Wireless Solar Powered DC Valve Controllers

**Authors: Ekaterina Rott, Orion Hollar,
Isaac Goshay, Salem Al Mazrouei**

Fall 2021 - Spring 2022

Table of Contents

1. Project Overview

1.1. Executive Summary

This document details the design and development process for a wireless farm irrigation control system that is powered by solar panels. Proper irrigation and watering is necessary for growth and return on investment on farms. To automate and simplify this process, our team has developed a solar powered system that will be controlled through an online web server that can remotely turn the irrigation system on and off.

The system is split into three main systems: the irrigation nodes, the control system hub, and the webserver. The control system hub communicates with a webserver to give the user an overview of the entire system, any errors occurring in the system, and control over the watering schedule. Solar powered irrigation nodes connect to solenoid controlled valves that stop or allow water flow through pipes. A primary hub wirelessly commands the irrigation nodes and requests data from them. Finally a web server connects to the primary hub which allows for watering schedules to be created to run automatically throughout the day by sending commands to irrigation nodes, show the current status of the system, and generate warnings if something is wrong in the system.

This project will be developed using a 30 week development cycle split into 3, 10 week stages. The first stage involves creating team standards and developing project requirements that fulfill the project partner's desires for the final product. During this first stage, we will create a block diagram and the high level descriptions of the system. The second stage involves the previously created diagram and high level descriptions and we will build and verify that our system works. The final stage will be a presentation stage, where we present the fully completed project and potentially transfer the technology developed for a future team to work on.

This document is split into seven sections. This first section of this document outlines the work structure within the group along with contact information for the members, the context for the role this project is meant to fill, and the timeline of the work invested in the project over the last 9 months. The second section of this document focuses on the requirements for the project to fulfill, the potential impacts our project will have, and the potential issues that could be encountered during the development process of this project. The third section outlines the design of the project, including the block diagram and descriptions along with the connections between the blocks and user input. The fourth through seventh sections have yet to be completed, and are part of future project development. The fourth section describes the verification testing done for each individual block, and the fifth section describes the verification for the entire system as a whole. The sixth section gives recommendations for future testing and development, and the final section is an appendix for additional reference materials.

1.2. Team Communication Protocols and Standards

Team Contact Information

Table 1.2.1: Team Contact Information

Team Member	Phone	Email
Ekaterina Rott	(971) 506-4558	rottek@oregonstate.edu
Orion Hollar	(360) 771-1875	hollaro@oregonstate.edu
Isaac Goshay	(808) 728-0852	goshayi@oregonstate.edu
Salem Almazrouei	(541) 908-7167	almazrsa@oregonstate.edu

Team Roles

Table 1.2.2: Team Roles and Descriptions

Role	Role Description	Role Owner
Team Meeting Coordinator	Coordinates Team Meetings, generates agendas, and takes meeting notes. Creates a list of the coming assignments, reminds the team about future assignments.	Ekaterina Rott
Project Partner Communicator	In charge of creating project partner communication and leading project partner interviews.	Orion Hollar
Chief Finance Officer	Maintains the budget, ensures that the team is not overspending	Isaac Goshay
Integration Officer	Regularly checks in with the team to insure everyone is hitting their deadlines for system integration, helps identify resources if a team member is falling behind	Salem Almazrouei
Parts Supplier	Is the primary contact for ordering parts, to ensure that minimum money is spent on shipping	Isaac Goshay
Project Progress Video Owner	Creates the slides used in progress videos, filling in areas of progress over the past week as well as future assignments to be looking towards in the next week.	Orion Hollar

Team Standards

Table 1.2.3: Team Standards and Protocols

Topic	Protocol	Standard
Team Meetings	Team Meetings will occur in person or via discord once a	Team Meetings are scheduled via a WhenIsGood poll every

	week, where the team members will check in and deliver progress reports as necessary.	week, occurring on Saturday or Sunday unless a team member is unable to attend one weekend.
Meeting Notes	Notes will be taken at every meeting via an agenda document hosted on the team google drive.	During meetings all members should have access to either the agenda or other project documents to ensure everything is recorded.
Team Google Drive	All Official documents should be placed into the team drive.	All documents should have an appropriate name and placed into the correct folder.
Team Discord	Using the App helps us to communicate with each other, plan for upcoming tasks or deadlines to be completed, plans for meetings, and share research.	Every team member can ask questions related to this project using this page and it helps us to check when the coming meeting is or any information about this project.
Weekly Tasks	Weekly tasks are self or group assigned at the team meetings.	Weekly tasks are to be finished by the created due date, unless the team member communicates that more time is needed, that they need more resources, or need help from another team member to complete it.
Absences	If a Team Member needs to be absent from a Team Meeting, they should notify the other team members via the discord as early as possible, preferably the night before.	A Team Member should not be absent from more than 2 Team Meetings a term. They are responsible for catching up on what the team has done via reading over the meeting notes and any relevant documentation.
Project Deadlines	Major Project Deadlines that are not Weekly Tasks should be noted in the Project Timeline, along with the due dates and deadline owner. Deadlines should be noted and discussed when necessary in Team Meetings.	A Team Member who is a Deadline Owner should notify the team as soon as possible if they suspect they will not be able to complete a deadline on time.
Project Partner Check-Ins	During meetings gather questions and progress for reports. Orion will take the info gathered to write up a check in email and send it by the following	Check-in emails should contain all questions gathered and any relevant information to the project partner from the meeting in a clear and concise manner.

	Wednesday.	
Modification Documentation	Any major software and hardware issues need to be documented in the Modifications document	Any changes should at least have a comment of what the problem was. Also to have it listed into the Modification document so that everyone else knows.
Weekly Video Records	The Team will create weekly progress videos starting Week 4 of Fall Term, 2021, and make a plan every Team Meeting regarding how it will be recorded, when, and what will be said.	Weekly plans for the videos will be created during the Team Meetings.
Project Timeline	The Project Timeline is a living document hosted on the Team Drive that describes all Major Project Deadlines and the critical path to project completion.	Any changes to the Project Timeline should be reviewed by the team.
Timesheet	The Timesheet is a living document hosted on the Team Drive that records the time spent by the team doing various tasks related to the project.	Team Members are responsible for filling out their tab of the Timesheet as accurately as possible

Communication Analysis

- Project Partners' interest in the project and main role(s): Mr. and Mrs. Small Farm are our primary customers and stakeholders
- Project Partners' profession/company: Small Farmers on a 5 acre farm
- Main types of information the Project Partners will want to know and why: They would like to hear about our timeline, if we are remaining within budget, and high-level summaries of our work.
- Project Partners' level of technical knowledge and terminology related to your project: Mr. and Mrs. Small Farm do not have a lot of technical knowledge and do not know most technical jargon.
- Preferred format and frequency for communication of various types of information: Bi-weekly emails for the requested project check-ins, phone calls for any urgent issues.
- Others who your Project Partners may show various types of communications and their levels of technical knowledge:

1.3. Gap Analysis

Small farms have a need for a low maintenance irrigation control system that would be able to withstand the thick cloudier months of October, and the rain in December [1]. Currently most systems are able to turn on the irrigation valves, but most run only on battery power [2], meaning that eventually they will have to recharge or even replace the battery. Our low maintenance system will charge the battery with solar power removing the need to replace the battery. The remote controlled system would also give the farmers control over the system, giving the farmers the space to work on other, more prominent, issues. The system will run automatically for long periods of time without technical maintenance.

A farm of this size (5 acres) is a hobby farm meant to feed the farmer and some friends [3]. This means that the user will be away for a large section of the day and would not be able to regularly check in on the crops, or their equipment. With the wireless communication between the control hub and irrigation node the user will be able to tend to their farm while they are gone. Also they would be able to get alerts concerning their field and equipment, and actively control the irrigation schedule. Our product helps avoid water waste and saves time, making small farming more affordable and accessible.

1.4. Project Timeline and Work Structure Breakdown

Stage 1

Week	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Dates	9/22 - 9/25	9/26 - 10/2	10/3 - 10/9	10/10 - 10/16	10/17 - 10/23	10/24 - 10/30	10/31 - 11/6	11/7 - 11/13	11/14 - 11/20	11/21 - 11/27	11/28 - 12/4
Task 1.1			Team Standards								
Task 1.2				Team Risks							
Task 1.3				Gap Analysis							
Task 1.4				Executive Summary							
Task 1.5					Project Requirements						
Task 1.6					Project Risk Analysis						
Task 1.7							Top Level Block Diagram				
Task 1.8								Block Descriptions			
Task 1.9								Interface Definitions			

Stage 1 Task Descriptions

Task	Champion	Stakeholder	Status
1.1 Team Standards	Ekaterina Rott	Team	Completed
1.2 Team Risks	Salem Almazrouei	Team	Completed
1.3 Gap Analysis	Isaac Goshay	Instructors	Completed
1.4 Executive Summary	Orion Hollar	Instructors	Completed
1.5 Requirements	Orion Hollar	Project Partner	Completed
1.6 Risk Analysis	Salem Almazrouei	Team	Completed
1.7 Block Diagram	Orion Hollar	Project Partner	Completed
1.8 Block Descriptions	Ekaterina Rott	Instructors	Completed
1.9 Interface Definitions	Isaac Goshay	Instructors	Completed

Stage 2

Week	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Dates	1/3 - 1/8	1/9 - 1/15	1/16 - 1/22	1/23 - 1/29	1/30 - 2/5	2/6 - 2/12	2/13 - 2/19	2/20 - 2/26	2/27 - 3/5	3/6 - 3/12
Task 2.1	Block 1 Verification									
Task 2.2					Block 2 Verification					
Task 2.3									System Level Integration	

Stage 2 Task Descriptions

Task	Champion	Stakeholder	Status
2.1 Block 1 Verification	Team	Instructors	Complete
2.2 Block 2 Verification	Team	Instructors	Complete
2.3 System Level Integration	Team	Team	Complete

Stage 3

Week	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Dates	3/28 - 4/2	4/3 - 4/9	4/10 - 4/16	4/17 - 4/23	4/24 - 4/30	5/1 - 5/7	5/8 - 5/14	5/15 - 5/21	5/22 - 5/28	5/29 - 6/4
Task 3.1	Presentation Preparation									
Task 3.2	System Final Touches									
Task 3.3					Project Closing					
Task 3.4										Tech. Transfer

Stage 3 Task Descriptions

Task	**Champion**	**Stakeholder**	**Status**
3.1 Presentation Preparation	Ekaterina Rott	Instructors	Complete
3.2 System Final Touches	Ekaterina Rott	Instructors	Complete
3.3 Project Closing	Ekaterina Rott	Project Partner	Complete
3.4 Technology Transfer	Orion Hollar	Project Partner	Complete

1.5. References and File Links

1.6. Revision Table

Table 1.6.1: Section 1 Revision Table

Date	Changes Made	Author
10/15/2021	Outline, Executive Summary, and Team Contacts added	Ekaterina Rott
10/17/2021	Team Standards, Communications Analysis, and Gap Analysis added	Ekaterina Rott
10/17/2021	Team Standards, Communications Analysis, and Gap Analysis added	Orion Hollar
10/17/2021	Team Standards, Communications Analysis, and Gap Analysis added	Isaac Goshay
10/17/2021	Team Standards, Communications Analysis, and Gap Analysis added	Salem Almazrouei

10/17/2021	Timeline Created	Ekaterina Rott
10/22/2021	Gap Analysis Revision	Ekaterina Rott
10/31/2021	Executive Summary Edits	Orion Hollar
10/31/2021	Gap Analysis Edits	Isaac Goshay
11/11/2021	Executive Summary Final Revisions	Orion Hollar
11/11/2021	Gap Analysis Final Revisions	Isaac Goshay
11/19/2021	Revision section 1	Salem Almazrouei
11/19/2021	Executive Analysis minor Revision	Orion Hollar
12/3/2021	Timeline Overhaul	Ekaterina Rott
05/05/2022	Timeline Completion	Ekaterina Rott

2. Requirements, Impacts, and Risks

2.1. Requirements and Constraints

2.1.1. Project Partner Requirements

- The system shall be solar powered.
- The system will have nodes that are 600 feet apart, and must communicate wirelessly between nodes.
- There will be a webpage that will display the status of the system, along with available schedules for turning the irrigation on and off.
- The system should not get in the way of proper functioning of farm equipment.
- There will be a user's guide that defines basic uses, set up, and troubleshooting.
- The system will be controlling DC solenoid set up at the ends of irrigation pipes.
- The webpage shall allow for irrigation schedule creation.

2.1.2. Engineering Requirements

Requirement 1: The system shall be powered directly by batteries and the batteries will be charged via a solar panel.

Verification: Demonstration

1) Solar Charging Test:
    a) Starting with the batteries below full charge, plug them into the solar charging PCB. Note the voltage of each battery cell.
    b) Connect the solar panel.
    c) Wait 20 minutes and check to see that the battery pack has increased in overall voltage with the solar panel disconnected.
2) Power via Battery Pack Test:
    a) Connect the batteries to the solar charging PCB.
    b) See that the Radio Communication board and 5V Buck Converter board LEDs are powered.
    c) Proceed through at least one opening and closing of the valve.

Pass Condition: If both the Solar Charging test and Power via Battery Pack test are complete and passed, the requirement is fulfilled.

Requirement 2: The system can communicate between nodes 600 ft apart.

Verification: Demonstration

1) Start with the nodes turned on and at least 10 ft apart.
2) The message should go through and the short range communication will receive the message and send it to the website.
3) Looking at the website, examine the "Current Node Status" table to make sure that a message has been received. [WSPDCVC Homepage (oregonstate.edu)]
4) Then move the node to at least 600 ft apart.
5) Again the message should be sent and received and the short range communication will receive the message and upload it to the website.
6) Examine the website to make sure that a message has been received.

Pass Condition: With both the close range and long range test successful, this requirement is verified.

Requirement 3: The system creates customizable automatic watering schedules through a web interface that shows the current status of the system.

Verification: Demonstration

1) Starting with both the watering schedule and the system data databases empty [WSPDCVC Homepage (oregonstate.edu)], navigate to the system scheduler page [Schedule Creator (oregonstate.edu)].
2) New Watering Schedule Test:
    a) Input a New Default watering schedule. Through inspection, see that the "Current Watering Schedule" table has changed.
    b) Input a second watering schedule, do not set it as Default. Through inspection, see that the "Current Watering Schedule" table has not changed.
3) Watering Schedule Manipulation Test:
    a) Navigate to the current schedule manipulation page webpage [Current Schedules (oregonstate.edu)], and perform the following tests:
    b) Change the current default schedule. See on the "Current Watering Schedule" that the default has changed.
    c) Change the current default schedule to no longer be default, such that there is no current default. See on the "Current Watering Schedule" that the default has changed.
    d) Set one of the two stored schedules to be default. See on the "Current Watering Schedule" that the default has changed.
4) System Status Test:
    a) Navigate to the system graphics page. Inspect the "Current Node Status" table. [Graphics Page (oregonstate.edu)]
    b) If the communication systems are running, the "Current Node Status" table will update 4 times an hour. Verify visually with the system turned on.
    c) If communication systems are not running, the "Current Node Status" table should say "No current system data".

Pass Condition: If the New Watering Schedule test, Watering Schedule Manipulation test, and System Status test pass, this requirement is passed.

Requirement 4: The system adjusts the valve within 5 seconds of the scheduled time.

Verification: Demonstration

1. Use the website to schedule a valve change every minute for the next 5 minutes.
2. Start a timer at the next minute, as displayed by a computer clock.
3. When the solenoid changes, mark the time and write it down.
4. After all prepared tests have been completed, compare the written times and the scheduled times.

5. Next, pull the SD card from the system, and inspect the schedule times to when the opening command for the SD card was sent.

Pass Condition: If a minimum of 90% of both the physical opening times and the opening command times are within 5 seconds of the scheduled time, this requirement is fulfilled.

Requirement 5: The system comes with a user guide that includes a title page, table of contents, introduction, explanation of system, system components, installation instructions and troubleshooting. [4].

Verification: Inspection

1) Present a hard copy version of the user's guide, with an online copy stored on google docs as a backup.
2) Going through the user's guide with our designated point-person, Ingreed Scheel, ensure that the user's guide includes all necessary sections.

Pass Condition: The requirement is fulfilled with point-person approval.

Requirement 6: The system sends messages through nodes with a wireless communication system that will successfully transmit 80% of sent messages.The system will send wireless confirmations that messages are received and re-attempt messages if confirmations are not received.

Verification:

1) Have 10 messages to be sent between the watering node and the base node prepared.
2) The messages should be sent from the radio of the base node to the short range communicator and from there the message should be sent to the website.
3) If at least 8 of the 10 messages are shown on the website, the test is passed.
4) Then, turn off the watering node to simulate a node being unable to receive messages.
5) The base node will give the message to the short range communicator and upload it to the website. This message will include how many retries in the message for us to see the "Error" column of the "Current Node Status" table.
6) If the node performs 3 retries before stopping, the test is passed.
7) With both step 3 and step 6 successfully passed, the requirement is fulfilled.

Pass Condition: If the message received by nodes and displayed on the website is greater than 80% of the messages sent and the retries error ends at 3 tries, then the system is working as specified.

Requirement 7: The system records messages sent and received by the system to an internal data backup, holding at minimum 1 month of messages.

Verification: Analysis

1. Run 3 tests where a different total number of messages are written to the SD card.
2. Using the number of messages sent and the total size of the output file, find the size of one message for each test. Average the 3 tests to get the average size of one message.

3. Multiply the average message size by the total number of messages per an average month, ensure that the total space required for the month is less than the space on the SD card.

Pass Condition: If the total space utilized is less than the space on the SD card, this requirement is fulfilled.

Requirement 8: The system circuitry will be protected by an enclosure that is water and dust proof, such that low pressure or low velocity water coming from all angles and dust larger than 1.0mm cannot penetrate the enclosure.

Verification: Demonstration

1) Begin with the system fully enclosed, with a paper towel inside the enclosure to catch any water that enters.
2) Mount the enclosure to a wooden pole as it would be mounted on a farm.
3) Using a hose, spray low velocity water onto the enclosure from various angles.
4) Toss soil onto the enclosure.
5) Take the system down, and remove the solar panel. Inspecting the inside of the enclosure, ensure that the paper towel is not wet, with no water with the enclosure otherwise, and no dirt within the enclosure.
6) If step 5 is successfully passed, the requirement is fulfilled.

Pass Condition: If there is no water and or any solid larger than 1.0 mm inside of the enclosure, the requirement is considered verified.

2.2. Design Impact Statement

Introduction:

The purpose of a Design Impact Statement is to explore and understand the economic, environmental, cultural, and public health and safety impacts of our design. The project is focused on enabling small farms to control their irrigation systems from afar with a wireless, solar powered irrigation control system. Our project encompases many aspects of engineering, from wireless communication to solar panel usage, but also has the potential to affect many people, from our end user, small farmers, to the people responsible for manufacturing the parts used in our product. This analysis will cover five impacts of our design, and conclude with some recommendations on how to mitigate potential negative impacts.

Public Safety Impacts:

One of the primary components of the project design is a solar panel intended to provide power so the user does not have to charge or replace on-board batteries. There are different styles of solar panels, which have different materials used in their construction, and all solar panels have some percentage of glass, aluminum, and silicon [6]. Silicon is easier to recycle than it is to manufacture on its own, and since a significant percentage of solar panels is made of silicon, it would be beneficial for tech companies to use recycled silicon. Glass and aluminum also have many recycling options and uses, and they can be repeatedly recycled. Recycling ensures that those materials are properly reprocessed, keeping any toxins out of the environment, while providing access to production companies to high quality (but cheaper) raw materials. Therefore, it is important to detail proper recycling procedures and how to find e-waste recycling locations for the end user, as we will not have direct control over how the product gets disposed of.

Welfare Impacts:

Another primary component of the project design is custom built PCBs, which come with their own realm of potential impacts. According to an Occupational Health and Safety Assessment Series (OHSAS) survey, the Taiwanese printed circuit board companies are only motivated to implement OHSAS policies due to their own customer requirements [7]. While OHSAS policies are not legally required, they do provide effective health and safety management systems meant to protect the workers. According to Chen et. al., success and failure of companies adopting these policies can be drilled down to top management's commitment and support and collaboration among company personnel [7]. Since we will be utilizing more than one custom PCB, it is our job as designers to be cognizant of who we are supporting with our purchases to ensure that we support a workplace that is safe for its workers.

Cultural Impacts:

While culture may not be what immediately comes to mind for farming, it is important to some communities and automating part of the farming process may actually take some of the purpose out of the activity. The study “Why Garden?” by Nancy Sonti found that most of the reasons behind community gardening were non-practical. Instead they were more focused around the experience, such as almost three fourths of the responders citing joy or getting to watch growth or simply loving the activity, largely focused on investing time and getting progress for their work [8]. Some also enjoyed being in the garden, since it is a step away from the urban dull and gray. Keeping those purposes in mind, our project would not only be unhelpful towards fulfilling what people get from the activity, but actively subtract from their experience by taking away one of the activities that they are able to enjoy in the space.

Environmental Impacts:

A common conversation around electricity and the environment is the impacts of fossil fuels on the air, water and soil. Liddle and Sadorsky [9] conducted research investigating how an increase in non-fuel usage in electricity generation reduced carbon dioxide emissions. The study results indicated that a 1% increase in non-fuel usage in electricity production decreases carbon dioxide emissions per capita by 0.82% [9]. These statistics suggest that adopting a technology using non-fuels like solar can help reduce such emissions, which could help mitigate global warming and ultimately climate change. As our project supports the use of small-scale solar power, we are ultimately supporting the overall trend of switching from natural gas and crude oil to solar, hydro, and wind based energies.

Economic Impacts:

Farming is humankind’s oldest profession, and there are political and economic ramifications that follow behind agricultural growth. Agricultural economic policy, on a broad level, benefits large farms and hurts small ones. Technologies that benefit small farms - such as irrigation control systems that help small farmers, who aren’t able to hire as many workers as large farms, for example - therefore work against this trend, benefiting small farms at the potential expense of large ones. While it may seem like empowering small farms has negative economic repercussions by encouraging a less productive company, small farms are actually no less productive than large ones [10]. Therefore, our project has the potential to help small farms be successful and compete with large farms by providing a method to cheaply automate their farms.

Conclusion and Recommendations:

Our project encompasses many aspects of engineering, and with those comes many global impacts that we, as designers, need to understand and mitigate within our project design. Some things are outside of our control, such as how a user chooses to dispose of our product at the end of its life cycle, but other things are, such as who we choose to purchase our PCBs from. There are potential positive impacts from our design, such as reducing reliance on fossil fuels and helping small farms build a competitive edge in the economy. After considering many potential impacts of our system, we will focus on the following recommendations, while acknowledging that these do not cover all potential impacts. First, we will ensure there are proper recycling methods provided to the end user so they have access to the best way to dispose of the product at the end of its life cycle. Secondly, we would like to help users make an informed decision about the reasons they garden, to prevent this product from taking enjoyment away from the experience.

