

## **Polycrop Smart Drip Irrigation System**

### **Team 17**

Vedant Chopra, Victor Carpio, Joshua Mica, Diego Tolentino (Team Leader)

**May 9, 2021**

**4800 Calhoun Rd.,  
N308-Building 579  
Electrical  
Engineering,  
University of  
Houston  
Houston, TX**

Faculty Advisor:  
**Dr. Leonard  
Trombetta**  
**Ph:** 713-743-4424  
**Email:**  
ltrombetta@uh.edu

Project Manager:  
**Dr. Steven Pei**  
**Ph:** 713-743-4433  
**Email:**  
peihu4@gmail.com

Hello to whom it may concern,

This report details our progress towards the completion of the Polycrop Smart Drip Irrigation System as of the end of the Spring 2021 Semester. The objective of this semester was to complete the assembly of our system and verify the functionality through implementation in an outside environment.

In the following report, we will explain the purpose of our project, elaborate on the user analysis and product specifications, provide an overview diagram, detail the engineering specifications, explain our test plan, provide updates from our schedule and budget, and address our risk matrix and mitigation plans. The appendix also contains a guide for the construction of our system.

So far our team has developed plans on how to tackle the issues outlined by the previous teams, while also adding impactful updates to the system so that our smart-drip irrigation system helps the Nicaraguan farmers more effectively. The main feature of the design we seek to improve on is providing the capability of watering different types of crops, making our system a polycrop irrigation system.

In spite of the continuation of the pandemic, we, Team 17, completed creating our system at the end of the Spring 2021 semester. Our team is thrilled to produce a functional polycrop smart drip irrigation system. We are excited that our project will help farmers in Nicaragua!

Warm regards,  
Team 17



# Polycrop Smart Drip Irrigation System

Team 17

---

## Third Progress Report

Presented to

The Department of Electrical and Computer Engineering

University of Houston

---

By

**Diego Tolentino (Team Leader)**- Applied Electromagnetics

**Vedant Chopra**- Power and Renewable Energy

**Victor Carpio**- Computer Engineering

**Joshua Mica**- Power and Renewable Energy

---

**Project Manager:** Dr. Steven Pei

**Faculty Advisor:** Dr. Leonard Trombetta

**Sponsor:** IEEE EDS, BioNica, and Universidad Nacional Agraria (UNA)



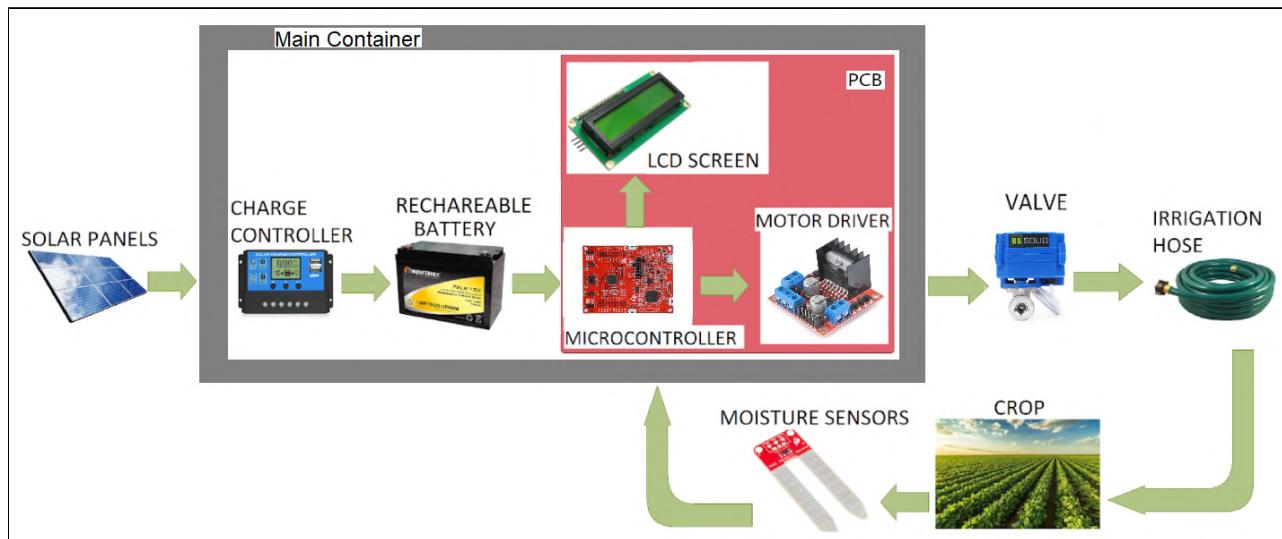
## **Purpose:**

The purpose of this project is to provide an updated and improved smart drip irrigation system (SDIS) to low-income Nicaraguan farmers as part of a continuation project. The previous system is designed to irrigate a single type of crop - a monocrop system - while our team looks to design upgrades that will allow for the irrigation of multiple crops - a polycrop system - depending on soil moisture levels. Currently, the agricultural activity in Nicaragua accounts for roughly 17% of the nation's Gross Domestic Product with roughly 31% of the population being employed in the agricultural industry, and an estimated 24.9% of those people are living below the poverty line, meaning they are living with less than \$3.20 per day [1][2][3]. The non-profit partnership of this project will look to provide economic and technological relief to farmers by providing a durable and easy-to-use irrigation system.

## **Overview Diagram:**

Our system begins with the solar panels, which harness the solar energy and convert it into electrical energy. That is then inputted into the charge controller, which regulates the rate at which the electrical current is added or drawn from the battery. Our rechargeable lead-acid battery then provides voltage to the microcontroller, which functions as the brain of our system and is responsible for managing the status of the moisture sensors and opening and closing the valve.

The microcontroller will also control the entire irrigation process, control the LCD screen outputs, and will have different options for the farmers, allowing them to customize the smart-drip irrigation system. The LCD screen will be the way our system will communicate to the farmers. The motor driver will control the motor, so whenever the crops need water, the motor will turn the valve, which in return will give water to the system through an irrigation hose. Once the moisture sensor finds that the soil is no longer dry, it will give that feedback to the microcontroller, which will then tell the motor driver to turn the valve off. In order to protect our system from the harsh Nicaraguan climate, our team decided to keep our components in an IP67 rated container, with the capability of water and dust proof resistivity.



**Figure 1.** Overview diagram depicting how the Polycrop Smart Drip Irrigation System

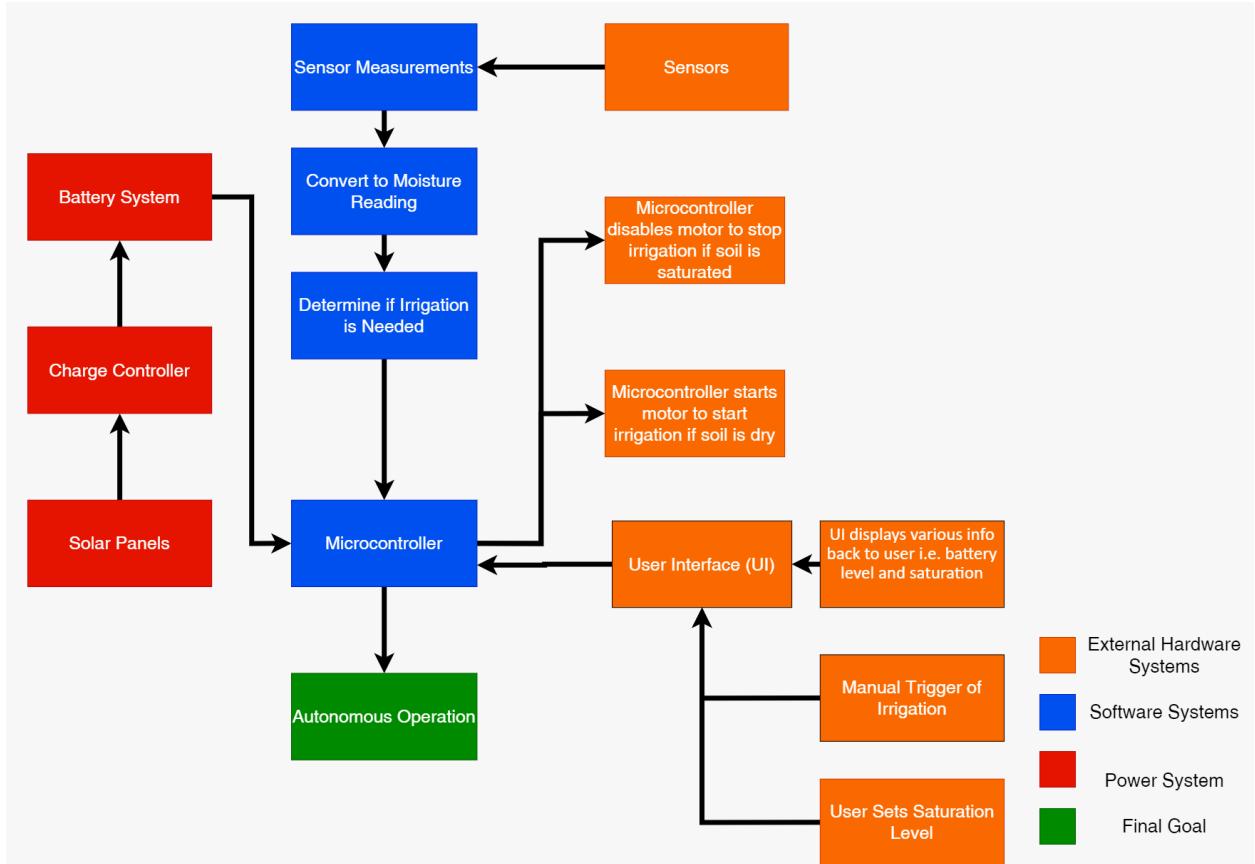
## Deliverables:

By the end of the semester we will have a fully operational Smart Drip Irrigation System. This system will have the ability to allow the user to select among predetermined saturation levels. If our saturation levels are not sufficient for the crop that is being grown in the plot of land, there will be a manual bypass that allows the user to open the valve and dispense more water. Our system will operate on 12 [V] DC which will be supplied by an array of solar panels which will be configured to have a voltage of 20 [V] DC. Our system will be housed in an IP67 rated enclosure. By that standard the enclosure will be waterproof, dustproof, and heat resistant which will allow it to survive the Nicaraguan weather since this system is to be operated outside. The electronics inside the container are rated to operate in temperature not to exceed 113°F, or 45°C, which is more than sufficient in the Nicaraguan environment. The system will have the ability to irrigate a maximum area of to 9 [m] x 16 [m] as this is the area of the plot of land the previous team installed in Nicaragua. Our system will periodically check if we need to open the valve every two hours currently, but this is subject to change as further testing continues. This contrasts what the previous team planned which was to water at night as that can cause a growth of mold on the crops. Our system will also have a new interface to display system information such as

current saturation, current mode, and any errors occurring. This interface will consist of an LCD and buttons to make selections and display the mentioned system information.

## **Test Plan:**

When the system is powered the user will be prompted to select the desired saturation level setting. This choice will be between a low, medium, and high level of saturation. Once the saturation point has been set the system will go into an idle state waiting for the moisture sensors to detect that the soil saturation is dry for our set saturation level. Once this occurs the microcontroller comes out of its idle state and starts the motor on the ball valve so that it will be in the open position. This will allow water to flow through the plumbing of the system and irrigate the crops. Once the sensors detect that the soil has reached the proper saturation threshold, the motor to the ball valve is triggered, closing the valve, and thus stopping the water from flowing. While this is occurring the LCD output will be displaying the saturation and current mode so that the user can ensure the system is working properly. Additionally, there will be a manual trigger that will open the ball valve should the system fail or if the highest saturation level is still not able to saturate the type of crop as needed.



**Figure 3:** Goal Analysis Diagram

In order to analyze the goals of our deliverables, we need to explore the features and behaviors of our system. In this section of the report, we analyze the sections of the system that are measurable and observable. Further, we provide Table 1, which contains our parameters, target values and tolerances needed to validate the product specs and engineering specs, and a description of how the testing will be conducted. Table 1 describes the relation between the input of the moisture sensors and what is expected to be outputted to the user interface (UI) and activate or deactivate the motor depending on the saturation. We have also provided our test plan for the main components of our system to verify their functionality.

**Table 1:** LCD Output Dependent on Probe Voltage

Probe Voltage	Soil Status	LCD Message
$0 - V_{SAT\ LOW}\ V$	Dry	“Under Saturated”
$V_{SAT\ LOW} - V_{SAT\ HIGH}\ V$	Sufficiently Saturated	“Saturated”
$>V_{SAT\ HIGH}\ V$	Over Saturated	“Over Saturated”

### Task 2.5-Moisture Sensors:

The sensors will be tested using three different soil samples that are representative of the soil conditions where the system is to be implemented. These soil samples were obtained from our project sponsor BioNica to accurately test the sensors and overall system. We will then attach the sensors to the microcontroller and take readings on test soil and output these readings to a computer terminal using a UART communication protocol. This will work with the user interface to test variable moisture settings. Different water levels will be added to each sample to attempt to replicate weather conditions.

### Task 1.5-Printed Circuit Board (PCB) Connections:

In order to check if the connections of the PCB are proper, we will have a two step plan. First, before ordering the physical PCB, we will run software simulations to check if the PCB will work as intended. EasyEDA, the software used to design the PCB, has a button which allows the software to run an analysis on the PCB. Once this analysis is complete, it will recommend any changes or incomplete connections it spots. After that, we can also check how the system will look in a 3D rendering. This is the time we can test if everything is good and as intended. Finally, once everything is complete and we order the designed PCB, we will commence physical testing. Using the lab at the University of Houston, we will send a voltage through connection paths using a bench power supply. Then, using the multimeter, we will measure the voltages and verify traces are pathed correctly. Once everything is functional, we will solder on the components. In case anything happens during testing, we will have duplicate PCBs as a precaution.

### **Task 2.3-Power Delivery:**

Our power delivery system's functionality will be verified by connecting the battery and solar panels to the charge controller. The charge controller's user interface will display the voltages of each. While connected, the solar panels will be toggled from exposure to sunlight and darkness. Through observation, the charge controller should show that the solar panel voltage is displayed to match the voltage of the battery. The battery voltage will also be displayed on the charge controller, and should display an average of 12 [V] when connected.

### **Task 3.1-User Interface:**

Our user interface consists of a Liquid Crystal Display (LCD) and three buttons that have certain functions like scrolling up/down and select. We will assemble the buttons and LCD and attempt printing various messages to the LCD and ensure that our buttons allow us to scroll up, scroll down, and make a selection. We will ensure that when a selection of the saturation level is made in the interface that it is properly adjusted within the microcontroller and the saturation points are adjusted as well.

### **Task 3-Assembled System:**

Once the system is fully assembled, the team will first verify the functionality of the ball valve by activating the system while the moisture sensor is dry. This should prompt the ball valve to open, which would allow water to flow through and irrigate the crops. The moisture sensor would then be dipped in water, shutting off power to the ball valve and closing it. For a final test, the design will be installed at a member's house and tested in a 4 [ft] x 10 [ft] garden.

