LIMA: Lovely Irrigation Monitoring App Final Report

Dec1717

Client: Dr. Ajay Nair
Advisor: Dr. Manimaran Govindarasu
Team Members:
Daniel Albers: Key Idea Concept Holder
Sam Jackson: Webmaster
Seth Lightfoot: Key Idea Concept Holder
Sierra Lucht: Team Leader
Landon Woerdeman: Team Communication Leader
Team Email: dec1717@iastate.edu
Team Website: http://dec1717.sd.ece.iastate.edu

Last Revised: December 5, 2017

Table of Contents:

1. Introduction and Motivation

1.1. Introduction

Crop irrigation monitoring via the use of "smart" sensors is an industry that is currently burgeoning (Vogt). This project, the Lovely Irrigation Monitoring Application (LIMA), focused on the development of a low cost, easy-to-use, wireless monitoring system that guarantees soil moisture accuracy. The goal of LIMA is to ensure farmers are equipped with easily-accessible and reliable data to help them determine their exact irrigation needs.

1.2. Motivation

Discerning the appropriate time for field irrigation is a problem faced by many crop growers. While some farmers adhere to a classic "gut feeling" approach of physically sampling their soil, many have begun deploying sophisticated soil moisture sensor networks in their fields. While the trend of adopting high-tech sensing solutions is expected to grow, many farmers are hesitant to adapt to these changes due to the cost implications and difficulty of use (Wood). The motivation behind LIMA is to alleviate both of these concerns by creating easily-configurable sensor networks at low costs; while leaving a positive environmental impact by reducing water consumption. Our project aims to help farmers reluctant to adopt new technologies save on irrigation costs, produce better yields, and reduce their overall water usage.

1.3. Existing Technologies

Our client introduced us to two of the most commonly-used moisture sensors deployed in fields today: Decagon Devices and Watermark Soil Moisture Sensors. Each one of these systems use a number of sensor nodes wired directly to a common collection station.

1.3.1. Decagon Devices

Sensor

Data Logger

Figure 1: Decagon Devices Components (METER Environment)

Figure 1 displays the sensor and collection station used in a typical Decagon Devices moisture sensing setup. This setup uses the highly accurate 10HS sensor, having a sensing radius of up to 1.3 liters (METER Environment). These sensors hook into the EM50 data logger, maxing out at 5 sensors per station. The logger is battery powered and will continuously collect data at a given interval. At $139 per sensor, and $475 per collection station, the overall cost of a 5 unit collection system would be $1,170.

1.3.2. Watermark Soil Moisture Sensors

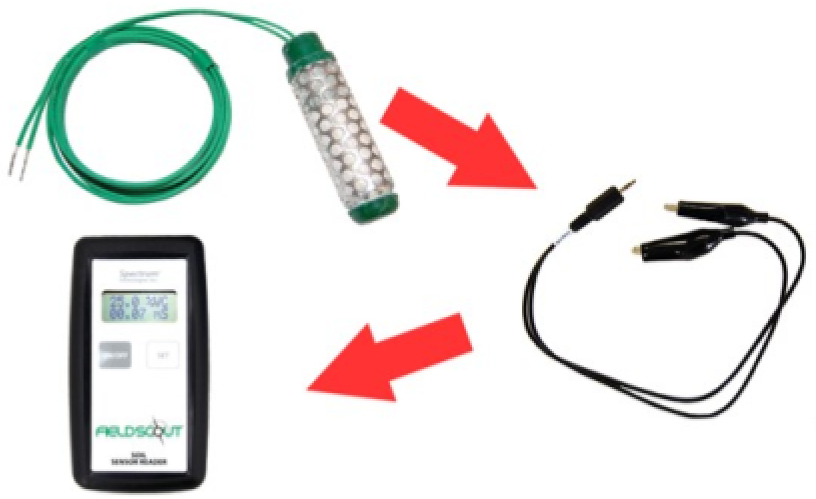

Figure 2: Watermark Soil Moisture Sensor Setup
(Watermark Soil, Watermark Logger)

Figure 2 displays the items necessary to use the Watermark Soil Moisture Sensors. This system is slightly different than the Decagon setup. In this setup, the sensors are not plugged into the collection unit at all times. The operator must go to each sensor and manually connect the sensor to the collection unit to get the moisture reading at that time, there is no long term collection solution. Each sensing node costs $79, each adapter $36, and the collection station costs $279 for a total of $854 for a 5 sensor setup.

2. Project Design

2.1. Prototype I

Originally, we had a design plan, as shown in Figure 1. This design depicts a sensor placed in the ground, which would communicate via Bluetooth with the farmer's mobile phone.

