Final Project Report

Plant Monitoring and Watering System

By Mouhannad Chehab, Mohammad Shehab, Hussein Mourtada

# Overview

The Plant Monitoring and Watering System is designed to simplify plant care by leveraging sensors, automation, and Bluetooth Low Energy (BLE) communication. The primary purpose of this project is to monitor soil moisture, temperature, and humidity levels in real-time, ensuring optimal plant health. When the soil moisture level drops below a critical threshold, the system automatically waters the plant. Additionally, the system provides a user-friendly interface through an OLED display and BLE communication for remote monitoring and alerts.

# Features

* Real-time monitoring of soil moisture, temperature, and humidity.
* Automatic watering when the soil is too dry.
* Bluetooth Low Energy (BLE) integration for remote monitoring via a smartphone or PC.
* OLED display for local data visualization.
* Sound alerts for critical conditions such as very dry soil.

# Hardware Setup

## Components

1. **Arduino Nano 33 BLE**: The central microcontroller used for processing sensor data and handling BLE communication.
2. **DHT11 Sensor**: Measures ambient temperature and humidity.
3. **Soil Moisture Sensor**: Monitors soil moisture levels.
4. **Relay Module**: Controls a water pump for automatic watering.
5. **OLED Display (SSD1306)**: Displays real-time data locally.
6. **Water Pump**: Delivers water to the plant as needed.
7. **Power Supply**: 5V power supply for the Arduino and other components.

## Wiring Diagram

1. **DHT11 Sensor**:
   * VCC to 3.3V
   * GND to GND
   * Data to D4
2. **Soil Moisture Sensor**:
   * VCC to 3.3V
   * GND to GND
   * Analog Output to A0
3. **Relay Module**:
   * VCC to 5V
   * GND to GND
   * Signal to D3
4. **OLED Display**:
   * VCC to 3.3V
   * GND to GND
   * SDA to A4
   * SCL to A5

**Software Setup**

**Arduino Code**

The Arduino sketch is written to:

1. Collect data from the DHT11 and soil moisture sensors.
2. Control the relay module for automatic watering.
3. Display real-time data on the OLED screen.
4. Transmit sensor data via BLE.

**Key Libraries Used:**

* U8g2lib: For OLED display management.
* DHT: For temperature and humidity readings.
* ArduinoBLE: For Bluetooth communication.

**Python GUI**

The Python GUI application provides remote monitoring capabilities using the BLE data sent from the Arduino. The interface displays soil moisture, temperature, humidity, and soil condition. It also plays an alert sound when the soil condition is "Very Dry."

**Key Libraries Used:**

* bleak: For BLE communication.
* tkinter: For GUI development.
* pygame: For sound playback.

**Step-by-Step Implementation**

**Hardware Setup**

1. Connect the soil moisture sensor, DHT11, OLED display, and relay module to the Arduino Nano 33 BLE as per the wiring diagram.
2. Mount the water pump and connect it to the relay module.
3. Ensure all connections are secure and provide power to the system.

**Software Setup**

1. **Arduino IDE**:
   * Install required libraries (e.g., ArduinoBLE, U8g2lib, DHT).
   * Upload the provided Arduino sketch to the Nano 33 BLE.
2. **Python Environment**:
   * Install required libraries (bleak, pygame, tkinter).
   * Run the Python script to monitor data and trigger alerts.

**Operation**

1. Power on the system and ensure the OLED displays initial readings.
2. Use the Python GUI to connect to the BLE device named "PlantMonitor."
3. Monitor the plant’s status and receive notifications for critical conditions.

**Screenshots and Illustrations**

**Device and Sensor Integration**

* A photo showing the Arduino Nano 33 BLE, DHT11 sensor, soil moisture sensor, OLED display, and relay module connected on a breadboard.

**OLED Display Output**

* Screenshot of the OLED showing:
  + Moisture: 650
  + Soil: Moist
  + Temp: 24.5°C
  + Humidity: 60%

**Python GUI**

* Screenshot of the Python GUI displaying real-time data:
  + Moisture Level: 650
  + Temperature: 24.5°C
  + Humidity: 60%
  + Soil Condition: Moist

**Bluetooth Communication**

* Screenshot from nRF Connect showing the BLE device and its characteristics.

**Challenges and Solutions**

**Challenge 1: Interfacing Components**

**Issue:** Ensuring the DHT11 sensor, soil moisture sensor, OLED display, and BLE module worked seamlessly together. **Solution:** Incrementally added components to the Arduino sketch, testing each individually before integrating them.

**Challenge 2: BLE Communication**

**Issue:** Maintaining stable BLE communication between the Arduino and Python GUI. **Solution:** Used the bleak library for reliable BLE interaction and optimized the data format for compatibility.

**Challenge 3: Power Supply**

**Issue:** Voltage drops when powering multiple components. **Solution:** Used a dedicated 5V power supply and verified current requirements for each module.

**Challenge 4: Moisture Sensor Calibration**

**Issue:** Determining the optimal moisture threshold for watering. **Solution:** Calibrated the sensor by measuring values for wet, moist, and dry soil conditions, adjusting thresholds accordingly.

**Reflections and Future Improvements**

This project successfully demonstrates how IoT technologies can simplify plant care. Real-time monitoring and automation reduce manual effort while ensuring plant health. However, there are areas for improvement:

1. **Battery-Powered Operation:** Adding a battery and low-power modes for portability.
2. **Mobile App Integration:** Developing a dedicated app for more advanced monitoring and control.
3. **Extended Sensors:** Including light and air quality sensors for more comprehensive monitoring.

Overall, this project highlights the potential of combining sensors, BLE communication, and automation to create smart systems for everyday tasks.