### **Schedule:**

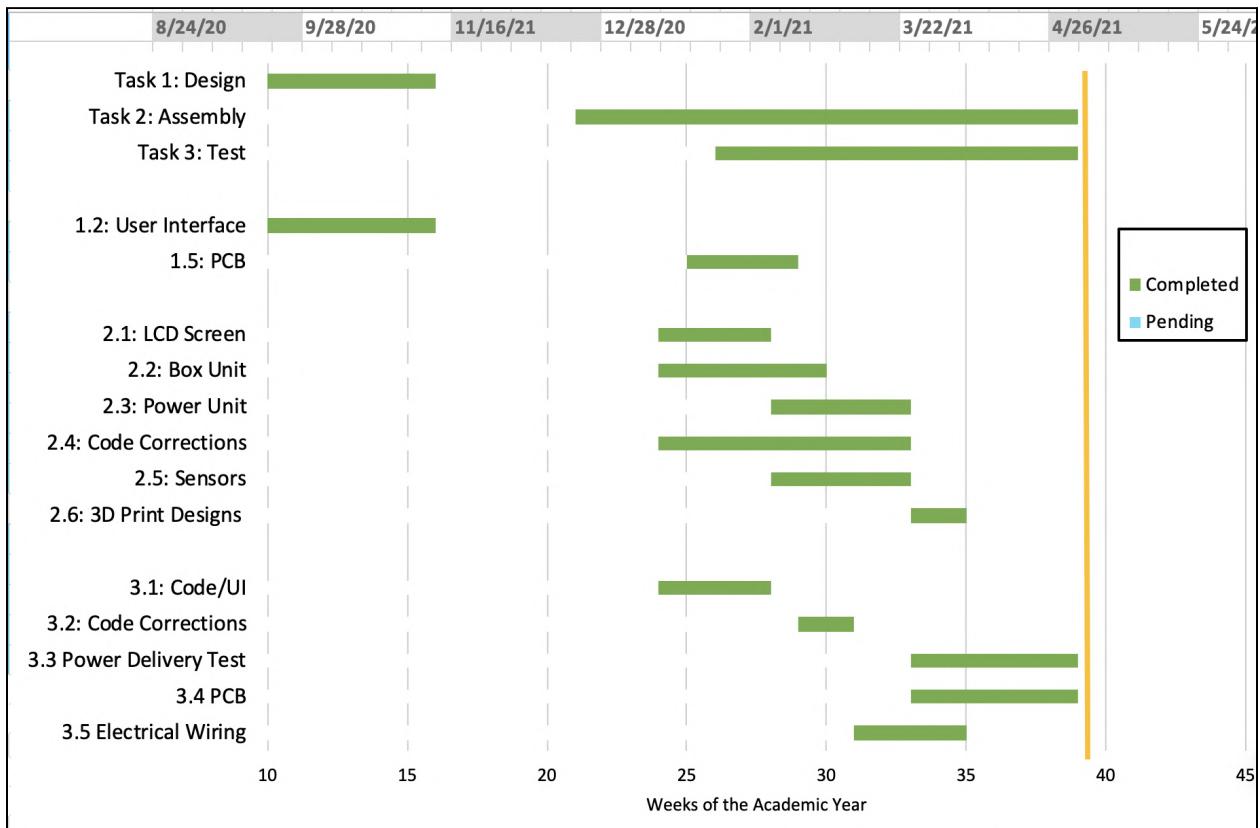
Table 2 depicts the intended schedule based on the Fall 2020 semester predictions. However, due to some circumstances our project schedule has changed from our first proposed schedule. Part of the code needed to test proper communication with the sensors is currently not yielding the proper output and therefore adjustments need to be made. Furthermore, because more sensors are

going to be added prior than intended, the team will need to reassess how the wiring that will lead connect back to the main enclosure. Although the team has been delayed in some parts, progress and an early start has been made in other areas. These areas include the design of the PCB board, code to output the system's message and warning to the users, and early developments on the user manual. Together, these changes are reflected in the schedule, which can be seen in Figure 4.

**Table 2:** Proposed schedule during Fall 2020.

Assignment	Due Date	Assigned Member
<b>Phase 1: Research</b>		
Analysis Written Report	September 18th, 2020	All
Oral Reort	October 1st, 2020	All
Individual Written Report	October 9th, 2020	All
Previous Team Meeting	October 15th, 2020	All
Sensor Research	October 16th, 2020	Diego
Power Unit Research	October 16th, 2020	Vedant & Josh
Protective Casing Research	October 16th, 2020	Vedant
Microcontroller Research	October 16th, 2020	Victor
Budget Proposal	October 22nd, 2020	All
Design Written Report	October 23rd, 2020	All
Contact Nicaruguan Team/Sponsors	October 26th, 2020	Diego
Duo Presentation (Team 1)	October 29th, 2020	Vedant & Victor
Individual Evaluation Report	November 6th, 2020	All
Duo Presentation (Team 2)	November 19th, 2020	Victor and Diego
Final Individual Written Report	November 20th, 2020	All
<b>Phase 2: Building and Testing</b>		
Order Parts	January 15th, 2021	All
Minimal User Interfance	January 29th, 2021	Victor
Build/Test Solar Panel System	February 1st, 2021	Vedant & Josh
Complete User Interface	February 26th, 2021	Victor
Complete Polycrop Code	February 17th, 2021	Victor
Integrated Power Supply	March 10th, 2021	Josh and Vedant
Print 3D Casings	March 10th, 2021	Vedant
Complete System Assembly	March 20th, 2021	All
<b>Phase 3: Documentation &amp; Shipping</b>		
3D Models	March 21st, 2021	Vedant
Write Assembly Manual (English and Spanish)	March 24th, 2021	Diego & Victor
Complete Propotype	March 31st, 2021	All
Ship tp Nicaragua	April 1st, 2021	Diego

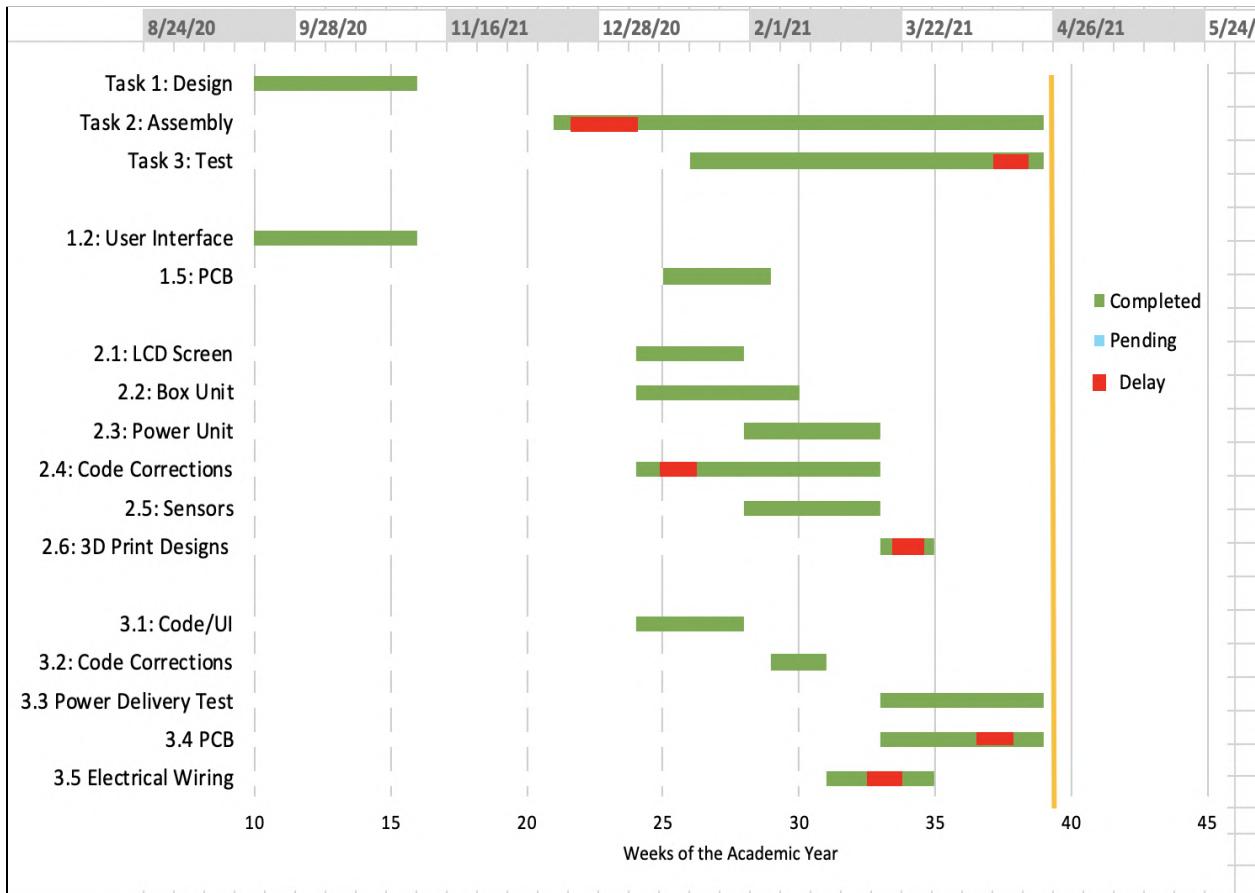
The final schedule of the project can best be represented by the Gantt Chart below in Figure 4(a). This chart is representative of the outlined goals of this project to deliver a functioning system with the ability to irrigate different types of crops. During the 36 weeks allotted for the completion of the SDIS, the team divided the overall goals into three distinct phases - Design (Phase I), Assembly (Phase II), and Test (Phase III). Each phase included subgoals for the overall project that eased the progress of completion. During the first academic term this project took place, the crux of the SDIS relied on designing and finding proper components that satisfied the client's needs. During the second academic semester, the attention to the project shifted to completing the assembly and testing phase based on the designs and measurements derived from Phase I.



**Figure 4(a):** Schedule with completion and progress of tasks.

During the completion of our project our schedule faced some significant setbacks. One setback that our team faced was the Winter Storm Uri in the early weeks of February. Due to the storm our operations were limited since all team members were without power and unable to move to a

meeting location. The result of this setback was a one week delay to our proposed schedule. Furthermore, our assembly phase started, there were multiple 3D print failures. The dimensions of our proposed designs were too big to handle at once, so many reiterations of 3D printing were needed. Under this same issue, the Senior Design Lab at the Cullen College of Engineering was only open for six hours while our designs required over eight hours of printing. This led the team to outsource 3D printing services and cost for the printing material. The assembly phase was delayed by another week as a consequence. In regards to the Phase III, testing of the overall system, we encountered a failure on the PCB as the soldering pads melted off and our proposed completion of the system was further delayed by another week. Lastly, while evaluating the microprocessor code, a failure of the system performance was encountered. The edits made to code allowed for proper operation of the system but delayed testing and resulted in no time to perform a field test, however. The schedule with the aforementioned setbacks are shown in Figure 4(b) below.



**Figure 4(b):** Final schedule with project delays.

Regardless of the setbacks that our team experienced, the overall project was successfully completed. However, as a consequence of the accumulation of setbacks, and under guidance of our faculty advisor, the team made the decision to not provide a video manual for future users. Instead, the team will include an operation guide as a form of Appendix found at the end of this report. In contrast to what we had previously anticipated, the complete system will no longer be shipped to Nicaragua but rather inherited by the next Senior Design Team working on the continuation of the SDIS and to the Cullen College of Engineering.

## **Task Updates:**

With the time that has passed, the control system for our device is complete. The control system gives the ball valve an opportunity to open every two hours if conditions are met. We monitor six soil sensors and once four of the sensors are detected as being dry then we queue our valve waiting for the next two hour cycle to begin irrigation. Once our sensors have been satisfied the valve is closed and irrigation is stopped. Now that all software has been completed we can finish assembly and begin testing our full system soon. In the testing phase we will determine our tolerances for measurements and check for any bugs. We will also have the opportunity to adjust our saturation levels should we determine that there needs to be any adjustments.

## **Task 2.3: Power Unit**

The power delivery system has been assembled and wired correctly. The solar panels were set out in cloudy weather conditions and read an output of 19 [V], which is more than sufficient for our system, as it needs to charge a 12 [V] battery. The solar panels, battery, and charge controller were then connected together to test, and the charge controller utilized voltage regulation to match the voltage from the solar panels to the voltage of the battery. This verifies that the solar panels were properly charging the batteries, through the charge controller.

## **Task 2.6: 3D Designs**

Our team has completed the 3D design for the Battery Platform. Our team decided to contain the battery in a custom 3D printed platform so that we can hold the battery in place. It will also allow mounting the battery to the back panel of the main container. A lip was added to keep the battery from slipping, and zip-ties will hold the battery in place. Using TinkerCad, the Battery Platform design was completed. The 3D model was then 3D printed using PLA plastics. Our resulting printed platform has a slight deformity, but it still functions correctly.

Our team also decided to make a Shield to hold our delicate, interactive components in place — charge controller, LCD screen, buttons, trigger, and potentiometer. We needed this Shield as it also helps make the layout of our interactive components for the user. Further, this Shield will cover our PCB while also hiding the wiring so that it remains protected from outside influences. The 3D designing aspect of this build is done, and our team is in the process of printing the design.

## **Task 3.4: PCB Testing**

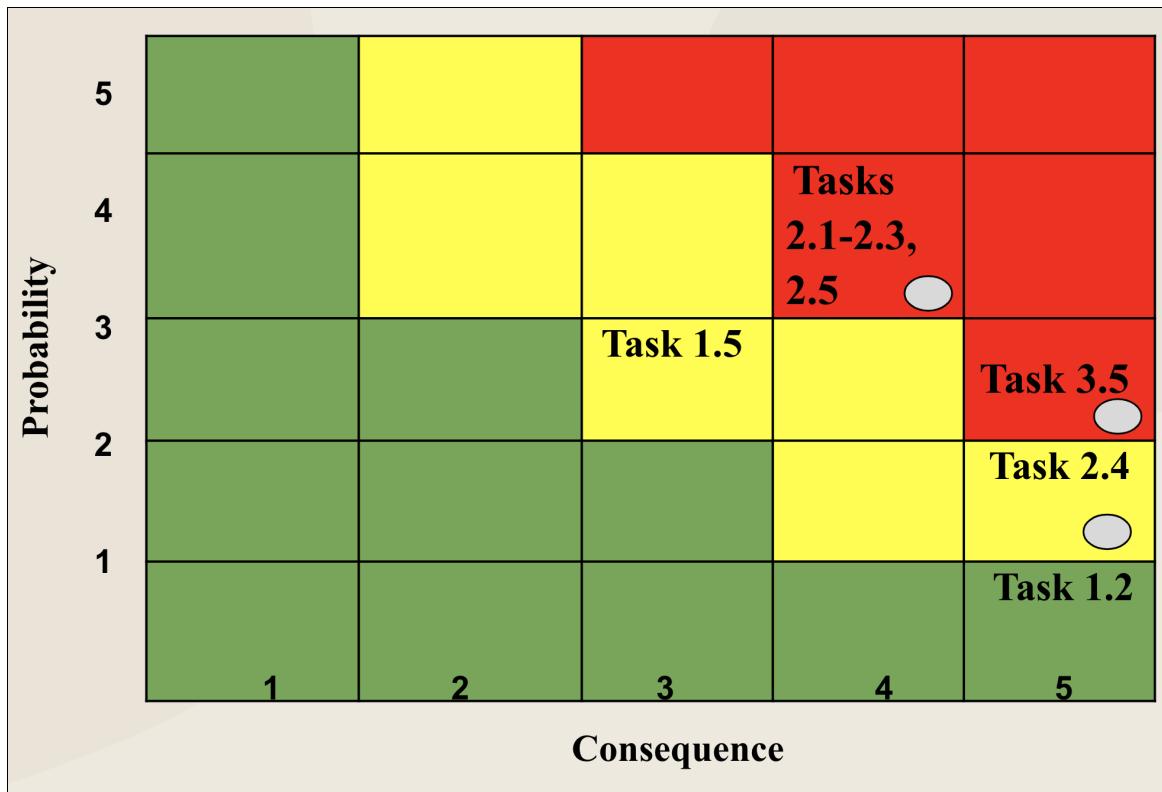
In order to test the integrity of our PCB, our team conducted a test to check the terminals. By doing this, our team could decide if every component in the PCB was connected properly. Using the ohmmeter at the electronics lab, we checked if the measured resistance was approximately resources. If the measured resistance was approximately 0, we know that the connection is clear and ready to use. Everything is perfectly connected, and is ready for use.