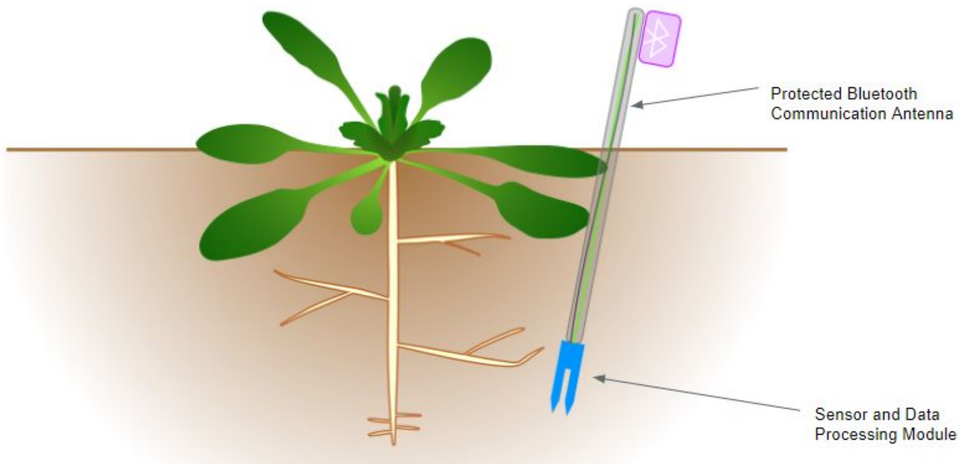

Figure 3: A drawing of the initial prototype

However, there were several reasons this design needed to be modified. First, the battery life of the sensor was not nearly long enough to last an entire growing season, as it was always looking to connect via Bluetooth. Additionally, we were unsure if the new sensors were measuring reliable information. For these reasons, we moved onto a new design, which is shown in Figure 2.
Additional information about prototype I can be found in Appendix II.

