California State University of Fullerton –

Multidisciplinary Projects in Computer Engineering

EGCP 471

Project Eden: Smart Irrigation System for the CSUF Arboretum Nursery

Team 4

Andy Nguyen

Timothy Trinh

Aaron Nguyen

Salem Edrees

Formal Project Report

Dr. Kiran George

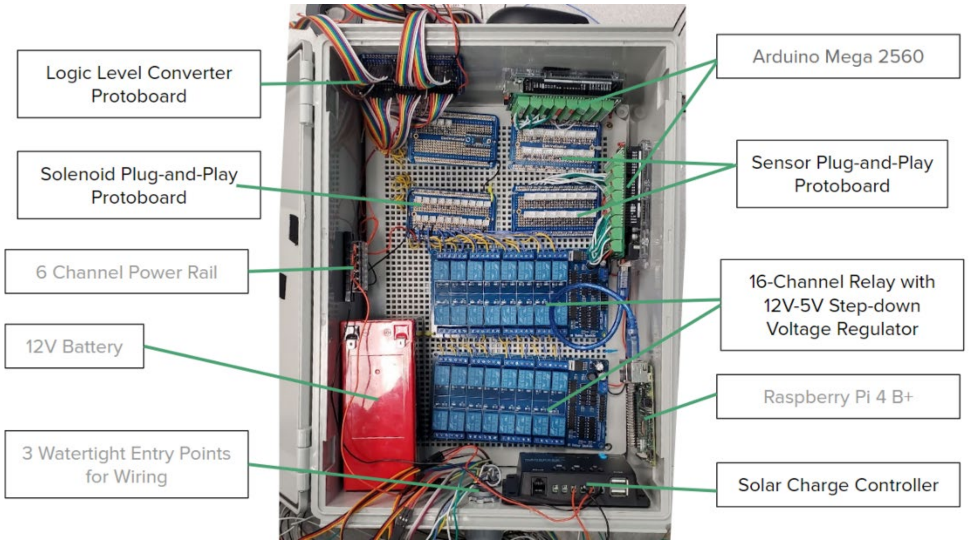
May 25, 2022

Contents

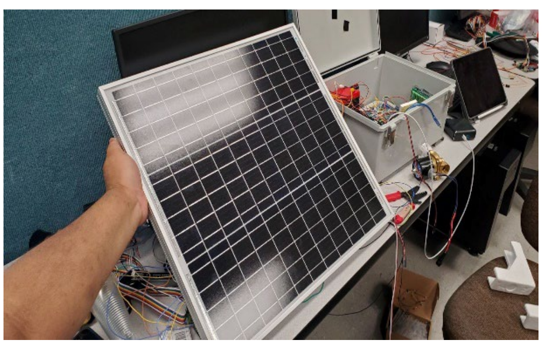
1 Abstract:

Being contracted by the CSUF Arboretum, our senior design capstone project involves the creation of a smart irrigation system that will reduce or eliminate the need for human interaction in the task of watering plants. To realize the design, the planned solutions include a fully customizable watering system with the two options of a user-inputted, semiautomatic watering schedule and a fully automatic, machine learning driven plant watering software. By combining a custom pipe structure, sprinklers, and a series of electric solenoids, it is possible to program and control the behaviors of the watering. This allows specific customizations that can be made to tailor watering patterns for different types of plants and their needs or to account for changes in season when plants require to be rotated out of the arboretum. To facilitate testing, a prototype frame was constructed with three solenoids as a proof of concept for individual solenoid control. While still under development, our prototype is currently performing as expected but will need to be appropriately scaled up before being applied to the desired area at the CSUF Arboretum's nursery section. This full design will be a collaboration between our Computer Engineering team and a corresponding Mechanical Engineering team in order to accomplish the multidisciplinary tasks of manufacturing, electrical work, programming, and project construction.

