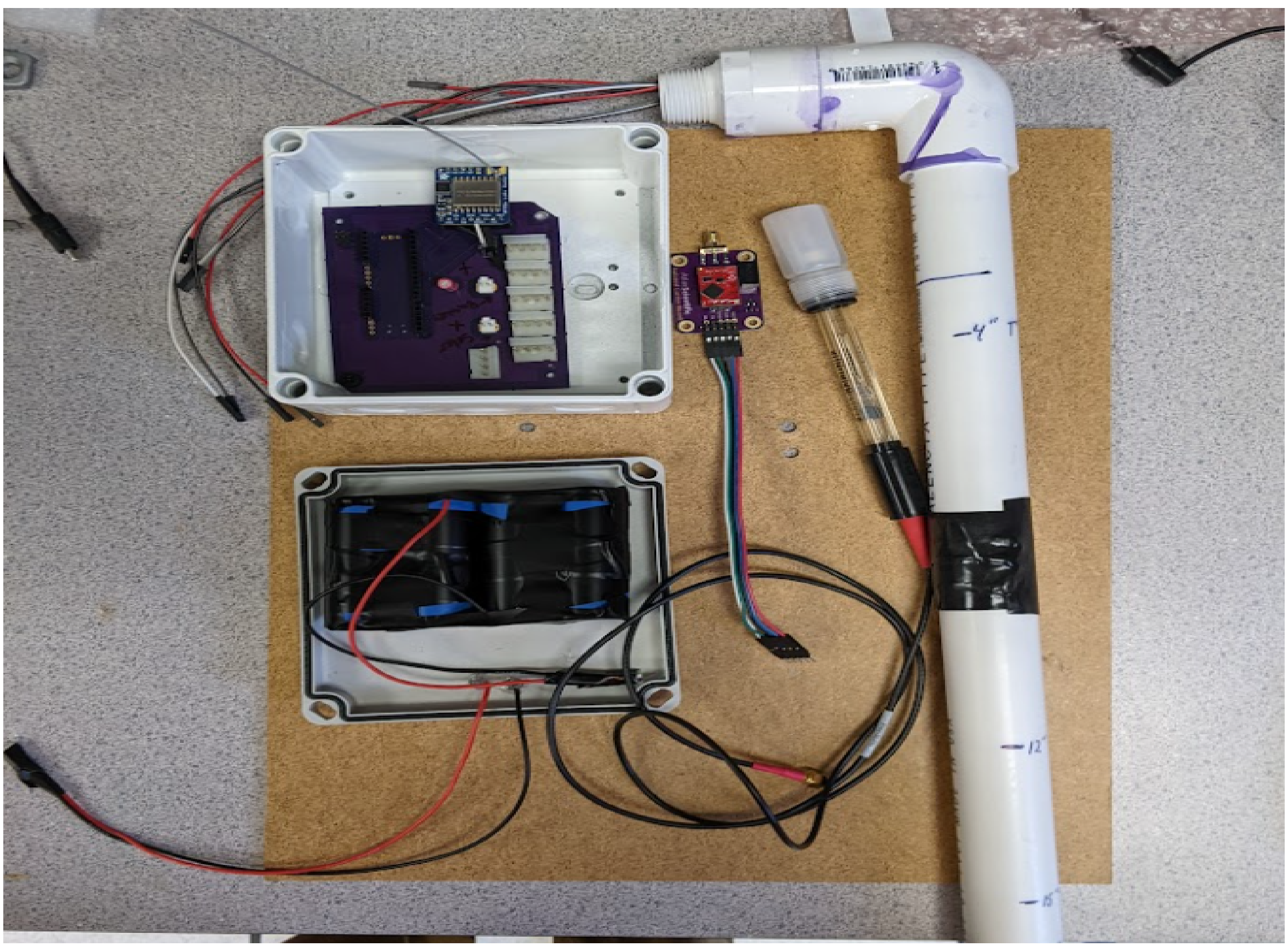
Soil Parameters Measurement and Monitoring

Brennan Ventura, Jeremiah Goddard, Caleb Walker



Oregon State University
11/2/2022

1. Overview

1.1. Executive summary

Soil Parameters Measurement and Monitoring. The goal of this project is to design a probe that is able to be either buried or placed into soil to measure a variety of different parameters and then display those parameters onto a monitor, .txt file or remote server. Some of the parameters to be measured by the probe include but are not yet limited to: soil moisture content, soil temperature, surface temperature, soil pH levels, and amount of water from either irrigation or rainfall. After collection of these various parameters, the user can choose how to display this data whether it be through a basic text file, or onto a table with graphical views onto an online server. Using capacitive measuring the project aims to measure the moisture content of the soil at certain times when there is a large amount of change in the moisture content such as after rainfall has occurred. Using a pH soil sensor that is premade, the pH of the soil can be measured at a set depth. Soil temperature is another important parameter that will affect how well a seed can germinate into the soil, as such, the soil temperature at depth just below the soil will be of importance and measured with a temperature sensor that connects to the microcontroller.

1.2. Team Contacts and Protocols

There are a total of three team members for our project. The information can be found in the table below, along with a table of team protocols.

Table Ⅰ

Group Member	Project Role	Email
Brennan Ventura	Scribe , Communications Systems	email
Caleb Walker	Firmware Programmer, Sensors	email
Jeremiah Goddard	Project partner liaison, power circuit design	goddarje@oregonstate.edu

Table Ⅱ

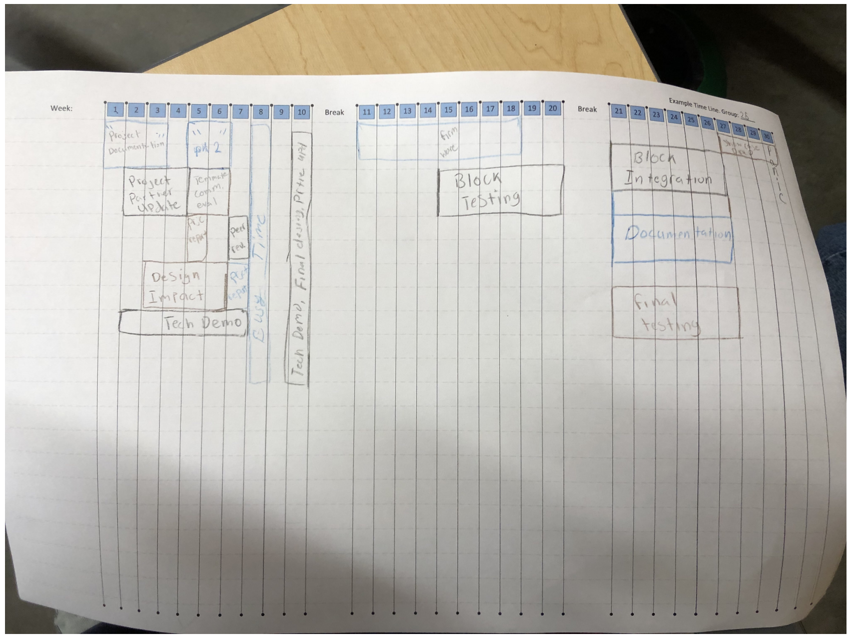
Topic	Protocol
Clear Communication	The team will strive to communicate often with each other and always let others know if they are falling behind, need help, are not able to make it to a meeting, etc.

Project Timelines	A project timeline or google calendar will be kept and updated regularly to keep the time aware of deadlines.
Group Collaboration	The team will work with each other, value everyone's ideas, and work towards making the best possible product in the allotted time.

1.3. Gap Analysis

Measurement of soil parameters can prove to be very helpful when doing long-term studies of soil and what is the optimal amount of water for root growth under certain conditions. Being able to take soil measurement at various sampling rates throughout the changing seasons can yield extremely valuable information. Moreover, on a smaller scale for a user who is simply interested in gardening in their backyard, a device such as this helps them to not over or under water their plants and learn information about the soil in their own garden beds. Long-term soil science studies would reap the most benefits from this end-product as they can see the trends of soil-moisture, as well as other parameters would help them towards achieving their hypothesis, whatever that may be. Some other projects similar to ours can be very helpful in gathering data as well as ideas to base our design on. The first project is done by Campbell Scientific[1], having the sensors rest on the edge of the screw like design is very helpful when burying the probe. Additionally, taking ideas from many other soil sensors that use the capacitance to measure the soil has been very helpful in the design process.

1.4. Timeline/Proposed Timeline



Phase one of the project which will occur during the fall term of 2022, September 2022 - December 2022. During phase one, the main goal will be the design of the project, breaking the project up into specific blocks that are more manageable to work on, creating a parts list, finalizing the design, and beginning to acquire parts for the next phase.

Phase Two. Phase two of the project will coincide with the winter term of the academic year, January 2023 - March 2023. The main goals of phase two will be block building, block implementation, block integration and block testing. Building individual blocks, prototyping blocks, and testing the block individually will take up the majority of the time in phase two.

Phase Three. Phase three is possibly the most important and stressful phase of the project. The main goal of this phase will be block implementation, final testing, and finalizing project documentation. Integrating the blocks with each other will prove to be the most challenging part of the project, however, it is also the most important step into achieving the final product. Field testing the final, end product and gathering measurements and data to be displayed at the final project showcase.

1.5. References and File Links

1.5.1. References

[1] “Soilvue10 - TDR soil moisture and temperature profile sensor,” *SoilVUE10: TDR Soil Moisture and Temperature Profile Sensor*. [Online]. Available: https://www.campbellsci.com/soilvue10. [Accessed: 18-Nov-2022].

1.5.2. File Links

No information for this section.

1.6. Revision Table

11/2/2022	Changed from latex to Google Docs - Brennan Ventura
11/2/2022	Updated Executive Summary - Brennan Ventura
11/2/2022	Moved Protocols table to correct section - Brennan Ventura
11/18/2022	Added Placeholder Timeline - Anthony Briggs
5/14/23	Fixed Formatting on document – Brennan Ventura

2. **Impacts and Risks**

2.1. Design Impact Statement

While there are not many impacts from a soil probe that measures the temperature, pH, and moisture of the soil and relays that information to the user, there are still a lot of important considerations to take into account while designing. There are a few important environmental impacts that could lead to damage over a long period of time. One of the most prevalent environmental impacts is the corrosion of batteries over time and the possible leakage of harmful chemicals into the surrounding environment. Typical lithium-ion batteries often corrode fairly quickly and this can lead to harmful chemicals contaminating the surrounding soil, water and air[1]. To account for this, a waterproof enclosure was designed to ensure that if corrosion of batteries were to occur during the typical system lifespan, no harmful chemical would leak out into the soil. Moreover, lithium-ion batteries were replaced with $LiFePO_4$ batteries which can last longer without any corrosion[2]. Some of the more complicated impacts stem from the public health, safety, welfare, and social and cultural impacts. This system uses solar panels in order to operate for long periods of time without any external wall power. However, solar panels are often very harmful to produce[3], which can lead to a worsening of public health. On a different note, this system can lead to a positive cultural and social impact due to its ability to help owners save and conserve water. This is especially important in drought stricken areas where conserving water can be the difference between going out of business and losing thousands of crops[4]. Furthermore, an affordable and useful soil probe can benefit both farmers and researchers in improving crop yields, an important metric as population increases continue [5]. Which is one of the positive environmental impacts stemming from the implementation and design of this soil probe. In conclusion, the soil parameters measurement and monitoring probe has impacts in many fields such as safety, society, the environment, and economics many of which are overlapping. The low cost of the probe and its ability to measure a variety of variables to aid with farming can have large impacts on the economics and societal problems of an area, especially those in urban areas.

2.2. Risks

Table Ⅲ

Risk ID	Risk Description	Risk Category	Risk Probability	Risk Impact	Performance Indicator	Action Plan
R1	Fertilizer Fire	Safety	Low	High	Fertilizer starts to smoke and catches fire	Attempt to put out fire while clearing out a safe distance from fire, contact 911
R2	Missing Deadline	Organizational	Medium	High	0 for assignment or message from TA	Contact the group member tasked with assignment and ask what happened ask if they need help for future assignments
R3	Battery Fire	Safety	Low	High	Batteries can no longer hold charge and possibly catch fire or explode	If any part of the system has been damaged, remove and test it separately. Replace battery and charging circuit.
R4	Water Damage	Technical	Medium	High	The system no longer works, there are shorts in the system. Water damage is visible inside the probe and enclosure.	Allow at least 48 hours for the system to fully dry before replacing any parts and resuming testing.

R5	Sensor/Wire Shorting	Technical	Medium	High	One of the sensors in the probe is no longer providing accurate measurement or no longer providing measurements at all.	Take out the sensors from the board and test each sensor separately then replace the damaged sensor and ensure proper spacing between wires so no shorts occur.
R6	Overcharged batteries	Technical	Low	High	The batteries can overcharge and damage the voltage regulator leading to a blown circuit and damaged microcontroller	Check the batteries with a multimeter, check the voltage regulator and replace any damaged components.

2.3. References and File Links

2.3.1. References

[1] W. Mrozik, M. A. Rajaeifar, O. Heidrich, and P. Christensen, “Environmental impacts, pollution sources and pathways of spent lithium-ion batteries,” *Energy & Environmental Science*, 13-Oct-2021. [Online]. https://pubs.rsc.org/en/content/articlelanding/2021/ee/d1ee00691f. [Accessed: 03-Nov-2022].
[2] A. Beck, “Lithium Iron Phosphate Vs. Lithium-Ion: Differences and Advantages,” *blog.epectec.com*. https://blog.epectec.com/lithium-iron-phosphate-vs-lithium-ion-differences-and-advantages
[3] S. A. O’Shaughnessy, M. Kim, S. Lee, Y. Kim, H. Kim, and J. Shekailo, “Towards smart farming solutions in the U.S. and South Korea: A comparison of the current status,” *Geography and Sustainability*, vol. 2, no. 4, pp. 312–327, Dec. 2021, doi: 10.1016/j.geosus.2021.12.002.
[4] J. Fu, “It’s the thirstiest crop in the US south-west. Will the drought put alfalfa farmers out of business?,” the Guardian, Sep. 12, 2022. https://www.theguardian.com/environment/2022/sep/12/colorado-drought-water-alfalfa-farmers-conservation
[5] H. Ritchie and M. Roser, “Crop Yields,” *Our World in Data*, 2017. https://ourworldindata.org/crop-yields

2.3.2. File Links

Design Impact Assessment	Design Impact Assessment.docx

2.4. Revision Table

11/2/2022	Added More Risks to risk table - Brennan Ventura
3/12/2023	Imported draft of design impact statement from previous term – Brennan
5/7/23	Brennan - Added citations to risks
5/12/23	Added sources and completely rewrote design impact statement - Brennan

3. Top-Level Architecture

3.1. Block Diagram

Early on in the design process of the system, a block diagram needed to be made not only to simplify the smaller parts of the project but also to streamline the work that needed to be done by each member of the team. During the inception of the block diagram there were four members of the team and it was decided that there be at least two blocks per team member. However, after time one of the members left leaving more work to be done by other members. Upon future iterations of the block diagram there are some blocks that can later be absorbed into others. The current block diagram can be seen below.

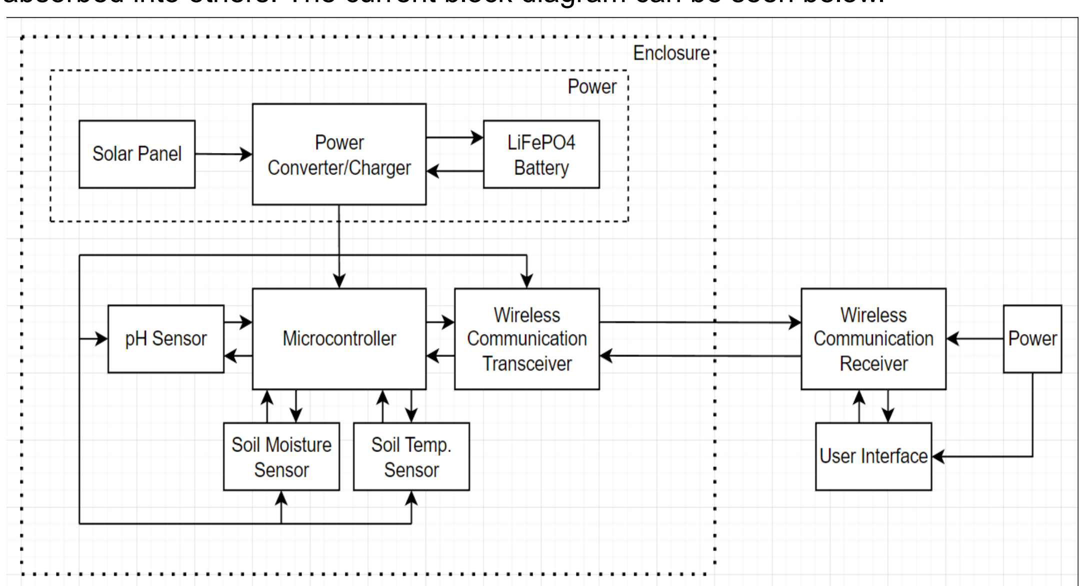


Figure Ⅰ: Block diagram

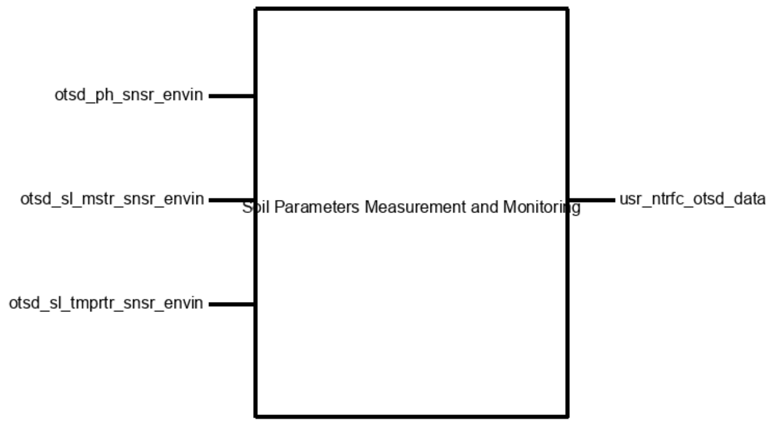


Figure ⅠⅠ: Black Box Diagram

3.2. Block Descriptions

3.2.1. Power Block

The power block is responsible for providing a steady 3.3 Volts to the other blocks located on the probe. This purpose is complemented by the structural support of buses and interconnects provided by the PCB (on which the power block is central). The operation of the block is directly tied to the mission lifespan (an engineering requirement requiring six months of continuous operation). Design decisions were made including: solar panel selection and capacity, battery selection and quantity, voltage regulator throughput, trace width, and others; to ensure the system mission lifetime could be met as well as the system being fully capable of powering every block at once (a situation which should never occur). Power is provided by the sun via the solar panels, which is then directly used to power a LFP battery charge controller. The battery sits between both the charge controller and the switching regulator, receiving current via the charge controller which can be tapped by the regulator if required (with the battery acting as a buffer to catch overflow and underflow from the charge controller).

3.2.2. Soil Temperature Sensor

This block's main purpose is to measure the soil temperature at various depths using multiple sensors that all report to the microcontroller. The sensors will be located inside the enclosure that is inserted into the earth at a location where the parameters of the soil either need to be monitored for research purposes or out of desire to better understand the soil for academic or hobby purposes. The temperature sensors will take in an external output (the outside temperature) and give those values to the microcontroller whenever it is asked for in a way that the microcontroller can understand, i.e. in a digital 12 bit format [1]. The temperature sensor block is powered by the system voltage of 3.3v nominal. Importantly, the sensor is accurate in the temperature ranges that are expected for the applications of farming or gardening in the Pacific NorthWest and meets the total accuracy requirement of being within 90% of the actual value [1].

