Agricultural Sensor Network

Emma Dacus
Grace Mackey
Blake Garcia
Garren Dutto

Section 1 – Overview

Table of Contents

Section 1 – Overview **2**
Table of Contents 2
1.1 Executive Summary 5
1.2 Team Contacts and Protocols 7
1.3 Gap Analysis 9
1.4 Timeline 10
1.5 References & File Links 11
1.6 Revision Table 12
Section 2 – Impacts & Risks **13**
2.1 Impact Statement 13
2.1.1 Introduction 13
2.1.2 Public Health, Safety, and Welfare Impacts 13
2.1.2.1 Electrocution 13
2.1.2.2 Degradation of Materials 13
2.1.3 Cultural and Social Impacts 14
2.1.3.1 Technology's Impact on Farming Practices 14
2.1.3.2 Reliance on Technology 14
2.1.4 Environmental Impacts 14
2.1.4.1 Effects of Flooding 14
2.1.4.2 Unreliable Data 15
2.1.5 Economic Factors 15
2.1.5.1 Economic Considerations 15
2.1.5.2 Equitable Mitigations 16
2.1.6 Conclusion 16
2.2 Risks 17
2.3 References & File Links 18
2.4 Revision Table 19
Section 3 – Top-Level Architecture **20**
3.1 Black Box & Block Diagrams 20
3.2 Block Descriptions 21
3.3 Interface Definitions 26
3.4 References & File Links 29
3.5 Revision Table 30
Section 4 – Block Validations **31**
4.1 Water Level Sensor 31
4.1.1 Description 31
4.1.2 Design 31

4.1.3 General Validation 33
4.1.4 Interface Validation 34
4.1.5 Verification Process 36
4.1.6 References & File Links 36
4.1.7 Revision Table 37
4.2 Application 38
4.2.1 Description 38
4.2.2 Design 38
4.2.3 General Validation 39
4.2.4 Interface Validation 39
4.2.5 Verification Plan 40
4.2.6 References & File Links 41
4.2.7 Revision Table 41
4.3 Radio Data Transfer System 42
4.3.1 Description 42
4.3.2 Design 42
4.3.3 General Validation 43
4.3.4 Interface Validation 44
4.3.5 Verification Plan 45
4.3.6 References & File Links 46
4.3.7 Revision Table 46
4.4 Wind Speed Sensor 47
4.4.1 Description 47
4.4.2 Design 47
4.4.3 General Validation 48
4.4.4 Interface Validation 50
4.4.5 Verification Plan 51
4.4.6 References & File Links 52
4.4.7 Revision Table 52
4.5 Humidity/Temperature Sensor 53
4.5.1 Description 53
4.5.2 Design 53
4.5.3 General Validation 55
4.5.4 Interface Validation 57
4.5.5 Verification Plan 60
4.5.6 References & File Links 62
4.5.7 Revision Table 62
4.6 Microcontroller 64
4.6.1 Description 64
4.6.2 Design 64
4.6.3 General Validation 66

4.6.4 Interface Validation 67
4.6.5 Verification Plan 74
4.6.6 References & File Links 76
4.6.7 Revision Table 77
4.7 Enclosure 78
4.7.1 Description 78
4.7.2 Design 78
4.7.3 General Validation 79
4.7.4 Interface Validation 80
4.7.5 Verification Plan 81
4.7.6 References & File Links 82
4.7.7 Revision Table 82
4.8 PWM Charge Controller 83
4.8.1 Description 83
4.8.2 Design 83
4.8.3 General Validation 85
4.8.4 Interface Validation 86
4.8.5 Verification Plan 90
4.8.6 References & File Links 92
4.8.7 Revision Table 93
4.9 Solar Panel 94
4.9.1 Description 94
4.9.2 Design 94
4.9.3 General Validation 96
4.9.4 Interface Validation 97
4.9.5 Verification Plan 99
4.9.6 References & File Links 100
4.9.7 Revision Table 101
Section 5 – System Verification Evidence 102
5.1 Universal Constraints 102
5.1.1 The system cannot contain a breadboard 103
5.2 Requirements 106
5.2.1 Sensor Requirement 106
5.2.2 Data Sending Range Requirement 111
5.2.3 Cost Requirement 113
5.2.4 Data Display 114
5.2.5 Data Sending Frequency 116
5.2.6 Documentation Quality 118
5.2.7 Ease of Assembly 119
5.2.8 Weatherproofing 120
5.3 References & File Links 122

5.4 Revision Table 124
Section 6 – File Closing **125**
6.1 Future Recommendations 125
6.1.1 Technical Recommendations 125
6.1.2 Global Impact Recommendations 126
6.1.3 Teamwork Recommendations 126
6.2 Project Artifact Summary 128
6.3 Presentation Materials 129
Project Showcase: Link 129

1.1 Executive Summary

The world's climate crisis disproportionately impacts developing countries and low-income populations within those communities at all levels of society; it can change the societal structure, community health, and individuals' economic situations [1]. Thailand is an example of a developing country dependent on the success of the agricultural sector, specifically rice, maize, and sugarcane, which have been estimated to have fallen by up to 3.6% to date and are projected to increase to a 15% decrease by 2050 [2]. The rural farmers dependent on the success of Thailand's agricultural industry make up over thirty percent of the total labor force, but only account for about ten percent of the gross domestic product, implying the average farmers are poor and vulnerable to the effects of climate change without much individual recourse to protect themselves, their families, and their communities from economic hardship if crop production continues to decline [3]. The purpose of this project is to develop an integrated sensor network to provide individual farmers with data about the conditions of their land to react quickly when their crops experience sudden harmful conditions - conditions that happen more frequently as the effects of climate change become more pronounced.

Bryan Hugill founded Raitong Organics to grow an active and social community of farmers who can implement sustainable rice farming practices in rural communities. The main challenges he faces include an aging population of farmers who have increasing difficulty overseeing all their farmland, which is spread out over non-contiguous plots. For example, an essential detail in the process of rice farming is flooding paddies with the correct amount of water between three centimeters and ten centimeters which can fluctuate if either too much rain falls on the paddy or if a bund cracks and water is released from the paddy [4]. Bryan wants to have a low-cost system capable of detecting these conditions and notifying the farmers where a problem has occurred and the severity of the issue allowing for a rapid response. He can also store the data for future analysis if necessary.

The sensor nodes included in the system developed by this project will measure four specific environmental factors: temperature, humidity, wind speed, and water level. They will each transmit the data periodically throughout an entire network of sensor nodes to a base station capable of displaying the data, the time it was measured, and the node which sent it. This project iteration is focused on creating the network topology and radio transmissions necessary for node-to-node communication through the non-contiguous plots of land and displaying multiple different nodes on the same user interface. A network has been fully implemented and the data is received by the host node, although the data display is still a work in progress to differentiate between different nodes.

1.2 Team Contacts and Protocols

This section consists of two tables focused on team communication and dynamics. The first table lists each student on the team with their contact information, role within the team, and their expected contributions to the project. The second table defines the expectations and protocols surrounding different aspects of the team dynamic and communication.

Table 1.2-1: Team Roles and Contributions

Team Member Name	**Contact Information**	**Role**	**Expected Contributions**
Emma Dacus	dacuse@oregonstate.edu	Treasurer	Web development
Grace Mackey	mackeyg@oregonstate.edu	Scribe	Power Management
Blake Garcia	garciabl@oregonstate.edu	Primary mentor contact	Embedded systems
Garren Dutto	duttog@oregonstate.edu	Researcher	Embedded systems

Table 1.2-2: Team Protocols and Standards

Topic	**Protocol**	**Standard**
Work Quality	Members will be assigned specific parts of the project to complete during weekly meetings. The assignment will include the purpose of the task, the scope of the expected work, and the completion date.	Work should be completed on time and to the best of the ability of the group member. If the task is too difficult, too time-consuming, or any unexpected situations arise delaying the completion of the work then the team member is expected to communicate their progress in the iMessage group chat before the expected date of completion.
Communication between Team Members	The main communication between group members outside of group meetings will take place within an iMessage group chat including all members of the project. All questions or concerns about each individual's work assignments are to take place within the group chat to keep information shared with each	A message should receive a reply within 24 hours. Communicate any issues a member encounters (personally and professionally) that may affect their ability to work effectively as soon as issues arise to avoid additional problems.

	group member.	
Communication between Team Members and the Project Partner	A WhatsApp group chat between all four team members and the Project Partner (Kendra Sharp) will be used for Oregon State University-related concerns. Blake Garcia will coordinate meeting times with the Project Partner and communication between the group as a whole and the Project Partner will be delivered by Blake.	Responses to the Project Partner will take place within 24 hours. Any communication from the Project Partner will be relayed to the entire group within 24 hours.
Communication between Team Members and the Project Partner	When available, the group will meet with the Project Partner (Kendra Sharp) in person. If unavailable then the meeting will be held via Zoom. The frequency of these meetings is subject to the Project Partner’s availability.	Meeting times will be coordinated by Blake directly to limit the confusion for the Project Partner.
Communication between Team Members and the Project Customer	Weekly emails including the progress made during the previous seven days will be completed in conjunction with the Agricultural Drone Team and sent to the Project Customer (Bryan Hugill) and the Project Partner (Kendra Sharp).	Blake will draft an update with input from the entire team and forward it no later than Friday at noon each week.
Communication between Team Members and the Project Customer	Meetings with the Project Customer (Bryan Hugill) will take place over Zoom at least once per term. Each meeting will be scheduled at least five days in advance of the meeting to ensure each member has access to participate.	The team will draft a list of progress, comments, and questions for the Project Customer (Bryan Hugill) before the meeting. Notes during the meeting will be taken by the Scribe (Grace Mackey) to refer to afterward.

1.3 Gap Analysis

The purpose of this project is to develop an integrated sensor network to provide individual farmers with data about the conditions of their land to react quickly when their crops experience sudden harmful conditions. The application of the system focuses specifically on farmers maintaining multiple plots of non-contiguous land which makes simultaneous supervision of each plot exceptionally more difficult and the landscape of the plots between the farmland makes direct point-to-point communication nearly impossible. Raitong Organics also intends to ease the burden of reacting to acute weather conditions on the elderly farmers within the population by alerting them early enough to reach out for help from the community. Collecting the data over time can also help guide farmers on different techniques they can implement and allows them to measure more factors of their crops beyond simply the yield of a harvest.

Completing the project requires the team to make a few assumptions surrounding the implementation and use of the system. Rural farmers in Thailand tend to have few economic resources on hand, so the final cost of the system has to include only the necessary components for operation. This required the exclusion of the soil moisture sensor and the wind direction sensor because the instruments were too expensive at the time of project design. The team also assumed the end user had access to a computer with a Universal Serial Bus port but had no relevant experience in operating one besides using simple programs with Graphical User Interfaces (“GUI”), so operating the system cannot require any programs more complex than a web browser. It must also have detailed instructions for initializing the system including a user manual.

The scale of the completed project will assume a farmer can measure the entirety of their farmland with sixteen individual sensor boxes. Documentation will provide details on how to scale up from sixteen sensors, but the implementation of this iteration of the project will focus more on the network topology and message transmission than on pushing the theoretical limits of the system. The constraining factor would be the hardware contained on the host node which would not have the capability of storing more than sixteen simultaneous sensor box readings at one time; if the cost of the system was unimportant then the software running the network could support up to two hundred and fifty-five sensor boxes.

This project is a continuation of a previous group’s implementation of the sensor box which mainly accomplished building a single box capable of taking measurements and displaying the data remotely on a website. The current iteration of the system is expanding upon that by supporting multiple sensor boxes on the same network and displaying them all at once. Each sensor box node has a transmission range of about one kilometer, although it can pass data through the network without direct communication with the host node. This range significantly decreases as the number of objects that get in between two radios, but the network can handle retransmissions and data relaying to ensure no data gets lost from any sensor box. Boxes will include their source of power generation (through a solar panel) and power storage (through a battery) and an enclosure to protect the electronics from both the weather and any insects or animals attempting to breach the box.

1.4 Timeline

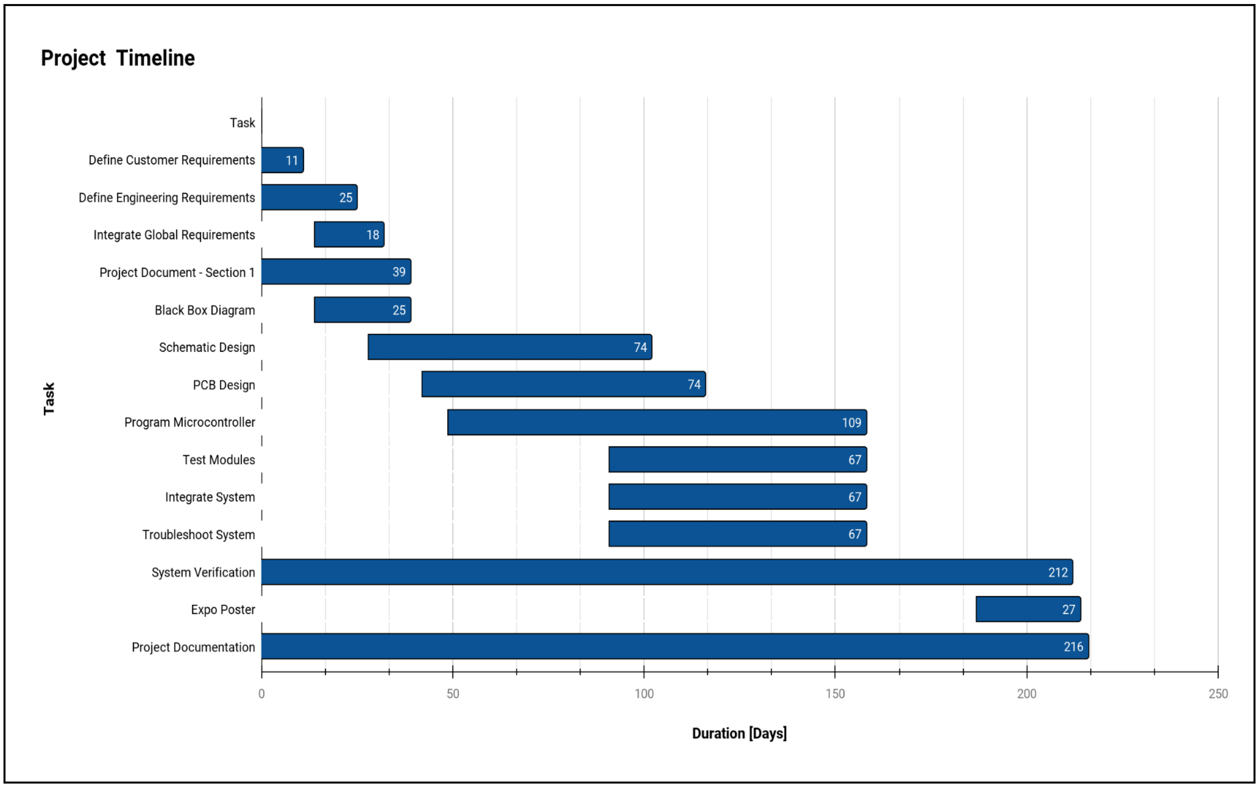


Figure 1.4-1: Project Timeline - Gantt Chart

1.5 References & File Links

[1] USGLC, “Climate change and the developing world: A disproportionate impact,” *USGLC*, 02-Mar-2021. [Online]. Available: https://www.usglc.org/blog/climate-change-and-the-developing-world-a-disproportionate-impact/. [Accessed: 07-Mar-2023].

[2] “Fao.org,” *Adapting to climate impacts in Thailand's agricultural sector – What to do first? | Integrating Agriculture in National Adaptation Plans (NAPs) | Продовольственная и сельскохозяйственная организация Объединенных Наций*. [Online]. Available: https://www.fao.org/in-action/naps/news-events/detail/ru/c/1118438/. [Accessed: 07-Mar-2023].

[3] W. Attavanich, “How Is Climate Change Affecting Thailand’s Agriculture? A Literature Review with Policy Update,” *FFTC Agricultural Policy Platform (FFTC-AP)*, Nov. 21, 2018. https://ap.fftc.org.tw/article/1359

[4] IRRI Technologies, “How to manage water,” *Water management - IRRI Rice Knowledge Bank*, n.d. [Online]. Available: http://www.knowledgebank.irri.org/step-by-step-production/growth/water-management#:~:text=Total%20seasonal%20water%20input%20to,for%20irrigated%20rice%20in%20Asia. [Accessed: 07-Mar-2023].

1.6 Revision Table

Date	Name	Revision Made
5/10/2023	Grace	Updated timeline image, finished section 1 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed the formatting in section 1
3/10/2023	Grace	Edited section 1 (grammar, spelling, etc.) and relabeled figures.
3/7/2023	Blake	Rewrote the Executive Summary to focus on the purpose of the project and to create a more abstract view of the technical details of the project.
3/7/2023	Blake	Reformatted the “Team Protocols” to fit within the “Topic / Protocol / Standard” table. The labeled table follows IEEE standards.
3/7/2023	Blake	Rewrote the Gap Analysis to align with the current state and scope of the project.
3/7/2023	Blake	Added citations in IEEE format.
11/18/2022	Grace	Updated the Executive Summary by including the current state of the project and also fixed grammatical errors.
10/31/2022	Emma	Switched the order of the table of contents and executive summary, rearranged the revision table, resized gap analysis, labeled figure, and table, and added numerical labels on subsections.
10/13/2022	Garren	Finished the rough draft of Gap Analysis and pasted a picture of the preliminary timeline.
10/13/2022	Grace	Added the first draft of the Executive Summary.
10/13/2022	Emma	Reformatted team contact info., and added potential team member roles, contributions, and protocols.
10/12/2022	Garren	Added rough Gap Analysis.
10/12/2022	Emma	Started the team contact/protocol section and reformatted the revision table.

Section 2 – Impacts & Risks

2.1 Impact Statement

2.1.1 Introduction

The purpose of this document is to evaluate the potential impacts of our design and the considerations we should make when completing our system. The main focuses of this evaluation are the possible impacts on the safety and health of stakeholders, the cultural and social impacts of farmers adopting our design, the economic factors going into our design, and the potential environmental impacts of our design. We want to provide a useful tool to allow farmers to increase their ability to maintain their farms, but we need to be aware of the possible repercussions of design flaws and barriers created by the adoption of our system.

2.1.2 Public Health, Safety, and Welfare Impacts

2.1.2.1 Electrocution

The main concern with many electrical projects is injury or death by electrocution. The sensor box uses a 50W solar panel for power, which sources 12 volts. According to a study conducted by medical professionals P. Mellen, V. Weedn, and G. Kao, only 30% of electrocutions are caused by "low-voltage" incidents, and of those incidents, only 40% were not caused by water, intoxication, or previous electric shock [1]. Using this data, the risk of electrocution with the sensor box is fairly low. This is compounded by the fact that almost all of the electrical circuitry will be enclosed, making it difficult to access any metal that may be carrying significant amounts of current.

2.1.2.2 Degradation of Materials

Another concern we have when it comes to the public health impacts of our project is making sure that none of the materials that we use in our system are prone to degradation from extended exposure to the elements. If any of our electronic parts were to deteriorate, it would not only affect the effectiveness and efficiency of our project and therefore Raitong Organic Farms, but it could also negatively impact the surrounding crops if toxins were to absorb into the soil and potentially spread through run-off. For example, electronic parts that have rust inhibitors on them may contain Chromium which in some forms, is capable of affecting plant growth in a multitude of ways. Chromium phytotoxicity is capable of inhibiting seed germination and degrading nutrient balance and antioxidant enzymes [6]. Therefore it's important to ensure that electronic parts that we choose to use don't contain harmful chemicals like Chromium that could potentially damage a large portion of crop yields if they degrade to protect the customer and ensure that their production can't be harmed and as a result, their way to economically support themselves and the recipients of their product.

2.1.3 Cultural and Social Impacts

2.1.3.1 Technology's Impact on Farming Practices

While technological advancements in farming cause largely positive impacts, such as an increase in production and less harm to the environment, there are also negative impacts to consider. According to the journal "Modernization of Agriculture vs Sustainable Agriculture" by Dariusz Kusz, the implementation of new agricultural systems that better organize farms has increased production and decreased employment levels. While these systems have mitigated burdensome tasks for farmers, the increase in production has decreased the value of the items produced. At first, the farmers see an increase in income but eventually, the value of their items will decrease, meaning the majority of the money made from these advancements goes to the consumers and not the farmers.

Another concern with the modernization of agriculture is the balance between profitability and sustainability not being observed. Corporations only being concerned with outputting as much product as fast as possible causes soil degradation and eventually infertility in the land. Not to mention the social stresses it places on farmers to put out more and more products only to receive the same amount of money as if they didn't modernize their farming system [7].

2.1.3.2 Reliance on Technology

Technology in agriculture has proven to be a double-edged sword in today's world. While the increase in production is good for a short time, in the long run, it will cause more harm than good unless the system is managed properly. The increased use of technology has caused a reliance on fossil fuels that are contributing to larger problems like climate change. Without using sustainable sources of energy, things like soil degradation and an increase in greenhouse gas emissions will permanently damage the environment [8]. The reliance on technology could also cause the loss of invaluable cultural practices if the system is not implemented with society's best interests in mind.

2.1.4 Environmental Impacts

2.1.4.1 Effects of Flooding

The overall purpose of the project focuses on measuring and collecting data about soil health to increase crop production with a specific use case of detecting flooded areas within a farm. If implemented correctly, our design should allow farmers to increase their crop production by responding to potentially harmful agricultural conditions within the soil. The article "Detection of flood and its impact on crops using NDVI - Corn case“ by Ranjay Shrestha investigates the impacts on crop production of flooding events by comparing the median weekly Normalized Difference Vegetation Index (NDVI) product of three different flooding events. One conclusion noted by the author focused on the impact of floods occurring during the growing season, specifically “unlike early season flood, when a flood occurs during the growing season it not only destroys the crop but also didn't give farmers an option to replant as it's too late in the growing season,” which could affect farmers using the sensor boxes developed as the result of our work by either detecting or not detecting the flooding on their farm [2].

2.1.4.2 Unreliable Data

Ideally, the project mitigates the impact of flooding, but two main risk factors could exacerbate the issue. The first would include a faulty design that fails to properly collect data and present it to the farmer - which should be mitigated by rigorous testing and the use of good design practices. The second risk factor would be faulty hardware within the sensors causing a misreading of the physical environment. This would likely give a false sense of security to the farmers using the malfunctioning device and risking the health of their crops, which in turn could affect both their livelihood and food security. This has been a noted problem within the development of Smart Homes and IoT (“Internet of Things”) applications. Dr. Munir authored an article titled “FailureSense: Detecting Sensor Failure Using Electrical Appliances in the Home” developing new techniques to detect the presence of errors within sensor readings. Most notably he comments on modern causes of errors when he states “Sensors just don't die, they experience failure in a variety of ways. For example, sensors installed on furniture are moved or covered and produce invalid data,” which also suggests the sensors in our system could be subject to unknown conditions causing unreliable data [3].

2.1.5 Economic Factors

2.1.5.1 Economic Considerations

Currently, the world is in the middle of a chip shortage, which is going to have impacts on which parts are available. In addition, it is not likely that the chip shortage will be over in the next year. Parts are still unavailable, and the supply chain is recovering slowly [4]. This may make some of the essential parts such as sensors or microcontroller components difficult or impossible to obtain within the time frame of the project.

In addition to the supply chain shortage, the project is meant to be buildable with parts that are available in Thailand. Even more careful consideration needs to be made when selecting parts so that they are not only available to us but also would be available to the project customer. It also needs to be buildable from the parts and schematic by the customer, meaning pre-built parts that are available are better than parts that we have to create on our own because we cannot assume that somebody with great technical knowledge is trying to recreate our design.

2.1.5.2 Equitable Mitigations

The customer for the project is a farm co-op in Thailand, so this means that the project should be as low-cost as possible as well. Using the cheapest available parts for each task is important to make it possible for a farm to produce a large number of sensor boxes for their operations. The overall sensor box should have a good return on its investment for the farmers by providing a product that will notify them of floods or unusual weather patterns for minimal cost. So, the economic impact of the project should be low so that it is affordable for farmers that would like to implement it in their fields. This is exacerbated by the chip shortage, as communication chips and microcontrollers are currently more expensive than they have been in the past, and the communication industry has been heavily impacted [5].

In addition to the cost of the actual materials, there is a financial impact associated with the flooding problems mentioned above. If the system malfunctions and stops reporting correct data when a field floods, the farmer will lose money because they will not get as much yield from their crops. Additionally, decisions could be made based on information reported from the sensor box that has a large impact on the productivity of the farm, and if the data is faulty, those decisions could instead have a strong negative impact and end up costing the farm money.

2.1.6 Conclusion

Some key takeaways from our research into the impacts of our project have allowed us to fully understand the potential risks of our system and how they could impact both ourselves and our customer. This has also allowed us to be able to prepare for any issues we may run into preemptively to decrease the influence of these events on our progress throughout the year. A large portion of our impacts such as our economic impacts and material degradation can be prevented by conducting research beforehand to avoid any issues. For example, economic impacts can be mitigated by double-checking part availability and pricing to ensure we don’t go over budget and will get the parts we need in time, and as long as we look into the composition of the materials that we use, we shouldn’t need to worry about electronic parts degrading and ruining crop production.

The environmental impacts can be countered simply by testing our prototype and ensuring that all data is being transferred to the user interface correctly and efficiently so we can present a successful product to the customer. Electrocution can be avoided by following safety protocols while working on our project and being careful to disconnect our electronic parts from any power source before working on them. When considering the cultural and social impacts of including technology in agriculture, it’s important to create a system that considers economic, social, and ecological impacts. This can be achieved by creating a sustainable system that operates within social standards and will have a positive impact on the community’s economy.

2.2 Risks

Table 2.2-1: Risk Assessment and Action Plans

ID	Description	Category	Probability	Impact	Indicator	Action Plan
R1	Team member stabbing themselves w/ soil sensor	Safety	H	L	Blood and/or pain	Rubber stops for ends of sensor
R2	Inability to get parts	Schedule/ Technical/ Organizational	H	H	No materials	Order/ Research availability ahead of time
R3	Electrocution with bare wires in power supply	Safety	L	M	Pain and/or death	Crimp power supply wires to cover the metal parts
R4	Scheduling time to work w/ team	Schedule	H	M	Miscommunication	- Try to make weekends available - Take good meeting notes so no one misses anything
R5	Degrading materials	Public health/ Environmental	L	H	Unexpected data results	Researching materials & parts extensively
R6	Weather-proof	Technical/ Environmental	L	H	Unexpected data results	Researching materials & parts extensively
R7	Misreading info	Technical/ Environmental	L	H	Unexpected data results	FAQs w/ customer or project partner to ensure the sensors are working properly
R8	LiPo battery catches fire	Safety	L	H	Fire	- If outside, safely kick fire to area away from burnable materials - Locate fire extinguisher and call 911 - Use extinguisher to prevent surrounding materials from burning
R9	Solar panel is damaged	Technical	L	H	Discoloration or cracking	- Replace solar panel

2.3 References & File Links

[1] P. F. Mellen, V. W. Weedn, and G. Kao, “Electrocution: A Review of 155 Cases with Emphasis on Human Factors”, *Journal of Forensic Sciences*, JFSCA, Vol. 37, No. 4, pp. 1016 - 1022, 1992.

[2] R. Shrestha, L. Di, G. Yu, Y. Shao, L. Kang, and B. Zhang, “Detection of flood and its impact on crops using NDVI - Corn Case,” *2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2013.

[3] S. Munir and J. A. Stankovic, “FailureSense: Detecting sensor failure using electrical appliances in the home,” *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, 2014.

[4] J. Voas, N. Kshetri, and J. F. DeFranco, "Scarcity and Global Insecurity: The Semiconductor Shortage," in *IT Professional*, vol. 23, no. 5, pp. 78-82, 1 Sept.-Oct. 2021, doi: 10.1109/MITP.2021.3105248.

[5] Wassen Mohammad, Adel Elomri, Laoucine Kerbache, “The Global Semiconductor Chip Shortage: Causes, Implications, and Potential Remedies”, IFAC-PapersOnLine, Volume 55, Issue 10, Pages 476-483, 2022, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.09.439.

[6] Panda, S. K., & Choudhury, S. “Chromium stress in plants”. *Brazilian journal of plant physiology*, *17*, 95-102. 2005. https://doi.org/10.1590/S1677-04202005000100008.

[7] D. Kusz, “Modernization of Agriculture vs Sustainable Agriculture,” Vol. 14, no. 1, pp. 171–177, 2014.

[8] C. J. Cleveland, “Resource degradation, technical change, and the productivity of energy use in U.S. agriculture,” *Ecological Economics*, vol. 13, no. 3, pp. 185–201, 1995, doi: https://doi.org/10.1016/0921-8009(95)00005-T.

2.4 Revision Table

Date	**Name**	**Revision Made**
5/10/2023	Grace	Finished section 2 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed the formatting in section 2
3/10/2023	Grace	Edited section 2 (grammar, spelling, etc.)
3/8/2023	Emma	Fixed heading for section 1 so they appeared correctly in the table of contents
3/6/2023	Grace	Added our risk assessment statement
11/4/2022	Grace	Wrote cultural and social impacts section
11/4/2022	Emma	Wrote public health, safety, and welfare sections.
11/4/2022	Blake	Edited environment impacts section
11/4/2022	Garren	Wrote economic impacts section
10/31/2022	Blake	Created the template for this document and wrote the section on environmental impacts
10/31/2022	Garren	Created revision table
10/31/2022	Emma	Filled in risk assessment table

Section 3 – Top-Level Architecture

3.1 Black Box & Block Diagrams

The following is a black box diagram of our system that depicts the general system interfaces.

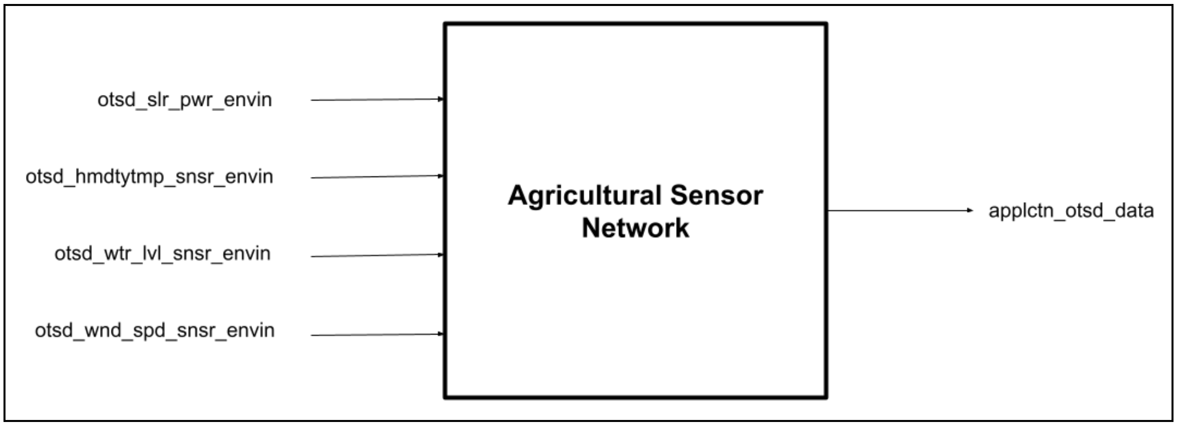


Figure 3.1-1: Black box diagram of the system.

The figure below is a block diagram of the entire system which is color-coded in accordance with the blocks that each team member was in charge of implementing.

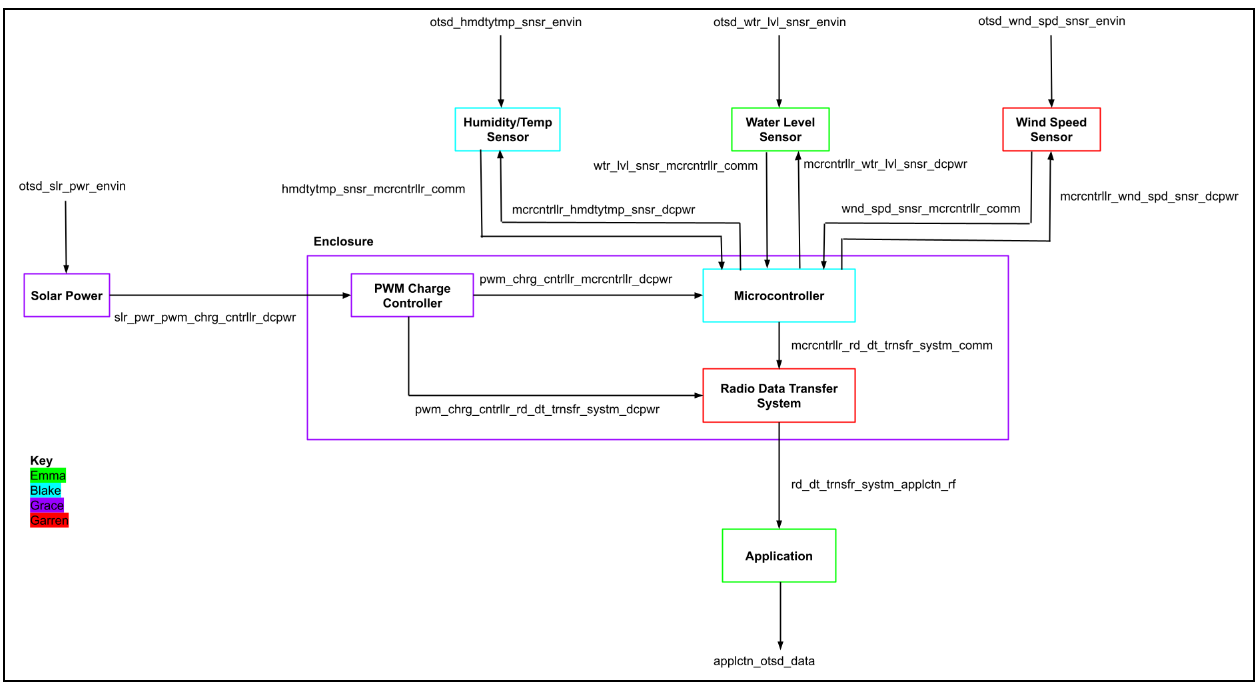


Figure 3.1-2: Block diagram of the entire system.

