Ultrasonic Automated Watering System

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia
In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science in Electrical/Computer Engineering

By
The Plant Whisperers

Fall 2023

Technical Team Members:
Alex Morris
Audrey Swart
Kate Van Meter
Sophia DeCleene

On our honor as University students, we have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

ADVISOR
Adam Barnes, Department of Electrical and Computer Engineering

Table of Contents

Table of Figures

Table of Tables

Statement of Work

Sophia DeCleene:

For my contributions to the project, I was in charge of everything related to the printed circuit board (PCB). This included the design of various subcircuits within the overall PCB. I designed the first couple iterations of the bandpass filter, the power supplies circuits, and the I-amp circuitry. These were all important tasks as they are what allow us to take in a signal from the microphone then conduct all the processing of the signal which dictates whether or not the plant being observed needs to be watered. I also designed a version of the PCB itself, however we ended up using Audrey's design. Along with the design of the circuitry and PCB comes testing to ensure the designs work. I conducted testing for all parts of the PCB in Multisim as an initial test and also assisted Audrey with testing the filters on a breadboard. Finally, I helped to solder components onto the PCB so that our final design could come together and we could have a signal processing system in place.

Alex Morris:

In terms of my contributions to the project. I was in charge of making the user interface needed to interface with sensors. I created a website that can be accessed through the user's laptop or phone where they can see their plants on the home page and see real time sensor data. The user can click the get started button in order to know how to enter a plant into their virtual greenhouse which will lead them to a plant submission form as shown the user fills in the details needed to take care of the specific plant including entering light preferences, temperature conditions, and ideal water schedule. In terms of the programming languages going into the website I used HTML, CSS, JavaScript ,PHP and MySQL. I created 3D printed models for the flower clip where the microphone is placed to direct the sound to the flower, a clip for the photoresistor, and the box to put the board and circuits into. Additionally, I assisted Kate in hosting the web server on the myRIO. Lastly, I was also responsible for updating the Gantt Chart and assigning tasks for each of us to do each week.

Audrey Swart:

My contributions to the project were mainly in regards to hardware. I worked with Sophia on the filter, the PCB, and soldering our components. I designed the final version of the filter after a great deal of troubleshooting and designing many iterations of varying types of filters. We also used my PCB design for the board that got manufactured. I also soldered the microphone. My work was important to the project because it allowed for the signal processing of the sounds the plants emit so we could use that to trigger the water pump to water the plant.

Kate Van Meter:

My contributions to the project were mainly related to the microcontroller part of the design, for which we used the myRIO. I used LabView to code the recording and handling of data for the different sensors, the coordination of the turning off and on of the water pump, and set up the NI web server to transfer the data from the code to the website. I also created the circuits for all of the sensors and the water pump/relay.

Abstract

Our technical project is focused on the design and implementation of an automated plant watering system that uses an ultrasonic microphone for precise and efficient plant hydration. Conventional methods of plant care often fall short in maintaining the delicate balance between underwatering and overwatering, leading to resource wastage and compromised plant health. The core objective of the watering system is to revolutionize plant care by creating a smart and adaptable solution capable of accurately addressing the hydration needs of plants. The heart of this project is the ultrasonic microphone sensor, which will detect sound made by plants when dehydrated that are outside of the human hearing range. We plan to replicate an ultrasonic microphone by purchasing a microphone and adding additional components to filter out the undesired frequencies. Additionally, a peristaltic dosing pump connected to a myRIO microcontroller will work alongside the ultrasonic microphone, delivering precise amounts of water when it is needed, thus eliminating the risks of overwatering and underwatering. A user-friendly interface accessible through a web portal will empower users to effortlessly monitor plant hydration levels, receive real-time notifications, and tailor watering schedules to their preferences. The automated plant watering system also prioritizes energy efficiency and eco-conscious components to minimize environmental impact.

Background

The solution most people offer upon hearing about forgetful people dealing with forgetting to water their plants is to set up an automated watering system on a timed schedule. However, plants are living things and do not thrive when they are watered on a strict schedule. Their water and nutrition needs change based on the seasons, light available to them in different areas, and how much a plant grows in their roots relative to their stems/leaves. The more intuitive approach to caring for plants involves paying attention to the physiological signs a plant shows when they start to become distressed. These are things like drooping stems, leaves curling inward, browning tips, and even dropping plant matter. The problem with waiting for these signals is that they indicate the plant has already begun to feel significantly distressed. Our solution aims to strike a balance between the automated and intuitive approach to plant watering, where the signal to water is based on an inadvertent scientific process that occurs when a plant first begins to show that they need water. As observed in the study by Itzhak Khait, plants produce popping noises at frequencies between 40kHz to 80kHz (outside of the human hearing range) when in distress, like being injured or dehydrated [1]. This popping sound is due "to air bubbles forming and bursting in the plant's vascular system in a process called cavitation" [19].

Using this concept to assist with plant care has not been widely applied to projects outside of academia. The actual watering action in the system will be completed by a water pump, where we will build on a previous project done by a student named Sonia Aggarwal [20]. Her project is on a Plant Ladder System where she included a physical water dispenser system that used a soil moisture sensor and a mobile application. We plan to use temperature, light, moisture, and water level sensors to monitor the plant environment. The main difference between ours and other designs is we plan to use an ultrasonic microphone and process the data using a myRIO microcontroller (a portable device that can be easily used for embedded WiFi solutions). The myRIO integrates the plant environment monitoring sensors to the website. When we host a web server through the myRIO that allows us to store and visualize the sensor readings from anywhere and water the plant remotely. The heart of the system is the ultrasonic microphone that uses a filter and instrumentation amplifier to filter out unwanted noise to isolate plant sounds accurately. In order to test the microphone we used an ultrasonic pest repeller and an ultrasonic emitter to stimulate a sound at the same range as the plant. These devices helped us determine we could detect signals coming from plants.

The volume of the plants is estimated to be around 60 dB, similar to a human conversation, which means that it should be able to be easily picked up by our ultrasonic sensor. Our coursework in Fundamentals II has prepared us well for this project since the final project was designing a circuit that was able to cascade several different types of filters to create a system that could isolate sounds in a certain frequency range. The semester-long project from Fundamentals III will also be of use to us, specifically the portion of the circuit where we had to amplify the signals to get them to a detectable range. Additionally, digital signal processing (DSP) concepts from the DSP class itself as well as the related unit in Fundamentals III will be helpful when we will need to refine our filtering to ensure our system only picks up sound coming from the actual plant, not any other noise.

Societal Impact Constraints

There are many different societal impact constraints that must be considered with respect to our project of an automatic plant-watering system. One consideration stakeholders must keep in mind is the water quality of the water being used to hydrate the plants. This is an environmental factor outside of the design's control which must be considered for successful use of the product. The quality of the water used is very important. Polluted water can harm plants; factors such as: salts, pH and alkalinity determine the suitability of water for use on foliage and flowering plants. Similarly, water conservation is an environmental factor to keep in mind. This is controlled with our design by not giving the plant too much water from the water pump at once. Thus only the water needed is used and water conservation is being observed. Another consideration regarding safety is the potential proximity of water to electrical components. Our design minimizes the risk of these interacting in a negative way by storing the majority of the electrical components and devices in a wooden box with a lid to protect what is inside from potentially harmful external factors.

Some considerations with respect to cultural, social, and economic factors include the following. Visual appeal can be a social and cultural factor as the device might seem out of place and cause people to be judged for having one. Our team has attempted to minimize these feelings by having a clean design with a box to hide all the inner workings of the project and by having a decorative yet functional flower clip the microphone attaches to so it can be placed near the plant. Other considerations include accessibility and affordability which seem to go hand in hand. The use of the myRIO microcontroller alone makes the cost of our project extremely expensive, making it unaffordable to many people and therefore inaccessible. As far as accessibility in the sense of who can and cannot use our design, the majority of people would easily be able to use our design as all they have to do is check when the software deems the plant dehydrated and give it the command to water the plant. This is something easily manageable by most adults and even some children, so the user base is quite large for our design.

Physical Constraints

Below we will be discussing the constraints of our project when it comes to cost constraints, tools utilized, design and manufacturability. These constraints will limit how we are able to build our prototype so it is important to address these challenges now.

Cost Constraints:

In the research study, the scientists used a CM16/CMPA condenser ultrasound microphone which costs around $3,000 [9]. Other high-quality ultrasonic microphones are also very expensive, making them unreasonable for use in our project. For this reason we looked into alternatives for the type of microphone needed and tried to replicate the design [11]. There are a wide variety of transducers to choose from that are relatively affordable and can easily be bought off the internet. From the several options researched, the cheapest alternative that will still pick up sounds in our desired frequency range is a MEMS microphone found on DigiKey. Another additional cost constraint we ran into was 3D printing costs. When trying to print a 8" by 10" box to fit all the circuitry, sensors, and boards one of our options was to get it printed at the Mechanical Engineering Building which would have cost us $274 so we decided to use a wooden box instead. Additionally, we ran into costs for accessing the content for some of the standards related to/used for our project such as the ISO 14000 series of standards.

Design and Manufacturability Constraints:

Our biggest challenge started in the beginning due to the limited documentation on how to make an ultrasonic microphone or how to use it to listen to the ultrasonic sound. Since the CM16/CMPA ultrasonic microphone was only available through Avisoft Bioacoustics, an international company, meaning it would take time to receive the part and is also extremely expensive, it was not feasible to use this product and we had to design our own ultrasonic sensor. When trying to manufacture the prototype of the ultrasonic automated water system we ran into some challenges that included power management, integration of multiple components, and microphone failure.

While we used a variety of different softwares there was some software which was not compatible with everyone's devices. The G Web software development, a software installation package needed to run a server on the myRIO, was only available through Microsoft devices and not Macbooks. This meant only one of the software-oriented group members could work on the server needed to connect the sensor data to the main web page. Also, Labview does not run smoothly on Apple devices and the correct version must be installed. The availability of free 3D printing machines is scarce at the University of Virginia because most of them require training and special permissions, and do not provide enough time to print larger components. Also, the size of the box we wanted to print was too large for the printing platforms of most standard 3D printers.

