

B.Tech. CSE (IoT & IS)

III Year (5th Sem)

Project Based Learning – III

IS3170

Project Report

On[[1]](#footnote-1)

**Plant Watering System using NodeMCU**

Group Members:

1. Aayush Bindal (229311202)

2. Siddharth (229311265)

3. Atharva Deshmukh (229311049)

|  |  |
| --- | --- |
| Submitted To  Dr. Hemlata Goyal | Submitted By:  Name:  Registration No.: |

**Index**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Page No.** |
| 1 | Objective | 3 |
| 2 | Problem Statement | 3 |
| 3 | Microcontroller Structure | 4 |
| 4 | Components Used | 5 |
| 5 | Circuit Diagram | 7 |
| 6 | Code | 8 |
| 7 | Result | 11 |

**Objective**

To develop an intelligent and automated irrigation system using NodeMCU ESP8266 that optimizes water usage and ensures plants receive adequate hydration with minimal human intervention.

**Problem Statement**

In today's busy world, people often neglect plant care due to a lack of time, resulting in improper watering. Overwatering can suffocate roots and waste water, while underwatering can cause plants to wilt or die. Traditional irrigation systems are inefficient, failing to account for specific plant hydration needs and leading to significant water wastage. Moreover, these systems lack remote monitoring and control capabilities, leaving plants vulnerable when users are away. Water scarcity further exacerbates the need for sustainable and efficient irrigation solutions. An automated, IoT-based system is required to address these issues by providing precise, data-driven watering, reducing human error, conserving water, and allowing remote access and control to ensure optimal plant health and resource management.

**Microcontroller Structure**

The NodeMCU ESP8266 microcontroller serves as the core component of the Plant Watering System, playing a critical role in managing the entire operation. It is specifically designed for IoT applications, featuring robust processing power and integrated Wi-Fi capabilities. The NodeMCU performs the following key functions:

1. **Data Collection:** The microcontroller collects real-time data from the soil moisture sensor, which measures the moisture level of the soil. This data is processed to determine whether the soil requires watering.
2. **Decision-Making:** Based on the predefined moisture thresholds programmed into the NodeMCU, it decides whether to activate the water pump. This logic ensures plants receive water only when necessary, optimizing hydration and conserving resources.
3. **Actuator Control:** The NodeMCU controls the relay module, which serves as a switch for the water pump. When the soil moisture level falls below the threshold, the NodeMCU sends a signal to the relay, turning the pump on. Once the soil is adequately hydrated, the pump is turned off.
4. **IoT Connectivity:** One of the standout features of the NodeMCU is its built-in Wi-Fi module, which facilitates connectivity to the internet. This allows the system to upload soil moisture data to cloud platforms and enable remote control through a mobile app or web dashboard.
5. **Power Efficiency:** The NodeMCU is designed to operate with low power consumption, making it suitable for long-term applications. It can be powered via a USB adapter or battery pack, ensuring flexibility in deployment.

**Features of NodeMCU ESP8266:**

* **Wi-Fi Capability:** The integrated Wi-Fi module simplifies IoT integration, enabling seamless data transmission and control.
* **Compact Design:** Its small form factor makes it easy to incorporate into compact systems.
* **Programming Flexibility:** Programmable via the Arduino IDE or Lua scripting language, the NodeMCU supports rapid development and prototyping.
* **Cost-Effectiveness:** As an affordable microcontroller, the NodeMCU is accessible for hobbyists and developers alike.
* **GPIO Pins:** The General-Purpose Input/Output (GPIO) pins allow the connection of various sensors and actuators, making it versatile for multiple applications.

In summary, the NodeMCU ESP8266 is a powerful yet cost-effective microcontroller that forms the backbone of the Plant Watering System. Its ability to collect data, make decisions, control actuators, and provide IoT connectivity ensures a reliable and efficient operation.

**Components Used**

**1. NodeMCU ESP8266**

The NodeMCU ESP8266 is a Wi-Fi-enabled microcontroller that is widely used in IoT projects.

Key Features:

Wi-Fi Module (ESP8266): Enables communication with the internet or local network.

GPIO Pins: Provides multiple General Purpose Input/Output (GPIO) pins for connecting sensors and actuators.

Programming: Can be programmed using the Arduino IDE, Lua, or MicroPython.

Cost: Affordable and easily available, making it a popular choice for hobbyists.

Use in Watering Systems:

Acts as the main controller, reading data from sensors and sending commands to actuators.

Interfaces with cloud platforms like Blynk for remote monitoring and control.

**2. Soil Moisture Sensor**

This sensor measures the water content in the soil, crucial for determining when plants need watering.

