IoT Environmental Monitoring

Design Document

Team 45

Advisor/Client: Dr. Geiger

Team Members/Roles

Tyler Fritz - Software Developer - Web Server and Home Node

Ahmed Abuhjar - Hardware Developer - Leaf Node and Home Node

Dong Xing - Hardware Developer -HomeNode&LeafNode Role

Haoyue Ma - Hardware Developer - Leaf Node and Home Node

Yuanzhe Wang - Hardware Developer - Leaf Node and Home Node

sdmay19-45@iastate.edu
https://sdmay19-45.sd.ece.iastate.edu/

Revised: 12/01/2018

Table of Contents

List of figures/tables/symbols/definitions

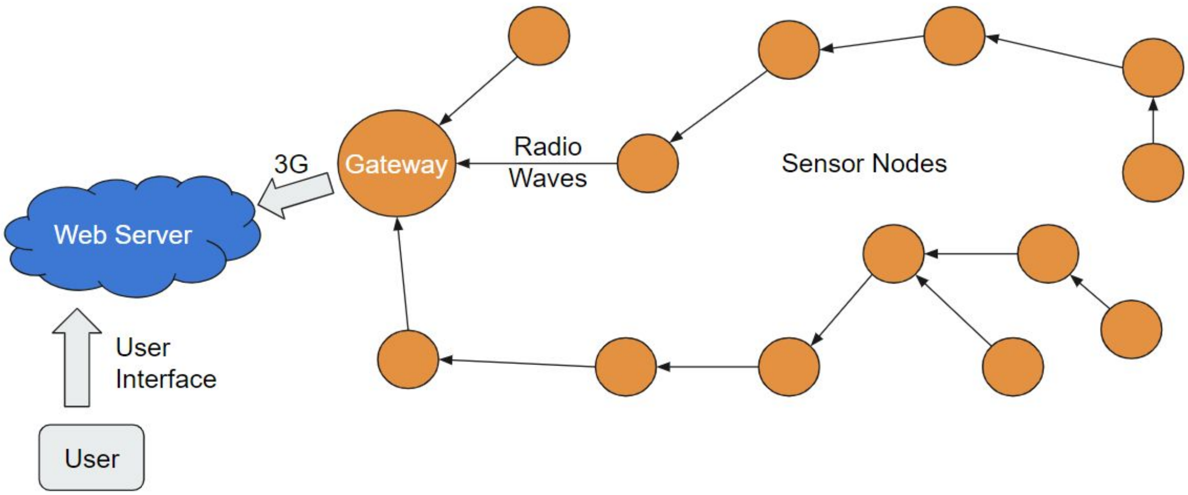

