Smart Herb Garden

Marcus Fernandez DeCastro, Jordyn Hayden, Lucas Jaramillo, and Charlee Mione

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — *The* Smart Herb Garden grants users the satisfaction of harvesting home-grown herbs, and most importantly making it convenient. The device will house herbs and feature an automated robotic watering system which replenishes an herb's moisture depending on the moisture level detected in the soil via sensors. This way, once the soil moisture level reaches a low reading, watering occurs without needed assistance from the user. The automated watering system is composed of an XY Motion robot attached above the planters. The software will plot the positions of the plants in an XY grid, while the mechanism will move the hose using stepper motors. This motion is dictated by the moisture sensor which will give an alert once a low moisture level has been reached. Conveniently, our Smart Herb Garden comes with a custom-built companion web application that allows a user to log in and view the historical and most recent moisture readings of their garden.**

***Index Terms* — *Automation*, internet, sensors, stepper motors, user interface.**

I. Introduction

People all over the world decide to take on gardening as a soothing and rewarding past time. Such a hobby can quickly become inconvenient, easy to neglect, and is thereby routinely given-up-on. The team harnesses the desire to spread the satisfaction of growing one's own fresh herbs and eliminate the inconveniences one would face doing so in a typical garden. The Smart Herb Garden takes a green thumb out of the equation for growing fresh herbs by implementing an automated watering system featuring a convenient web application to display heaps of data collected by use of sensors in the garden. The system will be self-guiding, easy to use, and generate an overall positive user experience. The team believes gardening and harvesting fresh herbs is a hobby all users of the Smart Herb Garden shall experience regardless of previous background in gardening or background in Smart Devices. The Smart Herb Garden solution utilizes a robotic arm to deliver water like an arcade claw game machine. Each plant has a moisture sensor inserted into the soil. The microcontroller will take readings of each plant and activate the robotic arm to move to whatever plant has low moisture. When the moisture level drops low, a robotic arm moves to the plant and activates a water pump to give it water. Plus, the user can log in to a web application to view the readings of the sensors, even if they are away from home.

II. System Components

The figure below, Fig. 1, introduces the overall project block diagram which highlights the main components of the project which clearly visualizes the solution design at a high-level view. In this section, a brief overview of each component featured in the following block diagram is detailed.

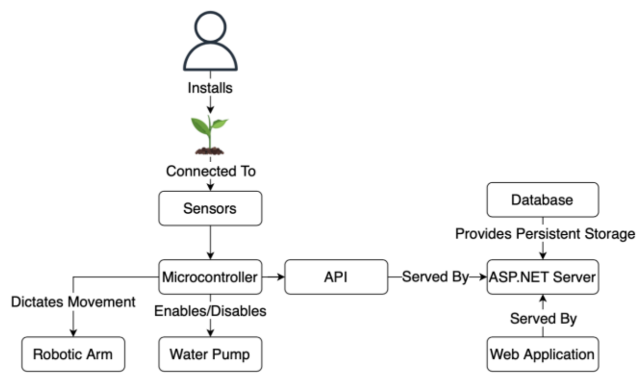

Fig. 1 Overall Project Block Diagram

A. Microcontroller

The microcontroller utilized in the PCB is the ATSAMD21G18A-MUT chip that is part of the SAM D21 family of chips. Due to the part shortage, workarounds had to be implemented to retrieve this chip. Several Arduino development kits with these chips were purchased and the microcontroller was desoldered using hot air. Fortunately, no damage to the chip was done in the process. One of the benefits of this process is that since the chip was desoldered from an existing kit, the firmware was already flashed onto the device, helping to reduce cost and development time. The microcontroller was selected because of its compatibility with the Arduino IDE which further significantly reduces development time, especially with the tight deadlines for the completion of the project. By using this microcontroller, the PCB can communicate effectively with the Arduino IDE and software can be flashed in a timely and effective manner.