We also ran into some issues with the PCB design and construction. The microphone we are using to pick up ultrasonic signals is an extremely small component and is not found in the Multisim database. Thus, in order to get the microphone on the PCB we would need to create a custom component footprint in Multisim to be exported into Ultiboard, and we could not figure out how to successfully do this for the microphone. Therefore, we had to carefully solder small wires onto the microphone so we could power it, connect it to ground, and measure the output. Since the microphone is so small this proved to be quite difficult and took multiple iterations to perfect. Additionally, this setup is very sensitive to breaking apart which constrains our design as people have to be cautious around the microphone so they do not knock wires out of place since it is not secured onto a PCB but rather is freely exposed. Another issue we encountered regarding the PCB was that we had ordered the specific op-amps and I-amp we needed for our PCB design, but when they arrived we discovered they were the incorrect size and would not fit on our PCB. We managed to order the correctly sized op-amps but had to order an IC adapter socket to get the I-amp on the board since we could not find any I-amps in the correct size that had the correct specs.

Tools Utilized:

In terms of the tools utilized we used a mixture of software for programming the user interface, 3D modeling, hardware simulation, and design testing. For the software side of our project we decided to create a webpage for the user interface that the user can access through their laptop using their local network. In order to achieve this we used the following as the front end programming languages: HTML, CSS, and JavaScript. We used these to create the layout and design of the website. Additionally, PHP and MySQL were used for the backend functionality which is in charge of the form handling and storing plant condition information in the database. Lastly, we hosted a web server through the myRIO using Systemlink Cloud and an API key that allows us to store and visualize the sensor readings. For the embedded programming we needed to learn Labview because it was useful in communicating with sensors and accessing the real time data. For the hardware side, we first used Multisim to simulate the behavior of the circuitry we designed. Once the functionality was verified in Multisim, we could create a printed circuit board (PCB) of our design using Ultiboard. We then used Waveforms to test the design on a breadboard by using it for power and to measure the signals that the microphone was picking up as well as the output from the designed ultrasonic sensor. This allowed us to experimentally verify our design. Lastly, for the assembly side of our project we used Fusion 360 in order to create a 3D printed model of a flower clip to place the ultrasonic microphone into. This attaches to the pot of the plant so the microphone is directed towards the sound coming from the plants so that it will minimize sounds picked up from the entire room where the device is being used.

External Standards

Ensuring that the automated watering system complies with relevant safety and certification standards can be a constraint. This may involve additional testing and certification expenses which is why it is important to make sure all aspects of our design meet certain standards. For our PCB design, this project made use of IPC guidelines and standards for board design including part spacing and track spacings, and also for soldering and component protection (IPC-4761) [5]. In regards to using battery power for the project, it needed to meet NEMA standards for portable cells and batteries (ANSI C18.4M) [7]. With respect to the automatic watering system, this project needed to meet NFPA standards for GFCI protection due to the proximity of electrical components to water (NFPA-70: NEC) [6]. Our setup is such that we maximize the distance between where the water will flow, or accidentally flow, and the hardware. In regards to the software and wireless communication aspect, the project needed to meet the standards from IEEE-802 [8].

The IEEE standards for sensor interoperability needed to be observed with respect to connecting sensors to the microcontroller and allowing the sensors to operate simultaneously [16]. We structured our design so each sensor gets its own power supply and does not interfere with the others. The ISO 14000 series of standards also came into play with our design. ISO 14001 concerns committing to environmental responsibility [17]. We accomplished this by considering the plant's needs and using low-power designs to not waste electricity. Something that must also be considered when using our design are the World Health Organization's guidelines on water quality to ensure the plant is getting nutritious water.

Intellectual Property Issues

There exist many different patents for differing designs of automatic plant-watering systems. These tend to show similarities to our design, but ours offers new functionality. Some of these patents are described in the following.

US patent US20070089365A1 [21] describes a plant-watering system that uses a controller to water the plants. The controller communicates with the watering device and activates a pump that waters the plants. It is also mentioned that this design may include moisture sensors that would send back data to the main controller and adjust the water output or generate a warning. US patent US3758987A [22] is another automatic watering device for plants. It is different because it uses a porous sensing device to determine the moisture content of the plant's soil and water the plant from the water reservoir based on feedback from that sensor. The US patent US262379A [23] is a third design for an automatic watering system for potted plants. It also includes a water reservoir but it collects rainfall for the water and has "downward extensions" for water to flow through.

Based on these patents, we believe our design is patentable. We believe this because there are many similarities across already-patented designs. For example, many of these designs involve using some type of sensor related to the soil the plant is kept in to detect when to water the plant. Our design uses a moisture sensor as patent US20070089365A1 does, however we offer many new components as well. Particularly the ultrasonic sensor component of our design is unique and due to that piece, our design is different enough from existing designs to ensure it is patentable.

Project Description

Performance objective and specifications:

The system aims to facilitate optimal plant care by using ultrasonic sounds from the plant as indicators of plant distress and utilizing a user interface accessible through a web portal that enables remote monitoring and control. When the ultrasonic sensor detects the sound made by the plant, it sends a notification to the user. The user can then decide to water the plant manually by looking at the data from the light, temperature, and moisture sensor then using the water button on the web page the user can trigger the water pump. This option will ensure accurate water delivery by monitoring soil moisture levels using the moisture sensor data. This approach optimizes plant care by responding to actual plant needs. The ultrasonic sensor uses a microphone to capture these sounds and processes the signal first on the PCB design itself, then on a myRIO microcontroller, which is a portable device that can easily be used for embedded Wifi-enabled solutions that allow the implementation of multiple design concepts.

How it works:

The ultrasonic sensor consists of three main parts: a microphone, a filtering system, and an instrumentation amplifier (see *Figure 1*). The first component is the microphone which picks up the ultrasonic sounds from our plants. We needed to use a specialized microphone that is able to pick up frequencies above the human hearing range for this. Based on functionality, price, and availability, we used a MEMS (Micro-Electro-Mechanical System) microphone. The sensitivity and size of the microphone was carefully selected in order to optimize the dynamic range and frequency range – a smaller diameter can measure higher frequencies but is not as good at picking up lower decibel sounds. Plants emit sounds at about the same decibel level as a human conversation, which is around 60dB, thus the MEMS microphone is a good fit. The microphone will be placed on a clip that attaches to the rim of the plant's pot so it is close to the plant and can pick up the sounds from the plant better. In testing, we determined the sounds from the plants will probably produce roughly an 80mV signal out of the microphone. The signal from the microphone will be fed into the filter. This is an op-amp based design with a buffer stage. The gain and Q values were calculated to maximize performance. The gain of the entire filtering system ended up being roughly 4 (it was calculated to be unity gain as seen in the Mathematical Analysis section, but the lack of exact component values late in the semester caused a slightly higher gain). It is a bandpass filter with a passband of 35kHz to 80kHz. This will isolate the desired ultrasonic frequencies from other sounds. Steep rolloffs are desired for this project, so we designed higher-order high and low pass filters to comprise the bandpass filter. Once the signal is filtered so we get only the desired frequency range, the signal will go into an instrumentation amplifier with a gain of 10 to amplify the small signals to a signal with several volts of amplitude before it is fed into the myRIO.

Figure 1: Block diagram of ultrasonic sensor

Mathematical Analysis

For the bandpass filter design, we used the simplification shown in Figure 2 which was provided in the Sallen Key Architecture document [24]. This involved setting the filter values as ratios and the gain as 1. We separated the filter into the first half, a second order high pass filter, and the second half, a second order low pass filter, to achieve an overall fourth order Chebyshev response. The values calculated and equations can be seen below. The Q values were determined using a document on active low-pass filter design [25].

$$f_c = \frac{1}{2\pi RC\sqrt{mn}}$$

$$Q = \frac{\sqrt{mn}}{n+1} \text{ for high pass, } Q = \frac{\sqrt{mn}}{m+1} \text{ for low pass}$$

$$R_1 = mR,\ R_2 = R,\ C_1 = C,\ C_2 = nC$$

Stage	**High pass 1**	**High pass 2**	**Low pass 1**	**Low pass 2**
Q value	1.0765	5.577	1.0765	5.577
Resistor ratio (*m*)	8.01	124.4	1.266	1
Capacitor ratio (*n*)	4.7	1	4.7	124 ≈ 100
Corner frequency	35 kHz	35 kHz	80 kHz	80 kHz

Table 1: Filter Calculation Results

After going through the calculations, this simplification resulted in the values seen below in Figure 2:

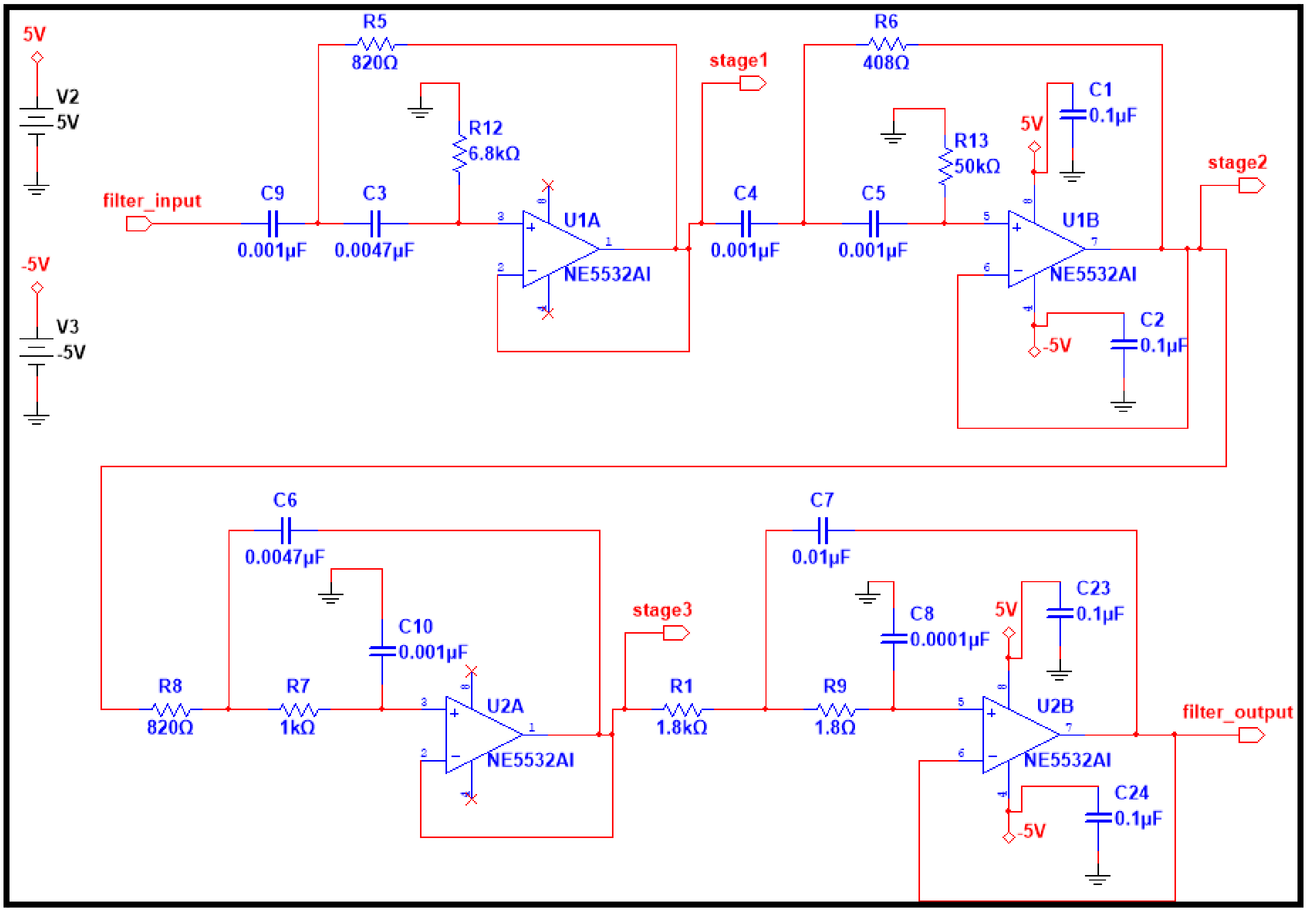

Figure 2: Filter Design

We also had to design a power circuit (see *Figure 3*) to properly distribute power amongst all the components that needed it. We used linear voltage regulators to reduce voltages down to their desired values as well as an op-amp with negative supplies and a voltage divider to produce the negative voltage needed to go into the negative input terminal of the I-amp.

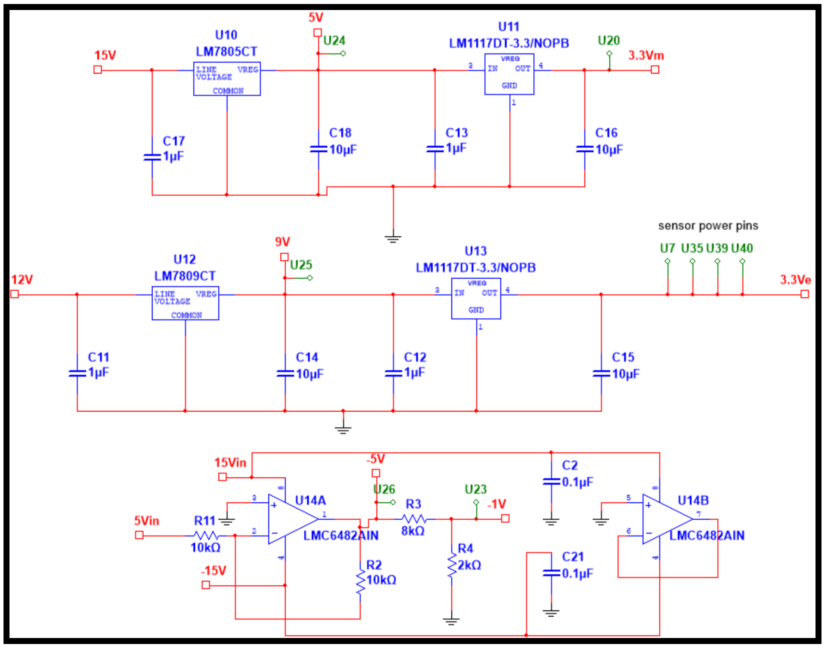

Figure 3: Power Circuit Design

These circuits came together in Ultiboard as seen in the board layout in Figure 4 below.

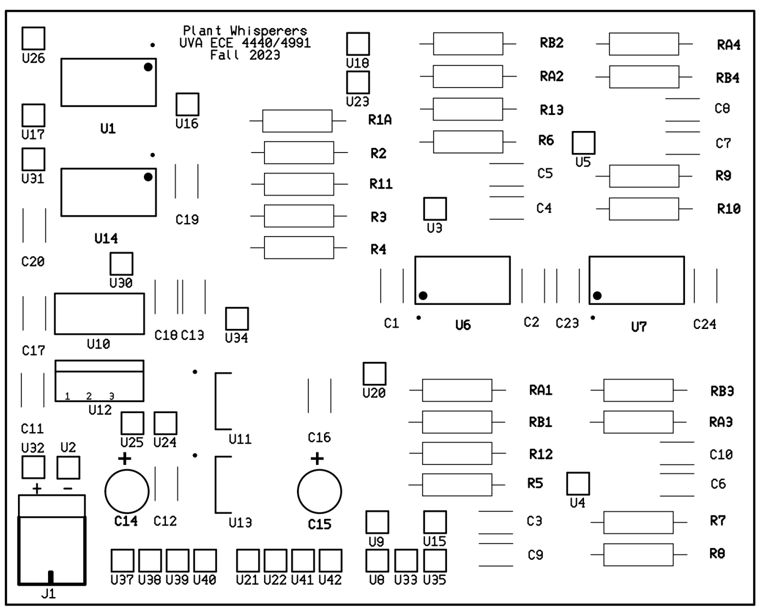

Figure 4: Final PCB Layout

To obtain a gain of 10 on the instrumentation amplifier as discussed previously, we used the formula (below) from the INA849 datasheet [26] and determined that RG should be 666 Ω. With the component values available to us, we used a 560 Ω resistor, making the gain just slightly higher.

$$Gain = 1 + \frac{6000}{R_G}$$

How it works continued:

An overview of the entire system can be seen below in Figure 5. The main sensors we used were a photoresistor which is a light-controlled variable resistor, a TMP36 analog temperature sensor, a 12V DC Dosing Water Pump Peristaltic, an Analog Capacitive Moisture Sensor, and lastly a water level sensor. All those sensors are then connected to the myRIO analog input/output pins and each is supplied the required power through the microcontroller. In order to power the myRIO we connect a power cable to the AC power source. For the water pump we supplied 9V and the rest of the sensors only needed 3.3V. Once the sensors are connected to the myRIO, then using the WiFi capabilities we can connect the system to the website. The website can be deployed using XAMPP which is used to manage servers and access database information. Once XAMPP is running a localhost server can be used to run the web pages which will then display the HTML/CSS files. In order to access the mySQL database information PHP is needed to fetch the dynamic information that was inputted by the user on the plant form.Then Javascript will be used to handle form errors and user actions. All the code used for the website was pushed to a github repository that can be accessed in the Appendix (p.31).

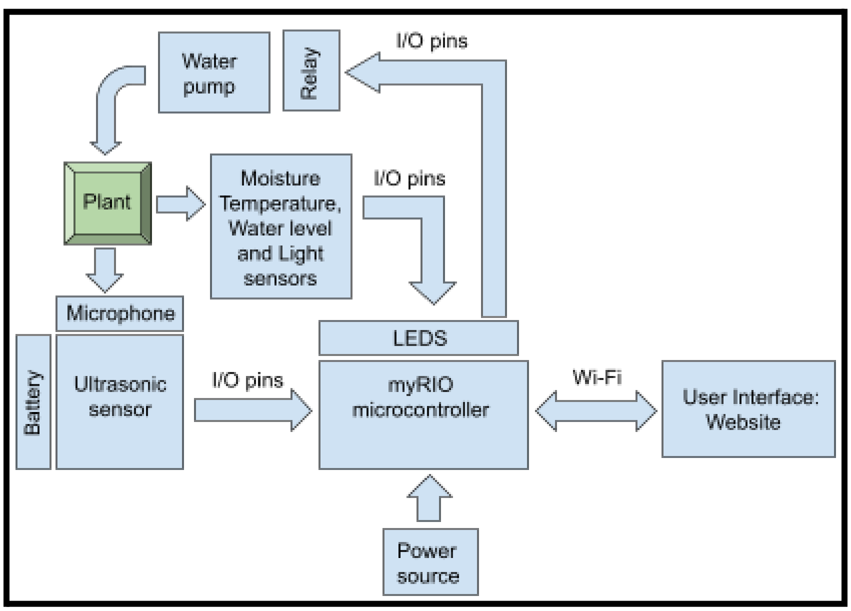

Figure 5: Block diagram of full system

We will now get in more depth of the myRIO interface and how to connect it to the sensors. We used a LabVIEW graphical diagram (see *Figure 6*) to access the sensor and then created string variables to display the data on the myRIO built-in User Interface and also created a button to turn the pump on. When the button is active high then it will switch on the relay which will then supply the respective power to the water pump, otherwise it will stay turned off. As indicators we decided to use the built in LEDs on the myRIO to notify the user when the plant has been sufficiently watered, not watered enough, or if the temperature is too high. In order to view the sensor data shown on the myRIO on the website we used hyperlinking to reference the URL of the global NI web server which is accessed using either an API key or HTTP request.