Figure 1 *Conceptual Sketch of the Proposed Design*

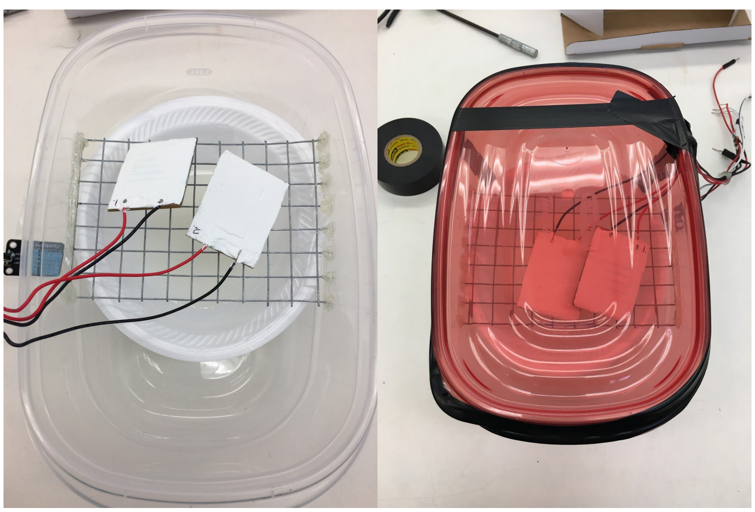

Figure 2 *Calibration test experiment*

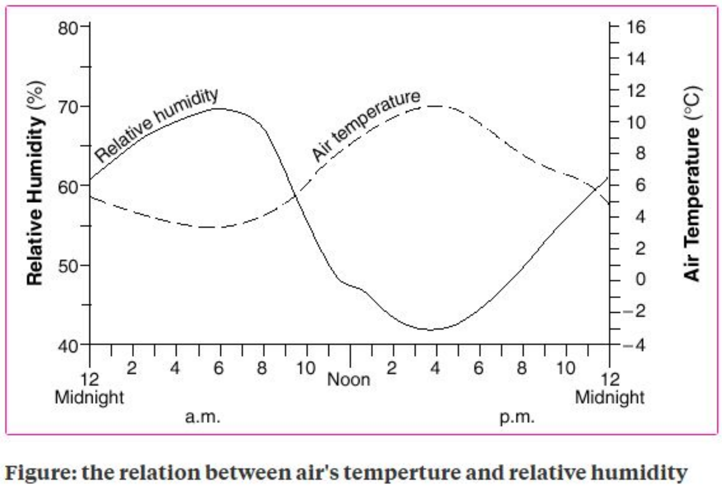

Figure 3 *The relation between air temperature and relative humidity*

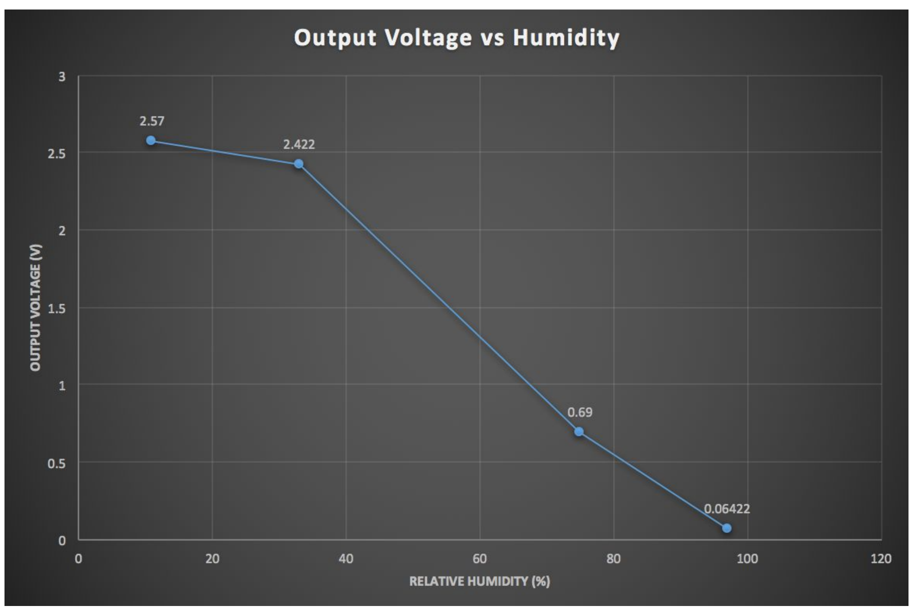

Figure 4 *Calibration test results*

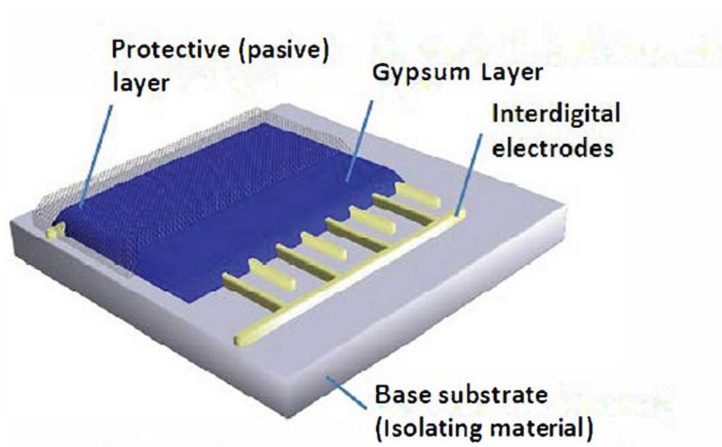

Figure 5 *protective layering strategy*

List of Definitions

Gateway Node / Home Node, which is the final node to receive all data from leaf nodes and then upload these data to the web-server.

