SWE30001 – IoT Programming

Assignment 4 (Group Assignment)

Authors: Christian Cheng (103075742), Tran Duc Anh Dang (103995439), Yiannis Kyritsis (103980370)

Date: 7/02/2024

Word Count: 2179

Table of Contents

1. Introduction

1.1. Topic

Smart Plant Monitoring System for Enhanced Plant Health Care in Home Gardening

1.2. Background

With the increasing trend of home gardening and the growing desire for self-sustainability, there is a need for innovative solutions to assist individuals in maintaining their plants effectively. Many enthusiasts embark on cultivating their own plants but face challenges in consistently monitoring vital parameters such as water levels, temperature, humidity, and growth progress. The Smart Plant Monitoring System addresses these challenges by providing an automated and efficient way to ensure optimal plant health.

1.3. Proposed System

The Smart Plant Monitoring System offers a holistic approach to plant care, addressing both internal and external factors crucial for plant growth. By integrating advanced sensors and automation technology, this system caters to the growing interest in home gardening while addressing the time constraints faced by individuals. The result is a more efficient and enjoyable gardening experience, fostering healthier and more robust plant growth. Automating the monitoring process, individuals can ensure their plants receive consistent attention even in the absence of manual oversight. The system can be connected to a mobile application, allowing users to receive real-time updates on their plant's status, make necessary adjustments remotely, and receive alerts when specific thresholds are met.

1.3.1. Internal Features Monitoring

1. **Water Level Sensor**

   The Water Level Sensor is employed to monitor the water level within the plant pot. This ensures that plants receive an adequate and consistent water supply, preventing both overwatering and underwatering, which are common issues affecting plant health.

1.3.2. External Features Monitoring

1. **LM35 Temperature, DHT11 Sensor**

   The LM35 sensor is dedicated to monitoring the ambient temperature. Temperature variations can significantly impact plant growth and overall health. The system adjusts the environment based on temperature data to create an optimal growth condition for the specific plant species.

2. **Motion Sensor**

   The motion sensor plays a key role in determining the plant's height and growth progress. Once the plant reaches the desired height for harvesting, the system can trigger alerts or automated actions, signaling that the plant is ready for harvest. This feature ensures timely harvesting and prevents overgrowth, enhancing the plant's yield and quality.

1.3.3. Other Enhancements

1. **Red LED Actuator**

   The Red LED Actuator not only provides a visual cue by illuminating when the plant has reached the desired height set by the user but also enhances its functionality by blinking intermittently. This blinking pattern serves as a dynamic indicator, signaling that the plant has achieved its set height and is ready for harvesting or simply allowing users to track its growth progress over time.

2. **Blue LED Actuator**

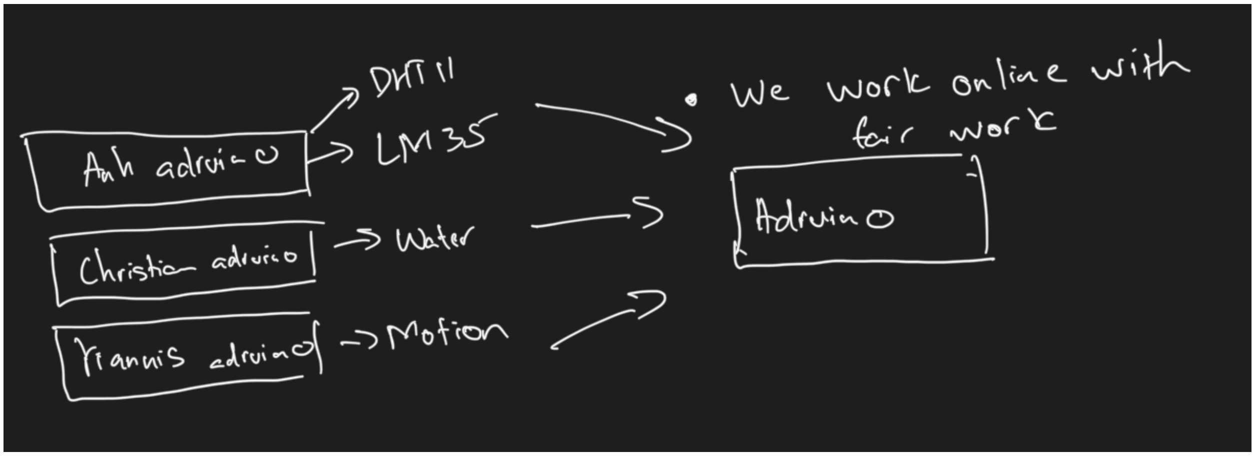
   The Blue LED Actuator provides a visual cue for users when the plant requires watering. When the water level is getting low, the Blue LED lights up to alert the user that the plant needs water. This serves as a proactive reminder for users to attend to their plants' hydration needs. Once the plant is watered and the water level rises to an adequate level, the Blue LED automatically turns off, indicating that the plant has been sufficiently watered. This feature helps users maintain optimal hydration levels for their plants, promoting healthy growth and minimizing the risk of over or under-watering.