## **Risk Matrix and Mitigation Plans**

Below are Figures depicting our risk matrix and our mitigation plans. On our risk matrix we can see Tasks 1.2, 1.5, 2.1-2.3, 2.4, 2.5, and 3.4. Task 1.2: User Interface Design, because of where we are in the project design the likelihood of this having a malfunction is low as it has been tested with test code and seems to be working. Though if we were unable to get a UI designed properly it would mean that we would be losing lots of the additions we aimed to add like the ability to set saturation modes and display system info. Task 1.5: PCB Design, is a medium risk

task because if we were unable to design and create our PCB then we would lose the goal of making more permanent and secure connections but we would be able to work around this by using protoboard. Tasks 2.1-2.3:LCD, Box Unit, and Sensor Assembly, are high risk tasks as if we are unable to connect these systems together then we would lose most of our system functionality. The likelihood that this occurs is not too high but if it does it would be devastating to our project. Task 2.4: Code Corrections: this task is a medium risk task as if an error does occur our system would not operate properly as this system manages the sensor readings, UI, and ball valve. The probability is low that this occurs as we are building off the code from the previous group which had an operational system. Furthermore, Task 2.5: Sensors is identified as a risk since obtaining wrong results from testing the moisture sensors will cause improper irrigation when is not needed. Failure in sensor testing is categorized as a high and probable risk because it will show that the main purpose of the system to irrigate on soil moisture readings has not been met. To overcome this obstacle, we will test all sensors independently with different level of water content in the soil to ensure that the output generated from the sensors are reflective if there is a need to irrigate or not. Task 3.4: Electrical Wiring:, is a high risk task as this task is essential to deliver power to our system and without power nothing would operate, the likelihood of this occurring is low as we are not completely redesigning the power system but rather improving it.

All of the outlined tasks were completed and thus rendered the completion of our project. The risks outlined on our risk matrix below were successfully completed even though some caused delays with respect to the proposed timeline of our projects. Upon proceeding to the assembly and testing phase, our team determined that the three risks that posed the biggest threat to the completion of the project were Task 3.5 (Electrical Wiring), Task 2.1 - 2.5 (System Assembly), Task 2.4 (Code Edits). All of these risks were successfully dealt with and the SDIS was successful. The risk matrix is shown in Figure 5.



**Figure 5:** Risk matrix for project design tasks



**Figure 6:** Risk matrix legend

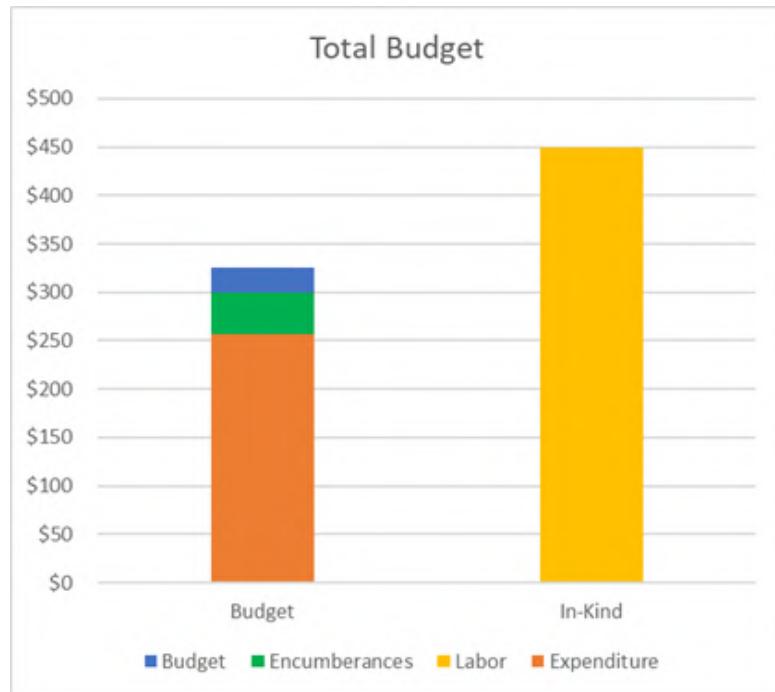
Table 3 below highlights the mitigation plans for the tasks specified in the Risk Matrix.

**Table 3:** Mitigation plan for potential risks

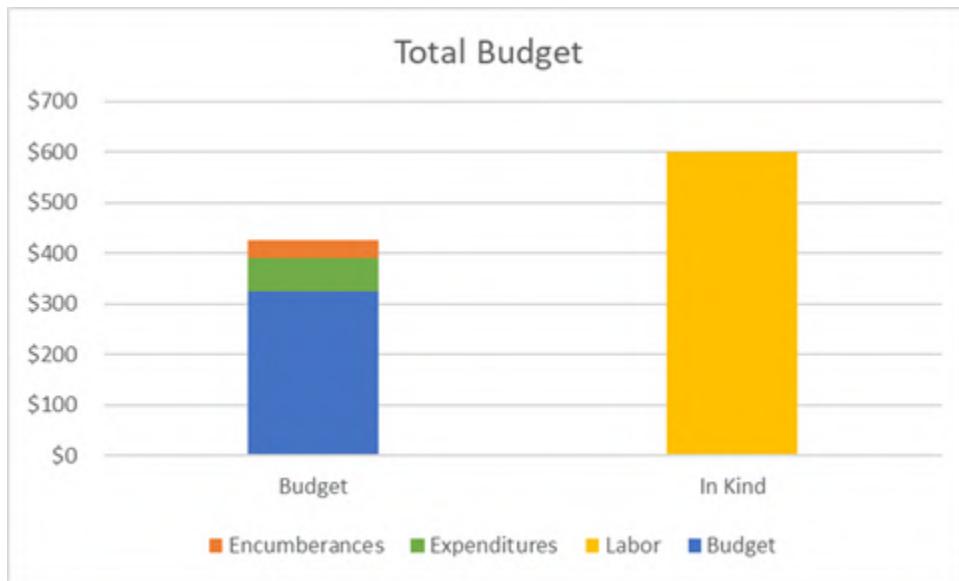
Task	Identified Risk(s)	Mitigation Plan
#2.1-2.3, 2.5 System Assembly	Error made during assembly of the system	<ul style="list-style-type: none"><li>Handle Components Carefully</li><li>Store in a safe area</li></ul>
#2.4 Original Code Edits	Improper System Operation	<ul style="list-style-type: none"><li>Redesign system operation</li><li>Reduce number of sensors</li><li>Remove time constraints on system</li></ul>
#1.5 PCB Design	Improper Connections made during design phase	<ul style="list-style-type: none"><li>Create Connections on piece of perf board</li><li>Connections will not be as robust, but will be more reliable than jumper cables</li></ul>
#3.4 Electrical Wiring	Faulty wiring connections made during assembly phase	<ul style="list-style-type: none"><li>Follow PCB Schematic</li><li>Troubleshoot system step by step to find errors</li></ul>

## **Budget:**

For this project, we've set the budget to be \$325. At the conclusion of this project we have spent \$425. Our project expenditures exceed the projected budget because some key components needed to be purchased in bulk - for instance, the PCB. In addition, with the unexpected changes to our 3D design, the team had to account for this additional cost. Observing Figure 7(a) below, our team has gone over budget with our project, and so future expenditures will be donated through our team members. The final budget with all expenditures is shown in Figure 7(b). Our faculty advisor, Dr. Trombetta, has informed us that a proposal has been issued to IEEE to get more funding in support of future projects with Bionica, and if approved, would allow us to delegate more funding to improve the infrastructure of our design.



**Figure 7(a): Original Budget**



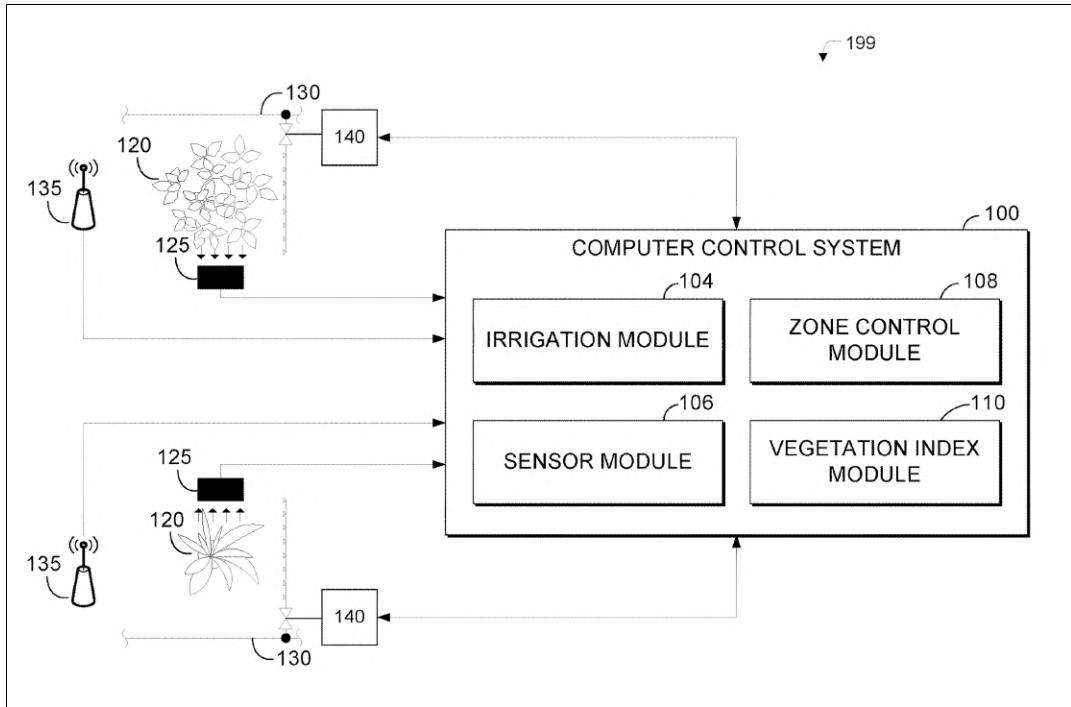
**Figure 7(b): Current Expenditures and Total Budget**

## **Patents and Standards:**

### **Patents:**

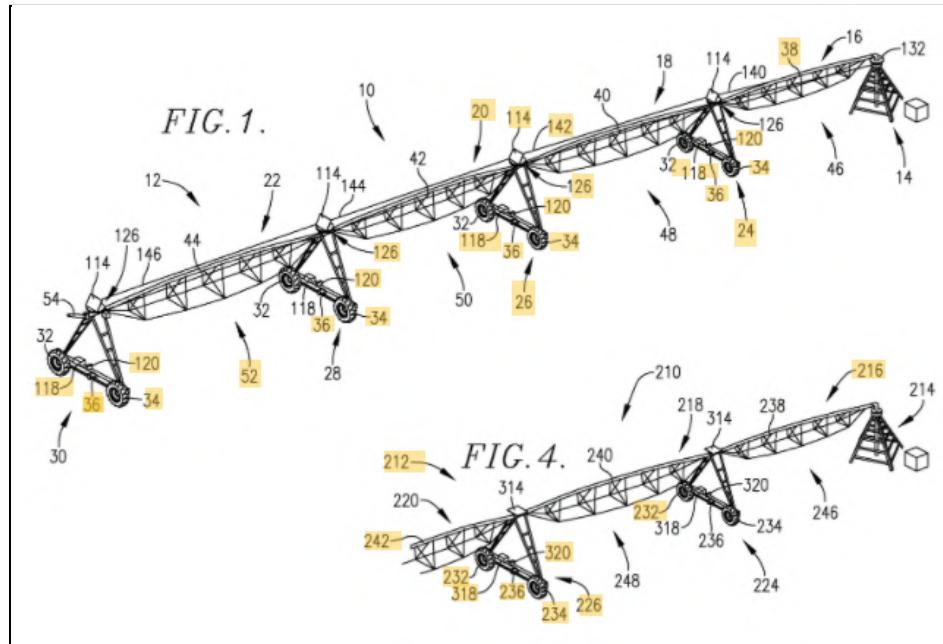
To ensure no intellectual property infringement with systems or processes intended to operate in a similar fashion to our proposed SDIS, the team evaluated the following patents. After careful consideration, a conclusion was derived that the system is not in violation of any patent law.

- S. . Bermudez Rodriguez, H. F. Hamann, L. Klein, and F. J. Marianno, “Automated Irrigation Control System,” 10 , 241 , 488 B2, 26-Mar-2019.
  - This patent by Bermudez, *et. al* seeks to use smart drip irrigation by taking into consideration the contrast of light for a designated crop. The system uses a photosensitive sensor to detect the contrast of light and send information to a central computer. After said data is collected the system then examines the sensor’s output and in addition to locally available weather information determines if the irrigation process should begin. Given our SDIS does not operate with either photosensitive sensors or a central computer, our proposed design is not in violation of Bermudez, *et. al* intellectual property.



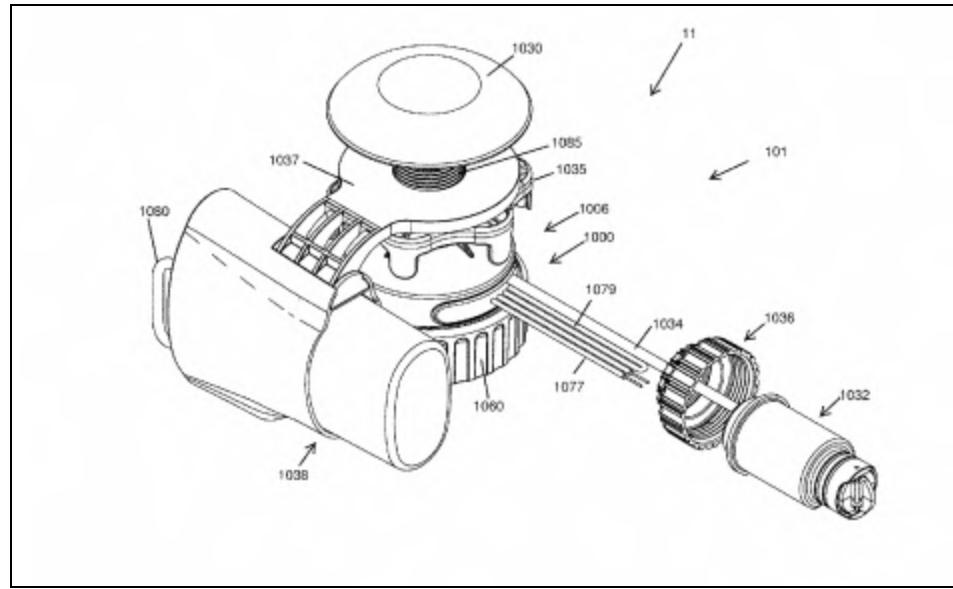
**Figure 2(a):** Irrigation system using photosensitivity sensors. [5]

- Richard W. Parod, C. H. M. (n.d.). “Solar Powered Irrigation Machine”. *Patent No. 8517289B2. 9-28-2007*
  - This patent is for a mobile irrigation machine that waters crops as it slowly moves across a field. It also utilizes solar tracking to determine the times at which the machine is to turn on and begin irrigating. The approach for our project differs from this patent since we use our solar panels strictly for power, not for solar tracking. Furthermore, while our design uses a machine, it is valve operated rather than motor operated.



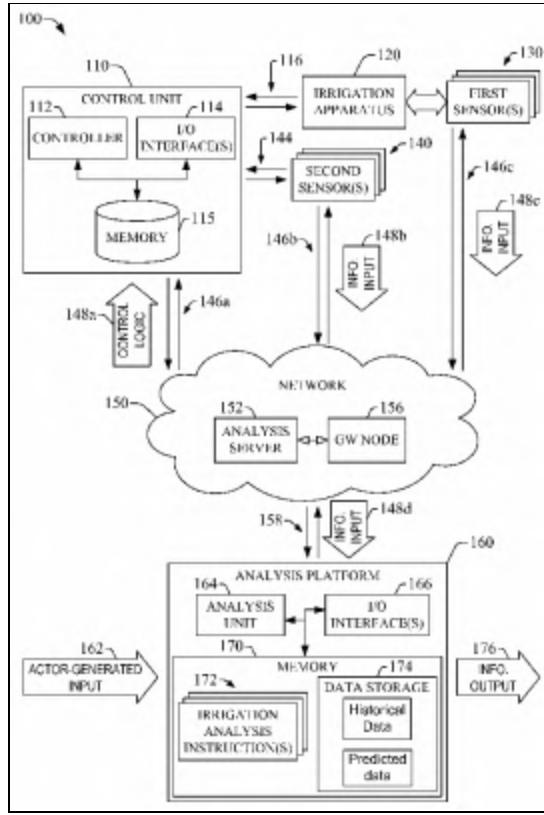
**Figure2(b):** Solar Powered Irrigation Machine [6]

- Paul, M. F. (2020). “WATERTIGHT ELECTRICAL COMPARTMENT FOR USE IN IRRIGATION DEVICES AND METHODS OF USE,” 10,965,109, 30-Mar-2021.
    - This patent covers the creation of a water tight compartment to be used for an irrigation system. The purpose of this compartment is to have a watertight environment to house the various components that go into making an irrigation system. This system is designed to avoid the need to have AC power wiring to be run into it as the electronics held within are assumed to be wireless and have battery power within the enclosure. Although this patent is designed to improve automatic irrigation systems, we are not designing or building our own enclosure.
    - This patent also makes claims on certain methods used to make the compartment watertight such as using O-rings and sealing caps in a certain manner. The patent also gives descriptions of how each compartment of the enclosure can be used such as the description of compartment 30 being able to be used to house wireless flow sensors.