Leaf Node / **Sensor Node**. Data will be tested and collected by sensors on each leaf node, and then the collected data will be transferred in-between leaf nodes and finally to the home node.

1 Introduction

1.1 Acknowledgement

Dr. Geiger has contributed a great amount of time and effort into this project. Without his contributions this project would not have been possible.

Tim Lindquist and his Senior Design group have also played a crucial role in our progress. Their knowledge and prior work have been critical to our progress.

1.2 Problem and Project Statement

General Problem Statement

The current market solutions for monitoring the environmental conditions of land are very expensive and it is only feasible to purchase a limited number of sensors. In most cases it is prohibitively expensive to purchase a system with a reasonably high level of data resolution. If a cheaper solution were available, better decisions could be made concerning the the use and maintenance of land.

General Solution Approach

We are going to make an IoT environmental monitoring system. It will consist of many sensor nodes that take measurements on the soil and send the results back to an internet gateway. The gateway will upload the received data to the internet where it can be viewed by the client. Our sensor nodes will be very inexpensive, at around $10 per node, which will make our solution cheaper, and have far better data quality, than the current solutions on the market.

1.3 Operational Environment

Our nodes will be evenly distributed over the land being monitored by our solution. This land will be primarily farmland in Iowa which will require our solution to account for rain, dust, mud, wind, snow, ice, intense sun, heat, cold, and hail. The sensor nodes should be able to survive in this climate for around 7 months, from early in Spring to late in Fall.

We will be designing our nodes to be able to handle this wide range variety of temperatures and events using a strong casing. Due to the timing of 491 and 492 we will only be able to test in the environmental conditions of Spring in Iowa.

1.4 Intended Users and Uses

The system is primarily intended for farmers, but it will also be useful for anyone doing other types of environmental monitoring. It will be easy to configure our nodes to capture any type of data a client might want. We are assuming that our users will have a low technical understanding which requires the setup for our system to be simple.

The product is intended to be used for collecting soil moisture and temperature levels over large areas of farmland. It could also be useful for collecting different types of data in different conditions however.

1.5 Assumptions and Limitations

Assumptions

- Sensor nodes will have a density of two nodes per square km.
- Sensor nodes will only be used for a span of 6-8 months.
- The radio transmitter from the sensor nodes will have a range greater than 1 km.
- There will be cellular reception where the gateway is placed.
- Our web server will not be required to serve very many people at a time.

Limitations

- The sensor nodes will be battery powered.
- Most sensor nodes will not be able to communicate directly with the gateway due to the range of their radios.
- It is not reasonable to expect that someone will replace the batteries in the sensor nodes when they die.
- Radio transmitters with a sizeable range are expensive and require lots of power.
- The price of each sensor node must be very low.
- We do not yet have a case to house our sensor nodes and no one on our team has experience with case design.

1.6 Expected End Product and Deliverables

By the end of December 2018 we will have a program for the sensor nodes that handles transmitting, receiving, and routing data through the network. Additionally this program will handle the sleep and wake cycles, and the recording of sensor data. This program will have been run and tested on Arduino Unos.

By the end of January 2019 we will have the sensor node program correctly functioning on the custom PCB made for these nodes.

By the end of March 2019 we will create and run a large scale test of our system. The results of these tests will be compiled into a detailed testing report.

By the end of January 2018 we will have a gateway that communicates with both the sensor nodes and the web server. It will be the only node in the mesh network that can communicate with the web server. Its function is to receive sensor data from the sensor nodes and upload the data to the web server.

By the end of January 2018 we will have a web server that receives data from the gateway and displays it to the users. This will require setting up the previous sensior design group's environment for this application on a public facing VM provided by the university. A protocol will be established for communication between the gateway and this web application.

By the end of February 2019 we will have gypsum based humidity sensors functioning accurately and with high levels of data resolution.

By the end of May 2019 we will have a solution for how to distribute the sensor nodes in a way that the web application knows their geographical location.

2. Specifications and Analysis

2.1 Proposed Design

Overview

A high level diagram of our solution can be seen by looking at *Figure 1* below. There are three main components in our solution which are the sensor nodes, the gateway node, and the web server. The sensor nodes will be recording environmental data which is then relayed through other sensor nodes back to the gateway, where the data is sent to its final destination, the web server. Clients will be able to see their data by accessing the web server through an intuitive UI.