2 Theory or Background:



Overview of our System



Solar Panel

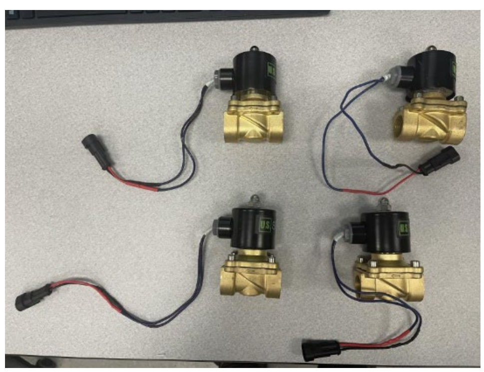
Sensors

After a period of dormancy due to the impact of COVID-19, the Arboretum of California State University, Fullerton reached out to the engineering department requesting assistance in assessing certain tasks to automate due to a heightened shortage of staff, including key members such as the arboretum's horticulturists. The selected and most time-inducive task was nursery plant watering which required up to 4 hours per watering session. To reduce workload, our goal, in collaboration with Mechanical Engineers, is to make a customizable smart irrigation infrastructure that can handle diverse watering needs and monitor output via sensors. In making the design, we emphasized user-friendly interfacing, modularity, and customization as core features. The design is planned to cover around 5 tables, each measuring around 4.83 feet wide and 30 feet long. At a high level, the computer engineering team is responsible for designing the main controls for the watering system, including the operation of the electric solenoids, the collection and synthetization of sensor data, and the creation of an appropriate, intuitive user interface as the agent of control. The computer engineering team is also responsible for the design of the electrical supply and distribution for the smart irrigation system, making sure that every component is properly supplied with the necessary power generated by the appropriate means within the system. On the other hand, the mechanical engineering team is responsible for the design and manufacturing of the physical pipe assembly which houses the solenoids, sprinklers, and control box for the irrigation system.

3 Design Specifications:

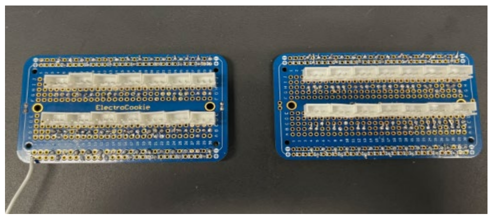
CpE SO(2): An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.

As a basic watering system at its core, our project is inherently stable and safe, containing zero moving parts and having little opportunity for accidents causing harm to people around. The majority of the structure uses PVC, which is highly used and valued for its low cost and high durability suited for outdoor conditions. Our project is powered with solar panels, allowing self-sufficiency and no operating costs at the expense of the Arboretum. The panels store their energy in a lithium-ion 12V battery, which is the greener choice when compared with acid batteries due to having less hazardous materials and a much longer lifespan. Along with lead acid batteries, they also have high recyclability. In making the system safe, modular, and user-friendly, many plug-and-play aspects were added to the hardware design and software design. Solenoid valves initially came with two open wires, however to allow the Arboretum staff the functionality to add, remove, and replace solenoid valves, waterproof connectors were soldered onto the solenoid valves to allow for easy access.



Soldered Waterproof Connectors

To increase modularity for sensors and solenoid valves, JST connectors were soldered onto several protoboard boards to allow easy plugging for Arboretum staff. Logic-level converters were also soldered to protoboards for storage and organization. These protoboards were then soldered to connect to appropriate components across the electronic system like relays, power strips, Arduino Megas, and the Raspberry Pi. In total, over 500+ soldering connections were made, which included soldering wires, logic-level converters, 2/3-pin JST connectors, and waterproof connectors. Shrink tubing was also wrapped around any exposed metal contacts from the soldered wire.



JST Connectors on Protoboards

Since our project is not novel and has well established practices used throughout the world, it is extremely unlikely to create any social or cultural disturbances.

CpE SO(5): An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives.

Considering that the required meetings only occur once a week, much of our communication was done through Discord, a free, accessible text chat application which can be used for collaborative work. The usage of this application allowed everyone in our project group to voice their opinions or share ideas during the development process of our project with ease. This also allowed us to remain updated with the Mechanical Engineering team, who constantly provided different designs to which we all discussed. At the beginning weeks of the project, major tasks were divided among each member to ensure total participation by everyone. Our preferred way of working revolved around freedom and self-responsibility. This means there were no “leaders” in the sense of an individual receiving updates or giving out tasks. Each member worked on their respective tasks whenever possible and without pressure as long as they maintained progress and shared their results. Any conflicts were resolved as objectively as possible.