**Figure 2(c): Patent #10,965,109 Drawing [7]**

- D. Lankford, “Crop-specific Automated Irrigation and Nutrient Management,” 10,512,226 B2, 24-Dec-2019.
  - In this patented system the user is responsible for initiating the irrigation process after the system alerts the user of detected dry soil. In addition to user data, the system operates by taking into account geographical location and weather data to further determine if irrigation is needed at that time. Unlike Lankford’s system, our proposed design focuses on a completely automated irrigation based on soil moisture data and excluding local weather data. After evaluation, our proposed SDIS is not in violation of Lankford’s patent.



**Figure 2(d):** Patent #10,512,226 B2 Drawing [8]

## Standards:

In compliance with industry and professional boards standards, we have consulted the following standards to be compliant with performance and data standards set by recognized institutions.

The consulted standards for this system are as follows:

- IEEE Standard for Sensor Performance Parameter Definitions [9]
  - This IEEE standard sets the parameters and limitations for sensor performance. This standard assures the performance of humidity and moisture sensors across different disciplines. Furthermore, the aforementioned standard is specifically constructed for systems with an input/output (I/O) system. The proposed SDIS meets these standards as the capacitive sensors used operate within the minimum and maximum parameters of said standard.

- IPC J-STD-001, “Requirements for Soldered Electrical and Electronic Assemblies.” [10]
  - Also known as the J-Standard, it covered the entire soldering process and the working environment to be performed in. The standard is important, particularly the guidelines for soldering through plated holes, since this is how connections are made on the PCB. It also details the effects of electrostatic discharge, which is crucial to understand since if we are not properly grounded before working on the electronic components in the PCB, could potentially damage them, rendering them inoperable.
- Recommended C Style and Coding Guidelines - Texas Instruments [11]
  - Standards created by Texas Instruments in 2013 for their C coding standards. These guidelines describe their preferred use of comments, formatting, and certain structuring of functions like “if” and “else” statements.
  - This document gives templates for function headers, file headers, and more to be able to give good descriptions on what your code is doing and how to use it. This is necessary as if the current code and data were to be passed onto another team they would be able to easily continue development of the code rather than having to spend time figuring out what each function or block of code does and test its application.
  - These standards reduce time used on unnecessary debugging and testing and allow for code to be easily updated since the programmer would know what to look for and where and make things much easier.

- IPC-2152 —Standard for PCB Trace Width [12]
  - This IPC Certified standard dictates what the Trace Width should be in regards to a PCB. Trace width is the width assigned for the connections of the PCB. Different operations require different currents, and the net width needs to accommodate these currents. For most smaller PCBs, like the one we are using, it is recommended to provide a 0.010” width. However, if there is more than .3A through the connection, the trace width should be larger. Most modern 3D CAD Softwares take this into consideration automatically, such as the one used for the PCB design.

## **Constraints:**

### **Lack of Feedback Data:**

The previous iteration of this project was intended to be fully assembled by an engineering team from the Universidad Nacional Agraria no later than October, 2020. However, due to the country experiencing two major hurricanes all efforts to assemble the system came to a halt until further notice. The new expected timeline for the assembly of the monocrop system is scheduled for February, 2021. As a result, no data on the previous system is available at this time thus resulting in our team being limited to the number of adjustments that can be carried out to further ensure the lifetime of the final system.

### **Unable to Establish Contact with UNA:**

Through the previous months, our team has made several attempts to establish a proper communication channel with BioNica. However, all efforts have yielded unsuccessful results. Our faculty advisor has become our main point of communication between our sponsors and the team. We remain optimistic that proper communication will occur in the upcoming days to finalize the delivery timeline of this project.

## **Future Works:**

As our Smart Drip Irrigation System depends on solar panels to power the entire system, optimizing this power would benefit anyone who is using our system. As a result, a future project could entail using a microcontroller to orient panels towards the sun. While our system already does power our system sufficiently, it is not optimized to get the most solar energy possible. While the Nicaraguan climate is mostly sunny and arid throughout the year, our system would suffer energy loss in more cloudy areas. Using a microcontroller and a light-sensor, a future team could track the solar panels to where the sun is most optimally poised for energy delivery.

The biggest criticism the first iteration of the Smart Drip Irrigation system received was that the sensor corroded, causing the entire system to fail. While our system does correct this by using six, new capacitive sensors, our team is not sure if our sensors will resist corrosion as testing for this was not done. However, for the Smart Drip Irrigation System to function reliably, our sensors need to be corrosion free. A future team could invest more of their time and budget towards more expensive sensors that are more corrosion free. A future team could also design a budget, corrosion-free sensor so that their design could be implemented to future projects. While testing our system, we found that our moisture sensors cannot operate properly with the “High” saturation range we have set because the operating range is too narrow for our system. To fix this, we recommend future teams purchase sensors with a higher sensitivity range that would allow the parameters of our microprocessor to work. Another variable our team did not consider was the issue of water pressure. The team in Nicaragua will use a large tank that will store 10,000 gallons of water. The tank is also 1 [m] high, with the water’s height varying from about 1.2 - 3.7 [m]. One concern about this is that height and water pressure are inversely proportional. In other words, our system will deliver faster when the tank is full, but it will slowly become slower as the tank empties. This is a concern since BioNica informed us that they would like to deliver 160 [L/hr], but our system can not guarantee it. Our team was informed about this request by BioNica before the last couple days of completing this project, so our team could not tackle this issue. However, a future team could design the Smart Drip Irrigation system so that it would deliver water at a reliable rate so that the system can reliably deliver water at a rate of 160 [L/hr].

## **Conclusion:**

We, Team 17, the Polycrop Smart-Drip Irrigation team, created a Polycrop Smart Drip Irrigation system. We accomplished creating an appealing design with important upgrades that improve on the previous iterations of this project. Our new system has an interactive UI that will allow the user to interact with our system, and has a PCB to make the system's wiring more clean and robust. Our team has finished our system on schedule, although we had to go slightly above the given budget. This was not an issue as our sponsor was willing to give us a budget increase. Our teams also arranged lab testing so that we could test our system and have arranged multiple tests to ensure our system works. Due to the current COVID-19 pandemic, our team will have issues with meeting up in person, but we are successfully getting important work done with frequent Microsoft Team calls. However, our system works perfectly, in spite of the restrictions placed on us. With our combined teamwork, we, Team 17, are excited to present the smart-drip irrigation system and to help improve Nicaraguan farmers' lives.

## **Sources:**

- [1] *Worldbank.org*. [Online]. Available:  
<http://documents1.worldbank.org/curated/en/532131485440242670/pdf/102989-WP-P152101-Box394848B-OUO-9.pdf>. [Accessed: 08-Dec-2020].
- [2] “Learn facts about Nicaragua, Poverty, and Development,” *Opportunity.org*. [Online]. Available: <https://opportunity.org/our-impact/where-we-work/nicaragua-facts-about-poverty>. [Accessed: 08-Dec-2020].
- [3] “Overview,” *Worldbank.org*. [Online]. Available:  
<https://www.worldbank.org/en/country/nicaragua/overview>. [Accessed: 08-Dec-2020].
- [4] Maloney, I., Woelfel, C., Lawal, A. and Serrano, C., 2020. SMART DRIP IRRIGATION.
- [5] S. . Bermudez Rodriguez, H. F. . Hamann, L. Klein, and F. J. . Marianno, “Automated Irrigation Control System,” 10 , 241 , 488 B2, 26-Mar-2019.
- [6] “US8517289B2 - Solar powered irrigation machine,” *Google Patents*. [Online]. Available: <https://patents.google.com/patent/US8517289B2/en>. [Accessed: 01-Apr-2021].
- [7] Paul, M. F. (2020). “WATERTIGHT ELECTRICAL COMPARTMENT FOR USE IN IRRIGATION DEVICES AND METHODS OF USE,” 10,965,109, 30-Mar-2021.
- [8] D. Lankford, “CROP-SPECIFIC AUTOMATED IRRIGATION AND NUTRIENT MANAGEMENT,” ) *World Intellectual Property Organization*, Jan. 2013.
- [9] IEEE, “IEEE Standard for Sensor Performance Parameter Definitions,” IEEE, IEEE Standards, Jan. 2018.
- [10] S. Bishop, “What is IPC-A-610 J-STD-001 For Electronic Assembly?,” *Levison Enterprises Blog*. [Online]. Available: <https://blog.levisonenterprises.com/ipc-j-std-001>. [Accessed: 01-Apr-2021].

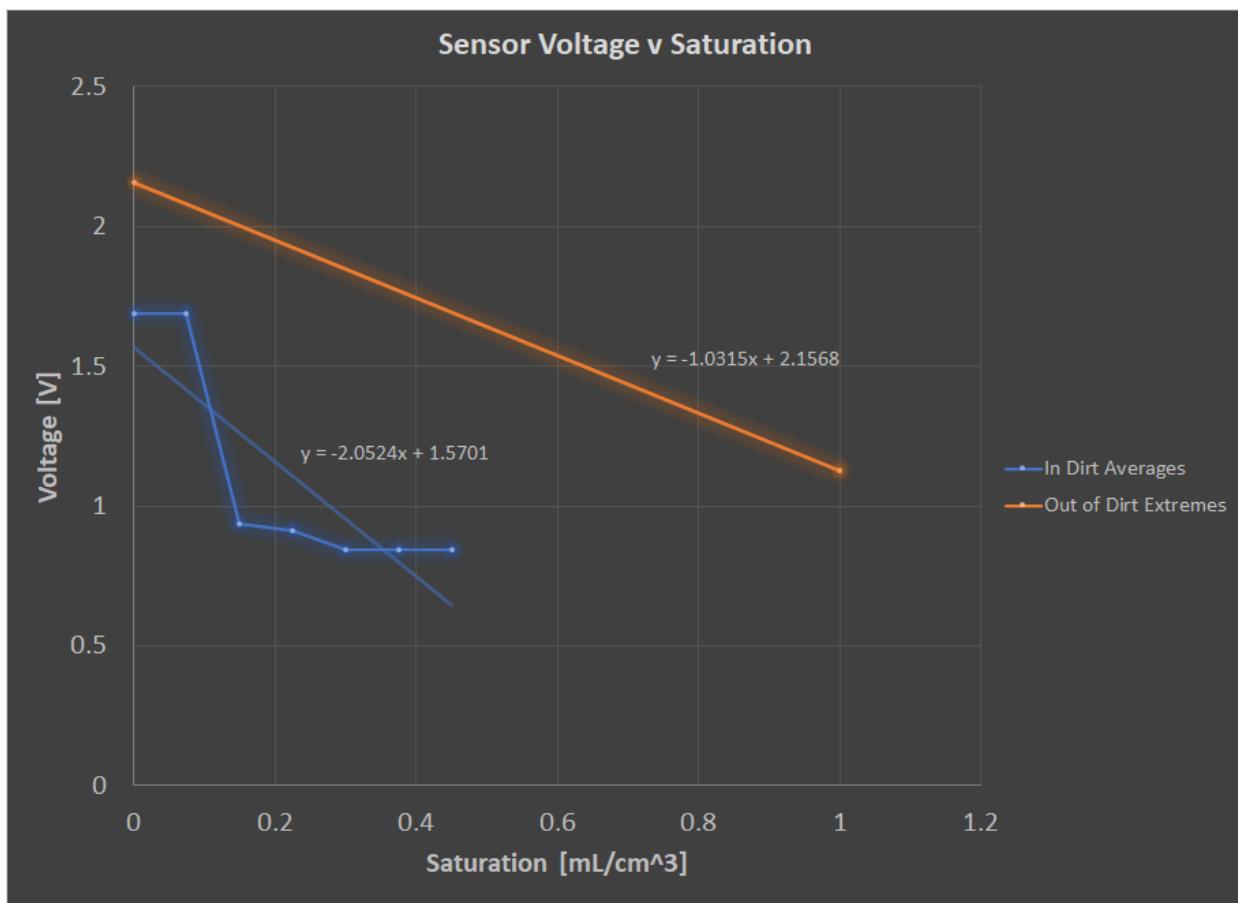
- [11] “Recommended C Style and Coding Guidelines.” [Online]. Available: [http://software-dl.ti.com/hercules/hercules\\_public\\_sw/HerculesMCU\\_C\\_CodingGuidelines.pdf](http://software-dl.ti.com/hercules/hercules_public_sw/HerculesMCU_C_CodingGuidelines.pdf). [Accessed: 31-Mar-2021].
- [12] J. Olson, “IPC-2152 Determining Current Carrying Capacity,” *IPC Standards*, May 2018.

## Appendix A:

### Design Specifications

**Table AA:** Critical Voltage Thresholds

Saturation Mode	High	Medium	Low	Manual Trigger
Irrigation Needed Voltage	N/A	1.21 [V]	1.61 [V]	<b>None</b>
Satisfied Voltage	N/A	0.97 [V]	1.36 [V]	<b>None</b>
Tolerance	+-. 0.1 [V]	+-. 0.1 [V]	+-. 0.1 [V]	<b>None</b>



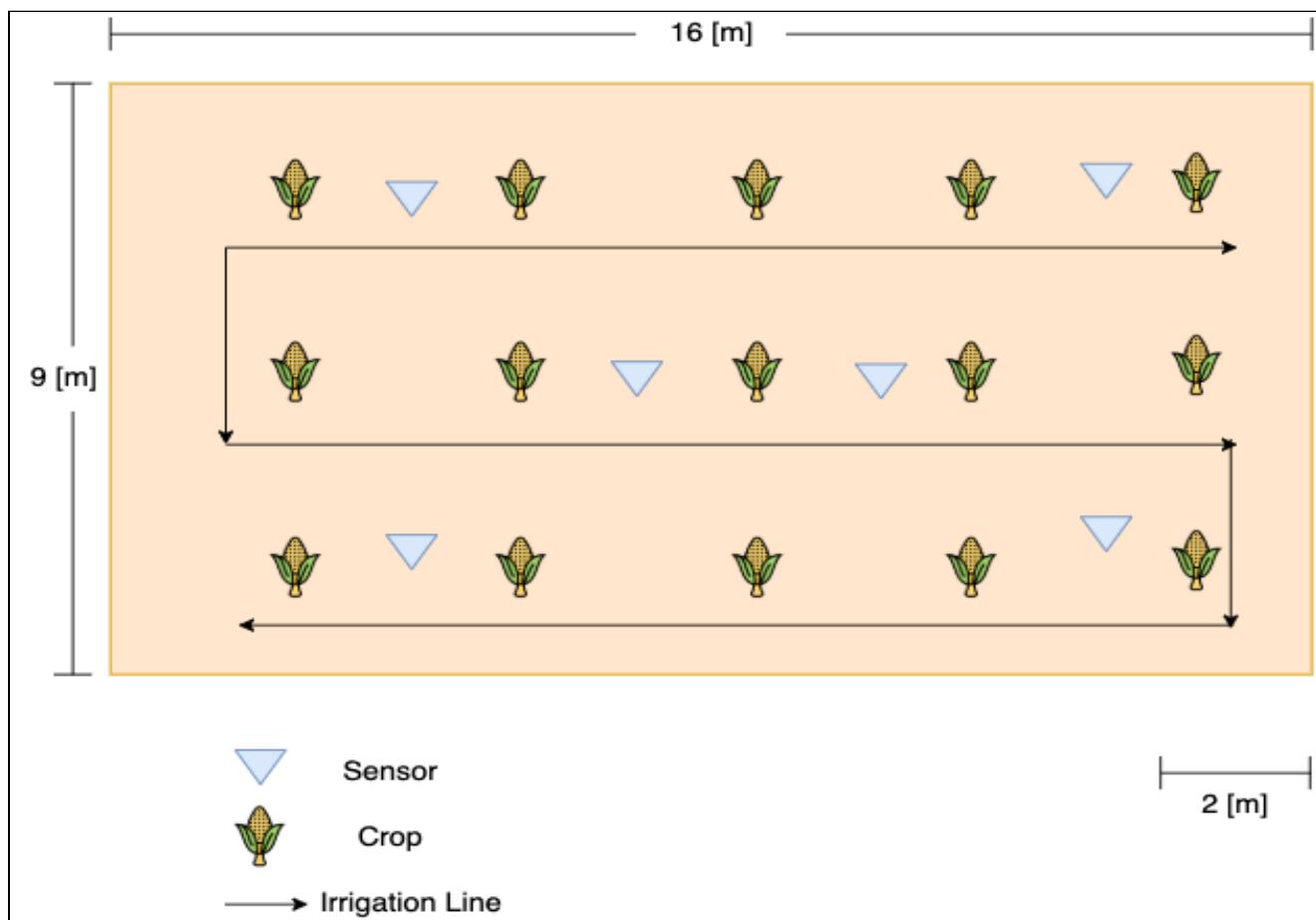
**Figure AA:** Sensor Operation Range

## **Improvements From Previous Group**

**Table AB:** Improvements From Previous Group

<b>Previous Group</b>	<b>Our Group</b>
Single Crop Design	Polycrop Design
One Moisture Sensor	Six Moisture Sensors
Loose Wiring	Printed Circuit Board
Used LEDs to Communicate	Using LCD Screen to Communicate
Three Solar Panels	Four Solar Panels
IP65 Container	IP67 Container
Normal User Manual	Video User Manual

