ECE 1000 Final Project Report: Automatic Plant Watering System

Dalton (Levi) Cantrell, Brenya Dickens

Tennessee Technological University
Department of Electrical and Computer Engineering

Cookeville, TN, USA

dlcantrell42@tntech.edu , brdickens42@tntech.edu

***Abstract* – The goal of the system outlined within this report is to supply a plant with water with nearly complete autonomy. The design incorporates use of a Raspberry Pi Pico breakout board, 5v DC submersible water pump, soil moisture sensor, and micropython code to control the behavior of the system. While the system primarily runs autonomously, the design also incorporates a button for user input that allows watering the plant manually. Our observations of the finished system concluded that user input was not necessary if the desired moisture levels were calibrated correctly.**

I. Introduction

This project is aimed at providing a failsafe to caring for plants, whether for decoration or consumption. Many plants have a volatile nature when water is concerned, leading to severe consequences if the caretaker were unable provide water on a regular schedule. The system we developed is capable of monitoring the moisture content of the soil for a given plant and automatically dispensing water at the desired parameters. Preservation of water resources in cultivation is another key utilization for such a system on a global scale. While our team's motivation for working on this project was our domestic concern for houseplants, this system is viable for use in arid climates where control of water supply is also vital to food supply.

II. Background

The development of this system was placed on a relatively brief schedule. As a result, the design was heavily influenced by referenced materials on the use of relays, a similar system designed by Collin Chidiac, his programming to detect and dispense proper amounts of water, and a box from a prior personal project used as a water storage container. Alterations to these source materials lead to an inversion of logic for the relay to correctly dispense water at desired intervals instead of constantly, variance in timespans that the system checks moisture levels, and a user input to allow manual watering via button press.

III. Materials

Most of the materials used in this project were provided by our instructors, and as such will be detailed in the below table outlining the cost of the project as PRV (designating provided). Other materials were scavenged from the personal supplies of our team. Due to this, the cost of these materials is an estimation based on market value and weight of material used.

Table 1: Project Material Costs

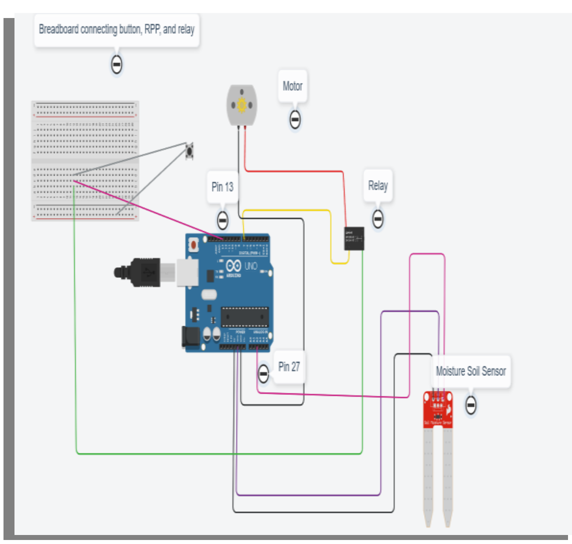
Material	Weight (kg)	Cost
Raspberry Pi Pico Breakout Board	n/a	PRV
5V DC Water Pump + Tubing	n/a	PRV
CW025 Relay Module	n/a	PRV
PLA Filament	0.145	$2.90
Yellow Hobby Paint	~ 0.040	$1.88
Acrylic Clear Coat	~ 0.040	$1.16
Total Cost		$5.94

IV. System Design

For the design of this system, we first researched how to properly connect the relay-DC motor subsystem in order to allow the Raspberry Pi to interact with the pump. One resource provided by our instructors allowed us to see what a functional system would look like in totality [1]. Another useful resource went into depth on the purpose of each connection within the relay-motor subsystem [2]. At this point, we discovered that our relay would need to be configured differently than our reference in that the connection between the motor and relay would be routed through the NO (Normally Open) port.

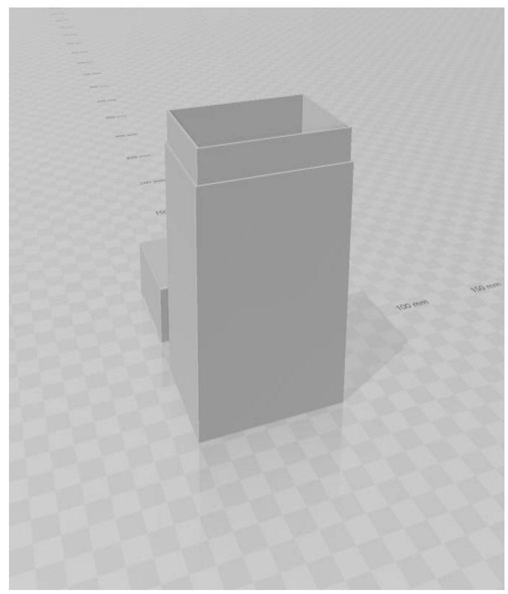
In the configuration using the NC (Normally Closed) port, the system would dispense water as a default behavior and only stop when signaled by the Raspberry Pie. With our configuration, we were able to achieve the desired effect, as well as add a button for the user to dispense manually. The physical configuration of the system is depicted in Figure 1 below, designed by Brenya Dickens. Although due to limitations with the simulation software we substituted our Raspberry Pie with an Arduino Uno model. Realistically, this configuration would achieve the same results using different hardware.

Figure 1: Tinkercad System Schematic



Next, we moved to design a water storage container with which to house the water pump and supply for the system. We decided to make use of a prior project from which Dalton Cantrell 3d printed a hobby card storage container. This specific 3d print [3] was a “failed” case where the bottom layer of the print was malformed due to heating issues. As this was not an obstacle to this project, we proceeded to sand, paint, and clear coat the container to allow it to hold water. Additionally, a hole was drilled through the lid portion to allow the tubing from the pump to pass through. This made the water storage closed off from contaminates.

Figure 2: Water Storage Container 3D Object



V. Conclusion

Throughout testing this system, we discovered few alterations were needed from our original concept. We needed to alter the logic for controlling the relay in order to allow user input and proper functionality for the pump. We wanted to implement an on-board screen to display the current moisture level of the plant, but out time constraints did not allow that portion of the project. This project has shown our team that the concept of automated plant watering systems are not as overly complicated as much as controlling the system.

REFERENCES

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

[2] Robojax. (Dec. 26, 2017). “How to use 5V Relay with Arduino to turn ON and OFF AC bulb or DC load”.https://www.youtube.com/watch?v=58XWVDnB7Ss

[3] caffeinewriter. (Feb. 25, 2013). “MTG Deck Box”. https://www.thingiverse.com/thing:54230