Arduino, ESP8266 can operate in three different modes: Wi-Fi station, Wi-Fi access point, and both at the same time. In the station mode, the ESP8266 could be connected to existing network. In the access point, other devices could be connected to the Wi-Fi generated by the ESP8266. Both modes could accomplish the data transfer between the web server and the Arduino. For the prototype testing, AP mode was used, and the code had been uploaded prior to the beginning of testing session. The Wi-Fi named “Greenhouse” was generated and connected successfully.

Description of Measurements

Our first step in testing consisted of verifying the measurements taken from the various sensors by displaying them on the Serial Monitor of the Arduino IDE and then applying conditions to change the measurements read from the sensor. The DHT11 sensor read an average value of the room temperature of 69.80°F with humidity of 11.0%. The TMP36 sensor read an average value of the room temperature of approximately 68°F. One of our team members brought an alarm clock that measures temperature to verify these readings, and it produced a reading of 20°C, or 68°F. The TMP36 sensor did have more variance in its readings. This is expected as the DHT11 sensor measures temperature with a thermistor whereas the TMP36 measures temperature with a transistor as the voltage between the base and the emitter chances proportionally with temperature. The TMP36 is more responsive to very slight alterations in temperature. In the senior design lab, the value read from the NSL19M51 photoresistor averaged around 415. We first applied the 100°F heat gun to the DHT11. This increased the temperature, and the red LED turned on when the temperature passed 80°F, with a reading at the time of 82.1°F. The red LED turned off when the temperature dropped below 80°F, though this was not immediate as the sensor had to cool down. When the photoresistor was covered to block light, the value read from it dropped to approximately 70. To verify that the humidity sensor was accurate, one of our group members breathed on the sensor, and we confirmed that the humidity reading increased to 12.0%.

The QLOUNI sensor uses capacitance to measure the dielectric permittivity of the surrounding medium. This characteristic becomes dependent upon water content when placed in soil. Absent of having data produced by a more sophisticated soil hygrometer for comparison, the sensor was calibrated by taking the average of measurements when the device was placed in dry air and submerged in fully moist soil. These values were then passed to a mapping function to convert the raw average electric potential exported by the sensor to a more user-friendly percentage. Three testing conditions were used to demonstrate the QLOUNI sensor's capability. First the sensor was placed in dry air, resulting in a reading of 0% soil moisture. Secondly, the sensor was placed in a pot with lightly watered soil, resulting in a water content reading between 11% and 12%. And finally, the sensor was placed in a pot with very moist soil, resulting in a reading between 98% and 100%.

For verifying the successful connection between the mobile device and the ESP8266, the code of presenting a counter on the browser is included in the setup code. After typing the IP address 192.168.4.1, the private IP address for use inside of a private network, in the browser, the counter is successfully presented on the browser page and increments every two seconds, which means the network connection is successful.

Conclusions based on Test Data

We successfully demonstrated that we could read accurate data from all of our sensors, activate an electrical device based on the sensor data, communicate with the Arduino over Wifi, and display data in a web application. One of the biggest challenges remaining ahead is integrating all of these different components together. Our test plan used three different Arduinos, and we hope to integrate the circuitry and software for these to a single Arduino for our final product. We will also have to think carefully about the algorithms we use to determine whether the plants are receiving too much or not enough heat, light, and water as these have largely not been implemented at this time.

As for the web application, a dashboard is created to demonstrate a set of local data saved on the server. The application will be able to demonstrate real-time data once the data pipeline between hardware and software is established. However, there’s an uncertainty as to whether the current AP mode implementation would work with the web application. It’s because AP makes the hardware a router and is sending html file stream to the client. It’s still uncertain how we would uses the Node.Js server as a client to connect to hardware. The professor has suggested the dashboard to display simple data and only display plot when the client requires extra data. We will take the professor’s advice when we implement the webapp.

Appendix

1. Sketch of the physical structure

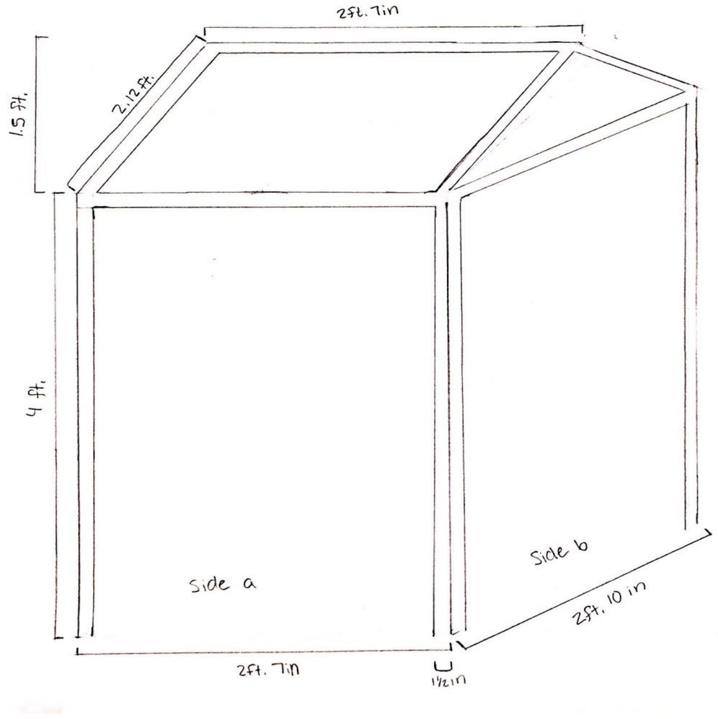

Figure 1: This design sketch is the three dimensional view of the PVC pipe frame of the greenhouse. The total height of the greenhouse frame will be 5 feet 6 inches tall. The PVC pipe width is 1.5 inches and will be joined together using pre-existing PVC pipe joints.

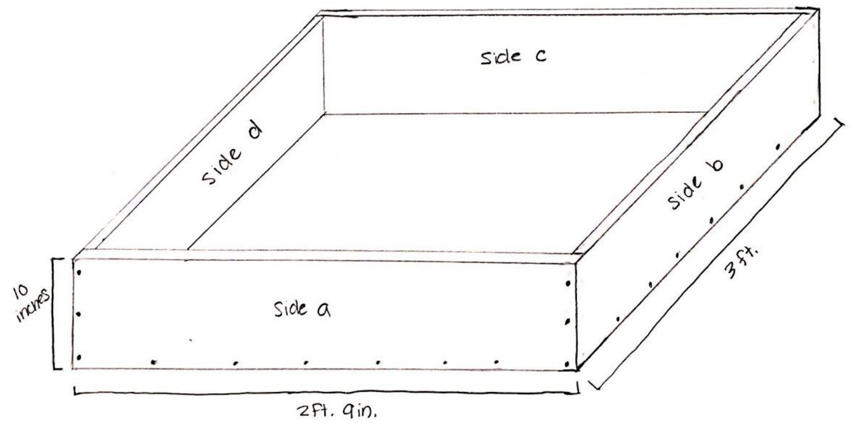

Figure 2: This design sketch is the three dimensional view of the box that is going to be the base of the greenhouse. The frame (depicted in Figure 1) will be mounted to this box. The box itself will be made using cedar lumber. The side boards will be 10 inches wide and 1 inch thick.