2.2. Prototype II

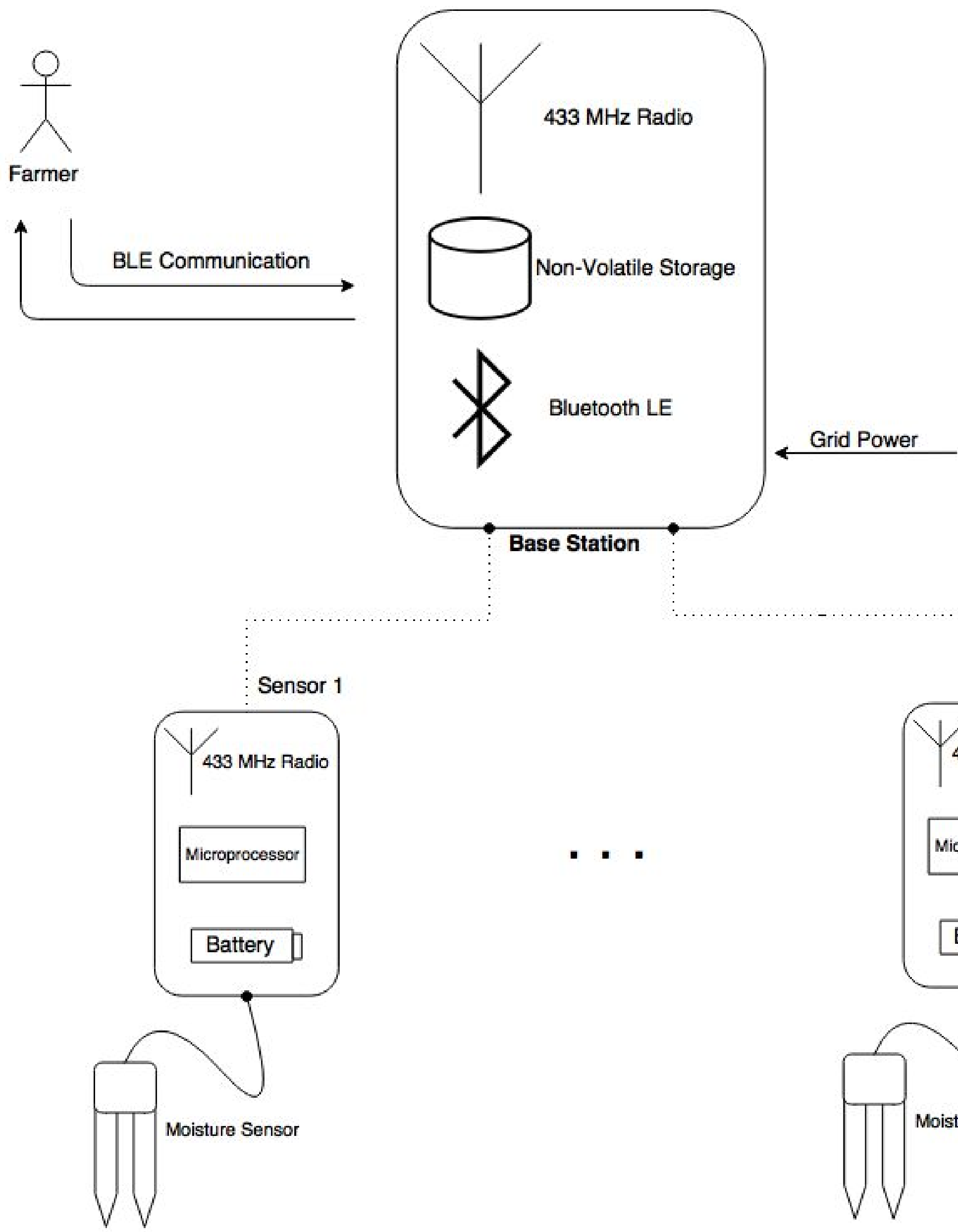

Figure 4: An architectural diagram of Prototype II

The design is intended to work so that the soil moisture sensor is buried into the dirt. It will then relay the information with a Bluetooth microcontroller above soil. When the farmer visits the field, they will press a button on the Base Station that will signal all of the sensors to relay their data. The microcontrollers will then wirelessly relay their information to the Base Station using 433Mhz radio. The Base Station will then transmit the data to the smartphone application using Bluetooth.

3. Implementation Details

3.1. Hardware

The details of how the irrigation monitoring system was implemented is broken down into two parts. Figure 3 displays the sensor nodes, and Figure 4 shows the Base Station setup.