Key Features:

Probes: Detect the moisture level by measuring the resistance or capacitance of the soil.

Output: Provides analog or digital signals based on moisture levels.

Range: Can detect moisture from dry to water-saturated soil.

Power: Low power consumption.

Use in Watering Systems:

Sends real-time soil moisture data to the NodeMCU.

Triggers the watering mechanism when the moisture level falls below a predefined threshold.

**3. Relay Module**

A relay module is an electrically operated switch that allows low-power microcontrollers to control high-power devices.

Key Features:

Channels: Commonly available in single or multi-channel versions.

Voltage Control: Allows a 3.3V or 5V microcontroller signal to control devices like water pumps running on higher voltages (e.g., 12V or 230V).

Isolation: Offers optocoupler isolation for safe operation.

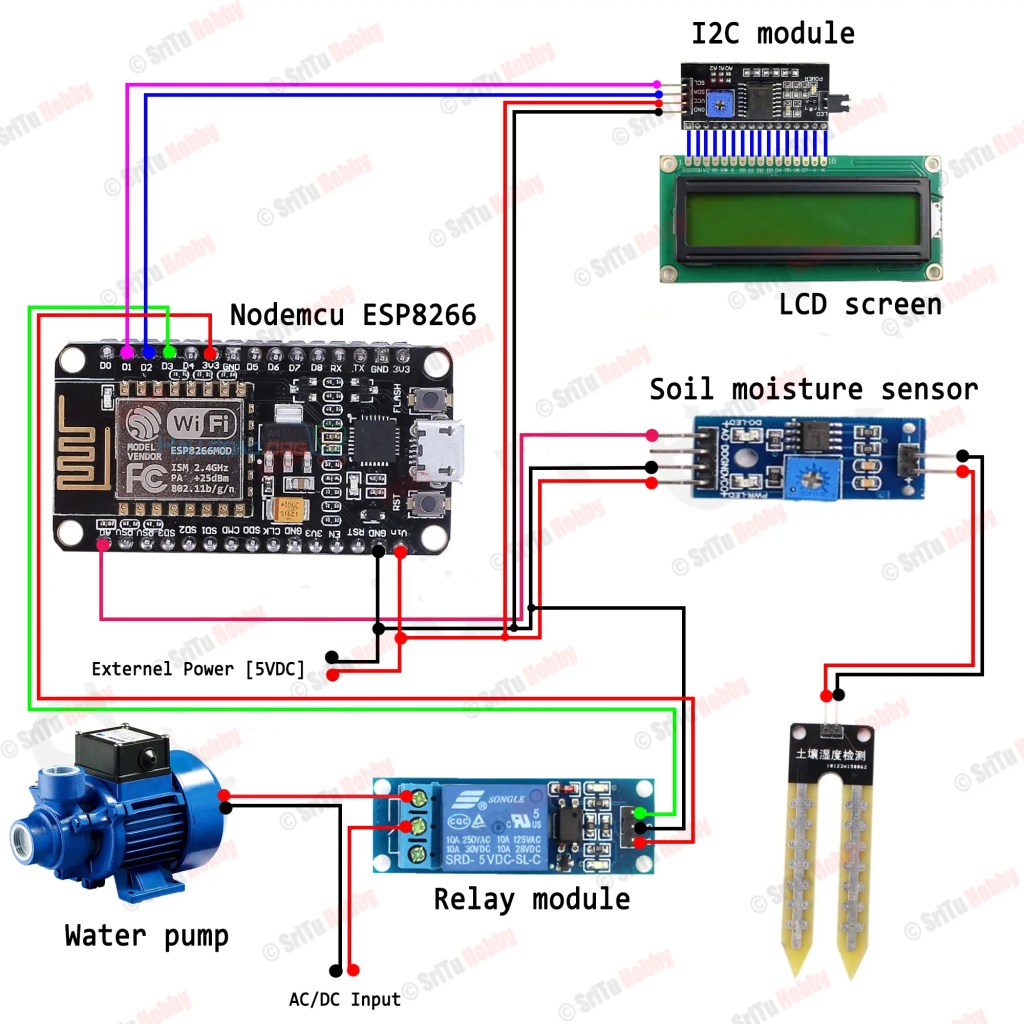
Use in Watering Systems:

Acts as an intermediary between the NodeMCU and the water pump or solenoid valve.

Turns the pump on or off based on signals from the NodeMCU.

These components together create an efficient system for automated plant watering.

**Circuit Diagram**



**Code**

//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[] = "";//Enter your Auth token

char ssid[] = "";//Enter your WIFI name

char pass[] = "";//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;

Blynk.virtualWrite(V0, value);

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

}

1. [↑](#footnote-ref-1)