2.3. Risks

Table 2.3.1: Project Risks and Action Plans

ID	**Risk**	**Category**	**Risk Probability**	**Risk Impact**	**Performance Indicator**	**Responsible Party**	**Action Plan**
R1	**Global parts shortage makes it so we are not able to obtain necessary parts.**	**Schedule, Organization**	90%	**High**	Parts going out of stock, lengthy shipping times	Isaac Goshay	Avoid by ordering parts as early as possible
R2	**Oregon weather damages the internal circuitry of the system.**	**Environmental**	50%	**High**	Gaps in the enclosure, moisture inside the enclosure	Salem Almazrouei	Reduce risk by thoroughly inspecting the enclosure any any connectors to the enclosure for gaps
R3	**Enclosure is damaged or otherwise unusable.**	**Technical**	30%	**Medium**	Improper handling of the enclosure. Difficulty closing enclosure	Isaac Goshay	Reduce the impact of the risk by keeping the enclosure design accessible to order another
R4	**Something pulls a team member away from the project and they are unable to complete their deadlines.**	**Personal, Schedule**	80%	**High**	Missed deadlines, repeated missed meetings, not replying in team discord	Salem Almazrouei	Reduce risk by having weekly team check ins to ensure all team members have necessary resources
R5	**Environmental noise begins to interfere with Wireless communication between hub and nodes.**	**Technical**	75%	**Medium High**	Gibberish messages or no received messages	Orion Hollar	Reduce risk by leap-frogging comms between nodes, error messages on the web interface, repeat attempts to send messages
R6	**System does not receive**	**Environmental**	10%	**Low**	Batteries not charging	Ekaterina Rott	Transfer risk to end user by utilizing

	full 3 peak hours of sunlight.						warning messages on the web interface that indicate when the batteries hit a critical level
R7	**Batteries completely lose power.**	**Environ mental, Technical**	10%	**High**	Warning message about critical battery power level, system turns off	Ekaterina Rott	Transfer risk to user by utilizing notifications that a node is no longer responding
R8	**Some external circumstance prevents team meetings, ex, pandemic, extreme weather, family issues.**	**Environ mental**	33%	**High**	Messages in Team Discord indicate that the regular, in person meeting will not be possible.	Orion Hollar	Reduce risk by having back up meeting possibilities, e.g. Zoom, Discord.

2.4. References and File Links

[1] "WeatherSpark.com," *Corvallis Climate, Weather By Month, Average Temperature (Oregon, United States) - Weather Spark*. [Online]. Available: https://weatherspark.com/y/400/Average-Weather-in-Corvallis-Oregon-United-States-Year-Round. [Accessed: 12-Nov-2021].

[2] *Digital Farming and Smart Water Irrigation Controller from Netafim.* [Online]. Available: https://www.netafim.com/en/digital-farming/netbeat/. [Accessed: 12-Nov-2021].

[3] "Home," *Flexfire LEDs*. [Online]. Available: https://www.flexfireleds.com/led-ip-ratings-led-flex-strip-waterproofing-explained-waterproof-v-nonwaterproof-led-strip-lights/. [Accessed: 12-Nov-2021].

[4] "IEEE/ISO/IEC 26511-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering - Requirements for managers of information for users of systems, software, and services," *standards.ieee.org*. https://standards.ieee.org/standard/26511-2018.html (accessed Oct. 29, 2021).

[5] Hyder, Z., 2021. "What is a peak sun hour? What are peak sun hour numbers for your state?". [Blog] *Solar Reviews*, Available at: <https://www.solarreviews.com/blog/peak-sun-hours-explained> [Accessed 12 November 2021].

[6] Y. Xu, J. Li, Q. Tan, A. L. Peters, and C. Yang, "Global status of recycling waste solar panels: A review," *Waste Management*, vol. 75, pp. 450–458, May 2018, doi: 10.1016/j.wasman.2018.01.036.

[7] C.-Y. Chen, G.-S. Wu, K.-J. Chuang, and C.-M. Ma, “A comparative analysis of the factors affecting the implementation of occupational health and Safety Management Systems in the printed circuit board industry in Taiwan,” *Journal of Loss Prevention in the Process Industries*, 29-Jan-2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950423009000060?casa_token=eo3fVn3SEgIAAAAA%3ATmrhG-7tKKLAAoiZ6eKtFOgw0jZAaHKOWuqaufu0jzUSnChUNo7BLxoEp0sqE-YiDXSXpBAjfKE. [Accessed: 30-Oct-2021].
[8] L. S. Chalmin-Pui, A. Griffiths, J. Roe, T. Heaton, and R. Cameron, “Why garden? – attitudes and the perceived health benefits of home gardening,” Cities, vol. 112, p. 103118, 2021. https://www.tandfonline.com/doi/full/10.1080/08941920.2018.1484971
[9] B. Liddle, and P. Sadorsky, How Much Does Increasing Non-Fossil Fuels in Electricity Generation Reduce Carbon Dioxide Emissions? Applied Energy, 197, 212-221, 2017. https://doi.org/10.1016/j.apenergy.2017.04.025 [Accessed Oct 23, 2021]
[10] M. A. Altieri, “Agroecology, Small Farms, and Food Sovereignty,” *Monthly Review*, vol. 61, no. 3, p. 102, Jul. 2009, doi: 10.14452/mr-061-03-2009-07_8.

2.5. Revision Table

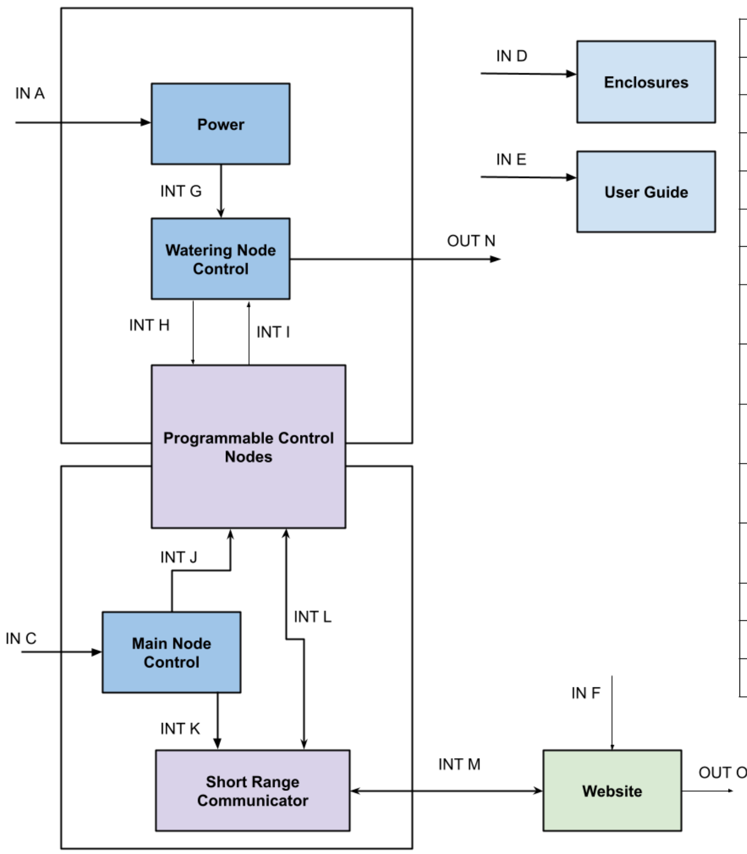
Table 2.5.1: Section 2 Revision Table

Date	Changes Made	Author
10/15/2021	Outline added	Ekaterina Rott
10/17/2021	Rough Requirements Created	Ekaterina Rott
10/17/2021	Rough Requirements Created	Orion Hollar
10/17/2021	Rough Requirements Created	Isaac Goshay
10/17/2021	Rough Requirements Created	Salem Almazrouei
10/24/2021	Risks Analysis added	Ekaterina Rott
10/24/2021	Risks Analysis added	Orion Hollar
10/24/2021	Risks Analysis added	Isaac Goshay
10/24/2021	Risks Analysis added	Salem Almazrouei
10/28/2021	Requirement Verification and Pass Conditions added for Requirements 5 and 6	Ekaterina Rott
10/29/2021	Requirement Verification and Pass Conditions added for Requirements 3 and 4	Orion Hollar
10/29/2021	Requirement Verification and Pass Conditions for Requirements 7 and 8	Salem Almazrouei

10/29/2021	Requirement Verification and Pass Conditions for Requirements 1, 2, and 9	Isac Goshay
11/10/2021	Requirements Revisions	Ekaterina Rott
11/11/2021	Risk Analysis Revisions	Salem Almazrouei
11/19/2021	Revision section 2	Salem Almazrouei
11/19/2021	Risk 3, 5 Revision	Orion Hollar
11/19/2021	Requirements Revisions	Orion Hollar
12/1/2021	Requirements Revisions	Orion Hollar
12/3/2021	Requirement Verification Clarification	Orion Hollar
12/3/2021	Requirement and Verification Revisions	Ekaterina Rott
12/3/2021	Requirement and Verification Revisions	Salem Almazrouei
05/03/2022	Adjusted Requirements to match Section 5	Ekaterina Rott
05/05/2022	Created the Design Impact Assessment	Ekaterina Rott
05/05/2022	Revised the Design Impact Assessment	Salem Almazrouei
05/05/2022	Revised the Design Impact Assessment	Orion Hollar
05/05/2022	Revised the Design Impact Assessment	Isaac Goshay
05/06/2022	Finished the Conclusion and Recommendations for the Design Impact Statement	Ekaterina Rott

3. Top Level Architecture

3.1. Block Diagram



Legend

Label	Interface Name
IN A	otsd_pwr_envin
IN B	otsd_mn_nd_cntrl_acpwr
IN C	otsd_enclsrs_envin
IN D	otsd_usr_gd_other
IN E	otsd_wbst_usrin
INT F	pwr_wtrng_nd_cntrl_dcpwr
ING G	wtrng_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr
INT H	prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig
INT I	mn_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr
INT J	mn_nd_cntrl_shrt_rng_cmmnctr_dcpwr
INT K	shrt_rng_cmmnctr_prgrmmbl_cntrl_nds_comm
INT L	shrt_rng_cmmnctr_wbst_rf
OUT M	wtrng_nd_cntrl_otsd_mech
OUT N	wbst_otsd_usrout

Figure 1: Top Level Block Diagram

3.2. Block Descriptions

<u>Block 1: Power</u>

- One of the project partner requirements was to power the system using a solar panel, to reduce the need for charging or changing the batteries to the watering node.
- The solar panel will be a 10W panel, connected to the charge controller, along with a 8V battery bank. The charge controller, which connects the solar panel and battery bank to the rest of the system, will be a DC-DC converter.
- Block Owner: Ekaterina Rott

Block 2: Watering Node Control

- The Node Control module will be the physical interface between the programmable control node, the power block, and the DC valve. This will include an on/off switch, a solenoid outside of the enclosure that controls a valve, a physical open/close push button to manually control the solenoid, and a PCB connecting them all.
- The PCB will be designed in KiCAD.
- Block Owner: Orion Hollar

Block 3: Programmable Control Nodes

- This block handles the long range communication between nodes as well as the code running the system. This includes receiving schedules from the website through the short range communicator, sending commands to the node control module to control the solenoid, ensuring the connection between nodes, storing a log of messages, and sending status updates to the website.
- This block will be using a Adafruit Feather M0 with RFM95 LoRa radio to communicate with each other on a 915 MHz frequency. We will also use the appropriate antenna length and the most suited for 915Mhz. There will be code that will control the radio for both receiving and sending information between the two nodes.
- Block Owner: Isaac Goshay

Block 4: Main Node Control

- This block converts AC power from a wall plug into usable DC power for the Programmable Control Node and the Short Range Communication blocks.
- Block Owner: Orion Hollar

Block 5: Short Range Communication

- For the Short Range communication it is going to be the wireless confirmations using the to send and receive messages for all the nodes and then all this data going to be saved onboard memory from the radios. It will receive communication from the Hub to the internet and throw all the Nodes. Also it will send all the data to the website to save it for a month and from there the website is going to send it the app.
- Since we do not have a project partner we can assume that Mr or Mrs farmer is going to have wifi or a good router and For this block we can use the feather m0 with winc1500 which needs wifi. Also there going to be some code that connect between the nodes in the Short Range Communication
- Block Owner: Salem Almazrouei

Block 6: Website

- The website is the primary user interface for the system, and will control the watering schedules and will display any errors present in the system.
- The website will be coded in MySQL and PHP, hosted on the OSU ENGR Web Servers.
- Block Owner: Ekaterina Rott

Block 7: Enclosure

- The enclosure will be used to house and protect the electronics from outside interference, conforming with IP 44 standards. Low velocity water and solids larger than 1.0mm shall not be able to enter the enclosure, and any connections entering and leaving the enclosure should be able to be disconnected.
- The enclosure will be designed in AutoCAD and 3D printed.
- Block Owner: Isaac Goshay

Block 8: User Guide

- The user guide is meant to provide the end user, someone with limited technical knowledge, the ability to troubleshoot and understand the system. The guide shall contain a high level summary of all system components and basic troubleshooting, along with any relevant set up and recycling information.
- The user guide will be a document that is written in Google Docs and then hosted on the website for future reference.
- Block Owner: Salem Almazrouei

3.3. Interface Definitions

Table 3.3.1: Interface Properties Table

Interface Name	Interface Properties	Diagram Label
otsd_pwr_envin	● Light: Minimum lux of 1000 ● Light: ~3 hr of sunlight per day on average ● Water: 8 oz of water poured directly from above	IN A
otsd_mn_nd_cntrl_acpwr	● Inominal: 20A ● Ipeak: 30A ● Other: Wall Socket Connection ● Vnominal: 120V	IN B
otsd_enclsrs_envin	● Other: The enclosure will be 7.55in length 6.9in width 4 in height. Main thing is to fit behind the solar panel ● Other: diameter of the holes for the wires will be 0.5in ● Water: Sprinkler spray and Rain make sure no water makes it inside	IN C
otsd_usr_gd_other	● Other: Section 1: Introduction ● Other: At least 10 pages ● Other: Section 4: Troubleshooting ● Other: Section 2: Installation procedure ● Other: Section 3: Operating the System	IN D
otsd_wbst_usrin	● Other: User will input the watering schedule.	IN E

	● Type: Typing and Clicking ● Type: Buttons, Scroll Menus	
pwr_wtrng_nd_cntrl_dcpwr	● Inominal: 500mA ● Ipeak: 550mA ● Vmax: 12V ● Vmin: 11.5V	INT F
wtrng_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr	● Inominal: 0.2 A ● Ipeak: 0.5 A ● Vmax: 5 V ● Vmin: 3.3V	INT G
prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig	● Logic-Level: 3.3 V (high) ● Other: Signal to turn motor open or closed	INT H
mn_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr	● Inominal: 0.05A ● Ipeak: 0.35A ● Vmax: 5.2V ● Vmin: 4.8V	INT I
mn_nd_cntrl_shrt_rng_cmmnctr_dcpwr	● Inominal: 0.05A ● Ipeak: 0.35A ● Vmax: 5.2V ● Vmin: 4.8V	INT J
shrt_rng_cmmnctr_prgrmmbl_cntrl_nds_comm	● Data Rate: 9600 bit/s ● Messages: Valve Status / New Schedules ● Protocol: UART	INT K
shrt_rng_cmmnctr_wbst_rf	● Data Rate: Messages will be sent a few times a minute. ● Messages: The data received will be in a .json file format. ● Protocol: HTTP POST or HTTP GET	INT L
wtrng_nd_cntrl_otsd_mech	● Other: Either extended or unextended, no in-between ● Other: Solenoid Extends > 0.5cm ● Other: Extends to 90% capacity within 1 second	OUT M
wbst_otsd_usrout	● Other: Website will display any error messages found in the json input files. ● Other: Website will display current watering schedule. ● Type: Website will display a visual that represents which watering nodes are open or closed.	OUT N

3.4. References and File Links

3.5. Revision Table

Table 3.5.1: Section 3 Revisions Table

Date	Changes Made	Author
10/31/2021	Section Outline Created	Ekaterina Rott
11/15/2021	Top Level Block Diagram added	Orion Hollar
11/17/2021	Block 1 description added	Ekaterina Rott
11/17/2021	Blocks 4, 7, and 11 descriptions added	Orion Hollar
11/17/2021	Blocks 5, 10, and 7 descriptions added	Salem Almazrouei
11/17/2021	Blocks	Isaac Goshay
11/19/2021	Interface Table Created	Ekaterina Rott
11/19/2021	Interfaces D,E,F,G,H,Q, R. Block Owners	Orion Hollar
11/19/2021	Interface N, Q.	Salem Almazrouei
12/1/2021	Block Diagram Editing, Block editing	Orion Hollar
12/3/2021	Block and Interface rewriting for new block diagram	Orion Hollar
12/3/2021	Block and Interface Definition Revisions	Ekaterina Rott
12/3/2021	Block and Interface Definition Revisions	Salem Almazrouei
3/4/2022	Block and Interface Updates	Orion Hollar
5/6/2022	Block Diagram and Interface Updates	Orion Hollar

4. Block Validations

4.1. Power Block

4.1.1. Block Overview

Block Champion: Ekaterina Rott

The power block is intended to be the connection between solar panels and the control circuitry for a node in a DC solenoid irrigation control network. This block needs to charge a 2 cell lithium ion battery pack and power a 12v load. The block will include solar panels, a battery charger, the battery pack, and a boost converter.

4.1.2. Block Design

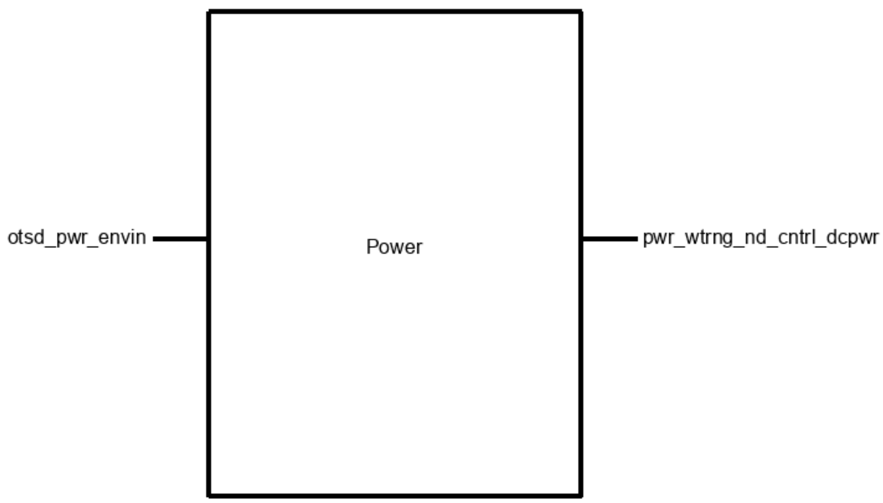


Figure 4.1.1: Top Level Black Box Diagram

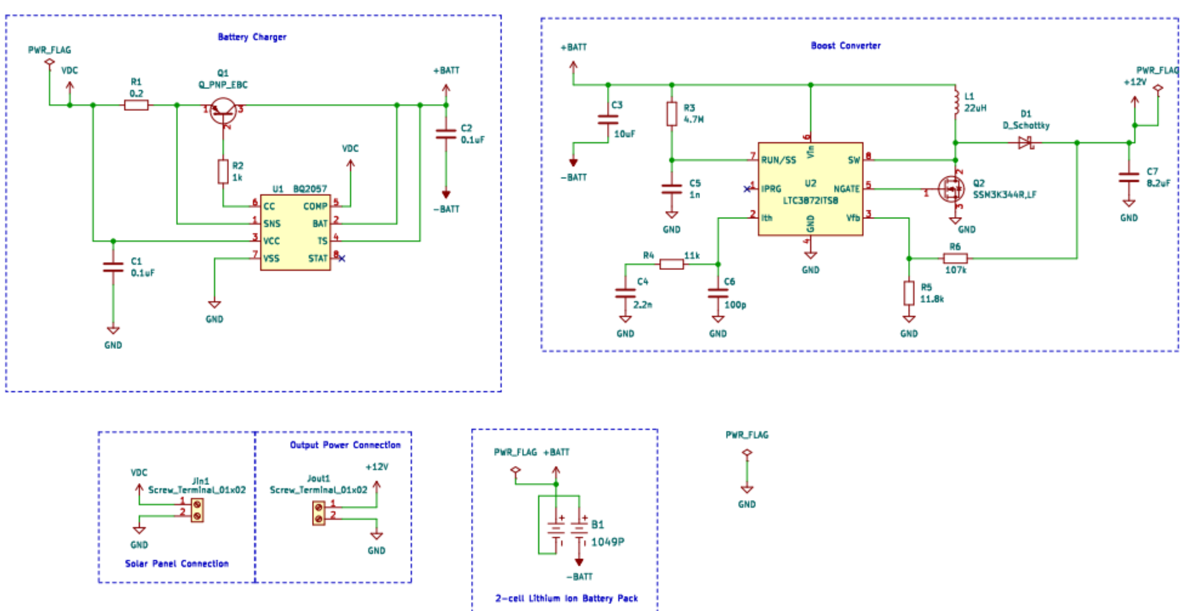


Figure 4.1.2: Circuit Diagram

4.1.3. Block General Validation

The design of this block requires connecting three primary components, those being the load, the battery, and the solar panels. The solar panels should be able to charge the battery cells, and the battery in turn powers the load. Since the system will always be pulling some amount of power, as after turning on the system it will remain on, it is better to have the battery powering the load consistently rather than relying on inconsistent power from the solar panels.

The solar panel chosen for this block is a 5W, 12V solar panel, because it is the most affordable and most available solar panel, and provides the best area for mounting the system. Similar panels that produce less voltage and were cheaper either didn't produce enough power or were physically too small to work with the system.

The batteries chosen were lithium ion. Lithium ion batteries are more expensive than other types, such as lead acid, and are more complex to charge, but they have higher efficiency and longer life spans. To avoid needing to replace the batteries often, lithium ion batteries were chosen over lead acid batteries. Next, when considering the battery size, the larger the battery the more expensive, but the longer the battery would last. A common size of lithium ion batteries is 18650, where the first two numbers represent the diameter in millimeters, and the second 3 represent the length. I chose 18650 due to the ease in finding applicable mechanical parts to hold the batteries and the relatively high battery life.

