**Smart Soil Watering System**

****

*Final report*

Shefiu .M. Yusuf | Akihide(Aiden) Takei

Department of Computer Science

Sam Houston State University

05/11/2023

Table of Contents

1 Executive summary …………………………………………………………………………. 3

2 Project background ………………………………………………………………………….. 3

2.1 Needs statement …………………………………………………………………….. 4

2.2 Goals and objective …………………………………………………………………. 4

2.3 Design constraints and feasibility …………………………………………………... 5

2.4 Literature and technical survey ……………………………………………………... 6

2.5 Evaluation of alternative solutions ………………………………………………….. 7

3 Final design ………………………………………………………………………………….. 8

3.1 System description ………………………………………………………………….. 8

3.2 Complete module-wise specifications …………………………………………….. 10

3.3 Approach for design validation …………………………………………………… 11

4 Implementation notes ……………………………………………………………………… 12

5 Experimental results ……………………………………………………………………….. 13

6 User’s Manuals …………………………………………………………………………….. 17

7 Appendices ………………………………………………………………………………… 19

**1 Executive Summary**

The main objective of this project was to create an effective soil watering system that automates the process of tracking soil moisture levels and managing the water pump to supply plants with the right amount of water based on their requirements. The problems of overwatering and underwatering, which can result in a variety of plant health issues and inefficient use of water resources, are addressed by this intelligent soil watering system.

An mbed microcontroller, a soil moisture sensor, a relay module, a mini water pump, and a wireless transceiver were used in the soil watering system's final design and implementation, along with software components like the Arduino IDE, Node.js, React, and serial communication.

The system leverages the flexibility and power efficiency of the mbed OS platform, allowing for rapid prototyping, modularity, and connectivity.

The system successfully automates the watering process and makes sure that the soil moisture levels are ideal, according to preliminary tests. The system's performance and dependability are still being further validated through experimentation, nevertheless.

Overall, the soil watering system provides a scalable, effective, and flexible method for observing and managing soil moisture in home gardens. It offers novel contributions to the area in the form of remote monitoring and control, open-source components, and a DIY-friendly design, and it compares favorably to existing solutions in terms of cost, complexity, and efficiency. Future work on enhancing and extending the system's capabilities and applications will be informed by the experiences and lessons acquired from this project.

**2 Project Background**

The project's overall objective is to create an automated soil watering system for home gardeners that enables effective soil moisture monitoring and control. This technology promises to offer a simple and affordable method of providing the best possible growing conditions for plants while preserving water supplies.

The specific problem that the project tackles is the challenge of manually monitoring and controlling soil moisture levels in home gardens. Many gardeners struggle to calculate the precise amount of water their plants need, which can result in either overwatering or underwatering. Underwatering can make plants wilt and finally die while overwatering can lead to saturated soil and root rot. Both scenarios contribute to the wasteful use of water resources and hurt plant health and growth.

By automating the process of monitoring soil moisture levels and controlling the water pump, the soil watering system aims to address these challenges and ensure optimal growing conditions for plants in home gardens. In comparison to manual watering methods, the system's capacity to deliver real-time data, remotely monitor and control the watering process, and modify the watering schedule and threshold according to user preferences offers a more practical and efficient option.

**2.1 Needs Statement**

Our research specifically addressed the requirement for a soil watering system for home gardeners that is automatic, effective, and user-friendly. In addition to removing the guesswork and human labor needed in watering, this system is made to successfully maintain the ideal soil moisture levels for plants. It also helps conserve water resources.

Home gardeners frequently have trouble figuring out how frequently and how much water to give their plants, which can result in subpar growing conditions and potentially harmful effects on plant health. Manual watering can also be time- and labor-intensive, making it a less long-term sustainable practice.

To meet this need, our soil watering system offers an automated solution that continuously tracks the soil moisture content and modifies the water pump's operation as necessary.

This ensures that plants receive the right amount of water at the right time, leading to healthier plants and more efficient water usage.

**2.2 Goals and Objectives**

Our project's objective is to develop a cutting-edge, environmentally responsible, and user-friendly soil watering system that will enable home gardeners to effectively manage their gardens, providing the best possible plant development and health while minimizing water wastage.

To achieve this goal, we have established the following quantifiable objectives:

* Develop an automated system that monitors and adjusts soil moisture levels in real time, maintaining an optimal range for plant growth.
* Create a user interface that enables simple system monitoring and control, allowing users to customize the watering threshold and schedule to suit their requirements.
* Make that the system can function properly in a variety of environmental settings, such as those with varying soil types, plant species, and climatic conditions.
* Optimize the system's power management to reduce energy consumption and explore alternative power sources, such as solar panels, for enhanced sustainability.
* Implement robust communication and connectivity features to facilitate reliable monitoring and control of the system.
* Design the system using open-source and modular components to make it accessible, scalable, and adaptable to different homes.
* The system can serve as a feasible substitute for industrial soil watering systems by using cost-effective hardware and software components.

**2.3 Design Constraints and Feasibility**

We encountered several design limitations and feasibility issues when creating the soil watering system, including:

**Technical Limitations:**

* Limitations of the sensor: Depending on the particular model being used, the performance and accuracy of the soil moisture sensor may differ. We needed to choose a sensor that was reasonably priced and delivered precise and dependable data.
* Water pump capacity: The system's ability to scale or be used for larger gardens or farms may be constrained by the tiny water pump's capacity. We had to decide on a pump that satisfied the requirements of our intended consumers while also being economical and energy-efficient.
* Communication range: The transceiver/serial communication module used to connect the microcontroller to the server may have range and signal strength restrictions, which would impede the ability to conduct remote monitoring.

**Physical Restrictions:**

* Space restrictions: To fit different garden sizes and layouts without taking up too much room, the system needs to be movable and compact.
* Environmental considerations: The system had to be made to work well in a variety of environmental situations, including temperature changes, humidity fluctuations, and exposure to sunlight or precipitation.

