*ECE 1000 Final Report: Automatic Plant Watering System*

Alex Bailey, Parker Foley

Tennessee Technological University

Department of Electrical and Computer Engineering

Cookeville, TN  
[abailey43@tntech.edu](mailto:abailey43@tntech.edu), [pjfoley42@tntech.edu](mailto:pjfoley42@tntech.edu)

***Abstract*** **-** **This report outlines the development of an automatic plant watering system revolved around the Raspberry Pi Pico microcontroller. The system integrates a moisture sensor to monitor soil hydration levels and a relay-controlled water pump to irrigate plants as needed. The project encompasses hardware assembly, including precise pin configurations on a breadboard, and software development using Thonny, with specific attention to the unique pin assignments of the Pico. Challenges encountered during the build process are discussed, providing insights into troubleshooting and calibration. The resulting system offers a cost-effective solution for automated plant care, with potential enhancements all around.**

*Introduction*

The motivation behind the automatic plant watering system is not only its real world application, but the possible expansion options this technology has to offer. Not only does this device teach and demonstrate fundamentals of engineering and problem solving, but it also provides a strong foundation to apply more engineering concepts to create various different devices, all with vastly different applications, but using the same core technology.

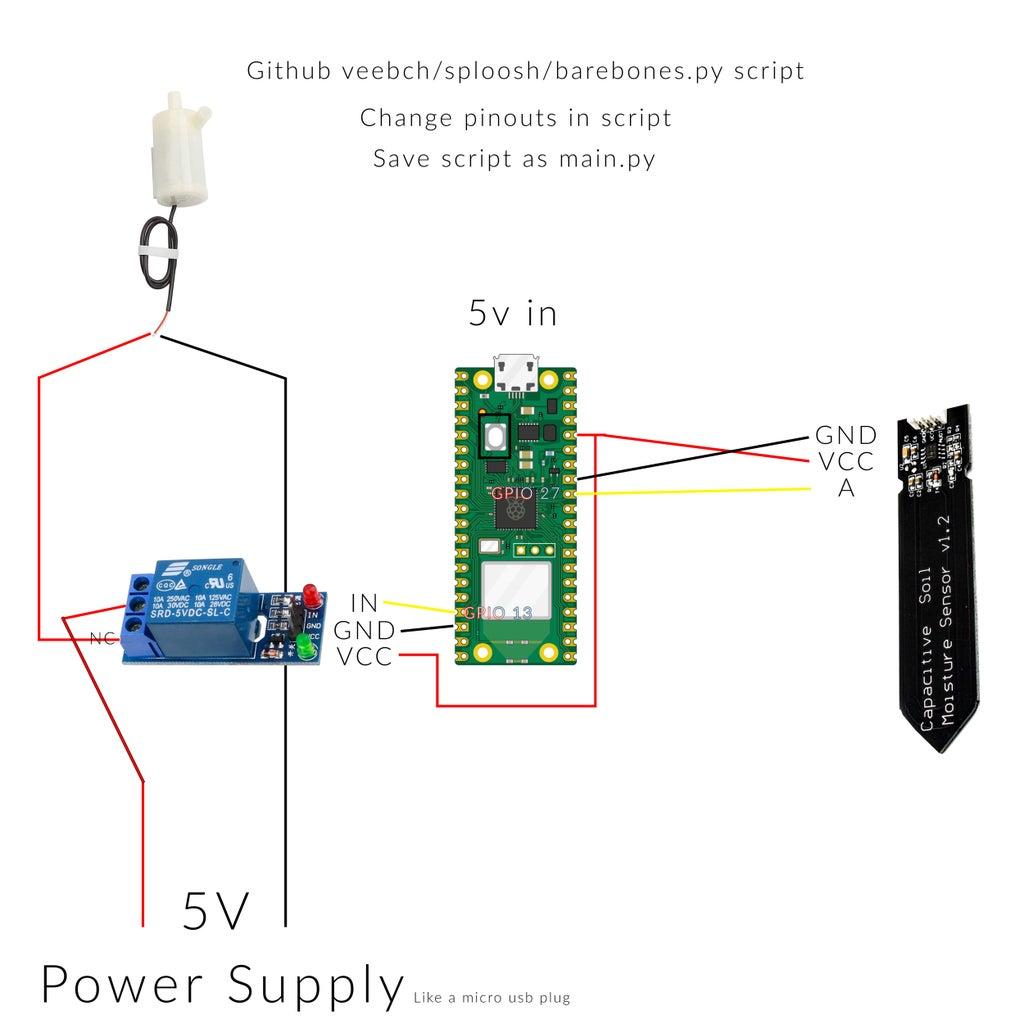
*Background*

**Materials:**

1. Raspberry Pi Pico: The Pi Pico is a budget-oriented, low power microcontroller that serves as a great platform for many projects. In this project, the Pico is in charge of the logic and turning the code into a functional watering system.
2. Breakout Board: The breakout board is a complex, but small form factor breadboard that has built in connectivity() for the Pi Pico. While it is not necessary to use this board, it makes the project a lot easier to set up, maintain, organize, and troubleshoot. The breakout board also gives us plenty of room to expand and improve this system.
3. Moisture Probe: The Moisture Probe is essentially just a giant capacitor that uses water as the dielectric. There are two electrodes, and the more water the medium it is inserted into has, the more capacitance there will be, which increases the voltage sent to the Pico, which then controls if the water will be added to the medium, in this case, soil.
4. Water Container: In this case, our container is a cup. The cup is used to store the water that is added to the soil.
5. Tubes: The tubes will allow the water to get moved from the cup, to the soil without worrying about any spills.
6. DC Water Pump: The DC Water pump uses a motor to pump water from our cup into the soil. It is connected to a relay that communicates with the moisture probe to know when to pump water into the soil.
7. Computer running Thonny: Thonny is an integrated development environment that allows the execution of python code to various different microcontrollers. Thonny is what allows the Pico to run the code.

