Plant Hydration and Weather Integration System

Aashish Chaubal, Jaeren Dadivas, Iker Uriarte

Maanas Agrawal

ECE 445

Introduction:

The world of outdoor plant care is challenged with the obstacles of weather uncertainty. In particular, a challenge resides in balancing plant hydration and minimizing water waste. Overwatering has been an obstacle for ages and it can lead to more than just the death of your plant. It can lead to beneath-the-surface issues such as root rot, fungal growth, and other plant diseases. On the other hand, underwatering plants remains a prevalent issue that can result in reduced growth and plant stress. In today's time, automated watering systems already exist to provide some convenience to plant care by irrigating plants at scheduled times. However, traditional watering systems remain inefficient with water usage as they are incapable of taking into consideration the weather conditions. The world is ever-changing and with the current concerns of weather uncertainty due to factors like climate change, the need for more efficient watering systems solutions has never been more prevalent.

Some smart irrigation systems already exist. Many of them focus primarily on moisture sensor readings without the integration of real-time weather forecasting and therefore their efficiency is limited. Without taking account for weather conditions, these systems are subject to water waste especially when rainfall is imminent. Outdoor plant owners would be able to take a greater advantage over their plants' health and growth if they made use of a system that could reduce water waste by taking into account both soil moisture readings and real-time weather forecast data.

Our project presents a solution to balancing plant hydration and minimizing water waste. We present a smart plant-watering system that utilizes the integration of both soil moisture sensors and real-time weather readings to improve water use efficiency. Our system will mainly utilize an ESP32 Microcontroller that will retrieve weather data via API (ie. to a third party app such as AccuWeather) and monitor soil moisture levels to determine whether it is necessary to activate water flow to the plant. If the weather forecasts rain precipitation throughout the day, the system will delay watering, preventing overwatering, and present the user with calculated results of water conserved due to rainwater.

Our system will include a rain detection subsystem using a weight sensor to measure and calculate accumulated rainfall. The microcontroller will calculate estimated water savings by comparing soil moisture levels before and after rainfall to identify how much water was conserved thanks to rainfall. The results of this data will be sent to a Firebase database and displayed on a website for the user to read the estimated water savings and track water consumption of the plants. The whole system will be battery-powered with rechargeable lithium-ion batteries to provide the user with better long-term sustainability and cost efficiency. Our solution provides outdoor plant care sustainability and reduces the amount of manual control required by the plant owners.

High-Level Requirements:

- Our system must accurately measure soil levels with a precision of ±5% using the moisture sensors and retrieve real-time weather data via API request at least once every 10 minutes. The weather data should update correctly and our system should store this retrieved data to be used for watering decisions.
- Our system must activate the water pump subsystem only when the soil moisture levels fall under 30% volumetric water content (VWC) and the current weather data indicate a probability of rain less than 70%. The system should respond to changes in soil moisture levels 30 seconds after a watering decision is made.

- The system must measure water savings using a weight sensor with a minimum resolution of 10 grams so that it can best detect changes in weight before and after watering or rainfall. Data from the sensor must be updated to Firebase within 5 seconds and displayed to the user via UI every 15 minutes.