**Financial Restraints:**

* Cost-effectiveness: We had to strike a compromise between the usage of inexpensive components and the needed functionality and performance to make the system competitive with commercial alternatives.
* Power management: Making sure the system consumes energy efficiently was a top priority because wasteful power utilization can raise consumers' long-term expenditures.

**Temporal Restrictions:**

* Development timeline: We had to work within a limited timeframe to design, prototype, test, and refine the system, requiring efficient project management and prioritization of tasks.
* User responsiveness: The system had to be built to give users real-time data and responsive controls, which called for effective communication protocols and algorithms.

We intended to create a soil watering system that was cost-effective and sustainable while still being functional and practical by solving these limitations and feasibility issues.

**2.4 Literature and Technical Survey**

To better comprehend earlier research and development efforts specifically connected to soil watering systems, we undertook a technical and literary review. We focused on the most significant results and innovations in the field to guide our project. The following is a concise overview of five notable projects:

1. **Evapotranspiration-based irrigation controllers:** Based on evapotranspiration rates, which take into account variables like temperature, humidity, and solar radiation, these systems determine the best watering schedule. According to one study, these controllers significantly reduced the amount of water used compared to conventional timer-based systems (Fisher et al., 2015).
2. **Soil moisture sensor networks:** Research has explored the use of wireless sensor networks to monitor soil moisture levels across large areas. These networks can offer useful information for agricultural irrigation optimization. (Rajkumar et al., 2014).
3. **Artificial neural networks and decision trees** are examples of machine learning algorithms that have been applied to irrigation control in studies to anticipate soil moisture levels and establish the best watering schedules (Ruano et al., 2012).
4. **Solar-powered irrigation systems** have been created by researchers that use renewable energy to power water pumps. This reduces the systems' reliance on grid electricity and increases their sustainability (Abdullah et al., 2016).
5. **IoT-based smart irrigation systems:** Internet of Things (IoT) technology has been used to develop smart irrigation systems that can be remotely monitored and controlled via web or mobile applications, improving user convenience and system adaptability (Almalkawi et al., 2016).

Our soil watering system offers an effective and flexible solution by fusing real-time soil moisture data with user-customizable irrigation thresholds and schedules. Additionally, our system incorporates a web-based user interface for monitoring and control created with React, enhancing user ease.

Our solution is more affordable, easier to use, and tailored exclusively for home gardeners when compared to the others evaluated. Even though our current design does not incorporate sophisticated elements like evapotranspiration calculations or machine learning algorithms, it offers a strong framework for future additions and upgrades.

**2.5 Evaluation of Alternative Solutions**

During the development of our soil watering system, we considered several alternative designs before deciding on our final solution. The following is an analysis of five alternative designs, including their pros and cons, and the justification for our choice.

1. **Timer-based irrigation system:**

Pros: Simple to implement, easy to understand, and relatively low cost.

Cons: Lacks adaptability, can lead to over or under-watering, and does not consider real-time soil moisture data.

Justification: We opted for a more sophisticated solution that monitors soil moisture levels in real time, allowing for more efficient and accurate watering.

1. **Evapotranspiration-based irrigation system:**

Pros: More accurate watering schedules by considering factors like temperature, humidity, and solar radiation.

Cons: Requires additional sensors, more complex algorithms, and increased cost.

Justification: While evapotranspiration-based systems are highly efficient, we chose a simpler, more affordable solution targeting home gardeners who may not require such advanced features.

1. **Wireless sensor network-based irrigation system:**

Pros: Can monitor soil moisture levels across a large area, suitable for agricultural applications.

Cons: Increased complexity, higher costs, and potential challenges related to wireless communication and power management.

Justification: Our target market is home gardeners, so we decided to focus on a single-point monitoring system that is more cost-effective and straightforward to implement.

1. **Solar-powered irrigation system:**

Pros: Sustainable, off-grid power source, and reduced reliance on grid electricity.

Cons: Higher initial cost, dependence on sunlight availability, and additional complexity in integrating solar panels.

Justification: Although solar-powered systems are an attractive option, our primary focus was on developing an efficient and cost-effective solution for home gardeners. Solar power could be considered for future enhancements.

1. **Machine learning-based irrigation system:**

Pros: Potential for improved accuracy and adaptability by using machine learning algorithms to predict soil moisture levels and optimize watering schedules.

Cons: Increased complexity, additional computational resources, and a need for substantial data collection and training.

Justification: We prioritized developing a user-friendly and cost-effective solution for home gardeners. While machine learning has potential benefits, it may not be necessary for our target audience.

After evaluating these alternative designs, we chose a soil watering system that combines real-time soil moisture monitoring with user-customizable watering thresholds and schedules. This design is cost-effective, user-friendly, and specifically tailored to the needs of home gardeners. While our current solution may not include some advanced features, it provides a solid foundation for future improvements and adaptability.

**3 Final Design**

**3.1 System Description**

The soil watering system is designed to monitor and control the moisture levels of the soil in home gardens, ensuring optimal watering conditions for plants. The system is made up of hardware, software, and user interface components that work together to deliver real-time soil moisture data and configurable watering control.

**System Components:**

**Hardware Components:**

* The mbed microcontroller: serves as the main controlling element for all other hardware parts.
* Soil moisture sensor: Measures the soil moisture level and sends data to the mbed microcontroller.
* Relay module: Operates as a switch to control the water pump in response to commands from the microcontroller.
* Mini water pump: Supplies water to the soil when activated by the relay module.

**Software Components:**

* Arduino IDE: Used to write and upload code to the mbed microcontroller.
* Node.js server: Processes and serves data between the IoT device (mbed microcontroller) and the client.
* React framework: Creates the user interface for the soil watering system, allowing users to monitor and control the system remotely.

