Agricultural Automation System with Weather Integration

Fatima Imran and Abdullah Wasi
Electronics Department, Ned University of Engineering and Technology.
fatimaimran.dev@gmail.com
wasia5297@gmail.com

Abstract— This project presents a prototype for a smart agricultural automation system designed to optimize irrigation by integrating IoT technologies. Utilizing sensors, a NodeMCU ESP8266 microcontroller, and actuators, the system monitors soil and environmental conditions to automate water delivery. The scalable prototype aims to address water management challenges in agriculture, offering a sustainable solution for large-scale implementation.

I. Introduction

Agriculture faces challenges like ***water scarcity***, unpredictable weather, and inefficient resource management.Our project addresses these issues by automating irrigation systems to optimize water usage and enhance crop health.

II. Objective

- Design and develop a prototype of a smart irrigation system for large-scale agricultural use.
- Automate water supply to plants based on soil moisture, and soil temperature.
- Reduce water wastage and improve crop productivity through efficient monitoring.

III. Methodology

A. Approach

Build a prototype using sensors and actuators to monitor soil and environmental conditions.

B. Tools and Techniques

1. ***Programming:*** Arduino IDE and NodeMCU for microcontroller programming.
2. ***Data Monitoring:*** Cloud integration for real-time updates (Blynk)
3. ***Libraries:*** Installed libraries for successful compilation of code.

C. Hardware

Node MCU ESP8266 (microcontroller for data processing and communication).

1) Sensors:

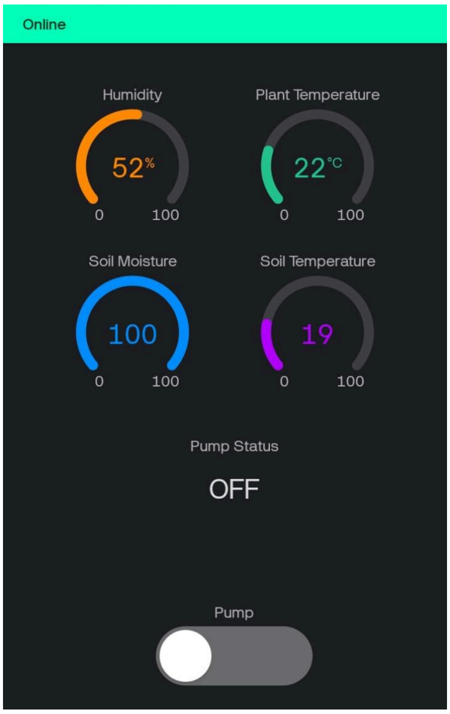
- DHT11 (temperature and humidity monitoring).
- Soil moisture sensor (soil water level measurement).
- DS18B20 waterproof temperature probe (accurate temperature readings).

2) Actuators:

- Mini water pump (for irrigation).
- Relay module (for controlling the pump).

C. Software

The system has been implemented on mobile and web, so as to have ease in control of the system and monitor the plant with the mobile app. In order to implement the app, we have used Blynk cloud server and built the dashboard using its free device manager. To connect our Esp8266 with Blynk, we installed the Blynk library, and used it in our code. The connection with Blynk is made via the WiFi.



IV. Code

Following is the code that has been implemented in our circuit to get data from the sensors and display it in our app.