3.2.3. Soil pH Sensor

This block's main purpose is to measure the pH of the soil at a single depth and report the collected information to the microcontroller. The sensor will report the information to the microcontroller upon request using the I2C protocol. The system will use 3.3v and have a nominal current of 50mA.

3.2.4. Soil Moisture Sensor

This block's main purpose is to measure the soil moisture. It uses a custom shape and size capacitor that is attached to a capacitance to digital converter to measure the relative permittivity of the soil it is in contact with. The capacitor penetrates the enclosure but leaves the IC inside to keep it protected from the elements.

3.2.5. Microcontroller

The microcontroller block is responsible for managing sensor and communications devices on the probe sub-system, as well as deciding when recorded sensor data has strayed far enough from previously reported results to report (primary guarantor of the Dynamic Polling engineering requirement). Other engineering requirements supported, but not exclusively the responsibility of the microcontroller block include Data Display (sensor readings are translated into human understandable units before being reported) and Power Requirement (aided by the frequently utilized builtin DORMANT mode). Design decisions for this block were made with these three requirements as guiding principles. Management is handled through two primary communication paths, one I2C bus containing the sensor devices and an SPI bus for the LoRa Transceiver block. Two digital output lines are also used to selectively control when sensors go to sleep (one for the pH sensor, and another for every other sensor). Internal to the block, a simple DIP switch bank will be used to set software settings (a debug mode and pH reading frequency). Management behavior is handled through firmware installed to the device from a development workstation well before system deployment. This firmware will likely not be final until project completion due to small tweaks in values to accommodate changes as desired. In general, the microcontroller will sleep for a predefined time interval. When the microcontroller wakes up, a quick reading of temperature and moisture values will be made across all sensors of these types after the sensors are woken up by the digital control pin. The pH sensor may only be read occasionally, depending on device settings. The data collected from sensors will then be used to generate a packet of information to be sent to the LoRa Transceiver for transmission.

3.2.6. Wireless Communication Transceiver

The soil moisture sensor project goal is to collect data parameters from the soil from three different probes and then send those measurements from the probe to a location that can be easily viewed by the consumer. In order for this to be achieved, there needs to be a wireless communications network setup. The Wireless Communication Transceiver block is one of the more important parts of the project as it will deal with sending all of the acquired data from the sensors over to a place to be stored and viewed by a user. The block will consist of a LoRa, or long range radio, module which meets several of our system level requirements and fulfills the needs of a long range wireless communication chip. LoRa, long range radio, is able to send smaller amounts of data over long distances at a lower power consumption which is exactly the specification the system needed to meet. In ideal conditions, LoRa advertises transmission ranges of nearly ten kilometers, however the system needs to only be able to transmit over 100 meters. Additionally, because the system needs to be able to last six months without external power, a low power cost wireless communication system was vital. Because of these restrictions, LoRa became the perfect choice for the systems wireless communication transceiver. Along with LoRa which is the physical hardware layer, there comes a protocol called LoRaWan that will be used to transmit our data to the wireless receiver block. LoRaWan uses established gateways and then sends the data from the onboard LoRa module through the gateway to an end device setup or connected to by the user such as a home PC or laptop. LoRaWan gateways are easy to use and have global coverage, additionally, if there is not a LoRaWan gateway in the users area they are not difficult to set up. Overall, the LoRa module and the LoRaWan protocol were a perfect fit for the project requirements and the benefit of having supported Arduino libraries for the LoRa module and Raspberry Pi Pico as our main microcontroller board.

3.2.7. Wireless Communication Receiver

The wireless communication receiver block is vital to storing, processing and displaying the information collected from the soil probe. Once the data collected from the soil probe is collated into a packet and sent over the wireless communication network through the LoRaWAN protocol. The packet is then received by the wireless communication receiver where it both saves a copy of the data onto a local storage device, in this project the local storage device will be a SD card, and exported to a .csv file through the user interface that the end user is able to interact with. The receiver is fundamentally very similar to the transceiver, however there are a few main differences which are important for the final behavior of the sub-system. The receiver sub-system will be held at a remote location, away from the main soil probe system, most likely in the end-users house connected to their PC or some other location due to the need for constant power. The receiver uses a RFM95W LoRa module paired with an Arduino Uno and SD card shield, because the receiver never knows when a packet will be transmitted , it is important for the sub-system to always remain on and be connected to a constant power source. The receiver will be able to communicate with the transceiver over a range of at least 100m and store the data received onto the SD card. The receiver system is very important for the final system, without it there is no way the user could easily view the sensor data without visiting the sensor probe.

3.2.8. Enclosure

The enclosure is a very important part of this system as one of the main concerns is protecting the components from the elements throughout the year. This is especially important for the climate where the system is intended to deploy in Oregon and Washington where a majority of the year there is near constant rainfall. There were many initial options when choosing an enclosure, however, because waterproofing was such an important component of the enclosure, it became much easier to buy and modify an enclosure already rated for waterproofing. The enclosure consists of a small polycarbonate case with a fiberglass coating and a PVC pipe to house the various system sensors embedded in the soil. The polycarbonate case is pre-rated for waterproofing however several coats of flex-seal were added to the enclosure in order to ensure that no water would seep through and damage any of the components. The enclosure also has several mechanical connections as the solar panel rests on the lid of the case and the batteries are connected to the top of the inside of the case.

3.2.9. Main PCB

The main pcb will connect the power supply to the microcontroller, sensors, and wireless communication transceiver. It will connect the microcontroller to the sensors and send the data to the wireless communication transceiver.

3.2.10. User Interface

Windows app user interface software capable of directly connecting to the receiver through a USB serial connection. Through this software, the user will be able to request data stored locally in the receiver. The user will operate the app using clickable buttons and an argument test field. Read only text fields are used to convey information to the user. Whenever the user enters an invalid input for the operation they request, a popup will warn them that their input type is incorrect. This block is important in fulfilling the data display (requiring a method of outputting sensor data as a csv file) and the clear instruction (requiring a consensus of 9/10 people asked) requirements.

3.3. Interface Definitions

There are several high level interfaces in this system. Between the three sensor blocks and the microcontroller there is a digital signal which uses the I2C protocol. The wireless communication transceiver sends a digital signal to the microcontroller through an SPI protocol. The power block provides a dc power interface to the microcontroller. The wireless communication transceiver and receiver both share a wireless communication protocol through the LoRaWAN protocol. The entire list of interfaces and their respective properties are provided in the table below.

Name	Properties
otsd_sl_ph_snsr_envin	• **Other:** pH Max 10.0 • **Other:** pH Min 4 • **Other:** pH Avg 7
otsd_sl_mstr_snsr_envin	• **Other:** Soil Moisture % (Very Dry 1/10) • **Other:** Soil Moisture % (Well Watered 5/10) • **Other:** Soil Moisture % (Very Oversaturated 10/10)
otsd_sl_tmprtr_snsr_envin	• **Temperature (Absolute):** Max 50C • **Temperature (Absolute):** Min -20C • **Temperature (Absolute):** Avg 15C
enclsr_mn_pcb_mech	• **Fasteners:** Mounting Screw 1x • **Pulling Force:** 5 N

	• **Shear Force:** 5 N
enclsr_mn_pwr_spply_mech	• **Fasteners:** Male to Female Molex Connectors • **Other:** Electrical Tape • **Pulling Force:** 5 N
wrlss_cmmnctn_trnscvr_wrlss_cmmnctn_rcvr_rf	• **Messages:** Packet: The packet will consist of {Timestamp: , Temp1: , Temp2: , Temp3: , Moisture1: , Moisture2:, pH: , Flags: } • **Other:** RF Frequency: American Civilian band 915MHz • **Protocol:** LoRaWAN
wrlss_cmmnctn_rcvr_usr_ntrfc_comm	• **Datarate:** 9600 Baud • **Messages:** {Timestamp: , Temp1: , Temp2: , Temp3: , Moisture1: , Moisture2:, pH: , Flags: } • **Protocol:** RS232
mcrcntrllr_mn_pcb_dsig	• **Logic-Level:** 3V3 • **Max Frequency:** 1kHz@100nF • **Other:** Imax: 10mA
mcrcntrllr_sl_ph_snsr_comm	• **Datarate:** 100kbps • **Other:** Logic level: 3V3 • **Protocol:** I2C
mcrcntrllr_wrlss_cmmnctn_trnscvr_comm	• **Datarate:** 100kbps • **Other:** Logic-Level: 3V3. • **Protocol:** SPI

mcrcntrllr_sl_mstr_snsr_comm	• **Datarate:** 100kbps • **Other:** Logic-Level: 3V3 • **Protocol:** I2C
usr_ntrfc_otsd_data	• **Messages:** Timestamp,Temp1,Temp2,Temp3,Moisture1,Moisture2,pH,Flags • **Other:** Operating System: Microsoft Windows • **Protocol:** CSV
mn_pwr_spply_sl_ph_snsr_dcpwr	• **Inominal:** 22mA • **Ipeak:** 30mA • **Vmax:** 5V • **Vmin:** 3V
mn_pwr_spply_wrlss_cmmnctn_trnscvr_dcpwr	• **Inominal:** Less than 1mA • **Ipeak:** 120mA • **Vmax:** 3.7V • **Vmin:** 1.8V
mn_pwr_spply_mcrcntrllr_dcpwr	• **Inominal:** 2mA • **Ipeak:** 90mA • **Vmax:** 5V • **Vmin:** 2V
mn_pwr_spply_sl_mstr_snsr_dcpwr	• **Inominal:** Less than 1mA • **Ipeak:** Less than 1mA • **Vmax:** 3.6V • **Vmin:** 1.6V • **Vnominal:** 3.3V
mn_pwr_spply_sl_tmprtr_snsr_dcpwr	• **Inominal:** Less than 1mA • **Ipeak:** Less than 1mA • **Vmax:** 3.6V • **Vmin:** 1.6V

sl_tmprtr_snsr_mcrcntrllr_data	• **Datarate:** 100kbps • **Other:** Logic-Level: 3V3 • **Protocol:** I2C

3.4. References and File Links

3.4.1. References

There are no references for this section

3.4.2. File Links

Black Box Diagram	Black_Box.png
Block Diagram	Block_Diagram.png

3.5. Revision Table

3/9/2023	Created Section - Brennan Ventura
3/12/2023	Added Interface Definitions - Brennan Ventura
5/14/23	Fixed Formatting - Brennan Ventura
5/14/23	Added in file links - Brennan Ventura

4. Block Validations

4.1. Soil Moisture Sensor

4.1.1. Description

This block's main purpose is to measure the soil moisture. It uses a custom shape and size capacitor that is attached to a capacitance to digital converter to measure the relative permittivity of the soil it is in contact with. The capacitor penetrates the enclosure but leaves the IC inside to keep it protected from the elements.

4.1.2. Design

Below are Fig. 1 and Fig. 2 which show the black box diagram for the block and the typical circuit for the main component of the block, the FDC1004, which is a capacitive to digital converter, respectively.The design of this block includes an input from the environment, an input from the microcontroller, an input from the power board, and input/output communication lines with the microcontroller. The input/output from the microcontroller takes the form of I2C lines that address the sensor and allow it to send the information that it gathers, in digital format, to the microcontroller over shared I2C lines using I2C protocol [1]. The power board input provides the supply voltage of approximately 3.3V. The final input to the board is the input from the environment in the form of capacitance to be measured and converted to a moisture level.

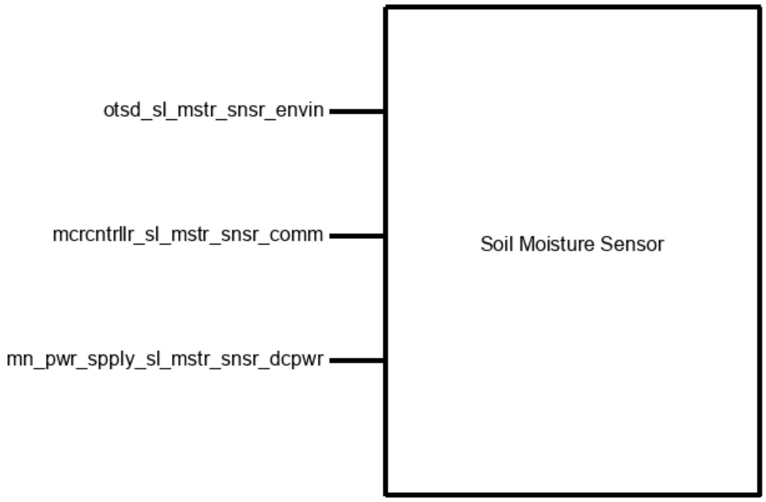


Figure Ⅰ: Soil Moisture Black Box Diagram

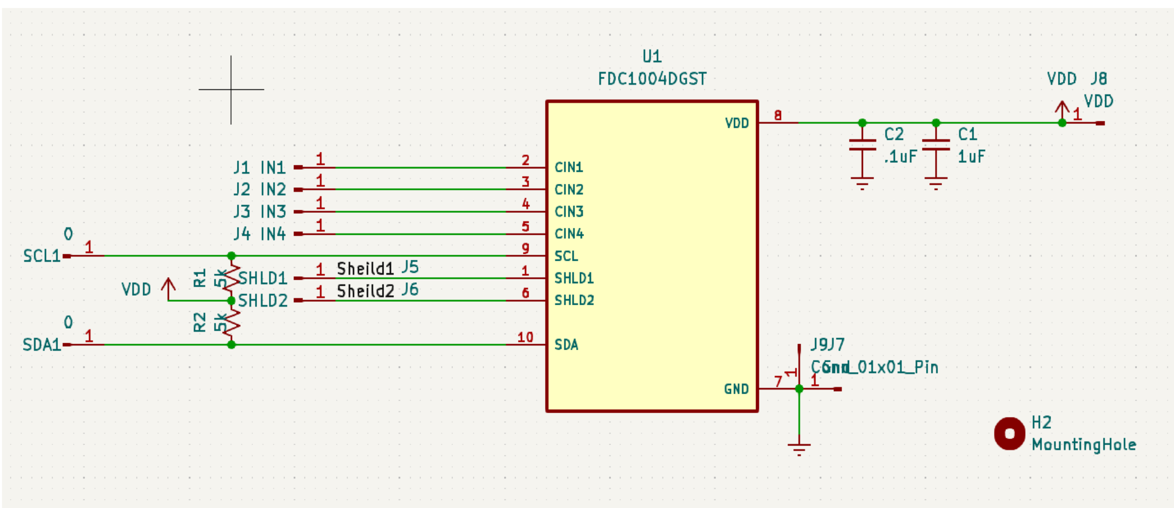


Figure Ⅱ: Soil Moisture Sensor PCB Schematic

4.1.3. General Validation

The block was designed to be the way that it is for many reasons. The chip itself is the capacitive to digital converter and covers any range that we would reasonably need to measure in its application as a soil moisture sensor [1]. The chip was also picked to be accurate within 90% of actual values. The inputs and outputs meet the demands of the system in that an I2C protocol meshes well with a system with a lot of sensors as they can share a bus limiting the size of the overall project and saving on the microcontroller, as one with fewer pins is needed.

The chip also has two other very important aspects that were decided on as well, it is small, 2x2 mm, and uses very little power while being used, μA Average current, as well as when it is put to sleep with an idle current of μA's [1]. These are very important for the project at handAny part that is used must use as little power as possible in order for the unit to function long term. The physical size is important in keeping the probe at a manageable size for installation.

4.1.4. Interface Validation

otsd_sl_mstr_snsr_envin

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Other: Soil Moisture % (Very Oversaturated 10/10)	The interface value is this because it is a measurable level that is easily interpreted across all soil types.	The FDC1004 converts capacitance to a digital value and using a calculator a capacitor can be made that uses the soils changing relative permittivity. The capacitor attached is of custom shape and size to stay within the limits of the chip [1][2][3].
Other: Soil Moisture % (Very Dry 1/10)	The interface value is this because it is a measurable level that is easily interpreted across all soil types.	The FDC1004 converts capacitance to a digital value and using a calculator a capacitor can be made that uses the

		soils changing relative permittivity. The capacitor attached is of custom shape and size to stay within the limits of the chip [1][2][3].
Other: Soil Moisture % (Well Watered 5/10)	The interface value is this because it is a measurable level that is easily interpreted across all soil types.	The FDC1004 converts capacitance to a digital value and using a calculator a capacitor can be made that uses the soils changing relative permittivity. The capacitor attached is of custom shape and size to stay within the limits of the chip [1][2][3].

mcrcntrllr_sl_mstr_snsr_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Datarate: 100kbps	The data rate was chosen to be 100 kbps because it is achievable by the microcontroller that was selected for the project.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

Other: Logic-Level: 3V3	This is the logic level of the pico that was decided on, the FDC1004 can only act as a slave so the logic level is always dictated by the microcontroller	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Protocol: I2C	The I2C protocol was selected because it allows multiple devices in the same bus allowing fewer connections with the microcontroller to be used.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

mn_pwr_spply_sl_mstr_snsr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Inominal: Less than 1mA	This value is put at 1mA because the actual value is rated in the uA which is not reasonably measurable with the equipment that is available.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Ipeak: Less than 1mA	This value is put at 1mA because the actual value is rated in the uA which is not reasonably measurable with the equipment that is available.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

Vmax: 3.6V	This value is at 3.6v because the system power supply is supposed to be 3.3v and 3.6v allows the device to operate around the fluctuations in the power supply.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Vmin: 1.6V	This value is at 1.6v because the system power supply is supposed to be 3.3v and 1.6v allows the device to operate around the fluctuations in the power supply.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Vnominal: 3.3V	This is the value of the system and the device needs to be able to run at nominal system voltage for its lifetime.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

4.1.5. Verification Process

Interface 1

1)The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.

2)The I2C clock line will then be hooked up to an oscilloscope as well as the info line.

3)The Pico will then request a reading from the sensor using I2C protocol.

4)If a reading is read on the pico, the clock is around 100 kbps on the oscilloscope, and the logic line is read at 3.3v then the interface has passed.

Interface 2

1)The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.

2)The pico will then request a reading from the sensor.

3)The value provided to the Pico will then be compared to the moisture level of the room using an independent moisture sensor.
4)The sensor will then be placed in contact with soil and the output compared to the .
5)If the sensor reads within 10% of the known moisture value (verified using another meter) it has met its requirement.
Interface 3
1)The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.
2)The Pico will then request a reading from the sensor using I2C protocol.
3)The supply voltage will then be lowered to 1.6v
4)The Pico will then request a reading from the sensor using I2C protocol
5)If the pico reads an accurate reading it has passed its min voltage rating
6)The supply voltage will then be raised to 3.6v
7)The Pico will then request a reading from the sensor using I2C protocol
8)If the pico reads an accurate reading it has passed its max voltage rating
9)The current will then be measured at 3.3v and 3.6v and verified to be less than 1mA to pass

4.1.6. References and File Links

4.1.6.1. References

[1] Texas Instruments, “FDC1004 4-Channel Capacitance-to-Digital Converter for Capacitive Sensing Solutions” SNNOSCY5B datasheet, Aug. 2014 [Revised April. 2015].
[2]“Coplanar Capacitance,” Electromagnetic Interference Software – EMI Software LLC. https://www.emisoftware.com/calculator/coplanar-capacitance/
[3]W. Patitz, B. Brock, and E. Powell, “SAND1A REPORT Measurement of Dielectric and Magnetic Properties of Soil DwraBunoN 0F ms DOCUMB aw rs VNisumm * 4.” Accessed: May 07, 2023. [Online]. Available: https://inis.iaea.org/collection/NCLCollectionStore/_Public/27/040/27040410.pdf

4.1.6.2. File Links

FDC1004 Datasheet	FDC1004 4-Channel Capacitance-to-Digital Converter for Capacitive Sensing Solutions datasheet (Rev. B)

4.1.7. Revision Table

Date	Revision
4/3/2023	Section Created By Jeremiah Goddard

4.2. Soil pH Sensor

4.2.1. Description

This block's main purpose is to measure the pH of the soil at a single depth and report the collected information to the microcontroller. The sensor will report the information to the microcontroller upon request using the I2C protocol. The system will use 3.3v and have a nominal current of 50mA.

