SMART PLANT WATERING SYSTEM

Table of Contents

Introduction

This project is about creating an automatic watering system for plants using a Nodemcu ESP8266 board. The main parts include a soil moisture sensor to check how wet the soil is, a relay module to control a small water pump, an LCD screen to show information, and the Blynk platform to control everything from a smartphone or computer.

To set it up, we have connected the Nodemcu board, soil moisture sensor, relay module, and LCD screen on a breadboard. Then, we have created a dashboard on the Blynk website and app, which will give us information about the soil moisture level and and by Blynk we can control the water pump remotely. This system helps make sure plants get the right amount of water without having to water them manually all the time.

Designed Model

The smart plant watering system for a smart garden comprises several components working together seamlessly to ensure efficient and precise watering of plants. Here's an overview of the key elements:

Components

1.Nodemcu ESP8266 board x 1

2.Soil moisture sensor x 1

3.Relay module x 1

4.LCD display x 1

5.I2C module x 1

6.Breadboard x 1

7.Jumper wires

8.Mini water pump and pipe

9.9v battery clip

10. Battery

Schematic Diagram of Project

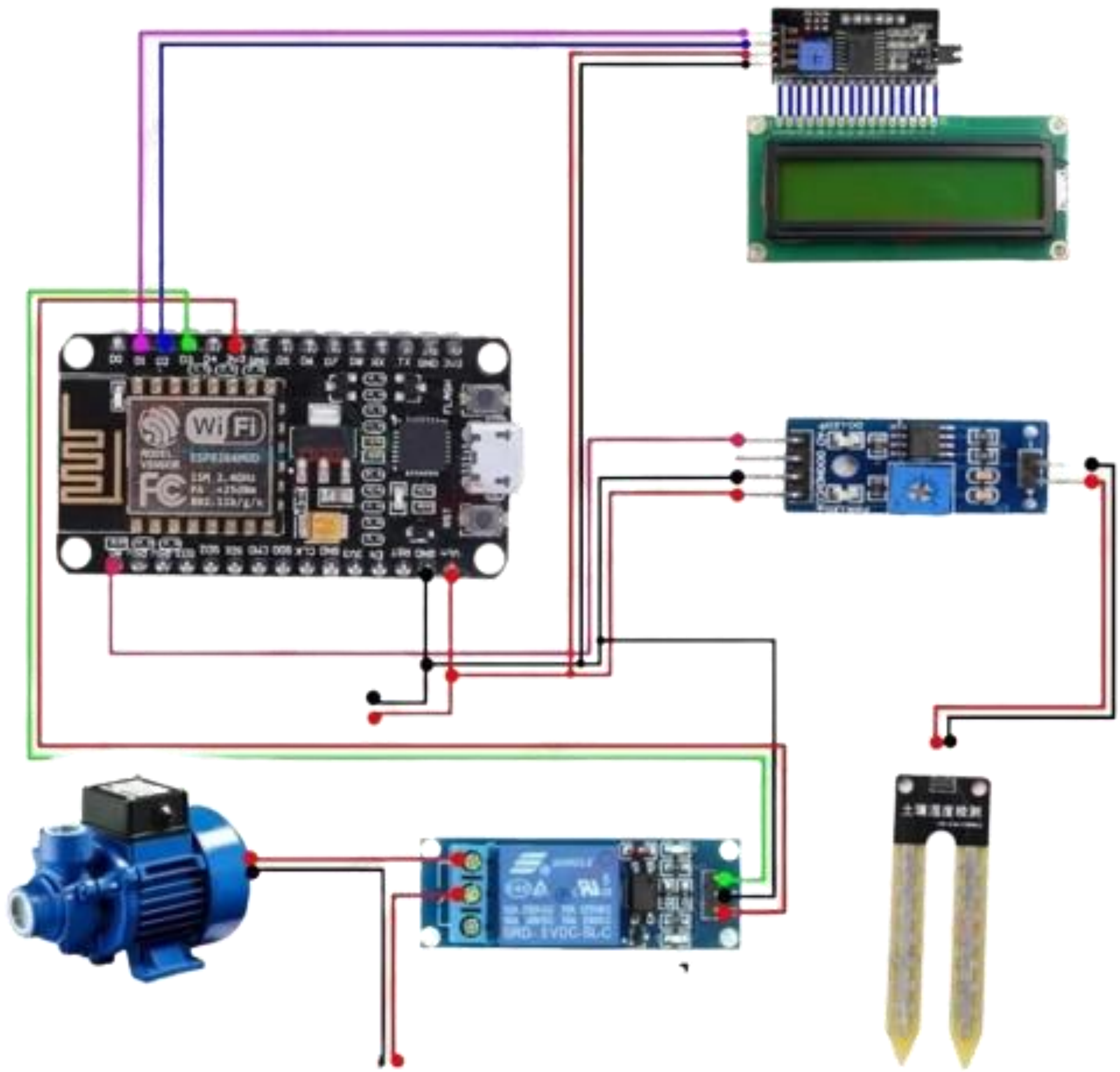

Fig: Smart plant watering system

Description of components along with pictures and Pin Diagrams

1. NodeMCU ESP8266

The NodeMCU ESP8266 board is a low-cost, open-source IoT platform that combines a powerful microcontroller and Wi-Fi connectivity on a single board. At its core is the ESP8266 chip, which features a 32-bit Tensilica L106 microcontroller with a clock speed of 80 MHz, 160 KB of RAM, and 4 MB of flash memory. This board supports the Lua scripting language as well as the Arduino IDE, making it highly versatile and accessible for developers of all skill levels. It includes built-in Wi-Fi, which allows for easy network connectivity, and has several GPIO pins for interfacing with sensors, actuators, and other peripherals. The NodeMCU ESP8266 is widely used in various IoT applications, such as home automation, remote monitoring, and wireless sensor networks, due to its ease of use, flexibility, and robust community support.

Fig: NodeMCU ESP8266

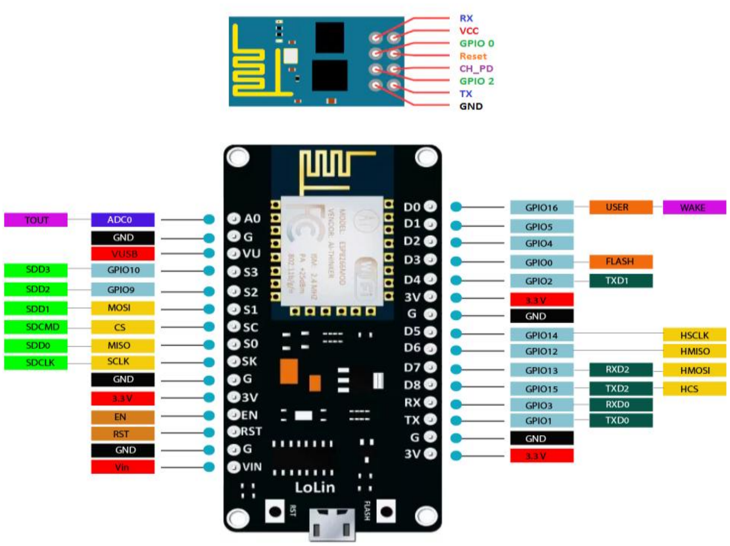

Fig: NodeMCU ESP8266 board Pin Diagrams

2.Soil moisture sensor

Soil moisture is basically the content of water present in the soil. This can be measured using a soil moisture sensor which consists of two conducting probes that act as a probe. It can measure the moisture content in the soil based on the change in resistance between the two conducting plates.

The resistance between the two conducting plates varies in an inverse manner with the amount of moisture present in the soil.

For more information about soil moisture sensor and how to use it, refer the topic Soil Moisture Sensor in the sensors and modules section