3.2 Block Descriptions

Table 3.2-1: Block Descriptions

Name	Description
User Interface (GUI) Champion: Blake Garcia and Emma Dacus	The GUI will take in information that was transmitted via radio communication from a microcontroller. This information includes environmental measurements from a temperature/humidity sensor, water level sensor, and wind speed sensor. This GUI will then display the data in a graphical format at 30-minute intervals in order to provide the farmers with accurate information they can use to determine agricultural operations. Our team decided that it would be the best option to write the program in C# rather than the previous team's UI which was a combination of PHP, Processing, and C++ as it can handle and store the data much easier. We also want this data to be able to be sent over a distance of approximately 1 km in the presence of obstructions as the farms in Thailand are non-contiguous. This can all be tested by setting up test code that provides the UI with dummy variables for agricultural data and illustrates that the website can display this information while each of the radio nodes are separated by 1 km and are not in line-of-sight. The ultimate goal of this project would be to have a network of sensor units spread throughout the entire farm that would be able to transfer and display all of their data, but for simplicity's sake, our team's goal is to figure out how to communicate this data through 2 or 3 nodes.

Humidity/Temp Sensor Champion: Blake Garcia	This block contributes to the satisfaction of the “Sensor Requirements” project requirement. The customer requested a system capable of measuring soil moisture, temperature, humidity, wind speed, and water level which was translated into an engineering requirement of measuring soil moisture up to 0.7 meters cubed of water per cubic meter of soil, temperature ranges of zero to forty-five degrees celsius, relative humidity from zero to one-hundred percent, wind speeds up to seventy-five kilometers per hour, and water levels up to twelve inches. Upon completion of this block, the microcontroller will receive data containing the air humidity and temperature readings via a Serial Peripheral Interface. The configuration used in this block operates a Bosch BME280 Digital humidity, pressure, and temperature sensor taking only temperature and humidity readings once per second and relaying them to the microcontroller via a Serial Peripheral Interface with data speeds of up to 2 million bits per second [1]. The sensor uses minimal power which does not directly satisfy any requirements for the system but was a consideration from the project partner - specifically, it needs an input voltage between three and five volts with a maximum current draw below 2 milliamperes. The datasheet also rates the temperature ranges from -40°C - 85°C but the realistic temperatures in Bangkok, Thailand have not dropped below or exceeded the requirement range in 2022 easing the extremes for which the system must operate [2]. Instead, the focus shifted to the humidity which can only satisfy the requirement between the temperature range of 0°C - 60°C [1]. The testing process for the block includes subjecting it to the extreme environmental conditions expected for the interface. This requires simulating relative humidity levels of zero percent and one hundred percent and temperatures of zero degrees Celsius and forty-five degrees Celsius. The SPI interface will use a visual inspection for the protocol (four-wire connection) and an oscilloscope to get the clock frequency of the data transmissions. Finally, the input DC power interface can implement two measurement tools (a controlled DC power supply and an ammeter) to ensure the device operates at both three and five volts with a nominal current draw of 1.2 mA and never exceeding 2 mA. If the block can successfully measure the environment, transmit it to the microcontroller, and operate under these input power conditions then it will meet the necessary interfaces to the other connected blocks.

Microcontroller Champion: Blake Garcia	This block facilitates the accomplishment of the Sensor, Data Display, and Data Sending Frequency requirements by powering and communicating with the temperature/humidity sensor, the wind speed sensor, and the water level sensor and then packaging the data every thirty minutes to broadcast via the Radio Data Transfer block. The customer requested a system capable of taking these measurements and sending them throughout a network which would be impossible without some control unit - in this case, the Microcontroller. It has three main categories of interfaces (with eight total): input power, output power, and peripheral communication. The input power interface includes the input from the power charge controller which will connect to the block via the USB port. It has an external regulator so the charge controller can provide close to five volts - although the Arduino can operate slightly below. Most of the input current is used to power the sensors instead of the microcontroller itself, but the current still travels through the block. The output power interfaces include three different DC values, although all of them should have the same vmin / vmax range and they just draw three different currents. The current draw is dependent on the sensor type and design, but the total peak draw is about 55 mA. As long as the supply pin is the 5V pin instead of an I/O pin on the microcontroller, then the difference between the sensors operating at nominal current draw and peak current draw is negligible. Finally, the block has to communicate with the sensors and the radio data transmitter. It uses two different communication interfaces to work with the sensors: analog sampling and a Serial Peripheral Interface connection. The analog sampling is used because the sensors will provide the data in the form of a range (i.e. the water level sensor might output 5 volts when fully submerged, 2.5 volts when half submerged, and no voltage when dry) and will operate entirely within the microcontroller itself whereas the SPI connection to the temperature and humidity sensor will be a two-way communication. The data rate for the Serial Peripheral Interface was chosen to match the radio data transfer interface so that the only code required to change between them is changing the logic levels of the Slave Select Pin. The radio data transmitter also uses a SPI connection, which will not be a problem because the transmission times between the sensor and the radio data transmitter are negligible.

Water Level Sensor Champion: Emma Dacus	This block will take in data from the environment in terms of how much water the sensor is submerged in. The data will then be displayed in terms of centimeters on a user interface (GUI) on a central computer once it's sent through a microcontroller and over a radio frequency. This sensor is able to measure the level of the water in which it's submerged through the compression of the sensor's envelope according to the liquid's hydrostatic pressure. The variable resistance that is output by the sensor is inversely proportional to the depth of the fluid. The maximum amount of resistance the sensor can output is approximately 2250 ohms when the liquid is anywhere between 0 and 1 inch deep, while the minimum amount of resistance the sensor can output is approximately 400 ohms [2]. Therefore, due to the sensor intaking 5V from the microcontroller, the maximum current that the sensor can consume is about 12.5 mA, the minimum being 2.22 mA. The customer needs the sensor to be able to send an alert to the GUI when the water on their farm reaches a certain level. While the sensor is built to be 14.1 inches, the length of the sensor that is active to measure fluid is 12.6 inches although my group only needs 12 inches. This is a large reason why we chose to utilize this sensor in particular as well as the fact that it was the most affordable option due to the fact that in order to be able to test our sensor in the field in Thailand, we want to have at least two complete systems functioning and able to communicate with each other. In order to validate that this particular block is able to work, I will need to connect the sensor to an Arduino and immerse the sensor in water, ensuring that the sensor is able to correctly measure the depth of the water according to the resistance it measures. Using a DMM, I can measure the input and output current and voltages, ensuring that these values are also all correct, fully validating the water level sensor block.
Solar Power Champion: Grace Mackey	The solar panel is connected to a PWM charge controller that is used to charge a battery and run a single sensor station. The panel will send power through the charge controller, which will in turn power the sensor station and charge a battery during the day. The battery will serve as the power supply at night. The panel has a rated power of 10 watts (W), a maximum voltage of 18 volts (V), a nominal current of 0.56 amps (A), and a minimum voltage of 6V. The sensor box and power charge controller require a minimum of 5V to work, this makes this solar panel ideal for the sensor box. The panel's dimensions are 13.8 inches (in) (length) by 8.6 in (width) with a thickness of 1.3 in. The size is ideal because the customer wanted something small and on the cheaper side so people will not want to steal the panel. The customer plans to mount the panel on a pole secured in a cement-filled bucket. This panel (Eco-worthy 10W 12V Solar Panel) is available in a kit with a charge controller for $33.99 on Amazon.

Enclosure Champion: Grace Mackey	The enclosure is a sealed container that is weatherproof and animal-proof. The container is 4.7 inches [in] tall, with a length of 10in, and a width of 7.9in. This is ideal for our needs since the tallest item, the battery, is 4.02in tall and 5.94in long. The enclosure also has twelve holes, three on each side. This will allow us to place the sensors that require environmental inputs outside the enclosure and the rubber seals will maintain the enclosure's weatherproofing. The box is made of silicon plastic and has an ingress protection (IP) rating of 65. IP65 is the highest level of dust protection and indicates how waterproof the box is. In this case, the box could withstand water jets hitting it from all sides and remain waterproof. While this only applies to low-pressure water jets, this equates to rain and flooding conditions making it perfect for our needs. Cost is a major concern for our project customer, so we needed to find a premade option that was easily accessible in Thailand and cheap to replicate if necessary. This box can be found on Amazon for $22.99 and it meets all of our projects and customers' needs.
Wind Speed Sensor Champion: Garren Dutto	This block is a sensor that is meant to measure wind speed, with an effective range of 0-75 kilometers per hour. This data can then be fed back to a microcontroller, which processes the sensor readings and then can send them via radio communications to a central computer for storage. As such, the sensor needs to be able to produce an output that can be interpreted by the microcontroller. The methodology that the wind sensor uses does not matter for the purposes of the project, as long as it is able to be interpreted by the microcontroller and also is able to be used in outdoor conditions. To do this, the sensor that was selected was the Rev C Wind Speed Sensor made by Modern Device. This sensor was selected because of its cost and accuracy, making it the best choice within the constraints of this project.
Radio Data Transfer System Champion: Garren Dutto	This block is a radio module that needs to be able to send data from one point to another point up to 1 km away. The data that is being transmitted will consist of sensor readings, address requests, and error messages. As such, the module should be able to receive data from a microcontroller, and then send that data to another radio module up to 1 km away, which should accurately pass that data into another microcontroller. To do this, the radio module that was chosen is the RFM95W transceiver breakout board from Adafruit, along with a 900 MHz antenna. The range of this is advertised to be up to 2 km when using a long wire antenna, and up to 20 km line-of-sight using directional antennas, which is much more than the project would need, and would certainly be able to accommodate some line-of-sight obstructions. In addition, it is operated through a SPI interface, which can be handled with libraries by the most common microcontrollers.

PWM Charge Controller Champion: Grace Mackey	The power charge controller (PCC) will manage the DC input from the solar panel and ensure the panel does not overcharge the backup battery or send too much power to the main system. The PCC is connected to the microcontroller and the radio data transfer system. It can take in a maximum voltage of 18 volts [V], a peak current of 10 amps [A], a nominal current of 0.56A, and a minimum voltage of 6V. This PCC arrived in the same Eco-worthy 10W 12V Solar Panel kit and it can be found for $33.99 on Amazon.

3.3 Interface Definitions

The following table contains all of the interfaces within the system as well as each of their properties.

Table 3.3-1: Complete system interface list

Name	Properties
otsd_hmdtytmp_snsr_envin	• **Humidity:** 95% (Max) • **Humidity:** 25% (Min) • **Temperature (Absolute):** 0°C (Min) • **Temperature (Absolute):** 45°C (Max)
otsd_wtr_lvl_snsr_envin	• **Other:** Water level coincides with a variable resistance • **Water:** Sensor immersed in 12 inches • **Water:** Sensor immersed in 0 inches
otsd_slr_pwr_envin	• **Light:** Can function in partly cloudy conditions (4-hour min.) • **Light:** Can function in sunset conditions (voltage min. met) • **Light:** Can function in cloudy conditions (4-hour min.)

	● **Light:** Can function in full sun (4-hour min.)
otsd_wnd_spd_snsr_envin	● **Other:** Maximum Wind Speed: 75 kmph ● **Other:** Minimum Wind Speed: 0 kmph ● **Temperature (Absolute):** 0-45 degrees C
rd_dt_trnsfr_systm_applctn_rf	● **Datarate:** 30 min ● **Messages:** Weather data from each of the sensors (Celsius, km/h, cm, etc.) ● **Other:** Written in C#
hmdtytmp_snsr_mcrcntrllr_comm	● **Datarate:** Serial Clock Speed: 1 MHz ● **Protocol:** 4-Wire (SS, SCK, MISO, MOSI) ● **Protocol:** Serial Peripheral Interface
mcrcntrllr_hmdtytmp_snsr_dcpwr	● **Inominal:** 1.2 mA ● **Ipeak:** 2 mA ● **Vmax:** 5 V ● **Vmin:** 3.3 V
mcrcntrllr_wtr_lvl_snsr_dcpwr	● **Inominal:** ~2.94 mA ● **Ipeak:** 4.2 mA ● **Vmax:** 5V

	• **Vmin:** 3.3V
mcrcntrllr_wnd_spd_snsr_dcpwr	• **Inominal:** 25mA • **Ipeak:** 40mA • **Vmax:** 5V • **Vmin:** 4V
mcrcntrllr_rd_dt_trnsfr_systm_comm	• **Datarate:** Serial Clock Speed: 1 MHz • **Protocol:** 4-Wire (SS, SCK, MISO, MOSI) • **Protocol:** Serial Peripheral Interface
wtr_lvl_snsr_mcrcntrllr_comm	• **Other:** Sampling Rate: 15 seconds • **Vmax:** 5V • **Vmin:** 0V
slr_pwr_pwm_chrg_cntrllr_dcpwr	• **Inominal:** 0.56A • **Ipeak:** 10A • **Other:** 10W (Max Power) • **Vmax:** 18V • **Vmin:** 6V
wnd_spd_snsr_mcrcntrllr_comm	• **Other:** Sampling Rate: 15 seconds • **Vmax:** 5V • **Vmin:** 0.5V

rd_dt_trnsfr_systm_usr_ntrfc_wbpg_rf	• **Other:** Range: 1 km • **Other:** Frequency: 923 MHz • **Protocol:** LoRa
pwm_chrg_cntrllr_mcrcntrllr_dcpwr	• **Inominal:** 80 mA • **Ipeak:** 150 mA • **Vmax:** 5V • **Vmin:** 4.7V
pwm_chrg_cntrllr_rd_dt_trnsfr_systm_dcpwr	• **Inominal:** 30mA • **Ipeak:** 120mA • **Vmax:** 5V • **Vmin:** 3V

3.4 References & File Links

[1] Bosch, “Combined humidity and pressure sensor”, BME280 datasheet, Jan. 2022. [Online]. Available: https://cdn-learn.adafruit.com/assets/assets/000/115/588/original/bst-bme280-ds002.pdf?1664822559. [Accessed: 10-Feb-2023].

[2] W. Underground, “Bangkok, Thailand weather history,” Weather Underground, 2022. [Online]. Available: https://www.wunderground.com/history/monthly/th/bangkok/VTBD/date/2022-1. [Accessed: 10-Feb-2023].

3.5 Revision Table

Date	Name	Revision Made
5/11/2023	Garren	Moved references to section 3.4, added link to BME280 Datasheet reference
5/10/2023	Grace	Finished section 3 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 3
3/13/2023	Emma	Added the block descriptions table
3/10/2023	Emma	Added black box diagram, block diagram, and interface definitions
3/10/2023	Grace	Added revision table to section 3

Section 4 – Block Validations

4.1 Water Level Sensor

4.1.1 Description

This block will take in data from the environment in terms of how much water the sensor is submerged in. The data will then be displayed in terms of centimeters on a user interface (GUI) on a central computer once it's sent through a microcontroller and over a radio frequency. This sensor can measure the level of the water in which it's submerged through the compression of the sensor's envelope according to the liquid's hydrostatic pressure [1]. The variable resistance that is output by the sensor is inversely proportional to the depth of the fluid. The maximum amount of resistance the sensor can output is approximately 2250 ohms when the liquid is anywhere between 0 and 1 inch deep, while the minimum amount of resistance the sensor can output is approximately 400 ohms [2]. Therefore, due to the sensor intaking 5V from the microcontroller, the maximum current that the sensor can consume is about 12.5 mA, the minimum being 2.22 mA.

The customer needs the sensor to be able to send an alert on the GUI when the water on his farm reaches a certain level. While the sensor is built to be 14.1 inches, the length of the sensor that is active to measure fluid is 12.6 inches although my group only needs 12 inches. This is a large reason why we chose to utilize this sensor in particular as well as the fact that it was the most affordable option because to be able to test our sensor in the field in Thailand, we want to have at least two complete systems functioning and able to communicate with each other. To validate that this particular block can work, I will need to connect the sensor to an Arduino and immerse the sensor in water, ensuring that the sensor can correctly measure the depth of the water according to the resistance it measures. Using a DMM, I can measure the input and output current and voltages, ensuring that these values are also all correct, fully validating the water level sensor block.

4.1.2 Design

This block is connected to the microcontroller we use in our system to both receive power (mcrcntrllr_wtr_lvl_snsr_dcpwr) as well as send data regarding the current water levels of the environment around the sensor (wtr_lvl_snsr_mcrcntrllr_comm).

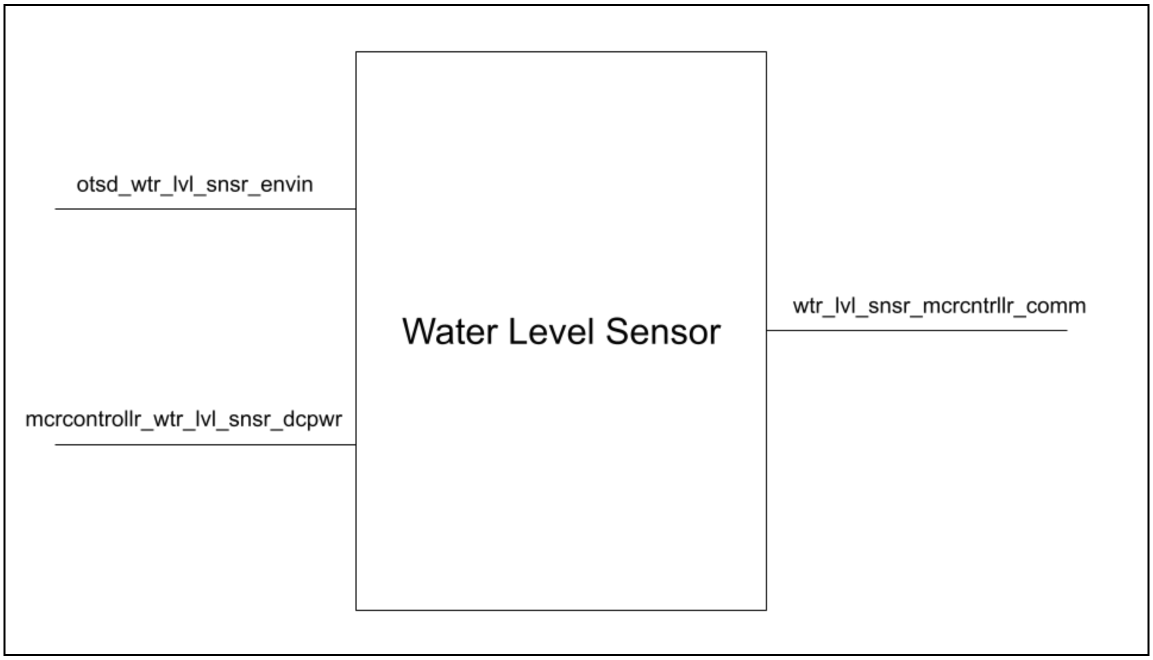


Figure 4.1-1: Black box diagram of the water level sensor and its interfaces.

To power the sensor, it is advised to connect pin 2 on the sensor to the ground pin on an Arduino, pin 3 to a 560-ohm resistor in series, and then to the VCC (5V) pin on an Arduino, then connecting the sensor and resistor via an ADC pin (A0) [1].

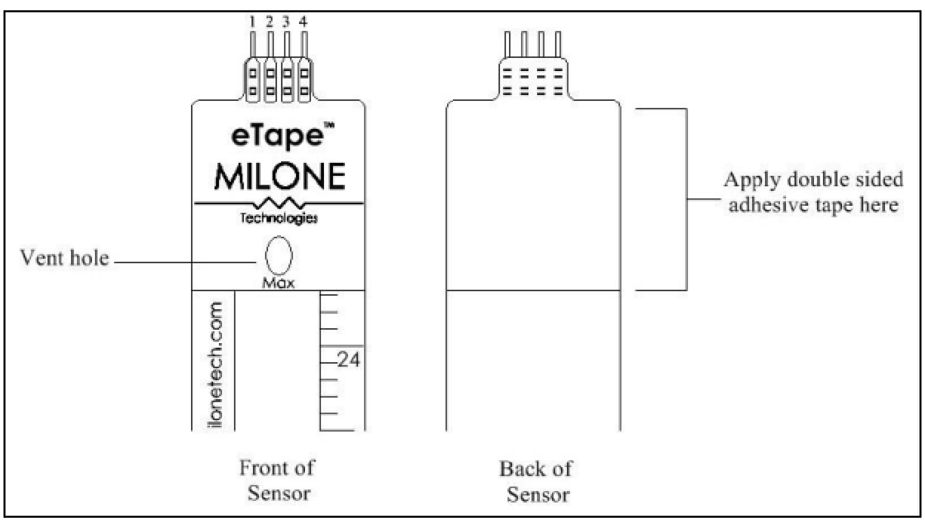


Figure 4.1-2: Water sensor pin assignments [2].

The amount of resistance that is output by the water level sensor is variable and coincides with how much compression is placed upon the sensor's envelope by the hydrostatic pressure of the fluid that it's submerged in. This can vary anywhere from 400 ohms, at the deepest (12 inches), to 2250 ohms at the most shallow measurement. There is also a window of error of approximately ± 10%.

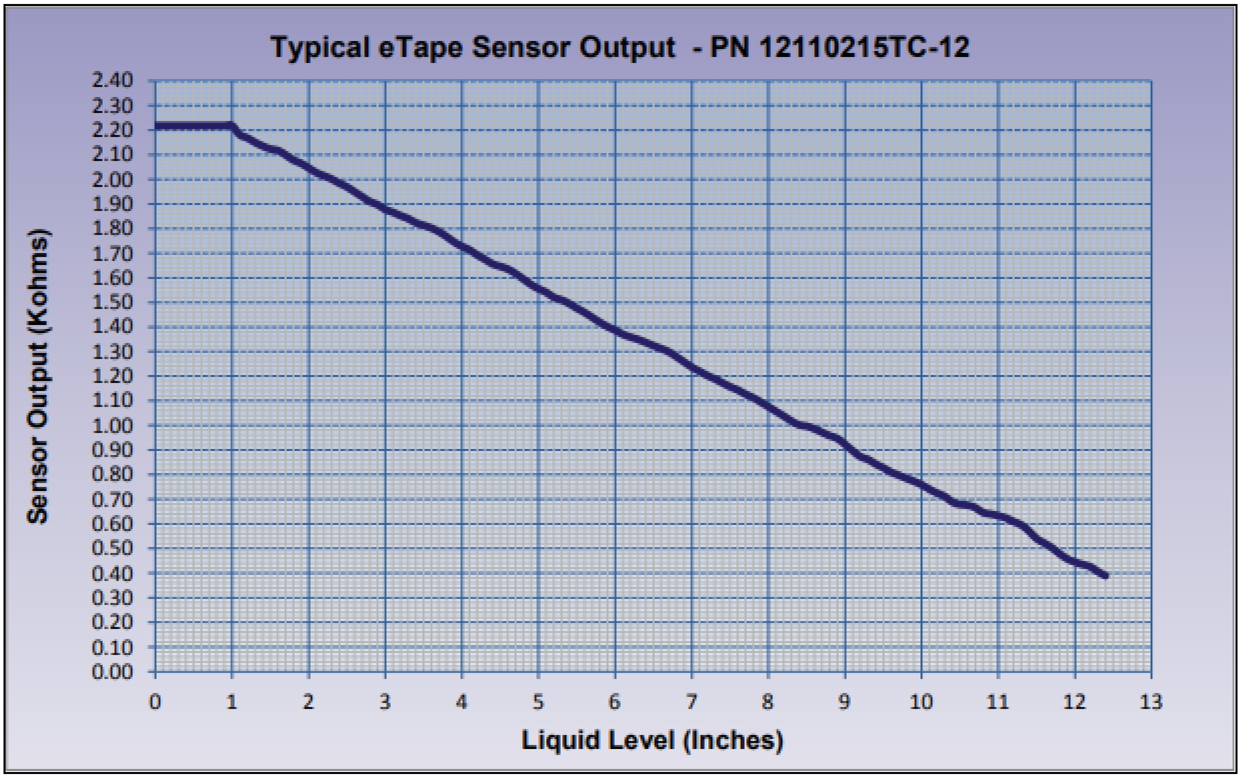


Figure 4.1-3: Variable resistance measured vs. the amount of water (inches) the sensor is submerged in [2].

4.1.3 General Validation

I chose this design because due to the number of sensors that we have in the entire system as well as the easiest way to send the data, we determined that a microcontroller would be the most straightforward way to connect the sensor to the rest of the system. Our customer requested that when the water reaches a predetermined height (in centimeters), an alert will be sent through the system that the area of the field in which the system is implemented is at risk of flooding and needs to be addressed. While the sensor measures fluid depth in inches according to the hydrostatic pressure on the sensor's envelope, this will be converted to centimeters in the programming process and as a result, displayed in centimeters on the GUI. The output resistance is inversely proportional to the depth of the liquid the sensor is in, so at its deepest, the resistance should be measured at 400 ohms, and at its most shallow, the resistance should measure at 2250 ohms [2]. This causes the output current to fluctuate between 2.2 mA and 12.5 mA as the voltage remains at a constant 5V.

We chose this specific sensor because we decided that 12 inches is a large enough measurement to use within the scope of the requirements we were provided by the customer and this sensor allows us to measure 12.6 inches and therefore covers the scope of the project [1]. This was the cheapest sensor option that was able to measure as deep as we needed to be although it was still quite expensive ($39.95 ea.) as we needed two to build our prototype network that, if we are successful, will be able to be field tested in Thailand [1].

One alternative solution that we could've implemented instead was buying several smaller sensors and connecting them to fulfill the length we needed, or building our own from scratch as buying components for a sensor is cheaper than buying one prebuilt. Unfortunately, this isn't the preferred option and we didn't end up choosing it due to our customer wanting our entire system to be able to be built quickly by non-engineers, primarily farmers who have no engineering prior experience.

4.1.4 Interface Validation

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

Table 4.1-1: otsd_wtr_lvl_snsr_envin: Input

Water: Sensor immersed in 0-12 inches	This is how the sensor determines the level of the water around it.	When enveloped in fluid, the envelope is compressed corresponding to a change in resistance (the lower the liquid level, the higher the output resistance, and vice versa)
Other: Water level coincides with a variable resistance	This water level was chosen based on the expected needs of the system overall.	For the 12-inch PN-12110215TC-24, the active length of the 14.2" sensor is 12.6" and exceeds the length that we need

This interface is potentially the most important for this block. The main function of the water level sensor is that it's able to measure the level of water the sensor provides the system with an alert when there is flooding in the agriculture fields the system is implemented in. This is important as it's one of the main functions of the entire system. How it does this is through variable resistance based on how much hydrostatic pressure from the fluid it's immersed in is measured by the sensor's envelope. Therefore, to validate and ensure that this sensor is working properly, the measurements that it makes must be accurate

Table 4.1-2: mcrcntrllr_wtr_lvl_snsr_dcpwr: Input

Ipeak: 12.5mA	This is the max amount of current that the sensor can run on, however, we don't expect that this will be necessary as it's not compatible with the rest of the system.	For the 12-inch PN-12110215TC-24, ● Max R ~ 2250 ohms ● Min R ~ 400 ohms ● Maintains enough current and voltage to sustain sensor and system
Vmax: 5V	This is the max amount of voltage that the sensor can have input, however, we don't expect to ever reach this level as the rest of the system doesn't need this much.	For the 12-inch PN-12110215TC-24, ● Max voltage it can intake is 10V > 6V
Vmin: 3.3V	This is the minimum voltage that the sensor can run at, however, we don't expect to use this value as it's too low for the rest of our system.	For the 12-inch PN-12110215TC-24, ● Min voltage it can intake is 3.3V < 4V

This interface is important because it impacts the rest of the blocks in the system as well. The peak current is important since the water level sensor in addition to the wind speed, temperature, and humidity sensors all feed into the microcontroller which can only supply a certain amount of current. Luckily all of the sensors require a small amount of current so it should not be an issue. The voltage requirement is important as all of the sensors also draw around the same amount of voltage. This also ensures that the power charge controller can provide enough power to the system through the use of our solar panel.

Table 4.1-3: wtr_lvl_snsr_mcrcntrllr_comm: Output

Vmax: 5V	This is the max amount of voltage that the sensor can output, however, we don't expect to ever reach this level as the rest of the system doesn't need this much.	For the 12-inch PN-12110215TC-24, ● Max voltage it can intake is 10V > 6V
Vmin: 3.3V	This is the minimum voltage that the sensor can run at, however, we don't expect to use this value as it's too low for the rest of our system.	For the 12-inch PN-12110215TC-24, ● Min voltage it can intake is 3.3V < 4V

Just like the input voltages, the output voltages are important parts of this interface because it's important to know how much power needs to be supplied to not only the water level sensor, but also the other environmental sensors, microcontrollers, and radio modules.

4.1.5 Verification Process

1. Connect the sensor to an Arduino Nano we're using in the system following what is recommended by adafruit (using a 560-ohm resistor, connected to the 5V pin)
2. Program the Arduino with code provided by adafruit to read data from the sensor
3. Submerge the sensor in water
4. Ensure the sensor is providing accurate readings (resistance – according to water depth)
5. Measure currents and voltages through the use of a function generator and dc power supply using a DMM
6. Check to see if they coincide with the values listed in the input and output conditions

4.1.6 References & File Links

[1] '12" eTape Liquid Level Sensor + extras. https://www.adafruit.com/product/464?gclid=CjwKCAiA68ebBhB-EiwALVC-NhmQ31Q75J4d-33izgz472_cPoMng5jPuq3LiBkD3mdBQyXlxS7yQRoCAfkQAvD_BwE (accessed January 16, 2023).

[2] 'eTape Datasheet'. https://cdn-shop.adafruit.com/datasheets/eTape%20Datasheet%2012110215TC-12_040213.pdf (accessed January 16, 2023).

4.1.7 Revision Table

Date	Name	Revision Made
5/10/2023	Grace	Finished section 4.1 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.1
3/10/2023	Grace	Edited section 4.1 (grammar, spelling, etc.) and relabeled figures
2/11/23	Emma	Added more specific information about verification steps
2/11/23	Emma	Updated interface validation tables, added IEEE labels and explanations for each of the tables
2/11/23	Emma	Added additional figures (2 & 3) to the design section
2/11/23	Emma	Added depth to description section
1/20/23	Emma	Completed filling in the rest of sections 1, 2, 3, 4, 6 and filled in 5
1/16/23	Emma	Created the initial doc
1/16/23	Emma	Added to sections 1, 2, 3, 4, 6

4.2 Application

4.2.1 Description

The GUI will take in information that was transmitted via radio communication from a microcontroller. This information includes environmental measurements from a temperature/humidity sensor, water level sensor, and wind speed sensor. This GUI will then display the data in a graphical format at 30-minute intervals in order to provide the farmers with accurate information they can use to determine agricultural operations. Our team decided that it would be the best option to write the program in C# rather than the previous team's UI which was a combination of PHP, Processing, and C++ as it can handle and store the data much easier. We also want this data to be able to be sent over a distance of approximately 1 km in the presence of obstructions as the farms in Thailand are non-contiguous. This can all be tested by setting up test code that provides the UI with dummy variables for agricultural data and illustrates that the website can display this information while each of the radio nodes are separated by 1 km and are not in line-of-sight. The ultimate goal of this project would be to have a network of sensor units spread throughout the entire farm that would be able to transfer and display all of their data, but for simplicity's sake, our team's goal is to figure out how to communicate this data through 2 or 3 nodes.

4.2.2 Design

The figure below is a black box diagram for the user interface, including both of the interfaces that enter and leave the block. The data enters the block via a LoRa radio transmission (rd_dt_trnsfr_systm_applctn_rf) over a 923 Hz frequency. This is then displayed on a single computer that is illustrated by the applctn_otsd_data interface that is added to/updated every 30 minutes.