3. **Yellow LED Actuator**

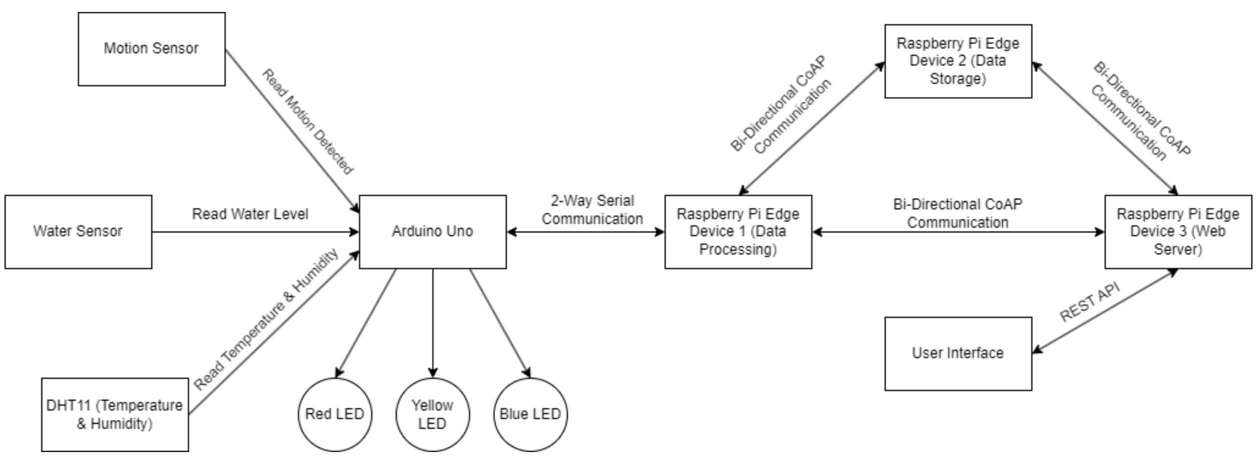
   The Yellow LED Actuator provides an additional visual cue for users by illuminating when temperatures are too hot, signaling the need to relocate the plant to a cooler environment for more optimal growing conditions. This proactive feature ensures that users can promptly take action to protect their plants from excessive heat stress, thereby safeguarding their health and promoting successful growth

2. Conceptual Design of Project

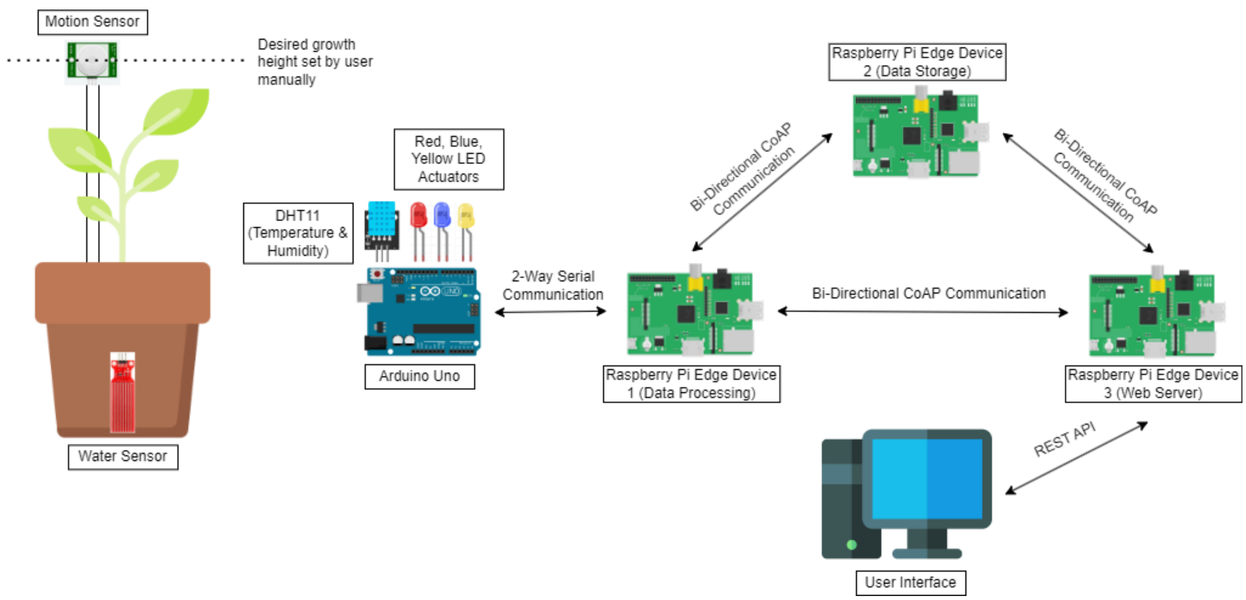
2.1 System Block Diagram



The system is scalable with multiple adruino's devices with flexible database structure



2.2 Real-Life System Configuration



3. Tasks Breakdown

Name (Student Id)	Tasks
Christian Cheng (103075742)	• Project Planning • Prototyping • Circuit Simulation • Diagram Designer • Report Writing • Quality Assurance
Tran Duc Anh Dang (103995439)	• Project Planning • Prototyping • System Engineer • Video Demonstration • Report Writing • Quality Assurance
Yiannis Kyritsis (103980370)	• Project Planning • Prototyping • Circuit Simulation • Video Demonstration • Report Writing • Quality Assurance

Note: We attempted to split up the roles as evenly as possible despite different schedules, ensuring each team member's unique strengths contribute to the project's success.

4. Implementation

4.1. Sensing System

The sensing system of our Smart Plant Monitoring System incorporates various sensors to monitor key parameters essential for plant health. These include the DHT11 sensor for temperature and humidity, a motion sensor to detect plant height, a water sensor to measure water levels, and LEDs for visual cues to the user.

4.2. Edge Servers

We utilize three edge servers to manage data processing, storage, and user interface:

1. Serial Data Reader Server

The Serial Data Reader Server plays a critical role in the Smart Plant Monitoring System by acting as the interface between the hardware components and the rest of the system. This edge server is responsible for reading serial data transmitted from the Arduino microcontroller, extracting sensor readings and status updates related to temperature, humidity, motion detection, and water levels. Once the data is received, the server processes it and forwards it to the appropriate destinations within the system for further analysis and storage.

2. Database Server

The Database Server serves as the centralized repository for storing all sensor data, user settings, and system configurations generated by the Smart Plant Monitoring System. This edge server ensures data integrity, reliability, and accessibility, allowing users to retrieve historical sensor data, monitor trends, and make informed decisions about plant care. By securely storing sensor data and user settings, the Database Server facilitates seamless communication between different system components and enables efficient data management and analysis.

3. Web Server

The Web Server, hosted on the third edge server, provides users with a user-friendly interface to interact with the Smart Plant Monitoring System. Built on the Python Flask framework, this server hosts the web application that allows users to access real-time sensor data, adjust system settings, and visualize plant health metrics through intuitive graphs and charts. The web interface enables users to monitor their plants remotely, receive alerts, and take proactive measures to ensure optimal plant care. Additionally, the Web Server facilitates communication between users and the system, allowing for seamless integration of user feedback and system updates.

4.3. Communication Protocols

We employ the Constrained Application Protocol (CoAP) as a lightweight IoT communication protocol. CoAP is a specialized web transfer protocol specifically designed for constrained devices and low-power networks, making it an ideal choice for IoT applications. Unlike traditional HTTP, CoAP operates over UDP, which minimizes overhead and reduces resource consumption, making it suitable for resource-constrained environments.

CoAP follows a client-server communication model, where clients initiate requests to servers to retrieve or modify resources. It supports various methods such as GET, POST, PUT, and DELETE, allowing clients to perform operations on resources. One of the key features of CoAP is its support for lightweight messaging formats such as CBOR (Concise Binary Object Representation) and efficient URI (Uniform Resource Identifier) mapping. This enables efficient encoding and transmission of data, reducing bandwidth usage and energy consumption.

In the context of the Smart Plant Monitoring System, CoAP facilitates communication between edge devices, allowing them to exchange sensor data, status updates, and user settings efficiently. For example, the edge server reading serial data from the Arduino communicates with the database server via CoAP to store sensor data securely. Similarly, user settings updates are transmitted back to the serial data reader server via CoAP, ensuring seamless communication within the system.

4.4. API or Website Details

Our API and website provide users with a comprehensive User Sensor Dashboard. Through this interface, users can access real-time updates from the sensors and actuators connected to the Arduino. Additionally, users can modify settings such as temperature thresholds for the yellow LED and water thresholds for the blue LED. The website offers visual representations such as graphs to track changes in sensor readings over time, empowering users with valuable insights into their plant's health and environmental conditions.

4.5. Cloud Computing

While our implementation primarily focuses on edge computing, we also integrate cloud computing for scalability and data analysis. Sensor data stored in the database server can be synchronized with cloud-based storage solutions for backup and remote access. Additionally, cloud-based analytics tools can process large datasets to derive actionable insights, further enhancing the functionality and effectiveness of our Smart Plant Monitoring System.

5. Smart Plant Monitoring System User Manual

5.1. Introduction

Welcome to the Smart Plant Monitoring System! This user manual provides guidance on operating and utilizing the features of our IoT system to ensure optimal care for your plants. With our system, you can monitor key parameters such as temperature, humidity, water levels, and plant growth status in real-time, empowering you to cultivate healthy and thriving plants.

5.2. System Setup

1. Ensure all hardware components are securely connected and powered on.
2. Access the User Sensor Dashboard through your web browser using the provided URL.
3. Familiarize yourself with the dashboard layout and available features.

5.3. Main Operations:

1. Monitoring Sensor Data

a. View real-time updates from sensors, including temperature, humidity, and water levels, on the dashboard.

b. Check the status of plant growth indicators, such as motion sensor readings and LED alerts.

2. Adjusting Settings

a. Modify temperature thresholds for the yellow LED indicator to suit your plant's needs.

b. Adjust water thresholds for the blue LED indicator to customize watering alerts.

c. Update other system configurations as needed through the settings menu.

3. Harvesting Plants

a. Monitor the red LED indicator, which lights up when the plant has reached the desired height for harvesting.

b. Harvest the plant when the red LED is illuminated, indicating optimal growth conditions.

4. **Watering Plants**
    a. Check the blue LED indicator, which illuminates when water levels are low, signaling the need for watering.
    b. Water the plant and ensure the blue LED turns off, indicating that the plant has been adequately hydrated.
5. **Responding to Temperature Alerts**
    a. Pay attention to the yellow LED indicator, which lights up when temperatures are too hot for optimal plant growth.
    b. Take action to relocate the plant to a cooler environment if necessary, ensuring optimal growing conditions.

5.4. Maintenance

1. Regularly check hardware components for any signs of damage or malfunction.
2. Keep sensors clean and free from debris to ensure accurate readings.
3. Monitor the system for any unusual behavior and troubleshoot as needed.

5.5. Conclusion

With the Smart Plant Monitoring System, you have the tools and insights to care for your plants with precision and ease. By leveraging real-time sensor data and intuitive indicators, you can create an ideal environment for your plants to thrive. Happy gardening!

6. Limitations of Smart Plant Monitoring System

While the Smart Plant Monitoring System offers an innovative solution for optimizing plant care and management, it is important to acknowledge its limitations. These limitations may impact the system's functionality, reliability, and overall user experience. Some limitations include:

1. Dependence on Internet Connectivity

The system relies on internet connectivity for accessing the User Sensor Dashboard and receiving real-time updates. In cases of internet outages or disruptions, users may experience delays in monitoring their plants.

2. Sensor Accuracy

While the sensors used in the system provide valuable data, they may have inherent limitations in accuracy and precision. Variations in sensor readings or occasional inaccuracies may occur, affecting the reliability of the information provided.

3. Limited Sensor Range

The sensing capabilities of the system are confined to the immediate environment surrounding the plant. Sensors may not detect conditions outside their range, potentially overlooking external factors that could impact plant health.

4. Battery Life

If the system relies on battery power for operation, battery life may be a limiting factor. Rechargeable batteries may require frequent recharging, while non-rechargeable batteries may need replacement, leading to downtime in monitoring.

5. Sensitivity to Environmental Factors

External environmental factors such as electromagnetic interference or extreme weather conditions may affect the functionality of the system components, leading to potential disruptions or inaccuracies in sensor readings.

6. Maintenance Requirements

Regular maintenance of hardware components and sensors is necessary to ensure optimal performance. Failure to conduct routine maintenance tasks may result in degradation of system functionality over time.

7. User Skill and Knowledge

Users may require a certain level of technical skill and knowledge to set up and operate the Smart Plant Monitoring System effectively. Inexperienced users may encounter challenges in configuring settings or troubleshooting issues.