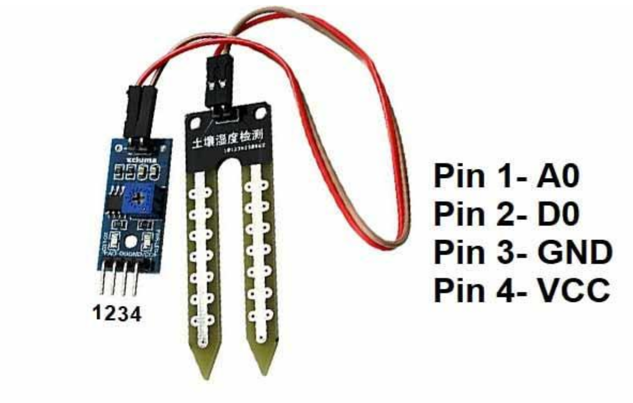

Fig: Soil moisture sensor pictures and Pin Diagrams

3.Relay module

A 5V single-channel relay module is an electronic component designed to control high voltage devices, such as appliances and lights, with a low voltage signal from a microcontroller like an Arduino or Raspberry Pi. It operates using a 5V input signal and includes an optocoupler to isolate the control circuitry from the high voltage circuit, ensuring safety and reliability. The module typically features screw terminals for easy connection to the controlled device and an LED indicator to show the relay status. It is widely used in home automation projects, industrial automation systems, and other applications requiring control of high power devices with a microcontroller.

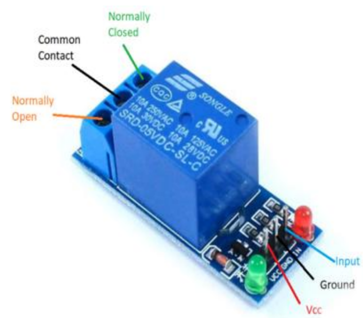

Fig: 5V Single-Channel Relay Module

4.LCD

An LCD (Liquid Crystal Display) is a flat-panel display technology commonly used in a variety of applications such as digital clocks, calculators, and electronic devices. It consists of a matrix of liquid crystals that can be aligned in different ways to block or allow light to pass through, thereby creating images or text. LCD displays are energy-efficient, providing clear and sharp visuals while consuming less power compared to other display technologies like CRTs. They come in various sizes and resolutions, making them suitable for a wide range of applications from small screens on digital watches to large screens on televisions and computer monitors. The technology is highly versatile, supporting both monochrome and color displays, and is known for its thin profile and lightweight characteristics.

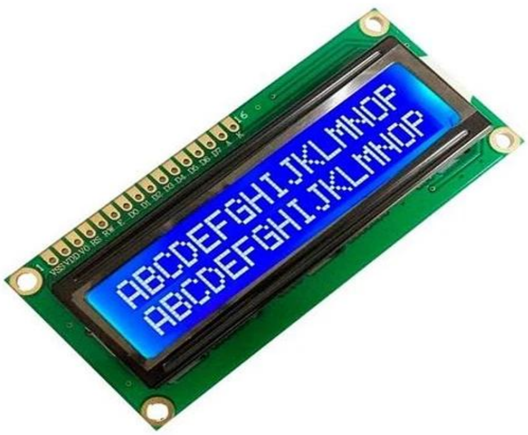

Fig: LCD display Picture

5.I2C module

An I2C (Inter-Integrated Circuit) module is a communication interface that allows multiple microcontrollers and peripheral devices to communicate with each other over a two-wire bus, consisting of a data line (SDA) and a clock line (SCL). The I2C protocol is widely used in embedded systems due to its simplicity and efficiency in handling multiple devices with minimal wiring. Each device on the I2C bus has a unique address, enabling the master device to communicate with specific slave devices. The I2C module is particularly useful for connecting sensors, displays, and other components to microcontrollers, facilitating easy data exchange and synchronization. Its ability to support multiple devices on the same bus makes it ideal for complex projects requiring several peripherals.

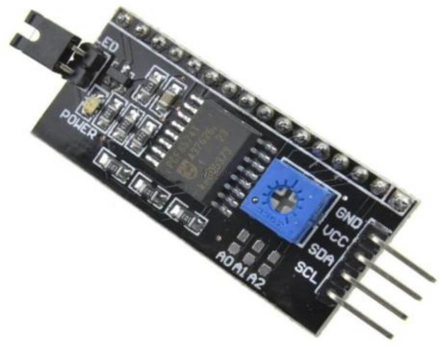

Fig:I2C module Picture

6.Breadboard

A breadboard is a reusable platform used for prototyping and testing electronic circuits without soldering. It consists of a grid of interconnected holes where electronic components, such as resistors, capacitors, integrated circuits, and wires, can be inserted to create temporary circuits. The holes are connected in rows and columns, allowing for easy and flexible connections between components. Breadboards typically have power rails on the sides for distributing power to the components. They are an essential tool for electronics enthusiasts, students, and engineers, enabling quick and efficient circuit design, modification, and troubleshooting.

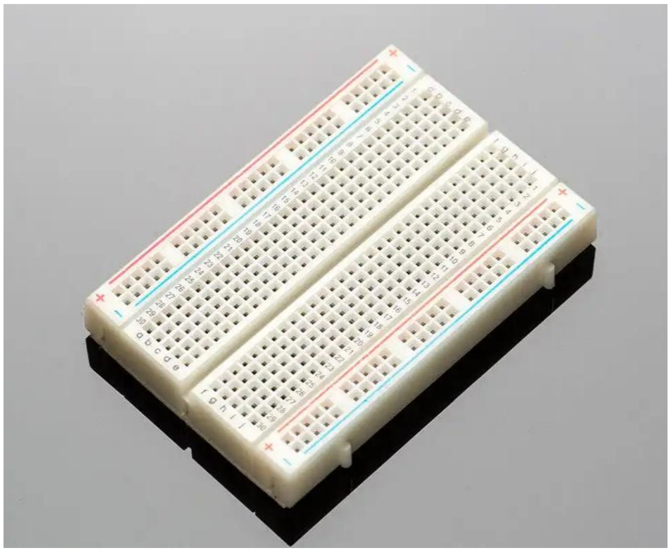

Fig:Breadboard Pictuer

7.Jumper wires

Though jumper wires come in a variety of colors, the colors don’t actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power.

Fig:Jumper wires Picture

8.Mini water pump

water pump that plays the role of trsanfer, lift or pressurize water, fuel, coolant in various water circluation, booster systems. Include small submersible water pump, small solar water pump, etc.

Mini water pumps are widely used for applications that requires use pump with small volume, low power consumption and low price. Such as applications for: Aquarium, fish tank, cat water fountain, solar water fountain, water cooling system, water booster, water heater, water circulation system, car wash, agriculture, medical industries and home appliance etc.

Fig:Mini water pump

10. Battery

A 9V battery is a small, rectangular battery commonly used in various electronic devices and projects. It typically features a compact design with two snap connectors on top, making it easy to connect to electronic circuits and components. The 9V battery provides a nominal voltage of 9 volts, making it suitable for powering a wide range of devices, including smoke detectors, portable radios, and small electronic projects. Its compact size and relatively high voltage make it a popular choice for portable and low-power applications, offering convenience and ease of use in powering electronic devices.

Fig:Battery

Methodology

1.Gather Components

We have collected all the necessary components for our project, including aa soil moisture sensor, a water pump, a Wi-Fi module (e.g., ESP8266), jumper wires, and an LCD display.

2.Setup Hardware

We have placed the Nodemcu board on the breadboard and connect the VIN and GND pins to the breadboard.
After that we have placed the soil moisture sensor on the breadboard and connect it to the Nodemcu board.
And then, we have connected the LCD display and relay module to the Nodemcu board.
And then relay with the board and the water pump.

For the setup process we have follow the circuit diagram below.

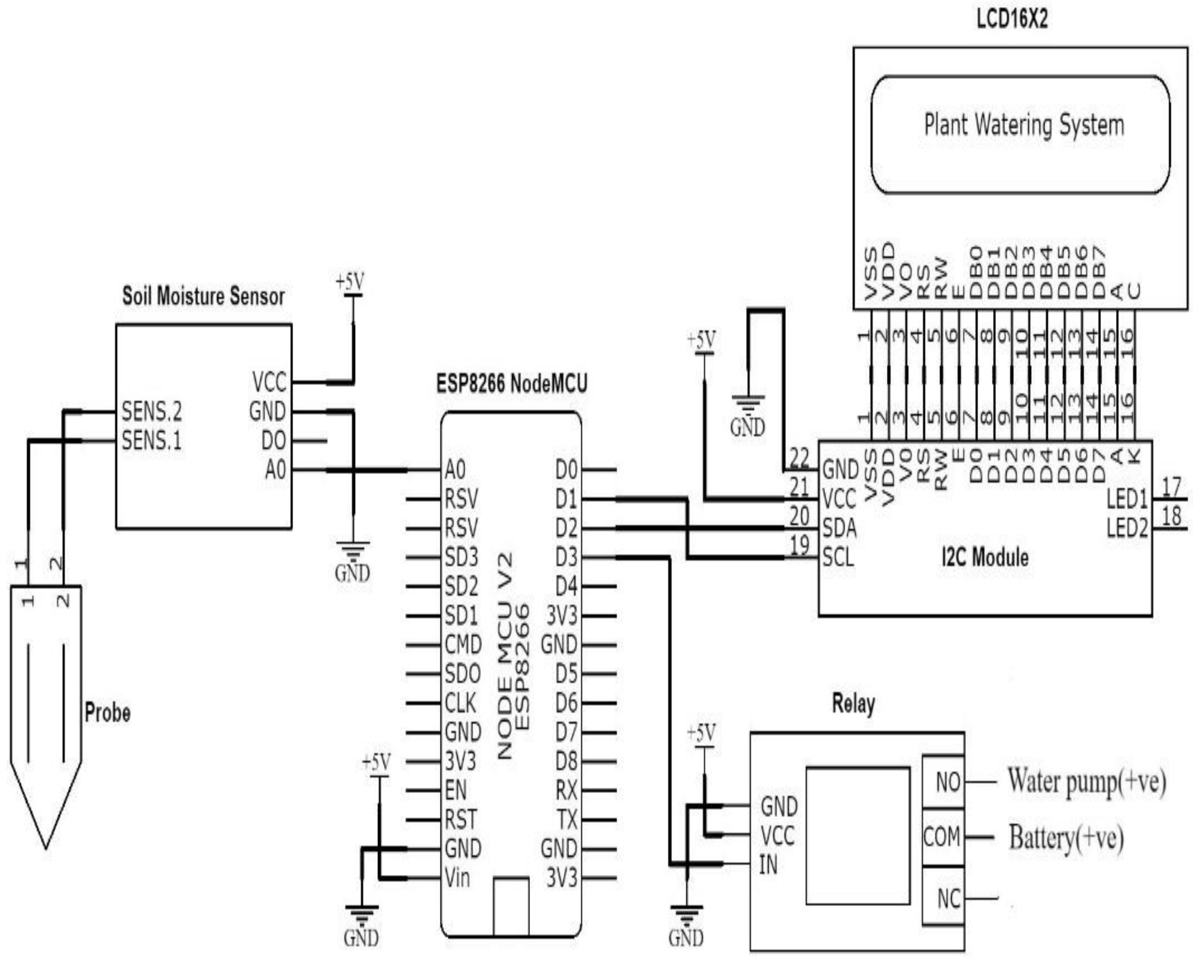

Fig: Circuit Diagram

3. Blynk Setup

Web dashboard

Let's create the Blynk web dashboard for our project. First, we need to go to the Blynk website and create a new account using our Gmail address. Once signed in, click the new Template button. Enter our project name as desired and click the done button. Next, click the "Datastreams" tab and create two data streams using the information below:

- Virtual PIN –> Name – Moisture value / PIN — V0 / MIN — 0 / MAX — 100

- Virtual PIN –> Name – Water pump / PIN — V1 / MIN — 0 / MAX — 1

Afterward, click the "Web dashboard" tab and include a button and a Gauge widget to the dashboard. Arrange these widgets as desired.

Next, click the settings buttons on these widgets one by one and select the data streams we created earlier. Then, click the save button. Now, click the search icon button and create a "New device," selecting the template we created earlier.

Upload Code

With the project connected to the computer, let's upload the following program to the Nodemcu board:

```
#include <LiquidCrystal_I2C.h>

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>


LiquidCrystal_I2C lcd(0x27, 16, 2);


char auth[] = "Enter your Auth token";
```

```
char ssid[] = "Enter your WIFI name";
char pass[] = "Enter your WIFI password";

BlynkTimer timer;
bool Relay = 0;

#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);

  // Display Title during initialization
  char text[] = "AIPHA";//Title
  for (int i = 0; i < 5; i++) {
    lcd.setCursor(i, 0);
    lcd.print(text[i]);
    delay(500);
```

```
  }
  lcd.clear();

  // Display loading animation
  lcd.setCursor(1, 0);
  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;

  Blynk.virtualWrite(V0, value);
  lcd.setCursor(0, 0);
  lcd.print("Moisture :");
  lcd.print(value);
  lcd.print(" ");

  // Check moisture level and control motor accordingly
  if (value < 10 || Relay == 1) {
    digitalWrite(waterPump, HIGH); // Turn on the motor
    lcd.setCursor(0, 1);
    lcd.print("Motor is ON ");
```

```
  } else if (value >= 30) {
    digitalWrite(waterPump, LOW); // Turn off the motor
    lcd.setCursor(0, 1);
    lcd.print("Motor is OFF");
  }
}

void loop() {
  Blynk.run(); // Run the Blynk library
  timer.run(); // Run the Blynk timer
}
```

Next, we enter our WIFI SSID and password, select the board and port, and finally, upload this code to the Nodemcu board.

Blynk Mobile Dashboard

Now, let's create the Blynk mobile dashboard. First, we download and install the Blynk app on our phone. Then, we sign in to our account and click the template we created in the Blynk web dashboard.

Next, we add widgets to the dashboard by clicking the + icon at the top right corner and adding one button and one gauge widget. After arranging these widgets as desired, we click each widget to select the data streams created in the Blynk web dashboard.

Finally, we connect the water pump to the relay module using the circuit diagram. We then insert the soil moisture sensor into the soil. For powering the water pump, we use a 9V battery.

Testing the Setup

We power on the NodeMCU ESP8266 board and monitor the Serial Monitor for any debugging messages. We verify that the LCD display shows the "AIPHA" text during initialization and then displays the "---------" animation. We ensure that the soil moisture level and motor status are displayed correctly on the LCD.

Testing Blynk Control

We open the Blynk app on our phones and connect to our project. We test the manual control of the water pump using the button widget. We ensure that pressing the button turns the motor on/off as expected, with the corresponding status displayed on the LCD.

Testing Automatic Control

We test the automatic control of the water pump based on the soil moisture level. We adjust the moisture sensor to simulate different moisture levels and observe how the motor responds accordingly. We ensure that the motor turns on when the soil moisture level is below the threshold and turns off when it is above the threshold.

4.Optimization

We fine-tune the threshold values in the code to optimize the watering process. We adjust the moisture level thresholds (e.g., lower threshold for turning on the pump, upper threshold for turning off the pump) based on specific plant watering requirements. This step may involve some trial and error to determine the optimal settings for maintaining soil moisture.

5.Deployment

Once the system is working as expected, we deploy the automatic watering system in our plant pots or garden. We ensure the soil moisture sensor is placed correctly and the water pump is positioned to effectively water the plants. We monitor the system over a period of time and make any necessary adjustments to ensure optimal plant health and efficient water usage.

6.Documentation

We document our project setup, including wiring diagrams, code explanations, and any modifications made during testing and optimization. This documentation will be valuable for future reference and for sharing the project with others who may want to replicate or build upon our work.

7.Inputs

- Soil Moisture Level: The project takes input from a soil moisture sensor placed in the soil of the plant pot or garden bed. This sensor measures the moisture level of the soil.

- Manual Control: The Blynk app provides a button widget that allows manual control of the water pump. Users can press the button to turn the pump on or off manually.

8.Outputs

- LCD Display: An LCD display connected to the Arduino provides visual feedback to the user. It displays:

  - The current moisture level of the soil.

  - The status of the water pump (whether it's on or off).

- Water Pump: The project controls a water pump connected to the Arduino. The pump delivers water to the plant when needed.

Working Procedure

To run the automatic plant watering system, follow these steps:

1. Set Up Hardware: Connect the NodeMCU ESP8266 board to the soil moisture sensor, relay module, and LCD display. Place the soil moisture sensor into the soil and connect the water pump to the relay module. Power the system with a 9V battery.

2. Configure Blynk:

   - Create a new project on the Blynk app and add a button and gauge widget.

   - Assign virtual pins V0 for the moisture value and V1 for the water pump control.

   - Save the project configuration.

3. Upload Code:

- Open the Arduino IDE, input the provided code, and replace the placeholders with your Blynk authentication token, WiFi SSID, and password.

- Select the correct board and port, then upload the code to the NodeMCU ESP8266 board.