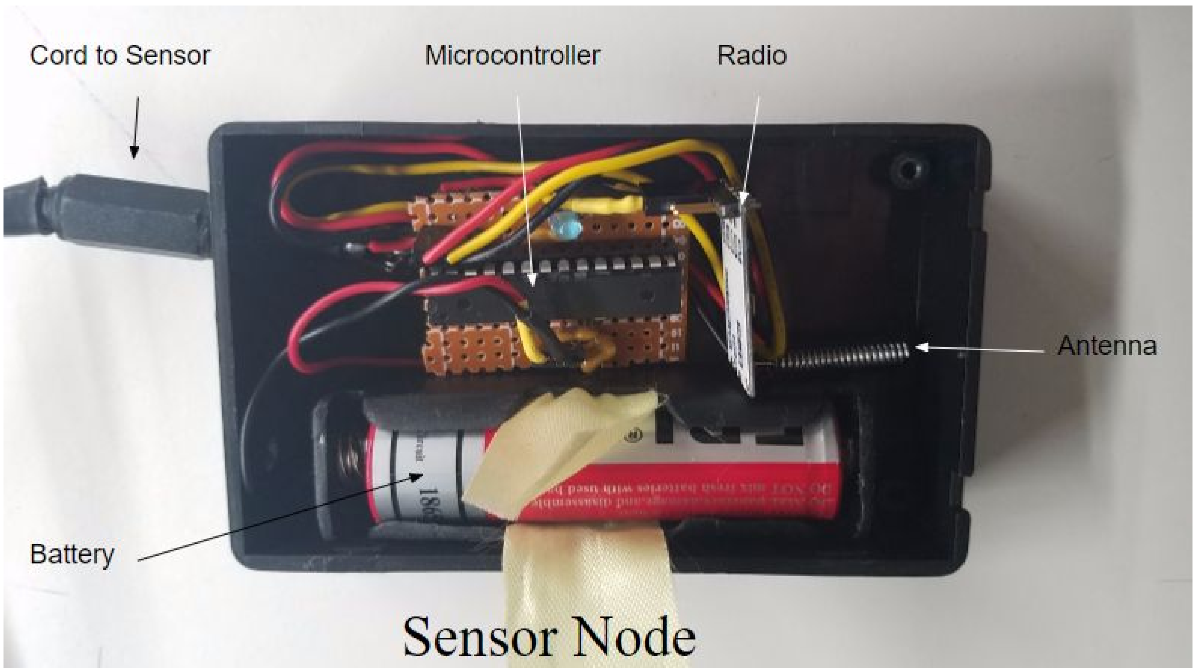

Figure 5: An image of the prototype of the sensor node

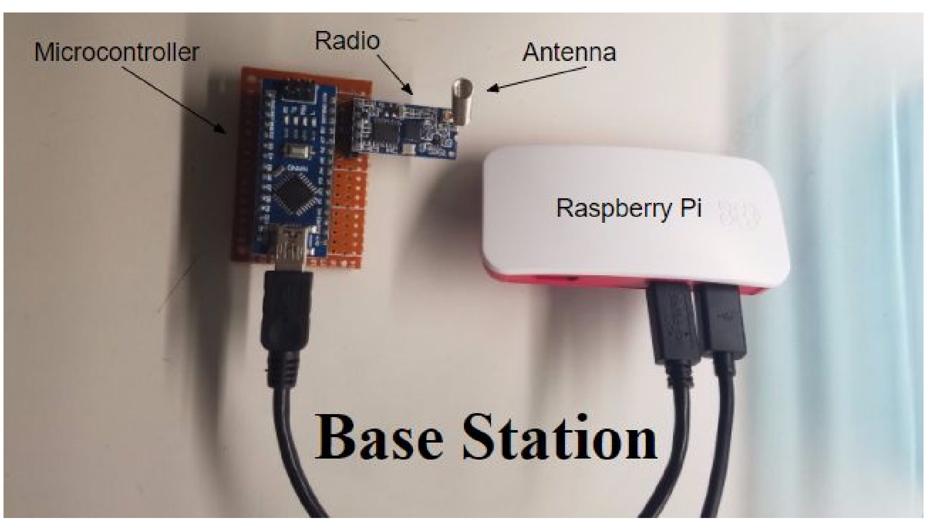

Figure 6: An image of the prototype of the Base Station

The sensor nodes in Figure 3 show the battery powering a microcontroller. This microcontroller then powers the radio with antenna to send values to the Base Station, as well as the sensor, which is connected to the cord. In Figure 4, the Base Station is shown. It too had a radio and antenna to communicate with the sensor nodes, as well as a microcontroller. The Base Station, however, also has the ability to communicate via Bluetooth LE with the farmer's smartphone.

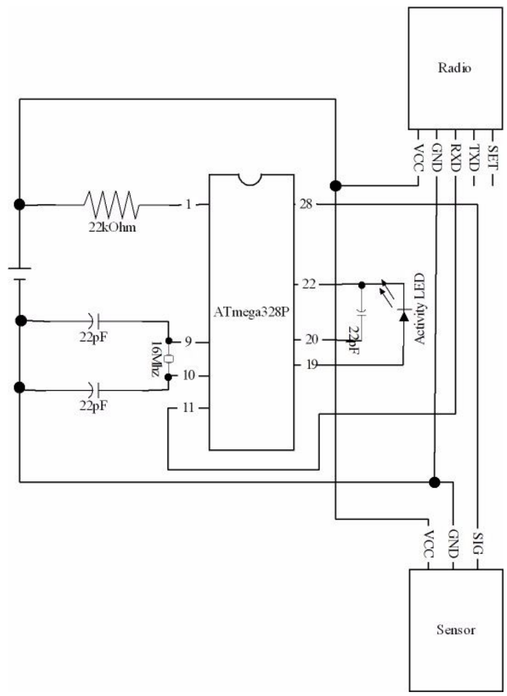

Figure 7: A diagram of the sensor node setup

To assemble each sensor node, a diagram was created, as shown in Figure 5. This diagram details the connections of each wire, as well as the overall setup. Using this diagram, we assembled each of the nodes.

For additional information on how to setup and run this system, an operation manual can be found in Appendix I.

3.2. Software

3.2.1. Sensor Nodes

Each sensing node consists of a base Arduino IC running custom software. This software is setup to read the sensor on a given time interval. To fight any kind of radio contention, the sensors transmit their reading, with the same transmission ID, three times with a random amount of delay between transmissions. Once the last transmission is completed, the node will transition into an ultra low power mode where the Arduino draws as little power as possible.

3.2.2. Base Station

The Base Station was implemented on a Raspberry Pi Zero running a combination of Node JS and Python code. The Python code governs "radio tasks," eg: reading, parsing, and storing messages received from each soil moisture sensor; while the Node JS handles Bluetooth Low Energy (BLE) connectivity and file transfers to the mobile device. The decision for using an amalgam of Python and Node JS stemmed from the usability of libraries supporting each feature we needed to implement on the Base Station. During implementation, we discovered Python was well-suited and easily approachable for reading incoming serial message data from the connected radio peripheral; however, we struggled to build out BLE functionality in Python and opted to switch to Node JS since there was a well-document library, "Bleno" available on Github (Sandeepmistry). It uses a Node JS interface to access the BLE hardware installed on the Pi.

3.2.3. Mobile Application

To develop the mobile application interface for LIMA, we sought a platform capable of cross-compiling a single codebase into both Android and iOS applications. Ultimately, the platform of choice was Flutter: an open-source package that compiles a single Dart project into native iOS and Android binaries (Flutter). Flutter allowed us to write an application code pertaining to UI display and state logic for both platforms simultaneously, while still giving us the leeway to implement device-specific functionality in Java and Swift independently.

Additional information about the software for this project can be found in Appendix III.

3.3. Cost

This is a cost estimate of the system, which lists all of the parts used to build this system.