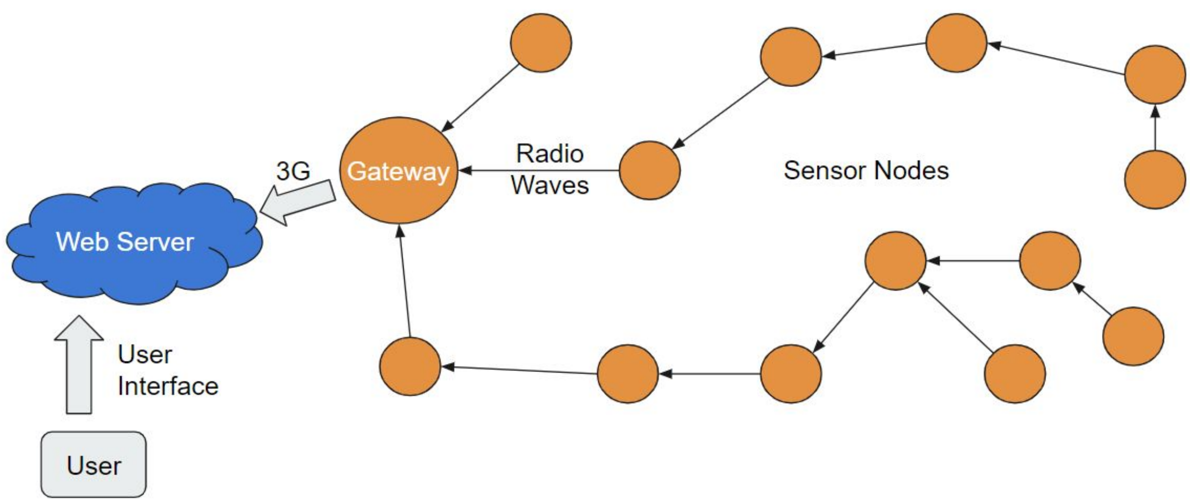

Figure 1. Conceptual sketch of mesh network

Web Server

The web server is written in a PHP framework called laravel. The backend is a Spring based API and a MySQL database. This application was written by the previous senior design group and will require some, but not an extensive amount, of additional work. The API is called using HTTPS requests, which is how the home node will be able to upload data.

One design alternative that we considered was creating a desktop application instead of using a web service. The data recorded by the mesh network would be stored in a local database and would be viewed by physically accessing the home node. We decided that this was a poor solution because the clients would not be able to access the data from anywhere and also increased the chance that the data would be lost in a disaster.

Leaf Nodes

The leaf nodes are comprised of a microcontroller, two sensors, a radio transmitter, and a pcb. The pcb was custom made for our application by a previous senior design group. The leaf nodes will record analog data with their sensors, put the data into a packet and transmit the packet to the home node by relaying it through the mesh network. The protocol used to route packets through the network will be one that is designed specifically for our application. Our protocol finds the shortest path for each node to get back to the gateway, and has each node use that path. These paths are discovered using an algorithm similar to Dijkstra's Shortest Path algorithm. Each node initially sets its distance to the gateway to infinity, and then periodically asks its neighbors what their distance to the gateway is. If a neighbor has a distance that is less than the current node's distance, the current node decides to send its messages to that node and updates its stored distance to be the neighbor's distance + 1. Then, when the node wants to send a message to the gateway, it simply needs to send its packet to that neighbor.

We need to be careful about how many times each node asks neighbors for their distance to the gateway because making these broadcasts uses some of the battery's energy. To account for this issue, each node only broadcasts these messages at the beginning of each wake cycle, or every ten seconds while the node does not know how to get to the gateway. This strategy strikes a balance between making the power usage of the network lower after it has finished setting up, while new paths are unlikely to be discovered, and ensuring that the network establishes itself quickly.

Each sensor node in our system will be powered solely by batteries, which poses a challenge because the nodes will have a limited amount of power they can use. Our approach to reduce power consumption will be through several techniques including having the nodes sleep when not in use, using an efficient routing protocol, and adjusting the range at which their radios communicate.