B. Sensors

The sensors utilized in the PCB are the Songhe Capacitive Soil Moisture Sensor. These sensors were chosen because of their low cost and because they already come with pre-manufactured housing. The sensors already come with connectors, allowing them to be quickly snapped into slots in the PCB. The mechanism of operation of the sensors is that as soil moisture changes, there is an interaction between the soil and the copper plane of the sensor causing there to be a change in capacitance. This change in capacitance is reported as a voltage level. A calibration procedure was performed by exposing the sensors to 100% moisture and 0% moisture. The reported voltage levels were mapped to the known values and a transfer function was utilized to convert these voltage levels to known percentages.

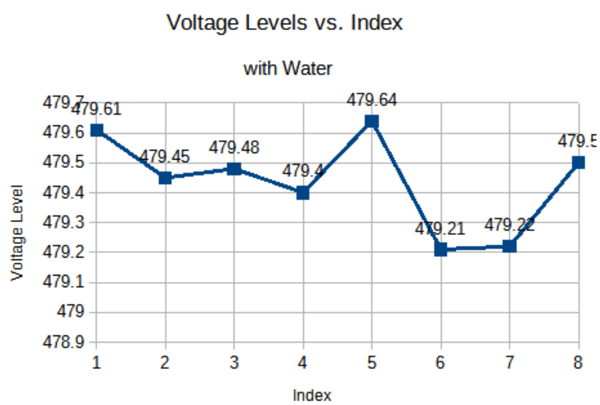

Fig. 2 Voltage levels retrieved from calibration with water

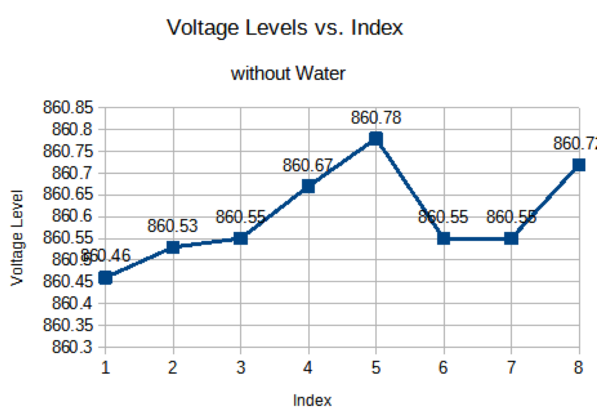

Fig. 3 Voltage levels retrieved from calibration without water

C. Robotic System

The purpose of the robotic system is to deliver water to the five different plants in the smart garden. When designing the system, the main consideration was to implement a fully automated watering system that utilizes the data collected by the moisture sensors. Once a moisture sensor sends data lower than the ideal threshold, the robot will move to the respective plant and the water pump will turn on. The robot is fixed on a wooden frame that fits around two planter boxes and moves a hose above each of the five plants. The robot moves in the x and y direction using a stepper motor for each motion.

D. Water Pump

The main design consideration behind the water pump was to ensure that the smart garden system maintained the ability to be mobile and versatile. This reasoning is why an external water pump is used rather than using the main water line to deliver water to the plants. The water pump is housed in the planter boxes to keep ease of mobility. The external water pump allows for this garden to be functional wherever the user has an outlet whether that be inside or outside.

E. Application Programming Interface (API)

Since the web application will require the detected data by the various sensors present in the Smart Herb Garden, naturally an API will be involved. Using an API allows for the simple exchange of information. An API could leverage a Representational State Transfer (REST) Protocol among others, in which case that API would be termed RESTful. REST is a popular protocol which the team will use all to allow the user to conveniently view moisture, humidity, and temperature data through the web application.

F. ASP.NET Server and Web Application

The user will interact with a web-based application served from the ASP.NET server. The data served with the page is handled internally with user-facing pages. ASP.NET was chosen for its ability to communicate with a database model directly mapping database entries to objects. Created by Microsoft, Razor Pages is a web development framework that is lightweight and notably cross-platform. Such web development is simplified through a page-based development model which is an attractive workflow for many developers which is one of many reasons why this technology has been used.

G. Database

The web application is developed using an impressive architecture which supports more than 25 different database providers. This is possible using Microsoft Entity Framework (EF) Core. EF Core grants the flexibility to interchange the utilized database in the solution without the requirement of altering a notable amount of code aside from small amounts of configuration.