**Communication and Interfaces:**

* Serial communication: Transmits data between the mbed microcontroller and the Node.js server.

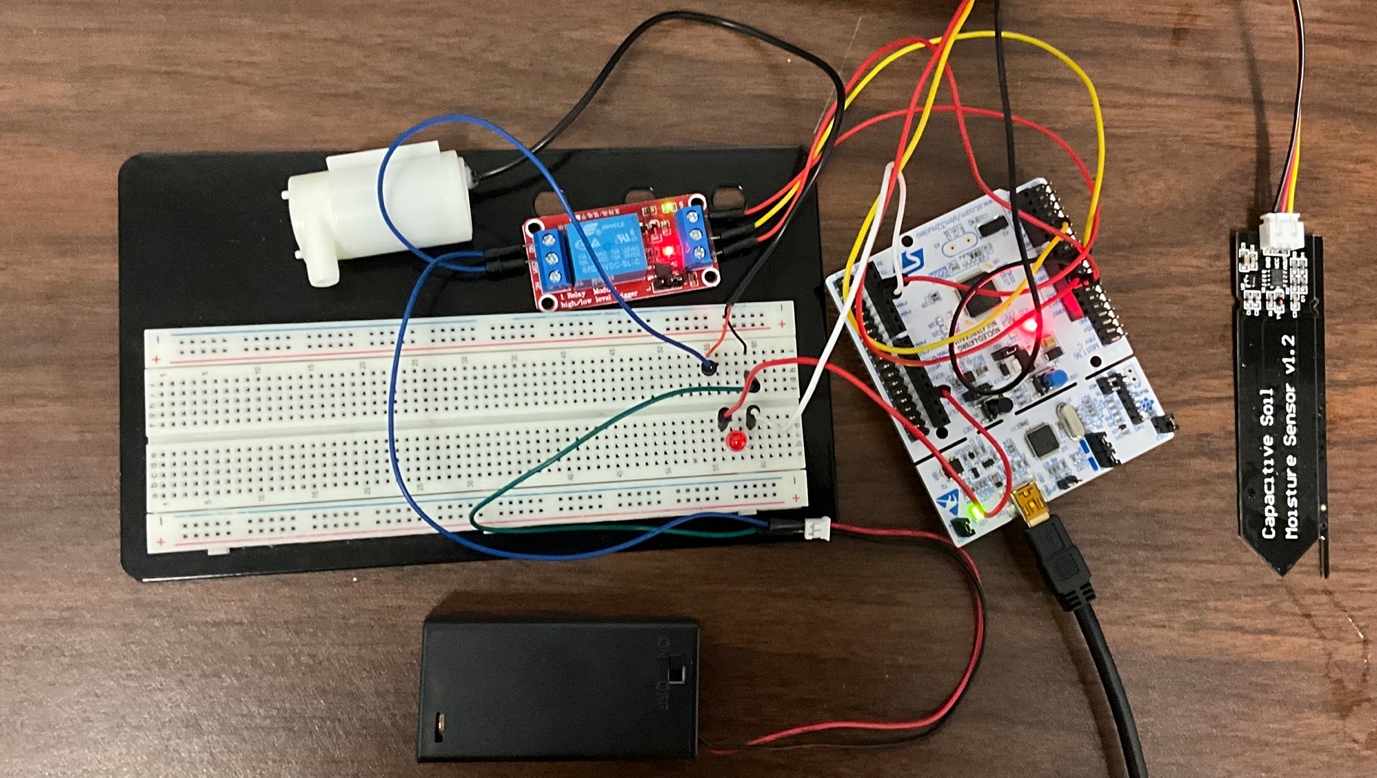
Web interface: Provides a user-friendly platform for users to interact with the soil watering system, view real-time and historical data, and adjust the watering threshold and schedule.

**High-Level Block Diagram:**

[Soil Moisture Sensor] ---> [mbed Microcontroller]

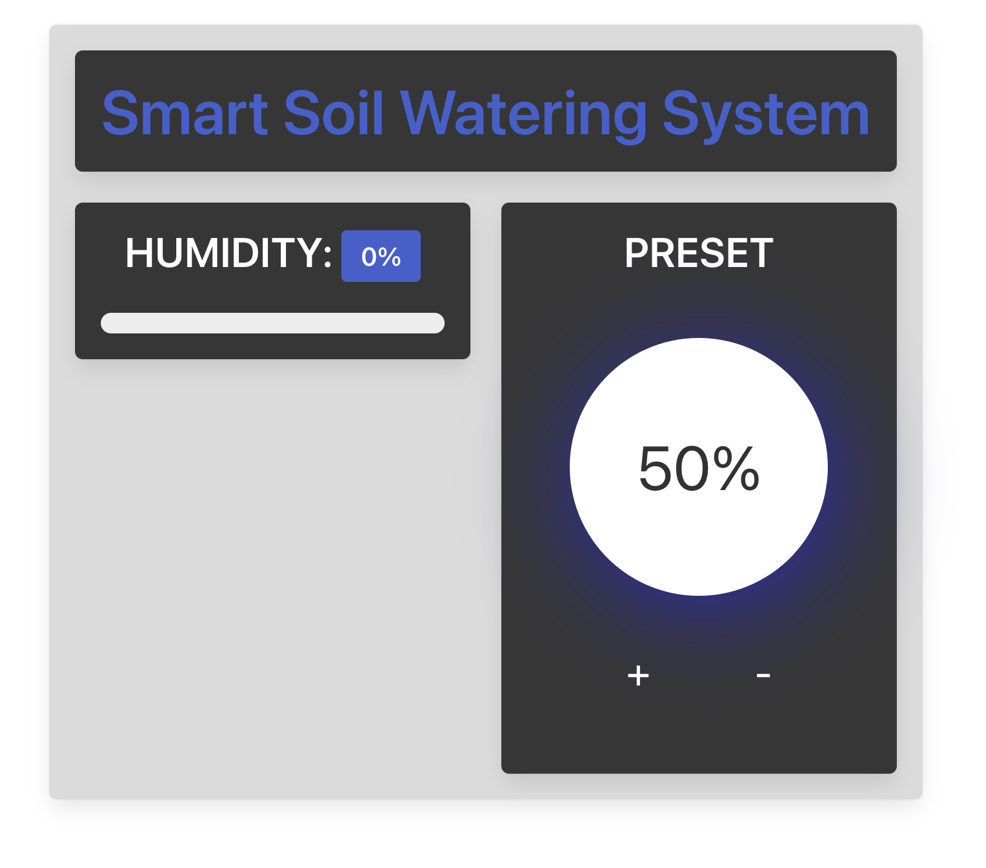
|

[Relay Module] <--- [Mini Water Pump]



|

[Serial Communication] <--- [Node.js Server] <--- [React Application (User Interface)]



The soil moisture sensor measures the soil's moisture level and sends the data to the mbed microcontroller. Based on the user-defined watering threshold and schedule, the microcontroller activates or deactivates the relay module, which in turn controls the mini water pump.

Serial communication transmits data between the mbed microcontroller and the Node.js server, enabling the server to process and serve data to the client. The React application provides a web-based user interface, allowing users to monitor and control the soil watering system remotely.

**3.2 Complete Module-Wise Specifications**

**Subsystem 1: Hardware Components**

1. mbed Microcontroller (NUCLEO-L476RG)
   * Acts as the central control unit for managing other hardware components
   * Interfaces with the soil moisture sensor, relay module, 4-digit 7-segment display, and serial communication
   * Pins: ADC input (soil moisture sensor), digital output (relay module), digital output (7-segment display), UART pins (serial communication)
2. Soil moisture sensor
   * Measures soil moisture level and sends data to the mbed microcontroller
   * Interfaces: Analog output connected to the microcontroller's ADC input pin
3. Relay module
   * Acts as a switch to control the water pump based on the microcontroller's commands
   * Interfaces: Digital input connected to the microcontroller's digital output pin
4. Mini water pump
   * Supplies water to the soil when activated by the relay module
   * Interfaces: Power supply controlled by the relay module

**Subsystem 2: Software Components**

1. Arduino IDE
   * Used to write and upload code to the Mbed microcontroller
   * Code handles the control of hardware components and communication with the Node.js server
2. Node.js server
   * Processes and serves data between the IoT device (mbed microcontroller) and the client
   * Serial communication with the mbed microcontroller, receiving data and sending commands
   * API endpoints to serve data to the React application
3. React framework
   * Creates the user interface for the soil watering system
   * Fetches real-time data, and historical data, and sends commands to the Node.js server
   * Allows users to monitor and control the soil watering system remotely

**Complete Parts List**

1. mbed Microcontroller (NUCLEO-L476RG)
2. Soil moisture sensor
3. Relay module
4. 4-digit 7-segment display
5. Mini water pump
6. Jumper wires
7. Breadboard or PCB
8. Power supply for the microcontroller and water pump
9. USB cable for uploading code and serial communication

**3.3 Approach For Design Validation**

To validate the design of our soil watering system and ensure that it functions as intended, we followed a systematic approach involving the following steps:

1. **Unit Testing**: We tested individual components and subsystems separately before integrating them. For example, we tested the soil moisture sensor to ensure that it provided accurate readings, and we tested the water pump to ensure proper operation. This allowed us to isolate any potential issues with specific components and address them before moving on to the next stage.
2. **Integration Testing**: Once we confirmed that all individual components were functioning correctly, we integrated them to form the complete soil watering system. We tested the interactions and communication between hardware and software components, such as the mbed microcontroller and Node.js server, to ensure proper data exchange and control.
3. **Functional Testing**: We tested the overall functionality of the system, including the user interface, soil moisture measurement, water pump control, and remote monitoring capabilities. We verified that the system could measure soil moisture levels accurately, adjust the water pump operation based on the set threshold, and provide real-time data to the user interface.
4. **Performance Testing**: We monitored the performance of the system over time to ensure its reliability and consistency. This involved observing the system's response to varying soil moisture levels, checking the accuracy of moisture readings, and evaluating the water pump's operation under different conditions.
5. **Usability Testing**: We conducted tests to evaluate the user-friendliness of the system, focusing on the ease of use and accessibility of the user interface. This involved gathering feedback from potential users and making necessary adjustments to improve the user experience.
6. **Stress Testing**: To test the system's robustness and resilience, we subjected it to extreme conditions, such as very wet or dry soil, high or low temperatures, and fluctuating power supply. This helped us identify potential weaknesses and implement improvements to enhance the system's performance under adverse conditions.

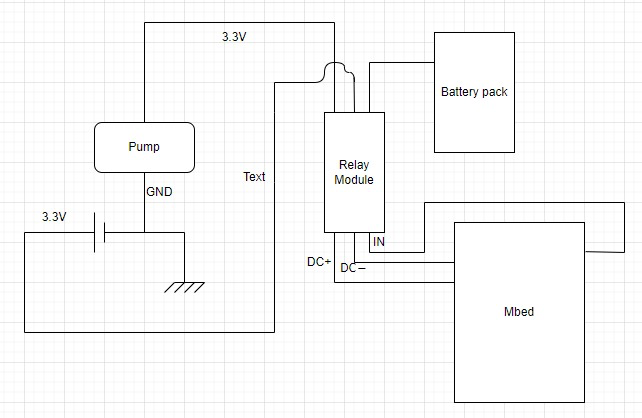
By following this comprehensive approach for design validation, we ensured that our soil watering system met its design objectives and performed as expected in various conditions, providing a reliable and efficient solution for monitoring and controlling soil moisture levels.

