Report for Industrial Training(EC781)

B. Tech in Electronics & Communication Engineering

B. P. Poddar Institute of Management & Technology

under

Maulana Abul Kalam Azad University of Technology

Submitted by

Name: Shivam Kumar Biswas
University Roll No. :11500321043
RegistrationNo.:211150100310049

Academic Year: 2024 - 2025

Department of Electronics & Communication Engineering
B.P.Poddar Institute of Management & Technology
137, V.I.P Road. Poddar Vihar
Kolkata – 700 052.

Industrial Internship Certificate

ARDENT COMPUTECH PVT. LTD.
High-End Technology Training and Project

N·E·A·T
National Educational Alliance for Technology

Ardent Verified

This certificate is awarded to

SHIVAM KUMAR BISWAS

of

B.P. PODDAR INSTITUTE OF MANAGEMENT AND TECHNOLOGY

for successfully completing the **Industrial Internship** on

INTERNET OF THINGS (IOT)

from

06-07-2024 to 06-08-2024

and implementing the project titled

PLANT WATERING SYSTEM

Certificate ID: ARDENT/112414
Issue Date: 06-08-2024

Microsoft Technology Associate

Director
Technology Services

Director
Operations

Industrial Training Certificate

ARDENT
COMPUTECH PVT. LTD.
High-End Technology Training and Project

Ardent Verified

This certificate is awarded to

SHIVAM KUMAR BISWAS

of

B.P. PODDAR INSTITUTE OF MANAGEMENT AND TECHNOLOGY

for successfully completing the **Industrial Training** on

INTERNET OF THINGS (IOT)

from

06-07-2024 to 06-08-2024

and implementing the project titled

PLANT WATERING SYSTEM

Certificate ID: ARDENT/112458
Issue Date: 06-08-2024

Microsoft Technology Associate | Delivery Partner | AUTODESK | ASD | UDYOG AADHAR MSME | ISO 9001:2015

Director
Technology Services

Director
Operations

ACKNOWLEDGEMENTS

I take this opportunity to express my heartfelt gratitude towards my teacher, **Mr. Shouvik Sarkar**, of IPESP Development Solution, who has guided me throughout the training "**Internship Program on Internet of Things (IoT)**" and in the project titled "**Plant Watering System**." His invaluable assistance in clarifying my doubts and solving my problems has been instrumental in my understanding of the topic.

I am also deeply indebted to my friends who have supported me and lent a helping hand at every stage of this project. Their contributions in researching and understanding the topic have been crucial in preparing this project.

The success of this project is the result of the collective efforts and contributions of many individuals who were involved directly or indirectly. It is my pleasure to extend my sincere thanks to all those who have supported me throughout this journey.

Shivam Kumar Biswas

Shivam Kumar Biswas

Mapping with PO and PSO:

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	P012	PSO1	PSO2
3	3	3	3	3	2	3	3	2	3	2	3	-	3

Justification of mapping:

Program Outcomes (POs):

Engineering knowledge(PO1):
Apply knowledge of engineering fundamentals, mathematics, science and an engineering specialization.

Problem analysis(PO2):
Identify, formulate, review research literature and analyze complex engineering problems.

Design/development of solutions(PO3):
The design solution for complex engineering problems that meet the specific needs with appropriate consideration for the public health and safety.

Conduct investigations of complex problems(PO4):
Use research-based knowledge and research methods to analyze, interpret and synthesis of the information to provide valid conclusion.

Modern tool usage(PO5):
Create, select and apply appropriate techniques, resources and modern engineering and IT tools to predict and model complex engineering activities with an understanding of the limitations.

The engineer and society(PO6): Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

Environment and sustainability(PO7): Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Ethics(PO8):
Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

Individual and team work(PO9):
Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

Communication(PO10):
Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Project management and finance(PO11): Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Life long learning(PO12):
Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSO):

PSO2: The knowledge we will collect and the grab throughout the project that will help us further in higher studies, many competitive exams for higher study and jobs. It will also help us for doing research work in near future.