The battery charging ICs are harder to locate due to cost, stocking available, and viable packaging. Most cheaper battery charging ICs are in a package that would be hard to solder without access to high-end equipment, so the cost of the chip is driven up by needing a package that I could reasonably solder and test. 3 cell battery charging ICs are harder to locate than 2 cell battery chargers, so I chose to go with a 2 cell design so I could buy multiple chips in case I damage one and need to swap it out.

Lastly, the system needs to generate 12V off of the 8.7 V battery pack. Therefore, a boost chip was chosen that would be able to facilitate that conversion. Boost converter technologies are fairly cheap and available, so the cost of the boost chip was not a major factor in the design. Boosting the voltage trades current, so the output voltage is higher but the output current is lower. While the DC solenoid in an adjacent block requires about 1.9A, that block will be utilizing a capacitor to generate those levels of current, so my design is acceptable for this block.

4.1.4. Block Interface Validation

otsd_pwr_envin Interface Properties		
Minimum Lux: 1000	Oregon is a cloudy state, and the lux of an overcast day is around 1000. Our system should still work when the lux is low. [1]	The solar panel chosen is a high enough wattage to provide power even in low light conditions.
Minimum Duration of Direct Contact: 3 hours	3 hours is the minimum number of peak hours of sunlight in Oregon. [2]	The solar panel chosen will be able to provide charge during this period of time.
Environmental: 8oz of Water poured directly above	The solar panel is intended to be large enough to cover the entire system and protect it from rain and other debris from above.	The solar panel chosen has an IP65 rating and is large enough to house the system underneath it.
pwr_wtrng_nd_ctrl_dcpwr Interface Properties		
Vmin: 11V	11V will still power the other system blocks, as there are capacitors supporting the powering of the DC solenoid.	The boost converter chip chosen for this block is calibrated such that the output voltage should be 12V.
Vmax: 12V	The DC solenoid this system is ultimately powering needs 12V to turn on and off.	The boost converter chip chosen for this block is calibrated such that the output voltage should be 12V.
Inominal: 500mA	500mA is the amount of current the other blocks in the system nominally need.	The design chosen provides 500mA according to the data sheet. [3]
Ipeak: 550mA	550mA is as high as the current can get without repeatedly damaging other system blocks.	The efficiency charts on the chosen boost chip go up to 550mA at a 12V output, implying the current will peak at 550mA. [3]
Other: Power can be supplied when system is not exposed to light	System needs to be able to operate in the dark, such as in the late evenings or early morning when sunlight might not hit the solar panels.	The integrated battery pack will power the system at all times, even when dark.

4.1.5. Block Testing

1. Set up the block with a lamp over the solar panel that can generate a minimum of 1000lx and a bucket underneath the panel to catch water. Have a multimeter, variable load, lux meter, a cup with 8oz of water, and a paper towel nearby for testing.
2. Turn on the lamp, and measure the ambient light to ensure that the system is exposed to at least 1000lux.
3. Connect output to a load that draws 500mA. Observe the voltage for 30 seconds.
4. Connect output to a load that draws 550mA. Observe for 3s.
5. If during steps 3 and 4, the voltage remains within [vmin] and [vmax], the block is verified.
6. Turn light off, repeat steps 3-5 to ensure the system works in darkness.
7. Turn the lamp off for safety, move the lamp if necessary. Pour the 8oz of water from directly over the solar panel while holding the panel at an angle to simulate the when the solar panel will be mounted.
8. Using a paper towel, wipe the underside of the solar panel to prove the solar panel supplies protection from water from above.

4.2. Watering Node Control Block

4.2.1. Block Overview

Block Champion: Orion Hollar

This block is part of a larger node that is solar powered. This node's job is to receive wireless commands and use them to control a valve. The Node Control module will be the physical interface between the programmable control node, where commands are received, the power block, which includes the solar panel, as well as containing the DC solenoid which will open and close the valve in a pipe. This will include an on/off switch, a switch to manually control the solenoid and force it to open the valve, the solenoid itself, and a voltage regulator allowing it to power the programmable control node using power from the power block.

4.2.2. Block Design

The Watering Node Control Block has three inputs, and two outputs. The first input is DC power from the power block, which powers this block. The first output is a mechanical movement of a DC Solenoid which can be controlled by the control node or manually. The second input is a pair of switches, one which turns the power for the system on and off while the other is a manual control for the Solenoid that forces it to open, or pull in. The third input comes from a separate block, the programmable control node, in the form of setting two wires high or low, which controls the opening and closing of the solenoid. The second output supplies dc power to the programmable control node.

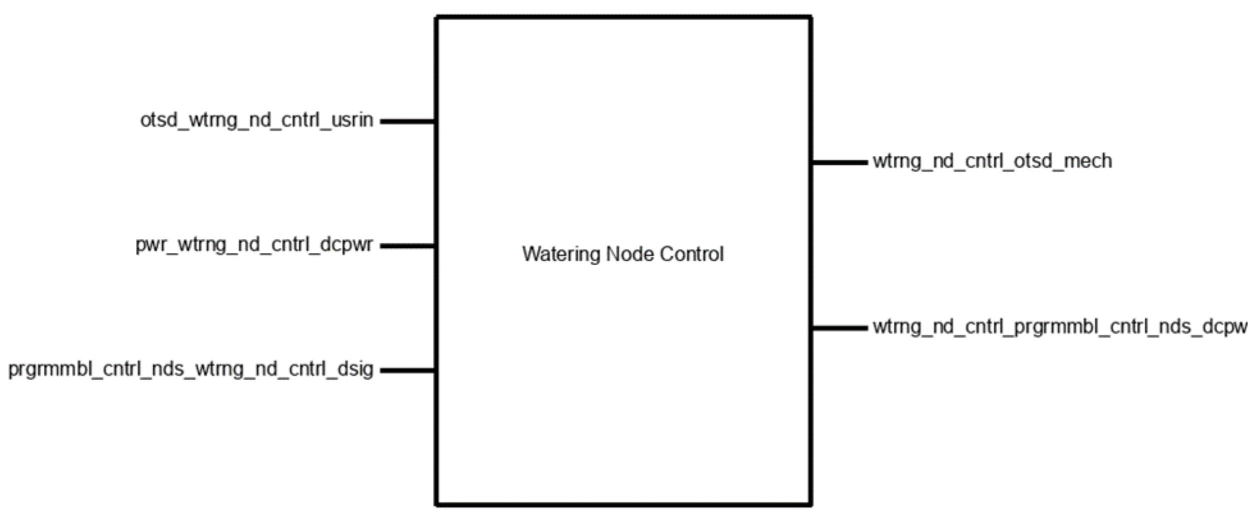


Figure 4.2.1: Watering Node Control Black Box Diagram

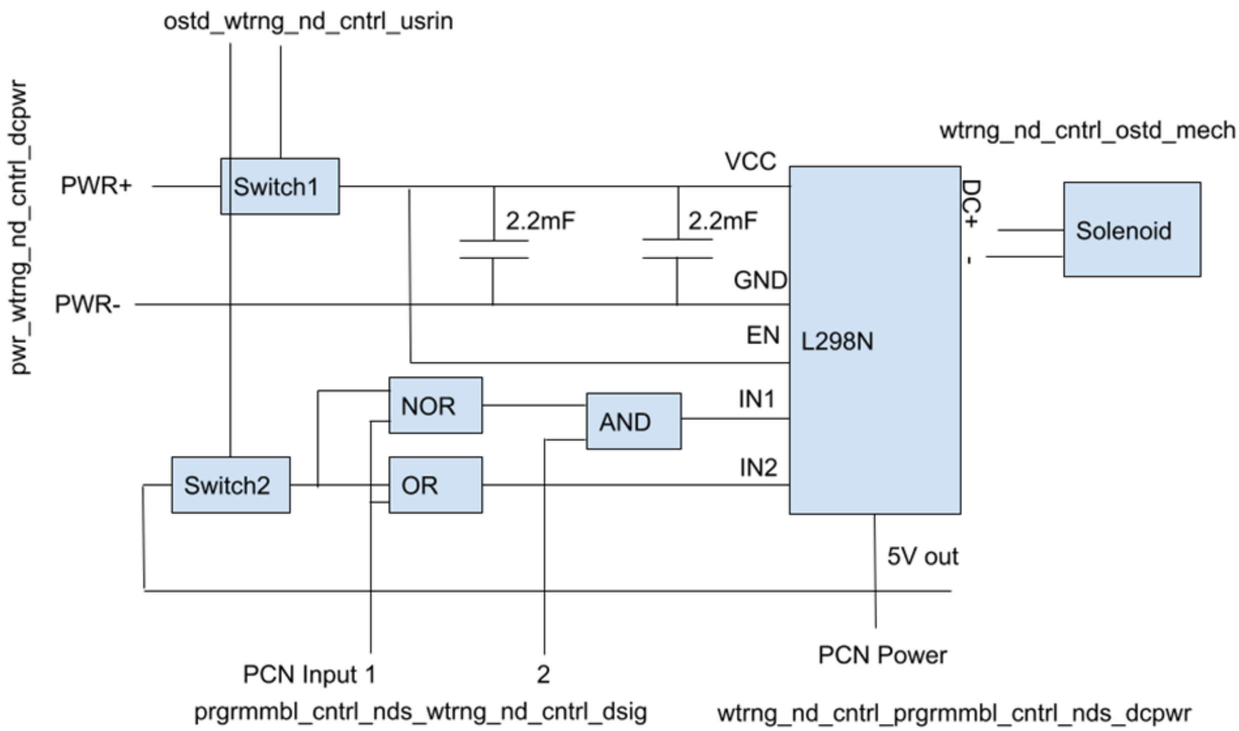


Figure 4.2.2: Watering Control Node Design

4.2.3. Block Design Validation

The primary requirements for the block are an on-off switch, a manual control switch, and a solenoid that can be controlled by the second switch or from the programmable control node. For my design, the first switch directly blocks or allows power to the block, effectively turning on or off all aspects of the block. The power goes directly into the L298N chip, which is a board that functions both as an H bridge and a 5V supply. The board has two inputs which route current through the solenoid depending on which of the inputs are high or low, allowing for current to be run one way to open, or the other way to close the solenoid. The programmable control node is connected to these two inputs, allowing it control over the solenoid. The second switch is also connected to these two inputs, which when switched will force one of the inputs high and the other low, forcing the pulled in state of the solenoid, or the open state of the valve.

A small logic circuit prevents both inputs from being high at once, which could be potentially harmful to the board. The current draw for the solenoid to initially switch states is 1.5 amps higher than what the power supply normally provides, which is solved using two large parallel capacitors that can provide a brief spike large enough to switch the states. The L298N board has already been purchased for less than $10, has already been shipped and has dimensions of less than 2inx2in, giving plenty of room for other parts within the node. The time required to assemble the circuit once all parts have been acquired should not take very long since it is primarily making connections between parts.

4.2.4. Block Interface Validation

otsd_wtrng_nd_cntrl_usrin : Input

Other: Switch is 1-2cm long	This length should be enough that it is easily usable, but not so long that it sticks out from under the system and catches water.	This will depend on the exact part number chosen for the switch, which is very open, with many kinds of switches available on the market.
Timing: Could potentially be used several times per second, user input takes less than a second	Under typical use, the switch should only be used once or twice with a few seconds in between, but a user could potentially grab the switch and flip it back and forth quickly, so the system must be able to accommodate.	Being directly connected to the circuitry the user input will immediately take effect, allowing the user to flip the switch quickly without causing internal issues.
Type: Flipping two switches	For this block the team wanted two interactions to be possible. The first being turning the node off and on, achievable with one switch, and the second being manual control of the solenoid in the block, achievable with another switch.	As shown in the diagram, two switches are included that allow the user to input.

pwr_wtrng_nd_cntrl_dcpwr : Input

Inominal: 500mA	This is the expected nominal current output by the power block.	The primary sensitive piece of circuitry in this design is the L298N board, which is able to handle at least 2A of current[4].
Ipeak: 550mA	This is the expected peak current output by the power block.	The primary sensitive piece of circuitry in this design is the L298N board, which is able to handle at least 2A of current[4].
Vmax: 12V	This is the expected maximum voltage output by the power block.	The primary sensitive piece of circuitry in this design is the L298N board, which can run all needed functions at up to 12 Volts[4].
Vmin: 11.5V	This is the expected minimum voltage output by the power block.	The primary sensitive piece of circuitry in this design is the L298N board, which can run all needed functions at up to 12 Volts[4].

prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig : Input

Other: Signal to turn motor open or closed	The Solenoid is typically meant to be controlled by the node control, so this interface is meant to receive those commands.	A connection from the interface goes into the H bridge block inputs, allowing for almost direct control of current flow through the solenoid.
Other: 2 digit binary signal	The H bridge board requires 2 inputs to allow both directions of current flow.	This design allows for two input pins from the programmable control node.
Other: Logic Level: 5V	According to the L298N board specifications, the logic level of the board is 5V.	As long as the And, Or, and Nor blocks can handle 5 V, there is nothing else that stops 5V from being accepted.

wtrng_nd_cntrl_otsd_mech : Output

Other: Either extended or unextended, no in-between	Our final design is meant to have two states, open and closed.	The design uses a DC solenoid, which is built to have 2 states of either fully extended or unextended.
Other: Solenoid Extends > 0.5cm	The valve mechanism used by the team is controlled by a shallow input of a cylinder which activates below the 0.5cm range.	The DC solenoid that I will use in the design extends to slightly below 0.5 cm.
Other: Extends to 90% capacity within 1 second	While the system generally operates on the scale of hours, the team prefers a reactive system. This amount of time allows for easy testing and confirmation that it functions.	The L298N's connection to the solenoid is primarily based on a H bridge, which is built of transistors. The transistors should not cause any significant lag between current being given and then run through the solenoid.

wtrng_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr : Output

Inominal: 0.5A	The programmable control node is able to run on 0.5A at 5V, which this interface will be powering.	The voltage regulator on the L298N that is being used for power has a listed Inominal current of 0.5A[5]. There are no circuit elements between the board and the output that should increase the current beyond the datasheet values.

Ipeak: 0.6A	The programmable control node is able to run on 0.5A at 5V, which this interface will be powering. This demand should not fluctuate widely.	There are no circuit elements between the board and the output that should increase the current beyond the datasheet values. The voltage regulator on the L298N that is being used for power has a listed Imax current of 0.7A[5].
Vmax: 5.2V	The programmable control node is able to run on 0.5A at 5V, which this interface will be powering. This demand should not fluctuate widely.	The voltage regulator on the L298N that is being used for power has a listed Vmax of 5.2V[5].There are no circuit elements between the board and the output that should increase the current beyond the datasheet values.
Vmin: 4.8V	The programmable control node is able to run on 0.5A at 5V, which this interface will be powering. This demand should not fluctuate widely.	The voltage regulator on the L298N that is being used for power has a listed Vmin of 4.8V[5]. There are no circuit elements between the board and the output that should increase the current beyond the datasheet values.

4.2.5. Block Testing

1. Ensure that both first and second switch are currently in the off position (**otsd_wtrng_nd_cntrl_usrin).**
2. Measure switches with a ruler or similar measurement tool to ensure length between 1-2cm (**otsd_wtrng_nd_cntrl_usrin).**
3. Attach DC voltage source set to 12V to the DC power input (**pwr_wtrng_nd_cntrl_dcpwr**).
4. Turn first switch to On (**otsd_wtrng_nd_cntrl_usrin).**
5. Connect DC power output to 10 ohm resistor (**wtrng_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr**).
6. Confirm that power output is between 0.5A and 0.55A with a multimeter connected from the resistor to ground.
7. If power output is less than 0.5A, use a smaller resistor until the output is within 0.05A over Inominal.
8. If power output is greater than 0.55A, use a larger resistor until the output is within 0.05A over Inominal.
9. Confirm that power output is between Vmin(4.8V) and Vmax(5.2V) with a voltmeter over the resistor.
10. Use smaller resistors until the output is 0.6A or slightly greater.

11. Confirm that power output is between Vmin(4.8V) and Vmax(5.2V) with a voltmeter over the resistor.
12. Remove Resistor, voltmeter and multimeter.
13. Ensure that Solenoid is currently in extended position (**wtrng_nd_cntrl_otsd_mech).**
14. Turn the second switch to On (**otsd_wtrng_nd_cntrl_usrin).**
15. Observe that Solenoid is pulled in from the extended position (**wtrng_nd_cntrl_otsd_mech).**
16. Turn the second switch to Off (**otsd_wtrng_nd_cntrl_usrin).**
17. Attach second DC voltage source set to 5V to 2nd programmable control node input while using finger to block solenoid from shooting out core (**prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig).**
18. Observe that Solenoid extends in reaction to simulated input. The reaction will be fast enough to confirm that it extended faster than 1 second without a measurement tool (**wtrng_nd_cntrl_otsd_mech).**
19. Use a ruler or similar measurement tool to observe that solenoid is extended less than 0.5cm (**wtrng_nd_cntrl_otsd_mech).**
20. Swap second DC voltage source set to 5V to 1st programmable control node input (**prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig).**
21. Observe that Solenoid pulls in from the extended position in reaction to simulated input (**wtrng_nd_cntrl_otsd_mech).**
22. Remove second DC voltage source (**prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig).**
23. Reduce DC voltage source connected to power voltage to 11.5V (**otsd_wtrng_nd_cntrl_usrin).**
24. Attach second DC voltage source set to 5V to 2nd programmable control node input while using finger to block solenoid from shooting out core (**prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig).**
25. Observe that Solenoid extends in reaction to simulated input (**wtrng_nd_cntrl_otsd_mech).**
26. Remove second DC voltage source from 2nd control node input (**prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig**).
27. Rapidly switch the first switch between off and on position, leave in off position **(otsd_wtrng_nd_cntrl_usrin).**
28. Rapidly switch the second switch between off and on position, leave in off position **(otsd_wtrng_nd_cntrl_usrin)**.
29. Remove DC voltage supply (**otsd_wtrng_nd_cntrl_usrin).**

4.3. Programmable Control Nodes

4.3.1. Block Overview

Block Champion: Isaac Goshay

The programmable control Nodes block will be the long range communication by radio between the watering nodes and the hub. This block consists of a feather m0 with a lora attachments, 915 Mhz radio antennas, and the code that allows for sending and receiving information between the node and hub.

4.3.2. Block Design

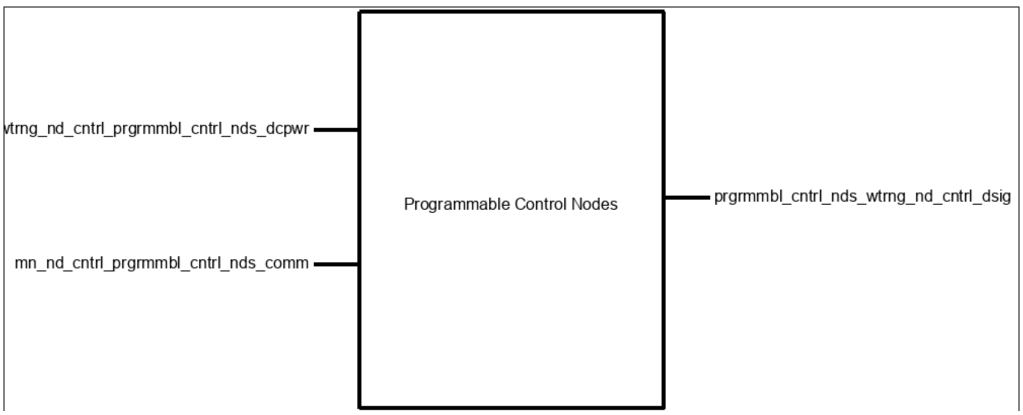


Figure 4.3.1: Programmable Control Nodes Black Box Diagram

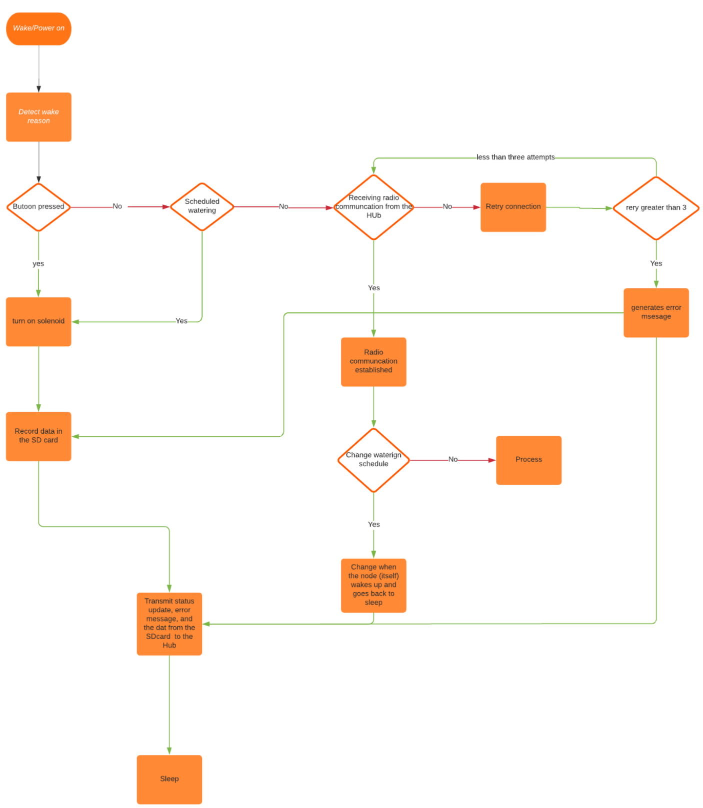


Figure 4.3.2: Programmable Control Node Design

4.3.3. Block General Validation

This design fits the needs of the system since it takes care of the main parts of the system such as the long range radio communication, watering schedule, and the button press. All of these interactions are key points of this project. Some of the related concerts would be the availability of parts since we will be relying on the adafruit featherwing LoRA board. If this board is not available then we would have to change the design drastically. Another big concern would be the time it would take to make sure this works consistently. Since radio signals can be interrupted by many things and without the proper knowledge we will think it is not working. Also knowing how to code the system to how we want it.