4.2.2. Design

Below is figure 1 which shows the black box diagram for the block. The main component is the pH sensor kit from atlas scientific which includes the pH probe as well as the circuits to interface the probe with electronic components.The design of this block is to take an input from the environment, and relay that to the microcontroller in a digital format to be easily accessed using the I2C protocol. The system supply voltage is used for this component at 3.3v which allows us to limit the complexity of the power supply.

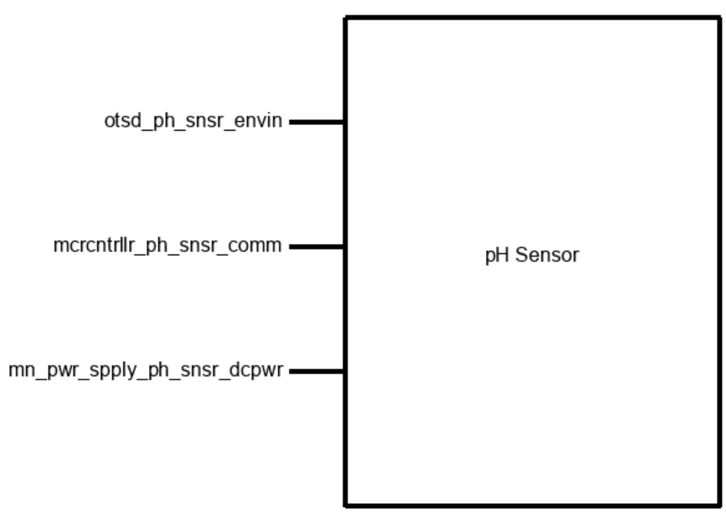


Figure Ⅰ: Soil pH Sensor Black Box Diagram

4.2.3. General Validation

For this block there was a need for robustness as the pH sensor will need to be buried for an extended period of time outside of the enclosure. The pH sensor also needs to be able to interface with the microcontroller easily. The established interface for the project is an I2C bus and the system voltage is at 3.3v. The consumer grade probe from atlas scientific meets these requirements. It can be buried in soil indefinitely, can be communicated with in a variety of methods, including I2C, and has a voltage input range of 3.0 -5.0v.

4.2.4. Interface Validation

otsd_ph_snsr_envin

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Other: pH Min 4	This is the minimum pH that we can reasonably expect to see. Ideally the soil should not ever reach this point but it is good to have a wide range.	This value was determined from the datasheet. [1]
Other: pH Avg 7	This is the midpoint of the scale. The sensor should be centered around 7 because that is what neutral pH is.	This value was determined from the datasheet. [1]
Other: pH Max 10.0	This is the maximum pH that we can reasonably expect to see. Ideally the soil should not ever reach this point but it is good to have a wide range.	This value was determined from the datasheet. [1]

mcrcntrllr_ph_snsr_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Datarate: 100kbps	The I2C protocol was selected because it allows multiple devices in the same bus allowing fewer connections with the microcontroller to be used.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [2].
Other: Logic level: 3V3	The I2C protocol was selected because it allows multiple devices in the same bus allowing fewer connections with the microcontroller to be used.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [2].
Protocol: I2C	The I2C protocol was selected because it allows multiple devices in the same bus allowing fewer connections with the microcontroller to be used.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [2].

mn_pwr_spply_ph_snsr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**

Inominal: 30mA	This is set to 30mA due to the current draw of the individual components inside the block adding up to 30v at its nominal.	This block is known to be able to meet the expectations set because of testing done on the system.
Ipeak: 50mA	This is set to 50mA due to the current draw of the individual components inside the block adding up to 50mA at its peak.	This block is known to be able to meet the expectations through testing of the system.
Vmax: 5V	This value is at 5v because the system power supply is supposed to be 3.3v and 5v allows the device to operate around the fluctuations in the power supply.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [2].
Vmin: 3V	This value is at 3v because the system power supply is supposed to be 3.3v and 3v allows the device to operate around the fluctuations in the power supply.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [2].

4.2.5. Verification Process

1. The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.
2. The I2C clock line will then be hooked up to an oscilloscope as well as the info line.
3. The Pico will then request a pH level from the sensor using I2C protocol.
4. If a pH Level is read on the pico, the clock is around 100 kbps on the oscilloscope, and the logic line is read at 3.3v then the interface has passed.

5. The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico. The probe will be placed into a liquid of a pH of 7.
6. The pico will then request a pH reading from the sensor.
7. The value provided to the Pico will then be compared to the known pH of the solution.
8. The sensor will then be placed in water to rinse and then placed in a solution with a pH of 4.
9. The pico will then request a pH reading from the sensor and the value provided to the Pico will then be compared to the known pH of the solution..
10. The process will be repeated for a pH of 10.
11. If the sensor has matched the known pH level of the solution it has passed
12. The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.
13. The Pico will then request a pH level from the sensor using I2C protocol.
14. The supply voltage will then be lowered to 3v
15. The Pico will then request a pH level from the sensor using I2C protocol
16. If the pico reads an accurate reading it has passed its min voltage rating
17. The supply voltage will then be raised to 5v.
18. The Pico will then request a pH level from the sensor using I2C protocol
19. If the pico reads an accurate reading it has passed its max voltage rating
20. The current will then be measured at 3.3v and 5v and verified to be less than 50mA to pass.

4.2.6. References and File Links

4.2.6.1. References

[1] Atlas Scientific, “Consumer Grade pH Probe” V2.7 datasheet [Revised July 2022]

[2] Atlas Scientific, “EZO-pH” V5.9 datasheet [Revised Oct. 2021]

[3] L. Reich, “FineGardening,” FineGardening, Apr. 25, 2014. https://www.finegardening.com/article/the-four-things-you-need-to-know-about-soil-ph

4.2.6.2. File Links

pH Reader PCB Datasheet	https://files.atlas-scientific.com/pH_EZO_Datasheet.pdf
pH Probe Datasheet	Consumer Grade pH Probe

4.2.7. Revision Table

Date	Revision
3/12/2023	Document Created - Jeremiah Goddard

4.3. Soil Temperature Sensor

4.3.1. Description

This block's main purpose is to measure the soil temperature at various depths using multiple sensors that all report to the microcontroller. The sensors will be located inside the enclosure that is inserted into the earth at a location where the parameters of the soil either need to be monitored for research purposes or out of desire to better understand the soil for academic or hobby purposes. The temperature sensors will take in an external output (the outside temperature) and give those values to the microcontroller whenever it is asked for in a way that the microcontroller can understand, i.e. in a digital 12 bit format [1]. The temperature sensor block is powered by the system voltage of 3.3v nominal. Importantly, the sensor is accurate in the temperature ranges that are expected for the applications of farming or gardening in the Pacific NorthWest and meets the total accuracy requirement of being within 90% of the actual value [1].

4.3.2. Design

The design of this block includes an input from the environment, an input from the microcontroller, an input from the power board, and input/output communication lines with the microcontroller. The input/output from the microcontroller takes the form of I2C lines that address the sensor and allow it to send the information that it gathers, in digital format, to the microcontroller over shared I2C lines using I2C protocol [1]. The power board input provides the supply voltage of approximately 3.3V. The final input to the board is the input from the environment in the form of temperature to be measured and converted. Below are Fig. 1 and Fig. 2 which show the black box diagram for the block and the typical circuit for the main component of the block, the temp1075, which is a temperature sensor, respectively.

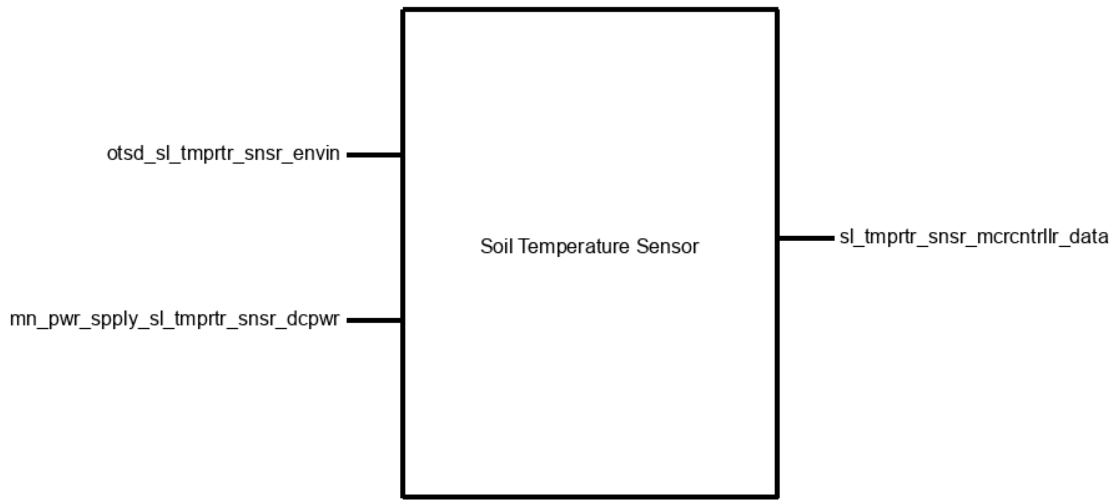


Figure Ⅰ: Soil Temperature Sensor Black Box

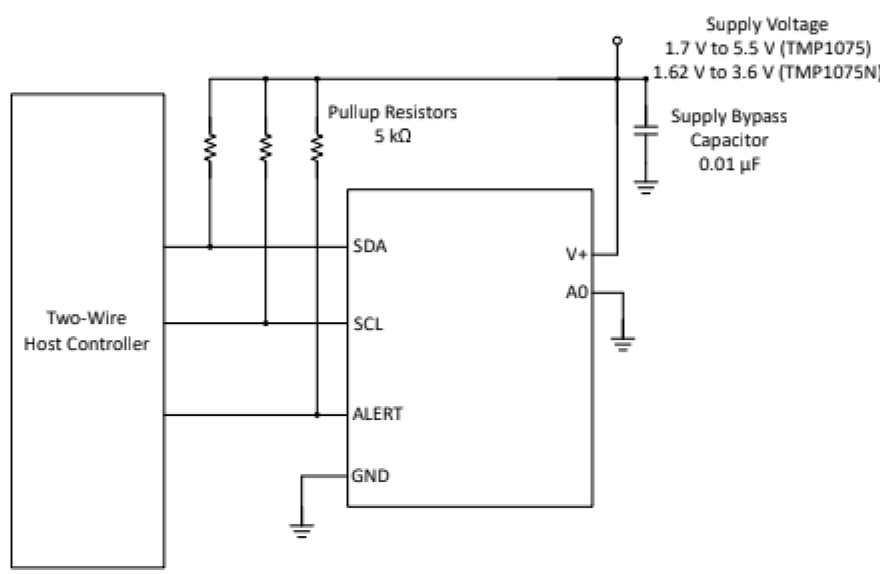


Figure Ⅱ: Sensor Circuit

4.3.3. General Validation

The block was designed to be the way that it is for many reasons. The chip itself, that is the temperature sensor, the TMP1075, was picked because its range of temperature measurement was large, approximately −55°C to +125°C, and covered any range that we would reasonably need to measure in its application as a soil temperature sensor left in a field [1]. The chip was also picked to be accurate within 90% of actual values, which it can easily meet at ±2°C as its maximum error [1]. The inputs and outputs meet the demands of the system in that an I2C protocol meshes well with a system with a lot of sensors as they can share a bus limiting the size of the overall project and saving on the microcontroller, as one with fewer pins is needed. It was also picked for its low cost of $1.45 and

it has been used in a variety of applications, so there is both support for the chip and a history of reliability and proven effectiveness.
The chip also has two other very important aspects that were decided on as well, it is small, 2x2 mm, and uses very little power while being used, 2.7-μA Average current, as well as when it is put to sleep with an idle current of about 0.37-μA [1]. These are very important for the project at hand. The probe must be able to last 6 months without needing any external power outside of what it can gather through solar generation. Any part that is used must use as little power as possible in order for the unit to function long term. The physical size is important in keeping the probe at a manageable size for installation. The temperature sensors need to go into the ground and the housing that protects it cannot be so large that it disturbs the ground around it too much. This chip also only has 6 pins and has a simple schematic and recommended layout [1]. This saves on time for designing the PCB for the sensor and limits the amount of mistakes that can be made in the design process.
An underrated advantage of using this chip is that even though it is newer than other TMP chips available it operates on the same protocols, so there is a large database of code available to help use the chip in this application. It is widely available and is heavily stocked on a multitude of electronic parts websites.
One other possible option for this block if the chip chosen above is not available or is faulty in some way is the TMP75. It is a chip with similar properties as listed for the TMP1075 but is older, a bit larger and more power hungry.

4.3.4. Interface Validation

otsd_sl_tmprtr_snsr_envin

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Temperature (Absolute): Avg 15C	This is within the range of both what is measurable for the validation process and is within the expected maximum temperatures that this	This block is known to be able to meet the expectations set because the datasheet has the component rated for a much higher value [1].

	sensor will need to detect.	
Temperature (Absolute): Max 50C	This is within the range of both what is measurable for the validation process and is within the expected maximum temperatures that this sensor will need to detect.	This block is known to be able to meet the expectations set because the datasheet has the component rated for a much higher value [1].
Temperature (Absolute): Min -20C	This is within the range of both what is measurable for the validation process and is within the expected maximum temperatures that this sensor will need to detect.	This block is known to be able to meet the expectations set because the datasheet has the component rated for a much higher value [1].

Mn_pwr_spply_sl_tmprtr_snsr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Inominal: Less than 1mA	This value is put at 1mA because the actual value is rated in the uA which is not reasonably measurable with the equipment that is available.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Ipeak: Less than 1mA	This value is put at 1mA because the	This block is known to be able to meet the

	actual value is rated in the uA which is not reasonably measurable with the equipment that is available.	expectations set because of the ratings on the datasheet [1].
Vmax: 3.6V	This value is at 3.6v because the system power supply is supposed to be 3.3v and 3.6v allows the device to operate around the fluctuations in the power supply.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Vmin: 1.6V	This value is at 1.6v because the system power supply is supposed to be 3.3v and 1.6v allows the device to operate around the fluctuations in the power supply.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

sl_tmprtr_snsr_mcrcntrllr_data

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Data Rate: 100kbps	The data rate was chosen to be 100 kbps because it is achievable by the microcontroller that was selected for the project.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

Other: Logic-Level: 3V3	This is the logic level of the pico that was decided on, the tmp1075 can only act as a slave so the logic level is always dictated by the microcontroller	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].
Protocol: I2C	The I2C protocol was selected because it allows multiple devices in the same bus allowing fewer connections with the microcontroller to be used.	This block is known to be able to meet the expectations set because of the ratings on the datasheet [1].

4.3.5. Verification Process

Each interface of the block needs to be verified in a way that proves it works properly. Below is a list of each verification process and a short description of what is being verified.

5.1 Microcontroller communication verification

The purpose of this verification plan is to prove that the block can communicate with another device using I2C protocol at 100kbps with a logic level of 3.3v.
1)The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.
2)The I2C clock line will then be hooked up to an oscilloscope as well as the info line.
3)The Pico will then request a temperature from the sensor using I2C protocol.
4)If a temperature is read on the pico, the clock is around 100 kbps on the oscilloscope, and the logic line is read at 3.3v then the interface has passed.

5.2 Outside Temperature Verification

The purpose of this verification plan is to prove that the temperature sensor can accurately read the outside temperature.

1)The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.
2)The pico will then request a temperature reading from the sensor.
3)The value provided to the Pico will then be compared to the temperature of the room using a thermometer.
4)The sensor will then be placed in contact with an object of a known value of -20C, likely using dry ice, and compared to what is given to the Pico.
5)If the sensor reads the temperature within 10% of the known temperature value (verified using another thermometer) it has met its minimum temperature reading.
6)The process will be repeated for 50C, likely using a microwave to heat water verified using another thermometer.
7)If the sensor reads the temperature within 10% of the known temperature value (verified using another thermometer) it has met its maximum temperature reading.

5.3 Power Supply Verification
The purpose of the supply verification is to ensure that the power supply will have a known draw and to verify that the block can handle the voltage and current coming from the power supply.
1)The chip will be supplied with a 3.3v nominal voltage from a DC power supply and the I2C pins will be connected to a raspberry pi Pico.
2)The Pico will then request a temperature from the sensor using I2C protocol.
3)The supply voltage will then be lowered to 1.6v
4)The Pico will then request a temperature from the sensor using I2C protocol
5)If the pico reads an accurate reading it has passed its min voltage rating
6)The supply voltage will then be raised to 3.6v
7)The Pico will then request a temperature from the sensor using I2C protocol
8)If the pico reads an accurate reading it has passed its max voltage rating
9)The current will then be measured at 3.3v and 3.6v and verified to be less than 1mA to pass

4.3.6. References and File Links

4.3.6.1. References

[1] Texas Instruments, “TMP1075 Temperature Sensor With I2C and SMBus Interface in Industry Standard LM75 Form Factor and Pinout” SBOS854E datasheet, Mar. 2018 [Revised Aug 2021]

4.3.6.2. *File Links*

IC Temperature Sensor Circuit	[TMP1075 Temperature Sensor datasheet]()

4.3.7. Revision Table

Date	Revision
2/11/2023	Revised Design Section. Added figure numbers and titles and updated black box diagram. - Jeremiah Goddard
2/11/2023	Added a table heading and updated table in the Interface validation section. - Jeremiah Goddard
2/11/2023	Updated the Verification process for each interface. - Jeremiah Goddard

4.4. Enclosure

4.4.1. Description

The enclosure is a very important part of this system as one of the main concerns is protecting the components from the elements throughout the year. This is especially important for the climate where the system is intended to deploy in Oregon and Washington where a majority of the year there is near constant rainfall. There were many initial options when choosing an enclosure, however, because waterproofing was such an important component of the enclosure, it became much easier to buy and modify an enclosure already rated for waterproofing.

The enclosure consists of a small polycarbonate case with a fiberglass coating and a PVC pipe to house the various system sensors embedded in the soil. The polycarbonate case is pre-rated for waterproofing however several coats of flex-seal were added to the enclosure in order to ensure that no water would seep through and damage any of the components. The enclosure also has several mechanical connections as the solar panel rests on the lid of the case and the batteries are connected to the top of the inside of the case.

4.4.2. Design

There were several very important design aspects that needed to be met when designing the enclosure. One of the first was planning how the various sensors were going to be deployed into the soil and how they would be housed and then connected to the top-lying enclosure. It was decided to house the sensors in a PVC pipe and use a 90° PVC bend and then couple the PVC into the enclosure and secure it using a ¾” connector and fastener. Because of this, it was very important to find an enclosure that came with knockouts so no major drilling or holes had to be made to the case. The final design of the enclosure consists of a top-lying polycarbonate case connected by a 90° turn to a 24’ PVC pipe that is buried in the soil which houses the sensors. A black vox diagram of the enclosure can be seen below describing its various interfaces which include a mechanical connection to the PCB, and a mechanical connection to the power supply.

