Soil Moisture Controller

(Pitched Project)

ECE 445 Final Report

Team 39
First Yingyord - phuriy2@illinois.edu
Isabel Alviar - ialviar2@illinois.edu
Ren Yi Ooi - rooi2@illinois.edu

University of Illinois at Urbana-Champaign - Electrical and Computer Engineering Department

May 4, 2023

Abstract

This project was pitched by the U.S. Department of Agriculture (USDA) based on the paper "A Cost-Effective and Customizable Automated Irrigation System for Precise High-Throughput Phenotyping in Drought Stress Studies." The need for the project was to create a device that provides constant moisture level sensing and water irrigation to different types of soil, to be used by the USDA during experiments involving plants in specialized CO2 growth chambers. Working in conjunction with a capstone team of students from the Department of Agricultural and Biological Engineering, our main goal was to create a cheaper, yet more efficient system than the one proposed in the paper that is also scalable up to 4 plants. This report discusses how our final system works, the high-level details and technical components of each subsystem, the steps we took to complete this project, and more.

Contents

1. Introduction

1.1 Purpose

One of the biggest limiting factors for gains in agricultural productivity is the ability to provide sufficient moisture in the soil for the growth of crops. To manage this issue, various water management strategies have been developed to ensure that there is sufficient water being applied over these crop lands. Currently, the measurement of soil moisture content in pots are performed manually with individuals monitoring the moisture level based on weight, or the use of gravimetric sensors. However, due to the many different components that make up the weight of the pot, it is difficult to measure the exact proportion of increase in plant mass to the change in soil moisture content to obtain an accurate indication of when the irrigation has to be activated. As a result, there is a need for a more precise method to measure and maintain the soil moisture conditions in these pots through the use of soil moisture sensors.

Our solution consists of a cheaper yet more effective device that provides constant moisture monitoring and water irrigation as needed to different types of soil. When the moisture within the substrate is below the predetermined target level (e.g. 25, 50, 75, or 100 percent), the water valve will be triggered to an extent where the moisture can be maintained at that level. In addition, there is also an interface so users are also allowed to check the current status of each pot, that is whether the substrate moisture is desirable as indicated by LED lights, and control the target level in the pot through a keypad. Our team worked alongside the team of ABE students to construct the irrigation system and ensure that our solution could be scaled up for high-throughput of at least 50 plants in the future.

1.2 Functionality

We set out with three main high-level functionality requirements for our project and were able to achieve all of them. First, the moisture sensors are able to detect the current level of moisture in the soil of each pot for the moisture level data to be logged on an SD card and displayed on an LED bar graph every 6 hours. The system is also able to provide irrigation when the moisture level falls beyond a set threshold level as inputted using a keypad by the user. Finally, the system was made scalable to four different pots and the moisture level was maintained at 100%, 75%, 50% and 25% in each of the respective pots.

1.3 Subsystem Overview

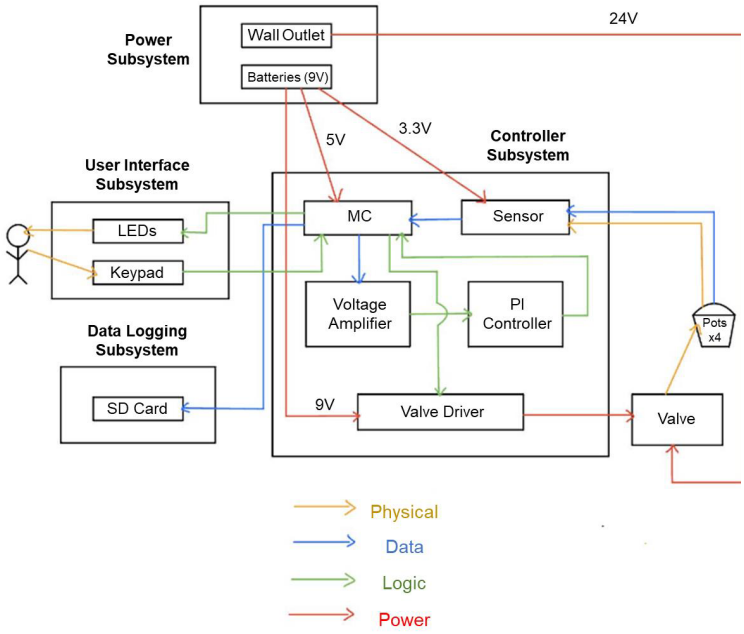

Figure 1: Block (Top-Level) Diagram

As shown in Figure 1, our project can be broken down into five subsystems: irrigation, data logging, user interface, controller, and power.

Irrigation Subsystem: The irrigation system for each pot consists of a valve that opens and closes based on the moisture level measured, in order to maintain a desired set of moisture conditions for different soil and soilless substrate mixes. When it is sensed that the moisture level falls below a certain value, relay switches are activated by a microcontroller to operate 12V irrigation valves that correspond to each sensor-controlled pot, and water will flow out until the soil reaches an ideal saturation rate again.

Data Logging Subsystem: The data logging subsystem consists of a data logging shield, an instrument that monitors and records changes in conditions over time. Eight times a day, the data logger takes in information from the sensors and microcontroller and records current volumetric water content, saturation rate, time, and date for each pot on an SD card. When the card is plugged into a laptop, all of this data is easily viewable in a text file.

User Interface Subsystem: The user interface subsystem consists of two main components: a 10-segment LED bar graph that shows the soil moisture level as detected by the sensors in each pot and a 4x4 keypad where the user is able to input the desired soil moisture level that the user would like the pot to be maintained at.

Controller Subsystem: The controller subsystem is originally designed with electronic components such as operational amplifiers, resistors, and capacitors, but later shifted to functions on the microcontroller which computes how much time the valve needs to be open to allow enough water into each pot.

Power Subsystem: The power subsystem consists of 9V batteries that are rechargeable and provide power to the other subsystems like user interface subsystem, data log subsystem and controller subsystems. A higher voltage of around 12 V DC is required for the irrigation valves, and the ABE team has requested for this voltage to be provided using a wall plug adaptor.

