Project Artifact Report

Section 1: Project overview

Background

This Smart Plant Monitoring system prototype is designed to assist indoor gardeners in maintaining optimal environmental conditions for their plants. It collects data from sensors such as a light sensor, temperature and humidity sensor, and soil moisture sensor to monitor the plant's environment. This data is automatically collected and transmitted via a cloud service like IFTTT. By enabling real-time alerts and logging, the system helps users respond promptly to their plants' needs, promoting healthier growth with minimal manual oversight.

Existing Work

Existing solutions for plant monitoring, such as smart pots, are bulky and not as easily portable. This limits their application for those who prefer flexible, mobile solutions for monitoring and managing their plants. Another example would be Parrot's Flower Power, which provides environmental readings but lacks integration with cloud services for real-time updates and alerts. This project enhances these existing solutions by enabling cloud connectivity through IFTTT services, Google Sheets and real-time monitoring.

Problem Statement

Indoor gardeners struggle to maintain the correct environmental conditions for their plants. While there are products available that monitor soil moisture, temperature, and light, there is a lack of systems that combine these variables, providing real-time feedback and automatic notifications when thresholds are met, especially without requiring manual intervention.

Requirements

This prototype aims to address two types of requirements: Functional and non-functional. Functional requirements address the sensors to monitor temperature, humidity, light intensity and soil moisture. Additionally, we want to track threshold data to alert the user when thresholds are exceeded, such as low moisture and high temperature. LCD interfacing addresses visual prompts to monitor the state of the sensor data. Additionally, this data will be automatically sent to Google Sheets and analysed into a graph. Non-functional elements cover the autonomous operation, allowing minimal involvement from the user.

Section 2: Design Principles

As someone who enjoys having plants around the house and occasionally overwaters them, I designed this product to help gardeners raise their awareness through plant care via accurate sensor data.

The design principles of this project are focused on creating a flexible, efficient, and user-friendly system for monitoring and maintaining plant health. Below are the key design principles that guided the development of the system:

1. **Portability and Modularity**
   - This system was designed with a portable, modular mindset, ensuring that components can be easily added or removed based on user needs. For instance, additional components can be sensors, water pumps and solar panels. Each of these elements contributes to the modularity of the final product and can be small enough to remain portable.
2. **Low power consumption**
   - One of the primary goals was to create a low-power, energy-efficient system, especially considering that the system is designed for portable, outdoor use with solar power implementation.
3. **Real-time Monitoring**
   - This system is designed to incorporate real-time monitoring of environmental parameters like temperature, soil moisture and light intensity. Data is transmitted and analysed continuously, providing users with up-to-date insights into the plant's environment over time.
4. **Cloud integration**
   - Another key aspect is the integration of cloud-based services into the system. This enables remote access to sensor data and data visualisation to see daily trends of their plant health data.
5. **Responsiveness and Reliability**
   - Responsiveness is a key design that allows a mix of continuous sensor value checking and data uploading through a 60-second timer. This ensures the user always has access to the most current readings. Additionally, the alternative low-power mode records a small set of data at set hourly intervals per day.

Section 3: Prototype Architecture

The prototype architecture for this project can be broken down into two sections: Hardware and Network Architecture.

Hardware

1. Microcontroller – Arduino Nano 33 IoT
   - The microcontroller coordinates all the components, such as sensor inputs, sensor data processing, LCD information, LED status and built-in Wi-Fi connectivity to manage communication through an IFTTT service.
   - This microcontroller was chosen based on the large number of digital and analog pins, allowing for adequate processing power for the sensors and IFTTT integration, and most importantly, because it includes built-in Wi-Fi.

2. Sensors
   - DHT11 has the role of measuring the temperature and humidity of the environment
   - GY-30's role is to measure ambient light intensity in the environment, and this sensor communicates through I2C
   - Soil Moisture Sensor reads the moisture level and is connected to analog pins on the Arduino, allowing for variable voltage from the soil to be recorded through the sensor.

3. LED indicators
   - These LEDs have the role of giving the user real-time feedback based on set thresholds.

4. LCD and potentiometer
   - The LCD is closely linked to the rest of the code by enabling visual feedback
   - All sensor readings that are produced and stored within Google Sheets via IFTTT can be displayed on the spot through the LCD
   - The Potentiometer is only used for text dimming and serves no other purpose.

