generated by Scienteer.

Research Plan/Project Summary

Scienteer Project ID:	**159897**
Category:	**Embedded Systems**
Student(s):	
Project Title:	**Designing a smart solution to water plants efficiently.**
Division:	**Senior**
School Name:	**CLEAR SPRINGS HIGH SCHOOL**
School City:	**League City, Texas, US**

Rationale

The project would involve designing a smart solution to watering plants efficiently. Based on researching what conditions are best for plant irrigation and based on the rain forecast, current weather conditions, and soil moisture, the system will water the plants accordingly. This project is important because it could help alleviate unnecessary water usage by not watering when it rains, it is forecast to rain, or when the soil doesn't need it. If successful, the end product would also be cost effective (saving people money) and it would also help people who don't have the time to make sure they water their plants every day, or when the conditions are correct for ideal watering. Automating this process could end up making everyday life easier for the end user because they have one less thing to worry about.

Hypothesis

Design a smart solution to watering plants efficiently.

Materials

- A Single Board Computer
- Relay
- Pump/Solenoid
- Soil Moisture Sensor
- Various Wires
- Soldering Iron
- Prototyping Board
- Weather API Key
- An internet connection
- Power Supply
- Analog to Digital Converter or ADC
- Potted Plant
- Email Account (optional)
- Another Computer
- Waterproof Case
- Watering Tubes

Procedure

1. Find a single board computer that meets the required specifications. In this case a Raspberry Pi W.
2. Find a Weather API which will allows to see the current weather and the forecast and investigate how to use it.
3. Connect a moisture sensor to the single board computer and use Python to read the output from the sensor using an Analog to Digital Converter.
4. Integrate this with a relay system that will turn a pump/valve on or off.
5. Next, integrate this with another program that polls the sensor and the Weather API and decides if it should water the plants and make it run on a predefined schedule.
6. Then put the completed circuit onto a board for the final revision.
7. After that, put the project into a place where it will be safe from the elements, in this case a weatherproof case.
8. Finally, bring the completed project outside and put the value so that when it activates it waters the plants, and put the moisture sensor into the potted plant.

Data Analysis

A moisture sensor would be used to collect the moisture levels of the soil. Also, an integration to a weather API would be used to collect the data on the rain forecast. Finally, while the Python script runs to water the plants, it will log the actions it takes and the data it collects (moisture sensor data, weather data, etc.) to a file for further analysis. With the file a table can be made which demonstrates it working based on the moisture level, among many other charts that can be made with this data, a chart based on weather data, moisture level, and when the plants are watered would interesting

Hazardous Materials/Activities/Devices/Exempt Microorganisms

Risk Identification

The main risk involved in the project would be the use of electricity. In the case that it is used there is a chance water will short or make contact with electricity. Another risk would be the use of a soldering iron.

Safety Precautions

I will try to minimize by being cautious around exposed wires/leads (try to minimize them) and inspecting the wiring before applying voltage to prevent shorts or electric shock. I will also follow the proper safety guidelines while using a soldering iron (avoiding accidental burns from hot areas, gloves, glasses, and reading the safety sheet/caution guide before use).

Description of Potential Hazards

The main risk involved in the project would be the use of electricity. In the case that it is used there is a chance water will short or make contact with electricity. Another risk would be the use of a soldering iron.

Disposal Methods

N/A

Sources of Safety Information

The manuals/spec sheets for the sbc (single board computer) and the related components being used with it. I will also read the manuals and safety guides for the soldering iron.

Project Summary

Changes from Original Research Plan

I have slightly updated the materials list to be more specific and have slightly reworded the procedure (the steps have not changed). Below are the updated versions. Note that the soldering iron is not present because it was never used throughout the project (instead a breadboard was used).