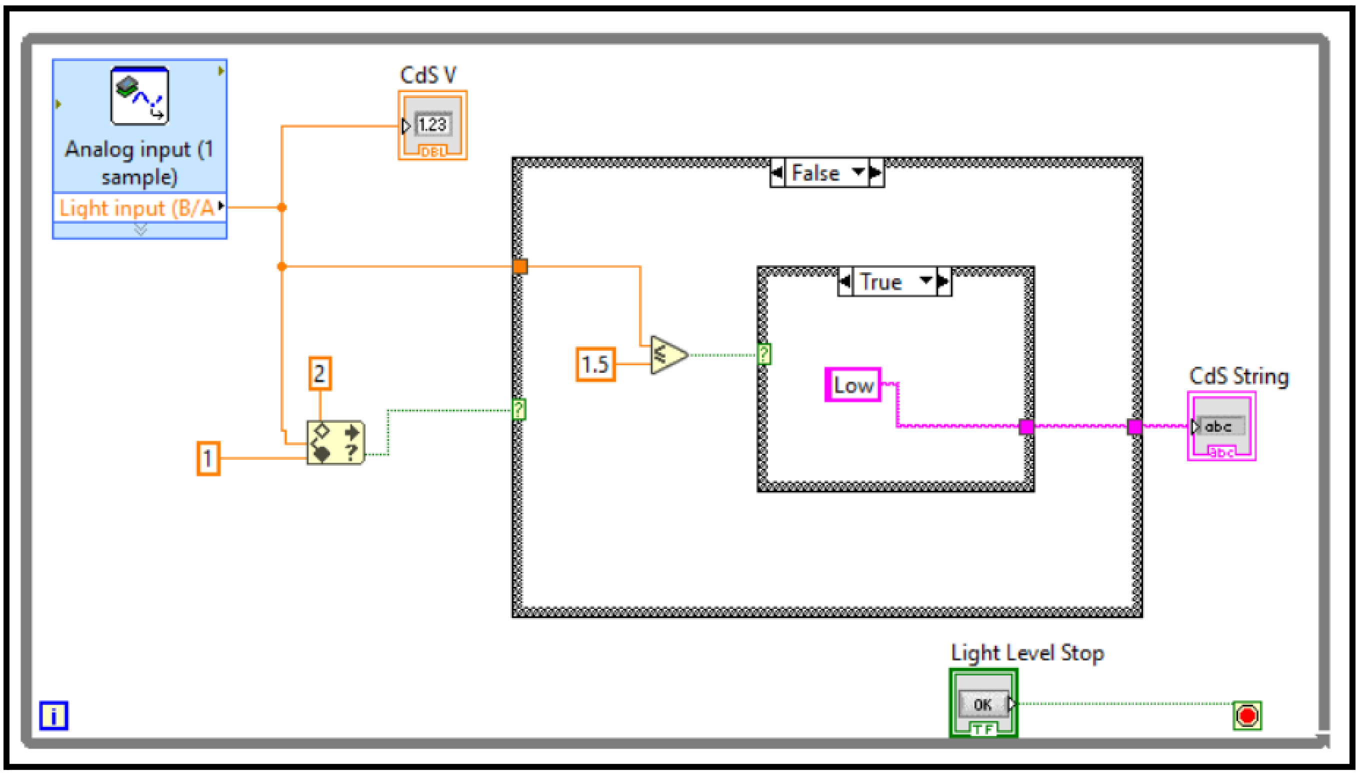

Figure 6: Labview code segment for controlling photoresistor

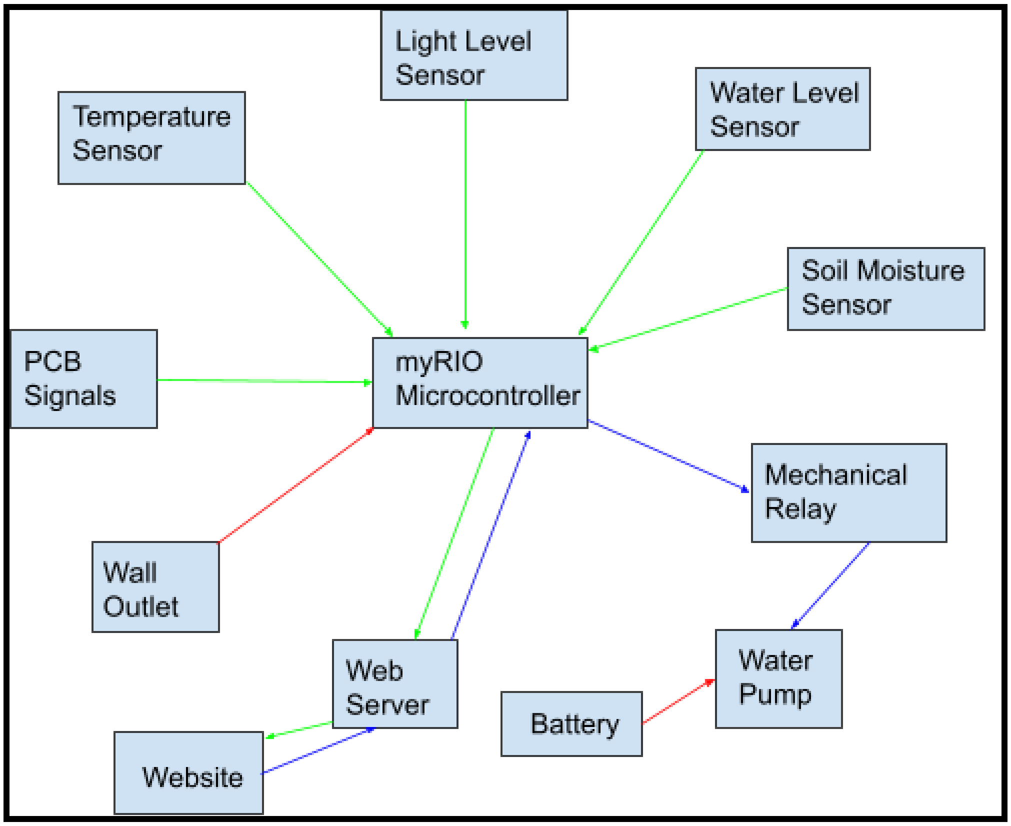

Figure 7: Microcontroller Block Diagram

In figure 7, the green arrows represent the flow of data, the red arrows represent the flow of power, and the blue arrows represent the flow of the control on/off signals from the website button to the water pump.

Technical Challenges:

The major challenge of our project was filtering out unwanted noise in order to isolate only the sounds coming from the plant. Originally, our microphone design used a passive bandpass filter and pre-amplifier. However, when implementing the filter on the breadboard we were not able to pick up any sound and achieve the desired results. After using Multisim to debug the problem, we discovered the bode plot was incorrect and the gain was way too low. This was because the unity gain bandwidth for our previous op-amps was too low for the frequencies we needed to work with. In order to fix this problem we redesigned the filter so there would be a sharper cutoff at 35kHz and 80kHz. To achieve this we used a fourth order Chebyshev filter and an instrumentation amplifier. We also chose components that had a large enough bandwidth which required buying new op-amps and a new instrumentation amplifier. Additionally, we first used the Sallen Key simplification 4 [24] which caused the gain of the filter to be too high and the output wave to clip at around 4V. We redesigned the filter to have unity gain so that the I-amp can supply all of the gain we need after the filtering stage to avoid clipping due to the non-ideal op-amps.

Another challenge that came up was when we had to interface the software with the hardware. In the beginning, we decided to connect the individual sensor to the Arduino UNO and hooked up the ESP8266 to the internet. This ended up being more complex than we anticipated since we had to integrate the microcontroller programming, signal processing libraries, push notification services, embedded software, and web development tools. Originally, we decided to use an ESP8266, Arduino, and MSP432 microcontroller to host a web server to interface with the sensors but it was overly complicated to connect all these additional softwares for each component so we thought the myRIO was better equipped to integrate all our features. At first, setting up the myRIO created challenges because there was new software that had to be downloaded to the microcontroller itself, which was not immediately obvious. We eventually fixed this problem after a lengthy debugging process by contacting the NI employee who first introduced the myRIO to us, and he was able to guide us to the correct software to download to fix the issue. Additionally, we ran into issues running a web server on the myRIO using their global server known as SystemLink cloud and a private API key to authorize the network connection so we contacted an employee from NI for assistance. Alternatively, we tried to use an HTTP request to connect to a local server using just a username and password to connect to the NI web server settings. Using the local NI server with the employee's guidance, we were able to determine that data had not been getting sent to the server correctly because some parameters of the data tags were not being defined. After fixing this, we were able to send regularly updating data to the web server.

Test Plans:

The ultrasonic sensor required the most amount of time and effort to test and fix. For the sensor, we tested the following components: microphone, filters, and amplifier.

To verify the microphone's functionality we first connected the output to an oscilloscope and tested that we were seeing pulses in the outgoing signal when we made sounds near the microphone such as clapping and talking. We then used an ultrasonic emitter to determine whether or not the microphone was able to pick up ultrasonic signals.

To test the filters we built the entire filter design on a breadboard then used the Waveforms software to generate an input signal. We used the network function to create bode plots for the filter by measuring at the final filter output. To ensure each subfilter was functioning properly, we tested the output at each individual filter's output and compared the results to those we simulated in Multisim. After testing by looking at the bode plots, we used the function generator to set functions to input to the filter at varying frequencies to observe if the appropriate response occurred. Finally, we tested the microphone and filters together by using the ultrasonic emitter to generate sounds for the microphone to pick up a signal to pass through the filters. The results of this can be seen in Figure 8 below. The yellow pulses are trigger signals to cause the emitter to produce a 40 kHz burst of sound, which is picked up by the microphone (the blue signal).

