

ECE 1000 Final Report: Wifi Auto Plant Waterer

Bryce Lander
Electrical and Computer Engineering (of Affiliation)
Tennessee Tech (of Affiliation)
Cookeville, United States
balander42@tntech.edu

Abstract—The project that I have made is a Automatic Plant Waterer. I am doing this project due to my dad's love of plants I thought it would be a nice thing to be able to make something capable of automatically waterer plants eventually with more time multiple all from one spot instead of having to individually go to all the plant waterers. My results I was able to successfully connect two raspberry pi pico wh's together via wifi and use one to create instructions for the plant waterer then send that data over to it and have the plant waterer be able to use those instructions properly.

Keywords—Data Transfer, Wifi

I. INTRODUCTION

The motivation behind this project is my own personal reasons of wanting to be able to create something for my dad to be able to water some of his plants automatically all from one device instead of multiple which could become hard to keep track of. My name is Bryce Lander and I am a Freshman pursuing a degree in Computer Engineering with a concentration in hardware and system security. This took quite of long time to create a lot of my own doing of going down fixes that were not as problematic as I had though.

II. BACKGROUND

I used multiple different sources to help me learn how to develop this project. I used two gitHubs to get the ssd1306 and picoBreadboard libraries I needed for my code. I watching a youtube video to teach me how to use my power supply and how to connect my Pico w's together via wifi. I found a tomshardware web page to show me how to connect OLED displays to my pico paired with a pihut blog to show me the .fill(0) function which allowed me to clear my screen to update my displays. I then used a how2electronics and instructables webpage to introduce me to wiring my soil moisture, relay, and water pump into my system and introduce me to how to code those things to work properly and most importantly learning what are the necessary functions from my utilized libraries I ne

III. PROJECT DESCRIPTION AND FORMULATION

How my project works is that one my server Pico w that programs my plant waterer pico I have three buttons connected into my GP16, GP17, and GP18 and then I have two OLED displays my display 1 having its SDA and SCL pins connected into GP0 and GP 1 and my display 2 having its SDA and SCL pins connected into GP2 and GP3. These two displays then are powered by the 5V VBUS pin from the pico in parallel. These connections stands the same for both devices. When you boot up the systems they both have the connection status's display onto the screens as they connect to Wi-Fi and the server and client connect to each other. On the programming device it then gives a display that tells you which buttons to press to lower are raise the water moisture level and lets you program what water moisture to start watering the plant then to click button 3 to continue which then brings you to program when to stop watering the plant and the button 3 finishes the setup process and proceeds to send the data over to the plant waterer and moves to a default screen. The Plant Water pico then takes these instructions and uses them and every second the Pico records data and then after 10 seconds averages the data to check if it needs to be water and then responds accordingly and if it does it will water the plant until it reaches the threshold it was set to stop watering at. The soil moisture sensor it plugged into the ADC pin G27 and is powered by the 3.3V pin on the Pico in parallel with the relay which the relay's input pin is connected into GP13. The water pump is then powered separately by a 5v power supply and is connected into the NO side of the relay on its negative side as I had troubles with it powering on before it needed to in any other different configuration.

IV. DISCUSSION AND RESULTS

The results of the project were to be able to program instructions from one pico that then send that data to the other Plant Waterer Pico which is then able to successfully use that data to properly run its functions as it is designed to. Things that I would change for when I continue to work on this project is to add the ability to program multiple rasberry pi pico's and then also reset plant waterer's I want to say if I change the plant its on. I'd also like to add more sensors and get more accurate readings on the plants to more

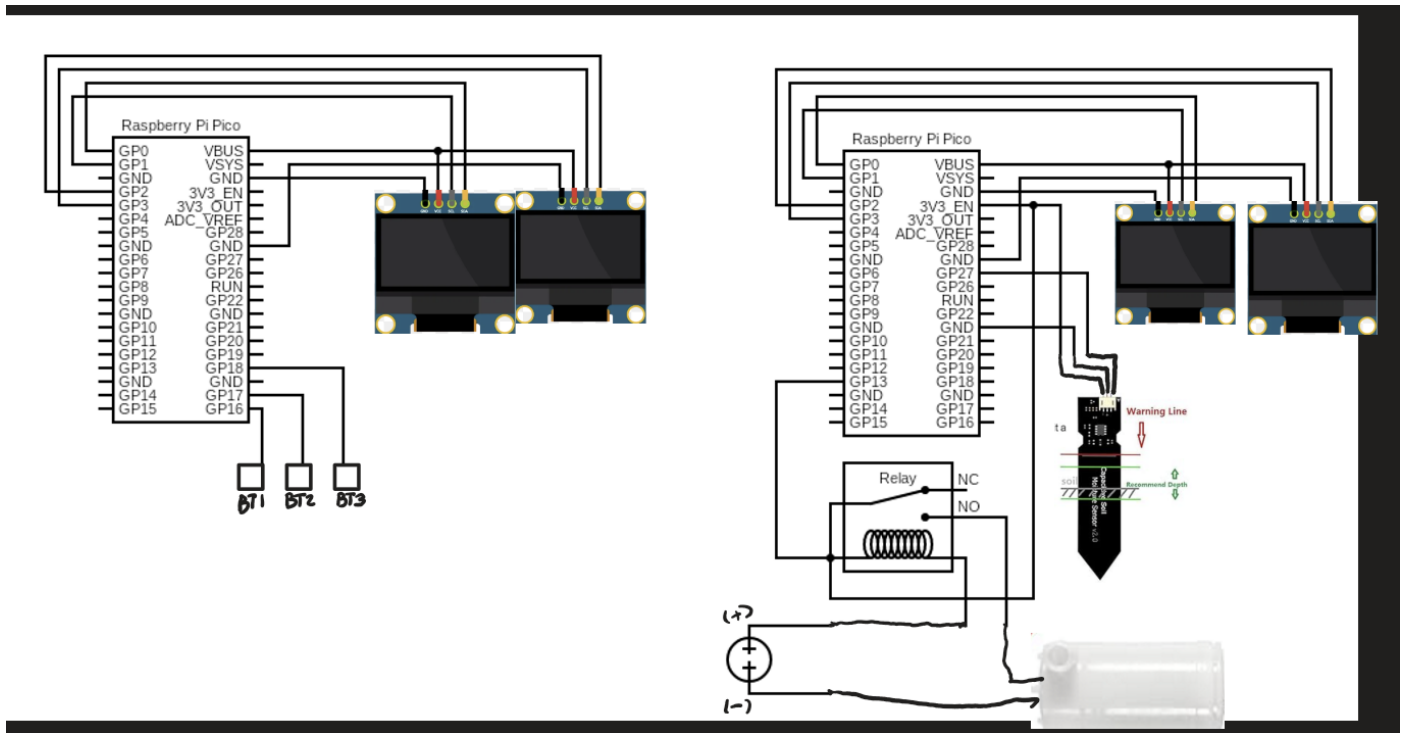
accurately be able to water them maybe even potentially adding misting to the system for plants that need it. What I enjoyed about this project was the process of going through and figuring out things and trying them out and even when I got stuck on something once I finally figured it out it was extremely rewarding. All tasks were completed by me as I did this project alone.

V. CONCLUSION

The purpose of this project was to be able to create an Auto Plant Watering system that is capable of being programmed and given its instructions via Wi-Fi. The skills I acquired through this project were better knowledge on wiring and how circuits work, I learned how to code in python better as well as learned a bunch of new libraries and some of their functions. I also started to learn how Wi-Fi works and getting devices to interact with each other over a subnet. The results of my project was being about to program a Automatic Plant Waterer via Wi-Fi and be able to have that Auto Plant Waterer be able to utilize the instructions sent to it by the device it was connected to by Wi-Fi.