## System Diagram



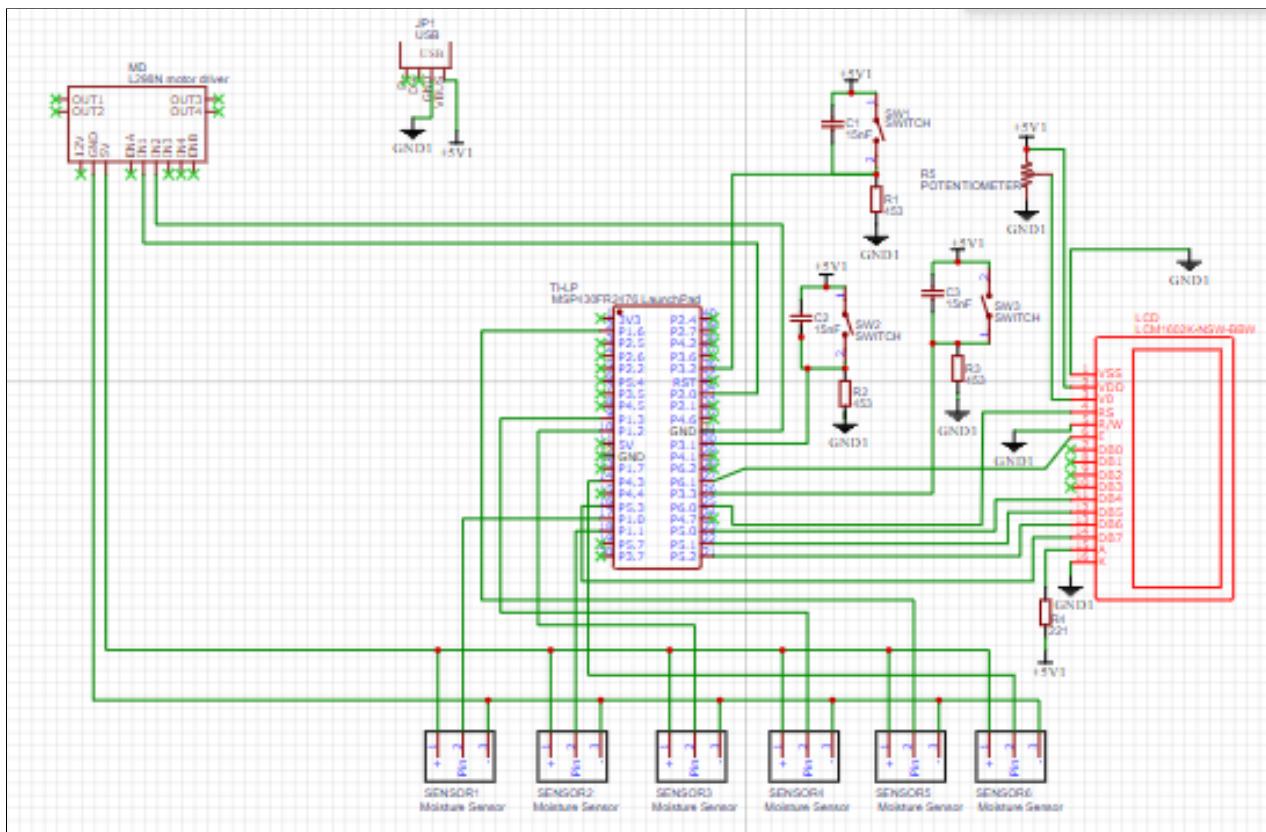
**Figure AB:** Diagram of Sensor Placements