e_Gardener_Final_2.ino

```
#define BLYNK_TEMPLATE_ID "TMPL6ldTVNSbF"
#define BLYNK_TEMPLATE_NAME "eGardener"
#define BLYNK_AUTH_TOKEN "IQy7saU4Nzp1M3YTGCGQBD-xnPXLjDFl"


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
#include <DallasTemperature.h>
#include <OneWire.h>

// Wi-Fi Credentials
char ssid[] = "";     // Replace with your Wi-Fi SSID
char pass[] = "";     // Replace with your Wi-Fi Password

// Pin Definitions
#define DHTPIN D5           // DHT sensor pin
#define DHTTYPE DHT11       // Define the type of DHT sensor (DHT11 or DHT22)
#define MOISTURE_PIN A0     // Soil Moisture sensor connected to analog pin A0
#define TEMP_PIN D7         // DS18B20 temperature sensor
#define PUMP D0             // Pump control pin

// Initialize Sensors
DHT dht(DHTPIN, DHTTYPE);
OneWire oneWire(TEMP_PIN);
DallasTemperature sensors(&oneWire);

// Timer for periodic tasks
```

```
OneWire oneWire(TEMP_PIN);
DallasTemperature sensors(&oneWire);

// Timer for periodic tasks
BlynkTimer timer;

// Pump control variables
bool pumpManualControl = false;  // To track manual control via Blynk
int moistureThreshold = 450;     // Moisture threshold for automatic control
bool pumpStatus = false;         // Track pump ON/OFF status

// Virtual Pin for Blynk Button
BLYNK_WRITE(V4) {  // Virtual pin V4 for pump manual control
  pumpManualControl = param.asInt();
  pumpStatus = pumpManualControl; // Update the pump status
  digitalWrite(PUMP, pumpStatus); // Update pump state based on manual control
  Blynk.virtualWrite(V5, pumpStatus ? "ON" : "OFF"); // Update pump status in Blynk
}

// Function to control pump automatically based on soil moisture
void controlPump() {
  int soilMoisture = analogRead(MOISTURE_PIN);

  // Map the soil moisture value to a 0-100 range
  int mappedSoilMoisture = map(soilMoisture, 1023, 300, 0, 100);

  if (!pumpManualControl) {  // Only control pump automatically if not in manual mode
    if (soilMoisture > moistureThreshold) {
```

```
  if (!pumpManualControl) {  // Only control pump automatically if not in manual mode
    if (soilMoisture > moistureThreshold) {
      pumpStatus = true;       // Pump is ON
      digitalWrite(PUMP, HIGH);
    } else {
      pumpStatus = false;      // Pump is OFF
      digitalWrite(PUMP, LOW);
    }
    Blynk.virtualWrite(V5, pumpStatus ? "ON" : "OFF"); // Update pump status in Blynk
    Blynk.virtualWrite(V2, mappedSoilMoisture);        // Send mapped soil moisture to
  }
}

// Function to read sensor data and send it to Blynk
void sendSensorData() {
  // Read soil moisture
  int soilMoisture = analogRead(MOISTURE_PIN);

  // Map the soil moisture value to a 0-100 range
  int mappedSoilMoisture = map(soilMoisture, 1023, 300, 0, 100);

  // Read DHT sensor values
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  // Read soil temperature from DS18B20
  sensors.requestTemperatures();
  float soilTemperature = sensors.getTempCByIndex(0);
```