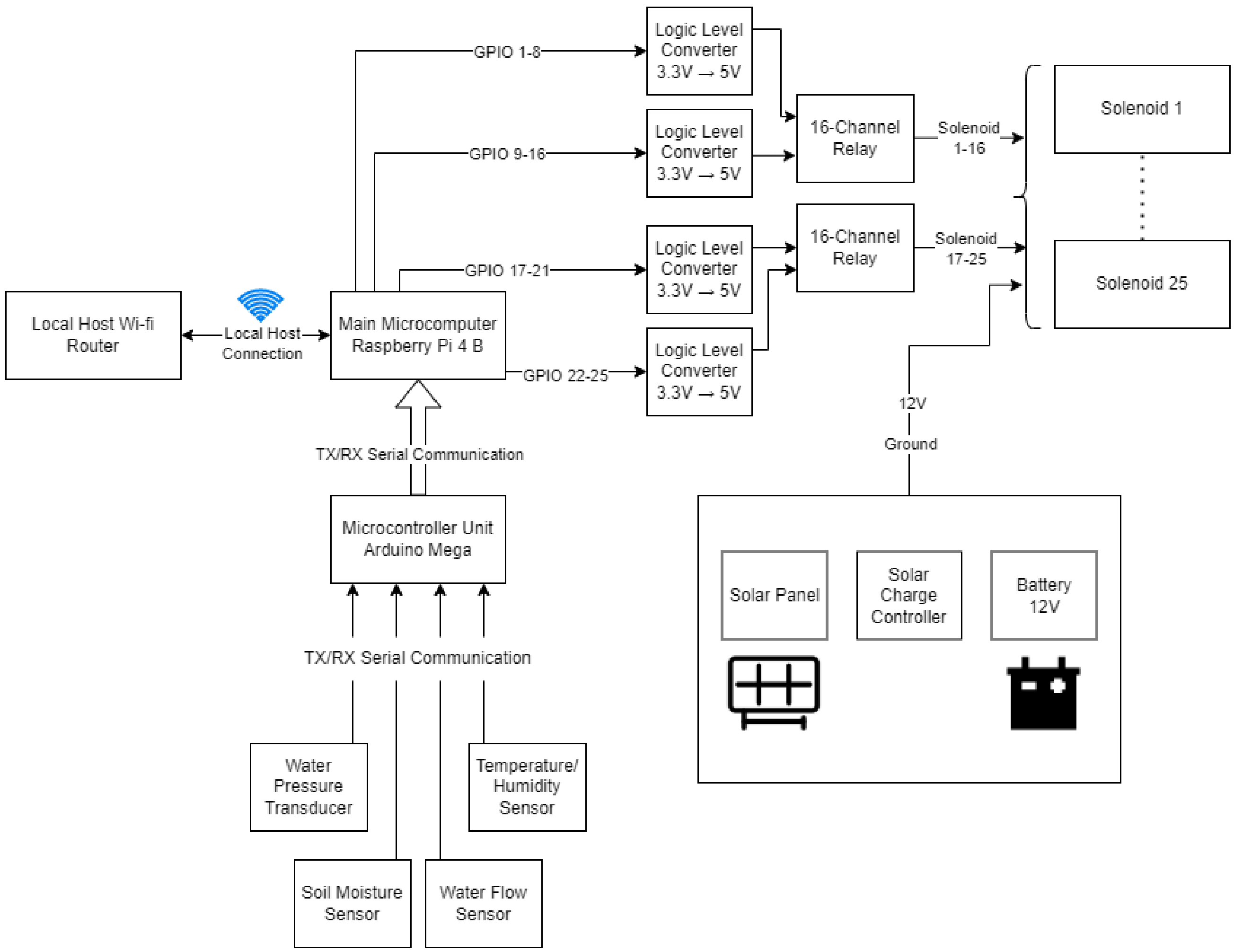
CpE SO(7): An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

This project required knowledge in many subjects that most members were not familiar with. For example, website development in both scopes of its front-end and back-end design, is an untaught skill in our major but is integral for the success of our project. With research and plenty of debugging time, a functional web server with backend capability was created. While it is still undergoing improvements, it provides necessary such as converting user input into digital behavior. In addition, a flexible scheduling system was also added, allowing for diverse and precise watering. Experimental features were tested but were not fully developed, such as smart watering and water prediction, both of which would make up the autonomous watering mode. Electrical engineering concepts were also reviewed and learned due to the necessity of relays and other components, demonstrating foreign device interfacing techniques. Additionally, another concept which required autodidacticism includes machine learning, which is important for our fully automated portion of the smart irrigation system’s design.