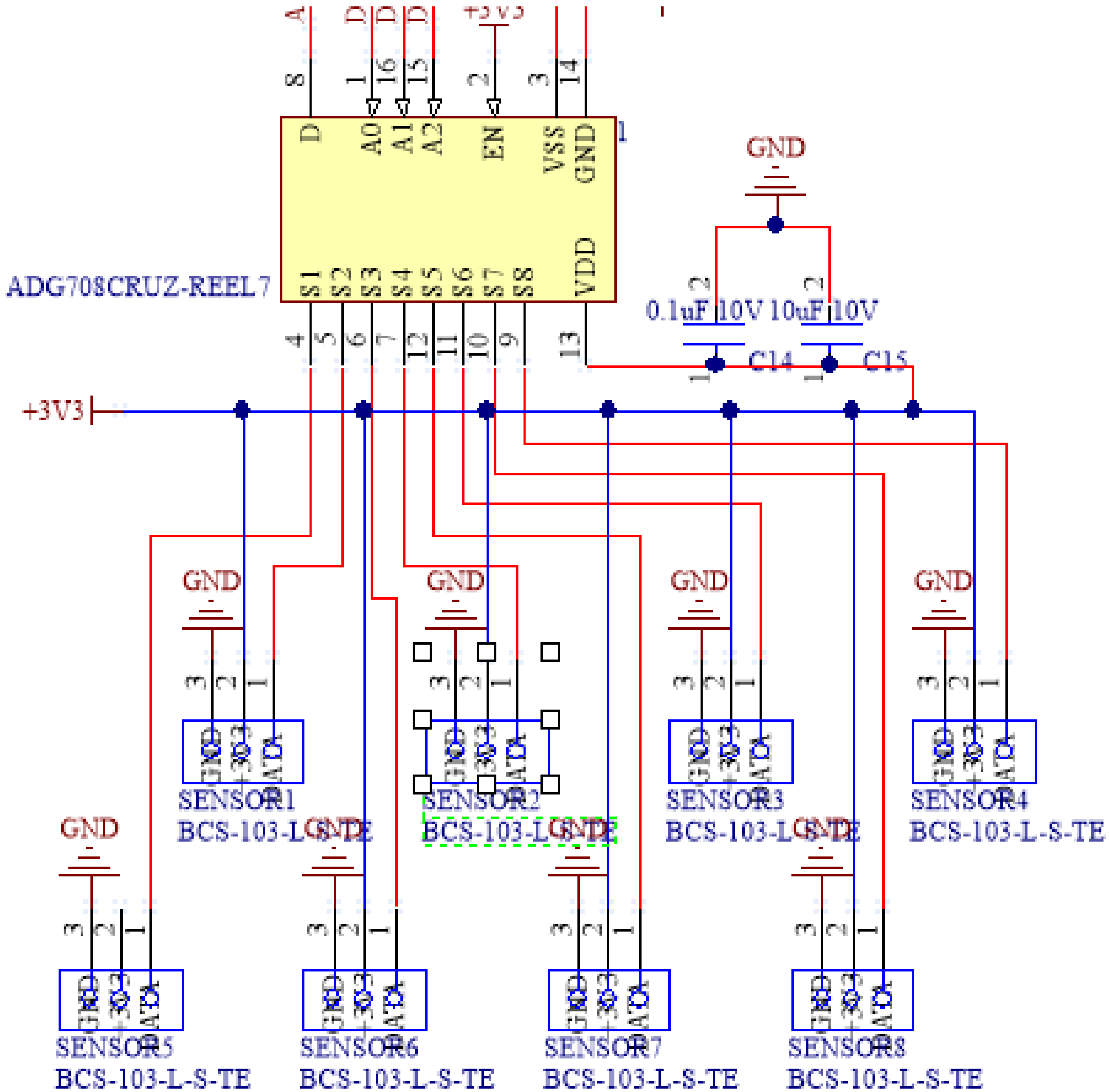

Fig. 4 PCB Design

I. PCB DESIGN

There are two PCBs that were manufactured for use in the project. Both boards were manufactured by JLCPCB and components were sourced from various vendors and were either hand-soldered or soldered by JLCPCB. The detailed design process and implementation are outlined below.