4.3.4. Block Interface Validation

prgrmmbl_cntrl_nds_wtrng_nd_cntrl_dsig

Other Signal to turn Motor open or closed	This is going to be a binary value since the valve will either be closed or open.	Our design will be able to meet the property since we are using an Adafruit feather M0 Lora we are able to use the General purpose input output pins. These pins are only able to go high or low. In this case it will mean on and off. [12]
Other Logic level 3.3 V	The logic level for our specific board, the Adafruit feather M0 Lora is 3.3V.	Our design will be able to meet the property since the board picked out the Adafruit feather M0 Lora operates at a 3.3V for a "high" value aka a 1.

wtrng_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr

IPeak 500 mA	The peak current was found based on the spec from the Adafruit M0 Lora board	For the Adafruit feather M0 with LORA attachment.[12] - During +20Bm transmission 120 mA peak This is also a given valued concerning how much max current it will be able to draw
I Nominal 200 mA	This nominal current was chosen based on the expected current needs of the system overall	For the Adafruit feather M0 with LORA attachment.[12] - While in active radio mode it is drawing 40 mA - 300 uA while sleeping
Vmax 5V	The max voltage was found based on the spec from the Adafruit M0 Lora board	For the Adafruit feather M0 with LORA attachment.[12]

		- This is the max voltage that the voltage regulator that the board is able to handle
Vmin 3.3 V	The minimum voltage was found based on the spec from the Adafruit M0 Lora board	For the Adafruit feather M0 with LORA attachment.[12 - The minimum voltage was found by that we are able to use the 3v line as input and be used to power the board. - This also makes sense since the onboard logic is 3.3 V

mn_nd_cntrl_prgrmmbl_cntrl_nds_comm

Messages: Receive Confirmation Messages, Send DC Solenoid Commands	The value for this is a 1 or a 0 since it is a matter of whether the system received the message and is able to send something back. Also if it did then it needs to issue the command to the solenoid	For the Adafruit feather M0 with LORA attachment.[12] - There is example code that has a confirmation sequence to make sure that the system is receiving a message . - There is example code set up already it should work with the box setup
Vmin: 3.3 V	The minimum voltage was found based on the spec from the Adafruit M0 lora board	For the adafruit feather M0 with Lora attachment [12] - This is the minimum voltage since the board minimum voltage is 3.3 V. Therefore the lora board will be running with the minimum 3.3 V logic.

4.3.5. Block Testing

1. Plug in a power source to two Adafruit feather M0 Lora board
2. Use the test code give to us by Adafruit to transmit messages to ensure that the board has the ability to send messages between each other
3. Create a Json File with information like watering schedule.
4. Use the adapted control node code and send Json files to one another
5. The board reads the json file and looks for information such as watering schedule
6. The Adafruit feather M0 Lora board will have an LED connected to a pin that it will light to simulate the solenoid turning on

4.4. Main Node Control Block

4.4.1. Block Overview

Block Champion: Orion Hollar

This block provides power to the short-range communication and the programmable control node which both need 5V, using power from a wall socket connection. It will use a single converter to create an output at an appropriate level and then split the output to connect to both other blocks.

4.4.2. Block Design

This block has a fairly straightforward design consisting of two major parts. An outlet connector, as the name implies, connects to a wall outlet and gives two connectors to access the 120V AC power from the wall socket. An AD-DC converter is used to bring this 120V AC down to a usable 5V to supply to other blocks. This 5V output is split into two connections with a positive and ground each.

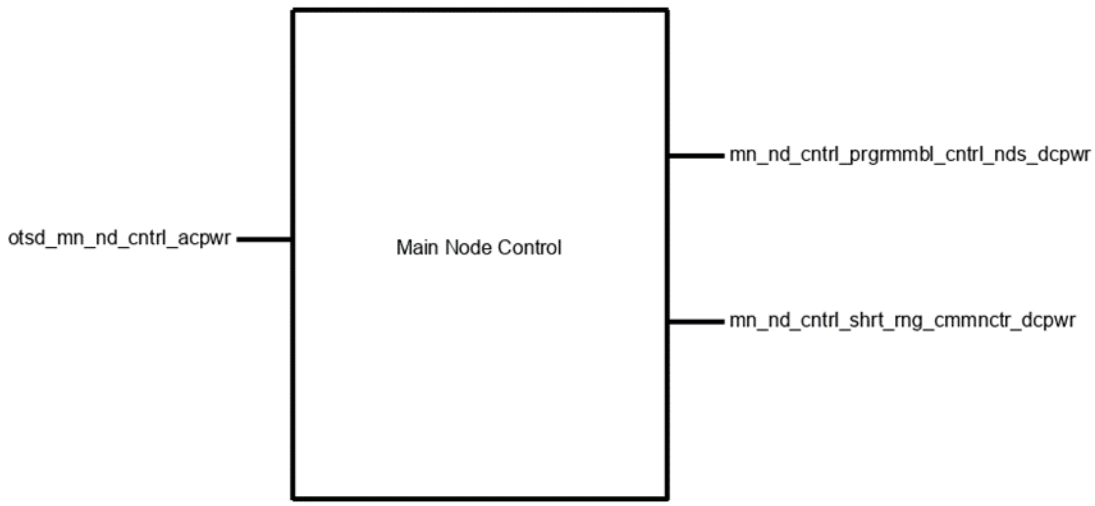


Figure 4.4.1: Main Node Control Block Black Box Diagram

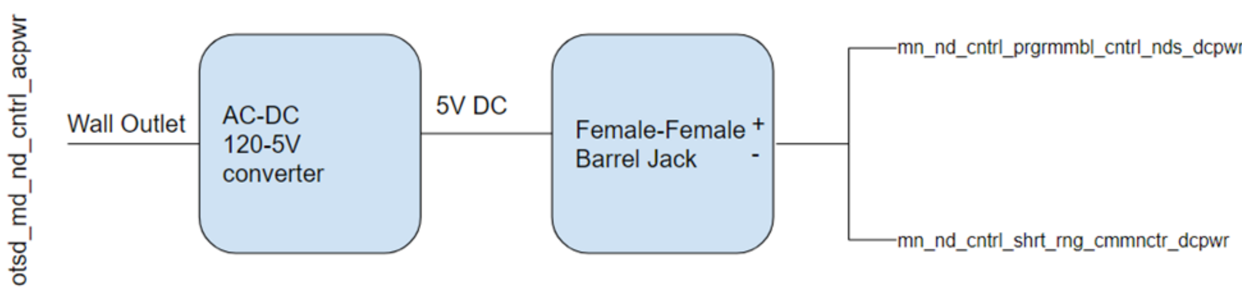


Figure 4.4.2: Main Node Control Block Design

4.4.3. Block General Validation

The most important function of this block is connecting to the wall socket and converting the 120V down to the more usable 5V. In these aspects, the most efficient solution is a plugged in converter and a connector to integrate that power into the system. Alternatively, two converters could have been used to give each board their own source, but it was decided this would be unnecessary as long as the one converter was able to handle the demand from both blocks.

4.4.4. Block Interface Validation

otsd_mn_nd_cntrl_acpwr : Input

Ipeak: 0.12A	To supply 5V and 2A takes 10 Watts. Being safe with a low estimate of 70% efficiency at 120V gives a current of 0.12A.	Modern wall socket converters are rated for 75% efficiency in the worst case, so the chosen converter will be above the calculated value.
Other: Wall Socket Connection	This is an identifier to clarify what the interface is. It is important that this interface is able to connect to a wall socket.	The wall socket connector will either include or be able to attack a power cord which will be able to plug into the wall socket.
Vnominal: 120V	In the United States wall socket connections are rated at 120V.	The power converter is rated for 120V, up to 230V. [6]

mn_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr : Output

Inominal: 50mA	This is a rough estimation for the current draw of the programmable board under normal conditions.	The power converter is rated to supply up to 2A. 2A is more than enough to handle both board's demands. [6]
Ipeak: 0.35A	This is a rough estimation for the maximum current draw of the programmable board.	The power converter is rated to supply up to 2A. 2A is more than enough to handle both board's demands. [6]
Vmax: 5.2V	The voltage needed by the board is 5V, this maximum is to give it an operable range.	The power converter is rated to supply 5V with a voltage ripple of 60mV. [6]
Vmin: 4.8V	The voltage needed by the board is 5V, this minimum is to give it an operable range.	The power converter is rated to supply 5V with a voltage ripple of 60mV. [6]

mn_nd_cntrl_shrt_rng_cmmnctr_dcpwr : Output

Inominal: 50mA	This is a rough estimation for the current draw of the short range under normal conditions.	The power converter is rated to supply up to 2A. 2A is more than enough to handle both board's demands. [6]
Ipeak: 0.35A	This is a rough estimation for the maximum current draw of the short range board.	The power converter is rated to supply up to 2A. 2A is more than enough to handle both board's demands. [6]
Vmax: 5.2V	The voltage needed by the board is 5V, this maximum is to give it an operable range.	The power converter is rated to supply 5V with a voltage ripple of 60mV. [6]

Vmin: 4.8V	The voltage needed by the board is 5V, this minimum is to give it an operable range.	The power converter is rated to supply 5V with a voltage ripple of 60mV. [6]

4.4.5. Block Testing

1. Connect the plug into an available wall socket (otsd_mn_nd_cntrl_acpwr).
2. Connect a multimeter over the + and – connectors for the mn_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr output.
3. Connect a current load over the + and – connectors for the mn_nd_cntrl_prgrmmbl_cntrl_nds_dcpwr output.
4. Set the current load to draw 50mA.
5. Observe voltage on current load to be within 4.8-5.2V.
6. Set the current load to draw 0.35A.
7. Observe voltage on current load to be within 4.8-5.2V.
8. Move the multimeter over the + and – connectors for the mn_nd_cntrl_shrt_rng_cmmnctr_dcpwr output.
9. Connect a second current load over the + and – connectors for the mn_nd_cntrl_shrt_rng_cmmnctr_dcpwr output.
10. Set the current load to draw 0.05A.
11. Observe voltage on current load to be within 4.8-5.2V.
12. Set the current load to draw 0.35A.
13. Observe voltage on current load to be within 4.8-5.2V.

4.5. Website Block

4.5.1. Block Overview

Block Champion: Ekaterina Rott

The website is the primary user interface for the system, and will control the watering schedules and will display any errors present in the system. The website MySQL server will also serve as a backup for data like the physical system status and past watering schedules. There will be a graphical representation of the system to visually show the user current node statuses. The website will be coded in MySQL and PHP, hosted on the OSU ENGR Web Servers.

4.5.2. Block Design

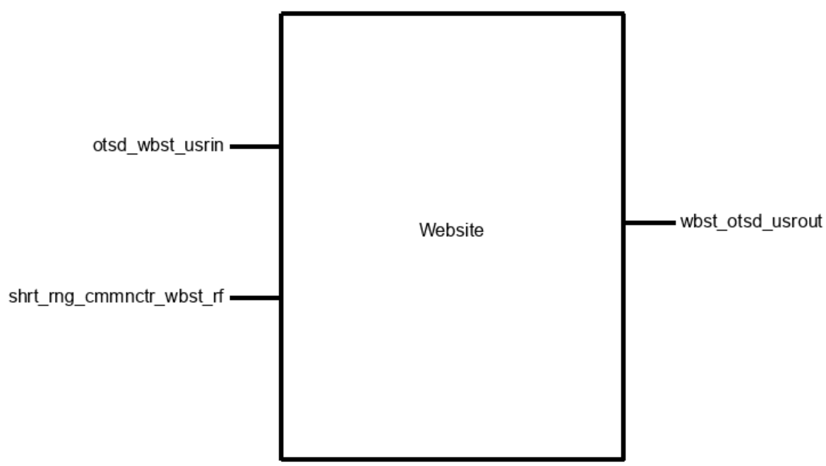


Figure 4.6.1: Black Box Diagram

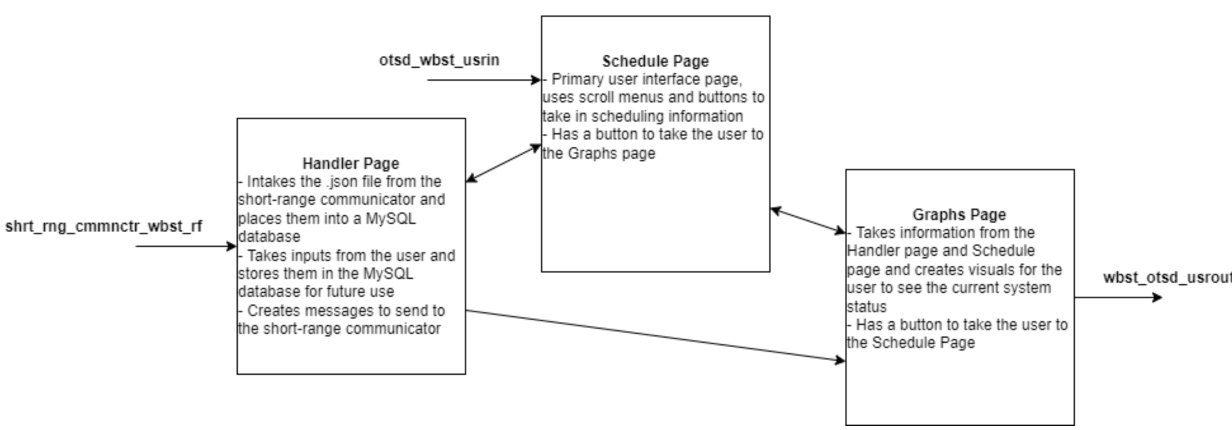


Figure 4.6.2: Block Object Hierarchy and Uses

4.5.3. Block General Validation

The website block is meant to be an easy-to-access interface with the physical system, so that the user does not need to go check on each watering individually to ensure the system is working. This site is where the user will input new watering schedules and see if there are any system status updates, including errors. These functions are necessary in order for the user to communicate with the system, as there are limited functionalities on the nodes themselves.

The block needs to be able to connect the user to the system via an intuitive user interface. Oregon State provides one MySQL database for every engineering student, so to mitigate the costs associated with database maintenance I will be using this MySQL database to store the information from the website and from the nodes. Next, I will be using PHP because it is a fairly simple coding language that is extensively documented and utilized for website and visual creation. Finally, use of the .json files is due to the file processing available on the node microcontrollers.

4.5.4. Block Interface Validation

otsd_wbst_usrin : Input

Other: User will input the watering schedule.	The user needs to provide the watering schedule so that the website can communicate the website to the physical system.	One of the website pages will include a location for the user to input a new watering schedule.
Type: Buttons, Scroll Menus	In order to reduce the need for error handling, buttons and scroll menus will be used to limit possible user inputs.	The PHP code that will create the areas that users can interact with will use buttons and dropdown/scroll menus.
Type: Typing and Clicking	The user input itself will be either typed out times or button clicks.	The PHP code will look for button clicks and typing in certain locations to gather user information.

wbst_otsd_usrout : Output

Other: Website will display any error messages found in the json input files.	The user needs to be able to see any errors at the website in order to reduce the need to go outside and check on the physical system.	One of the website pages in the design will have an error messaging system that will display if there is an error message from the input files.
Type: Website will display a visual that represents which watering nodes are open or closed.	The user needs to be able to see which watering nodes are currently on/open.	One of the website pages will display a visual that will show the current open/closed status of the nodes.
Other: Website will display current watering schedule.	The user needs to see which watering schedule is currently operating, even if the nodes are	One of the website pages will display what the current/active watering schedule is.

	not actively watering.	

shrt_rng_cmmnctr_wbst_rf : Input

Data rate: Messages will be sent a few times a minute.	The short range communication block and website will only need to communicate a few times a minute in order to communicate the watering schedule and current system status.	The PHP code will recall the short range communication block a few times a minute and will send information at the same rate.
Messages: The data received will be in a .json file format.	JSON files are easy to send and receive through PHP and the short range communication block.	The PHP code will use JSON files in both the sending of data and receiving of data when filling in the MySQL database.
Protocol: HTTP POST/HTTP GET.	The website and the short range communication device need to be in contact with each other, one way to do this is through an HTTP POST or HTTP GET protocol.	The PHP code in the website will use the HTTP POST or HTTP GET protocol to communicate with the short range communication block.

4.5.5. Block Testing

1. Boot up website and site servers.
2. First, look at the Schedule page. Look at the user input area, and inspect that the scroll menu has times from midnight to midnight in 5 minute intervals.
3. Put in an example watering schedule. Check that any buttons on the web page work.
4. Submit the watering schedule, check that it appears in the MySQL database. This will confirm the otsd_wbst_usrin interface.
5. Secondly, start by inspecting the code itself. Look at the PHP and make sure that the protocol being used to get data is the HTTP POST or HTTP GET protocol.
6. Next, inspect the Handler page. Make sure that the page is able to receive JSON files and parse them into the SQL database, and that it is able to generate JSON files to send back.
7. Finally, send 5 JSON files via HTTP POST/GET to the Handler page, and make sure they are all in the SQL database. This will confirm the shrt_rng_cmmnctr_wbst_rf interface.
8. Next, looking at the Graphs page. With the JSON files and watering schedules from previous steps, there should be some data shown on the webpage. Visually inspect for the system status, current watering schedule, and system errors. Send new JSON files to the webpage and submit a new watering schedule. If these visuals have changed, wbst_otsd_usrout is verified.

4.6. Short Range Communication

4.6.1. Block Overview

Block Champion: Salem Almazourei

This block is called the short-range communication block, this block will be responsible for communicating between the server (internet) and the programmable block. This block should act as a data buffer which means it will receive the incoming data from the server (using the internet which is connected to this block using WIFI) and the programmable nodes.

The connection between this block and the programmable block will be done using the radio frequency transceivers, the incoming data from the WIFI network is stored in this block and it will be sent to the other block using locally RF link.

4.6.2. Block Design

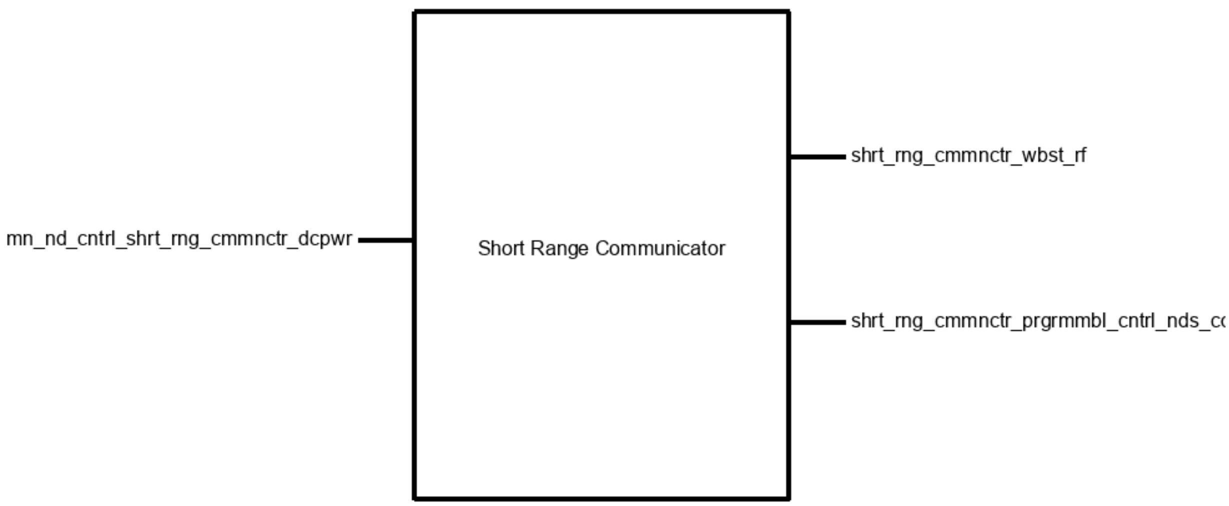


Figure 4.6.1: Short Range Communication Black Box Diagram

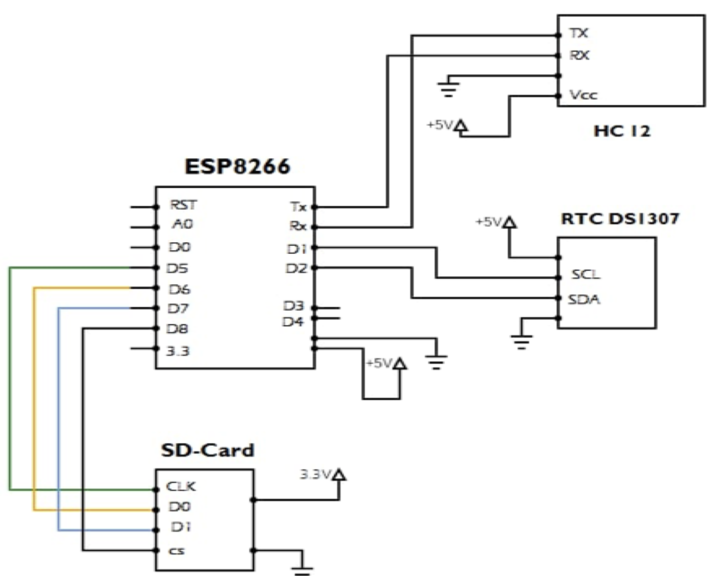


Figure 4.6.2: Short Range Communication Design

4.6.3. Block General Validation

In this section of the report, we will focus on the reasons behind choosing those components for the design, all the general features were introduced in the previous sections but in this section, we will highlight the features that make those parts better than others.

4.6.4. Block Interface Validation

WIFI connection	Everyone is familiar with WIFI networks, even people with low education levels can deal with WIFI and can connects to those networks, also it is available everywhere and the routers as well as the repeaters can be found easily in the market	Using the ESP8266 board that has a built-in WIFI module can reduce the hardware components as well as the power consumption, and also this chip is widely supported and has a lot of libraries of all WIFI connection types.
Protocol: wifi		

Serial interface to RF transceiver	Most of the transceivers communicates using the serial ports via Tx pin and Rx pin, this protocol is supported by the ESP in two types, the hardware serial and the software serial	The hardware serial has an internal buffer, this means that if the CPU is executing any command, it still can receive and store data coming from the serial port, this will eliminate the chance of losing data in transmission.
Protocol: UART		

SPI connection	Serial peripheral interface is a protocol used by the microcontrollers to other devices such as SD cards, this protocol is high speed.	It is very important in this block because it is the only protocol that is supported by the SD card.
Protocol: SPI		

I2c connection	I2C protocol is used to interface the RTC chip, this protocol is multi drop channel which means more than one device can be connected to the same channel and the ESP which will be the master in this case will use devices addresses to send and receive any data to the connected devices	I2c is supported by the ESP and it is the only protocol that can be used with RTC (DS1307).
Protocol: I2C		

4.6.5. Block Testing

1. Connects the USB cable to the ESP8266 board
2. Board should be started and the blue led on the board should start blinking
3. Once the esp board connects to the WIFI network (SSID: project, Pass:12345678) the blinking will stop
4. The ESP board should store the starting time on the SD card, this includes that the read of the RTC and the write on the SD card was successfully done
5. The SD card can be detached from the board and connected to the desktop PC to check the log file, it should be in txt format and it should include the time and date of the start time on the first line of the text line
6. When the ESP board receives any command from the server using the WIFI network, this command should be stored in a separate line on the SD card with its time stamp, the verification can be done by removing the SD card from the board and connect it to the PC to observer the command
7. After the esp board receives the command from the server, it should forward the command to the RF transmitter (via UART port) and wait for the acknowledgement, the result will be stored on the SD card with its time stamp.

