ECE 445

Spring 2026

Soilmate

Ysabella Lucero (ylucero2), Emma Hoeger (ehoeger2), and
Sigrior Vauhkonen (sigrior2)
Team Number: #32
Date: 02/13/2026
TA: Zhuchen Shao
Instructor: Arne Fliflet

Table of Contents

Introduction

1. Problem

Many houseplant owners struggle to take proper care of their plants. In fact, “60 percent of millennials said that what worries them most is ensuring plants get the proper amount of sunlight. Other anxieties include: watering (56 percent of respondents), [and] keeping the plants alive (48 percent)” (Castillo). It can be difficult to keep track of when to water them and where to keep them, based on the species of plant and stage of life. Since all plants require water at different frequencies and amounts, it’s also easy to forget to water the plants on time and meet their different schedules. This causes many plants to die, and many owners become disheartened and avoid taking care of plants altogether.

2. Solution

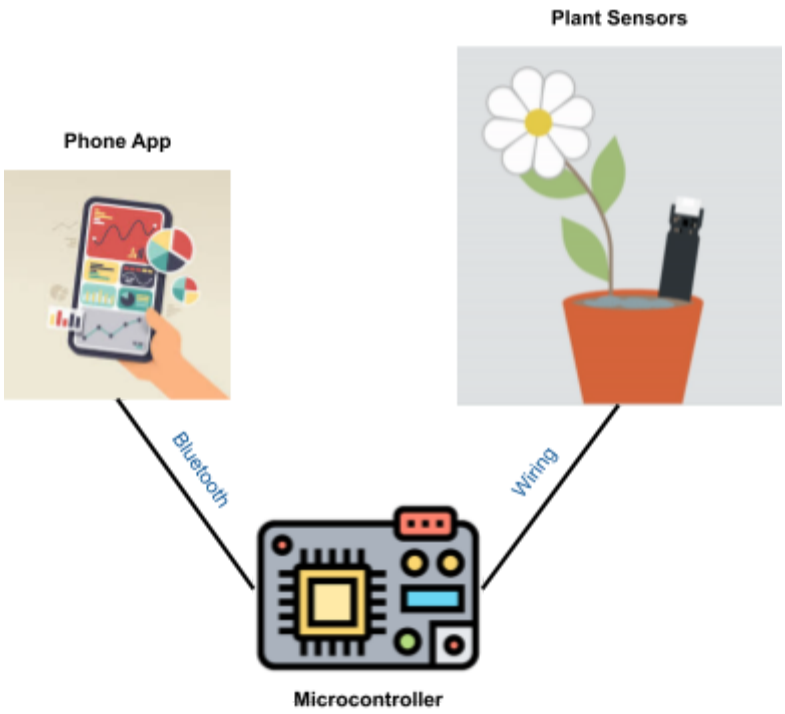
Our solution is to create a notification system to inform houseplant owners of when they should water their different plants. It will also monitor and notify the owner of the conditions of the plant based on various sensors. This will be done by creating an app that the owner can download on their phone, where they will be able to enter their type of plant. There have been many apps created to act as a reminder to water plants; however, the majority of them rely on a schedule rather than live data gathered from the plant. Also, those that do have live data from the plant do not track the weather. Our app will use the location of where a plant is originally from and use the weather patterns in that area to determine when it should be watered (ie. when it’s raining). In addition, there will be a soil moisture sensor, humidity sensor, light sensor, and temperature sensor. The soil moisture sensor acts as a

failsafe to prevent dangerously dry or damp conditions by alerting the owner to water the plant if the moisture is too low, and prevent overwatering of the plant if the moisture is too high. The humidity sensor will alert the owner when the humidity is dangerously too high or too low for the plant, which is especially useful for tropical plants in a non-tropical environment (many houseplants are of a tropical background). The temperature sensor will alert the owner when the room temperature is not in the optimal range for the specific plant. The light sensor will ensure that the plant is in a place with the correct light exposure, and will inform the owner when it is not.

With the integration of software and hardware subsystems, this effective plant notification system will make taking care of houseplants much more convenient for both beginner and experienced plant owners. Beginner plant owners will find it easier to learn about and keep track of the demands of their plants, preventing the most common mistakes that result in their death. Many experienced plant owners can have upwards of 20 plants, and this notification system would make it much simpler to keep track of when to water them all and meet their different needs.

3. Visual Aid

Our design will combine the plant sensors, the user's phone via Bluetooth, and the ESP32 Microcontroller to create this useful tool, which can be seen from our visual aid diagram below.