4. Start the System:

- Power on the NodeMCU board. The LCD will first display "AIPHA" during initialization, followed by a loading animation.

- After initialization, the LCD will display the current soil moisture level and the motor status.

5. Using Blynk:

- Open the Blynk app on your smartphone and connect to the project.

- Use the button widget to manually control the water pump. The gauge widget will display the soil moisture level.

6. Automatic Watering:

- The system will automatically turn the water pump on when the soil moisture level falls below 10% and turn it off when the moisture level reaches 30%. This ensures the soil maintains optimal moisture levels for plant health.

This procedure allows you to effectively manage the watering of your plants both automatically based on soil moisture and manually via the Blynk app.

Result Discussion

Initialization

Upon power-up, the LCD display shows the text "AIPHA" followed by a "System Loading" animation to indicate that the system is initializing.

Expected Outcomes:

Efficient Water Use: Reduced water consumption by targeting only the areas and times where irrigation is needed.

Enhanced Plant Health: Consistent and optimal watering promotes better growth and health of plants.

User Convenience: Minimal manual intervention required, with the option for remote monitoring and control.
Data Insights: Detailed logs and data provide insights for further optimization and understanding of plant and soil conditions.

Automatic Mode

The project continuously monitors the soil moisture level using the soil moisture sensor.

If the moisture level falls below a predefined threshold (e.g., 20%), indicating that the soil is dry, the water pump is automatically turned on to irrigate the plant.

The LCD display updates to indicate that the pump is running.

-Once the moisture level rises above another predefined threshold (e.g., 30%), indicating that the soil is adequately moistened, the water pump is automatically turned off.

Manual Mode

-Users can open the Blynk app and use the button widget to manually control the water pump.

- Pressing the button turns the pump on, and the LCD display updates to indicate that the pump is running.

- Pressing the button again turns the pump off, and the LCD display updates to indicate that the pump is off.

Water Distribution:

Controlled delivery of water to plants based on their specific needs.

Optimized water usage to prevent wastage and promote plant health.

By following this methodology, the automated watering system will operate effectively, ensuring plants receive the right amount of water at the right time, leading to healthier growth and resource conservation.

Conclusion

The automatic watering system works well to keep plants properly watered by using the soil moisture sensor to check if the soil is too dry. If it is, the water pump turns on automatically. The LCD screen shows the soil moisture level and whether the water pump is on or off.

Using the Blynk platform, you can also see this information and control the water pump from your phone or computer. You can even turn the water pump on or off manually if you need to.

Tests showed that the system is reliable. It correctly detects when the soil is dry and turns the pump on, then turns it off when the soil is wet enough. The LCD screen and Blynk app both show the correct information.

This project makes watering plants easier and saves water by only turning the pump on when needed. It’s great for gardeners, farmers, and anyone who wants an easy way to keep their plants healthy. This system shows how useful smart technology can be in taking care of plants and using water efficiently.

References

1. Blynk Documentation: The official Blynk documentation provides detailed information on setting up and using Blynk for IoT projects.
   - Website: https://docs.blynk.io

2. Random Nerd Tutorials: Comprehensive guides on using the ESP8266 NodeMCU, including setup, programming, and various projects.
   - Website: [https://randomnerdtutorials.com/esp8266-nodemcu]

3. Arduino LiquidCrystal_I2C Library: Documentation for the LiquidCrystal_I2C library used to control I2C LCD displays with Arduino.
   - Website: [https://www.arduino.cc/reference/en/libraries/liquidcrystal_i2c

4. SparkFun Soil Moisture Sensor Guide: Information on soil moisture sensors, including how they work and how to interface them with microcontrollers.
   - Website: [https://learn.sparkfun.com/tutorials/soil-moisture-sensor-hookup-guide]

5. How to Use a Relay Module with Arduino: Detailed guide on using relay modules with microcontrollers for controlling high voltage devices.

- Website: [https://www.circuitbasics.com/how-to-use-a-relay-module-with-arduino]

6. IoT Based Smart Plant Monitoring System: Example of a similar project that uses IoT for automated plant watering.
   - Website: https://www.hackster.io/harshju/iot-based-smart-plant-watering-410bd8

These references provide valuable information and support for the various aspects of our project, from hardware setup to software programming and IoT integration.