4.7. User Guide

4.7.1. Block Overview

Block Champion: Salem Almazrouei

The Wireless Solar-Powered DC Valve Controllers system is a solar-powered system that is used to control the operation of a DC solenoid valve to automate and monitor the irrigation process on farms. In this document, we will provide you with an easy-to-understand guide for first-time users that contains everything about the Wireless Solar-Powered DC Valve Controllers system and how it will run, and also in this guide you will find all the required data for initializing the system as well as troubleshooting any future issues.

4.7.2. Block Design

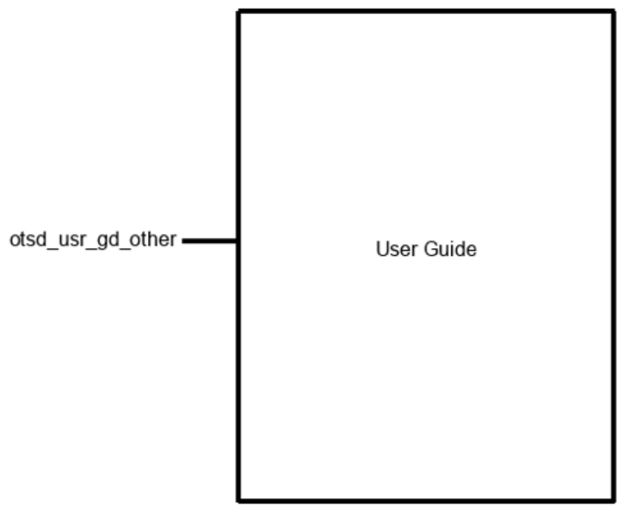


Figure 4.7.1: User Guide Black Box Diagram

This system comes with all required ready components to rapidly start the installation in your farm those components

4.7.3. Block General Validation

User can interface the system in two methods

A- Local operation

- This mode is used when there is no internet connection to the main node where the user cannot send or receive data from the website.
- User will not lose control because he can control the valves manually from the node unit using the manual button.
- In case of power loss on the farm this mode is still available since the valve nodes operate on batteries and can operate for 3 days without being recharged. For more information on how to connect a solar panel to a battery, please see [5].

B- Remote Mode

- In this mode the user can control irrigation from the website.
- In this mode users can send instant commands to open or close any valve.
- User can schedule the watering procedure using this mode.

4.7.4. Block Interface Validation

In this block we have a printed and an online copy that shows all required sections that's meant to provide the end user a guide that contains a high level summary of all system components and basic troubleshooting, along with any relevant set up and recycling information.

4.7.5. Block Testing

After finishing the work with the hardware and software the verification of the user guide can be achieved by following those steps

- Giving the user guide to a new customer and giving him the chance to install the hardware.
- User should install all the components with less than one hour.
- User should use the website to control the valves and observe the results.
- User should be able to schedule the irrigation and observe the results.
- User should know how to start any node manually from the node unit using the buttons.
- User should be satisfied by the results and should be able to compare the history from the website and from the SD card.

4.8. Enclosure

4.8.1. Block Overview

The enclosure will be used to house and protect the electronics from outside interference, conforming with IP 44 standards. Low velocity water and solids larger than 1.0mm shall not be able to enter the enclosure, and any connections entering and leaving the enclosure should be able to be disconnected.

4.8.2. Block Design

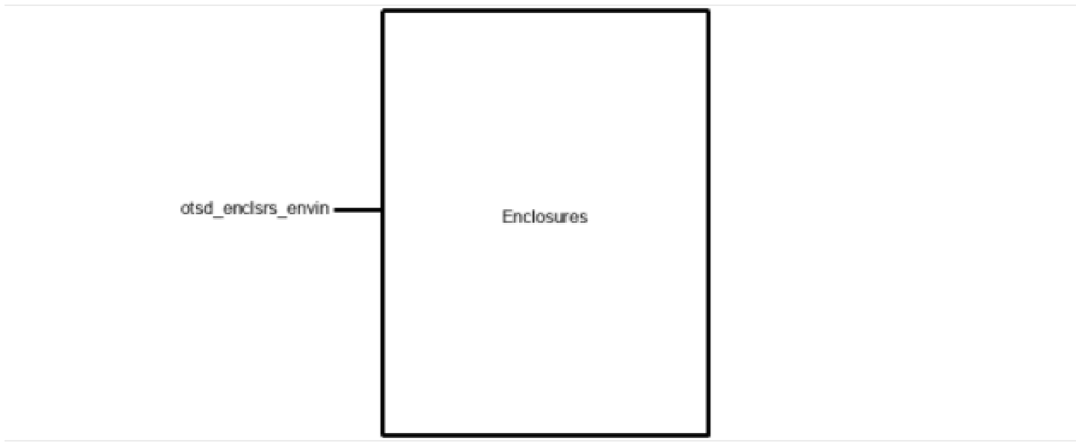


Figure 4.8.1: Enclosure Block Diagram

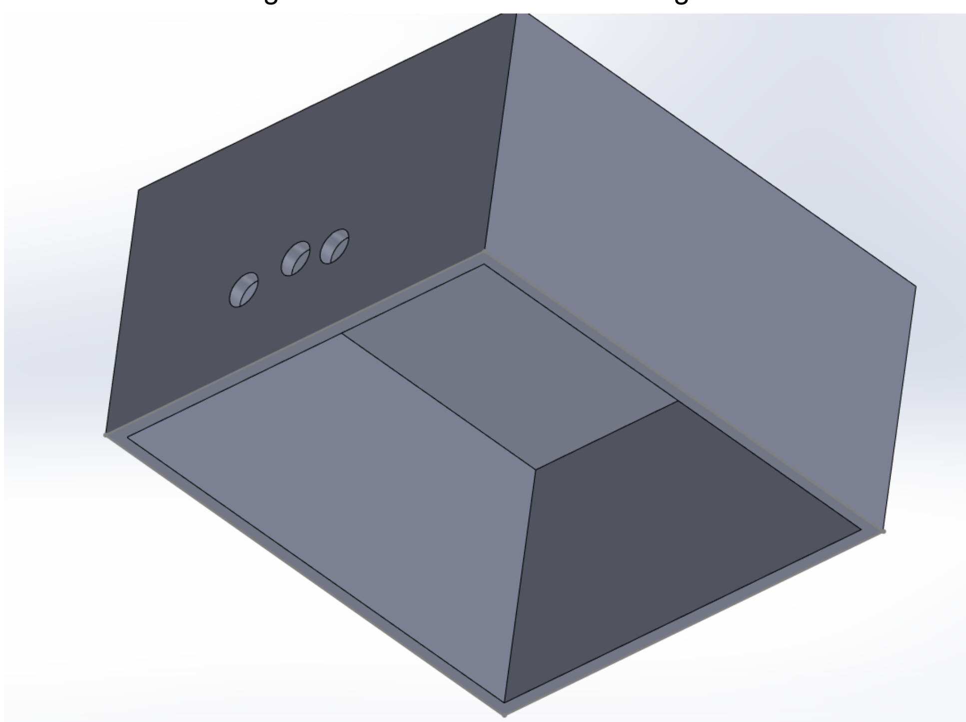


Figure 4.8.2: Enclosure CAD Model

4.8.3. General Validation

This design fits the needs of the system since it is able to house all of the components in the system, and is small enough to fit behind the solar panel. One of the big reasons it has to fit behind the solar panel would be to reduce the amount of space that the system will take up. Since it would not make sense to have two separate systems especially if space is an issue. The materials we will use will most likely be a PLA plastic if we are 3d printing. There will be a couple of wires that will be coming in and out of the enclosure. This will include the power for the solar panel, and the radio antenna. There will be waterproof glands used to make sure that water does not enter the enclosure through the hole for the wires. The mount for the enclosure will most likely be a metal strip with holes that will fasten the enclosure onto the solar panel.

This design fits the needs of the system since it takes care of the main parts of the system such as the long range radio communication, watering schedule, and the button press. All of these interactions are key points of this project. Some of the related concerts would be the availability of parts since we will be relying on the adafruit featherwing LoRA board. If this board is not available then we would have to change the design drastically. Another big concern would be the time it would take to make sure this works consistently. Since radio signals can be interrupted by many things and without the proper knowledge we will think it is not working. Also knowing how to code the system to how we want it.

4.8.4. Interface Validation

otsd_enclsrs_envin

Other Dimensions of the box is 7.55in length 6.9in width 4 in depth	This is the size of the solar panel so the enclosure must be small enough to fit behind the solar panel	Currently with the solar panel we are using it is measured to be 8.27 x 7.68 inches. Since the dimensions of the enclosure will be smaller than these values that means they should be able to fit behind the solar panel.
Other Hole size 0.5 in	We will be using this hole for the cable glands. The part of the cable glands that is thickest is roughly 0.49 in	This will work with our design since we will be having a couple of wires that will be going in and out of the enclosure we need to make sure no water gets in. Also by doing it this way we can just get a longer cable instead of designing a way to make sure the connector part of the cable is able to meet the ends of the enclosure.

Low velocity water proof and dust proof	The IP 44 standard is to say that low velocity water and solids larger than 1.0mm will not be able to get inside of the enclosure.	Our design will be able to meet the blocks needs since the enclosure will be designed so that the top part is slanted so that water will slide off. Also we will add a rubber substance that will make it harder for liquids and solids to enter the enclosure[11]

4.8.5. Verification Plan

1. Will bring ruler and measure the length and width of the enclosure
2. Place enclosure onto the backside of the solar panel to see if it fits
3. Measure the holes of the enclosure to makes sure that the cable glands will fit
4. Use a hose to spray water on to the enclosure to simulate low velocity water
5. Throw soil on the enclosure
6. Check if the enclosure has any water or soil inside.

4.9. References and File Links

[1] "Daylight," *Wikipedia*, Jun. 02, 2020. https://en.wikipedia.org/wiki/Daylight.

[2] "Oregon," *TurbineGenerator*. https://www.turbinegenerator.org/solar/oregon/ (accessed Jan. 22, 2022).

[3] "LMR62014XMF/NOPB Texas Instruments | Integrated Circuits (ICs) | DigiKey," *www.digikey.com*. https://www.digikey.com/en/products/detail/texas-instruments/LMR62014XMF-NOPB/3179256 (accessed Jan. 22, 2022).

[4] Amazon.com. 2022. Qunqi L298N Motor Drive Controller Board Module Dual H Bridge DC Stepper For Arduino. [online] Available at: https://www.amazon.com/Qunqi-Controller-Module-Stepper-Arduino/dp/B014KMHSW6/ref=as_li_ss_tl?ie=UTF8&qid=1502219420&sr=8-1&keywords=L298N&linkCode=sl1&tag=howto045-20&linkId=e79eafb4726064dde2a572cb4a8e1c00 [Accessed 22 January 2022].

[5] MC78M00, MC78M00A, NCV78M00 series. West Florida Components. (2008). Retrieved March 13, 2022, from https://www.westfloridacomponents.com/mm5/graphics/H01/MC78M05BDTRK.pdf

[6] https://www.amazon.com/dp/B08TG6YD28?psc=1&ref=ppx_yo2_dt_b_product_details

[7] "ESP32 HTTP GET and HTTP POST with Arduino IDE | Random Nerd Tutorials," Apr. 08, 2020. https://randomnerdtutorials.com/esp32-http-get-post-arduino/ (accessed Feb. 19, 2022).

[8] "How to use JSON in PHP? - Programming Dive." https://programmingdive.com/how-to-use-json-in-php/#:~:text=PHP%20read%20JSON%20file%20JSON%20file%20always%20comes (accessed Feb. 19, 2022).

[9] "PHP Connect to MySQL," *W3schools.com*, 2012. https://www.w3schools.com/php/php_mysql_connect.asp.

[10] "How do I send a POST request using PHP?," *ReqBin*. https://reqbin.com/code/php/ky6hlmcs/php-post-request-example#:~:text=The%20HTTP%20POST%20request%20method%20is%20used%20to (accessed Feb. 19, 2022).

[11] "The Complete Guide to IP rating - IP44, IP54, IP55, IP65, IP66, IPX4, IPX5, IPX7," *IP Rating Guide | IP44, IP54, IP55, IP65, IP66, IPX4, IPX5, IPX7*. [Online]. Available: https://www.coolstuffshub.com/ip-rating-guide/. [Accessed: 05-Feb-2022].

[12] L. Ada, "Adafruit Feather M0 Radio with Lora Radio Module," *Adafruit Learning System*. [Online]. Available: https://learn.adafruit.com/adafruit-feather-m0-radio-with-lora-radio-module/power-management. [Accessed: 22-Jan-2022].

4.10. Revision Table

Table 4.10.1: Section 4 Revision Table

Date	Changes Made	Author
10/31/2021	Section Outline Created	Ekaterina Rott
01/06/2022	Added outlines for each block	Ekaterina Rott
03/06/2022	Added 4.2 and 4.6	Salem Almazrouei
3/07/2022	Added 4.2 and 4.4	Orion Hollar
03/10/2022	Added 4.6 and 4.7	Salem Almazrouei
3/10/2022	Added 4.3 and 4.8	Isaac Goshay
3/13/2022	Edited sections to include titles and sources	Orion Hollar

5. System Verification Evidence

5.1. Universal Constraints

5.1.1. **The system may not include a breadboard**

All system connections are made directly with wires, connectors, or PCB traces.

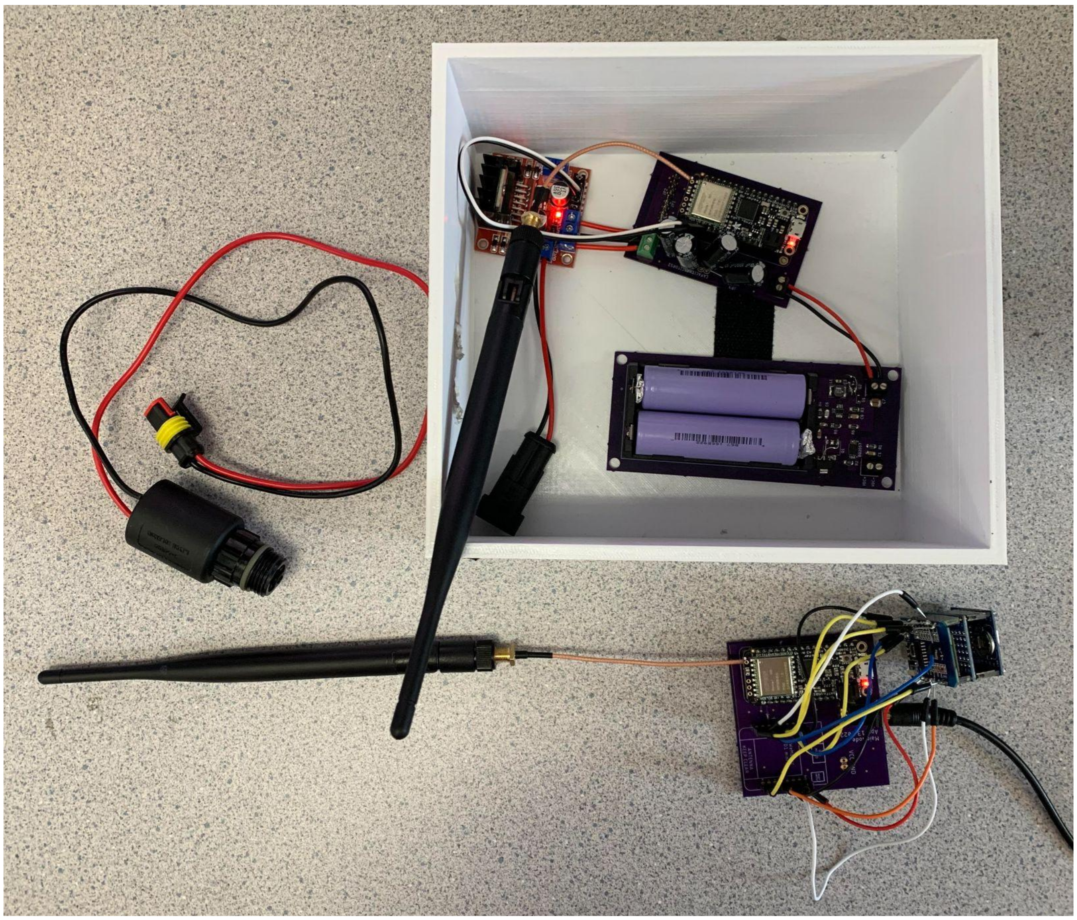


Figure 5.1.1: System with No Breadboards

5.1.2. **The final system must contain both of the following: a student designed PCB and a custom Android/PC/Cloud application.**

Student Designed PCBs - There are three student designed PCBs in the system, a solar charging boost converter, a watering node connector and a control node connector.

a, Solar Charging Boost Converter

Figure 5.1.2: Solar Charging PCB

b, Watering Node Connector

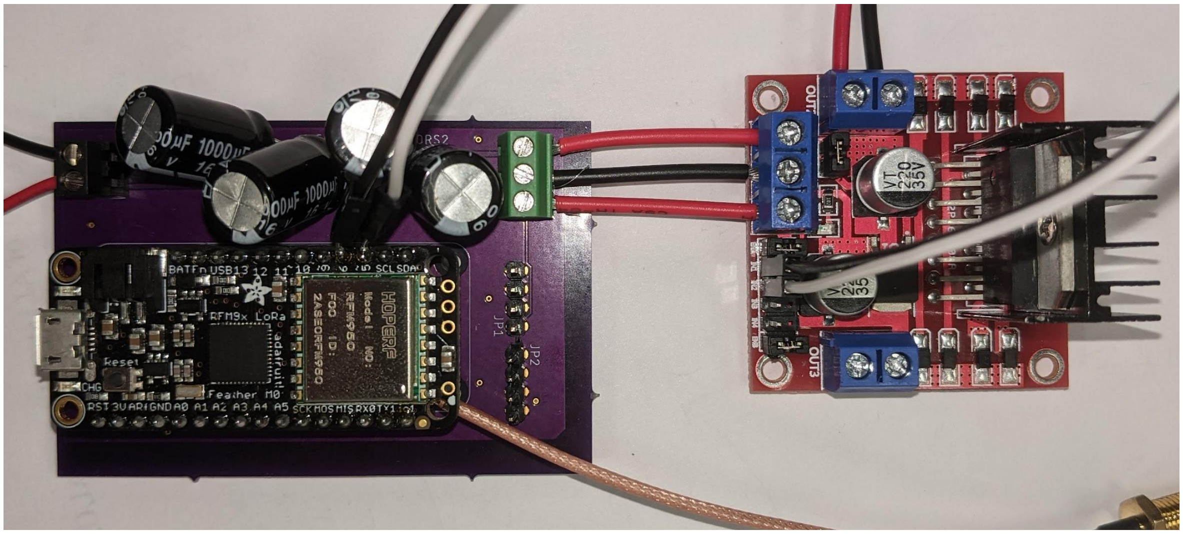


Figure 5.1.3: Watering Node PCB

c, Control Node Connector

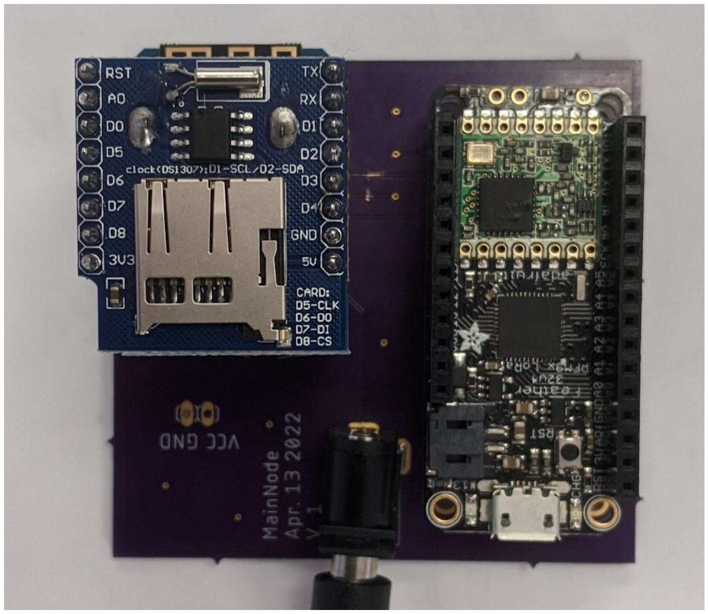
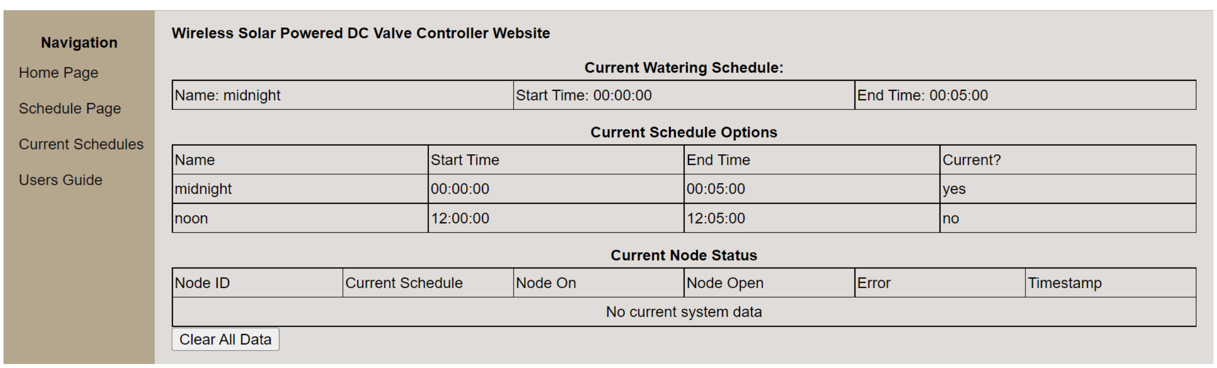


Figure 5.1.4: Control Node PCB

PC Application - The system includes a custom website hosted on the OSU web servers, utilizing a MySQL database to hold system data, and designed with PHP. The PC application can be found at the following address (will not be sustained past 2022): https://web.engr.oregonstate.edu/~rottek/home.php. The website allows users to set custom watering schedules and see current system statuses, along with clearing all system data and schedules.