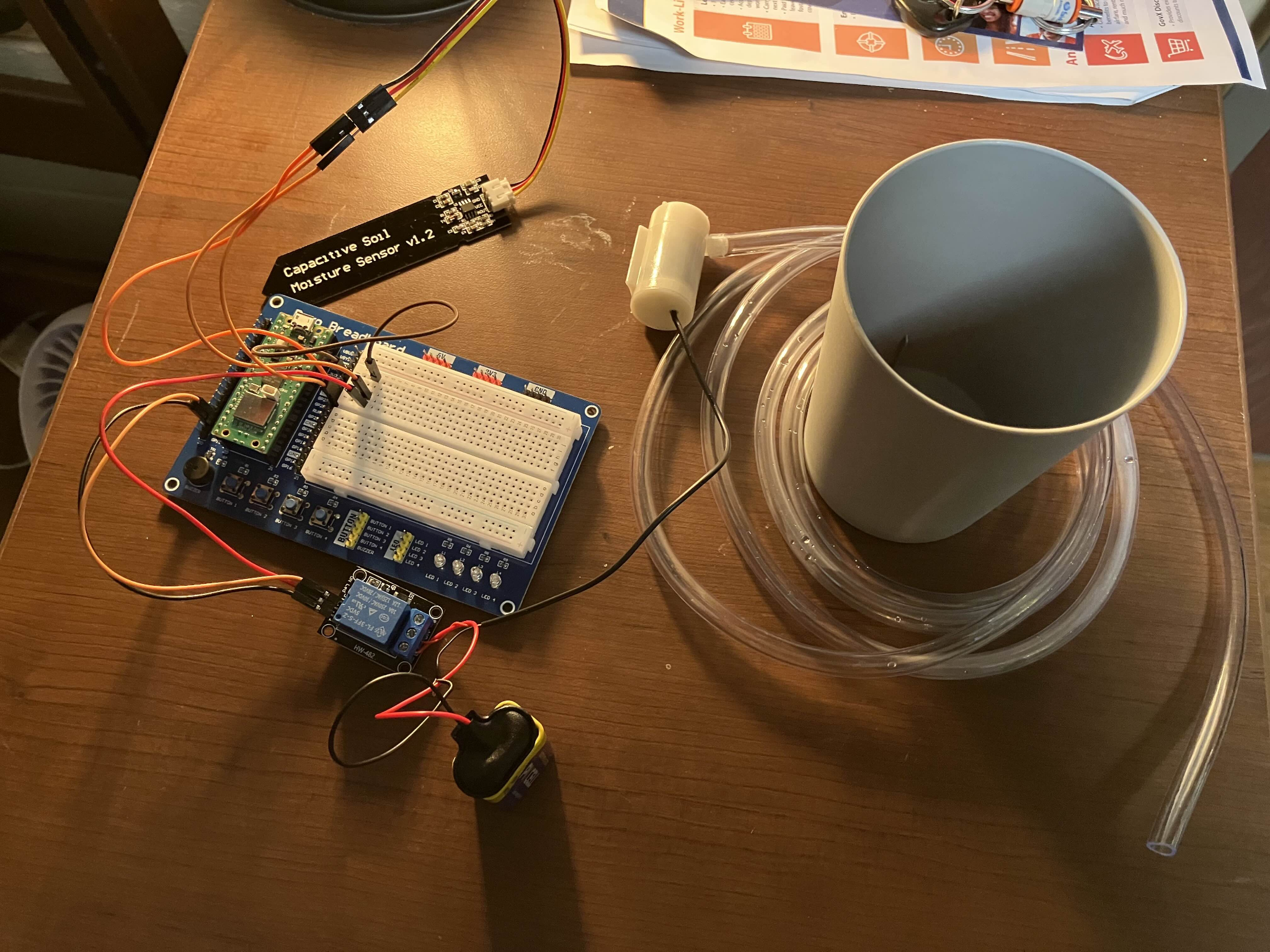
**Diagram:**

Figure 1 shows the basic layout of components that are used in our system. The diagram was featured in Collin Chidiac’s guide.

Figure 1: Simulated System  


**Full System:**  
Figure 2 shows our working system built in person. We did not have a plant to test on at the time the picture was taken, so we used a second cup to dispense the water into. The moisture sensor can use a different dielectric to work, so if you squeeze the sensor hard enough, it will think there is enough water and will quit dispensing.

Figure 1: Real System



**Functionality:**

The Pico joins all of the components together with its General Purpose Input/Output pins. These pins allow for each component to communicate with each other very quickly. However, the Pico doesn’t instinctively know what to do, so code is run from an external source via Thonny. Once the Pico receives the code, it takes full control of everything until stopped.

The Moisture probe generates an analog signal, which is converted in real time to a digital signal via pin 34 on the Pico. A weaker analog signal corresponds to a lower capacitance, which is translated to a lower voltage. If the probe detects a higher capacitance, the analog signal will be stronger, sending a higher voltage to the Pico to alert the Pico that there is enough water in the soil.

If the moisture level in the soil falls below a given value, then the probe will alert the Pico, and the Pico will send power to the DC Water Pump. The pump will then send a certain amount of water into the soil. If more water is required, then the pump will activate again until there is enough water in the soil.

*Discussion and Results*

The plant watering system worked successfully, it watered our target plant when needed, and stopped watering once the soil was fully moisturized. No issues were found, besides a potential error within Thonny that does not stop the Pico from executing the code unless the power is cut from the Pico. Using another IDE would likely solve this problem in the future.

There are a lot of different approaches one could take to further enhance this device due to its expandable nature. The code could be adjusted to allow the pump to send different amounts of water throughout the day. Or going another direction, the moisture sensor could be used to detect moisture in environments where there should not be moisture, and cut power to prevent damage to other things.

*Conclusion*

The Plant Watering System represents a platform that can both introduce the basics of electrical engineering while being easy to expand upon. The accessibility this system provides can open the doors to electrical engineering to anyone, while being able to be morphed into whatever somebody would desire it to be.

*References*

[1] Collin Chidiac. Automatic Raspberry Pico W Watering System. Instructables. https://www.instructables.com/Automatic-Raspberry-Pico-W-Watering-System/