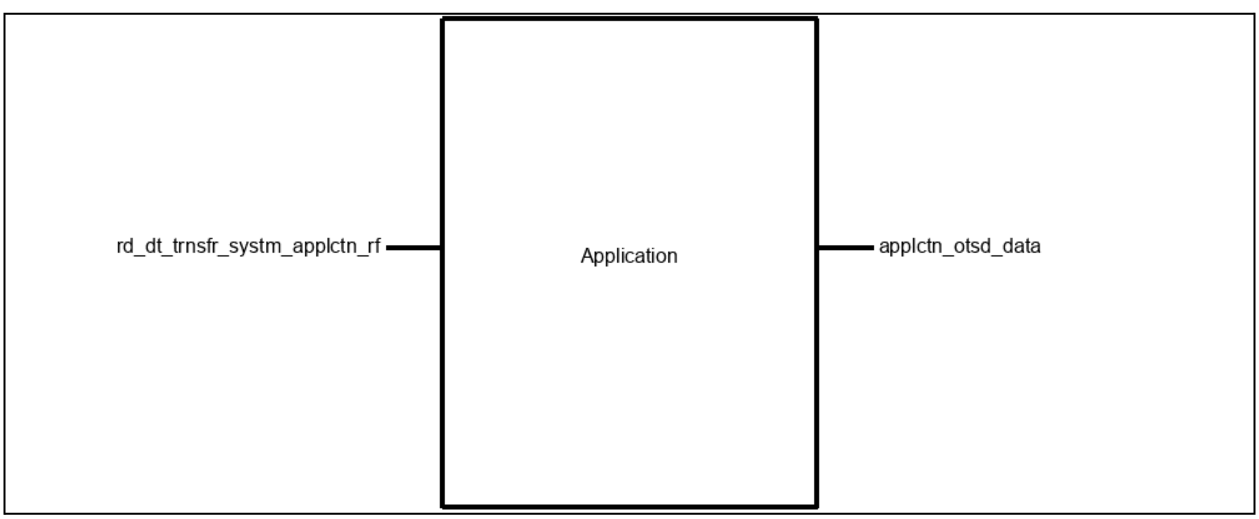


Figure 4.2-1: Black box diagram for the GUI.

As previously mentioned, the team that worked on this project before us wrote all of their UI in the PHP, Processing, and C++ coding languages. We were initially going to do this as well, but after conducting some research of our own we determined either Python or C# would be better suited to this project and ultimately decided on C#. C# has a better ability to collect, store and display the data we need it to.

4.2.3 General Validation

As I have mentioned before, the previous team designed their UI using PHP, Processing, and C++. It successfully displayed data from sensors, however, we determined that there was an easier way to do it. Our team decided that using a language like Python or C# would be a more conducive way to implement the UI, not only because we have more knowledge in these languages, but also because we believe that it will be simpler to implement and do everything that the prior team did in three separate languages. We chose to write the program in C# instead as it’s able to do everything the previous code did, but more efficiently.

4.2.4 Interface Validation

The following tables go into detail about each of the interfaces of this block as well as the specific properties that make them up.

Table 4.2-1: applctn_otsd_data : Output

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Datarate: 30 min	This time interval was selected by our project customer.	When coding, you can set this value as a variable so it maintains constant and reliable transmissions/displays.
Messages: Weather data from each of the sensors (Celsius, km/h, cm, etc.)	This data was selected by our project customer.	The graphical representation of the data on the web page will display this in a clear format.
Other: Written in C#	This was determined by the team as a whole.	The entirety of the code written for the UI is in C#.

Table 4.2-2: rd_dt_trnsfr_systm_applctn_rf : Input

Other: Range: 1 km	The sending range was specified by the project customer.	RFM95 Module: - The RFM95 is advertised as having a range of up to 2km with a simple wire antenna [2]
Other: Frequency: 923 MHz	The frequency selected was based on current regulations in Thailand, and the band from 920-925 MHz is recommended to use for unlicensed IoT applications.	From rfm95 library reference: - Main frequency range is 868-915 MHz, but can go higher, it has been tested as far as 950 MHz [1]
Protocol: LoRa	The LoRa protocol was chosen because of the long-range potential that it has while having a low power requirement.	RFM95 Module: - The RFM95 uses the LoRa protocol because of its effectiveness at long ranges [2]

4.2.5 Verification Plan

Output Interface:

1. Run the program
2. Wait more than 30 minutes
3. Check to see if environmental data is being displayed every 30 minutes
4. Check the GitHub file to ensure the code is written in C#

Input Interface:

1. Connect two separate radio modules to two microcontrollers with the appropriate SPI pins (On Arduino, SCK goes to pin 13, MISO is pin 12, MOSI is pin 11, and CS is configurable but is usually pin 10 by default). Then, assign RST and IRQ pins in the code (pins 9 and 2, respectively, by default) and make the connections. Lastly, connect 5V and ground pins.
2. Power on both of the microcontrollers with test code for sending and receiving
3. Verify that the radios are communicating with each other over a short distance

4. Slowly move one of the radios away from the other, making sure that it is still receiving messages from the other radio
5. Continue moving away until either the radios lose connection to one another or they are 1 km apart and still transmitting data

4.2.6 References & File Links

[1] "RFM95 Class Reference", *LoRaLib*. [Online]. Available: https://jgromes.github.io/LoRaLib/class_r_f_m95.html. [Accessed: 8-Mar-2023]

[2] "Adafruit RFM95W LoRa Radio Transceiver Breakout - 868 or 915 MHz - RadioFruit", *Adafruit*. [Online]. Available: https://www.adafruit.com/product/3072. [Accessed: 7-Mar-2023]

4.2.7 Revision Table

Date	**Name**	**Revision Made**
5/11/2023	Emma	Made updates to black box diagram and interface table consistent with design changes
5/10/2023	Grace	Finished section 4.2 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.2
3/10/2023	Emma	Finished sections 4-7
3/9/2023	Emma	Added section 1, 2, 3
3/9/2023	Emma	Created revision table

4.3 Radio Data Transfer System

4.3.1 Description

This block is a radio module that needs to be able to send data from one point to another point up to 1 km away. The data that is being transmitted will consist of sensor readings, address requests, and error messages. As such, the module should be able to receive data from a microcontroller, and then send that data to another radio module up to 1 km away, which should accurately pass that data into another microcontroller.

To do this, the radio module that was chosen is the RFM95W transceiver breakout board from Adafruit, along with a 900 MHz antenna. The range of this is advertised to be up to 2 km when using a long wire antenna, and up to 20 km line-of-sight using directional antennas, which is much more than the project would need, and would certainly be able to accommodate some line-of-sight obstructions. In addition, it is operated through a SPI interface, which can be handled with libraries by the most common microcontrollers.

4.3.2 Design

The block consists of an RFM95W breakout board, an antenna mount with an antenna, and connections to pins on a microcontroller for MOSI, MISO, SCK, CS, Reset, and IRQ. The block needs to be powered by a 3-5V supply to function properly and uses the LoRa protocol when sending radio messages.

The three interfaces to the block are a radio input/output interface, 5V DC power, and an SPI interface with the microcontroller. For SPI communication, the clock speed is 1 MHz, and the module uses the 4-wire version of the protocol. The data that is being sent is a series of bytes, which can be represented in the microcontroller’s code as an array of characters or an array of bytes. A black box diagram of the radio module is shown below.

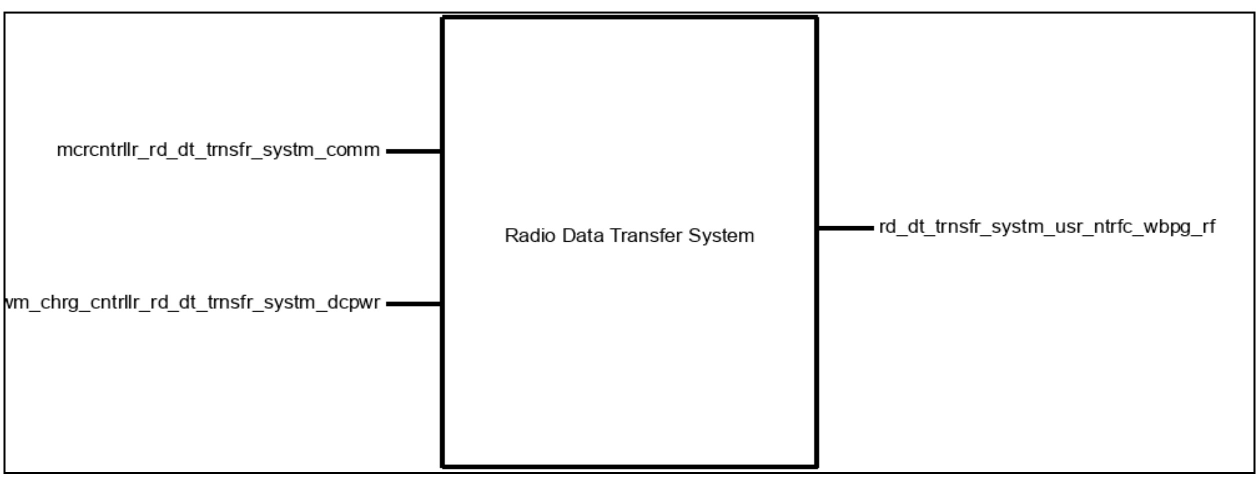


Figure 4.3-1: Black Box Diagram

The power should be provided by an external power supply rather than directly from the microcontroller because the radio module can draw a significant amount of current during transmission. While it may not draw more than the microcontroller can provide by itself, when combined with the other sensors that are connected to the microcontroller, it may draw too much current and cause system failure.

The microcontroller being used for this project is the Arduino Nano, which has libraries available for running the RFM95 radio module, without needing to access the individual registers. This handles both ends of the SPI interface and does most of the difficult tasks automatically.

For the radio transfer system, a specialized omnidirectional antenna for 900 MHz is included as part of the system. This greatly increases the potential range that the radios can transmit over and accommodates line-of-sight disruptions much better than a simple wire antenna. This will make for more consistent data transfer as well, as packet losses will occur much less frequently.

4.3.3 General Validation

The selected parts will meet the engineering requirements set out by the team and the requirements set out by the project customer. These include an overall cost and availability requirement, requirements for the data transmission range, and requirements for what types of data are being transmitted. In addition, the block should be easily understood and buildable by somebody who does not have an engineering background, because the farmers that the sensors will be provided to will have to be able to assemble units on their own, without instruction from the project group.

The most important requirement that this block needs to fulfill is the range requirement. The official requirement set out by the team is that the radio module should be able to transfer data over 1 km with a 99% success rate. Because another requirement is that data transmissions only need to occur every 30 minutes, getting a 99% success rate requires the sensor data to be successfully received once every half hour with 99% accuracy. The RFM95 radio modules can transmit up to 2 km with a simple wire antenna according to the product page and can achieve further ranges with better antennas [1]. This will easily meet the range of requirements for the system.

Another important requirement that needs to be met by the system is an overall cost requirement, so all of the components in the system should fulfill the other requirements while being the lowest possible overall cost. The RFM95 radio modules from Adafruit are relatively cheap for a LoRa module at $20 apiece [1]. This allows for significantly greater performance than is required at a relatively low cost.

An alternative to using these radio breakout boards with an Arduino running them would be to purchase microcontrollers that have radio functionality built in. This is what was implemented in the previous project, as they used the Adafruit Feather M0 with the RFM95 module attached, which overall does cost less than an Arduino and RFM95 breakout board [2]. However, these boards are not in stock and have not been in stock at any point during the planning process for this project. Since part availability is one of our requirements, a product that is not in stock very often should not be chosen so that the project customer can purchase more of the parts on their own without waiting too long.

One potential downside to using the modules is the complication that it adds when assembling the project. The project customer must be able to assemble our project without any input from the people who were involved in the creation of the project. However, this should not be a major issue for the radio module, since it does not require an extensive amount of connections, and does not require additional configuration to work with an Arduino board.

4.3.4 Interface Validation

The following tables outline the properties of each of the interfaces to and from the radio module. This includes range requirements, radio frequencies, and power requirements.

Table 4.3-1: mcrcntrllr_rd_dt_trnsfr_systm_comm: SPI Interface

Datarate: 1 MHz	The data rate was set because of the libraries for the radio modules, which operate at 1 MHz	RFM95 Module: - The SPI clock speed set in the RFM95 library for Arduino is 1 MHz [1]
Protocol: SPI	Communication with the radio module should be done using the Serial Peripheral Interface protocol to edit the registers in the chip	RFM95 Module: - The adafruit website states that the module is based on the SX1276 with SPI interface [1]
Protocol: 4-Wire	The SPI protocol used should have 4 connections: MOSI, MISO, SCK, and CS	RFM95 Module: - 4-Wire is the standard SPI interface, and the pinout for the RFM95 breakout module has 4 SPI connections [1]

Table 4.3-2: rd_dt_trnsfr_systm_usr_ntrfc_wbpg_rf: **Radio Output**

Frequency: 923 MHz	The frequency selected was based on current regulations in Thailand, and the band from 920-925 MHz is recommended to use for unlicensed IoT applications	From rfm95 library reference: - Main frequency range is 868-915 MHz but can go higher, it has been tested as far as 950 MHz [3]
Sending	The sending range was specified by	RFM95 Module:

Range: 1 km	the project customer	- The RFM95 is advertised as having a range of up to 2km with a simple wire antenna [1]
Protocol: LoRa	The LoRa protocol was chosen because of the long-range potential that it has while having a low power requirement	RFM95 Module: - The RFM95 uses the LoRa protocol because of its effectiveness at long ranges [1]

[3]

Table 4.3-3: pwm_chrg_cntrllr_rd_dt_trnsfr_systm_dcpwr: **Input**

$I_{nominal}$: 30 mA	The nominal current is based on the expected current requirement of the radio module when it is not actively transmitting	RFM95 Module: - The radio consumes around 30 mA of current during active listening [1]
I_{Peak}: 120 mA	The peak current is the maximum current that might be seen if the radio module is actively transmitting at maximum power	RFM95 Module: - The radio consumes around 100 mA of current when transmitting at maximum power (20 dBm) [1]
V_{max}: 5V	The maximum voltage is based on the voltage range that the radio module is rated to operate at	RFM95 Module: - The RFM95 module is designed to work with both 5V and 3V circuits, with a regulator and level shifter [1]
V_{min}: 3V	The minimum voltage ensures that the device will also be compatible with 3.3V logic if necessary	RFM95 Module: - The RFM95 module is designed to work with both 5V and 3V circuits, with a regulator and level shifter [1]

4.3.5 Verification Plan

The process for validation of the radio module is as follows:

1. Connect two separate radio modules to two microcontrollers with the appropriate SPI pins (On Arduino, SCK goes to pin 13, MISO is pin 12, MOSI is pin 11, and CS is configurable but is usually pin 10 by default). Then, assign RST and IRQ pins in the

code (pins 9 and 2, respectively, by default) and make the connections. Lastly, connect 5V and ground pins.

2. Power on both of the microcontrollers with test code for sending and receiving
3. Verify that the radios are communicating with each other over a short distance
4. Slowly move one of the radios away from the other, making sure that it is still receiving messages from the other radio.
5. Continue moving away until either the radios lose connection to one another or they are 1km apart and still transmitting data

4.3.6 References & File Links

[1] “Adafruit RFM95W LoRa Radio Transceiver Breakout - 868 or 915 MHz - RadioFruit”, *Adafruit*. [Online]. Available: https://www.adafruit.com/product/3072. [Accessed: 7-Mar-2023]

[2] “Adafruit Feather M0 with RFM95 LoRa Radio - 900MHz - RadioFruit”, *Adafruit*. [Online]. Available: https://www.adafruit.com/product/3178. [Accessed: 7-Mar-2023]

[3] “RFM95 Class Reference”, *LoRaLib*. [Online]. Available: https://jgromes.github.io/LoRaLib/class_r_f_m95.html. [Accessed: 8-Mar-2023]

4.3.7 Revision Table

Date	**Name**	**Revision Made**
5/10/2023	Grace	Finished section 4.3 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.3
3/10/2023	Grace	Edited section 4.3 (grammar, spelling, etc.) and relabeled figures
3/10/2023	Garren	Updated references to IEEE format
3/9/2023	Garren	Updated Interface Validation Section
3/7/2023	Garren	Added revision table
3/7/2023	Garren	Added all sections

4.4 Wind Speed Sensor

4.4.1 Description

This block is a sensor that is meant to measure wind speed, with an effective range of 0-75 kilometers per hour. This data can then be fed back to a microcontroller, which processes the sensor readings and then can send them via radio communications to a central computer for storage. As such, the sensor needs to be able to produce an output that can be interpreted by the microcontroller. The methodology that the wind sensor uses does not matter, for this project, as long as it can be interpreted by the microcontroller and also is able to be used in outdoor conditions.

To do this, the sensor that was selected was the Rev C Wind Speed Sensor made by Modern Device. This sensor was selected because of its cost and accuracy, making it the best choice within the constraints of this project. It provides an analog output that can be interpreted by the Arduino microcontroller being used to provide an accurate wind speed and requires 5V which is consistent with the other portions of the project.

4.4.2 Design

The entire block consists of one wind speed sensor, which has 5V and Ground inputs, and an analog output that outputs 0-5V. The sensor selected uses a heating element to measure the wind speed, by measuring the current required to maintain a certain temperature on a metal plate.

The three main interfaces that are required for the wind speed sensor are 5V of DC power, an interface with the environment that is producing the wind, and communications with the microcontroller. The communications are going to be in the form of an analog signal that varies from 0-5V depending on the measured wind speed. From there, the microcontroller will need to process the signal to determine the exact numerical value of the wind speed. Figure 4.4-1 shows the black box diagram of the overall block with the 3 interfaces labeled.

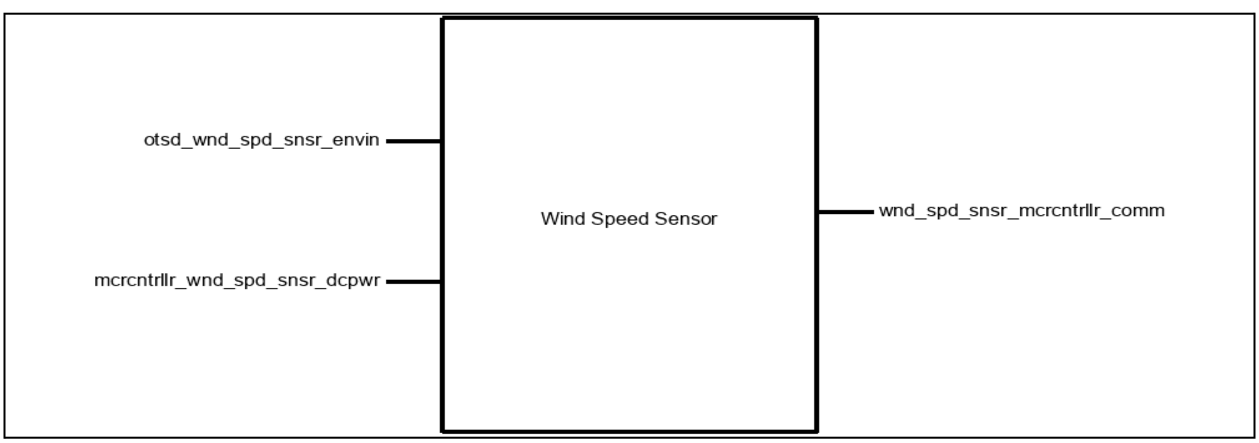


Figure 4.4-1: Black Box Diagram

The power interfaces would be connected directly to the power coming from the same power source that is powering the microcontroller, rather than from the microcontroller, because the sensor may draw more current at high wind speeds than the microcontroller would be able to provide if it were also powering other sensors. Figure 4.4-2 shows the connection diagram for the sensor's 3 connections, which are ground, 5V, and analog output.

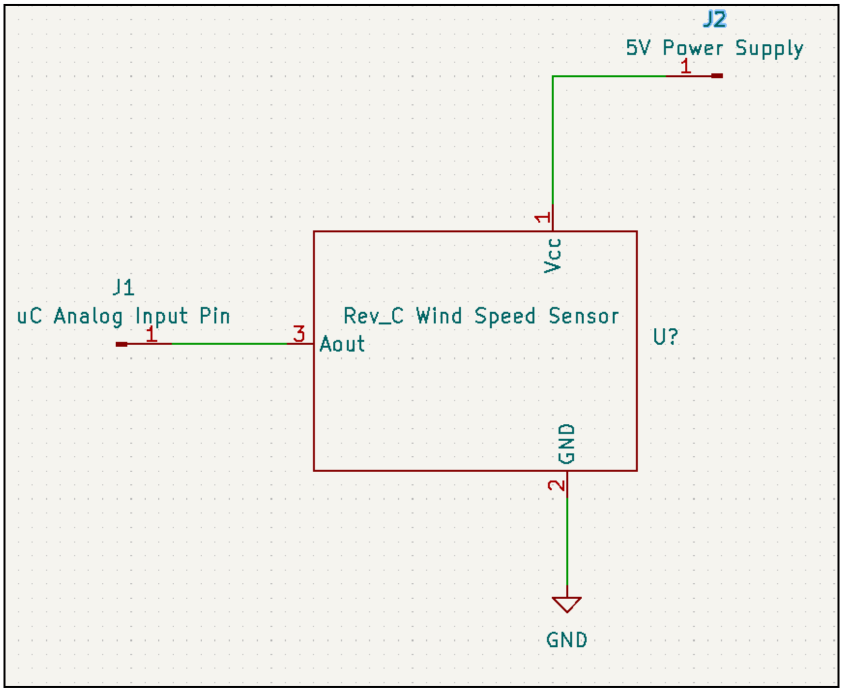


Figure 4.4-2: Connection diagram for the Rev C Wind Sensor

The wiring diagram is simple, which means that the majority of the difficulties in validating the block will come from making sure that the microcontroller is able to correctly interpret the analog output from the wind speed sensor and convert it into a useful value that can then be interpreted by humans who are using the system. This will require some experimentation in the code to find the correct constant to multiply the analog voltage by to derive the current wind speed that is being measured by the sensor.

4.4.3 General Validation

The selected parts and overall design of this sensor system meet both the requirements that were given by the customer and the engineering requirements set by the team. These include an overall cost and availability requirement, requirements for the type of outputs that the sensor should provide, and range requirements for the measurements. In addition, the block should be easily understood and buildable by somebody who does not have an engineering background, because the farmers that the sensors will be provided to will have to be able to assemble units on their own, without instruction from the project group.

The most important requirement that needs to be met is that the sensor can accurately measure the wind speed. Both of the main methods of measuring wind speed are fairly accurate, those using a spinning rotor and using a heating element to measure heat loss from the wind. In general, the heating element method is more accurate and also cheaper, with the drawback that it may not function properly at very high wind speeds and it has to be waterproofed in order to be used outside. Because the wind speeds of the area where the sensors would be deployed typically do not reach extreme levels, the heating method of the sensor should produce accurate results once calibrated properly.

In addition, the cost of a heating element wind speed sensor plus the cost of materials to make it water resistant would be less than the cost of a rotary wind sensor, which is why it was chosen. Cost is an important factor for this project because it is meant to eventually get mass-produced and placed in many locations in farmer's fields. As such, each individual unit should have as low of a cost as possible, and all of the parts used for the project should also be available for purchase in Thailand, where it will be deployed, or else have viable alternatives in Thailand that can be easily swapped out for the original sensors. For these reasons, the heating element wind sensor was chosen for this project. There was originally a plan to also include a wind direction sensor, but that was scrapped because sensors that measure both wind speed and direction are significantly more expensive than sensors that measure wind speed only, and wind direction was not a completely necessary component of the sensor box.

The sensor is also supposed to produce an output that can be easily read by a microcontroller. For this to be the case, the two easiest options are to use the SPI protocol or to use an analog input pin on the microcontroller to directly read the output from the sensor. Because pin space is limited on the microcontroller due to other sensors using the SPI protocol, an analog output from the sensor is ideal so that it only takes up one pin on the microcontroller. As such, a sensor was selected that provided an analog output. This means that the microcontroller will need to do some processing on the data, but overall keeps the pin usage and the cost to a minimum.

As a result of all of these requirements, the sensor chosen is the Rev C Wind Speed Sensor from Modern Device. It is a heating element wind sensor that was the least expensive out of all the options while retaining accuracy, costing $21.95 [1]. This would allow for the units to be produced as cheaply as possible while also accurately measuring the wind speed so that the data can be reported to the farmers. This sensor gives an analog signal from 0-5V as output and is also rated for wind speeds of up to 60 mph before it saturates and stops giving accurate readings [1].

There are other alternative sensors that could be used for this block, with the main drawback of the other sensors being the cost. If cost were not prohibitive, a sensor that measures both speed and direction could be used, or a sturdier rotor-type wind sensor could be used, such as the wind sensor made by Adafruit, which costs $44.95 and would give the same kind of output [2]. However, because the Rev C sensor is calibrated to give a logarithmic output for more accurate readings at lower wind speeds, some differences would have to be implemented in the code in order to get accurate wind speed readings from the Adafruit sensor [1].

There is also a requirement that was set by the customer that they should be able to replicate one of the units that the team builds on their own by just ordering the parts and then following instructions left behind by the team. Because of this, a simple sensor with documentation on how to implement it was chosen. On the sensor's product page, there are links to the code for the sensor and instructions for calibrating the sensor, which would be supplementary to the instructions and code provided by the engineering team. This is also the reason that a wind speed sensor was not created from scratch using a small encoder, even though it would have been the most cost-effective option because the ease of assembly, use, and accuracy of a pre-made analog output sensor outweighs the cost-benefit of a self-made sensor.

4.4.4 Interface Validation

The following tables outline the properties of each of the interfaces to and from the wind sensor. This includes wind speed ratings, maximum temperature ratings, and voltage and current ratings for the power and analog output interfaces.

Table 4.4-1: mcrcntrllr_wnd_spd_snsr_dcpwr: Input

$I_{nominal}$: 25mA	The nominal current is based on the expected current need of the sensor	Wind Sensor Rev. C: - Supply Current: 20-40mA (depending on wind speed) - Device self-regulates current to maintain a constant temperature - From the website: The sensor draws ~25 mA of current when taking normal measurements
I_{min}: 20 mA	Minimum current is based on the minimum current draw required by the sensor	Wind Sensor Rev C: - Supply Current: 20-40mA
I_{max}: 40 mA	Maximum current is the maximum amount of current that the sensor will draw from the power source	Wind Sensor Rev C: - Supply Current: 20-40mA
V_{min}: 4V	Minimum voltage is based on the range given by the sensor's datasheet	Wind Sensor Rev C: - The sensor is rated for 4-5 volt supply voltage
$V_{nominal}$: 5V	Nominal voltage is based on the optimal voltage that the wind speed sensor was designed for	Wind Sensor Rev C: - The sensor is rated for 4-5 volt supply voltage
V_{max}: 10V	Maximum voltage is based on the maximum voltage that the wind speed sensor can handle	Wind Sensor Rev C: - The sensor is rated for up to 10 volts

[1] Rev C Product Page, Accessed 1/17/2023

Table 4.4-2: wnd_spd_snsr_mcrcntrllr_comm: Output

V_{min}: 0.5V	Analog output minimum should be the minimum voltage.	Wind Sensor Rev C: - From the website: output signal is analog, from 0.5V to Vcc
V_{max}: 5V	Analog output maximum should be equal or almost equal to the supply voltage.	Wind Sensor Rev C: - From the website: output signal is analog, from 0.5V to Vcc

[1] Rev C product page, Accessed 1/17/2023

Table 4.4-3: otsd_wnd_spd_snsr_envin: Input

Wind Speed: 0-75 km/h	The wind speed measurement requirement is based on the maximum wind speeds that would be encountered in Thailand, with the highest recorded being around 75 kmph.	Wind Sensor Rev C: - From the website: Useful in wind velocities 0-60 mph (0-96 kmph)
Temp: 0-45 ℃	The temperature measurements are based on the temperature in Thailand, which typically does not get below around 10 degrees C, with the highest temperature recorded being 43.7 degrees C [3].	Based on common electronic tests, it is reasonable to assume that the wind speed sensor will work within the specified temperature sensor.
Water Tolerance: Rain/Flooding	The block should be able to tolerate some water exposure because it will be outside in a field, which will experience rainy conditions and possibly flooding.	- Using epoxy or some other kinds of water-resistant finish on the wind speed sensor will not inhibit its ability to measure temperature and would make it resistant to short circuits caused by water.

[1] Rev C Product Page, accessed 1/17/2023

4.4.5 Verification Plan

The process for validation of the wind speed sensor is as follows:

1. Connect the sensor to 5V and GND (on a microcontroller or just raw voltages)
2. Connect some device used to measure the output voltage such as a microcontroller or multimeter

3. Test the sensor at different wind speeds. The most accurate way to do this would be a wind tunnel, but if the team is unable to get access to one, the best alternative would be to drive a car at set speeds and hold the sensor out the window
4. Measure the voltage at different wind speeds and show that the output is an analog signal between 0 and 5 volts, with the low end being at no air movement and the high end being at top wind speeds tested

4.4.6 References & File Links

[1] "Wind Sensor Rev C Product Page," *Wind Speed Sensor Rev C*. [Online]. Available: https://moderndevice.com/products/wind-sensor. [Accessed: 17-Jan-2023].

[2] *Anemometer Wind Speed Sensor*. [Online]. Available: https://www.adafruit.com/product/1733?gclid=Cj0KCQiA8aOeBhCWARIsANRFrQHrFifXoADK4Y4QZxIxqIeSj5dxNztUrfSP2KS16zF2Ilg9rpqFqMIaAgqeEALw_wcB. [Accessed: 17-Jan-2023].

[3] "That's how warm it is in Thailand: 33.0 °C on average per year and over 2100 hours of sunshine!," *Worlddata.info*. [Online]. Available: https://www.worlddata.info/asia/thailand/climate.php. [Accessed: 20-Jan-2023

4.4.7 Revision Table

Date	**Name**	**Revision Made**
5/10/2023	Grace	Finished section 4.4 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.4.
3/10/2023	Grace	Edited section 4.4 (grammar, spelling, etc.) and relabeled figures.
2/10/2023	Garren	Edited interface tables to more accurately reflect numbers on the datasheet, and added an overview to the interface section.
2/8/2023	Garren	Edited introduction section and added content
1/20/2023	Garren	Added revision table
1/17/2023	Garren	Added all sections

4.5 Humidity/Temperature Sensor

4.5.1 Description

This block contributes to the satisfaction of the “Sensor Requirements” project requirement. The customer requested a system capable of measuring soil moisture, temperature, humidity, wind speed, and water level which was translated into an engineering requirement of measuring soil moisture up to 0.7 meters cubed of water per cubic meter of soil, temperature ranges of zero to forty-five degrees celsius, relative humidity from zero to one-hundred percent, wind speeds up to seventy-five kilometers per hour, and water levels up to twelve inches. Upon completion of this block, the microcontroller will receive data containing the air humidity and temperature readings via a Serial Peripheral Interface.

The configuration used in this block operates a Bosch BME280 Digital humidity, pressure, and temperature sensor taking only temperature and humidity readings once per second and relaying them to the microcontroller via a Serial Peripheral Interface with data speeds of up to 2 million bits per second [1]. The sensor uses minimal power which does not directly satisfy any requirements for the system but was a consideration from the project partner - specifically, it needs an input voltage between three and five volts with a maximum current draw below 2 milliamperes. The datasheet also rates the temperature ranges from -40°C - 85°C but the realistic temperatures in Bangkok, Thailand have not dropped below or exceeded the requirement range in 2022 easing the extremes for which the system must operate [2]. Instead, the focus shifted to the humidity which can only satisfy the requirement between the temperature range of 0°C - 60°C [1].

The testing process for the block includes subjecting it to the extreme environmental conditions expected for the interface. This requires simulating relative humidity levels of zero percent and one hundred percent and temperatures of zero degrees Celsius and forty-five degrees Celsius. The SPI interface will use a visual inspection for the protocol (four-wire connection) and an oscilloscope to get the clock frequency of the data transmissions. Finally, the input DC power interface can implement two measurement tools (a controlled DC power supply and an ammeter) to ensure the device operates at both three and five volts with a nominal current draw of 1.2 mA and never exceeding 2 mA. If the block can successfully measure the environment, transmit it to the microcontroller, and operate under these input power conditions then it will meet the necessary interfaces to the other connected blocks.

4.5.2 Design

The block measuring the temperature and humidity of the environment and communicating with the microcontroller centers around the Bosch BME280 combined Digital humidity, pressure, and temperature sensor. It is placed on an Adafruit breakout board along with a voltage regulator to allow the board to receive a five-volt input from the microcontroller with minimal additional current draw [3]. The microcontroller communicates with the sensor using the Serial Peripheral Interface protocol (SPI) at a synchronous clock speed of one megahertz allowing for both data input and output from either device simultaneously.

The firmware necessary to operate the microcontroller initializes the device in "normal mode" effectively allowing the sensor to operate independently from the microcontroller and only interact with each other when the microcontroller queries the most recent temperature or humidity reading. These readings update automatically once per second by the sensor when it takes a measurement. Interestingly the breakout board includes a MIC5225-3.3YM5 Dropout Regulator between the input voltage supply pin and the BME280 to ensure a consistent 3.3-volt difference is produced for the sensor allowing a much higher voltage supply from the microcontroller. Signals produced by the sensor go through a level-shifting process back up to five volts via two transistors connected to pull-up resistors. This process requires more current than simply powering the sensor itself but it provides flexibility to the system to allow for different power inputs.