The sensor nodes will only wake to collect and transmit data within certain periods of time. When asleep, the power consumption of each node is reduced almost to zero. We plan for the nodes to only be awake for around five minutes per day, which means that this strategy saves around 99.6% of the power that keeping them always on would use.

To wake up at the same time, nodes will use an external crystal as their timer because the internal clock of the microcontroller we are using has an accuracy of only 90%. We still have an error issue with the crystal however because the crystal we plan to buy has an error of 20 parts per million. This means that over the course of a month each microcontroller would become out of sync by a minute. This would be a problem because the sensor nodes would not wake at the same time and the network would be able to transmit data. To get the leaf nodes back in sync, we will have the gateway broadcast a signal that when received, each sensor node will reset its clock. This event will occur daily ensuring that the sensor nodes will always be in sync in the order of a few seconds.

Depending on application, the number of sensor nodes or leaf nodes will differ from one system to another. The range between the sensors will pose a challenge as the radio signal will consume more power when transmitting data longer ranges. However, the longer the distance between the nodes the less number of sensors will be utilized in our system; hence reducing the cost of the system. A trade off is involved in deciding whether we will place the sensor nodes further apart or closer to each other. Once we verify that the system is functioning properly in receiving and transmitting data by the end of this semester, we will run several tests during the next semester to determine what range would be the best to utilize for the system.

Home Node

The home node has two primary functions, sending data it receives from the mesh network to the web server, and syncing the clocks of the leaf nodes. To make it so that the home node is able to send data to the web server it will have a cellular data connection which will allow it to send HTTPS requests. The web server will have an API which is what the home node will use to upload data. To enable the home node to be able to sync the clocks, it will have a radio that is powerful enough to contact every leaf node.

The home node will require a radio with the same range as the sensor nodes in addition to its longer range radio. This is due to the fact that during the route discovery phase, the home node must respond that it is distance 0 from the home node and only nearby nodes must be able to hear this message. In addition to that, data packets are not passively received by nodes. Before a sending nodes marks a packet as received, it must receive a confirmation that the receiving node got it. So the home node needs to be able to send these confirmation messages to any nodes that want to communicate with it.

2.2 Design Analysis

We have not started implementing our design so far. That is because we are continuing the work of a previous senior design group that had already designed the solution and implemented a great deal of it. We have spent the semester so far simply getting the project set up the way the previous group had it. The design decisions we get to make are based on the protocol that the nodes use to route messages. We only recently configured the nodes to be able to send messages so we are only now able to start implementing our design.

One strength of our design is its low cost. By distributing the network and hopping messages between nodes, we are able to use low power radio transmitters which are cheaper. Each of our nodes will end up costing around $10 and our base node will cost around $150. If our system is employed to monitor 50 km2 of land, the total cost will be $650. This is significantly cheaper than our competition, which costs $3,000 per node, and provides data at a much better resolution.

The overall scope of the system was discussed by all the team members and determined to be reasonable given the project deadline. After examining the requirements with the highest priority, we determined that the majority of the work will be with the gateway and the web application. Overall, after analyzing the requirements and prioritizing them based off importance we have evaluated that the project is within scope and that probability for our team to finish the project before the deadline is high.

We have examined where our technical strengths and weaknesses were for this project. Most of our team members have had experience with the majority of the individual components that the project should consist. The primary hardware components are the microcontroller and soil moisture sensors. We have examined each piece of the hardware to determine how they will interact and function with each other. The majority of our team members are familiar with the microcontroller (Arduino). This is an advantage for our team as the microcontroller is a fundamental part for the communication between the various parts of the system.

3 Testing and Implementation

3.1 Interface Specifications

Humidity sensors need to be able to detect and measure the humidity in a wide range, roughly between 20% and 90%, and that's sufficient to indicate the water requirement for the soil. Calibrations needs to be done on the sensors to determine the lower and the upper level of humidity the sensor can detect accurately.

The sensor nodes should be able to communicate with each other by radio signals. The sensor node should be able to interface with two other sensor nodes by using an Arduino microcontroller and an Atmega328P chip on a PCB which means if one of these two sensor nodes is broken or out of the range, the sensor node can still send the data to the other one. The gateway will send the data to the web server by 3G connection where the users can see the application and the data from any web browsers.