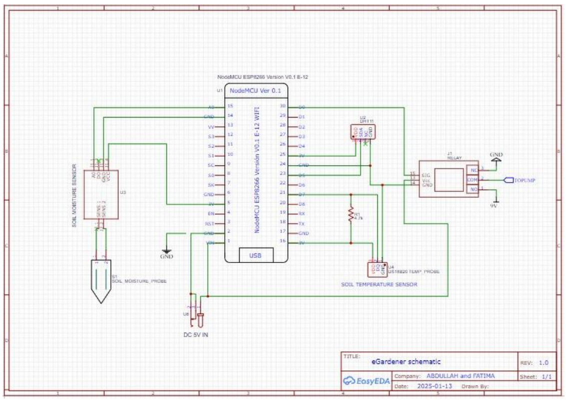
```
sensors.requestTemperatures();
float soilTemperature = sensors.getTempCByIndex(0);

// Validate readings
if (isnan(humidity) || isnan(temperature)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

if (soilTemperature == DEVICE_DISCONNECTED_C) {
  Serial.println("Failed to read from DS18B20 sensor!");
  return;
}

// Print data to Serial Monitor
Serial.print("Soil Moisture: ");
Serial.print(mappedSoilMoisture);
Serial.print(" | Temperature: ");
Serial.print(temperature);
Serial.print("°C | Humidity: ");
Serial.print(humidity);
Serial.print("% | Soil Temp: ");
Serial.print(soilTemperature);
Serial.println("°C");

// Send data to Blynk
Blynk.virtualWrite(V0, temperature);     // Virtual pin V0 for Temperature
Blynk.virtualWrite(V1, humidity);        // Virtual pin V1 for Humidity
Blynk.virtualWrite(V2, mappedSoilMoisture);  // Virtual pin V2 for Soil Moisture
```

```
// Send data to Blynk
Blynk.virtualWrite(V0, temperature);     // Virtual pin V0 for Temperature
Blynk.virtualWrite(V1, humidity);        // Virtual pin V1 for Humidity
Blynk.virtualWrite(V2, mappedSoilMoisture);  // Virtual pin V2 for Soil Mois
Blynk.virtualWrite(V3, soilTemperature); // Virtual pin V3 for Soil Temperat
}

void setup() {
  // Debug Console
  Serial.begin(115200);

  // Connect to Wi-Fi and Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Initialize sensors
  dht.begin();
  sensors.begin();

  // Initialize pump pin
  pinMode(PUMP, OUTPUT);
  digitalWrite(PUMP, LOW);  // Ensure pump is off initially

  // Set a timer to send sensor data every 2 seconds
  timer.setInterval(2000L, sendSensorData);

  // Set a timer to control the pump every 1 second
  timer.setInterval(1000L, controlPump);
```

