Mini-Project Report On:

SMART IRRIGATION SYSTEM USING ESP8266

Contents

LIST OF FIGURES

CHAPTER 1

1.1 INTRODUCTION

In today's world, the growing trend of smart home systems is extending into agriculture and gardening. With the advancement of the Internet of Things (IoT), it's now possible to automate tasks that traditionally required manual effort, like watering plants. The smart irrigation system is an IoT-based solution aimed at monitoring and watering plants based on real-time soil moisture levels. This project utilizes the NodeMCU ESP8266 board—a low-cost microcontroller that features integrated Wi-Fi—along with a soil moisture sensor to detect the soil's wetness. Based on the moisture readings, the system can activate a water pump through a relay module. Additionally, the system is integrated with the Blynk platform, which enables remote monitoring and control via a smartphone application. With this automated system, you can ensure your plants are always properly watered without needing to be physically present, and you can make manual adjustments or receive alerts directly from your smartphone.

1.2 OBJECTIVES

- To design and implement an automated plant watering system using NodeMCU ESP8266.
- To monitor soil moisture levels in real-time and automatically water plants based on sensor data.
- To integrate the system with the latest Blynk IoT platform for remote control and monitoring.
- To provide real-time data visualization and manual override through the Blynk mobile app.
- To promote water conservation by ensuring watering only when the plant actually needs it.
- Water Conservation: This helps prevent overwatering, which can be harmful to the plants and waste water.

1.3 METHODOLOGY

The implementation of this project involves a structured series of steps that covers hardware interfacing, embedded programming and real time monitoring of the system. The methodology is broken down into the following stages :

1. **System Initialization :**
   - When the system is powered on, the NodeMCU ESP8266 connects to Wi-Fi.
   - The LCD screen shows the message "Soil Moisture: 0%" to show the system is ready.
   - The Blynk app connects to the NodeMCU, allowing the user to control the water pump with a button.

2. **Measuring Soil Moisture:**
   - The soil moisture sensor checks the moisture level of the soil.
   - The NodeMCU reads the sensor's data to figure out how wet or dry the soil is.
   - This data is shown as a percentage on the LCD screen (0% for dry soil, 100% for wet soil).

3. **Displaying Moisture on LCD:**
   - The LCD screen updates in real-time, showing the current soil moisture percentage so you can always see if your plant needs watering.

4. **Controlling the Water Pump with Blynk App:**
   - In the Blynk app, there is a button to manually control the water pump.
   - When you press ON in the app, the NodeMCU turns on the relay, which powers the water pump.
   - When you press OFF, the pump is turned off.

5. **Relay and Water Pump:**
   - The relay acts like a switch for the water pump.
   - When the relay is ON, the water pump is powered and starts watering the plant.
   - When the relay is OFF, the water pump stops working.

6. **Continuous Monitoring:**
   - The system keeps checking the soil moisture all the time, updating the LCD with the current moisture level.
   - You can control the water pump whenever you want using the Blynk app.

7. **Feedback and Control:**
   - The LCD screen always shows the soil moisture, so you know if your plant needs watering.
   - The Blynk app lets you manually turn the water pump on or off, depending on what you see on the LCD.

CHAPTER 2

2.1 DESIGN

1. System Design Overview:

The design consists of several key components:

- NodeMCU ESP8266: A Wi-Fi-enabled microcontroller that serves as the central unit for data processing and communication.
- Soil Moisture Sensor: Detects the moisture level in the soil to determine when watering is needed.
- Water Pump: Activated to irrigate the plant when the soil is dry.
- Relay Module: Acts as a switch to control the water pump based on signals from the NodeMCU.
- Blynk App: An IoT platform that allows users to monitor soil moisture levels and control the watering system remotely via a smartphone.
- +12v power supply, wires, breadboard

Functionality:

- The soil moisture sensor continuously monitors the moisture level of the soil.
- When the moisture level drops below a predefined threshold, the NodeMCU activates the relay, turning on the water pump to irrigate the plant.
- The system sends real-time data to the Blynk app, allowing users to monitor soil conditions and manually control the pump if necessary.

2. Circuit Design:

Below is a simplified circuit diagram of how the components are connected:

Connections:

- Soil Moisture Sensor:
  - VCC → 3.3V on NodeMCU
  - GND → GND on NodeMCU
  - A0 (Analog Pin) → A0 on NodeMCU (to read moisture data)

- Relay Module:
    - VCC → 5V on NodeMCU (or external power supply)
    - GND → GND on NodeMCU
    - IN → D1 (digital pin) on NodeMCU (used to control the relay)

- Water Pump:
    - VCC → External power supply (depends on pump voltage, e.g., 12V)
    - GND → GND of external power supply
    - Relay Output (NO terminal) → One terminal of the water pump
    - Other terminal of pump → GND of the external power supply

3. Software Configuration:

- **Blynk App :**Create a new project and add widgets such as a gauge to display soil moisture levels and a button to manually control the pump. Obtain the authentication token provided by Blynk for the project.

- **Arduino IDE :**Install the necessary libraries: Blynk, ESP8266WiFi.Write a program that reads the analog value from the soil moisture sensor, sends this data to the Blynk app, and controls the relay based on the moisture level .Include the Blynk authentication token, Wi-Fi SSID, and password in the code.

2.1.1 CIRCUIT DIAGRAM

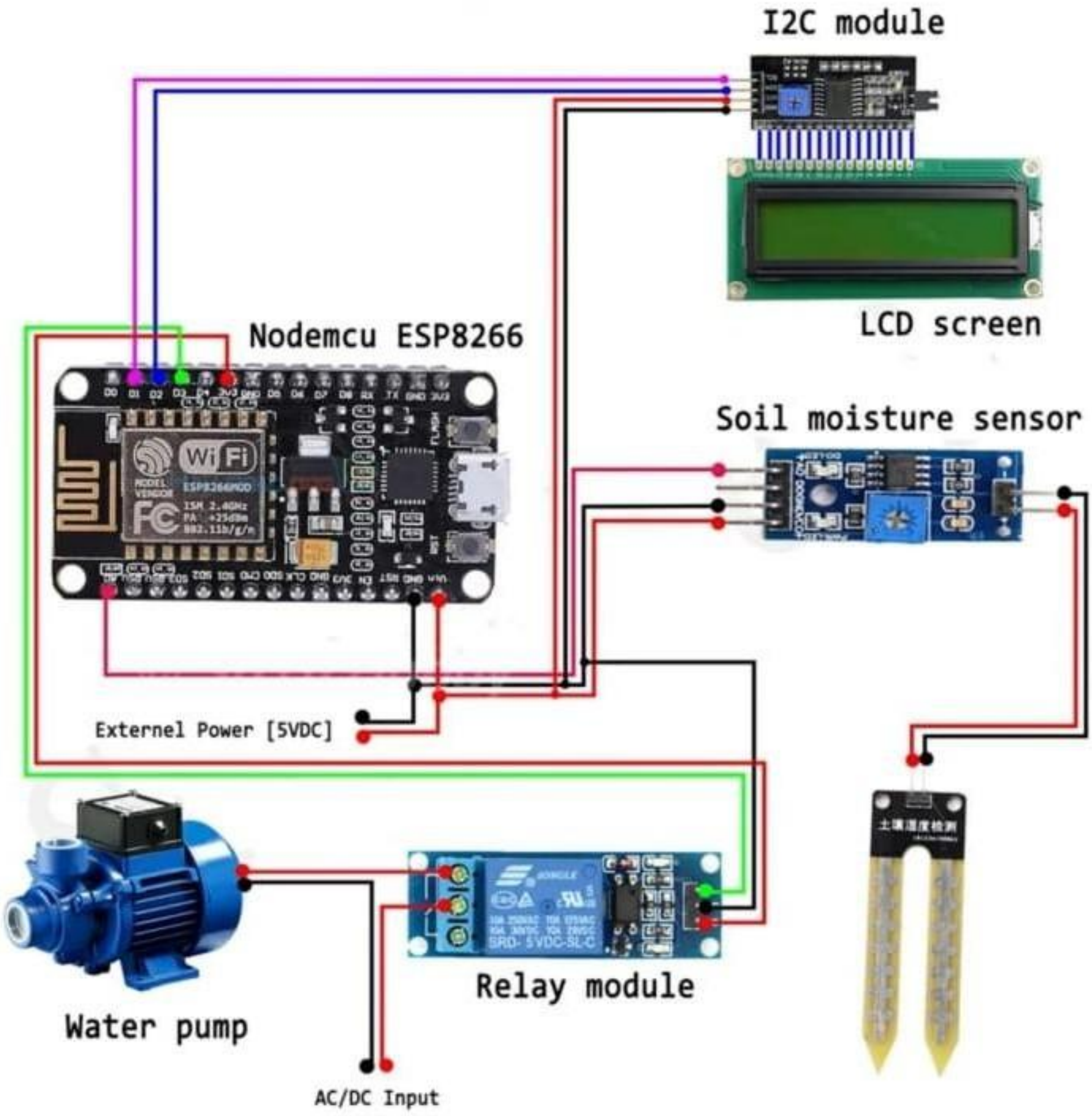

Figure 2.1: Circuit Diagram Of Smart Irrigation System

As shown in figure 2.1 illustrates a smart irrigation system using the NodeMCU ESP8266 Wi-Fi module. The main components are a soil moisture sensor, LCD display (via I2C), a relay module, and a water pump. Below is a detailed explanation of the connections and operation:

COMPONENTS:

1. NodeMCU ESP8266 – Main controller with Wi-Fi capability.
2. Soil Moisture Sensor – Measures the moisture level in soil.
3. I2C LCD Module (with 16x2 LCD) – Displays sensor readings and status.
4. Relay Module – Acts as a switch to control the AC water pump.
5. Water Pump – Delivers water when soil is dry.
6. External 5V Power Supply – Powers NodeMCU and connected modules.

HARDWARE CONNECTIONS:

1. NodeMCU ESP8266:

- D1 (GPIO5) → SCL (I2C LCD module)
- D2 (GPIO4) → SDA (I2C LCD module)
- D5 (GPIO14) → IN (Relay module)
- A0 → Analog Output of Soil Moisture Sensor
- Vin → Relay module VCC
- 3V3 → Soil Moisture Sensor VCC
- GND → Connected to all GNDs (Relay, LCD, Sensor)

2. Soil Moisture Sensor:

- AO (Analog Output) → A0 of NodeMCU
- VCC → 3V3 of NodeMCU
- GND → GND of NodeMCU
- Probes inserted into soil

3. I2C LCD Module:

- SDA → D2 (GPIO4) of NodeMCU
- SCL → D1 (GPIO5) of NodeMCU
- VCC → 3V3
- GND → GND

4. Relay Module:

- IN → D5 (GPIO14) of NodeMCU
- VCC → Vin
- GND → GND
- Relay Output (NO, COM) connected to AC water pump

5. Water Pump:

- Connected to AC supply through Relay switch

OPERATION:

1. Moisture Detection:
   - Soil moisture sensor reads moisture level and sends analog data to A0.
   - NodeMCU reads the value and decides whether soil is dry.
2. Display:
   - I2C LCD shows current soil moisture status (e.g., "Soil is Dry" or "Moist").
3. Watering Trigger:
   - If the soil is dry, NodeMCU sends a HIGH signal to the relay via D5, turning ON the water pump.
   - When soil is wet enough, NodeMCU turns OFF the relay, stopping the pump.
4. Remote Monitoring (Optional):
   - ESP8266 can send data over Wi-Fi (to Blynk, Firebase, etc.) for remote monitoring/control.

2.1.2 FLOWCHART

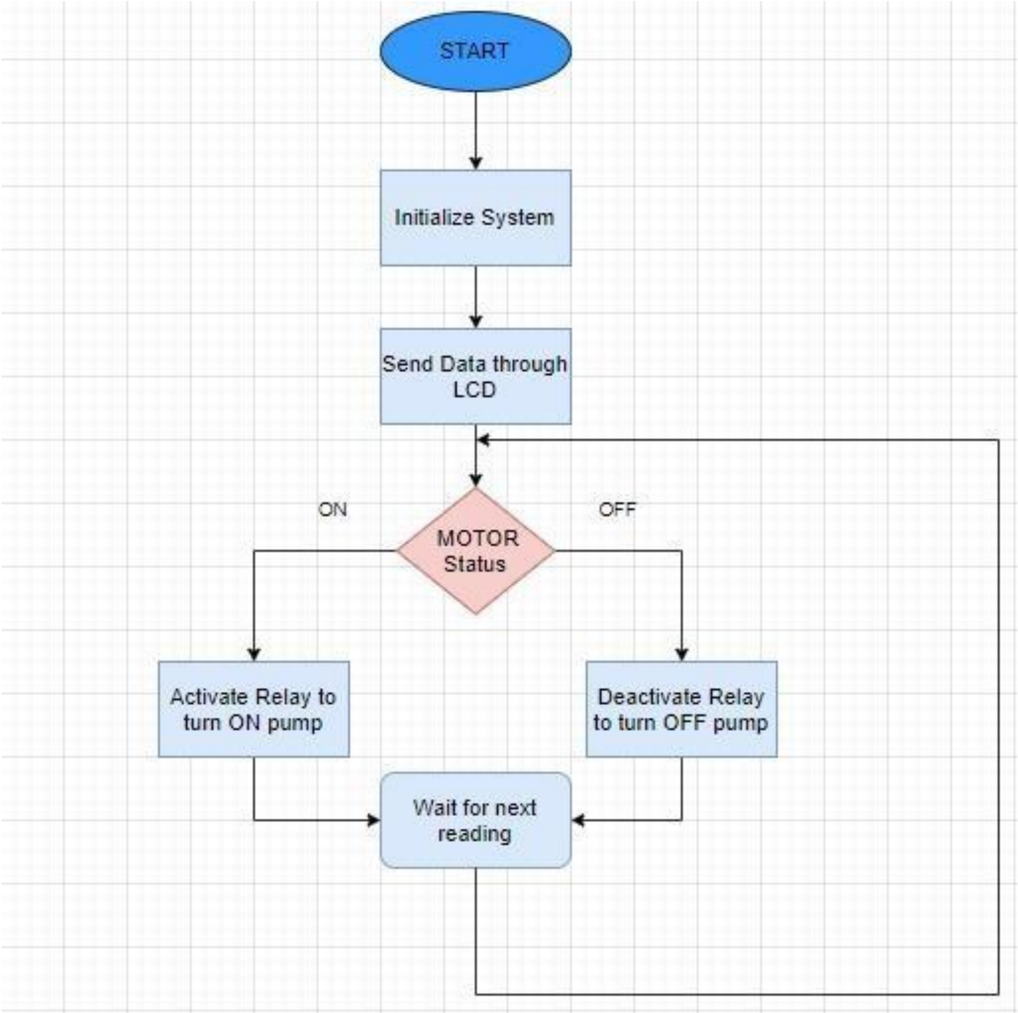

Figure 2.2: Flowchart Showing The Smart Irrigation System Operation

As shown in figure 2.2 shows explanation of flowchart

- START

The system begins its operation.

- Initialize System

All necessary hardware and software components (e.g., sensors, relays, microcontroller) are set up.
Additionally, the system connects to a Wi-Fi network if remote monitoring or control is required.

- Send Data through LCD

Real-time data such as system initialization status, connectivity, or motor readiness is displayed on the LCD.

- Check MOTOR Status

Decision block to check the status command for the motor:

If "ON" → proceed to activate the pump.

If "OFF" → ensure pump remains off.

- If MOTOR = ON: Activate Relay to Turn ON Pump

The system turns ON the water pump by activating the relay switch.

- If MOTOR = OFF: Deactivate Relay to Turn OFF Pump

The system turns OFF the water pump by deactivating the relay.

- Wait for Next Reading

A delay period is introduced before the next cycle begins.

This delay allows for system stability and avoids rapid polling.

- Loop Back

After the wait, the system checks the motor status again and repeats the process.

2.2 CODE

```
#define BLYNK_TEMPLATE_ID "TMPL3F6yajGCJ"
#define BLYNK_TEMPLATE_NAME " Quickstart Template"
//Include the library files
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

//Initialize the LCD display
LiquidCrystal_I2C lcd(0x27, 16, 2);

char auth[] = " IXELf_b4qioeucFRWvqhf2_eOD96F3vO ";//Enter your Auth token
char ssid[] = " vivo t4x ";//Enter your WIFI name
char pass[] = "9113599393";//Enter your WIFI password

BlynkTimer timer;
bool Relay = 0;

//Define component pins
#define sensor A0
#define waterPump D3
void setup() {
Serial.begin(9600);
  pinMode(waterPump, OUTPUT);
  digitalWrite(waterPump,  HIGH);
  lcd.init();
  lcd.backlight();
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  lcd.setCursor(1, 0);
  lcd.print("System Loading");
  for (int a = 0; a <= 15; a++) {
  lcd.setCursor(a, 1);
  lcd.print(".");
    delay(500);
  }
  lcd.clear();

  //Call the function
  timer.setInterval(100L, soilMoistureSensor);
}
//Get the button value
BLYNK_WRITE(V1) {
  Relay = param.asInt();
  if (Relay == 1) {
    digitalWrite(waterPump, LOW);
    lcd.setCursor(0, 1);
    lcd.print("Motor is ON ");
  } else {
    digitalWrite(waterPump, HIGH);
    lcd.setCursor(0, 1);
    lcd.print("Motor is OFF");
  }
}

//Get the soil moisture values
```

```
void soilMoistureSensor() {
  int value = analogRead(sensor);
  value = map(value, 0, 1024, 0, 100);
  value = (value - 100) * -1;
  lcd.setCursor(0, 0);
  lcd.print("Moisture :");
  lcd.print(value);
  lcd.print(" ");
}
void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```

CHAPTER 3

3.1 IMPLEMENTATION

The Smart Irrigation System was implemented using a combination of hardware components and software programming to automate the watering process based on soil moisture levels. The system is built around the NodeMCU ESP8266 microcontroller, which serves as the central processing unit. It continuously monitors soil moisture data from a sensor and controls a water pump via a relay module.

A soil moisture sensor is inserted into the plant's soil to detect the current moisture level. The sensor outputs an analog signal proportional to the soil's moisture content. This analog value is read by the NodeMCU’s ADC pin and compared against a predefined threshold value set in the program. If the moisture level is found to be below the threshold, indicating dry soil, the NodeMCU sends a HIGH signal to the relay module, which switches on the water pump. Water is then supplied to the soil through a pipe connected to the pump.

A 16x2 LCD screen (connected via an I2C module) displays real-time values of soil moisture and the motor's current status (ON or OFF), helping users visually monitor system performance. Additionally, the NodeMCU is connected to the internet via Wi-Fi, enabling remote monitoring and control through a mobile application (like Blynk). The user can view the moisture level and manually override the motor state using the app if required.

All components are mounted on a breadboard and powered using a 5V DC supply. The setup ensures a compact, energy-efficient, and scalable design. Through this implementation, the system achieves automated irrigation, remote control, and efficient water usage, making it suitable for both small-scale home gardens and larger agricultural applications.

3.2 RESULT

WHEN SOIL MOISTURE IS ZERO(0)

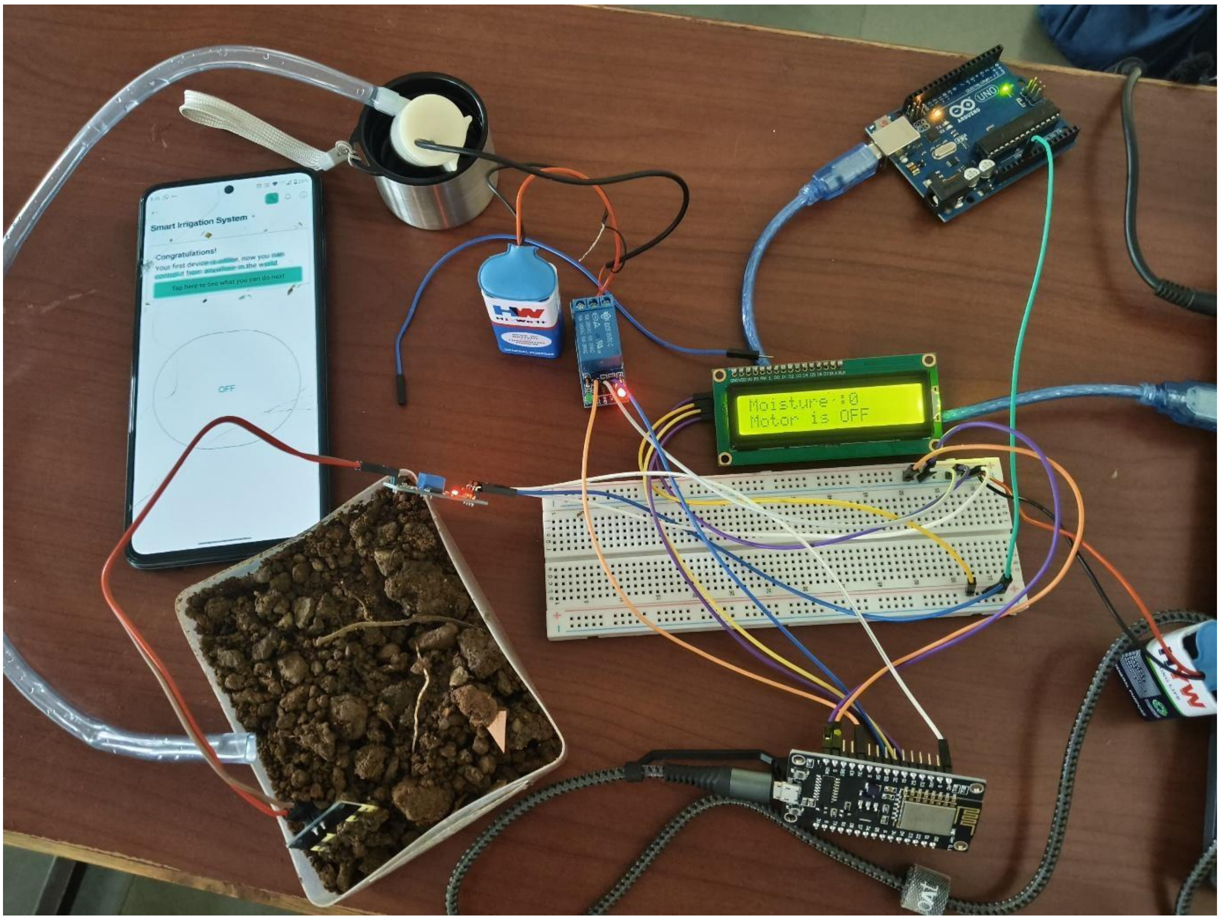

Figure 3.1 shows the output when soil is in dry condition

As shown in figure 3.1 when the soil moisture sensor detects a value of zero, it signifies that the soil is completely dry and requires immediate watering. The sensor sends this analog reading to the NodeMCU ESP8266, which continuously monitors the soil condition. Upon receiving this low moisture value, the NodeMCU triggers a relay module by sending a HIGH signal to its input pin. This, in turn, activates the water pump connected to the relay, allowing water to be supplied to the soil through a pipe. At the same time, the 16x2 LCD screen updates to display the current moisture level as "Moisture: 0" along with the message "Motor is ON", clearly indicating that the system has detected dryness and has taken action.

Furthermore, if integrated with a mobile application (e.g., Blynk), the system also sends real-time updates to the user’s smartphone over Wi-Fi. The app interface reflects the motor status as ON, and optionally allows the user to override the automatic behavior via a control button. The pump continues to operate until the moisture level rises above a pre-defined threshold (indicating sufficient water content in the soil). Once this happens, the NodeMCU automatically turns off the relay, thereby switching off the motor to conserve water. This intelligent decision-making process ensures that plants are watered only when necessary, promoting efficient water usage, remote monitoring, and automated smart irrigation without human intervention.

WHEN SOIL CONDITION IS WET

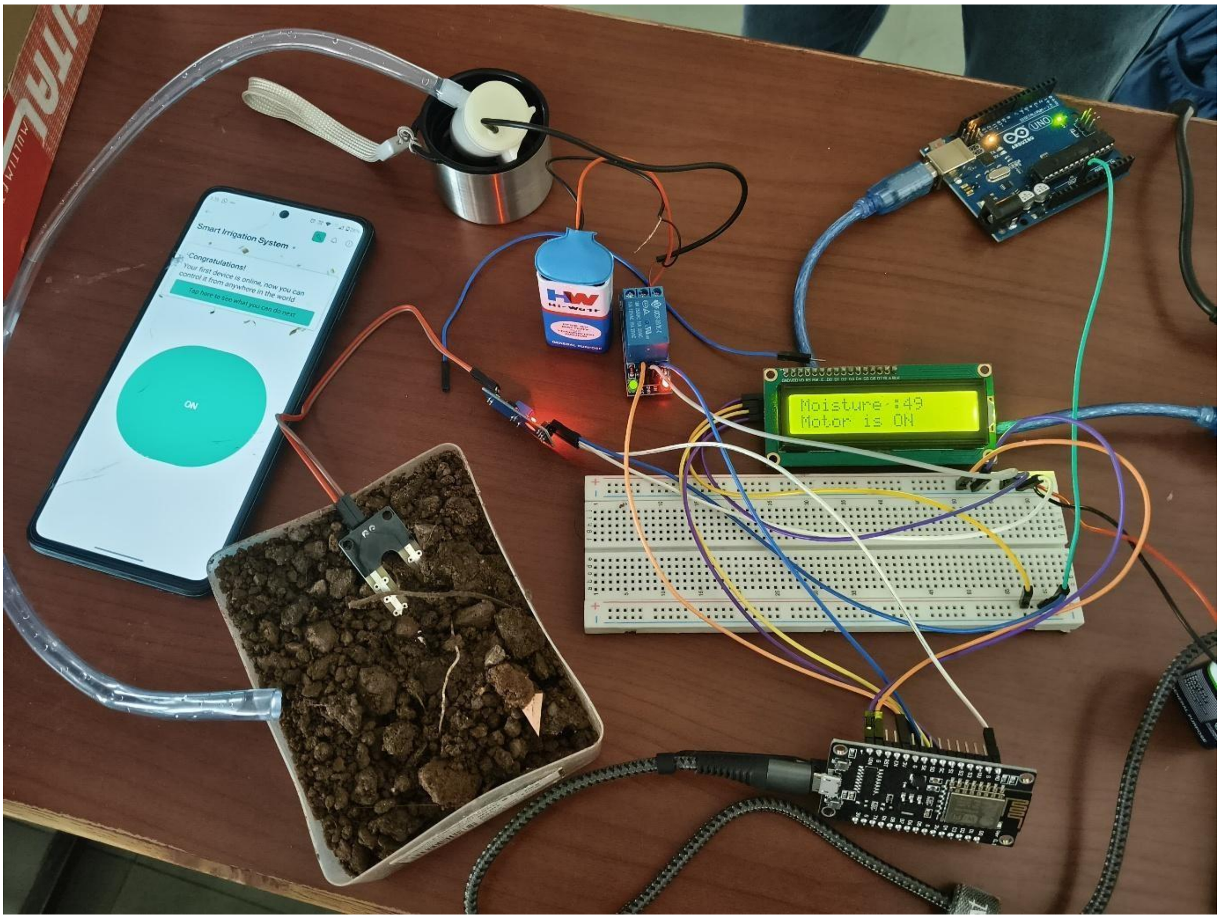
Smart Irrigation System
Congratulations!
Your first device is online, now you can control it from anywhere in the world
Tap here to see what you can do next
ON
Moisture :49
Motor is ON
UNO

Figure 3.2 shows the output when soil is in wet condition

As shown in the figure 3.2 the soil moisture sensor detects a value of 49, which is considered below the preset moisture threshold required for healthy plant growth (typically around 60 or higher). The NodeMCU ESP8266 microcontroller reads this sensor value and identifies that the soil is not sufficiently moist. As a result, it triggers the relay module, which activates the water pump to begin irrigating the soil.

This indicates that the system is actively responding to the detected soil condition and supplying water to the plants. Additionally, the smartphone application interface (such as Blynk) updates in real time, displaying the motor status as "ON", along with a large green button that allows the user to manually toggle the motor if needed.

Water is delivered through a pipe connected to a submersible pump, which is powered through a relay switch and a 9V battery or DC power source. As the pump operates, the moisture level in the soil gradually increases. The system continuously monitors the moisture readings and, once the value surpasses the desired threshold, it will automatically turn OFF the motor, thereby conserving water and preventing over-irrigation.

This stage in the operation demonstrates the system's ability to make autonomous decisions based on real-time environmental data, enhancing agricultural efficiency through automated irrigation and remote control capabilities.

3.3 CONCLUSION

The Smart Irrigation System using the ESP8266 NodeMCU successfully demonstrates the application of IoT in automating agricultural practices. By integrating a soil moisture sensor with a relay-controlled water pump and the Blynk IoT platform, the system ensures plants are watered based on real-time soil moisture levels. The solution provides a reliable and cost-effective method for conserving water, reducing human intervention, and promoting smart farming practices. The system also provides user-friendly remote monitoring and control through a smartphone interface and displays real-time data via an LCD, ensuring transparency and ease of use.

3.2.1 Future Work

- Integration with Weather Forecasting APIs: Incorporate weather prediction data to optimize irrigation cycles based on expected rainfall.
- Multi-Sensor Network: Extend the system to support multiple plants or fields using several moisture sensors.
- Solar Power Support: Power the system with renewable energy sources like solar panels for sustainable deployment in remote areas.
- Automatic Nutrient Dispensing: Add a mechanism to deliver fertilizers or nutrients based on plant growth cycles.
- Cloud Database and Analytics: Store historical data in a cloud database for trend analysis and machine learning applications.
- Voice Assistant Integration: Integrate with Google Assistant or Alexa for voice-based control of irrigation functions.

3.4 REFERENCES

[1] NodeMCU ESP8266 Official Documentation – https://nodemcu.readthedocs.io/

[2] Blynk IoT Platform – https://blynk.io/

[3] LiquidCrystal_I2C Library for LCD – Arduino Library Manager

[4] Soil Moisture Sensor Datasheet – Various open-source electronics vendors

[5] Arduino IDE Official Website – https://www.arduino.cc/