5. Battery Powered
   - Currently, the system is run by a battery source of 6 AA batteries
   - This can further be evolved by utilising solar power.
   - Solar power integration proved challenging due to the limited number of available pins on the microcontroller.
   - Solar power will also require additional components such as: Charge controllers for power distribution, and a step-down module for power voltage conversion

Network Architecture

1. Wi-Fi connection
   - The role of the Wi-Fi connection is essential for the sensor data to be sent through the IFTTT service
   - It's not critical for the project to run, but it is critical for data recording

2. IFTTT Integration
   - IFTTT is used to automate actions based on sensor readings.
   - It records all values produced by the sensors (temperature, soil moisture and light intensity)
   - The IFTTT will trigger an event that sends the log data to Google Sheets.

3. Cloud storage and alerts
   - Cloud storage through Google Sheets stores and visualises sensor data over time.
   - Alerts are not currently implemented, but will be to enable the user to monitor any issues within the product
   - For example, an alert can happen when a threshold is not met over time.

Section 4: Link to Prototype Code

https://github.com/TimTrev/SIT210_Task11.1_Prototype

Section 5: Testing Approach

The testing approach for this project was designed to ensure the system performs as intended, focusing on sensor accuracy, data transmission reliability, user interface functionality, and system responsiveness. Below are the key components of the testing approach:

1. Unit testing was conducted to verify that each individual sensor and system component operates as expected. This approach helped isolate and identify issues in specific parts before they affect overall performance.
2. Wi-Fi connectivity testing helps ensure the Wi-Fi connection is stable and can handle frequent data transmissions. Tested the system's ability to reconnect if the Wi-Fi connection drops.
3. LED feedback testing verifies whether LED indicators correctly turn on and off based on sensor readings and thresholds
4. Integration Testing was performed to verify that all components work together seamlessly, and that the system meets the desired functionality.

5. Sensor Data integration includes verifying that data from the temperature, humidity, light intensity, and soil moisture sensors are correctly aggregated and sent to the cloud (Google Sheets) via IFTTT. Additionally, these tests needed to be tested in real-time environments.
6. Failing handling and recovery is a system test that simulates sensor failures by disconnecting sensors or providing extreme conditions (e.g., very low or high values). This ensured the system handled failures gracefully by displaying appropriate error messages on the LCD and sending error notifications to IFTTT if necessary.
7. Testing of Wi-Fi disconnection and re-establishment by temporarily turning off the Wi-Fi network and verifying the system's ability to reconnect and resume sending data after reconnection was incorporated.
8. Lastly, performance testing measured the system's response time when sensors detect changes (e.g., a drop in soil moisture or a temperature change). Ensuring that the LEDs and IFTTT notifications are triggered within an acceptable time frame (less than a few seconds).

Section 6: User Manual

The User Manual can be found and downloaded through the GitHub Repository

Section 7: Link to Demonstration video

https://youtu.be/dZKQn37gzRU

Section 8: Conclusion

Developing the Smart Plant Monitoring system prototype has been a rewarding and insightful experience. The project lays a strong foundation for future development in smart gardening systems through monitoring and maintaining the ideal environmental conditions for their plants. By integrating multiple sensors, including light, temperature, humidity, and soil moisture sensors, with real-time cloud data logging via IFTTT and Google Sheets, the system provides users with an accessible and efficient way to ensure their plants thrive.

Throughout the development process, key challenges were encountered, especially in integrating cloud-based services like IFTTT and Google Sheets for real-time monitoring. The issue of limited pin availability for sensors and solar power integration also posed obstacles that were creatively addressed by focusing on battery power and modular sensor integration.

The testing phase confirmed the system's reliability and responsiveness in real-world conditions, ensuring that data is transmitted seamlessly and sensor thresholds are respected. LEDs for real-time feedback, combined with notifications via IFTTT, provide an intuitive user interface, making the system user-friendly.

If given a second chance, a more streamlined solution for handling solar power integration would be considered, along with further optimisations in the code to enhance system performance and lower power consumption. The system could also benefit from additional features such as more robust alerting mechanisms, integration with other smart devices, and further customisation options for users.

In conclusion, this project serves as a functional and scalable prototype for indoor plant monitoring, providing a foundation for future enhancements in both hardware and software. It successfully demonstrates the potential of combining embedded systems, IoT, and cloud services to improve the care and maintenance of plants in a modern, connected world.