

IoT Plant Watering System

This IoT project involves a smart plant watering system that monitors the last watering time, calculates the interval since the last watering, and decides whether to activate a water pump. Additionally, it tracks the water level in the tank, sends critical water level notifications, and records the timestamp of the ESP32's deep sleep mode. The system also includes a Node-RED dashboard for real-time monitoring.

Features

1. Watering Schedule Management:

- Fetch the last watering time from a MariaDB/MySQL database.
- Calculate the time difference between the current time and the last watering time.
- Determine whether the water pump should be activated based on the calculated interval.

2. Water Level Monitoring:

- Measure the water level in the tank when the pump is activated.
- Store the measured water level in the MariaDB table.
- Send critical water level notifications via Telegram or another messaging app.

3. Sleep Mode Tracking:

- Record the timestamp of the ESP32's deep sleep mode in the MariaDB table to save energy and optimize performance.

4. Real-time Dashboard:

- Develop a Node-RED dashboard to display the tank's water level and the last watering date for easy monitoring.

Components Used

- **ESP32:** Microcontroller for managing the watering schedule, water level measurement, and deep sleep mode.
- **MariaDB/MySQL:** Database for storing and retrieving watering data and timestamps.
- **Node-RED:** Dashboard for real-time monitoring of the system's status.
- **Telegram:** Messaging app for sending critical water level notifications.

Installation and Setup

1. Database Setup:

- Install and configure MariaDB/MySQL.
- Create the necessary tables for storing watering data, water levels, and deep sleep timestamps.

2. ESP32 Configuration:

- Program the ESP32 to fetch data from the database, measure water levels, and record timestamps.
 - Implement logic to decide when to start the water pump based on the watering schedule.
 - Set up deep sleep mode to optimize power consumption.
3. **Node-RED Dashboard:**
- Install Node-RED on your server or local machine.
 - Create a dashboard to display the tank's water level and the last watering date.
 - Connect Node-RED to the MariaDB/MySQL database for real-time data visualization.
4. **Notification Setup:**
- Configure Telegram
 - Implement the notification logic in the ESP32 program.

Usage

1. **Monitor Plant Watering:**

- Use the Node-RED dashboard to keep track of the last watering date and the current water level in the tank.
- Ensure the system is connected to the database for real-time updates.

2. **Receive Alerts:**

- Receive notifications on Telegram

3. **Optimize Water Usage:**

- Adjust the watering schedule based on the data collected to ensure optimal water usage for your plants.

Conclusion

This project aims to provide a smart, efficient, and user-friendly solution for plant watering management. With real-time monitoring, automated decision-making, and critical notifications, you can ensure your plants are well-watered without wasting resources.