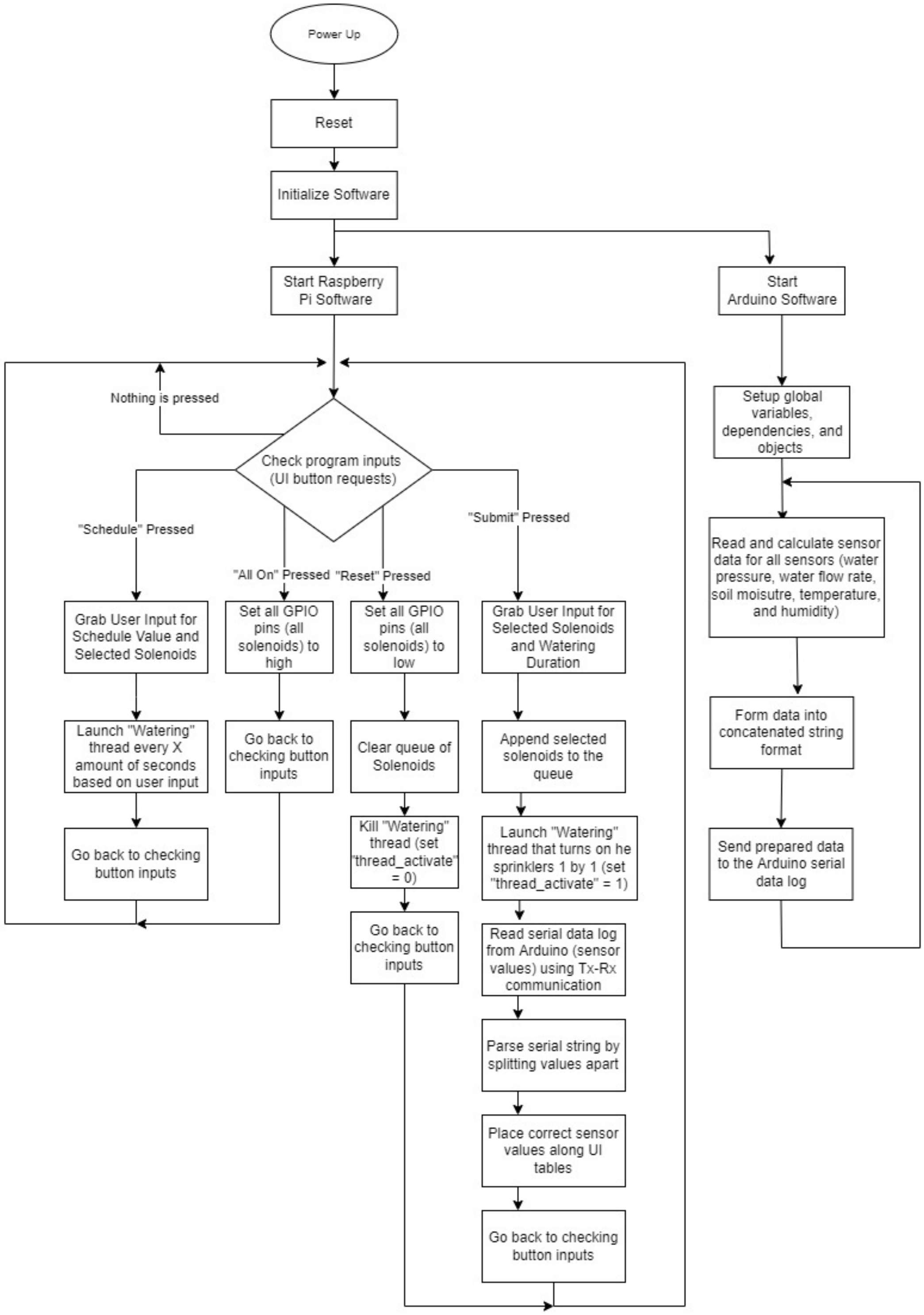
4 Performance Analysis:

The webserver and solenoids are functioning as expected. For example, since we do not know the sprinkler and flow rate, the solenoids are programmed to take time as input. When inputting numbers as seconds, the solenoid correctly powers on for the specified amount of time with imperceptible delay. Our latest model contains 5 solenoids with the possibility for expansion of up to 25 solenoids. On the software side, the UI would be able to program itself additional rows up to the limit, each corresponding to an additional physical solenoid. The program also automatically detects the number of connected sensors. These sensor values are stored in a database as well as a graph, both of which were tested to display real time updates. Each solenoid has its own schedule due to the great amount of plants that it is responsible for. The results, as seen through our video demos, were successful as we were able to operate the solenoids under different settings as well as work with varying numbers of them. In terms of the user interface, the web server was tested using a Samsung smartphone, Apple iPad, and Windows laptop, which all were able to access the local host through their respective browsers. The sensor readings were taken into two Arduino Mega 2560s which successfully transmitted its data to be displayed through the Raspberry Pi through Tx/Rx serial communication channels. Each device is able to interface with the webserver and the system in order to successfully request user-inputted operations. In conclusion, our system successfully demonstrates that it can interface with sensor data, show said data on the website, and control solenoid valves via the website. Our system also successfully demonstrated modularity as all sensors and solenoid valves have been properly developed both on hardware and software side to be easily removed and added by the user.