A. Design

Version 1.0 of the PCB was used strictly for prototyping and was designed using KiCad. Fig. 4 displays the first PCB design. Version 2.0 of the PCB is the final version and was

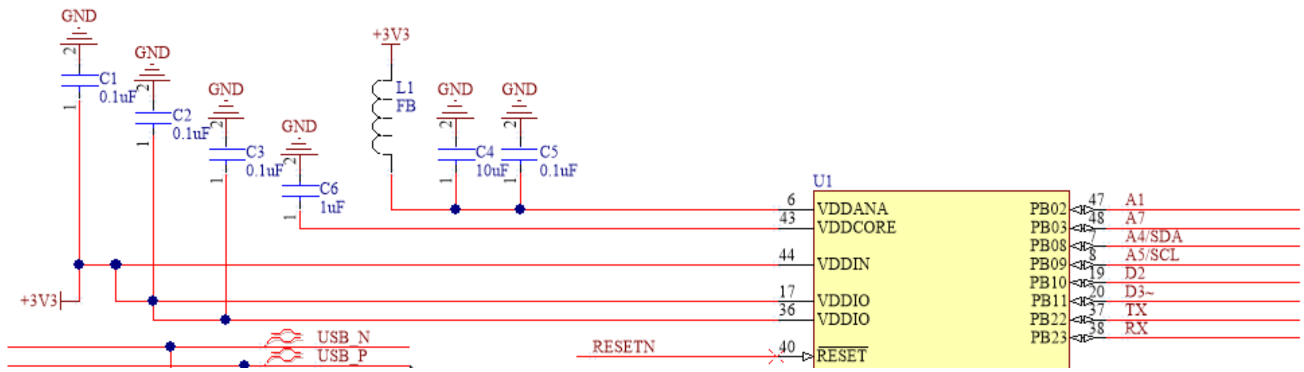

Fig. 5 Example of decoupling capacitors used to ensure smooth voltage levels for the microcontroller

designed using Altium Designer. Version 1.0 consists of a multiplexer, 8 connectors, and two slots to plug in a development kit. Version 2.0 consists of a microcontroller, Wi-Fi/Bluetooth module, cryptographic module, buck regulator, multiplexer, switch, and various connectors and LEDs. Version 2.0 contains significant PCB design and is based on the design of the Arduino Nano 33 IoT. The PCB takes in readings from various sensors and multiplexes them into a signal pin. Then, it processes the data to form it into an HTTP POST request containing JSON-formatted data. The PCB connects to a router and then communicates via Wi-Fi with the server for the web application. The PCB contains a micro-USB port for power from a traditional phone charger and allows for debugging with the Arduino IDE. There is a reset switch and two status LEDs. The red LED indicates power, while the yellow LED indicates communication with the Arduino IDE. The PCB provides support for various potential features such as Bluetooth, allowing room for further development beyond the scope of the project.

B. Implementation of v2.0

Version 2.0 will be utilized as the final version; therefore Version 1.0 will no longer be discussed. Regarding the implementation of v2.0, the process began with developing a multiplexer circuit. The multiplexer circuit allows for all data from the sensors to be sent to 1 pin on the microcontroller, helping to reduce the number of pins utilized. There are 8 slots for each sensor, and each slot contains pins for GND, VCC, and AOUT. Each of the GND pins is connected to the board's ground plane, and each of the VCC pins is connected directly to the same 3.3V power supply. Using a multimeter, it was determined that the current draw of 8 sensors is highly negligible in the microamp range, thus it is safe to tie each slot to the same VCC pin without overloading the wire trace on the PCB. Each of the AOUT pins connects to an individual pin on the 8-1 multiplexer. Three select lines are connected to the microcontroller. One pin is utilized for the multiplexed output and connects to the microcontroller. Two decoupling capacitors are connected in parallel to the multiplexer's voltage supply pin. Fig. 4 is a schematic of this circuit (and also serves as schematic of Version 1.0).

