Research Archive of
Rising Scholars (preprint)

Where bright minds share their learnings

Dynamic plant watering system

Howard Li

SUMMARY

The purpose of this paper and project was to create a dynamic plant watering system that could adjust based on weather and sunlight data. The system utilizes data from the National Weather Service, as well as the plant's orientation and location to adjust the watering frequency. A working proof-of-concept was built and tested on an indoor potted plant over a couple of weeks. Testing showed that the system was able to successfully adjust based on local weather conditions and changing sunlight hours. This project serves as an example of how open-source data and resources can be used to make day-to-day tasks much more effortless.

INTRODUCTION

Watering a plant is usually considered a mundane task that must be done out of pure necessity. It requires people to keep track of watering schedules manually. The amount of water a plant needs changes based on conditions such as temperature, sunlight intensity, and humidity (1). Sometimes people use assistants, such as apps, to help them keep track, but those apps aren't fully comprehensive and only suggest a fixed interval between watering sessions. Other projects use local temperature and humidity sensors to determine when plants should be watered, but they require many components and don't give the user the full picture (2).

This project aims to create an automatic watering system that uses an adaptive algorithm to dynamically update its watering schedule, based on weather data collected from external sources. No local sensors are required, and all parts, except for the servo and Raspberry Pi and servo, can be 3D printed.

RESULTS

An experiment was conducted to test the effectiveness of the system and its ability to adjust to real-world weather patterns. It was conducted by running the system in the same setup for 12 days using location data of Howard’s house and generic plant data. The daily sunlight amount, 7-day rolling average, and adjusted watering frequency were recorded once a day throughout the experimental period.

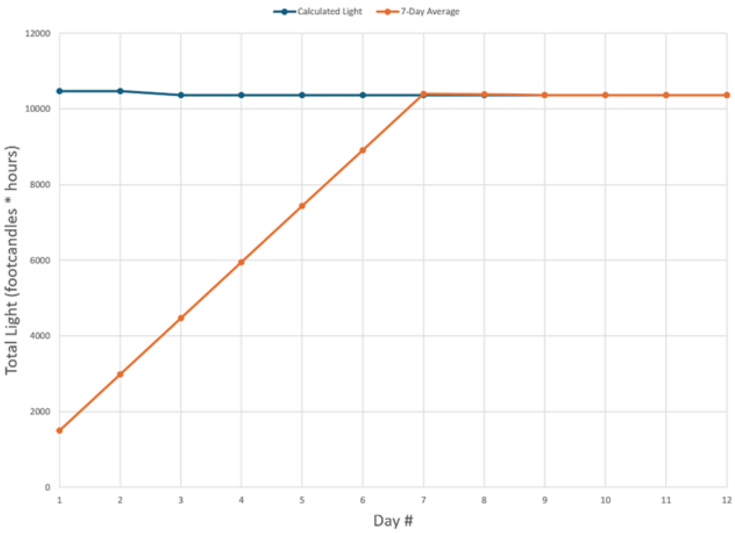

Figure 1. *Measured total light (footcandles * hours) for every day in the 12-day experimental period. Line graph showing the trend of sunlight values over the course of the experiment. Real US zip code was inputted and plant orientation was set as "East," with the program printing out calculated values every day at noon.*

Over the course of the 12-day period, the daily total sunlight value remained roughly constant at around 10,000 foot-candles * hour and the 7-day average was increasing for the first 7 days until it plateaued at around 10,000 foot-candles * hour starting from day 8, which is when the system began adjusting watering frequency (Fig. 1).

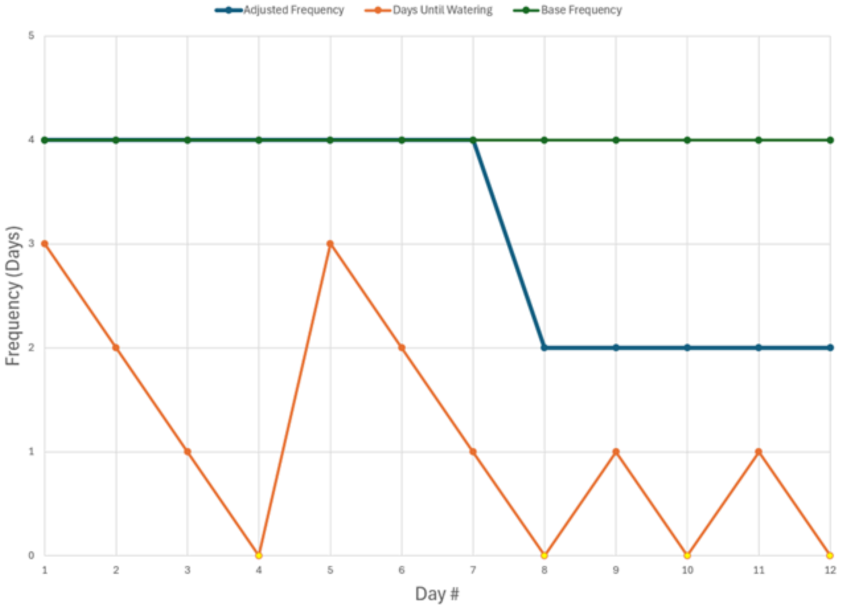

Figure 2. *Watering frequencies over the course of the experimental period. Line graph showing how the change in average sunlight lead to an adjustment in watering frequency. Generic plant with base frequency of 4 was chosen and adjusted frequency was printed out daily.*