Update Materials List:

- Raspberry Pi Zero W
- Relay Board
- Water Pump
- Water Source
- Capacitive Soil Moisture Sensor
- Various Wires
- Breadboard
- OpenWeatherMap Free API Key
- An internet connection
- Power Supply
- Analog to Digital Converter or ADC
- Potted Plant
- Another Computer
- Waterproof Enclosure
- Water Tubing

Updated Procedure:

1. Find a single board computer that meets the required specifications. In this case a Raspberry Pi Zero W will work.
2. Find a Weather API which will allows to see the current weather and the forecast and investigate how to use it. In this case OpenWeatherMap was used.
3. Connect a moisture sensor to the single board computer and use Python to read the output from the sensor using an Analog to Digital Converter.
4. Integrate this with a relay system that will turn a pump on or off and put the circuit onto the breadboard.
5. Next, integrate this with two programs that decide to water the plant and to query the Weather API, then make them run during the morning, afternoon, and night using Crontab (and also run-on boot).
6. The program should decide to water or not by checking the rain conditions and the current moisture level of the soil.
7. After that, put the project in a place where it will be safe from the elements, in this case a weatherproof case.
8. Then bring the completed project outside and situate the pump so that when it activates it waters the plants and put the moisture sensor into the potted plant.
9. To make sure that it will water at the right levels of moisture, tuning is highly recommended by seeing the moisture level of the plant dry and the moisture level of the plant manually watered and incorporating those into the program.

I have also updated the data analysis to meet the results from my project. The updated version is below.

Updated Data Analysis:

Overall, after analyzing the data it can be seen that the software preformed as expected. The Raspberry Pi only watered the plant when it had low moisture, was not raining, and it was not forecasted to rain later in the day and in the coming days.

Project Conclusions

At the beginning I set out with the goal of designing a smart solution to water plants efficiently. The hypothesis was that a system could be designed for watering plants efficiently, in the end it was designed/successful, since it watered the plants only when the correct conditions were met, thus proving the hypothesis. The success of this project is important because it can help reduce water usage by not watering when it rains, is forecasted to rain, or when the soil doesn't need it. The final solution designed was also cost effective compared to commercially available systems. One possible source of error that could have presented itself throughout the project would be external factors like temperature that might have affected some of the moisture readings. Though, since the data followed a steady trend, and mirrored the real world well, I can rule that out as a source of major error.

Bibliography

Reference Source

Family Handyman. "10 Smart and Effective Ways to Water Your Lawn." Family Handyman, Family Handyman, 24 Aug. 2020, https://www.familyhandyman.com/list/smart-and-effective-lawn-watering-tips/.

"HTML." W3Schools Online Web Tutorials, https://www.w3schools.com/

Industries, Adafruit. "ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier." Adafruit Industries, https://www.adafruit.com/product/1085.

"Lawn Water Management - How Often Should Grass Be Watered?" Texas A&M AgriLife Extension Service, 17 July 2019, https://agrilifeextension.tamu.edu/library/landscaping/lawn-water-management/.

"PHP Tutorial." PHP Tutorial, 7 Oct. 2021, https://www.phptutorial.net/.

Purkayastha, Shyam. "Top 10 Weather Apis [2021] Premium + Free Weather API (30+ Reviewed)." Rakuten RapidAPI Blog, 3 July 2021, https://blog.api.rakuten.net/top-weather-apis/.

Python.org, https://www.python.org/.

Raspberry Pi. "Raspberry Pi Zero W." Raspberry Pi, https://www.raspberrypi.com/products/raspberry-pi-zero-w/.

Staff, RapidAPI, et al. "Top 8 Best Free Weather Apis (2021) [25+ Reviewed]." The Last Call - RapidAPI Blog, 29 Sept. 2021, https://rapidapi.com/blog/access-global-weather-data-with-these-weather-apis/.

"Water Your Way to Happy Potted Plants." Proven Winners, https://www.provenwinners.com/learn/water-your-way-happy-plants.

Adafruit. "Adafruit/Fritzing-Library: Adafruit Parts, Components, Breakouts, Etc...in Fritzable Format!" Github, https://github.com/adafruit/Fritzing-Library.

Buscher, Noah. "Photo by Noah Buscher on Unsplash." Beautiful Free Images & Pictures, 19 Nov. 2018, https://unsplash.com/photos/x8ZStukS2PM.

Fritzing, https://fritzing.org/.

"Ogretransporter." GitHub, https://github.com/OgreTransporter.

OpenWeatherMap.org. "Current Weather and Forecast." OpenWeatherMap, https://openweathermap.org/.

Richard. Omnigatherum, 11 May 2021, http://omnigatherum.ca/wp/.