After designing the multiplexer circuit, the remainder of the PCB was designed based on the Arduino Nano 33 IoT and its open-source schematics as a guide. The same components were primarily used, with the exception of not using the gyroscope/accelerometer since it is not needed. A micro-USB port was included, and it was connected to an electrostatic discharge protection device to prevent any integrated circuits from frying. However, components on the board need to be powered strictly with 3.3V, but USB provides 5V. Therefore, a buck regulator was utilized to convert the 5V down to 3.3V. There were various passive components utilized such as decoupling capacitors and such. The numerical values for the specific capacitances, inductances, and resistances were provided by the example applications of the datasheet [1]. Fig. 6 demonstrates the example application of the data sheet that was utilized [1].

Finally, the microcontroller, Wi-Fi module, and cryptographic modules were included in the PCB design. Various decoupling capacitors were utilized to ensure that the voltage levels are stable. Certain pins were connected to the multiplexer circuit, meanwhile other pins were connected in the same manner as the open-source schematics to ensure that the board would be compatible with the Arduino IDE. Two test points were included in

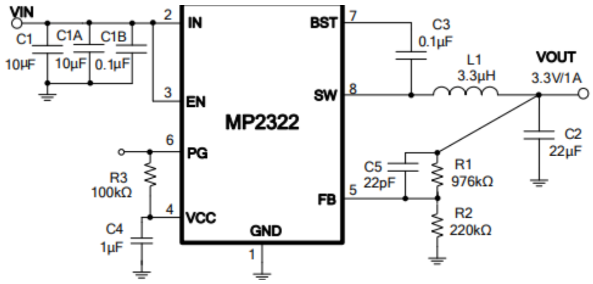

Fig. 6 The buck regulator circuit in the PCB was based on the application example provided by the datasheet

the PCB, in addition to 30 pins that can be used to connect additional devices or to be used as test points themselves. There is also a reset switch, and two status LEDs. A portion of the schematic can be seen in Fig. 5.

II. Robotic System Detail

The robotic system is designed to deliver water to the plants most efficiently. The robotic system consists of two NEMA17 stepper motors each with their own A4988 stepper motor driver. These specific stepper motors were chosen due to their high torque and low cost, which brings down the total price of the smart garden system, while also precisely and accurately moving the tubing of the watering system to the specific plant. The stepper motor drivers were chosen because of their simple implementation, low cost, and variety of features. The robotic system also consists of two linear motion shafts that carry two wooden platforms in the x and y directions, as well as two timing belts attached to the platforms that facilitate their movement. The stepper motor drivers will be connected to the PCB designed by the team to allow for the movement of the robotic system to correlate with a low moisture level reading from the sensors.

A. Design

The garden consists of two plastic planter boxes surrounded by a wood frame made with 2in thick wood. It is measured to be 28in x 14in x 12in. and the robot is fixed on the top of this frame above the planter boxes. The robot is designed using linear motion shafts and linear ball bearing slide block units. This design choice was based on both 3D printers and XY plotter robots. Two linear motion shafts are fixed to the wooden frame in the x-direction and two slide block units are placed on to each shaft. Attached to these units are pieces of wood where the y-direction shafts are placed. These y-direction shafts have two slide block units that will hold up the vinyl tubing used for watering via another piece of wood. There are two stepper motors assisting in the x-direction and y-direction movement. The stepper motor for the x-direction is attached directly to the frame and will remain still whereas the stepper motor for the y-direction will be moving with the y-direction linear motion shafts. As seen in Fig. 7, this demonstrated the application of the microcontroller, stepper motor driver, and stepper motor attached.

