*ECE 1000 Final Report: Automatic Plant Waterer*

Gavin Cooper & Jackson Hamblin  
[gcooper42@tntech.edu](mailto:gcooper42@tntech.edu), [jhamblin42@tntech.edu](mailto:jhamblin42@tntech.edu)

*My partner and I decided to design, build, and program an Automatic Plant Watering System. We chose this project as a way to further our understanding of basic programming and circuit construction. The project had a few bumps in the road as any project would but ultimately, it was a success. During the initial testing, we had a few issues with the code and a few simple wiring issues. Upon fixing these issues, the project performed exactly as planned.*

# Introduction

Jackson and Gavin, both Electrical Engineering Majors, designed, built, and programmed an apparatus that automatically waters plants. The Automatic Plant Watering system can appear to be a simple apparatus; however, it is a little more complex. The apparatus uses a sensor to detect moisture level. When the moisture level declines to a certain percentage, a relay is triggered. This allows for a pump to turn on and feed water into a plant. Once the soil reaches a certain moisture level, the relay is then triggered again, shutting the pump off. Aside from a few simple problems with the code and wiring, the apparatus performed very well.

# Background

The initial construction of the project came with a few hiccups. When we first tested the project, we realized there was a wring issue that was causing the relay to not send power to the pump. We seeked assistance with this from a website known as “Autodesk Instructables.” We came to the realization that the issue was simply just mixed connection from the relay to the pump. We later realized we were having issues with our code completing a loop. Following collaboration with fellow peers, we were able to identify the issue and get it corrected allowing the apparatus to perform continuously.

# Project Description and formulation

While the idea of an automatic plant waterer might sound complex, our circuit is quite simple. With the help of the Raspberry Pi Pico and a simple MicroPyhton code, our soil moisture sensor sends a reading to our Pi Pico, which is then turned into a percentage in the code. Using an if statement in MicroPython, that percentage is determined as sufficient for the plant, causing our submersible water pump to shut off or remain off, or insufficient for the plant, causing the pump to turn on or remain on. As far as the wiring goes for our plant waterer, we decided to have the signal that comes from the moisture sensor run into GP27 of the breakout board. The signal coming from the Pi Pico into the relay for the pump we had coming out of GP13.

# IV. Discussion and results

The results were exactly as we had hoped. We were able to make the apparatus detect moisture level and pump or not pump water exactly when needed. One improvement we would make would consist of powering the pump using a separate power source allowing there to be a display that would accurately show what stage the apparatus is in as well as show the readings and the actions being performed. The team enjoyed all aspects of the project, most of all being the callaboration and troubleshooting. Mostly the entire project was done collaboratively with a few minor aspects being done individually. Gavin took the lead on housing the parts needed for the project while Jackson took a lead on scheduling times for meetings.

# V. Conclusion

Ultimately the project was a success. We were able to accurately take moisture readings, use the readings to signal a relay, and turn a water pump on or off. The main skills we obtained from the project were troubleshooting and collaborating skills as well as programming basics using MicroPython language.

##### References

1. https://www.instructables.com/Automatic-Raspberry-Pico-W-Watering-System/

U,{710450b3-c68d-4502-8b88-afe6b595042a}{222},12.722222805023193,7.4999996821085615