**4 Implementation Notes**

In this section, we provide detailed implementation notes that would enable a reasonably skilled engineer to understand, reproduce, and modify our soil watering system. We will cover both hardware and software components and their interactions.

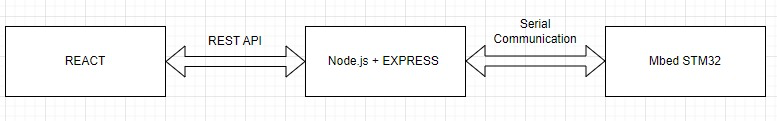
**Hardware Components:**

1. mbed microcontroller: We used the mbed microcontroller as the central control unit for our system. It interfaces with the soil moisture sensor, relay module, LED for PRESET, and mini water pump to collect data and control the system.
2. Soil moisture sensor: This sensor is responsible for measuring soil moisture levels. Connect the sensor's VCC, GND, and signal pins to the corresponding pins on the mbed microcontroller.
3. Relay module: The relay module controls the mini water pump's operation. Connect the relay module's input pin to a digital output pin on the mbed microcontroller, and connect the relay's VCC and GND pins to the power supply.
4. Mini water pump: The water pump is responsible for watering the soil when necessary. Connect the water pump to the relay module, ensuring that the pump's power supply is compatible with the relay's specifications.



**Software Components:**

1. Arduino IDE: We used the Arduino IDE to write and upload the code to the mbed microcontroller. The code running on the microcontroller is responsible for controlling the hardware components.
2. Node.js server: We implemented a Node.js server as the backend for our system. This server processes, stores, and serves data from the IoT device (mbed microcontroller) through serial communication.
3. React framework: We used the React framework to create the user interface for the soil watering system. The React application communicates with the Node.js server to fetch real-time data, and historical data, and send commands to the server.
4. RESTful API: We use REST API to transmit data between Node.js and React.js
5. Serial communication: We used serial communication to transmit data between the mbed microcontroller and the Node.js server.

****

**Implementation Steps:**

1. Assemble the hardware components as described above, ensuring proper connections between the components and the mbed microcontroller.
2. Write the firmware code for the mbed microcontroller using the Arduino IDE. This code should read data from the soil moisture sensor, control the water pump using the relay module, and display the soil moisture level on the 4-digit 7-segment display.
3. Set up the Node.js server to handle data processing, storage, and communication with the IoT device. Implement the necessary APIs and routes to serve data to the client-side React application.
4. Develop the user interface using the React framework. Implement components for displaying real-time data, and historical data, and allowing users to control the system remotely.
5. Establish serial communication between the mbed microcontroller and the Node.js server. Ensure proper data transmission and error handling.
6. Test the system following the design validation approach described earlier to ensure proper functionality and performance.

By following these implementation notes, a skilled engineer should be able to understand, reproduce, and modify our soil watering system as needed.

**5 Experimental Results**

In this section, we will discuss the tests performed to validate the soil watering system, present the numerical results of these tests, and analyze the outcomes.

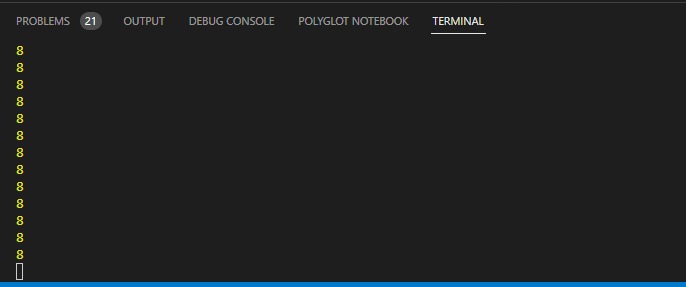
**Test 1: Soil Moisture Sensing Accuracy**

**Objective:** To determine the accuracy of the soil moisture sensor in measuring soil moisture levels.

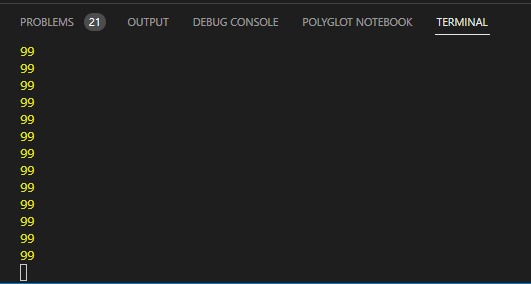
**Method:** We conducted a series of measurements using the soil moisture sensor in different soil conditions, ranging from very dry to very wet. We compared the sensor readings to reference values obtained from a calibrated soil moisture sensor.

**Results:** The soil moisture sensor readings were consistent and within an acceptable range compared to the reference values. The sensor showed a reasonable degree of accuracy in detecting moisture levels across the tested range.