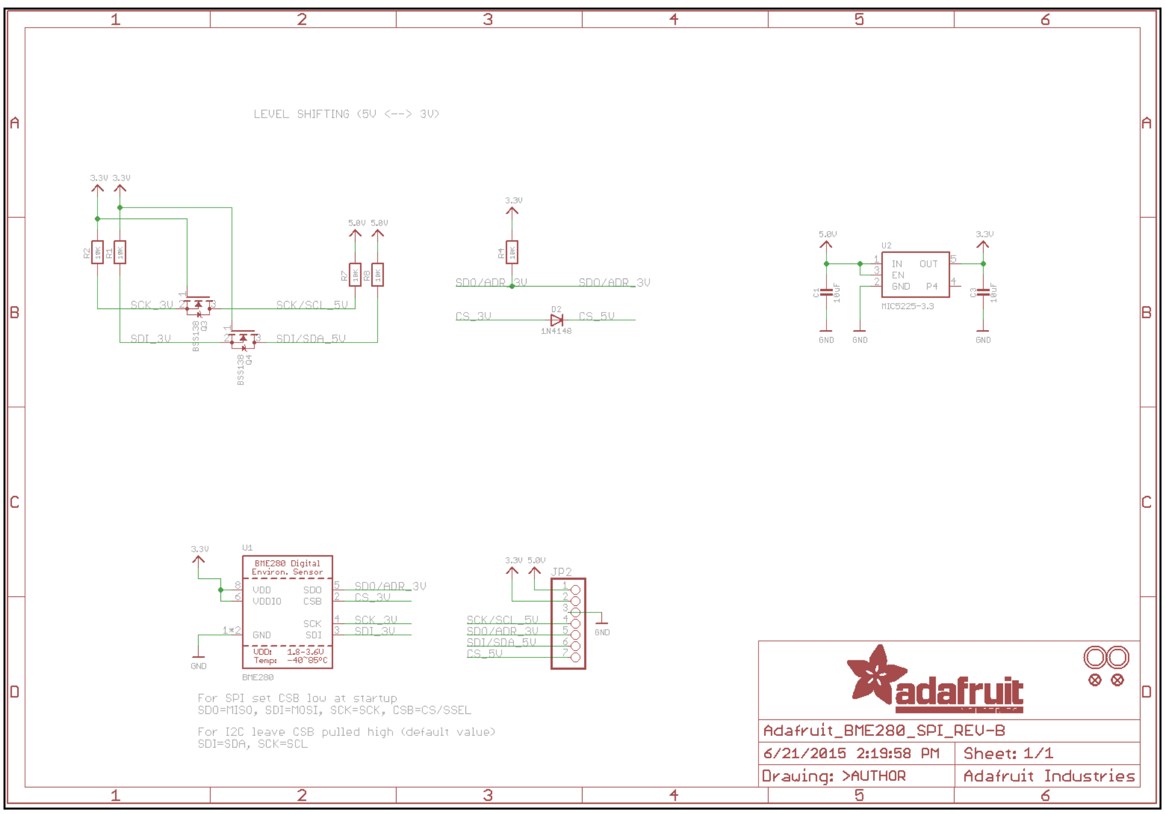


Figure 4.5-1: Breakout board schematic [3]

An additional aspect of the design for this section was defining the interfaces between the exterior world and communicating it via interfaces to the microcontroller. The interface to the microcontroller had to allow for several different types of processors including different logic levels during the initial design phase of the project, so the flexibility of either a five-volt or three-volt supply allowed for that interface to wait for definition without slowing down the overall process and individual block design. The environmental interfaces were defined both for the hardware limitations (i.e. only measuring the full range of humidity between 0°C - 60°C) and the realistic environmental conditions in Thailand [4].

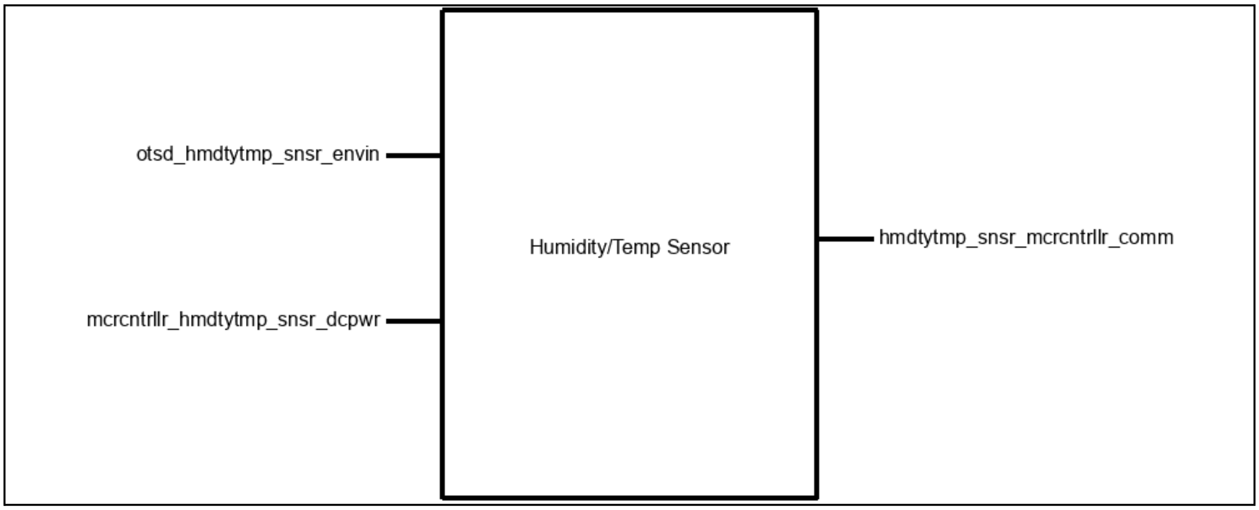


Figure 4.5-2: Block diagram

4.5.3 General Validation

The block centers around the Bosch BME280 combined Digital humidity, pressure, and temperature sensor placed on an Adafruit breakout board. This product was chosen by the team because it provides both relative humidity and temperature data from the environment to the microcontroller via commonly used embedded data transfer protocols. Additionally, Adafruit provides an open-source Arduino library with examples available for any programmers to use and experiment with which decreases the amount of engineering time required to create a program to use the sensor. Finally, it includes a voltage regulator allowing for digital inputs into the block between three and five volts but only requires about a milliampere of current which gives the team more flexibility for other aspects of the design. Considering all these conditions, the Adafruit breakout board containing the Bosch BME280 combined humidity and pressure sensor was the best choice to satisfy this block.

The datasheet of the BME280 states the integrated circuit can detect temperature ranges between -40°C - 85°C with an accuracy of ±1°C, however, the interfaces defined by the team only require a temperature range between 0°C - 45°C to reflect realistic conditions in Bangkok, Thailand [1][2][4]. The decreased range was decided for an additional two reasons: it is easier to simulate these temperatures with the appliances available to the team and according to [1, Fig. 1] the sensor can only read the full humidity range (0% - 100%) in the temperature range of 0°C - 60°C. Additionally, the datasheet for the BME280 suggests an accuracy of ±3% for each relative humidity reading [1]. Once the sensor physically collects the data it can use one of three communication protocols to transmit it to a central device; specifically a two-wire Integrated Circuit connection, a three-wire Serial Peripheral Interface (SPI), or a four-wire SPI [1]. This design implemented the four-wire SPI connection because other sensors or peripheral devices can use the same physical connections with an addition of a single I/O pin on the microcontroller per sensor. Finally, the clock speed of the interface can vary between 0 - 10 MHz but the Arduino Library used in the implementation defaults the speed at 1 MHz [1][5].

The Adafruit breakout board including the BME280 was particularly attractive to the team as a module to use for the block because it includes an open-source GitHub repository complete with several examples [5]. This decreased both the total cost of the system by not requiring any purchasable firmware and decreased the engineering effort required to create a functional block; instead, most of the effort for this block was focused on simulating the high-humidity environment of the project customer. Another request from the project partner was to use easily accessible modules to increase the likelihood of availability in Thailand. Custom-designed PCBs and individual circuit components are difficult to assemble in large quantities for the project partner.

The project partner voiced concern for the current amount of theft in the deployment region of the system and requested a constraint of a cheap, low-wattage solar panel for easy replacement which limits the power consumption available to each block within the system. The BME280 sensor itself requires a peak current of 350 μA when in the process of making a measurement and a nominal current of 3.6 μA when not in the process of measuring which is negligible compared to the rest of the system [1]. However, the breakout board also includes a linear voltage regulator to reduce the input supply voltage for the BME280 and then subsequently raise the output voltage logic signal on the BME280 from three volts to five volts which in turn increases the overall current draw of the block to about one mA [6][7]. Assuming the worst-case scenario of supplying two milliamps at five volts the block consumes only ten milliwatts of power - insignificant to a 50-watt supply. Thus the power draw of the temperature and humidity sensor interacts nicely with the other requirements and constraints of the system.

This block is designed to easily provide temperature and humidity data to the microcontroller block of the system with low power consumption. The sensor used to accomplish this task is the Bosch BME280 combined Digital humidity, pressure, and temperature sensor placed on an Adafruit breakout board. Included with the implementation of the breakout board is a detailed Arduino library with other project examples already included in the GitHub repository. The block could also implement an AM2320 sensor but as mentioned in the product description the device lacks quality documentation and has a high variance in the reading [8][9]. The AM2320 costs only four dollars but uses more current at a higher voltage which consumes more power [8]. Adafruit also warns developers to expect low-quality documentation with a high variance in the reading [9]. If the block implemented the AM2320 instead of the BME280 it would require some physical protection to ensure the sensor does not get damaged as the product ships as a singular integrated circuit, whereas the BME280 is already placed on a board [3]. Finally, the AM2320 does not have any example projects on the Adafruit website, so the development time would significantly increase. Due to these factors, the team decided to spend the extra eleven dollars to get a higher quality sensor requiring less development time.

4.5.4 Interface Validation

This section includes a table detailing the three different interfaces between this block and the overall system. These interfaces include the outside environment as an input, the DC power source supply as an input, and the data communication protocol as an output. The first column details the values of each property within the interfaces, the second column describes the reasoning behind the value of the property, and the third column explains the reasoning behind why the values were set at each value.

Table 4.5-1: otsd_hmdtytmp_snsr_envin: Input

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Humidity: 25 - 95%	This range was chosen for three reasons: it encompasses the relative humidity range of Thailand, the sensor can read all values within the range, and it is easier to simulate with the materials available to the team [1][4].	According to the datasheet of the Bosch BME280 combined Digital humidity, pressure, and temperature sensor: ● [1, Figure 4.5-1] shows a range from zero to one hundred percent between the temperatures of zero to sixty degrees Celsius. ● [1, Table 2] shows the operating range between zero and one hundred percent.
Temperature (Absolute): 0 - 45 [Celsius]	This is within the maximum rating of the sensor, it encompasses the temperatures expected in a tropical environment, and it ensures the sensor can measure the humidity ranges from 25 - 95% [1][2].	According to the datasheet of the Bosch BME280 combined Digital humidity, pressure, and temperature sensor: ● Table 2 (humidity parameter specification) shows the operating range between negative forty and eighty-five degrees Celsius.

Table 4.5-2: hmdtytmp_snsr_mcrcntrllr_comm: Output

Datarate: 1 MHz	The range for this value is between 0 - 10 MHz so we chose an arbitrary clock speed. The Arduino library defaults to a 1 MHz clock speed and there is no need in this project to change it [1][5].	According to the datasheet of the Bosch BME280 combined Digital humidity, pressure, and temperature sensor: ● Table 34 (SPI Timings) states the clock input frequency should be set between zero and ten megahertz.
Protocol: SPI	SPI communication has been easier to debug and supports multiple peripherals at a time on the same microcontroller if necessary. The connection will include 4 pins: a clock, a slave select, a master-in slave-out, and a slave-out master-in to allow for two-direction data transfer [1].	According to the datasheet of the Bosch BME280 combined Digital humidity, pressure, and temperature sensor: ● The “General Description” section on page three states “The sensor provides both SPI and I²C interfaces…”. ● The “Key Features” on page two states the device can handle both three and four-wire SPI with clock speeds up to 10 MHz.

Table 4.5-3: mcrcntrllr_hmdtytmp_snsr_dcpwr: Input

Inominal: 1.2 mA	The Bosch BME280 itself only uses about 3.6 microamps of nominal current however the rest of the breakout board draws more current to level shift from the five-volt signal to a 3.3-volt input to the sensor and then back out to a 5-volt output. At five volts this increases the nominal current draw [1].	According to the datasheet of the Bosch BME280 combined Digital humidity, pressure, and temperature sensor: ● The “Key Features” on page two states a 3.6 microamp current consumption on a 1-hertz frequency of humidity, pressure, and temperature. The pressure will not be enabled, so the actual

		current should be less. According to the schematic provided by Adafruit [3]: ● The breakout board includes a linear regulator and multiple pull-up resistors which will increase the current draw from the voltage supply.
Ipeak: 2 mA	The nominal current draw is the amount of current when the sensor is not operating in the measurement mode. Once per second, the sensor increases the current draw by up to 360 microamps for temperature and humidity but it could increase more if it automatically measures the air pressure too. Expecting this peak value should ensure the current never reaches a value too high in unforeseen conditions.	According to the datasheet of the Bosch BME280 combined Digital humidity, pressure, and temperature sensor: ● Table One (General Electrical specification) lists the current during the humidity measurement as 340 uA, reaching the maximum at eighty-five degrees Celsius. ● Table One (General Electrical Specification) lists the current during the temperature measurement as 350 uA, reaching the maximum at eighty-five degrees Celsius.
Vmax: 5 V	The breakout board from Adafruit includes a voltage regulator for the sensor so it can draw directly from a microcontroller.	According to the description of the Adafruit BME280 I2C or SPI Temperature Humidity Pressure Sensor: ● The breakout board includes a 3.3V level regulator, so while the sensor itself can only manage 3.6 volts the block will expect the standard 5-volt supply from a microcontroller.
Vmin: 3 V	The sensor itself only requires a	According to the description of

	1.71-volt supply but the breakout board appears to assume a 3-volt or 5-volt supply pin supply [1][3]. The behavior below this value is undefined so we should keep it within 3 and 5 volts.	the Adafruit BME280 I2C or SPI Temperature Humidity Pressure Sensor: ● The description of the level regulator on the datasheet is somewhat vague, but the breakout board does not describe the behavior if the board receives below 3 volts.

4.5.5 Verification Plan

This section covers the testing procedure for all three separate interfaces. Some interface verifications require the same initial procedure - otherwise, they are independent of each other. It also assumes a computer has downloaded the Arduino program, the BME280 Arduino library, and has a test file to read from BME280 once per second [5].

4.5.5.1 Initialization

1. Using an Arduino Nano and the Adafruit BME280 breakout board connect the serial clock pin (labeled SCK) on the BME280 breakout board to digital pin thirteen on the Arduino Nano Every, the master-in slave-out pin (labeled SDO) to digital pin twelve, the master-out slave-in pin (labeled SDI) to digital pin eleven, and the slave select pin (labeled CS) to digital pin ten.

2. Connect the Arduino Nano Every to the computer using a USB connection and boot the Arduino program. Compile and download the test program firmware onto the Arduino Nano Every and open up the serial monitor.

4.5.5.2 Verifying Input Power Interface

1. Attach two leads to a controlled DC power supply and two more leads to an ammeter. Connect the high potential lead to the high potential side of the ammeter and the low potential side of the voltage source to a ground pin on the Arduino Nano Every. Connect the low potential side of the ammeter to the input voltage pin (labeled VIN) of the BME280 breakout board and a ground pin from the Arduino Nano Every to the ground pin on the breakout board.

2. Turn on the DC power supply and set the voltage difference to three volts. It may be necessary to restart the programming running on the Arduino Nano Every.

3. Observe the serial monitor. New readings should appear once per second.

4. Observe the ammeter and verify the amount of current is lower than the peak current and allow the program to operate for at least thirty seconds to observe the nominal current into the block.

5. Increase the DC power supply voltage to five volts. Repeat steps four and five to ensure the block is operational and the input supply has not exceeded the interface definitions.

4.5.5.3 Verifying Output Communication Interface

1. Connect the five-volt supply and the ground pins from the Arduino Nano Every to the input supply voltage pin (labeled VIN) and the ground pin on the BME280 breakout board.

2. Attach an oscilloscope probe to the serial clock pin on either the BME280 serial clock pin (labeled SCK) or to digital pin thirteen on the Arduino Nano Every.

3. Observe the frequency of the clock speed. Ensure it meets the interface definition of one megahertz. The protocol interface is verified if the connections between the microcontroller and sensor are made and the program is still operational.

4.5.5.4 Verifying Environmental Input Conditions

1. Connect the five-volt supply and the ground pins from the Arduino Nano Every to the input supply voltage pin (labeled VIN) and the ground pin on the BME280 breakout board.

2. Place the BME280 breakout board in a zero percent humidity environment with an external hygrometer verifying the value. Observe the serial monitor for the current value of the humidity the sensor is reading.

3. Place the BME280 breakout board in a one-hundred percent humidity environment with an external hygrometer verifying the value. Observe the serial monitor for the current value of the humidity the sensor is reading.

4. Place the BME280 breakout board in a zero-degree Celsius environment with an external thermometer. Observe the serial monitor for the current value of the humidity the sensor is reading.

5. Place the BME280 breakout board in a forty-five-degree Celsius environment with an external thermometer. Observe the serial monitor for the current value of the humidity the sensor is reading.

4.5.6 References & File Links

[1] Bosch, “Combined humidity and pressure sensor”, BME280 datasheet, Jan. 2022

[2] W. Underground, “Bangkok, Thailand weather history,” *Weather Underground*, 2022. [Online]. Available: https://www.wunderground.com/history/monthly/th/bangkok/VTBD/date/2022-1. [Accessed: 10-Feb-2023].

[3] A. Industries, “Adafruit BME280 I2C or SPI temperature humidity pressure sensor,” *adafruit industries blog RSS*, 2020. [Online]. Available: https://www.adafruit.com/product/2652#technical-details. [Accessed: 10-Feb-2023].

[4] C. Temps, *Relative humidity in Bangkok, Thailand*, 2017. [Online]. Available: http://www.bangkok.climatemps.com/humidity.php#:~:text=The%20average%20annual%20relative%20humidity,January%20to%2085%25%20in%20September. [Accessed: 10-Feb-2023].

[5] Adafruit, (2021), BME280 Library (2.2.2) [Source Code]. https://github.com/adafruit/Adafruit_BME280_Library

[6] Mouser, “Ultra-Low Quiescent Current 150mA µCap Low Dropout Regulator”, MIC5225 datasheet, July 2008.

[7] “Low-dropout regulator,” *Wikipedia*, 11-Jan-2023. [Online]. Available: https://en.wikipedia.org/wiki/Low-dropout_regulator. [Accessed: 10-Feb-2023].

[8] Aosong, “Digital temperature and humidity sensor”, AM2320 datasheet, Unknown.

[9] A. Industries, “AM2320 digital temperature and humidity sensor,” *adafruit industries blog RSS*, 2018. [Online]. Available: https://www.adafruit.com/product/3721?gclid=Cj0KCQiA8aOeBhCWARIsANRFrQGF6yGiqPPc2awjZ2UxGFbaSNiKd3MNvgOBi76NJfBCeLI-tIAnnpwaAtmGEALw_wcB. [Accessed: 10-Feb-2023].

4.5.7 Revision Table

Name	**Date**	**Revision Made**
5/10/2023	Grace	Finished section 4.5 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.5.

3/10/2023	Grace	Edited section 4.5 (grammar, spelling, etc.) and relabeled figures.
2/9/2023	Blake	Finished updating the General Validation section.
2/9/2023	Blake	Addressed questions about the design of the block. Included a schematic picture and description of the components within the block.
2/9/2023	Blake	Rewrote the Verification Plan portion with separate interface sections. Removed specific details regarding creating the necessary environments and allowed for the testing procedure to create those environments as an abstract detail of the verification.
2/8/2023	Blake	Began rewriting the General Validation to address the question: What constraints/requirements did our project partner give us that made us choose this block?
2/7/2023	Blake	Rewrote the General Description section to add strong specific statements about the "need" this block satisfies with more sources (i.e. what part it plays, what function it has, and connecting it to the overall system requirements). Also incorporated how the testing and validation process satisfies the requirements.
2/10/2023	Blake	Added a description to the beginning of the Interface Validation section. Reviewed and altered the current input interfaces to include the additional current from other components on the breakout board. Updated the Description section to match the new interface.
2/10/2023	Blake	Formatted the references section to match IEEE standards, added in-text citations, and labeled figures and tables according to the IEEE standard.
1/20/2023	Blake	Finished the rough draft of sections one through seven in preparation for peer review.
1/19/2023	Blake	Made changes to the description, general validation, and interface validation.
1/18/2023	Blake	Copied the format from the template.

4.6 Microcontroller

4.6.1 Description

This block facilitates the accomplishment of the Sensor, Data Display, and Data Sending Frequency requirements by powering and communicating with the temperature/humidity sensor, the wind speed sensor, and the water level sensor and then packaging the data every thirty minutes to broadcast via the Radio Data Transfer block. The customer requested a system capable of taking these measurements and sending them throughout a network which would be impossible without some control unit. In this case, the Microcontroller. It has three main categories of interfaces (with eight total): input power, output power, and peripheral communication.

The input power interface includes the input from the power charge controller which will connect to the block via the USB port. It has an external regulator so the charge controller can provide close to five volts - although the Arduino can operate slightly below. Most of the input current is used to power the sensors instead of the microcontroller itself, but the current still travels through the block. The output power interfaces include three different DC values, although all of them should have the same vmin / vmax range and they just draw three different currents. The current draw is dependent on the sensor type and design, but the total peak draw is about 55 mA. As long as the supply pin is the 5V pin instead of an I/O pin on the microcontroller, then the difference between the sensors operating at nominal current draw and peak current draw is negligible.

Finally, the block has to communicate with the sensors and the radio data transmitter. It uses two different communication interfaces to work with the sensors: analog sampling and a Serial Peripheral Interface connection. The analog sampling is used because the sensors will provide the data in the form of a range (i.e. the water level sensor might output 5 volts when fully submerged, 2.5 volts when half submerged, and no voltage when dry) and will operate entirely within the microcontroller itself whereas the SPI connection to the temperature and humidity sensor will have two-way communication. The data rate for the Serial Peripheral Interface was chosen to match the radio data transfer interface so that the only code required to change between them is changing the logic levels of the Slave Select Pin. The radio data transmitter also uses a SPI connection, which will not be a problem because the transmission times between the sensor and the radio data transmitter are negligible.

4.6.2 Design

The Microcontroller block handles three main tasks: sensor initialization, reading sensor values, and sending encapsulated data to the Radio Data Transmitter block for transmission to the host node. The microcontroller performs the sensor initialization either through direct communication for the temperature and humidity sensor or by continuously reading values until the sensor has settled into a steady state condition. These operations happen as soon as the microcontroller receives a power input from the charge controller without any explicit design of the block, but it is still a necessary operation to accomplish the other two tasks.

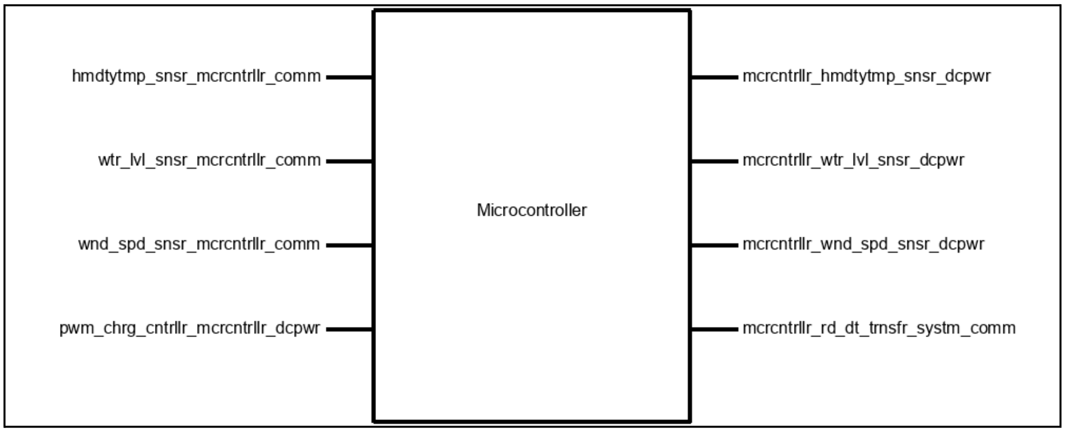


Figure 4.6-1: Microcontroller Block Diagram

The design considerations for this block were focused on the encapsulation of data for transmission throughout the network. The microcontroller receives data from three different inputs (specifically the input communication interfaces) and the structure it packs it into before transmission. Each additional byte adds overhead to the amount of memory the host node needs to store and the probability of dropped packets so the packet structure includes only the necessary information for the system to function. Finally, the microcontroller sends the data to the Radio Data Transmitter block through a Serial Peripheral Interface for transmission to the host node.

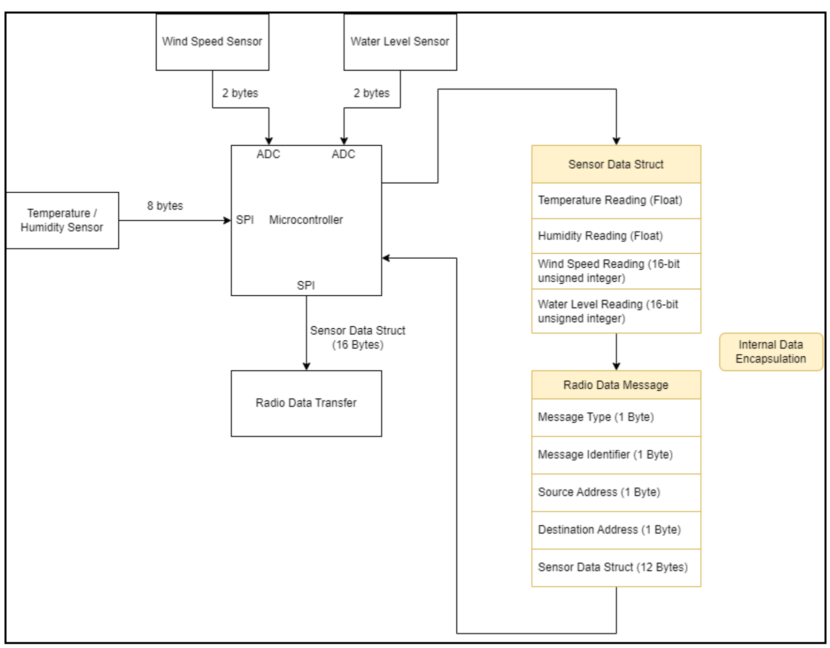


Figure 4.6-2: Internal Data Processing Procedure

The microcontroller has the additional responsibility of powering each sensor. All of them operate between four and five volts which the microcontroller can easily supply and have a relatively small combined peak load of under fifty milliamps, so there was no need to complicate the system design by rerouting power from any other block. The input to the block was set high enough to meet all the output power requirements plus allow the consumption of the microcontroller itself, which can vary depending on the processing power required.

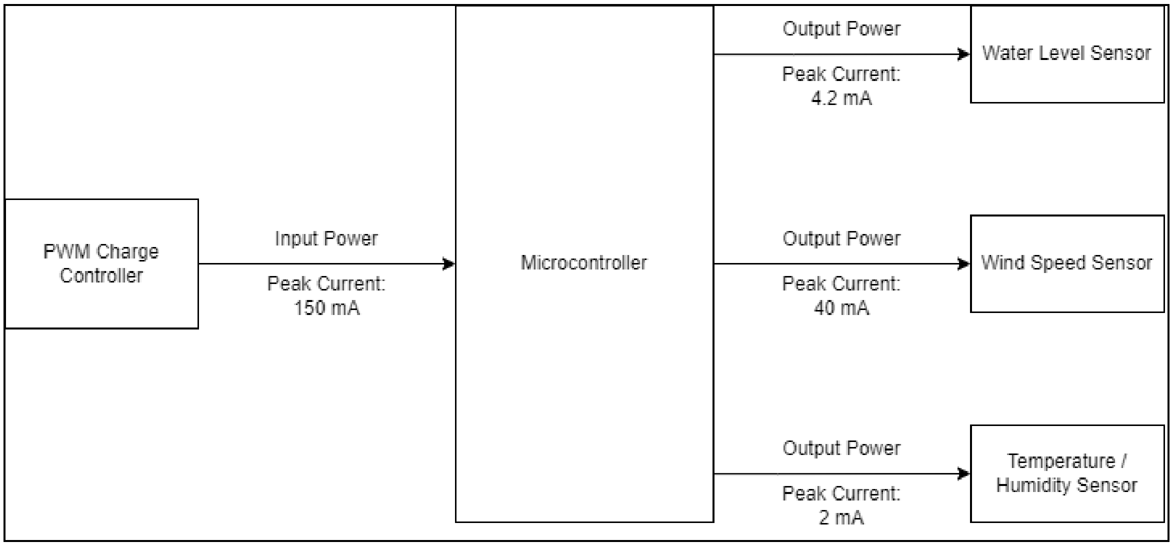


Figure 4.6-3: DC Power Interface Representation

4.6.3 General Validation

The “Description” section of the block validation describes the necessity of a block capable of reading analog voltage inputs, reading data in and out from a Serial Peripheral Interface, and supplying between four and five volts to various peripheral sensors. It also has to run on a five-volt input itself either from a Universal Serial Bus connection or a direct connection to a voltage input pin. Additionally, it must be compatible with both the BME280 Arduino Library by Adafruit and the RadioHead Packet Library to interact with the Temperature and Humidity sensor block and the Radio Data Transfer block. This restricts the choice to only a few options but the two simplest choices are the Arduino Nano and the Arduino Uno. Both of these choices can take analog readings [1], have access to the Arduino built-in Serial Peripheral Interface library [2], include a five-volt power supply output pin with a maximum draw of over fifty milliamps [3][4], and can run on a USB connection. The biggest differences between the two boards are their costs and the number of I/O pins each can support, but the interfaces defined for the block only require a total of eight total I/O pins so the cheaper option was more desirable. Another limitation of the project is the amount of data memory available for each microcontroller, but the UNO and Nano both have 2KB so the deciding factor ended up as the cost difference.

For this documentation, a byte refers to a string of eight consecutive binary bits; thus for the wind speed and water level readings the ADC value must fit within a sixteen-bit string. Included onboard the Arduino Nano is the ATmega328 microprocessor which operates a ten-bit

Analog-to-Digital Converter with at least 6 input pins for comparison [5] which indicates it can measure both sensors separately and fit them within the correct amount of data space. These ten bits readings measure the relative voltage compared to the voltage supply, so it can range anywhere from zero to five volts [5]. Additionally, the ATmega328 has a built-in Serial Peripheral Interface with a three-wire hardware setup to communicate over a shared Serial Clock, Master-In Slave-Out, and Master-Out Slave-In configuration [5]. All three wires are shared between both SPI connections with another digital output sending signals to each peripheral device indicating incoming data. Another important detail for the serial clock data is the speed of oscillation on the serial clock; both the Temperature and Humidity sensor and the Radio Data transfer require speeds of one megahertz which is well within the zero to eight megahertz range [5].

4.6.4 Interface Validation

The table below (Table 4.6-1: Complete List of Interface Definitions) specifies the DC power input interface from the PWM Charge Controller block and all three DC power output interfaces to the Water Level Sensor, Wind Speed Sensor, and Temperature and Humidity Sensor blocks. It also defines the different communication interfaces between the blocks of the system; namely the serial communication from the Temperature and Humidity Sensor block and to the Radio Data Transfer System block and the analog input voltage values from the Water Level Sensor and Wind Speed Sensor blocks. Each entry in the table includes the specified value of the property, why the value is necessary, and how the design can meet the specifications of the value.

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

Table 4.6-1: hmdtytmp_snsr_mcrcntrllr_comm: Input

Datarate: Serial Clock Speed: 1 MHz	The serial clock connects both the microcontroller to the Temperature and Humidity Sensor block and the Radio Data Transfer System block, so this property has to take a value compatible with all three blocks.	The Arduino Nano includes an ATmega328 8-bit microcontroller [3]. According to the datasheet of the ATmega328 [5]: ● The ATmega328 runs at 16 MHz, so a maximum SPI transfer speed of 8 MHz. ● The maximum SPI speed is half the frequency of oscillation of the CPU [5, Table 18-5] while the minimum is 128th of the

		CPU speed at 125 KHz [5, Table 18-5].
Protocol: 4-Wire (SS, SCK, MISO, MOSI)	The microcontroller sends and receives serial communication data but a three-wire connection would only allow a unidirectional flow of information. A four-wire setup allows for both the slave and master to communicate.	The Arduino Nano includes an ATmega328 8-bit microcontroller [3]. According to the datasheet of the ATmega328 [5]: ● Three-wire synchronous data transfer (MOSI, MISO, and SCK) is supported. ● The ATmega238 has ten other digital pins available for I/O communication which can drive the slave select pin operation.
Protocol: Serial Peripheral Interface	A system communicating over a Serial Peripheral Interface can support multiple slaves with a single master by consuming one additional I/O pin per connection. This block design requires two peripheral connections, so a serial peripheral interface can work with both simultaneously.	The Arduino Nano includes an ATmega328 8-bit microcontroller [3]. According to the datasheet of the ATmega328 [5]: ● [5, Table 13-3] includes the alternate functions of PORTB. This includes dedicated SPI hardware.