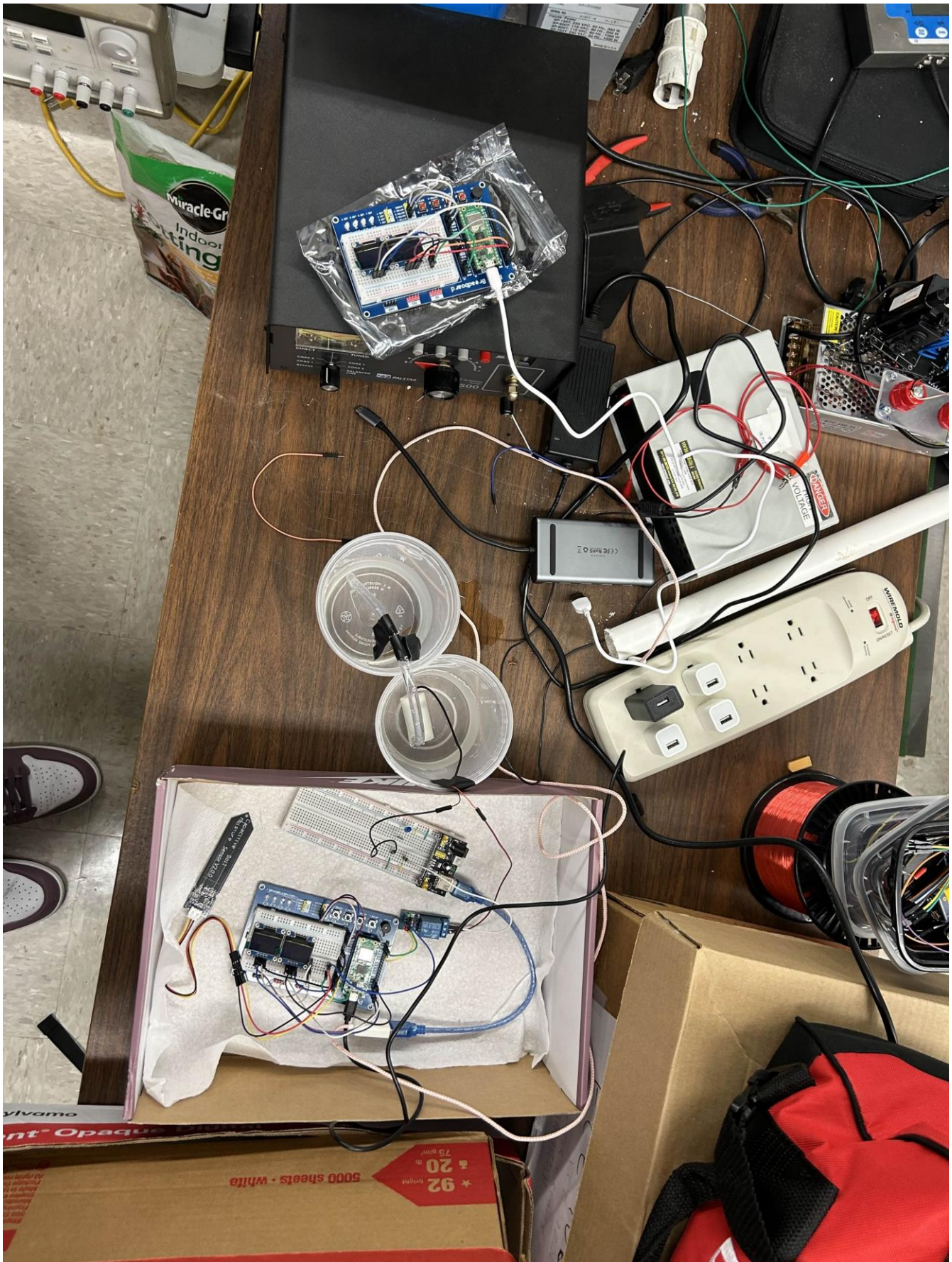
REFERENCES

- Admin. "Capacitive Soil Moisture Sensor with Raspberry Pi Pico." *How To Electronics*, 6 Aug. 2023, how2electronics.com/capacitive-soil-moisture-sensor-with-raspberry-pi-pico/.
- Chidiac, Collin, and Instructables. "Automatic Raspberry Pico W Watering System." *Instructables*, Instructables, 15 May 2023, www.instructables.com/Automatic-Raspberry-Pico-W-Watering-System/.
- electronzapdotcom. "How to Use a Breadboard Power Supply Module for DIY Learning Electronics." *YouTube*, YouTube, 3 Sept. 2020, www.youtube.com/watch?v=1er6XQ-BMp4.
- Goodhew, Tony. "MicroPython Skill Builders - #7 SPI OLED Displays." *The Pi Hut*, 14 Dec. 2023, [thepihut.com/blogs/raspberry-pi-tutorials/micropython-skill-builders-7-spi-oled-displays#:~:text=The%20pixel%20at%20the%20top%20right%20is%20at%20\(0%2C%20127,the%20screen%20with%20BLACK%20pixels](https://thepihut.com/blogs/raspberry-pi-tutorials/micropython-skill-builders-7-spi-oled-displays#:~:text=The%20pixel%20at%20the%20top%20right%20is%20at%20(0%2C%20127,the%20screen%20with%20BLACK%20pixels).
- Makerportal. "Rpi-Pico-Ssd1306/Micropython/Logo_display/Ssd1306.Py at Main · Makerportal/RPI-Pico-SSD1306." *GitHub*, github.com/makerportal/rpi-pico-ssd1306/blob/main/micropython/logo_display/ssd1306.py. Accessed 6 Dec. 2024.
- Pounder, Les. "How to Use an OLED Display with Raspberry Pi Pico (Updated)." *Tom's Hardware*, Tom's Hardware, 23 Apr. 2021, www.tomshardware.com/how-to/oled-display-raspberry-pi-pico.
- Sbcshop. "Raspberry-Pi-Pico-Breadboard-Kit/Picobreadboard.Py at Main · Sbcshop/Raspberry-Pi-Pico-Breadboard-Kit." *GitHub*, 2023, github.com/sbcshop/Raspberry-Pi-Pico-Breadboard-Kit/blob/main/PicoBreadboard.py.
- Shilleh. "Tutorial: Sending Data between Multiple Raspberry Pi Pico W's in Thonny." *YouTube*, YouTube, 27 May 2023, www.youtube.com/watch?v=rJ2A3E6z4wQ.



^ System Diagram ^

BELOW IS PICTURE OF THE SYSTEM



ECE 1000 Spring 2024 Final Project