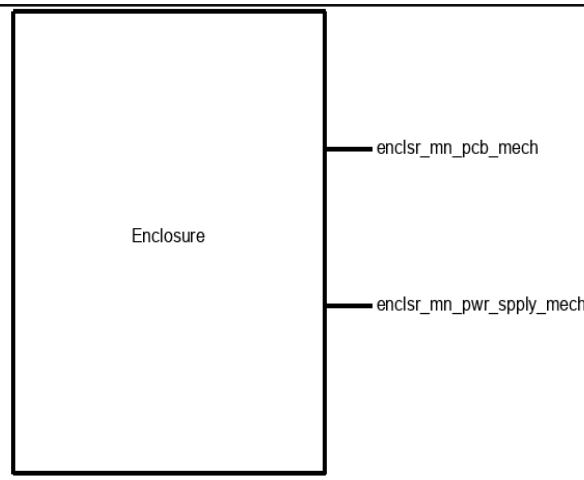


Figure Ⅰ: Enclosure Black Box Diagram

Figure Ⅱ: Assembled System with Enclosure

4.4.3. General Validation

There were many important design aspects that the enclosure needed to meet. Some of the most important were waterproofing, large enough to fit all the components and batteries, and modifications could be easily made to the case to make sure all connections were secure and more could be made if necessary. Because of this choosing the right material for the case and probe housing were extremely important. Choosing a very hard material would have made structure more rigid and possibly more weatherproof, however it would have been much more difficult to make any necessary modifications and adjustments without more serious machinery. After some research, it was decided that polycarbonate with a fiberglass coating would fit all the specifications needed for the enclosure. This material would provide a solid enough base to withstand the rigid structure requirement while also being pliable enough to make smaller adjustments while not damaging the structural integrity. Moreover, the glass fiber coating provides more flexibility and is more ideal for waterproof applications[1]. The cases found had several sizing options, so finding one that would fit the space requirements would not be difficult.

The next part of the enclosure was researching how to house the various sensor boards that need to sit inside the soil while still being protected. Because PVC pipes are often used for piping and plumbing in outdoor environments they would also make a great, waterproof housing for the various sensors in the system. Additionally, PVC is readily available at hardware stores and has a lot of various fittings to secure it to the top-lying case. Sealing the PVC pipe with PVC cement would also provide a watertight seal on both ends. Overall, using PVC and the polycarbonate case were the ideal combinations to fit all of the requirements for the enclosure.

4.4.4. Interface Validation

enclsr_mn_pcb_mech

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Pulling Force : 5 Newtons	There will never be that much force pulling on the PCB inside the enclosure so it does not need to withstand a great deal of pulling force.	Once the PCB is securely connected to the enclosure it can withstand a pulling force exertion of greater than 25 Newtons.
Sheer Force: 5 Newtons	There will never be that much force pulling on the PCB inside the enclosure so it does not need to withstand a great deal of pulling force.	Once the PCB is securely connected to the enclosure it can withstand a sheer force exertion of greater than 25 Newtons.
Fasteners: Mounting Screw	There are several mounting holes in the enclosure and one at each corner of the PCB. Mounting the PCB is very important so it does not move with the enclosure.	There are several mounting posts in the enclosure where the PCB can be screwed into to ensure a secure connection.

enclsr_mn_pwr_spply_mech

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Fasteners: Molex Connectors	The power supply which is connected to the enclosure needs to connect to the PCB through modular connectors.	Molex connectors are easy to disconnect and reconnect making them a great choice for this fastener.
Other: Electrical Tape	The batteries/power supply will be fastened to the physical enclosure using electrical tape and hot glue providing a very secure connection.	The combination of hot glue and electrical tape provide a very secure connection between the power supply and the enclosure that is stable.
Pulling Force : 5 N	There will never be that much force pulling on the PCB inside the enclosure so it does not need to withstand a great deal of pulling force.	The only force that should typically be pulling on the mounting batteries/power supply is gravity which does not provide enough force to remove them from their mounting.

4.4.5. Verification Process

There are not many properties that need to be proven separately from others as they can all be proven with the same steps.

1. Fasten the PCB and power supply to the enclosure with the various fasteners
2. Using a digital scale apply a pulling force to the enclosure and measure the pulling force on the digital scale

3. Check the connections in the enclosure to make sure nothing has come loose
4. Apply a sheer force to the enclosure once again using the digital scale
5. Recheck the connection to make sure all the connections are secure
6. If all the connections have remained and are secure after the force has been applied, the block passes and has been verified

4.4.6. References and File Links

4.4.6.1. References

[1]“Fibreglass Concrete Linings & Coatings: Guide,” Resin Library. https://www.resinlibrary.com/knowledge/guide/grp-concrete-lining-tips/#:~:text=Fibreglass (accessed May 10, 2023).

4.4.6.2. File Links

Polycarbonate Case	https://www.polycase.com/sk-13

4.4.7. Revision Table

4/7/23	Created document - Brennan Ventura
5/7/23	Added in picture of assembled system - Brennan Ventura
5/9/23	Revised design section + validation section - Brennan Ventura

4.5. Wireless Communication Receiver

4.5.1. Description

The wireless communication receiver block is vital to storing, processing and displaying the information collected from the soil probe. Once the data collected from the soil probe is collated into a packet and sent over the wireless communication network through the LoRaWAN protocol. The packet is then received by the wireless communication receiver where it both saves a copy of the data onto a local storage device, in this project the local storage device will be a SD card, and exported to a .csv file through the user interface that the end user is able to interact with. The receiver is fundamentally very similar to the transceiver, however there are a few main differences which are important for the final behavior of the sub-system.

The receiver sub-system will be held at a remote location, away from the main soil probe system, most likely in the end-users house connected to their PC or some other location due to the need for constant power. The receiver uses a RFM95W LoRa module paired with an Arduino Uno and SD card shield, because the receiver never knows when a packet will be transmitted , it is important for the sub-system to always remain on and be connected to a constant power source. The receiver will be able to communicate with the transceiver over a range of at least 100m and store the data received onto the SD card. The receiver system is very important for the final system, without it there is no way the user could easily view the sensor data without visiting the sensor probe.

4.5.2. Design

The wireless communication receiver consists of three main parts: an Arduino Uno microcontroller, a RFM95W LoRa breakout board, and a data logging shield for the Arduino Uno. Both the LoRa breakout board and the data logging shield communicate with the microcontroller through an SPI bus which simplifies the wiring as they can both share the same pin connections for SPI aside from the clock select pin. The LoRa breakout board will be handling the wireless communication between the receiver and the probe through the LoRaWAN protocol operating at the civilian broadband frequency of 915MHz. The data logging shield provides a simple means for the data retrieved by the LoRa module to be saved locally onto an SD card. Moreover, the data logging shield also provides a RTC, real-time clock, which allows a time-stamp to be appended onto each received packet so the user knows what time the measurements were taken throughout the day. However, because there is no way for the receiver to know when a packet is being transmitted it is very important for the receiver sub-system to have constant power, ideally connected to the end users computer so they can make use of the user interface provided with the system .Overall, the design of the wireless communication receiver can be seen as its own sub-system as it has its own microcontroller unit, and will be held at a separate location away from the main probe system.

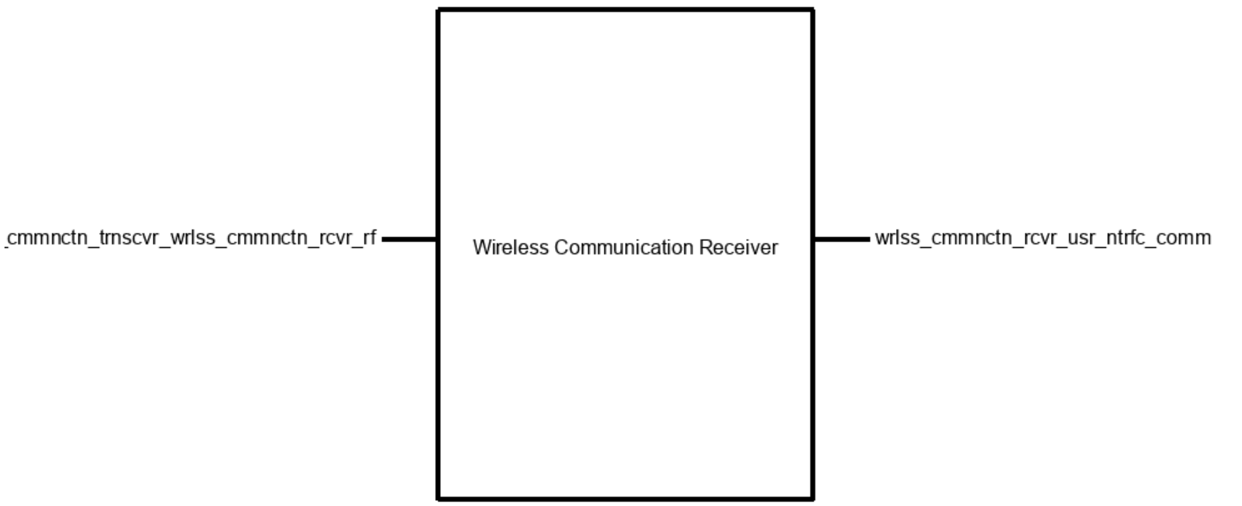


Figure Ⅰ: Wireless Receiver Block Diagram

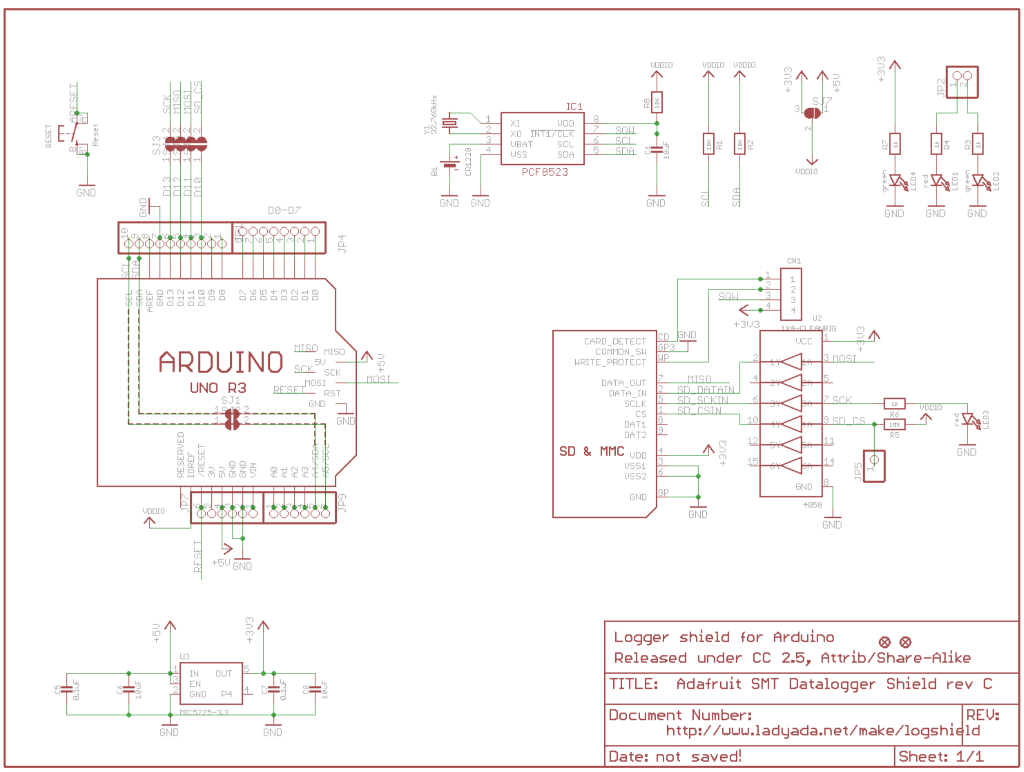


Figure Ⅱ: Schematic of Arduino Data Logging Shield

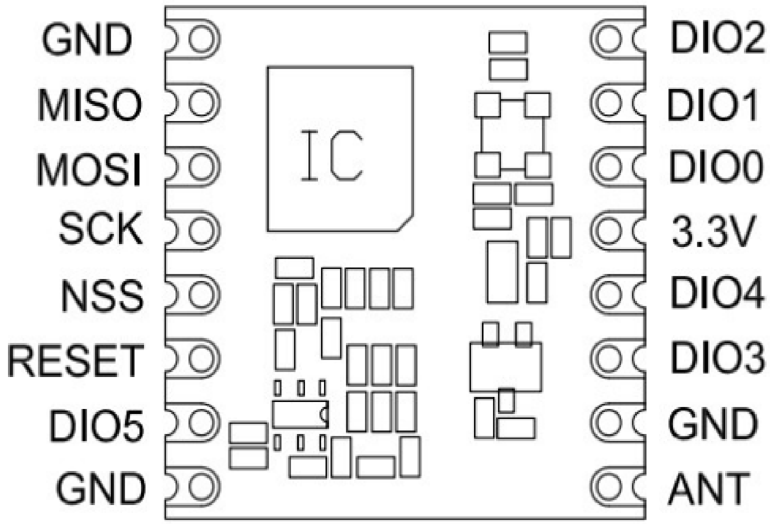


Figure Ⅲ: LoRa Breakout Board Pinout

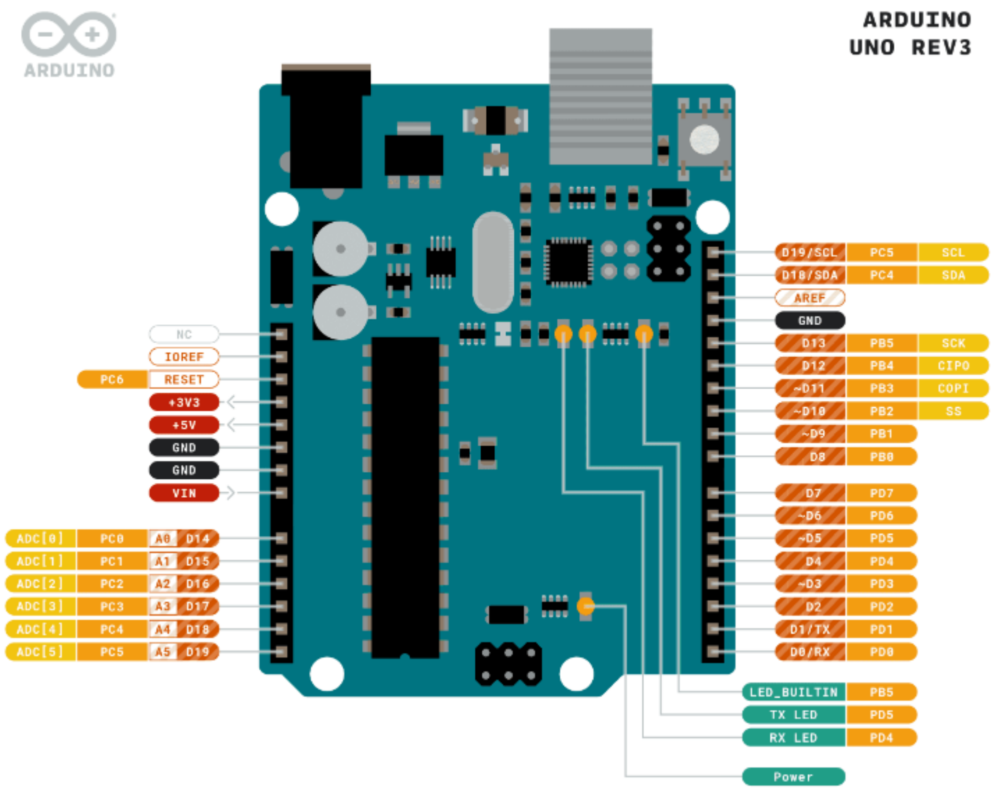


Figure Ⅳ: Arduino Uno Pinout

4.5.3. General Validation

There are many specifications that the wireless communication receiver design needs to meet. It was very important for the receiver to be able to store the data retrieved from the probe locally on some form of physical media. The simplest way to accomplish this was through a data logging shield designed for the Arduino Uno that could interface with the Uno and store data onto a SD card that the user could then access at any time. A data logging shield that supports an SD card made the local data storage problem very simple while also solving some other minor problems. For one, the data logging shield comes with a real time clock which is not present anywhere else in the system[1]. This means that each packet of data transmission can be accompanied by a timestamp that shows the user the time that each of the measurements were taken at. Moreover, a data logging shield that supports an SD card is very helpful in firmware where libraries for SD cards are readily available in the arduino IDE[1]. Additionally, both the SD card and LoRa module communicate through SPI which means they can be run off the same bus. Another very important design aspect the receiver had to meet was low power consumption, and long range wireless transmission. The LoRa breakout board helps the receiver to meet both of these requirements.

4.5.4. Interface Validation

wrlss_cmmnctn_trnscvr_wrlss_cmmnctn_rcvr_rf

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Protocol: LoRaWAN	The LoRa module makes its best long range, low-power performance when paired along with the LoRaWAN protocol for wireless communication.	The wireless communication transceiver can make very good use of the LoRaWAN protocol as its preferred method of wireless communication.
Messages: Packet:	The sensor blocks connected to the microcontroller will be arranged into packets of data containing pH	The LoRa module excels at being able to send smaller packets of information over the

	sensor data, temperature sensor data, and soil moisture sensor data to be sent from the transceiver over the wireless channel to the end device.	LoRaWAN protocol to the end device. These packets will be relatively small as they will only contain number values and not very many of them.
Other: rF. LoRaWAN operates over the 915MHz channel or the civilian band in the US	LoRaWAN operates over the 915MHz channel or the civilian band in the US	The US LoRa module is designed to work in the civilian band at a frequency of 915MHz.

wrlss_cmmnctn_rcvr_usr_ntrfc_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Data Rate: 9600 Baud	Typical arduino sketches run at a baud rate of 9600. This data rate is fast enough to support all of the necessary interactions with the code.	The code has been tested before and is even able to run at several different baud rates but 9600 is the most suited for this application.
Messages: Packet	The sensor blocks connected to the microcontroller will be arranged into packets of data containing pH sensor data, temperature sensor data, and soil moisture sensor data to be sent from the transceiver over the wireless channel to the end device.	The LoRa module excels at being able to send smaller packets of information over the LoRaWAN protocol to the end device. These packets will be relatively small as they will only contain number values and not very many of them.

Protocol: RS232	Using a USB serial connection requires the RS232 protocol for the user interface.	The user interface code takes use of the RS232 protocol to communicate over the serial port and communicate with the receiver.

4.5.5. Verification Process

The wireless communication receiver verification plan will consist of proving the interface between the transceiver and receiver, and the interface between the receiver and user interface.

1. Upload the wireless receiver code onto the Arduino Uno set at a frequency of 915 MHz.
2. Send a packet of data from the wireless communication transceiver to the receiver.
3. Confirm that the packet was received by opening the serial monitor in the Arduino IDE thus proving the packet was sent over the LoRaWAN protocol and the packet structure meets the specifications.
4. Show that the code is running at 9600 baud and the serial monitor is also open at 9600 baud.
5. Using the user interface confirms that the receiver is correctly connected by sending a handshake over the serial monitor using “HRH” and receiving an “ACK” back.
6. Send “RD” into the serial monitor to print out the data collected by the wireless receiver to prove the wireless receiver is correctly interfacing with the user interface.