Table 4.6-2: mcrcntrllr_hmdtytmp_snsr_dcpwr: Output

Inominal: 1.2 mA	The sensor will draw this amount of current on average, so the Arduino Nano has to be able to supply it.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Ipeak: 2 mA	The sensor breakout board could theoretically draw this amount of current in the harshest environmental conditions operating at the highest voltage, so the Arduino Nano has to have the capability of withstanding a current surge of this magnitude.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].

Vmax: 5 V	The Arduino Nano pinout includes a 5-volt supply pin on the breakout board. This has a voltage regulator from input to output if supplied with more than five volts on Vin or a direct connection to the Universal Serial Bus input at the supplied voltage.	Looking at the schematic of the Arduino Nano [3], the VIN pin connects to a LM1117IMPX-5.0 Linear voltage regulator. The datasheet of the LM1117 suggests the device will output a typical value of 5 volts [7].
Vmin: 3.3 V	The Arduino Nano pinout includes a 3.3-volt supply pin on the breakout board. This has an additional 3.3-volt regulator between the five-volt supply on the board to bring it to the correct output.	Looking at the schematic of the Arduino Nano [3] an FT232R is powered by a five-volt supply from either the USB input or the VIN pin input. The datasheet for the FT232R [8] details the output of a 3.3-volt supply, which is in turn connected to the 3.3-volt output on the Arduino Nano.

Table 4.6-3: mcrcntrllr_wtr_lvl_snsr_dcpwr: Output

Inominal: ~2.94 mA	The sensor will draw this amount of current on average, so the Arduino Nano has to be able to supply it.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Ipeak: 4.2 mA	The sensor breakout board could theoretically draw this amount of current in the harshest environmental conditions operating at the highest voltage, so the Arduino Nano has to have the capability of withstanding a current surge of this magnitude.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Vmax: 5V	The Arduino Nano pinout includes a 5-volt supply pin on the breakout board. This has a voltage regulator from input to output if supplied with more than five volts on Vin or a direct connection to the Universal Serial	Looking at the schematic of the Arduino Nano [3], the VIN pin connects to a LM1117IMPX-5.0 Linear voltage regulator. The datasheet of the LM1117 suggests the device will output a typical value of 5 volts [7].

	Bus input at the supplied voltage.	
Vmin: 3.3V	The Arduino Nano pinout includes a 3.3-volt supply pin on the breakout board. This has an additional 3.3-volt regulator between the five-volt supply on the board to bring it to the correct output.	Looking at the schematic of the Arduino Nano [3] an FT232R is powered by a five-volt supply from either the USB input or the VIN pin input. The datasheet for the FT232R [8] details the output of a 3.3-volt supply, which is in turn connected to the 3.3-volt output on the Arduino Nano.

Table 4.6-4: mcrcntrllr_wnd_spd_snsr_dcpwr: Output

Inominal: 25mA	The sensor will draw this amount of current on average, so the Arduino Nano has to be able to supply it.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Ipeak: 40mA	The sensor breakout board could theoretically draw this amount of current in the harshest environmental conditions operating at the highest voltage, so the Arduino Nano has to have the capability of withstanding a current surge of this magnitude.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Vmax: 5V	The Arduino Nano pinout includes a 5-volt supply pin on the breakout board. This has a voltage regulator from input to output if supplied with more than five volts on Vin or a direct connection to the Universal Serial Bus input at the supplied voltage.	Looking at the schematic of the Arduino Nano [3], the VIN pin connects to a LM1117IMPX-5.0 Linear voltage regulator. The datasheet of the LM1117 suggests the device will output a typical value of 5 volts [7].
Vmin: 4V	The Arduino Nano pinout includes a 3.3-volt supply pin on the breakout board. This has an additional 3.3-volt regulator between the five-volt supply on the board to bring it to the correct	Looking at the schematic of the Arduino Nano [3] an FT232R is powered by a five-volt supply from either the USB input or the VIN pin input. The datasheet for the FT232R [8] details the output of a

	output.	3.3-volt supply, which is in turn connected to the 3.3-volt output on the Arduino Nano.

Table 4.6-5: mcrcntrllr_rd_dt_trnsfr_systm_comm: Output

Datarate: Serial Clock Speed: 1 MHz	The serial clock connects both the microcontroller to the Temperature and Humidity Sensor block and the Radio Data Transfer System block, so this property has to take a value compatible with all three blocks.	The Arduino Nano includes an ATmega328 8-bit microcontroller [3]. According to the datasheet of the ATmega328 [5]: ● The ATmega328 runs at 16 MHz, so a maximum SPI transfer speed of 8 MHz. ● The maximum SPI speed is half the frequency of oscillation of the CPU [5**,** Table 18-5] while the minimum is 128th of the CPU speed at 125 KHz [5, Table 18-5].
Protocol: 4-Wire (SS, SCK, MISO, MOSI)	The microcontroller sends and receives serial communication data but a three-wire connection would only allow a unidirectional flow of information. A four-wire setup allows for both the slave and master to communicate.	The Arduino Nano includes an ATmega328 8-bit microcontroller [3]. According to the datasheet of the ATmega328 [5]: ● Three-wire synchronous data transfer (MOSI, MISO, and SCK) is supported. ● The ATmega238 has ten other digital pins available for I/O communication which can drive the slave select pin operation.
Protocol: Serial Peripheral Interface	A system communicating over a Serial Peripheral Interface can support multiple slaves with a single master by consuming one additional I/O pin per connection. This block design requires two peripheral connections, so a	The Arduino Nano includes an ATmega328 8-bit microcontroller **(include source)**. According to the datasheet of the ATmega328 **(add source)**: ● [**datasheet reference number,** Table 13-3]

	serial peripheral interface can work with both simultaneously.	includes the alternate functions of PORTB. This includes dedicated SPI hardware.

Table 4.6-6: wtr_lvl_snsr_mcrcntrllr_comm Input

Other: Sampling Rate: 15 seconds	The Water Level Sensor block outputs a continuous-time reading of the water level, so the sampling rate only affects how much data the microcontroller needs to store and how abnormal readings skew transmitted data. At fifteen-second intervals, a single bad data read would have less than a one percent change on the data set. It also reduces the amount of CPU time required to operate the sensor, although realistically when the sampling rate is in the range of seconds there is a negligible effect on the processing power.	The ATmega328 has a 16-bit Timer/Counter capable of triggering interrupts at configurable times [5]. This forgoes any wasted CPU time while still allowing for relative accuracy with the data. Additionally only the water level sensor readings and the wind speed sensor readings require any filtering, so the system can dedicate several hundred bytes (each reading only requiring two bytes individually) to leveling sensor data.
Vmax: 5V	The largest possible value for the sensor output is the supplied voltage at five volts. This ensures there are no input values outside the range of the sensor interface.	According to the ATmega328 datasheet [5] the Analog to Digital converter pins measure between 0 and Vcc, which will use the 5-volt supply from the USB connection.
Vmin: 0V	The lowest possible output of the sensor is a short to ground, which would have a voltage reading of zero volts.	According to the ATmega328 datasheet [5] the Analog to Digital converter pins measure between 0 and Vcc, which will use the 5-volt supply from the USB connection.

Table 4.6-7: wnd_spd_snsr_mcrcntrllr_comm: Input

Other: Sampling Rate: 15 seconds	The Wind Speed Sensor block outputs a continuous-time reading of the water level, so the sampling rate only affects how	The ATmega328 has a 16-bit Timer/Counter capable of triggering interrupts at configurable times [5]. This

	much data the microcontroller needs to store and how abnormal readings skew transmitted data. At fifteen-second intervals, a single bad data read would have less than a one percent change on the data set. It also reduces the amount of CPU time required to operate the sensor, although realistically when the sampling rate is in the range of seconds there is a negligible effect on the processing power.	forgoes any wasted CPU time while still allowing for relative accuracy with the data. Additionally only the water level sensor readings and the wind speed sensor readings require any filtering, so the system can dedicate several hundred bytes (each reading only requiring two bytes individually) to leveling sensor data.
Vmax: 5V	The largest possible value for the sensor output is the supplied voltage at five volts. This ensures there are no input values outside the range of the sensor interface.	According to the ATmega328 datasheet [5] the Analog to Digital converter pins measure between 0 and Vcc, which will use the 5-volt supply from the USB connection.
Vmin: 0.5V	The lowest possible output of the sensor is 0.5 volts above ground.	According to the ATmega328 datasheet [5] the Analog to Digital converter pins measure between 0 and Vcc, which will use the 5-volt supply from the USB connection.

Table 4.6-8: pwm_chrg_cntrllr_mcrcntrllr_dcpwr: Input

Inominal: 80 mA	The nominal draw of each of the three output sensors plus the nominal draw of the ATmega328 should reach around this amount, depending on the voltage applied by the charge controller.	Looking at the schematic of the Arduino Nano [5], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Ipeak: 150 mA	This is well beyond the maximum draw of all sensors plus the ATmega328, but in the event of an added sensor by future groups this gives room to avoid an entire hardware redesign.	Looking at the schematic of the Arduino Nano [3], the USB voltage input connects to a SS1P3L diode with a maximum forward current of 1.0 amps [6].
Vmax: 5V	The system provides power to the microcontroller via the PWM	Looking at the schematic of the Arduino Nano [3], two components

	Charge Controller block, which outputs on a USB connection expected at 5 volts.	on the board require the 5-volt internal supply. This includes the FT232R and the ATmega328P. The FT232R has a maximum voltage supply of 5.25 volts [8] and the ATmega328P has a maximum operating voltage of 5.5 volts [5], so both can operate at a maximum of 5 volts.
Vmin: 4.7V	The USB connection will not always operate exactly at five volts, so including a small tolerance in the design allows for the Charge Controller block to operate within a small tolerance.	Looking at the schematic of the Arduino Nano [3], two components on the board require the 5-volt internal supply. This includes the FT232R and the ATmega328P. The FT232R has a minimum voltage supply of 4.0 volts [8] and the ATmega328P has a maximum operating voltage of 2.7 volts [5], so both can operate at a minimum of 4.7 volts.

4.6.5 Verification Plan

To reduce the number of overall tests required to verify all eight interfaces of this block combine all DC power output interfaces, run both the Serial Peripheral Interface communications simultaneously, and sample on both analog pins in turn. This effectively reduces the number of interfaces from eight to four while preserving all individual properties of the entire block.

4.6.5.1 Initialization

1. Attach an Arduino Nano to either a PCB with dedicated connectors or to a breadboard. Ensure room exists for multiple connections to the SCK, MOSI, and MISO pins. Grab an independent DC power supply with measured output values, a digital multimeter, a laptop with a USB connection, an oscilloscope with a probe, and an electronic load.

2. Load a test program to the Arduino Nano and open the serial monitor on the laptop.

4.6.5.2 Verifying Combined Serial Data Communication Interfaces

1. Attach two devices capable of two-way Serial Peripheral Communication to the Arduino Nano. Both devices should share a connection to the MOSI, MISO, and SCK pins while having distinct digital output pins responsible for the slave select signal.

2. Visually verify the four-wire connection is complete.

3. Attach an oscilloscope to the SCK pin. Set the oscilloscope to trigger on a voltage of about 3 volts to avoid any noise from the block. Visually verify on the oscilloscope a clock signal of 1 MHz is present on the line.

4. Send and receive data via the Serial Communication Interface. Print the data to the serial monitor. If data can successfully be sent to the devices from the Arduino Nano and received from the devices by the Nano, then the verification for both interfaces is complete.

4.6.5.3 Verifying Both Analog Sampling Interfaces

1. Attach the DC power supply to two analog pins on the Arduino Nano. Load a testing program capable of reading the analog inputs from both pins and then printing out the data to the serial monitor every fifteen seconds.

2. Vary the voltages on both pins starting at zero volts and up to five volts. The resolution of the analog-to-digital converter is about five millivolts, so test values should vary at least ten millivolts to ensure no sampling errors.

3. Verification of these interfaces includes a visual inspection of the serial monitor to ensure values print every fifteen seconds and can read both the minimum value of zero volts and the maximum value of five volts.

4.6.5.4 Verifying Combined Output Power Interface

1. Disconnect the Arduino Nano from the USB interface. Attach the DC power supply to the VIN pin and provide at least seven volts from the DC power supply. Measure the 5-volt pin and the 3.3-volt pin with the digital multimeter. Since all three DC power outputs have a minimum at or below 3.3 volts and all have a maximum of five volts a visual inspection of the readings from the digital multimeter verifies correct output interfaces.

2. Connect the 5-volt pin to an independent load of at least 50 milliamps. Verification of the combined peak current occurs if the LED on the Arduino Nano remains on (ensuring the board is still on and functioning).

4.6.5.5 Verifying Input Power Interface

1. Repeat the process for the output power interface verification but connect an ammeter in series with the VIN pin. Watch the current draw and ensure it remains below the peak and in the range of the nominal for current draw verification. The input voltage will also be complete if the board is still operating because the DC power supply should give a voltage differential of seven volts (two volts above the maximum).

2. Disconnect the Arduino Nano from all external connections. Measure the voltage provided by a USB supply. Plug in the Arduino Nano to the USB supply and ensure operation. If the LED is still blinking, then the minimum voltage is also verified.

4.6.6 References & File Links

[1] Arduino, “Arduino Reference,” *Arduino. cc*, 2019. https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

[2] Arduino, “Arduino & Serial Peripheral Interface (SPI) | Arduino Documentation,” *docs. arduino.cc*, 2023. https://docs.arduino.cc/learn/communication/spi

[3] Arduino, “Arduino® Nano,” Jul. 2023. Accessed: Mar. 10, 2023. [Online]. Available: https://docs.arduino.cc/static/e084a9b942d5c43ee5a38bbaac9d99ac/A000005-datasheet.pdf

[4] Arduino, “Arduino® UNO R3,” Jul. 2023. Available: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[5] Atmel, “ATmega328P Datasheet,” 2015. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[6] Vishay, “SS1P3L, SS1P4L Vishay General Semiconductor Low V F High Current Density Surface Mount Schottky Barrier Rectifiers,” 2012. Accessed: Mar. 10, 2023. [Online]. Available: https://www.tme.eu/Document/861b13b13a2be75a1db38cf23ba0d8e0/SS1P3L-M3-84A.pdf

[7] Texas Instruments, “LM1117 800-mA Low-Dropout Linear Regulator Datasheet,” *https://media.digikey.com/pdf/Data%20Sheets/Texas%20Instruments%20PDFs/LM1117.pdf*, 2016. https://media.digikey.com/pdf/Data%20Sheets/Texas%20Instruments%20PDFs/LM1117.pdf (accessed Mar. 10, 2023).

[8] FTDI Chip, “FT232R USB UART IC Datasheet Version 2.15,” 2020. Available: https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

4.6.7 Revision Table

Date	Name	Revision Made
5/10/2023	Grace	Finished section 4.6 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.6.
3/10/2023	Grace	Edited section 4.6 (grammar, spelling, etc.) and relabeled figures.
3/10/2023	Blake	Added citations and completed the References and File Links section.
3/10/2023	Blake	Completed the Interface Definitions section.
3/9/2023	Blake	Completed the Verification Plan section.
3/9/2023	Blake	Began filling out the interface definition table.
3/9/2023	Blake	Created the Revision Table and began filling it out, continually updating it as changes are made.
3/8/2023	Blake	Began writing the Interface Validation section to address the inputs and outputs to the block including the value itself, why the value was chosen, and how the block can successfully support those values.
3/8/2023	Blake	Wrote the General Validation section to address how the block design can meet the requirements.
3/8/2023	Blake	Inserted Figure 1 and Figure 2 as images to illustrate the design of the block, then added the text to explain why the block includes the design itself.
3/7/2023	Blake	Copied the format from the template, including the General Description section

4.7 Enclosure

4.7.1 Description

The enclosure is a sealed container that is weatherproof and animal-proof. The container is 4.7 inches [in] tall, with a length of 10in, and a width of 7.9in. This is ideal for our needs since the tallest item, the battery, is 4.02in tall and 5.94in long. The enclosure also has twelve holes, three on each side. This will allow us to place the sensors that require environmental inputs outside the enclosure and the rubber seals will maintain the enclosure's weatherproofing.

The box is made of silicon plastic and has an ingress protection (IP) rating of 65. IP65 is the highest level of dust protection and indicates how waterproof the box is. In this case, the box could withstand water jets hitting it from all sides and remain waterproof. While this only applies to low-pressure water jets, this equates to rain and flooding conditions making it perfect for our needs. Cost is a major concern for our project customer, so we needed to find a premade option that was easily accessible in Thailand and cheap to replicate if necessary. This box can be found on Amazon for $22.99 and it meets all our projects and customers' needs.

4.7.2 Design

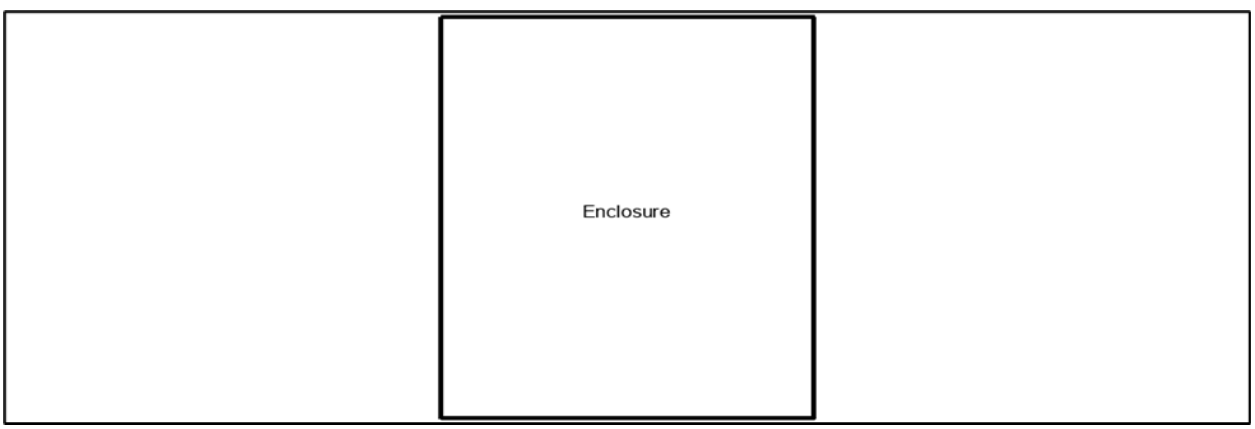


Figure 4.7-1: Black Box view of Enclosure block

The enclosure needs to be able to house and protect our backup battery, radio data transfer system, the microcontroller managing the sensors, and the power charge controller. Due to the size of the battery and PCC, the enclosure needed to be at least 4.02in tall, 6in long, and 7in wide. The enclosure we chose is a silicon plastic box with the outer dimension: 10in length, 7.9in width, and 4.7in height (including the lid). The inner dimensions of the box are 9.6in length and 7.6in width. The dimensions for both the inner and outer sides of the enclosure can be seen in Figures 3 & 4 below. Figure 2 below shows the mechanical drawing provided by the company of the enclosure. To be animal-proof the enclosure we order includes 4 screws so we can seal the enclosure's lid tightly. This will also contribute to the waterproofing.

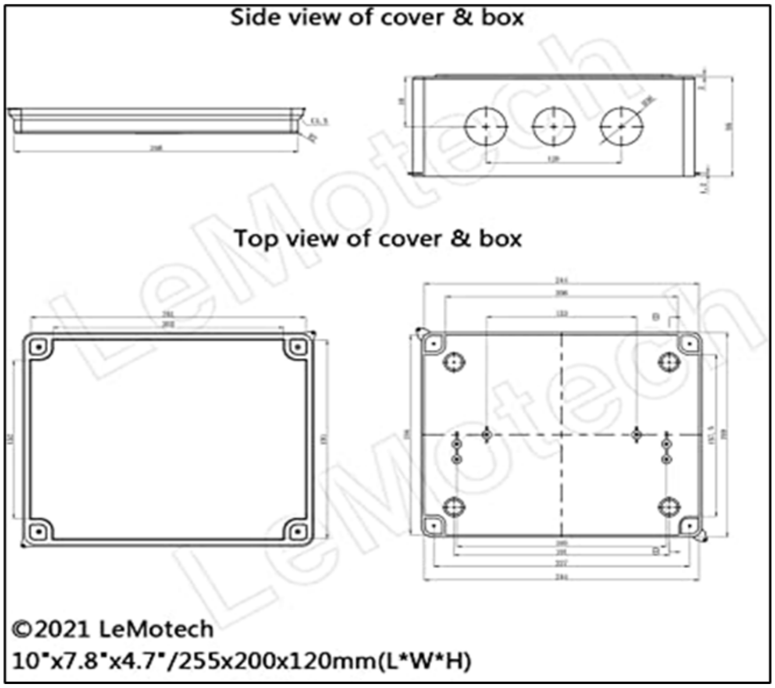


Figure 4.7-2: Mechanical drawing of the enclosure

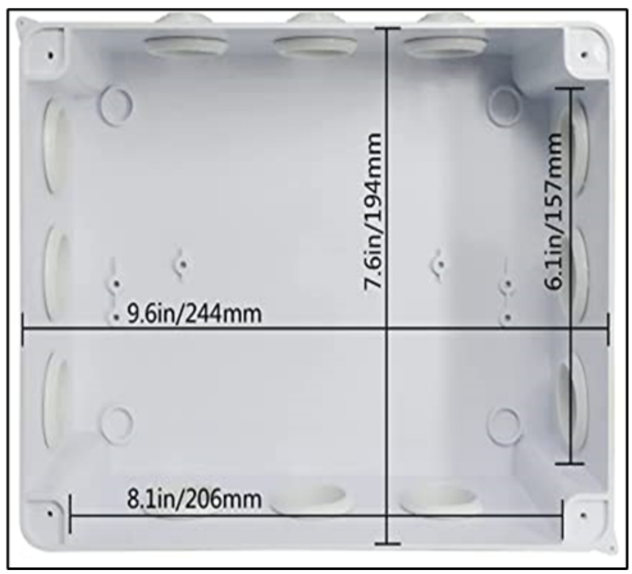


Figure 4.7-3: Outer dimensions of enclosure

Figure 4.7-4: Inner dimensions of the enclosure

4.7.3 General Validation

The enclosure has an IP65 rating meaning it will survive flooded conditions as well as prevent dust and other contaminants from entering the enclosure. IP65 is the highest level for ingress protection. This is ideal for our system since the rice patty fields have a high chance of flooding. Plastic is ideal for the humidity and heat in the area of Thailand where we are designing the box for. A metal box would heat up too much and there would be a chance of condensation in a metal container. Silicon is also cheaper than other materials and it would be easy for our project customer to purchase more enclosures like this.

The enclosure also has twelve holes filled with rubber stoppers. This allows us to connect our sensors that require environmental to the microcontroller without sacrificing the weatherproofing of the enclosure. There are also four screws that are used on the corners of the enclosure to prevent animals from getting into the box and it also contributes the waterproofing.

4.7.4 Interface Validation

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

Table 4.8-1 Parameters of enclosure

Dimensions	The outer dimensions are: ● 10in length ● 7.9in width ● 4.7in height The inner dimensions are: ● 9.6in length ● 7.6in width These dimensions will allow us to fit our system perfectly in the box and allow	The backup battery is the tallest object in the enclosure, while the power charge controller is the widest object in the system. The battery is 4.02in tall, and the added width of the PCC and the battery requires about 7in. The minimum length should be about 6in as well. This enclosure exceeds our needs and the extra space should allow for neat wire management and ease of repair.
Weatherproof/Animal-proof	The enclosure has an ingress protection (IP) level of 65. IP65 is the highest level offered and is ideal for our system's needs.	IP65 is the maximum level for dust protection and can handle water jets hitting all the sides of the enclosure at low pressure [1]. This is ideal for the flooding conditions we are designing for. This level and quality of the enclosure also contribute to the

		animal-proof aspect of the enclosure.
Cost	Despite the size and quality of the box, it is relatively cheap in price. This is ideal since the customer wants a low-cost design.	The cost is $22.99 on Amazon. The box is made with silicon plastic, four screws, and included twelve rubber stoppers.

4.7.5 Verification Plan

Waterproof:

1. Place a piece of cardboard inside the enclosure.
2. Place the four screws that came with the enclosure in their dedicated holes.
3. Use a Philips screwdriver to tighten the screws and make sure to tighten them in the following order:
    a. Corner 1, corner 3, corner 2, and then corner 4
4. Tighten the screws until you cannot tighten them any further.

    Note: if you hear any creaking when tightening the screws down, stop immediately. The screw is tight enough.
5. Pour 1 liter [L] of water onto the top of the box.
6. Pour 1L of water onto the left side of the box.
7. Pour 1L of water onto the right side of the box.
8. Pour 1L of water onto the bottom of the box.
9. Unscrew the box's lid and check if the piece of cardboard is wet.
10. Dry cardboard and no water in the box indicate that the enclosure is in fact

Animal-proof:

1. Place the four screws that came with the enclosure in their dedicated holes.
2. Use a Philips screwdriver to tighten the screws and make sure to tighten them in the following order:
    a. Corner 1, corner 3, corner 2, and then corner 4
3. Tighten the screws until you cannot tighten them any further.

    Note: if you hear any creaking when tightening the screws down, stop immediately. The screw is tight enough.
4. Place the enclosure outside for one 24-hour period.
5. Take a photo of the box when you start the test.
6. Once the 24-hour period has passed, take another photo of the enclosure.
7. Open the enclosure and see if any dust or bugs made it into the box.
8. Take another picture to prove there are no bugs or dust inside the enclosure.

4.7.6 References & File Links

[1] ZLT Electrical, “What is IP65? | the electrical counter,” *The Electrical Counter*. [Online]. Available: https://www.electricalcounter.co.uk/what-is-ip65. [Accessed: 13-Mar-2023].

4.7.7 Revision Table

Date	Name	Revision Made
5/10/2023	Grace	Finished edits to section 4.7 - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.7
3/12/2023	Grace	Added section 4.7

4.8 PWM Charge Controller

4.8.1 Description

The power charge controller (PCC) will manage the DC input from the solar panel and ensure the panel does not overcharge the backup battery or send too much power to the main system. The PCC is connected to the microcontroller and the radio data transfer system. It can take in a maximum voltage of 18 volts [V], a peak current of 10 amps [A], a nominal current of 0.56A, and a minimum voltage of 6V. This PCC arrived in the same Eco-worthy 10W 12V Solar Panel kit and it can be found for $33.99 on Amazon.

4.8.2 Design

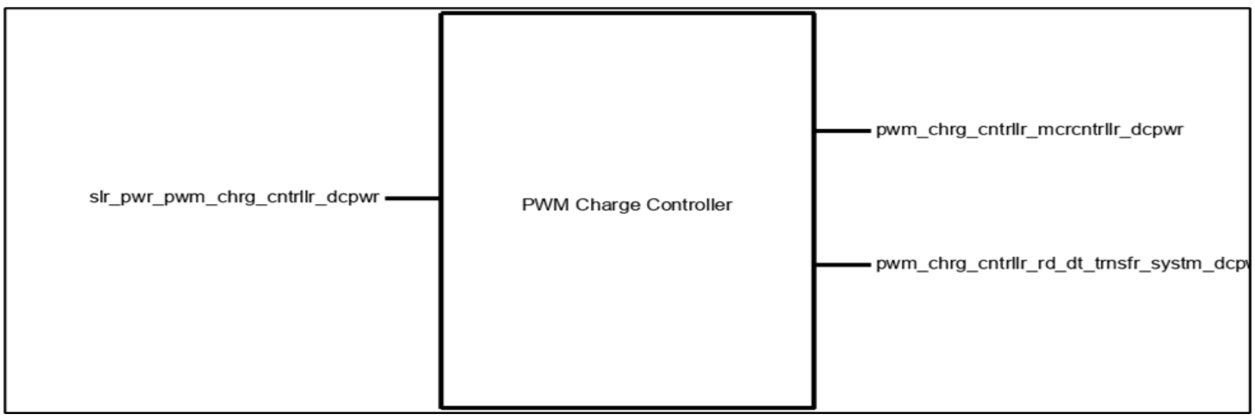


Figure 4.8-1: Black Box view of Power Charge Controller Block

This block represents the power charge controller that we are using in this project. The PCC can take in a maximum voltage of 18V and a peak current of 10A. The PCC has three interfaces as seen in Figure 4.8-1 above. The first interface, slr_pwr_pwm_chrg_cntrllr_dcpwr, is the power supplied by the solar panel (DC power). The second interface, pwm_chrg_cntrllr_mcrcntrllr_dcpwr, is where the microcontroller is plugged into the USB port that outputs 5V and 1.2A. The third interface, pwm_chrg_cntrllr_rd_dt_trnsfr_systm_dcpwr, is where the radio data transfer system will be connected. The radio data transfer system is connected to the same load position as the Arduino Nano which supplies 5V and 1.2A.

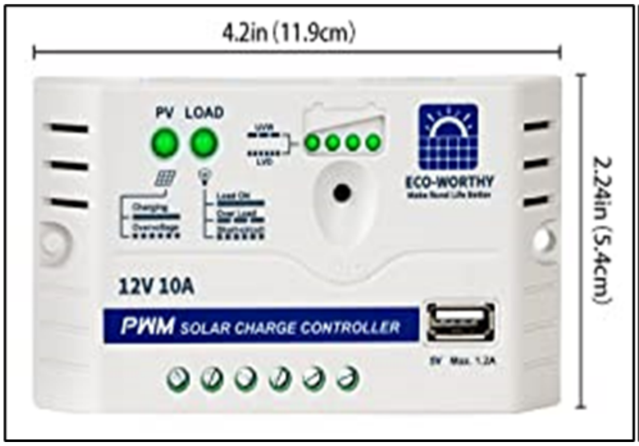


Figure 4.8-2: Front view of PCC

Figure 4.8-3: Back view of PCC

The PCC has four connection points as seen in Figure 4.8-2 above: Photovoltaic (PV), battery, load, and USB connection. The solar panel is connected to PV, a 12V rechargeable and Sealed Lead Acid battery is connected to the battery port, and the load will be connected to the radio data transfer system. Instead of connecting our system to the load port, we have opted to use the USB load port, as seen in Figure 4.8-2. The Arduino can withstand the 1.2A and 5V output of that load port better than the load port (indicated by the light bulb symbol in Figure 4.8-2) at the base of the PCC. The dimensions of the PCC are 4.2in wide and 2.24in long, which can also be seen in Figure 4.8-2 above.

The PCC has three sets of LEDs that indicate various operations in the system. These LEDs and descriptions can be seen in Figure 2 above. If the PV LED is solid, the solar panel is connected, charging the battery, and powering the load. If the PV LED is blinking, the panel is supplying too much voltage/current and the PCC is opening the circuit to prevent damage to the system. The load LED indicates whether the PCC is allowing the load to be powered. When the load LED is solid the load is powered, when the load LED is blinking slowly the load is experiencing an overload which puts the system in danger of short-circuiting, and if the load LED is blinking quickly the PCC shuts off power to the load to avoid a short-circuit. The final set of LEDs indicates the battery type and how the PCC charges the battery. For this system, all four LEDs should be lit to indicate a Sealed Lead Acid battery.

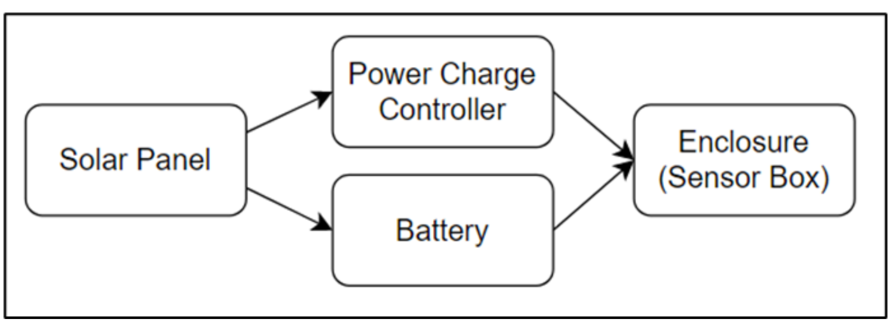


Figure 4.8-6: Basic block diagram showing how Solar Panel, PCC, and Battery are connected to the main system

Figure 4.8-6 above shows in a block diagram how the PCC and battery are placed in the system. The solar panel will be connected to the PCC and a 12V battery which are connected in parallel.