## **Plot of Land for System**

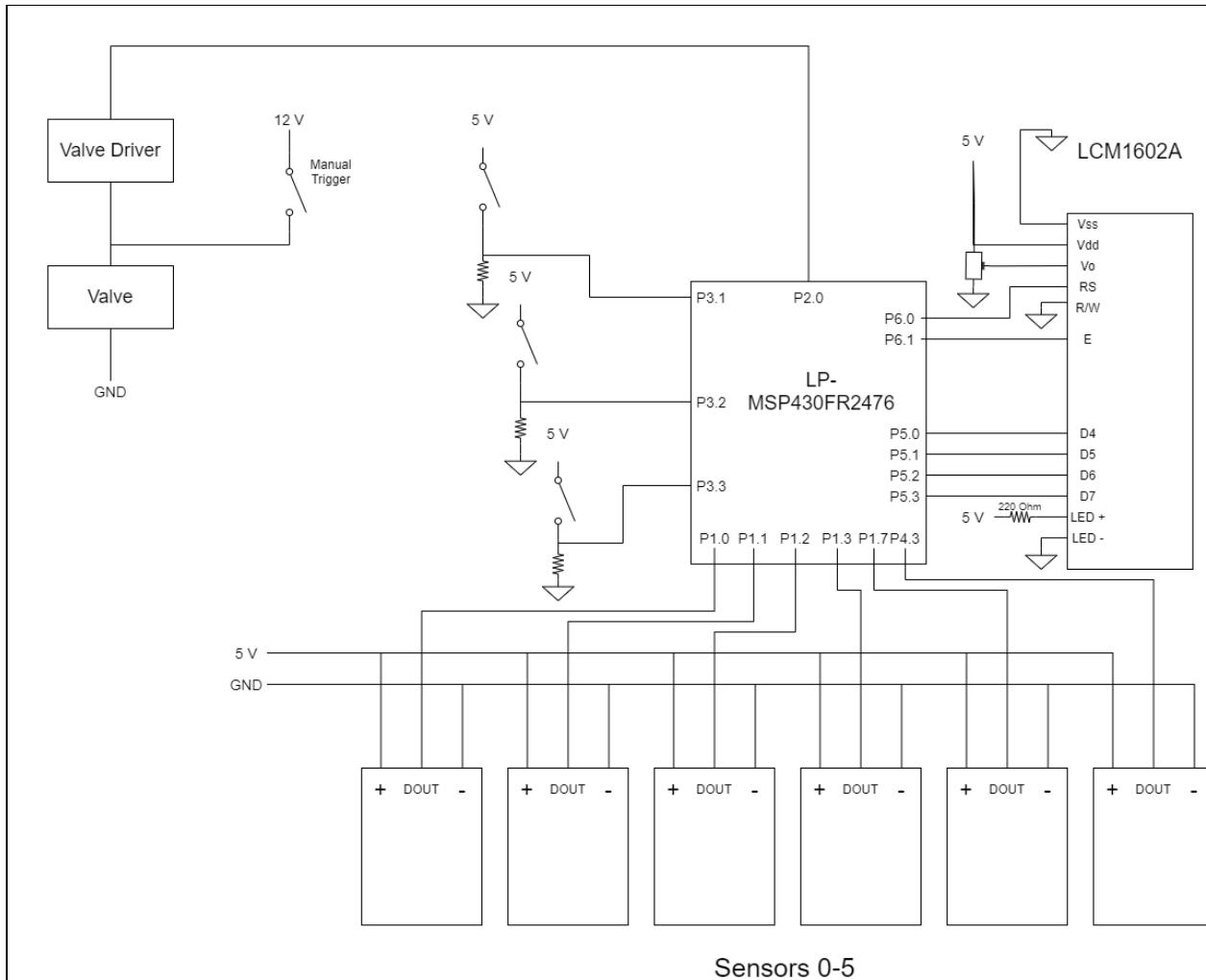


**Figure AC:** Plot of land (Nicaragua) for SDIS System.

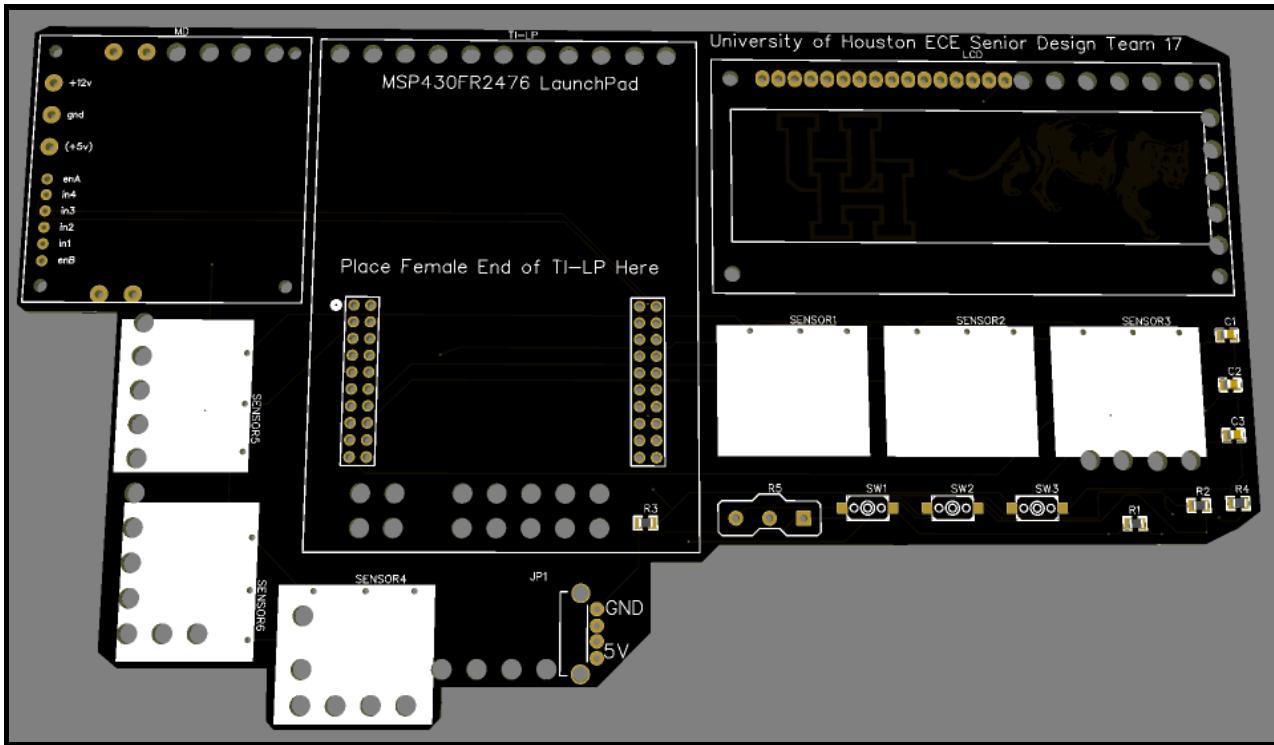
## PCB Design



**Figure AC: PCB Design Schematic**

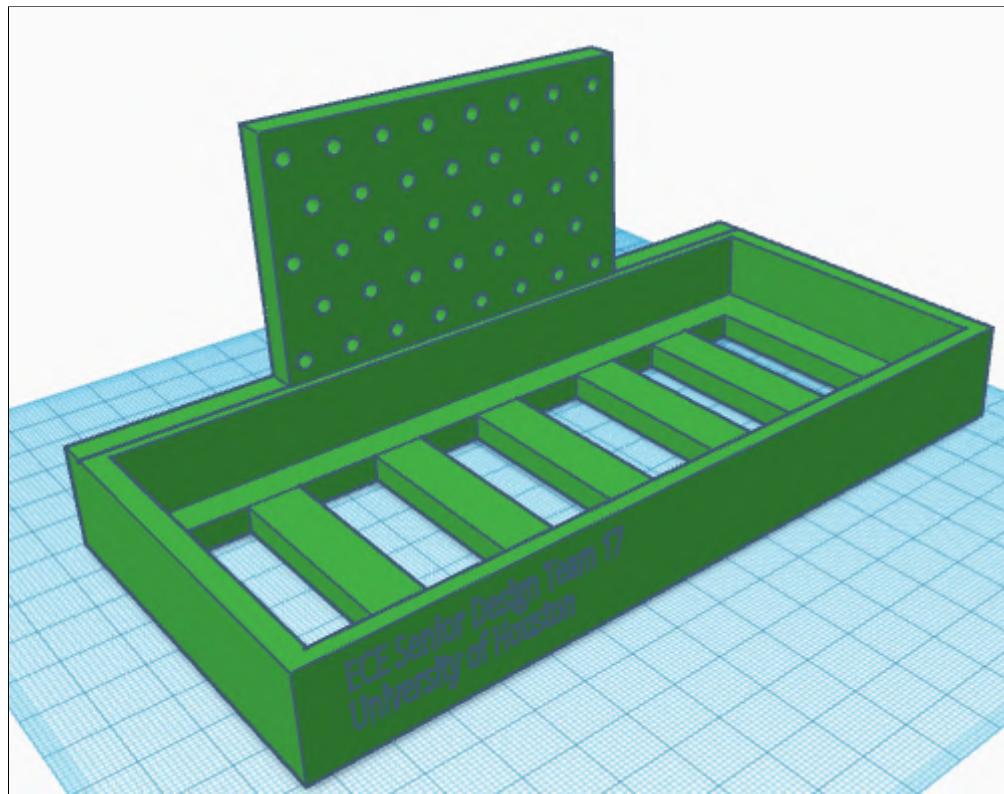


**Figure AD:** Block Diagram of Controls System



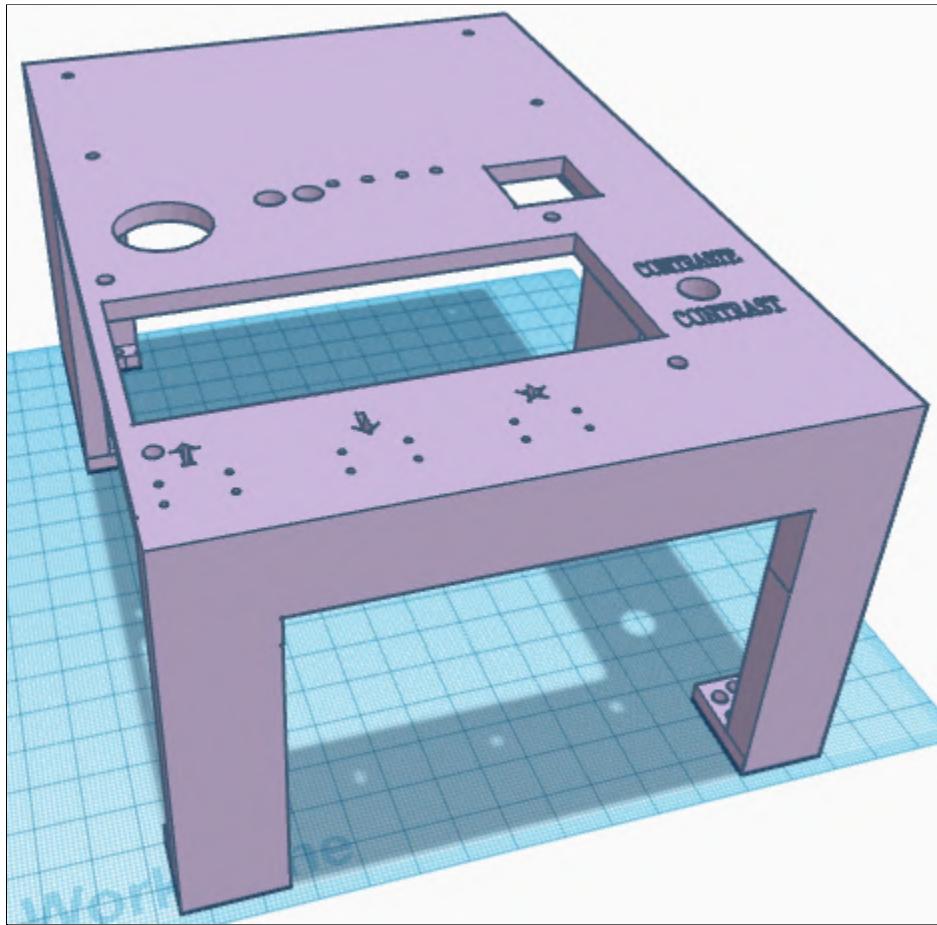
**Figure AE:** Final PCB Design

**3D Design for Battery Platform:**



**Figure AF:** Battery Platform

**3D Design for Shield Platform:**



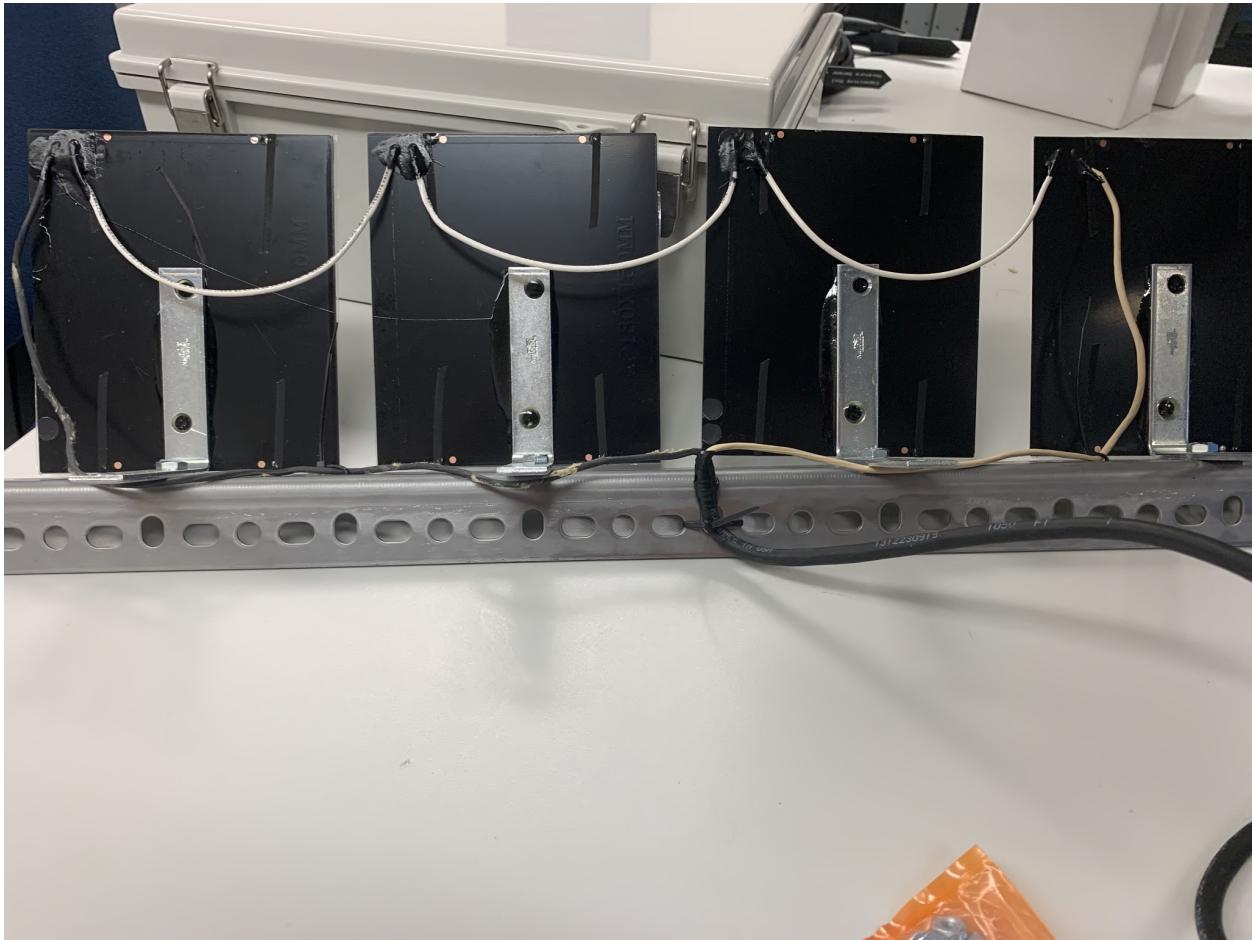
**Figure AG:** Shield Platform

## **Appendix B: Build Guide:**

Testing references

### **Power Delivery System:**

#### **Solar Panels:**



**Figure BA:** Solar Panels connected in Series

#### **Materials:**

- soldering wire
- epoxy for hardware
- 18-gauge wire
- #8 ½” screws
- Liquid Electrical Tape

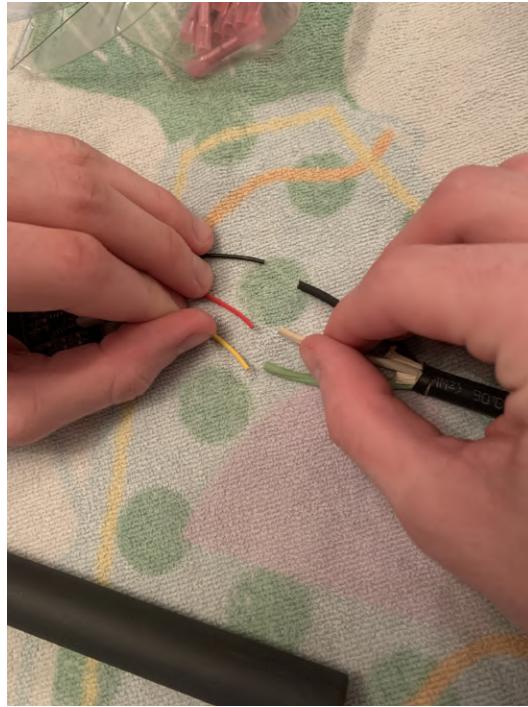
**Directions:**

1. Mount the solar panels on the rail using screws and epoxy (spacing does not matter)
2. Solder the panels in series, from the positive to negative terminals (The terminals are defined on the panel)
3. Apply Liquid Electrical tape to Soldered connections on solar panels

**Sensor Wires**

**Materials:**

- Wire strippers
  - Heat shrink
  - 18-22 [AWG] butt splices
  - Moisture sensors
  - 18/3 gauge burial wire
  - Crimper
1. Cut wires on the sensors to an appropriate length, and match them with the wires from the burial cable (yellow to green), (red to white), and (black to black).



**Figure BB:** Wire connections between moisture sensors and burial wire

2. Butt splice the connections together and crimp them



**Figure BC:** Butt-Spliced wires

3. Apply Heat Shrink around the butt splices and heat them until they are snug.

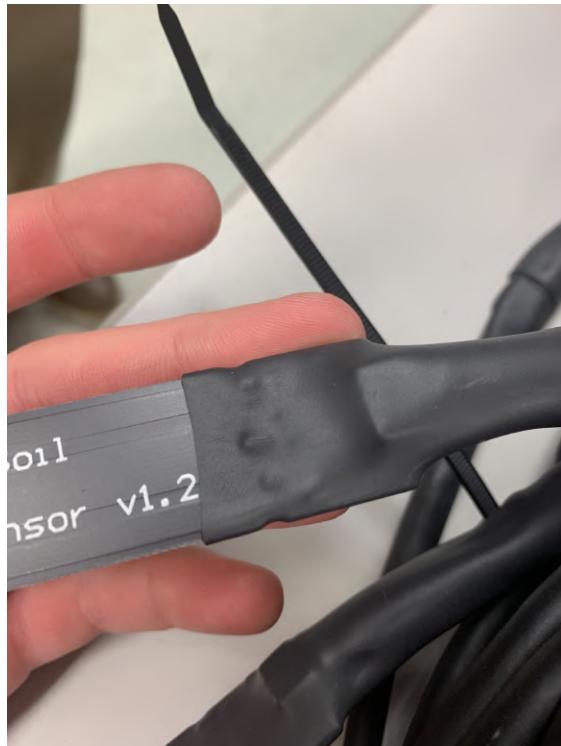


**Figure BD:** Butt-spliced wire with heat shrink applied

4. Continue to apply heat shrink around the wires until it's tight, with no leaks or exposure, until it covers the electronics on the sensor.



**Figure BE:** Butt-Spliced moisture sensor cable



**Figure BF:** Fully Heat-Shrunk moisture sensors

5. Repeat the Process for all 6 moisture sensors

### **Valve:**

#### **Materials:**

- Heat shrink
- Valve
- 18/2 Burial Wire

1. Repeat the process from the moisture sensors, and heat shrink the wire connections.

Cable matching: (Red to White), (Black to Black)



**Figure BG:** Heat shrinked valve wire

## **PCB Soldering**

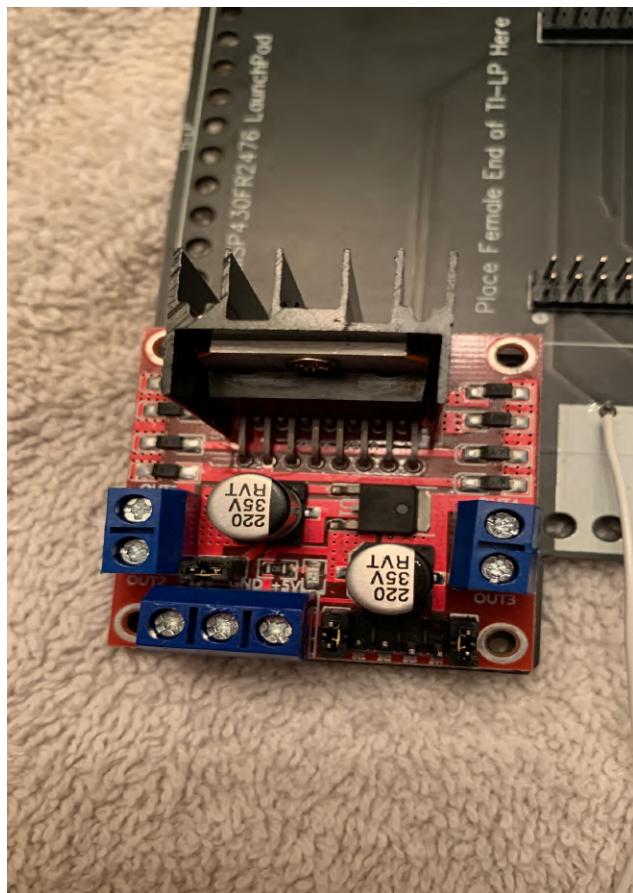
### **Materials:**

- Soldering Iron
- Solder
- 22-gauge wire (3 different colors)
- Motor driver

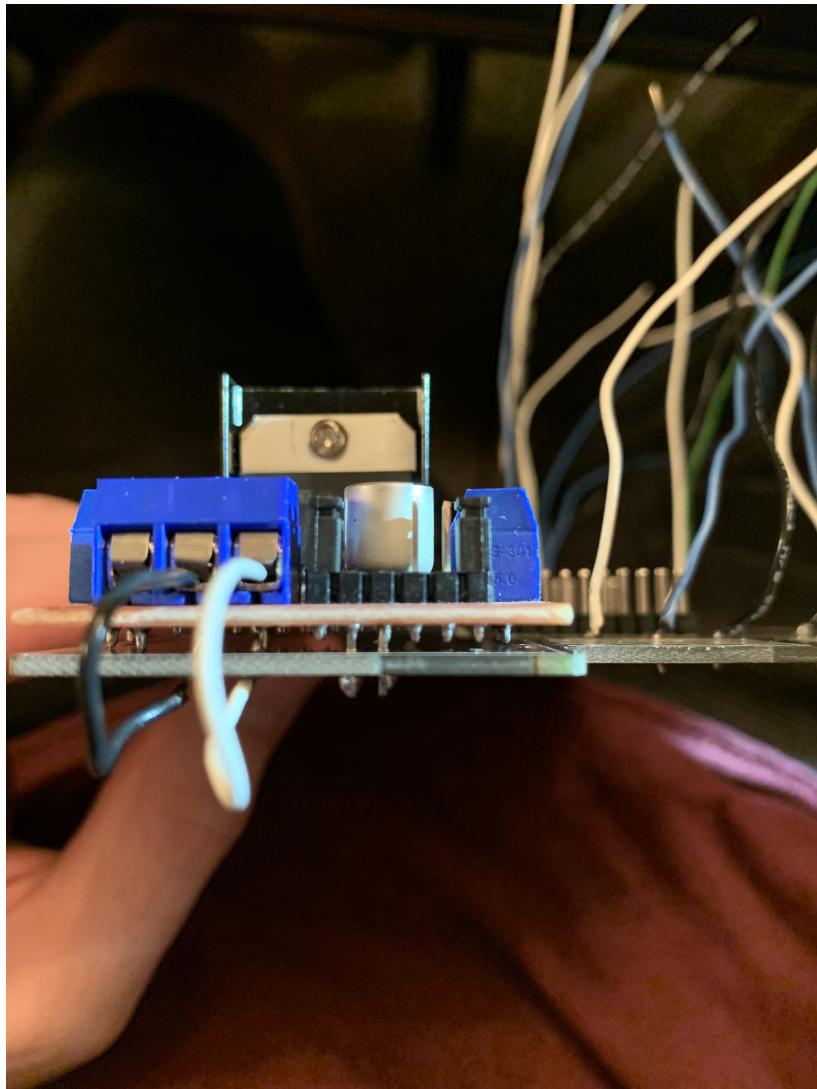
- male to male pin header connectors

## **Motor Driver**

1. Pull Pins IN1 and IN2 down under the motor driver. Fit them through the holes designated IN3 and IN4 and solder them in. Refer to the orientation below for how to place the motor driver onto the PCB board. Use 22-gauge wire to connect the 5 [V] supply and ground to the designated ports on the PCB. Solder them in place.