Design

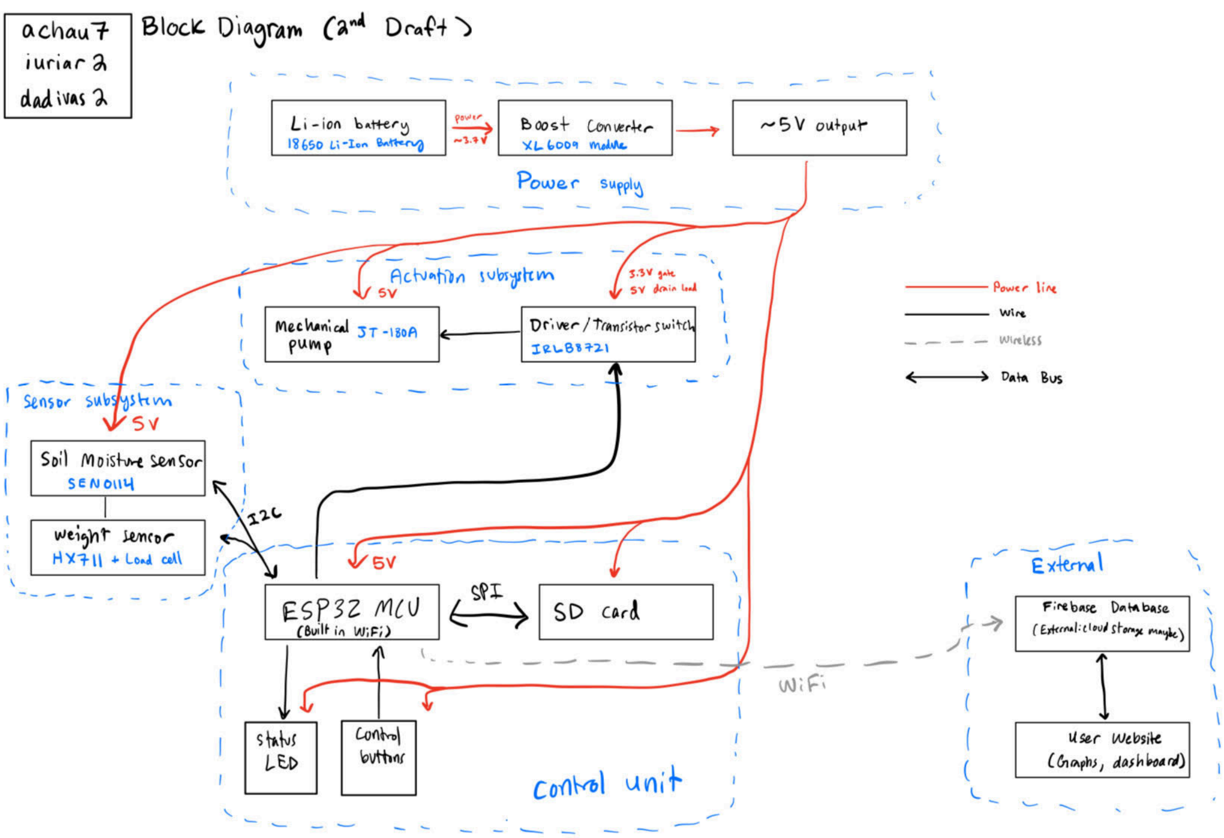

Fig 1. Block Diagram of System

Subsystem Overview

#1 Control Unit

The ESP32 will serve as the microcontroller for this project, chosen for its built-in Wi-Fi and Bluetooth functionality, which enhances efficiency and reduces design complexity. Its widespread use and extensive documentation provide a valuable resource for troubleshooting and development.

The ESP32 will handle soil moisture readings from sensors and determine when watering is needed. If the soil moisture level falls below a predefined threshold—set manually based on the plant type—the ESP32 will retrieve real-time weather data from a third-party API (e.g., AccuWeather) to check the probability of rain. If rainfall is likely, the system will delay watering, allowing the plant to naturally absorb moisture from rain. Otherwise, the ESP32 will activate the water pump to hydrate the plants.

Additionally, the ESP32 will calculate the amount of water saved by comparing moisture levels when watering is delayed due to rain versus when the water pump is used. Since the relationship between moisture levels and water usage is non-linear, developing an accurate calculation method will be a key challenge. The results will be transmitted to a website, where users can view a graph displaying water savings over time.

#2 Power Supply

Since our project is designed for outdoor plants, we will use one 3.7 V lithium-ion battery, as they offer high energy efficiency and rechargeability. However, standard lithium-ion batteries struggle in cold temperatures, which is why we will use cold-weather-resistant lithium-ion batteries to ensure reliable performance in various environmental conditions. The battery will be connected to a boost converter, which will regulate the voltage and provide a stable 5 V output to power the ESP32 microcontroller. We will have a buck converter which will provide 3.3 V to sensors.

#3 Watering Subsystem

To actively monitor soil moisture levels, we will use SEN0114 moisture sensors, which will be pinned into the soil and connected to the 3.3V pins of the ESP32. These sensors will continuously measure the moisture content and provide data to the microcontroller.

The moisture sensors will work in coordination with our water pump, a JT-180A, to ensure proper hydration of the plants. When the moisture levels drop below a set threshold and there is no forecasted rain, the ESP32 will trigger the water pump to irrigate the plants. This setup ensures efficient water use while preventing both overwatering and underwatering.

#4 External Subsystem

Our ESP32 will collect and upload data to a Firebase database, a cloud-based storage solution that will serve as our central repository for all collected data throughout the semester. This database will store key metrics, such as soil moisture levels, weather conditions, water usage, and calculated water savings.

To visualize this data, we will develop a custom website that connects to Firebase, allowing us to generate plots and display trends over time. This website will provide users with an intuitive interface to track their plant's hydration status and monitor water conservation efforts in real-time. Firebase will act as our reliable "storage box", ensuring all recorded data remains secure, accessible, and up-to-date.

#5 Rain Detection Subsytem

To enable our control board to detect rainfall, we will use a Load Cell + HX711 sensor module. This sensor will be placed beneath a container (e.g., a cup) that collects rainwater. By measuring the change in weight of the container, the system can determine how much rain has accumulated.

The ESP32 will use this weight data to calculate the amount of water saved by delaying irrigation. If significant rainfall is detected, the system will confirm the presence of rain and prevent unnecessary activation of the water pump. This subsystem ensures that the watering process is dynamically adjusted based on actual rainfall, improving water conservation and efficiency.

Risk Analysis

We have identified the rain detection subsystem using a load cell and HX711 ADC as the most challenging component to implement. The main reason for this is the sensitivity of the load cell to environmental factors such as wind, debris, and temperature fluctuations, which can introduce errors in weight measurements. Unlike digital sensors, load cells require precise calibration and may experience drift over time, leading to incorrect readings. The system must differentiate between actual rainfall and other factors that may affect the weight of the cup. **The load cell should detect weight changes as small as 1g to ensure rain is accurately measured.** We will also be manually testing (placing known weights) this sensor routinely.

Ethics & Safety

Our plant hydration and weather integration system design provides innovation for sustainability and more efficient resource consumption and therefore aligns with the IEEE Code of Ethics. The purpose of our system is to reduce water waste by utilizing real-time weather data to optimize water usage and reduce environmental impact. However, since we plan to use Firebase for data collection, which is a cloud–based storage system, it is important that we ensure ethical data management by setting secure user access and taking precaution to prevent unauthorized data manipulation. Furthermore, our project design does not involve testing of neither humans or animals alike, meaning this project does not require IRB approval.

A few safety concerns should be taken into consideration as our project is planned to provide solutions to outdoor plant life involving electrical components. Electrical safety will be followed as we are working with an ESP32 microcontroller, lithium batteries, and water based subsystems. We must ensure that components are properly insulated and waterproofed to prevent the possibility of shorting circuits and more electrical hazards. Mechanical safety will be followed as well. We must take into consideration the stability of the water pump system and the weight sensor system to prevent things such as tipping hazards. Another very important consideration for safety is that we follow safe battery handling since we will be handling the use of lithium batteries which are prone to overcharging, overheating, or leaking which may all pose a risk of fire.