Figure 4.8-5: Front view of 12V rechargeable Sealed Lead Acid Battery

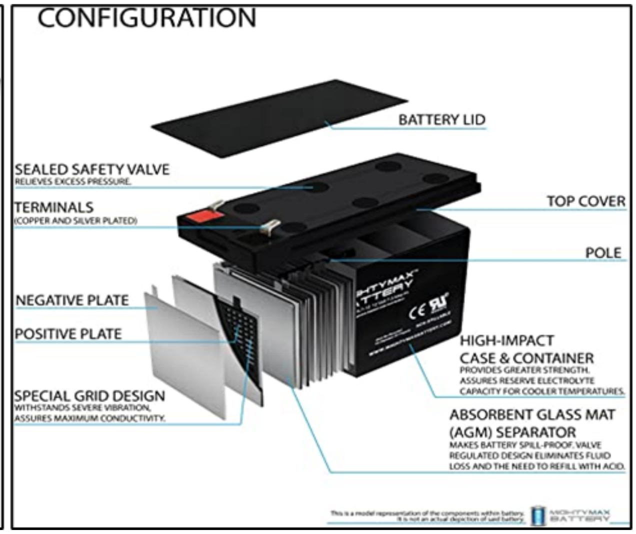


Figure 4.8-4: Configuration for 12V Rechargeable Sealed Lead Acid Battery

The battery we chose is a 12V rechargeable Sealed Lead Acid battery with 7 amp-hours [Ah] (Figures 4.8-4 & 4.8-5). The battery will remain charged throughout the day and regulate the current in the system along with the power charge controller. The battery will take over as the main power supply for the system during the night, as previously mentioned. The battery's dimensions are as follows: height 4.02 inches [in], length 5.94in, and width 2.56in. This battery is ideal for our project because it is cheap ($19.99 on Amazon), it can stay charged for up to 14 hours, and it will fit in our enclosure perfectly. The reason why a 14-hour charge is so important is that charging a completely discharged battery takes much longer than a battery with some charge left over.

4.8.3 General Validation

This PCC is the best option for our system's design because each sensor box will have sensitive components always connected. A power surge would ultimately cause irreparable damage to the system if the circuit is not opened immediately. The PCC will be able to stop a power surge from causing any damage by opening the circuit [1]. The charge controller also prevents current leakage from going to the panel during the night.

When the sun sets, the solar panel will put out less charge which will cause the battery to take over and equalize the system through the PCC. If the battery were connected to the solar panel directly, it would cause a leakage of current going to the solar panel as that would be the path of least resistance [2]. A diode in the PCC placed between the battery and the solar panel prevents this leakage from occurring, so the battery will continue to power the sensor box.

We chose a Sealed Lead Acid battery in favor of a Nickle Cadmium (NiCad) battery, which is generally cheaper because NiCad suffers from the memory effect. The memory effect is when the battery "remembers" how much power was drawn and then performs best only during that certain power draw [3]. Avoiding the memory effect will result in a longer lifespan for our battery, making the Lead Acid battery the cheaper option over time. The reason why the price is important is due to our project customer requested the cheapest build possible for our system, without sacrificing functionality.

4.8.4 Interface Validation

Table 4.8-1: slr_pwr_pwm_chrg_cntrllr_dcpwr: Input

Interface Property	**Why is this interface this value?**	**Why do you know that your design details <u>for this block</u> above meet or exceed each property?**
Inominal: 0.56A	The power charge controller can accept a maximum of 10A, our nominal current value is significantly less than that. This value will support the charge controller, but not overload the controller or the sensor box.	Judgment: We need a current that can supply the 5V and 1.2A USB output without overloading the sensor box. This amperage meets our system's needs and if it were to be lowered, the battery backup would compensate for the loss.
Ipeak: 10A	The power charge controller can accept a 10A maximum. This value accounts for a sudden surge in power or anything else that could cause a spike in current.	Judgment: The current peak value is high enough that the system will not be damaged in case of a surge. Environments can be unpredictable and having a peak current value of 10A ensures the safety of our system. If this value was to be reached, it should be noted that the solar panel would most likely be damaged, which could have also caused the surge.

Other: 10W (Max Power)	The panel's power value is 10W maximum because our customer wanted a panel that was cheap, but able to effectively power the system.	Judgment: Based on our system's needs (5V input voltage to the microcontroller) a 10W panel with a 12V output is more than enough to support the sensor box. As shown in Figure 4, the panel has a power maximum of 10W and a maximum output voltage of 18V. These values can be managed by the power charge controller which accepts the maximum voltage and a current up to 10A.
Vmax: 18V	Each cell in an average solar panel outputs about 0.46V per cell, and our current panel has about 40 cells [4]. Therefore, the maximum voltage out is about 18V. The system will take the 18V input to charge a battery and power the rest of the system which operates on a 5V input.	Judgment: The system requires a 12V input, this panel has a maximum of 18V which exceeds our expectations. This makes our choice of panel ideal for the sensor box and power charge controller.
Vmin: 6V	This is the approximate minimum value that the solar panel can output and continue to power our system with the use of the battery. The minimum voltage needed to run our system is 5V, so 6V exceeds our needs for the sensor box. The battery will manage the current value since it sits in parallel with the charge controller connected to the panel.	Judgment: 6V would be the lowest possible voltage that the system could run on. This value would be shown when the solar panel is not in full sun or when the sun has set. It should be noted that the absolute minimum would be 0V (absolute darkness), but that condition will not be met in the environment for which we are building this system.

Table 4.8-2: pwm_chrg_cntrllr_mcrcntrllr_dcpwr: Output

Inominal: 80 mA	The microcontroller can handle a peak current of 150mA, but the nominal operating value is 80mA. This current value will ensure the microcontroller does not receive too much current, resulting in damage to the microcontroller. This is also an ideal value for the microcontroller because all the sensors connected to it will not draw more current than the nominal value.	Judgment: We need a current that can supply the microcontroller with enough power to run all the sensors connected to the Arduino Nano. This amperage meets our system's needs without risking any damage to the Arduino Nano. The PCC will supply 1.2A maximum from the USB output, which exceeds the controller's needs.
Ipeak: 150 mA	The microcontroller can accept a 150mA maximum. This value accounts for a sudden surge of power or anything else that could cause a spike in current.	Judgment: The current peak value is high enough that the system will not be damaged in the event of a power surge. The peak current value is 70mA above our nominal current value. This should ensure the safety of our microcontroller in case of a power surge. The PCC will supply 1.2A maximum from the USB output, which exceeds the controller's needs.
Vmax: 5V	The microcontroller can accept a 5V maximum without causing damage to the Arduino Nano.	Judgment: The microcontroller requires at least 4.7V to operate, but no more than 5V. The voltage supplied by the PCC is exactly 5V, making it the perfect choice for the Arduino Nano.

Vmin: 4.7V	The microcontroller will continue to operate normally at 4.7V, the voltage minimum for an Arduino Nano.	Judgment: 4.7V would be the lowest possible voltage that the Arduino Nano could receive and still function. If there is a pause between the battery taking over for the solar panel at sunset, this would be the lowest value reached during that moment.

Table 4.8-3: pwm_chrg_cntrllr_rd_dt_trnsfr_systm_dcpwr: Output

Inominal: 30mA	The radio data transfer system can accept a maximum of 120mA, our nominal current value is significantly less than that. This value will support the radio system's power needs without risking damage to the system.	Judgment: We need a current that can supply the radio data transfer system. This amperage meets our system's needs without risking any damage to the data transfer system. The PCC will supply 1.2A maximum from the USB output, which exceeds the system's needs.
Ipeak: 120mA	The radio data transfer system can accept 120mA maximum. This value accounts for a sudden surge in power or anything else that could cause a spike in current, like the exact moment when the system is plugged in.	Judgment: The current peak value is high enough that the system will not be damaged in the event of a power surge. The peak current value is 90mA above our nominal current value. This should ensure the safety of our radio system in case of a power surge. The PCC will supply 1.2A maximum from the USB output, which exceeds the system's needs.

Vmax: 5V	The radio data transfer system can accept a 5V maximum without causing damage to the system.	Judgment: The radio data transfer system requires at least 3V to operate, but no more than 5V. The voltage supplied by the PCC is exactly 5V, making it the perfect choice for the radio system.
Vmin: 3V	The radio data transfer system will continue to operate normally at 3V, the voltage minimum for this system. Note: Normal function refers to the fact that packets are not being lost during the data transmission, at least not due to power supply.	Judgment: 3V would be the lowest possible voltage that the radio data transfer system could receive and still function. If there is a pause between the battery taking over for the solar panel at sunset, this would be the lowest value reached during that moment.

4.8.5 Verification Plan

1. Create a test load for this process. This can be done with a breadboard, a resistor, and an LED. Connect the resistor in series to the LED and the load. Make sure all connections on the breadboard are secure.

   Note: The test load can be used to show that the PCC will supply the combined values for nominal current. So, these steps will show you how to test for an Inom of 0.11A.

2. Place the solar panel with a clear view of the sun (it will also work in cloudy conditions but the sunnier the better).
3. Connect the battery to the battery ports located at the base of the PCC.
4. Connect the test load to the load ports located at the base of the PCC.
5. Connect the solar panel to the PV ports located at the base of the PCC.

   Steps 3-5 must be followed, do not connect to the PCC in any other order!

6. Once the LED indicators on the PCC turn on, use a multimeter to check the voltage and current values of the system.

7. Turn on your multimeter and make sure it is set to read in the 20-200V range.
8. Probe the screws used to secure the PV port, and ensure the solar panel is supplying the 18V maximum.
9. Probe the screws used to secure the battery port, and ensure the battery is charging.
10. Probe the screws used to secure the load port, and ensure the PCC is outputting the 12V maximum to the test load.
11. Change the multimeter to measure current. Make sure your probe leads are connected to the high current port on the meter, otherwise you will blow the fuse in the meter.
12. Probe the screws used to secure the PV port, and verify the nominal current of 0.56A is being supplied by the solar panel.
13. Probe the screws used to secure the load port, and verify the nominal current of 0.11A is being supplied to the test load.
14. Once you have all nominal values and your battery has been charged for a minimum of 4 hours, cover the solar panel with a cloth or an opaque material to simulate darkness (night-time).
15. Set your multimeter to measure voltage and make sure it is set to read in the 20-200V range.
16. Probe the screws used to secure the PV port, and ensure the 6V operating minimum is reached by the solar panel.
17. Probe the screws used to secure the battery ports, and verify that the battery is now supplying the 12V maximum to the PCC.
18. Probe the screws used to secure the load port, and verify that the load is still receiving the 12V maximum.
19. Change the multimeter to measure current. Make sure your probe leads are connected to the high current port on the meter.
20. Probe the screws used to secure the load port, and verify the nominal current is still being supplied to the test load.
21. Probe the screws used to secure the PV port, there should be little to no current being supplied by the panel at this point.

    Note: If you see a negative current value, and you are certain your probes are connected correctly, then there is a leakage current going from the battery to your panel. You should disconnect your PCC because it is likely faulty.

22. Disconnect the Solar Panel from the PV port on the PCC.

23. Disconnect the test load from the load port on the PCC.

24. Disconnect the battery from the battery port on the PCC.

Steps 22-24 must be followed, do not disconnect the PCC in any other order!

4.8.6 References & File Links

[1] Eco-Worthy, “Getting started with solar power,” *Getting Started with Solar Power*. [Online]. Available: https://www.eco-worthy.com/pages/getting-started-with-solar-power. [Accessed: 10-Mar-2023].

[2] N. A. Z. S. E. Posted in Batteries & Charging, “Solar charge controller basics,” *Solar Charge Controller Basics*. [Online]. Available: https://www.solar-electric.com/learning-center/solar-charge-controller-basics.html/. [Accessed: 10-Mar-2023].

[3] North East Battery, Ed., “Battery 101 three things need to know lead acid batteries: Northeast Battery Blog,” *Northeast Battery*, 18-Feb-2021. [Online]. Available: https://northeastbattery.com/battery-101-three-things-need-know-lead-acid-batteries/#:~:text=By%20fully%20discharging%20your%20lead,to%20the%20memory%20effect%20issue. [Accessed: 10-Mar-2023].

[4] “Solar Power Panel: Assembly of solar cells that can generate 230 to 275 watts of power,” The Economic Times. [Online]. Available: https://economictimes.indiatimes.com/small-biz/productline/powergeneration/solar-power-panel-assembly-of-solar-cells-that-can-generate-230-to275-watts-of-power/articleshow/69131072.cms. [Accessed: 20-Jan-2023].

4.8.7 Revision Table

Date	Name	Revision Made
5/10/2023	Grace	Finished section 4.8 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.8
3/12/2023	Grace	Edited Figure 4.8-2
3/12/2023	Grace	Edited section 4.8 to exclude intermediate circuit. The Arduino Nano and RDTS will be connected to the same load port and can handle the voltage and current without risking damage
3/10/2023	Grace	Edited section 4.8 (grammar, spelling, etc.) and relabeled figures
3/10/2023	Grace	Added PCC validation to this document
3/10/2023	Grace	Corrected citations in sections 1-4
3/10/2023	Grace	Generated the citations for section 6
3/10/2023	Grace	Wrote sections 3-5
3/10/2023	Grace	Wrote sections 1-2
3/10/2023	Grace	Template generated by the student portal website

4.9 Solar Panel

4.9.1 Description

The solar panel is connected to a PWM charge controller that is used to charge a battery and run a single sensor station. The panel will send power through the charge controller, which will in turn power the sensor station and charge a battery during the day. The battery will serve as the power supply at night.

The panel has a rated power of 10 watts [W], a maximum voltage of 18 volts [V], a nominal current of 0.56 amps [A], and a minimum voltage of 6V. The sensor box and power charge controller require a minimum of 5V to work, this makes this solar panel ideal for the sensor box. The panel's dimensions are 13.8 inches [in] (length) by 8.6 in (width) with a thickness of 1.3 in. The size is ideal because the customer wanted something small and on the cheaper side so people will not want to steal the panel. The customer plans to mount the panel on a pole secured in a cement-filled bucket. This panel (Eco-worthy 10W 12V Solar Panel) is available in a kit with a charge controller for $33.99 on Amazon.

4.9.2 Design

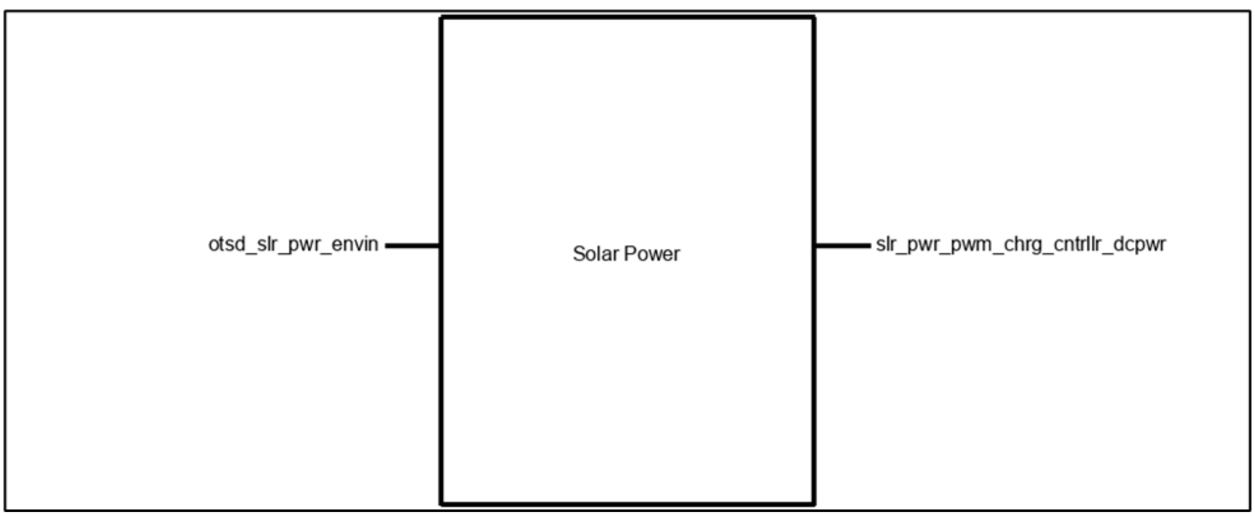


Figure 4.9-1: Black Box view of Solar Panel block

This block represents the solar panel that we are using in this project. It has a 10W maximum power and a maximum voltage of 18V. The solar panel has two interfaces as seen in Figure 4.9-1 above. The first interface, otsd_slr_pwr_envin, is the power supplied by the environment through ultraviolet (UV) rays. The second interface, slr_pwr_chrg_cntrllr_dcpwr, refers to the power charge controller. The power charge controller will manage how much charge goes into our system. The second interface is also where the system will split the charge so that excess charges a battery. The battery will act as the night-time power supply referenced in the description.

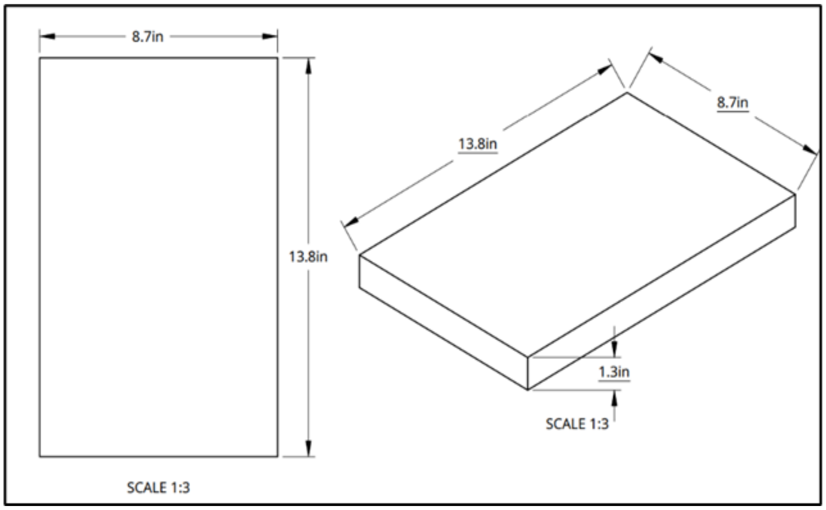
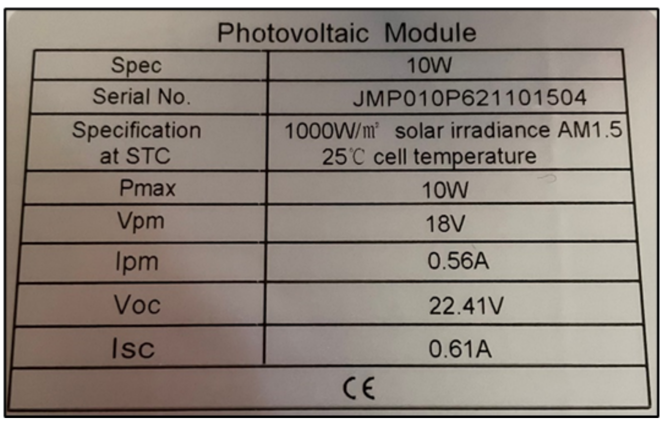


Figure 4.9-3: Mechanical Drawing of Solar Panel

Figure 4.9-2: Image of Solar Panel

The panel is 8.7 in wide, 13.8 in long, and 1.3 in thick as shown in Figure 4.9-3 above. The panel has an aluminum frame and a glass front covering the crystalline material that the photovoltaic cells, as shown in Figure 4.9-2 above.



Photovoltaic Module

Spec	10W
Serial No.	JMP010P621101504
Specification at STC	1000W/㎡ solar irradiance AM1.5 25℃ cell temperature
Pmax	10W
Vpm	18V
Ipm	0.56A
Voc	22.41V
Isc	0.61A

CE

Figure 4.9-4: Picture confirming Solar Panel values (back of the panel)

Figure 4 confirms the device parameters of the solar panel from the introduction. Where Vpm is the maximum voltage the solar panel will output, Pmax is the maximum power, Ipm is the nominal current, Voc is the open circuit voltage (which does not apply to our system), and Isc is the source current value. The section labeled “Specification at STC” refers to the standard test conditions of the panel. Under standard test conditions, the solar panel was tested at 25 degrees Celsius, 1000 watts per square meter of solar irradiance, and an air mass of 1.5.

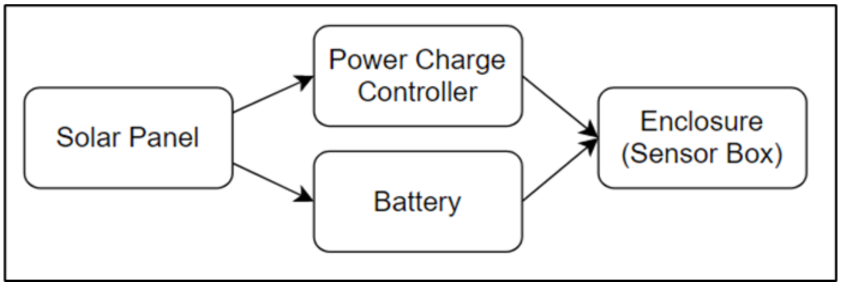


Figure 4.9-5: Basic block diagram showing how Solar Panel is connected to the main system

Figure 4.9-5 above shows in a block diagram how the solar panel is going to be connected to the system. As previously stated, the panel will be connected to the power charge controller and a 12V battery which are connected in parallel. The battery will remain charged throughout the day and regulate the current in the system along with the power charge controller. The battery will take over as the main power supply for the system during the night, as previously mentioned. The power charge controller essentially acts as a breaker that prevents too much current or voltage from entering the main system in the enclosure.

The reason why we chose this design over other options is the cost for both the solar panel and the charge controller was low, which meets our customer's request. The panel is also lightweight and easy to mount, which was another of our customers' concerns. Having the solar panel connected to the power charge controller and battery (which are parallel to each other) allows for the current coming from the solar panel to be efficiently managed. If the solar panel is not providing enough current, the battery will equalize the circuit so the box will continue to function. This occurs in low-light environments, more specifically when the sun sets.

4.9.3 General Validation

Solar power is the best option for our system's design because each sensor box will be out in a field and submerged in soil. Having solar power and a battery backup allows our system to run continuously without the constant need to replace the power supply or run an extension cord through the field. Thailand, where we are designing these devices, has a tropical environment so there should be plenty of sunlight for the panel to absorb.

The reason why we decided to include a cheaper solar panel with low wattage in our system is our customer requested it. Cost is a key consideration in our system because our customer wants to make as many boxes as possible to collect data across several farms once the design is finalized. Our customer is also worried about the theft of the system. By having a cheaper panel and mounting it in a bucket of cement, thieves will likely not want to go through the trouble of stealing the panel.

The solar panel will be expected to perform its purpose on days with full sun or partly cloudy conditions. According to an article written by Boston Solar, the average solar panel (12V panel) can perform in low light conditions, however, they need at least 4 hours of full sun to operate at full potential [2].

4.9.4 Interface Validation

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

Table 4.9-1: otsd_slr_pwr_envin: Input

Light: Panel needs a minimum of 4 hours of direct sunlight a day	The solar panel will take in energy from the sun and convert it to 12V DC on average. The voltage output is dependent on how much sunlight the panel is exposed to and for how long the panel is allowed to absorb that light.	Judgement: Based on our customer's needs and where the sensor box will be in the rice patty field, a solar panel is the ideal power source. The panel has a maximum voltage of 18V and a nominal current of 0.56A. This panel will provide more power than our system will need, but so much that it will damage our system. The panel also meets our customer's cost and power requirements.

Table 4.9-2: slr_pwr_pwm_chrg_cntrllr_dcpwr: Output

Inominal: 0.56A	The power charge controller can accept a maximum of 10A, our nominal current value is significantly less than that. This value will support the charge controller, but not overload the controller or the sensor box.	Judgement: We need a current that can supply the 5V and 1.2A USB output without overloading the sensor box. This amperage meets our system's needs and if it were to be lowered, the battery backup would compensate for loss.

Ipeak: 10A	The power charge controller can accept a 10A maximum. This value accounts for a sudden surge in power or anything else that could cause a spike in current.	Judgement: The current peak value is high enough that the system will not be damaged in case of a surge. Environments can be unpredictable and having a peak current value of 10A ensures the safety of our system. If this value was to be reached, it should be noted that the solar panel would most likely be damaged, which could have also caused the surge.
Other: 10W (Max Power)	The panel's power value is 10W maximum because our customer wanted a panel that was cheap, but able to effectively power the system.	Judgement: Based on our system's needs (5V input voltage to the microcontroller) a 10W panel with a 12V output is more than enough to support the sensor box. As shown in Figure 4, the panel has a power maximum of 10W and a maximum output voltage of 18V. These values can be managed by the power charge controller which accepts the maximum voltage and a current up to 10A.
Vmax: 18V	Each cell in an average solar panel outputs about 0.46V per cell, and our current panel has about 40 cells [1]. Therefore, the maximum voltage out is about 18V. The system will take the 18V input to charge a battery and power the rest of the system which operates on a 5V input.	Judgement: The system requires a 12V input, this panel has a maximum of 18V which exceeds our expectations. This makes our choice of panel ideal for the sensor box and power charge controller.

Vmin: 6V	This is the approximate minimum value that the solar panel can output and continue to power our system with the use of the battery. The minimum voltage needed to run our system is 5V, so 6V exceeds our needs for the sensor box. The battery will manage the current value since it sits in parallel with the charge controller connected to the panel.	Judgement: 6V would be the lowest possible voltage that the system could run on. This value would be shown when the solar panel is not in full sun or when the sun has set. It should be noted that the absolute minimum would be 0V (absolute darkness), but that condition will not be met in the environment we are building this system.

4.9.5 Verification Plan

1. Attach the solar panel to the power charge controller securely.
2. Take the solar panel and place it outside when the weather is clear (alternatively if weather conditions do not cooperate you can use a grow light to simulate the sun).
3. Attach a multimeter to the output leads on the back of the panel to measure the voltage output.
4. Ensure the voltage output maximum is in fact 18V and take a picture as proof.
5. Change the multimeter settings to measure the current output (make sure the multimeter leads are connected to the right inputs on the meter) and take a picture as proof.
6. Take the voltage and current values that you measured and calculate the power to verify the panel's wattage, 10W.
7. Once the panel values are measured and calculated, disconnect the panel, and take it inside (if you are not using the grow light method).
8. Cover the panel with a cloth or opaque material so that it can discharge. Also, make sure the leads are not touching each other so you do not damage the panel.
9. Once the panel has had enough time to discharge, reattach the multimeter to verify the panel has a 6V minimum in low-light conditions. Again, take a photo for confirmation.
10. Once the values of the panel are confirmed, connect the panel to the power charge controller and the 12V battery.
11. Connect an Arduino Nano to the 5V output USB port on the power charge controller.

12. Confirm the nano is powered and fully functional. Make sure to videotape this portion for later verification.

    Note: there is no need to test the output values of the power charge controller, that block will be verified separately. You simply need to confirm the nano is operational as that will simulate the microcontroller in the sensor box that needs to be powered.

4.9.6 References & File Links

[1] “Solar Power Panel: Assembly of solar cells that can generate 230 to 275 watts of power,” The Economic Times. [Online]. Available: https://economictimes.indiatimes.com/small-biz/productline/powergeneration/solar-power-panel-assembly-of-solar-cells-that-can-generate-230-to275-watts-of-power/articleshow/69131072.cms. [Accessed: 20-Jan-2023].

[2] Solar, “Do solar panels need direct sunlight to work?,” *Boston Solar*, 01-Mar-2022. [Online]. Available: https://www.bostonsolar.us/solar-blog-resource-center/blog/do-solar-panels-need-direct-sunlight-to-work/#:~:text=Despite%20their%20ability%20to%20perform,of%20direct%20sunlight%20per%20day. [Accessed: 11-Feb-2023].

4.9.7 Revision Table

Date	Name	Revision Made
5/10/2023	Grace	Finished section 4.9 edits - grammar, figure labels, etc.
5/4/2023	Grace	Fixed formatting in section 4.9
3/10/2023	Grace	Edited section 4.9 (grammar, spelling, etc.) and relabeled figures
3/10/2023	Grace	Added solar panel validation to this document
2/11/2023	Grace	Updated document based on the reviews I was given from the first submission
2/11/2023	Grace	Added the updates from the template generated by the student portal added to this document
1/20/2023	Grace	Wrote sections 3-6
1/19/2023	Grace	Wrote sections 1-2
1/19/2023	Grace	Template generated by the student portal website

Section 5 – System Verification Evidence

5.1 Universal Constraints

The following figures display different aspects of the system meeting the universal constraints. Each requirement section references the specific figures used to display the accomplishment of the requirement, or a description explaining why the requirement does not apply to the system.

Figure 5.1-1: System PCB

5.1.1 The system cannot contain a breadboard

The system will avoid using breadboard connections by implementing a PCB to make connections to multiple pins on the Nano. This avoids any use of a breadboard in the system. Figure 5.1-1 displays the PCB used to connect all the SPI pins between the microcontroller, radio data transfer, and temperature and humidity sensor blocks.

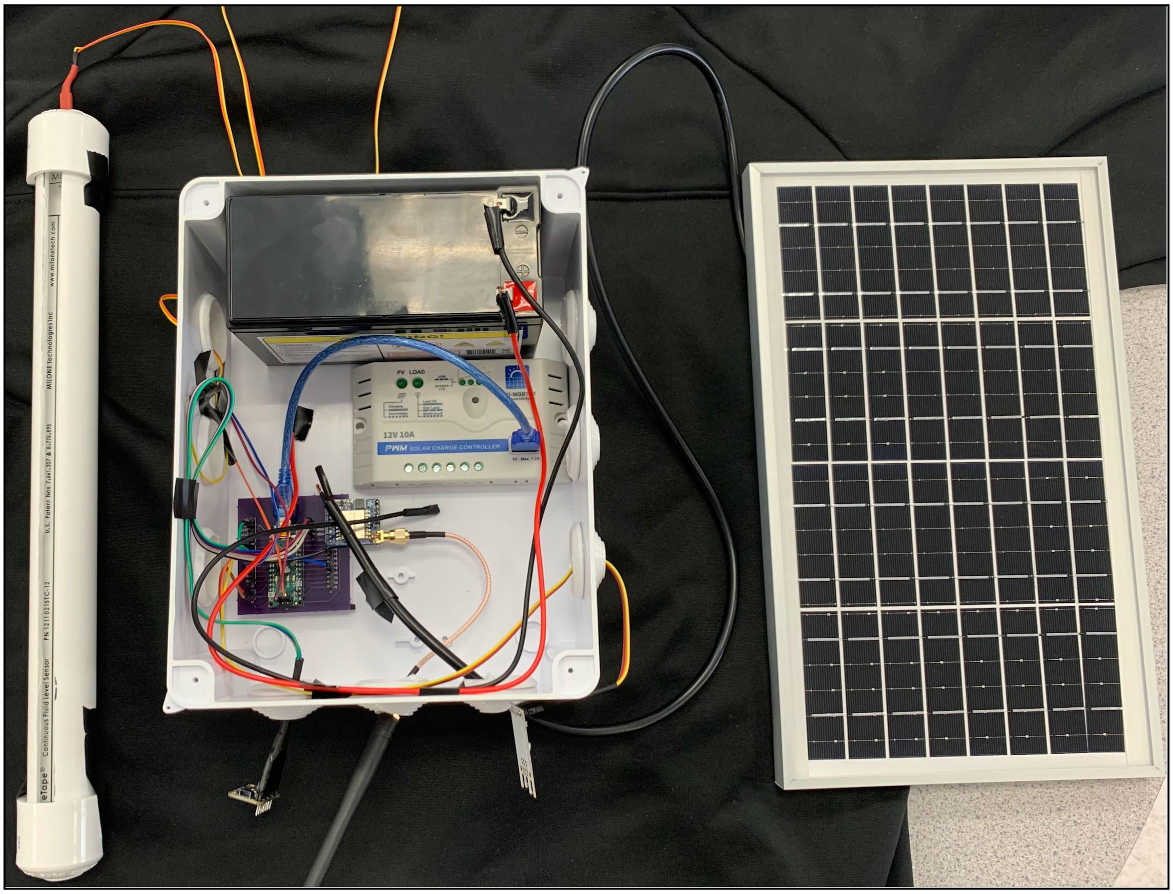


Figure 5.1-2: Image of completed system showing there is no breadboard

5.1.2 The final system must contain a student-designed PCB

The system does not meet this requirement because the PCB on the board does not have 30 SMD connection pads. It will contain a PCB designed by the team specifically to connect three of the SPI pins to both the Temperature and Humidity Sensor and the Radio Data Transmitter simultaneously (shown in Figure 5.1-1), but neither of these blocks require any extra components with SMD connections. Bryan Hugil, the project partner, has explicitly asked for devices and components easily accessible to him in Thailand. Soldering thirty separate surface mount connections would cause a huge barrier in his ability to assemble the system.