Technology Stack :

ARDUINO

BLYNK IOT

INTRODUCTION

The rapid advancement of technology, particularly in the field of the Internet of Things (IoT), has opened up new possibilities for automating everyday tasks, making them more efficient and convenient. One such application is the development of smart plant watering systems, which leverage IoT technology to provide precise, automated irrigation for plants. This project focuses on creating a plant watering system that integrates IoT cloud technology to monitor and manage plant health, ensuring optimal growth conditions with minimal human intervention. With the growing interest in smart home technology and sustainable living, the integration of the Internet of Things (IoT) into plant care has emerged as an innovative solution to traditional gardening challenges. This project focuses on developing an automated plant watering system that leverages IoT cloud integration to monitor and manage the irrigation process. By connecting soil moisture sensors and a water pump to a microcontroller, the system can automatically water plants based on real-time data.

Overall, this project demonstrates the potential of IoT in enhancing traditional plant care practices, offering a modern solution that aligns with the growing trend of smart homes and sustainable living. By integrating IoT technology with cloud computing, the system provides a convenient and reliable way to care for plants, reducing the need for manual intervention and promoting healthier, more resilient plant growth.

PROBLEM DEFINITION AND FEASIBILITY STUDY

Problem Definition

Traditional plant care often relies on manual watering, which can lead to either overwatering or underwatering, affecting plant health. For individuals with busy schedules or those who are frequently away, maintaining consistent and optimal plant care can be challenging. Additionally, manual methods are inefficient and may result in unnecessary water waste. There is a need for an automated solution that can monitor and manage plant watering, ensuring plants receive adequate care while conserving water.

Feasibility Study

Technical Feasibility

- Hardware Compatibility: Availability of soil moisture sensors and microcontrollers (Arduino, ESP8266) that can easily interface with each other.
- Cloud Integration: Use of established IoT cloud platform, BLYNK Cloud IoT that support data storage, analysis, and remote control functionalities.
- Connectivity: Reliable internet connection for data transmission between the system and cloud services, ensuring real-time monitoring and control.
- Power Supply: Adequate power source (battery or mains) to keep the system operational, especially for remote or outdoor installations.

Economic Feasibility

- Cost of Components: Affordable sensors, microcontrollers, and cloud service subscriptions; initial investment is reasonable for hobbyists or small-scale applications.
- Maintenance Costs: Low recurring costs, primarily related to electricity and potential cloud service fees; minimal maintenance required.
- Scalability: Potential to scale up the system for larger agricultural projects or multiple plants, balancing initial investment with long-term savings on labor and water usage.

Operational Feasibility

- Ease of Use: User-friendly interfaces for setting up, monitoring, and controlling the system via a web or mobile app, even for users with minimal technical expertise.
- Automation: System can operate autonomously based on preset conditions, reducing the need for constant human intervention.
- Reliability: High reliability in detecting soil moisture levels and triggering the watering process, with fail-safes (like manual overrides) in case of sensor errors.
- Environmental Impact: Positive environmental impact by optimizing water usage and reducing waste, contributing to sustainable practices.

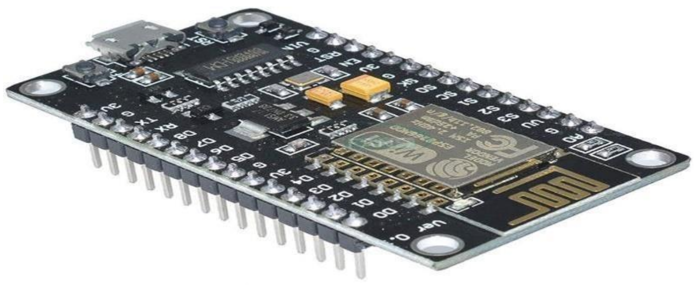
In summary, it is technically, financially, and operationally possible to analyze a concise overview of the feasibility aspects of implementing a plant watering system using IoT and cloud integration.

COMPONENTS:

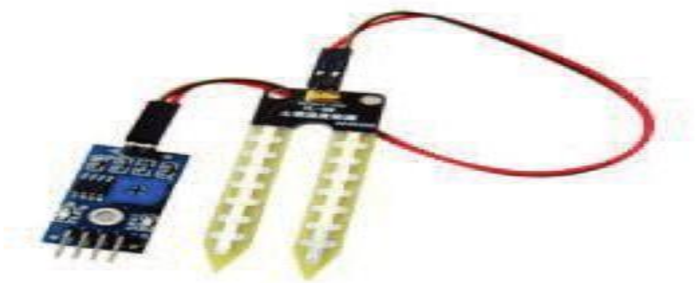
- Esp8266 12e
- Soil moisture sensor
- Motor Driver L298
- Breadboard
- Jumper wires (male to male) (female to male)
- mini water pump
- mini water pipe
- 9v battery clip
- 9v battery

COMPONENTS DETAILS:

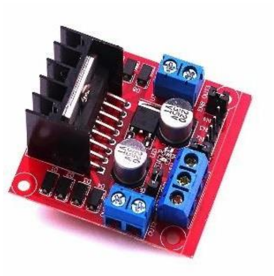
- **Esp8266 12e**: The ESP8266-12E has a built-in microcontroller (Tensilica Xtensa) that can run your program. It has built-in Wi-Fi, allowing the plant watering system to connect to a network for remote monitoring and control. These allow the module to interface with sensors, relays, and other components.



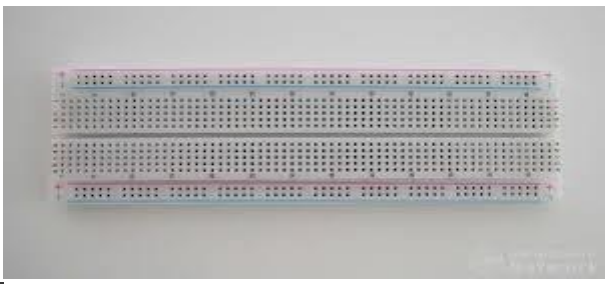
- **Soil moisture sensor** ; Metal or conductive material probes inserted into the soil to measure its moisture level. They detect changes in electrical resistance based on soil moisture content.A microcontroller (e.g., Arduino, ESP8266) processes the sensor's data and controls the watering mechanism. It can be programmed to trigger watering when the soil is dry. Converts the analog signals from the sensor probes into digital data that the microcontroller can process. Provides power to the sensor and microcontroller. This can be a battery, USB power, or a direct connection to a power source. Controlled by the microcontroller, this pump delivers water to the plants when the soil moisture falls below a certain threshold.Acts as a switch, allowing the microcontroller to control high-power devices like the water pump.



- **Motor Driver L298** : The L298 motor driver is a dual H-Bridge IC used to control the water pump in a plant watering system. The driver enables the pump to turn on and off based on sensor readings. It's wired with IN1/IN2 for control signals, OUT1/OUT2 for motor connections, VCC for motor power, and GND for grounding.



- **Breadboard:** A breadboard is a useful tool for prototyping electronic circuits without soldering. In a plant watering system, the breadboard can be used to connect and test various components. These components are interconnected on the breadboard, allowing for easy modification and testing of the plant watering system before creating a more permanent setup.



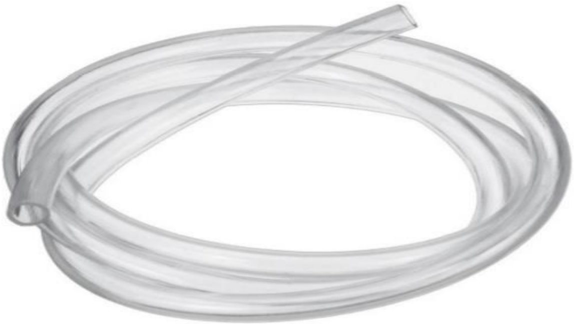
- **Jumper wires (male to male) (female to male):** Jumper wires (male-to-male and female-to-male) are essential components in a plant watering system for connecting various electronic parts such as sensors, microcontrollers, and relays. Here's a brief overview:

1. Male-to-Male Jumper Wires: These are used to connect different pins on a breadboard or between components with female headers. In a plant watering system, they might link a moisture sensor to an input pin on a microcontroller.
2. Female-to-Male Jumper Wires: These are used to connect components with male headers to a breadboard or other male connections. For example, they can connect the output pin of a moisture sensor (which typically has male pins) to a breadboard, which then routes the signal to the microcontroller.

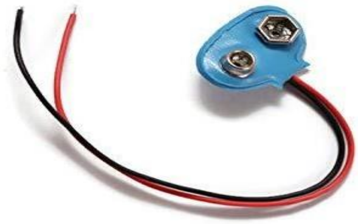
These jumper wires ensure flexible and reliable connections, enabling the plant watering system to function by transmitting signals and power between components like sensors, controllers, and actuators.

- **Mini water pump :** A DC motor mini water pump is a small, electrically powered pump commonly used in plant watering systems for its simplicity and efficiency. The DC motor mini water pump ideal for automated plant watering systems, allowing consistent and controlled watering based on the needs of the plants.

- **Mini water pipe :** A mini water pipe in a plant watering system is typically part of a micro-irrigation setup designed to deliver water directly to the base of plants. This system is efficient and helps conserve water while ensuring plants receive adequate moisture.

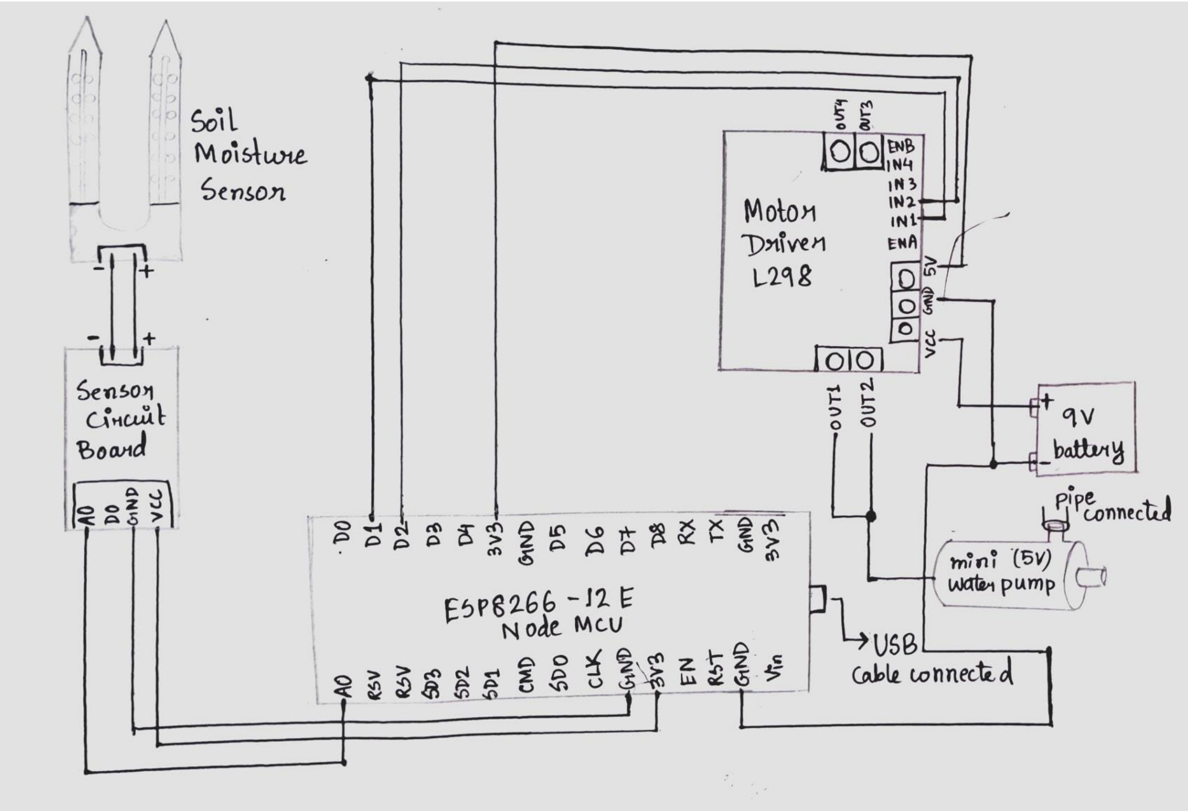


- **<u>9v battery clip:</u>** In a plant watering system, a 9V battery clip is used to connect a 9V battery to the circuit, providing power to the system. Here's how it typically fits into the system:The 9V battery serves as the main power supply for the components of the plant watering system, such as sensors, a microcontroller (like an Arduino), and a water pump.The clip ensures a secure connection between the battery and the circuit. It's usually attached to the battery terminals and then connected to the circuit through wires. If the system requires lower voltage (e.g., 5V for most microcontrollers), a voltage regulator might be used to step down the 9V to the required level.Using a 9V battery with a clip allows the system to be portable, making it ideal for small or temporary setups.Overall, the 9V battery clip is a simple but essential component that ensures the plant watering system has a reliable power source.



- **<u>9v battery</u>** : A 9V battery provides a portable and compact power source, making the plant watering system versatile and easy to install in various locations.

BLOCK DIAGRAM



SOURCE CODE:

```
#define BLYNK_TEMPLATE_ID "TMPL3-WC2HCNt"

#define BLYNK_TEMPLATE_NAME "Plant watering system"

#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>
```

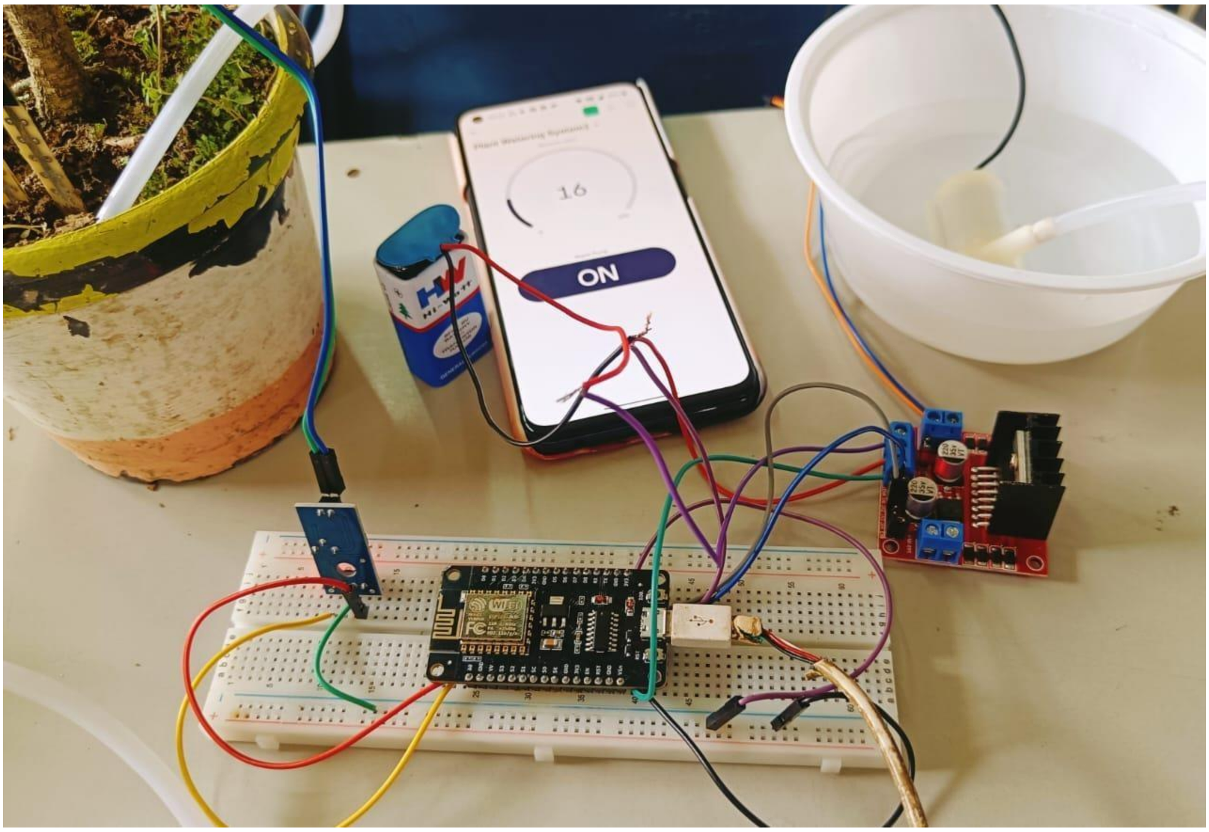
```
// Blynk credentials
char auth[] = "bjeV02fwcClp3UfnL6idGl78Pq_ORwdk"; // Enter your Auth token
char ssid[] = "OnePlus Nord2 5G"; // Enter your WIFI name
char pass[] = "r7nmrn3h"; // Enter your WIFI password

BlynkTimer timer;

// Define component pins
#define sensor A0
#define IN1 D1
#define IN2 D2

void setup() {
  Serial.begin(9600);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(sensor, INPUT);

  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

  Serial.println("System Loading");
```

```
for (int a = 0; a <= 15; a++) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.println("System Loaded");

  // Call the function
  timer.setInterval(5000L, soilMoistureSensor);
}

// Get the button value from Blynk
BLYNK_WRITE(V1) {
  int Relay = param.asInt();

  if (Relay == 1) {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    Serial.println("Pump is ON");
  } else {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    Serial.println("Pump is OFF");
  }
}
```

```
// Get the soil moisture values
void soilMoistureSensor() {
  int value = analogRead(sensor);
  value = map(value, 0, 1024, 0, 100);
  value = (value - 100) * -1;

  Blynk.virtualWrite(V0, value);
  Serial.print("Moisture: ");
  Serial.print(value);
  Serial.println(" ");
 // Threshold value for soil moisture
  int threshold = 40;
 if (value < threshold) {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    Serial.println("Soil is dry, turning on the pump.");
  } else {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    Serial.println("Soil is wet, turning off the pump.");
  }
}
void loop() {
  Blynk.run(); // Run the Blynk library
  timer.run(); // Run the Blynk timer
}
```