After these steps have been taken all the interface properties have been proven between the wireless receiver and the wireless transceiver and the wireless receiver and user interface.

4.5.6. References and File Links

4.5.6.1. References

[1] A. Industries, “Adafruit Assembled Data Logging Shield for Arduino,” adafruit industries blog RSS, 2018. [Online]. Available: https://www.adafruit.com/product/1141#technical-details. [Accessed: 25-Apr-2023].

4.5.6.2. File Links

RFM95W LoRa Radio Transceiver Breakout	https://www.hoperf.com/data/upload/portal/20190801/RFM96W-V2.0.pdf
Product Link for Data Logging Shield	https://www.adafruit.com/product/1141#technical-details
Arduino Uno Datasheet	https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

4.5.7. Revision Table

4/3/23	Created Template + Filled out Design - Brennan Ventura
5/6/23	Added new diagrams for pinouts on important parts of system - Brennan Ventura

4.6. Wireless Communication Transceiver

4.6.1. Description

The soil moisture sensor project goal is to collect data parameters from the soil from three different probes and then send those measurements from the probe to a location that can be easily viewed by the consumer. In order for this to be achieved, there needs to be a wireless communications network setup. The Wireless Communication Transceiver block is one of the more important parts of the project as it will deal with sending all of the acquired data from the sensors over to a place to be stored and viewed by a user. The block will consist of a LoRa, or long range radio, module which meets several of our system level requirements and fulfills the needs of a long range wireless communication chip. LoRa, long range radio, is able to send smaller amounts of data over long distances at a lower power consumption which is exactly the specification the system needed to meet. In ideal conditions, LoRa advertises transmission ranges of nearly ten kilometers, however the system needs to only be able to transmit over 100 meters. Additionally, because the system needs to be able to last six months without external power, a low power cost wireless communication system was vital. Because of these restrictions, LoRa became the perfect choice for the systems wireless communication transceiver.

Along with LoRa which is the physical hardware layer, there comes a protocol called LoRaWan that will be used to transmit our data to the wireless receiver block. LoRaWan uses established gateways and then sends the data from the onboard LoRa module through the gateway to an end device setup or connected to by the user such as a home PC or laptop. LoRaWan gateways are easy to use and have global coverage, additionally, if there is not a LoRaWan gateway in the users area they are not difficult to set up. Overall, the LoRa module and the LoRaWan protocol were a perfect fit for the project requirements and the benefit of having supported Arduino libraries for the LoRa module and Raspberry Pi Pico as our main microcontroller board.

4.6.2. Design

The wireless communication transceiver will communicate with the system microcontroller, a Raspberry Pi Pico, through SPI or serial peripheral interface. The LoRa module needs a 3.3 V logic level through its 3.3 V pin and a ground connection to its GND pin in order to receive power. The pinout for the LoRa board can be seen below in figure II. The LoRa module uses radio frequency in order to achieve its wireless communication , here in the US the civilian frequency band, which is the range of frequencies that can be used by the general population without any permits, is 915 MHz[1] which is the frequency at which the LoRa module will send its data over to the LoRa gateway established in the Corvallis area. The wireless communication transceiver also shares an interface from the other wireless communication transceiver which can send commands to the microcontroller to control the sensors. For instance, a command can be sent from the users end device through the LoRaWan gateway to the LoRa module connected to the soil moisture probe, telling the microcontroller to collect the pH sensor data and send it in the next packet along with the scheduled information. The data collected from the various sensors will then be organized into packets and sent at a specified timer interval selected by the user and then hardcoded into the firmware. Most of this block design will consist of the hardware components between the LoRa module used for sending the data collected from the sensors connected to the Raspberry Pi Pico and through the channel through the LoRaWAN gateway to the end device that the user has connected. However, there is a software portion of this design as well as the data collected from the various sensors needs to be collected and organized into packets. Typically, the sensors being read most often will be the temperature and soil moisture so they will take priority in the packet. This is because pH is not that important of a parameter for typical plant growth and will typically stay fairly constant over a plant's lifetime. Because of this, a typical packet will consist of the most recent data collected from the temperature and moisture sensor, with pH sensor data only being sent once every month, or if the user specifies for it to be taken and sent along with the next packet transmission.

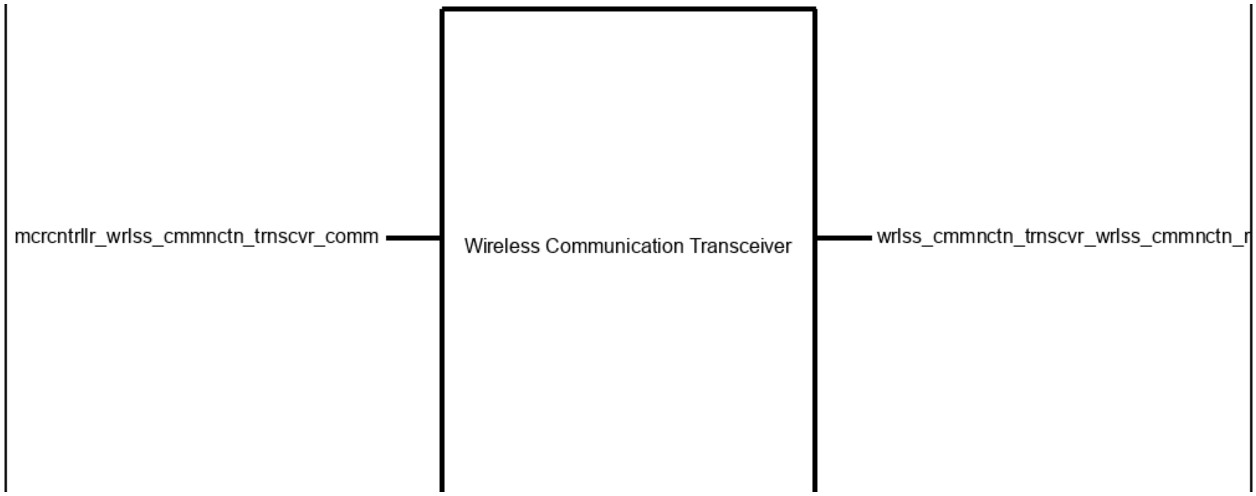


Figure I : Black Box Diagram

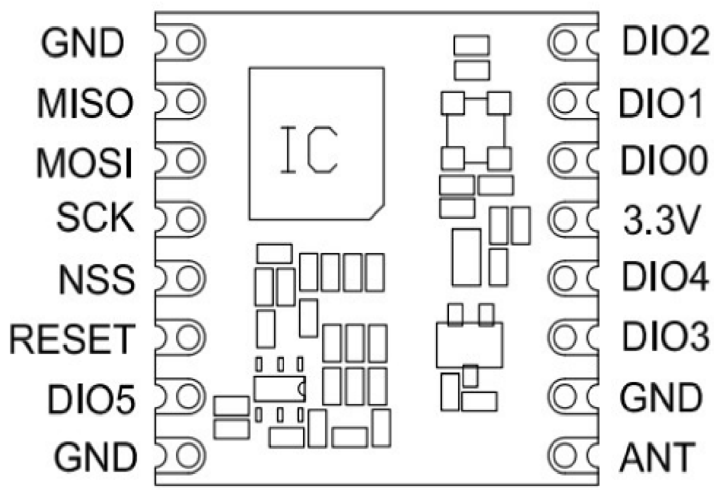


Figure II : LoRa Module Pinout

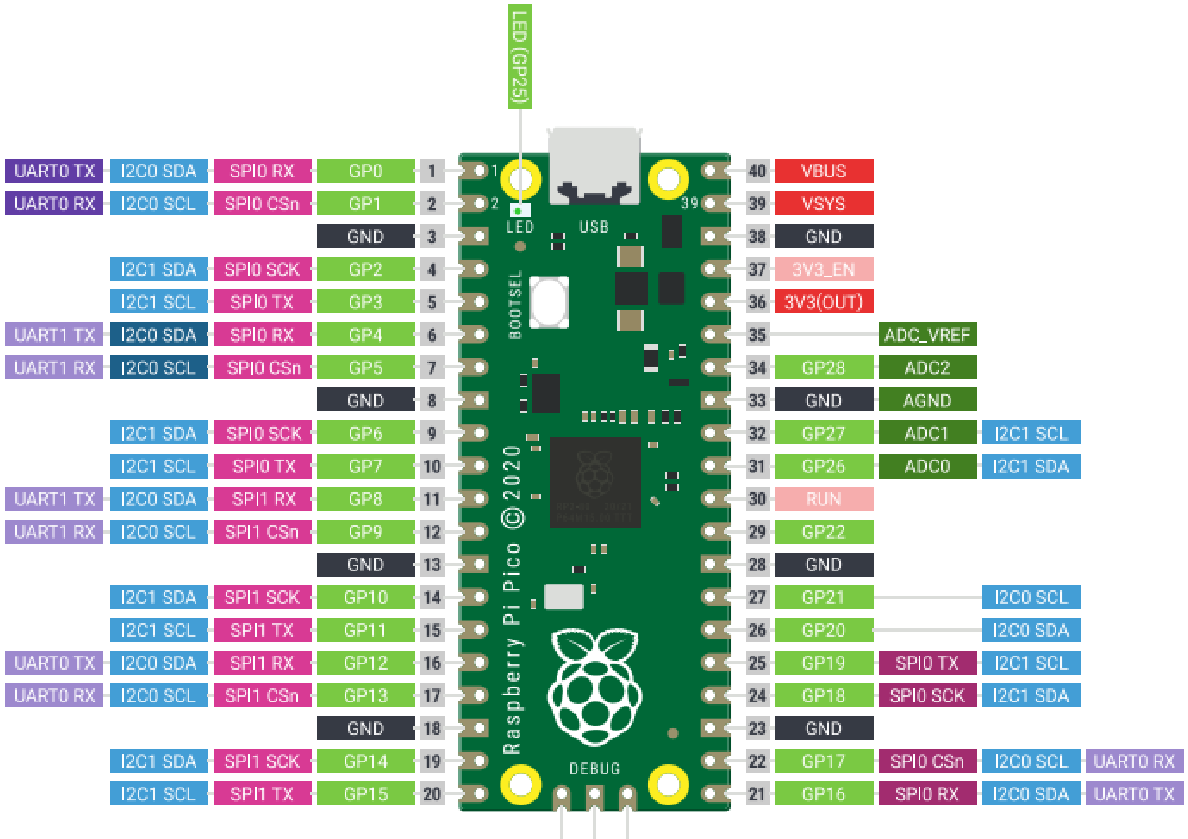


Figure Ⅲ: Raspberry Pi Pico Pinout Diagram

4.6.3. General Validation

There are several different requirements that this block needed to meet in order to be included into the system. Some of the most important specifications for this block were its ability to send data over long distances at a low power cost. After some research a LoRa module was decided as the best choice to meet all the requirements and specifications of the other blocks, the engineering requirements, as well as being able to easily interface with the microcontroller board and the other wireless communication transceiver.

The first main requirement from this block was its ability to transmit data over long distances of at least one hundred meters. With the LoRa module as our form of wireless communication, the board is able to transmit data over ten kilometers which far exceeds the system requirement. Because of this, LoRa became much more of a reliable choice at the beginning of the design phase.

Another very important requirement of this wireless communication transceiver block is low power consumption when transmitting and the ability to go into sleep mode while not transmitting to save on the overall power cost of the board. Once again, the LoRa module meets and exceeds both of these requirements. With a typical draw of 0.1 uA during sleep mode , the LoRa module makes it easy to save power whenever possible. And with a maximum of 130 mA draw during transmit mode and only 16 mA during receive mode, the module makes power consumption very manageable for long periods of time. Because the overall system needs to be powered for up to six months without external power, meaning the system will receive its power from a solar panel and batteries and not be connected to a constant power source such as a wall plug, having a low power wireless transceiver became very important.

Finally, the LoRa breakout board has the same operating voltage range as the sensors used in the other blocks, as well as an interface that is easily integrated into the system. The LoRa board uses the SPI protocol to interface with the raspberry pi pico microcontroller which makes it very easy to program and receive and send data to. Additionally, with this specific breakout board there are many supported arduino libraries and online documentation on support for using LoRa and LoRaWan with raspberry pi boards.

Overall, the LoRa breakout board fits all of the requirements of the project such as power consumption, easily interfaceable, and same voltage operating range as the other blocks in the project. Finally, choosing a specific version of the LoRa breakout board was important as well, thankfully the RFM95W board from Adafruit was in stock, cheap, and had great documentation and online support.

4.6.4. Interface Validation

Mcrcntrllr_wrlss_cmmnctn_trnscvr_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Other: Logic Level 3V3	The specific RFM95W LoRa module specifies an operating voltage of 3.3V logic level.	The raspberry pi pico will be able to generate a 3.3V logic level signal from any of its outputs which is enough to power the LoRa module.
Protocol: SPI	The LoRa module specifies SPI as its	The microcontroller Pico is able to provide

	chosen protocol to interface with the microcontroller or other wired devices.	and receive SPI signals and according to the LoRa datasheet it requires SPI for communication.
Data Rate: 100Kbps	The connection between the microcontroller and LoRa module can be programmed up to 300 Kbps, having a high enough data rate between these two connections is important.	The datasheet for the RFM95W LoRa breakout board boasts a programmable data rate of up to 300Kbps with a specified data rate of 100 Kbps is very easy to achieve.

wrlss_cmmctn_trnscvr_wrlss_cmmctn_rcvr_rf

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Protocol: LoRaWAN	The LoRa module makes its best long range, low-power performance when paired along with the LoRaWAN protocol for wireless communication.	The wireless communication transceiver can make very good use of the LoRaWAN protocol as its preferred method of wireless communication.
Messages: Packet:	The sensor blocks connected to the microcontroller will be arranged into packets of data containing pH sensor data, temperature sensor data, and soil moisture sensor data to be sent from the transceiver over the wireless channel to	The LoRa module excels at being able to send smaller packets of information over the LoRaWAN protocol to the end device. These packets will be relatively small as they will only contain number values and

	the end device.	not very many of them.
Other: rF. LoRaWAN operates over the 915MHz channel or the civilian band in the US	LoRaWAN operates over the 915MHz channel or the civilian band in the US	The US LoRa module is designed to work in the civilian band at a frequency of 915MHz.

4.6.5. Verification Process

The overall verification plan for the wireless communication transceiver block is quite simple and can be seen as a step-by-step process detailed below. A brief overview of the verification plan follows. In order to verify that the wireless communication transceiver is working as intended, there simply needs to be some data sent over the wireless communication protocol to the end device thus proving the connection was successful and the information was sent without error.

1. Connect the LoRa module to the Raspberry Pi Pico microcontroller with an LED on the breadboard to light up whenever the LoRa sends information over the wireless network
2. Connect to the LoRaWan gateway that has been setup in the general Corvallis area.
3. Generate arbitrary data through an onboard sensor of the Pico or simply create arbitrary data and hardcode them in as the information sent does not matter.
4. Send the information packet from the Pico and LoRa module through the gateway and to the end device. This should be shown on the circuit with a green LED lighting up when the LoRa is transmitting the packet.
5. On the end device, find the location where the packet was received and check if the information in the packet sent matches the information in the packet that was received. If it does, the block works correctly and its functionality has been verified.

4.6.6. References and File Links

4.6.6.1. References

[1] “Federal Communications Commission | The United States of America.” [Online]. Available: https://transition.fcc.gov/oet/spectrum/table/fcctable.pdf. [Accessed: 11-Feb-2023].

4.6.6.2. File Links

Adafruit RFM69HCW and RFM9X LoRa Packet Radio Breakouts Datasheets	https://learn.adafruit.com/adafruit-rfm69hcw-and-rfm96-rfm95-rfm98-lora-packet-padio-breakouts/downloads
Raspberry Pi Pico Datasheet	https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

4.6.7. Revision Table

2/20/23	Added verification plan to document - Brennan Ventura
2/11/23	Rewrote block description and updated block description on student portal to match - Brennan Ventura
2/11/23	Added title page and table of contents - Brennan Ventura

4.7. User Interface

4.7.1. Description

Windows app user interface software capable of directly connecting to the receiver through a USB serial connection. Through this software, the user will be able to request data stored locally in the receiver. The user will operate the app using clickable buttons and an argument test field. Read only text fields are used to convey information to the user. Whenever the user enters an invalid input for the operation they request, a popup will warn them that their input type is incorrect. This block is important in fulfilling the data display (requiring a method of outputting sensor data as a csv file) and the clear instruction (requiring a consensus of 9/10 people asked) requirements.

4.7.2. Design

The user interface block consists entirely of a C++ Win32 API[1] driven application that communicates with the receiver over a standard serial USB connection. The general structure of the application is event driven: the code will idle until an event (button press) occurs, and then the application will execute code associated with the pressed button. This reactive architecture is the default for simple GUI driven applications. For access to the exact source code used in the project, please see the File Links section below.

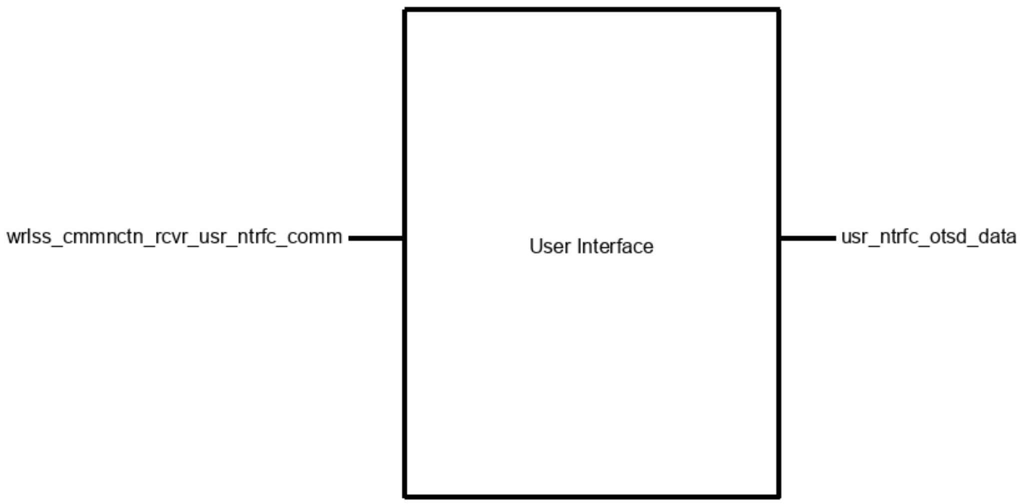


Figure Ⅰ: User Interface Black Box

4.7.3. General Validation

As a block consisting entirely of software meant to run on a standard PC, regular design choices such as cost, manufacturability, source of components, etc; have absolutely no bearing on design decisions made. The only aspects of design that factored into the system architecture design included simplicity (to aid development and maintenance) and API choice. I chose the Win32 API because of the low level unified control it allowed me to take, without needing to use multiple libraries to solve the different tasks of serial communication and GUI rendering.

In hindsight, the benefits offered by the Win32 API may not have outclassed the benefits presented by unified cross platform frameworks. It may have been simpler to use a few Python libraries. Although the Win32 API does provide a unified interface for all system utilities, the documentation for the API is dense and iteration on the GUI is slow and comes at a significant time expense compared to other blocks in the system.