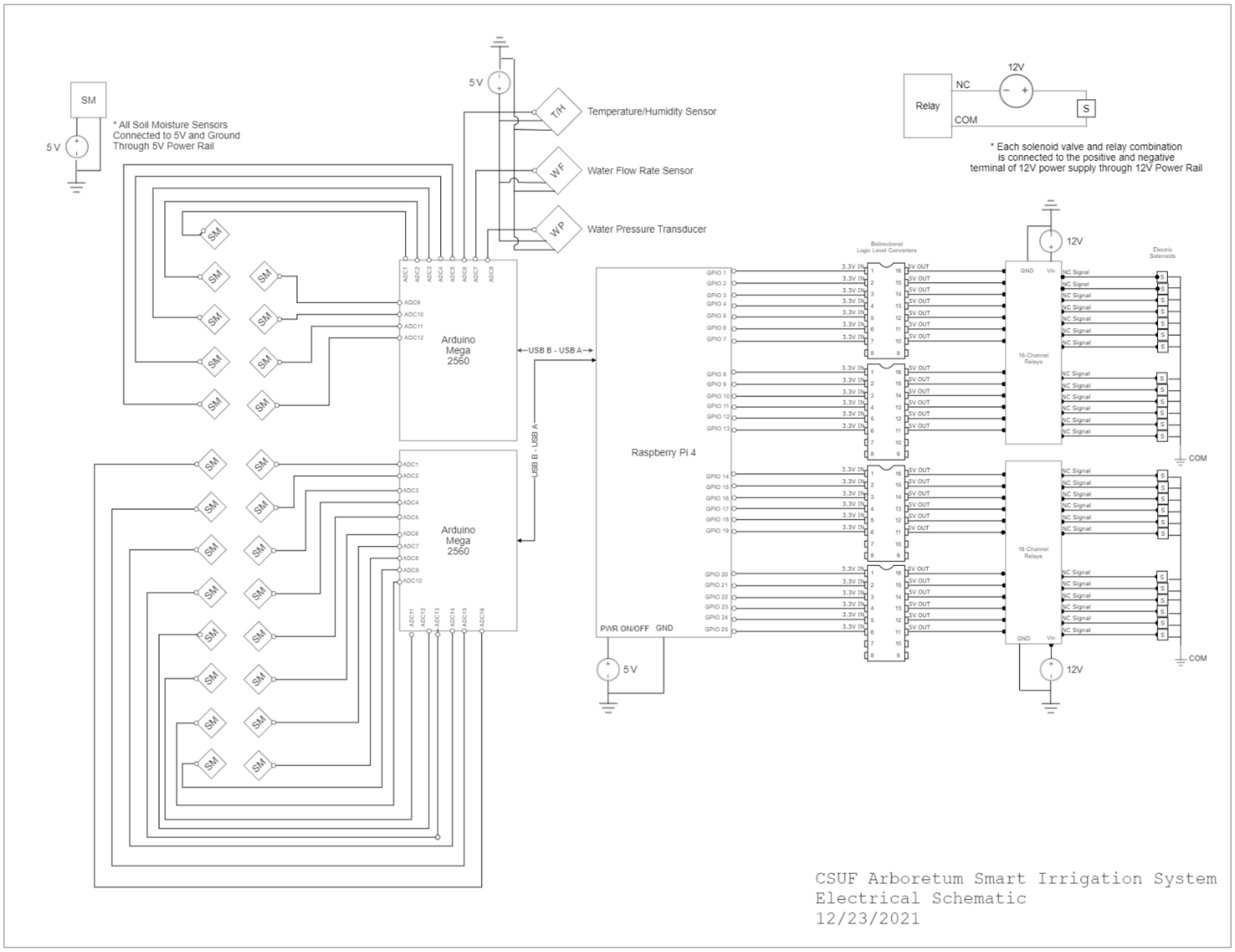
5 Block Diagram:



6 Design Flow Chart:



7 Complete Schematic Diagram:



The complete schematic diagram of the full arboretum smart irrigation system is shown above. At its core, a Raspberry Pi 4 acts as the main microcomputer responsible for handling all the sensor data transmitted by the Arduino Mega 2560 devices. The Arduino Mega 2560 devices are connected to the Raspberry Pi via USB-A to USB-B and their power is supplied by the Raspberry Pi itself. The sensors onboard the Arduino devices, including the soil moisture sensors, temperature/humidity sensor, water pressure transducer, and water flow rate sensor, all operate optimally at 5V and require an analog pin (ADC) to properly interface between the Arduinos and the Raspberry Pi. The Raspberry Pi also controls the operation of the solenoids by activating digital pins which drives the normally closed signals (NC) from the relays to the solenoids. Due to the relays requiring a 5V digital input to operate properly, a bidirectional logic level converter was implemented in order to step-up the 3.3V driven by the Raspberry Pi's GPIO pins to be 5V for the relay. To implement different power sources from the 12V battery source such as 5V, a stepdown voltage regulator (12V – 5V) is used in order to supply devices requiring 5V. These regulators also carry an appropriate amount of current to properly drive the 5V devices.

8 Parts List:

Prototype Parts:

Item	Price	Quantity	Description	Link
Renogy 50W Monocrystalline 12V Solar Panel Kit	$89.99	1	Solar Panel and Solar Charge Controller	Amazon
ECI Power 12V 10Ah Lithium LiFePO4 Deep Cycle Rechargeable Battery	$54.99	1	Main Battery/Power Cell	Amazon
1/4" 12V DC U.S. Solid Solenoid Valve	$39.95	3	Main Solenoids controlled for water flow	Electric Solenoid Valves
ELEGOO 8 Channel DC 5V Relay Module	$10.99	1	Device Driver between Raspberry Pi and Solenoid	Amazon
HiLetgo TXS0108E 8 CH Bi-Directional Converter Module	$8.49	1	Bidirectional logic level converter to convert 3.3 V to 5V	Amazon
Plywood Planks 2’ x 4’	$15.98	1	Platform for Prototype Testbench	Home Depot
PVC Pipes 1/2" - 5 ft	$2.44	2	Support for Prototype Testbench	Home Depot

Cable Zip Ties 20 Pc	$2.48	1	Fastening for Prototype Testbench	Home Depot
1/4" Pipe Fittings	$47.64	1 (total 10)	Tubing connections	Home Depot
Plastic Tubing	$2.96	1	Tubing for water in Prototype testbench	Home Depot
Songhe Capacitive Soil Moisture Sensor 5pcs	$9.71	1	Collecting soil moisture level for data	Amazon
GREDIA 1/4" Water Flow Sensor Food-Grade Switch Hall Effect Flowmeter	$11.99	1	Collecting water flow rate	Amazon
G1/4 Pressure Transducer Sensor, Input 5V Output 0.5-4.5V / 0-5V Pressure Transmitter for Water (0-80 PSI)	$22.09	1	Collecting water pressure	Amazon
DHT11 Temperature Humidity Sensor Module Digital Temperature Humidity Sensor 3.3V-5V	$10.29	1	Collecting local humidity, temperature values	Amazon
Raspberry Pi 4 B 4GB RAM	$55.00	1	Main Microcomputer responsible for controlling all devices	Pishop.us

Arduino MEGA 2560	$38.00	1	Multiple ADC pinouts for analog-output sensors	Amazon
Total	**$505.33**			

Full Design Parts:

Item	Price	Quantity	Description	Link
Renogy 50W Monocrystalline 12V Solar Panel Kit	$89.99	1	Solar Panel and Solar Charge Controller	Amazon
ECI Power 12V 10Ah Lithium LiFePO4 Deep Cycle Rechargeable Battery	$54.99	1	Main Battery/Power Cell	Amazon
3/4" 12V DC U.S. Solid Solenoid Valve	$39.95	25	Main Solenoids controlled for water flow	Electric Solenoid Valves
SainSmart 16-Channel Relay Module	$10.99	2	Device Driver between Raspberry Pi and Solenoid	Amazon