Navigation

Home Page

Schedule Page

Current Schedules

Users Guide

Wireless Solar Powered DC Valve Controller Website

Current Watering Schedule:

Name: midnight	Start Time: 00:00:00	End Time: 00:05:00

Current Schedule Options

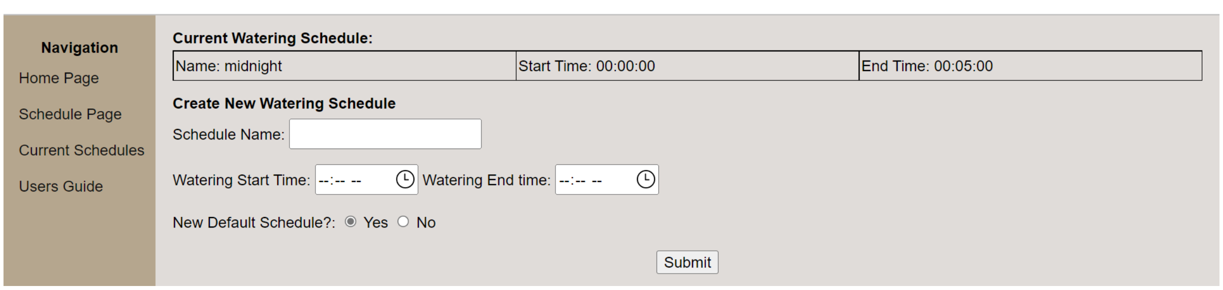
Name	Start Time	End Time	Current?
midnight	00:00:00	00:05:00	yes
noon	12:00:00	12:05:00	no

Current Node Status

Node ID	Current Schedule	Node On	Node Open	Error	Timestamp
No current system data					

Clear All Data

Figure 5.1.5: Home Page of the Website



Navigation

Home Page

Schedule Page

Current Schedules

Users Guide

Current Watering Schedule:

Name: midnight	Start Time: 00:00:00	End Time: 00:05:00

Create New Watering Schedule

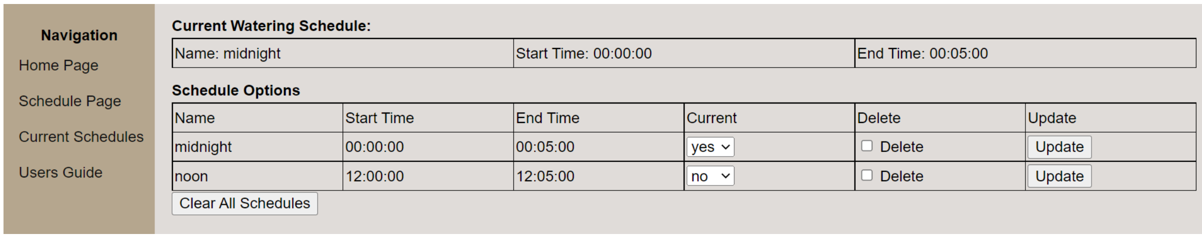
Schedule Name:

Watering Start Time: --:-- -- Watering End time: --:-- --

New Default Schedule?: Yes No

Submit

Figure 5.1.6: Schedule Creation Page of the Website



Navigation

Home Page

Schedule Page

Current Schedules

Users Guide

Current Watering Schedule:

Name: midnight	Start Time: 00:00:00	End Time: 00:05:00

Schedule Options

Name	Start Time	End Time	Current	Delete	Update
midnight	00:00:00	00:05:00	yes	Delete	Update
noon	12:00:00	12:05:00	no	Delete	Update

Clear All Schedules

Figure 5.1.7: Current Schedule Manipulation Page on the Website

5.1.3. **If an enclosure is present, the contents must be ruggedly enclosed/mounted as evaluated by the course instructor.**

The enclosure that the electronics were to be mounted in was not completed due to setting aside the enclosure requirement in order to pursue other engineering requirements. The mounting method was intended to be velcro attached to the bottom of the PCBs and connected to the base and walls of the enclosure, which includes 4 inches of vertical space for the electronics to take up.

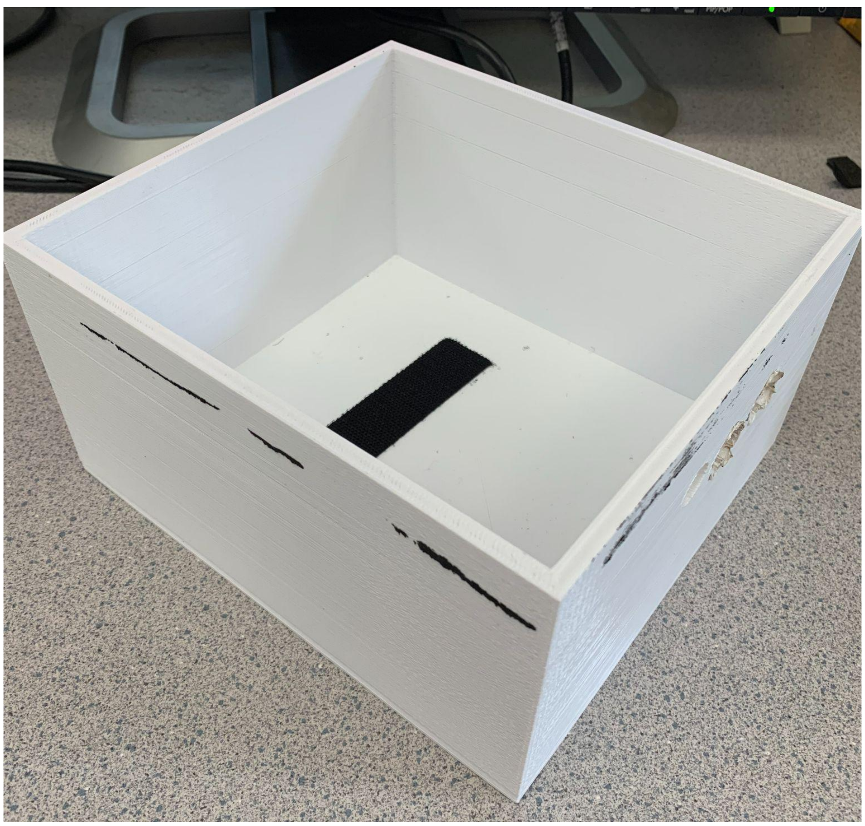


Figure 5.1.8: Enclosure Side View

5.1.4. **If present, all wire connections to PCBs and going through an enclosure (entering or leaving) must use connectors.**

Figure 5.1.9: Holes exiting the Enclosure for Connectors

In Figure 5.1.9, the three holes in the enclosure for the connectors to be seated. The left-most hole is for the radio board antenna, the other two are for the two MUYO connectors to the solenoid and the back up power.

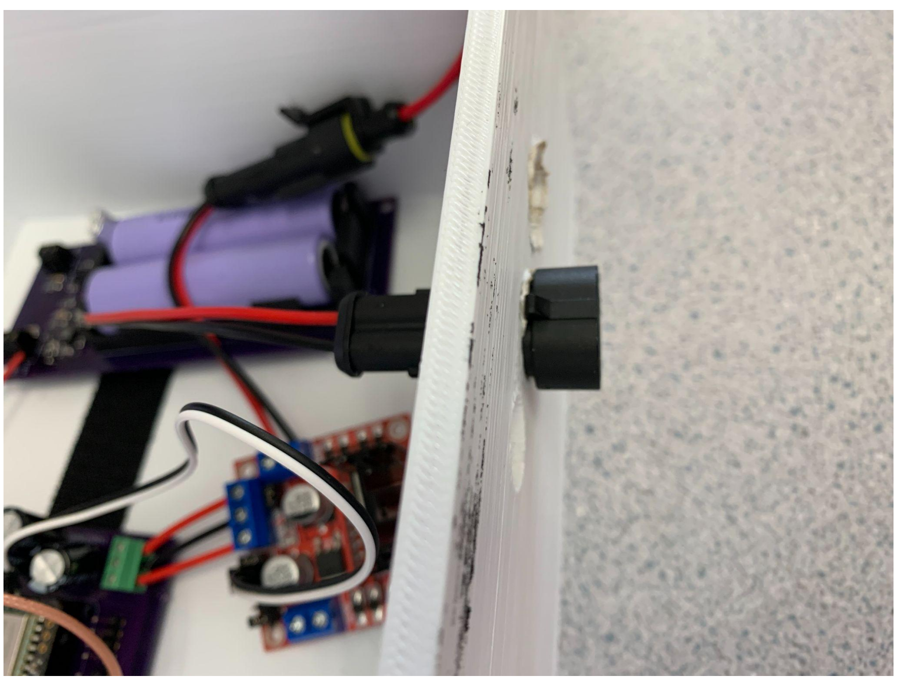


Figure 5.1.10: Solenoid and Power MUYI Connector in the Enclosure

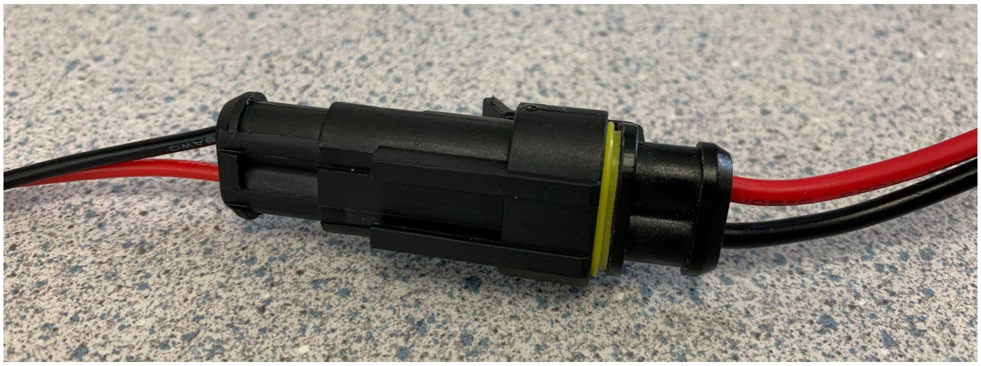


Figure 5.1.11: Close up of the Power and Solenoid MUYI Connector

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

Solar Charging Boost Converter	The power block has one primary efficiency drop, and that is in the boost converter, has a minimum efficiency of 78% at 0.5A. [1]
Watering Control Node	This power supply is using a pre-bought AC DC external power supply. As shown in the test below we can draw 4.8W when running at 1 amp, while using 7W from the wall for an efficiency of 68%.

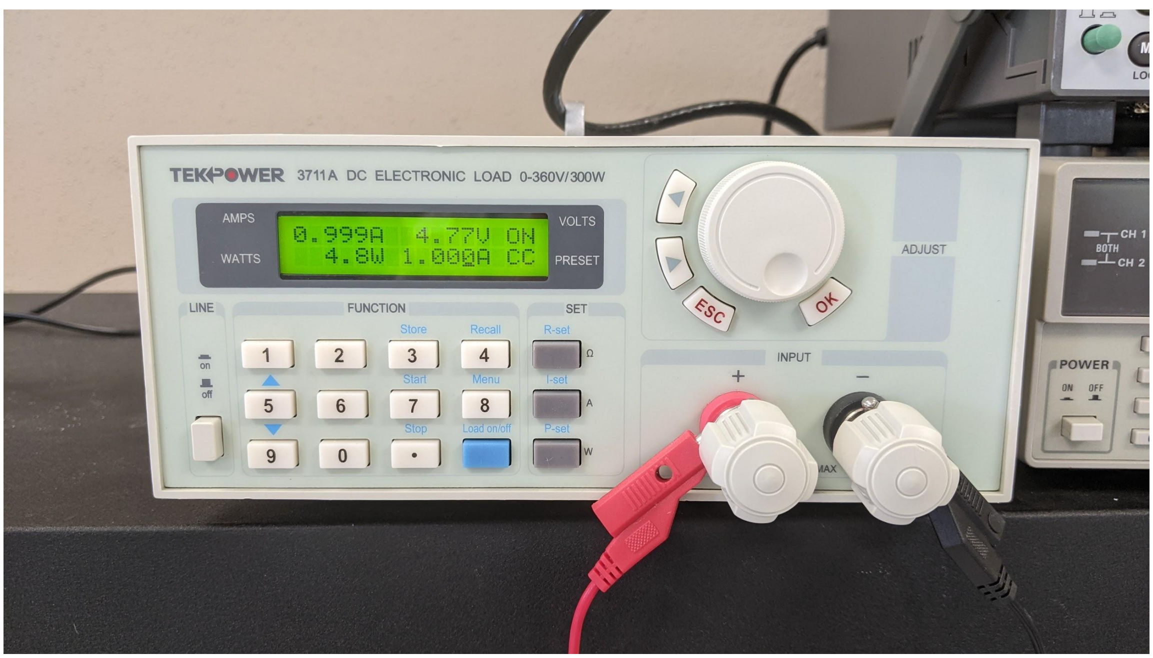


Figure 5.1.12: Efficiency Testing of the 12V Wall Jack Power Supply

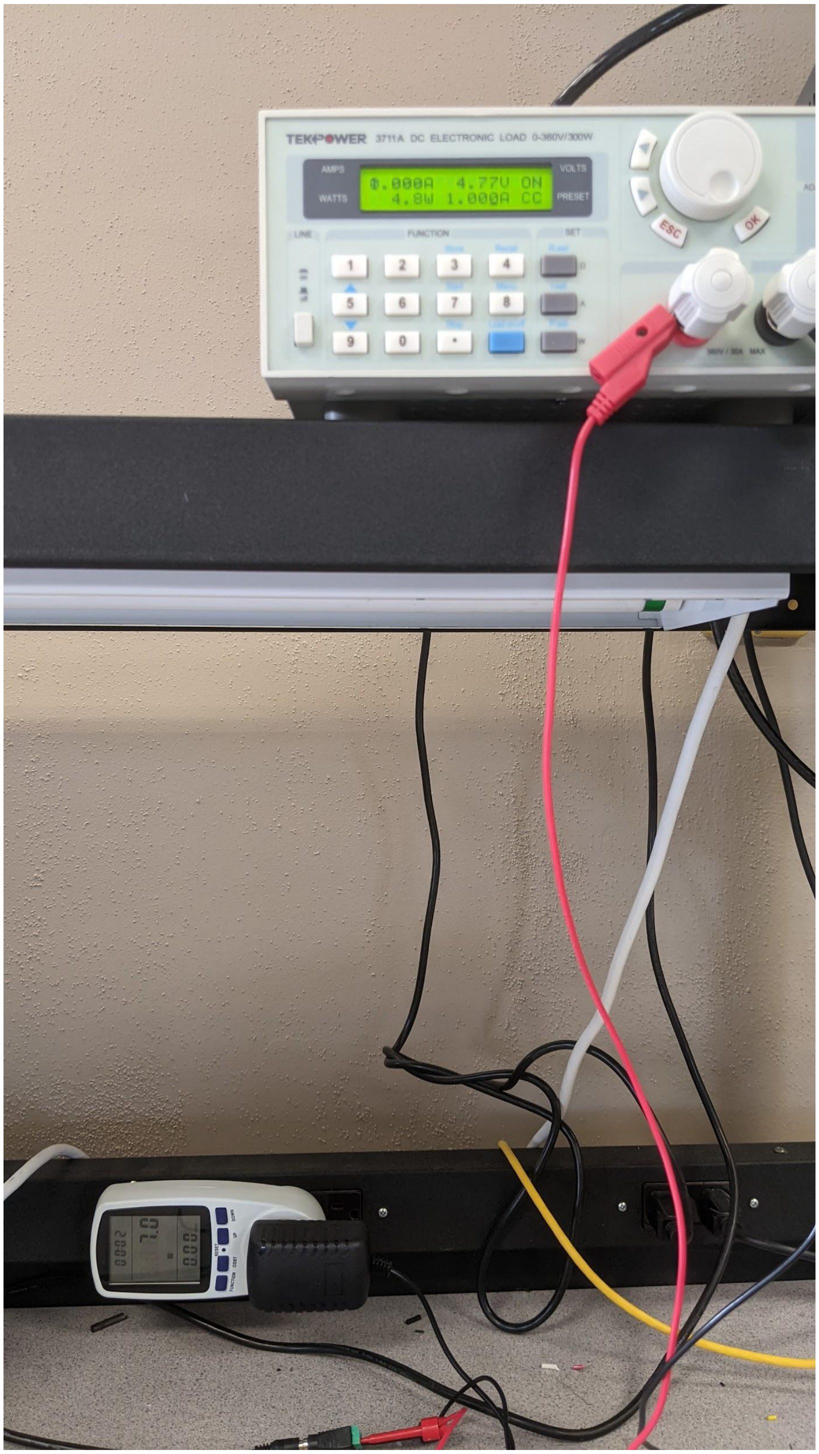


Figure 5.1.13: Full Efficiency Test Setup

Figure 5.14: Joulemeter Setup for Efficiency Test

5.1.6. **The system may be no more than 50% built from purchased 'modules.'**

Block	Contents
Solar Charging Boost Converter	Custom PCB designed with a BQ2057 IC and a LTC3872 IC to connect a 2 cell 18650 Li-Ion battery pack and a 10W/12V Solar Panel while providing a 12V, 0.5A output.
Watering Node Control	4x 1000uF Capacitor, Valve Solenoid, MUYI female and male connector, H Bridge Stepper Motor Driver, assembled together with soldering on a custom PCB.
Programmable Control Nodes	Adafruit feather m0 Lora, Lora antenna and ufl to sma female connector. Code was written by team
Watering Control Node	5V 2A wall mounted power supply, female-female power jack, soldered onto a custom PCB.
Short Range Communication	ESP8266 board, data logger module with SD card and ProtoBoard Shield connected together with short and long female pins. Code was written by team.
Website	Coded in full by Ekaterina Rott.
Enclosure	Enclosure was designed and printed by the team. Cable gland and insulation foam was purchased
User Guide	The User Guide was written by the team.

5.2. Engineering Requirement 1

5.2.1. Requirement: The system shall be powered directly by batteries and the batteries will be charged via a solar panel.

5.2.2. Testing:

Proof via Demonstration:

1. Solar Charging Test:
    a. Starting with the batteries below full charge, plug them into the solar charging PCB. Note the voltage of each battery cell.
    b. Connect the solar panel.
    c. Wait 20 minutes and check to see that the battery pack has increased in overall voltage with the solar panel disconnected.
2. Power via Battery Pack Test:
    a. Connect the batteries to the solar charging PCB.
    b. See that the Radio Communication board and 5V Buck Converter board LEDs are powered.
    c. Proceed through at least one opening and closing of the valve.
3. If both the Solar Charging test and Power via Battery Pack test are complete and passed, the requirement is fulfilled.

5.2.3. Evidence:

Solar Charging Test Evidence:

This portion of the requirement has not been met. In the design of the PCB, which is based on the design found on Figure 4 in the BQ2057 datasheet, the TS pin was left floating and the indicator LEDs in the 0.5A charger were removed due to the incorrect assumption that these were not necessary for chip functioning [2]. However, after PCB debugging, it was found that these elements are necessary for the BQ2037 chip to work. During testing of the Solar Charging PCB, the amperage into the batteries was nearly 1A and the PNP transistor heated up within a few seconds. The current should have been limited to 0.5A, and while the PNP transistor would generate some heat, it should not have been as hot as it got. This error in design was not found early enough to make the design changes.

Power via Battery Pack Evidence:

Figure 5.2.1 shows the Solar PCB connected to the Watering Control PCB and the solenoid. Figure 5.2.2 is the same image as Figure 5.2.1, but with the individual system elements labeled for easier reference. The batteries are fully connected, and the Solar PCB generates 12V at its output, which passes through the buck converter board to generate a 5V output to power the radio communication. Both the buck converter board and the radio board have their red status LEDs on - indicating they are fully powered.

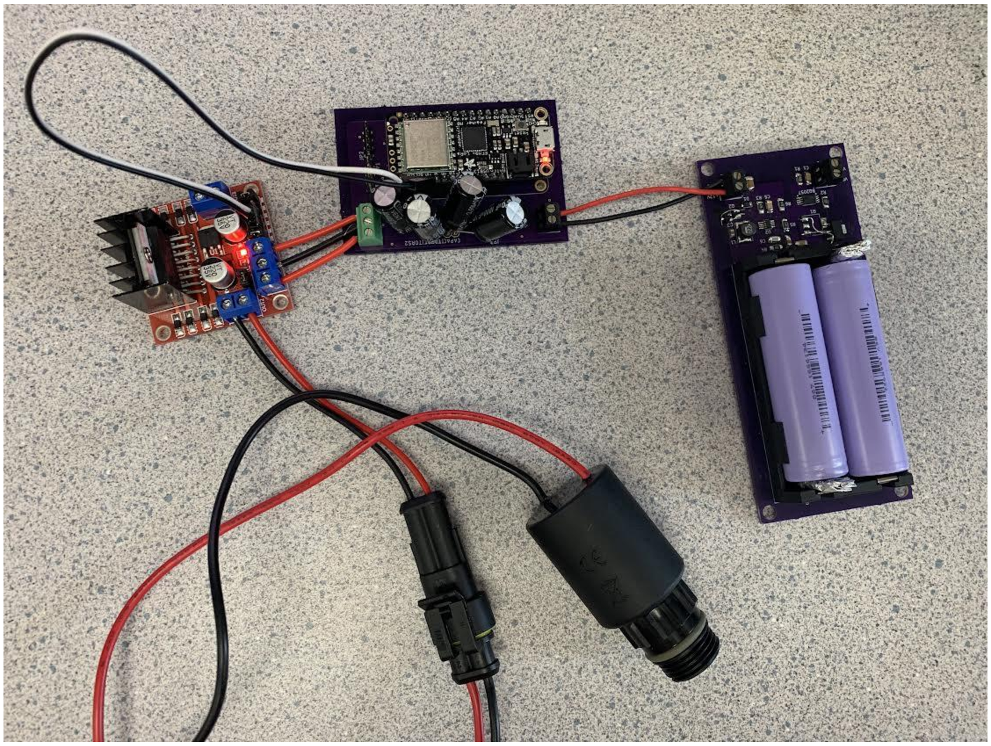


Figure 5.2.1: Fully Powered System with Batteries in the Solar PCB

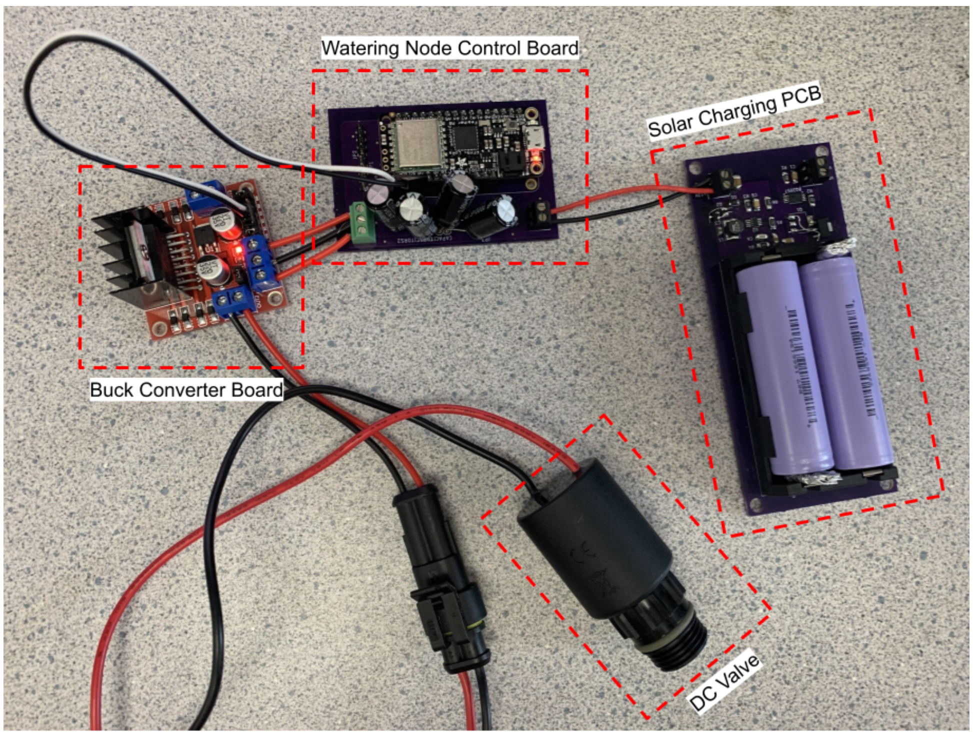


Figure 5.2.2: Fully Power System, labeled

Since the communication systems are not working, the valve movement was simulated by directly powering the solenoid control pins, which serves to demonstrate that the system can be fully powered by the batteries, if the communication systems were working. As shown in the evidence video linked below, the valve is able to be repeatedly opened and closed without any of the LEDs flickering or turning off - which would indicate loss of comms due to loss of power.