2. Design

2.1 Equations and Simulations

Control System

Originally, we wanted our system to constantly monitor the moisture level of each pot's soil and maintain it within 5% of the desired saturation rate. The most challenging part of our project was to construct a closed-loop control system that satisfied this requirement. Since the PI controller was implemented with operational amplifiers (op-amps), resistors, and capacitors, the majority of the tuning was done by adjusting the resistance, capacitance, and ratios between resistance on each side of the amplifier. In order to test the feasibility of the control system itself, a simulation of the closed-loop system in response to a unit step function was run on LTSpice, as shown in Figure 2 and Figure 3 below. All op-amps in the simulation are in the LM741 model.

It is important to note that later in the semester, our project requirements changed per the request of the USDA representative so that the pots only be watered four times a day, in order to prevent overwatering that would cause the plants' roots to rot. Therefore, our control system was scrapped and redesigned to an open-loop system. However, it was helpful to have started with creating simulations of our original design.

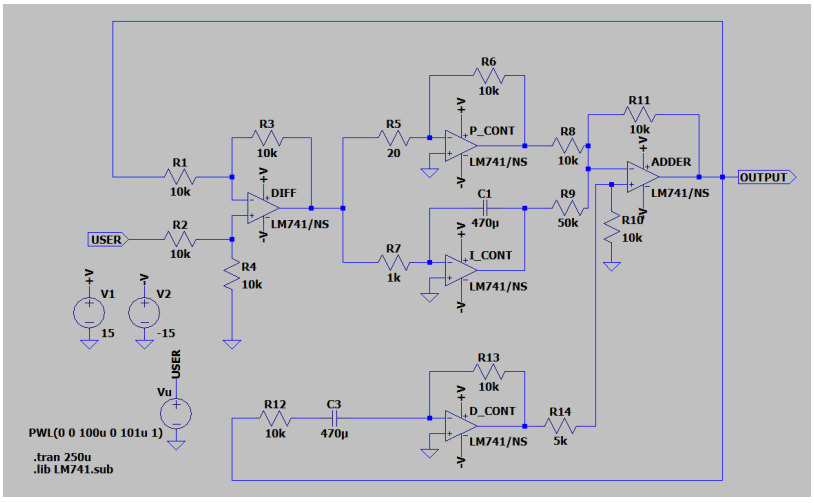

Figure 2: Closed-loop control system circuit for simulation

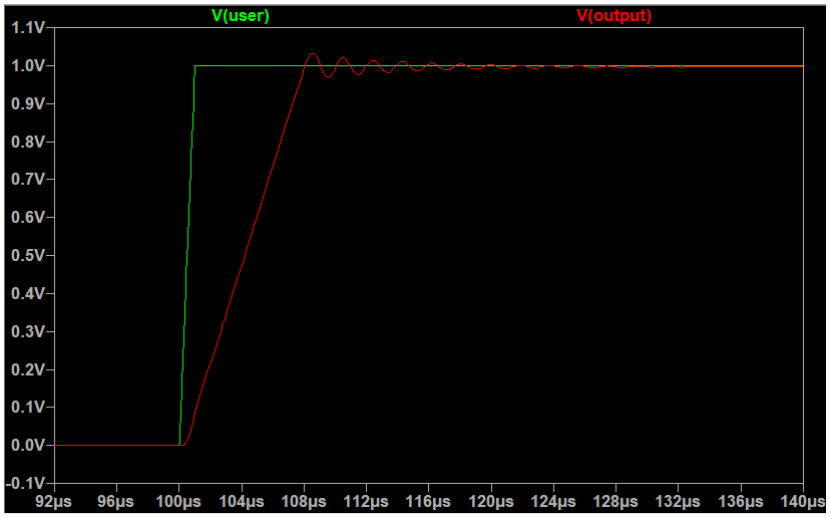

Figure 3: Simulation Result

Sensor Calibration and Converting VWC to Saturation Rate

The type of moisture sensor we are using is the ECH20 EC-5 from MeterGroup. The sensor is a three prong design (power, analog output, and ground) that determines volumetric water content (VWC) by measuring the dielectric constant of the media using capacitance and frequency domain technology. The number that is output by the sensor is an excitation voltage in mV, that then needs to be put into a formula that gives a volumetric water constant (our measurement of moisture level) between 0 and 1 (0-100%).

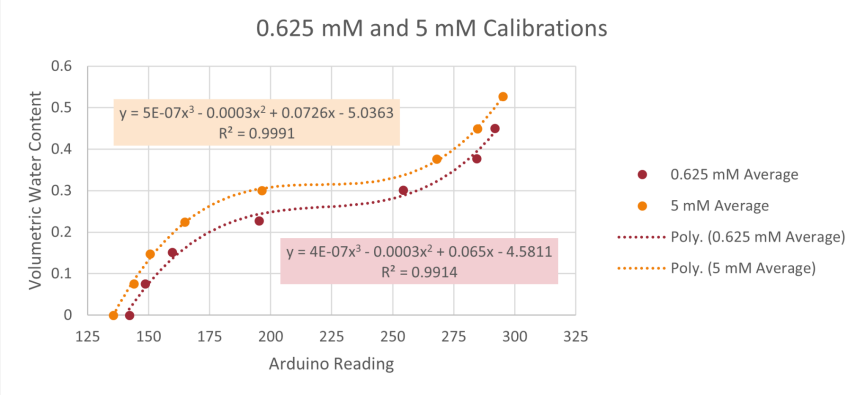

Figure 4: Sensor Calibration Equations

Each sensor had to be calibrated specifically to the type of soil being used. For the purpose of our project, we prepared two types of soil: 5mM and 0.625mM. First, the ABE team oven-dried the soils in the lab. Then, we derived two unique formulas to convert the output of the sensor (VWC) to a saturation rate between 0-100, so we could compare against the desired value and determine if irrigation was needed. The equations for the 5mM (orange) versus 0.625mM (red) soil are shown in Figure 4.