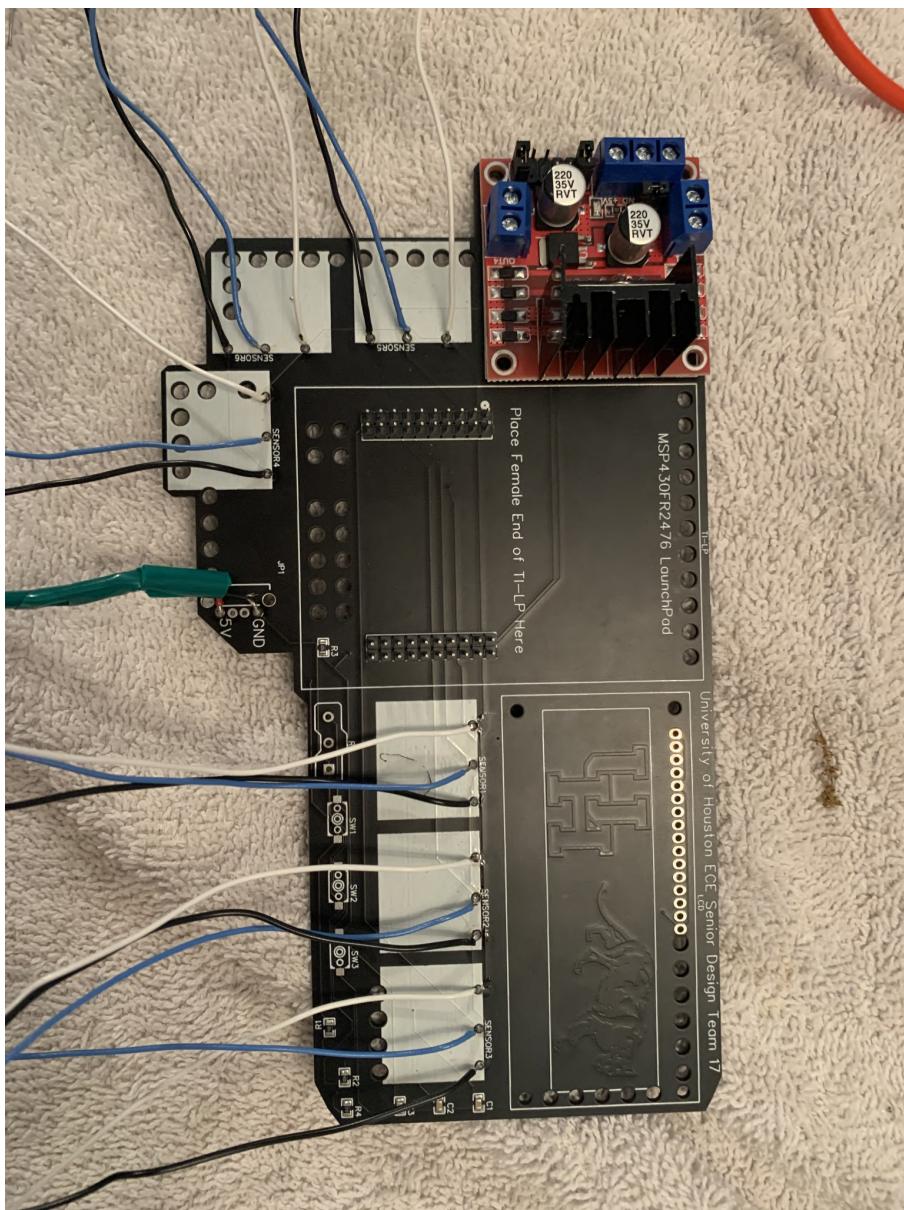
**Figure BH:** Motor driver Configuration on PCB



**Figure BI:** Motor Driver solder and wire connections

## **Moisture Sensor Wires:**

1. Orientation for Wires: Leftmost or top port, depending how it's angled, is power. We used white 22-gauge wire for this. The middle port is data, designated with a blue 22-gauge wire, and the remaining port is ground, a black wire. Solder all of these through their respective ports. Leave ample slack for the wires. Some of them will need to be longer than others to connect to the DIN Rail. They can be cut later to an appropriate length. Lastly, strip a USB and solder the red and black wires to the 5 [V] [Red] and Ground [Black] ports sitting at the bottom of the PCB.



**Figure BJ:** Soldered wires going to moisture sensors

**Microprocessor and LCD Setup:**

1. Solder male to male pin header connectors for the ports where the microprocessor and LCD are designated on the PCB. The short end should be the ones going into the ports and soldered on the other side.

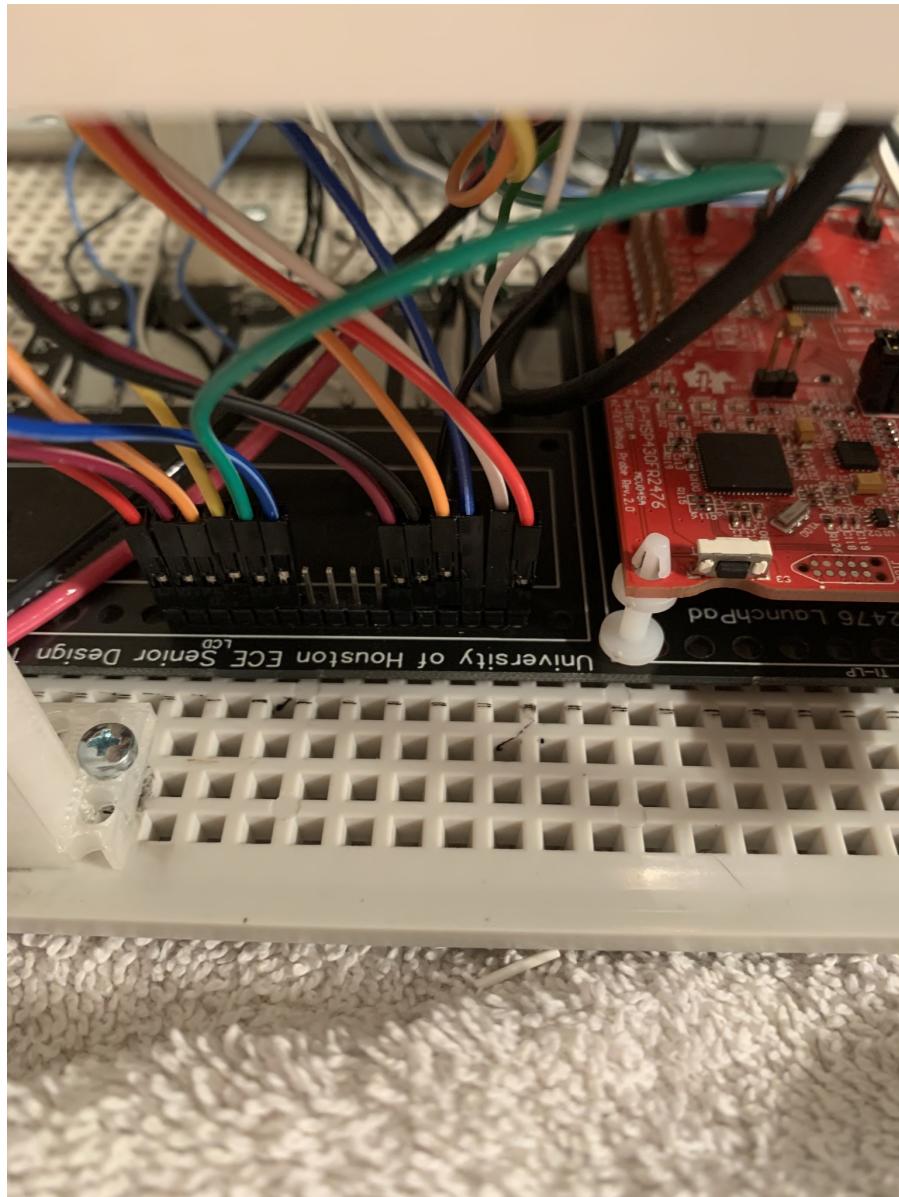


**Figure BK:** Soldered male to male header connectors (Front side)



**Figure BL:** Soldered male to male header connectors (back side)

2. The microprocessor, using the female ports on the bottom, can be directly placed onto the board. For the LCD, we will be using flexible female to female wire to connect it to the user interface. This will be addressed later. For the following picture, notice how the microprocessor is placed on the board.



**Figure BM:** Microprocessor directly placed on male header pins

### Potentiometer and Button Connections

1. On the PCB, the potentiometer is designated as “R5”. The left port is power (white wire), the middle is variable resistance (green wire), and the right port is ground (black wire). Using 22-gauge wire, we solder the connections to the potentiometer. Looking at the orientation of the figure below, the white wire is soldered on the left, the green wire is

soldered in the middle, and the black wire is soldered on the right. The potentiometer will be mounted on the 3D platform, which will be addressed later.



**Figure BN:** Potentiometer Orientation

2. For the switches on the PCB, the “UP” function is designated as “SW1”, the “Down” function is “SW2”, and the “Select” Function is “SW3”. The Cyclewet buttons used for our design have 2 connections on either side of the button. Each side classifies itself as a loop, meaning one side needs to be connected for functionality. For the ports on the PCB, the left port needs to be connected to the top node of the switch, and the right port needs to be connected to the bottom node. The connections will need to be soldered. Take care to not damage the button with the soldering iron. The wires coming from the PCB will need to be run through the holes of the 3D platform and then soldered onto the button, seen by the picture below.



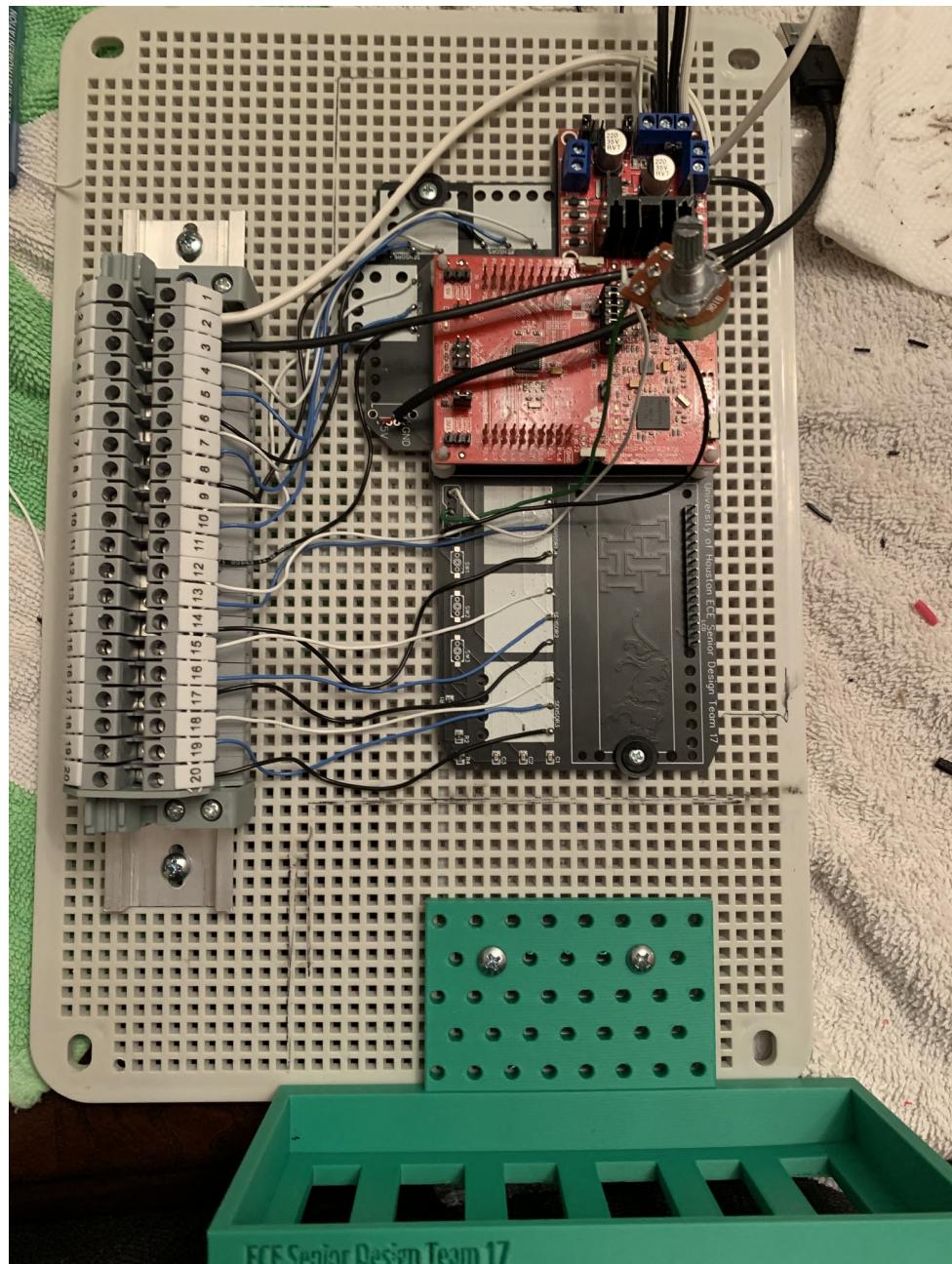
**Figure BO:** 3D platform Button Connections

## Container Interior Setup

### **Materials:**

- Din rail with 20 terminals
- #8 ½" screws
- ½" washers
- ½" nuts
- ½' rubber washers

1. For placement of the platform for the battery, sit the back plate in the container and place the platform in the bottom right corner. Then, line up the screw holes with the holes in the back plate. Screw in the battery platform with the specified screws, nuts, and washers.
2. For placement of the DIN Rail, it needs to be on the left side. Actual placement needs to take into account wire lengths of the moisture sensor wires. Make sure all moisture sensor wires from the PCB can reach their respective DIN Rail terminals. Screw the Din rail with the specified screws, nuts, and washers.
3. For the PCB, place the 3D printed platform where it does not overlap the PCB to get placements. The PCB should fit between the legs of the platform. Mark on the back plate where the legs of the platform will sit with the PCB placed underneath. To prevent metal screws from contacting the PCB, place rubber washers on either side of the screw going into the PCB.
4. For connecting the moisture sensor wires, they should be wired in their respective sequence. The black wire (ground) should be connected to the DIN rail terminal labeled “20” and then every third terminal after that is another ground wire. The blue wire “data” should be connected to the terminal labeled “19” and then every third terminal after should be another data wire. The white wire [5V] should be connected to the terminal labeled “18”, and every third wire after should be another 5[V] wire. On the DIN rail, terminals 1 and 2 will be saved for the motor driver, since those wires will carry voltage to our valve. On the motor driver, the terminal labeled Out 1 will have a 18-gauge wire connected and will go to Terminal 1 on the DIN rail. For Terminal 2, connect a small 18-gauge wire from Terminal 2 to Terminal 5 to ground the valve.