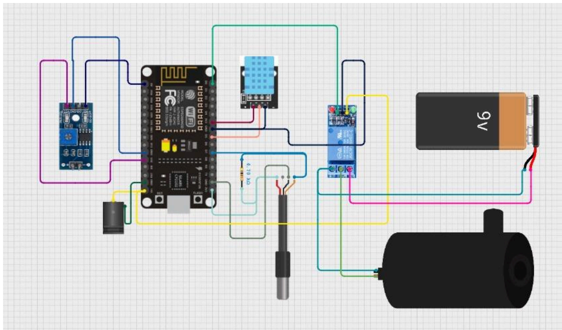
```
void setup() {
  // Debug Console
  Serial.begin(115200);

  // Connect to Wi-Fi and Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Initialize sensors
  dht.begin();
  sensors.begin();

  // Initialize pump pin
  pinMode(PUMP, OUTPUT);
  digitalWrite(PUMP, LOW);  // Ensure pump is off initially

  // Set a timer to send sensor data every 2 seconds
  timer.setInterval(2000L, sendSensorData);

  // Set a timer to control the pump every 1 second
  timer.setInterval(1000L, controlPump);
}

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

V. Result

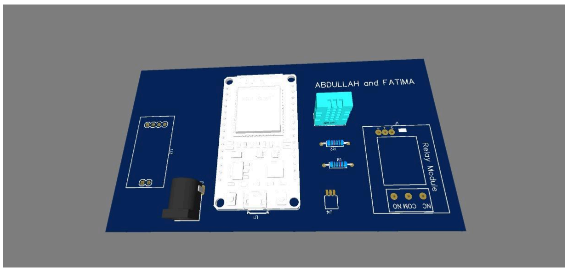
This is the final outcome of our project that can be implemented in a plant pot to monitor the data, control pump through the app over the internet from any location. Apart from this, the ESP8266 will itself provide irrigation whenever the soil moisture drops below a certain level. You can view the plant temperature, soil temperature, humidity and soil moisture with this system.



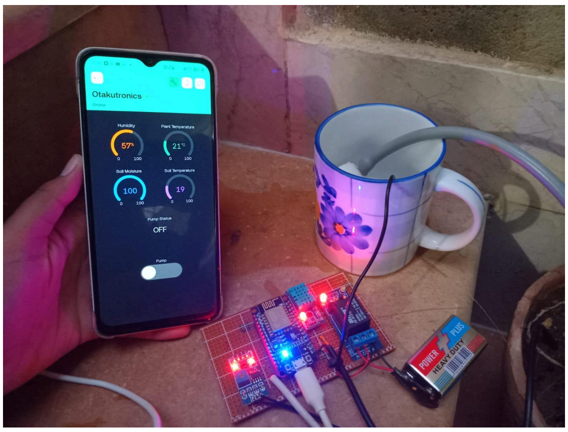
Schematic Diagram



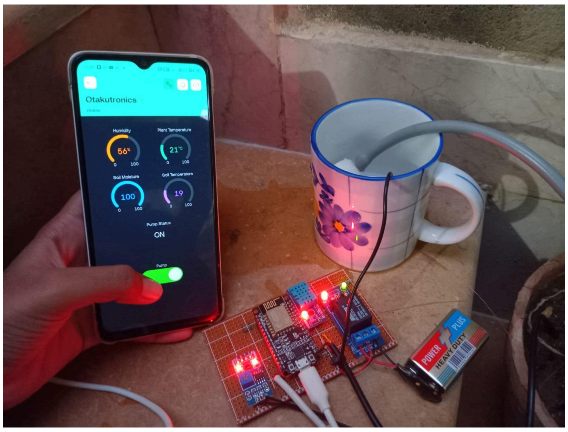
2d model of the system



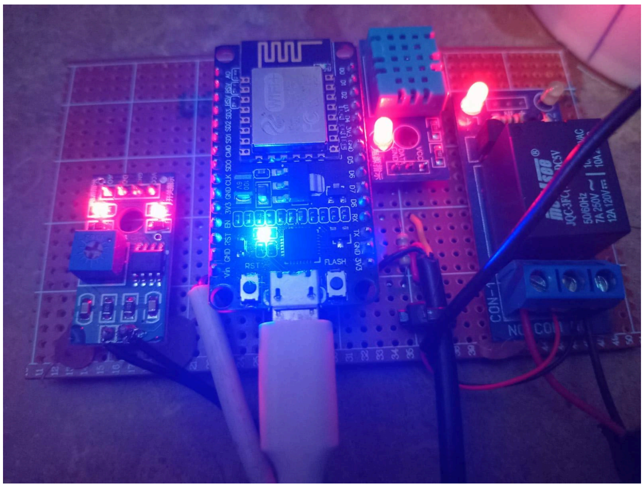
3d view of the system



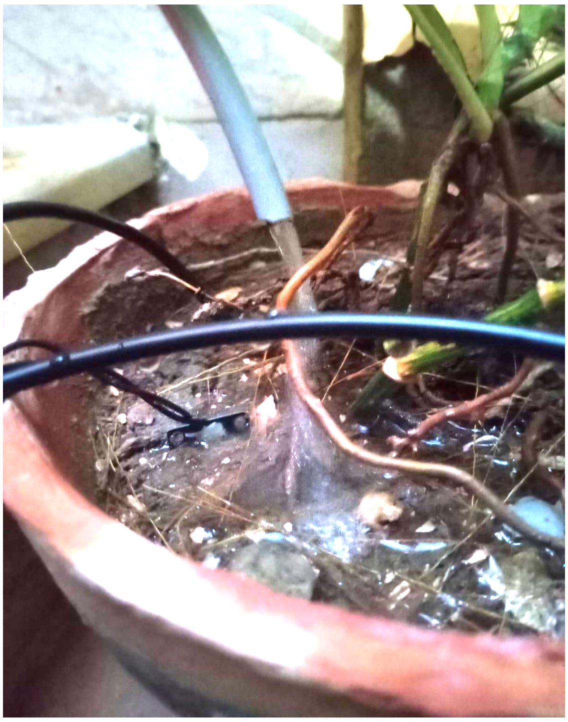
Working of the system whilst the pump is off because the soil moisture is 100%



Working of the system when the pump is switched ON by using the app



Close view of the circuit when it is connected with the app



The water pump provides water to the plant