ER 1 Evidence Video

This requirement was originally Engineering Requirement 1 (“The system shall fully recharge dead 12V batteries in 7 days.”) and Engineering Requirement 2 (“The system shall run on fully charged 12V batteries for 3 days.”). However, these two requirements were too specific and we decided to condense the two requirements into one that better fit the Customer Requirement they were based off of (“The system shall be powered via a solar panel.”). This requirement change occurred midway through Phase 3 of the project, on 04/22/2022, and the approval for this change is shown in Figure 5.2.2.

Group 16 Engineering Requirement Change

External Inbox ×

Rott, Ekaterina <rottek@oregonstate.edu> 2:38 PM (1 hour ago)
to ece44x-ta@engr.oregonstate.edu

Hello,

We would like to change one of our engineering requirements as follows:

Old Engineering Requirement 1: The system shall fully recharge dead 12V batteries in 7 days.
Old Engineering Requirement 2: The system shall run on fully charged 12 V batteries for 3 days.

New Engineering Requirement 1: The system shall be powered directly by batteries and the batteries will be charged via a solar panel.

We would like to change the engineering requirement to better fit the Customer Requirement of "The system shall be powered by a solar panel", and be more descriptive of what we have actually accomplished this year. Since we previously had 9 Engineering requirements, collapsing these two requirements still leaves us with the 8 required.

Thank you,
Ekaterina Rott

Scheel, Ingrid 2:45 PM (1 hour ago)
to me, ece44x-ta@engr.oregonstate.edu

This is approved.

Figure 5.2.2: Approval for Engineering Requirement 1 Change

5.3. Engineering Requirement 2

5.3.1. Requirement: The system can communicate between nodes at least 600 ft apart.

5.3.2. Testing:

Proof via Demonstration:

1. Start with the nodes turned on and at least 10 ft apart.
2. The message should go through and the short range communication will receive the message and send it to the website.
3. Looking at the website, examine the "Current Node Status" table to make sure that a message has been received. [WSPDCVC Homepage (oregonstate.edu)]
4. Then move the node to at least 600 ft apart.
5. Again the message should be sent and received and the short range communication will receive the message and upload it to the website.
6. Examine the website to make sure that a message has been received.
7. With both the close range and long range test successful, this requirement is verified.

5.3.3. Evidence:

10 ft Distance Testing:

ER 2 Close Range Evidence Video shows that the system can successfully send messages to the website when in close range, at approximately 10 feet apart.

600 ft Distance Testing:

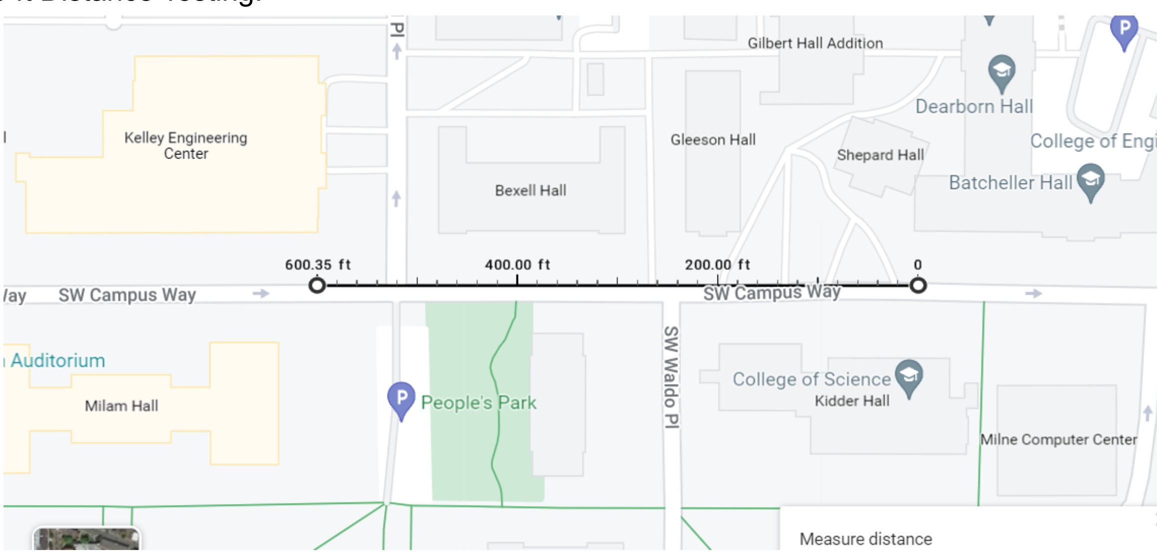


Figure 5.2.3: Distance between Two Nodes during Testing

ER 2 Distance Evidence Video 1 (Distance Node) shows where the "distance" node is during Video 2.

ER 2 Distance Evidence Video 2 (Home Node) shows the home node and the computer that has the website open to prove that messages are being received from the distance node 600 feet away.

With the current system, messages are able to be sent from a distance of at least 600 feet. we will be able to send messages with a distance of at least 600 ft. The successful test in Video 2 shows that the distance requirement is verified.

5.4. Engineering Requirement 3

5.4.1. Requirement: The system creates customizable automatic watering schedules through a web interface that shows the current status of the system.

5.4.2. Testing:

Proof via Demonstration:

1. Starting with both the watering schedule and the system data databases empty [WSPDCVC Homepage (oregonstate.edu)], navigate to the system scheduler page [Schedule Creator (oregonstate.edu)].
2. New Watering Schedule Test:
   a. Input a New Default watering schedule. Through inspection, see that the "Current Watering Schedule" table has changed.
   b. Input a second watering schedule, do not set it as Default. Through inspection, see that the "Current Watering Schedule" table has not changed.
3. Watering Schedule Manipulation Test:
   a. Navigate to the current schedule manipulation page webpage [Current Schedules (oregonstate.edu)], and perform the following tests:
   b. Change the current default schedule. See on the "Current Watering Schedule" that the default has changed.
   c. Change the current default schedule to no longer be default, such that there is no current default. See on the "Current Watering Schedule" that the default has changed.
   d. Set one of the two stored schedules to be default. See on the "Current Watering Schedule" that the default has changed.
4. System Status Test:
   a. Navigate to the system graphics page. Inspect the "Current Node Status" table. [Graphics Page (oregonstate.edu)]
   b. If the communication systems are running, the "Current Node Status" table will update 4 times an hour. Verify visually with the system turned on.
   c. If communication systems are not running, the "Current Node Status" table should say "No current system data".
5. If the New Watering Schedule test, Watering Schedule Manipulation test, and System Status test pass, this requirement is passed.

5.4.3. Evidence:

Proof without Communication Systems: ER 3 Video Evidence (w/o comms)

Since the communication systems were not completed for final system verification, the communication systems test could not be run. However, this component of the website was still verified through communication system testing. Figure 5.4.1 shows a test done to verify that the Wifi and Radio boards could communicate to each other and, in turn, to the website. The system messages shown in Figure 5.4.1 come from the communication system, successfully uploaded to the website. Further, the top row of the table states "System has not sent any data in more than 15 minutes. Please check that the system notes are still connected", which is the status condition wherein no communication from the system is working, where it had been previously. These tests prove that the website status functionality works as it should, both with and without the communication systems connected.

Current Node Status

Node ID	Current Schedule	Node On	Node Open	Error	Timestamp
System has not sent any data in more than 15 minutes. Please check that system nodes are still connected.					
ss	be	bl	ba	nO	2022-05-03 02:22:00
ss	be	bl	ba	nO	2022-05-03 02:15:51
ss	be	bl	ba	nO	2022-05-03 02:15:45
ss	be	bl	ba	nO	2022-05-03 02:15:37
ss	be	bl	ba	nO	2022-05-03 02:15:28
ss	be	bl	ba	nO	2022-05-03 02:15:16

Figure 5.4.1: Test of Communication System Connection to the Website

5.5. Engineering Requirement 4

5.5.1. Requirement: The system adjusts the valve within 5 seconds of the scheduled time.

5.5.2. Testing:

Proof Via Demonstration:

1. Use the website to schedule the valve opening and closing a minimum of 10 times total. [Schedule Creator (oregonstate.edu)]
2. Start a timer at the next minute, as displayed by a computer clock.
3. When the solenoid changes, mark the time and write it down.
4. After all prepared tests have been completed, compare the written times and the scheduled times.
5. Next, pull the SD card from the system, and inspect the schedule times to when the opening command for the SD card was sent.
6. If a minimum of 90% of both the physical opening times (step 4) and the opening command times (step 5) are within 5 seconds of the scheduled time, this requirement is fulfilled.

5.5.3. Evidence:

This requirement was not able to be verified in time for system verification. This requirement is heavily dependent on the entire communication system working: from the website all the way to the DC solenoid, and back. Theoretically, the communication between the Wifi and Radio boards are fast enough to communicate to the solenoid within 5 seconds, but the communication system has not gotten up and running after debugging. The primary bug in the communication systems is the inconsistency of the arduino communication between the Wifi and Radio board and between the Wifi board and the Website.

Figure 5.5.1 shows a single test where no variables were changed and no wires were shifted, and what the website was able to receive from the Wifi board. Throughout the same test, the website saw a blank JSON file, then correctly received the JSON information 4 times, before going back to seeing a blank JSON.

1354						2022-05-03 00:38:43
1353						2022-05-03 00:38:33
1352						2022-05-03 00:38:22
1351						2022-05-03 00:38:12
1350	you	are	my	sunshine	my	2022-05-03 00:38:02
1349	you	are	my	sunshine	my	2022-05-03 00:37:51
1348	you	are	my	sunshine	my	2022-05-03 00:37:40
1347	you	are	my	sunshine	my	2022-05-03 00:37:32
1346						2022-05-03 00:37:07
1345						2022-05-03 00:36:57
1344						2022-05-03 00:36:50
1343						2022-05-03 00:36:39

Figure 5.5.1: Radio Board to Website Test Example

The error in JSON communication could come from a couple places. First, there may be inconsistencies in the way arduino boards parse char* arrays into JSON objects, which is not controllable through the code without developing an entirely new library. Secondly, the issue could be with the length of the message themselves. There is a likelihood that there is a limit to how big of a char* array that the parsing function is able to handle, which includes both the JSON variable names and the content of the variables. While this might be the problem for other messages that had more content within them, this doesn't explain why the messages would randomly start and stop transmitting. The third potential reason for the incorrect transmission is the delay between transmissions from the Radio board. When the transmission delay was increased to 5 seconds, the above message in Figure 5.5.1 in rows 1347 through 1350 was successfully transmitted 89 times before stopping the test. Combining reasons 2 and 3, length of messages and transmission delays, the messages were sent more consistently, sending 75 messages with no issue before stopping the test.

id 1	nodeID	currentSchedule	nodeOn	nodeOpen	error	timestamp
1738	ss	be	bl	ba	nO	2022-05-03 02:22:00
1737	ss	be	bl	ba	nO	2022-05-03 02:15:51
1736	ss	be	bl	ba	nO	2022-05-03 02:15:45
1735	ss	be	bl	ba	nO	2022-05-03 02:15:37
1734	ss	be	bl	ba	nO	2022-05-03 02:15:28
1733	ss	be	bl	ba	nO	2022-05-03 02:15:16
1732	ss	be	bl	ba	nO	2022-05-03 02:15:05
1731	ss	be	bl	ba	nO	2022-05-03 02:14:58
1730	ss	be	bl	ba	nO	2022-05-03 02:14:49
1729	ss	be	bl	ba	nO	2022-05-03 02:14:44

Figure 5.5.2: Short Message with Long Delay Test

The test performed to create Figure 5.5.2 may have been the solution to the radio to website issues. Unfortunately that solution was found too late to attempt integration with the entire Website to Solenoid and Back communication system.

5.6. Engineering Requirement 5

5.6.1. Requirement: The system comes with a user guide that includes a title page, table of contents, introduction, explanation of system, system components, installation instructions, and troubleshooting.

5.6.2. Testing:

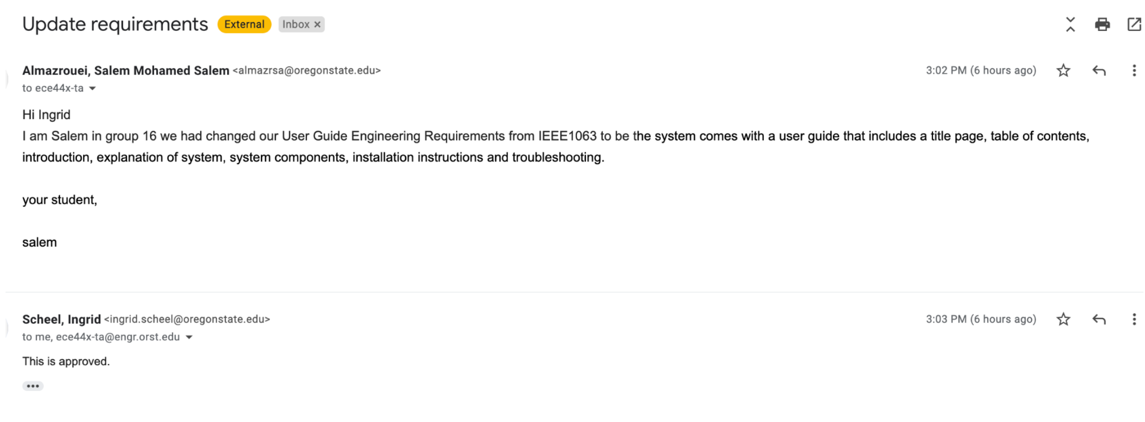
Proof via Inspection:

1. Present a hard copy version of the user's guide, with an online copy stored on google docs as a backup.
2. Going through the user’s guide with our designated point-person, Ingreed Scheel, ensure that the user’s guide includes all necessary sections.
3. The requirement is fulfilled with point-person approval.

5.6.3. Evidence:

[WSPDCVC User Guide]

Engineering Requirement 6 was updated late into the third phase to be more specific, specifically by changing the working from mentioning IEEE 1063 to referencing the specific aspects of the IEEE standard that this requirement is looking for. Figure 5.6.1 is the email for the approval for this late change.



Update requirements External Inbox ×

Almazrouei, Salem Mohamed Salem <almazrsa@oregonstate.edu> 3:02 PM (6 hours ago)
to ece44x-ta

Hi Ingrid
I am Salem in group 16 we had changed our User Guide Engineering Requirements from IEEE1063 to be the system comes with a user guide that includes a title page, table of contents, introduction, explanation of system, system components, installation instructions and troubleshooting.

your student,

salem

Scheel, Ingrid <ingrid.scheel@oregonstate.edu> 3:03 PM (6 hours ago)
to me, ece44x-ta@engr.orst.edu

This is approved.

Figure 5.6.1: Engineering Requirement Change Approval

5.7. Engineering Requirement 6

5.7.1. Requirement: The system sends messages through nodes with wireless communication systems that will successfully transmit 80% of sent messages. The system will send wireless confirmation that messages are received and will do 3 re-attempt messages if confirmations are not received.

5.7.2. Testing:

Proof Via Demonstration:

1. Have 10 messages to be sent between the watering node and the base node prepared.
2. The messages should be sent from the radio of the base node to the short range communicator and from there the message should be sent to the website. [WSPDCVC Homepage (oregonstate.edu)]
3. If at least 8 of the 10 messages are shown on the website, the test is passed.
4. Then, turn off the watering node to simulate a node being unable to receive messages.
5. The base node will give the message to the short range communicator and upload it to the website. This message will include how many retries in the message for us to see the "Error" column of the "Current Node Status" table.
6. If the node performs 3 retries before stopping, the test is passed.
7. With both step 3 and step 6 successfully passed, the requirement is fulfilled.

5.7.3. Evidence:

ER 6 Video

This part of the system was not fully tested but is partially fulfilled. Communication between radio boards has never failed unless one of the boards was off or on the wrong COM port, and the communication from the wifi board and the website was always able to transmit something, even if the JSON object received was empty. Once the radio and wifi boards were coded to properly connect, we never saw a failed message, only the contents of the JSON were unable to be properly parsed (as explained in the Section 5.5.3, Evidence for Requirement 4). Therefore, we are reasonably confident that the system does successfully transmit more than 80% of sent messages, such as is shown in the video linked above.

However, the communication system is currently in a spot where the messages sent within the system are not meaningful to either us or the user. Since there isn't an error messaging standard, there is no way to determine whether a message failed, or if a message failed and was retried.

5.8. Engineering Requirement 7

5.8.1. Requirement: The system records messages sent and received by the system to an internal data backup, holding at minimum 1 month of messages.

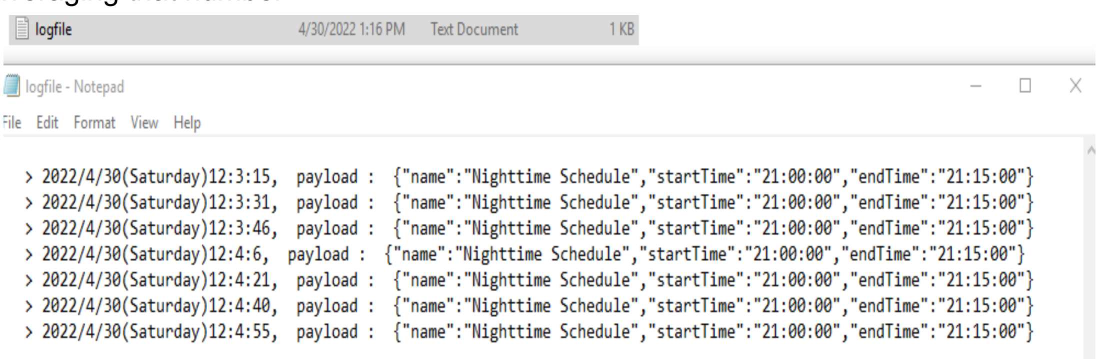
5.8.2. Testing:

Proof via Analysis:

1. Run 3 tests where a different total number of messages are written to the SD card.
2. Using the number of messages sent and the total size of the output file, find the size of one message for each test. Average the 3 tests to get the average size of one message.
3. Multiply the average message size by the total number of messages per an average month, ensure that the total space required for the month is less than the space on the SD card.
4. If the total space utilized is less than the space on the SD card, this requirement is fulfilled.

5.8.3. Evidence:

With the communication system running, there will be 4 updates per hour, every hour, for 30 days (the average size of a month). $4 * 24 * 30 = 2880$ messages on average, every month. Since there isn't a way to fetch the size of a message, a single average message size is pulled from Figures 5.8.1, 5.8.2, and 5.8.3 by finding the size of a single message in each instance and then averaging that number.



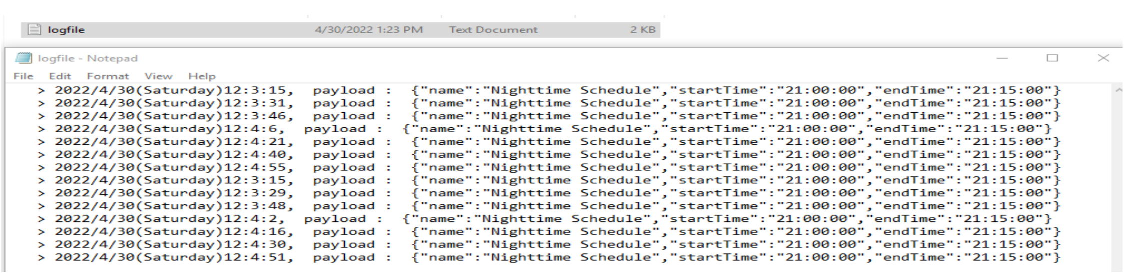
logfile 4/30/2022 1:16 PM Text Document 1 KB

logfile - Notepad

File Edit Format View Help


```
> 2022/4/30(Saturday)12:3:15,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:31,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:46,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:6,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:21,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:40,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:55,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
```


Figure 5.8.1: 7 messages sent, the SD card has 1kB of space filled.