Item	**Unit Price**	**Quantity**	**Sub-Total**
10HS Soil Sensor	$139	5	$695
Raspberry Pi Zero W	$10	1	$10
SD card	$3	1	$3
CH340G NANO	$2.90	1	$2.90
Arduino (IC)	$2.18	5	$10.90
Battery	$1.50	5	$7.5
Enclosure	$.87	5	$4.35
Radio	$4.00	6	$24
Add. Electrical Components	$20	1	$20
		Base Station Price:	$20
		Price Per Node:	$151.53
		5 System Total:	**$777.65**

Figure 8: A table which displays the cost of the system

Another benefit of this solution is the reduced cost after five nodes due to the limitations of the current system. The existing system can only support five nodes per base station, whereas this solution can support a much larger amount of nodes.

Another benefit of this solution is the adaptability of the existing system. If the customer already owns the existing system, it can be adapted to work with Prototype II. This would subtract the price of the sensors themselves ($139.00). This would bring the 5 system total to $82.65, a very significant cost difference.

4. Testing Processes

To ensure the project worked as designed, testing was split into several stages, with modifications being made based on the results of the previous stages.

4.1. Prototype I Test

Initial testing was done with Prototype I in the greenhouses on campus.The sensor was placed in a plant, and the data was remotely monitored. However, this sensor died rather quickly, posing the issue of battery life. Additionally, the results were somewhat unreliable. The sensor seemed to have a low resolution, only really distinguishing if the soil was wet or not, with little differentiation of the levels in between. This issue led to the question of if this new sensor was reliable enough to be used.

4.2. Sensor Reliability Test

To address the issues found in Prototype I, the design of Prototype II was created. To test this issue of sensor reliability in Prototype II, two identical sensor node systems were created, one using the cheaper SparkFun sensor, and one using the more expensive 10HS sensor. It was found that the cheaper sensor was unable to detect many of the changes in moisture level outside simply if the soil was wet or not. This discovery led to the use of the more expensive 10HS sensor with the system.
Details of the results of this test can be found in section 4.4, Raw Sensor Data

4.3. Comparative Test

With the decision to move forward with the more expensive 10HS sensor, one final test needed to be performed. The new system and the existing system were set up to be tested against each other. It was found that the data from the new system was very similar to that of the existing system, showing that this system was reliable enough to use in the field.
Details of the results of this test can be found in section 4.4, Raw Sensor Data

4.4. Raw Sensor Data

Data was collected from both the LIMA system and the client’s existing solution in the ISU Department of Horticulture greenhouses during the testing described in Section 4.1-4.3. Using two LIMA sensor nodes we gathered data with a high accuracy 10HS sensor (referred to as T5) provided by the client, and a low cost SparkFun Soil Moisture Sensor (referred to as T6). Figure 9 shows T5 raw ADC readings in millivolts over the test period. Figure 10 shows T6 raw ADC readings in millivolts over the test period.