4.7.4. Interface Validation

wrlss_cmmnctn_rcvr_usr_ntrfc_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**

Data Rate: 9600 Baud	Standard serial communication rate	Standard PC hardware supports this baud rate
Messages: {Timestamp: , Temp1: , Temp2: , Temp3: , Moisture1: , Moisture2:, pH: , Flags: }	Communication format designed by group	Designed to conform to this format
Protocol: RS232	Selected serial protocol	Standard PC hardware supports this protocol

usr_ntrfc_otsd_data

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Messages: Timestamp,Temp1,Temp2,Temp3,Moisture1,Moisture2,pH,Flags	The sensor blocks connected to the microcontroller will be arranged into packets of data containing pH sensor data, temperature sensor data, and soil moisture sensor data to be sent from the transceiver over the wireless channel to the end device.	Having a specified packet structure makes converting the data into a .csv file much easier.
Other: Operating System: Microsoft Windows	Supported by Win32 API	Compiles into valid Windows exe
Protocol: CSV	Standard tabled data format	Test outputs conform to the file convention

4.7.5. Verification Process

The verification plan for the user interface will include the use of a dummy interface, specifically an Arduino Mega 2560 that has been flashed to behave like a receiver without actually having to use a wireless protocol to receive data.

1. Flash dummy with test code so that it responds to requests from the user interface as if it was the receiver block.
2. Establish a connection with the dummy by finding the port it is on and completing the handshake procedure.
3. Request data from the dummy receiver and store the data in the application by clicking on the appropriate button.
4. Paste path information for the file to store the data in csv format into the argument field and press the appropriate button.

4.7.6. References and File Links

4.7.6.1. References

[1] Drewbatgit, “Programming reference for the win32 API - win32 apps,” Win32 apps |
Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/api/. [Accessed: 07-May-2023].

4.7.6.2. File Links

User Input Code	UI_Code.cpp

4.7.7. Revision Table

5/7/23	Caleb Walker: Initial draft of block validation

4.8. Microcontroller

4.8.1. Description

The microcontroller block is responsible for managing sensor and communications devices on the probe sub-system, as well as deciding when recorded sensor data has strayed far enough from previously reported results to report (primary guarantor of the Dynamic Polling engineering requirement). Other engineering requirements supported, but not exclusively the responsibility of the microcontroller block include Data Display (sensor readings are translated into human understandable units before being reported) and Power Requirement (aided by the frequently utilized builtin DORMANT mode). Design decisions for this block were made with these three requirements as guiding principles.

Management is handled through two primary communication paths, one I2C bus containing the sensor devices and an SPI bus for the LoRa Transceiver block. Two digital output lines are also used to selectively control when sensors go to sleep (one for the pH sensor, and another for every other sensor). Internal to the block, a simple DIP switch bank will be used to set software settings (a debug mode and pH reading frequency).

Management behavior is handled through firmware installed to the device from a development workstation well before system deployment. This firmware will likely not be final until project completion due to small tweaks in values to accommodate changes as desired. In general, the microcontroller will sleep for a predefined time interval. When the microcontroller wakes up, a quick reading of temperature and moisture values will be made across all sensors of these types after the sensors are woken up by the digital control pin. The pH sensor may only be read occasionally, depending on device settings. The data collected from sensors will then be used to generate a packet of information to be sent to the LoRa Transceiver for transmission.

4.8.2. Design

The design of the block itself centers largely around the Raspberry Pi Pico, selected for reasons that will be presented in the General Validation section. Presented figures show the block-level design of the microcontroller with appropriate input and outputs, a circuit schematic for the block made in the KiCad schematic editor (labels on wires are used to convey to which interfaces wires belong to), and pseudo code describing the firmware behavior of the microcontroller block. For full firmware code, please see the References and File Links section, which will contain a file link to the most up to date code.

The firmware, although important to the operation of the block; will not be under review for the block verification. This is because the behavior of the block during intended operation will be very different from a desirable testing environment. This is due to factors including but not limited to: long wait times between measurements, low utilization of bus bandwidths per unit time (we aren’t reading from sensors or communicating wirelessly very frequently), and very infrequent use of the pH sensor (meaning no way to test the switching speed of the digital interface). Instead, custom firmware will be created, specifically to make verification an easier process.

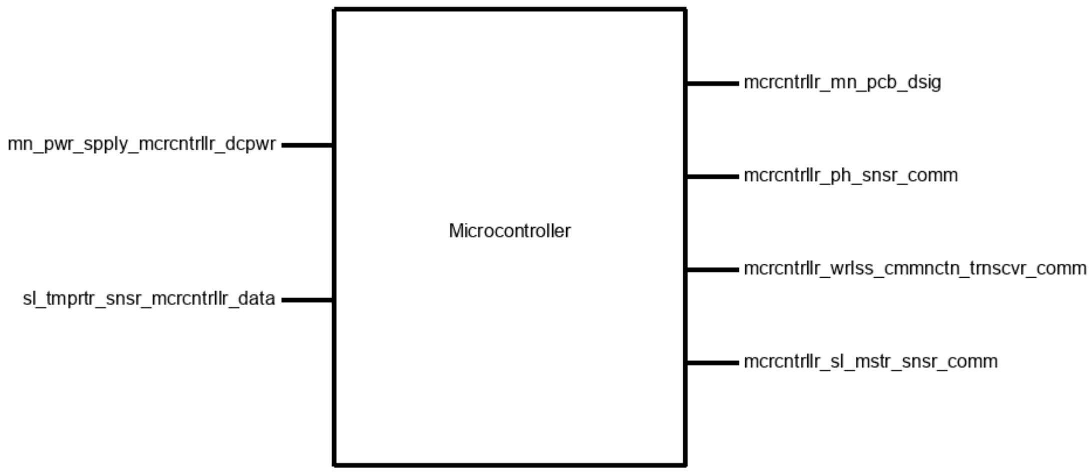


Figure Ⅰ : Black Box Diagram

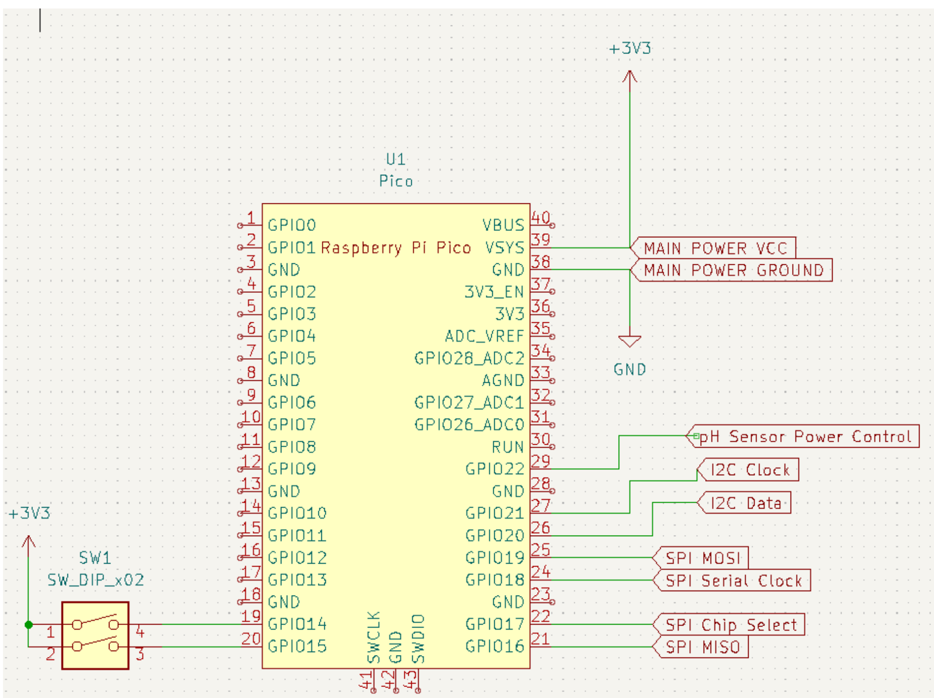


Figure Ⅱ : Microcontroller Wiring Schematic

```
MAIN LOOP:
        WAIT for READ_INTERVAL
        WAKE temp_sensors and moisture_sensors
        IF PH_SWITCH OR PH_TIMER:
                WAKE pH_sensor
                READ pH_sensor
                SLEEP pH_sensor
                CLEAR PH_TIMER
        READ temp_sensors and moisture_sensors
        SLEEP temp_sensors and moisture_sensors
        CHECK_VARIANCE new_data vs old_data
        IF VARIANCE is large enough OR DEBUG_SWITCH:
                SEND PACKET
                LISTEN for 2S
                IF PACKET_RECEIVED:
                        INTERPRET PACKET
        CLEAR READ_INTERVAL
        IF DEBUG_SWITCH:
                SET READ_INTERVAL to SHORT_INTERVAL
        ESLE:
                SET READ_INTERVAL to NORMAL_INTERVAL
        GOTO MAIN LOOP
```

Figure Ⅲ: Pseudocode for Microcontroller

4.8.3. General Validation

The Raspberry Pi Pico was selected for this system for a number of key reasons, first of which being hardware and software support. As a big player in the hobby electronics scene, the Raspberry Pi Foundation provides high quality and easy to find documentation on any and all features of the Pico. This includes very good software toolchain support, with resources on the official website for setting up both MicroPython and C/C++ toolchains. On top of these resources, a vast library of example and sample code can be found for many different types of peripherals as well as standardized features on the heart of the Pico, the RP2040 microcontroller [1]. Another factor that makes the Pico so easy to develop for is the micro USB programming port, and the fact that programming is as simple as dragging and dropping a binary file to a storage device. The other group of factors that made the Pico a top contender to fulfill the requirements of this block were the electrical characteristics of the board. Of particular interest in a battery powered scenario is the current consumption of a device. According to the official datasheet, the Pico draws below 2mA when in SLEEP mode and below 90mA during periods of peak activity [2], which is an activity level we will likely not reach in our simple application. This relatively low current draw aids in the system meeting its Power Requirement (one of the overall engineering requirements) allowing the system to run off of an attached rechargeable battery for at least six months by reducing the rate at which the battery is depleted. The board also supports a very broad range of input voltages (1.8V to 5.5V) [2] which is nice, but not entirely necessary due to the separate power supply sub-system already supplying 3.3V to the microcontroller and all sensors.

In terms of processing power, the Pico delivers in spades for our application. A dual core ARM 32 bit processor, 264kB of SRAM, and 2MB of FLASH [1] is overkill for a simple measure and report system but the hardware has many other builtin features (such as flexible pin locations for builtin I2C and SPI ports). Higher processing capabilities however, will aid in meeting the system's Dynamic Polling requirement, as the Pico can quickly make comparisons with previously recorded values and make quick decisions; with a tack on effect of reducing the amount of time we are not in DORMANT mode. These features (as well as the previously discussed software and hardware support) make the Pico well worth the $4 USD for our system. This price tag may have worked against the Pico had the system been designed to be made at massive scale (a cheaper microcontroller would surely suffice), but in the design window given, the support and detailed documentation of the Raspberry Pi Foundation won out. Without the constraint of time, a better option might have been a specialty low power microcontroller, which can be found to draw current in the area of a few microamps during active operation, however extra time would be needed for both the software toolchain as well as developing programming hardware.

4.8.4. Interface Validation

mcrcntrllr_mn_pcb_dsig

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Logic-Level: 3V3	Standard value provided by microcontroller [2]	Reported in RP2040 datasheet [1]
Max Frequency: 1kHz@100nF	Acceptable switching rate for expected capacitive load (MOSFET gate)	Performed RC circuit analysis of charging capacitor that would not violate 10mA Imax value

Other: Imax: 10mA	To protect the microcontroller from sinking/sourcing too much current	Well below allowable GPIO current for RP2040 [1]

mcrcntrllr_ph_snsr_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Datarate: 100kbps	Lowest I2C standard speed (wide device support)	Reported in RP2040 datasheet [1]
Other: Logic level: 3V3	Standard value provided by microcontroller [2]	Reported in RP2040 datasheet [1]
Protocol: I2C	Protocol supported by blocks on either side of the interface	Reported in RP2040 datasheet [1]

mcrcntrllr_wrlss_cmmnctn_trnscvr_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Other: Logic-Level: 3V3	Standard value provided by microcontroller [2]	Reported in RP2040 datasheet [1]
Protocol: SPI	Protocol supported by blocks on either side of the interface	Reported in RP2040 datasheet [1]
Datarate: 100kbps	Chosen to be in sync with I2C bus for	Reported in RP2040 datasheet [1]

	simplicity	

mcrcntrllr_sl_mstr_snsr_comm

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Datarate: 100kbps	Lowest I2C standard speed (wide device support)	Reported in RP2040 datasheet [1]
Other: Logic-Level: 3V3	Standard value provided by microcontroller [2]	Reported in RP2040 datasheet [1]
Protocol: I2C	Protocol supported by blocks on either side of the interface	Reported in RP2040 datasheet [1]

mn_pwr_spply_mcrcntrllr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Inominal: 2mA	Power draw in Pi Pico DORMANT mode	Pi Pico will frequently sleep in regular operation
Ipeak: 90mA	Power draw demonstrated in intense use case in RP2040 documentation [1]	Pico will never be used for an intense computation load as demonstrated in documentation
Vmax: 5V	Standard value above	Within allowable

	3V3 (selected for flexibility)	voltage in Pico documentation [2]
Vmin: 2V	Lower than expected battery input, but higher than Pi Pico minimum limit [2]	Within allowable voltage in Pico documentation [2]

sl_tmprtr_snsr_mcrcntrllr_data

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Datarate: 100kbps	Lowest I2C standard speed (wide device support)	Reported in RP2040 datasheet [1]
Other: Logic-Level: 3V3	Standard value provided by microcontroller [2]	Reported in RP2040 datasheet [1]
Protocol: I2C	Protocol supported by blocks on either side of the interface	Reported in RP2040 datasheet [1]

4.8.5. Verification Process

Verifying the microcontroller block will require specialized hardware and firmware, as devices specifically designed to test serial interfaces specifically are difficult to come by and expensive. Therefore, specialized testing hardware has been designed specifically for testing the output interfaces. A schematic has been included in this section, depicting the schematic of the hardware added to the block for testing.

The primary components added for testing are an I2C compatible EEPROM storage device and another Raspberry Pi Pico. The storage device will be loaded with a known dataset, which will be read by the primary Pico (the one within the microcontroller block). This data will then be sent through the SPI port to the other Pico, which will then send the data through its USB serial port to be viewed by a connected computer for

visual verification. This second Pico has the benefit of allowing the SPI interface to be tested, as well as preventing the primary Pi from being connected to the debugging USB interface, which would interfere with testing the DC power interface (because some or all power would be provided by the USB port).
After verification hardware has been assembled and programmed, the block is ready for testing. Testing procedure will be outlined below:

1. Apply a 2V voltage DC signal from a benchtop power supply to the DC Power interface, subsequent steps will confirm that the block operates at the minimum interface rated voltage.
2. Connect MCU to PCB digital signal interface to load tester, set to a simulated load of 10mA. Hold for at least 30 seconds.
3. Remove simulated load, and connect the interface to the frequency test circuit. Switch Pico to toggle test mode, using the attached DIP switch. Then attach an oscilloscope probe to the interface, analyzing the waveform to ensure that a logical HIGH and LOW value are met before the signal switches (should see a square wave with a capacitive charging/discharging curve on the leading edge).
4. Attach oscilloscope probes to both the I2C serial clock pin and I2c data pin, and either an oscilloscope or a frequency counter (whichever is available), ensuring that the clock cycles at at least 100kHz (specifically for the clock channel), and that voltages stay between 3.3 and 0 Volts, on both channels.
5. Attach Oscilloscope probes to the SPI serial clock, MISO, and MOSI channels. Make sure that the clock pin has a frequency of at least 100kHz, and that all channels stay between 3.3 and 0 Volts.
6. Observe data being output to the serial terminal emulator, and ensure that it matches with the known data stored on the EEPROM device.
7. Apply a 5V voltage DC signal from a benchtop power supply to the DC Power interface.
8. Repeat steps 2-6 for new input voltage.

4.8.6. References and File Links

4.8.6.1. References

[1] Raspberry Pi Ltd, RP2040 Datasheet: A microcontroller by Raspberry Pi, Raspberry Pi
Ltd, 2020.

[2] Raspberry Pi Ltd, Raspberry Pi Pico Datasheet: An RP2040-based microcontroller board, 2020.

4.8.6.2. File Links

Microcontroller Block Firmware	MCU_BLOCK_Firmware
Microcontroller Block Test Firmware (Master)	MCU_Block_Test_MASTER
Microcontroller Block Test Firmware (Slave)	MCU_Block_Test_SLAVE
Microcontroller Block Test Firmware (EEPROM Programmer)	MCU_Block_Test_EEPROM_Programmer

4.8.7. Revision Table

01/20/23	Caleb Walker: Initial draft
2/11/23	Caleb Walker: updated written content taking into account provided feedback. Updated interfaces, and verification plan to reflect changes leading up to project verification.

4.9. Power/Main PCB

4.9.1. Description

The power block is responsible for providing a steady 3.3 Volts to the other blocks located on the probe. This purpose is complemented by the structural support of buses and interconnects provided by the PCB (on which the power block is central). The operation of the block is directly tied to the mission lifespan (an engineering requirement requiring six months of continuous operation). Design decisions were made including: solar panel selection and capacity, battery selection and quantity, voltage regulator throughput, trace width, and others; to ensure the system mission lifetime could be met as well as the system being fully capable of powering every block at once (a situation which should never occur). Power is provided by the sun via the solar panels, which is then directly used to power a LFP battery charge controller. The battery sits between both the charge controller and the switching regulator, receiving current via the charge controller which can be tapped by the regulator if required (with the battery acting as a buffer to catch overflow and underflow from the charge controller).

4.9.2. Design

The block's design is composed of four main stages that power travels through. The solar panels act as the probe's power capture, and its primary purpose is to simply capture energy from the sun for use by the rest of the probe assembly. The second stage, the charge controller, takes the power generated by the solar panels and converts it into a safe charging algorithm to charge the LFP batteries (preventing damage, fire, and other issues that would arise from directly charging the batteries via the solar panels). The third stage, the batteries, serve as a very large energy buffer: capturing excess energy during surplus periods and supplying the stored energy back during lean times. The final stage takes power directly from the batteries and converts it into a voltage that is useful to the myriad of devices in the probe assembly.

All devices that receive power from the power block due so through a shared 3.3V power rail. With all devices capable of drawing from the same line, it is important that trace widths are selected appropriately to prevent excessive voltage drop across the line.