8. Data Privacy and Security

The system may collect sensitive data about users' plant care habits and environmental conditions. Ensuring data privacy and implementing robust security measures is essential to protect user information from unauthorized access or breaches.

9. Cost

The initial cost of purchasing hardware components and setting up the system may be a barrier for some users. Additionally, ongoing costs related to maintenance, replacement parts, and internet connectivity should be considered.

10. Scalability

The system's scalability may be limited, particularly in larger-scale applications such as commercial agriculture or industrial settings. Scaling up the system to accommodate a larger number of plants or expanding its functionality may pose logistical and technical challenges.

7. Resources

1. Pallets Projects. (2024). Flask Documentation. Pallets Projects. https://flask.palletsprojects.com/en/3.0.x/

2. Giacomo Tanganelli (Tanganelli). (2019). CoAP Installation GitHub Repository. Github (Tanganelli). https://github.com/Tanganelli/CoAPthon

3. Ian Craggs. (2022). MQTT vs CoAP for IoT. HiveMQ. https://www.hivemq.com/article/mqtt-vs-coap-for-iot/

4. Mukhadin Beschokov. (2023). What Is CoAP Protocol? Meaning & Architecture. Wallarm. https://www.wallarm.com/what/coap-protocol-definition

5. Alun Williams. (2016). How to build a plant monitor with Arduino. ElectronicsWeekly.com. https://www.electronicsweekly.com/blogs/gadget-master/arduino/build-plant-monitor-arduino-2016-09/

6. Stregoi. (2016). Arduino Plant Monitor. Instructables. https://www.instructables.com/Arduino-Plant-Monitor/

7. Components101. (2021). DHT11–Temperature and Humidity Sensor. Components101. https://components101.com/sensors/dht11-temperature-sensor

8. Components101. (2021). HC-SR501 PIR Sensor. Components101. https://components101.com/sensors/hc-sr501-pir-sensor

9. Last Minute Engineers. (2018). How Water Level Sensor Works and Interface it with Arduino. Last Minute Engineers. https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/

10. Arduino Get Started. (2018). Arduino - Water Sensor. Arduino Get Started. https://arduinogetstarted.com/tutorials/arduino-water-sensor

11. MariaDB. (2024). MariaDB Server Documentation. Maria DB. https://mariadb.com/kb/en/documentation/

12. Abdelhadi Dyouri. (2022). How To Make a Web Application Using Flask in Python 3. Digital Ocean. https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3

13. Raspberry Pi Foundation. (2017). Build a Python Web Server with Flask. Python Pi Foundation. https://projects.raspberrypi.org/en/projects/python-web-server-with-flask

14. London Grow. (2020). Understanding optimum Temperature and Humidity for Plants. London Grow.

https://www.londongrow.com/blogs/grow-tips/understanding-optimum-temperature-and-humidity-for-plants

15. Living Water Smart Darwin Region. (2013) How much water does your garden need? Living Water Smart Darwin Region. https://www.livingwatersmart.com.au/gardenwateringneeds

8. Appendix

8.1. Hardware Component

1. Raspberry Pi (Edge Devices): Used as the edge computing device, the Raspberry Pi collects sensor data directly from the connected sensors. It processes this data at the edge, performing initial analyses and filtering before sending it to the communication server. This setup reduces latency and network traffic, enabling real-time responsiveness.

2. Raspberry Pi (Communication Server with CoAP): Raspberry Pi acts as the communication server, utilizing the CoAP (Constrained Application Protocol) for efficient IoT communications. This device serves as the intermediary, receiving processed data from the edge device (the first Raspberry Pi) and forwarding it to the database hosted on a PC. It handles requests from the web server and interacts with the edge device to retrieve or update sensor data.

3. PC (Database Server): A dedicated PC hosts the database server, storing all sensor data, user settings, and system configurations. This server ensures data integrity, reliability, and accessibility, allowing for complex queries and historical data analysis. It communicates with the Raspberry Pi communication server to receive and store data from the edge device.

4. PC (Web Server): A dedicated PC hosts the web server, providing a user-friendly interface for accessing the Smart Plant Monitoring System. This server runs the web application, allowing users to view real-time sensor data, adjust system settings, and visualize plant health metrics. It communicates with the Raspberry Pi communication server to send commands to the edge device and to retrieve data for the user interface.

8.2. Software and Libraries

Software components and libraries used in the system's backend and frontend development, including CoAPython3, Flask, React, Material-UI, Axios, Chart.js, React-Chartjs-2, Date-Fns, Chartjs-Adapter-Date-Fns, Chartjs-Adapter-Moment, and @mui/styles.