CloudGarden
Team 6

Date: April 19, 2021

Primary Author: Donovin Lewis

This project was created for the 2020-2021 term of the University of Kentucky's (UK) Electrical and Computer Engineering (ECE) Senior Design and sponsored by an industry partner. It has been released to the public after conversation and confirmation with our project's sponsor regarding its release. Sections and details have been removed to provide anonymity.

Table of Contents

List of Figures 3
List of Tables 5
1. Abstract 7
2. Problem Statement 7
2.1 Need 7
2.2 Objective 7
2.3 Background 8
3. Requirements Specification 8
3.1 Marketing Requirements 8
3.2 Objective Tree 8
3.3 Engineering Requirements 10
3.4 Broader Impact 14
3.4.1 Economic Impact 14
3.4.2 Environmental Impact 14
3.4.3 Manufacturing Impact 14
3.4.4 Standards Impact 15
3.4.5 Health and Safety Impact 15
4. Design 15
4.1 Functional Decomposition 15
4.1.1 Level 2 Functional Decomposition 17
4.1.2 Level 3 Functional Decomposition 19
4.2 Design Summary 22
4.2.1 Wireless Data Transmission 22
4.2.2 Hub Processing and Presentation 25
4.2.3 Power Sourcing and Sensing 26
4.2.4 Soil Moisture Sensing 30
4.2.5 Controls and Timing 36
4.2.6 Enclosure and Waterproofing 41
5. Integration, Verification and Validation 42
5.1 Integration 42
5.1.1 Hardware Integration 42
5.1.2 Software Integration 45
5.1.3 Integration Difficulties 46
5.2 Verification/Validation Plans 49
5.2.1 Wireless Transmission 49
5.2.2 Central Hub 50

5.2.3 Power Sourcing and Sensing 50
5.2.4 Soil Moisture Sensing 51
5.2.5 Controls and Timing 52
5.3 Completed Verification and Validation 53
5.3.1 Wireless Transmission Test 53
5.3.2 Central Hub Test 54
5.3.3 Power Sourcing and Sensing Test 55
5.3.4 Soil Moisture Sensing Tests 57
5.3.5 Controls and Timing Tests 62
5.3.6 Enclosure and Waterproofing Tests 65
6. Final Costs 66
7. Results and Lessons Learned 68
7.1 Project Results Summary 68
7.2 Lessons Learned 70
8. References 71
9. Appendix 76
9.1 AHP Decision Making Process 76
9.2 Waterproofing Standards: IP-54 80
9.3 Design Alternatives 81
9.3.1 Wireless Data Transmission 82
9.3.2 Hub Processing and Presentation 83
9.3.3 Power Sourcing and Sensing 84
9.3.4 Capacitive Moisture Sensing 86
9.3.5 Controls and Timing 87
9.3.6 Mechanical Assembly and Enclosure 88
9.4 Detailed Designs 90
9.4.1 Final Code 94

List of Figures

Figure 1: Objective Tree for Outdoor Soil Sensor 10
Figure 2: Level 0 – CloudGarden System 16
Figure 3: Level 1- CloudGarden Functional Decomposition 16
Figure 4: CloudGarden Behavior Model 17
Figure 5: Level 2- Soil Moisture Sensor Functional Decomposition 17
Figure 6: Level 2- Hub Functional Decomposition 18
Figure 7: Level 3- Radio Functional Decomposition 19
Figure 8: Level 3- Capacitive Moisture Sensing Functional Decomposition 20
Figure 9: Level 3- MCU Functional Decomposition 20
Figure 10: Level 3- Temperature Sensing Functional Decomposition 21

Figure 11: Mechanical Assembly and Enclosure Abstract Diagram 21
Figure 12: Wireless Transmission Protocol Objective Tree 22
Figure 13: RFM69HCW-Arduino Connection 23
Figure 14: RFM69HCW Breakout Board Example 24
Figure 15: RFM69HCW Transceiver Application Circuit 25
Figure 16: Central Hub Behavior Flowchart 26
Figure 17: Battery Temperature/Voltage Relationship 26
Figure 18: Battery Life Estimate Spreadsheet 27
Figure 19: Lithium Thionyl Chloride Battery Discharge Plot 28
Figure 20: Battery Monitor IC Application Circuit 29
Figure 21: Power Sourcing and Sensing Circuit 29
Figure 22: Bandgap Voltage Measurement Method 30
Figure 23: Soil Moisture Sensing Objective Tree 30
Figure 24: Milled PCB plates used for capacitance testing 31
Figure 25: LCR Capacitance Measurement Example 31
Figure 26: Patented Interdigitated Finger Capacitor Example 32
Figure 27: Example of soil moisture sensor input and output 32
Figure 28: Capacitive Soil Moisture Sensing Circuit Example 33
Figure 29: Equations representing astable 555 Timer Waveform Generation 33
Figure 30: Capacitive Soil Moisture Sensor Output 34
Figure 31: Simplified Capacitive Soil Moisture Sensing 34
Figure 32: Soil Moisture Sensing Prototype 35
Figure 33: Gravity Capacitive Soil Moisture Sensor Example 35
Figure 34: PCB Capacitor Design 35
Figure 35: Soil Moisture Sensing Example Code 36
Figure 36: Arduino Pro Mini Schematic 37
Figure 37: Microcontroller Application Circuit 38
Figure 38: ICSP Pinout 38
Figure 39: PCB Connectors 38
Figure 40: USB to FTDI Breakout Board Schematic 39
Figure 41: USB to FTDI Application Circuit 40
Figure 42: Timing and Mode Code 40
Figure 43:NTC Thermistor Code 41
Figure 44: 1st Soil Moisture Sensor PCB Prototype 43
Figure 45: Partially Populated Prototype 43
Figure 46: Revision of Soil Moisture Sensor Prototype 44
Figure 47: Final Prototype 45
Figure 48: Arduino as ISP Connection 46
Figure 49: Arduino as ISP Programming 47
Figure 50: USBasp Programming 47
Figure 51: Teensy 3.2 ADC Attempt 48
Figure 52: Teensy Microcontroller ADC Attempt Code 48
Figure 53: Wireless Communication Troubleshooting/Testing 53
Figure 54: Transceiver Setup Code 54
Figure 55: Central Hub Webpage 55
Figure 56: Batteries received from our Sponsor 56
Figure 57: Battery Capacity Testing 56
Figure 58: Wireless Program Testing 57
Figure 59: Functional Battery Operation Test 57
Figure 60: Using PuTTY to log soil moisture measurements. 58
Figure 61: SongHe Capacitive Soil Moisture Sensor 58

Figure 62: Commercial Capacitive Soil Moisture Sensing Testing 58
Figure 63: SongHe Capacitive Soil Moisture Sensor Testing Results 59
Figure 64: Custom Soil Moisture Sensor Testing 60
Figure 65: Custom Soil Moisture Sensing Results 60
Figure 66: Final Prototype for Soil Moisture Sensing Tests 61
Figure 67: Soil Moisture Measurement in Air 61
Figure 68: Voltage Change with Capacitamce Variation 62
Figure 69: Soil Moisture Measurement in a Wet Plant 62
Figure 70: Microcontroller Power Draw Test Circuit 63
Figure 71: Measured voltage for active state current calculation 63
Figure 72: Measured voltage for low power mode current calculation 64
Figure 73: Temperature sensor output without applied heat 64
Figure 74: Temperature lowering voltage output 65
Figure 75: Prototype prior to IP-54 Testing 65
Figure 76: Dry prototype after removing enclosure 66
Figure 80: Soil Moisture Sensing Device Prototype Cost Breakdown 67
Figure 81: Final Soil Moisture Sensor Cost Breakdown 68
Figure 82: Central Hub Cost Breakdown 68
Figure 83: IP Ratings Guide 81
Figure 84: Battery Life Estimate Spreadsheet 91
Figure 85: Soil Moisture Sensor EAGLE Prototype Schematic 92
Figure 86: Soil Moisture Sensor Prototype V1 EAGLE Board File 93
Figure 87: Soil Moisture Sensor Prototype V2 EAGLE Board File 94

List of Tables

Table 1: Engineering Requirement, Rationale, and Justification 10
Table 2: Engineering Requirement Verification 12
Table 3: CloudGarden Level 0 Functional Decomposition Explanation 16
Table 4: CloudGarden Level 1 Functional Decomposition Explanation 16
Table 5: Soil Moisture Sensor Level 2 Functional Decomposition Explanation 18
Table 6: Central Hub Level 2 Functional Decomposition Explanation 18
Table 7: Radio Level 3 Functional Decomposition 19
Table 8: Capacitive Moisture Sensing Level 3 Functional Decomposition Explanation 20
Table 9: MCU Level 3 Functional Decomposition Explanation 20
Table 10: Temperature Sensing Level 3 Functional Decomposition Explanation 21
Table 11: Results of PCB Capacitor Testing 32
Table 12: Wireless Transmission Testing 49
Table 13: Central Hub Testing 50
Table 14: Power Sourcing and Sensing Testing 50
Table 15: Soil Moisture Sensing and Enclosure Tests 51
Table 16: Microcontroller Testing 52
Table 18: ER Progress 69
Table 19: AHP Matrix for the Outdoor Wireless Soil Moisture Sensor 77
Table 20: AHP Matrix for the Soil Moisture Sensor's Data Collection 77
Table 21: AHP Matrix for the Soil Moisture Sensor's Accessibility 77
Table 22: AHP Matrix for the Soil Moisture Sensor's Data Quality 77
Table 23: AHP Matrix for the Soil Moisture Sensor's Longevity 77
Table 24: AHP Matrix for the Wireless Transmission Protocol 78
Table 25: AHP Matrix for the Wireless Transmission Protocol's Cost 78
Table 26: AHP Matrix for the Wireless Transmission Protocol's Power Consumption 78

Table 27: AHP Matrix for Central Hub Hardware Selection.. 78
Table 28: AHP Matrix for Soil Moisture Sensing 78
Table 29: AHP Matrix for Soil Moisture Sensing's Waveform Generation 79
Table 30: AHP Matrix for Soil Moisture Sensing's PCB Capacitor Sourcing............................... 79
Table 31: AHP Matrix for Waterproofing 79
Table 32: AHP Matrix for MCU Selection.. 79
Table 33: Wireless Transmission Protocol Design Alternatives.. 82
Table 34: Hub Processing Design Alternatives 83
Table 35: Power Sourcing and Sensing Design Alternatives... 84
Table 36: Capacitive Soil Moisture Sensor Design Alternatives... 86
Table 37: Controls and Timing Design Alternatives 88
Table 38: Mechanical Assembly and Enclosure Design Alternatives 89

1. Abstract

Our team, CloudGarden, developed a design for a wirelessly connected outdoor soil moisture sensor for home gardeners and hobbyists who want to simplify their daily gardening routine or may not be able to check on their crops with great frequency. For this purpose, our primary goal was the sensing and indication of soil moisture alongside the wireless communication of measurements to a user-friendly hub. This report begins by detailing the problem addressed by our device, the requested features from our sponsor, and the technical requirements necessary for those features alongside a review of the impacts of our device and constraints placed upon it. Keeping these requirements in mind, we designed a solution and broke it down into the individual subsystems: wireless data transmission between devices; the interface or logging of data at the central hub; power source and sensing system for deployed devices; soil moisture sensing component and waterproofing; and basic algorithm design, device control, and microcontroller selection. We systematically compared and selected from possible solutions for each subsystem before integrating these subsystems into one product, testing what worked, and troubleshooting what did not up until the deadline for our project's completion.

2. Problem Statement

2.1 Need

Our project sponsor, recently acquired some land in Mercer County and after a couple of weeks of gardening, quickly realized that he wanted a way to keep track of what was going on with his plants. Currently, to check the soil moisture content of his plants, a gardener would have to walk all the way to his plant and evaluate it physically while still running the risk of over or under watering it [1]. Maintaining the necessary amount of water within a plant's soil is essential for its growth and survival. If there is not enough water for a plant, nutrients cannot travel through the plant, starving it of what it needs to thrive [1]. If plants are watered too much, the roots can rot, causing the plant to receive an insufficient amount of oxygen from the soil and potentially killing it [1, 2]. With our wirelessly connected outdoor soil moisture sensor, home gardeners or hobbyists will be able to conveniently and reliably check whether they need to water their plants without having to leave their house or significantly invest in an industrial system. Utilization of this product would reduce the amount of time necessary for checking soil moisture physically, conserve wastewater by notifying the user when over watering, ensure plants have sufficient soil moisture, and enables users to improve performance of their gardens and save time by tracking and predicting moisture retention trends using acquired data.

2.2 Objective

The objective of this project was to design a wirelessly connected outdoor soil moisture sensor that measures relative soil moisture and reports the data to the user from a central hub. These sensors were designed for deployment in a wide range of environments including an outdoor field or potted plant with the intention of creating a distributed network of wireless soil moisture sensors for ranged moisture tracking. Measurements, such as the ambient air temperature and soil moisture, were intended to be remotely available for analysis and easy to understand for a home gardener. Each device is powered by a primary battery to facilitate year-round data collection without the limitation of being connected to the local electrical grid. To survive outdoors, the device's electronics are enclosed and waterproofed so that water cannot disable the device through ingress. Since there are a large range of already available solutions for data logging and visualization, the design team focused on the development of wirelessly connected soil moisture sensing devices rather than that of the central hub. By the end of this project, we intended to have a reliable, easily reproducible, and effective wirelessly connected outdoor soil moisture sensor for the approximation and transmission of local soil moisture levels.

2.3 Background

Common methods for checking soil moisture require either physical interaction like picking the pot up or sticking a finger in [1]. Widely available soil moisture sensors typically measure using electrical contacts and display the moisture content on a small screen where it is deployed [2]. These sensors measure soil moisture resistively and as a result, suffer from corrosion if left within plant pots or gardens for an extended period and have limited wired range. Numerous papers have presented an alternative capacitive method of soil moisture sensing that relies on the fringe effect to identify changes in permittivity within the surrounding medium or surrounding soil [3, 4, 5, 6]. While these technologies have been fleshed out for academic purposes, some even with wireless transmission [6], their implementation in the commercial market is limited to products such as the Adafruit STEMMA sensor [7], DFRobot Gravity: Analog Capacitive Moisture sensor [8], and the Grove Capacitive Moisture sensor [9]. These commercially available sensors use wires to communicate data values, which significantly limit their range; are not waterproof, so they cannot easily be utilized for outdoor gardens or farms; and are designed to interface with microcontrollers, making them difficult to use or interpret for those who are not tech savvy.

3. Requirements Specification

3.1 Marketing Requirements

The following are *Marketing Requirements* (MR) derived from a meeting with our sponsor for the outdoor soil sensor to fulfill the need of home gardeners:

1. The device shall sense the relative moisture content of surrounding soil.
2. The device shall hourly measure and wirelessly transmit all measurements.
3. Reported data shall be remotely accessible and easy to interpret for a home gardener.
4. All measurements shall be consistent and reliable.
5. The device shall have long battery life.
6. The device shall monitor and report battery life status.
7. The device shall be waterproof.
8. The device should be compact.
9. The device should be low cost and easily replaceable.
10. The device should measure temperature.
11. Multiple devices should be supported for wireless connection.
12. The device's soil moisture sensing shall be able to be recalibrated for varying environments.

3.2 Objective Tree

Our project's objective tree, shown in Figure 1, resulted from the translation and compartmentalization of the marketing requirements to allow for relative importance weighting, task to task. The marketing requirements for the soil moisture sensor were first divided into four major categories – data collection, accessibility, data quality, and longevity - that make up the top level of the objective tree. Analytical Hierarchy Process (AHP) decision matrices, shown and described in greater detail within Appendix 9.1, were used to determine branch weights by comparing the importance of categories and grouped aspects to one another. Throughout this in-depth objective tree explanation, categories will be described as first, second, third, and fourth assuming the reader will move left to right across the objective tree.

The top branch of the objective tree was allocated relative importance weightings using the comparisons made through Table 18 within the Appendix. As shown in the objective tree in Figure 1 and Table 18, data collection and accessibility are the two most important categories for our project. Specifically, they contain the two main focuses of the project – soil moisture sensing and wireless accessibility – the categories themselves are of equal importance because therein lies the hardware design basis upon which the entire project is built upon. If the product fails to sense relative soil moisture and allow for remote access to this data, then we have failed at the most basic level while the data quality and longevity categories can be altered through alteration of the components and device's enclosure. The longevity category is second in terms of priority because of the focus of the sponsor and intended audience on leaving the moisture sensor outdoors or in soil for extended periods of time. Without a long battery life or waterproofing, it would have to be replaced often due to breaking or dying quickly. The data quality category is the lowest priority because compared to longevity, accessibility, and data collection, it does not contain aspects that are necessary for the device to function and can be improved by altering the device's sensors.

The first category, data collection, is divided into 4 aspects whose relative importance weights were made using AHP Table 19 in the Appendix: soil moisture sensing, derived from marketing requirement 1; battery life sensing, derived from marketing requirement 6; temperature measurement, derived from marketing requirement 10; and soil recalibration, derived from marketing requirement 12. Soil moisture sensing is the highest priority because it is crucial for effective implementation of the intended device. Battery life sensing follows in priority because it is necessary to indicate to the user when to replace the device and ensure continued measurements. Soil recalibration is third in priority for this section because it isn't necessary for deployment into varying types of soil but rather due to differences in proximal water's complex permittivity, the main electrical property used for capacitive soil sensing [3, 4, 5, 6]. Last in this branch's priorities is temperature measurement as it does not factor into the essential functionality of the device, sensing soil moisture, and was suggested by the sponsor because of the inexpensive nature of temperature sensors.

The second category, accessibility, is divided into 3 aspects whose relative importance weights were made using AHP Table 20 in the Appendix: wireless transmission, derived from marketing requirement 2; remote ease of use, derived from marketing requirement 3; and multiple device support, derived from marketing requirement 11. Wireless transmission refers to the transmission of all measured data to a remote hub and remote ease of use refers to the easy access to this data for the end user. In terms of priority, wireless transmission was the highest followed by remote ease of use and multiple device support. While remote ease of use is essential to satiate the need of home gardeners, if the data is wirelessly transmitted, its interpretation can be altered to make it more user-friendly and easier to access. Lowest in priority was multiple device support which is not essential for our product solution but could also be altered after the hardware design has been completed if device transmissions can be separately addressed. Both remote ease of use and multiple device support are not design factors for the device solution but rather the hub that it transmits information to which was specifically addressed as not a focus by the sponsor due to the number of off-the-shelf solutions available.

The third category, data quality, is divided into 3 aspects whose relative importance weights were made using AHP Table 21 in the Appendix: consistency, derived from marketing requirement 4; sampling rate, derived from marketing requirement 2; and inexpensive, derived from marketing requirement 9. Consistency refers to consistent and reliable measurements, sampling rate refers to the periodic time when measurements are taken and transmitted, and inexpensive refers to the cost of the device with majority of the cost coming from the sensors themselves according to our sponsor. Consistency was ranked the highest because if the measurements cannot be trusted reliably, the device's effectiveness is significantly reduced for the user as they must verify moisture physically or risk killing the plant. Sampling rate follows in priority because soil moisture and battery life are likely to change very little hour by hour but could be useful for tracking and predicting soil moisture trends. Inexpensive cost of the device's sensors is lowest in priority because increases to the sampling rate and consistency at higher costs could drastically increase effectiveness of the device.

The fourth and final category, longevity, is divided into 3 aspects whose relative importance weights were made using AHP Table 22 in the Appendix: battery life, derived from marketing requirement 5; waterproof, derived from marketing requirement 7; and compact, derived from marketing requirement 8. Compact means the size of the device should be as small as possible to allow for minimal physical obstruction and is in this category because if the device is too large, it could inhibit growth of the plant. Battery life and waterproofing are of the same priority because they both contribute to the essential functionality of the device. If splashes of water destroy the integrity of the device and its capability to perform, then it cannot be left in a pot or garden. Likewise, if the battery dies shortly after being put into the pot or garden, it necessitates frequent physical interaction. Compared to these two aspects, being compact is of a lower priority because -unless larger than the plant or pot- the device's size will not harm its essential functionality or usage. However, the size of the device is largely dependent on the size of the battery which will most likely be one of the largest parts of the system and the method of waterproofing due to the enclosure created to protect the electronics, battery included.

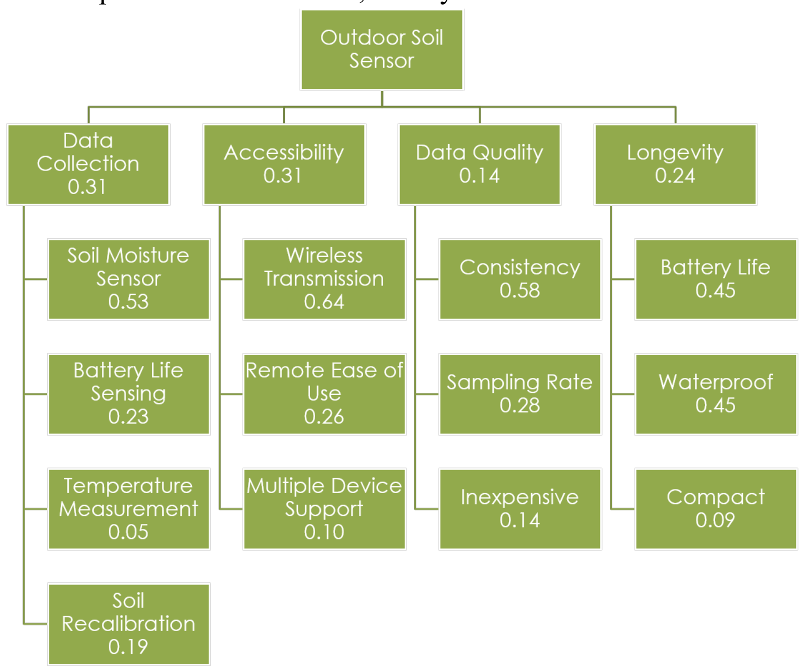


Figure 1: Objective Tree for Outdoor Soil Sensor

3.3 Engineering Requirements

Each marketing requirement for the outdoor soil sensor was broken down into technical *Engineering Requirements* (ER) that are *Abstract,* specifying what the system will do, *Verifiable,* able to be demonstrated or measured, *Unambiguous,* having a single meaning and with short complete sentences, and *Traceable,* able to be traced back to marketing requirements. For each numbered engineering requirement in Table 1, there is a rationale or marketing requirement origin and a justification or reason for the derivation of the engineering requirement from the rationale. Table 2 correlates each engineering requirement with a strategy for verifying or validating that it has been completed in the final device.

Table 1: Engineering Requirement, Rationale, and Justification

#	Engineering Requirements	Rationale	Justification
1	The device shall measure soil moisture and report soil moisture level.	Based on MR #1 and 3	Soil moisture level should be reported to the user to allow for easier interpretation.
2	Soil moisture readings shall be taken without direct electrical contact to the soil.	Based on MR #1	Exposure to electrical contacts to soil could corrode the sensor prematurely, leading to less accurate readings over time.
3	Soil moisture sensing reliability shall not degrade over time.	Based on MR #1 and 4	The user should not have to second guess or physically verify moisture readings.

4	The device shall wirelessly transmit data over more than 500 meters without line-of-sight.	Based on MR #2	Wireless transmission over more than 500 meters and without line-of-sight was requested by our sponsor.
5	Measurements on the device shall be taken every hour.	Based on MR #2	Taking soil moisture measurements hourly captures the rate of change of soil moisture, determined experimentally. This sampling rate was also requested by our sponsor.
6	The device shall wirelessly transmit all measurements to a central hub.	Based on MR #2	Remote access to outdoor soil sensor measurements is one of the main goals of the project and was requested by our sponsor.
7	Measurements shall be easily accessible and understandable from the central hub.	Based on MR #3	Data needs to be easy to interpret and access for the intended customer, a home gardener, as specified by our sponsor.
8	The device shall track battery parameters to alert the user when the voltage drops below 2.8V prior to device deactivation.	Based on MR #3 and 6	It was important to our sponsor to indicate to the user when the device or the device's battery needs to be swapped out.
9	The device shall be able to measure battery voltage with a resolution of .01 V to predict device end-of-life.	Based on MR #6	A resolution of .01 V in necessary to ensure a timely alert of battery/device end-of-life due to the rapid change in voltage with our selected battery, shown in Figure 19.
10	The device should measure the battery's voltage with a margin of error of a maximum of ± .01V.	Based on MR #4 and 6	Ensuring the margin of error is less than our desired resolution ensures that our end-of-life estimate is reliable to not alert the user too early of device death.
11	The device's microcontroller will enter deep sleep and cut power to all subsystems to conserve power between samplings.	Based on MR #5 and 6	Minimizing current draw between samplings - majority of the device's lifetime- is essential to being able to sustain battery life for 2 years or more.
12	The device shall have a battery life greater than 2 years.	Based on MR #5	A 2-year battery life was specified by our sponsor to ensure replacement is infrequent.
13	The device shall continue working through temperatures ranging from -20ºF to 126ºF.	Based on MR #4	A range of -20ºF and 105ºF covers the most extreme air temperatures recorded in Kentucky [10] with a 20% safety factor on the max due to the increased ground temperature.
14	The device's electronics shall be enveloped within an enclosure.	Based on MR #7	Enclosing the electronics is the requested method of environmental protection from our sponsor.
15	The device's enclosure shall have a waterproofing rating of IP-54 or greater.	Based on MR #7	This standard of waterproofing was requested by the sponsor as the device is intended to be used outdoors.
16	The device shall be no larger than 10" long, 2" wide, and 5" deep.	Based on MR #8	The maximum size constraint for our device was created by analyzing other PCB capacitive moisture sensors and estimating the space needed for an attached battery [7, 8, 9].
17	The total parts cost for one device shall not exceed $30.	Based on MR #9	Our sponsor proposed this budget within the original project proposal after perusing commercially available alternatives.
18	Costs for the entire system shall not exceed $500	Based on MR #9	Our sponsor proposed this budget within the original project proposal.

19	The device's electrical components shall be off-the-shelf or available online.	Based on MR #9	Ensures that the device can be reproduced at large scale by confirming component availability.
20	The device should be able to measure air temperature with a resolution of at least 1ºF.	Based on MR #10	Resolutions higher than 1ºF are not necessary to determine whether the local area is beneath freezing or 32ºF.
21	The device should be able to measure temperature with a margin of error of a maximum of ± 1º F.	Based on MR #4 and 10	Error values larger than ± 1º F are not trusted to accurately predict whether the surrounding environment is beneath freezing or 32ºF.
22	Multiple devices should be supported for simultaneous usage.	Based on MR #11	Enables the user to utilize multiple devices with the same hub for centralized access to multiple soil moisture data points.
23	The device shall be able to be recalibrated to report relative soil moisture level.	Based on MR #12	Ensures that soil moisture values can be used to determine the level of moisture relative to the calibrated end points of air and water readings.

Table 2: Engineering Requirement Verification

#	Engineering Requirements	Verification and Validation
1	The device shall measure soil moisture and report soil moisture level.	Tested by leaving the wireless device in soil and varying the moisture content to verify correct soil moisture level calibration.
2	Soil moisture readings shall be taken without direct electrical contact to the soil	Verified by dipping the device into water and air during initial calibration and recording the measured capacitance values.
3	Soil moisture sensing reliability shall not degrade over time.	Tested using 5 trials leaving the device in soils with varying levels of moisture content for multiple days and analyzing the resulting trend line
4	The device shall wirelessly transmit data over more than 500 meters without line-of-sight.	Tested by taking the device out of line-of-sight 501m away from the central hub and verifying data values are still received.
5	Measurements on the device shall be taken every hour.	Tested using a software defined testing mode that records and sends a time value alongside other measurements every minute rather than every hour and comparing the time of reception at the hub. To ensure measurements are varying in all sensors, soil moisture and temperature will be changed between samplings.
6	The device shall wirelessly transmit all measurements to a central hub.	Tested through wireless transmission to the central hub using a software-defined testing mode.
7	Measurements shall be easily accessible and understandable from the central hub.	Tested by introducing the device to someone who is not tech savvy and verifying that they are able to effectively use it.
8	The device shall track battery parameters to alert the user when the voltage drops below 2.8V prior to device deactivation.	Tested using a partially discharged battery, connecting it to the circuit, and verifying that the battery life indicator is sent to the user when it falls below the voltage threshold while in rapid sensing mode.

9	The device shall be able to measure battery voltage with a resolution of .01 V to predict device end-of-life.	Tested using an external digital multimeter across the battery connections and comparing that to what is measured by the device.
10	The device should measure the battery's voltage with a margin of error of a maximum of ± .01V.	Tested using an external digital multimeter across the battery connections and comparing that to what is measured by the device at 10 different voltages and calculating the average error.
11	The device's microcontroller will enter deep sleep and cut power to all subsystems to conserve power between samplings.	Tested by putting the microcontroller into deep sleep and disconnecting other subsystems while estimating the current draw of the device and comparing that to when everything is activated.
12	The device shall have a battery life greater than 2 years.	Tested by measuring the current draw through one hour of operation and verifying the energy storage system will be able to supply that amount of current while sampling for 2 years through a computational estimate.
13	The device shall continue working through temperatures ranging from -20ºF to 126ºF.	Verified through waterproofing method, enclosure material selection, and electrical component datasheet analysis to confirm structural integrity at the extremes and in between.
14	The device's electronics shall be enveloped within an enclosure.	Verified through visual inspection of the final device.
15	The device's enclosure shall have a waterproofing rating of IP-54 or greater.	Tested by splashing the deactivated device with water from all angles and covering it in sand in accordance with its IP-54 rating (detailed in Appendix 5.2). Verify that water or sand was unable to move into the device's enclosure.
16	The device shall be no larger than 10" long, 2" wide, and 5" deep.	Verified through direct measurement of the final device, using a ruler.
17	The total parts cost for one device shall not exceed $30.	Verified by summing the total cost of all parts for the deployed soil moisture sensor to ensure the sum is less than $30.
18	Costs for the entire system shall not exceed $500	Summing total cost prior to construction of the final prototype system to ensure the sum is less than $500.
19	The device's electrical components shall be off-the-shelf or available online.	Verified through inspection of the origin of components for the device.
20	The device should be able to measure air temperature with a resolution of at least 1ºF.	Verified through component selection and external thermometer testing at 5 different temperatures to compare with the device's own measurements.
21	The device should be able to measure temperature with a margin of error of a maximum of ± 1º F.	Tested at 5 different temperatures and comparing with an external thermometer's measurements.
22	Multiple devices should be supported for simultaneous usage.	Verified through hub and wireless radio selection, creation of a second device, and testing the functionality of multiple devices and the hub.
23	The device shall be able to be recalibrated to report relative soil moisture levels.	Verified through initial calibration of the device using soil moisture sensor values while inserted in water and air and comparing to values from another water sample with added fertilizer.

3.4 Broader Impact

3.4.1 Economic Impact

The outdoor wireless soil moisture sensor could improve farmer's bottom line by reducing overconsumption of water, as described in the environmental impact, and allowing for more efficient utilization of time. According to the EPA, soil moisture-based control technologies allow for more efficient watering of plants by tailoring the irrigation schedule to according to the moisture content of the soil [11]. Instead of having to physically walk and check moisture across plots of plants in a large farm, a farmer could save time by checking the moisture from a central hub and planning to maximize irrigation efficiency. Increasing irrigation efficiency reduces the overall cost spent on water and could increase yields or returns on plants that would have faltered or died otherwise due to overwatering. The opportunity cost of physically travelling and checking moisture would be eliminated by using our product, allowing farmers or home gardeners to perform other tasks to increase yields or make money through other methods. Two of the marketing requirements for the device involve being inexpensive and easily reproducible. Shifts in the economy could increase the price of electrical components and our custom PCBs, upsetting our design's inexpensive target. If the electrical components used are uncommon or custom made, their production could cease which would reduce reproducibility or possibly force a total redesign of the electrical system after passing our work to our sponsor.

3.4.2 Environmental Impact

The outdoor soil sensor will impact the environment by reducing wastewater caused by overwatering plants [2]. Since overwatering is one of the most common causes of plant death, this could serve to further water conservation efforts and save plants from being killed due to a lack of available oxygen [1, 2, 11]. Enabling users to collect data and use it to track trends in soil moisture retention allows for more efficient deployment of water when the plant really needs it, reducing the likelihood of overwatering [2]. Decreasing water consumption within environments where water is scarce and precious, like deserts or within space, through the usage of soil moisture sensing units could greatly impact the survivability of denizens. The deployment of our soil moisture sensor within varying environments also constrains or defines some of the necessary functionalities for the device. Exposure to the elements requires a waterproofed enclosure to defend the electronics used for sensing and transmission of measured data against rainfall, splashing, and condensation. Differences in soil and water content directly impact the effectiveness of the device as they alter complex soil permittivity, the electrical property used to capacitively sense soil moisture, necessitating recalibration of the soil moisture sensing. As described within the health and safety impact, potential pollution or littering of sensors, leaving them forgotten to decay in the ground, would also be detrimental to the local environment so we indicate to the user whenever the device needs to be replaced [12].