**Figure BP:** Container Back-Plate mounting

### **3-D Platform Setup**

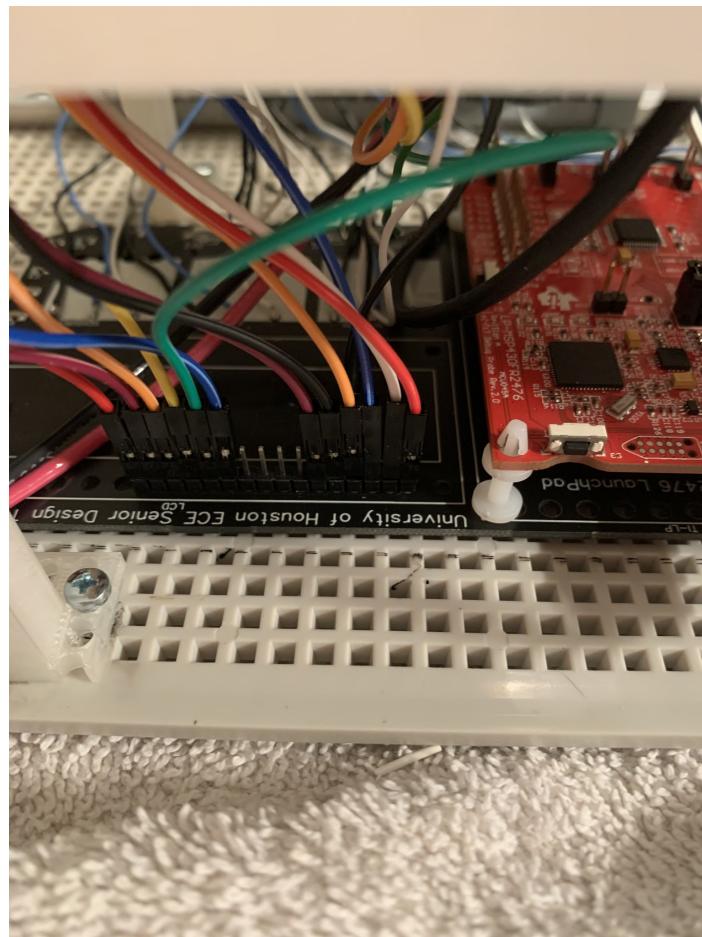
#### **Materials:**

- Charge Controller
- Manual trigger
- 18-gauge wire (red, black, and white)

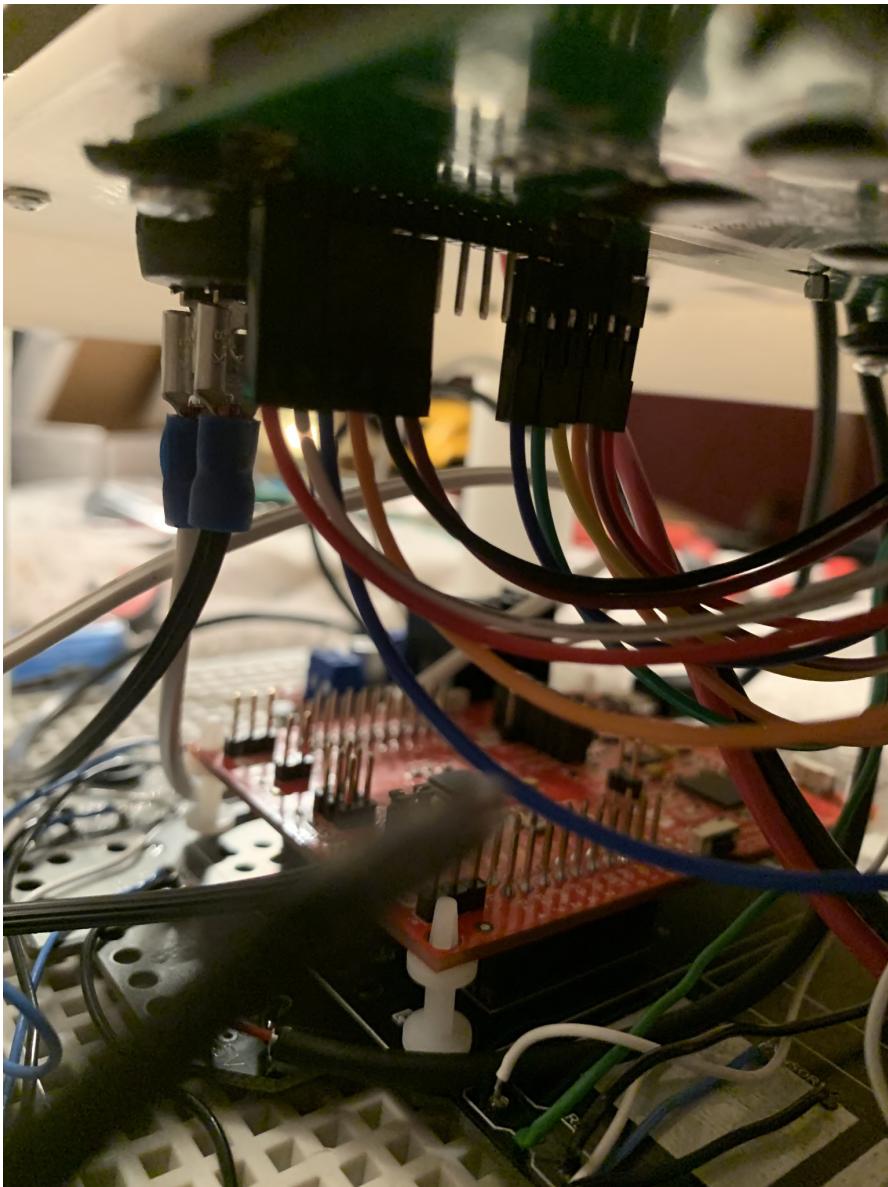
- 2 USB cables
- LCD Screen
- #8 ½” screws
- ½” nuts
- ½” washers
- 3 Cyclewet buttons
- 10K Potentiometer
- rubber washers
- 8” flexible female to female wire

1. USB will be connected into one of the ports on the charge controller and connected into the charge controller. The other will be stripped down to its red and black wires and soldered into the PCB, as prescribed in an earlier step.
2. From the locations that the legs were designated when determining the placement of the PCB, screw in the legs. 1 screw each is sufficient, provided a washer and nut are placed on the other side of the back plate.
3. Install the charge controller on the top of the 3D platform. The screw placements line up with the holes on the charge controller. The nuts would not fit on the back side of the platform because of proximity to the edges of the platform, so the screws were placed in the reverse configuration, with the washers and nuts on top.
4. Place the manual trigger in its respective place to the left and below the charge controller. It should fit snugly in there. One 18-gauge wire will connect from the “O” node to the “12[V]” port on the motor driver. The other 18-gauge wire will connect on the “I” side to Terminal 1 on the DIN Rail.
5. Place the buttons in their respective slots on the 3D platform. If the solder connections made were thin enough, they should slip into the holes printed for them. Remember that the “UP” button is “SW1” on the PCB, “DOWN” is “SW2”, and “SELECT” is “SW3”
6. Place the potentiometer in its slot right of the LCD screen. The Potentiometer comes with a washer and nut that can be tightened to secure it to the platform.
7. Install the LCD screen in the slot above the buttons and to the left of the potentiometer. Screw it in with the same configuration as the charge controller. Include rubber washers

on either side of the screw to mitigate conductivity. For the wires that are connected to the LCD, Pins 1-6, and 11-16 are used. On the PCB, if viewed where one can read the text, Port 1 starts on the left, and ends at port 16 on the right. On the LCD, the pins are labeled on either side. On one side it is labeled as Port 1, and the other end is labeled as Port 16. Using the flexible female to female wire, make the connections. It is best to use diverse colors in order to keep track of the port designations.



**Figure BQ:** LCD wiring to PCB



**Figure BQ:** LCD wiring to LCD



**Figure BR:** 3D platform component placement

## **Exterior Wiring to the DIN Rail**

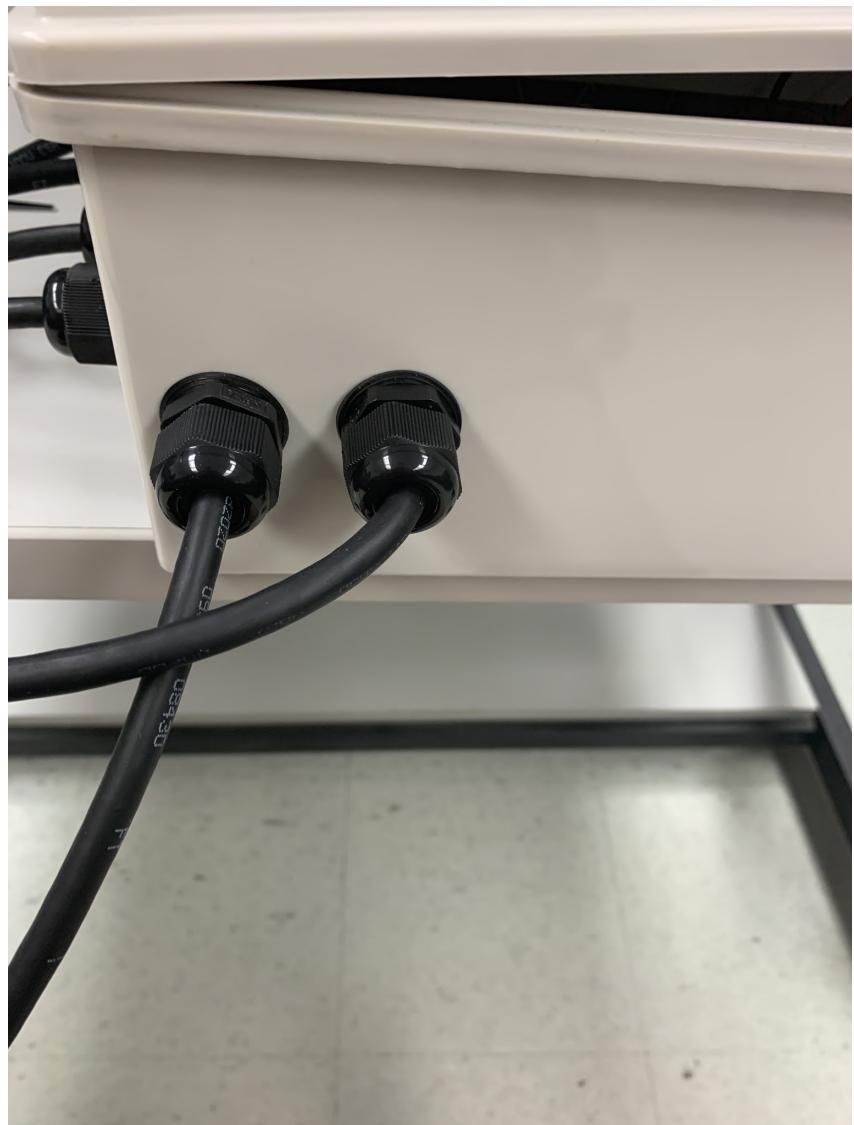
### **Materials:**

- $\frac{1}{2}$ " CIG Connectors
- $\frac{1}{2}$ " drill bit
- 18/3 and 18/2 burial wire

1. Using the  $\frac{1}{2}$ " drill bit, drill 6 holes on the left side of the container, and 2 on the bottom left side. Then, install the  $\frac{1}{2}$ " CIG connectors for the holes. Run all the cables through the CIG connectors. The top 2 holes on the left side of the container are reserved for the solar panel and valve, using the 18/2 wire.. The solar panels will go through the top hole and connect through the charge controller in the 2 left ports. The valve wires will run through the 2nd to the top hole and connect into Terminal 1 and 2 on the DIN Rail. The remaining 6 holes will be used for the moisture sensors. The wires will be stripped and placed in their respective terminals on the DIN rails, to match up with the wires from the PCB. The white wire (power) will first connect into Terminal 3, and then sequentially connect into every third one after that. The green wire (data) will connect into Terminal 4, and the will sequentially connect into every third, and the black wire (ground) will connect into Terminal 5, and sequentially connect into every third after that one.



**Figure BS:** CIG connector holes (left container side)



**Figure BT:** CIG Connector holes (bottom container side)

2. For the figure below, notice how the wiring from the exterior comes into the container and connects into the DIN Rail.



**Figure BU:** Exterior Wiring coming into the DIN Rail

## Appendix C

Vendor	Item	Quantity	Unit Price (US \$)	Sub-Total (US \$)
Amazon.com	80 Pieced M and Female Pin Header Connector 40 Pin 2.54 [mm] Single Row Straight Pin for Arduino Shield	1	11.99	11.99
Amazon.com	KeeYees 3 Pcs Capacitive Soil Moisture Sensor Module - High Sensitivity Corrosion Resistant Moisture Detection Module for Arduin	3	7.99	23.97
U.S Solid Invoice	U.S Solid Motorized Ball Valve - 1/4" Stainless Steel Electrical Ball Valve with Full Port, 9-24 V AC/DC, 2 Wire Auto Return	1	29.99	29.99
Amazon.com	ExpertPower Standard 12V 7AH rechargeable SLA Battery, (EXP1270-2) - 2 Pack	1	33.99	33.99
Amazon.com	MGI SpeedWire Strain Relief NPT Nylon Cord Grip Cable Glands - 20 Pack (1/2", Black)	1	19.95	19.95
Amazon.com	Erayco DIN Rail Terminal Blocks Kit, 20 Pcs UK-2.5N 12 AWG Terminal Blocks, 2 Pcs Ground Blocks, 2 Pcs Fixed Bridge Jumpers, 4 Pcs End Bracket, 4 Pcs End Covers, 4 Pcs	1	26.99	26.99
Amazon.com	GenBasic 40 Piece Female to Female Jumper Wires (8 Inch)	1	4.99	4.99

	QILIPSU Hinged Cover Stainless Steel Latch 370x270x150mm Junction Box with Mounting Plate, ABS Plastic DIY Electrical Project Case IP67 Waterproof Dus			
Amazon.com	RioRand LCD Module for Arduino 20 x 4, White on Blue	1	57.99	57.99
Mouser Electronics	595 - LP - MSP430FR2476	1	17.99	17.99
JLCPCB	PCB Assembly	5	3.618	18.09
JLCPCB	PCB Samples	5	2.7	13.5
Amazon.com	Potentiometer Kit - 5pcs Multi-Turn Trimmer (1K - 100K Ohm), 10 pcs Single-Turn Linear Variable Resistor Pots	1	5.99	5.99
Amazon.com	DaierTek 5pcs Circle Round Rocker Switch Wired KCD1 20mm Mini Snap Black Toggle	1	6.99	6.99
Amazon.com	Cylewet 25 Pcs 12x12x7.3mm Momentary Tact Tactile Push Button Switch Touch Switch Micro Switch 4 Pins SMD PCB with Cap for Arduino	1	7.89	7.89
Amazon.com	ALLPOWERS 2 Pieces 2.5 W 5V/500mAh Solar Panel DIY Battery Charger Kit Mini Encapsulated Cell Epoxy for Battery Power LED 130x150mm	1	12.9	12.9
Amazon.com	TIMESETL 5PAck L298N Stepper Motor Driver Controller Board Dual H Bridge Module for Arduino Electric Projects	1	12.99	12.99
SJOOW	18/2 SJOOW Portable Cord 300V UL/CSA	1	2.6	2.6

SJOOW	18/3 SJOOW Portable Cord 300C UL/CSA	1	18.6	18.6
			<b>Discounts</b>	-8.00
			<b>Shipping Fees</b>	51.53
			<b>Total</b>	378.92

**Appendix D: User Guide:**

# Smart Drip Irrigation System

## User Guide

By: Victor Carpio, Vedant Chopra, Joshua Mica, Diego Tolentino

## **Introduction:**

Welcome to your new Smart Drip Irrigation System (SDIS)! This system is designed to assist you by determining when irrigation for your crops is needed and taking care of the irrigation process for you, saving you lots of time and effort. Please read through the following document fully prior to operating your SDIS to ensure you have a good understanding of how to operate the system and properly configure it to your operation.

## **System Overview:**



**Figure 1:** User Interface Layout

In Figure 1 we can see the parts of the user interface (UI). We have three buttons with functions that will allow the user to scroll the menus up, down and to make selections on menus and sub-menus. Additionally, there is a potentiometer that can adjust the contrast of the LCD and allow the viewing to be easier. There is a toggle switch which manually triggers the SDIS, which when in the off position keeps the valve closed and once it is flipped to the on position it will open the valve. This is not dependent on the reading from the sensors and will allow the valve to open at the user's will. Finally, we have our charge controller which will display the information about the battery and the input voltage from the solar panels.

## **Initialization:**

To power the system once the battery has been connected the “Enter” button must be pressed to power on the system. Once the system is on if the LCD is hard to see, use the potentiometer on the UI to adjust the contrast and allow for the LCD to be viewable. Once the system is powered and the LCD is viewable it is time to set the saturation point. To do this scroll down using the UI to the “CHANGE MODE” option and select the desired operation mode. Please refer to Table 1 to see what the saturation points are for each setting.



Figure 2: Setting the Saturation Mode

<b>Saturation Mode</b>	Saturation Range [mL/cm <sup>3</sup> ) Dry Point - Saturated Point
<b>Low</b>	0.075059645 - 0.150119289
<b>Medium</b>	0.150119289 - 0.300238579
<b>High</b>	Non-Operational

### **Current Mode:**

To view the current mode that the system has been set to scroll to the “CURRENT MODE” option. The system will display a string that will notify the user of which saturation mode the system is currently in.

### **System Information Screen:**

Once the saturation point has been set the system is already running and will determine if saturation is needed actively. To view this scroll down to the “CURRENT MODE” this screen will display the system information such as: Valve Status, Queue Status, Flow Error, Sensing Error. If the valve is queued after two hours of operation the valve will open since the system has determined that irrigation is needed. Once the sensors detect that the soil has hit its saturation point it will immediately close the valve. This combats the chance that it starts raining during irrigation. Rather than leaving the valve open for a set amount of time it is dependent on the current soil status. If the valve is not queued and the two hour mark the valve will not open since the valve is not queued. The system will report a flow error if the sensors detect that the soil has gotten drier since the valve has opened meaning that there is now water flowing reaching the sensors. The system will report a sensing error if the sensor's readings are not within the operational range.

## **Appendix E: Links to Google Drive**

<https://drive.google.com/drive/folders/1HNw8XwRi3Vfa8td3vLfA8BSOCOotDIOP?usp=sharing>