*ECE 1000 Final Report: Automatic Plant Watering System*

Hannah Poston, Lydia Garrett, Lloyd Galvez

Tennessee Technological University

Department of Electrical and Computer Engineering

Cookeville, TN, USA

[haposton42@tntech.edu](mailto:haposton42@tntech.edu), [lggarrett42@tntech.edu](mailto:lggarrett42@tntech.edu), [ldgalvez42@tntech.edu](mailto:ldgalvez42@tntech.edu)

***Abstract—*The Automated Watering Plant System mentioned in this report applies the knowledge of many principles of engineering, including mechanical and electrical, the usage of physical electronic components, and programming, using the Raspberry Pi Pico microcontroller. This system measures water percentages in soil by utilizing a soil moisture probe, and activates the water pump, through the relay, as necessary. This report explores the design, operation, and potential benefits, of the Automatic Plant Watering System, emphasizing its role in promoting environmental sustainability and efficiency.**

1. INTRODUCTION

Driven by the interest in automation and plant life, we explored a possible solution for watering automation by integrating microcontroller programming, physical electronic components, and Electrical and Mechanical Engineering. This project helps enhance our understanding of Engineering and Engineering principles as well as signifying our interest of environmental and agricultural sustainability. Comprised by Hannah Poston, Lydia Garrett, and Lloyd Galvez, our team is devoted to using technology to improve agricultural productivity and water sustainability.

1. BACKGROUND

When developing our automatic plant watering system, we relied on a multitude of resources. We utilized Tinker cad’s simulation software as well as referencing instructables for circuit design and programming. Our citations have been referenced below to acknowledge any outside aid we have received on this project.

1. PROJECT DESCRIPTION AND FORMULATION

**Materials:**

1. Raspberry Pi Pico : Serves as a central processing unit to analyze data and control other physical electronic components.
2. Soil Moisture Probe : Measures the moisture level at constant intervals. Consists of a large diode that measures the soils’ electrical conductivity which correlates to the moisture level.
3. Water Tank (Cup) : Acts as a storage unit for water to be pumped into the plant.
4. Tube : Transports water from the water tank to the plant’s soil and roots.
5. DC Motor Water Pump : Draws water from the tank to the tubes to water the plant.

**Diagram:**

The diagram below shows the circuit with an Arduino instead of a Raspberry Pi Pico due to the limited library in Tinker Cad; however, the functionality should still be the same due to the Arduino also being a microcontroller. This diagram also portrays the adaptability of this design with other microcontroller sources.

A computer screen shot of a circuit board

Description automatically generated

**Full System:**

The next photo shows our system including the Raspberry Pi, Relay, Water Pump, and Moisture Sensor Probe. This picture doesn’t represent the tube or water tank; however, the addition of those elements would be attached to the pump, causing water to be pushed through the tube and into the soil of the plant.

A circuit board with wires

Description automatically generated

**Functionality:**

Physical components that are utilized in this lab are connected through GPIO Pins that are connected to the Raspberry Pi Pico. The Code running on the Pico is communication through those pins to each of the components, allowing the relay to turn off and on, resulting in the water pump to turn off and on, based on the data collected from the soil sensor.

The Raspberry Pi Pico converts the analog data collected from the Soil Moisture Sensor and transforms it into a digital signal (due to the microcontrollers’ Analog-to-Digital Converter that is built into it) that will be useful for the controller. The Raspberry Pi Pico can collect this data by correlating higher moisture levels from the sensor with low electrical resistances in the soil.

Based on the reading that the soil sensor collects, the Raspberry Pi Pico will determine whether the moisture level is above or below the moisture threshold we set and decide whether to turn on or off the relay. The threshold value was determined by seeing the “appropriate” moisture level of soil for plants and allowing that threshold value determine if the relay will be on or off.

If the moisture level is below the threshold, then the Pico will turn on the relay that is connected to the pump. This will allow water to be added to the plant until the threshold is reached. Even after appropriate hydration is reached, the system will continuously monitor the moisture level, if the code is running, to see when to activate the motor again.

1. DISCUSSION AND RESULTS

The automatic watering plant system proved to be successful in reading the moisture sensor and correlating that percentage to starting the relay. Even though our project ended up being successful, we went through many trials. In the beginning, we couldn’t accurately read and start the motor. After a further look into the code, and the entire circuit design, we deduced that there was an error in the functionality of our original code that needed to be rectified. After fixing this, and our microcontroller, the system was able to accurately read and respond to changes in the moisture percentage. In the future, the design of this system can be improved by refining the reaction time of the system and increasing the scale of this system to aid in the maintenance of modern agricultural projects.

Additionally, to improve the functionality of the project, the team may consider the addition of monitoring systems that are able to control the system remotely on a larger scale. This would improve the versatility and usage of this project for other agricultural and sustainability projects. Overall, we enjoyed many parts of this project, including the circuit design and the coding introduction we had while using the Pi Pico.

Individual Contributions:

Hannah Poston was responsible for formulating and testing the code on the Raspberry Pi Pico. This includes the research and development of the statements used for each physical component as well as general code trial testing.

Lydia Garrett undertook the task of sourcing the materials and main circuitry for the system. She also provided insight into the formulation and analysis of the 3D design for this project.

Lloyd Galvez modified code, helped with the formulation of the main circuit, and ensured that the system remained at optimum efficiency. He kept the project organized and orderly as we went along the project.

By utilizing each persons’ strengths, we were able to sort through any issues that came our way when developing this system. With our teamwork, we were able to achieve each of the objectives we needed to reach.

1. CONCLUSION

The Raspberry Pi Pico serves as a vital central processor for collecting moisture data, analyzing and transferring data into different units, and controlling the physical electronic elements in the system.

Its integration capabilities make this component vital for this system. Without the Raspberry Pi Pico, many of the functions needed for this automation wouldn’t be able to operate. Its quick analysis capabilities allow for infinite opportunities with different projects.

The Automated Plant Watering System is a representation of the culmination of engineering principles and values as well as innovation to aid in water management and efficiency.

REFERENCES

Chidiac, C., & Instructables. (2023, May 15). *Automatic raspberry pico W watering system*. Instructables. https://www.instructables.com/Automatic-Raspberry-Pico-W-Watering-System/

JCWilliams1003. (n.d.). *ECE-1000-spring-2024-final-project-insert-project-name/example Micropython Codes/ECE\_1000\_Soil\_Moisture\_Sensor\_Example.py at main · JCWILLIAMS1003/ECE-1000-SPRING-2024-FINAL-PROJECT-INSERT-PROJECT-NAME*. GitHub. https://github.com/JCWilliams1003/ECE-1000-Spring-2024-Final-Project-Insert-Project-Name/blob/main/Example%20Micropython%20Codes/ECE\_1000\_Soil\_Moisture\_Sensor\_Example.py

OpenAI. (2022). ChatGPT (Version 3.5) [Computer software].  
https://openai.com/chatgp