3.4.3 Manufacturing Impact

Manufacturing capability affects our design because it determines whether our intended designed PCB and electronic enclosure can be created. For example, there are limitations on the maximum amount of copper traces or layers on PCBs, reducing the maximum amount of current available to the device and increases in copper width or layers which increase current carrying capacity drastically increases the price of the PCB. If larger amounts of current passed through the device increased the sensitivity of soil moisture measurements, there would only be so much that could be done to act upon this discovery. Manufacturing tolerances concerning the construction of the physical enclosure could drastically affect our waterproof capabilities. If a method of enclosing the electronics with 3D printing was developed but had too much of a gap or error in the fit between the enclosure and the PCB, water and particles could negatively affect the device's performance. The cost of manufacturing also effect's our product solution's design as the total cost of each sensor should not exceed thirty dollars, according to our sponsor. This means that during design and prototype creation, parts had to be picked to reduce cost without sacrificing essential functionality. Keeping the cost of additional functionality and the total cost limitation in mind helped us make decisions on which optional features will be added to the device. As mentioned in our economic impact, manufactured electrical component availability also determines what can be utilized for our electrical components.

3.4.4 Standards Impact

Current standards will have an impact on the implementation of the outdoor soil sensor product solution. These include standards regarding the construction of the enclosure for the electronics (IP-54) and the general design of Printed Circuit Boards (IPC-2221). IP-54 is an environmental protection rating that we were instructed to prepare our product for due to the surrounding environmental conditions when placed within the soil, as discussed in the environmental impact section and in the Appendix 9.2 [13]. Specifically, it means our device must provide a high level of protection against particles such as dirt and a moderate level of protection against water damage or splashes aimed towards the device. Therefore, an enclosure was created for the electronics with a plan on verifying using the IP-54 test, splashing the device and water and sand while off to see how much breaches into the container. IPC-2221 or the "Generic Standard on Printed Circuit Board Design" created by IPC, formerly known as the Institute for Printed Circuits, is a document that details the thermal, electrical, and mechanical properties of PCBs and the limitations or requirements that must be made to account for them. These limitations have been adapted into Design Rule Checks for PCB design software like Autodesk Eagle and KiCad EDA to ensure that digitally designed PCBs can be physically constructed by PCB mills or manufacturers.

3.4.5 Health and Safety Impact

The health and safety impact of our device must be considered due to the potential impact on local crops and humans through groundwater and soil pollution. If our device is improperly waterproofed or forgotten in the ground, the batteries could corrode, soaking chemicals into the nearby soil and contaminating ground water [12]. Depending on the chemical composition of the battery, if malfunctioning to the point of thermal runoff, the battery could combust, causing a fire that destroys nearby plant life, or vent, releasing toxic chemicals into the atmosphere. Both could negatively impact local human health through groundwater pollution as numerous toxic metals and corrosive acids used in batteries can cause burns to the eyes and skin or are carcinogens increasing the likelihood of developing cancer [12]. For these reasons, we need to we verify the functionality of the battery life sensing and take all necessary steps to protect interaction between the battery and the local environment. To ensure that our device is not left too long after deactivation to degrade into electronic waster, we will notify the user when the device's battery has almost run out or within 5% of end-of-life or not functioning.

4. Design

The following sections detail the design steps taken to plan and actualize our product prototype for an outdoor wireless soil moisture sensor. First, we performed a functional decomposition to discern the systems that would be needed to provide the intended functionality described within our marketing requirements. With these details in mind, we made final selections on the components that were used to create and test our soil moisture sensor prototypes.

4.1 Functional Decomposition

Functional Decomposition (FD) is a recursive process that iteratively describes the functionality of all system components using the inputs, outputs, and transformations in between. The Top-Down analysis performed within this section focuses on the overall vision or functional requirements of what the final system will do and breaks it down to the subsystem or component level. Figure 2 represents Level 0 of our functional decomposition or the most abstract functional requirements for the system whose inputs, outputs, and functionality are described in Table 3. Figure 3 is the 1st level of the functional decomposition or main design architecture of the system and breaks down the functional requirements into the deployable soil moisture sensor subsystem used for measurements and the central hub subsystem intended to deliver outputs to the user which are described in further detail in Table 4.

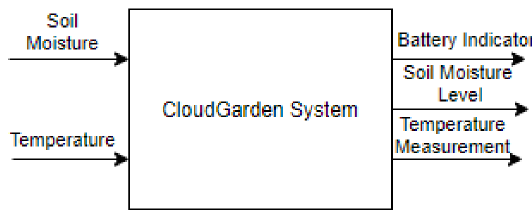


Figure 2: Level 0 – CloudGarden System

Table 3: CloudGarden Level 0 Functional Decomposition Explanation

Module	**CloudGarden Wireless Soil Moisture System**
Inputs	• Soil moisture content is approximated. • Temperature of the surrounding air is measured.
Outputs	• Soil moisture level at the device's location is available to the user. • Ambient temperature at the device's location is available to the user. • A battery life indicator is displayed when voltage falls below voltage threshold.
Functionality	Outdoor wireless device enables hobbyists and home gardeners to monitor soil moisture and temperature remotely at a centralized location.

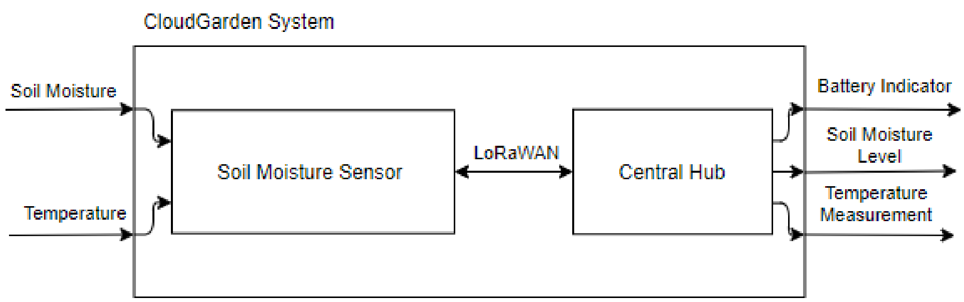


Figure 3: Level 1- CloudGarden Functional Decomposition

Table 4: CloudGarden Level 1 Functional Decomposition Explanation

Module	**Soil Moisture Sensor and Central Hub**
Inputs	• Soil Moisture: Surrounding soil moisture is measured by the device using changes in the fringing field produced by a capacitive region of the PCB. • Temperature: Temperature of the surrounding air is measured by the device within a range of -20ºF to 126ºF.
Outputs	• Battery Indicator: A battery end-of-life indicator is delivered to the user to notify when the device's battery is within 5% of being empty/below 2.8V. • Soil Moisture Level: Relative soil moisture levels are available to the user telling them whether the plant's surrounding soil is wet or dry. • Temperature: Temperature in Fahrenheit is given to the user based on surrounding air temperature within a range of -20ºF to 126ºF.
Functionality	Measures surrounding soil moisture and ambient temperature every hour, and converts the data into an easily readable form, and presents it to the user. Battery energy consumption is monitored with an alert sent to the user when voltage dips below 2.8V.

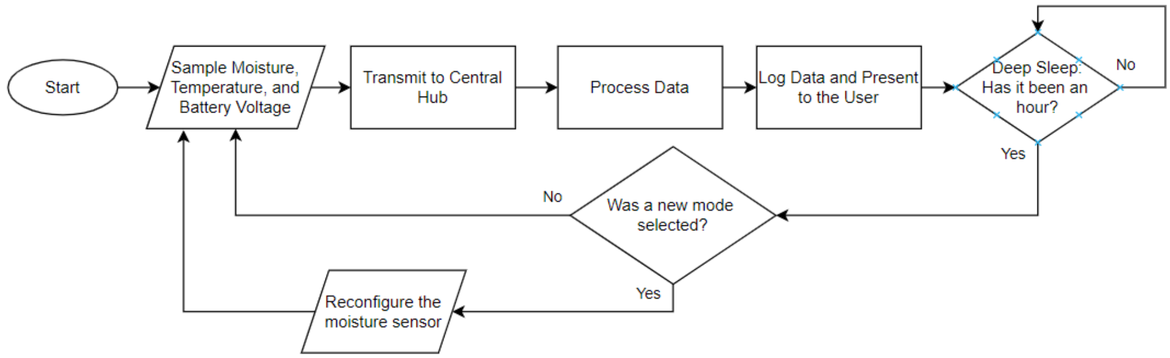


Figure 4: CloudGarden Behavior Model

The behavior model shown in Figure 4 represents our system's software-defined behavior. At the beginning of every sampling cycle, it samples surrounding soil moisture, ambient air temperature, and battery voltage. Using the ADCs, the microcontroller will read the voltage inputs, translate the measurements into byte-wise encoded messages before transmitting them to the central hub for processing to conserve device-side power. The central hub processes this data within a database using conversion equations necessary for each method of measurement and logs it for easy access on a web server. While waiting between hour samplings, the device will go into deep sleep and turn off parts of the circuit that consume power. Upon the completion of an hour, the device will repeat the above processes and check to see if the Central Hub has ordered the change to a new operating mode. While the default mode is to sample every hour, other modes change the time between samples on the device to allow for soil moisture calibration or sensor validation.

4.1.1 Level 2 Functional Decomposition

Figure 5 is the decomposition of the deployable outdoor soil moisture sensor, named the *Dirt Dock,* with the subsystems necessary for the input, processing, and transmission of measurements from the deployed outdoor device to the central hub. As detailed in Table 5, the capacitive moisture sensing and temperature sensing subsystems are powered by the microcontroller's Digital I/O pins to allow for easy control and measure the surrounding temperature and moisture. To ensure that the user is alerted when the device's battery is dead, battery voltage is measured internally using the internal bandgap as a reference. All three values are converted from analog voltages to digital signals that are compiled within a wireless message and sent across LoRaWAN communication to the central hub. Mode control input from the central hub to the microcontroller changes the period between device samplings to allow for soil moisture calibration or sensor validation.

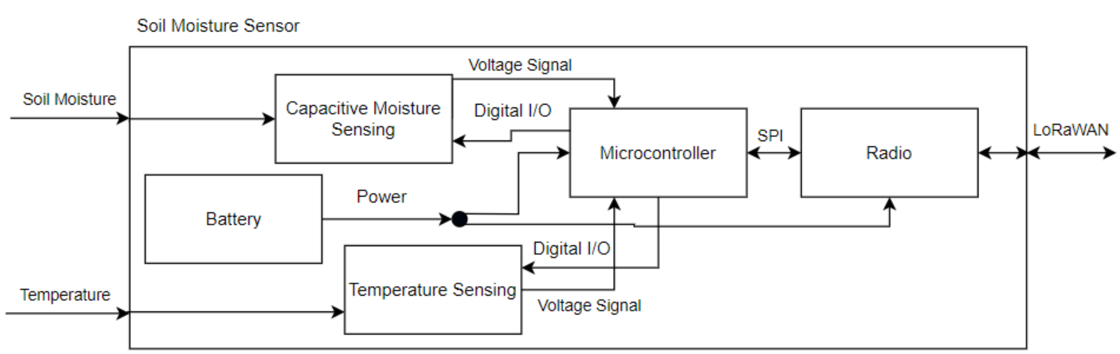


Figure 5: Level 2- Soil Moisture Sensor Functional Decomposition

Table 5: Soil Moisture Sensor Level 2 Functional Decomposition Explanation

Module	**Soil Moisture Sensor**
Inputs	• Soil Moisture is measured via the Capacitive Moisture Sensing Block and sends a voltage to the microcontroller. • Temperature is measured via the Temperature Sensing Block and sends a voltage to the microcontroller.
Outputs	• All 3 measurements are transmitted to the central hub via LoRaWAN. • Mode Control input via LoRaWAN is used to calibrate the system, deactivate it remotely, or undergo rapid sensing testing.
Functionality	The outdoor soil moisture sensor device is battery powered and hourly measures the soil moisture and temperature surrounding the device to send its readings to the hub device.

The Central Hub subsystem revolves around the logging, processing, and display of measurements taken from devices to the user at the hub. Figure 6 is the functional decomposition for this subsystem that is further described in Table 6. This system consists of three parts: a radio system for handling or initiating communication, a database for logging and processing data, and a web server to distribute information locally to users. The web server provides user input in the form of a mode select and allows access to the measurements captured by individual devices within a webpage in the local network. Mode select messages can be sent to the device tell it to sleep for different periods, allowing higher sampling rates for initial calibration or sensor validation. A special mode is also available that stores and transmits time when sensing to validate period timing. The central hub is intended to be deployed in the gardener's house, protected from the elements, connected to a local wireless network, and powered by a grid connection.

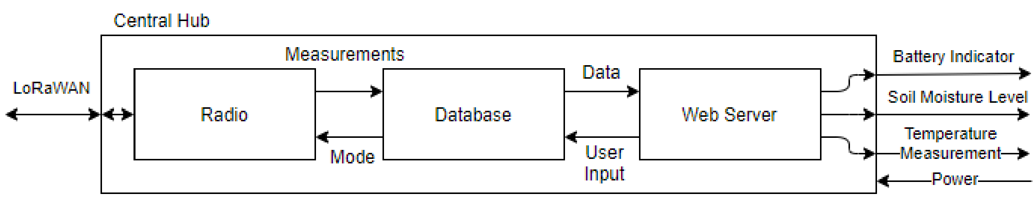


Figure 6: Level 2- Hub Functional Decomposition

Table 6: Central Hub Level 2 Functional Decomposition Explanation

Module	**Central Hub**
Inputs	• Measurements: Battery energy consumption and temperature and soil moisture measurements are transmitted via LoRaWAN, processed by the webserver, and logged by the central hub's database. • Power: The system is powered through a connection to the grid
Outputs	• Soil Moisture Level: Soil moisture level is presented based on the initial calibration of water and air capacitance values provided before long-term installation and segmenting the range in between the two values. • Temperature: Temperature in Fahrenheit is displayed using the Stein-Hart equation for the NTC thermistor • Mode Select: The user can activate several modes for device's operations via transmissions to the device via LoRaWAN that change device sampling rate and output. • Battery Indicator: When the battery voltage falls below 2.8V, the user is alerted to a low battery.

Functionality	Processing, data logging, and user presentation/interfacing is handled by the central hub to allow easy access and interpretation of data. Information is accessed through the localized wireless network and can be retrieved for multiple time points.

4.1.2 Level 3 Functional Decomposition

The wireless Radio subsystem is described in the functional decomposition shown in Figure 7 and focuses on sending and receiving data between the Dirt Docks and Central Hub. Communication between the microcontroller and wireless transceiver is facilitated through Serial Peripheral Interface (SPI) protocol allowing for commands regarding mode and the transfer of data to and from the device. The transceiver encodes or decodes data through the LoRaWAN transmission protocol for long-range low-power wireless transmission. Encoded data is sent through an antenna which radiates energy by drawing current or is received through an antenna by intercepting radio wave power to produce current. As described in Table 7, the block both receives and sends data and wireless messages to and from the device, establishing wireless communication.

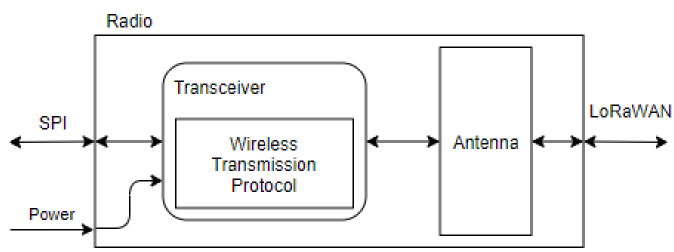


Figure 7: Level 3- Radio Functional Decomposition

Table 7: Radio Level 3 Functional Decomposition

Module	**Radio**
Inputs	• SPI: Battery, soil moisture, and temperature sensing results are transmitted to the transceiver alongside commands for transmission and sleep mode. • LoRaWAN: Mode select is transmitted from the central hub to the device. • Power: Input 3.6V power comes directly from the battery output.
Outputs	• LoRaWAN: Measurement values are wirelessly transmitted to the central hub via LoRaWAN transmissions. • SPI: Mode select information is decoded and sent to the microcontroller through a communication bus.
Functionality	The radio system handles wireless communication between individual devices and the central hub device by encoding input data and transmitting it via modulation of radio waves through an antenna.

The Capacitive Moisture Sensing functional decomposition is shown in Figure 8 and described in the functionality section of Table 8. a waveform generator provides 619 kHz square waves to an RC filter containing the variable PCB capacitor whose capacitance varies with changes in soil moisture. The output analog voltage signal is converted to a digital value by the microcontroller mapped between a segmented range of calibrated water and air endpoints to create soil moisture levels.

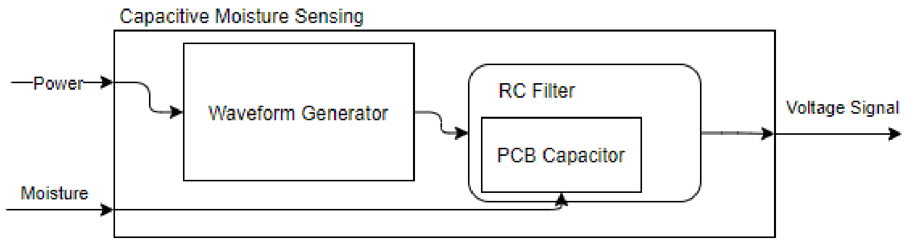


Figure 8: Level 3- Capacitive Moisture Sensing Functional Decomposition

Table 8: Capacitive Moisture Sensing Level 3 Functional Decomposition Explanation

Module	**Capacitive Moisture Sensing**
Inputs	• Power: 3.6V powers a waveform generator circuit which passes a signal through a lowpass RC filter containing the PCB capacitor • Moisture: Moisture content of the surrounding soil modifies the capacitance of the PCB depending on its ionic and chemical composition
Outputs	• Voltage Signal: The analog voltage output varies proportional to changes in impedance caused by soil moisture impacts on the capacitor's fringing field within the capacitive portion of the RC lowpass filter.
Functionality	The PCB capacitor's capacitance value varies depending on the complex permittivity of the surrounding medium, altering the value of a low pass filter. Passing a waveform through the low pass filter results changes the output voltage which can be read by the microcontroller's ADC.

The Microcontroller functional decomposition, depicted in Figure 9, and described in Table 9 converts analog voltages into digital signals from sensing subsystems, powering sensing subsystems, transmitting data to and from the radio, and changing modes depending on received mode select messages.

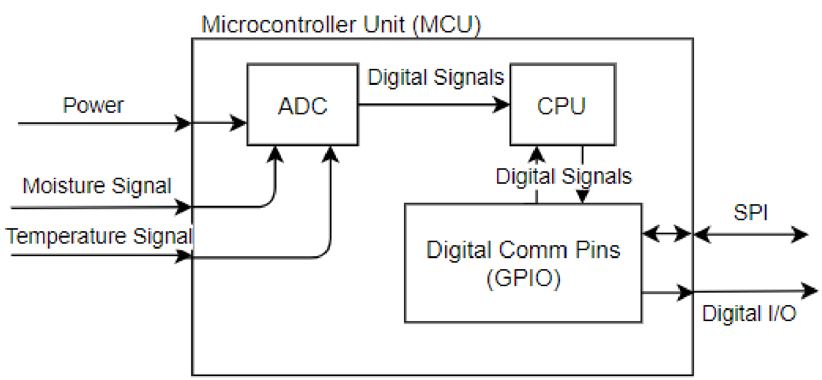


Figure 9: Level 3- MCU Functional Decomposition

Table 9: MCU Level 3 Functional Decomposition Explanation

Module	**Microcontroller Unit**
Inputs	• Power: 3.6V power powers the microcontroller and is always being supplied to power either deep sleep mode with sensors off or to power the entire device. • Moisture Analog Signal: Voltage input from the capacitive moisture sensor is read by the microcontroller's ADC and converted to a digital signal. • Temperature Analog Signal: Voltage input from the temperature sensor is read by the microcontroller's ADC and converted to a digital signal.

	• SPI: Digital messages sent from the central hub allow for activation of various software defined modes of operation
Outputs	• SPI: Soil moisture, temperature, and battery measurements are encoded for transmission and sent to the transceiver through SPI. • Digital I/O: The external temperature sensing and soil moisture sensing circuits are deactivated by pulling the respective digital outputs low
Functionality	The microcontroller device handles all inputs from the other sensors and transmits them to the transceiver for wireless transmission. It also toggles temperature and soil moisture sensing via Digital I/O.

Ambient air temperature sensing's functional decomposition is shown in Figure 10 and described in Table 10. The temperature sensing circuit measures ambient temperature using an NTC thermistor as part of a voltage divider so that the voltage output from the system changes depending on the thermistor's value.

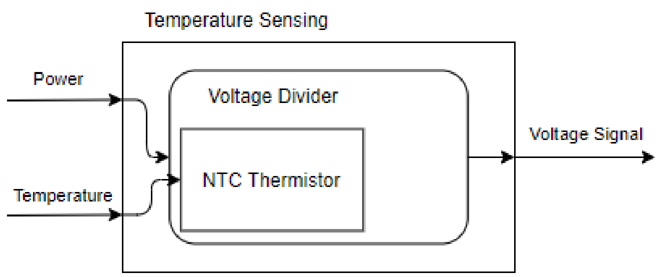


Figure 10: Level 3- Temperature Sensing Functional Decomposition

Table 10: Temperature Sensing Level 3 Functional Decomposition Explanation

Module	**Temperature Sensing**
Inputs	• Power: 3.6V is divided across the NTC thermistor and a 10kOhm resistor. • Temperature: As the temperature increases, the resistance of the NTC thermistor decreases which reduces the voltage across the thermistor.
Outputs	• Voltage Signal: The voltage output to the microcontroller across the thermistor varies as the resistance of the thermistor varies.
Functionality	Within the voltage divider, the resistance of the NTC thermistor reduces as the temperature changes, changing the output voltage of the system. The microcontroller's ADC can read the analog value which can later be converted to a Fahrenheit temperature using the Stein-Hart equation [14, 15].

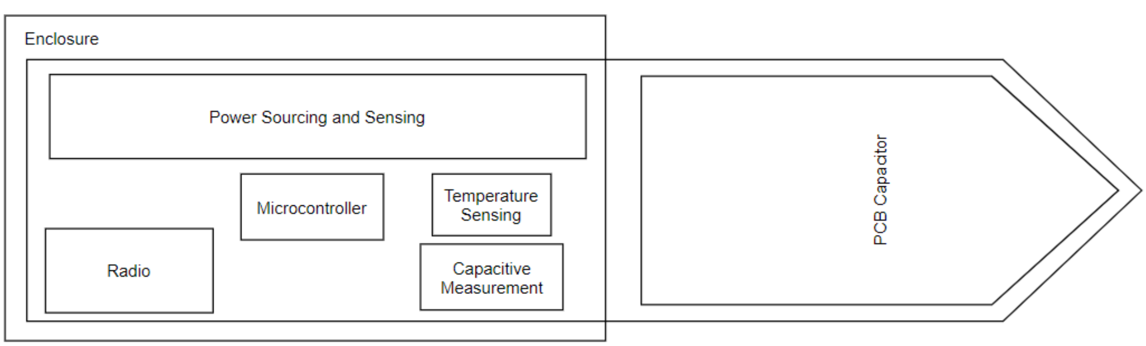


Figure 11: Mechanical Assembly and Enclosure Abstract Diagram

Figure 11 is an abstract drawing showing a preliminary design for the mechanical assembly for the deployed device as a top-heavy stake. This included the electronics at the top of the stake with an enclosure and waterproofing applied to all exposed ports of the enclosure. The battery will be attached to the back of the PCB, defining its horizontal length. Water proofing is essential for effective protection of the electronics as water entering the device could corrode the circuitry or short parts of the battery and is not necessary for the PCB capacitor section as it is not susceptible to water damage. The only parts of the device that are outside of the enclosure are the antenna to maximize transmission range of LoRaWAN communication.

4.2 Design Summary

The following sections summarize the designs and steps taken to turn the functional decomposition systems into reality within our final prototype.

4.2.1 Wireless Data Transmission

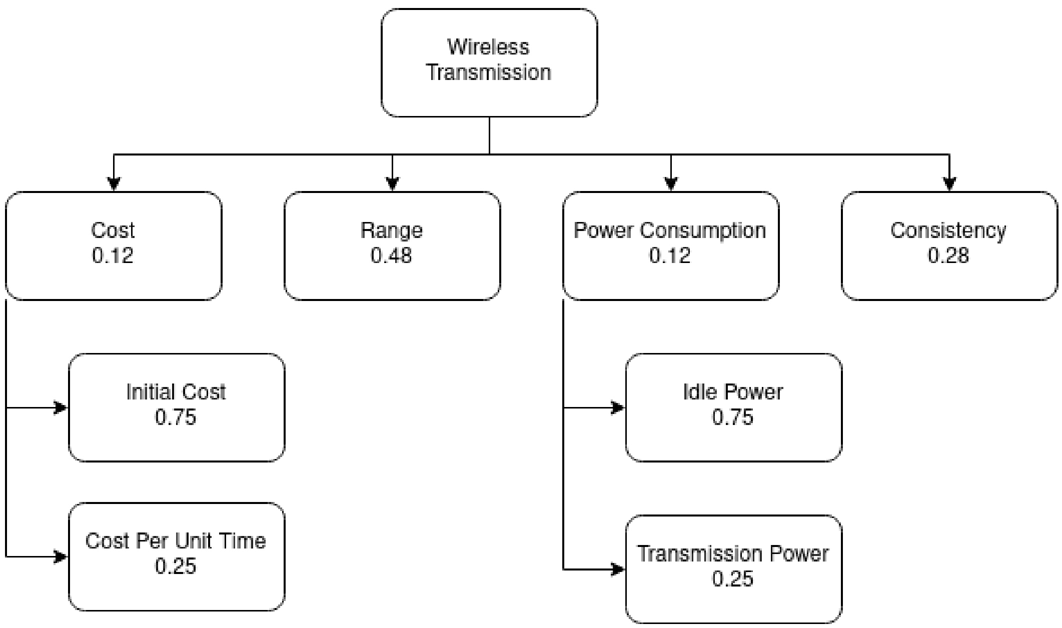


Figure 12: Wireless Transmission Protocol Objective Tree

Using the relative weights described in Figure 12 and derived through Table 23, Table 24, and Table 25 in the appendix, LoRaWAN is the best option for our wireless transmission protocol. As a transmission protocol, LoRaWAN is growing in popularity as a robust bi-directional communication network designed with IoT data acquisition and deployment in mind. The range of NB IoT and LoRaWAN transmission systems are approximately equivalent as they both satisfy the minimum range requirement of 500 meters and are known to be consistent and have security protocols in place to protect transmitted information. NB IoT systems require monthly subscriptions for utilization of the licensed spectrum whereas LoRaWAN systems are free to operate after the initial purchase of a gateway to transmit to and from devices. The consistency of NB IoT is lower than LoRaWAN since we can manage retransmission more easily using the hub. Within a portable device such as ours, which is meant to be left outside for long periods of time, minimizing current draw and power losses is an important factor to our device's lifetime. With half of the current draw of NB IoT systems, LoRaWAN proves to be superior for our purposes due to smaller idle and transmission power consumption and total cost.

Transceivers are necessary for the application of wireless transmission protocols, producing AC radio waveforms that are radiated using an antenna. As described in Table 32, the design alternatives for wireless transmission, transmitters can either be deployed in standalone chips or commercial modules with full application circuits, prebuilt, and pretested. Since the module is produced for easy deployment and connection to the breakout board, its application is tested and known to be effective. While a mass-produced version of this device would be cheaper with a standalone chip mounted onto its surface, each individual chip would cost more buying them one at a time. Power consumption is in favor of the transceiver chip, but the circuitry needed to develop the application circuit for the chip results in an approximately equivalent current draw. Due to the greatly reduced time from acquisition to implementation and reduced cost, a surface mountable LoRaWAN transceiver module is the best option to fulfill our current engineering requirements.

For this purpose, we selected the surface mount RFM69HCW module because it is inexpensive, commonly used with Arduino microcontrollers through SPI communication protocol, capable of hundreds of meters of transmission over open land, and runs off 3.3V, the voltage for the rest of the system [16]. The RFM69HCW transmits on the ISM (Industry Scientific and Medical) band, at 915MHz, as it is legal in North America to freely use for low-power, short-range, license free radios for embedded systems purposes [16]. This module also comes with ultra-low power modes that can be used to enter sleep stated which draw significantly less current than active. We combined example applications from a sparkfun article that detailed out the RFM69HCW should be hooked up to a traditional Arduino [16], shown in Figure 13, and open source hardware diagram/CAD file for an RFM69HCW breakout board also developed by sparkfun [17], shown in Figure 14, to create the application circuit for our transceiver, shown in Figure 15. Both the tutorial and the breakout board are licensed under the Creative commons by ShareAlike 4.0 license allowing for adaptation and sharing if appropriate credit is given and any changes led to distribution of contributions under the same license [18].

RFM69HCW pin	328 (Pro/Mini/Redboard/Uno)	Mega	32U4 (Leonardo/Due/Pro Micro)
O / MISO	12 or ICSP-1	50 or ICSP-1	ICSP-1
I / MOSI	11 or ICSP-4	51 or ICSP-4	ICSP-4
C / SCK	13 or ICSP-3	52 or ICSP-3	ICSP-3
S / NSS	10	53	10
0 / DIO0	2	2	3
3.3V	3.3V (labeled "VCC" on Pro/Mini)		
G / GND	GND		
A / ANT	See the next section for antenna information		

Figure 13: RFM69HCW-Arduino Connection

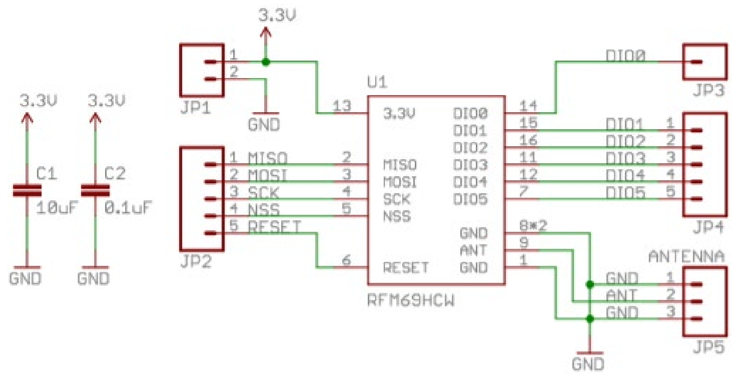


Figure 14: RFM69HCW Breakout Board Example

Antennas are necessary for the modulated transmission of radio waves over long distances from the transceiver. For our purposes, we compared between internal antennas, wires or patterns created within the PCB used to house the capacitive moisture sensor, or external antennas, which provide longer range and can be directionally oriented in the same direction to maximize transmission effectiveness. An external antenna will provide fewer limitations upon waterproofing options as well as greatly reduce the implementation time of radio transmission. An SMA connector will afford the end user the ability to replace the included antenna with a more powerful one if so desired, provided sufficient current can be supplied to it, increasing the number of use cases for the end device. Due to the increased ease of use and transmission range of external antennas, we decided upon using an external antenna attached to an SMA connector soldered onto the PCB.

Figure 15 showcases the final hardware implementation of the RFM69HCW transceiver and antenna circuit for wireless LoRaWAN transmission from the deployable soil moisture sensor devices to the central hub. The transceiver's pinout was translated row by row from Figure 13 with the addition of the two optional power capacitors included in the breakout version of the board, shown in Figure 14, to reduce noise in the voltage input and reduce transient behavior. At the request of our sponsor, we added an optional pi filter output, sourced, and described in further detail in [19], between the antenna, labelled X1, and transceiver board, labelled U3, to allow for impedance matching to maximize effective transmission power and remove voltage ripple. If we decide not to use the pi filter circuit, we can add a 0-ohm resistor or bridge the pads meant for the resistor. A solder jumper was also added, labelled SJ1, to allow for utilization of an external antenna connected through a through hole connection. Object-oriented programming and abstraction will be enabled through the LowPowerLab RFM69HCW library to send and receive messages from the transceiver [20]. This library is covered under the GNU General Public License V3.0, a copyleft license or one where software derived from the source must be freely distributed under the same or similar license [21].

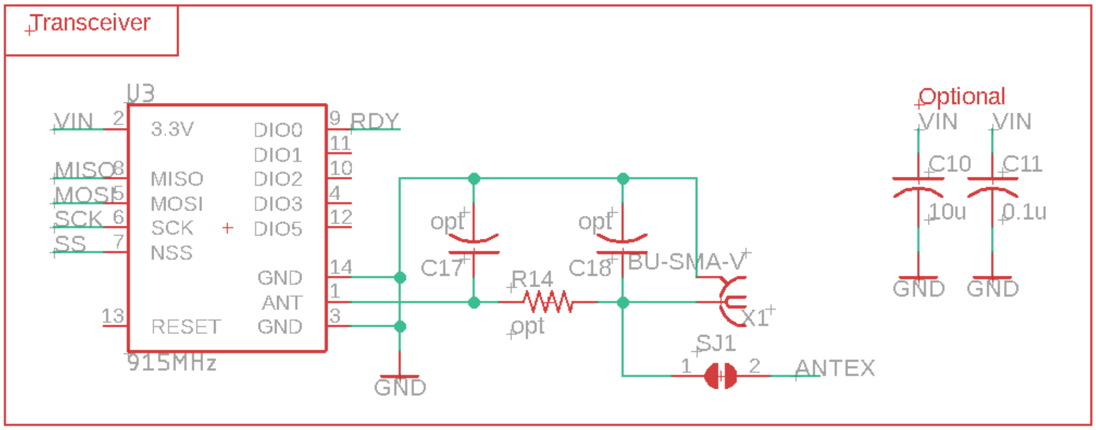


Figure 15: RFM69HCW Transceiver Application Circuit

4.2.2 Hub Processing and Presentation

As specified by the sponsor and described in our marketing requirements, our device needs to deliver all measurements so that they are remotely accessible and easy to interpret for a home gardener. Our solution is to create a central hub that can receive transmissions, store measurements, process them, and output the data in a form that is easy to access and understand by an end user such as a hobbyist or home gardener.