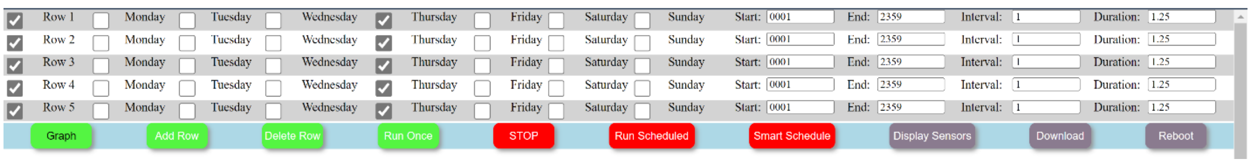
HiLetgo txs0108e 8 CH Bi-Directional Converter Module	$8.49	4	Bidirectional logic level converter to convert 3.3 V to 5V	Amazon
BANKEE 12V to 5V 5A Converter Step Down Regulator	$10.98	1	Stepdown Voltage Regulator to step from 24V sources to 12V devices (main voltage level)	Amazon
Songhe Capacitive Soil Moisture Sensor 5pcs	$9.71	5	Collecting soil moisture level for data	Amazon
DIGITEN Water Flow Hall Sensor Switch Flow Meter Flowmeter Counter 1-60L/min	$11.99	1	Collecting water flow rate	Amazon
G1/4 Pressure Transducer Sensor, Input 5V Output 0.5-4.5V / 0-5V Pressure Transmitter for Water (0-80 PSI)	$22.09	1	Collecting water pressure	Amazon
DHT11 Temperature Humidity Sensor Module Digital Temperature Humidity Sensor 3.3V-5V	$10.29	1	Collecting local humidity, temperature values	Amazon

Raspberry Pi 4 B 4GB RAM	$55.00	1	Main Microcomputer responsible for controlling all devices	Pishop.us
Arduino MEGA 2560	$38.00	2	Multiple ADC pinouts for analog-output sensors	Amazon
Total	**$1398.11**			

* It should be noted that the budget for this project in the scope of Computer Engineering is supplied by both the CpE department and the Mechanical Engineering Senior Design project fund

9 Discussion of Results:

Due to the nature of this project, there is less emphasis on quantitative results when compared to other projects such as those controlling drones underwater. Our goal was to create a UI that is able to control solenoids, work with various sensors, and integrate our components with mechanical team's irrigation system. Due to many Mechanical Engineering set-backs in creating their irrigation infrastructure, we were unable to truly stress test the system as much as we should. However, as shown through our presentation and demo, we completed our own goals with satisfactory results. We were able to stay within our budget as well as make several innovative accomplishments with our design.

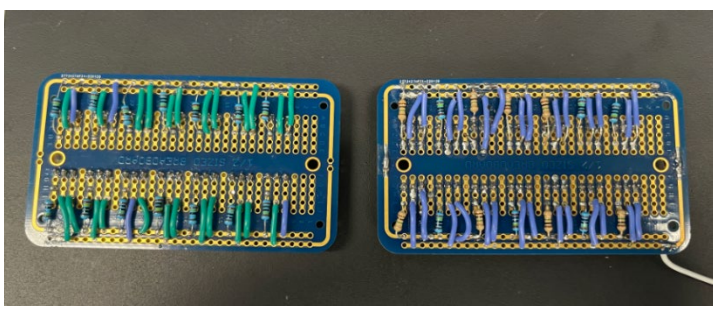


A screenshot of our user interface allowing for precise water scheduling as well as options to expand.

Our electronics integrated with Mechanical Engineering's structure

10 Problems Encountered:

One of the main problems encountered was finalizing a design with the mechanical engineering team as well as with the Arboretum curator, Mr. Greg. For example, the mechanical engineering team wanted to include a pump along with the system to allow precise pressure. However, the curator disliked the idea of the pump possibly due to its carbon footprint and operating noise. Due to this development, the mechanical team had to calculate and design a series of different variations of pipe layouts that would allow proper water flow and pressure throughout the structure. With their simulations, they were able to simulate and test designs without having to physically build any. Another problem that we had was the system not being able to detect when a sensor was plugged in or not plugged. This is due to the analog pins of the Arduino always producing a voltage level similar to a sensor plugged in even when none are. In order to combat this, we had to implement external pull down resistors (1 Mohm) that were coupled with the JST connector signal pins, in order to drive the Arduino analog value down to zero when no sensors were plugged in.



Backside of JST connector Protoboard.

Another persistent problem that we encountered was wiring organization. Although our wires are sturdier and more permanent, the connectors were left detachable to allow for easy reorganization. However, this meant wires could easily detach themselves when transporting the prototype around.

A problem that was beyond our reach were networking problems. Because we were using eduroam, a lot of network settings were not accessible to us. Hence, we worked with what we had, which limited accessibility. For example, users not only had to be on the same network, but also use the same credentials. Also, the eduroam coverage at the arboretum was not as stable or consistent. A solution to these problems would be to set up another network, but this would not be practical as users would have to switch to the new network anytime they wanted to access the UI.

11 Individual Task Assignments

Team Member's Name	Andy Nguyen
Description of Task Assignment	• Helped design and assemble hardware from original full schematic for upscaled, electrical control box • Contributed to wiring for electrical control box and for all custom socket boards • Helped onsite testing as well as purchased components for implementation • Assisted on some pieces of code debugging and functionality for UI
Team Member's Name	Timothy Trinh
Description of Task Assignment	• Focused on UI and interfacing with Raspberry Pi 4, developing features such as scheduling, input processing, local hosting, communication with Arduino, expandability and automatic IP retrieval • Helped test and debug prototype

	• Assisted other members in other areas of the project (brainstorming, organizing reports, research etc…)
Team Member's Name	Aaron Nguyen
Description of Task Assignment	• Researched and implemented finalized design (component wiring and soldering 500+ connections) • Helped design schematic and purchased components for the project • Helped make Arduino code that outputted information read from the 4 types of sensors to the Raspberry Pi • Assisted in project ME group meetings, reports/presentations, budget lists, and arboretum implementations
Team Member's Name	Salem Edrees
Description of Task Assignment	• Assisted in wiring

12 Conclusions:

The designs and plans of the original semester came to fruition in the second phase of the capstone project year. Using the original schematic as well as a constructed timeline to handle both the software and hardware progression. In terms of the software, great strides were made in order to carry out the high specificity and modularity outlined by the Arboretum curator. A great amount of time was spent on the full stack development, but mostly for the backend. The backend was modified to handle object-oriented programming for better modularity, and many features were added such as a dynamically changing graph for all sensors, automatic Wi-Fi handling for changing IP addresses, and the ability to receive phone messages through text for notifications on connecting to the UI. The core functionality as well as the additional features were successful in execution. In terms of the hardware, the original schematic was realized fully and translated in the form of the electrical control box that was mentioned previously in the report. Multiple custom PCBs were drafted and created through the usage of protoboards, wiring, and other components which were all individually hand soldered. Other custom parts including the solenoids were also modified to accommodate for the changing system as well. The system was successfully running with remote operation between the UI and the control box, allowing for the system to successfully be complete.

13 Future Work

There is still additional work to be done as well as more bonus features that can further enhance the smart irrigation system. The core functionality of the system for basic water scheduling and sensor irrigation was complete with remote accessibility. Additionally, in terms of full modularity, the system can dynamically change both with hardware and software. However, to further increase the system’s smart capabilities, the system can have more machine learning integrated such as a full dataset on predicting water usage. Security measures can also be integrated into the system such as implementing a security camera for detecting and warding off intruding animals and unauthorized individuals. Optimizations can be made for the smart adjustment watering such as discussing with the onsite horticulturists for more details on how to adjust watering based off of external variables. As of now, the system adjusts the watering based off our own research and prediction on how much watering should be added/subtracted, but the full insight of the horticulturist will be more intimate for the purposes of the adjustment algorithm. Another way to improve the system is to further increase the system's durability to the elements by encasing each wire with the solenoid and sensors. This will double down on improving wire management and protect the system at the cost of more materials.

14 References

"Why most Arduino Soil Moisture Sensors suck (incl. solution)", *Youtube,* uploaded by Adreas Spiess, 16 June 2018, https://www.youtube.com/watch?v=udmJyncDvw0

Band, Amey. "How to Find the Optimal Value of K in Knn?" *How to Find the Optimal Value of K in KNN*, Towards Data Science, 10 Apr. 2021, https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb.

Navlani, Avinash. "KNN Classification Using Sklearn Python." *KNN Classification Using Scikit-Learn*, Datacamp , 2018, https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn.

Reljič, Andrej. "Smart Irrigation Controller: Why Use Smart Watering for Your Garden." *Qubino*, 22 Nov. 2019, https://qubino.com/smart-irrigation/.

Domínguez-Niño, Jesús María, et al. "Differential Irrigation Scheduling by an Automated Algorithm of Water Balance Tuned by Capacitance-Type Soil Moisture Sensors." *Agricultural Water Management*, Elsevier, 7 Nov. 2019, https://www.sciencedirect.com/science/article/pii/S0378377419315641.

"Grow Room Automation & Automated Grow Systems." *Autogrow*, https://autogrow.com/.