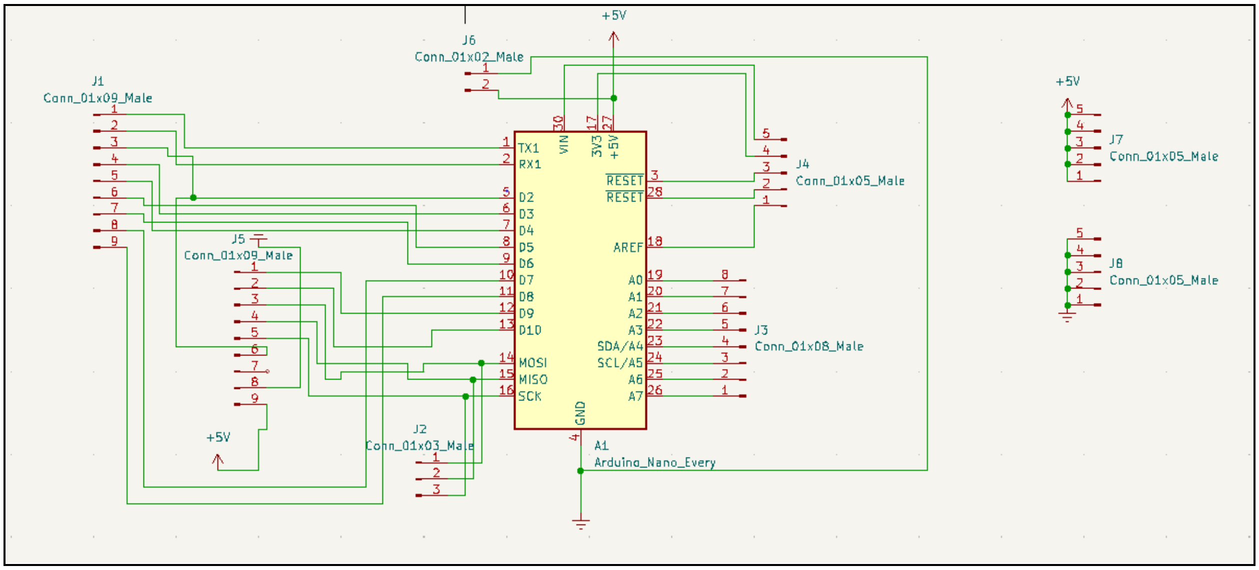


Figure 5.1-3: PCB Schematic

This requirement has been waived by Don in [Figure 5.1-4].

5.1.3 All connections to the PCBs must use connectors

The system meets this requirement by including connectors to the PCB for the Microcontroller, Temperature, and Humidity Sensor, Water Level Sensor, and Radio Data Transmitter block. Figure 5.1-1 shows the connectors on the PCB.

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

The system meets this requirement. The solar panel outputs 18V (Vin1), the PCC only requires 12V (Vo1) [2]. The solar panel also outputs 0.56A (Iin1) and the PCC uses a maximum of 0.50A (Io1) [2]. The resultant power in (Pin1) in watts [W] is:

$$Pin1 = Vin1 \times Iin1 = 18V \times 0.56A = 10.08\,W$$

The resultant power out (Po) is as follows:

$$Po1 = Vo1 \times Io1 = 12V \times 0.50A = 6.0\,W$$

Therefore, the solar panel's efficiency is:

$$Efficiency1 = \frac{Po1}{Pin1} \times 100\ (\%) = \frac{6.0W}{10.08W} \times 100\ (\%) = 59.5\%$$

The PCC outputs 5V (Vo2) and 1.2A (Io2) to the system. The system requires 5V (Vin2) and 0.50A (Iin2). The resultant power and efficiency are as follows:

$$Pin2 = Vin2 \times Iin2 = 5V \times 0.50A = 2.5\,W$$

$$Po2 = Vo2 \times Io2 = 5V \times 1.2A = 6.0W$$

$$Efficiency2 = \frac{Po2}{Pin2} \times 100\ (\%) = \frac{6.0W}{2.5W} \times 100\ (\%) = 240\%$$

The battery backup, that runs during the night cycle, inputs 12V (Vin3) and a current of 7 amp-hours [Ah] (Iin3). The PCC requires 12V (Vo1) and uses a maximum of 0.50A (Io1) every 30 minutes. Therefore the efficiency of the battery is as follows:

$$Pin3 = Vin3 \times Iin3 = 12V \times 7.0Ah = 84Watt - hours\,[Wh]$$

$$Po3 = Vo1 \times Io1 = 12V \times 1.0A = 12\,Wh$$

$$Efficiency3 = \frac{Po3}{Pin3} \times 100\ (\%) = \frac{12Wh}{84Wh} \times 100\ (\%) = 14.3\%$$

The power draw for the PCC to the system remains the same at night. Based on the calculations above, we can calculate that the daytime and nighttime efficiency of the system.

$$Overall\ Daytime\ Efficiency = \frac{59.5\% + 240\%}{2} = 149.75\%$$

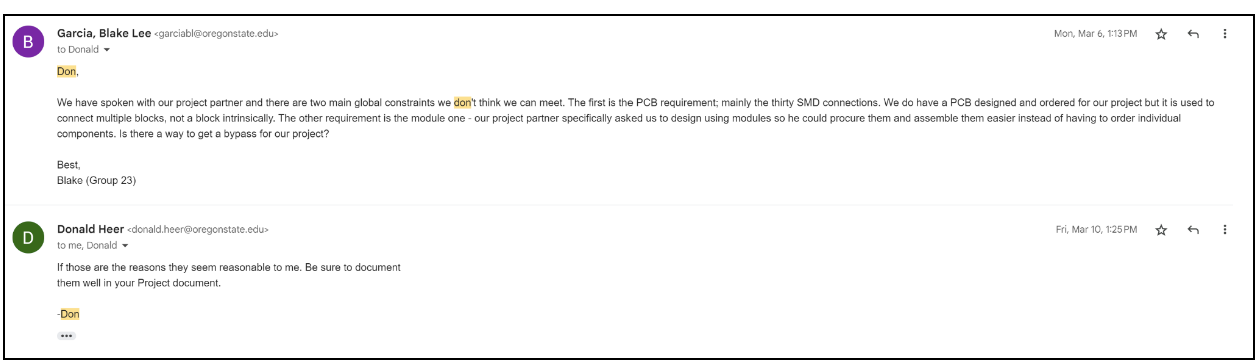
$$Overall\ Nighttime\ Efficiency = \frac{14.3\% + 240\%}{2} = 127.15\%$$

The reason why our efficiencies are so high is due to the PCC outputting more current than it receives to our system. This is due to the internal circuitry in the PCC amplifying the current.

5.1.5 The system may be no more than 50% built from 'modules'

The system does not meet this requirement because the project customer, Bryan Hugil, has requested the system contain easily accessible parts capable of operating with limited additional setup. Designing a system to meet the other engineering and customer requirements would not be possible without surface-mounted connections which would make assembly of the system near impossible with the tools Bryan has on hand in Thailand.

This requirement has been waived by Don in [Figure 5.1-4].



Garcia, Blake Lee <garciabl@oregonstate.edu> — Mon, Mar 6, 1:13 PM
to Donald

Don,

We have spoken with our project partner and there are two main global constraints we don't think we can meet. The first is the PCB requirement; mainly the thirty SMD connections. We do have a PCB designed and ordered for our project but it is used to connect multiple blocks, not a block intrinsically. The other requirement is the module one - our project partner specifically asked us to design using modules so he could procure them and assemble them easier instead of having to order individual components. Is there a way to get a bypass for our project?

Best,
Blake (Group 23)

Donald Heer <donald.heer@oregonstate.edu> — Fri, Mar 10, 1:25 PM
to me, Donald

If those are the reasons they seem reasonable to me. Be sure to document them well in your Project document.

-Don

Figure 5.1-4: Email from Don Waiving Global Constraints

5.2 Requirements

5.2.1 Sensor Requirement

5.2.1.1 Project Partner Requirement

The system will measure temperature, humidity, wind speed, and water level.

5.2.1.2 Engineering Requirement

The system will measure temperature from 0-45 degrees C, humidity from 25% to 95%, wind speed from 0-75 kmph, and water level from 0-12 inches.

5.2.1.3 Verification Processes

Verifying this requirement takes eight distinct environments. Each environmental factor has a maximum and minimum condition to verify and the requirement lists four different measurements. The process below includes the initialization and a description of how to verify each individual maximum or minimum condition.

Initialization:

1. Set up the sensor box with the radio antenna exposed and the most recent copy of the transmitter node code flashed onto the microcontroller. Ensure the temperature sensor, water level sensor, wind speed sensor, and solar panel are present outside of the enclosure while the rest of the electronics are inside.

2. Set up the receiver node connected to an external computer. Ensure the microcontroller of the external node receives enough power to begin operating. Open the serial monitor and begin observing each printed value.

Temperature Values:

3. Place the system in an environment below zero degrees Celsius. Verify the current temperature of the environment with an external thermometer. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the temperature reading to ensure it is below zero degrees Celsius.

4. Place the system in an environment above forty-five degrees Celsius. Verify the current temperature of the environment with an external thermometer. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the temperature reading to ensure it is above forty-five degrees Celsius.

Humidity Values:

5. Place the system in an environment at or below twenty-five percent relative humidity. Verify the current humidity of the environment with an external hygrometer. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the temperature reading to ensure it is at or below twenty-five percent relative humidity.

6. Place the system in an environment at or above ninety-five percent relative humidity. Verify the current humidity of the environment with an external hygrometer. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the temperature reading to ensure it is at or above ninety-five percent relative humidity.

Wind Speed:

7. Place the system in a car moving at or above 75 km/h. Verify the current speed of the car using a speedometer. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the wind speed reading to ensure it is at or above 75 km/h.
8. Place the system in a closed environment with still air. Completely close off the box to ensure there is no outward or inward movement of the wind. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the wind speed reading to ensure it is at 0 km/h.

Water Level:

9. Place the water level sensor vertically in a dry bucket. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the water level reading to ensure it is at zero inches.
10. Place the water level sensor vertically in a dry bucket. Gradually add more water until there are twelve inches present in the bucket. Visually inspect the ruler marking on the water level sensor for accurate water height. Open the serial monitor or serial plotter on the computer connected to the receiver node and observe the values of the water level reading to ensure it is at zero inches.

5.2.1.4 Testing Evidence

[5], [6] are videos of the testing setup used to simulate the necessary.

1) Low Temperature:

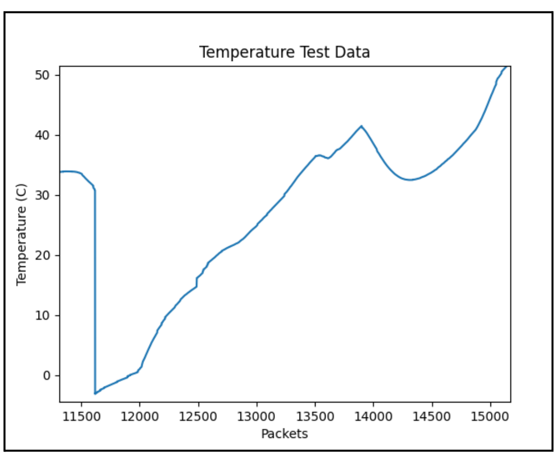


Figure 5.2-1: Low Temperature Readings

Figure 5.2-2: Low Temperature Thermometer

2) High Temperature:

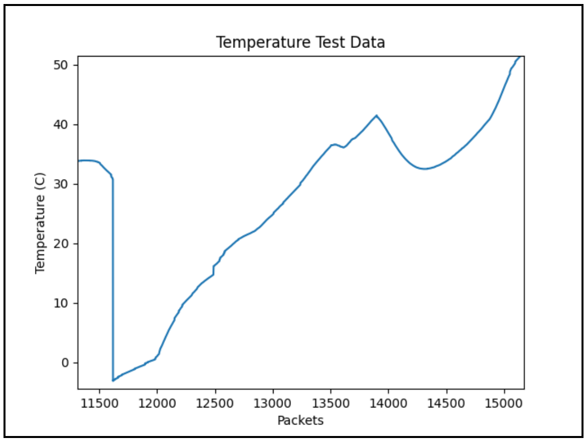


Figure 5.2-3: High Temperature Readings

Figure 5.2-4: High Temperature Thermometer

3) Low Humidity:

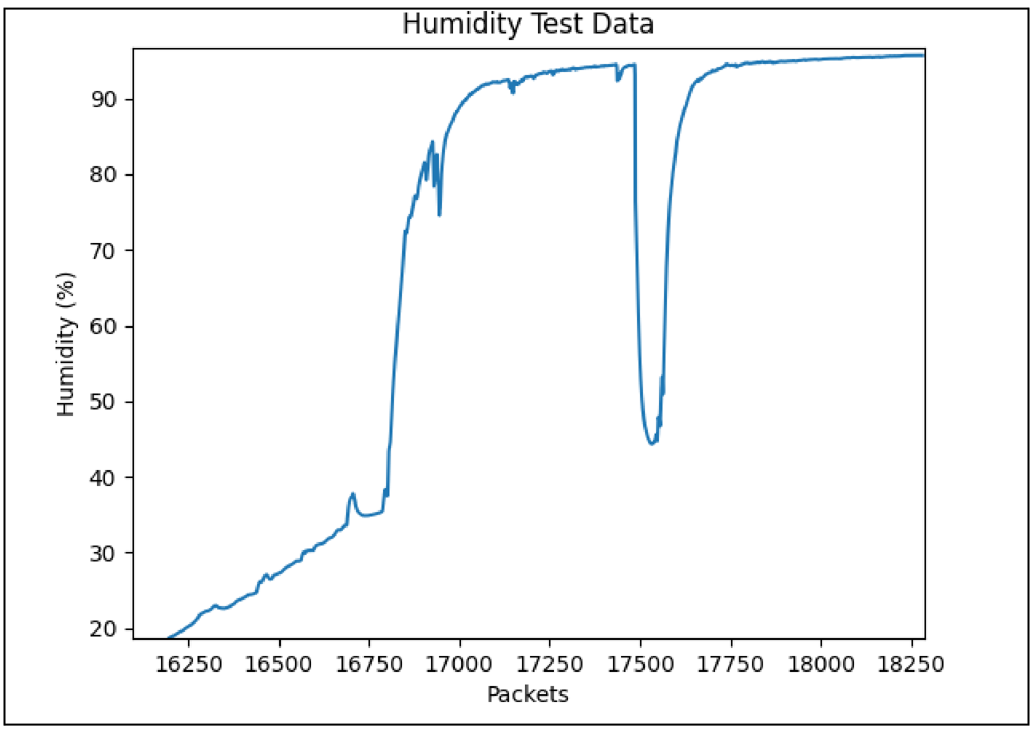


Figure 5.2-5: Low Humidity Graph

4) High Humidity:

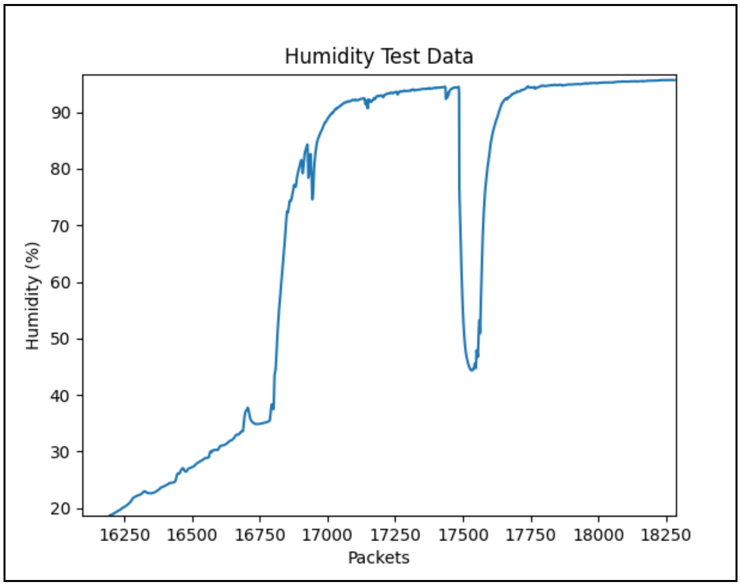


Figure 5.2-6: High Humidity Graph

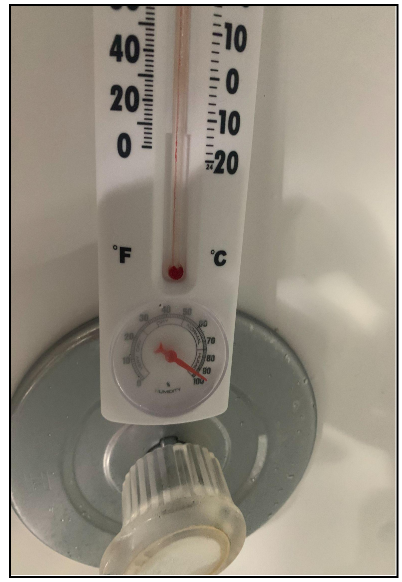


Figure 5.2-7: High Humidity Hygrometer

5) Low Wind Speed:

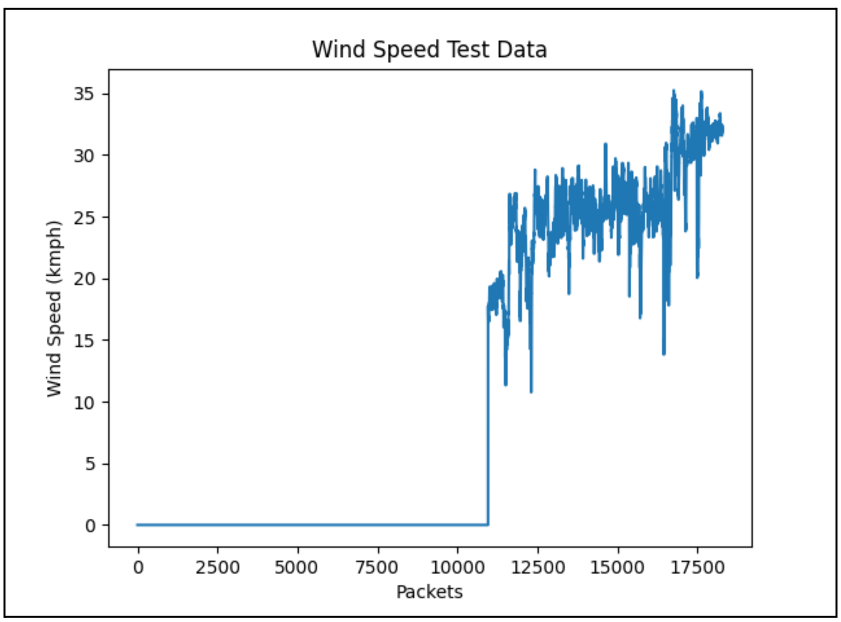


Figure 5.2-8: Low Wind Speed Graph

6) High Wind Speed:

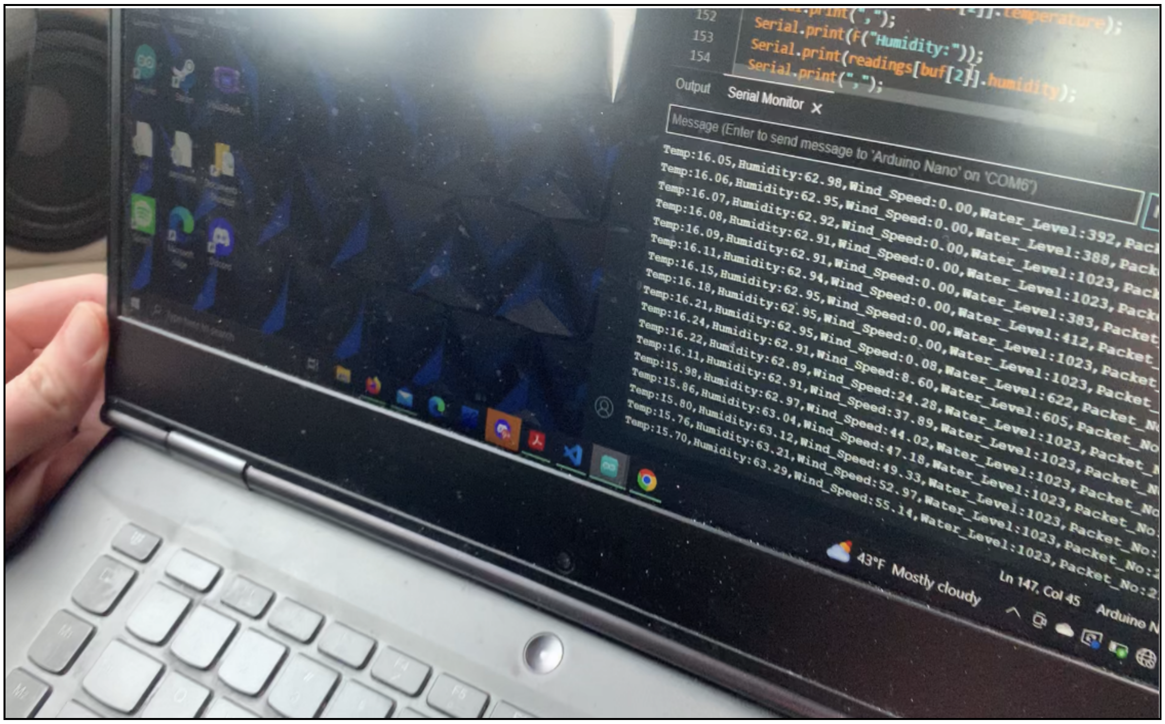


Figure 5.2-9: High Wind Speed Readout

7) High and Low Water Levels

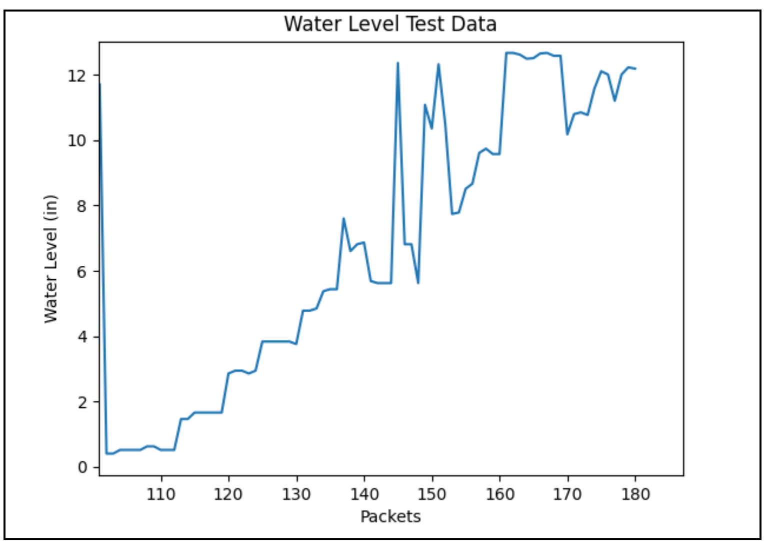


Figure 5.2-8: Full water level range

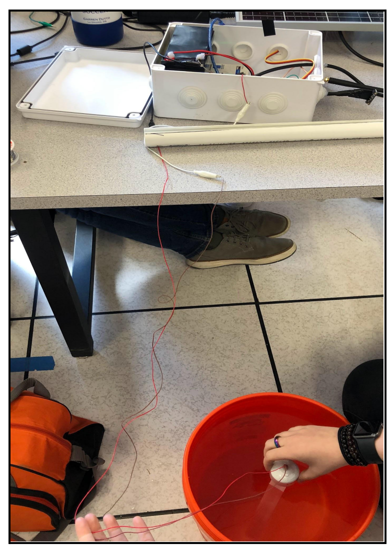


Figure 5.2-9: Water level sensor setup for 12 inches of water

Figure 5.2-10: Water level setup for 0 inches of water.

5.2.2 Data Sending Range Requirement

5.2.2.1 Project Partner Requirement

Adjacent sensor boxes in non-contiguous rice paddy fields will relay information to each other.

5.2.2.2 Engineering Requirement

The system will relay information to a separate identical system over 1 km away with a 99% success rate without line-of-sight between devices.

5.2.2.3 Verification Processes

To verify this requirement, the nodes need to be 1 km away from each other and get 99% of data packets through before the next data transmission.

1. Turn on the device and base station
2. Verify that they are connected to each other
3. Move the two nodes 1 km away from each other
4. If the connection is spotty, add a relay node in between the two nodes
5. Verify that every sensor data packet is being received
6. Run the system until 100+ data packets have been received, and count how many drops there were using the sequence numbers of the packets

5.2.2.4 Testing Evidence

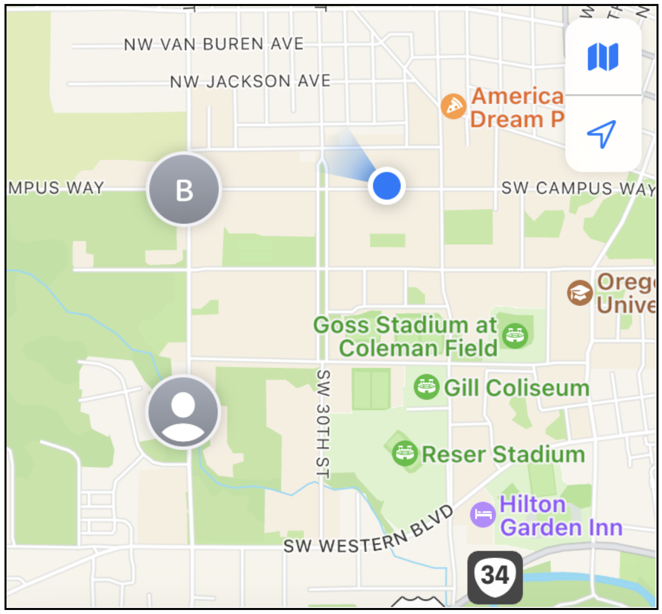


Figure 5.2-11: Sensor box locations during range test.

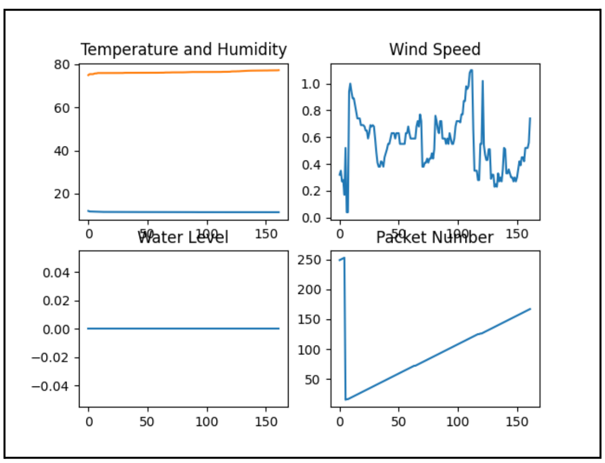


Figure 5.2-12: Sensor Box Data At Sending Range of 1km

Figure 5.2-13: Range Test Distance

Drops would be indicated by a vertical “jump” in the Packet Number graph in the lower-right corner of the figure (other than the rollover where the packet number jumps from 255 to 0).

5.2.3 Cost Requirement

5.2.3.1 Project Partner Requirement

Each sensor unit must prioritize low-cost design methods to reduce the overall cost of the project without any recurring monthly costs.

5.2.3.2 Engineering Requirement

The system hardware in total should cost less than $300 at the time of purchase.

5.2.3.3 Verification Processes

An accurate bill of materials verifies this requirement by documenting the different purchases necessary to recreate the system and tracking the cost. Each price listed on the BOM was recorded at the time of purchase. To verify this requirement is met:

1) Open the Bill of Materials available in [1].

2) Sum the total cost of the system.

5.2.3.4 Testing Evidence

Figure 5.2-12 displays a screenshot of the Bill of Materials from [1]. The row labeled “Total” indicates the total cost of the system ($212.26). This meets the requirement of a cost value under $300.

	A	B	C
1	Item	Price	Link
2	12" water level sensor	39.95	https://www.adafruit.com/product/464?gclid=CjwKCAiA68ebBhB-EiwALVC-NhmQ31Q75J
3	Wind Speed Sensor	21.95	https://moderndevice.com/products/wind-sensor
4	Temp/Humidity Sensor	14.95	https://www.adafruit.com/product/2652
5	Radio Module	19.95	https://www.adafruit.com/product/3072
6	Board	24.9	https://store.arduino.cc/products/arduino-nano
7	Solar Panel & Power Charge Controller	33.99	https://www.amazon.com/dp/B09RZRY4QB?psc=1&ref=ppx_yo2ov_dt_b_product_details
8	12V Rechargeable Battery	19.99	https://www.amazon.com/ML7-12-Battery-Mighty-Brand-Product/dp/B088FZ7MM4/ref=sr_
9	Enclosure Box	22.99	https://www.amazon.com/LeMotech-Dustproof-Waterproof-Electrical-255mmx200mmx80r
10	Antenna Cable	5.5	https://www.amazon.com/gp/product/B086JJM654?ref=ppx_pt2_dt_b_prod_image&th=1
11	Antenna	8.09	https://www.digikey.com/en/products/detail/kyocera-avx/9000046-XLPDNB/14825444?utr
12			
13	**Total**	212.26	

Figure 5.2-14: Bill of Materials

5.2.4 Data Display

5.2.4.1 Project Partner Requirement

The data should be graphically displayed in a user interface.

5.2.4.2 Engineering Requirement

The system will graphically display each sensor's data on a personal computer over a 1 month time frame.

5.2.4.3 Verification Processes

To verify this requirement obtain a computer and download the user application executable file from [4]. Collect or generate one full month of testing data in a valid Environmental Data File with the proper *.edf extension. Once these are available:

1. Run the program. It will open an initial window pictured in Figure 5.2-15. Select the option “Display Environmental Data”. If the images in the window appear disproportionate to Figure 5.2-15 ensure the operating system does not automatically scale the display.

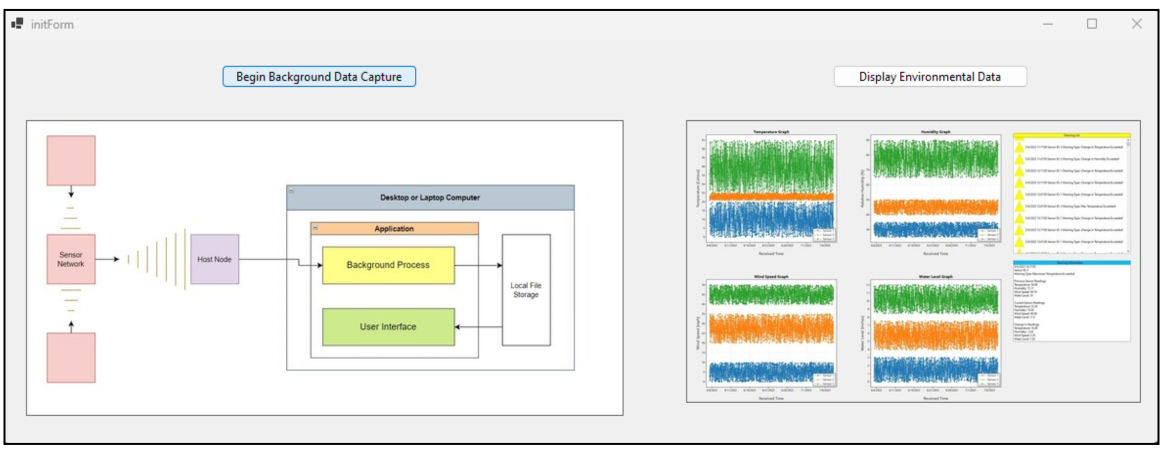


Figure 5.2-15: Opening Application Window

2. Navigate to the valid Environmental Data File in the File Explorer Window. The text box at the top of the window can receive text input for a file location, or the “Back” and “Go” buttons can interact with selecting folder icons in the window. Any file with a green check mark (pictured in Figure 5.2-14) has the correct file extension. Click on the desired file and push the “Select” button.

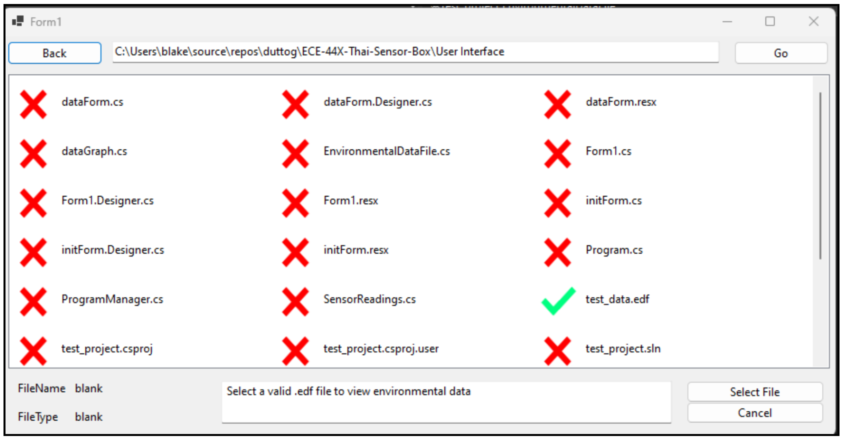


Figure 5.2-16: File Explorer Window

3. Once selected the program will open a screen with the graphing data and generate the warnings. This process may take a few minutes depending on the size of the data file and the number of warnings. The completed data window should look like Figure 5.2-17 below.