Irrigation Time with Constant Rate Control

Because of the constraint that the plants should only be watered four times a day – 00:00, 06:00, 12:00, and 18:00 – we determined how much each plant should be watered by using a set flow rate of 31.5mL/second and comparing its measured versus desired saturation rate. Once the saturation rate for each plant is known, we use those values as inputs into a formula that outputs how long the valve should be open and water should flow out for all pots which need the irrigation. The equation is:

$$t_{open} = (SR_{desired} - SR_{actual}) * 0.46/31.5$$

In the equation, we compute the difference between the user-input and actual saturation rate, then multiply by the porosity of the soil, which is 0.46 in this case, to get the amount of water needed to fill the moisture difference. Finally, the amount of time the valve is open is derived by dividing the amount of water needed by the fixed flow rate.

2.2 Design Descriptions, Diagrams, and Alternatives

2.2.1 Irrigation Subsystem

Irrigation is the process of artificially applying controlled amounts of water to land, plants, or crops. For our system, this was done by pumping water directly from the pipes in Edward R. Madigan Laboratory to irrigate experimental pots of soil in a CO2 growth chamber. Our irrigation subsystem consisted of 12V solenoid valves, drip emitters, a pressure control valve, a flow splitter, and a system of plastic tubes. For each pot of soil, the solenoid valve would open and close based on the moisture level measured, in order to maintain a desired set of moisture conditions for the four pots. An important note is that the water source outside the plant chamber was always turned on so that there was a constant supply of water ready due to pressure build up. Another note is that the pressure control valve was used to set the flow rate of water coming out of the pipes at a constant 31.5 mL/second. This eliminated the need to factor in flow rate to how long each plant should be watered, and instead we could solely focus on comparing the measured and desired saturation rates.

Irrigation is needed in a given pot if it is sensed that the moisture level falls below a certain saturation rate (for example, below 75 for 5mM soil). When this happens, relay switches activated by our control system and microcontroller open the irrigation valves, powered at 12V, that correspond to each sensor-controlled pot, and water will flow out for a calculated amount of time until the soil reaches an ideal value again. When the plants are done being irrigated, the valve will click shut and water will stop flowing out.

Figure 5: Irrigation system setup inside the growth chamber

2.2.2 Data Logging Subsystem

The data logging subsystem consists of a microcontroller, data logging shield, SD card, and 9V battery. The data logger takes in current volumetric water content as analog output from the sensors, as well as time and date from the microcontroller. For each pot, volumetric water content, saturation rate, time, and date will be stored on an SD card. When the card is plugged into a laptop, the data is easily viewable in a text file. When having to switch from PCB to breadboard, it was convenient that the data logging shield sat nicely on top of the Arduino Uno, our microcontroller. An alternative option with a PCB is to create a data logger from scratch using a voltage regulator, real time clock, level shifter, SD card holder, resistors, capacitors, 6-pin ICSP header, and coin cell battery.

Originally, our data logging subsystem was designed to record data every 60 minutes. However, the sample rate has since changed to 8 times a day, per the USDA representatives requirements. At 00:00, 06:00, 12:00, and 18:00, the sensors will be checked and data will be logged. Then, irrigation will happen and the system will wait 30 minutes for the water to settle. After, the sensors will check VWC again and data for each pot will be logged to see if any more irrigation was necessary. Then, the system will wait five and a half hours and repeat again.

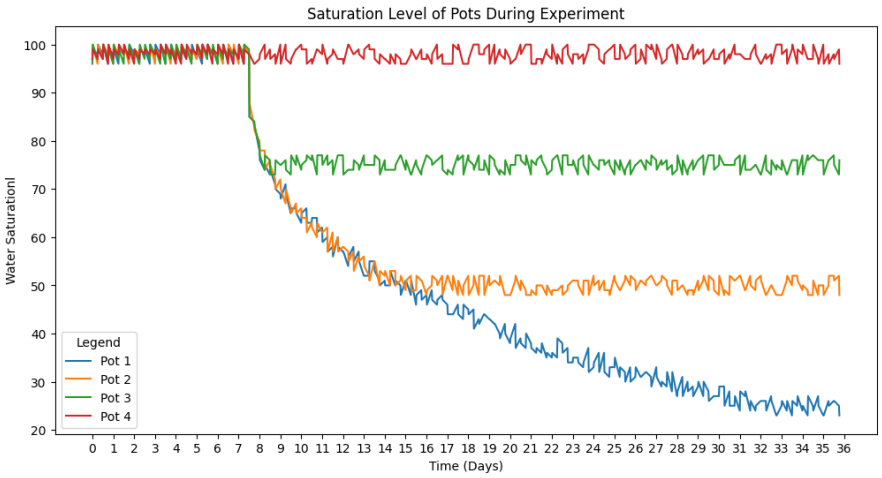

Figure 6: Saturation level of pots over time

With this method, there are only 8 data points logged per day. Not only did this make it difficult to demonstrate our system for our Final Demo in the 30 minute allotted time, it also made it difficult to run and collect data over multiple days while we were still putting the finishing touches on our whole system. The maximum time period we collected data was from midnight on April 19 to 12:30 P.M. on April 22. Instead, Figure 6 shows how the pots would perform over a more accurate span of 36 days. With the soil starting at full saturation (100%), it would take a few days for the plants to start losing any water until finally they would slowly drop down to their desired saturation rate. As each pot reaches its desired saturation rate, its curve peaks up and down slightly each time it receives irrigation as its ideal level is being maintained.

2.2.3 User Interface Subsystem

One of the biggest challenges faced when researching how to connect the LED bar graphs to the microcontroller is figuring out a method to reduce the number of wires that would have to be connected to the IO pins of the microcontroller. Each 10-segment LED bar graph acts like a combination of 10 different LEDs, and to wire it, there would be 10 resistors and 10 wires required, utilizing 10 digital pins on the microcontroller. Having four 10-segment LED bar graphs would entail having 40 resistors and 40 wires, utilizing 40 digital pins on the microcontroller. This would not only be very messy, but if the other subsystems like datalogging are all connected to the same microcontroller, there would not be sufficient digital pins.

As a result, there is a need to come up with a solution to reduce the number of pins necessary to wire four 10-segment LED bar graphs. There are several methods that this can be achieved, including using a matrix network to select the different LEDs on the bar graph and combining them such that the LEDs would all run using the same wire. However, the solution that was used eventually involved the use of power MOSFETs – specifically the IRF520 – and connecting them to each 10-segment LED bar graph. The power MOSFETs consist of NPN transistors that would allow us to switch between each 10-segment LED bar graph by turning them on and off sequentially. If this switch were to be made very rapidly, it would allow the user to see all four 10-segment LED bar graphs together. This would allow for the reduction from the 40 digital pins previously required to only 10 digital pins and an additional four pins for each of the four power MOSFETs giving a total of 14 digital pins required on the microcontroller.