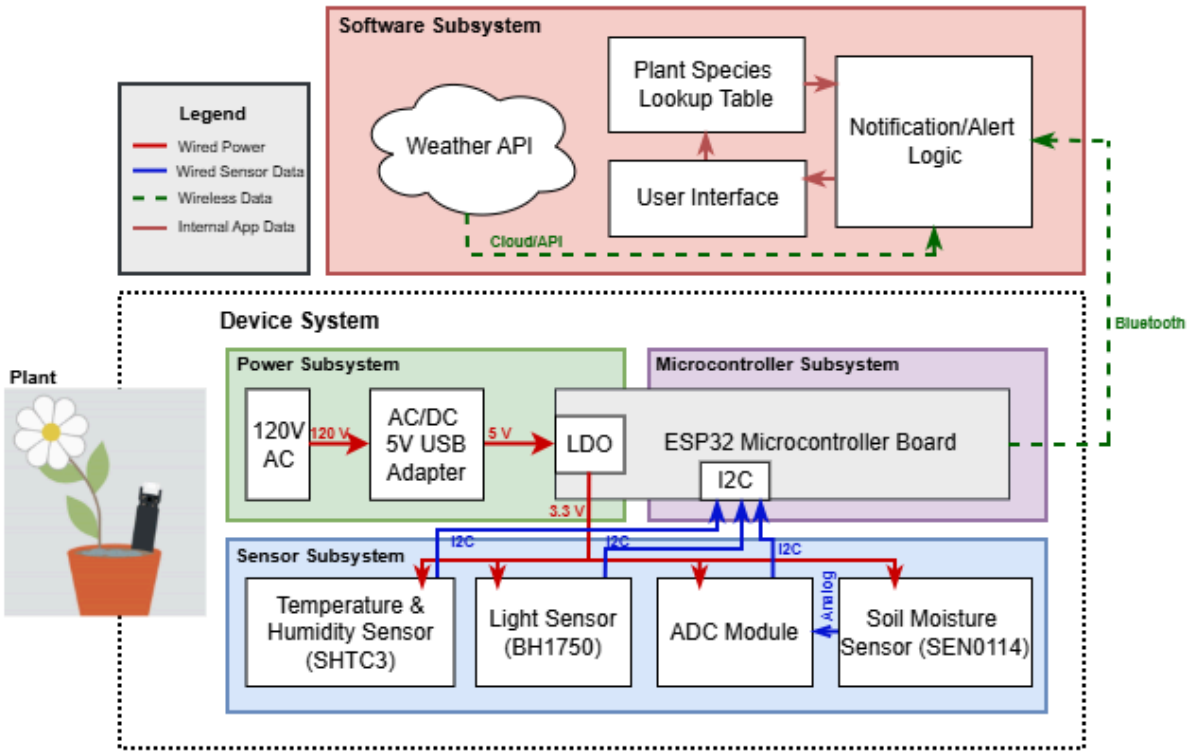
4. High-Level Requirements

1) The device must be able to gather accurate data from the sensors. This will be done by having a PCB with our microcontroller and the majority of the sensors, excluding the soil temperature sensor, since this has to be placed inside the soil. The PCB will be placed near the plant and will gather the environmental data of the plant. It will then communicate the data to the microcontroller, which will transmit it via Bluetooth to the user's phone app.
2) Another basic requirement is to gather weather data from the plant's original region. This will be done using a weather API that will update in the app once a day. If there is a chance of rain, then the app will know when the plant should be watered.
3) Lastly, the app that we create must send notifications to the user when to water the plant or to notify them when the plant's environment must change in combination

with the sensor and weather API (ie. the humidity is too high/low). This will be done using Flutter, a software that is used to create apps.

Design

1. Block Diagram



2. Subsystem Overview and Requirements

a. App Configuration Subsystem

The app (developed using Flutter/Android Studio) will allow the user to add a plant for monitoring- the user will select the plant species, size, light exposure, and the size of the pot. With this information, using a lookup table that holds information

for various house plant species, the app will store target ranges for soil moisture, temperature, humidity, and light, as well as an “origin location” (later used to check the weather). In the event that a plant species is unknown to the app (not in the lookup table), the user can manually add this information.