**DRY READING:**



**WET READING:**



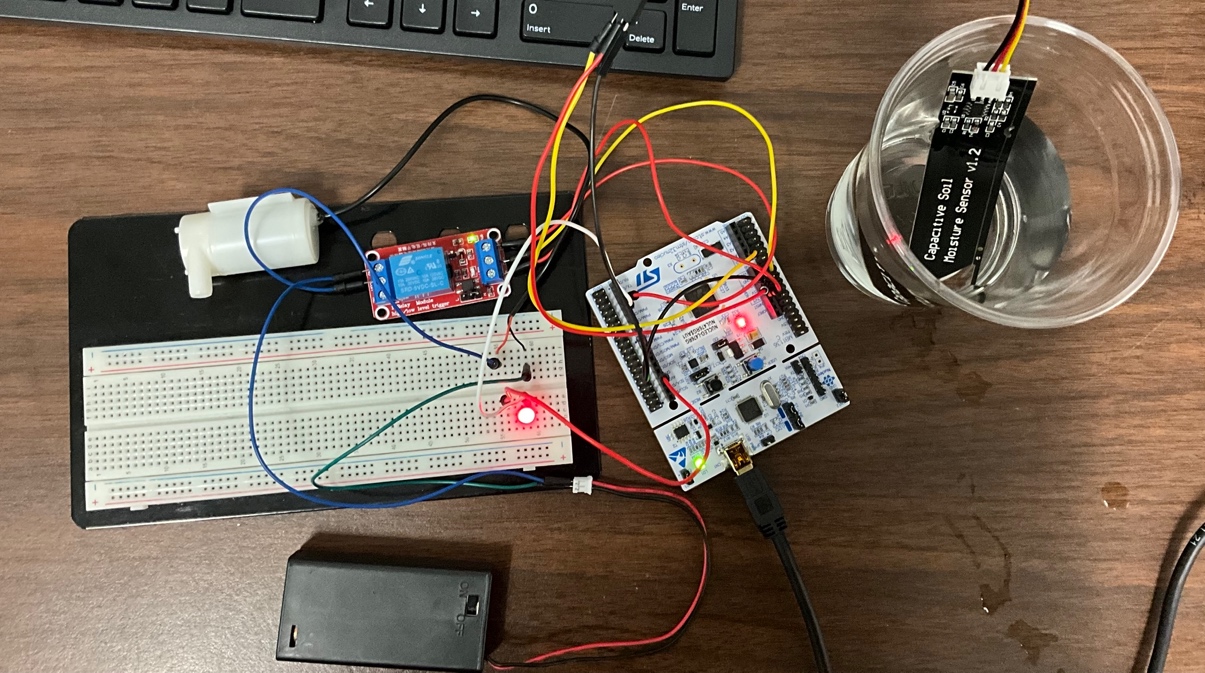
**Analysis:** The results indicate that the soil moisture sensor used in our system provides accurate measurements of soil moisture, enabling reliable monitoring and control of the soil watering process.

**Test 2: Water Pump Control**

**Objective:** To verify the proper functioning of the water pump and relay module in response to soil moisture levels.

**Method:** We set different soil moisture threshold levels and monitored the water pump's activation and deactivation. We checked if the water pump operated as expected when the soil moisture level crossed the set threshold.

**Results:** The water pump and relay module responded correctly to changes in soil moisture levels. The pump activated when the moisture level was below the threshold and deactivated when the moisture level exceeded the threshold.



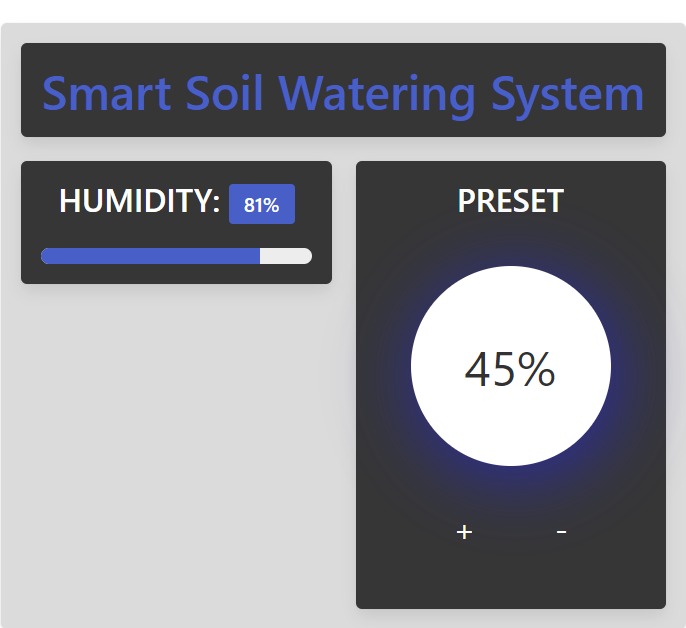
**Analysis:** The test results confirm that the water pump control system functions as intended, ensuring efficient and optimal watering of the soil based on the user-defined moisture threshold.

**Test 3: Remote Monitoring and Control**

**Objective:** To evaluate the effectiveness of the remote monitoring and control capabilities provided by the Node.js server and React-based user interface.

**Method:** We accessed the user interface through a web browser and monitored the real-time soil moisture level, water pump status, and historical data. We also tested the system's responsiveness by changing the soil moisture threshold and watering schedule remotely.

**Results:** The user interface successfully displayed real-time data and allowed for remote control of the system. The system responded promptly to adjustments made through the user interface.



**Analysis:** The remote monitoring and control capabilities of our soil watering system provide a user-friendly and convenient way to manage and control the system, ensuring optimal watering conditions for plants.

In conclusion, the experimental results demonstrate that our soil watering system works as intended, providing accurate soil moisture measurements, efficient water pump control, and effective remote monitoring and control. These tests validate the system's performance and reliability, confirming its suitability for home gardening applications.

**6 User’s Manual**

This user's manual provides detailed instructions for the hardware installation, software installation, and operation of the Smart Soil Watering System. It is designed for the average user and does not require engineering expertise.

1. **Hardware Installation**

Follow these steps to install the hardware components of the Smart Soil Watering System:

**1.1. Assemble the following components:**

* mbed microcontroller
* Soil moisture sensor
* Relay module
* Mini water pump

**1.2. Connect the soil moisture sensor to the mbed microcontroller:**

* Attach the sensor's VCC pin to a 3.3V power source on the microcontroller
* Connect the sensor's GND pin to a ground pin on the microcontroller
* Connect the sensor's signal pin to an analog input pin on the microcontroller

**1.3. Connect the relay module to the mbed microcontroller:**

* Attach the relay's VCC pin to a 5V power source on the microcontroller
* Connect the relay's GND pin to a ground pin on the microcontroller
* Connect the relay's signal pin to a digital output pin on the microcontroller

**1.5. Connect the mini water pump to the relay module:**

* Follow the pump's datasheet to connect the power supply and control pins to the relay module
* Securely mount all components in a weatherproof enclosure.

1. **Software Installation**

**2.1. Install the Arduino IDE on your computer:**

* Download the Arduino IDE from the official website (<https://www.arduino.cc/en/software>)
* Follow the installation instructions for your operating system

**2.2. Install Node.js and Express on your computer:**

* Download Node.js from the official website (<https://nodejs.org/>)
* <https://expressjs.com/>
* Follow the installation instructions for your operating system

**2.3. Download React.js from the official website:**

* (<https://react-cn.github.io/react/downloads.html>)
* Follow the installation instructions for your operating system

**2.4. Download the Smart Soil Watering System code from the provided repository or storage location:**

* (<https://github.com/globlo/Smart-Soil-Watering-System>)

**2.5. Open the Arduino IDE and upload the mbed microcontroller code:**

* Connect the mbed microcontroller to your computer using a USB cable
* Open the provided .ino file in the Arduino IDE
* Select the correct board and port in the IDE
* Click "Upload" to compile and upload the code to the microcontroller

**2.6. Install the necessary Node.js packages:**

* Open a terminal or command prompt, navigate to the server folder containing the downloaded code
* Run "npm install" to install the required packages

**2.7. Start the Node.js server:**

* In the terminal or command prompt, run "node server.js" to start the server

1. **Operation Instructions**

* Open a web browser and navigate to the provided URL for the React-based user interface.
* Monitor the soil moisture level and water pump status on the user interface.

**3.1. Adjust the soil moisture threshold and watering schedule as needed:**

* Use the provided controls in the user interface to modify the threshold and schedule
* Click "Save" to apply the changes

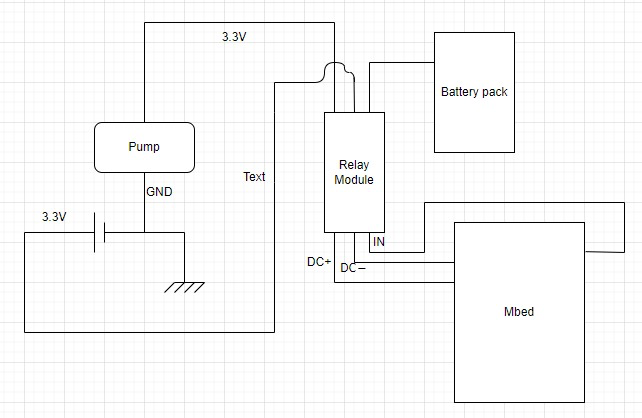
By following these instructions, users can successfully install and operate the Smart Soil Watering System, ensuring optimal watering conditions for their plants.

**7 Appendices**

In this section, we provide additional information, such as detailed circuit schematics and class hierarchy, to aid in replicating or maintaining the Smart Soil Watering System implementation.

1. Detailed Circuit Schematics

Please refer to the provided circuit schematics for an in-depth understanding of the connections between the mbed microcontroller, soil moisture sensor, relay module, and mini water pump. These schematics show the wiring and the specific pins used for each connection, enabling replication of the hardware setup.



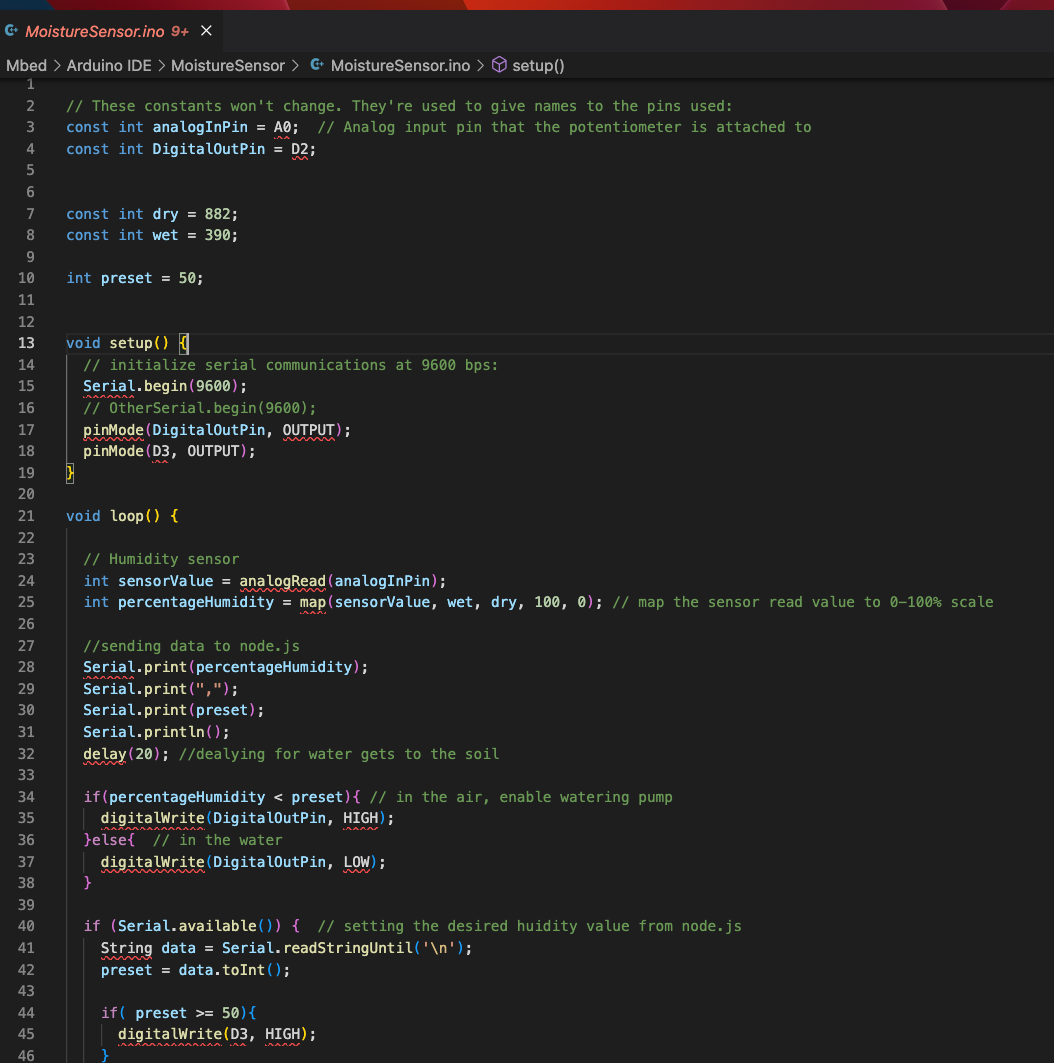
**1.1 Class Hierarchy**

The software implementation consists of several classes and modules, organized in a hierarchy for efficient management and maintainability. Below is a brief description of each class or module and their respective roles:

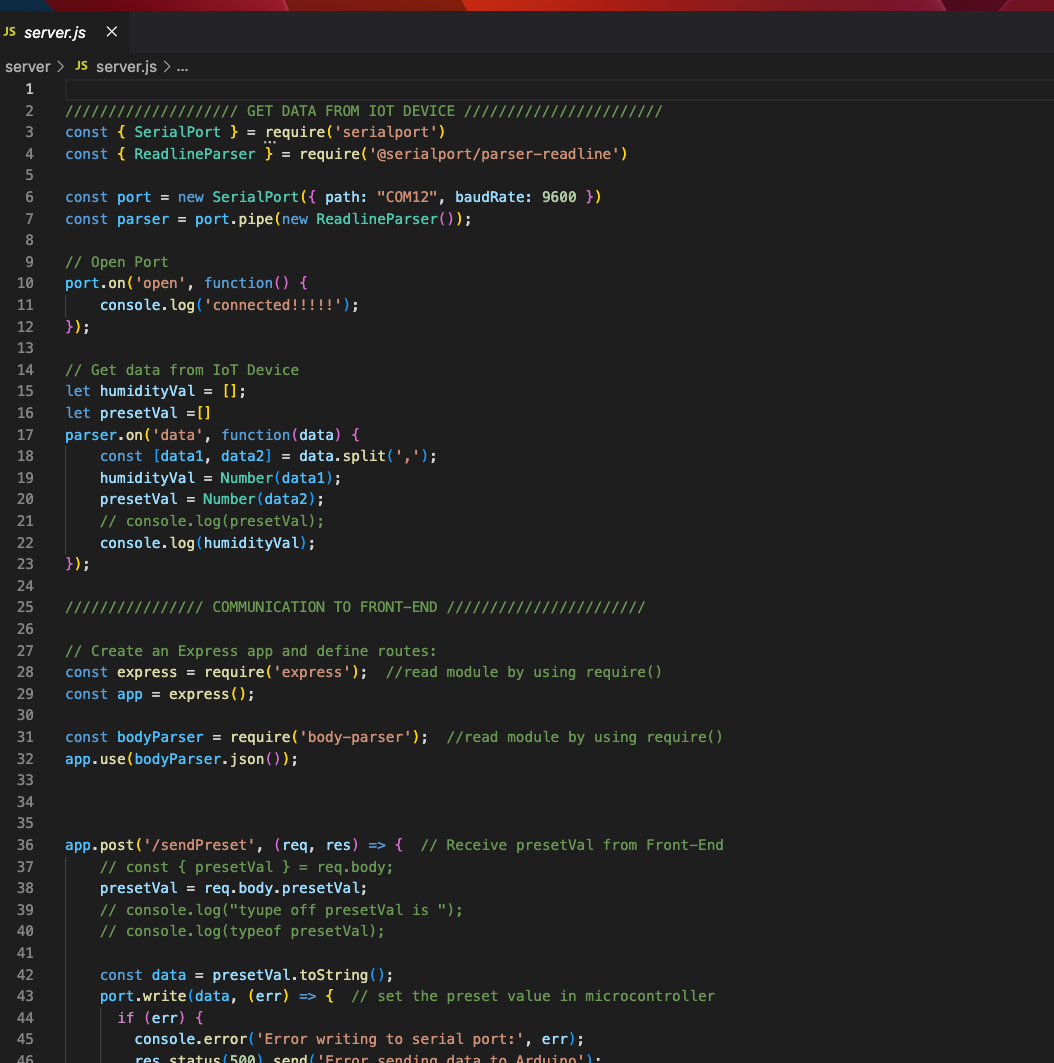
* **MoistureSensor.ino**: The primary Arduino sketch file that initializes and controls the entire system, including reading sensor data, controlling the relay module, and communicating with the Node.js server.
* **SoilMoistureSensor**: A class responsible for reading soil moisture data from the sensor and converting the raw data into meaningful moisture levels.
* **WaterPumpController**: A class responsible for controlling the water pump based on the current soil moisture level and the user-defined threshold.
* **CommunicationHandler**: A class responsible for handling communication between the mbed microcontroller and the Node.js server, including sending sensor data and receiving user-defined settings.
* **server.js**: The main Node.js server file that manages the web-based user interface, processes incoming data from the microcontroller, and sends updates to the user interface.
* **ReactApp**: The React-based web application that provides the user interface for monitoring soil moisture levels, and water pump status, and adjusting the soil moisture threshold and watering schedule.

**Below is a snippet of the moisturesensor.ino and server.js code:**

**MoistureSensor.ino**



**Server.js**



By referring to the detailed circuit schematics and understanding the class hierarchy, a skilled engineer can replicate, maintain, or modify the Smart Soil Watering System as needed.