3.2 Hardware and software

Hardware:

- Arduinos - Microcontroller that is easy to program and connect with external devices.
- Radio Transmitter Receiver - specific hardware still undetermined must have at least 1km range.
- Hygroscopic Moisture sensor - Will be a gypsum based sensor because gypsum is cheap and hydroscopic.
- Adafruit Cellular Link - Connects the gateway to the internet through a cellular connection.

Software:

- NRF24 Arduino Library - Manages low level communication over radios
- Laravel - Simplifies web development
- MySQL - Stores our data safely with fast retrieval times
- PHP - Allows for custom web pages
- Java - Creates backend API
- Arduino IDE - Allows for easily programming our Arduinos.

3.3 Functional Testing

Several tests needs to be conducted in our project, and theses includes the following:

1. Calibration of humidity sensor.
2. Test routing protocol by placing many sensor nodes in a field and recording how many nodes are able to accurately get their data to the gateway.
3. Test longevity of sensor nodes by doing a full scale test for a month and measuring the amount of energy left in the batteries by the end.
4. Determine the need of Polystyrene to protect the Gypsum and to stabilize the sensor.
5. Preliminary test for the accuracy of signal transmitted. (One-to-one)

3.4 Non-Functional Testing

Data tables which are collected and transmitted from nodes and CAD annotated drawings of the hydroscopic probe will be submitted.
According to different types of data, like temperature under different time periods or humidities in different fields, these data are all going to to be finally created to drawings and reports.
In order to make sure the documentation security and usability, the system requires pin numbers and keeps checking and reporting the results to the user.

3.5 Process

Sensor Calibration: One test plan is to calibrate the humidity sensor to determine the lower and upper level of the output measurement from the sensor. This can be done by placing the sensor in different atmosphere with different RH level.

We can use two methods to control relative humidity. One is to utilize air/water vapor flow. The second method involves placing a reservoir with saturated salt solution in the chamber, which gives discrete number of values of the RH, depending on the kind of salt used. Refer to the **Table 1** for details. (i.e. NaCl for 75% RH and K2SO4 for 97% RH). ("Equilibrium Relative Humidity Saturated Salt Solutions")[7]

Table 1. Salt solutions and expected relative humidity in the Vaisala´s calibrator

Salt	Water content	Expected relative humidity
15g LiCl	12 ml	11%
30g MgCl2	3 ml	33%
20g NaCl	10 ml	75%
30g K2SO4	10 ml	97%

The later method is what we are using for calibrating the sensors, and some constraints had to be put into consideration for this test.

- Uniform temperature environment
- Two sensors per trial to insure consistency with the measurements.
- Consistent container dimensions to ensure the consistency in measurement.
- To place the sensors in the containers without affecting the internal environment, every hole or opening for passing the wiring has to be sealed. In this case tape was used to seal the gap between the cap and the container. As shown in **Figure 2**

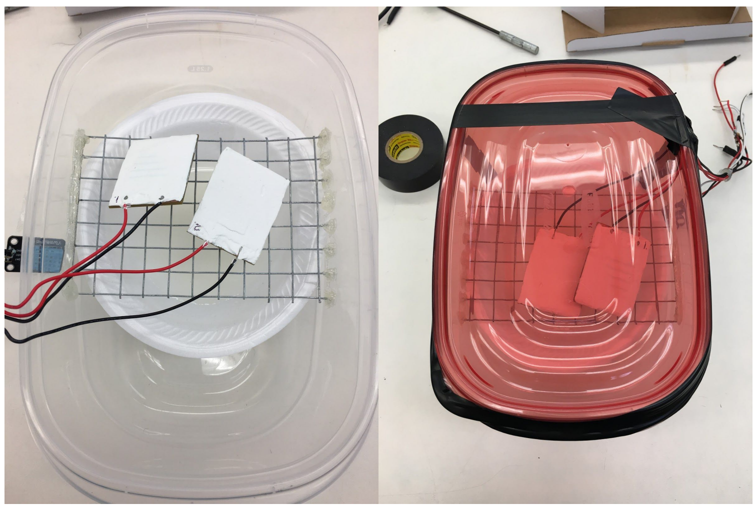

Figure 2 *Calibration test experiment*

Both methods require a uniform temperature environment. As already known, and illustrated in the graph **Figure 3**, a small temperature fluctuation or a temperature gradient could easily result in ±1% to ±2% error in RH. Thus, when testing, it's necessary to control the ambient temperature in order for us to have accurate measure of the RH, which is very difficult to achieve with a small tolerance, and that could be because the change of gypsum impedance is abrupt.

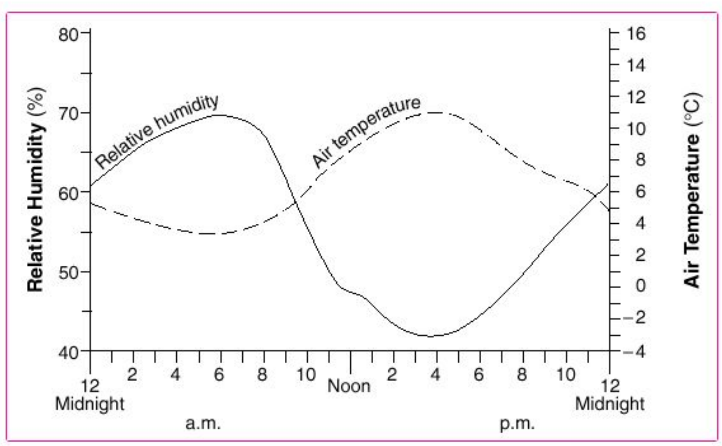

Figure: the relation between air's temperture and relative humidity

Figure 3 *The relation between air temperature and relative humidity*

After theses testing, we can collect the data and draw the trend lines. We can observe whether the trend is generally satisfactory and whether any unexpected results appeared.

The preliminary test has shown a satisfactory results as shown in **Figure 4.**

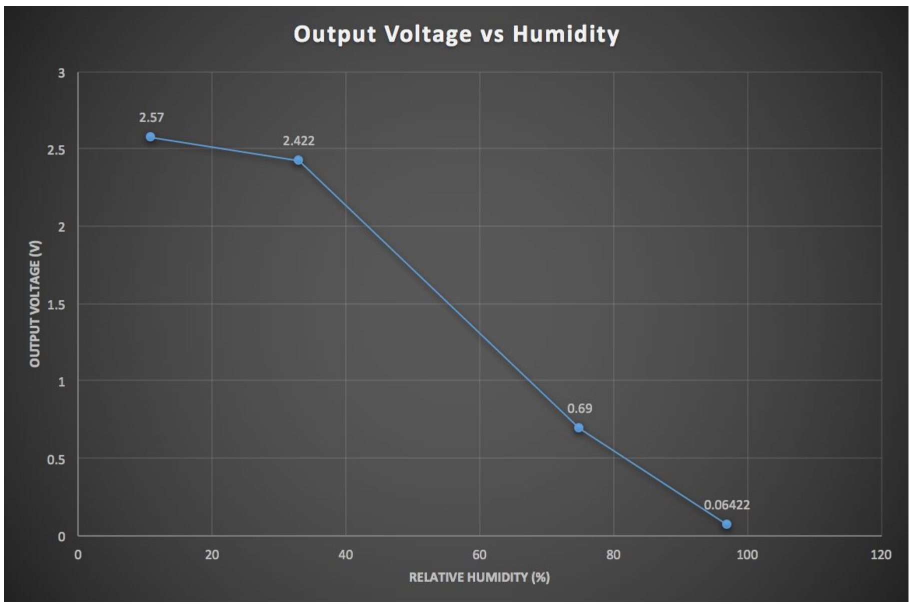

Figure 4 *Calibration test results*

As can be seen from this figure, the voltage output tends to deviate from the linear curve in the region less than roughly 33% relative humidity level. The mean error, averaged over all these four samples, was found to be relatively low (of the order of 10%). This is sufficient to accurately indicate the water requirement of soil and can be taken as insignificant error value.

Nodes Testing

Our hygroscopic sensor prototype is going to be tested on this step which has the purpose of checking if the sensors could test accurate data, the transceivers work well or some unexpected environmental issues may influence the nodes.