Once per day, the app will call a weather API (OpenWeatherMap API) using the origin location of a plant to check for rain in that region. With the data from the soil moisture sensor as a supplementary failsafe, a decision will be made on whether to water the plant or not. If the plant should be watered, a notification will be generated to inform the user. The data from the temperature, light, and humidity sensor will also generate notifications if the temperature and/or humidity is out of the recommended range, informing the user that the environment is too hot or too cold, or too moist or dry. It will give recommendations to either turn down/up the temperature, place the plant in a different facing window (north, east, south, west), mist with water if too dry, or open windows if too humid. This will make the app much more beginner plant owner friendly.

b. Sensors Subsystem

The sensor subsystem will use a resistive moisture sensor (SEN0114), a temperature and humidity sensor (SHTC3), and a light sensor (BH1750). All of these sensors, except the SEN0114, which requires an ADC module, will use an I2C interface that is compatible with our microcontroller (ESP32). The sensors will send their measurements to the microcontroller to be interpreted and relayed through the

app. Our power subsystem will supply the correct voltages to the rated amounts of the sensors.

c. Microcontroller Subsystem

We must be able to blend our app configuration with our live sensor subsystem to send an alert. We can do this by using the ESP32 microcontroller. It will provide WiFi and Bluetooth connectivity for our sensor devices to easily transfer the data to our app. It is cost-effective and has low power consumption, which will make it easy to integrate with our design. Furthermore, our group has experience with this specific microcontroller, so we are confident in its capabilities.

d. Power Subsystem

The power subsystem will deliver power to the sensors and microcontroller systems. The ESP32 requires 5V, while the temperature, humidity, moisture, and light sensors require 3.3V. The 3.3V will come from the LDO on the microcontroller, and we will use a 5V USB adaptor to convert the 120V AC from the bench to 5V.

3. Tolerance Analysis

There are some devices within our design that could possibly risk successful completion. The first is that we need to have our soil moisture sensor translate data to our A2D module and then to the microcontroller. Having this module in between the two could create slightly skewed data when it translates analog to I2C. We will be using an

ADS1115 for our converter, which provides 16-bit precision at 860 samples/second. The extra bits will allow us to be more precise with the voltage values it reads from the sensor. The module will not sample from the sensor too quickly since the sensor continuously sends out data. The module also allows us to program the adjusting gain if we notice the values varying too much.

Additionally, the soil moisture depth may vary with plants, so there can be inaccuracies in depth. Some plants need to stay consistently moist, while others may need to have the first two inches be dry before it has to be watered again. To mitigate this, we will have our app tell the owner how deep they must insert the sensor when they first get the device.

The last concern is that we will update our weather data only once a day. This can become a problem if there is no rain at that point or there is no chance of rain, yet it randomly rains later in the day. We wanted to update it only once to eliminate giving the owner multiple notifications. This is so that they do not receive conflicting notifications within a quick time span of each other. This is especially a concern in tropical environments where it intermittently rains throughout the day for a short period of time, so it could possibly overwhelm the owner.

Ethics and Safety

With this design, there are a few aspects that pose a risk. To begin with, the PCB we create will not be waterproof. Thus, when the owner goes to water the plant, some drops may spill and possibly ruin the board. To prevent the hazards of water meeting

electronics, we are working with the machine shop to create a container for our PCB to ensure that it is not exposed.

Our project is using weather forecasting data that is publicly available; we are not tracking the location of the plant owner and using the weather forecast information of their location. Since all of the data we are using is public and we are not collecting any personal data from the user, only information about their plant, we should not run into any ethical dilemmas relating to the security of the user's personal information.

Our project must meet the standards IEEE 802.11 / 802.15.1 for WiFi and Bluetooth communication. This is largely handled by our choice of microcontroller, being an ESP32 module, which meets the necessary standards. We will also make sure to use the standard software libraries for WiFi and Bluetooth when developing the app.

References

[1] filo, "Mobile device data statistics phone," iStock. Accessed: Feb. 04, 2026. [Online]. Available: https://www.istockphoto.com/illustrations/mobile-phone-market

[2] A. Chaturvedi, "How to NOT become a plant murder 101," Medium. Accessed: Feb. 04, 2026. [Online]. Available: https://avantikachat.medium.com/how-to-not-become-a-plant-murder-101-8d561ba9c3bb

[3] Flaticon. Accessed: Feb. 04, 2026. [Online]. Available: https://www.flaticon.com/free-icon/microcontroller_2752878

[4] I. D. Castillo, "This Is How Many Houseplants the Average Plant Parent Has Killed," *Apartment Therapy*, May 11, 2022. Accessed: Feb. 04, 2026. [Online]. Available: https://www.apartmenttherapy.com/dead-houseplants-average-37076429

[5] Ray, Brian. "Examining 5 IEEE Protocols - Zigbee, WIFI, Bluetooth, Ble, and WiMax." *IoT For All*, Dec. 02, 2024. Accessed: Feb. 13, 2026. [Online]. Available: www.iotforall.com/ieee-protocols-zigbee-wifi-bluetooth-ble-wimax.