At the start of the experimental period, the adjusted watering frequency was 4 days, equal to the generic plant's base watering frequency, due to the lack of a full 7 days' worth of sunlight data. After 7 days, the watering frequency was adjusted to 2 days based on the average daily sunlight value of 10,370 foot-candles * hour and stayed constant for the rest of the 12 days (Fig. 2).

DISCUSSION

According to experimental data, once the 7-day average was fully populated with data, the system reduced the watering frequency based on the sunlight levels being sufficient for the user-inputted plant type. The results of the experiment proved the system's ability to adjust based on National Weather Service and user-inputted data.

Letting the system run for a longer period would greatly increase the amount of data collected, leading to a more accurate picture of real-world performance. Additionally, testing the system on multiple plants would show any potential variation and the effectiveness of the adjustments on multiple samples.

The experiment's success shows the potential of the plant-watering system as a household product. It greatly reduces the amount of effort needed to keep track of a watering schedule by hand and prevents over and under-watering, helping to reduce water usage and ensure healthy plant growth.

The successful usage of publicly available data sources in this project demonstrates its effectiveness as a replacement for locally collected data. A similar principle can be utilized for other projects to make end-user operation simpler.

MATERIALS AND METHODS

This project was programmed entirely using the Python programming language, with the help of open-source libraries and the National Weather Service API. All code was run on a Raspberry Pi 4B.

Software Libraries

The Python libraries used in this project are: Datetime, used to track date and time and determine the length of intervals between watering; Pgeocode, used to convert a user-provided zip code into geological coordinates (latitude, longitude); Time, used to pause the program for intervals determined using Datetime; Gpiozero, used to interface with RPi4B GPIO pins and control servo actuator to dispense water; Numpy, used to calculate the rolling average of sunlight data across 7 days; Noaa-SDK, a Python package that serves as a wrapper for the NOAA National Weather Service API; and Suntime, used to find sunrise and sunset times, which are used to calculate the total amount of sunlight in a day.

Program Algorithm

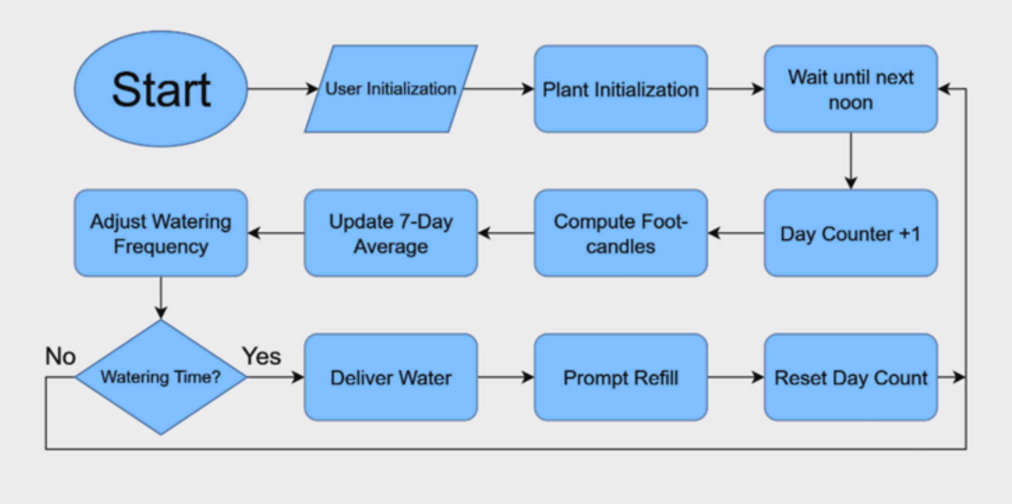

Figure 3. *Software operation diagram. Flowchart showing the inputs the program takes, the conditions it checks, and the adjustment process it goes through.*

The adjustment of the watering schedule is based on a calculation of how much light the plant will receive in a day. This number is based on several factors, such as weather conditions, plant orientation, and time of year (Fig. 3).

Weather conditions are determined by processing the short forecast that is queried from the NOAA NWS API. A possible example of such a forecast could be, "Partly Cloudy." This forecast is split into individual words and parsed to search for key phrases such as "rain," "cloud," "fog," and, "sun." If a key phrase is found, the short forecast is assigned a numerical value, *W*, in foot-candles, that corresponds with the amount of sunlight present during those conditions. For this project, "Rain" was assigned a value of 100 foot-candles, "Fog" a value of 400 foot-candles, "Cloud" a value of 700 foot-candles, and "Sun" a value of 1,000 foot-candles based on reference sources (3).

