

Code For Iot-Based-Plant-Monitoring-And-Irrigation-System:

```
#define BLYNK_TEMPLATE_ID "TMPL3ThmHnAMQ"
#define BLYNK_TEMPLATE_NAME "1"
#define BLYNK_AUTH_TOKEN "-"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>

// WiFi credentials
char ssid[] = "arun12"; // Replace with this Wi-Fi SSID
char pass[] = "12121234"; // Replace with this Wi-Fi Password

// Pin definitions
#define MOISTURE1_PIN 34
#define MOISTURE2_PIN 35
#define RELAY1_PIN 23
#define RELAY2_PIN 25
#define LED_PIN 2
#define PIR_PIN 5
#define DHT_PIN 19
#define DHT_TYPE DHT11

// Create DHT object
DHT dht(DHT_PIN, DHT_TYPE);

// Blynk Virtual Pins
#define VMOISTURE1_PIN V0
#define VMOISTURE2_PIN V1
#define VTEMP_PIN V2
#define VHUMIDITY_PIN V3
#define VRELAY1_PIN V4
```

```

#define VRELAY2_PIN V5

// Variables

bool manualMode1 = false;
bool manualMode2 = false;
unsigned long lastMotionTime = 0;
unsigned long motionDelay = 2000; // Delay for PIR motion debounce
bool motionNotified = false;    // To prevent repeated notifications for motion
float tempThreshold = 35.0;     // Temperature threshold for alert

void sendDataToBlynk() {
    int moisture1Level = analogRead(MOISTURE1_PIN);
    int moisture2Level = analogRead(MOISTURE2_PIN);
    int moisture1Percentage = map(moisture1Level, 0, 4095, 100, 0);
    int moisture2Percentage = map(moisture2Level, 0, 4095, 100, 0);

    Blynk.virtualWrite(VMOISTURE1_PIN, moisture1Percentage);
    Blynk.virtualWrite(VMOISTURE2_PIN, moisture2Percentage);

    if (!manualMode1) {
        if (moisture1Percentage < 30) {
            digitalWrite(RELAY1_PIN, LOW);
            Blynk.virtualWrite(VRELAY1_PIN, HIGH);
        } else {
            digitalWrite(RELAY1_PIN, HIGH);
            Blynk.virtualWrite(VRELAY1_PIN, LOW);
        }
    }

    if (!manualMode2) {
        if (moisture2Percentage < 30) {
            digitalWrite(RELAY2_PIN, LOW);
            Blynk.virtualWrite(VRELAY2_PIN, HIGH);
        } else {
            digitalWrite(RELAY2_PIN, HIGH);

```

```
Blynk.virtualWrite(VRELAY2_PIN, LOW);  
  }  
}  
}
```

```
BLYNK_WRITE(VRELAY1_PIN) {  
  int buttonState = param.asInt();  
  manualMode1 = buttonState;  
  
  if (buttonState) {  
    digitalWrite(RELAY1_PIN, LOW);  
    Serial.println("Pump 1 ON via Blynk");  
  } else {  
    digitalWrite(RELAY1_PIN, HIGH);  
    Serial.println("Pump 1 OFF via Blynk");  
  }  
}
```

```
BLYNK_WRITE(VRELAY2_PIN) {  
  int buttonState = param.asInt();  
  manualMode2 = buttonState;  
  
  if (buttonState) {  
    digitalWrite(RELAY2_PIN, LOW);  
    Serial.println("Pump 2 ON via Blynk");  
  } else {  
    digitalWrite(RELAY2_PIN, HIGH);  
    Serial.println("Pump 2 OFF via Blynk");  
  }  
}
```

```
void setup() {  
  Serial.begin(115200);
```

```

Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

pinMode(MOISTURE1_PIN, INPUT);
pinMode(MOISTURE2_PIN, INPUT);
pinMode(RELAY1_PIN, OUTPUT);
pinMode(RELAY2_PIN, OUTPUT);
pinMode(LED_PIN, OUTPUT);
pinMode(PIR_PIN, INPUT);

dht.begin();

digitalWrite(RELAY1_PIN, HIGH);
digitalWrite(RELAY2_PIN, HIGH);
digitalWrite(LED_PIN, LOW);

Serial.println("Plant Monitoring System Initialized");
}

void loop() {
  Blynk.run();

  static unsigned long lastSendTime = 0;
  if (millis() - lastSendTime >= 5000) {
    sendDataToBlynk();
    lastSendTime = millis();
  }

  if (digitalRead(PIR_PIN) == HIGH && millis() - lastMotionTime > motionDelay) {
    if (!motionNotified) {
      Blynk.logEvent("motion_alert", "Motion Detected!");
      Serial.println("Motion Detected!");
      motionNotified = true; // Prevent repeated notifications
    }
    for (int i = 0; i < 10; i++) {

```

```

digitalWrite(LED_PIN, HIGH);
delay(100);
digitalWrite(LED_PIN, LOW);
delay(100);
    }
lastMotionTime = millis();
    } else if (millis() - lastMotionTime > 10000) {
motionNotified = false; // Reset motion notification after 10 seconds
    }

static unsigned long lastDHTTime = 0;
if (millis() - lastDHTTime >= 5000) {
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();

    if (temperature > tempThreshold) {
Blynk.logEvent("temp_alert", "Temperature Exceeded Threshold!");
Serial.println("High Temperature Alert!");
    }

Blynk.virtualWrite(VTEMP_PIN, temperature);
Blynk.virtualWrite(VHUMIDITY_PIN, humidity);
lastDHTTime = millis();
    }
}

```