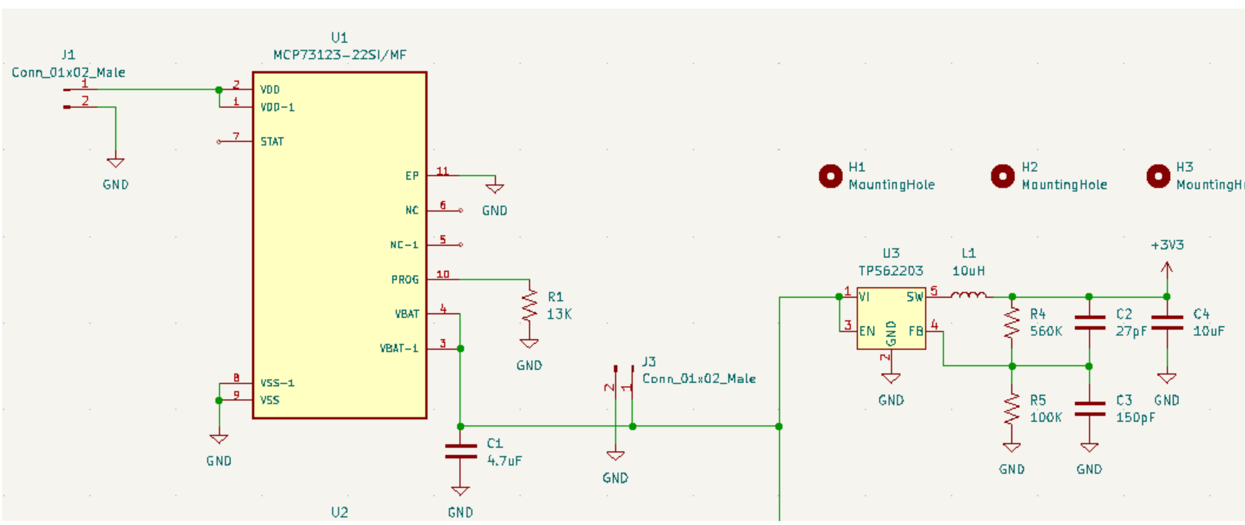


Figure Ⅰ: Schematic of Power Block in KiCad

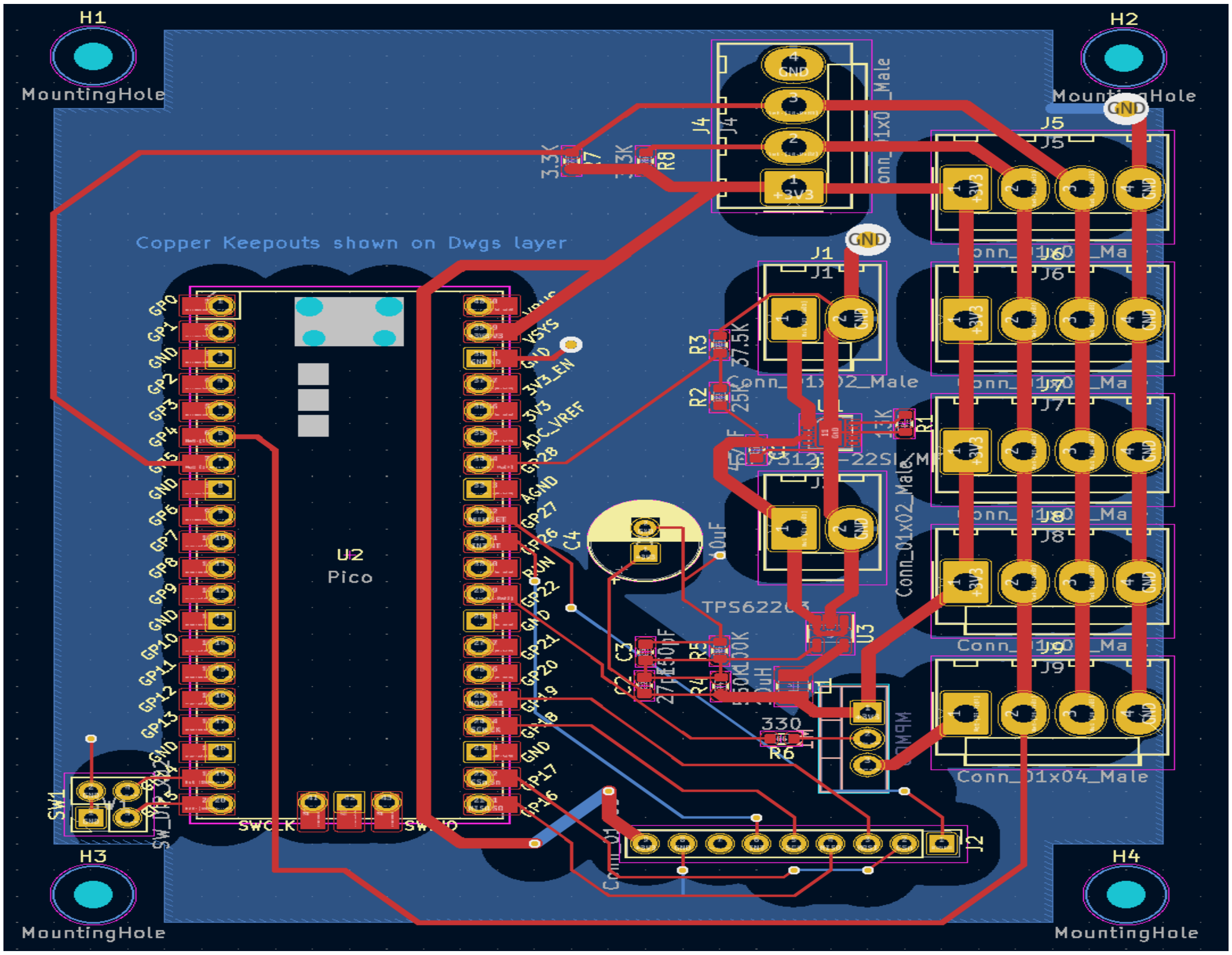


Figure Ⅱ: Final PCB containing Power Block, Microcontroller Block, as well as interconnects for other blocks

4.9.3. General Validation

The overall architecture of the block (the simple four stage structure) made the most sense for the probe assembly with its meager power requirements and simplified what could have otherwise been a complex power architecture. The solar panels and batteries selected for capacity were chosen with mission lifetime in account, with batteries alone capable of supplying the probe assembly for several weeks. As for battery chemistry, LiFePo4 was selected due to its superior performance at more extreme temperatures compared to traditional lithium ion cells[1].
The regulator selected was chosen primarily for minimum required cost while meeting the current and voltage requirements of the output interfaces. The battery charge controller was primarily selected based upon cost and availability, as not many charge controllers are easy to source based on our selected battery chemistry.

4.9.4. Interface Validation

mn_pwr_spply_ph_snsr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Inominal: 22mA	Current needed by device	Regulator and battery can both provide enough current
Ipeak: 30mA	Maximum current device may draw	Regulator and battery can both provide enough current
Vmax: 5V	Max allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)
Vmin: 3V	Minimum allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)

mn_pwr_spply_mcrcntrllr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Inominal: 2mA	Current needed by device	Regulator and battery can both provide enough current
Ipeak: 90mA	Maximum current device may draw	Regulator and battery can both provide enough current

Vmax: 5V	Max allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)
Vmin: 2V	Minimum allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)

mn_pwr_spply_sl_mstr_snsr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Inominal: Less than 1mA	Current needed by device	Regulator and battery can both provide enough current
Ipeak: Less than 1mA	Maximum current device may draw	Regulator and battery can both provide enough current
Vmax: 3.6V	Max allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)
Vmin: 1.6V	Minimum allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)

mn_pwr_spply_sl_tmprtr_snsr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details for this block above meet or exceed each property?**

Inominal: Less than 1mA	Current needed by device	Regulator and battery can both provide enough current
Ipeak: Less than 1mA	Maximum current device may draw	Regulator and battery can both provide enough current
Vmax: 3.6V	Max allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)
Vmin: 1.6V	Minimum allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)

mn_pwr_spply_wrlss_cmmnctn_trnscvr_dcpwr

Interface Property	**Why is this interface property this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Inominal: Less than 1mA	SLEEP MODE current	Regulator and battery can both provide enough current
Ipeak: 120mA	Maximum current in highest power transmission mode	Regulator and battery can both provide enough current
Vmax: 3.7V	Max allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)
Vmin: 1.8V	Minimum allowed voltage by device	Regulator regulates to 3.3V (and variation does not exceed limits)

4.9.5. Verification Process

The verification process for this block includes two aspects: a runtime test: where current and voltage outputs are tested, and a lifetime simulation: where we analyze whether the system could operate on solar power for the entire mission lifetime. In the File Links section below, a link to the Python simulation script can be found. For the live test the following steps can be followed.

1. Attach battery to battery connection point, and apply 5 Volts power over the solar connection point (to remove dependance on solar power for short term testing).
2. Set load simulator to maximum possible load current at once (the sum of all Ipeak values)
3. Observe voltage while the load simulator runs for 30 seconds. If voltage stays above 3V, we have satisfied the power requirements of all devices.

4.9.6. References and File Links

4.9.6.1. References

[1] ECOFLOW, “LiFePO4 vs. Lithium Ion Batteries: What’s the Best Choice for You?,” EcoFlow US Blog, Mar. 08, 2023. https://blog.ecoflow.com/us/lifepo4-vs-lithium-ion-batteries/#:~:text=LiFePO4%20batteries%20offer%20a%20wider%20operating%20temperature%20range. (accessed May 07, 2023).

4.9.6.2. File Links

Python Solar Simulation Script	**Solar_sim.py**

4.9.7. Revision Table

5/6/23	Caleb Walker: Initial draft

5. <u>System Verification Evidence</u>

5.1. Universal Constraints

5.1.1. The system may not include a breadboard.

The final system contains no breadboard and uses student designed PCBs throughout the system. The different sensor boards are pictured outside of the probe because they would be very difficult to see while inside the probe. A picture of different components outside the system can be seen below along with a picture of the assembled system. As can be seen in the photos there are no breadboards in the system

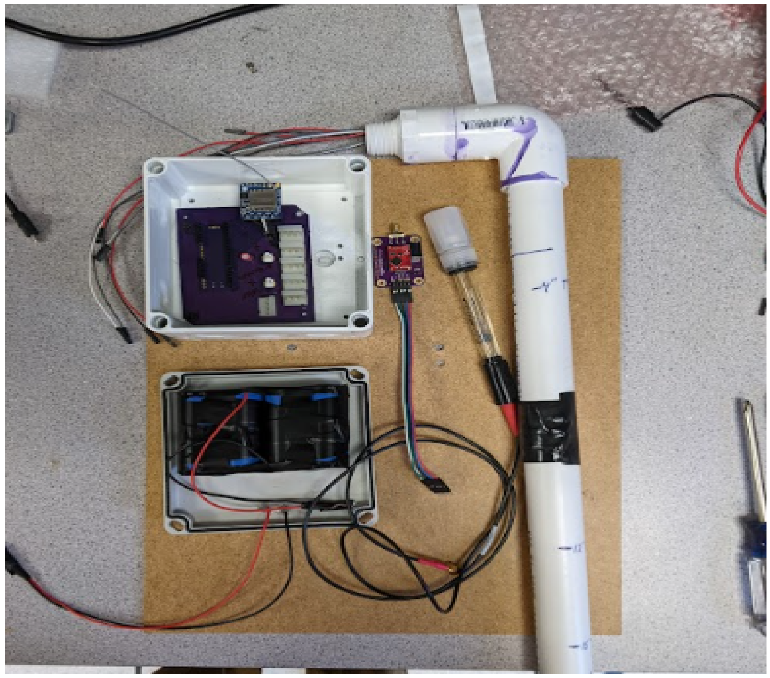


Figure Ⅰ: Assembled, Unconnected System

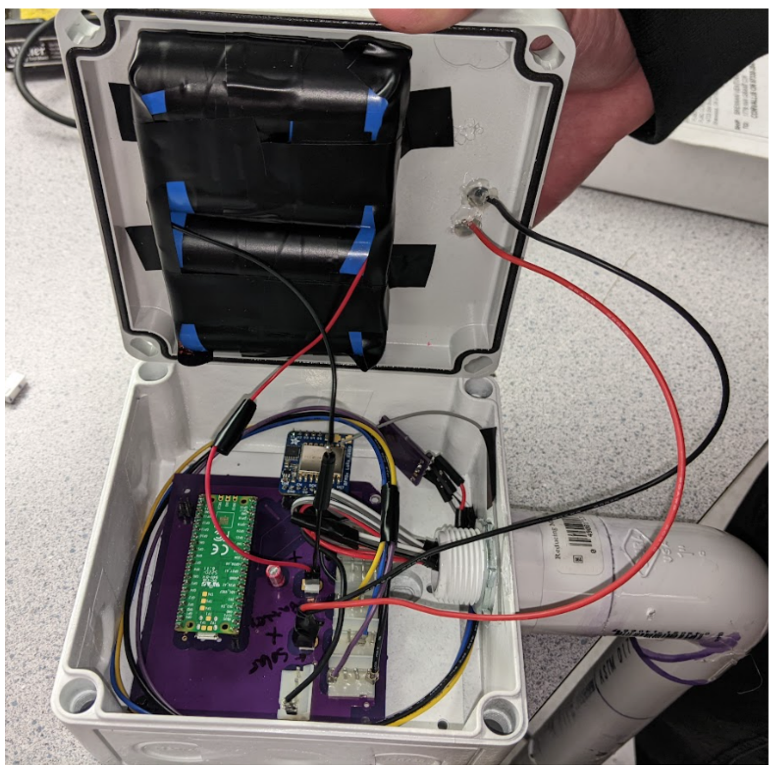


Figure Ⅱ: Assembled System

5.1.2. The final system must contain a student designed PCB.
The system contains several student designed PCBs in the implementation. There is a student designed PCB for the temperature sensor unit, soil moisture sensor unit, and the main PCB which houses

the microcontroller, power charger/converter circuit, and the connections for each of the various components. The main PCB contains over 30 SMT pads and the final schematic along with a picture of the fully assembled board can be seen below.

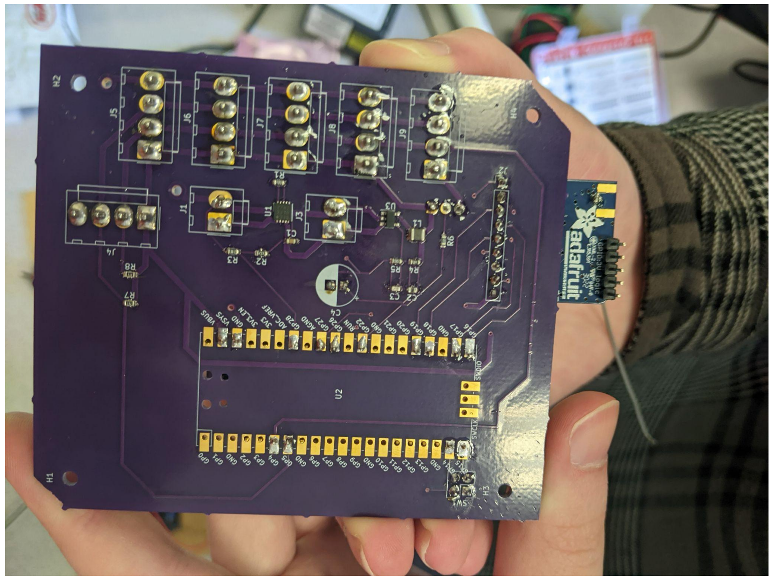


Figure Ⅲ: Assembled PCB with 41 SMT Pads

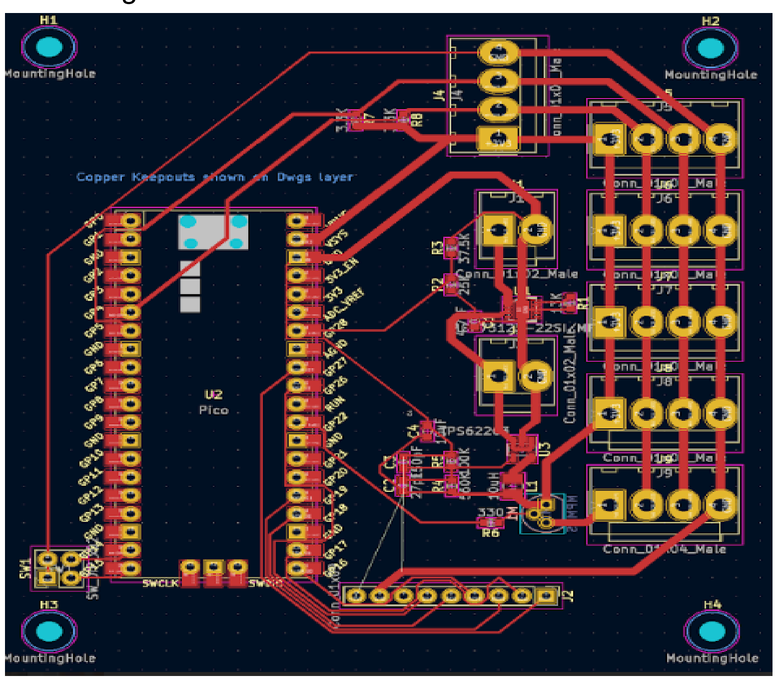


Figure Ⅳ: Final PCB Schematic

5.1.3. All connections to PCBs must use connectors.

The final system contains three student designed PCBs along with a modified purchased PCB. All three of the sensor PCBs use male header pins to connect to the main PCB. The main PCB uses a combination of regular male header pins and male Molex pins to connect the peripheral devices. The wireless receiver sub-system uses an arduino shield along with male header pins and jumper wires to form the connections. A picture of each of the PCBs is provided below showing the male header pins attached.

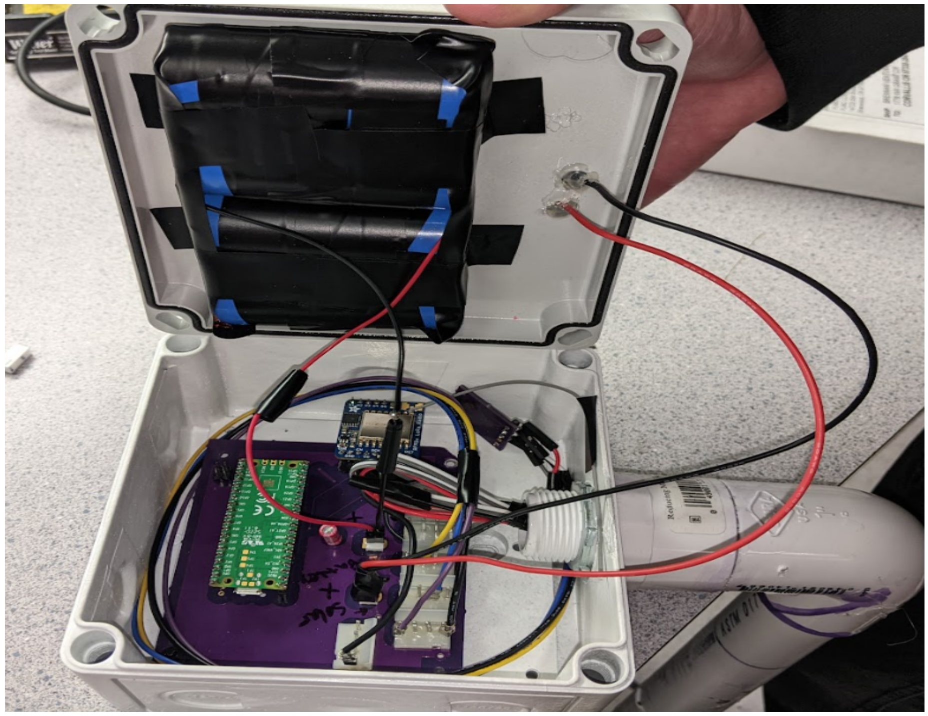


Figure Ⅴ: Assembled System with no wires directly soldered

5.1.4. All power supplies in the system must be at least 65% efficient.

The system power supply consists of two solar panels as the energy source along with four LiFePO4 batteries to hold charge. Because of this most of the devices draw a very small amount of power allowing the system to last long amounts of time on its own. A simulation of the system's power usage was modeled in python that includes how much power the solar cells would provide using data from sun movement patterns to more accurately model how much power the photovoltaic cells could collect. The simulation results can be seen below with the python code provided in the file links section. Moreover, the power converter used in the system has a rated accuracy up to 95% with our load always being between 10mA and 100mA.