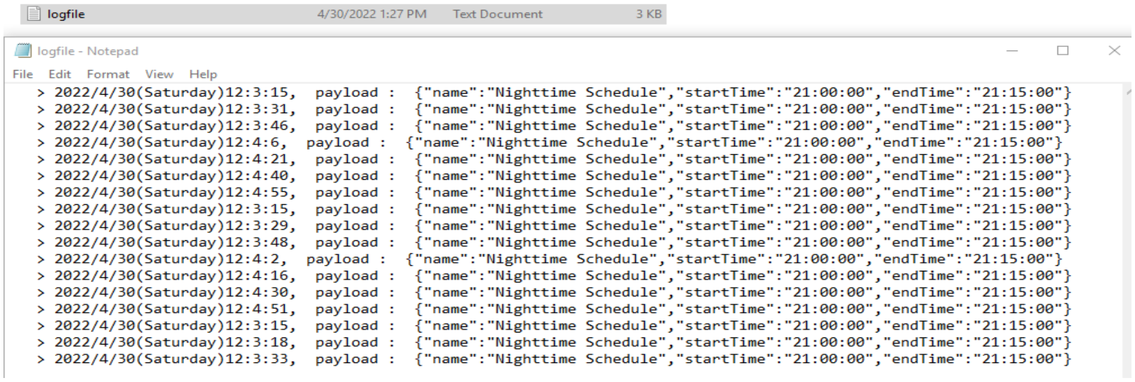
logfile 4/30/2022 1:23 PM Text Document 2 KB

logfile - Notepad

File Edit Format View Help


```
> 2022/4/30(Saturday)12:3:15,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:31,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:46,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:6,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:21,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:40,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:55,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:15,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:29,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:48,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:2,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:16,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:30,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:51,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
```


Figure 5.8.2: 14 messages sent, the SD card has 2kB of space filled.



logfile 4/30/2022 1:27 PM Text Document 3 KB

logfile - Notepad

File Edit Format View Help

```
> 2022/4/30(Saturday)12:3:15,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:31,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:46,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:6,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:21,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:40,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:55,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:15,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:29,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:48,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:2,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:16,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:30,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:4:51,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:15,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:18,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
> 2022/4/30(Saturday)12:3:33,  payload :  {"name":"Nighttime Schedule","startTime":"21:00:00","endTime":"21:15:00"}
```

Figure 5.8.3: 17 messages sent, 3kB of space on the SD card filled.

1. $\frac{1kB}{7mgs} = 142.86B$
2. $\frac{2kB}{14\ mgs} = 143.68B$
3. $\frac{3kB}{17mgs} = 176.47B$
4. $\frac{142.86+142.86+176.47}{3} = 154.063$ bites per message, on average

Therefore, with an average of 2880 messages per month, a month’s worth of messages should be

$154.063 * 2880 = 443.7kB$, on average.

The SD card has 28.8 GB of space total. Since 443.7kB is much less than 28.8GB, the SD card can hold a minimum of 1 month of messages.

5.9. Engineering Requirement 8

5.9.1. Requirement: The system circuitry will be protected by an enclosure that is water and dust proof, such that low pressure or low velocity water coming from all angles and dust larger than 1.0mm cannot penetrate the enclosure.

5.9.2. Testing:

Proof via Demonstration:

1. Begin with the system fully enclosed, with a paper towel inside the enclosure to catch any water that enters.
2. Mount the enclosure to a wooden pole as it would be mounted on a farm.
3. Using a hose, spray low velocity water onto the enclosure from various angles.
4. Toss soil onto the enclosure.
5. Take the system down, and remove the solar panel. Inspecting the inside of the enclosure, ensure that the paper towel is not wet, with no water with the enclosure otherwise, and no dirt within the enclosure.
6. If step 5 is successfully passed, the requirement is fulfilled.

5.9.3. Evidence:

This requirement has not been fulfilled.

First, the holes in the side of the enclosure are not suited for the connectors that we have ended up using in the final system. While this is solvable by melting out the connector locations, melting out the holes has left space for water and dust to enter. This would be solvable with a glue like hot glue, although none of the team members have easy access to a glue gun.

The second major issue with the enclosure is re-sealability. In order to perform other tests, such as the verification for Requirement 4 or 7, the user has to be able to enter the enclosure. The current sealing system is water-tight tape, which has to be broken and reset each time the user wants to get in.

Finally, the current enclosure does not include a way to mount the whole system to something. Without this mount, enclosure testing isn’t able to be done.

5.10. References and File Links

[1] Linear Technology. “No Rsense Current Mode Boost DC/DC Controller,” LTC3872-1 datasheet, Accessed May 2022. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/38721f.pdf

[2] Texas Instruments. “Advanced Linear Charge Management IC For Single- and Two-Cell Lithium-Ion and Lithium-Polymer,” BQ2057TTSTR datasheet, May 2001 [Revised Jul. 2002]. Available: https://www.ti.com/lit/ds/symlink/bq2057.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1651887716106&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsuppproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%252Flit%252Fgpn%252Fbq2057

5.11. Revision Table

Table 5.6.1: Section 5 Revision Table

Date	Changes Made	Author
10/31/2021	Section Outline Created	Ekaterina Rott
03/05/2022	Added Universal Constraints, Engineering Requirement 3, and Engineering Requirement 6	Ekaterina Rott
03/05/2022	Filled in Universal Constraints 1, 2, 4, 5, 6	Orion Holar
03/06/2022	Added requirement for Engineering Requirement 6	Salem Almazrouei
03/12/2022	Added information to Universal Constraints 5 and 6	Ekaterina Rott
04/22/2022	Condensed Requirements 1 and 2 into the current Requirement 1. Removed Requirement 9	Ekaterina Rott
04/22/2022	Updated requirements 5 and added requirement 7.	Salem Almazrouei
04/24/2022	Reformated and Reworded Universal Constraints proof	Ekaterina Rott
04/24/2022	Rewrote all testing procedures to better describe testing processes	Ekaterina Rott
04/26/2022	Added in Solar Charging PCB picture, worked on filling in Universal Constraints with current system build	Ekaterina Rott
04/29/2022	Redid formatting to be consistent across all Engineering Requirements	Ekaterina Rott
05/01/2022	Added in evidence for Requirement 7	Salem Almazrouei
05/02/2022	Completed Requirement 1, 3, and 7 write ups	Ekaterina Rott

05/02/2022	Adjusted requirement testing procedures	Orion Hollar

6. Project Closing

6.1. Future Recommendations

The purpose of this section is for any future team that picks up this project, so that they may learn from our work and know what we would change in future iterations. There are 3 types of recommendations, those being technical, global impact, and teamwork recommendations. The technical recommendations revolve around what we would change about the design and our design process. Global impact recommendations include how we would adjust our design or design process to reduce the global impacts of this design. Finally, teamwork recommendations include how we would have structured our team or worked together differently to better achieve design and timeline goals.

6.1.1. Technical recommendations

Recommendation 1: Start design early, even though there aren't assignments for it early on.

Rushing the design process to make an assignment deadline can cause lasting damage to your ability to work on the system. By the time issues with a rushed design are discovered, it may be too late in the design or integration process to start over. Having at minimum vague design in mind within the first couple weeks is also helpful for making the design impact assessment in the first term, as you could do more specific and more useful research based off what will actually be in the design.

Recommendation 2: Write requirements that are independent of each other.

One reason for this is for the sake of efficiency. If some requirements depend on others being complete, it can waste time during integration by forcing a critical path in integration that may take too long to complete. If they are independent, work can be done in parallel and be more productive, and also prevents one requirement failing from cascading into several other failures. This also means ensuring that system blocks relate to only one or two requirements at most.

We ran into issues completing our Engineering Requirements because two of the blocks, the Short Range Communication and Programmable Control Nodes, were critical to the success of about half our requirements. During integration, we faced issues were we could not get those two blocks to connect properly, which meant the failure of that one connection cascaded into half of our requirements, hurting our ability to create a finished system.

Recommendation 3: Be thorough in interface design, it will save time later on. Test interface connections in the second term when possible.

In the second term, block validations consist of showing that a block can produce or take in the desired interface properties. While it is easy to cheat this process and force an interface to give some desired value, it is a better use of time to build the block as if it were actually part of the system to save time integrating later on. For example, saying a program can output to a screen could be done either by just doing a simple looping print statement, or by creating the real program that prints relevant data out without wasting time making a dummy program.

Interfaces are the primary hurdle of integration, so working on them early will make for an easier final term. While our project passed all block validations, we did not do them in such a way that ensured the blocks on either end of the interface would actually connect. This left us having to redo the interface validations during integration, holding us back from making progress and running requirement tests.

Recommendation 4: Make sure to allocate a lot of time for the code, and do a lot of research. Avoid arduino code where possible - it is buggy and difficult to work with.

It has a lot of work to do and you should know how to connect between other blocks like for example the wireless connection or between blocks to send and receive data which is something not easy to do if you are not interested in the arduino code in the project.

During integration we had to do a lot of debugging to figure out what was wrong with our code. This was due to not having enough experience with functions. For example we ran into an issue with the serial.read and serial.write functions. Even though we were using example codes it was still not running as it should have. After a bit of time we learned that on the adafruit board serial refers to the USB or the visible serial monitor. While Serial1 is the actual rx and tx pins on the board. Another coding issue we ran into was not being able to send the whole JSON string to the website. This could have been an issue of the size of the string. For the most part if we had more time we would have been able to figure out a way to solve the problems.

Recommendation 5: When designing a PCB, make sure to look at the order of the pads for MOSFETs, BJTs, and other transistors, and make sure the pads on your design and the pads on the transistor's datasheet match up.

During the design of the Solar PCB, the schematic symbol for the N-Channel MOSFET and the PNP BJT were chosen at random, due to limited understanding of how many transistors set up the pads of their components differently. Both of the transistors in the Solar PCB design had the wrong footprints, meaning the connections had to be jumped to the correct pads, decreasing the amount of time to work on other issues and test the PCB.

For example, figures 6.1.1, 6.1.2, and 6.1.3 show the schematic symbols available for an NMOS transistor, which one was chosen for the Solar PCB, and what the actual NMOS transistor's footprint is. The schematic would require a gate/source/drain footprint rather than the gate/drain/source symbol chosen, and this can only be determined after the actual transistor has been chosen.

If the Solar PCB had time for a second revision, these footprints would have gotten corrected. There are other issues with the current PCB design, but having these footprints correct to begin with would have sped up locating other issues.

Device	Generic symbols for common devices
Q_NMOS_DGS	N-MOSFET transistor, drain/gate/source
Q_NMOS_DSG	N-MOSFET transistor, drain/source/gate
Q_NMOS_GDS	N-MOSFET transistor, gate/drain/source
Q_NMOS_GDSD	N-MOSFET transistor, gate/drain/source, drain connected to mounting plane
Q_NMOS_GSD	N-MOSFET transistor, gate/source/drain
Q_NMOS_SDG	N-MOSFET transistor, source/drain/gate
Q_NMOS_SDGD	N-MOSFET transistor, source/drain/gate, drain connected to mounting plane
Q_NMOS_SGD	N-MOSFET transistor, source/gate/drain

Figure 6.1.1: NMOS Schematic Symbol Options in KiCAD

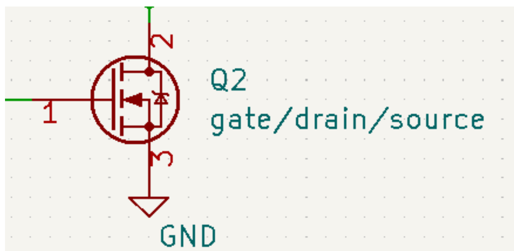


Figure 6.1.2: NMOS Schematic Symbol Chosen in PCB Design

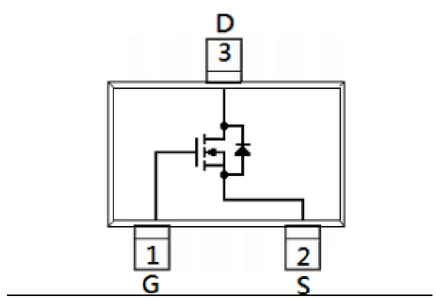


Figure 6.1.3: Actual NMOS Transistor Chosen [1]

Recommendation 6: When designing and ordering a PCB, make sure that everyone who has blocks connected to the PCB validates and approves the PCB design.

While designing the watering and home node PCBs, there were some last minute changes made to post-PCB production. This could have easily been avoided if all team members connected to the PCBs had validated the design before ordering them, and would have saved time during integration as jump wires would not have to be soldered to the PCBs.

Another benefit of this recommendation is that in general having multiple eyes on a designs makes a huge difference. Small mistakes in PCBs have a tendency to roll into larger, harder to correct mistakes once a PCB is ordered, so having a PCB validation process before ordering can save time and frustration later.

6.1.2. Global impact recommendations

Recommendation 1: Provide more recycling options for the end user.

Disposal of project parts is potentially a large source of waste. For a single system being made it may not seem like you will have much unused materials, but there can be several intermediate stages of the project with parts that get cut away and need to be dealt with in one way or another. Keeping track of sustainable ways to dispose of as many parts as possible ensures that you aren't left with a pile of prototype parts to try and deal with at the end of the project. Properly recycle the battery when changing and dispose of the replaced battery properly.

Recommendation 2: Be careful with the type of battery you use.

One potential global impact that we did not consider was making sure to use non toxic batteries for this project. One of the big ideas for this project was the irrigation system for a farm, and since the watering node will be out in the field that also means the battery would be as well. Having a battery leak potentially toxic substances into the field would greatly impact the farm and the crops that are growing. Here is a quick guide provided by the government for how to dispose of many different battery types [2].
https://portal.ct.gov/DEEP/Reduce-Reuse-Recycle/Batteries/Managing-Household-Batteries

6.1.3. Teamwork recommendations

Recommendation 1: Re-read engineering requirements regularly while designing.

Over the lifespan of the project make sure to keep re-reading the engineering requirements while designing the system. Revisiting requirements helps ensure that a block design will actually fulfill a given requirement, or will inform if the requirement needs to be changed as early as possible. This also prevents missing minor details in a requirement. We began with 9 engineering requirements and ended up changing 4 of those 9 requirements at the last minute due partially to scope creep and partially to our project blocks not fully fulfilling the requirements. Had we been reading over the requirements while creating our block designs, we could have either made those requirement changes earlier or built designs that fulfilled the requirements from the beginning.

Recommendation 2: For each block in the design, have a primary block champion and a secondary one.

This recommendation is twofold. First, another team member would be able to help the block champion make progress on the block if they get stuck. Secondly, this would mean that information and technical knowledge is distributed within the team and not centralized onto one person. Having technical knowledge spread amongst the team is critical during the integration phase of the project, as it allows integration progress to occur without needing the entire team present.

This is especially helpful if there are two blocks with a critical interface. We ran into issues connecting the Short Range Communication and Programmable Control Node blocks, but since the code and overall technical knowledge (such as which drivers were needed for each board) were isolated to the block champions alone, we were unable to troubleshoot this connection unless everyone was present, leading to major slow-downs in integration.

The following is a short article on the importance of cross-training, which we are recommending [3]. https://www.reliableplant.com/Read/31316/back-up-champion

Recommendation 3: Utilize a file sharing system like GitHub to share technical information like code, schematics, mechanical drawings, etc, so that the whole team has access to the entire project.

While we did utilize Google Drive for sharing files such as the Project Document and project update slides, we did not have a way to readily share technical information such as code, requisite drivers for the communication boards, or schematics. This became a problem when trying to troubleshoot connections in the system, because the entire team needed to be present in order to work on the project, but if we had a sharing system where all technical information was stored, we would be able to proceed even with a missing team member.

This recommendation, when combined with recommendation 2, means that half the team could be missing but progress could still be made.

Recommendation 4: Do not wait till the last minute to build the whole project and verify every requirement.

Ideally, have the whole project built one week before the final system verification. When it comes to integration, the earlier difficult bugs are caught, the faster troubleshooting could happen. We waited too long to test blocks on either side of an interface together, like the short range and the long range communications. This left us with bugs in our system along critical interfaces, which single handedly took 4 days to fix, eating up almost all of the integration time. Had we started integration work earlier, we would have had more time to work on fixing the communication systems and might have gotten to requirement testing. Starting as early as possible means having more time to troubleshoot and most likely have fewer late nights preparing for verification.

Recommendation 5: Ensure each team member knows about and has access to the fastest communication method with each team member.

There are many times during the project that you need information from another team member to progress. It can be disruptive to work flow to reach that point and need to stop and wait for potentially long periods of time waiting for a response because the message was sent through a method that they don't check as regularly. Different team members will likely have different fastest ways to communicate with them, even if you all have one shared way to communicate it may not be the best for reaching a single person. For example the fastest way to reach other team members is by sending them an email or by calling them if they did not respond quickly.

6.2. Project Artifact Summary with Links

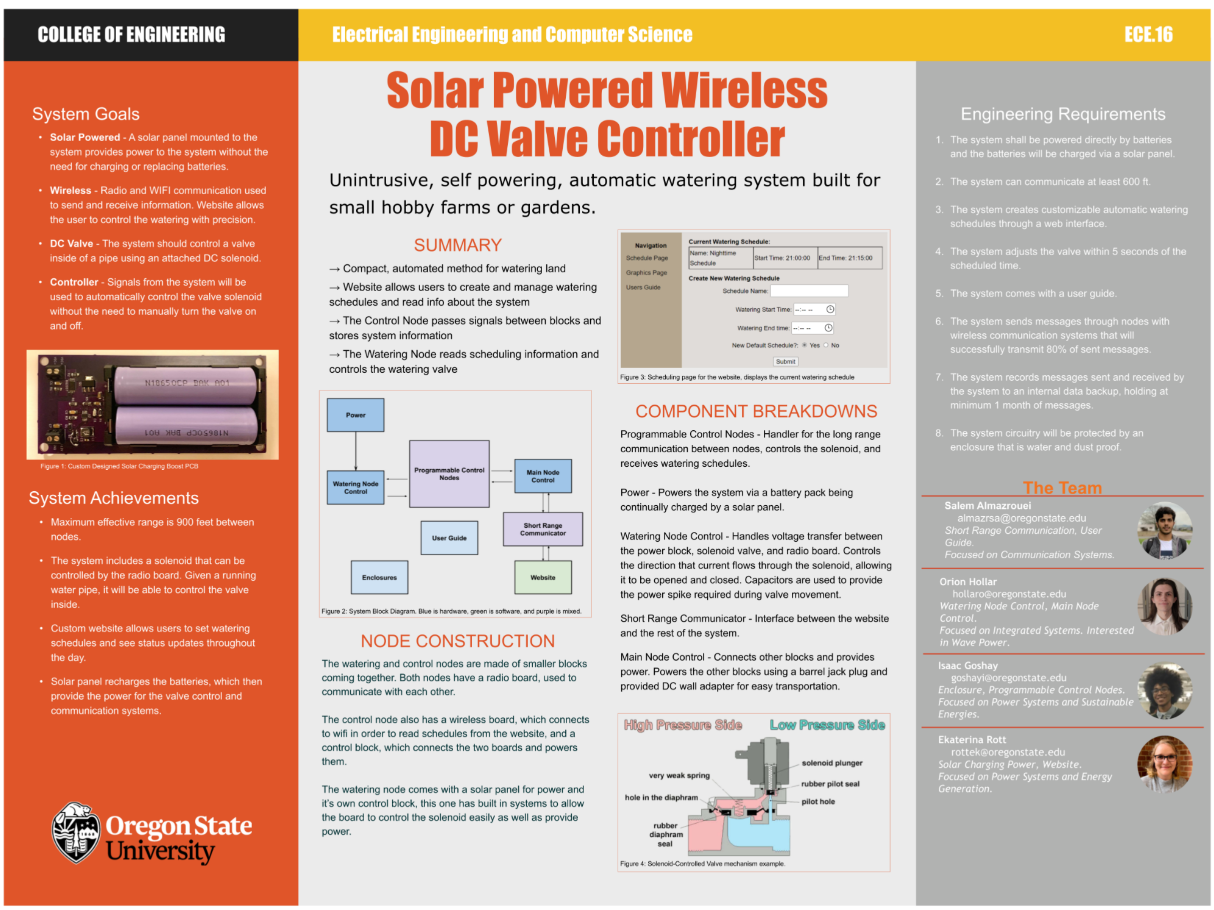
All PCB design files, coding files, and mechanical drawings can be found on the Project Showcase website for download. [Showcase Project | OSU (oregonstate.edu)]

Table 6.2.1: All Artifacts Available on the Project Showcase Website

Artifact Name	Artifact Description
Solar PCB KiCAD Development Files	One of the custom PCBs in the project is a solar charging boost converter. This PCB takes in the voltage output from a solar panel, charges a 2-cell Lithium-Ion battery pack, and boosts the battery pack voltage to 12V. This PCB was developed in KiCAD, and the .zip file contains the schematic, component footprints, PCB design, and gerber files. This Solar PCB is in Revision 1, changes include fixing the MOSFET and PNP BJT footprints to match the transistors used in the design, adjusting some capacitor and resistor values to reduce output oscillations, and redoing the solar charging side of the PCB.
Watering Node PCB EagleCAD Development Files	One of the custom PCBs in the project is a base for the Watering Node. This PCB connects to the Solar PCB, holds the long range communication board, and connects to the solenoid driver. This PCB was developed in EagleCAD and includes the schematic and board files.
Control Node PCB EagleCAD Development Files	One of the custom PCBs in the project is a base for the Control Node. This PCB holds the short range and long range communicator, connects them and allows for a barrel jack power plug in. This PCB was developed in EagleCAD and includes the schematic and board files.
Sending and Receiving code	The main code for the long range communication using the Adafruit feather M0 LoRa. The code makes sure that the radio runs at 915Mhz sends and receives message, and sends the information using UART communication to the short range commination board.
Connecting between the website and the main code	The Short Range Communication is connecting to the Website with wifi to get and receive data from the Main node block. Also ESP8266 is the board that connected to the main node with connecting Rx to Tx and Tx to Rx and ground to ground. Also we have the SD card on the data logger board which save the data for month that was sent between the Website and the Main Node.
User Guide	This User Guide has all the instructions for new users and experts for installing and operating the system and it has trouble shooting section which helps to avoid any issues us comes up.

Enclosure CAD Development Files	The Enclosure is what holds the rest of the project and protects the system for water and dust. The enclosure should be able to prevent low velocity water from all angles making it inside. This enclosure was 3d printed so that it would fit behind our solar panel.
Website PHP and HTML Scripts	This project included a custom website that allows users to set custom watering schedules and see current system statuses. This website includes the ability to clear all system data and schedule data from the MySQL server that acts as a backup for system data, and allows the user to change which watering schedule is currently operating. The MySQL server this code is attached to will not be supported past June of 2022.
Project Expo Poster	PDF Copy of the Project Expo Poster.

6.3. Presentation Materials



6.4. References and File Links

[1] Panjit Semiconductor. “30V N-Channel Enhancement Mode MOSFET,” PJA3404 datasheet, March 2014. Available: https://www.panjit.com.tw/upload/datasheet/PJA3404.pdf

[2] R. Brieck, “Back up your champion, not just your database,” Reliable Plant, 03-Oct-2018. [Online]. Available: https://www.reliableplant.com/Read/31316/back-up-champion . [Accessed: 06-May-2022].

[3] “Managing household batteries,” CT.gov, 2020. [Online]. Available: https://portal.ct.gov/DEEP/Reduce-Reuse-Recycle/Batteries/Managing-Household-Batteries . [Accessed: 06-May-2022].