#define BLYNK\_TEMPLATE\_ID "TMPL3sFnahYPk"

#define BLYNK\_TEMPLATE\_NAME "SMART IRRIGATION WITH ESP32"

#define BLYNK\_AUTH\_TOKEN "8baMSzL2MOrStqSTyjoc\_JJaPqh8LvNN"

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

// Pin setup

#define SOIL\_MOISTURE\_PIN 34    // Analog pin for soil moisture sensor

#define THRESHOLD\_MOISTURE 60  // Moisture threshold percentage (0–100)

#define PUMP\_PIN 2             // D2 connected to relay IN pin

#define PUMP\_SWITCH V6         // Virtual pin on Blynk for manual control

// 💧 CRITICAL CALIBRATION - ADJUST WET\_VALUE BASED ON YOUR MEASUREMENT 💧

#define DRY\_VALUE 4095         // Raw reading for 0% moisture (sensor in dry air - uses max ADC value)

#define WET\_VALUE 1750         // ⚠️ CHANGE THIS: Raw reading for 100% moisture (sensor in water).

                               // Use the lowest raw value you recorded when fully submerged.

// WiFi credentials

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Ram Ram";       // 🔹 your WiFi name

char pass[] = "12345678";      // 🔹 your WiFi password

BlynkTimer timer;

bool isPumpOn = false;

// Function to read & smooth soil moisture data

int readSoilMoisture() {

  int total = 0;

  for (int i = 0; i < 10; i++) {

    total += analogRead(SOIL\_MOISTURE\_PIN);

    delay(10);

  }

  return total / 10;

}

// Function to send data and control pump

void sendSensorData() {

  int soilMoisture = readSoilMoisture();

  // 💥 CORRECTED MAPPING 💥

  // Maps raw value from DRY (0%) to WET (100%)

  int soilPercent = map(soilMoisture, DRY\_VALUE, WET\_VALUE, 0, 100);

  // Clamp values to prevent display errors outside 0-100% range

  if (soilPercent > 100) soilPercent = 100;

  if (soilPercent < 0) soilPercent = 0;

  Serial.print("Soil Moisture (Raw: ");

  Serial.print(soilMoisture);

  Serial.print("): ");

  Serial.print(soilPercent);

  Serial.println("%");

  Blynk.virtualWrite(V5, soilPercent); // Send to Blynk app gauge

  // ACTIVE-LOW RELAY CONTROL (Correctly prioritizing manual control)

  if (isPumpOn) { // Priority 1: Manual Control is Active (Pump must be ON)

    digitalWrite(PUMP\_PIN, LOW); // Relay ON

  } else {

    // Priority 2: Automatic Mode (Only runs if manually OFF)

    if (soilPercent < THRESHOLD\_MOISTURE) {

        // Automatic ON

        digitalWrite(PUMP\_PIN, LOW);  // Relay ON

        Blynk.logEvent("moisture\_alert", "Soil moisture below threshold!");

        Serial.println("💧 Pump turned ON automatically (low moisture)");

    } else {

        // Automatic OFF

        digitalWrite(PUMP\_PIN, HIGH); // Relay OFF

        Serial.println("🌿 Pump OFF (soil moist enough)");

    }

  }

}

// Manual control from Blynk app

BLYNK\_WRITE(PUMP\_SWITCH)

{

  isPumpOn = param.asInt();

  if (isPumpOn) {

    digitalWrite(PUMP\_PIN, LOW);  // Relay ON

    Serial.println("✅ Pump manually turned ON (Blynk)");

  } else {

    digitalWrite(PUMP\_PIN, HIGH); // Relay OFF

    Serial.println("⛔ Pump manually turned OFF (Blynk)");

  }

}

void setup() {

  Serial.begin(9600);

  pinMode(PUMP\_PIN, OUTPUT);

  digitalWrite(PUMP\_PIN, HIGH); // Relay OFF initially (active LOW)

  Blynk.begin(auth, ssid, pass);

  timer.setInterval(3000L, sendSensorData); // Check every 3 seconds

  Serial.println("🌱 Smart Irrigation System Starting...");

}

void loop() {

  Blynk.run();

  timer.run();

}