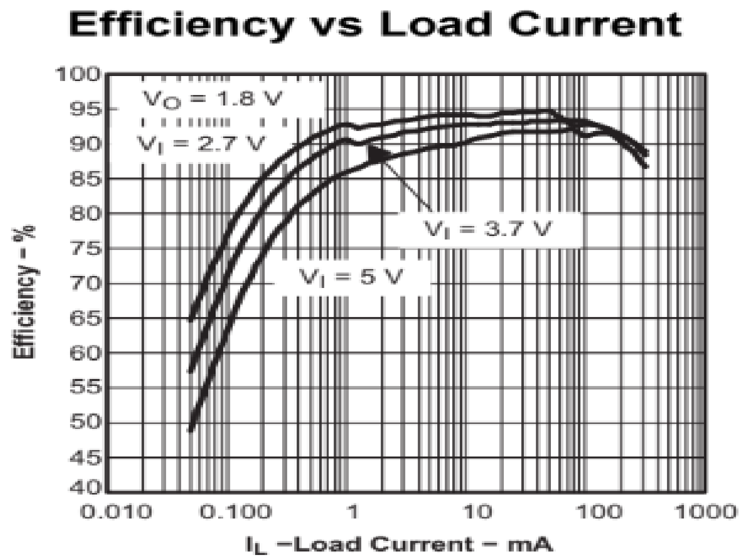


Figure Ⅵ: Step-Down DC Converter Efficiency Chart

5.1.5. The system may be no more than 50% built from purchased modules. There are many devices in the system so to best prove that the final system is no more than 50% built from purchased 'modules' a table is provided below to show the part along with whether it is built or bought and the reasoning to defend that choice.

Part	Built/Bought	Reason
Microcontroller	Built	The Raspberry Pi Pico board was coded with designed firmware specifically for this project and set of sensors, therefore it is a built module.
Enclosure	Built	The polycase enclosure was bought online, however there were several modifications made to the enclosure thus making it a built module.
pH Sensor	Bought	The pH sensor and interpreter board were purchased online and there were no modifications made to

		it. Moreover, there was a premade library in arduino used to collect the reading, thus making this a bought module.
Moisture Sensor	Built	The moisture sensor uses a student designed PCB along with student written code to collect the reading from the board, thus making it a built module.
Temperature Sensor	Built	The temperature sensor uses an IC sensor along with a student designed PCB board. This makes the temperature sensor a built module.
Wireless Communication Transceiver	Bought	The wireless communication transceiver uses an adafruit LoRa module connected to the main microcontroller. However, because the code is written to the microcontroller unit this is counted as a bought block.
Wireless Communication Receiver	Built	The wireless receiver consists of a data logging shield, LoRa module, and an arduino uno. There was student designed code for this module along with modifications to the data logging shield, making this a built block.
Main Power Supply	Built	The main power

		supply is built from two photovoltaic cells, four LiFePO4 batteries, and a charging/converting circuit. There were modifications made to the solar cells and batteries making this a built block.
User Interface	Built	The user interface consists of code written for the system and a GUI. The code was not entirely written from a premade library and had heavy student influence, thus making this a built block.
Main PCB	Built	This module was completely designed by a student and assembled with the SMT and thru-hole components by a student, thus making this a built block.
Percentage of Blocks Built	Total number of modules: 10 Percentage of Built Blocks: 80%	

5.2. Requirements

5.2.1. Accuracy Rating

5.2.1.1. **Project Partner Requirement:** The sensor values will have an adequate accuracy rating.

5.2.1.2. **Engineering Requirement:** The system will report moisture, and pH that is 90% accurate, and temperature within 5 % or 3 degrees whichever is greater.

5.2.1.3. **Testing Method**: Test

5.2.1.4. **Verification process:** The pH sensor will first be put into a liquid of known pH and the output .csv file pH column will be compared to the known pH this will then be repeated for a second known pH and once again compared to the captured value recorded by the probe. The temperature values on the output .csv file will be compared to an independent thermometer. Using an independent soil moisture sensor a benchmark for soil moisture levels will be taken. Then using the student designed soil moisture sensor the value from the independent soil moisture sensor will be compared with the data collected from the probe.

Pass Condition: The output values for moisture in soil and pH are within 90% of the value known or based on independently found values, and temperature is within 5% or 3 degrees, whichever is greater.

5.2.1.5. **Testing evidence**: This requirement could not be met due to several reasons. The most important reason was due to code not working on the Raspberry Pi pico. The blocks were originally tested with an Arduino Uno however the libraries using the Arduino Uno were not compatible with the Raspberry Pi pico.

5.2.2. Clear Instruction

5.2.2.1. **Project Partner Requirement:** The project will be usable by non-engineering parties with no previous experience.

5.2.2.2. **Engineering Requirement:** The system will have clear documentation and instructions that at least 9/10 people report as 'easy to understand'.

5.2.2.3. **Testing Method:** Analysis

5.2.2.4. **Verification process:** A user manual describing how to deploy the system along with how to use the provided user interface will be provided on a google form and given out to at least 10 people. At least 9/10 people must report that they understand how to use the system based on the user manual.

Pass Condition: 9/10 people report that they understand how to use the system after reading the documentation and instructions.

5.2.2.5. **Testing evidence:** A survey was sent out along with a link to the user manual describing how to use the system. The testing evidence is a link to all the responses from that survey along with the percentage of people who claimed it was easy to understand. Verified 5/10, Capstone Survey (Responses) , Clear_Instruction_Verification.MOV

5.2.3. Communication Distance

5.2.3.1. **Project Partner Requirement:** The soil probe will be able to communicate wirelessly over a range of 100m.

5.2.3.2. **Engineering Requirement:** The system will function when the probe sub-system is at least 100m from the receiver sub-system.

5.2.3.3. **Testing Method:** Demonstration

5.2.3.4. **Verification Process:** To prove that the system can communicate wirelessly over a range of at least 100m. A video showing the system taking measurements and then sending those measurements to the receiver which is 100 meters away and seeing those same measurements show up in the user interface thus proving that the system can communicate over a range of 100 meters.

Pass Condition: The receiver/user interface correctly receives the sent packet consisting of the taken sensor outputs are displayed after being sent over a range of at least 100m.

5.2.3.5. **Testing Evidence:** The video starts with the receiver offline while the transmitter inside the system is powered. Two teammates holding the receiver then walk 100m away and power on the receiver and then check to confirm data is properly received. Verified 5/8 Comm_Distance_Verification.mp4

5.2.4. Data Display

5.2.4.1. **Project Partner Requirement:** The system must have a user interface where the data collected from the sensors can be displayed.

5.2.4.2. **Engineering Requirement:** The system will save the sensor data and present it as csv files.

5.2.4.3. **Testing Method:** Inspection

5.2.4.4. **Verification Process:** Using the user interface, we will have a number of packets collected from the probe sent over to the receiver. After some time has passed, using the user interface the data will be exported from the receiver onto a .csv file where it will be opened and inspected by the team to ensure the data is correct and different.

Pass Condition: The user interface correctly exports a .csv file with data collected from the various sensors. As long as the data is not the exact same for every packet the system will pass this requirement.

5.2.4.5. **Testing Evidence:** This video is to prove the data display requirement. The video starts with the receiver offline and not receiving any packets from the transmitter which is powered. Next the receiver is powered on and then we allow some time to pass in order for some data to populate the receiver. Then using the GUI it is confirmed that data is correctly read onto a .csv file as can be seen in the file explorer. Verified 5/8 Data_Display_Verification.mp4

5.2.5. Dynamic Polling

5.2.5.1. **Project Partner Requirement:** System should increase the amount of measurements taken when great change is detected from the sensors.

5.2.5.2. **Engineering Requirement:** The system will report data more frequently when a change greater than 5% is detected from the previous sensor data.

5.2.5.3. **Testing Method:** Demonstration

5.2.5.4. **Verification Process:** To show that the sensors can correctly dynamically poll a large stimulus will be provided to one of the sensors and then the data will be sent over to the receiver. When viewing the data, if the time between packets has decreased along with an appropriately large variation in one of the sensor fields for the packet then the dynamic polling works as intended.
Pass Condition: The data can be seen to have decreased time between the sending of each packet along with a large variation/deviation between same sensor fields, i.e. temp1 at time1 is much smaller than temp2 and time 2.

5.2.5.5. **Testing Evidence:** Because the libraries mentioned in the accuracy rating section were not compatible with the Raspberry Pi pico, dynamic polling of the sensors could not be implemented in time for the system verification and thus this requirement could not be met. For future projects, it is important to test and prototype on the same board as the final system implementation

5.2.6. Power Requirement

5.2.6.1. **Project Partner Requirement:** The system is able to be left unattended for at least six months.

5.2.6.2. **Engineering Requirement:** The system will operate for at least six months without any external power.

5.2.6.3. **Testing Method:** Analysis

5.2.6.4. **Verification Process:** Using a python simulation to simulate the solar panel input throughout the year, along with power ratings from the various data sheets, a calculation will be shown to prove that the system can last at least 6 months without any external power.
Pass Condition: The system takes in enough power from the solar panel on a day to day basis to last for six months according to the calculations and the python simulation does return an "unhealthy discharge" state.

5.2.6.5. **Testing Evidence:** In order to prove the power requirement, a python simulation was made to test each of the power consumptions by the various components against how much power would be received by the solar panel each day. The system passed the power requirement and the python script is attached below. Verified 5/7 Solar_sim.py

```
>>>
=========================== RESTART: C:\Users\caleb\SolarSim.py ===========================
setting up time ranges...
Calculating solar position over time range...
Grabbing weather data...
Calculating irradiance on panel over time range...
Calculating battery charge over time...
Checking for dangerous battery levels...
SIMULATION COMPLETE
 1409.3647215366364  Seconds elapsed
Power Test: PASSED
>>>
```

5.2.7. Rigid Structure

5.2.7.1. **Project Partner Requirement:** The system must be rigid and survive deployment in the soil.

5.2.7.2. **Engineering Requirement:** The system will support a top load of at least 25 lbs and withstand a lateral force exertion of at least 25 Newtons.

5.2.7.3. **Testing Method:** Demonstration

5.2.7.4. **Verification Process:** A top load of 25 lbs will be measured using a digital scale and then placed on the top of the system. After the weight has been placed, the system will still transmit data properly and be observed in either the serial monitor in the Arduino IDE or through the user interface as a .csv file. To prove the system can withstand a lateral force of 25 Newtons, using a digital scale, a lateral force will be applied to the system using a rope tied around the enclosure and then measured with the digital scale, the reading should be around 5 lbs on the digital display.
Pass Condition: The system will continue to properly send packets over the wireless communication network after the two force tests have taken place. If the packets are still being received by the receiver after both tests the system will pass.

5.2.7.5. **Testing Evidence:** There were two tests done in this video to prove the rigidity requirement. First a lateral force of 25 Newtons was exerted by pulling on the enclosure with a rope tied to a digital scale, 25 Newtons is equal to around 6 lbs which is less than the actual force that was exerted. Next a weight was placed on top of the enclosure which was around 30 lbs. After this we confirmed that the system still correctly functions by confirming data is still being sent over the wireless channel that is unique and different. Verified 5/8 Rigid_Structure_Verification.mp4

5.2.8. Weatherproof

5.2.8.1. **Project Partner Requirement:** The system must be able to last at least six months under normal weather conditions for the area.

5.2.8.2. **Engineering Requirement:** The probe will function normally after being submerged in water up to 12 inches and the top-lying system will function normally after being submerged in up to 3 inches of water.

5.2.8.3. **Testing Method:** Test

5.2.8.4. **Verification Process:** The enclosure along with the probe will be submerged in at least 12 inches of water. After the system has been submerged, data will be sent over the wireless channel and observed to see if it is still sending data reliably.
Pass Condition: The system will continue to work as intended and data will be observed at the receiver end either through the

arduino IDE or through the user interface to prove that the system continues to work after the water test.

5.2.8.5. **Testing Evidence:** This testing evidence consists of a video link where the system probe is dunked in a 5-gallon bucket which is approximately 13 inches deep. After it is dunked and held in the water, it is then confirmed that data is still being collected by the receiver thus proving it still functions normally after a 12 inch submersion. Verified 5/10 Weatherproof_Verification.MOV

5.3. References and File Links

5.3.1. References

There are no references pertaining to this section.

5.3.2. File Links

Picture of Modules	Built_Modules+Connections.jpg
Main PCB Schematic	Main_PCB_Schematic.png
Connected System	Assembled_System.jpg
SMT Connection Photo	SMT_Connection_Proof.jpg
Assembled but Unconnected System	Assembled_Unconnected_System.jpg
Python Script for Power consummation + solar cell input	Solar_sim.py
Verification Video for Comm Distance	Comm_Distance_Verification.mp4
Verification for Clear Instruction + Link to Form	Capstone Survey (Responses) Capstone Survey Clear_Instruction_Verification.MOV
Verification Video for Rigid Structure	Rigid_Structure_Verification.mp4
Verification Video for Data Display	Data_Display_Verification.mp4
Verification Video for Weatherproof	Weatherproof_Verification.MOV

5.4. Revision Table

3/12/23	Section Created - Jeremiah Goddard
5/7/23	Revised section headers to match the outline - Brennan Ventura
5/7/23	Added necessary information for universal constraints - Brennan
5/14/23	Added in verification evidence for missing requirements - Brennan
5/14/23	Fixed formatting and revised some section - Brennan

6. Project Closing

6.1. Future Recommendations

6.1.1. Technical Recommendations

Overall the implementation and design of the system went very smoothly and there was not a lot of disagreement over the various technical aspects of the aspects of the system during the design phase. However, after implementing the various components and building the different blocks there were many things that in hindsight could have been improved upon. One primary technical improvement that would benefit the system greatly would be either connecting to a premade LoRaWAN gateway or creating your own with a LoRa module and a microcontroller. Using a LoRaWAN gateway would help the UI end of the project as the live datastream from the sensors can be sent directly over the LoRaWAN network and accessed on the gateway website thus allowing the user to be able to view the sensor data from anywhere with internet access. Another change that would benefit the overall system would be a different temperature sensor. The current temperature sensor is an IC temperature sensor which while very accurate, may not be the best for the reading of soil temperature [1]. Using a thermocouple sensor such as a waterproof wire sensor kit, where the sensor can extend into the soil would thus provide a more accurate reading while also providing the additional benefit of weatherproofing. While the system reads a variety of different parameters from the soil, there are still many other important parameters that could be measured which affect the way in which plants grow. The chemical makeup of soil is very important when it comes to the way a plant grows. There are three elements nitrogen, phosphorus and potassium; which are crucial to plant growth so being able to measure those three elements in your own soil would provide massive benefits for growth. Implementing a sensor to measure these different essential nutrients in soil would be very beneficial for the end user. One of , if not the most important part of this system is the data the soil probe collects and knowing what to do with that data. In the final implementation of this system, the data is merely presented to the user in a .csv format with no extra processing or calculations. And while this can be okay for many gardeners who may be more experienced, those with less green thumbs would benefit from more information about the data they are receiving. A more flushed out user interface that provides the user with additional information about their soil in the form of trendlines, charts and even the opportunity to compare their soil parameter measurements with the ideal measurement would provide a much more enjoyable experience to the end user. Allowing the user to enter in the plant they are wanting to grow and then showing the ideal soil conditions for that plant would provide a great user experience and greatly improve the final system. Overall, all of these technical recommendations would improve the final system and provide a more challenging and rewarding project.

6.1.2. Global Impact Recommendations

Global impacts include the potential elimination of the use of culture based farming practices specifically those of indigenous peoples of colonized lands. The probe is a form of smart farming which can improve efficiency and help develop new methods of farming and tracking water usage and the effects that it has on crops. In this way smart farming bypasses knowledge handed down between generations, limiting its need and potentially eradicating it completely. Efforts should be made to preserve these practices through thorough documentation and the encouragement of its use wherever possible because preserving a culture has been proven to be beneficial to society in many ways [2]. Other possible impacts on a global scale include the over reliance on this technology which could prove to cause issues in the long term. This technology is new and still needs to be improved, imperfections in the system or any potential failures could leave anyone relying on this technology for food with no other knowledge of farming or gardening adversely affected. Users should be encouraged to only use the technology if they have a strong background in farming or to not have an overreliance on technology.

6.1.3. Teamwork Recommendations

While the project team got along very well and worked together to accomplish tasks and finish assignments before deadlines there is still room for improvement across each phase of the project. Just because people are working together towards a common goal it does not make that work productive or fruitful [3]. During the first phase while designing the system it's very important to keep in mind how difficult each block of the project will be and its importance to the overall system implementation. One very important recommendation is how the various blocks are divided amongst each team member. In the design phase, it may seem like certain blocks may pertain more to one person's set of skills but this block may take up a majority of the project. It is very important to split the workload of the individual blocks evenly between team members so one person is not left with a majority of the work. On the same note, how each of the blocks is designed is very important. Sometimes it makes more sense to combine blocks into one larger sub-system. For instance, the final system contains a wireless communication transceiver and receiver block which could have been combined into one wireless communication network or system block with a relatively similar amount of work. The design of the blocks is crucial to building and implementing the system later on. Another very important part of any capstone project is communication with the project partner. In many projects the project partner is there to help and guide the project. Sometimes it can be difficult to start the design of a block or how to build it, asking your project partner for advice is a very good place to start as they often have some type of experience in the project field and can guide the design into a direction that is beneficial for both parties. Finally, proper and frequent communication with a project team can lead to a better final system. There are many times where it can seem like there is nothing to work on for an individual, so communicating with the team and offering to help others or get a head start on other parts of the project will be a massive help later on when deadlines approach much quicker.

6.2. Project Artifact Summary with Links

6.2.1. References for Recommendations

[1]“TMP1075 Temperature Sensor With I 2 C and SMBus Interface in Industry Standard LM75 Form Factor and Pinout.” Accessed: May 08, 2023. [Online]. Available: https://www.ti.com/lit/ds/symlink/tmp1075.pdf?ts=1683507012198

[2]T. H. Tuan and S. Navrud, “Capturing the benefits of preserving cultural heritage,” Journal of Cultural Heritage, vol. 9, no. 3, pp. 326–337, Jul. 2008, doi: https://doi.org/10.1016/j.culher.2008.05.001.

[3]G. Sherwood and J. Barnsteiner, Quality and Safety in Nursing: A Competency Approach to Improving Outcomes. John Wiley & Sons, 2021. Accessed: May 08, 2023. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=KzxKEAAAQBAJ&oi=fnd&pg=PA131&dq=teamwork+and+collaboration&ots=YM21-uMo-y&sig=7bQCh2rjccbpwpDLm7aL4EYtRlk#v=onepage&q=teamwork%20and%20collaboration&f=false

6.2.2. Project Artifacts

Main PCB Layout	Main_Board_Layout.png
Main PCB Schematic and .pcb File	Main_PCB.zip
PCB Schematic and board for Soil	Moisture_PCB.zip

Moisture Sensor	
PCB Schematic and board for Soil Temperature Sensor	Temperature_PCB.zip
Arduino Wireless Receiver Software	LoRaSimpleGateway_Final.ino
Main Microcontroller Firmware	SOIL_SENSE.ino
System Block Diagram	Block_Diagram.drawio
User Manual	SoilSense User Manual

6.3. Presentation Materials

There are no presentation materials for this project besides the poster for the engineering expo and the Senior Capstone showcase website.. A link to the PDF version of the poster as well as a ppt version can be found below in the file links table along with a link to the showcase website.

Important Links	
Project Expo Poster PDF	SoilSense_ECE.25_Expo_Poster.pptx.pdf
Project Expo Poster .pptx	SoilSense_ECE.25_Expo_Poster.pptx
Senior Showcase Website	Senior_Showcase_Website