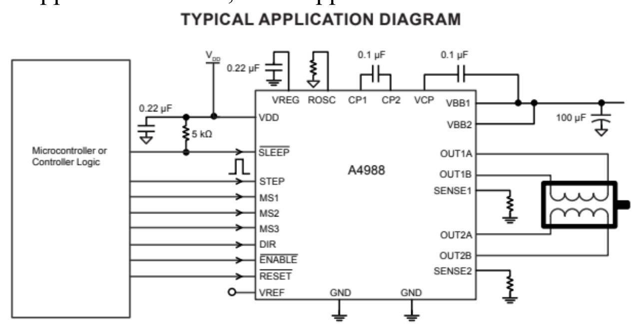

Fig. 7 Application Diagram

B. Stepper Motors and Motor Drivers

The stepper motor chosen for this project is the NEMA17 stepper motor. It has a rated voltage of 12 volts and a rated current of 1.2A at 4V. It also has a step angle of 1.8 degrees, meaning that at 200 steps per revolution, each revolution will turn the motor 1.8 degrees, allowing for a high degree of control. This allows for a high torque motor with high precision. The motor has six wires, connected to two split windings. The black, yellow, and green wires are part of the first winding where black is the center tap and the yellow and green are the coil end wires. The red, white, and blue wires are part of the second winding where white is the center tap and the red and blue are the coil end wires. The center taps of the windings are wires to the positive supply, while the two ends of the windings are grounded through a drive circuit. The order of the stator poles in the motor is A, B, A', B' as seen in Fig. 8. [2]

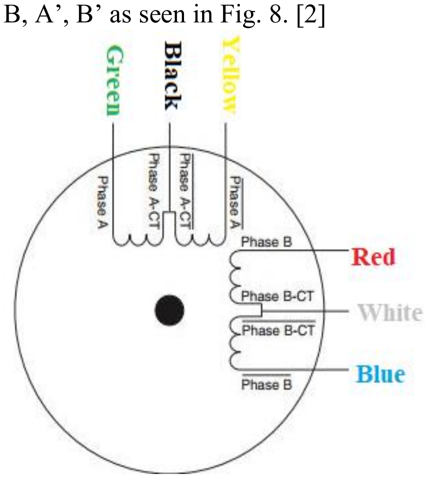

Fig. 8 Stator Pokes in the Stepper Motor

The motor drivers used to control the stepper motors are the A4988 stepper motor drivers. Due to its simplicity as well as the variety of stepping modes, it is the ideal solution for the application of a smart garden which requires precise and reliable stepper motor control. The drivers require only two pins to control the speed and direction of the bipolar Nema 17 stepper motors that are used. The A4988 motor driver has an output drive capacity of up to 35V and +/- 2A, which allows for controlling a bipolar stepper motor at up to 2A output current per coil. The output current is regulated which allows for noiseless operation of the stepper motor and eliminates resonance or ringing that is common in unregulated stepper driver designs. The motor driver also supports micro stepping, which can be completed by dividing a single step into smaller steps, achieved by energizing the coils with intermediate current levels. The built-in translator allows for the straightforward operation of the motor driver. This allows for the reduction of the number of control pins to two, one for controlling the steps, and one for controlling the direction that the motor spins. Some other features of the driver include under-voltage, shoot-through, overcurrent, and thermal protection in the form of a heatsink. [3]

For this project the stepper motors will utilize the moisture sensors' collected data to aid in the watering of low moisture plants. If a plant's moisture level is below the minimum moisture for the specific plant type the stepper motors will move the watering system to the center of the plant quickly and efficiently. Fig. 9 models the movement of the stepper motors for both the x-direction and the y-direction. [4]

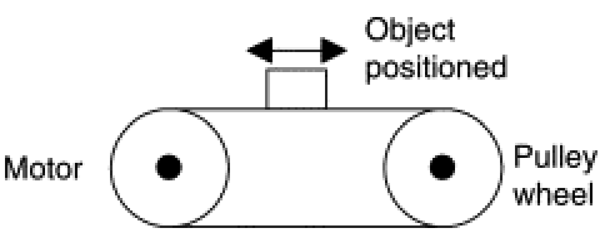

Fig. 9 Model of Robot's Movement

C. Water Pump, Tubing, and Relay

The water pump chosen for the project is a Submersible Mavel Star Mini Water Pump that has a flow rate of 240 liters per hour. The water pump will be connected to PVC Clear Vinyl tubing to facilitate the water to the low moisture plants. The water pump is programmed via a relay. A relay is an electrical switch that is programmable and is used to control the on and off function of a device. The relay has two groups of pins: input and output. The input group is connected to our microcontroller and will allow the relay to receive the control signal. The output group is connected to the water pump and has a "normally open" pin and a "normally closed" pin. These modes are opposite in how they work. The "normally open" mode implies that the current does not flow in its "normal" state and the "normally closed" mode implies that the current does flow in its "normal" state. For most of the time the water pump will be off and for this reason the "normally open" mode will be used. [5]