PROTOTYPE MODE OF SYSTEM

Circuit diagram



Implementation

Sensor Reading: The soil moisture sensor measures soil moisture levels and sends data to the Arduino or ESP8266.

Data Processing:

- **Arduino + ESP8266**: The Arduino reads the sensor data and sends it to the ESP8266.
- **ESP8266 Only**: The ESP8266 reads the sensor data directly if used standalone.

Control Logic:

- The Arduino or ESP8266 compares the moisture level with a predefined threshold.
- If moisture is below the threshold, the relay activates to turn on the water pump.

Blynk Integration:

- **Data Upload**: Sensor data and pump status are sent to the Blynk server.
- **Control:** Commands from the Blynk app can manually control the pump and display moisture levels.

User Interaction:

- The Blynk app displays real-time moisture readings and pump status.
- Users can control the watering manually or view status updates via the app.

Continuous Operation:

- The system periodically checks soil moisture and adjusts watering based on conditions.
- Blynk updates the app with current data and status.

OUTCOME:

```
Output    Serial Monitor ×
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM6')
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
Moisture: 27
Soil is dry, turning on the pump.
```

FUTURE SCOPES

Advanced Sensors: Integrate additional sensors for monitoring temperature, humidity, and light levels to create a more comprehensive plant care system.

Data Analytics: Utilize machine learning algorithms to analyze historical data and optimize watering schedules based on plant type, weather patterns, and growth stages.

Automated Weather Integration: Connect to weather APIs to adjust watering schedules based on real-time weather forecasts, preventing overwatering during rainy periods.

Solar Power: Implement solar panels to power the system, making it more sustainable and suitable for remote locations.

Smart Notifications: Enhance the Blynk app with smart notifications for maintenance alerts, soil health updates, and personalized care tips.

Integration with Other Smart Devices: Connect the system with other smart home devices, like automated greenhouses or climate control systems, for a fully integrated plant care ecosystem.

CONCLUSION

An IoT plant watering system using Arduino, ESP8266, and Blynk automates plant irrigation by monitoring soil moisture levels. The soil moisture sensor sends data to the Arduino (or directly to the ESP8266). Based on this data, a relay controls the water pump to water the plants when needed. The ESP8266 connects to Wi-Fi and communicates with the Blynk app, allowing users to monitor and control the system remotely. This setup provides automated watering, real-time monitoring, and remote control for efficient plant care.

REFERENCES

- ✓ https://iotstarters.com
- ✓ https://www.ijert.org
- ✓ https://www.instructables.com
- ✓ https://www.researchgate.net