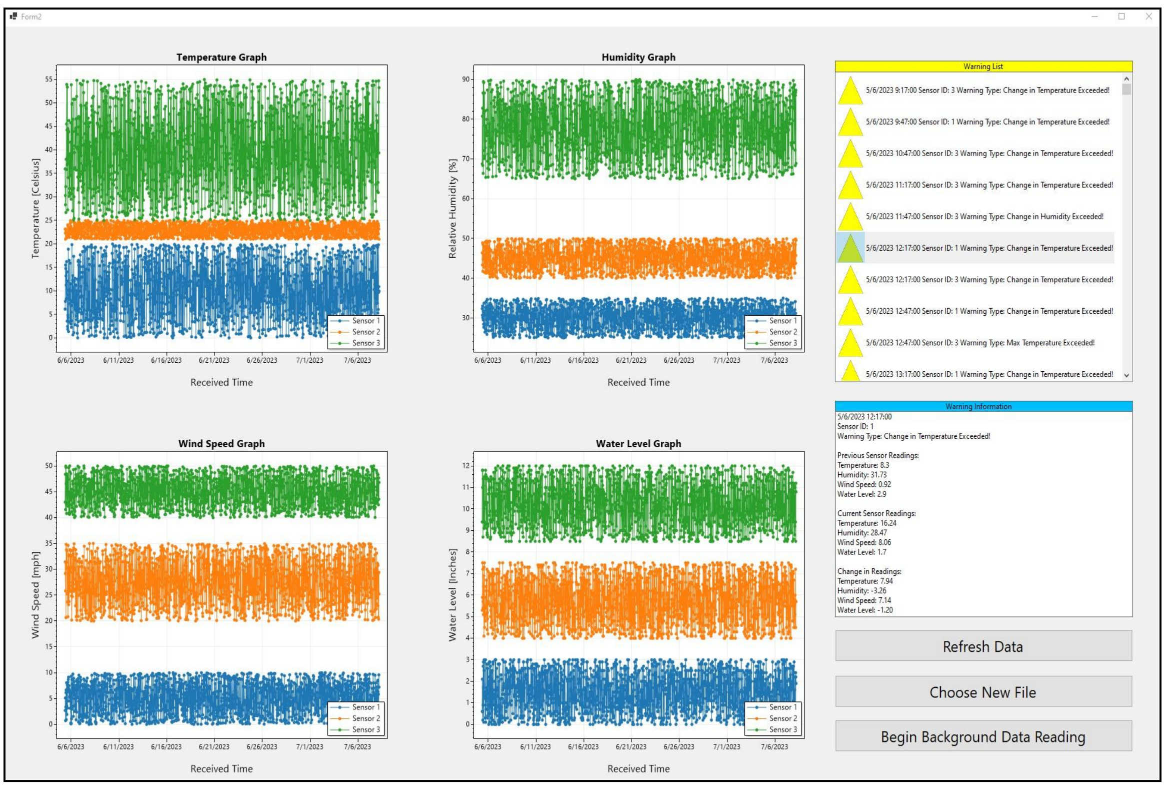


Figure 5.2-17: Application Data Window

5.2.4.4 Testing Evidence

There are two sources of evidence for this requirement. The first source is Fig. 5.2-17 which is an image with over a month of data in it and the second is [3] which is a Youtube video with a recording of the program.

5.2.5 Data Sending Frequency

5.2.5.1 Project Partner Requirement

The sensors will relay information non-continuously but record significant changes over the course of a single night.

5.2.5.2 Engineering Requirement

The system will relay sensor data at least every 30 minutes for eight continuous hours.

5.2.5.3 Verification Processes

To verify this requirement, run the system for 8 continuous hours, and record the data being sent from the sensor box to the base station. After the 8 hours are finished, verify through timestamps that the sensor box transmitted data at least every 30 minutes for the entire 8 hours.

5.2.5.4 Testing Evidence

A time-lapse of the data collection is present at [7] and the resulting *.csv file is available at [8]. [Figure 5.2-20] is a screenshot of the time-lapse with the time taken due to the blurriness of the video itself for further evidence.

The data was sent every minute, with the largest gap between two data points being around 18 minutes, which is under the 30-minute parameter set in the requirement. The test started at 4:18 PM and ended at 12:21 AM.

test.csv

```
21.12,50.78,5.1,1252.9113924050637,17,2023-05-06 16:18:00.507820
21.11,50.8,4.06,1336.953642384106,18,2023-05-06 16:19:05.465099
21.1,50.84,3.61,1318.295081967213,19,2023-05-06 16:20:05.662390
21.1,50.89,4.57,1343.2558139534888,20,2023-05-06 16:21:05.917005
21.09,50.92,6.21,1324.4736842105265,21,2023-05-06 16:22:06.134847
21.09,50.85,5.65,1306.058631921824,22,2023-05-06 16:23:06.367965
21.09,50.83,5.03,1306.058631921824,23,2023-05-06 16:24:06.570039
21.07,50.87,4.57,1324.4736842105265,24,2023-05-06 16:25:06.766854
21.06,50.93,5.88,1276.1538461538462,25,2023-05-06 16:26:07.007693
21.04,50.93,4.5,1276.1538461538462,26,2023-05-06 16:27:07.252441
```

Figure 5.2-18: Data from the start of the test

test.csv

```
20.98,49.16,6.93,1436.0975609756097,241,2023-05-07 00:14:50.292083
20.98,48.95,8.74,1436.0975609756097,242,2023-05-07 00:15:50.512725
20.97,48.54,9.45,1436.0975609756097,243,2023-05-07 00:16:50.733760
20.96,48.59,0.36,1436.0975609756097,244,2023-05-07 00:17:50.954379
20.95,48.35,0.28,1436.0975609756097,245,2023-05-07 00:18:51.174792
20.92,49.03,6.19,1422.2837370242212,246,2023-05-07 00:19:51.391232
21.0,49.47,4.29,1422.2837370242212,247,2023-05-07 00:20:51.615957
21.04,49.17,6.93,1422.2837370242212,248,2023-05-07 00:21:51.832844

```

Figure 5.2-19: Data from the end of the test

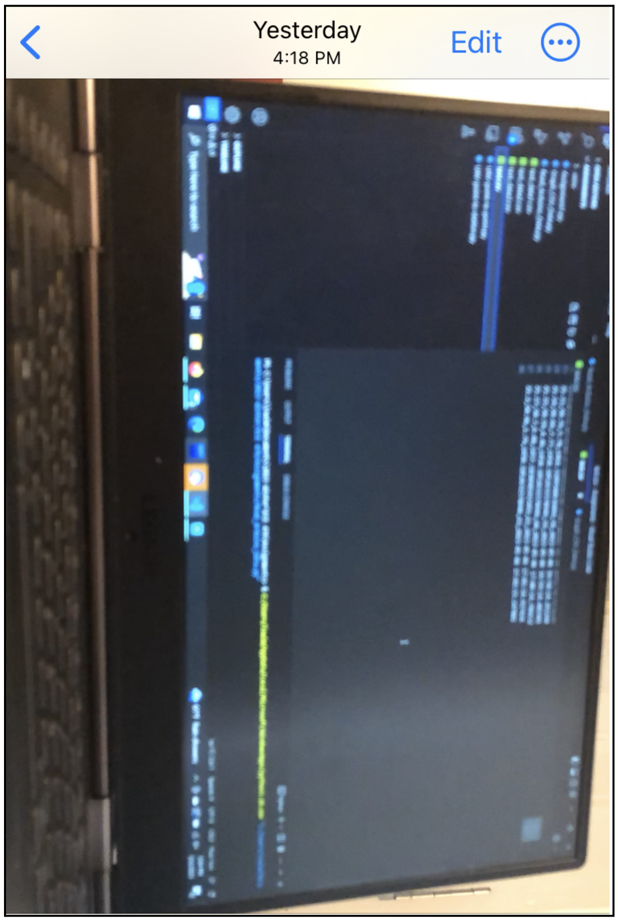


Figure 5.2-20: Timestamp showing the test was started at 4:18 PM

5.2.6 Documentation Quality

5.2.6.1 Project Partner Requirement

The overall system design will have enough information for both future manufacturing and development

5.2.6.2 Engineering Requirement

The project will have appropriate documentation as approved by the project partner

5.2.6.3 Verification Processes

The project partner asked for customer-facing documentation to ease the assembly process and an explanation of using the software. To verify this requirement we went through the following steps:

1. Create a rough draft of each type of document with group members individually writing the sections they contributed to the most.
2. Peer-review each piece of documentation as a group to ensure a layman's understanding is enough.
3. Send the documents to the project partner for review. In this email, we also asked for feedback or suggestions for a second round of revisions (if necessary).
4. Send the complete documentation and acquire approval.

5.2.6.4 Testing Evidence

The documents approved by the project partner include [1], [3], [9], [10]. They were initially sent via email in Figure 5.2-21 and later approved via email in Figure 5.2-22.

Sharp, Kendra
to me

Thu, May 4, 2:52 AM (5 days ago)

[This email originated from outside of OSU. Use caution with links and attachments.]

Hi Blake,

Let me know when your user guide is complete and I'll review. When is the deadline? I will be available this Friday between about 12-3 and would love to talk to your group. Is that possible? I'm also coordinating with Caleb on the other team to talk to them.

Kendra

From: "Garcia, Blake Lee" <garciabl@oregonstate.edu>
Date: Wednesday, May 3, 2023 at 10:26 PM
To: "Sharp, Kendra" <kvsharp@nsf.gov>
Subject: [EXTERNAL] - Documentation Approval

This email originated from outside of the National Science Foundation. Do not click links or open attachments unless you recognize the sender and know the content is safe.

Kendra,

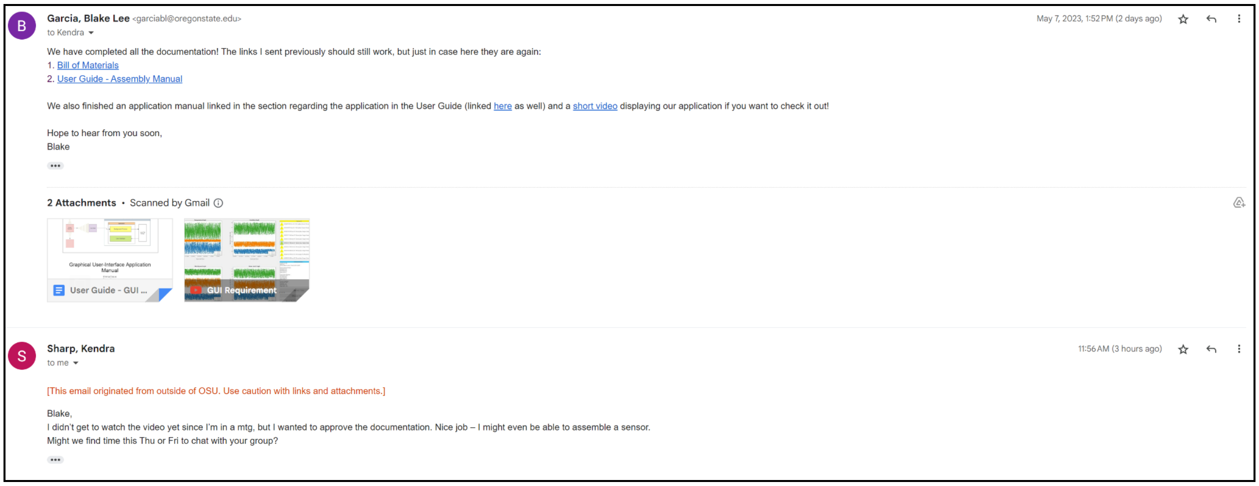
As we mentioned in one of our previous meetings, we have eight specific requirements we need to fulfill for our final system verification. Most of them involve the system itself (i.e. "The system can communicate over 1 km without line of sight with 99% success rate") but we have two involving the documentation. Both of them are phrased to require your approval, so I wanted to reach out with plenty of time for you to review them and ensure they are satisfactory (or make edits you feel are necessary).

Here are the documents requiring review:
1. Bill of Materials
2. User Guide - Assembly Manual

Due to some scheduling issues, we are still editing the User Interface Guide, but that is linked in the User Manual and should be complete by early next week. We will still be sending our weekly update as usual, so expect to hear from us again before the end of the week!

Best,
Blake

Figure 5.2-21: Initial Email to project partner



Garcia, Blake Lee <garciabl@oregonstate.edu>
to Kendra
May 7, 2023, 1:52 PM (2 days ago)

We have completed all the documentation! The links I sent previously should still work, but just in case here they are again:
1. Bill of Materials
2. User Guide - Assembly Manual

We also finished an application manual linked in the section regarding the application in the User Guide (linked here as well) and a short video displaying our application if you want to check it out!

Hope to hear from you soon,
Blake

2 Attachments • Scanned by Gmail

Graphical User-Interface Application Manual
User Guide - GUI ...
GUI Requirement

Sharp, Kendra
to me
11:56 AM (3 hours ago)

[This email originated from outside of OSU. Use caution with links and attachments.]

Blake,
I didn't get to watch the video yet since I'm in a mtg, but I wanted to approve the documentation. Nice job – I might even be able to assemble a sensor.
Might we find time this Thu or Fri to chat with your group?

Figure 5.2-22: Confirmation Email from project partner

5.2.7 Ease of Assembly

5.2.7.1 Project Partner Requirement

The project partner and other farmers within Raitong Organics Farm must have the necessary information and skills to construct modules without contact with the engineering team.

5.2.7.2 Engineering Requirement

The system must contain only parts that are approved by the project customer and that are available to purchase in Thailand.

5.2.7.3 Verification Processes

This verification included two specific documents getting checked off by the project partner. The first included the Bill of Materials and the second was a build guide to ensure both the parts used and the techniques implemented are accessible by the project customer. To verify:

1. Draft a Bill of Materials and an Assembly Guide for all parts used in the system.
2. Send both documents to the project partner and ask for revision or approval if anything is unavailable.
3. Make edits to the documentation or system until the project partner is satisfied.

5.2.7.4 Testing Evidence

The testing evidence for this section includes an initial email Figure 5.2-21 to the project partner indicating the purpose of the email and including all necessary information with enough time for revision (if necessary). An additional email with confirmation of the met requirement from the updated documentation is also included in Figure 5.2-22.

5.2.8 Weatherproofing

5.2.8.1 Project Partner Requirement

The sensors and electronics must not break when the sensor is placed in the outdoor environment of the customer.

5.2.8.2 Engineering Requirement

The enclosure will not allow water inside under moderate rain conditions (0.10 - 0.30 inches of rain per hour) for over 24 hours.

5.2.8.3 Verification Processes

To test the entire enclosure in rainy conditions outside for 24 hours and then check if water made it into the enclosure. Since there was no consistent rainy weather and no place we could safely leave the sensor unit for the test period, we placed the enclosure in a bathtub for 24 hours. We also turned on the shower head for one minute every 6 hours to simulate the required rainfall. This process followed these steps:

1. Run the shower uninterrupted for one minute. Collect all of the water into a bucket or other measuring device.
2. Ensure the amount of water from the shower head is more than 0.1 inches per hour.
    a. i.e. if 0.6 inches of water are collected, run the shower at least every six hours.
3. Place the box in the bathtub, start the timer, and run the shower for the first minute.
4. Continually run the water over the course of the 24-hour period.
5. Open the box at the end for a visual inspection of the enclosure and hardware. There should not be any moisture in the enclosure and the system should still be operational.

5.2.8.4 Testing Evidence

The testing evidence for this section is a series of four videos proving that the shower head was turned on every 6 hours for 60 seconds. First, I measured that my shower head puts out about 8 liters of water every 60 seconds. This is equivalent to 3 - 4 inches of rain every 6 hours, which is more than enough for the set requirement of 0.10 - 0.30 inches of rain per hour as stated in section 5.2.8.3. Videos references [11-14], linked in section 5.3, prove that “rain” occurred every 6 hours. Video reference [15] shows the enclosure shortly after the test in a video [14] and proves that the system is still running and no water entered the enclosure. Figures 5.2-23 to 5.2-27 below show time stamps proving when the “rain” occurred and the water-free enclosure.

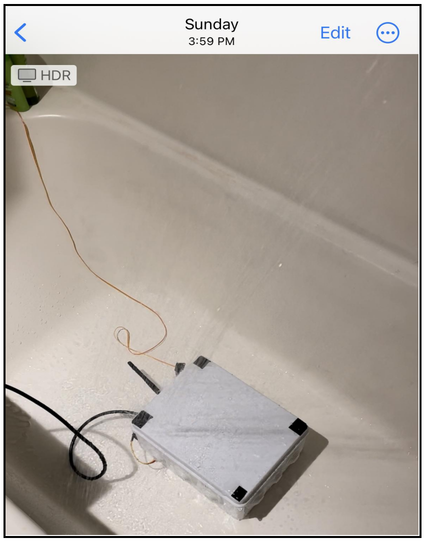


Figure 5.2-23: Timestamp proof of “rain” after 6 hours

Figure 5.2-24: Timestamp proof of “rain” after 12 hours

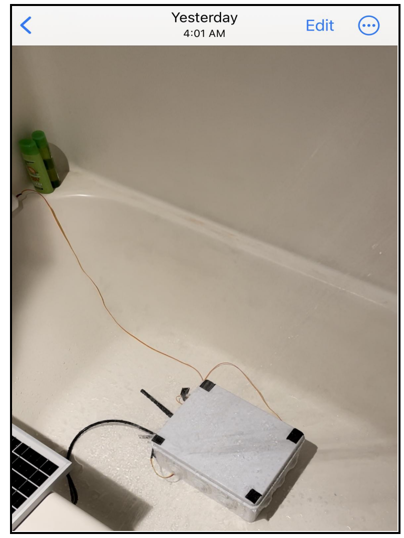


Figure 5.2-25: Timestamp proof of “rain” after 18 hours

Figure 5.2-26: Timestamp proof of “rain” after 24 hours

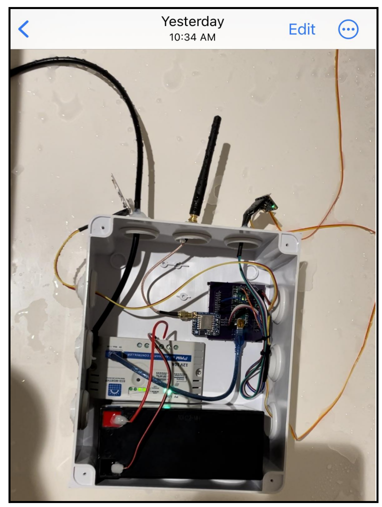


Figure 5.2-27: Timestamp proof enclosure being waterproof after 24 hour test

5.3 References & File Links

[1] "Bill of Materials," *Excel Spreadsheet* [Online]. Available: https://docs.google.com/spreadsheets/d/1tthBiTt9ZK8cjy0BLbihnEUPt_epiSR_avoK7uvaK7M/edit?usp=sharing. [Accessed 3-May-2023].

[2] "Solar Charge Controller User Guide," *Document scan* [Online]. Available: https://drive.google.com/file/d/1CP684vjrpmw5ma0IPD2ph5LgOUoE2_il/view?usp=share_link. [Accessed 4-May-2023].

[3] "GUI Requirement Video", *Youtube Video* [Online]. Available: https://www.youtube.com/watch?v=FBAv5AhQp84&ab_channel=BlakeLeeGarcia. [Accessed 7-May-2023].

[4] "application.zip", *Compressed ZIP File* [Online]. Available: https://oregonstate.box.com/s/bx2q04zdpsmba1w0bwt25c75kd28bfhd. [Accessed 7-May-2023].

[5] "environmental_testing_setup.MOV", *Video File* [Online]. Available: https://drive.google.com/file/d/1jSb3RHSHcZPtoLrsTJDob64A2LFPRgci/view. [Accessed 9-May-2023].

[6] "wind_speed_test.MOV", *Video File* [Online]. Available: https://drive.google.com/file/d/1InE16x-JXdivEbytY7mw22w8BijyBPM6/view. [Accessed 9-May-2023].

[7] "time_lapse_sending_frequency.MOV", *Video File* [Online]. Available: https://drive.google.com/file/d/1KFJFiFzxeAZD0gsNto2UmOB6Hf_Mbbwk/view. [Accessed 9-May-2023].

[8] "time_lapse_sending_frequency.csv", *Comma-Separated Value File* [Online]. Available: https://drive.google.com/file/d/1z9XRsx3NJ3z7Z2q4FM0VWn5Y--37pD1u/view. [Accessed 9-May-2023].

[9] "User Guide - Assembly Manual", *Google Docs File* [Online]. Available: https://docs.google.com/document/d/1RK-p2FFcgupmwx4BLD9Sd_0z8Aem9MsTlsiB0qlVICQ/edit?usp=sharing. [Accessed 9-May-2023].

[10] "User Guide - GUI Application", *Google Docs File* [Online]. Available: https://docs.google.com/document/d/1W_nz7PYnNcfc10-aih8cZHH0EWD4HIweg3crf67pCu8/edit?usp=sharing. [Accessed 9-May-2023].

[11] "Weatherproof_test_6_hours", *Video File* [Online]. Available: https://drive.google.com/file/d/10FUcbu_gTUWgRUXQWuC4nBJSJIxqOPpV/view?usp=sharing. [Accessed 9-May-2023].

[12] “Weatherproof_test_12_hours”, *Video File* [Online]. Available: https://drive.google.com/file/d/1GiZG3LSdhfIILGSsEuNuIiTKcw_pyc7-/view?usp=sharing. [Accessed 9-May-2023].

[13] “Weatherproof_test_18_hours”, *Video File* [Online]. Available: https://drive.google.com/file/d/1SQPND0WL56KPGRtgqVrNjWxEn8DLAYVK/view?usp=sharing. [Accessed 9-May-2023].

[14] “Weatherproof_test_24_hours”, *Video File* [Online]. Available: https://drive.google.com/file/d/1Z0eRZ9xqcHt8N8DThxv_bPB84J-ww0EY/view?usp=sharing. [Accessed 9-May-2023].

[15] “Final_Weatherproof_24_hours_completed”, *Video FIle* [Online]. Available: https://drive.google.com/file/d/1CSvTpvLbvaBpxkfqeVZjzKkKBxxWK_Je/view?usp=sharing. [Accessed 9-May-2023].

5.4 Revision Table

Date	Name	Revision Made
5/11/2023	Garren	Replaced temperature and humidity plots with neater versions
5/10/2023	Grace	Finished edits to section 5 - grammar, figure labels, etc.
5/9/2023	Grace	Finished weatherproofing verification and added all relevant links to the reference section
5/9/2023	Blake	Added the GUI testing evidence and reformatted some references to match IEEE standards
5/5/2023	Grace	Added the weatherproofing verification information
5/4/2023	Grace	Fixed formatting in section 5
5/4/2023	Grace	Added system image to 5 and references
5/3/2023	Garren	Added PCB schematic
3/13/2023	Garren	Added pictures for range requirement
3/12/2023	Grace	Wrote section 5.1.4
3/10/2023	Blake	Added verification process for 5.2.2.1
3/10/2023	Garren	Added verification process for 5.2.2.2
3/10/2023	Blake	Reformatted section 5.2 with the same information as before
3/10/2023	Blake	Added the Universal Constraint section and completed section 5.1 (except for subsection 5.1.4)
3/8/2023	Emma	Added each of the initial requirements (with filled in fields for partner and engineer)

Section 6 – File Closing

6.1 Future Recommendations

6.1.1 Technical Recommendations

Embedded Web Server - The project partner wanted the data accessible without having the raw values stored in a local file, but due to the high costs of the necessary hardware (i.e. a Raspberry Pi, costing over $150 on Amazon [1]) at the time of designing the system it was not feasible. A small-scale computer with a USB port could read the serial data from the host node and upload it to a web server on top of which a graphical interface would display the data.

Custom Part Layouts - Creating custom layouts of each module could reduce bulk manufacturing costs by removing any unneeded components on the modules and purchasing the printed circuit boards in bulk. Additionally, a single piece of hardware combining the radio module, student-built PCB, and Arduino Nano could ease the assembly complexity because the connections between the blocks would not require external wires. This alteration to the project would also give future flexibility to incorporate external hardware; for example, adding a module of EEPROM or flash memory to store node-specific data in the case of power loss or network disconnection. These are relatively cheap (for example, [3] is $0.17) and can easily fit onto a PCB.

Static Node Addresses - Right now each node identifies itself with an address (1 byte of data) given by the host node during the network initialization procedure. If any node loses connection for a full measurement cycle the network automatically detects and drops them from the expected responses; to return to the network the node has to undergo the address procurement process again. This system effectively deals with nodes losing power or getting moved out of range but it would cause significant confusion to the user if a temporary condition interferes with the regular transmissions and the same node randomly changes the identifier. One suggestion to get around this problem is to define an additional address request message capable of asking for the previous address the node had. This could get saved in ROM to maintain the same network addresses even after a power down.

Two-Way Node Communication - The final system reads the serial port from the Arduino Nano USB connection into an interface. Adding the capability to talk back to the host node would allow for more user customization of the network. Two quick examples of usefulness include a process for the user to instantaneously request data on the interface and to add more informative names. Every message on the network identifies itself with the single-byte address to reduce the amount of wireless data and increase the likelihood of successful transmissions, but these limitations do not exist in the wired connection between the host node and a computer. In other words, the user program could send a message to the host node to label node address 7 as “East Field” so whenever a sensor reading from node 7 arrives the file labels the data as “East Field” instead of seven.

[1] “Raspberry Pi 4 Model B 2019 Quad Core 64 Bit Wifi Bluetooth (4GB)”, *Amazon Listing* [Online]. Available: https://www.amazon.com/Raspberry-Model-2019-Quad-Bluetooth/dp/B07TC2BK1X/ref=asc_df_B07TC2BK1X/?tag=&linkCode=df0&hvadid=380145854123&hvpos=&hvnetw=g&hvrand=3237457730203903640&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1024429&hvtargid=pla-818643320764&ref=&adgrpid=85982211068&th=1. [Accessed 11-May-2023].

[2] “M24C02-FMN6TP”, *Digi-Key Listing* [Online]. Available: https://www.digikey.com/en/products/detail/stmicroelectronics/M24C02-FMN6TP/4729279. [Accessed 11-May-2023].

6.1.2 Global Impact Recommendations

Cost Reduction - As it stands, with the current state of our economy, each unit will cost just over $200 (USD). If these units are used on a global scale our team would suggest finding a way to build the sensors in-house or find a retailer willing to give a mass production discount.

More Sensors - The addition of other sensors, such as a soil sensor [3], would provide more workable data to the farmers. The GUI would have to be updated to accommodate these changes as well, but everything in this system is scalable so that won’t be a challenge.

Additional Units - With the addition of new sensor units in the network make sure the code allows for all of the units, properly addresses the units, and that the hub can store all of the incoming data.

Dedicated Hub - As this network grows, it will become more important to have a dedicated station for receiving and parsing through the incoming data. We would suggest a laptop dedicated to this task or something similar running the application we created.

[3] “TEROS 12 Advanced Soil Moisture Sensor + Temperature and EC”, *Meter Listing* [Online]. Available: https://www.metergroup.com/en/meter-environment/products/teros-12-soil-moisture-sensor?creative=428743883689&keyword=soil+moisture+sensor&matchtype=p&network=g&device=c&gclid=Cj0KCQjw3a2iBhCFARIsAD4jQB2EacwhedAtiHTcynxCJkYTrIEdQtqzQaKK41HQuAIIBgj_EIlbPAoaAoeSEALw_wcB. [Accessed 11-May-2023].

6.1.3 Teamwork Recommendations

Group Chats - Our team found it really important and much easier to communicate with each other by forming a group chat. This makes it easy to address any potential concerns that may come up as well as informing each other of issues that we run into and may need help with. If possible, creating a group chat with the project partner is also very helpful to ensure efficient communication between them and the team so everyone is well-informed on meeting locations and times, and anything that needs to be addressed regarding the project and its progress.

Joint Virtual Workspace - One thing that also worked really well for our team is setting up both a google drive that we shared amongst our team members, but also between our team and our project customer so he was able to be easily updated on any progress that we made. This makes it significantly easier to collaborate with one another. The initial process of setting up was quite difficult, but we eventually found a reference [4] walking us through the process.

Weekly Project Partner Updates - Although the capstone class itself has a set number of assignments dedicated to updating the project partner, we found that our project partner and customer both appreciate being updated every week. This makes communication easier between our three different parties. This also helps them be better informed on what we are able to accomplish and provides them with a good idea of what we are going to work on and achieve in the near future.

Weekly Team Meetings - We have found that meeting one day every week, even if just for an hour, is a great way to facilitate a good working environment. Whether we're doing something small like a project partner update that only takes around a half hour, or we're working on building and testing our actual project for four hours, it helps us build a good culture within our team. This, in turn, makes communicating any issues we have with our project, or conflicts within our group easier as we're very comfortable around one another.

[4] "Create a Shared Google Drive", *Google Workspace Learning Center Document* [Online]. Available: https://support.google.com/a/users/answer/9310249?hl=en. [Accessed 11-May-2023].

6.2 Project Artifact Summary

The main project artifacts include the firmware source code for the microcontrollers, the software source code for the Graphical User Interface, the schematic and gerber files for the custom printed circuit board, and the Bill of Materials. The group has also written and provided two user manuals; one describing the assembly process and one detailing the use of the application. Each artifact has a link below to view and download the information.

All Source Code: Github Link
PCB Schematic Files: Zip File Link
BOM / Part Information: Link to Google Sheets
User Interface Guide: Google Docs Link
Assembly Guide: Google Docs Link
Network Protocol Overview: Google Docs Link

6.3 Presentation Materials

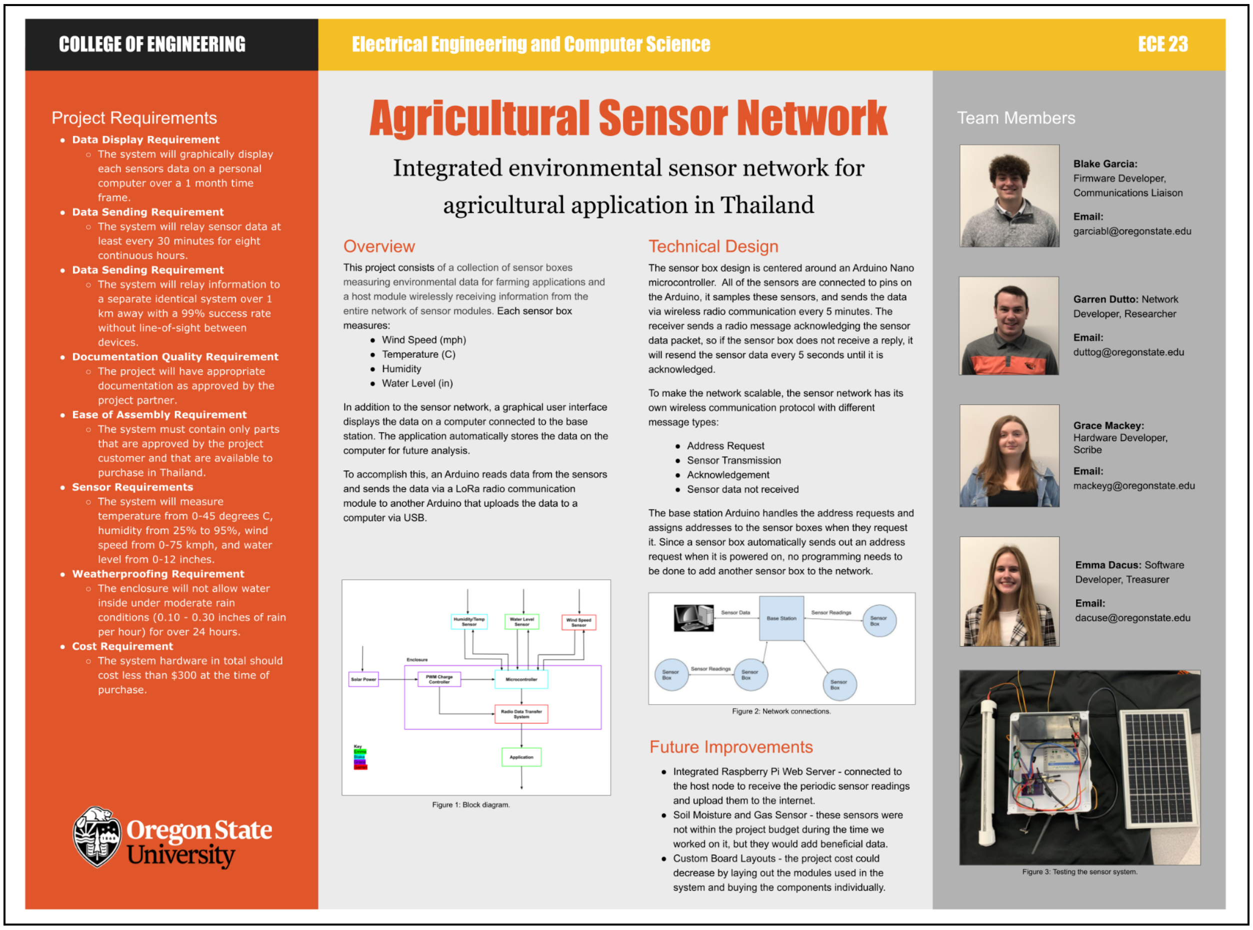


Figure 6.3-1: Expo Poster

Project Showcase: [Link]