For this project the water pump will be connected to the COM pin and NO (normally open) pin. If the input pin is programmed to LOW the switch is open and the device is off. If the input pin is programmed to HIGH the switch is closed and the device is on. This functionality will allow for the water pump to be on only when it is needed. When a plant's moisture level falls below the minimum set and the tubing is moved to the correct plant, the relay's input pin will be programmed to HIGH using digitalWrite() and the water pump will begin to water the plant.

III. Web Application Detail

The purpose and motivation behind the inclusion of a web application is to allow the user to conveniently and remotely view the data collected in a tabular form from the sensors apart of their garden(s) regarding moisture, temperature and humidity. The factors that were taken into consideration and drove the web application design are common principles that make for a good software design. Most importantly maintainability, testability and loose coupling are instances of a few that have been chosen by the team to prioritize. Designing a software solution using these principles will improve the product quality dramatically and enhance not only the experience for the developer but importantly the customer as well.

The team's web application is modern, performant, and custom-built. In this section the main components are detailed including why they were chosen as well.

A. The Microsoft Stack

The suite of Microsoft tools and applications that have been utilized to develop the web application include ASP.NET Core, Razor Pages, Web API, Azure, and the C# programming language. These available tools are well documented online and are designed for modern web experiences. Most notably, Microsoft's ASP.NET Core is the chosen framework due to its high performance, cross-platform capability, dependency injection design pattern support, and its testability which greatly improves the quality and integrity of the software.

B. Entity Framework (EF) Core

EF Core is an object-relational (OR) mapper specifically for Microsoft .NET and due to the team's choice of the Microsoft stack, this tool integrates seamlessly into project development. Using EF Core, the team can develop the web application utilizing a database yet strip away the usual complexities of SQL. By removing the necessity of SQL, EF Core also promotes code maintainability and readability. SQL statements can get complicated and EF Core thereby enhances the productivity of a developer. Although database knowledge is still needed database schema creation, object-oriented code will be at the forefront of the database management.

EF Core also has a very useful tool for data migrations which allows for the application data to always be up to date with database schema. So, any changes made to the EF model will be reflected to the application, keeping any current data intact, and remove the necessity of constant database dropping recreation. Finally, a main reason the team has decided to use EF Core is due to its ability to work with many different database options. By abstracting out the database, this enables the project to be maintainable and adaptable to the needs of the product may it fluctuate over time. The project will leverage an in-memory database for testing purposes, and SQLite for development since it is lightweight. As customer demands grow the database can scale appropriately. Such as, the team can opt for SQL Server since it is a choice database for enterprise-level data management.

C. API Protocol

The team has decided to utilize a REST API, which is an architecture style which simplifies data communication between devices on the internet. MQTT was considered since it is a popular protocol in the world of IoT, however due to the simplicity and the team's familiarity with RESTful APIs, REST was ultimately chosen.

D. Data Message Format

JSON has been chosen as the format for which the data is sent to and read by the API. JSON is already integrated to work with directly with JavaScript, and the speed of which messages are transferred are notably fast due to comparatively small file sizes than that of using XML. Therefore, due to overall ease of use and the team's familiarity with the format, JSON was chosen. Below is a shortened example of a JSON-formatted message that is sent to the API, validated, and used to update the web application with new values detected by the sensors. During garden operations, the messages will include sensor values for "id" values from 1 to 5 inclusive in order to support the 5 available moisture sensors in the Smart Herb Garden design. However, due to the flexible nature of JSON format, this can easily be expanded upon and promotes the project's scalability.


```
{
 "sensors": [
  {
   "sensorID": 1,
   "values": {
   "moisture": 32,
   "temp": 99,
   "humidity": 44
    }
  },
…
}
```