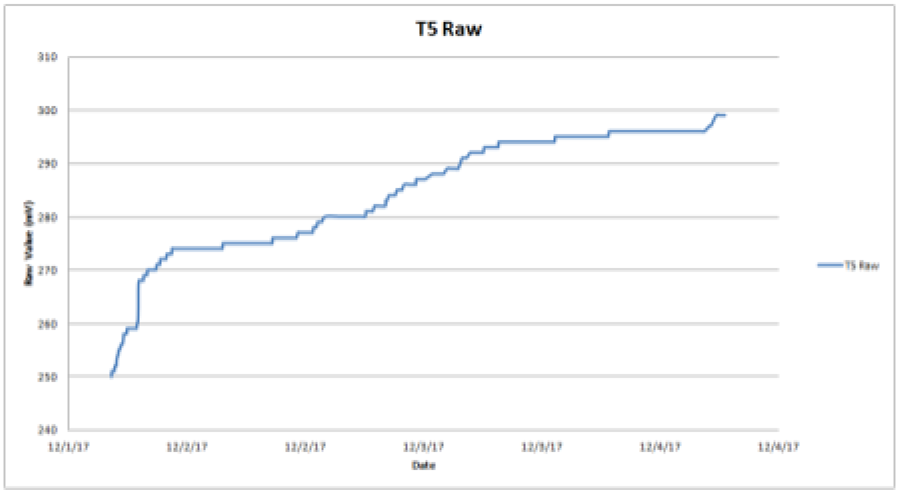

Figure 9 - T5 Raw ADC Value

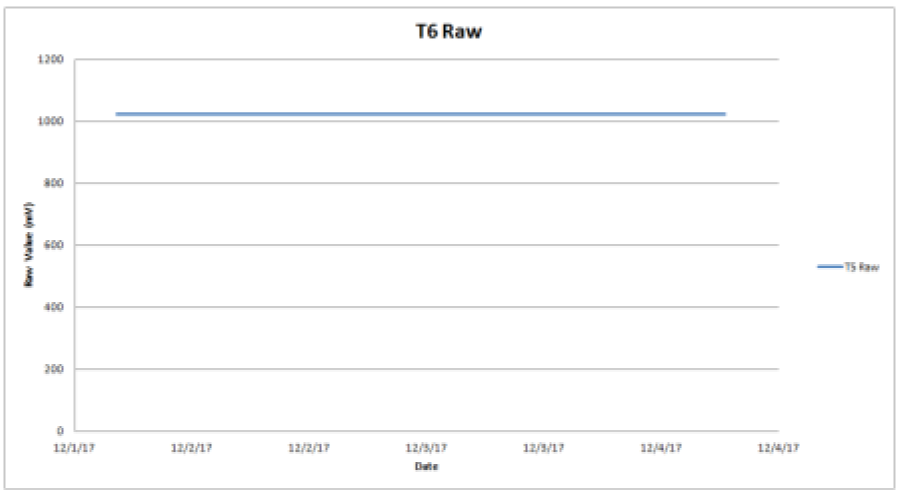

Figure 10 - T6 Raw ADC Value

The trend line indicates that T5 is collecting data with much higher accuracy than T6. T6 ADC was capping at 1023 for the duration of the test. Thus, the less expensive Sparkfun Sensor does not report with a fine enough accuracy and wasn't considered for final use. No further data analytics were conducted using this sensor.

4.5. Sensor Data Conversion

In effort to make the ADC values reported by the LIMA node readable, we must convert this raw voltage to Percent Volumetric Water Content (% VWC). The formula used was found in the data sheet for the 10HS sensor (T5) and is pictured in Figure 11.

ABS(2.589E-10 * mV^4 * -5.010E-7 * mV^3 + 3.523 E-4 * mV^2 - 9.135E-2*mV + 7.457)/2 + 20

Figure 11 - T5 %VWC Conversion Equation

Using this equation, we were able to convert the raw ADC values reported by the LIMA node to a usable metric. Figure 12 shows the conversion over the course of the test period.

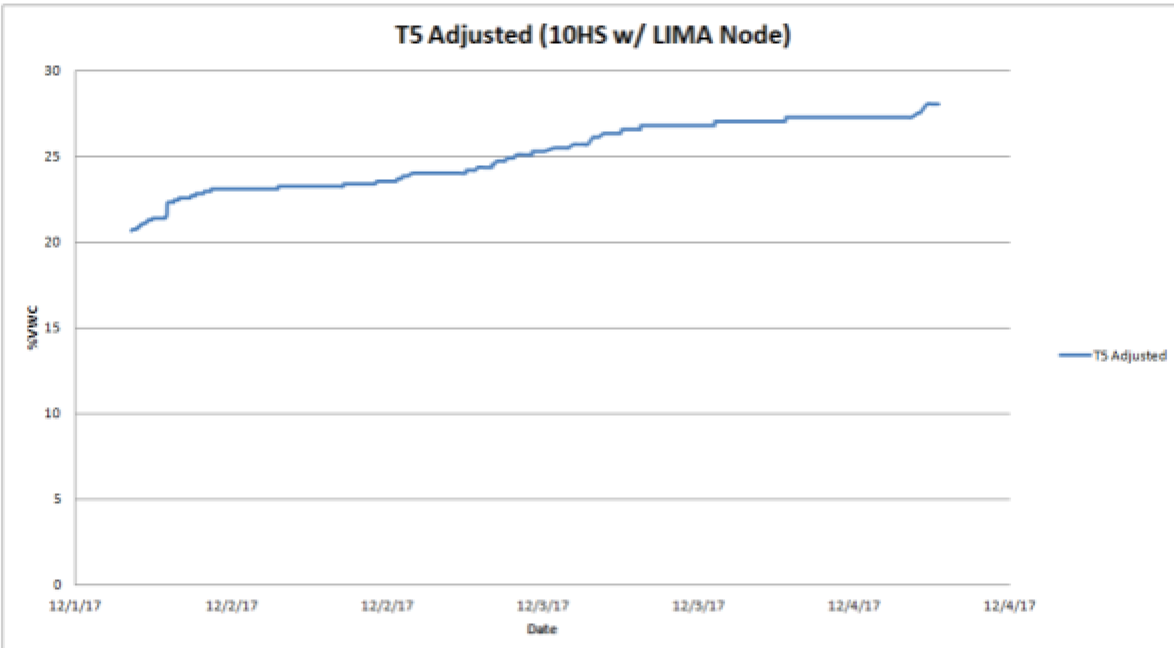

Figure 12 - Converted T5 ADC values

4.6. Comparison with Existing Solution

Initial impressions of the results are good. While the trendlines for the existing solution and LIMA are not completely overlapping, they are operating within an acceptable range. Figure 13 shows the existing system data collection over the test period vs. LIMA system data collection over the same period.

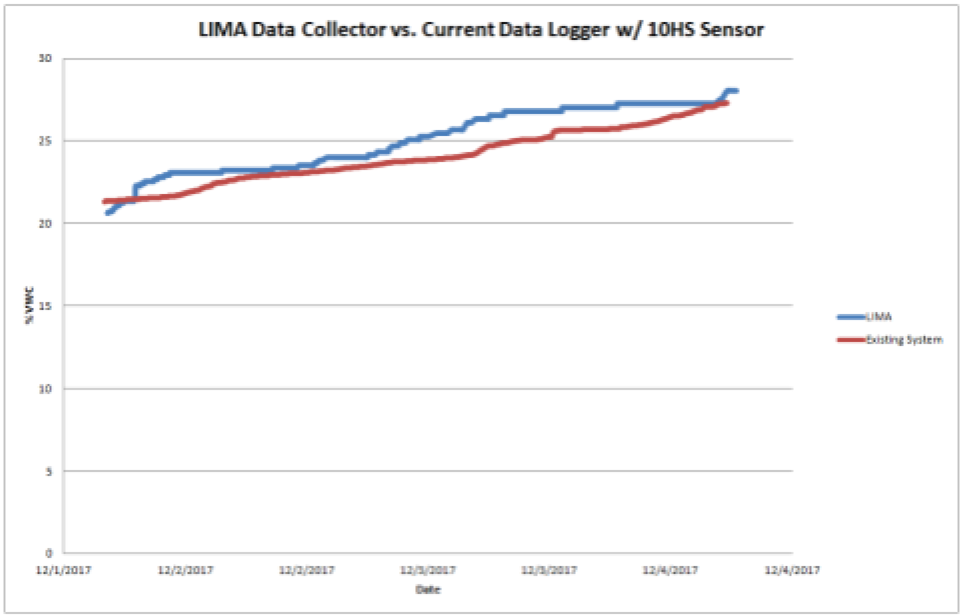

Figure 13 - Client's current system vs. LIMA system

Although we could not totally minimize cost by using the less expensive SparkFun Soil Moisture Sensor, we are still able to achieve a lower overall system cost while using the higher accuracy 10HS sensors. By using 10HS sensors with LIMA sensor nodes and Base Station, the client should see an increase in system usability at a slightly decreased cost without sacrificing accuracy. LIMA expands the current systems feature set by providing users with a hassle free way to connect to the Base Station via their Bluetooth LE enabled mobile device and read the %VWC of many sensors at a time. Section 6 outlines the expansion of this functionality, bringing more to the client at a less expensive price point.

5. Concluding Remarks

5.1. Lessons Learned

Throughout the course of developing LIMA, we experienced triumphs and challenges completely new to us. Working with a team on a multi-semester project of this size ensured that proper communication, workflow and organization would be key to success. This project helped us immensely to develop these practical skills. We also faced many technical challenges we had to overcome including hardware design, application development, and working with new software packages and SDKs such as Flutter, which was used to develop the mobile application. Appendix IV details just some of the resources we leveraged as we developed LIMA.

6. Future Expansion

In the future, this project could be expanded and build upon in several ways. We examined several similar projects that have been documented to get a better feel for what was on the market. This led us to several hardware ideas, as well as several software ideas.

- Integration of control hardware
    - Turn on irrigation system when moisture levels are low
- Solar panels
    - Longer battery life
    - Environmentally friendly
- Power Management
    - Mosfet to control sensor’s regulator
- Metrics
    - Identify trends
    - Make educated decisions about crops

7. Appendix I – Operations Manual

7.1. Assembling Moisture Sensors

In order to use the LIMA system, each sensor in the network must be assembled and configured. For each soil moisture sensor to be added to the network, connect the male-end barrel connector on the sensor to the enclosure. Then, open the sensor enclosure and place a charged battery into the battery housing, using adequate force as necessary. Finally, bury the sensing-end of each sensor in the field at their desired locations and depths. Once the devices power on, each will begin to automatically sample the soil moisture and attempt to report it to the Base Station at periodic intervals.

7.2. Assemble Base Station

Configuring the Base Station is a straightforward process of attaching the radio antennae and powering-on the device. Attach the radio antennae to the Base Station using the provided, doubly-ended Mini-USB cord; then, using the provided Mini-USB power adapter, connect the Mini-USB end to the Base Station and plug the outlet into a 110V wall socket. Once the device is powered on, allow two minutes for the device to finish booting; after such, the Base Station will begin recording all samples collected from the soil moisture sensors and is ready to communicate with mobile devices.

7.3. Using the LIMA application

Data may be pulled from the Base Station using the LIMA mobile application after the soil moisture sensors have been planted and activated, the Base Station has been powered on, and the Base Station has received soil moisture samples from the sensors in the network. This document makes no assumptions of the users device, apart from it having Bluetooth Low-Energy capabilities.

To begin using the LIMA mobile application, the user should turn on their device's Bluetooth. If they haven't enabled Bluetooth prior to opening the app, the app will load with a prompt to enable the Bluetooth. Clicking the prompt button, the user will be redirected to the device's settings page where they can turn on Bluetooth. After LIMA has permission to access and use the phone's Bluetooth radio, the app begins to scan for the Base Station in the background. If the device was not found the phone will prompt the user to connect to the Base Station. Upon a successful scan where the station was found in near proximity to the phone, LIMA connects to the peripheral and prompts the user to get the most recent data from it.

After clicking the prompt button, the phone subscribes to Bluetooth Low Energy notifications from the Base Station. This allows the Base Station to send multiple packets in a row to be received on the phone. Since BLE only allows 20 bytes per packet and the packets are sent every 50 ms, this step is relatively slow, and it can last up to several seconds depending on the size of data being transmitted. After receiving the last chunk of data, LIMA terminates the connection by unsubscribing from the Base Station. The data is then saved to an internal application file, and is loaded into the application GUI.

The homepage of LIMA displays the current weather, each Base Station's average soil moisture, and the button prompt. In order to view the specific sensor readings, the user should click on the Base Station's icon. This will load a new page that displays each sensor that has broadcasted to the station and their respective average moistures. In order to view the moisture content for a specific sensor, the user can click on the sensors icon. Here a new page is shown that has the moisture history, and options to rename the node.

8. Appendix II – Alternative Versions & Designs

8.1. Prototype I

The first design was much more simplistic than the current version. This plan involved each sensor communicating via Wifi. However, after implementing this system, we quickly found that the batteries died very quickly. In order to fix this, we transitioned to a Base Station communicating via Bluetooth LE, but all other nodes using a 433Mhz radio.

Item	**Unit Price**	**Quantity**	**Sub-Total**
Soil Moisture Sensor	$4.70	10	$47.00
BLE Transceiver	$1.50	10	$15.60
CH340G NANO	$1.589	10	$15.89
Battery	$4.80	10	$48.00
Enclosure	$.87	10	$8.70
		10 System Total:	$135.20
		Price Per System:	$13.52

9. Appendix III – Code

The code for the application is stored on GitLab through Iowa State. The link is provided:

https://git.ece.iastate.edu/groups/lima_492

10. Appendix IV – Acknowledgements

We would like to acknowledge the people who supported Dec1717 and project LIMA and the resources we leveraged during the course of this project, CprE/EE/SE 491 and CprE/EE/SE 492:

- Dr. Manimaran Govindarasu, Department of Electrical and Computer Engineering
- Dr. Ajay Nair, Department of Horticulture
- Dr. Goce Trajcevski, Department of Computer Science
- Mr. Leland Harker, Electronics Technology Group
- Iowa State University Department of Electrical and Computer Engineering
- Iowa State University Department of Computer Science
- Iowa State University Electronics Technology Group

11. Appendix V – References

"Flutter Makes It Easy and Fast to Build Beautiful Mobile Apps." Easy and Fast SDK for Beautiful Apps - Flutter, flutter.io/

Markets, Research and. "Soil Moisture Sensors Market to 2025 - Increase in the Adoption of Smart Irrigation Techniques." GlobeNewswire News Room, 28 June 2017, globenewswire.com/news-release/2017/06/28/1030039/0/en/Soil-Moisture-Sensors-Market-to-2025-Increase-in-the-adoption-of-smart-irrigation-techniques.html.

METER Environment. "LARGE VOLUME SOIL MOISTURE SENSING". MeterGroup, 2017, http://www.decagon.com/en/soils/volumetric-water-content-sensors/10hs-large-volume-vwc/.

METER Environment. "METER Legacy Data Loggers". MeterGroup, 2017, https://www.decagon.com/en/data-loggers-main/data-loggers/.

Sandeepmistry. "Sandeepmistry/Bleno." GitHub, 24 Apr. 2017, github.com/sandeepmistry/bleno

"SparkFun Soil Moisture Sensor." SparkFun Start Something, SparkFun Electronics Inc, www.sparkfun.com/products/13322.

"Watermark Monitor Data Logger. Read Soil Moisture Automatically." Irrometer Watermark Monitor (900M), www.berryhilldrip.com/Watermark-900M-Monitor-Data-Logger.-Read-Soil-Moisture-Automatically..html?_vsrefdom=adwords&gclid=CjwKCAiApJnRBRBlEiwAPTgmxDBQ13TSwRyDnItM8wPdC2zfBXBkjq-PrJ2LISI61eIe1SEBN5EBMBoCAtYQAvD_BwE.

"Watermark Soil Moisture Sensors." Watermark Soil Moisture Sensors | Spectrum Technologies, www.specmeters.com/soil-and-water/soil-moisture/soil-moisture-sensors/watermark-soil-moisture-sensors/.

Willie Vogt | Jan 05, 2015. “Soil Sensors for Efficient Irrigation.” Farm Industry News, 16 May 2017, www.farmindustrynews.com/tech-work/soil-sensors-efficient-irrigation.