Table 33, the design alternatives for the central hub, describes multiple alternative devices can receive I/O and processing/storing them on a Web Server that will can be accessed by a URL of a device located on the same local network. All three options were compared while keeping the relative importance of device parameters in mind which were determined in Table 26 in the appendix. According to the systematic comparison performed via the AHP, simplicity and processing power were more important than cost or available storage of the device as the budget for the central hub is very large compared to the budget for the deployable devices and the available storage on the device can be easily upgraded. The Raspberry Pi and Linux Mini are small computers that have open-source software available for the logging, processing, and deployment of data onto a localized webserver enabling easy integration of these systems with our device’s primary function. However, the Raspberry Pi rules supreme over the Linux Mini PC for our project due to its cheaper cost, greater availability, and the sheer size of the open-source development community surrounding Pi Programming. The third option, an offsite server, would still require a local device to receive transmissions from the sensors and pass on the data. An offsite server could provide much of the same services but has an ongoing cost of the cloud server. Because of this, we have selected to use a Raspberry Pi for the main Processing and Web Hosting capabilities essential to the function of the central hub.

The Raspberry Pi processing and web hosting unit will have an attached a Raspberry Pi SX1262 LoRa HAT working as a 915MHz gateway for the wireless system and capable of serial debugging. Both will be located indoors and powered through the grid with the Raspberry Pi hosting a webserver to allow the user easy access to logged data.

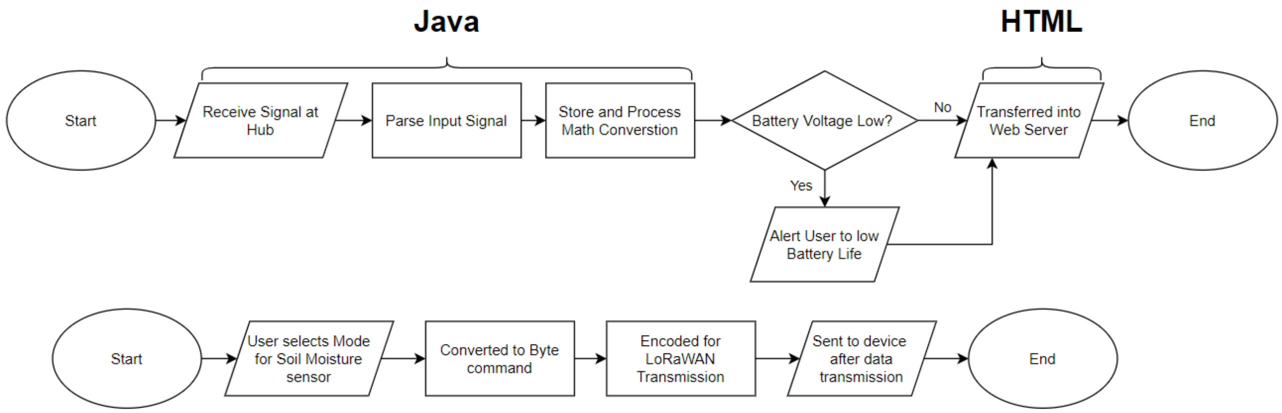


Figure 16: Central Hub Behavior Flowchart

The flowchart shown in Figure 16 summarizes the software workflow for data received by the hub, shown at the top, and selecting modes for the soil moisture sensing device, shown at the bottom. To describe the workflow for receiving messages, the device sends data messages hourly that are received by the LoRaWAN gateway Pi Hat and transmitted to the Raspberry Pi through the shared GPIO between the two. This message is then parsed using Java to decode the bitwise transmissions and process them using their respective math conversions. Calculations are performed on the hub side to reduce the amount of time needed for microcontroller processing of data for power usage minimization. If the battery voltage is low, taking voltage variations in account due to ambient temperature measurements as shown in Figure 17, and compared to typical voltage, shown in Figure 19, then the user is alerted to low battery/device life. This data is then transferred to a Web Server, coded in HTML, that is accessible as a local webpage to the user. Mode selection between device sensor sampling periods and a rapid sampling soil moisture calibration mode will be available at the web server to confirm functionality and establish relative soil moisture range. Upon selection, the mode will be encoded byte wise and sent to the device following the next transmission of data, reducing deployed device uptime in favor of longer battery life.

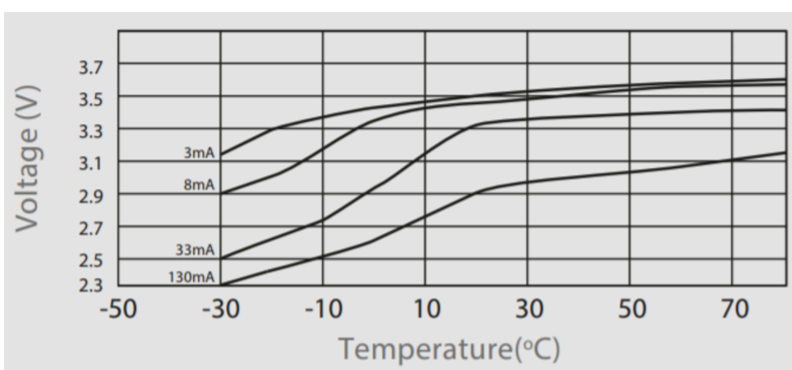


Figure 17: Battery Temperature/Voltage Relationship

4.2.3 Power Sourcing and Sensing

Since our device is meant to be left outdoors and wirelessly connected to transmit sensing results regarding the local environment, it needs to be powered for a substantial amount of time and able to notify the user when to replace the battery or device after the battery has reached an end-of-life state. To run the system for the target time of 2 years as designated by the sponsor, current draws for Dirt Dock systems were estimated alongside the expected time of operation using an hour sampling rate. Shown in Figure 18 and in greater detail within the Appendix in Figure 81, our derived spreadsheet estimates battery life in hours based on the current draw per hour resulting from active mode duty cycle and subsystem current consumption [22]. The battery capacity is then divided by the hourly current draw is to estimate the battery life of the entire system in hours which is converted to years. To create our current model, estimates of subsystem power usage were derived from other sites that had tested the microcontroller power consumption [23], a soil moisture sensing devices datasheet stating a consumption of 5mA [8], and RFM69HCW specifications for transmit and reception power consumption [17]. The Arduino Pro Mini microcontroller chip was quoted to have a wake current of 3.87 mA and a sleep current of 4.3 uA [23, 24, 25, 26]. The bluish grey portion is an estimate of current leakage through the I/O pins for the soil moisture and temperature sensing blocks that are powered by the microcontroller and the temperature sensing estimate is assuming max current and least resistance. The results of this model have shown that, with 2000 mAh of current capacity, the battery life of the device will be around 3 years assuming these approximations hold true.

Battery Capacity (mAh)	2000	Input	Calculated Values	Idle means "Not doing anything"		
Desired Battery Life (hours)	17520	Defined by Switch	Assumed Max Current	Standby means "Shutting down temporarily"		

Component	On (mA)	Idle (mA)	Standby(mA)	On Time (seconds)	Idle Time (seconds)	ON Duty Cycle (%/h)	Idle Duty Cycle (%/h)	Standby Duty Cycle (%/h)	mAh
ATMEGA 328P	3.58	0.7	0.0045	5	0	0.138888889	0	99.86111111	0.009466
Radio Receive	16	0.0012	0.0001	3.164	0	0.087888889	0	99.91211111	0.014162
Radio Transmit	130	0.0012	0.0001	0.289	0	0.008027778	0	99.99197222	0.010536
Soil Moisture Sensing	5	0.0001	0.0001	5	0	0.138888889	0	99.86111111	0.007044
Temperature Sensing	0.329	0.0001	0.0001	0.1	0	0.002777778	0	99.99722222	0.000109
Battery Monitor IC	0.035	0.009	0.0007	3600	0	100	0	0	0.035

Results	
Average mA/h	0.076318
Estimated Battery Life (hours)	26206.26
Desired Life Remaining (years)	-0.99158

Figure 18: Battery Life Estimate Spreadsheet

Our preliminary evaluation of battery sources compared combinations of batteries with solar energy, connecting multiple batteries in parallel, and utilizing a single battery in terms of current carrying capacity and size of the system as the biggest constraints. We initially decided upon a mixture of solar power and rechargeable or secondary batteries but at the recommendation of our sponsor dropped the solar power component due to our new battery estimate indicating it was unnecessary and realizing the self-discharge rate was more than our system could handle. The self-discharge rate of Lithium-Ion secondary, or rechargeable, cells or amount of capacity lost without being consumed is about 5% in the first 24 hours and then loss of about 1-2% each month [27]. For the target two-year time, the battery capacity would have by then decreased to 48% of the total capacity just from self-discharge alone. Factoring this into the battery life estimate approximations, we found that we would still be able to power the system on a Li-ion battery with 3 years of battery life rather than the margin of 7 years that we had predicted prior. This prediction fully cemented the alteration to remove solar panel recharging as it proved unnecessary with our current estimation results.

We were recommended by our sponsor and have chosen to use Lithium Thionyl Chloride batteries for our system. While discussing purchasing and acquisition of materials for prototyping, our sponsor offered a cache of primary, or non-rechargeable, UHR-ER17505-H Lithium-Thionyl Chloride batteries which have a total capacity of 2.0Ah [28] which, when plugged into the spreadsheet, still received around 3 years of predicted device battery life and had a very stable voltage output throughout its effective lifespan. As a primary or non-rechargeable battery, LiSOCl batteries benefit from a long shelf life and very small self-discharge compared to their secondary Lithium-Ion counterparts while still boasting around the same energy density or current capacity [29]. The AA battery packaging of the provided battery is also small and commonly occurring so if the user did want to swap batteries, they could purchase AA batteries off the shelf albeit with a shorter device life due to lower current capacity from other chemistries. The relative stability of the Thionyl Chloride battery voltage, shown in Figure 19, means we will not need any voltage regulator to step up or down and maintain the overall shape/function as the battery discharges until reaching an end-of-life state [28].

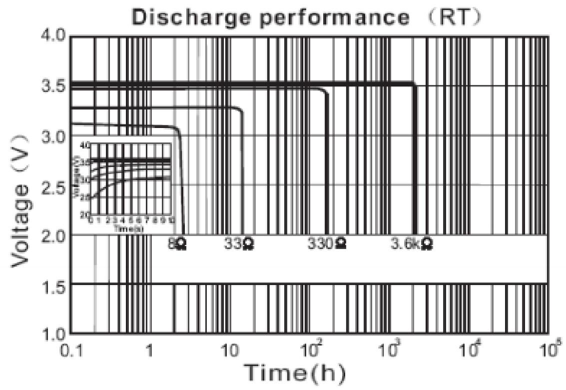


Figure 19: Lithium Thionyl Chloride Battery Discharge Plot

We decided to use voltage monitoring as a battery life indicator to predict when the system will fail to continue operation. This, however, would require utilization of the internal voltage reference of 1V, changing the resistor configuration of the soil moisture and temperature subsystem and decreasing the resolution of our ADC measurements [30, 31]. To this end, we did add a 100Ω current limiting resistor to an ADC pin to allow direct battery voltage reading. However, since voltage varied little with increased battery discharge according to the characteristic curve of the battery, shown in Figure 19, it was decided to include a battery monitor circuit in case the direct to ADC version failed to work as intended. For this, we chose the Maxim Integrated MAX17201G+T IC battery monitor as it is directly applicable to LiSOCl primary battery cells, runs on 3.3V, and can be programmed and interacted with via I2C rather than SPI which is used for the integrated LoRaWAN transceiver [32].

The IC's documentation for a LiSOCl battery's application circuit, shown in Figure 20 and provided by the manufacturers, was used to create the circuit integrated into the battery circuit, displayed in Figure 21 [33]. This circuit contains a resistor for current limiting the input into the microcontroller's ADC, terminating the I2C lines, and using an external thermistor on the battery monitor for more advanced SOC measurements if desired. The IC is programmed and communicated with via I2C as using a combination of parameters described in the Primary Cell Battery guide for the device [33], a GitHub library focusing around easy utilization of the MAX17201 to retrieve battery parameters [34], and supplementary documentation surrounding I2C application [35]. The GitHub library serving as the basis for our software development with the battery monitor is also licensed on the copy-left GNU GPL v3 mentioned previously [21].

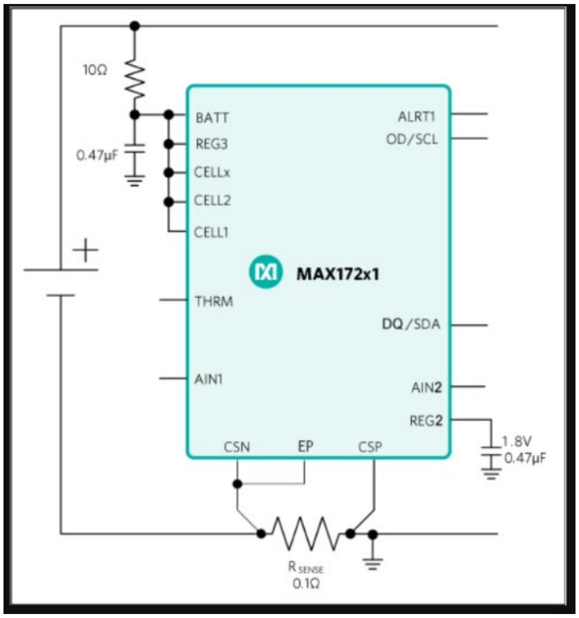


Figure 20: Battery Monitor IC Application Circuit

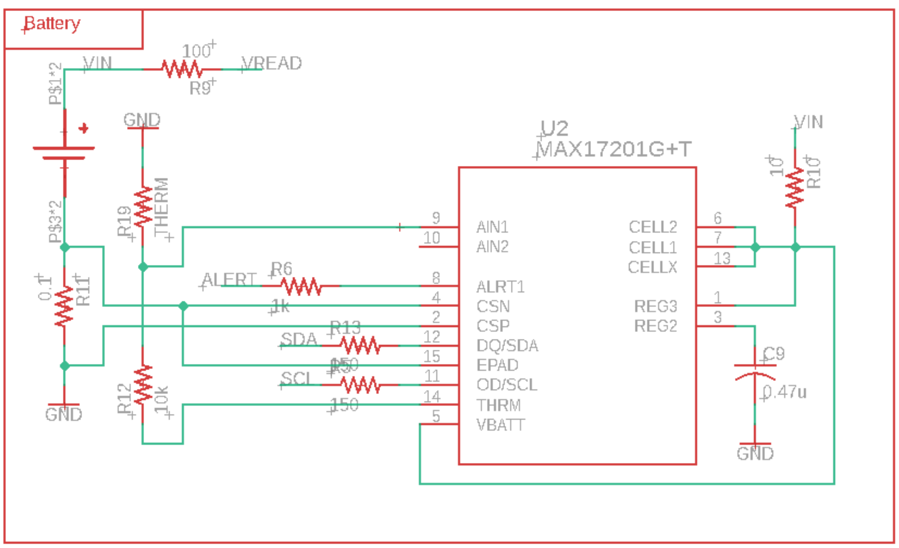


Figure 21: Power Sourcing and Sensing Circuit

During our scramble to assemble and test a functional prototype for the device, we stumbled upon a third method of battery estimation, that would better serve us than both methods mentioned prior. This method uses the internal bandgap voltage as a reference to measure the voltage supplied to the ATMEGA 328P chip from the battery [36, 37]. The code, shown in Figure 22 and available for ATMEGA 328 boards, sets the voltage reference to VCC or the power supply, measures the internal reference which fluctuates with VCC, and approximates VCC using this value [37]. This method of battery voltage estimate does not require any additional hardware components or modifications and uses no extra pins on the microcontroller, reducing overall system power consumption and cost for the devices. Band gap voltage battery measurement must be calibrated prior to voltage estimation but once found does not change for other microcontroller chips in the same series [36]. To attain this constant, a 0.1 uF capacitor must be connected from Aref to ground and A0 should be read and saved as “InternalReferenceVoltage” for the voltage estimation formula, shown in Figure 22, using the internal voltage reference for the ADC.

```
int getBG(void) {
  const long InternalReferenceVoltage = 1050L;  // Adust this value to your specific internal BG voltage x1000
  // REFS1 REFS0          --> 0 1, AVcc internal ref.
  // MUX3 MUX2 MUX1 MUX0  --> 1110 1.1V (VBG)
  ADMUX = (0 << REFS1) | (1 << REFS0) | (0 << ADLAR) | (1 << MUX3) | (1 << MUX2) | (1 << MUX1) | (0 << MUX0);
  // Start a conversion
  ADCSRA |= _BV( ADSC );
  // Wait for it to complete
  while ( ( (ADCSRA & (1 << ADSC)) != 0 ) );
  // Scale the value
  int results = (((InternalReferenceVoltage * 1024L) / ADC) + 5L) / 10L;
  return results;
}
```

Figure 22: Bandgap Voltage Measurement Method

4.2.4 Soil Moisture Sensing

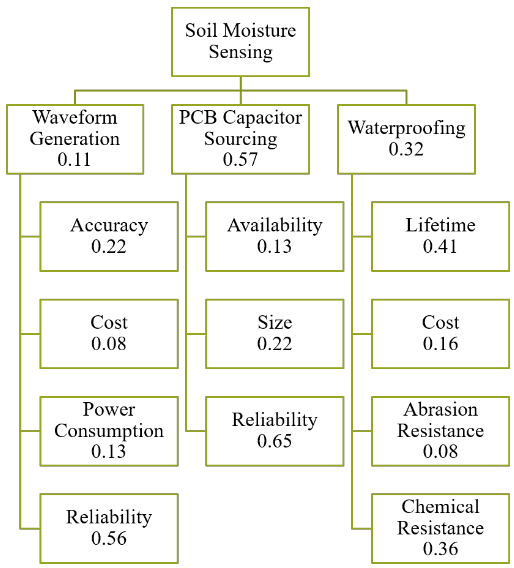


Figure 23: Soil Moisture Sensing Objective Tree

Capacitive soil moisture sensing, whose functional decomposition is shown in Figure 8, is the key measurement necessary for effective implementation of our outdoor soil moisture sensor. The soil moisture monitoring circuit is a combination of a PCB capacitor whose capacitance varies as the surrounding ionic content or soil moisture changes and waveform generation that generates PWM pulses of a fixed high frequency. Shown in Figure 23 is the objective tree or systematic breakdown of importance for the soil moisture sensing subsystem that was derived through AHP analysis using Table 27, Table 28, Table 29, and Table 30 in the Appendix. While waterproofing is indirectly related to the soil moisture sensing subsystem, waveform generation and PCB capacitor pattern/sourcing alternatives were explored in Table 35 and compared based on the rankings of importance shown in Figure 23.

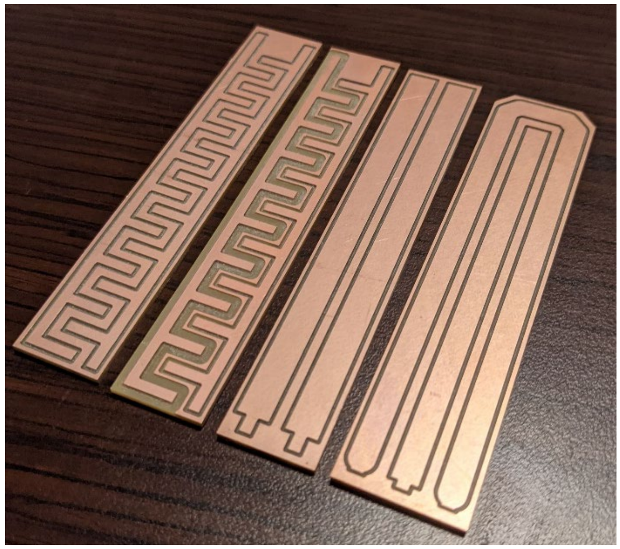


Figure 24: Milled PCB plates used for capacitance testing

When it comes to the creation of the PCB capacitor, we decided to create our own custom soil moisture sensor PCB due to its potential for integrating with the other systems needed for the device. Creating our own custom PCB allows for greater customization of the PCB capacitor pattern, reduces the size of the system, and was favored by our sponsor within the initial sponsor meeting. To determine the most effective PCB capacitor pattern for our purposes, 4 PCB capacitor prototypes, shown in Figure 24, were created at the Innovation Center, and the magnitude of their capacitances measured using an LCR meter, shown in Figure 25, with the assumption that the largest capacitance was the best [3]. With the results shown in Table 11, the parallel plate pattern had the largest capacitance of 54.836 pF, having more than double the capacitance of the other patterns and is what we chose for our PCB. Patents, WO2009066992A2 and US20150338363A1, exist detailing capacitive moisture sensors exist but they focus on the usage of interdigitated finger PCB capacitor pattern, shown in Figure 26, rather than fringing field effect measurement of soil moisture variation [38, 39].

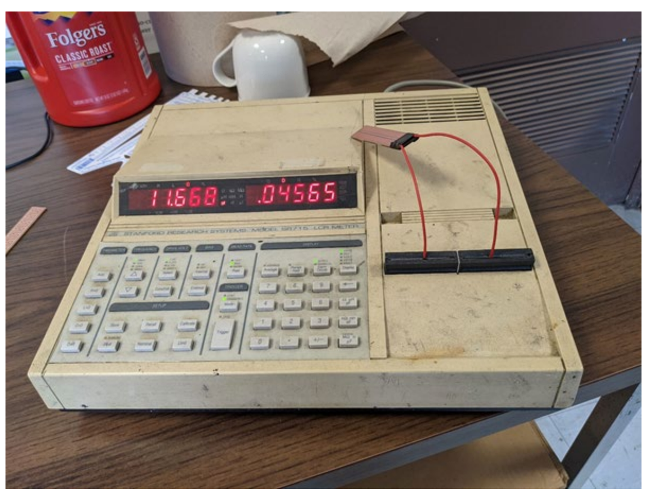


Figure 25: LCR Capacitance Measurement Example

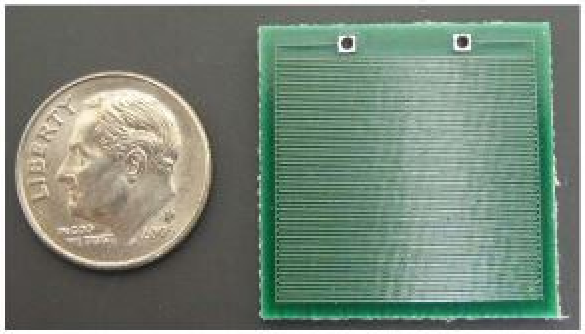
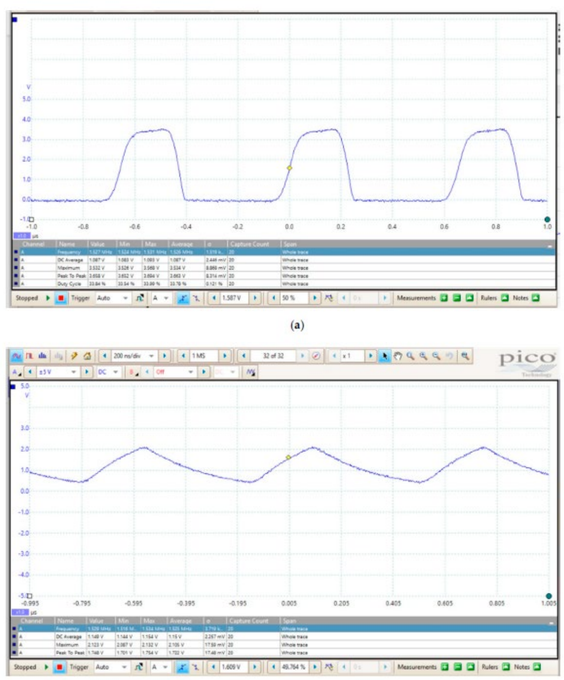


Figure 26: Patented Interdigitated Finger Capacitor Example

Table 11: Results of PCB Capacitor Testing

PCB Style	C	D
Interdigitated Fingers	9.78pF	.04585
Parallel Traces	11.713pF	.04542
Circle Around (Large Area)	13.150pF	.04478
Circle Around	7.4389pF	.04662
Parallel Plates	54.836pF	.06174



(b)

Figure 27: Example of soil moisture sensor input and output

For waveform generation, the most important aspect of the component or system is reliability followed by accuracy because the functionality of the entire device is based around having accurate and trusted soil moisture measurements. Most commercially available systems utilize a hardware generated circuit that creates a square wave, as shown in the top picture in Figure 27 from [40], pass it through an RC filter comprised of the PCB capacitor and known resistance, and converts the voltage output, the bottom picture in Figure 27, to a digital signal. The voltage change is used to determine either a percentage or level due to its position in a range created between two calibrated end points of the voltage output while submersed in water and air [8, 9, 40].

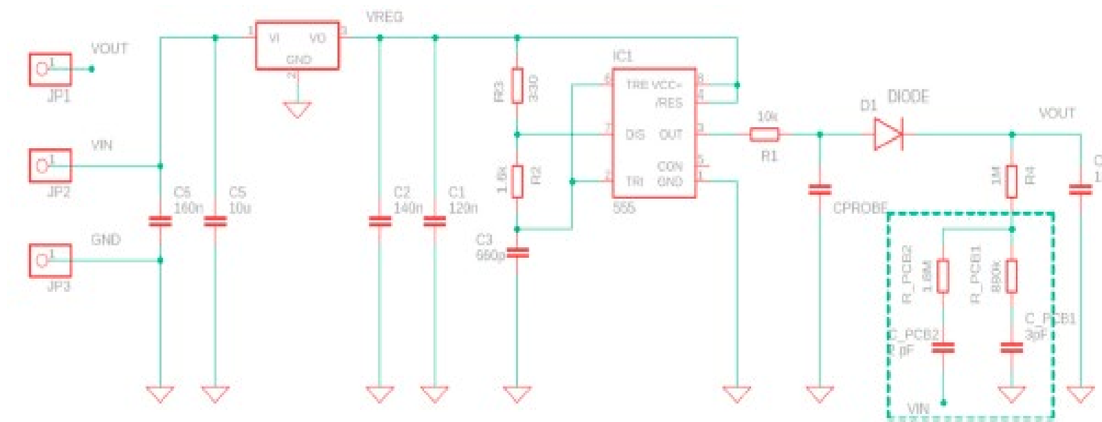


Figure 28: Capacitive Soil Moisture Sensing Circuit Example

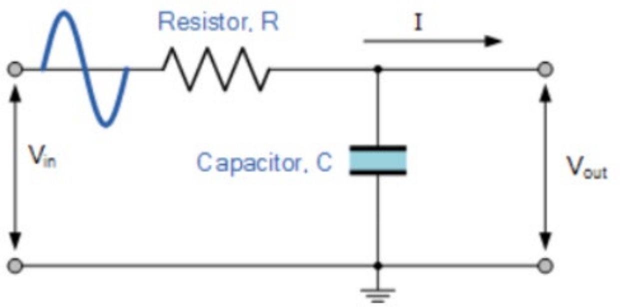
$$t_H = 0.693(R_A + R_B)C$$
$$t_L = 0.693(R_B)C$$

Other useful relationships are shown below:

$$\text{period} = t_H + t_L = 0.693(R_A + 2R_B)C$$
$$\text{frequency} \approx \frac{1.44}{(R_A + 2R_B)C}$$
$$\text{Output driver duty cycle} = \frac{t_L}{t_H + t_L} = \frac{R_B}{R_A + 2R_B}$$

Figure 29: Equations representing astable 555 Timer Waveform Generation

We adapted the hardware implementation found in most capacitive soil moisture sensors, shown in Figure 28, as the template for our soil moisture sensor [40]. Moving left to right through the system shown in Figure 28, traditional capacitive soil moisture sensors regulate voltages down to 3.3V to provide a stable operating voltage for the 555-timer generating the waveform as well as allow 5V and higher devices to power the device. Depending on the resistor voltage division and capacitor voltage decay time, whose relationships are shown in Figure 29, the 555 timer operates as an astable waveform generator [41]. Since our version includes the values of 330Ω for Ra, 1600Ω for Rb and 660 nF for C for the input voltage divider into the 555 timer, a square wave is created of 619 kHz. This square wave is passed across an RC filter which limits the current to the microcontroller and whose impedance changes with surrounding soil moisture content, changing the voltage output akin to what is shown between the top and bottom images in Figure 27. Since the battery output is incredibly stable until the end-of-life period, we opted to simplify this circuit by removing the voltage regulation as shown in Figure 31 with the capacitor still connected at the net labeled CAP between the Schottky diode and the resistor labeled R3. The voltage output of this system can be modeled via voltage division using the imaginary part of the capacitor's impedance with the relationship shown in Figure 30 [42]. As the capacitive reactance changes due to changes in surrounding permittivity, the output voltage changes in proportion, filtering out different parts of the input waveform to do so.



$$V_{out} = V_{in} \times \frac{X_C}{\sqrt{R^2 + X_C^2}} = V_{in} \frac{X_C}{Z}$$

Figure 30: Capacitive Soil Moisture Sensor Output

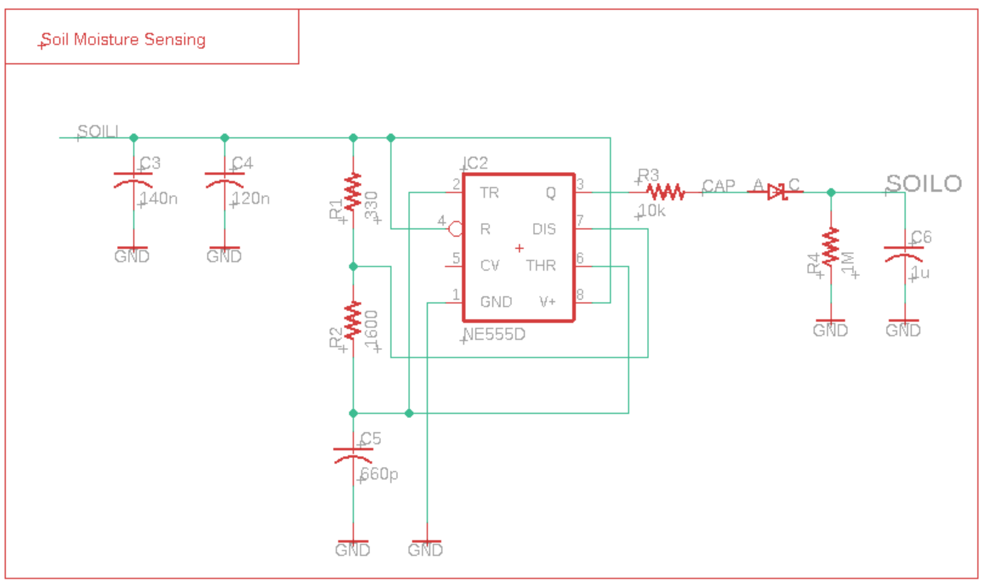


Figure 31: Simplified Capacitive Soil Moisture Sensing

Having compared PCB capacitor effectiveness in previous testing, we decided upon a parallel plate capacitor built into the PCB and created a prototype for the soil moisture sensing subsystem to test our PCB capacitor as shown in Figure 32. We found that when a ground plane is below the PCB capacitor the device acts like a parallel-place capacitor where all the field is between the plates and there is only small fringing near the edges, leading to less variation in capacitance. The length of the soil moisture sensing prototype capacitor, 3.17 inches, also led to an offset in soil moisture within the test period due to a large percentage of the capacitor sitting above of the soil level. Talking to our sponsor on senior design day, he suggested looking at the Gravity Capacitive Soil Moisture sensor which he gave us for inspiration on how to alter the pattern. In this sensor, shown in Figure 33, the PCB capacitor is 2 inches long and single layered for the capacitive or measuring portion of the PCB [8]. To adjust for a happy medium between the two examples, we reduced the PCB capacitor to 2.35 inches as shown in Figure 34.

Figure 32: Soil Moisture Sensing Prototype

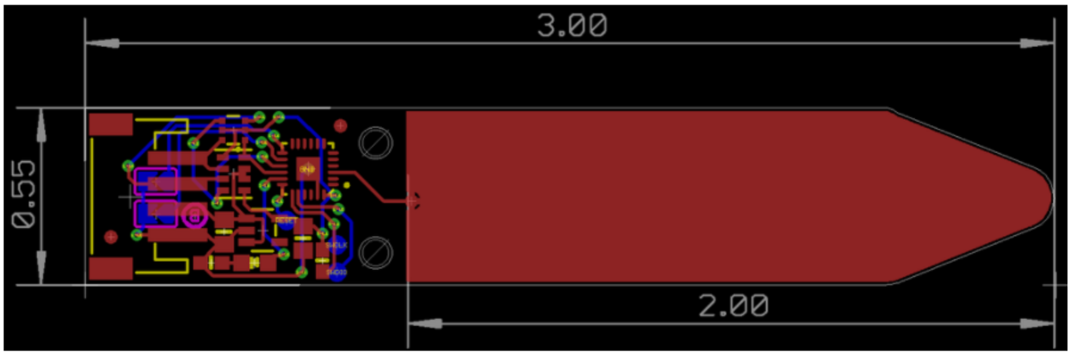


Figure 33: Gravity Capacitive Soil Moisture Sensor Example

Figure 34: PCB Capacitor Design

To process the resulting voltage change, we adapted code from the DFRobot commercial capacitive moisture sensor auxiliary documentation [8] and an How2Electronics article about the sensor [43] to create soil moisture logging code, shown in Figure 35. Recalibration of the capacitive soil moisture sensing depending on water composition is necessary because water and its ionic concentration have a permittivity of approximately 80 while soil has a permittivity of around 5-8 so changes in the water content have a much larger impact on capacitance compared to the soil type [44]. Prior to initial use, calibration values are obtained using the analog values read while the device is in air and water and saved to initialize the level mapping. Using this range, the analog input from the soil moisture sensor is read and transmitted to the central hub to mapped between the two values and approximate a relative soil moisture percentage. Our testing was mainly focused on soil moisture percentage values because they showed the change in soil moisture value over time, but levels are also easy to implement with if statements indicating levels of wet or dry depending on the percentage measured.

SecondLog_SoilMoist | Arduino 1.8.13

File Edit Sketch Tools Help

SecondLog_SoilMoist

```
// Integer values that need to be calibrated to set mapping
const int AirValue = 564;
const int WaterValue = 293;

int soilMoistureValue = 0;
int soilmoisturepercent = 0;

unsigned long curSeconds = 0;

void setup() {
  Serial.begin(9600); // open serial port, set the baud rate to 9600 bps
  Serial.println(" Time (s) , Analog Reading , Percent Value ");
}

void loop() {
soilMoistureValue = analogRead(A0);  //put Sensor insert into soil
curSeconds = millis()/1000;
Serial.print(curSeconds);

Serial.print(" , ");
Serial.print(soilMoistureValue);

soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
Serial.print(" , ");
Serial.println(soilmoisturepercent);

  delay(1000);
}
```

Figure 35: Soil Moisture Sensing Example Code

4.2.5 Controls and Timing

Functioning as the brains of the operation, the microcontroller, whose functional decomposition is depicted in Figure 9, converts analog voltages to digital signals, control which sensors are powered, and communicates with the onboard transceiver to transmit and receive between the soil moisture sensing devices and the central hub. AHP Table 31 in the appendix assisted in the systematic comparison of important microcontroller parameters wherein we found that power consumption and device size are the most important parameters to for microcontroller selection. We found multiple microcontrollers that had Analog to Digital Converters (ADC), I/O, and can communicate with our transceiver. We decided to use a microcontroller in the chip form factor due to the lower current draw as we can selectively implement necessary systems on the custom Printed Circuit Board even though it is more difficult to program due to a lack of preloaded bootloader or connection to the board. Specifically, we modeled our hardware system off the Arduino Pro Mini because it is low cost, has a low power draw, and a small form factor while containing all the functionality described previously [45].

The microcontroller chip used on the Arduino Pro Mini board is the ATMEGA 328P which supports ultra-low power draw around 5 mA while active [23], is capable of being used with the Arduino IDE, can handle SPI and I2C communication alongside ADC inputs, operates at 3.3V, and within the target operational temperature range of the system [46]. Using the Arduino Pro Mini schematic as a template, shown in Figure 36, for the application circuit with this microcontroller chip we simplified the system for our purposes, removing a voltage regulator and pins for external connection (not pictured) and integrated it into our own circuit, shown in [47]. According to the ATMEGA 328P data sheet, temperature differences result in a difference of about 10% error in the internal clock value [46] so we decided to use an externally connected 8MHz crystal with accompanying 22 pF capacitors as were used on the Arduino Pro Mini 3.3 V version [45]. Since the maximum input/output current for the ATMEGA 328P is 40mA per output pin, we will use the digital I/O ports to toggle the soil moisture sensing and temperature sensing subsystems since they use less than the max according to our power consumption estimates [46]. In case we wanted to assign an external ADC reference, holes for headers, shown in Figure 39, were created alongside a capacitor intended to deal with transient waveforms and noise by stabilizing power input for applied references. A capacitor was added within the battery circuit to allow for the same functionality for voltage into the microcontroller and was placed near the microcontroller as such. The optional pull up resistors for the I2C connections were also adapted from the schematic in the case of excess noise or erroneous input on the data line to the battery monitor IC due to floating when messages are not sent across the bus.

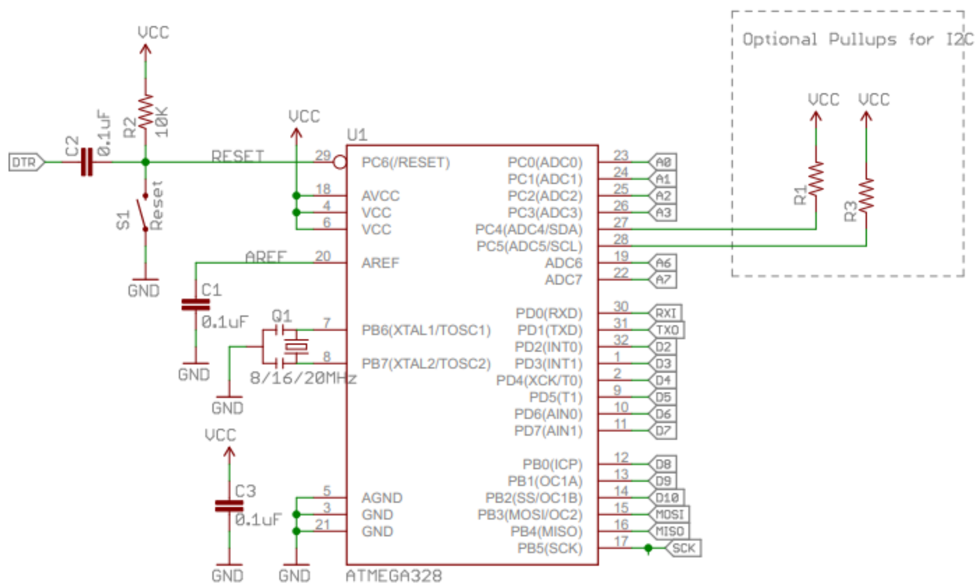


Figure 36: Arduino Pro Mini Schematic

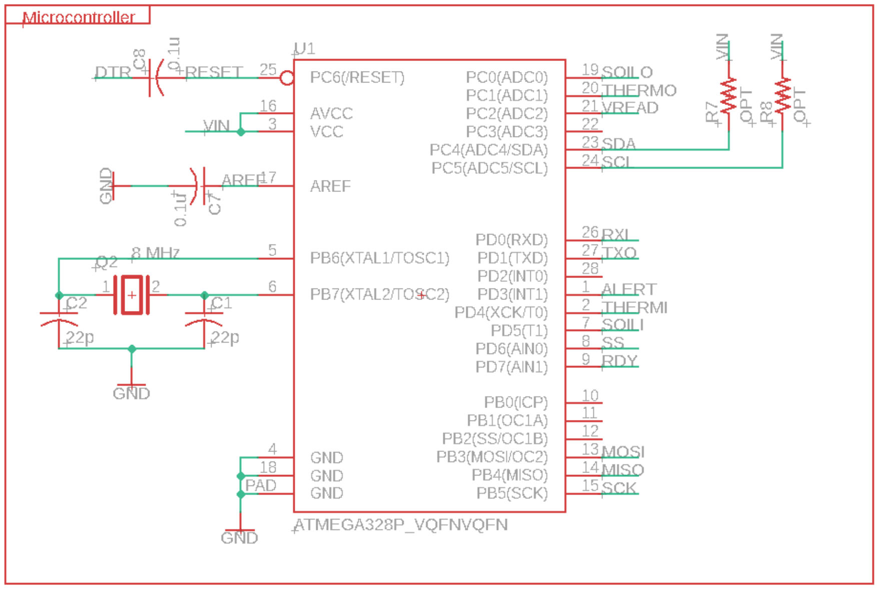


Figure 37: Microcontroller Application Circuit

The In-Circuit Serial Programming (ICSP) connection can be used to burn the bootloader and associated code onto an Arduino microcontroller chip to allow for programming and utilization of the Arduino IDE [48]. To enable the ability to use the Arduino IDE on the ATMEGA 328P chip, we intend to use another Arduino to load the bootloader onto the chip using an ICSP connection [48]. The bootloader, once loaded to the device through an ICSP-to-ICSP connection, is what looks for sketches to be uploaded from the serial/USB port through the Arduino IDE. For this purpose, we broke out the ICSP pins, shown in Figure 38, or MISO, SCK, MOSI, Reset, VCC, and Ground into individual header, serving as the main connection to the microcontroller and shown in Figure 39 [48]. The other header described in Figure 39 are for optional external antenna connections, optional outputs for the thermistor and soil moisture sensor outputs for troubleshooting, and the external analog reference input.

Figure 38: ICSP Pinout

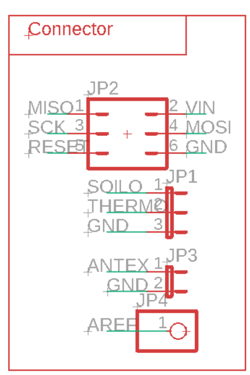


Figure 39: PCB Connectors

The Arduino Pro Mini itself lacks a USB port, leaving users to often use FTDI to USB converters to easily program the devices [49]. Because of this, we decided to integrate an FTDI to USB converter into the custom PCB to aid in prototyping to be able to power the system from a computer when a battery is not loaded in during prototyping alongside an ICSP connection for boot loading. For this, we found the schematic for the Beefy 3 FTDI to USB connection which is typically used with full Arduino Pro Minis to allow powering of the microcontroller from the USB connection [50]. This system, shown in Figure 40, uses an FTDI USB converter with an external voltage regulator to handle larger current loads up to 600mA from a USB voltage [50]. The capacitors used throughout the schematic are used to stabilize power input to the system from the source whether it is into the voltage regulator, into the microcontroller, or out of the USB connection. The DTR pin allows for the Arduino target to auto-reset when a new sketch is downloaded through the capacitor input shown previously in Figure 37. The circuit was simplified by removing the power LEDs and tying the CTS pin to ground as is typically done with Arduino Pro Minis to create the system shown in Figure 41.

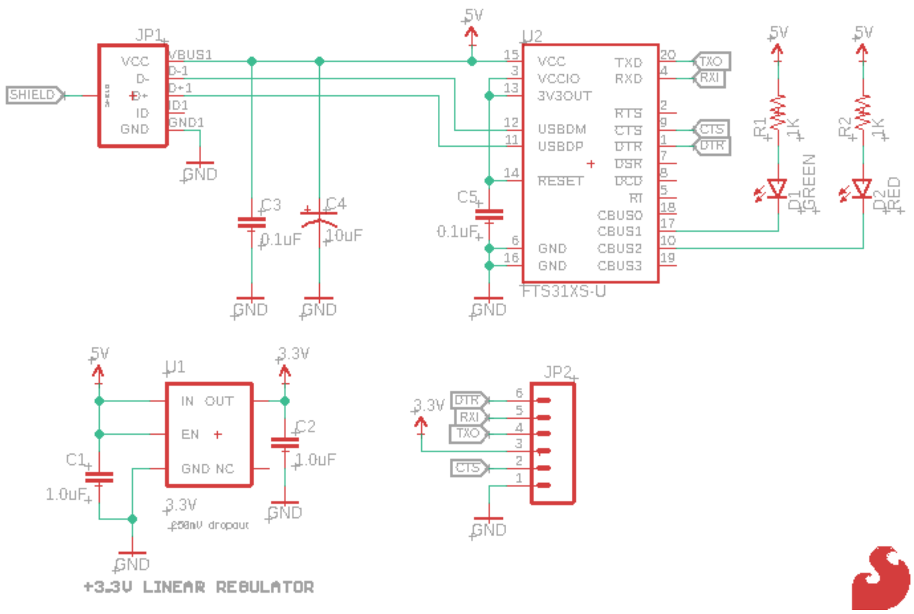


Figure 40: USB to FTDI Breakout Board Schematic

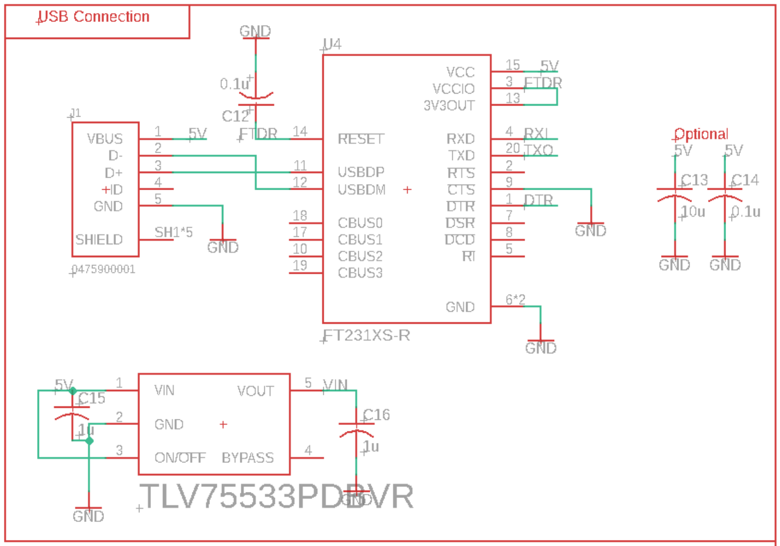


Figure 41: USB to FTDI Application Circuit

Using the Arduino LowPower Lab library, we created test code, shown in Figure 42, that would continuously put the device into a low power sleep mode until the duration stamp selected via mode select was reached [26]. A while loop was used to continuously call the power down command because it has a max sleep period of 8 seconds with the "durStamp" recording the last time sampling occurred. The difference between this and the current time sample were compared to decide whether to leave the while loop. The mode and cal options represent decisions that would change the period between wakeup for the device and assign the current soil moisture sensing value as air or water locally.

Timing_Testing

```
int mode = 0;
int cal = 0;
unsigned long durStamp, curSec, duration = 0;

void setup() {
}

void loop() {

  if (mode == 0) duration = 3600;

  if (mode == 1) duration = 5;

  if (mode == 2) {
    duration = 1;
    if (cal == 1) calAir = soilVal;
    if (cal == 2) calWat = soilVal;
    if (cal != 1 || cal != 2) mode = 0;
  }

  durStamp = millis();
  duration  *= 12.5; //Converts to millis with sleep time factored in

  while ((millis() - durStamp) < duration) {
    LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
  }
}
```

Figure 42: Timing and Mode Code

Temperature sensing focuses on the application of an NTC thermistor in a 10kΩ voltage divider circuit connected directly to a microcontroller analog input and toggleable through microcontroller digital I/O [51]. An example of temperature sensing code is shown in Figure 35 and uses the Stein-Hart equation to convert between the resistance derived from changes in voltage with a temperature in Kelvin that is converted to a temperature in Fahrenheit [14, 15].

Theristor_Testing §

```
#define THERMI PD4
#define THERMO PC1

const int A = 1.009249522e-03;
const int B = 2.378405444e-04;
const int C = 2.019202697e-07;

int tempVal = 0;

void setup() {
  Serial.begin(9600);

  pinMode(THERMI, OUTPUT);
  pinMode(THERMO, INPUT);
  digitalWrite(THERMI, HIGH);
}

void loop() {
tempVal = therm(analogRead(THERMO));
Serial.println(tempVal);
delay(1000);
}

int therm(int V0) {
  float logRT, T, TC, TF = 0;
  logRT = log(10000 * (1024 / (V0 - 1)));
  T = (1 / (A + B * logRT + C * logRT * logRT * logRT));
  TC = T - 273.15;
  TF = (TC * 1.8) + 32;
  return TF;
}
```

Figure 43:NTC Thermistor Code

4.2.6 Enclosure and Waterproofing

For the physical construction and enclosure of the outdoor wireless soil moisture sensor, we followed through with the idea of the top-heavy stake, shown in Figure 11, akin to commercially available soil moisture sensors. This topology is defined by the top half of the PCB holding the electronics and protecting them with a 3D printed plastic enclosure while the bottom half functions as the PCB capacitor and changes with changes in surrounding soil moisture. When asked about this during our initial design meetup, our sponsor indicated that he preferred this implementation and it fits the intended purpose of the product as a series of devices that could be monitor various locations, reporting to a central hub. This also meshes well with the choice to create a custom PCB as it allows for integration of all subsystems for the devices onto one cohesive piece and is quick and easy to modify the enclosure as the shape changes, taking about a day or less to manufacture at the IC.

Concerning the waterproofing of the device, we performed AHP analysis of the parameters important for device waterproofing, shown in Table 30 and graphically depicted in Figure 23 with soil moisture sensing. Chemical resistance and lifetime were determined to be the two most important factors assuming all choices seal the device against surrounding water or moisture to ensure that the waterproofing sustains effectiveness while exposed to the elements over the lifetime of the device. Keeping this in mind, Silicone resin is the chosen waterproofing method for the final product due to it's long lifespan, good performance in extreme temperatures and against chemicals, cost effective nature, and is unlikely to be removed through abrasive methods inside the enclosure while the device is in operation. The plastic enclosure combined with the silicone sealing will ensure proper operation of the outdoor wireless top-heavy stake shaped device for its intended life cycle of two years. After creation and electrical testing of our final PCB prototype comes to completion, we will design and print a 3D-printed plastic enclosure for the top portion and apply silicone to the top of the electronics before attaching the cover.

5. Integration, Verification and Validation

To act upon our design, we designed and ordered a hardware PCB, integrated software development using a GitHub repository, and populated the system. Using this system, we tested what was functional and attempted to troubleshoot what was not in relative importance.

5.1 Integration

5.1.1 Hardware Integration

For the Dirt Dock devices, we created a PCB to tie the battery monitoring, power sourcing and sensing soil moisture sensing, and wireless transmission subsystems together, shown in Figure 44, and in greater detail in Figure 82 within the Appendix. All the previously individually described subsystem circuits were combined onto this one PCB, shown in greater detail in Figure 83 in the Appendix, while being implemented and tested incrementally. A 3D printed cover was created to provide an enclosure for the device's electronics like created in [52] due to its simplicity and effectiveness. Silicone should be applied to the electronics and the battery as a waterproofing seal before putting the enclosure on prior to device deployment.

Concerning the physical layout of the board, power and decoupling capacitors were put near the input to the microcontroller or intended system to minimize input noise effectively. The through hole headers planned for ICSP programming and located away from edges to allow for a 3D printed enclosure to fit snugly without component removal alongside mounting holes positioned just above the capacitive sensor to allow the enclosure to snap in. The top and bottom planes on the electronics section of the board are both ground planes to maximize current carrying capability of the ground traces. For planar conductor trace width calculations based on current draw, we found a PCB Trace Width Calculator provided by Digikey which stated the minimum size of traces was 10 mil for the max expected current draw from the battery to ensure the heat dissipated by the trace is not too much to significantly damage the board [53]. Based off this idea, we decided to go with 6 mil for data/communication wires that use lower current values and 10 mil for the power circuits/connections. System placement based on reducing transmission noise led to the following setup: the capacitive moisture sensing circuit is located near the PCB capacitor section at the bottom and shown in greater detail in Figure 34; the transceiver circuit is on the right side of the microcontroller to reduce communication lines between the antenna and transceiver and microcontroller; the USB to FTDI circuit is located on the other size and more spaced out due to low priority; and the battery monitoring circuit was placed on the opposite side of the device to maximize space for higher priority systems.

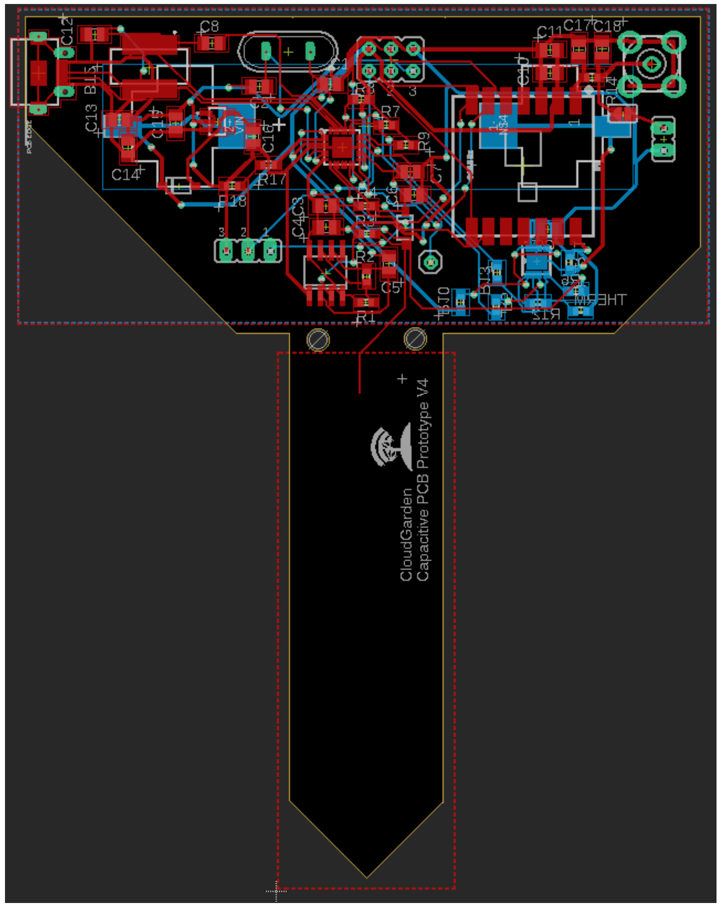


Figure 44: 1st Soil Moisture Sensor PCB Prototype

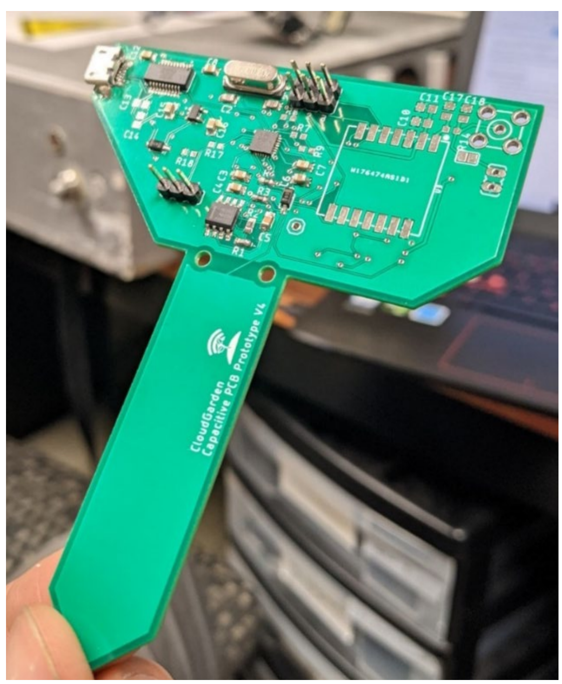


Figure 45: Partially Populated Prototype

The physical transceiver on one the first prototype boards were partially populated, as shown in Figure 45, with the USB to FTDI circuit, Soil Moisture Sensing circuit, and the microcontroller application circuit to attempt boot loading and programming of the chip. Initial development revealed several errors including: a surface mount (SMD) 8MHz crystal oscillator purchased for the intention of at through hole connection, the AA battery holder ordered was through hole for an SMD connection, the micro-USB connector did not match the PCB footprint, and, most importantly, the footprint for the RFM69HCW is wrong and has a different pin count. To counter this, we took the liberty of creating a new PCB revision, shown in Figure 46 and at larger scale as Figure 84 in the Appendix, that integrated an easier to solder microcontroller package and increased header along with correcting the previously mentioned errors. A fully populated version of this final prototype is displayed in Figure 47.

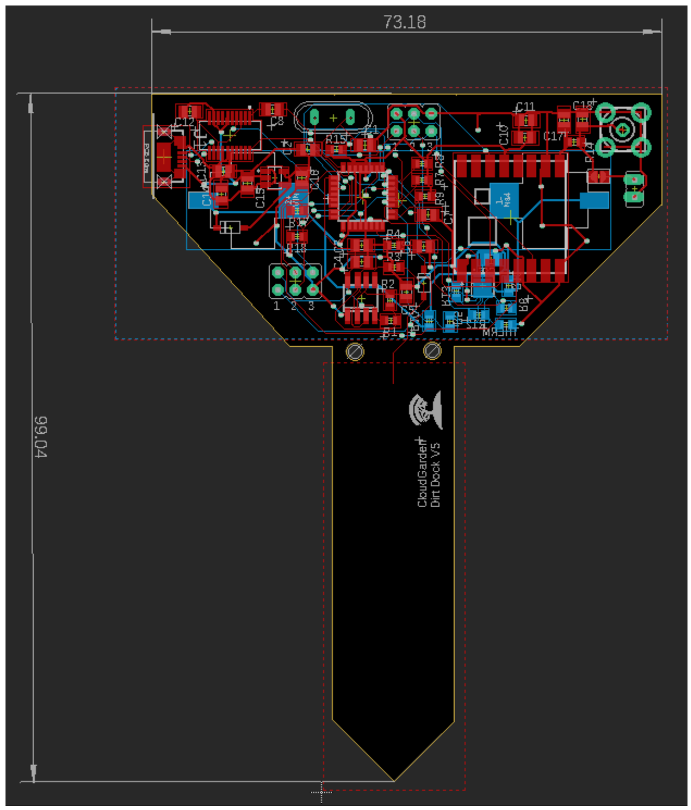


Figure 46: Revision of Soil Moisture Sensor Prototype

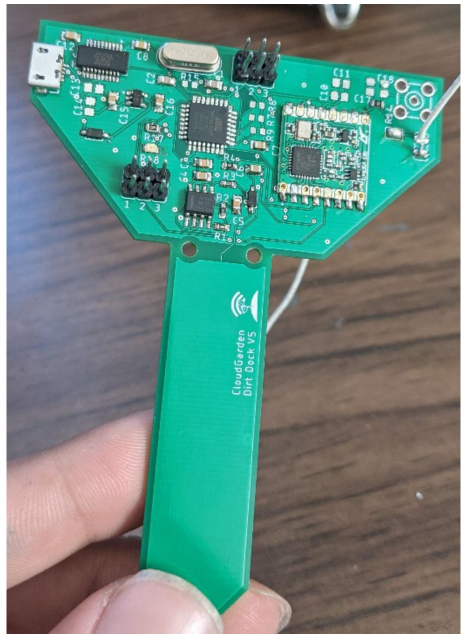


Figure 47: Final Prototype

5.1.2 Software Integration

Software integration for the Dirt Docks focused on creating a mixture of C++ libraries that would be accessed via the Arduino IDE for easy implementation and programming to the ATMEGA328P chip. Object-oriented programming was intended to be used to implement communication to the transceiver, soil moisture sensing, temperature sensing, and battery voltage measurements.

Progress towards this system was captured within a private GitHub repository as our sponsor communicated the potential intention to make a feasible product with what we come up with [54]. Soil moisture sensing uses an initial calibration with local water/fertilizer and air to set the setpoints for the upper and lower end of the soil moisture range that later values are compared within as described in Figure 35. This calculation was intended to be handled on the Hub side with a command to switch to a higher sampling rate calibration mode and saving them into private calibration variables on the hub-side of processing for the final prototype. Communication with the transceiver was intended to use SPI to send and receive commands/messages to send data wirelessly using the LowPower Lab RFM69 Library [20]. Traditional sampling activates the soil sensing and temperature sensing circuits to take measurements, with a small delay to deal with transient waveforms, and measurement of the battery voltage using the internal bandgap as a reference. The testing mode operates at a higher rate of sampling to validate that we are gaining effective data through movement of the device into soil, water, and air of varying temperatures during the prototyping phase. Outside of the sampling activation, another LowPower Lab library was used to put the system into deep sleep with the temperature sensing and soil moisture sensing subsystems deactivated to conserve power as shown in Figure 42 [26].

As we began to panic with difficulties split between hardware and software incompatibilities, a single Arduino file was created that was capable of all these functions within a loop and is available at the Appendix within Section 9.4.1 Final Code. We developed a webserver and back-end database built for the Raspberry Pi/Central Hub to support a local webpage that is remotely accessible over the local Wi-Fi network. Battery measurement, soil moisture measurement, and temperature measurement as covered in the code examples described within their individual design summaries were implemented but left untested digitally or on our prototype due to a lack of serial communication.

5.1.3 Integration Difficulties

Majority of the month of prototyping between the CDR and this report focused on attempting to program or load programs onto the microcontroller attached to our custom PCB, a problem that we foresaw but was unnecessary as defined by our ER or could have been avoided altogether with a different design. The FTDI to USB circuit, created for this very purpose and necessary even if we had used an Arduino Pro Mini board, failed, and could power but not program the PCB even after installing the drivers as described by multiple troubleshooting websites [55]. The lack of documentation of avrdude or Arduino IDE errors led us to scavenge for information across Arduino forums to find potential solutions to improve device effectiveness. Even after deciding to move on from this system, a lack of a structured method for soldering verification led to confusion whether our programming issues were the result of software or hardware issues with PCB population.

Starting with the Arduino as ISP [48] connection, the method of programming we originally planned and shown in Figure 49, we were unable to program the device until we corrected our reset connection as pin 10 of the programmer should be connected to the reset pin on the device to program correctly as shown in Figure 48 [56]. Using this method allowed us to either program the bootloader, using the “Tools -> Burn the Bootloader” option, or program directly to the device by selecting “File -> Upload Using Programmer” which wipes whatever program was previously on the microcontroller [57, 58]. The procedure for programming using the Arduino as ICSP system was to: 1) Load the Arduino as ISP example to the programmer board; 2) Connect the ICSP connection between the Arduino Programmer and the device as shown in Figure 48 [56]; 3) Select the board as Arduino Pro Mini with a 3.3V 8Mhz ATMEGA 328P with a programmer to Arduino as ISP; and 4) Select from the previously mentioned options. The bootloader itself enables serial communication from the board and allows for programming using the USB connection [59, 58]. While we were able to establish programming, the power supply was still from a 5V Arduino Uno, so we had to transfer to another programmer or power method for testing and communication between devices as the systems were calibrated for 3.3V.

```
ISP - Target
+5V - +5V
GND - GND
11  - 11
12  - 12
13  - 13
10  - RST
```

Figure 48: Arduino as ISP Connection

After figuring out a suitable method for programming, we did not know how to correctly define I/O pins to be able to input and output as if we were using an Arduino Pro Mini and as detailed in the EAGLE Schematic [47]. Since the schematic for the microcontroller circuit was identical to an Arduino Pro Mini [45] and used the same chip, we reasoned that using similar pin names would allow us to control or program the system in a similar manner. As were able to load programs to the PCBs through the ICSP, we decided to set the output of the thermistor circuit input high with a 6.8kΩ resistor and red LED where the NTC thermistor voltage divider circuit typically sits to serve as a verification that the PCB was programmed and powered as intended. After substantial testing, we found that rather than the pin number on the Eagle schematic, we needed to define pins as what it was known as to the microcontroller chip which is defined within the IDE during programming.

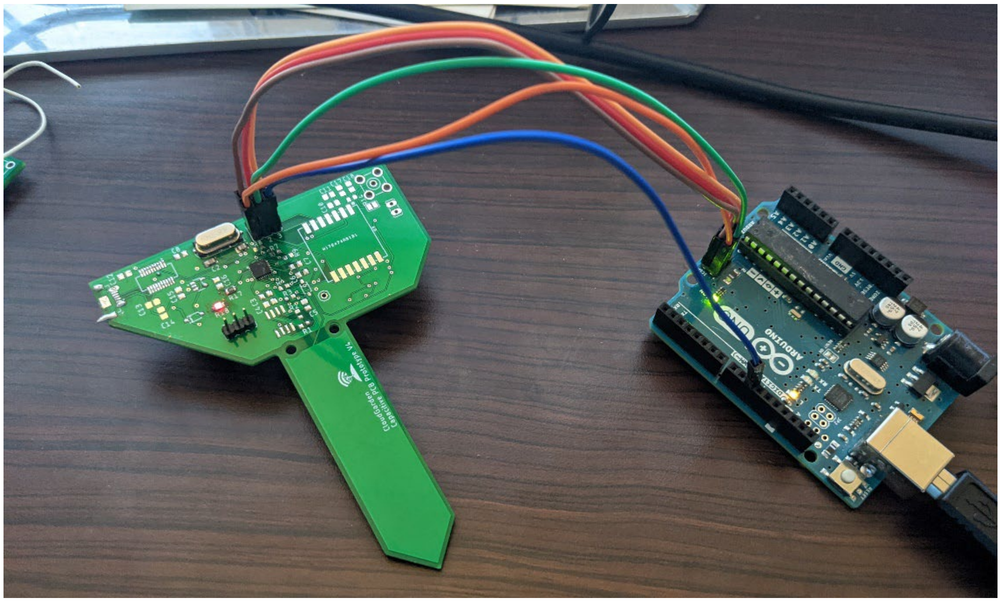


Figure 49: Arduino as ISP Programming

To enable 3.3V programming and decrease programming difficulty, we purchased USBasp programmers, shown in Figure 50. To begin programming, we used Zadig, a generic driver downloader, to install the libusb (V3.0.7.0) USBasp driver to our systems [60] and updated the firmware on the USBasp devices by following a YouTube tutorial [61]. The USBasp programmer works in a similar fashion to the Arduino as ISP where either the “Upload using Programmer” or “Burn bootloader” options are used but this time without a port selected and effectively powers the circuit at 3.3V [62]. The USBasp programmers were used for the remaining programming and testing performed through our prototypes up until the project’s prototype deadline.

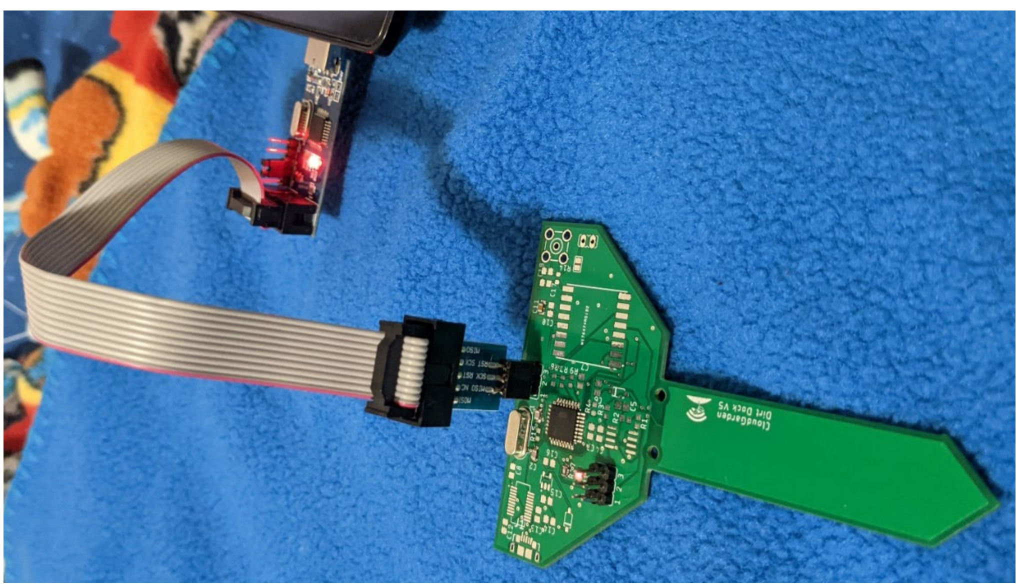


Figure 50: USBasp Programming

Having discovered methods of powering and programming the device outside of the FTDI to USB circuit, we could now analyze the sensing subsystem’s wireless performance or variation by outputting power to their respective subsystems. However, since our FTDI to USB circuit did not work, we were unable to receive serial messages or other digital communication from microcontroller onboard the Dirt Dock. We also could not use the Arduino Uno or like 5V microcontrollers for this purpose without using a level shifter to step down to the 3.3V level used by the ATMEGA328P chip for communication. In a last-ditch attempt, we connected the output of the soil moisture sensing devices to a Teensy 3.2, a 3.3V microcontroller used by solar car to control PCBs, and attempt to read the analog input voltage from the battery-powered sensors on the Dirt Dock as shown in Figure 51. The code for this, shown in Figure 52, while simple did not function correctly even with the Teensyduino library needed for Arduino IDE compatibility and following guidelines of the Teensy manufacturer's website [63].

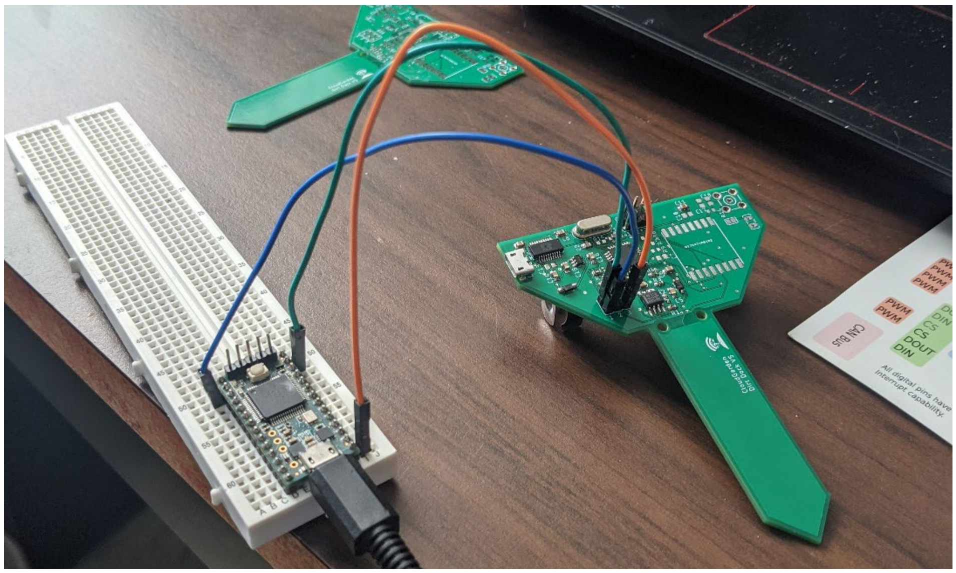


Figure 51: Teensy 3.2 ADC Attempt

Calibration_Teensy

```
int val = 0;

void setup() {
  Serial.begin(9600); // open serial port, set the baud rate as 9600 bps
  pinMode(1,OUTPUT);
  pinMode(A0, INPUT);
  digitalWrite(1, HIGH);
}
void loop() {


  val = analogRead(A0); //connect sensor to Analog 0
  Serial.println(val); //print the value to serial port
  delay(1000);
}
```

Figure 52: Teensy Microcontroller ADC Attempt Code

Having now given up on using an external microcontroller to read analog values, we tried one last Hail Mary of integrating the RFM69HCW transceiver with the microcontroller, programming it, and attempting to send wireless values. On the final day of device prototyping/testing, we found out that integrating the RFM69HCW communication and the ICSP programming method for the microcontroller chip disables the ability to program the chip through the SPI circuit [64]. Other devices such as the

RFM69HCW can respond and transmit out of synch or messages unexpected by the ISCP programmer, leading to a failed connection. Since -within our limited research- we had not encountered any other instances of combining a RFM69HCW into a PCB with a microcontroller chip programmed through the same ICSP, we did not have the time or the resources on the final day to take off the transceiver chips, program, then put the chips back on. As such, we gave up on development of the prototype and completed whatever verification/validation that we could.

5.2 Verification/Validation Plans

The following tests that were planned for the individual subsystems and the final product/prototype to confirm satisfactory creation of a system that fulfills all ER Verification described in Table 2. Tests or validations methods are described in table form with the associated ER and MR that they strive to fulfill. Tests were meant to be performed individually first per each previously described subsystem with documentation of the testing methodology, parameters tested, and results of the tests.

5.2.1 Wireless Transmission

The Wireless Transmission subsystem of the soil moisture sensing devices was meant to be tested for the conditions described in Table 12. One and two-way communication should be ensured in all these scenarios and testing should have been done initially using a breakout board transceiver before integration with the rest of the systems. Tests revolving around transmission with the full device would have been performed during the rapid sampling testing mode to check for effective reception as messages were sent.

Table 12: Wireless Transmission Testing

ER	MR	Testing Procedure
4	2	Wireless transmission and reception of messages from the soil moisture sensor will be confirmed over the minimum range of 500 meters without line-of-sight. To suit the intended usage for potted plants or field deployment, maximum range testing environments will include: • Dense urban settings through walls and concrete • Flat rural areas without obstruction • Hilly, forested areas with natural obstructions
5	2, 4	Transmission reliability will be determined by the percentage of data packets lost between the soil moisture sensors and the central hub. The central hub will know the number of bytes or packets that should be received and will calculate this percentage upon reception and processing of transmissions from the sensors.
10	5	Power or current consumption of the transceiver will be measured for the following modes or operations: • Single transmission • Single reception/processing • Low power idle state

The first test described in Table 12 focused on the marketing requirement for wireless transmission and the engineering requirement for wireless transmission over a minimum distance of 500 meters without line-of-sight. The test's purpose was to determine if the transmission strength and range fall significantly depending on the surrounding environment or barriers between the sensor and central hub and if we can transmit the previously stated minimum distance. If the device was not sufficient, changes could've been made to the antenna and transceiver, if time allowed, to increase signal strength.

The second test focused on the reliability of transmission to assured consistent, reliable, and hourly measurement transmissions by counting the number of received data within the transmission and comparing that to what is expected at the central hub. The result of transmission testing should have been no more than one sampling period's worth of data lost a day, to ensure that the user only missed at most one (1) hourly reading. The protocol is not subject to change due to the construction of the user accessible hub but as stated before, the antenna could have been modified to enhance effective transmission.

The final test was intended to confirm transceiver power consumption in low power state and active state transmission/reception using an oscilloscope in the completed prototype. By reading the voltage over a known resistor, we would have estimated the current draw and ensure the actual power consumption is within a reasonable range of the estimate derived from the datasheet for battery life estimation. If varying from our current estimates, we would have recalculated the new battery life estimate and verified whether it fits within our target device life of 2 years.

5.2.2 Central Hub

Testing of the Central Hub revolved around the effective reception, processing, and presentation of data retrieved from the soil moisture sensors to the user as described in Table 13. Early testing revolved around the usage of a Raspberry Pi and a LoRaWAN Gateway or receiver to verify message reception capabilities, encoding transmissions, and data presentation. The final prototype central hub hosts a webpage on a local network to handle the input for mode selection and showcase received measurements to the user. Transmission comparison and validation was meant to confirm successful and effective usage of the LoRaWAN wireless transmission. "Tech savvy" as mentioned within Table 13's second test is meant to indicate someone who is not familiar with electrical connections, circuits, or adept computer usage as is assumed of the average home gardener.

Table 13: Central Hub Testing

MR	ER	Testing Procedure
3	6	Transmissions will be compared between those recorded at the deployed sensor and the received data at the central hub to verify successful wireless transmission.
3	7	User accessibility will be tested by introducing the web interface to an individual who is not tech savvy and confirming that they are able to use it to clearly interpret output data.
3	8	Using a mostly discharged battery and a higher sampling rate, run the battery dry and verify that the user is given an indication before the device dies.
11	22	Multiple prototype soil moisture sensing units will be populated and used simultaneously to confirm parallel data transmission, logging, processing, and presentation from multiple sensors.

5.2.3 Power Sourcing and Sensing

The Power Sourcing and Sensing subsystem focused on the testing of the battery sensing capabilities our prototype system and confirming the current draw estimated within our battery life spreadsheet/calculations as described in Table 14.

Table 14: Power Sourcing and Sensing Testing

MR	ER	Testing Procedure
3, 6	8	Tested by applying voltages to the device above and below the voltage threshold to ensure we send an end-of-life indicator as it drops below the voltage threshold. A follow-up test will involve discharging a battery using a programmable load at 3.4V using a 1A current draw, attaching it to a soil moisture sensor, and monitoring the device's state as it deactivates when the battery runs out.

4, 6	9, 10	We will retrieve voltage measurements from the battery monitor IC or the microcontroller's ADC and compare to an external multimeter measurement in 3 different trials at varying levels of discharge.
5	11, 12	Apply the data sampling algorithm and analyze the current draw through a known resistor's voltage drop alongside the battery monitor to verify expected current draw.

The first test revolved around verifying that our planned method of end-of-life estimation, voltage measurement, was processed correctly and sends a message to the user at the central hub when we fall below the intended voltage. The follow-up test was meant to ascertain the time difference between the voltage end-of-life warning and the end-of-life of the device to ensure the set voltage threshold is not too early or late of a notification than the desired 5-10%. Too early in this case would have been a warning multiple months before end-of-life and too late would be within the hour of device death operating under the typical data sampling algorithm. The second test is integral to the first test but separate in its purpose as it is meant to verify the precision of the voltage measurements from the microcontroller ADC or battery monitor IC. We intended to ensure that the currently constructed battery monitor code effectively collected the battery voltage and sent it to the microcontroller, printing it out while using the maximum current load to check out own battery life predictions. This would have been double checked with either an external multimeter or oscilloscope depending on the necessary precision or noise reduction needed for an accurate measurement and comparison. If not of the desired accuracy, we would swap modes of measurement or use another method to estimate battery SOC using the battery monitor IC. The final test would have served as a confirmation of total system current draw while temperature and soil moisture were being actively sampled to compare with our current battery life estimate. If lower or matching current draw on the battery estimate spreadsheet, we would have surpassed the target device lifetime target and if higher, components or pieces of the system may have had to change to achieve the target current draw.

5.2.4 Soil Moisture Sensing

The tests shown in were created to validate the integrated soil moisture sensing subsystem's effectiveness in delivering upon the constraints and goals relevant to non-contact soil moisture sensing.

Table 15: Soil Moisture Sensing and Enclosure Tests

MR	ER	Testing Procedure
1,3,4	1, 3	Tested by leaving the wireless device in soil and varying the moisture content to verify correct soil moisture level calibration.
12	23	Test 2 different water compositions within potted plants to ensure the device can predict soil moisture level after recalibration with low variability.
2	5	Test using a software mode to send timing values and compare with the time of reception at the central hub.
5	12	System current consumption will be verified using a known resistor and an oscilloscope or multimeter reading its voltage to acquire system current draw
5	11	Soil moisture sensing waveform output will be plotted shortly after toggling the system on to decide whether a delay should be enacted for soil moisture sensing sampling due to transient behavior.
7	14, 15	The enclosure's waterproofing will be tested with the IP-54 Test described in section 8.2 in the Appendix by splash the device with water from all angles, covering it in sand, and verifying neither enter the enclosure.

The first test was intended to have been completed ensure that the device can measure and decide between levels of soil moisture without measurement reliability degrading over time as described in ER 1 and 3. The physical verification of soil moisture through a resistive soil moisture sensor or touch would have served as the control that the measurements would be compared against to ensure correct estimation of soil moisture percentage. The second test focused on ensuring low variability due to differences in nearby water's ionic moisture content through recalibration prior to entering the ground or potted plant. A software defined testing mode would have been made to increase the sampling rate and report the time value recorded on the microcontroller, transmit a signal, and compare between the time of transmission and reception at the central hub. This ensures that we could have effectively controlled the sampling time of measurements taken on the device without large time delays between measurement acquisition and transmission. Confirmation that the circuit consumed the 5 mA of current as previously found in the DFRobot datasheet for a commercial capacitive soil moisture sensor [9] would be evaluated using similar methods as used for other subsystems, voltage measurements across a resistor of a known value. Finally, whether transient behavior is witnessed shortly after toggling the soil moisture sensing subsystem would have determined if a delay is necessary between activation and stable waveform generation and measurement.

Waterproofing trials for the soil moisture sensors, or verification of ER 14 and 15, are described by the IP-54 rating requested by our sponsor. A rating of IP-54, described in the Appendix and in Figure 80, indicates that a device can be splashed with water from all angles and covered in sand without either water or sand being able to move into the device's enclosure.

5.2.5 Controls and Timing

The Controls and Timing subsystem focused on tests, shown in Table 16, for the control of subsystem power and configuration as well as the periodic sampling timing which are both dependent on the Dirt Dock microcontroller. The first test focused on ensuring our current estimates, derived from third party websites [23], were accurate to the power consumption of our system. If not, we would have changed our battery life estimate accordingly and evaluated whether other systems needed to change to meet this requirement. Testing the sampling period of the microcontroller as mentioned in the second test could have been accomplished by logging measurement time in an accelerated sampling mode verifying the sleep mode was being enacted for the desired time. The next 4 tests in the list were related to checking the effective utilization of the microcontroller's I/O pins to read analog values and communicate with other subsystems to receive, process, and transmit measurements.

Table 16: Microcontroller Testing

MR	ER	Testing Procedure
5	10	Verification of ATMEGA328P current draw estimates during normal operation using a known resistance value and the voltage over it with an oscilloscope or multimeter.
2	5	Testing of our sleep mode programming through logged measurement times on the microcontroller side and comparison with our prediction of expected sampling time per period.
2, 3	6	Verification that we can communicate via I2C to the transmitter from the microcontroller and back.
1, 3	1	Verification that the ADC inputs of battery voltage, soil moisture, and temperature sensing can be read by the microcontroller.
4, 10	20, 21	Verification that the recorded ambient temperature read from the temperature sensing subsystem matches an externally recorded temperature from a nearby thermometer or infrared camera.

5.3 Completed Verification and Validation

Using what we had of a functional system, we completed as many of the verifications or tests as we could to confirm engineering requirements from completed systems using partial or full tests.

5.3.1 Wireless Transmission Test

Following much debugging and trial and error testing, wireless transmission became functional for a short amount of time. The testing setup used in Figure 53 was between a PCB prototype for the project with no onboard microcontroller and an RFM69HCW transceiver, together with a RFM69HCW breakout board. Each of these were connected to different microcontrollers to test functionality. Functionality was eventually achieved with this setup. When tests were repeated with the same setup later it ceased to function. There remain a few hardware issues in the setup which prevent consistency. Functionality was never achieved for a long enough time to test range according to the testing plan above.

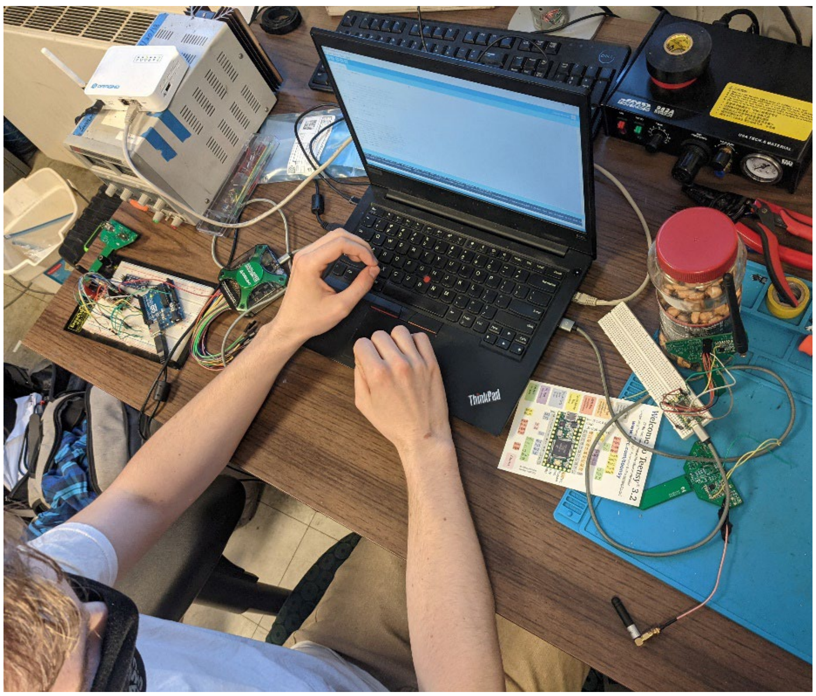


Figure 53: Wireless Communication Troubleshooting/Testing

```
  // NOTE: some RF69 modules use high power output,
  //       those are usually marked RF69H(C/CW).
  //       To configure RadioLib for these modules,
  //       you must call setOutputPower() with
  //       second argument set to true.
  Serial.print(F("[RF69] Setting high power module ... "));
  state = radio.setOutputPower(20, true);
  if (state == ERR_NONE) {
    Serial.println(F("success!"));
  } else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
  }
  printFailure(radio.setFrequency(900.0));
  printFailure(radio.setBitRate(1.2));
  printFailure(radio.disableAddressFiltering());
  uint8_t syncWord[4] = {0x01, 0x23, 0x45, 0x67};
  printFailure(radio.setSyncWord(syncWord, 4));
  printFailure(radio.enableSyncWordFiltering());
  printFailure(radio.setPromiscuousMode(false));
  printFailure(radio.disableAES());
  printFailure(radio.setCrcFiltering(false));
  printFailure(radio.setDataShaping(RADIOLIB_SHAPING_NONE));
  printFailure(radio.setEncoding(RADIOLIB_ENCODING_WHITENING));
  printFailure(radio.fixedPacketLengthMode(14));
}
```

Figure 54: Transceiver Setup Code

The setup code, shown in Figure 54, was for the previously specified RFM69HCW transceiver, connected to a ATMEGA328P microcontroller. The options chosen here were to set the minimum frequency and bit rate to increase range, remove address filtering to accept packets from any device on the network, enable sync word filtering to isolate all devices using the same sync word, disable encryption and disable cyclical redundancy checking. This design resulted in a successful transmission range of approximately two feet. Further ranges were not tested.

During testing, it was noted that interfacing between most non RFM69HCW transceivers required additional setup due to the default modulation schemes of other transceivers. All other transceivers used for testing of the device defaulted to LoRa modulation, which was a modulation scheme unsupported by the RFM69HCW. Instead, FSK modulation was used for testing. This issue could be remedied by replacing the RFM69HCW with an RFM95 series transceiver, which supports LoRa modulation.

5.3.2 Central Hub Test

For the central hub, we were unable to verify the wireless or data transmission components of the system due to unreliable wireless transmission via the LoRaWAN transceiver. Still, the website, with a webpage shown in Figure 55, works and can be accessed remotely but has minor issues with correct formatting and network configuration of webpage attributes. Some attributes intended to be stored locally or hosted by the Raspberry Pi show up as alias's pointing back to the Raspberry Pi's routing address. Fake data was input to the system from previous soil moisture sensor testing to test data processing and we were able to confirm that processing and logging should work with multiple connected devices.

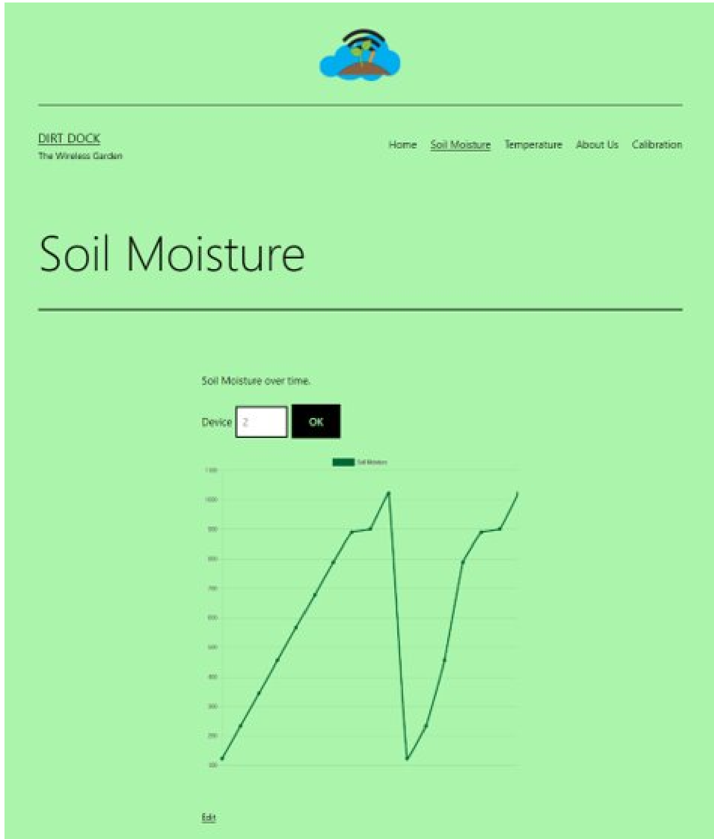


Figure 55: Central Hub Webpage

5.3.3 Power Sourcing and Sensing Test

Upon receiving the batteries for our project from our sponsor, shown in Figure 56, we used a Keithley 2380-500-30 Programmable Load to attempt capacity and voltage tests, shown connected using leads attached to the batteries in Figure 57 and described in the earlier verification/validation plan section. Since the batteries arrived in two lots, one labelled “DEAD” and one labeled “USED”, we took a battery of each type and had intended to discharge at a high current of 5A or so to calculate how much capacity was left in the batteries as we had received them. However, checking the battery data sheet led us to realize that the maximum continuous discharge was 130mA which was just enough for our device as 180mA was available in pulses but insufficient for the 5A test [28]. We pivoted towards using a 130mA constant current draw and found that within 180 seconds, the battery labeled “DEAD” fell to 1.360 V while the battery labeled “USED” fell to 2.915 V. Extrapolating this to a 8mA voltage discharge curve from Figure 19 to approximate our typical use for most of the sampling period indicated that the “DEAD” batteries were mid voltage discharge within the exponential decline of the curve while the “USED” batteries were at about half capacity.

Receiving these batteries at half capacity meant we would be unable to determine an accurate estimate of when a voltage threshold would trigger hours or life wise for the device. Since the device does peak at around 130mA for wireless transmission which pulls the voltage as low as 2.8V within most of the device lifetime, we decided to put the voltage threshold around 2.8V or around 5% health within the voltage discharge curve. Since the voltage of the battery would be read before the transmission, what draws the peak voltage, this voltage value covers both microcontroller usage with other subsystems activated and the transmission period. Battery voltage measurement using a bandgap reference worked as designed on an ATMEGA328P chip of an Arduino Mini Pro connected via breadboard to a battery power source but was not tested with our prototype device due to a lack of digital output.

Figure 56: Batteries received from our Sponsor

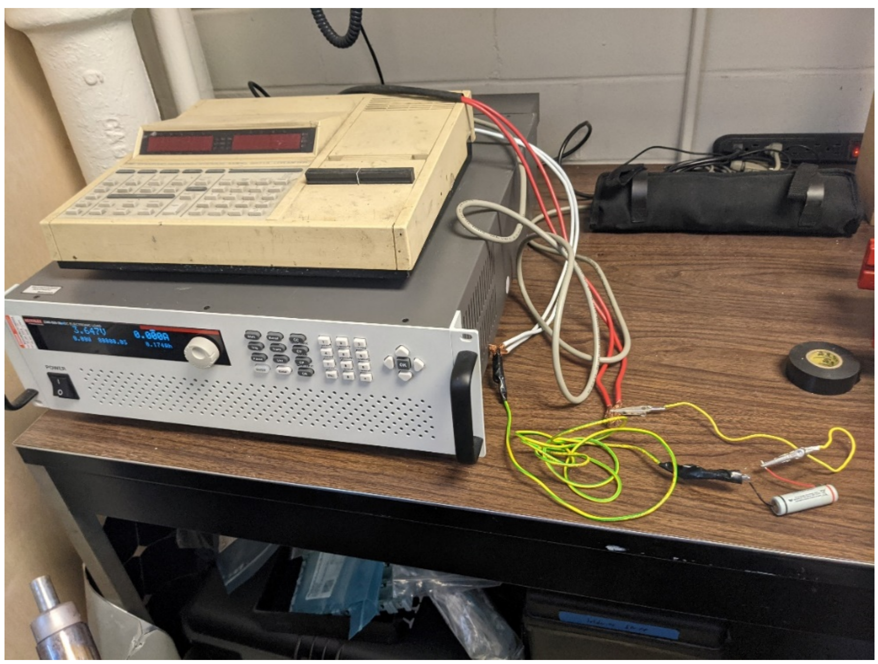


Figure 57: Battery Capacity Testing

After the initial creation of a prototype with a functional microcontroller and prototyping setup, we attached a 6.8kΩ resistor and red LED in place of the thermistor voltage divider circuit and programmed it with a simple blink function, shown in.Figure 58. Plugging a battery into the battery holder activated the light as intended within the sketch and, shown in Figure 59, confirmed that the battery could effectively power the device wirelessly.

```
Blink_Test
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

#define LEDP PD4

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(LEDP, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(LEDP, HIGH);// turn the LED on (HIGH is the voltage level)
  delay(500);
  digitalWrite(LEDP,LOW);
  delay(1000);
}
```

Figure 58: Wireless Program Testing

Figure 59: Functional Battery Operation Test

5.3.4 Soil Moisture Sensing Tests

Our testing setup used the code shown in the subsystem summary from Figure 35 alongside the serial port of the Arduino Uno connected to our computer through PuTTY, a free SSH client for Windows, as shown in Figure 60, to log the values in .csv, convert the csv to an Excel or .xlsx file, and plot the resulting soil moisture change over time. This code was verified to be functional using a SongHe capacitive soil moisture sensor, shown in Figure 61, as the baseline.

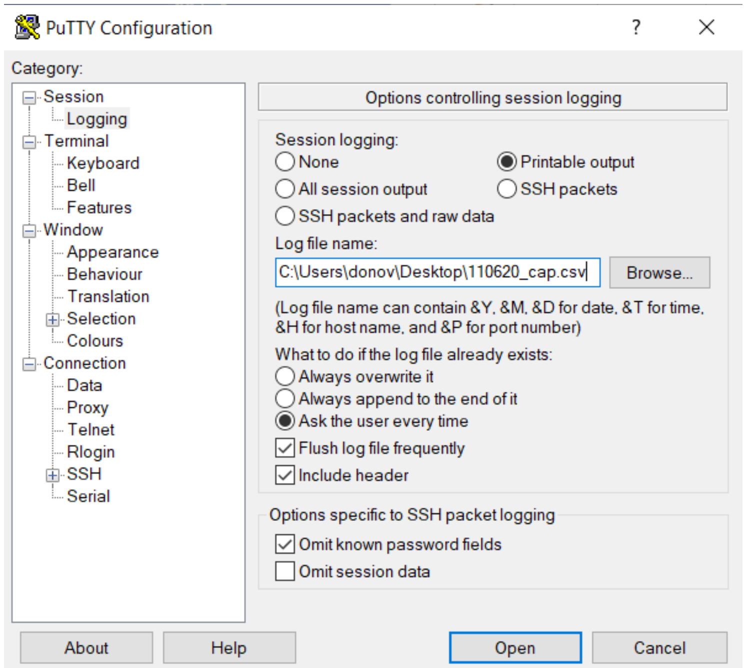


Figure 60: Using PuTTY to log soil moisture measurements.

Figure 61: SongHe Capacitive Soil Moisture Sensor

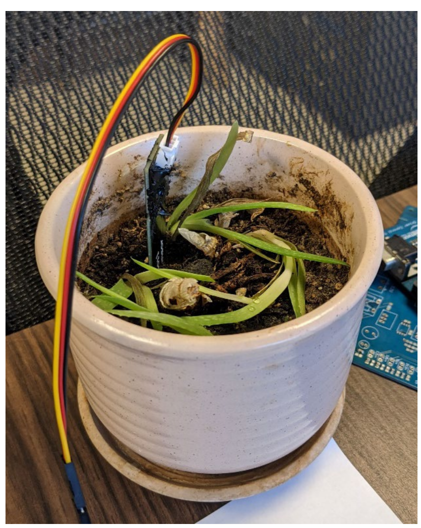


Figure 62: Commercial Capacitive Soil Moisture Sensing Testing

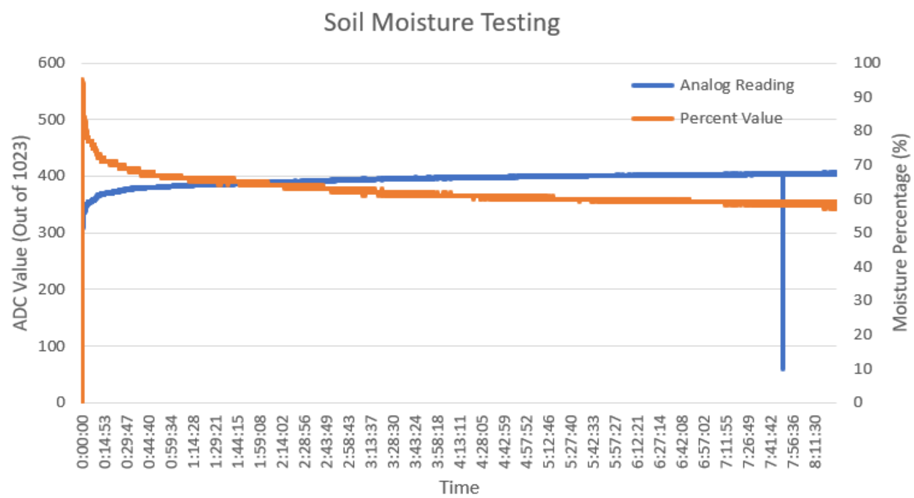


Figure 63: SongHe Capacitive Soil Moisture Sensor Testing Results

Using the logging soil moisture sensing code, we connected a SongHe capacitive soil moisture sensor [65], shown in Figure 61, to an Arduino Uno and inserted the device -covered in liquid electrical tape to quickly waterproof it- into a plant as shown in Figure 62. With an initial pour of water right next to the capacitive soil moisture sensor, we sampled the relative soil moisture surrounding the device every second for 8 hours and plotted it in Figure 63. As shown in the legend, the analog reading or ADC output of the read voltage is displayed in orange with the left axis and the percentage value between the two calibrated endpoints is shown in blue and oriented with the right axis. The voltage spike around the 8 hours mark could have been an issue but was excluded by the map because it was not within the range between air and water, clearing the error from the percentage output. While there is an initial spike in moisture percentage within 15 to 20 minutes of initial water exposure to the area, the overall trend was very consistent second to second over the 8-hour period. This performance indicated that the data could be sampled hourly to capture the majority of soil moisture variation within the topsoil while still maintaining relative stability.

To test our own capacitive soil moisture sensing circuit and decide between hardware and software defined waveform generation, we designed and purchased a custom PCB, shown in Figure 64, and implemented the same or a similar type of test. To allow for greater utilization of the ADC range, however, we also tied the ADCs to an external voltage reference of 3.3V with the resulting logged soil moisture values shown in Figure 65. We increased the overall length of the PCB with the assumption that the larger the area or coverage of soil, the more the capacitance would change as soil moisture did. What we found, however, was that the moisture percentage increased overtime rather than decreasing and variation with additional water was incredibly small compared to the commercial sensor test, as shown within Figure 65. At around 81 minutes in, we added a significant amount of water and there was little to no change to the reading but sticking my thumb to the top portion of the PCB capacitor sticking out of the soil led to a large jump in percent value around what we would have expected with the added water. Our assumption for what caused this variation is that the PCB Capacitor section was too long for the pot we tested it with and the ground plane stretching across the back of the board had offset the soil moisture sensing measurements with its own capacitive behavior with the sheet on the top layer meant for sensing. The capacitance measurement was then largely skewed towards the fringing field value from the air calibration, so we corrected our earlier assumption about size and focused more on fitting it more exactly to the potted plant or topsoil depth.

Figure 64: Custom Soil Moisture Sensor Testing

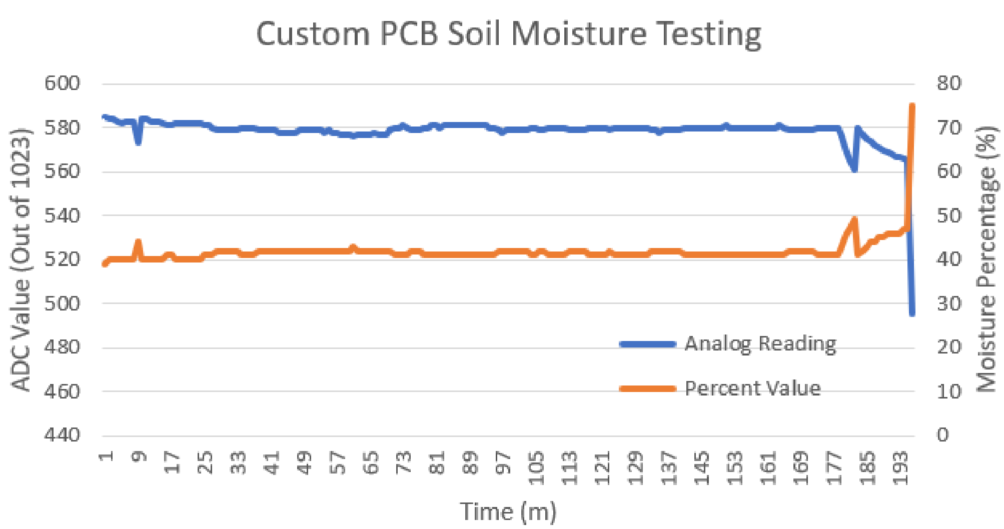


Figure 65: Custom Soil Moisture Sensing Results

Soil moisture sensing was tested physically on our own prototype, shown in Figure 66. However, due to a lack of Serial communication and failure to read ADC values using an external 3.3V microcontroller, we were unable to record and plot these values. As this was after attempting final integration testing, all electronics were removed from a RFM69HCW module to allow the microcontroller to be programmed again. To test this system, we programmed the microcontroller to output power to the soil moisture sensing circuit and read the output by connecting an oscilloscope across the soil moisture sensing output header and ground, as shown in Figure 67, which read 3.21 V by default. Pressing my finger on the capacitive plane, shown in Figure 68, linearly changed the output voltage to 1.88 V as the RC filter's impedance changed. Further testing involving a plant showcased the similar behavior as water was added to the plant around the microcontroller, shown in Figure 69 and at 1.67 V. The haphazard nature of both the connections to the header and a lack of waterproofing at the time of this test did not allow for long duration testing of the device.

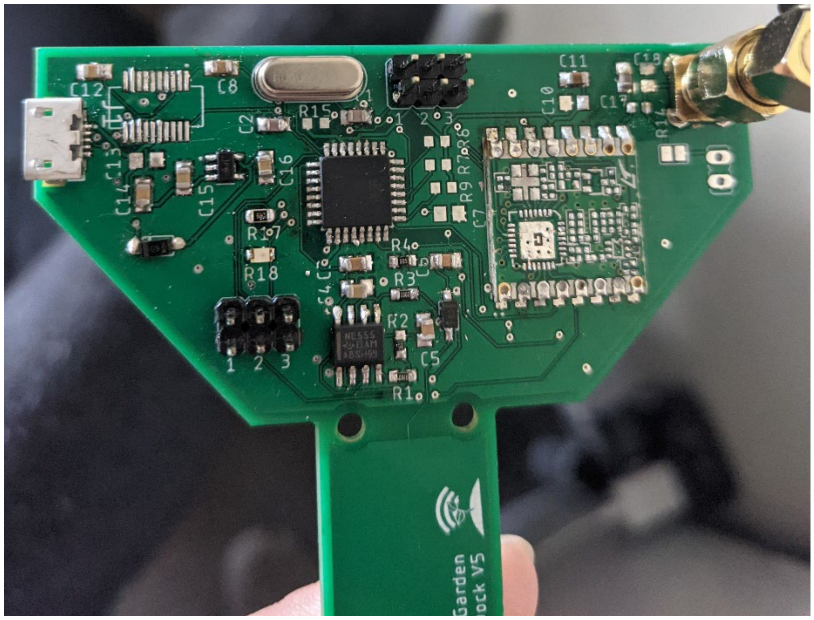


Figure 66: Final Prototype for Soil Moisture Sensing Tests

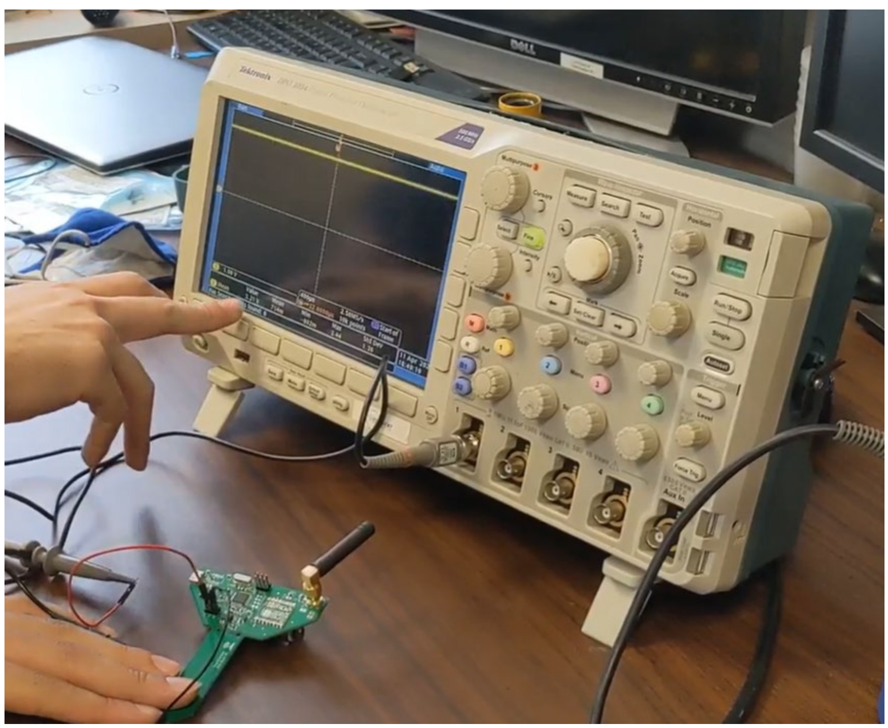


Figure 67: Soil Moisture Measurement in Air

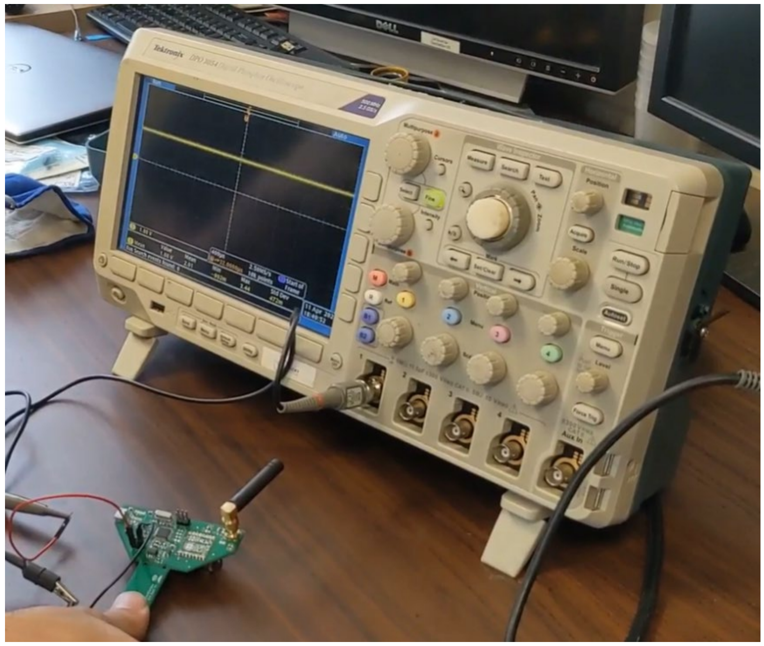


Figure 68: Voltage Change with Capacitamce Variation

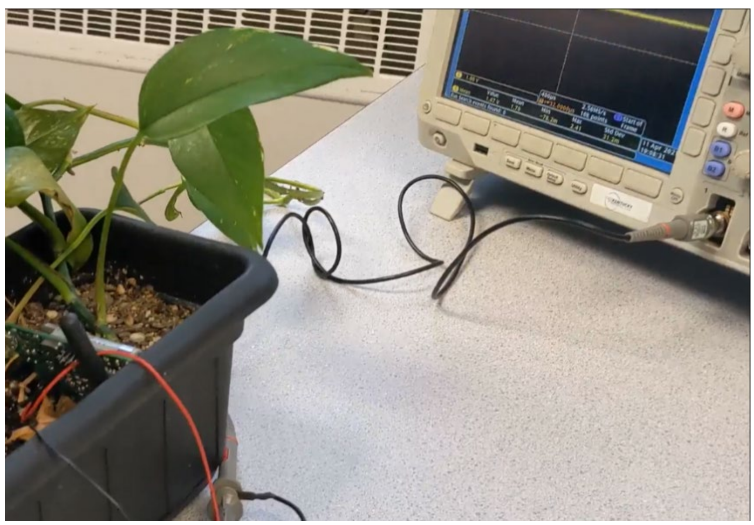


Figure 69: Soil Moisture Measurement in a Wet Plant

5.3.5 Controls and Timing Tests

We tested the low power mode code and current consumption for the microcontroller and basic algorithm configuration using an Arduino Pro Mini, USB to FTDI converter, and a USB cord connected to a computer to turn LEDs on and off with hibernation/low power mode. For voltage measurement, we used an Analog Discovery 2 along with Waveforms software to act as an oscilloscope with a resistor of value 1.1Ω in series with the Pro Mini on the side of ground. The board was powered by a voltage converter connected to a 9V battery to reduce noise on the input system voltage. After powering the board, we obtained the voltage across this resistor and using the known value of the resistor, calculated the current flowing through it using Ohm's law as shown in Figure 70. This process was repeated for the active and sleep state tests using otherwise blank code to determine the device's default power draw using the low-power library like what was described in the design summary for Figure 42 [26].

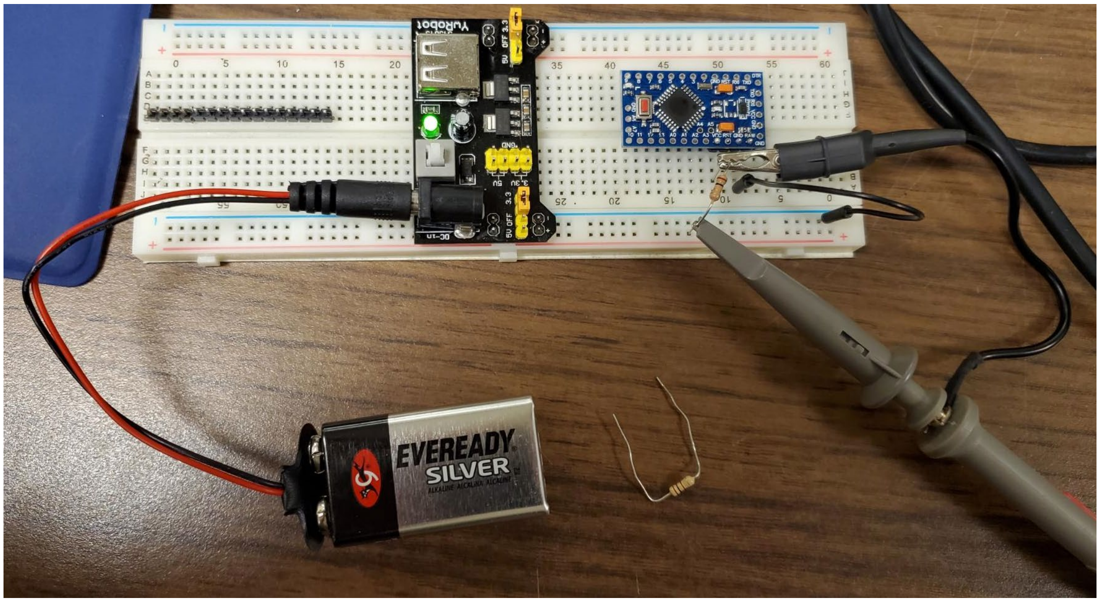


Figure 70: Microcontroller Power Draw Test Circuit

In Figure 71, you can see the voltage readings we obtained by measuring in the active state which peaks around 25mV. Calculating the current draw using the 1.1 Ω resistor results in a peak current draw of around 22.7 mA which is nearly 5 times our estimated current draw of 5mA source for the same device. Figure 72 showcases the voltage readings across the resistor when repeated in the device's low-power mode. As shown in the figure, the voltage across the resistor appears to peak at around 6.5mV resulting in a current draw of about 5.91 mA in the low power mode. While it is a reduction in overall current draw to around a fifth, it is still significantly higher than the 4.3 uA reported by external sources [23, 24]. We believe that these values may partially be a result of a lack of resolution or large noise, however we were unable to prove otherwise.

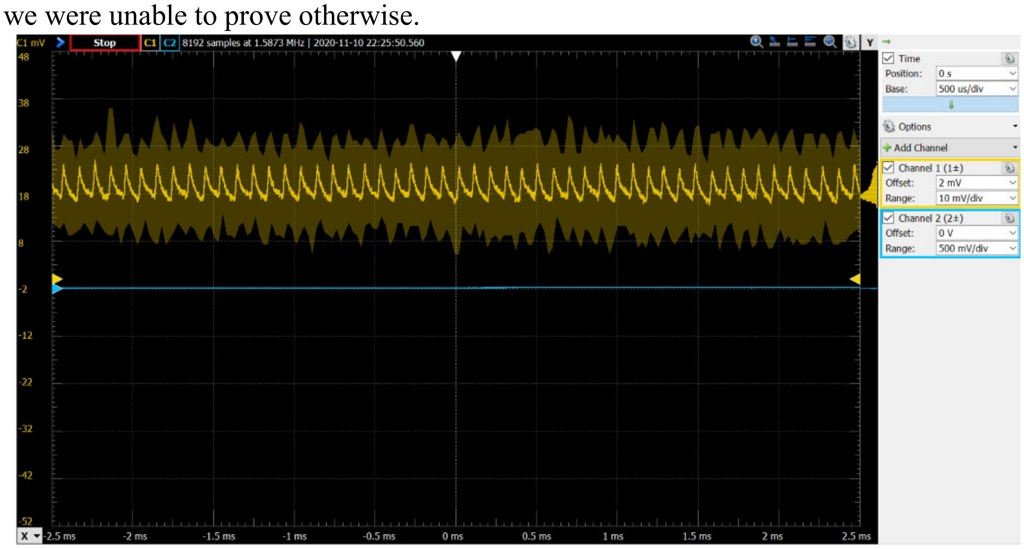


Figure 71: Measured voltage for active state current calculation

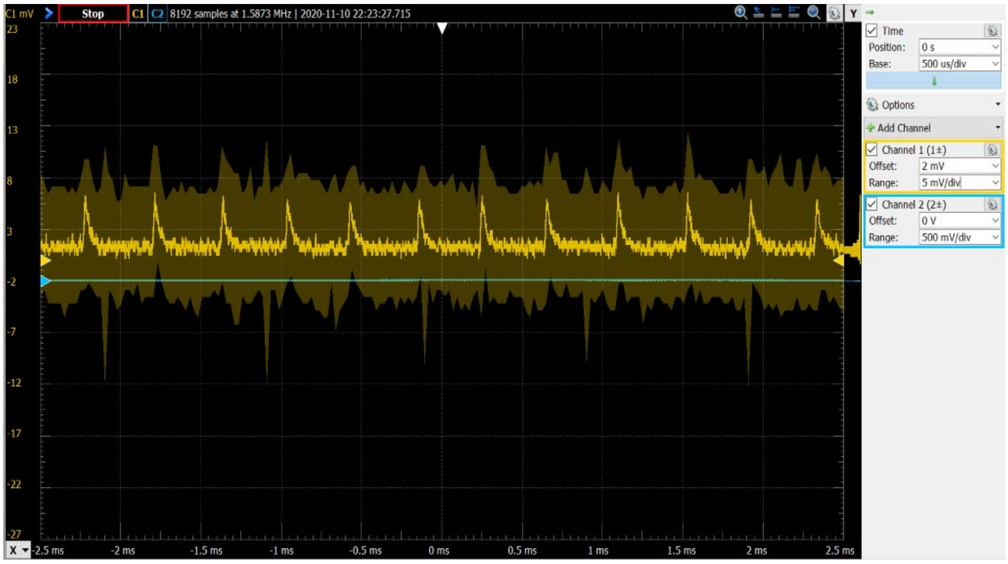


Figure 72: Measured voltage for low power mode current calculation

As described in section 5.1.3 and 5.3.3, we were able to successfully program the microcontroller and output power from the microcontroller to turn subsystems on and off. SPI communication functioned properly as it was used to program the boards but once the transceiver was attached, sync issues across the SPI led to communication failure. As shown above, activating low power mode did work effectively using a while loop, drastically reducing current consumption of the microcontroller. The thermistor circuit also worked as intended with a prototype board functioning off battery power showing voltage fluctuations on an oscilloscope with significant changes in temperature. Figure 73 shows the voltage of about 1.44V at ambient room temperature and Figure 74 shows a voltage of around 833 mV with 200ºC of heat applied with a heat gun.

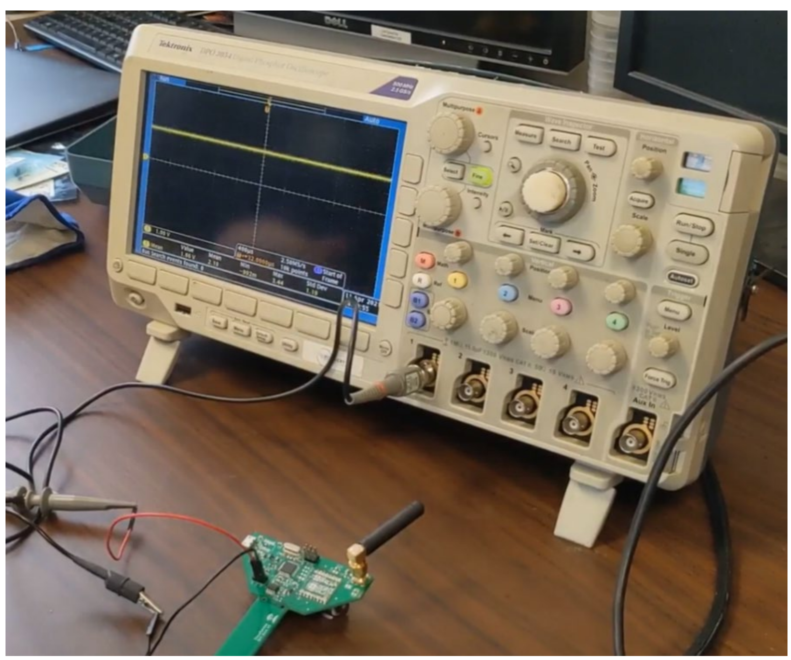


Figure 73: Temperature sensor output without applied heat

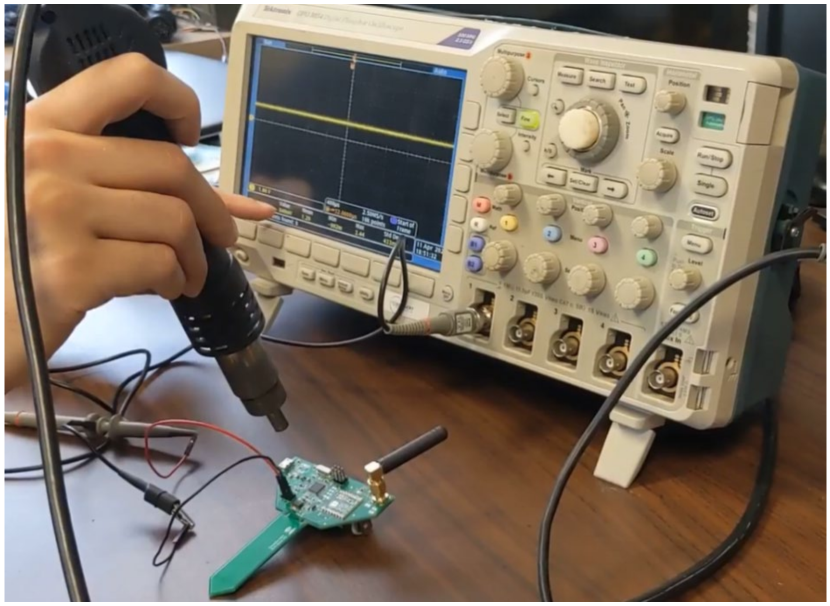


Figure 74: Temperature lowering voltage output

5.3.6 Enclosure and Waterproofing Tests

With the attached enclosure, the Dirt Dock, pictured in Figure 75, is within the size constraints at 4.10” long, 1.95” wide, and 3.277” deep. Splashing water on the device’s enclosure as shown in Figure 75 and described in the IP-54 test description within the Appendix resulted in no particle or water ingress into the electronics inside shown in Figure 76. As the final prototype would be coated in silicone, condensation, and other methods of water entrance into the system would be fully blocked or sealed off.

Figure 75: Prototype prior to IP-54 Testing

Figure 76: Dry prototype after removing enclosure.

6. Final Costs

To ensure compliance with the budgetary constraints concerning our project, we created cost breakdowns for the Dirt Docks and the central hub. The devices themselves are limited to a budget of $30 which is included with the central hub and other pieces necessary for initial prototyping in a total budget of $500. Figure 77 is a tabular breakdown of the device cost per unit while Figure 79 is a breakdown of the cost for a single central hub.

As it currently stands and without shipping or tax factored in, a device costs $34.33 to make while the central hub and processing costs $78.43 to create which is well within the total budget but not within the budget for individual devices. Within the device cost estimate, the yellow highlighted rows are for FTDI/USB prototyping and would not be necessary for the final product with programming performed through the ICSP connection [48]. The PCB prototypes also come in a 5 for 5 deal which is why its price is divided by 5 to result in $1.00 each and the price would be smaller for bulk purchases. If the FTDI/USB and battery monitor IC circuits were removed, as totaled in Figure 78, the total cost of the device would be $27.60, meeting our sponsor's budget restrictions. The price would most likely drop with increased quantity and the batteries and transceiver chip were the most expensive components in the system, totaling 43% of the total device cost.

Prototyping costs include two PCB revisions and 4 times the components needed for one Dirt Dock for use in initial construction/testing of the system. The total prototyping cost comes to $179.42 with the two PCB revisions which cost $26.00 each for medium speed shipping, two Digikey orders, the first with enough components for 1 prototype which cost $49.24 and a second with enough parts for 3 prototypes and costing $56.56. For prototyping, USBasp programmers were purchased from Amazon for $10.59 and an RFM69HCW breakout board was purchased from Amazon for $11.03. When reached out to regarding funds for project prototyping, the sponsor indicated that we should purchase it ourselves and would then be refunded for these costs later.

Qty	Value	Device	Parts	Digikey Part Number	Digikey Price	Itemized
1.0		JP2E	JP1	HDR100IMP40M-G-V-TH	$0.69	$0.69
1.0		JP3Q	JP2			$0.00
1.0	0.1	R-US_R0603	R11	A103277CT-ND	$0.23	$0.23
6.0	0.1u	C-USC0805	C7, C8, C11, C12, C14	1276-1099-1-ND	$0.10	$0.60
1.0	0.47u	C-USC0805	C9	1276-6480-1-ND	$0.10	$0.10
1.0	1M	R-US_R0603	R4	541-CRCW06031M00JN	$0.10	$0.10
1.0	1k	R-US_R0603	R14	541-3949-1-ND	$0.10	$0.10
3.0	1u	C-USC0805	C6, C15, C16	1276-1026-1-ND	$0.10	$0.30
1.0	8 MHz	XTAL/S	Q2	2151-AS-8.000-20-ND	$0.18	$0.18
1.0	10	R-US_R0603	R10	13-RE0603FRE0710RLCT-ND	$0.10	$0.10
3.0	10k	R-US_R0603	R3	541-5136-1-ND	$0.10	$0.30
3.0	10u	C-USC0805	C10, C13	1276-6455-1-ND	$0.11	$0.33
2.0	22p	C-USC0805	C1, C2	1276-1047-1-ND	$0.10	$0.20
1.0	100	R-US_R0603	R9	541-3951-1-ND	$0.10	$0.10
1.0	120n	C-USC0805	C4	399-9171-1-ND	$0.13	$0.13
1.0	140n	C-USC0805	C3	732-8040-1-ND	$0.10	$0.10
2.0	150	R-US_R0603	R15, R16	541-4023-1-ND	$0.10	$0.20
1.0	330	R-US_R0603	R1	541-CRCW0603330RFKEBCC	$0.10	$0.10
1.0	660p	C-USC0805	C5	732-8067-1-ND	$0.10	$0.10
1.0	915MHz	RFM69HCW-915MHZ	U3	1568-1394-ND	$5.95	$5.95
1.0	1600	R-US_R0603	R2	541-5237-1-ND	$0.10	$0.10
1.0	4.76E+08	475900001	J1	2057-MCR-B-S-RA-SMT-CS5A	$0.34	$0.34
1.0	ATMEGA32	ATMEGA328P_VQFNVQ	U1	ATMEGA328P-AU-ND	$2.32	$2.32
1.0	BU-SMA-V	BU-SMA-V	X1	2057-RF2-04A-T-00-50-(	$1.29	$1.29
1.0	DIODE_SCH	DIODE_SCHOTTKY_CMI	D11	1727-7328-1-ND	$0.13	$0.13
1.0	FT231XS-R	FT231XS-R	U4	768-1129-1-ND	$2.12	$2.12
2.0	KEYESTONE	KEYESTONE_53A_TYPE	U$1	36-53-ND	$0.37	$0.74
1.0	LP2985-33I	LP2985-33DBVR	TLV75533PDBVR	296-50411-1-ND	$0.42	$0.42
1.0	MAX17201	MAX17201G+T	U2	MAX17201G+TCT-ND	$3.85	$3.85
1.0	NE555D	NE555D	IC2	296-6501-1-ND	$0.41	$0.41
2.0	THERM	R-US_R0603	R6, R12	490-16279-1-ND	$0.10	$0.20
0.2	PCB			PCB Manufacturer	$5.00	$1.00
1.0	Antenna		ANT	939-9000962-XLPDNW-I	$3.95	$3.95
1.0	Battery		BAT	1908-ER17505J-SA-ND	$7.55	$7.55
					Total	$34.33

Figure 77: Soil Moisture Sensing Device Prototype Cost Breakdown

Qty	Value	Device	Parts	Digikey Part Number	Digikey Price	Itemized
1.0		JP2E	JP1	HDR100IMP40M-G-V-TH	$0.69	$0.69
1.0	0.1	R-US_R0603	R11	A103277CT-ND	$0.23	$0.23
6.0	0.1u	C-USC0805	C7, C8, C11, C12, C14	1276-1099-1-ND	$0.10	$0.60
1.0	0.47u	C-USC0805	C9	1276-6480-1-ND	$0.10	$0.10
1.0	1M	R-US_R0603	R4	541-CRCW06031M00JN	$0.10	$0.10
1.0	1k	R-US_R0603	R14	541-3949-1-ND	$0.10	$0.10
3.0	1u	C-USC0805	C6, C15, C16	1276-1026-1-ND	$0.10	$0.30
1.0	8 MHz	XTAL/S	Q2	2151-AS-8.000-20-ND	$0.18	$0.18
1.0	10	R-US_R0603	R10	13-RE0603FRE0710RLCT-ND	$0.10	$0.10
3.0	10k	R-US_R0603	R3	541-5136-1-ND	$0.10	$0.30
3.0	10u	C-USC0805	C10, C13	1276-6455-1-ND	$0.11	$0.33
2.0	22p	C-USC0805	C1, C2	1276-1047-1-ND	$0.10	$0.20
1.0	100	R-US_R0603	R9	541-3951-1-ND	$0.10	$0.10
1.0	120n	C-USC0805	C4	399-9171-1-ND	$0.13	$0.13
1.0	140n	C-USC0805	C3	732-8040-1-ND	$0.10	$0.10
2.0	150	R-US_R0603	R15, R16	541-4023-1-ND	$0.10	$0.20
1.0	330	R-US_R0603	R1	541-CRCW0603330RFKEBCC	$0.10	$0.10
1.0	660p	C-USC0805	C5	732-8067-1-ND	$0.10	$0.10
1.0	915MHz	RFM69HCW-915MHZ	U3	1568-1394-ND	$5.95	$5.95
1.0	ATMEGA32	ATMEGA328P_VQFNVQ	U1	ATMEGA328P-AU-ND	$2.32	$2.32
1.0	BU-SMA-V	BU-SMA-V	X1	2057-RF2-04A-T-00-50-(	$1.29	$1.29
1.0	DIODE_SCH	DIODE_SCHOTTKY_CMI	D11	1727-7328-1-ND	$0.13	$0.13
2.0	KEYESTONE	KEYESTONE_53A_TYPE	U$1	36-53-ND	$0.37	$0.74
1.0	NE555D	NE555D	IC2	296-6501-1-ND	$0.41	$0.41
2.0	THERM	R-US_R0603	R6, R12	490-16279-1-ND	$0.10	$0.20
0.2	PCB			PCB Manufacturer	$5.00	$1.00
1.0	Antenna		ANT	939-9000962-XLPDNW-I	$3.95	$3.95
1.0	Battery		BAT	1908-ER17505J-SA-ND	$7.55	$7.55
					Total	$27.50

Figure 78: Final Soil Moisture Sensor Cost Breakdown

Qty	Value	Device	Parts	Digikey Part Number	Price	Itemized
1.0		LoRa Gateway	LoRa HAT for Raspberr	https://www.waveshare	$25.99	$25.99
1.0		Processing	Raspberry Pi 3 B+ 4 GB	1690-1025-ND	$35.00	$35.00
1.0		Processing Power	Pi Power Supply	Amazon.com: CanaKit 5'	$9.95	$9.95
1.0		Processing Storage	Pi SD Card	Amazon.com: Samsung (	$7.49	$7.49
					Total	$78.43

Figure 79: Central Hub Cost Breakdown

7. Results and Lessons Learned

The following sections detail the progression or results of our senior design project as explained prior in our verification/validation and integration sections.

7.1 Project Results Summary

Try as we might, we were unable to complete our prototype as originally designed or intended, completing 13 of 23 ER as shown and explained in Table 17. Soil moisture sensing, temperature sensing, microcontroller programming, and battery operation were confirmed to function as intended on the prototypes, but we lacked a method of communication to get these values from the device, either wirelessly or through a serial connection. Unforeseen hardware integration problems with the RFM69HCW transceivers blocked programming of the devices through the shared connection used for sending and receiving wireless transmissions. A lack of early development regarding wireless transmission and reception prototyping, as described in the work breakdown, led to little understanding as to whether our issues were hardware, software, or both. The Central Hub worked, for the most part, but without wireless transmission to and from the Dirt Dock devices, does not have data to work with, process, and show. Numerous of the failed engineering requirements below, highlighted in red, could have worked but without the digital output or wireless transmissions, we were unable to take long duration measurements in a meaningful way.

Table 17: ER Progress

#	Engineering Requirements	Verification/Validation Status
1	The device shall measure soil moisture and report soil moisture levels.	Soil moisture can be measured and reported between calibration points of completely dry (air) and completely wet (water) to determine between various soil moisture levels.
2	Soil moisture readings shall be taken without direct electrical contact to the soil	Soil moisture readings are taken through capacitive soil moisture measurement.
3	Soil moisture sensing reliability shall not degrade over time.	We were unable to sustain long duration soil moisture sensing tests due to a lack of wireless or stable wired communication.
4	The device shall wirelessly transmit data over more than 500 meters without line-of-sight.	Wireless communication is not functional within our final prototype.
5	Measurements on the device shall be taken every hour.	Measurements can be taken hourly using the onboard microcontroller.
6	The device shall wirelessly transmit all measurements to a central hub.	Successful transmission from device to hub was never accomplished.
7	Measurements shall be easily accessible and understandable from the central hub.	Measurements, while easily accessible, do not make it to the central hub.
8	The device shall track battery voltage to alert the user when the voltage drops below 2.8V prior to device deactivation.	Battery voltage measurements were unable to be tested on the final prototype due to a lack of digital or wireless communication.
9	The device shall be able to measure battery voltage with a resolution of .01 V to predict device end-of-life.	Battery voltage measurements were unable to be tested on the final prototype due to a lack of digital or wireless communication.
10	The device should measure the battery's voltage with a margin of error of a maximum of ± .01V.	Battery voltage measurements were unable to be tested on the final prototype due to a lack of digital or wireless communication.
11	The device's microcontroller will enter deep sleep and power will be cut all subsystems, excluding the energy storage and SOC sensing, to conserve power	Testing was completed to verify power is sufficiently cut to subsystems not currently in use and current draw was greatly reduced within deep sleep mode.
12	The device shall have a battery life greater than 2 years.	Assuming that our approximation of the full device's power consumption is correct, the device would last more than 2 years on a single battery once fully functional. However, we were unable to correct or verify our current draw estimates for the whole system.

13	The device shall sustain functionality through temperatures ranging from -20ºF to 126ºF.	All electronics used within the bill of materials were checked to verify functionality within the effective temperature region of -20ºF to 126ºF.
14	The device’s electronics shall be enveloped within an enclosure.	An enclosure was designed and fully encloses the device’s electronics.
15	The device’s enclosure shall have a waterproofing rating of IP-54 or greater.	IP-54 waterproofing was tested with the enclosure attached but without Silicone waterproofing and the device passed due to the snug fit of the enclosure.
16	The device shall be no larger than 10” long, 2” wide, and 5” deep.	The Dirt Dock is within the size constraints at 4.10” long, 1.95” wide, and 3.277” deep
17	The total parts cost for one device shall not exceed $30.	The cost of the system excluding non-functional or unnecessary systems (battery monitor IC and FTDI to USB circuit) is less than $30.
18	Costs for the entire system shall not exceed $500	The cost of the entire system with nonfunctional systems included is $112.76, well under $500.
19	The device’s electrical components shall be off-the-shelf or available online.	All electrical components for the prototype were purchased from and are available on Digikey.
20	The device should be able to measure air temperature with a resolution of at least 1ºF.	NTC temperature sensing has proven to function as intended and the thermistor voltage divider circuit has a smaller resolution than 1ºF
21	The device should be able to measure temperature with a margin of error of a maximum of ± 1º F.	Long duration or substantial temperature sensing tests were unable to be performed without a digital output
22	Multiple devices should be supported for simultaneous usage.	Single devices were unable to transmit wirelessly
23	The device shall be able to be recalibrated to report relative soil moisture levels.	Software-defined calibration of soil moisture sensors function off of the final prototype and can be implemented on it via programming.

7.2 Lessons Learned

When reviewing this project, there is a large amount of context not present within the technical documentation that predates or led to the overall results in terms of device or prototype effectiveness which will be summarized in the following bullet points:

- Delivery of PCB prototypes during the Chinese New Year
  - What usually takes a week to be made and delivered took 4 weeks because of shipping during the Chinese New Year
  - PCB ordered during the 2nd week of class for the semester and arrived much later for actual prototyping or integration testing.
- Work on implementation and preliminary testing should be accomplished or begun by the time the design specification has been completed.
  - This would have allowed for such revisions as necessary to create a functional prototype in time. Mistakes were made in the initial specification because the parts had not been tested before being used.
  - This would have remedied the issue of not being able to utilize LoRa radio frequency modulation as stated in section 5.3.1.
  - Would have greatly assisted in knowing where problems originated while troubleshooting and integration
- Lack of definition of roles within the beginning and expectations for individuals or subsystems

  - Could have helped with keeping subsystem development on track and helping ensure no one fell behind of the others.
  - This led to confusion when flexibility occurred as team members did now know who to talk to pick up slack or ask how they could help.
- Knowing when to give up and move on
  - Several developments did not occur because members became fascinated with specific portions of their design or stuck within troubleshooting.
  - Timelines should have been established for when to give up in favor of greater progress.
  - Breaks from working on senior design or course related material should've been focused upon for certain members of the group to reduce mental fatigue.
- Work should have been done between semesters.
  - We should have had prototypes ready to go when we got back to school, knowing the delays that other courses would bring.

To summarize these points, we needed to take breaks away from senior design and course work to work more effectively, well define our responsibilities and expectations, start early, perform preliminary testing in the IRDI and distribute IRDI's to all team members, and know when to give up and move on.

8. References

[1] S. Armstrong, "How Does Water Affect Plant Growth?," Gardening Know-How, 16 2020 September. [Online]. Available: https://www.gardeningknowhow.com/special/children/how-does-water-affect-plant-growth.htm. [Accessed October 2020].

[2] "Soil & Plant Moisture Meter," Conservation Mart, [Online]. Available: https://www.conservationmart.com/p-2142-soil-plant-moisture-meter-am-conservation-mm071.aspx. [Accessed 16 September 2020].

[3] R. N. Dean, A. K. Rane, M. E. Baginski, J. Richard, Z. Hartzog and D. J. Elton, "A Capacitive Fringing Field Sensor Design for Moisture Measurement Based on Printed Circuit Board Technology," *IEEE Transactions on Instrumentation and Measurement,* vol. 61, no. 4, pp. 1105-1112, 2012.

[4] J. Mizuguchi, J. Piai, J. A. d. Franca, K. Yamashita and L. C. Mathias, "Fringing Field Capacitive Sensor for Measuring Soil Water Content: Design, Manufacture, and Testing," *IEEE Transactions on Instrumentation and Measurement,* vol. 64, no. 1, pp. 212-220, 2015.

[5] E. F. D. Costa, N. E. D. Oliveira, F. J. O. Morais, P. Carvalhaes-Dias, L. F. C. Duarte, A. Cabot and J. A. S. Dias, "A Self-Powered and Autonomous Fringing Field Capacitive Sensor Integrated into a Micro Sprinkler Spinner to Measure Soil Water Content," *Sensors,* vol. 3, no. 17, p. 575, 12 March 2017.

[6] F. Morais, P. C. Dias, L. Duarte, E. Costa, A. Ferreira and J. S. Dias, "Fringing Field Capacitance Smart Sensor Based on PCB Technology for Measuring Water Content in Paper Pulp," *Journal of Sensors,* vol. 2020, pp. 1-13, 2020.

[7] l. ada, "Adafruit STEMMA Soil Sensor - I2C Capacitive Moisture Sensor Downloads," adafruit, 6 November 2020. [Online]. Available: https://learn.adafruit.com/adafruit-stemma-soil-sensor-i2c-capacitive-moisture-sensor/downloads) . [Accessed October 2020].

[8] "Gravity: Analog Capacitive Soil Moisture Sensor - Corrosion Resistant," DFROBOT, 27 August 2015. [Online]. Available: https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193. [Accessed October 2020].

[9] "Grove - Capacitive Soil Moisture Sensor (Corrosion Resistant)," Seeed Studio, [Online]. Available: https://www.seeedstudio.com/Grove-Capacitive-Moisture-Sensor-Corrosion-Resistant.html. [Accessed September 2020].

[10] D. Baker, "Kentucky Annual Temperatures and Records," coolweather.net, [Online]. Available: https://coolweather.net/statetemperature/kentucky_temperature.htm. [Accessed September 2020].

[11] US EPA, "Watering Tips," United States Environmental Protection Agency, [Online]. Available: https://www.epa.gov/watersense/watering-tips. [Accessed November 2020].

[12] "Battery Recycling is Important for Environmental Health," Gallegos Sanitation Incorporated, 20 January 2020. [Online]. Available: https://gsiwaste.com/battery-recycling-is-important-for-environmental-health/.. [Accessed September 2020].

[13] "IP-54 Enclosures," Nema Enclosures, [Online]. Available: https://www.nemaenclosures.com/enclosure-ratings/ip-enclosures/ip54-enclosures.html.. [Accessed September 2020].

[14] ElectronicWings, "Thermistor Interfacing with Arduino UNO," ElectronicWings, [Online]. Available: https://www.electronicwings.com/arduino/thermistor-interfacing-with-arduino-uno. [Accessed 9 April 2021].

[15] P. Khatri, "Interfacing Thermistor with Arduino to Measure and Display Temperature on LCD," CircuitDigest, 18 December 2017. [Online]. Available: https://circuitdigest.com/microcontroller-projects/arduino-thermistor-interfacing-code-circuit. [Accessed 9 April 2021].

[16] M. Grusin, "RFM69HCW Hookup Guide," sparkfun, 2016. [Online]. Available: https://learn.sparkfun.com/tutorials/rfm69hcw-hookup-guide/all. [Accessed 1 January 2021].

[17] sparkfun, "SparkFun RFM69 Breakout (915MHz)," sparkfun, [Online]. Available: https://www.sparkfun.com/products/12775. [Accessed 6 March 2021].

[18] creative commons, "CC BY-SA 4.0," creative commons, [Online]. Available: https://creativecommons.org/licenses/by-sa/4.0/. [Accessed 16 February 2021].

[19] elprocus, "What is Pi Filter : Circuit, Working and Its Applications," elprocus, [Online]. Available: https://www.elprocus.com/pi-filter-circuit-working-and-its-applications/. [Accessed 6 February 2021].

[20] F. Rusu, "RFM69 Library," GitHub, 26 January 2021. [Online]. Available: https://github.com/LowPowerLab/RFM69/. [Accessed 16 February 2021].

[21] Free Software Foundation, "GNU General Public License," Free Software Foundation, [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.en.html. [Accessed 18 February 2021].

[22] Beningo Embedded Group, "7 Tricks for Estimating Battery Life," Beningo Embedded Group, 27 July 2017. [Online]. Available: https://www.beningo.com/7-tricks-for-estimating-battery-life/. [Accessed 19 January 2021].

[23] J.-M. Dehcriste, "Arduino Pro Mini: Power consumption," IoT Experiments, 25 August 2016. [Online]. Available: https://www.iot-experiments.com/arduino-pro-mini-power-consumption/. [Accessed 3 March 2021].

[24] J. Gruber, "Low power Arduino Pro Mini," GitHub, 27 December 2019. [Online]. Available: https://jackgruber.github.io/2019-12-27-Low-power-Arduino-Pro-Mini/. [Accessed 9 April 2021].

[25] sparkfun, "Reducing Arduino Power Consumption," sparfkun, [Online]. Available: https://learn.sparkfun.com/tutorials/reducing-arduino-power-consumption/all. [Accessed 9 April 2021].

[26] RocketScream, "Lightweight Low Power Arduino Library," RocketScream, 4 July 2011. [Online]. Available: https://www.rocketscream.com/blog/2011/07/04/lightweight-low-power-arduino-library/. [Accessed 19 April 2021].

[27] Battery University, "BU-802b: What does Elevated Self-discharge do>," Cadex Electronics Inc., 2011. [Online]. Available: https://batteryuniversity.com/learn/article/elevating_self_discharge. [Accessed 28 January 2021].

[28] UltraLife, "3.6V 2.0Ah LiSOCl2 Non-rechargeable AA-Size Spiral Cell," UltraLife, [Online]. Available: https://www.ultralifecorporation.com/ECommerce/product/er-legacy/er-legacy/uhr-er14505-h/aa-36v-2000mah-able-label. [Accessed 2 February 2021].

[29] Battery University, "BU-106a: Choices of Primary Batteries," Cadex Electronics, 2011. [Online]. Available: https://batteryuniversity.com/learn/article/choices_of_primary_batteries. [Accessed 19 January 2021].

[30] Arduino, "Measure voltage on the same battery powering the arduino," Arduino, 1 January 2018. [Online]. Available: https://forum.arduino.cc/t/solved-measure-voltage-on-the-same-battery-powering-the-arduino/500237. [Accessed 29 March 2021].

[31] Arduino2021, "Measuring battery voltage conditionally," Arduino, June 2015. [Online]. Available: https://forum.arduino.cc/t/solved-measure-voltage-on-the-same-battery-powering-the-arduino/500237. [Accessed 29 March 2021].

[32] Maxim Integrated, "MAX17201 Fuel Gauge Datasheet," Maxim Integrated, [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX17201-MAX17215.pdf. [Accessed 30 January 2021].

[33] H. Singh, "HOW TO USE THE MAX17201/MAX17211 AND MAX17205/MAX17215 TO FUEL-GAUGE PRIMARY CELL BATTERIES," maxim integrated, 14 July 2017. [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/app-notes/6/6416.html. [Accessed 30 January 2021].

[34] L. Mustafa, "max1720x GitHub Library," GitHub, 5 May 2017. [Online]. Available: https://github.com/IRNAS/max1720x. [Accessed 14 February 21].

[35] Dejan, "How I2C Communication Works and How To Use It with Arduino," How To Mechatronics, [Online]. Available: https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/. [Accessed 27 February 2021].

[36] Arduino, "Measurement of Bandgap voltage," Arduino, January 2011. [Online]. Available: https://forum.arduino.cc/t/measurement-of-bandgap-voltage/38215. [Accessed 25 March 2021].

[37] Scott, "Secret Arduion Voltmeter- Measure Battery Voltage," Provide Your Own..., 9 July 2012. [Online]. Available: https://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/. [Accessed 9 April 2021].

[38] A. S. Tamsir, A. Manut, D. M. Alias, S. Sulaiman and A. Zakaria, "Capacitive Sensor". Malaysia Patent WO2009066992A2, 28 May 2009.

[39] J. Robert Neal Dean, E. A. Guertal and M. E. Baginski, "Capacitive fringing field sensors and electrical conductivity sensors integrated into printed circuit boards". USA Patent US20150338363A1, 26 November 2015.

[40] P. Placidi, L. Gasperini, A. Grassi, M. Cecconi and A. Scorzoni, "Characterization of Low-Cost Capacitive Soil Moisture Sensors for IoT Networks," *Sensors,* vol. 20, p. 3585, 2020.

[41] "TLC 555 LinCMOS Timer," July 2019. [Online]. Available: https://www.ti.com/lit/ds/symlink/tlc555.pdf. [Accessed October 2020].

[42] ElectronicsTutorials, "Passive Low Pass Filter," https://www.electronics-tutorials.ws/filter/filter_2.html, [Online]. Available: https://www.electronics-tutorials.ws/filter/filter_2.html. [Accessed 9 April 2021].

[43] "Interface Capacitive Soil Moisture Sensor v1.2 with Arduino," how2electronics , 2 September 2020. [Online]. Available: https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino/#Capacitive_Soil_Moisture_Sensor_Schematic. [Accessed October 2020].

[44] N. R. Harris and A. Stonard, "A Printed Capacitance Sensor for Soil Moisture Measurement," in *Eurosensors*, Graz, Austria, 2018.

[45] s. fun, "Arduino-Pro-Mini Schematic," Arduino, 9 April 2014. [Online]. Available: https://www.arduino.cc/en/uploads/Main/Arduino-Pro-Mini-schematic.pdf. [Accessed 30 January 2021].

[46] Microchip, " ATmega48A/PA/88A/PA/168A/PA/328/P megaAVR® Data Sheet," Microchip, [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf. [Accessed 28 january 2021].

[47] Spark Fun Electronics, "Arduino Pro Mini Schematic," Adafruit, 9 April 2014. [Online]. Available: https://www.arduino.cc/en/uploads/Main/Arduino-Pro-Mini-schematic.pdf. [Accessed 30 January 2021].

[48] SM, "Arduino as ISP and Arduino Bootloader," Arduino, 20 January 2018. [Online]. Available: https://www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoISP#use-arduino-as-isp. [Accessed 1 January 2021].

[49] Brent, "SparkFun USB to Serial UART Boards Hookup Guide," sparkfun, [Online]. Available: https://learn.sparkfun.com/tutorials/sparkfun-usb-to-serial-uart-boards-hookup-guide?_ga=2.222279000.1392414038.1612473373-1317233020.1611841922. [Accessed 1st January 2021].

[50] sparkfun, "SparkFun Beefy 3 - FTDI Basic Breakout," sparkfun, [Online]. Available: https://www.sparkfun.com/products/13746. [Accessed 31 January 2021].

[51] ElectronicsTutorial, "Thermistors," ElectronicsTutorials, [Online]. Available: https://www.electronics-tutorials.ws/io/thermistors.html. [Accessed 28 January 2021].

[52] "Wireless Soil Moisture Sensor," instructables, [Online]. Available: https://www.instructables.com/Wireless-Soil-Moisture-Sensor/. [Accessed October 2020].

[53] Digikey, "PCB Trace Width Calculator," Digikeyq, [Online]. Available: https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-pcb-trace-width. [Accessed 31 January 2021].

[54] D. Lewis, "CloudGarden GitHub Repository," GitHub, February 2021. [Online]. Available: https://github.com/donovinlewis/CloudGarden. [Accessed 19 April 2021].

[55] P. Smith, "How to Install FTDI Drivers," sparkfun, [Online]. Available: https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/windows---in-depth. [Accessed 1 April 2021].

[56] F. Valla, "Programming Arduion via ICSP," StackExchange, 1 November 2017. [Online]. Available: https://arduino.stackexchange.com/questions/38699/programming-arduino-via-icsp. [Accessed 25 March 2021].

[57] Arduino, "Can't get ArduinoISP to work," Arduion, October 2015. [Online]. Available: https://forum.arduino.cc/t/cant-get-arduinoisp-to-work/341978. [Accessed 25 March 2021].

[58] Arduino, "Burning sketches to the Arduino board with an external programmer," Arduino, [Online]. Available: https://www.arduino.cc/en/hacking/programmer. [Accessed 2 April 2021].

[59] Arduino, "Problem with uploading sketch to Pro Mini using FTDI," Arduino, March 2016. [Online]. Available: https://forum.arduino.cc/t/problem-with-uploading-sketch-to-pro-mini-using-ftdi/371574. [Accessed 2 April 2021].

[60] Arduino, "Problems using USBasp with Arduino IDE 1.6.10/Arduino AVR Boards 1.6.12," Arduino, July 2016. [Online]. Available: https://forum.arduino.cc/t/solved-problems-using-usbasp-with-arduino-ide-1-6-10-arduino-avr-boards-1-6-12/399660. [Accessed 30 March 2021].

[61] Z. Amps, "How to Upgrade Firmware in USBasp Programmer," YouTube, September 16 2019. [Online]. Available: https://www.youtube.com/watch?v=1tU7cAFwzig. [Accessed 5 April 2021].

[62] Arduino, "AVR USBasp not working with Arduino IDE (COM error) - Windows 8.1 x64," Arduino, April 2014. [Online]. Available: https://forum.arduino.cc/t/avr-usbasp-not-working-with-arduino-ide-com-error-windows-8-1-x64/229861. [Accessed 5 April 2021].

[63] PJRC, "Teensy 3.2 Development Board," PJRC, [Online]. Available: https://www.pjrc.com/store/teensy32.html. [Accessed 19 April 2021].

[64] Reddit, "Problem burning bootloader to Arduino Nano," Reddit, December 2020. [Online]. Available: https://www.reddit.com/r/arduino/comments/jmn06h/problem_burning_bootloader_to_arduino_nano/. [Accessed 2 April 2021].

[65] SongHe, "Capacitive Soil Moisture Sensor Corrosion Resistant for Arduino Moisture Detection Garden Watering DIY," Amazon, 13 October 2020. [Online]. Available: https://www.amazon.com/gp/product/B07SYBSHGX/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1. [Accessed 13 October 2020].

[66] "18650 Batteries," 18650 Battery Store, [Online]. Available: https://www.18650batterystore.com/collections/18650-batteries. [Accessed November 2020].

[67] "21700 Batteries," Battery Junction.com, [Online]. Available: https://www.batteryjunction.com/21700-batteries.html. [Accessed November 2020].

[68] Battery University, "BU-903: How to Measure State-of-charge," 1 May 2019. [Online]. Available: https://batteryuniversity.com/learn/article/how_to_measure_state_of_charge. [Accessed October 2020].

[69] Linear Technology, "LTC4150 - Coulomb Counter/Battery Gas Gauge," 2003. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/4150fc.pdf. [Accessed November 2020].

[70] element14, "Calculating Battery Life in IoT Applications," element14, [Online]. Available: https://sg.element14.com/calculating-battery-life-in-iot-applications#calculator. [Accessed 17 November 2020].

[71] ethertonics, "X9000984-4GDSMB Datasheet," ethertonics, [Online]. Available: https://datasheets.avx.com/ethertronics/AVX-E_X9000984.pdf. [Accessed 1 January 2021].

9. Appendix

9.1 AHP Decision Making Process

To develop the branch weightings of the objective tree, we grouped related aspects and systematically compared the relative importance of categories another using Analytical Hierarchy Process (AHP) tables. In each AHP table, the categories are listed as the rows and columns of the matrix with two extra columns added for calculation of the weights, after grouping like terms with one another, the weights were determined using pairwise comparison where each criterion was systematically compared to all others in the same group. Using the following scale from the senior design textbook, we compared each row's value to that of the column.

- 1 = equal
- 3 = moderate
- 5 = strong
- 7 = very strong
- 9 = extreme

Values > 1 designate the importance of the row criteria over the column criteria and values < 1 show the importance of column criteria over the row criteria. A value at 1 means the row and column are of equal importance. This meant we could fill out one triangle of the matrix and the values on the opposite side of the matrix's diagonal would be the reciprocals of what is on the other side. To deal with innate inconsistency, the geometric mean of each row was recommended for calculation to reduce the bias in the skewed data. The ratio of each criteria's geometric mean to the sum of means was used to create the relative importance rankings.

Table 21's AHP will be explained as an example with the assumption that all other tables follow this same procedure. Table 21 compares the Consistency, Sampling Rate, and Inexpensive nature of the device, specifically of its sensors. In all AHP tables, the importance of a row is equal in importance to its column assignment. To start with the first row. Consistency is moderately more important than sampling rate because if the values are not consistent, the rate that data is sampled at is irrelevant, and it is moderately more important than the device being cheap because if the data isn't consistent, the device won't be trusted or useful. Sampling rate is less important than consistency as explained prior but moderately more important than being inexpensive as the device should measure and send recent data to be effective. The low cost of the device is moderately less important than the other categories for the previously explained reasons. The geometric mean is calculated for each row by taking the square root of the product of the row's terms. For each row, this geometric mean's ratio relative to the sum of geometric means is calculated to create the weights.

This same process is followed each AHP matrix stored within this section. The rationale for the weights, derived in Table 18, Table 19, Table 20, Table 21, and Table 22, for the project's objective tree are explained in greater detail in Section 3.2.

Table 18: AHP Matrix for the Outdoor Wireless Soil Moisture Sensor

Outdoor Soil Sensor	Data Collection	Accessibility	Data Quality	Longevity	Geometric Mean	Weight
Data Collection	1	1	1	3	1 1/3	0.31
Accessibility	1	1	3	1	1 1/3	0.31
Data Quality	1	1/3	1	1/3	4/7	0.14
Longevity	1/3	1	3	1	1	0.24

Table 19: AHP Matrix for the Soil Moisture Sensor's Data Collection

Data Collection	Soil Moisture Sensing	Battery Life Sensing	Temperature Measurement	Soil Recalibration	Geometric Mean	Weight
Soil Moisture Sensing	1	3	7	3	2 ¾	0.53
Battery Life Sensing	1/3	1	7	1	1 1/3	0.23
Temperature Measurement	1/7	1/7	1	1/3	2/7	0.05
Soil Recalibration	1/3	1	3	1	1	0.19

Table 20: AHP Matrix for the Soil Moisture Sensor's Accessibility

Accessibility	Wireless Transmission	Remote Ease of Use	Multiple Device Support	Geometric Mean	Weight
Wireless Transmission	1	3	5	1 5/7	0.64
Remote Ease of Use	1/3	1	3	1 4/9	0.26
Multiple Device Support	1/5	1/3	1	2/5	0.10

Table 21: AHP Matrix for the Soil Moisture Sensor's Data Quality

Data Quality	Consistency	Sampling Rate	Inexpensive	Geometric Mean	Weight
Consistency	1	3	3	2	0.58
Sampling Rate	1/3	1	3	1	0.28
Inexpensive	1/3	1/3	1	1/2	0.14

Table 22: AHP Matrix for the Soil Moisture Sensor's Longevity

Longevity	Battery Life	Waterproof	Compact	Geometric Mean	Weight
Battery Life	1	1	5	1 5/7	0.45
Waterproof	1	1	5	1 5/7	0.45
Compact	1/5	1/5	1	1/3	0.10

Table 23, Table 24, and Table 25 are AHP tables that were created to rank the priorities for the development of the wireless transmission protocol for our device and used to deliver data to the central hub and receive messages for selecting between various modes. The results of these systematic comparisons were used to make the Objective Tree for the Wireless Transmission Protocol, shown in Figure 12, and compare between potential wireless transmission options.

Table 23: AHP Matrix for the Wireless Transmission Protocol

Wireless Transmission	Cost	Range	Power Consumption	Consistency	Geometric Mean	Weight
Cost	1	1/3	1	1/3	4/7	0.12
Range	3	1	3	3	2 2/7	0.48
Power Consumption	1	1/3	1	1/3	4/7	0.12
Consistency	3	1/3	3	1	1 1/3	0.28

Table 24: AHP Matrix for the Wireless Transmission Protocol's Cost

Cost	Parts cost	Cost over time	Geometric Mean	Weight
Parts cost	1	3	1 5/7	0.75
Cost over time	1/3	1	4/7	0.25

Table 25: AHP Matrix for the Wireless Transmission Protocol's Power Consumption

Power Consumption	Standby Power	Transmission Power	Geometric Mean	Weight
Standby Power	1	3	1 5/7	0.75
Transmission Power	1/3	1	4/7	0.25

Table 26 is an AHP table used to prioritize parameters for the hardware used within the central hub for processing and presentation of data from the deployable devices to the user.

Table 26: AHP Matrix for Central Hub Hardware Selection

Central Hub	Cost	Processing Power	Simplicity	Storage	Geometric Mean	Weight
Cost	1	1/3	1/3	1	4/7	0.125
Processing Power	3	1	1	3	1 3/4	0.375
Simplicity	3	1	1	3	1 3/4	0.375
Storage	1	1/3	1/3	1	4/7	0.125

Table 27, Table 28, Table 29, and Table 30 are all AHP tables created to systematically rank the importance of sections of the soil moisture sensing system that will be integrated into our device to report soil moisture content of surrounding soil. The resulting weights were used to make the Objective Tree for Soil Moisture Sensing, shown in Figure 23, and compare options displayed within the design selection section.

Table 27: AHP Matrix for Soil Moisture Sensing

Soil Moisture Sensing	Waveform Generation	PCB Capacitor Pattern	Waterproofing	Geometric Mean	Weight
Waveform Generation	1	1/3	1/5	2/5	0.11

PCB Capacitor Pattern	3	1	3	2	0.57
Waterproofing	5	1/3	1	1 1/5	0.32

Table 28: AHP Matrix for Soil Moisture Sensing's Waveform Generation

Waveform Generation	Accuracy	Cost	Power Consumption	Reliability	Geometric Mean	Weight
Accuracy	1	3	3	1/5	1 1/6	0.22
Cost	1/3	1	1/3	1/3	4/9	0.08
Power Consumption	1/3	3	1	1/5	2/3	0.13
Reliability	5	3	5	1	3	0.56

Table 29: AHP Matrix for Soil Moisture Sensing's PCB Capacitor Sourcing

PCB Capacitor Sourcing	Availability	Size	Reliability	Geometric Mean	Weight
Availability	1	1/3	1/3	1/2	0.13
Size	3	1	1/5	5/6	0.22
Reliability	3	5	1	2 1/2	0.65

Table 30: AHP Matrix for Waterproofing

Waterproofing	Lifetime	Cost	Abrasion Resistance	Chemical Resistance	Geometric Mean	Weight
Lifetime	1	5	3	1	2	0.41
Cost	1/5	1	5	1/3	3/4	0.16
Abrasion Resistance	1/3	1/5	1	1/3	2/5	0.08
Chemical Resistance	1	3	3	1	1 3/4	0.36

Table 31 is an AHP table used to rank the priorities for the device control subsystem to integrate the power, transmission, and soil moisture subsystems onto a deployable device.

Table 31: AHP Matrix for MCU Selection

MCU Selection	Power Consumption	Device Size	Cost	Parts Accessibility	Geometric Mean	Weight
Power Consumption	1	3	3	3	2.28	0.483
Device Size	1/3	1	2	3	1.19	0.252
Cost	1/3	1/2	1	2	0.76	0.161
Parts Accessibility	1/3	1/3	1/2	1	0.49	0.104

9.2 Waterproofing Standards: IP-54

The following information has been obtained from [13] and describes IP-54 enclosure protective ratings: International (or Ingress) Protection IP ratings systems define an enclosure's protective capacity. The code labels the enclosure's IP followed by two numbers: The first shows the extent to which equipment is protected against particles, and the second digit indicates the extent of protection against water. IP-54 means a high level of protection against particles and a fair amount of protection against water and sports the following characteristics:

- Protection from dirt, dust, oil, and other non-corrosive material
- Protection from contact with enclosed equipment
- Protection from splashing water

In Figure 18, retrieved from [13], the IP rating of 54 correlates to 5 for solids or dust protection with limited ingress of dust and equipment interference for 2 to 8 hours. For water, this correlates to protection against water splashes from all directions and limited permitted ingress. Ingress, in this context, means the act of entering the enclosure surrounding vulnerable electronics. So for testing of the device, we should verify it is IP-54 water resistant by splashing the deactivated device with water from all angles and covering it in sand then verify that neither was able to move into the device's enclosure.

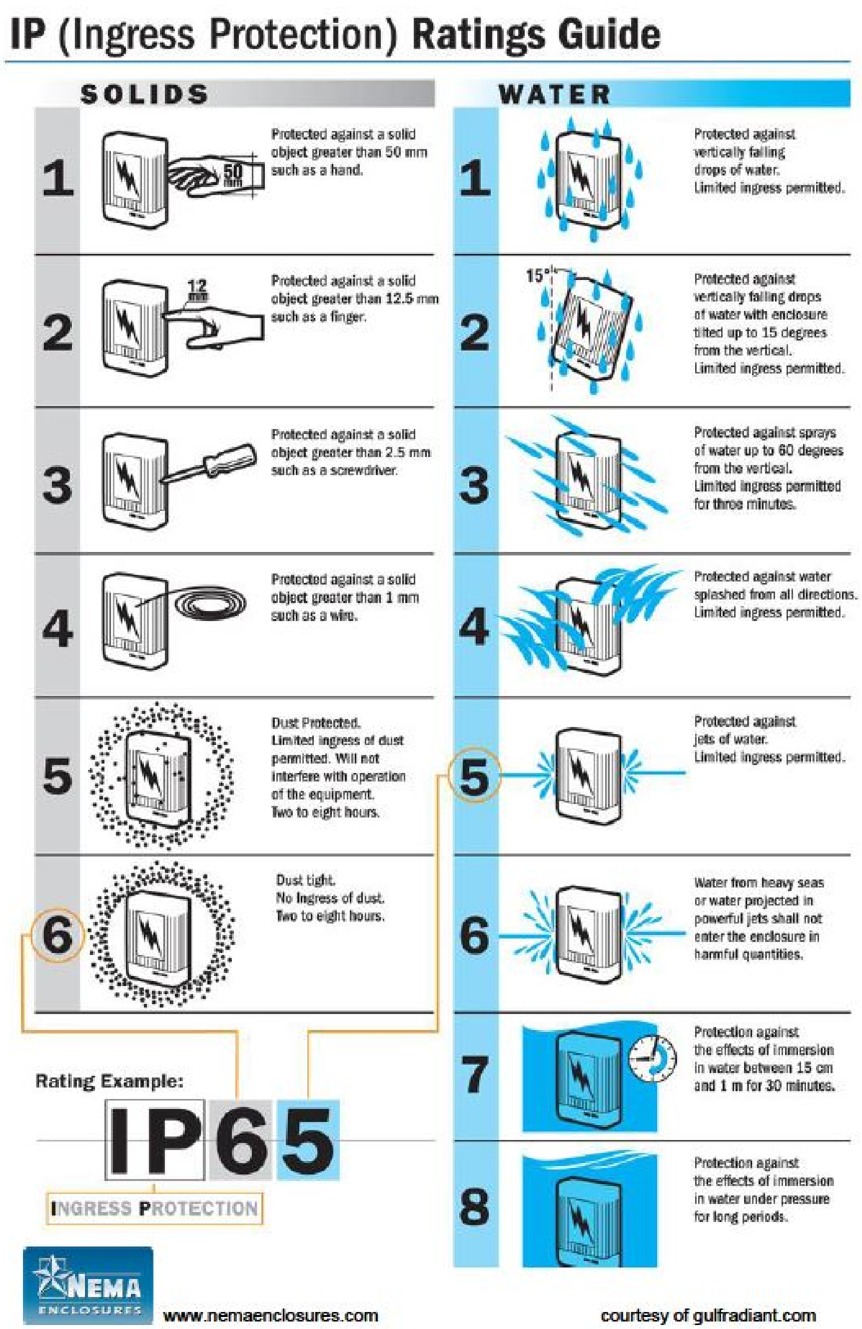


Figure 80: IP Ratings Guide

9.3 Design Alternatives

Following the Functional Decomposition of the outdoor wireless soil moisture sensor system, we split the work as evenly as possible between all five team members centering on the major or most important aspects of our design: Wireless Data Transmission, Hub Processing and Data Presentation, Powering the Device, Capacitive Moisture Sensing, and the Microcontroller and Algorithm needed to run the device. Within the following section, we detail numerous alternatives to solve the issues we were independently tasked with studying including the requirements for each section, the description of options for that section, and the pros and cons associated with each. These alternatives are for systems or blocks derived and described within the Functional Decomposition of our project.

9.3.1 Wireless Data Transmission

Based on the Wireless Data Transmission FD from Figure 7, the components necessary for wireless communication are a wireless transmission protocol to encode and decode messages while minimizing power consumption, a compatible transceiver for sending or receiving messages of the chosen protocol, and an antenna capable of transmitting across the desired range of 500m without line-of-sight. Table 32 is a comparison between various alternatives for each part of the system with comparisons of the pros and cons relative to each other.

Table 32: Wireless Transmission Protocol Design Alternatives

<table>
<tr><th colspan="3">Wireless Transmission Protocol</th></tr>
<tr><td colspan="3">Requirements:
• The device shall transmit data over a range greater than 500m without line-of-sight.
• The device shall have a battery life greater than 2 years.
• The device shall wirelessly transmit measurements to a user accessible hub.
• Multiple devices should be supported for simultaneous usage.</td></tr>
<tr><th>#</th><th colspan="2">Option</th></tr>
<tr><td rowspan="3">1</td><td colspan="2">Long Range Wireless Area Network (LoRaWAN):
Low power and long-range solution explicitly created for the internet of things. It requires significantly lower transmission current than anything else for the range it provides. LoRaWAN operates in unlicensed frequency bands, meaning that data transmission is free.</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Low current draw
• Works anywhere
• Free frequency ranges</td><td>• Low data speeds
• Requires hub nearby</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">Narrow Band Internet of Things (NB IoT):
NB IoT is a version of cellular transmission designed for the internet of things. As a result, it has lower data rates and longer ranges than normal cellular signals. It also operates in licensed frequency bands, so data transmission requires a subscription.</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Longest range option
• High data rates</td><td>• High current draw
• Proprietary frequency ranges</td></tr>
<tr><th colspan="3">Transceiver Selection</th></tr>
<tr><td colspan="3">Requirements:
• The device shall transmit data over a range greater than 500m without line-of-sight
• The device shall have a battery life greater than 2 years
• The device shall wirelessly transmit measurements</td></tr>
<tr><th>#</th><th colspan="2">Option</th></tr>
<tr><td rowspan="3">1</td><td colspan="2">SMD Transceiver Chip
Using an individual chip would be the most cost effective and power efficient but would be far more difficult to implement. On a larger scale with more time, this would be strictly better. On a small scale this would dramatically increase the time from part acquisition to useful implementation.</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Cost effective for larger scale operation
• Smaller form factor</td><td>• Difficult to implement firmware and hardware layout.
• Requires design and implementation of supplementary components</td></tr>
</table>

<table>
<tr><td rowspan="3">2</td><td colspan="2">SMD Transceiver Module
A surface mounted module would not give as much design freedom since many of the pins would have other circuitry between. A module would, however, provide a much simpler implementation for a working prototype.</td></tr>
<tr><td>Pros</td><td>Cons</td></tr>
<tr><td>• Simple implementation and testing
• Tested schematic design and implementation
• Often come with other useful features built-in</td><td>• Higher current draw
• Higher cost on average</td></tr>
<tr><td colspan="3">Antenna Selection</td></tr>
<tr><td colspan="3">Requirements:
• The device shall transmit data over a range greater than 500m without line-of-sight
• The device shall have a battery life greater than 2 years</td></tr>
<tr><td>#</td><td colspan="2">Option</td></tr>
<tr><td rowspan="3">1</td><td colspan="2">Internal Antenna
An internal antenna is would be a helical, wire, or PCB antenna. All of which would be much less reliable and more difficult to implement than any alternative. But they would allow us to waterproof the antenna entirely and possibly allow more configuration options.</td></tr>
<tr><td>Pros</td><td>Cons</td></tr>
<tr><td>• Compact
• Allows more configuration options
• Minimal cost</td><td>• Potentially shorter ranged
• Increased enclosure size</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">External Antenna
An external antenna could take a few shapes, the primary being a rod like structure jutting out of the enclosure. If we waterproof around the connector, this will not negatively affect waterproofing. An external antenna is larger and more powerful across the board than an internal antenna.</td></tr>
<tr><td>Pros</td><td>Cons</td></tr>
<tr><td>• Long range
• User configurable
• Simple implementation</td><td>• More power consumption
• Limits design possibilities</td></tr>
</table>

9.3.2 Hub Processing and Presentation

Regarding the Central Hub FD shown in Figure 6, the radio system is largely dependent on the wireless transmission protocol chosen through Wireless Data Transmission and will be purchased off the shelf but the hub processing device needs to be able to store, process, and present data to the user through a database and web server. Since the central hub is going to be sheltered and intended to be powered through a grid connection and is not limited budget wise other than a max budget of $500, power draw and cost are not as big of a factor. Table 33 contains multiple alternatives for the hub processing device, comparing performance, usability, and capacity.

Table 33: Hub Processing Design Alternatives

<table>
<tr><td>Hub Processing Device</td></tr>
<tr><td>Requirements:
• Reported data shall be remotely accessible and easy to interpret for a home gardener.
• Multiple devices should be supported for simultaneous usage.</td></tr>
</table>

• The device should be able to be recalibrated to relatively measure soil moisture depending on the soil content.		
#	**Option**	
1	Raspberry Pi	
	Pros	**Cons**
	• Small form factor • Strong performance for the small size and power draw • Large community of code to implement webserver • Built-in networking • Ability to perform without internet connection (able to receive data from sensors, unable to access data remotely) • Cost effective	• Not optimized for our specific use case • Small internal storage • Unable to support many remote connections • Less powerful processor
2	**Option**	
	Linux Mini PC	
	Pros	**Cons**
	• Open-source community for easy integrations • Very customizable • High power processor and storage • Ability to perform without an internet connection	• Less I/O Options • Higher power consumption • Higher cost
3	**Option**	
	Offsite Server	
	Pros	**Cons**
	• Expandable capacity • Easy and reliable remote access • Data stored in the 'cloud'	• Require onsite receiver to connect to server • Requires constant internet connection • Ongoing cost • Data stored in the 'cloud'

9.3.3 Power Sourcing and Sensing

The Power and Sourcing subsystem involves the usage of an energy storage device to power the other subsystems on the outdoor soil moisture sensing alongside SOC sensing to alert the user to the battery life left in the battery. Based on the current draw from other components or systems of the outdoor soil moisture sensor, the energy storage system must have enough capacity to allow current draw for the full duration or battery life of 2 years. SOC sensing must be able to be accurately acquired while the device continues to continuously draw current and sample every hour. Table 34 contains alternatives for both the energy storage and SOC estimation systems with these requirements in mind.

Table 34: Power Sourcing and Sensing Design Alternatives

Power Sourcing	
Requirements: • The device shall have a battery life greater than 2 years.	
#	**Option**
1	Single Secondary Lithium-Ion Battery:

1.1	18650 Cell: An 18650 Battery is a cylindrical cell with a diameter of 18mm and a length of 65mm. Commonly used for laptop battery packs, flashlights, electric vehicles, and vape pens [66].	
	Pros	**Cons**
	• Smaller size • Cheap	• Could generate heat and would require something to manage that
1.2	21700 Cell: 21700 batteries are cylindrical cells with a diameter of 21mm and a length of 70mm. Often used for flashlights and other devices that require increased battery capacities [67].	
	Pros	**Cons**
	• Larger capacity • Greater power output capability	• Larger size • Slightly more expensive
2	Multiple Batteries in Parallel:	
	Pros	**Cons**
	• Capacity increases with each battery added in parallel	• More expensive • Much larger size • Greater potential failure points • Batteries could discharge into one another if at different voltages • More complex power sensing circuits
3	Battery with Solar Panel Generation:	
	Pros	**Cons**
	• Environmentally friendly • Increased device longevity through battery energy regeneration	• Requires a battery charger/charger controller to interface with the solar panel and the battery • Larger overall size • More expensive
	Power Sensing	
Requirements:	• The device shall calculate and send the battery's State of Charge (SOC) or lifetime in days and months to the user. • The device shall reliably measure an electrical parameter from which SOC can be derived	
#	**Option**	
1	Coulomb Counting: A method of estimating SoC using the inward and outward flowing current. Connection is made across or through a resistor attached to the main input and output of the battery to the rest of the device [68].	
	Overall Pros	**Overall Cons**
	• More accurate • SOC calibration can be overcome with more advanced coulomb counters • Increased resolution of the SOC range	• Need to calibrate the initial battery SOC • More expensive • Some losses in delivered energy
1.1	Chip Integration	
	Pros	**Cons**
	• Allows for greater customization of the circuit and connectors • Smaller form factor • Cheaper	• Must source and solder additional parts for the circuit • Incorrect placement of filters could lead to larger amounts of noise

	• Implementation schematic is provided within the data sheet	
1.2	Breakout Board Integration	
	Pros	**Cons**
	• Arrives ready to use with no soldering required • Confirmed to work prior to commercial sale	• Must be mounted to the PCB • Larger size • May contain unnecessary connectors
2	Voltage Measurement:	
	Pros	**Cons**
	• Simple to implement • Minimal current draw or ohmic losses	• Temperature alters the voltage and battery capacity relationship • The battery needs to rest at open circuit for at least 4 hours for an accurate voltage reading • Resolution dependent on cell chemistry

9.3.4 Capacitive Moisture Sensing

From the Capacitive Moisture Sensing FD in Figure 8, there are two main components that need to be defined: a method of waveform generation for what is supplied to the RC filter and the PCB capacitor pattern and type used. The PCB capacitor pattern will be defined through testing of various PCB capacitor prototypes, but the type used, either commercial or custom, will drastically effect the overall shape of the final product and the level of integration possible between subsystems. Through research it was found that adding a porous material along the length of the PCB capacitor section increases accuracy as water is closer to the capacitor used to measure it so the pros and cons of that implementation are also explored as a potential design choice [5]. Table 35 compares the alternatives for waveform generation and PCB capacitor utilization while presenting the general idea of adding a porous material to the capacitive moisture sensing circuit.

Table 35: Capacitive Soil Moisture Sensor Design Alternatives

Waveform Generation		
Requirements: • Soil moisture sensing reliability shall not degrade over time. • The device's soil moisture sensing shall be able to be recalibrated for varying environments. .		
#	**Option**	
1	Hardware-defined Timing Circuit: 555 or other timer-based circuit that generates a square wave of a high frequency to maximize real impedance within the low pass filter and reading the resulting voltage value [8, 9, 40].	
	Pros	**Cons**
	• Trusted method for commercial devices • Simple processing • Configurable duty cycle and frequency	• Dependent on ADC resolution • Increased hardware complexity • Cost of components
2	Software-defined Timing Circuit: Using a microcontroller to provide waveforms and measure the output of the RC filter [5, 6, 44].	
	Overall Pros	**Overall Cons**
	• Low hardware complexity • Overall algorithm is available	• Increased software complexity • Dependent on clock speed

	• Easily configurable or modified	
2.1	Direct Microcontroller Discharge Timing: Compare voltage thresholds and measuring the time needed for many full capacitor cycles [6].	
	Pros	**Cons**
	• Independent of the ADC's resolution • Allows quantifiable calculation of capacitance using known resistance	• Increased software complexity • Dependent on clock speed
2.2	Direct Microcontroller Waveform Generation: Reading the resulting voltage value from the RC filter.	
	Pros	**Cons**
	• Simple to compare	• Complicated timing for both ADC and Waveform Generation • Dependent on ADC resolution
PCB Capacitor Utilization		
#	**Option**	
1	Integrate Commercial PCB: Commercial PCB capacitive moisture sensing products exist and integrating them into our systems while adapting for wireless communication would satisfy our basic marketing requirements.	
	Pros	**Cons**
	• Known to be effective • Costs about the same as printing off multiple PCBs	• Flimsy connections • Necessitates substantial wired connection
2	Custom Soil Moisture Sensor PCB: Multiple subsystems could be integrated with the Capacitive PCB circuit on a custom designed PCB that fulfills the requested form factor.	
	Pros	**Cons**
	• Favored by our sponsor • Allows for alteration of the PCB capacitor pattern • Reduces size of the total system	• Components must be soldered onto the board • Requires basic PCB design
Porous Material Addition		
General Idea To mitigate the influence of pores or airgaps between the PCB and the soil, a porous material could be added on the outside of the device over the traces of the PCB capacitor. As the water level surrounding grows or falls, the water level of the porous material would grow or fall accordingly [5].		
	Pros	**Cons**
	• Less susceptible to changes in porosity/air gap of surrounding soil • More accurate to proximate water content	• More expensive • Susceptible to decay • Difficult to source

9.3.5 Controls and Timing

The FD for Controls and Timing is covered in Figure 9 and revolves around the selection of a microcontroller for power draw minimization during the sleep period, majority of the device's lifetime, analog to digital conversion of measured values, and the deactivation of systems that are not currently in use from the battery system to conserve power. Table 36 includes alternatives for types of microcontroller form factor and the type of power switches necessary to enable this functionality on the deployable soil moisture sensor devices.

Table 36: Controls and Timing Design Alternatives

<table>
<tr><th colspan="3">Microcontroller Selection</th></tr>
<tr><td colspan="3">Requirements:
• All measurements on the device shall be taken every hour.
• The device shall have a battery life greater than 2 years.</td></tr>
<tr><th>#</th><th colspan="2">Option</th></tr>
<tr><td rowspan="3">1</td><td colspan="2">Microcontroller Breakout Board</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Arrives ready to use
• Tested for commercial use
• Allows for easier prototype creation and implementation</td><td>• Depending on the intended method of connection, may be difficult to attach to a PCB
• Larger form factor</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">Microcontroller Chip</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Simpler connections for attaching to a custom PCB
• Lower current draw due to selective component usage</td><td>• Harder to Program
• Requires more in-depth application circuit design and implementation
• Very difficult to solder onto a PCB</td></tr>
<tr><th colspan="3">Power Switching/Control</th></tr>
<tr><td colspan="3">General Idea
To reduce the power consumed by circuits when they are not needed or not in use, connection can be broken with the power bus to conserve energy.</td></tr>
<tr><th>#</th><th colspan="2">Option</th></tr>
<tr><td rowspan="3">1</td><td colspan="2">Relays:</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Can handle higher current and voltages
• Does not leak current- either on or off
• Low Resistance</td><td>• Require more power to operate
• Larger form factor
• Contacts can bounce when the switch is operated</td></tr>
<tr><td rowspan="3">2</td><td colspan="2">MOSFETs:</td></tr>
<tr><th>Pros</th><th>Cons</th></tr>
<tr><td>• Faster switching speed
• Smaller form factor
• Allows one signal to switch a larger load
• Smaller power consumption</td><td>• Could leak current within modes in between on and off
• Increased complexity of using three terminal devices</td></tr>
</table>

9.3.6 Mechanical Assembly and Enclosure

To integrate all subsystems together, and reliant on the type of PCB capacitor chosen, the mechanical enclosure and waterproofing of the device are integral to maintaining functionality of the device while exposed to the elements. Table 37 compares alternatives for mechanical enclosures or assembly types as well as methods of waterproofing to ensure the device sustains integrity through long term operation.

Table 37: Mechanical Assembly and Enclosure Design Alternatives

Enclosure		
Requirements: • The device's electronics shall be enveloped within an enclosure. • The device shall be no larger than 10" long, 2" wide, and 5" deep.		
#	**Option**	
1	Top-Heavy Stake: The idea is to have a PCB that holds everything needed for the sensor that is then stuck into the ground as shown in Figure 11. The top half of the PCB holds the electronics and protects them within an enclosure while the bottom half functions as the PCB capacitor to sense changes in soil moisture.	
	Pros	**Cons**
	• Compact • No wired connections • Preferred by the sponsor	• Requires design and implementation of a custom PCB • Requires robust waterproofing for the electronics protection
2	Stake connected to Box: A PCB capacitor is wired to a box that holds the solar panel, battery, and the rest of the circuitry to allow for greater sensing area and space for components.	
	Pros	**Cons**
	• Allows for integration of commercially available systems into one cohesive system • Does not require PCB design or creation • Allows the PCB capacitor to be larger or sense over a larger area	• Consumes valuable topsoil area • Wired connections add more potential points of failure •
Waterproofing		
Requirements: • The device's enclosure shall have a waterproofing rating of IP 54 or greater. • Soil moisture readings shall be taken without direct contact or exposure of electrical contacts to the soil		
#	**Option**	
1	Acrylic Resin Spray:	
	Pros	**Cons**
	• Moisture resistant • Abrasion resistant • Cheap	• Can be removed by solvents or solvent vapors
2	Silicone Resin:	
	Pros	**Cons**
	• Performs well in extremely wet environments • Excellent Humidity resistance • Good chemical resistance • Excellent corrosion resistance • Challenging to remove • Performs well in extreme temperatures • Long life	• Vulnerable to abrasion • Expensive
3	Liquid Electrical Tape:	

	Pros	Cons
	• Very cheap for prototyping • Waterproof seal • Excellent acid resistance • Excellent abrasion resistance • Good extreme weather resistance • Easy to remove with cutting tools	• Requires multiple applications • Difficult to apply correctly
4	Epoxy Resin:	
	Pros	**Cons**
	• Great moisture resistance • Great abrasion resistance • Great chemical resistance • Good humidity resistance • Difficult to remove	• Soldering iron required for rework or repair
5	Potting Compound:	
	Pros	**Cons**
	• Able to withstand various weather conditions. • Excellent moisture resistance • Excellent chemical resistance	• Not able to rework it after applied • Expensive • Requires placement and application within a container

9.4 Detailed Designs

We submitted a testing plan draft and received the feedback that the pictures we had added had too much detail to be placed mid page or as part of a page and to move the images to the Appendix rather than take entire pages mid-report for the images.

Battery Capacity (mAh)	2000		Input	Calculated Values		Idle means "Not doing anything"			
Desired Battery Life (hours)	17520		Defined by Switch	Assumed Max Current		Standby means "Shutting down temporarily"			

Component	**On (mA)**	**Idle (mA)**	**Standby(mA)**	**On Time (seconds)**	**Idle Time (seconds)**	**ON Duty Cycle (%/h)**	**Idle Duty Cycle (%/h)**	**Standby Duty Cycle (%/h)**	**mAh**
ATMEGA 328P	3.58	0.7	0.0045	5	0	0.138888889	0	99.86111111	0.009466
Radio Receive	16	0.0012	0.0001	3.164	0	0.087888889	0	99.91211111	0.014162
Radio Transmit	130	0.0012	0.0001	0.289	0	0.008027778	0	99.99197222	0.010536
Soil Moisture Sensing	5	0.0001	0.0001	5	0	0.138888889	0	99.86111111	0.007044
Temperature Sensing	0.329	0.0001	0.0001	0.1	0	0.002777778	0	99.99722222	0.000109
Battery Monitor IC	0.035	0.009	0.0007	3600	0	100	0	0	0.035

Results

Average mA/h	0.076318
Estimated Battery Life (hours)	26206.26
Desired Life Remaining (years)	-0.99158

Figure 81: Battery Life Estimate Spreadsheet

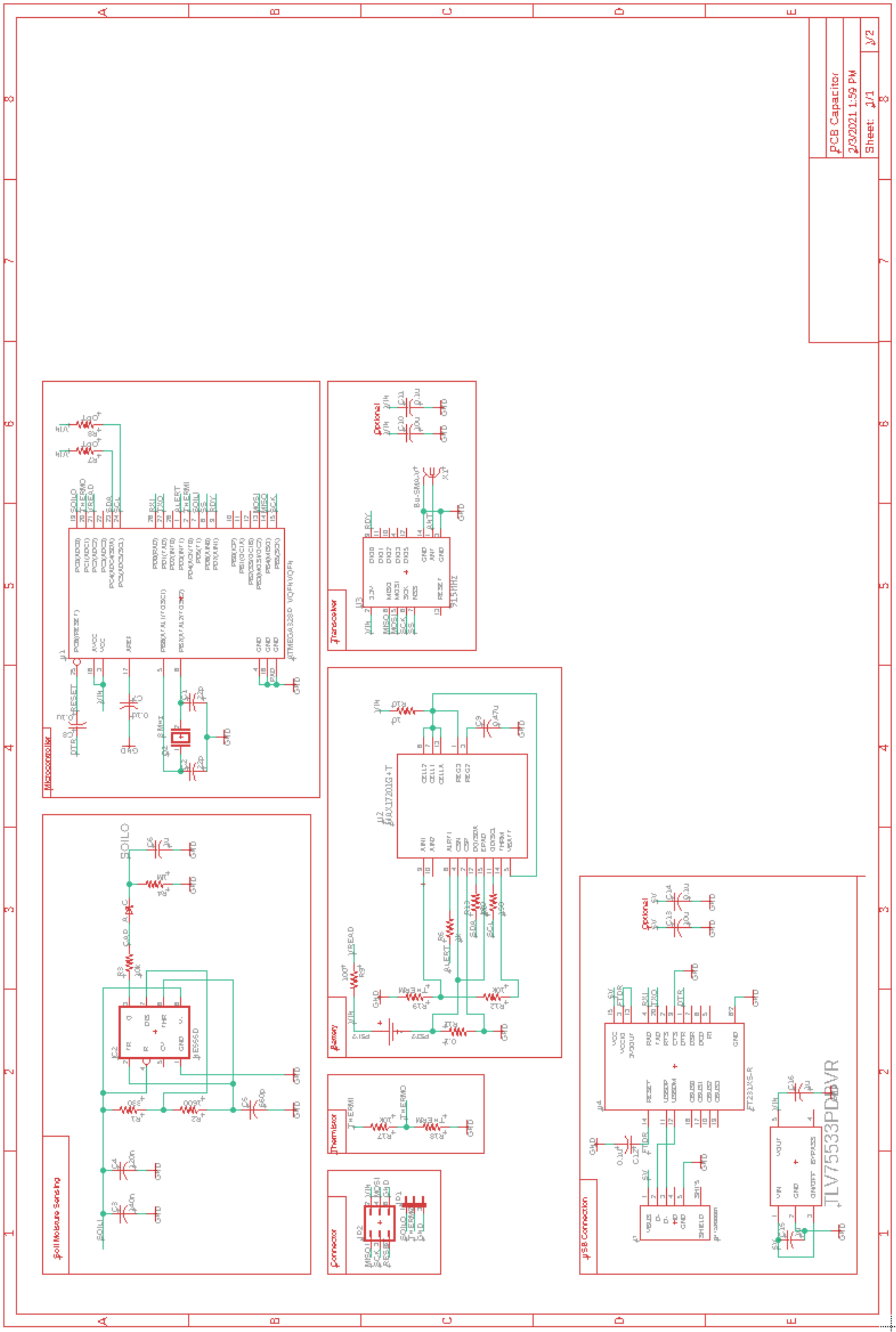


Figure 82: Soil Moisture Sensor EAGLE Prototype Schematic

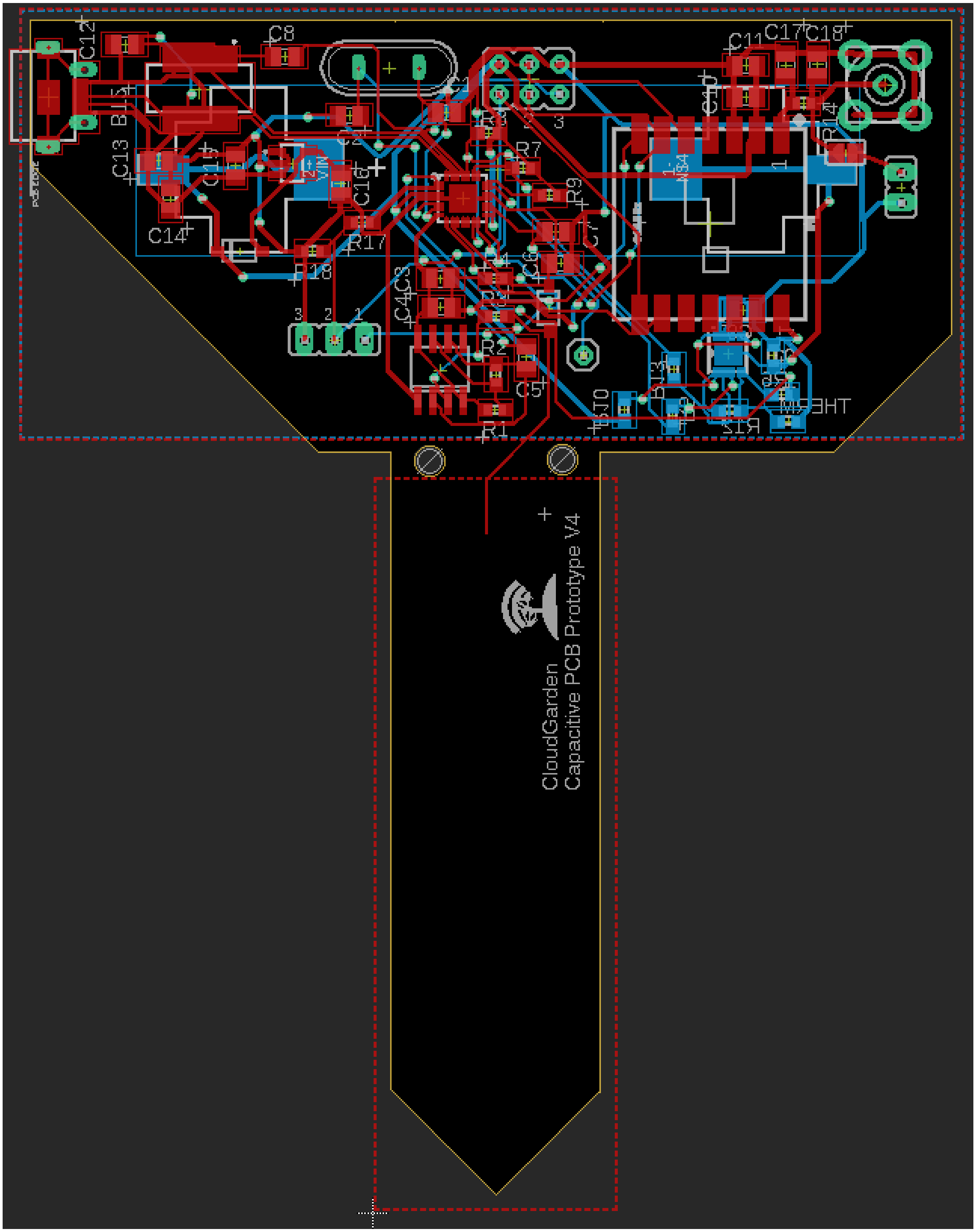


Figure 83: Soil Moisture Sensor Prototype V1 EAGLE Board File

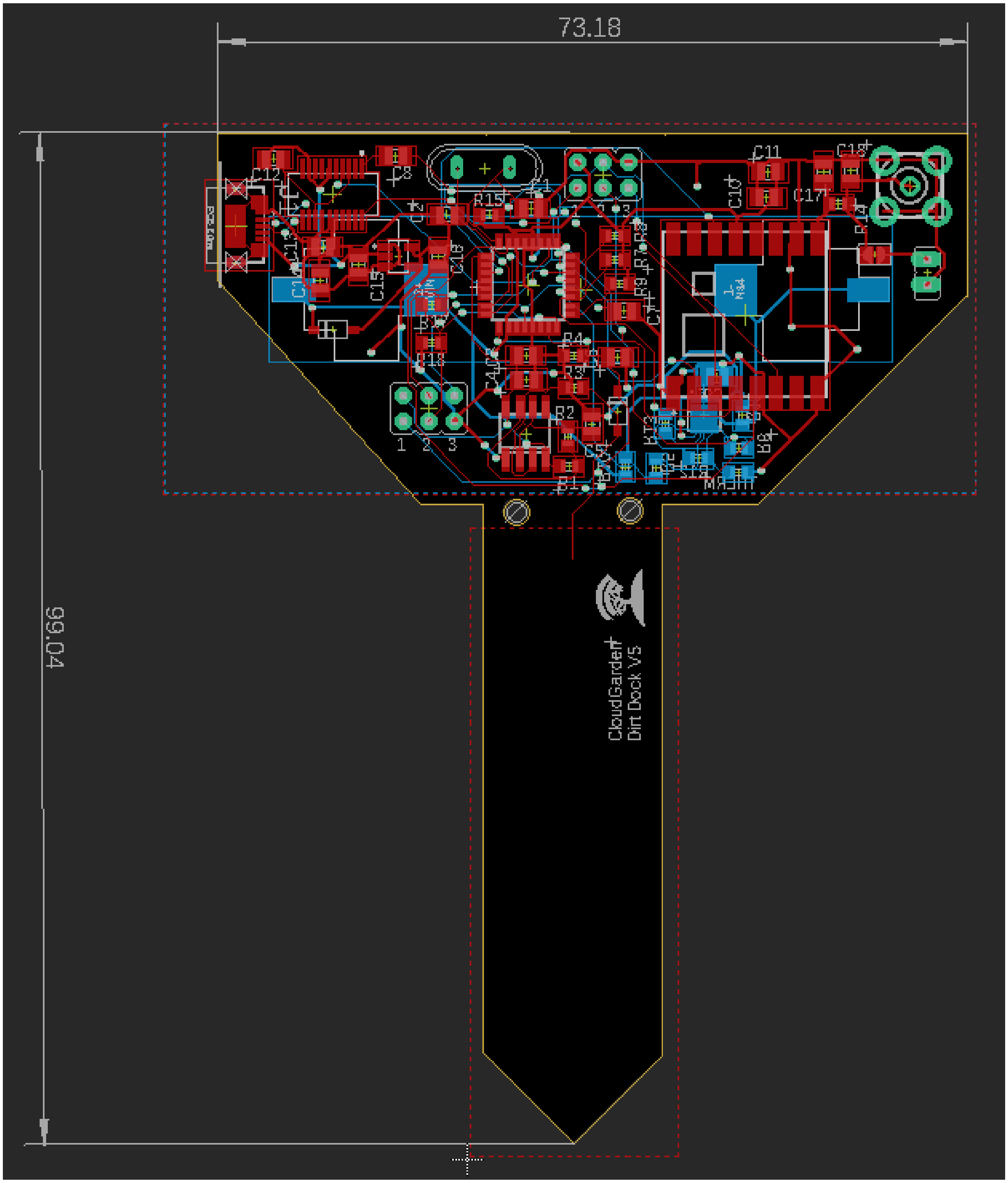


Figure 84: Soil Moisture Sensor Prototype V2 EAGLE Board File

9.4.1 Final Code

```
/* Quick and Dirty Implementation of
*/
```

```
#include "LowPower.h"

//#define LENGTH 9

#define SOILI PD5
#define SOILO PC0

#define THERMI PD4
#define THERMO PC1

int mode = 1;
int cal = 0;
unsigned long durStamp, curSec, duration = 0;

int calAir = 564;
int calWat = 293;
int soilVal, soilPct = 0;

const int A = 1.009249522e-03;
const int B = 2.378405444e-04;
const int C = 2.019202697e-07;

int tempVal = 0;

int battVolts = 0;

void setup() {
  //Serial.begin(9600);

  pinMode(SOILI, OUTPUT);
  pinMode(SOILO, INPUT);

  pinMode(THERMI, OUTPUT);
  pinMode(THERMO, INPUT);
}

void loop() {
  /*
    if (Serial.available() > LENGTH-1) {
      Serial.readBytes(rec_msg, LENGTH);

      //Prints a serial message containing the serial values sent from the master for debugging
      for (int i = 0; i < LENGTH; i++) {
        Serial.print(rec_msg[i], HEX);
        Serial.print(" ");
      }
      Serial.println();
    }
```

```
 */

 digitalWrite(SOILI, HIGH);
 digitalWrite(THERMI, HIGH);
 delay(100);

 soilVal = analogRead(SOILI);
 soilPct = map(soilVal, calAir, calWat, 0, 100);

 tempVal = therm(analogRead(THERMO));

 for (int i = 0; i <= 2; i++) battmVolts += getBandgap();
 battmVolts = battmVolts* / 3;

 if (mode == 0) duration = 3600;

 if (mode == 1) duration = 5;

 if (mode == 2) {
  duration = 1;
  if (cal == 1) calAir = soilVal;
  if (cal == 2) calWat = soilVal;
  if (cal != 1 || cal != 2) mode = 0;
 }

 durStamp = millis();
 duration  *= 12.5; //Converts to millis with sleep time factored in

 while ((millis() - durStamp) < duration) {
  LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
 }
}

void printVal() {
 curSec = millis() / 1000;
 Serial.print(curSec);
 Serial.print(" , ");
 Serial.print(soilVal);
 Serial.print(" , ");
 Serial.print(soilPct);
 Serial.print(" , ");
 Serial.print(tempVal);
 Serial.print(" , ");
 Serial.println(battVolts);
}

int therm(int V0) {
 float logRT, T, TC, TF = 0;
```

```
  logRT = log(10000 * ((1024 / V0) - 1));
  T = (1 / (A + B * logRT + C * logRT * logRT * logRT));
  TC = T - 273.15;
  TF = (TC * 1.8) + 32;
  return TF;
}

int getBG(void) {
  const long InternalReferenceVoltage = 1050L;  // Adust this value to your specific internal BG voltage
x1000
  // REFS1 REFS0          --> 0 1, AVcc internal ref.
  // MUX3 MUX2 MUX1 MUX0  --> 1110 1.1V (VBG)
  ADMUX = (0 << REFS1) | (1 << REFS0) | (0 << ADLAR) | (1 << MUX3) | (1 << MUX2) | (1 << MUX1) | (0
<< MUX0);
  // Start a conversion
  ADCSRA |= _BV( ADSC );
  // Wait for it to complete
  while ( ( (ADCSRA & (1 << ADSC)) != 0 ) );
  // Scale the value
  int results = (((InternalReferenceVoltage * 1024L) / ADC) + 5L) / 10L;
  return results;
}
```