E. Cloud Service Provider

Microsoft was named a leader for Cloud Platforms as depicted by the 2021 Gartner® Magic Quadrant™ and due to the team's decision to leverage the Microsoft stack, naturally it has been decided upon to use Microsoft Azure as the cloud service provider. The team also recognizes several advantages such as the fact it is built into the Visual Studio integrated development environment, it implements DevOps practices including pipelines for CI/CD, and ultimately the team has previous experience with this cloud service provider specifically.

F. User Interface (UI)

The popular Bootstrap CSS framework was used which allows for ease of front-end development for various CSS components which grants the application with a professional look and feel. Once the user interface is established, the custom-built functionality of the Smart Herb Garden's web application is developed through the software tools and architecture previously described.

IV. CONCLUSION

The Smart Herb Garden is an excellent option for a small-scale plant management system. The system will automatically water plants when the moisture levels are low, alert users of undesired environmental conditions and actions the Smart Herb Garden takes and allows for convenience and versatility in gardening. The goal is to implement the absolute requirements and basic functionality a user can expect to experience. The project currently involves robotics, sensors, microcontrollers, and a web application. The Senior Design project will be an excellent demonstration of the skills that have been acquired throughout the years in Electronics and Computer Science.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the assistance and support of the team at Sparton Corporation, especially Angi Drew and Aldo Powell for their expertise in microsoldering and PCB design with Altium Designer.

We would also like to acknowledge and thank Dr. Lei Wei for the assistance throughout these semesters. His input has been extremely valuable to our success in this project.

Fig. 10 Our Sponsor's Logo

Finally, we also offer thanks to the owners of DC Carpet for sponsoring our project. Their extensive knowledge of the tools and materials needed to create the robotic structure greatly affected the structural integrity of the garden. They also offered insight into the user base being in the home improvement business.

BIOGRAPHY

Charlee Mione is a 23-year-old Computer Engineering student with interests in pursuing a career in robotics. She has been tutoring K-12 students in math throughout her college years and hopes to continue sharing her love for math for years to come. Charlee plans on enrolling in graduate school this Spring.

Marcus Fernandez DeCastro is a 22-year-old Computer Engineering student who has completed an extended 6-month internship at Stryker doing medical robotics. Marcus plans to continue a second internship during the summer at Stryker and applying for graduate school in the Spring.

Lucas Jaramillo is a 22-year-old Computer Engineering student who is currently in a 6-month electrical engineering internship with maritime defense contractor, Sparton Corporation. Lucas plans on continuing with his career path after graduation with the intent to put his education into practice and gain more hands-on experience.

Jordyn Hayden is a 22-year old Computer Engineering student a part of the Tau Beta Pi engineering honor society, participates in the University of Central Florida's CECS Accelerated Bachelor's to Master's program, and has held several internships throughout her academic career. Jordyn is looking forward to pursuing her graduate degree in computer engineering in the near future.

REFERENCES

[1] *1A, 3-22V, 5µA IQ, Synchronous, Step-Down Converter in 1.5x2mm QFN.* Monolithic Power Systems, https://www.monolithicpower.com/en/documentview/productdocument/index/version/2/document_type/Datasheet/lang/en/sku/MP2322/document_id/4852/.

[2] Allegro Microsystems. (n.d.). DMOS Microstepping Driver with Translator and Overcurrent Protection. A4988 Stepper Motor Driver. Retrieved from http://www.allegromicro.com/

[3] Shultz, George Patrick. "Fractional-Horsepower Motors." Transformers and Motors, Newnes, Boston, 1989.

[4] Cope, Liam, and Liam CopeHi. "Normally Open vs Normally Closed: What Do They Mean?" Engineer Fix, 24 Oct. 2022, https://engineerfix.com/normally-open-vs-normally-closed-what-do-they-mean/.

[5] PBC Linear. Stepper Motor Nema 17. https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf.