A

MINI-PROJECT REPORT ON

"PLANT WATERING SYSTEM USING IOT"

Submitted in partial fulfilment of requirements for the award of degree of

Bachelor of Technology

in

Electronics and Telecommunication Engineering

Submitted by

Khushwant Digambar Bhure :2210121372014

under the guidance of

Prof. S. S. Sarode

Department of Electronics and Telecommunication Engineering

Government College of Engineering, Yavatmal

Maharashtra State, India

2024-2025

Department of Electronics and Telecommunication Engineering

Government College of Engineering, Yavatmal

Maharashtra State, India

2024-2025

CERTIFICATE

This is to certify that the Mini-Project report entitled,

"PLANT WATERING SYSTEM USING IOT"

is a bona fide Mini-Project work and has been carried out by

Khushwant Digambar Bhure (2210121372014)

of third year B-Tech class under the guidance of **Prof. S. S. Sarode** during the academic year 2024 -2025 (Sem- VI).

Prof. S. S. Sarode
Mini Project Guide

Prof. Dr. S. R. Shiledar
Head 0f Electronics and Telecommunication Department

Prof. Dr. S. D. Londhe
Principal

ACKNOWLEDGEMENT

Apart from individual efforts, the success of any mini-project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental throughout the mini-project work. It is our privilege to express our gratitude towards our mini-project guide, **Prof. S. S. Sarode** for her valuable guidance, encouragement, inspiration and wholehearted cooperation throughout the mini-project work. We thank him for being a motivation through all our highs and importantly, our lows.

We deeply express our sincere thanks to our Head of department **Prof. Dr. S. R. Shiledar** for encouraging and allowing us to present the mini-project on the topic **"Plant Watering System Using Iot"** and providing us with the necessary facilities to enable us to fulfill our mini-project requirements as best as possible. We take this opportunity to thank all faculty members and staff of the Electronics and Telecommunication Engineering department who have directly or indirectly helped us in our mini-project.

We pay our respects to honourable Principal **Prof. Dr. S. D. Londhe** for his encouragement and motivation. Our thanks and appreciations also go to our family and friends, who have been a source of encouragement and inspiration throughout the duration of the Mini-Project.

ABSTRACT

The **Plant Watering System using IoT** is an intelligent and automated solution designed to take care of plants by monitoring soil moisture and controlling the water supply without manual intervention. This project is particularly useful for home gardens, indoor plants, and small-scale agricultural setups where consistent watering is essential for healthy plant growth.

The core of the system is the **ESP32 microcontroller**, which is a powerful and affordable Wi-Fi-enabled board ideal for Internet of Things (IoT) applications. It is interfaced with a **soil moisture sensor**, a **relay module**, and a **mini water pump**. The soil moisture sensor continuously checks the water content in the soil. When the moisture level drops below a preset threshold, the ESP32 sends a signal to the relay module, which in turn powers the water pump. The pump waters the plant until the soil moisture reaches a safe level, at which point it automatically stops.

What makes this project truly smart is the integration of the **Blynk IoT platform**. The ESP32 sends real-time sensor data to the Blynk cloud, allowing users to monitor soil moisture and control the pump from anywhere using a smartphone. The Blynk app provides a user-friendly interface that displays moisture levels, system status, and manual control buttons. Users can also receive push notifications or automate watering schedules based on predefined rules.

This system brings several advantages, such as efficient water usage, reduced manual effort, and increased convenience, especially for people who are busy or travel frequently. It also supports scalability—multiple sensors and pumps can be added to control larger garden areas. Additionally, the system is energy-efficient and can be powered using a USB adapter or a small battery pack. It is also possible to enhance the setup with solar panels for outdoor applications.

From an educational and practical standpoint, the project provides hands-on experience with microcontroller programming, sensor interfacing, IoT connectivity, and mobile app integration. It is a great example of how embedded systems and IoT can solve everyday problems sustainably.

In conclusion, this Plant Watering System demonstrates how automation and IoT can be combined to create smart agricultural solutions. It is a cost-effective and scalable system that not only simplifies gardening but also contributes to resource conservation and sustainable living.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 What is an Automatic Plant Watering System?

An automatic plant watering system is a smart solution that eliminates the need for manual watering by using electronic sensors and actuators to monitor soil conditions and irrigate plants based on real-time data. Traditional gardening and agricultural methods rely on human observation and periodic watering, which may lead to overwatering or underwatering. An automatic system leverages technology to maintain the ideal moisture level in the soil, thus ensuring healthier plant growth.

These systems typically consist of a soil moisture sensor, a microcontroller, and a water-pumping mechanism like a relay-controlled pump. The sensor continuously checks the moisture level in the soil. If the soil is too dry, the controller activates the water pump to supply water to the plant. Once the required moisture level is reached, the pump turns off. Some systems also include data monitoring and remote control features via Wi-Fi, making them suitable for integration with IoT platforms such as Blynk, ThingSpeak, or Ubidots.

The proposed system in this project makes use of ESP32, a low-cost microcontroller with built-in Wi-Fi and Bluetooth, enabling real-time communication with the Blynk IoT application. It allows users to remotely monitor the system and even operate it manually if required.

1.2 Importance of Automation in Agriculture

Agriculture is the backbone of many economies, and with the global population increasing, the demand for food and efficient farming practices is also growing. Automation in agriculture, commonly referred to as smart farming, is an evolving technology aimed at enhancing productivity while reducing manual labor and conserving resources.

One of the major challenges in agriculture is water management. Water scarcity is a global issue, and inefficient irrigation methods can lead to a significant waste of this valuable resource. Traditional watering methods are time-consuming and often inaccurate in determining plant needs. Here is where automation plays a critical role.

Using smart irrigation systems like the one proposed in this project can drastically reduce water wastage, save time, and ensure that crops receive optimal water levels, resulting in better yield and healthier plants. Such systems are especially beneficial for urban gardeners, greenhouse owners, hobbyists, and even large-scale farmers when scaled up.

Additionally, integrating the system with IoT platforms offers real-time data collection, historical tracking, and remote operation. This not only enhances the system's intelligence but also enables better decision-making and predictive analysis.

1.3 Objective of the Project

The main objective of this mini-project is to design and implement an automatic plant watering system that can monitor soil moisture levels and water the plant accordingly. The key goals of this project are as follows:

1. **Automated Monitoring**: To continuously monitor the moisture level of the soil using a soil moisture sensor and ensure the plant gets water only when necessary.
2. **Efficient Water Usage**: To avoid overwatering or underwatering, thereby saving water and promoting healthier plant growth.
3. **Remote Accessibility**: To allow users to check moisture levels and control the water pump remotely using the **Blynk IoT app** on their smartphones.
4. **Low-Cost and Scalable Design**: To create a budget-friendly system using the ESP32 microcontroller, suitable for small-scale or hobby farming, and easily scalable for larger applications.
5. **User-Friendly Interface**: To design a simple and intuitive interface in the Blynk app where users can view data, control actions, and receive alerts or notifications.
6. **Energy-Efficient Operation**: To develop a system that can run on minimal power, optionally using batteries or solar panels for outdoor applications.

This project combines embedded systems, IoT technology, and automation to deliver a smart solution that simplifies plant care and supports sustainable practices in modern agriculture.

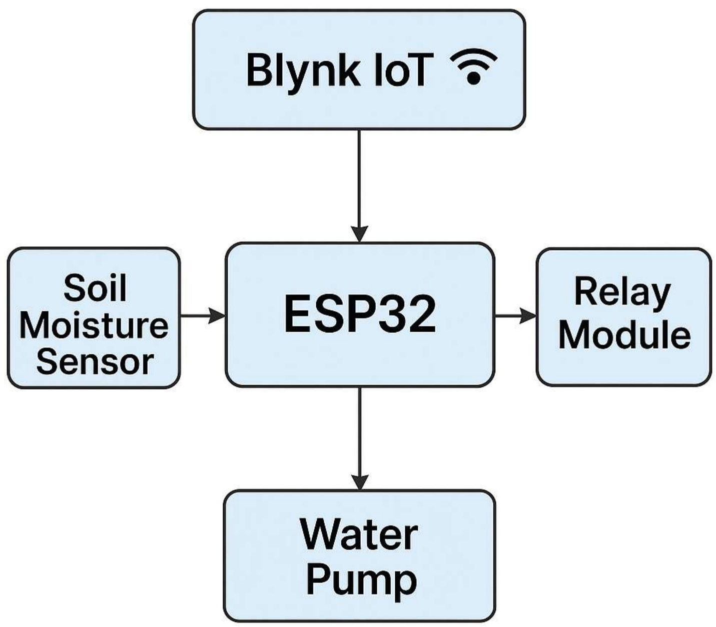

Fig. 1.1: Block Diagram of Plant Watering System

CHAPTER 2
LITERATURE REVIEW

2.1 Background

The integration of automation and Internet of Things (IoT) technologies into agricultural practices has opened new avenues for improving irrigation efficiency. Traditional plant watering techniques often rely on manual observation and fixed schedules, which can lead to overwatering or underwatering, negatively impacting plant health. As the demand for smart and sustainable solutions increases, researchers and hobbyists have focused on designing automated systems that utilize real-time environmental data to control irrigation dynamically.

B. Traditional and Early Automation Approaches

Manual watering is the most basic form of irrigation, requiring users to observe plant conditions and water them accordingly. However, this process is prone to inconsistency, human error, and inefficient water usage. Early attempts to automate irrigation involved using mechanical or digital timers to switch pumps on at fixed intervals. While these systems reduced labor, they lacked adaptability to changing soil conditions.
Saiyidah Nafisah Binti Rosli [2] explored one such timer-based system enhanced by microcontrollers like 8051 and PIC. Although these systems introduced automation by interfacing with soil moisture sensors, they lacked internet connectivity and remote control features, limiting their flexibility and applicability in remote or urban settings.

C. Emergence of IoT in Smart Irrigation

The evolution of wireless communication and low-power embedded systems has led to a new generation of IoT-based irrigation systems. K. Soderby [1] demonstrated a smart watering solution using Arduino IoT Cloud that allows users to monitor and control irrigation remotely using mobile applications. The system reads soil moisture values in real-time and triggers watering actions accordingly, improving efficiency and reducing resource wastage.
T. Prasanth et al. [3] further validated the effectiveness of IoT by developing a plant monitoring system with live feedback using the ESP8266 microcontroller. Their work emphasized the role of wireless connectivity and cloud-based analytics in optimizing irrigation schedules and minimizing water consumption.

D. Microcontroller Selection

Microcontroller selection plays a vital role in the design of IoT systems. The Arduino Uno, while widely adopted, lacks native wireless capabilities and often requires additional modules for connectivity. On the other hand, the ESP32 microcontroller has emerged as a popular choice due to its integrated Wi-Fi and Bluetooth, low power consumption, and higher processing capabilities. The Circuit Digest team [4] highlighted the ESP32 as a cost-effective and powerful solution for scalable IoT applications, especially in agriculture.

E. IoT Platform Utilization

Platforms like Blynk simplify IoT implementation by providing mobile and web-based interfaces to control hardware over the cloud. As explained by IoT Starters [5] and various community tutorials [6][7], Blynk supports the use of virtual pins, real-time data visualization, notifications, and remote control. Its drag-and-drop mobile interface makes it ideal for students and hobbyists developing smart plant watering systems.

F. Soil Moisture Sensors and Accuracy

Soil moisture sensors form the cornerstone of smart irrigation systems. They are typically categorized into resistive and capacitive types. Resistive sensors are low-cost but degrade over time due to corrosion. Capacitive sensors, although slightly more expensive, offer greater accuracy and durability. Rosli [2] emphasized the importance of sensor calibration and environmental adaptation in achieving consistent irrigation performance.

CHAPTER 3

COMPONENTS

Hardware Components

The Plant Watering System using ESP32 and Blynk IoT is designed using various hardware components, each playing a vital role in the functionality of the system. The selection of components is based on criteria such as cost-effectiveness, availability, compatibility with ESP32, power efficiency, and ease of interfacing. Below is a detailed explanation of all the components used in the system.

3.1 ESP32 Development Board

Description:

The ESP32 is a powerful microcontroller with built-in Wi-Fi and Bluetooth functionalities. Developed by Espressif Systems, it is widely used in IoT applications due to its robust performance, low cost, and extensive features.

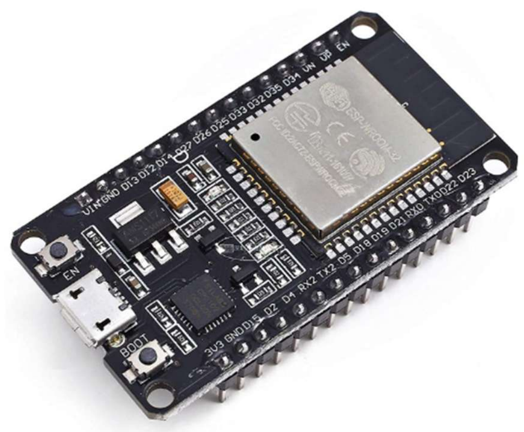

Fig. 3.1: ESP32

Technical Specifications:

- Dual-core 32-bit LX6 microprocessor
- Clock frequency: up to 240 MHz
- 520 KB SRAM
- 4 MB Flash memory
- Integrated 802.11 b/g/n Wi-Fi
- Bluetooth v4.2 (BR/EDR and BLE)
- 34 GPIO pins
- 3.3V logic level
- Onboard voltage regulator

Function in Project:

The ESP32 serves as the central processing unit of the project. It reads the analog signal from the soil moisture sensor, processes the data, and decides when to activate or deactivate the water pump via the relay module. It also handles Wi-Fi communication and sends real-time data to the Blynk IoT platform, allowing users to remotely monitor and control the system via a smartphone.

3.2 Soil Moisture Sensor

Description:

The soil moisture sensor is responsible for measuring the water content in the soil. It typically works on the principle of resistance or capacitance, where the conductivity between two probes changes depending on the soil's moisture level.

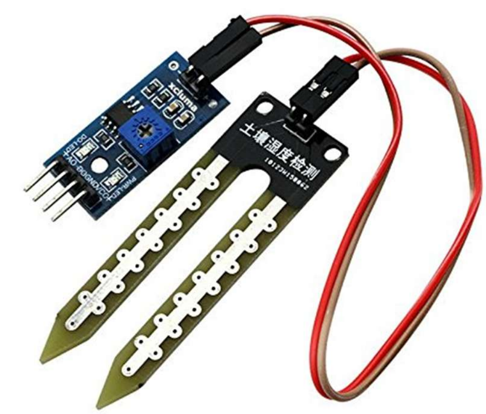

Fig. 3.2: Soil Moisture Sensor

Technical Specifications:

- Working voltage: 3.3V–5V
- Output type: Analog (AO) and Digital (DO)
- Corrosion-resistant probes (in capacitive types)
- Compatible with microcontroller ADCs

Function in Project:

This sensor is inserted into the soil near the plant. It detects the moisture level and provides an analog voltage signal to the ESP32, which is then read using the Analog-to-Digital Converter (ADC). If the soil is dry (below the preset threshold), the ESP32 initiates the watering process. If the soil is wet, no action is taken.

Note: For long-term outdoor use, capacitive soil moisture sensors are preferred over resistive types due to better durability and corrosion resistance.

3.3 Relay Module (1-Channel)

Description:

A relay module is an electrically operated switch that allows low-voltage circuits (like ESP32) to control high-power devices (like pumps or motors). A typical relay module consists of a mechanical relay, a driving transistor, a freewheeling diode, and sometimes an optocoupler for isolation.

Fig. 3.3: Relay Module

Technical Specifications:

- Operating voltage: 5V (can work with 3.3V logic via transistor)
- Switching voltage: up to 250V AC / 30V DC
- Load current: up to 10A
- Optoisolated (optional)

Function in Project:

The ESP32 sends a digital signal to the relay module, which energizes the internal coil to switch the water pump on or off. The relay isolates the low-power circuit from the high-power section, ensuring electrical safety and reliability. The use of a 1-channel relay module is sufficient for a single plant setup.

3.4 Water Pump (DC Mini Submersible)

Description:

A **DC water pump** is used to draw water from a container and deliver it to the plant. A **mini submersible pump** or a **peristaltic pump** is commonly used in small-scale irrigation systems.

Fig. 3.4: Water Pump

Technical Specifications:

- Voltage: 5V–12V DC
- Flow rate: 80–300 L/hr (varies by model)
- Current: 300mA–1A
- Submersible in water
- Lightweight and compact

Function in Project:

The pump is powered ON/OFF through the relay module. It provides water to the plant when the ESP32 detects that the soil moisture level is below the required threshold. The water source can be a water tank or container kept beside the plant.

3.5 Blynk IoT Platform

Description:

Blynk is a popular cloud-based IoT platform that allows users to build mobile applications for controlling and monitoring hardware projects. It supports devices like ESP32, Arduino, NodeMCU, and Raspberry Pi.

Key Features:

- Smartphone control via Android/iOS apps
- Custom widgets: Gauge, Display, Button, Graph
- Cloud server integration
- Virtual pins for data communication
- Real-time notifications and history logs

Function in Project:

Blynk is used to display the soil moisture level, control the water pump manually, and send notifications to the user. The ESP32 communicates with the Blynk cloud server via Wi-Fi and updates the moisture readings in real-time. Users can manually override the auto-watering feature through a button widget in the app.

3.6 Power Supply

Description:

The system requires a power source to operate the ESP32 and the water pump. For simplicity and portability, either a 5V USB charger, battery pack, or regulated power adapter can be used.

Types of Power Sources:

- **USB 5V Power Bank** (for ESP32 only)
- **9V or 12V Adapter** with Buck Converter
- **Lithium-ion Battery with TP4056 Charging Module**

Function in Project:

ESP32 operates at 3.3V logic, but most development boards include a 5V to 3.3V voltage regulator. The water pump, however, may require higher current and voltage (9V–12V), so a separate power supply line (with common ground) is typically used for the pump and relay.

Connecting wires ensure secure and efficient signal transmission between components.

CHAPTER 4
WORKING MECHANISM

4.1 Introduction

The Plant Watering System using ESP32 and Blynk IoT is designed to automate the irrigation process for potted plants, gardens, and indoor plantations. The system aims to eliminate the need for manual watering by automatically detecting the soil's moisture level and supplying water when required. This is achieved by integrating a soil moisture sensor, ESP32 microcontroller, relay module, DC water pump, and the Blynk IoT mobile application. The hardware is soldered onto a zero PCB, making it a durable, compact, and deployable system.

This chapter explains the detailed working mechanism of the system, covering both hardware and software operations, logic implementation, data communication, and interaction with the cloud-based Blynk IoT platform.

4.2 System Overview

The system functions by continuously monitoring the moisture content in the soil through the moisture sensor. The analog data from the sensor is fed into the ESP32 microcontroller, which is programmed using the Arduino IDE. The ESP32 analyzes the moisture level and makes a decision whether to activate or deactivate the water pump through a relay. Simultaneously, real-time data is pushed to the Blynk cloud, allowing the user to monitor and control the system remotely via a smartphone.

4.3 Block Diagram

The basic working principle can be visualized with the following components:

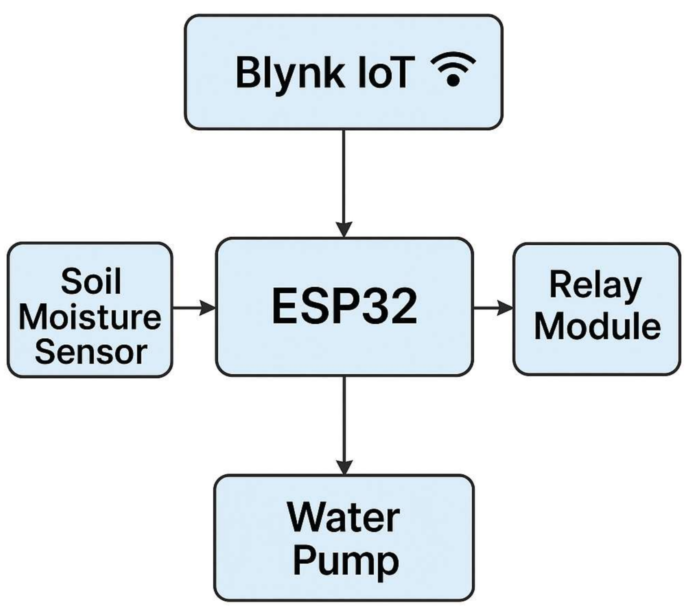

Fig. 4.1: Block Diagram of Plant Watering System

- Sensor reads the soil condition.
- ESP32 processes the value and sends data to Blynk.
- Based on threshold, ESP32 triggers relay.
- Relay activates the water pump.
- Pump irrigates the plant.

4.4 Hardware Functionality

4.4.1 ESP32 Microcontroller

The ESP32 serves as the brain of the system. It performs the following operations:

- Reads analog input from the soil moisture sensor.
- Converts analog values into moisture percentage.
- Compares the moisture value with the user-defined threshold.
- If the soil is dry, it turns ON the relay to activate the water pump.
- It also continuously maintains connection with Wi-Fi and the Blynk cloud server.
- Sends real-time moisture data and pump status to the Blynk app.
- Accepts user commands (like turning the pump on/off manually) from the Blynk app via virtual pins.

Key Functions:

- Analog to Digital Conversion (ADC)
- Digital Output to Relay
- Wi-Fi Communication
- Blynk Cloud Integration

4.4.2 Soil Moisture Sensor

This sensor detects the volumetric water content in the soil. When the soil is moist, conductivity between the sensor's probes is higher, resulting in a lower analog voltage. When the soil is dry, resistance increases, resulting in a higher voltage.

Working Logic:

- Sensor gives analog output (0–1023 in ESP32).
- Value is inversely proportional to moisture.
- Values above a certain threshold (e.g., 800) indicate dryness.

Sample Mapping:

- 0–400 = Wet
- 401–700 = Moist
- 701–1023 = Dry

4.4.3 Relay Module

The **relay** acts as an electronic switch that allows the ESP32 to control high current and voltage devices such as the water pump. When the ESP32 outputs a HIGH signal (or LOW depending on relay type), the relay coil energizes and closes the circuit, turning the pump ON.

Relay Logic:

- Trigger pin connected to digital output of ESP32.
- Relay switches the pump ON when moisture < threshold.
- Turns OFF when soil is sufficiently wet.

4.4.4 DC Water Pump

The **DC submersible water pump** is used to draw water from a tank and deliver it to the plant. It operates at 5V to 12V and is powered through the relay circuit.

Pump Action:

- Activated by relay module.
- Waters the plant through a connected pipe.
- Runs for a programmed duration (e.g., 10–30 seconds) or until sensor detects sufficient moisture.

4.5 Power Supply Mechanism

- ESP32 is powered via USB or a 5V regulated adapter.
- The pump uses a separate 9V–12V source, connected through the relay.
- All grounds are connected to a common ground to ensure proper voltage reference.

4.6 Software and Control Mechanism

4.6.1 Arduino IDE Programming

The entire control logic is written and uploaded to the ESP32 via the **Arduino IDE**. The code includes:

- Wi-Fi setup with SSID and password.
- Blynk authentication token for secure cloud access.
- Pin configuration for the sensor and relay.
- Threshold setup for moisture detection.
- Loop functions to monitor and trigger outputs.
- Virtual pin control to allow app-based interaction.

Code

```
#define BLYNK_TEMPLATE_ID "TMPL3ssIcLB35"
#define BLYNK_TEMPLATE_NAME "esp32222"
#define BLYNK_AUTH_TOKEN "F7TH206mQckf9_UX_92sVSvVpBGyvhKu"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Khushwant";
char pass[] = "7588731861";

#define SOIL_MOISTURE_PIN 34
#define RELAY_PIN 18

void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.println("Starting ESP32 Plant Watering Project...");

  analogSetAttenuation(ADC_11db);  // Extend ADC range

  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH);  // Relay OFF initially

  WiFi.begin(ssid, pass);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\n✅ WiFi Connected");
  Serial.println("IP Address: " + WiFi.localIP().toString());

  Blynk.config(auth);
  if (Blynk.connect(10000)) {
    Serial.println("✅ Blynk Connected");
  } else {
    Serial.println("❌ Blynk Connection Failed");
  }
}

void loop() {
```

```
  Blynk.run();

  int moistureRaw = analogRead(SOIL_MOISTURE_PIN);
  Serial.print("Raw Moisture: ");
  Serial.println(moistureRaw);

  // Map raw ADC value to percentage (0% dry, 100% wet)
  int moisturePercent = map(moistureRaw, 4095, 0, 0, 100);
  moisturePercent = constrain(moisturePercent, 0, 100); // Clamp values between 0-100

  Serial.print("Moisture %: ");
  Serial.println(moisturePercent);

  // Send moisture % to Blynk Gauge widget on V0
  Blynk.virtualWrite(V0, moisturePercent);

  // Watering control threshold (adjust if needed)
  if (moisturePercent < 30) {   // If soil is less than 30% moist → water it
    digitalWrite(RELAY_PIN, LOW);  // Relay ON (pump ON)
    Blynk.virtualWrite(V1, "Watering 🌧");
    Serial.println("💧 Soil dry - Pump ON");
  } else {
    digitalWrite(RELAY_PIN, HIGH); // Relay OFF (pump OFF)
    Blynk.virtualWrite(V1, "Soil OK ✅");
    Serial.println("✅ Soil moisture OK - Pump OFF");
  }

  delay(2000);
}
```

4.6.2 Blynk IoT Integration

The system connects to the Blynk IoT cloud, allowing the following:

- Remote visualization of moisture level (e.g., Gauge or LCD widget).
- Control button to manually turn the pump ON or OFF.
- Notification alerts when soil is too dry.
- History graph for long-term moisture trends.

Communication:

- Uses Blynk virtual pins to send/receive data.
- Every sensor reading is pushed to Blynk virtual pin V0.
- Pump status is reflected on virtual pin V1.
- Manual button (V2) allows user to override automation.

4.7 Steps to Create IoT Dashboard on Blynk

1. Open Blynk IoT Web Console

- Visit: https://blynk.cloud
- Log in with your registered email ID and password.
- If you don't have an account, sign up and verify your email.

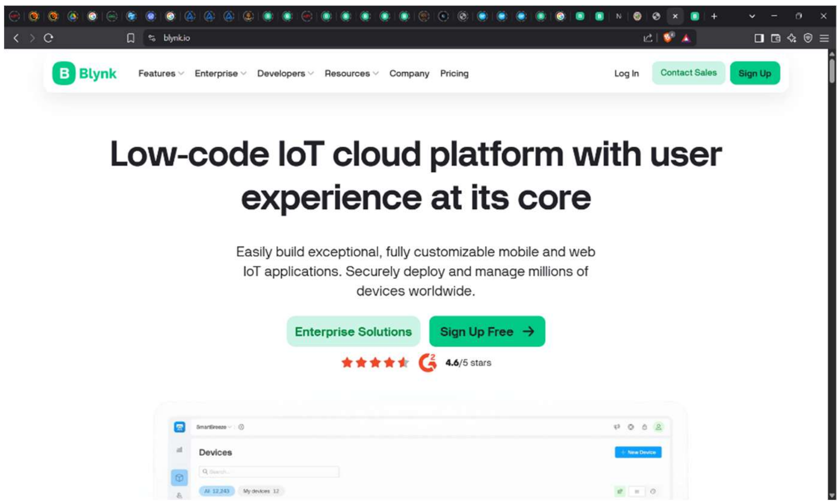

Fig. 4.7.1: Step 1

- Click on **"Templates"** in the left menu.
- Click **"+ New Template"**.
- Fill in the template details:
  - **Template Name**: ESP32 Plant Watering
  - **Hardware**: ESP32
  - **Connection Type**: WiFi
- Click **"Done"**.

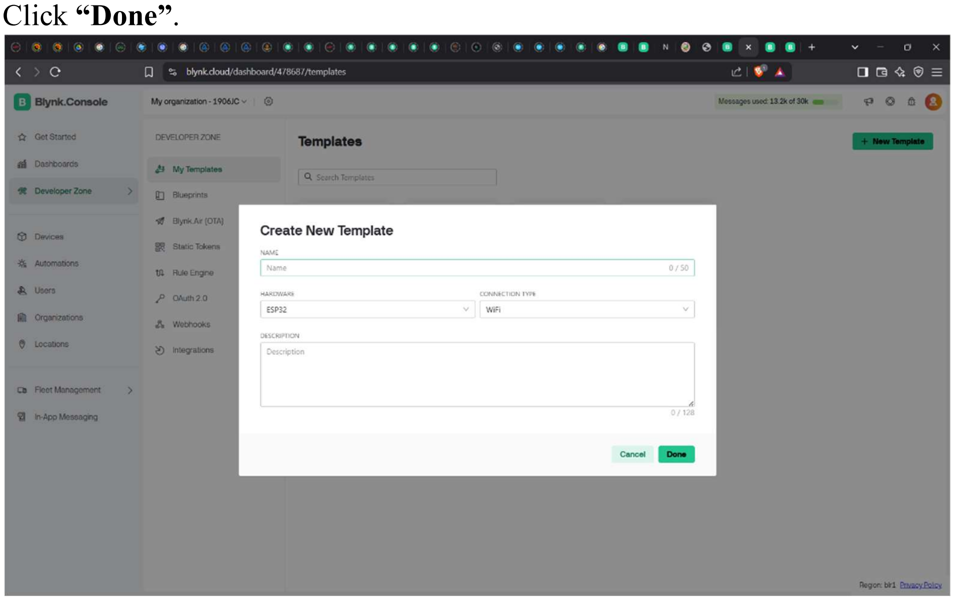

Fig. 4.7.2: Step 2

3. Add Datastreams

- After creating the template, go to the **“Datastreams”** tab.
- Click **“New Datastream”** and select **Virtual Pin**.
- Create the following datastreams:
    - **V0**: Soil Moisture (%)
        - Data type: Integer
        - Min: 0, Max: 100
    - **V1**: Pump Status
        - Data type: String

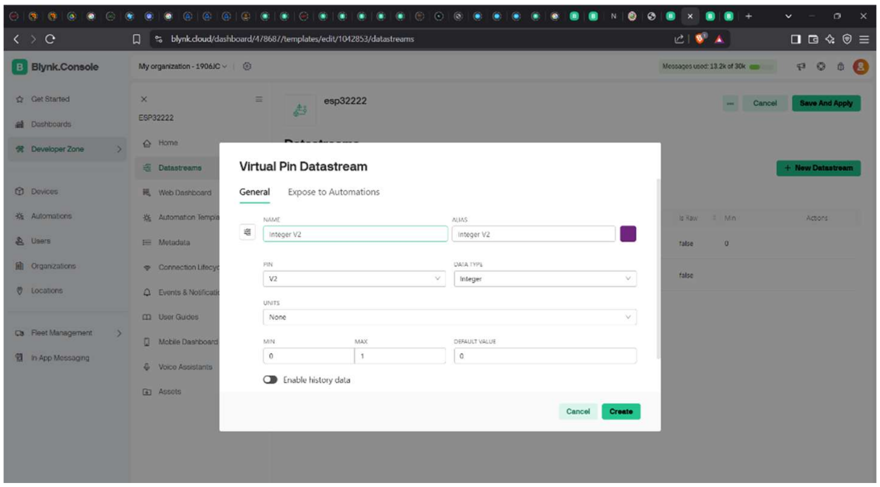

Fig. 4.7.3: Step 3

4. Design Web Dashboard

- Go to the **"Web Dashboard"** tab in your template.
- Drag and drop widgets:
    - **Gauge** for Soil Moisture (Link to V0)
    - **Label** for Pump Status (Link to V1)
- Click **“Save”** when done.

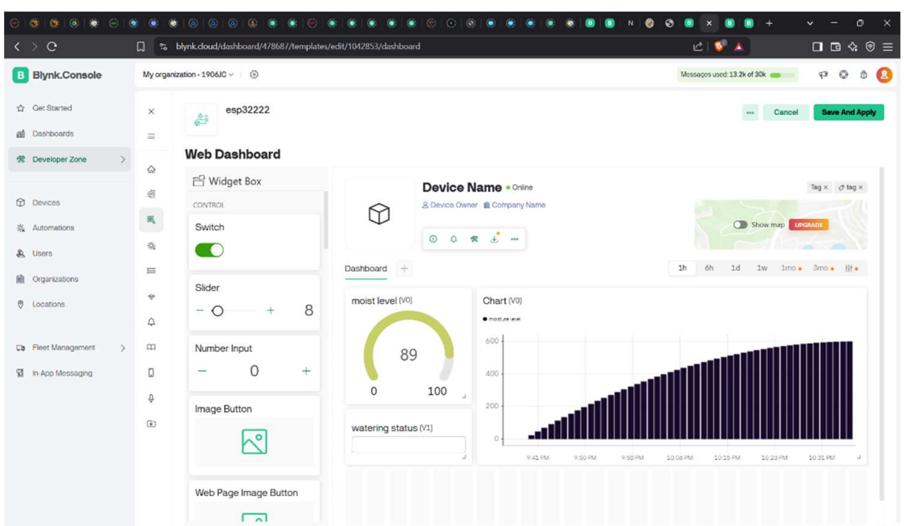

Fig. 4.7.4: Step 4

5. Create a Device

- Go to **"Devices"** in the sidebar.
- Click **“+ New Device” → From Template**.
- Select your template.
- Name your device (e.g., *My Watering System*).
- Click **“Create”**.

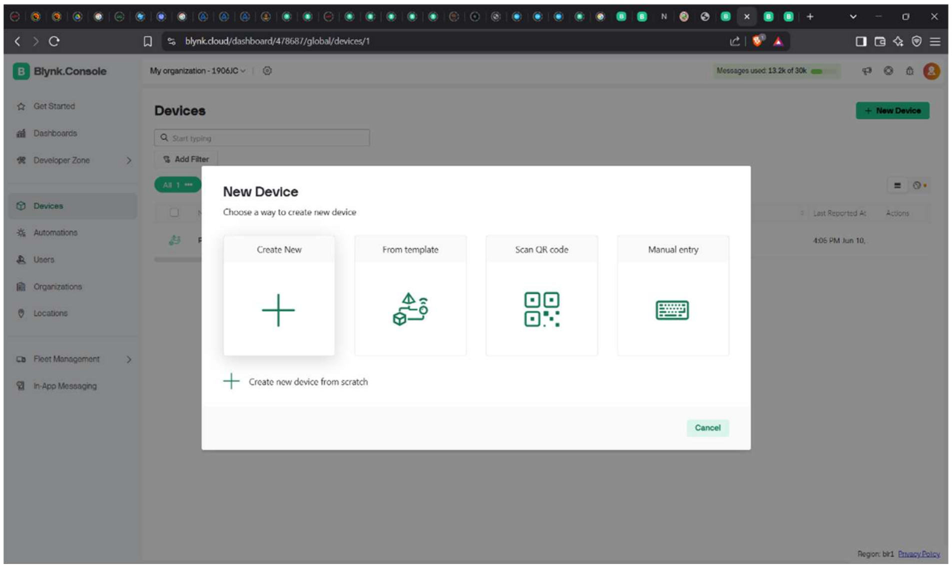

Fig. 4.7.5: Step 5

6. Copy Auth Token

- Open the created device.
- Go to **“Device Info”** tab.
- Copy the **BLYNK_AUTH_TOKEN** for use in your ESP32 code.

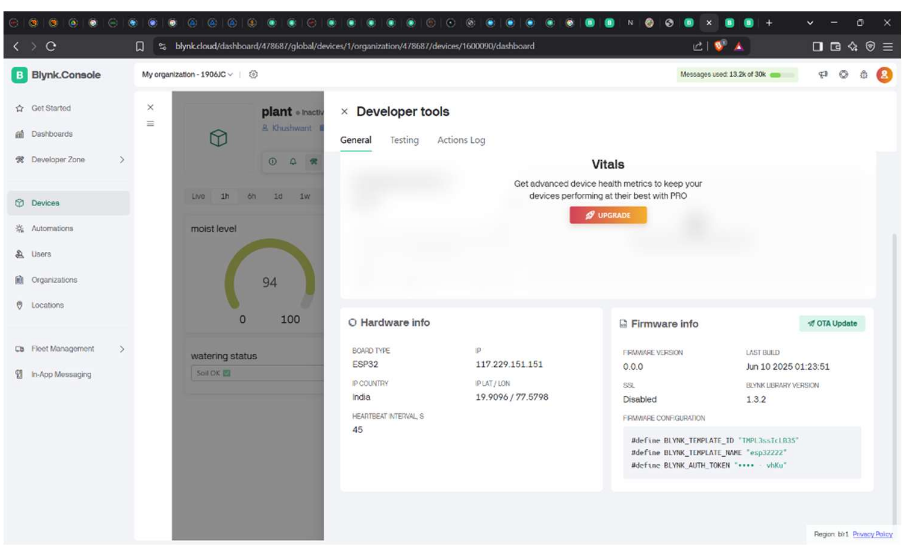

Fig. 4.7.6: Step 6

7. Upload Code to ESP32

- Use Arduino IDE or similar to upload your ESP32 code.
- Paste the BLYNK_AUTH_TOKEN, WiFi SSID, and Password in the code.
- After uploading, ESP32 will start communicating with the dashboard.

1. **Monitor and Control via Web Dashboard**

- Return to **Blynk Web Dashboard**.
- Live values of **soil moisture** and **pump status** will update automatically.

CHAPTER 5
RESULT

1. When the soil moisture is above the 30% then the our pump is off.

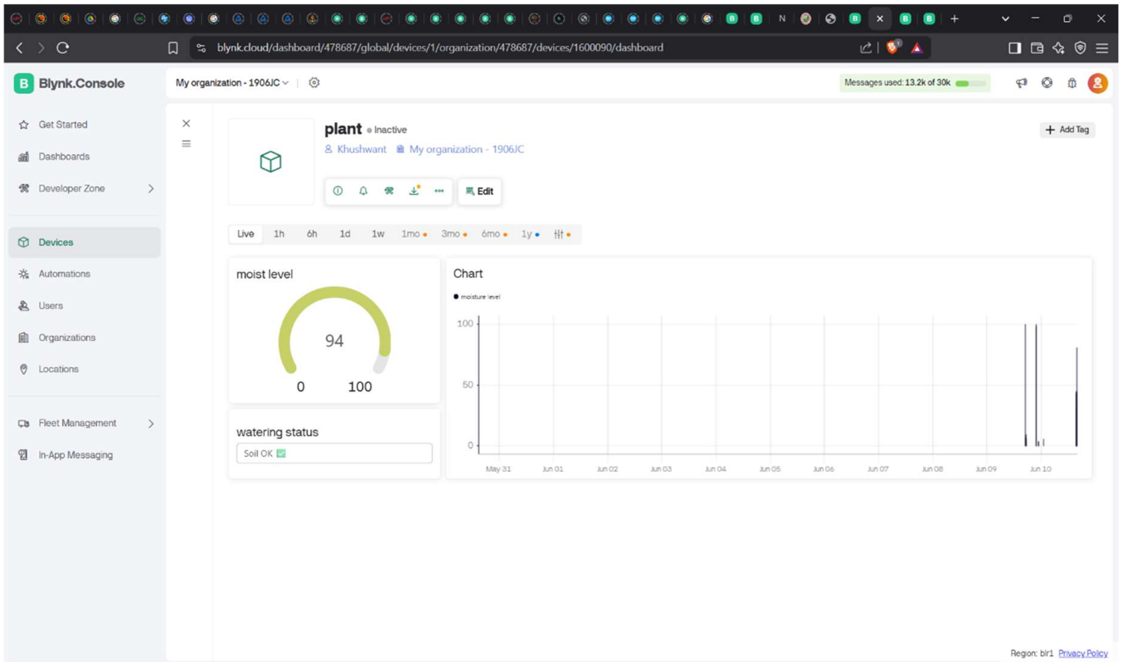

Fig. 5.1: Soil Moisture>30

2. When the soil moisture is below the 30% then the our pump is on.

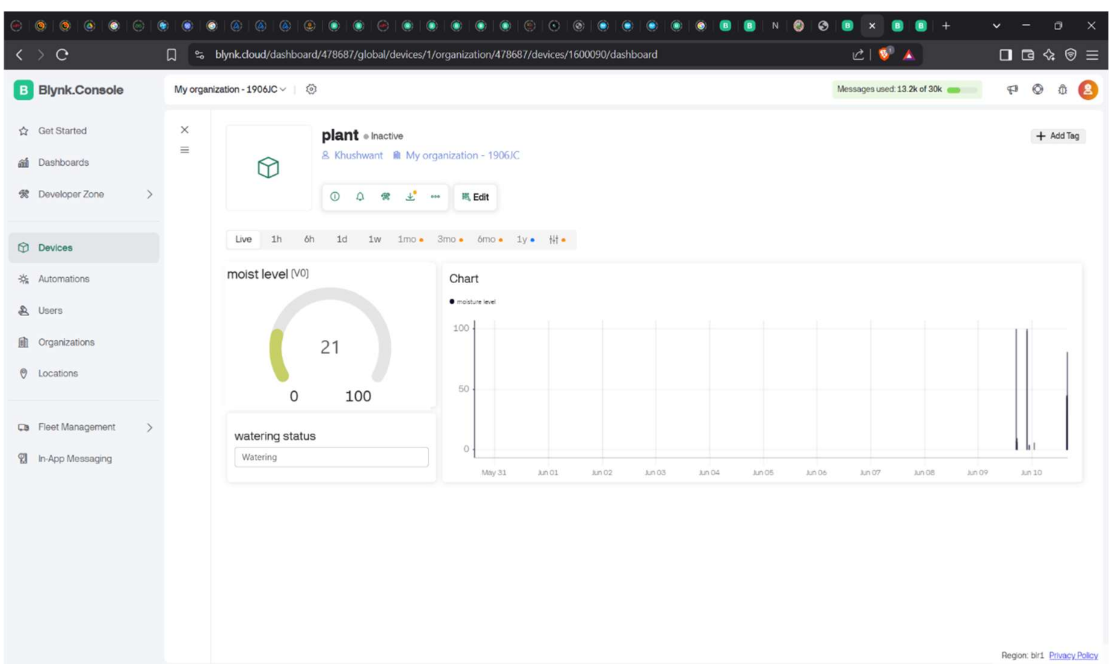

Fig. 5.2: Soil Moisture<30

CHAPTER 6
ADVANTAGES AND LIMITATIONS

1. Advantages

1.1 Automation of Routine Tasks

One of the biggest advantages of this system is the automation of daily watering tasks. Plants require regular and optimal amounts of water for healthy growth. However, many individuals forget or are unable to water plants consistently due to busy schedules. This system ensures that watering occurs only when necessary, removing the burden of routine care from users.

1.2 Water Conservation

Traditional watering methods often result in overwatering, leading to water wastage and possible plant root rot. By using a soil moisture sensor to determine the exact need, the system ensures water is used only when required. This not only conserves water but also promotes healthier plant growth by avoiding water stress.

1.3 Remote Monitoring and Control

With the integration of Blynk IoT, users can monitor real-time soil moisture data and control the water pump from anywhere using a smartphone. This is especially useful for people who travel frequently or are away from home for extended periods. The user can also manually override the system if needed.

2. Limitations

2.1 Soil Sensor Inaccuracy

Most low-cost soil moisture sensors, especially resistive types, degrade over time due to corrosion when left in the soil continuously. This can lead to inaccurate readings. Furthermore, their sensitivity varies across different soil types, requiring repeated calibration.

2.2 Limited Battery Support

While the ESP32 supports low-power modes, the current design requires either USB or external DC power. The absence of a battery backup or solar charging limits the system's application in remote or outdoor settings without reliable electricity.

2.3 Dependence on Wi-Fi

The system's remote features are highly dependent on Wi-Fi connectivity. In rural or isolated areas where internet connectivity is unstable or unavailable, Blynk-based features become unusable. Offline mode functionality is limited unless additional local logic is implemented.

CHAPTER 7
APPLICATIONS

1. Home Gardening

The primary application is in indoor or balcony gardens. Many individuals now grow vegetables and ornamental plants at home, and this system helps maintain them easily. It also acts as a learning tool for children and beginners interested in technology and nature.

2. Agricultural Fields

With some enhancements, this system can be adapted for small agricultural plots. A network of ESP32s and sensors can be installed across a farm to manage irrigation more intelligently, saving water and increasing crop yields.

3. Greenhouses and Nurseries

In controlled environments like greenhouses, where climate and moisture are critical, this system provides accurate monitoring and control. It can be used to automate watering schedules based on soil condition rather than fixed timers.

4. Urban Landscaping

For public parks, hotels, or office buildings, maintaining plant life can be labor-intensive. Integrating this system into the landscape irrigation infrastructure can automate watering, saving manpower and resources.

5. Smart Cities and IoT Projects

The project aligns with smart city initiatives, promoting resource conservation and automation. It can be expanded and integrated with centralized dashboards that manage multiple devices across different city locations.

6. Educational and Research Use

This project is ideal for engineering students, DIY enthusiasts, and researchers. It offers hands-on experience in embedded systems, sensor interfacing, IoT, cloud platforms, and basic automation.

CHAPTER 8
CONCLUSION

The Plant Watering System using ESP32 and Blynk IoT represents an effective and innovative solution for automating irrigation tasks in home gardening and small-scale agriculture. By leveraging the capabilities of the ESP32 microcontroller, a soil moisture sensor, a relay-controlled water pump, and the Blynk IoT platform, this project enables real-time monitoring and control of plant watering based on actual soil conditions.

The primary goal of this system—to ensure timely and adequate watering—has been successfully achieved. It eliminates the dependency on manual irrigation, reduces water wastage, and promotes healthier plant growth. The integration with the Blynk mobile application further enhances the user experience by offering remote access, live data visualization, and manual override controls.

The use of a zero PCB makes the hardware compact and cost-effective, allowing for easier assembly and future scalability. Through this project, we have also gained practical knowledge in the fields of embedded systems, IoT communication, sensor interfacing, and cloud-based data exchange.

While the system has certain limitations, such as reliance on internet connectivity and sensor calibration, these can be addressed in future developments with the inclusion of AI algorithms, solar power support, and environmental sensors.

Overall, this project serves as a strong foundation for smart irrigation systems and contributes meaningfully to sustainable living practices. It demonstrates how low-cost technology can solve real-world problems and support smarter, greener, and more efficient agriculture and gardening.

References

[1] K. Soderby, "Smart Plant Watering with Arduino IoT Cloud," *Arduino Project Hub*, 2021. [Online]. Available: https://projecthub.arduino.cc/ksoderby/smart-plant-watering-with-arduino-iot-cloud-0dff1f

[2] S. N. B. Rosli, "Automatic Plant Watering System with IoT," *Politeknik Sultan Salahuddin Abdul Aziz Shah*, 2020. [Online]. Available: http://repository.psa.edu.my/bitstream/123456789/4286/1/AUTOMATIC%20PLANT%20WATERING%20SYSTEM%20WITH%20IOT%2008DJK20F1002%20SAIYIDAH%20NAFISAH%20BINTI%20ROSLI.pdf

[3] T. Prasanth, M. Siva, and B. Bharath, "IoT Based Smart Plant Monitoring System," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 5, no. 3, pp. 1–5, 2021.

[4] Circuit Digest Team, "Build a Simple Plant Watering System using Arduino," *Circuit Digest*, 2021. [Online]. Available: https://circuitdigest.com/microcontroller-projects/build-a-simple-plant-watering-system-using-arduino

[5] IoT Starters, "IoT Smart Plant Watering System Project," *IoTStarters.com*, 2020. [Online]. Available: https://iotstarters.com/iot-smart-plant-watering-system-project/

[6] Instructables Community, "How to Make IoT Plant Watering System With Web Server," *Instructables*, n.d. [Online]. Available: https://www.instructables.com/How-to-Make-IoT-Plant-Watering-System-With-Web-Ser/

[7] Scribd User, "Mini Project: Plant Watering System," *Scribd*, n.d. [Online]. Available: https://www.scribd.com/document/782514080/mini-project-Plant-Watering-System

[8] SlideShare Contributor, "Automated Plant Watering System," *SlideShare*, n.d. [Online]. Available: https://www.slideshare.net/slideshow/automated-plant-watering-system/61615536

[9] YouTube – ElectroSoftLab. *IoT Based Smart Plant Watering System*. Video: https://www.youtube.com/watch?v=r13VFOAHheA