In order to get the accuracy percent, we have to starting testing in different water content type of soils which have already known about their moisture situation, and after the tested data come out, we can just compare the result with the known conditions.
By checking and reading the data from different nodes can directly know if the transceivers work well or not.

Under different weathers, like raining, snowing or sunshine, out team wants to keep testing the data and compare the accuracy for these dates to make sure if there is any issues to influence the test process.

Polystyrene cover layer test:

A direct connection between the gypsum used in the sensor and water can cause the electrode to take a long time to respond to a change of humidity level from high to low and hence longer time to stabilize. It can also cause some distortion to the gypsum structure. Therefore it is necessary to be stabilized by a material, which is porous for water vapour and at the same time prevents penetration of water into the gypsum.

The response time of the adsorption can take about 2 minutes, but it could potentially take long time during the desorption process. This is because there direct contact between the gypsum and the moist soil. We don’t want the sensor to suffer from long response times, and so in case this situation happened in our testing, we came with an idea to stabilize it using a protective layer at the top of the gypsum. This layer or protective cavity is porous and is made from Polystyrene **Figure 5**.

When the water vapour is trying to move from the soil through the polystyrene to the gypsum layer, the gypsum can sense the humidity but not necessarily be in touch with water. Cold surfaces is what can cause condensation, and studies show that polystyrene can less likely cause condensation and that’s why we see lots of houses nowadays use polystyrene in the walls.

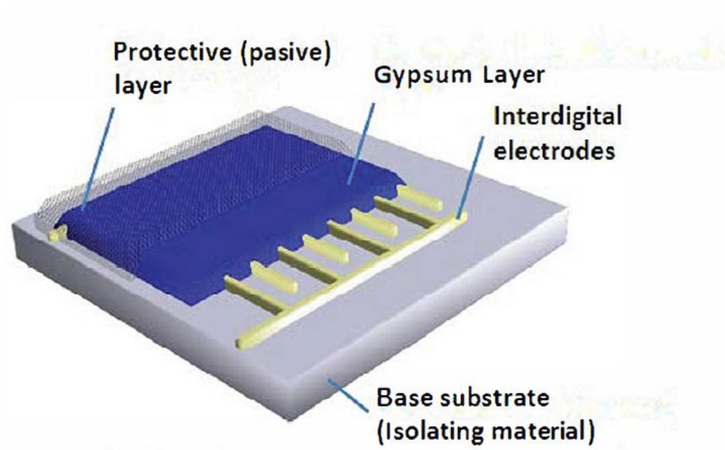

Figure 5 protective layering strategy

Software Testing

Our web-application designer will keep checking the accuracy and performance of displayments and classifications of the web server. Also, our team will ask for feedback from users to make sure if the data are readable or understandable.

4 Closing Material

4.1 Conclusion

We are setting out to solve the problem that monitoring large expanses of land is a very costly endeavor. Our solution will be a wireless sensor network that will provide better data than our competitors at a much lower cost. A soil moisture sensor and humidity sensor will be developed and equipped on leaf nodes. Each node will collect data from sensors which will be sent to the home node. The project will have a website that will display the data that be uploaded by the home node to our clients in a simple and friendly way. We are working on the nodes connection.After that we are going to find a good solution for power supply then we are ready to test our products in the field. We have a very strong head start because we are continuing the work of a previous senior design group. This fact along with our passion for solving this problem make us confident we will be successful in developing this system.

4.2 References

[1] Miner, Andrew. "Networks 2." Lecture.

[2] Tuttle, Gary, Barebones Arduino.Print

[3] TempuTech, "Wireless Sensor Monitoring," TempuTech, [Online]. Available: http://www.temputech.com/26-home/slider/113-wireless-sensor-monitoring. [Accessed 21 02 2017].

[4] Banner Engineering Corp. "Wireless I/O & Data Radios | Products for Industrial & Process Automation." Banner Engineering. Banner Engineering, n.d. Web. 21 Feb. 2017.

[5] The 5 Step Problem Solving Approach, "eba," eba, 2016. [Online]. Available: http://www.educational-business-articles.com/5-step-problem-solving/. [Accessed 21 02 2017].