Plant orientation is user-inputted, and it can be one of four possible values: "N," for North, "E," for East, "S," for South, or "W," for West. A directional foot-candle value, *D*, is assigned based on this value. North is assigned a value of 100 foot-candles, East and West are assigned a value of 500 foot-candles, and South is assigned a value of 1,000 foot-candles (4).

The time of year affects sunlight levels because it changes how long the sun is out every day. This value, *T*, is calculated by finding the difference in time, in hours, between sunrise and sunset on the current day.

These three variables are used in the following equation to calculate the total amount of sunlight that the plant will receive in one day, *S*, in foot-candle * hours:

$$S = (W + D) * T$$

Starting from noon on the day after the program is first run, it keeps track of the number of days since the plant was last watered. Every following day at noon, the total sunlight for that day is calculated.

Once the value for each day is calculated, it is stored in a NumPy array, and a 7-day rolling average is computed and used to adjust the watering schedule. The purpose of this is to smooth out unusual weather events and highlight repeating weather patterns, which have a much more significant effect on plant growth.

The watering frequency is adjusted by mapping the 7-day average sunlight value to a number between -1 and 1 based on calculated theoretical minimum and maximum sunlight amounts. Then, this value is multiplied by a base watering frequency, dependent on the type of plant specified by the user, and added to the original base watering frequency. Currently, the code contains a dictionary of 5 different plants and their base watering data. The equation for such a calculation is:

$$F_1 = (F_0 * scale) + F_0$$

where F_1 represents adjusted frequency, F_0 represents base frequency, and *scale* represents the mapped multiplier value.

Once the count of days since the last watering equals the adjusted frequency, the servo-controlled actuator is opened, dispensing all the water inside of the reservoir into the pot. Then, the user is prompted through the terminal to refill the reservoir for the next watering session and the day count is reset.

The code for this project can be found at the link: https://github.com/howli4/PolygenceProject

Hardware Setup

Figure 4. *Real picture of the system. A photograph of what the assembled system looks like in real life, showing the individual components assembled together.*

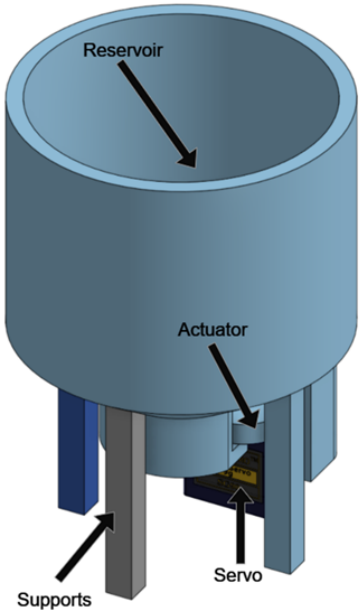

Figure 5. *Watering mechanism CAD diagram. Labeled picture of the CAD (Computer aided design) model used to 3D-print the required parts of the mechanism.*

All major components of this project's proof-of-concept are 3D printed, except for a TowerPro SG90 Servo and a Raspberry Pi 4 Model B (Fig. 4). The main physical parts of the working concept are: the reservoir, a 3D printed container that holds the water to be dispensed on the plant and has a small hole at the bottom which is plugged up by the actuator and sealed with an O-ring; the actuator, a 3D printed piece that is used to plug up the bottom of the reservoir, and is moved in order to let water flow out of it; the servo, a TowerPro SG90 digital servo controlled by a Raspberry Pi that is used to move the actuator and dispense water; a Raspberry Pi 4 Model B, a compact microcontroller that runs the project code and controls the servo; and the supports, 3D printed columns which are used to hold the reservoir, actuator, and servo above the soil and provide stability (Fig. 5).

Experiment

To test the effectiveness of this system, data such as base and adjusted watering frequency and calculated sunlight amounts were collected periodically over the course of a 12 day-long observation period. The program would print out its predicted daylight amount for the day at noon every day, as well as its calculated adjusted day interval until the next watering period. A real zip-code and plant type was used in order to simulate realistic conditions.

REFERENCES

[1] C. Cerro, “Future of dwelling: Indoor plants and produce,” WIT Transactions on Ecology and the Environment, vol. 260, pp. 493–502, 2022.

[2] K. Shah, et al., “Proposed automated plant watering system using iot,” Conference on Technologies for Future Cities, 2019.
[3] “How Much Light Is Enough?” Take 3 Lighting, www.takethreelighting.com/light-levels.html. Accessed 16 Oct. 2024.
[4] “Lighting for Indoor Plants.” University of Maryland Extension, University of Maryland, 10 Mar. 2023, extension.umd.edu/resource/lighting-indoor-plants/. Accessed 16 Oct. 2024.