As for the 4x4 keypad, it was first necessary to study the datasheet to determine how it can be interfaced with the microcontroller, and it turned out that there is an 8-pin access to the 4x4 matrix. The matrix keypad uses a combination of four rows and four columns to provide button states to the microcontroller. Underneath each key is a push button, with one end connected to one row, and the other end connected to one column.

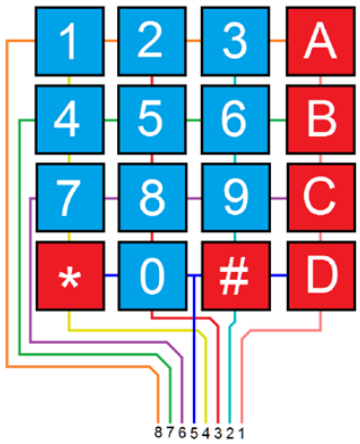

Figure 7: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed. For example, the microcontroller would first pull all four columns low and then pull the first row high. It then reads the input states of each column and reads pin 2 high. This means that a contact has been made between column 3 and row 1, so button '3' has been pressed. In terms of the hardware connection on the schematic, this would mean that the keypad can be represented as an element with four inputs and four outputs, which is shown in Figure 7.

2.2.4 Controller Subsystem

The controller subsystem is the subsystem which has undergone the biggest changes since the original design. As can be seen from the simulations section, the original controller is intended to be a PI controller, which allows proportional, and integral control over the moisture value. In other words, not only can the controller command the amount of water to be dispensed but also the flow rate fluctuations which allow the fine tuning of water dispensing for the moisture value to settle within the desired range. In the original design, the controller mainly consists of operational amplifiers to amplify or combine multiple signals, potentiometers to allow physical tuning for each particular controller unit, and capacitors to enable signal integration. While the input of the controller is an indication of the error between the measured moisture value and the desired user value set by the keypad, the output of the controller determines the flow rate of the valve to dispense the water for each pot.

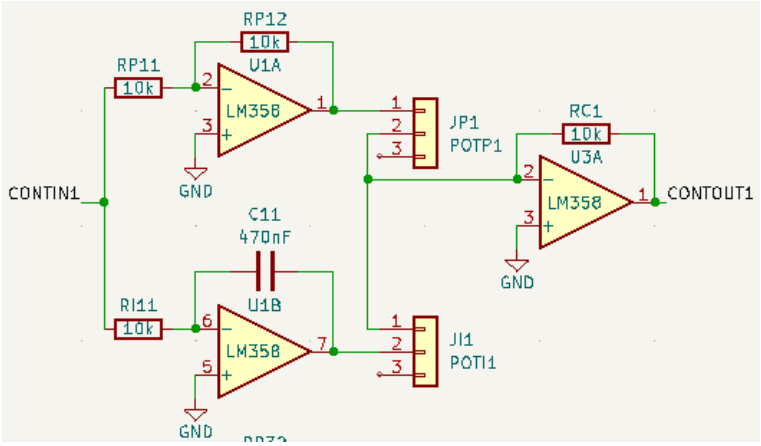

Figure 8: Original controller design

However, the new requirements set by the USDA representative force the change of the scope and design of the controller. The most significant change is in considering the controller system implementation within future experiments. Because the moisture controller will be used with actual plants, the controller should not water the soil more than four times a day to prevent the roots from rotting. Another minor change is in shifting the control parameters from flow rate to the length of time the valve is open. With these two changes, we lose the advantages of using a PI controller because we cannot frequently accept feedback from the moisture sensor and open the valve when the moisture drops below the desired range, and the manual tuning which can indicate the flow rate difference between each valve, is dismissed from the constant flow rate control equations. Therefore, the physical parts for the controller subsystem have been scrapped, and the controller design is shifted to coding the valve opening time dependent on the amount of water needed for each pot.

2.2.5 Power Subsystem

It is intended that the soil moisture sensors would be powered by the Arduinos which would be connected to the computer. This allows for a stable power supply to the sensors compared to using a rechargeable battery since a rechargeable battery would be depleted over time. The sensors would require a minimum of 2.5 VDC to accurately pick up the data on soil moisture levels. The Arduino microcontrollers accept 7-12 V DC through the power jack and the onboard voltage regulator steps down the voltage to the required 5 and 3.3 V DC. This would allow for the data log and user interface subsystems to be used as portable devices without a need for mains voltage, which is often the situation in a greenhouse.

As for the irrigation valves, they run on 12V DC and typically do not have more than 18W of power. The power supply that is used to power the irrigation valves is the RS-75-12, which runs on 12V DC and has a maximum supply of 72W. For power to be drawn from the power supply, a wall plug adapter was stripped into its Live, Neutral and Ground wires, and they were connected to the respective pins of the RS-75-12 power supply via the screws. For the output pin on the

power supply (V+), an 18-gauge wire was chosen to handle the amount of current flowing out of the power supply into the valves. This singular 18-gauge wire would then branch into four separate 18-gauge wires for each of the four valves. A pictorial representation of the power supply is shown in Figure 9 below.

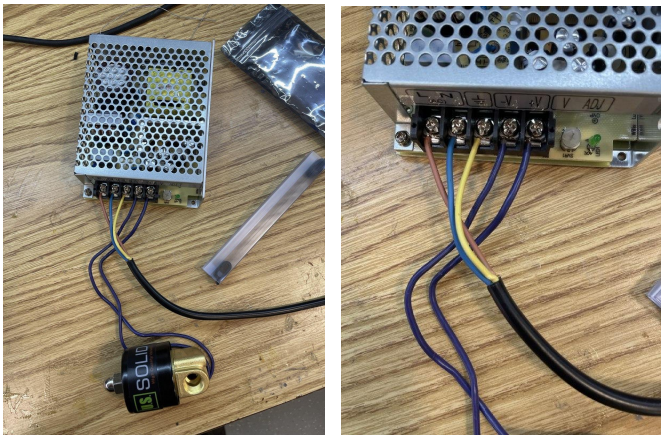

Figure 9: 12V Power Supply Connections

2.2.6 PCB

The PCB Design Schematic is split into different systems as represented in Figure A1 in the Appendix. The microcontroller that was chosen for the project is the ATmega2560 microcontroller, also found on the Arduino Mega 2560 board. This microcontroller has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The 4x4 keypad is connected directly to the microcontroller in the schematic.