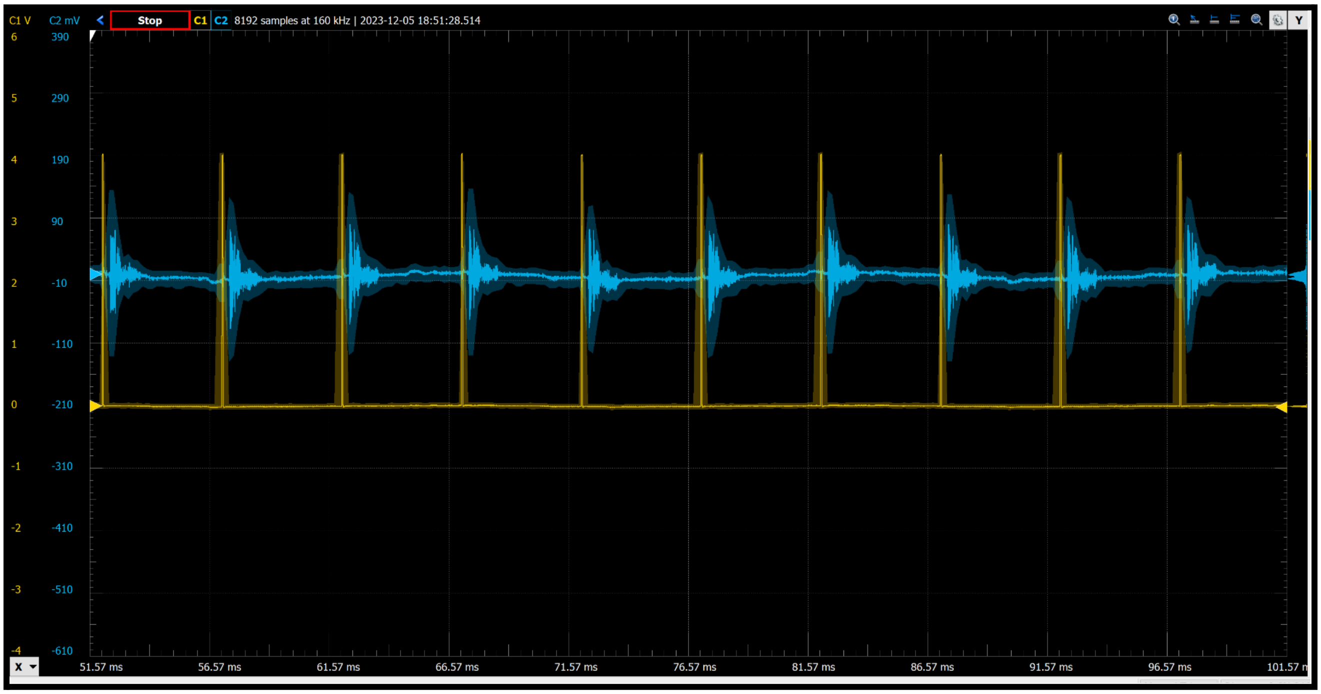

Figure 8: Output from filter using ultrasonic emitter as microphone input

To test the instrumentation amplifier (I-amp) we built only the subcircuit for the I-amp and fed a generated signal from Waveforms into the positive input terminal and measured the value at the output. We then tested all parts of the PCB by connecting the microphone output to the filters, the filter output to the I-amp, and used the ultrasonic emitter to generate sounds for the system to pick up.

For the other components of this project (the software, microcontroller, and physical watering system), we tested each one on its own using manually generated signals, then integrated parts together and tested using experimental signals. The output type of the humidity/moisture level sensor was evaluated early on to determine how to best incorporate it into our algorithm. For the microcontroller interface, we first tested the most basic input and output functionality of microcontroller pins, sending and receiving bytes. We then tested each branch of the software including all failure modes.

Timeline

We divided our project into four parts: Hardware (includes circuit design for ultrasonic microphone), water mechanism (includes water pump and moisture sensor), microcontroller (includes sensor communication), and user interface (includes the webpage). In terms of adjustments made, we ended up changing who led each subsection further along the semester as the demands of some sections require more work than others. Originally Sophia was lead for the microphone but due to unforeseen circumstances we made Audrey lead and Sophia secondary but the lead for power. Then we made Kate the lead for the watering mechanism system so she could fully focus on the embedded system with myRIO and Alex lead for the User Interface. As shown below in Figure 9 and Figure 10, the highlighted green boxes are the important tasks that were being completed at the time of the project, the red ToDo are tasks that still need to get done, and the dark green boxes are the tasks that were completed.

Proposal Gantt Chart:

GANTT CHART

Part 1

WBS	Task	Person Assigned	Start	Deadline	Completion
1	**Important Deadlines**	**All**	**Fri 8/18/23**	**Fri 12/08/23**	**To Do/ Doing/ Done**
1.1	Update Gantt Chart	Alex	Wed 9/13/23	Tue 9/19/23	Done
2	**Ultrasonic microphone/ Hardware**	**Lead: Sophia**			**To Do/ Doing/ Done**
2.1	Figure out exact frequency range of plants	Sophia/Audrey	Fri 8/18/23	Fri 9/01/23	Done
2.2	Research and figure out ultrasonic	Audrey	Fri 9/15/23	Tue 9/19/23	Done
2.3	What can the ultrasonic sensor can and	Kate	Sun 9/17/23	Wed 9/27/23	Done
2.4	Microphone testing how to read data and	Sophia/Audrey	Wed 9/27/23	Wed 10/04/23	Done
2.5	Do calculation for circuit	Sophia	Sun 9/17/23	Mon 10/09/23	Done
2.6	Build instr amplifier circuit to boost the weak electrical signals generated by ultrasonic transducer.	Sophia	Wed 9/27/23	Wed 10/11/23	Done
2.7	Make a filter circuit to filter out unwanted	Audrey	Wed 10/04/23	Wed 10/11/23	Done
2.8	Connect microphone to amplifier circuit	Sophia	Mon 10/09/23	Fri 10/20/23	Doing
2.9	Test filter	Audrey	Mon 10/09/23	Fri 10/20/23	Doing
2.10	Need an ADC to convert the analog signal	Sophia	Mon 10/09/23	Wed 10/18/23	Doing
2.11	Figure out power suppplies for each component	Sophia	Wed 10/04/23	Wed 10/18/23	Doing
2.12	Create circuit on bread board and test	Sophia/Audrey	Fri 10/06/23	Wed 10/18/23	To Do
2.13	Create PCB design on Multisim	Sophia/Audrey	Fri 10/13/23	Mon 10/23/23	To Do
2.14	Record data	Sophia/Audrey	Mon 10/23/23	Mon 10/30/23	To Do
2.15	Test with ultrasonic pest repeller	Sophia/Audrey	Mon 10/30/23	Fri 11/03/23	To Do
3	**Water mechanism**	**Lead: Audrey**			**To Do/ Doing/ Done**
3.1	Research what type of water pump to use for	Audrey	Fri 9/15/23	Sun 9/17/23	Done
3.2	Figure out specs (voltage) of water pump	Alex	Fri 9/22/23	Wed 9/27/23	Done
3.3	Read datasheet for how to work watering	Alex	Fri 9/22/23	Wed 9/27/23	Done
3.4	Connect water pump to arduino and test	Alex	Sun 10/01/23	Fri 10/06/23	Done
3.5	Connect water system to battery	Alex	Thu 10/05/23	Mon 10/09/23	Done
3.6	Research connect water pump to transistor use as trigger	Audrey	Mon 10/09/23	Fri 10/13/23	Doing
3.7	Read moisure sensor input	Kate	Wed 10/11/23	Wed 10/18/23	To Do
3.8	Research what moisure sensor and read data	Kate	Wed 10/11/23	Wed 10/18/23	Doing
3.9	integrate moisture sensor to water pump	Kate	Thu 10/05/23	Wed 10/18/23	To Do

Figure 9: First two subsections of Gantt Chart

GANTT CHART

Part 2

WBS	Task	Person Assigned	Start	Deadline	Completion
1	**Important Deadlines**	**All**	**Fri 8/18/23**	**Fri 12/08/23**	**To Do/ Doing/ Done**
1.1	Update Gantt Chart	Alex	Wed 9/13/23	Tue 9/19/23	Done
4	**Microcontoller/Embedded**	**Lead: Kate**			**To Do/ Doing/ Done**
4.1	Connect all sensors including temperature , photoresistor, and waterpump to breadboard	Alex	Fri 10/13/23	Wed 10/11/23	Done
4.2	Display all the sensor data on the serial monitor on the Arduino	Alex	Fri 10/13/23	Wed 10/11/23	Done
4.3	Connect water pump to microcontroller and trigger pump	Kate	Mon 10/09/23	Fri 10/13/23	Doing
4.4	LabVIEW tutorial	Kate	Wed 10/11/23	Fri 10/13/23	Doing
4.5	Look into wifi capabilities	Kate/Alex	Fri 10/13/23	Sun 10/15/23	Doing
4.6	Look into connecting ESP8266 to the internet	Kate/Alex	Mon 10/16/23	Tue 10/17/23	Doing
4.7	Connect ESP to temp sensor and display data	Alex	Tue 10/10/23	Wed 10/18/23	To Do
4.8	Stop water pump after moisure sensor	Kate	Wed 10/18/23	Fri 10/20/23	To Do
4.9	Develop algorithms to analyze the digital signal and convert it to byte or binary value	Kate/Alex	Mon 10/16/23	Wed 10/25/23	To Do
5	**User Interface**	**Lead: Alex**			**To Do/ Doing/ Done**
5.1	Research into sending mesages	Kate	Fri 9/15/23	Wed 10/04/23	Done
5.2	Create login page html	Alex	Fri 9/15/23	Sun 9/24/23	Done
5.3	Create home page html	Alex	Fri 9/15/23	Sun 9/24/23	Done
5.4	Create Plant from html	Alex	Sat 9/16/23	Mon 9/25/23	Done
5.5	Create help page html	Alex	Fri 9/22/23	Wed 9/27/23	Done
5.6	Create water preference page	Alex	Wed 9/27/23	Wed 10/04/23	Done
5.7	Create dropdown menu for watering	Alex	Wed 10/04/23	Fri 10/06/23	Done
5.8	Log in page backend	Alex	Wed 9/27/23	Mon 10/09/23	Done
5.9	Documentations upload on Github	Alex	Sat 10/07/23	Mon 10/09/23	Done
5.10	Write backend to input image into form	Alex	Fri 10/06/23	Wed 10/18/23	Doing
5.11	Display data on webpage	Alex	Fri 10/06/23	Wed 10/18/23	Doing
5.12	Create Instructions for water preferences	Alex	Fri 10/06/23	Wed 10/18/23	Doing
5.13	Look into how to connect webpage buttton to water pump	Kate/Alex	Mon 10/09/23	Fri 10/20/23	To Do
5.14	Take digital microphone signal to send alert to user	Kate	Mon 10/16/23	Fri 10/20/23	To Do

Figure 10: Second two subsections of Gantt Chart

Towards the end of our project, we created additional tasks to account for the technical challenges faced during our project. In the final chart, we had to pivot the hardware design to adjust for the change of components and the filters needed to detect the correct frequency. We also needed to redirect our attention to getting everything connected to the myRIO as we decided this approach would be faster and less complicated. Compared to the proposal Gantt Chart, we have way more tasks completed than we initially had as seen by Figure 9 and Figure 10. There are still some tasks that we were not able to complete because we had to push back the deadlines for certain tasks that were taking longer than expected. For instance, since we couldn't fully implement the microphone we were not able to process the signal and send the notification to the user when the plant sound was detected. Lastly, we added additional sections (*see Figure 13*) such as "Testing/Assembly" to account for integrating all the subsections together and "Supplies Needed" to keep track of everything we needed to buy for our Capstone project.

Final Gantt Chart:

10	Ultrasonic microphone/ Hardware	Lead: Audrey			To Do/ Doing/ Done
10.1	Figure out exact frequency range of plants	Sophia/Audrey	Fri 8/18/23	Fri 9/01/23	Done
10.2	Research and figure out ultrasonic	Audrey	Fri 9/15/23	Tue 9/19/23	Done
10.3	What can the ultrasonic sensor can and	Kate	Sun 9/17/23	Wed 9/27/23	Done
10.4	Microphone testing how to read data and	Sophia/Audrey	Wed 9/27/23	Wed 10/04/23	Done
10.5	Do calculation for circuit	Sophia	Sun 9/17/23	Mon 10/09/23	Done
10.6	Build instr amplifier circuit to boost the weak electrical signals generated by ultrasonic transducer.	Sophia	Wed 9/27/23	Wed 10/11/23	Done
10.7	Make a filter circuit to filter out unwanted	Audrey	Wed 10/04/23	Wed 10/11/23	Done
10.8	Connect microphone to amplifier circuit	Sophia	Mon 10/09/23	Fri 10/20/23	Done
10.9	Create circuit on Multisim	Sophia/Audrey	Fri 10/13/23	Wed 10/25/23	Done
10.10	Create circuit on bread board and test	Sophia/Audrey	Fri 10/06/23	Wed 10/18/23	Done
10.11	Figure out power suppplies for each component	Sophia	Wed 10/18/23	Wed 10/25/23	Done
10.12	Need an ADC to convert the analog signal	Sophia	Mon 10/09/23	Wed 10/18/23	Done
10.13	Test filter	Audrey	Mon 10/09/23	Fri 10/20/23	Done
10.14	Create PCB board on Ultiboard	Audrey	Mon 10/23/23	Fri 10/27/23	Done
10.15	Send out PCB	Audrey	Mon 10/23/23	Wed 11/01/23	Done
10.16	Recalculate values for filter	Sophia/Audrey	Mon 10/30/23	Wed 11/22/23	Done
10.17	Order parts for the PCB	Audrey	Mon 11/13/23	Wed 11/22/23	Done
10.18	Test new filter design on breadboard	Sophia/Audrey	Wed 11/22/23	Fri 12/01/23	Done
10.19	Solder components to PCB	Sophia/Audrey	Wed 11/22/23	Fri 12/01/23	Done
10.20	Record data	Sophia/Audrey	Fri 12/01/23	Wed 12/06/23	Done
10.21	Test with ultrasonic pest repeller	Sophia/Audrey	Fri 12/01/23	Wed 12/06/23	Doing
11	**Water mechanism**	**Lead: Sophia**			**To Do/ Doing/ Done**
11.1	Research what type of water pump to use for	Audrey	Fri 9/15/23	Sun 9/17/23	Done
11.2	Figure out specs (voltage) of water pump	Alex	Fri 9/22/23	Wed 9/27/23	Done
11.3	Read datasheet for how to work watering	Alex	Fri 9/22/23	Wed 9/27/23	Done
11.4	Connect water pump to arduino and test	Alex	Sun 10/01/23	Fri 10/06/23	Done
11.5	Connect water system to battery	Alex	Thu 10/05/23	Mon 10/09/23	Done
11.6	Read moisure sensor input	Kate	Wed 10/11/23	Wed 10/18/23	Done
11.7	Research what moisure sensor and read data	Kate	Wed 10/11/23	Wed 10/18/23	Done
11.8	connect moisture sensor , temperature sensor, and photoresistor to myRIO	Kate	Mon 10/23/23	Wed 11/01/23	Done
11.9	Research connect water pump to relay	Kate	Wed 11/01/23	Wed 11/15/23	Done
11.10	Connect water pump to myRIO microcontroller and trigger pump	Kate	Wed 11/01/23	Wed 11/15/23	Done
11.11	Clean up sensor display and put all sensor on the MyRIO breakout board	Kate	Fri 11/24/23	Fri 12/01/23	Done

Figure 11: First section of Final Gantt Chart

12	Microcontoller/Embedded	Lead: Kate			To Do/ Doing/ Done
12.1	Connect all sensors including temperature , photoresistor, and waterpump to breadboard	Alex	Mon 10/09/23	Wed 10/11/23	Done
12.2	Display all the sensor data on the serial monitor on the Arduino	Alex	Mon 10/09/23	Wed 10/11/23	Done
12.3	LabVIEW tutorial	Kate	Wed 10/11/23	Fri 10/13/23	Done
12.4	Look into wifi capabilities for the myRIO	Kate/Alex	Mon 10/23/23	Fri 10/27/23	Done
12.5	Connecting ESP8266 to the internet	Alex	Mon 10/30/23	Fri 11/03/23	Done
12.6	Host web server using ESP8266	Alex	Mon 11/06/23	Thu 11/09/23	Done
12.7	Stop water pump after moisture sensor	Kate	Mon 11/06/23	Wed 11/15/23	Done
12.8	Watch LABVIEW webUI tutorial	Kate/Alex	Mon 11/06/23	Wed 11/15/23	Done
12.9	Work on the email notification	Kate	Wed 11/15/23	Wed 11/29/23	Doing
12.10	upload sensor reading to the myRIO WebServer and host it publicly	Kate	Wed 11/15/23	Wed 11/29/23	Done
12.11	Take digital microphone signal to send alert to user	Kate	Thu 11/30/23	Mon 12/04/23	To Do
13	**User Interface**	**Lead: Alex**			**To Do/ Doing/ Done**
13.1	Research into sending mesages	Kate	Fri 9/15/23	Wed 10/04/23	Done
13.2	Create login page html	Alex	Fri 9/15/23	Sun 9/24/23	Done
13.3	Create home page html	Alex	Fri 9/15/23	Sun 9/24/23	Done
13.4	Create Plant from html	Alex	Sat 9/16/23	Mon 9/25/23	Done
13.5	Create help page html	Alex	Fri 9/22/23	Wed 9/27/23	Done
13.6	Create water preference page	Alex	Wed 9/27/23	Wed 10/04/23	Done
13.7	Create dropdown menu for watering	Alex	Wed 10/04/23	Fri 10/06/23	Done
13.8	Log in page backend	Alex	Wed 9/27/23	Mon 10/09/23	Done
13.9	Create Instructions for water preferences	Alex	Fri 10/06/23	Wed 10/18/23	Done
13.10	Documentations upload on Github	Alex	Sat 10/07/23	Mon 10/09/23	Done
13.11	Write backend to input image into form	Alex	Mon 10/23/23	Wed 10/25/23	Done
13.12	Get image to output on plants page	Alex	Wed 10/25/23	Thu 10/26/23	Done
13.13	Update Github	Alex	Fri 10/27/23	Sun 10/29/23	Done
13.14	Put form information on homepage	Alex	Wed 10/25/23	Tue 10/31/23	Done
13.15	Look into how to connect webpage buttton to water pump	Kate/Alex	Mon 11/13/23	Wed 11/15/23	Done
13.16	Include error handling on the form	Alex	Mon 11/13/23	Wed 11/22/23	Done
13.17	hyperlink myRio sensor website with the mainpage	Alex	Thu 11/16/23	Wed 11/29/23	Doing
13.18	Display data on webpage	Alex	Mon 11/13/23	Wed 11/29/23	Doing

Figure 12: Second section of final Gantt Chart

7	Testing/Assembly				To Do/ Doing/ Done
7.1	Assemble microphone on breadboard	Sophia/Audrey	Wed 11/01/23	Mon 11/20/23	Done
7.2	Test individual sensors	All	Fri 10/20/23	Tue 10/31/23	Done
7.3	Integration and testing	All	Mon 11/20/23	Thu 11/30/23	Done
7.4	Demo webpage and fix any potential bugs	Alex	Mon 11/27/23	Wed 11/29/23	Done
7.5	Create 3D printed clip for microphone	Alex	Mon 11/20/23	Wed 11/22/23	Done
7.6	Create box to put boards in and organize	Alex	Mon 11/20/23	Wed 11/22/23	Done
7.7	Print clip and box	Alex	Tue 11/21/23	Wed 11/29/23	Done
7.8	Test Microphone on PCB with Pest Reject	Sophia/Audrey	Mon 11/27/23	Fri 12/01/23	Done
8	Supplies Needed				To Do/ Doing/ Done
8.1	Pots plant	Kate	Sun 10/01/23	Wed 10/04/23	Done
8.2	Peristaltic dosing pump	Audrey	Tue 9/19/23	Wed 9/20/23	Done
8.3	MSP432	Alex	Tue 9/19/23	Wed 9/20/23	Done
8.4	Arduino UNO	Alex	Tue 9/19/23	Wed 9/20/23	Done
8.5	Microphone	Audrey	Tue 9/19/23	Wed 9/20/23	Done
8.6	SparkFun RedBoard	Alex	Tue 9/19/23	Wed 9/20/23	Done
8.7	Moisture sensor	Kate	Sun 10/01/23	Wed 10/04/23	Done
8.8	Ultrasonic testing device	Alex	Wed 9/27/23	Wed 10/04/23	Done
8.9	Ethernet ESP8266.	Alex	Tue 9/19/23	Wed 9/20/23	Done
8.10	Instumental Amplifier	Sophia	Mon 10/09/23	Wed 10/11/23	Done

Figure 13: Third section of final Gantt Chart

Costs

For this project, our total cost was $285.99. This does include parts that we decided not to use, so the cost is a little higher than the true cost for our group. We did, however, get to use the myRIO for free which costs $814.96 on Digikey, so that adds a substantial amount to the total cost if we were to manufacture this product, clearly making it an unreasonable price for consumers to pay. Looking at bulk prices, if we were to manufacture 10,000 units of our product it would cost $8,886,518.83. The details for this manufacturing cost as well as the individual cost of the project can be found in the appendix in figure 18 and figure 19.

Final Results

For our final product we were able to get many of the individual parts of our project working, but struggled to piece them together. Based on the criteria we defined ourselves, we would get a grade of a B for getting the water system to work with the user interface which interacts with all sensors except the microphone. However, we were able to get all aspects of the microphone PCB working on a breadboard (*see figure 14 for filter on breadboard*) but did not have time to solder all the components onto the board by the time this report was due. This was mainly due to the hardware difficulties and having to order new parts several times. When testing the board we were able to use the microphone to pick up ultrasonic sounds, filter out undesired frequencies, and amplify the filtered signal to produce a good quality signal to pass into the myRIO. We did this using the ultrasonic emitter device, not the plants. We did not get the coding done for observing the number of sounds from the plant per hour recorded to use to trigger the pump. Additionally, we successfully connected all the environmental monitoring sensors to the myRIO; these connections can be seen in Figure 15.

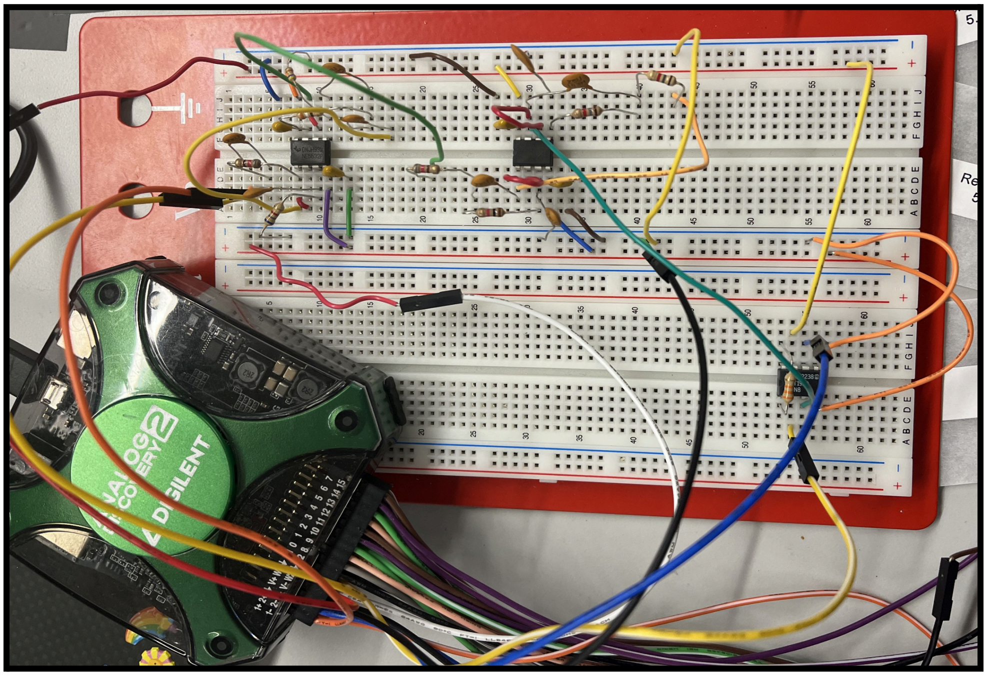

Figure 14: Breadboard with filter and I-amp design

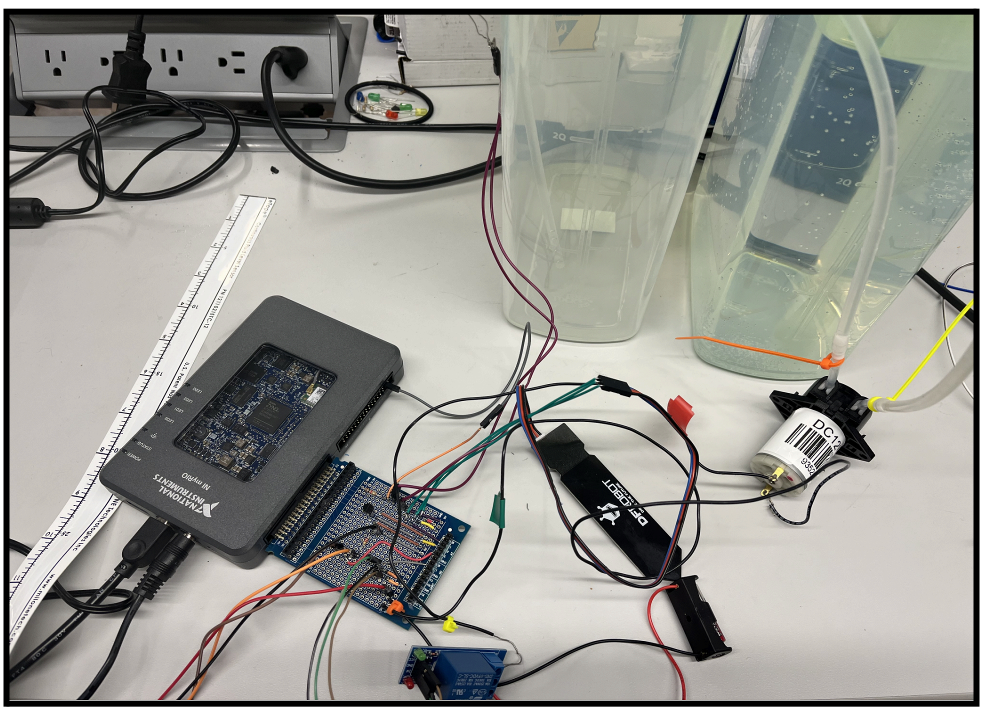

Figure 15: Breakout board

To explain the microcontroller sensors set up above (*Figure 15*) in more detail we will first discuss the main component: the watering mechanism which is a simple water pump that goes into the soil of the plant pot that will activate and release water when instructed by the user through the user interface. To log the amount of water being released for the plant, a moisture sensor is also placed deep into the soil. After a period of time, the exact length to be determined by testing, if the moisture sensor detects minimal changes in water levels, then the watering mechanism is shut off and the plant is recorded as watered. In addition to those, there is also a temperature sensor and a light sensor that inform the user if their plant's environment is optimal for growth. There is also a water level sensor, which turns on an LED on the myRIO if the water container needs to be refilled.

The user interface is a website that has three essential pages: a login page, home page, and settings. The user can login to access their personalized plant profiles and settings. Once logged in they will be navigated to the home page as shown in Figure 16 which provides an overview of the connected plants and their current status. Each plant will be represented as an image or icon with the plant's name and recommended environment conditions. The user can click an icon to see a detailed view of the sensor data for that plant, as seen in Figure 17. This screen displays a larger image of the plant, its name, and detailed information about its moisture level as a percentage value; the user will be able to adjust the water settings if needed. Users will also have the option to manually trigger the water pump if needed to hydrate the plant with a button that allows them to initiate this process immediately. There also is an input field that allows users to set the desired moisture level for the plant. If the ultrasonic microphone detects dehydration the alert will prominently display on this screen along with a timestamp. If there are issues with the water pump or other components of the system, such as a low water reservoir or a malfunctioning sensor, the user interface should be able to navigate to the help.html page to find the solution. These alerts should include descriptive error messages and guidance on how to resolve the issue.

Figure 16: User Interface

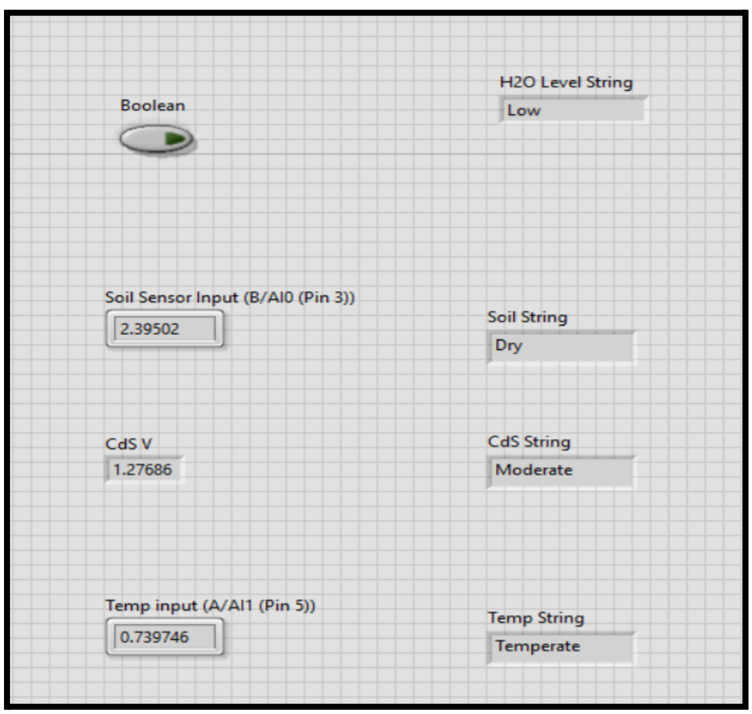

Figure 17: Labview code user interface for sensors

Future Work

Future work for this project includes aspects of the design we had aimed to complete — the remaining tasks on the project timeline — that we unfortunately did not get to. One major functionality to be added in future work is the plant communication aspect between the signal processing and the user. We are currently able to detect when the plant is dehydrated, however we did not get the system set up where the user would get an email and be able to water the plant via the website based on that feedback. That had been a goal of our design that we fell short on, but an improvement to also consider in future work is to use texts instead of emails for the alerts since texts are a much more popular medium of communication these days.

Another section of work we did not manage to accomplish was writing the code to process the filtered and amplified signal from the microphone picking up sounds from the plant. This code would track the number of pulses per time period (example: 30 minutes) deemed to be the ultrasonic popping noises from the plant and use those numbers to determine when the plant is dehydrated.

Another improvement we did not get to was having one microphone point to the plant contained in a cone, another pointing away from this, and subtracting that out from the first microphone signal to get a better plant signal with minimal noise. For future work an improvement would be to find a better system for collecting the sounds from the plants than this as this method could easily cause issues. Issues could occur if the cone blocks enough of the surrounding sound that subtracting the second microphone signal from the first could result in a negative signal even when the plant is making noises.

A major task delegated to future work is to find ways to reduce the cost of our system. The use of the myRIO as our microcontroller alone brings the minimum cost up to over $800. To make a profit off this design, the price would have to be over $1,000 which is not a reasonable price for the average person to pay for such a device. Some ways the project could be improved or expanded upon include figuring out a way to determine specifically how much water a plant should be given when it is detected as dehydrated rather than relying on the soil moisture level sensor using immediate feedback to determine when to stop the water flow.

A recommendation we would have for someone attempting to recreate or expand upon our design is to act with a sense of urgency throughout the project to avoid as much delay as possible. We struggled with budgeting our time properly as we underestimated how long everything would take. Another pitfall included spending too much time looking at the individual pieces of the design without considering deeply how to connect them all together. Although it is important to get smaller pieces of the project working as it would be impossible to look at the whole design at once, it is important to have a clear plan for connecting all the pieces.

References

[1] I. Khait *et al.*, “Plants emit informative airborne sounds under stress,” Dec. 2018, doi: https://doi.org/10.1101/507590.

[2] F. Frongia, L. Forti, and L. Arru, “Sound perception and its effects in plants and algae,” *Plant Signaling & Behavior*, vol. 15, no. 12, p. 1828674, Oct. 2020, doi: https://doi.org/10.1080/15592324.2020.1828674.

[3] S. Sundari.M, J. M. Mathana and T. S. Nagarajan, "Secured IoT Based Smart Greenhouse System with Image Inspection," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020, pp. 1080-1082, doi: 10.1109/ICACCS48705.2020.9074258.

[4] Group, J. Gilbert -Cpe, A. Lam -Ee, C. Dany, M.-E. Megan, and M.-E. Ii, “SmartLeaf Indoor Greenhouse System.” Accessed: Sep. 16, 2023. [Online]. Available: https://www.ece.ucf.edu/seniordesign/su2019fa2019/g12/docs/SD1%20Final%20Paper.pdf

[5] IPC, Design Guide for Protection of Printed Board Via Structures. 2006.

[6] NFPA, National Electrical Code. 2022.

[7] NEMA, American National Standard for Portable Cells and Batteries – Environmental. 2023.

[8] IEEE, IEEE802. 2023.

[9] “Price list & Ordering information,” Avisoft Bioacoustics, https://www.avisoft.com/price-list-ordering-information/ (accessed Sep. 19, 2023).

[10] D. KOCH, “Ultrasonic Microphone Part 1 ,” DIYODE Magazine, https://diyodemag.com/projects/ultrasonic_microphone_heterodyne_lm324_lm386_lm833 (accessed Sep. 20, 2023).

[11]“CM16/CMPA,”Avisoft Bioacoustics, https://www.avisoft.com/ultrasound-microphones/cm16-cmpa/ (accessed Sep. 19, 2023).

[12] T. A. Team, “Send SMS,” Arduino Documentation, https://docs.arduino.cc/library-examples/gsm-library/GSMExamplesSendSMS (accessed Sep. 19, 2023).

[13] “Plants emit ultrasonic sounds in rapid bursts when stressed, scientists say,” The Guardian, https://www.theguardian.com/environment/2023/mar/30/plants-emit-ultrasonic-sounds-in-rapid-bursts-when-stressed-scientists-say (accessed Sep. 19, 2023).

[14] C. Freid, "Are required courses meeting industry demands?," IEEE Potentials, vol. 20, no. 3, pp. 39-40, 2001.

[15] "Zotero: Your personal research assistant," [Online]. Available: www.zotero.org. [Accessed 02 Sep 2020].

[16] “IEEE Standards Association,” *IEEE Standards Association.* https://standards.ieee.org/ieee/1451.99/10355/

[17] “ISO - ISO/TC 207/SC 1 - Environmental management systems,” www.iso.org. https://www.iso.org/committee/54818/x/catalogue/p/1/u/0/w/0/d/0

[18]“Audio Solutions Question of the Week: What Is a Microphone’s Frequency Response and Why Is It Important To Know?,” *www.audio-technica.com*. https://www.audio-technica.com/en-us/support/microphones-frequency-response-important-know/#:~:text=Many%20microphones%20have%20a%20frequency

[19] “Plants emit ultrasonic popping sounds when stressed,” *www.nhm.ac.uk*. https://www.nhm.ac.uk/discover/news/2023/april/plants-emit-ultrasonic-popping-sounds-when-stressed.html#:~:text=Scientists%20have%20recorded%20plants%20producing

[20] S. Aggarwal, University of Virginia School of Engineering and Applied Science. “Plant Ladder System; Peer-to-Peer Variable Service Transaction; Parenting: Past, Present, Future,” *libraetd.lib.virginia.edu*. https://libraetd.lib.virginia.edu/public_view/8p58pd804

[21] R. Rowe, “Plant watering system.” https://patents.google.com/patent/US20070089365A1/en

[22] W. Crane, “Automatic plant watering device.” https://patents.google.com/patent/US3758987A/en

[23] “Automatic watering device for pot-plants.” https://patents.google.com/patent/US262379

[24] “Analysis of the SallenĆKey Architecture Mixed Signal Products Application Report SLOA024B,” 1999. Accessed: Dec. 06, 2023. [Online]. Available: https://www.ti.com/lit/an/sloa024b/sloa024b.pdf?ts=1701834508390&ref_url=https%253A%252F%252Fwww.google.com%252F

[25] J. Karki, “Active Low-Pass Filter Design.” Accessed: Dec. 06, 2023. [Online]. Available: https://www.ti.com/lit/an/sloa049d/sloa049d.pdf?ts=1701883018163&ref_url=https%253A%252F%252Fwww.google.com%252F

[26] “INA849 Ultra-Low-Noise (1 nV/√Hz), High-Bandwidth, Instrumentation Amplifier 1 Features.” Accessed: Dec. 07, 2023. [Online]. Available: https://www.ti.com/lit/ds/symlink/ina849.pdf?ts=1701892619871&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FINA849

Appendix

Budget Outline:

Qty Req'd	Per Unit Price	Cost	Comments
5	$0.77	$3.85	100 Hz - 80 kHz MEMS analog mic
5	$1.22	$6.10	50 Hz - 80 kHz MEMS analog mic
1	$4.40	$4.40	Wifi SOC module
1	$5.90	$5.90	Analog capacitive soil moisture sensor
2	$9.04	18.08	Instrumentation amplifier
1	$7.65	$7.65	ESP wifi module
10	$0.55	$5.50	BATTERY ALKALINE 12V 23A
1	$6.49	$6.49	12V battery holder with wires
1	$5.25	$5.25	Digital thermometer. Might be $6.49 if promotional price ends before time of order
1	$2.45	$2.45	Extra temperature sensor
2	$9.49	$18.98	Water jug with green lid
3	$1.95	$5.85	Extra water level sensors
1	$39.95	$39.95	Long water level sensor
1	$17.95	$17.95	Silicone tape
1	$17.13	$17.13	Flex seal
3	$0.79	$2.37	Op-amps
1	$1.80	$1.80	SMD to DIP 8-SOIC
3	$10.00	$30.00	Plants
1	$35.00	$35.00	PCB
1	$46.97	$46.97	3D-printed cable management system
1	$4.32	$4.32	3D-printed clip
		Total Cost	
		$285.99	

Figure 18: Cost of individual project

Qty Req'd	Per Unit Price	Cost	Comments
10000	$0.66	$6,600.00	50 Hz - 80 kHz MEMS analog mic
10000	$5.90	$59,000.00	Analog capacitive soil moisture sensor
10000	$0.42	$4,200.00	BATTERY ALKALINE 12V 23A
1667	$6.49	$10,818.83	12V battery holder with wires
10000	$5.25	$52,500.00	Digital thermometer. Might be $6.49 i
10000	$2.45	$24,500.00	Extra temperature sensor
10000	$9.49	$94,900.00	Water jug with green lid
10000	$39.95	$399,500.00	Long water level sensor
30000	$0.79	$23,700.00	Op-amps
10000	$1.80	$18,000.00	SMD to DIP 8-SOIC
10000	$4.32	$43,200.00	3D-printed clip
10000	$814.96	$8,149,600.00	myRIO
		Total Cost	
		$8,886,518.83	

Figure 19: Cost of 10,000 Units of the Product

Github Repository:

Link: https://github.com/alamo2000/CapstonePlantWhisperers.git