Smart Irrigation System Documentation

Made by: Alexia Oaida, Doris Haiduc

**Project Idea**

The goal of the IIOTCA laboratory project is to construct a smart irrigation system using the power of the Internet of Things (IoT). This system's purpose is to measure soil moisture levels continuously and accurately, ensuring optimal watering for plant health and survival. The design comprises a Raspberry Pi device and a soil moisture sensor, creating an automated solution that facilitates efficient irrigation strategies based on real-time data.

To enhance the user interaction with the system, a mobile application is proposed. This application will allow users to monitor soil moisture levels remotely, providing real-time data from the soil moisture sensor.

**Main Features**

• *Real-Time Soil Moisture Monitoring*: The system continuously measures and reports soil moisture levels, providing actionable data to guide irrigation.

• *Automated Irrigation Control*: Based on the moisture readings, the system can recommend when to water the plants.

• *LCD Display*: The system incorporates an LCD display connected to the Raspberry Pi, displaying the current soil moisture level.

• *Mobile Application*: A user-friendly mobile application will allow users to access real-time soil moisture data and receive recommendations for watering.

**Architecture of the System and Subsystems**

The architecture of the smart irrigation system is designed with a view to providing real-time soil moisture updates and remote monitoring capabilities. The system is built on top of several key technologies and platforms, including Raspberry Pi, AWS IoT Core, Firebase Realtime Database, and Android mobile technology.

Raspberry Pi and Soil Moisture Sensor

The Raspberry Pi, loaded with a soil moisture sensor, serves as the primary data collection point. It is responsible for gathering raw data about soil moisture levels, which is then processed and forwarded to other subsystems for analysis and action. The Python script runs on the Raspberry Pi to read data from the sensor.

AWS IoT Core

AWS IoT Core is used as the middleware that ensures real-time communication between the Raspberry Pi and other components of the system. It provides a robust and secure platform for devices to interact with cloud applications and other devices. In this case, the Raspberry Pi is configured to publish the soil moisture data to an AWS IoT Core MQTT topic.

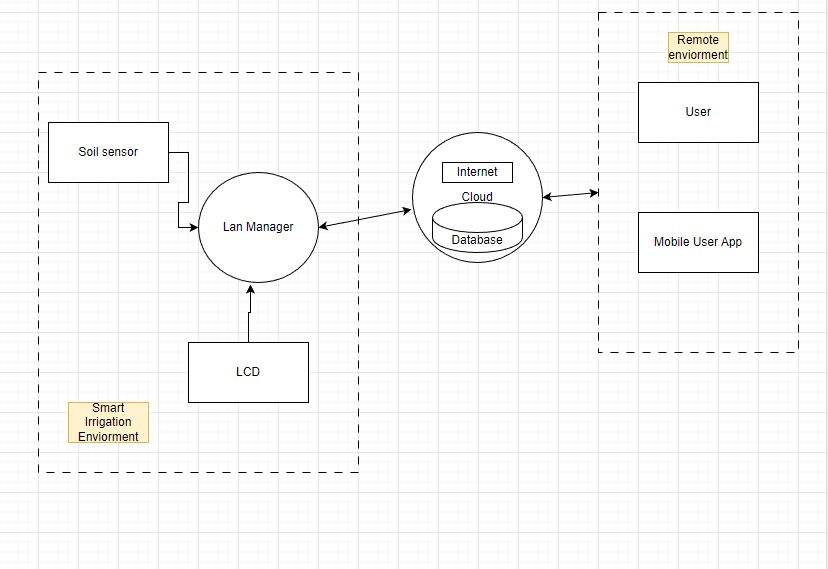
Firebase Realtime Database

Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. This ensures that all components of the system have access to the most current data. In this project, soil moisture data is pushed from the Raspberry Pi to Firebase Realtime Database, where it is immediately available to the Android application.

Android Mobile Application

The Android mobile application allows users to monitor the soil moisture level remotely. It retrieves the latest soil moisture data from the Firebase Realtime Database and updates the user interface.

This architecture enables the smart irrigation system to provide real-time updates and facilitates remote monitoring of soil moisture levels. Furthermore, it provides a foundation for future expansion, such as adding additional sensors or developing more sophisticated data analysis capabilities.

Please refer to the attached diagram for the visual representation of the system architecture. 

**Short Descriptions of Technologies Used**

The project uses several technologies, including:

• Raspberry Pi: A single-board computer, serving as the central processing unit for the system.

• Java Programming Language: Used to program the logic and functioning of the system.

• AWS IoT services: Amazon Web Services' cloud platform used for data storage and IoT device management.

• Android Studio: The official IDE for Google's Android operating system, used to develop the mobile application.

• Python: The Python script controls the soil moisture sensor and the communication with the Firebase Realtime Database.

• Firebase Realtime Database: A cloud-hosted NoSQL database by Google, where every piece of data is stored as JSON and synchronized in real-time to every connected client.

**How the Project actually looks**

A close-up of a circuit board

Description automatically generated with low confidence

**Code Examples and Descriptions**

This project uses a mixture of Python for the Raspberry Pi, Firebase Realtime Database, and Java for the Android application.

Python code is responsible for initializing the soil moisture sensor, fetching sensor readings, and updating the database with the current moisture level.

A screenshot of a computer code

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated with medium confidence

The Java code for the Android application fetches the latest soil moisture data from the Firebase Realtime Database and updates the user interface.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated with medium confidence

We also have some Python code that leverages the AWS SDK for Python (Boto3) to publish soil humidity data to a specified AWS IoT Core topic. Let's look at how it works:

A screenshot of a computer program

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

**Future Work**

The current version of the smart irrigation system is a viable solution for small scale or home-based gardening. For future iterations, the system could be further enhanced to cater to larger agricultural lands, including features like support for multiple soil sensors, predictive irrigation based on weather forecasts, and analytics dashboards for long-term irrigation strategy planning.

**References**

• Raspberry Pi Foundation. Raspberry Pi 4 Model B. Raspberry Pi Foundation, 2019.

• Oracle Corporation. Java Development Kit (JDK) 11. Oracle Corporation, 2018.

• Amazon Web Services, Inc. AWS IoT Core. Amazon Web Services, Inc, 2019.

• Google LLC. Android Studio. Google LLC, 2020.

• Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

• Google LLC. Firebase Realtime Database. Google LLC, 2020.

*Note: All citations are in accordance with the IEEE referencing guidelines.*

Github repo: https://github.com/SuperxJoker/IotcaProj