For the valve driver and controller unit, a PI controller takes the feedback input from the moisture sensor, compares the measured value with the desired value, and triggers the water valve if the measured value is below the desired value. The output of the controller will be amplified and connected to a LM555 Timer chip in order to generate a PWM signal to the water valve so that the amount of water being given is sufficient to each pot.

For the user interface, each 10-segment LED bar graph would be connected to a IRF520 Power MOSFET on its drain side. The three terminals of the MOSFET, the “Gate”, “Drain” and “Source” are connected to the digital input-output pin of the microcontroller, the 10-segment LED bar graph and ground respectively. Each of the LEDs in the 10-segment LED bar graph would be connected on the same bus, which is then connected to a single 470 ohm resistor before it is wired to the microcontroller.

3. Requirements and Verification

3.1 R&V Tables

3.3.1 Irrigation Subsystem

Requirement	Verification
Valves will open/close to dispense a correct amount of water to the appropriate pot based on the signal received from the controller	- When the 24V valve for a pot receives a signal that irrigation is needed, it should flip open like a lid and water flows out. Once moisture level is reached, the valve should close within 1 second. Overall, when irrigation is not needed, the valve should remain closed. - Valves are powered when closed, non-powered when open. Use a multimeter to measure if there is any power (>0.5W) in a valve currently dispensing water versus in a valve that does not need to dispense water (should be 0W).
The irrigation system should be scalable so that at least 4 pots can be watered in by respective amounts at the same time (within 5 minutes for 4 pots)	- Set up 4 pots with different soils that require 4 different moisture levels (eg. 25%, 50%, 75%, 100%). Make sure all are under-watered and need irrigation - Plug in SD card and run program - All 4 pots should be watered at the same time so the display reads their correct desired moisture level within 5 minutes (start to finish) - The irrigation will happen simultaneously through 4 different valves
Water will be pumped from a supply tank using a series of tubes until a predetermined desired moisture level between 0 and 100 is reached for a given pot	- Connect the control system to the irrigation system. Irrigation system includes pipes and valves, and water is pumped from an external source - Test using one pot of soil that is below its desired moisture level, for example, 50%. Water should continue being pumped into the pot until the sensor detects that it has reached 50% again - For a sanity check, the user should be able to see the moisture level for that Pot say 50% on the LCD display

3.3.2 Data Logging Subsystem

Requirement	Verification

Data logger should send a pulse I/O of 0 or 5V to signal to the controller whether or not irrigation is needed for each pot	- Test using an LED (controls is the part that actually activates irrigation valve opening/closing, and irrigation is dependent on the ABE team, so we just need a way to simulate if a pulse of 0 or 5V is being sent for data logger) - Using a multimeter, measure the voltage of sensor for one pot when it is at an ideal irrigation level (should 0V if no irrigation needed) - After water is removed so that moisture is below a level and irrigation should be needed, measure if the voltage of signal is 5V
Record data every 60 minutes* (sample rate) for each pot and store on an SD card *Note: Our sample rate is subject to change, based on the USDA's requests	- Allocate a few hours of time to be able to check the data logger each hour (eg. 4pm-6pm, 12-9pm, etc) - Make sure SD card is plugged into the card socket on the front of the Arduino - Start running Arduino program from scratch - Plug in SD card to computer to extract the data then read data from SD card with Arduino → file.read() method → results get printed in a text or CSV file - Verify if data is actually printed with timestamps 60 minutes apart
Data logger should be able to read more than 2 inputs such as voltage, current, temperature, etc. at once	- Set up data logger to take readings every 10 seconds of 2 inputs from the sensor, such as temperature and voltage - Over 24 hours, starting on a specific day, begin running the system. We know the logger should produce 8640 data points (1 day = 86400 seconds) - Logger should produce data for each point (1 rows/point) - Extract data into CSV file with temperature and voltage as the columns if correct

3.3.3 User Interface Subsystem

Requirement	**Verification**
The 10-segment LED bar graph must receive and display data from the microcontroller for visualization of soil moisture level data collected by the sensors in the pot.	Connect the 10-segment LED bar graph to the Arduino microcontroller. Verify that the 10-segment LED bar graph is able to display the correct amount of soil moisture within a 10% range.

The user must be able to input any percentage value of soil moisture level from 0 to 100 using 4x4 keypad.	Connect the 4x4 keypad to the Arduino microcontroller together with the 10-segment LED bar graph. Verify that the value that is entered in the keypad is reflected accurately by the 10-segment LED bar graph.
The microcontroller must be able to send and receive data to and from the voltage comparator in the controller subsystem.	Test the situation where the user enters a soil moisture value that is below the current soil moisture level detected by the sensors. Ensure that the valves in the irrigation subsystem do not open. Test the situation where the user enters a soil moisture value that is above the current soil moisture level detected by the sensors. Ensure that the water valves in the irrigation subsystem opens within one minute and subsequently maintains the soil moisture level at the level set by the user.

3.3.4 Controller Subsystem

Requirement	**Verification**
The voltages from the moisture sensor and user input must be calibrated to the same scale in order to acquire a reasonable voltage difference in a linear range from 0-5V.	Connect the scaled outputs of user input and moisture sensor to a digital multimeter to check the output voltage. Verify that the voltage of both inputs can be linearly converted to moisture percentage. The characteristics between the readings and voltage can be obtained from manually measuring the voltage at each reading and plotting the graph between the two variables. An additional amplifier and voltage divider would be needed to adjust the voltage offset and range from both sources.
The control system must be able to remain stable.	Test the stability of the control system by connecting the power supply as a temporary input to the system and oscilloscope to the output of the system to check that the current and voltage does not become unstable. As a safety measure, connect the voltage clipper circuit to the system output so that the voltage is only limited to 5V.
The controller must be able to send the information on the amount of water needed for each pot so that the relative error would be less than 5% within ten minutes.	Obtain the amount of water being transmitted to the substrate by connecting the power supply and timer chip to the valve input and varying the duty cycle of the signal into the valve. Collect the data for the water needed to fill the pot at each moisture baseline (25%, 50%, 75%, and 100%) to gather an estimated

	amount of water to increase the moisture percentage by one percent. Verify by inputting the desired duty cycle and compare the moisture percentage of the substrate after ten minutes have passed.

3.3.5 Power Subsystem

Requirement	Verification
The 12V power supply should be able to power the irrigation valves and allow it to switch between on and off.	There should be a ‘click’ sound indicating that the power supply opens when the irrigation valve is connected to the V+ and V- pins of the 12V power supply.
The batteries should be able to recharge to its full capacity within 5 hours.	Place the rechargeable battery with the charger and check if full-capacity charge is achieved after 5 hours.

3.2 Discussion and Quantitative Results

While some of our requirements and verifications have changed with the scope of our project and high level requirements changing due to the requests of the ABE team and/or USDA representative, any changes made to the system are not a reflection of our inability to meet our original requirements. Quantitative results of all of our requirements being met are best seen in the data collected from our system running over the span of a few hours, since all of our subsystems – from irrigation to data logging to user input to controls to power – must work correctly for the system to function properly as a whole. Figure 10 shows a screenshot of a text file with data collected on the four pots while running the system. It is important to note the correct time intervals between data logs and saturation values changing as a result of the pots being watered successfully.

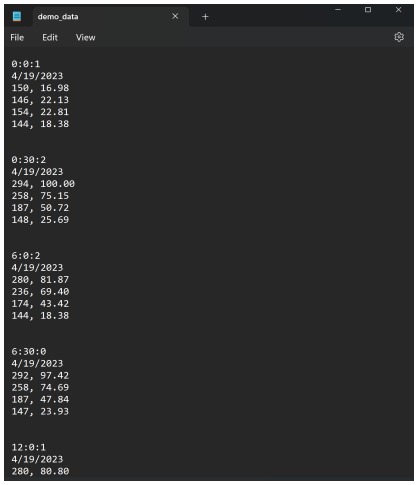

Figure 10: Text file of actual data collected from running system

4. Costs and Schedule

4.1 Parts

The cost of parts for this project is significant. Due to shortages in global semiconductor supply, the cost of many parts of the system have increased dramatically. The list of all parts is shown in Appendix B1. The total parts cost comes to $1,196.83. This is covered by the funding provided by USDA.

4.2 Labor

The average starting salary of ECE graduates is $80,000/year, or $40/hour. Assuming that each team member works 100 hours total, this would amount to ($40/hour) x 2.5 x 100 hours to complete = $10,000 per person. Multiplying this by the number of team members, $10,000 x 3, the total cost of labor for our group would be $30,000. Additionally, we worked alongside a team of 3 ABE students that will be paid separately through their department.

The total project cost amounted to **$31,196.83.**

4.3 Schedule

The overall schedule our team followed for the project is shown in Appendix B2.

5. Conclusion

5.1 Accomplishments

Overall, our group was satisfied with the final product we delivered. More importantly, as a pitched project, it was important for us to satisfy the USDA representative that we were building this prototype for. He expressed his gratitude for our work and is excited to see how it can be utilized with his future experiments. Our main success is that the prototype was implemented and tested successfully inside the experiment's growth chamber, which indicates future success once the experiment starts. Other notable accomplishments were that all of our high level requirements were met and each subsystem worked well with each other as expected.

5.2 Challenges and Uncertainties

Like in industry or any "real-world" project, work on a single product actually spans across multiple teams in many different disciplines. While it was challenging, this project has been a reminder of the world outside of just Electrical and Computer Engineering classes, as our team had to learn how to work alongside another team in a completely different department with different specializations, skill sets, and timelines than ours. This led to communication issues with different expectations per group and updates on meeting times and project issues. However, as a team of six motivated students, and nice people in general, we learned how to compromise.

There was also the constant challenge of our requirements changing throughout the semester. Since the USDA representative still does not know the exact specifications of his experiment, the

project scope started vague and changed as he saw fit. This caused us to have to redesign our system many times throughout the semester, even after parts were ordered.

While we did not have any specific "unsatisfactory" results, the main uncertainty is how our project can be scalable up to more than 40 pots, instead of just the 4 we designed our system for, given the time period of our class. This will be further discussed in Section 5.3 "Future Work." One way we are actively addressing this uncertainty is that our final deliverable to the USDA representative, outside of our required documents for ECE 445, will be to create detailed documentation of how to operate our product in the growth chamber, and plan how we think it can be scaled in the future.

5.3 Future Work

Extension to 40 pots

The original intention of USDA was for the project to function for at least 40 different pots. Our current design is intended to only be able to control the irrigation for four different pots, with all parts of the project being housed on a single PCB. It remains to be studied how the grouping of the different subsystems can be re-organised such that the solution can be extended to any number of pots required. For instance, it might be useful to compartmentalize all components of the solution such that there would only be one PCB required for each pot, so that if the solution needs to be scaled, more PCBs can be printed to replicate the same function. Another solution is that, since we currently designed our PCB to work with 4 pots, it would be efficient to find a way to multiply our work by 10 to accomplish scalability of 40 pots.

User Interaction

The current user interface is a coarse display of the moisture level content in each pot rounded to the nearest tenth shown on a 10-segment LED bar graph. A more comprehensive display of the exact moisture level in each pot, as well as the amount of time left until the pot is irrigated sufficiently, = can be created on a LCD screen which would give the user a better idea of how the irrigation is faring at any point of time.

Calibration to Different Substrate Types

The ECH20 EC-5 sensors used in this project determine the Volumetric Water Content (VWC) based on the soil's dielectric constant and how well the soil holds electric charge. They are sensitive to temperature, ion concentration, and soil bulk density and the specific calibration with chosen substrate allows for an accuracy of ± 1-2%. The substrates that a calibration curve has been developed for are namely a solution with 0.625 mM NO_3^- and a solution with 5 mM NO_3^- (both were also treated with a 0.5-strength solution of Hoagland's, a nutrient blend). A greater range of calibration curves can be developed so that the sensors could detect moisture content in a wider range of soil and substrate types in different applications.

5.4 Ethical Considerations

One of the main ethical standards that was followed in this project is keeping our project ideas original and different from other ideas that are already found on the market. There should not be any plagiarism of product designs and other applicable sources found during the research process should be properly cited and credited. (IEEE Code of Ethics II.5). In the context of our project, other automated soil moisture-based irrigation control technologies that utilize smart watering to tailor irrigation schedules to meet the water needs of the area are present. Careful consideration was made to keep our project idea distinct from these existing technologies, in terms of the methods we use to achieve the implementation of such a system. Additionally, the IEEE Code of Ethics I.5 states that the claims and estimates based on available data should be honest and realistic. As our project dealt with making soil moisture data available for the user to make decisions on irrigation schedules, our team ensured the solution is sufficiently tested such that the data shown by the sensor system is accurate and precise. This includes a comparison of the water moisture level detected by the sensor across multiple occasions such that the measurements made by the sensor are repeatable.

Specific safety concerns in our project were also addressed. Upon review of the laboratory safety regulations as set by the Division of Research Safety in the Office of the Vice Chancellor for Research and Innovation, our team followed these regulations strictly in the lab sessions that we engage in during the semester. Our team would always work in a team of at least two people, report any broken equipment immediately, clean up after each lab session, and avoid bringing any food into the lab (ECE 445 p.3). One of the potential safety risks that could have been encountered during the project was the possibility of short circuits when the electronic components are wired to a power source resulting in damage to all the combustible crops around the soil moisture controller. Accidental skin contact with the circuit with moisture would also increase the risk of dangerous electric shocks through our body. Special attention was paid to ensure that electrical systems are kept entirely separate from any external irrigation systems that release water over croplands.

Finally, our group upheld IEEE Code of Ethics II.7, to treat all personas fairly and with respect. One of the greatest strengths of our group (ECE and ABE combined) was how diverse we were in our backgrounds, races, genders, experiences, passions, etc. We never discriminated against one another and were always open-minded to how our differences benefit our project.

Citations / References

1. Ortiz, Diego, et al. “A Cost-Effective and Customizable Automated Irrigation System for Precise High-Throughput Phenotyping in Drought Stress Studies.” *PLOS ONE*, vol. 13, no. 6, 2018, https://doi.org/10.1371/journal.pone.0198546.
2. ECE 445 Lab. Lab : ECE 445 - Senior Design Laboratory. Retrieved February 2, 2023, from https://courses.engr.illinois.edu/ece445/lab/index.asp
3. IEEE code of Ethics. IEEE. (n.d.). Retrieved January 24, 2023, from //www.ieee.org/about/corporate/governance/p7-8.html.
4. Liao, R., Zhang, S., Zhang, X., Wang, M., Wu, H., & Zhangzhong, L. (2021). Development of smart irrigation systems based on real-time soil moisture data in a greenhouse: Proof of concept. *Agricultural Water Management, 245,* 106632.
5. Team, Arduino. “Guide to Arduino & Secure Digital (SD) Storage.: Arduino Documentation.” *Arduino Documentation | Arduino Documentation*, https://docs.arduino.cc/learn/programming/sd-guide.
6. Agnihotri, Nikhil. “How to Log Sensor Data Using SD and Micro SD Card with Arduino.” *Engineers Garage*, 2023, https://www.engineersgarage.com/arduino-sd-card-micro-sd-card-reader-sensor-data-logging/#:~:text=Reading%20data%20from%20SD%2FMicro%20SD%20card%20involves%20use,number%20of%20lines%20or%20until%20the%20file%20ends.
7. Dahlen, Aaron. (2005) “The PID Controller – Part 1” https://www.nutsvolts.com/magazine/article/the_pid_controller_part_1

Appendix A – PCB Schematic and Layout

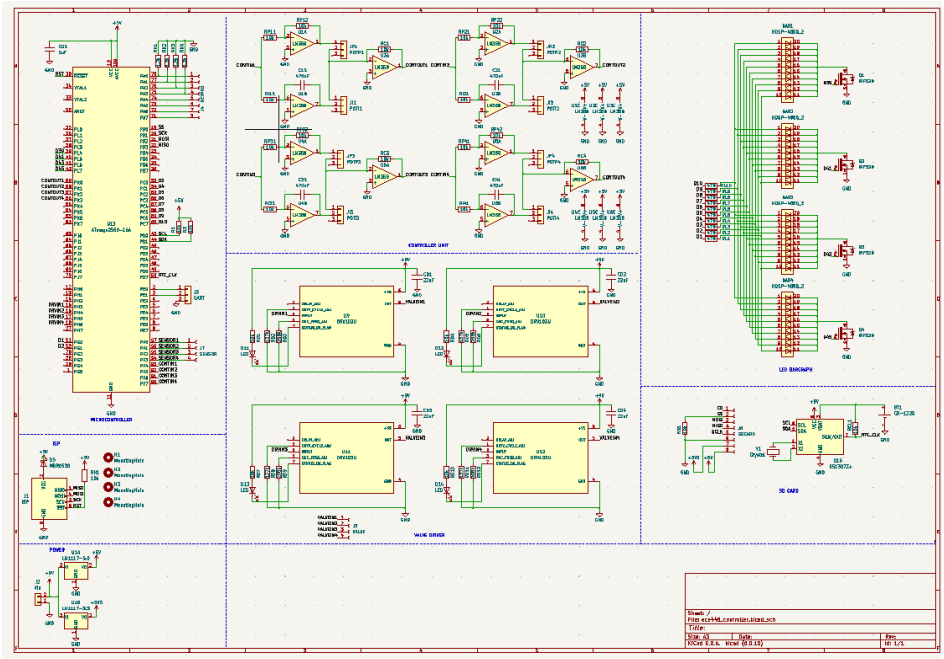

Figure A1. PCB Schematic

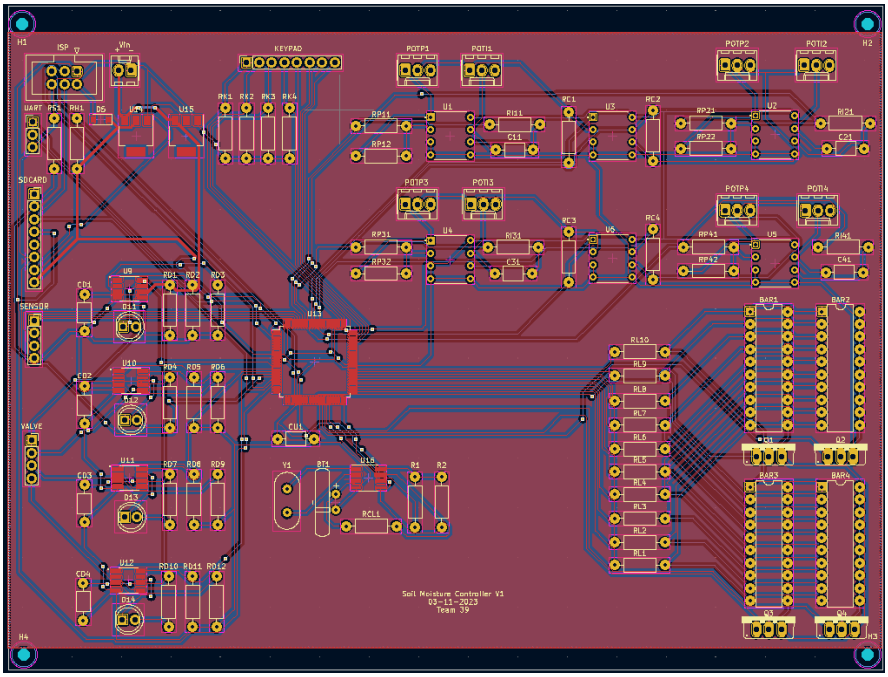

Figure A2: PCB Design Layout

Appendix B – Cost and Schedule

Appendix B1 – Cost

Description	Manufacturer	Quantity	Unit Price ($)	Cost ($)
3/4-inch Solenoid Globe Valve AC 24V	Galcon	4	$31.96	$127.84
3/4-inch Clear Vinyl Plastic Tubing Flexible (10 feet)	Eastrans	2	$11.39	$22.78
40 pcs 304 Stainless Steel Hose Clamp	Selizo	1	$8.99	$8.99
ECH20 EC-5 Soil Moisture Sensor	Meter Group	4	$145.00	$580.00
DS1307 for Arduino Uno Data Logging Shield (3pcs)	AITRIP	1	$10.99	$10.99
32GB SD Memory Card	LaView	1	$8.99	$8.99
10-segment LED bar graph	Lite-On Inc.	4	$1.38	$5.52
4x4 keypad	Parallax Inc.	1	$9.27	$9.27
Arduino Uno R3 ATMEGA328P Eval microcontroller	Arduino	2	$27.60	$55.20
Breadboard	DFRobot	1	$2.90	$2.90
Resistor 10K Ohm 1/4 Watt PTH - 20 pack (Thick Leads)	Sparkfun	1	$1.25	$1.25
LM358DR2G Operational Amplifier	Digikey	12	$0.48	$5.76

NE555DR Timer	Texas Instruments	4	$0.46	$1.84
470 µF 16 V Aluminum Electrolytic Capacitors Radial	Rubycon	8	$0.48	$3.84
10k Ohm Linear Rotary Potentiometer	Sparkfun	24	$1.05	$25.2
9V rechargeable battery	Energizer Battery Company	4	$19.62	$78.48
9V Energizer Recharge 4 Battery Black Universal Battery Charger	ACE Hardware Store	1	$35.99	$35.99
Snap in connector for batteries	SparkFun Electronics	4	$3.50	$14.00
9V to 24V voltage converter	XP Power	1	$88.88	$88.88
9V to 15V voltage converter	Recom Power	1	$28.56	$28.56
180 x 135 mm 2-layer HASL main PCB	PCBway	5	$80.55	$80.55
Total				**$1196.83**

Appendix B2 – Schedule

Week	Task	Team Member
February 27th-March 3rd	**Design Document Check - 2/27**	Everyone
	Start PCB design for the controller	First
	Research on how to connect the microcontroller to 4x4 keypad and LED bar graph for user interface	Ren Yi
	Research data logger connection to microcontroller	Isabel
	Finalize irrigation system design with ABE group; order parts	Everyone
March 6th-10th	**PCB Order - 3/7**	Everyone
	Teamwork Evaluation I - 3/8	Everyone
	Work on a basic prototype of the user interface	Ren Yi
	Test the battery connections to microcontrollers, control system and valves	Ren Yi
	Tune the controller for stability	First
March 13th-17th	Work on a basic prototype of the user interface	Ren Yi
	Test the battery connections to microcontrollers, control system and valves	Ren Yi
	Work on prototype of data logger, reading mock data from one pot of soil	Isabel
	Document the range of input and output of the controller	First
March 20th-24th	Combine the user interface and data logger with moisture sensor	Ren Yi, Isabel
	Combine the irrigation subsystem with the controller	First
	Design container for prototype to fit in for protection in plant chambers; order parts	Everyone
March 27th-31st	**PCB Order - 3/28**	Everyone
	Individual Progress Report Due - 3/29	Everyone

	Combine the sensor with the controller and irrigation system for real moisture input	Everyone
	Scale data logger to read from 4 different pots of soil through sensors	Isabel
April 3rd-7th	Refine prototype to make it more efficient and effective	Everyone
	Build container for prototype to fit in for protection in plant chambers	Everyone
April 10th-14th	Test the soil moisture controller with potted plants	Everyone
	Assemble prototype in container	Everyone
	Film video of working system for mock demo	Everyone
	Fix minor errors	Everyone
April 17th-21st	**Mock Demo**	Everyone
	Fix any remaining errors	Everyone
	Test the soil moisture controller with potted plants	Everyone
	Finalize assembly	Everyone
	Re-film video for final demo	Everyone
April 24th-28th	**Final Demo**	Everyone
May 1st-5th	**Final Presentation**	Everyone
	Final Paper Due - 5/3	Everyone