AUTOMATIC PLANT WATERING POT

```
#define BLYNK_PRINT Serialn

#define BLYNK_TEMPLATE_ID "TMPL3prsC5fTE"

#define BLYNK_TEMPLATE_NAME "smart irrigation"

#define BLYNK_AUTH_TOKEN "ZC_-uWq-PLpEqNfsDkfEyBgoKE9fdzr5"

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>

#include <DHT_U.h>

#define DHTPIN 23

#define Soil 36

#define Relay1 26

#define Relay2 27

#define DHTTYPE    DHT11     // DHT 11

//#define DHTTYPE    DHT22     // DHT 22 (AM2302)

//#define DHTTYPE    DHT21     // DHT 21 (AM2301)

char ssid[] = "iotdata";

char pass[] = "123456789";

int sensorValue = 0;

int SoilValue = 0;

int Count = 0;

DHT_Unified dht(DHTPIN, DHTTYPE);

void setup()

{

  Serial.begin(9600);
  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  digitalWrite(Relay1, LOW);
  digitalWrite(Relay2, LOW);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  dht.begin();
  Serial.println(F("DHTxx Unified Sensor Example"));
  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  Serial.println(F("------------------------------------"));
  Serial.println(F("Temperature Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("°C"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("°C"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("°C"));
  Serial.println(F("------------------------------------"));
  dht.humidity().getSensor(&sensor);
  Serial.println(F("Humidity Sensor"));
  Serial.print  (F("Sensor Type: ")); Serial.println(sensor.name);
  Serial.print  (F("Driver Ver:  ")); Serial.println(sensor.version);
  Serial.print  (F("Unique ID:   ")); Serial.println(sensor.sensor_id);
  Serial.print  (F("Max Value:   ")); Serial.print(sensor.max_value); Serial.println(F("%"));
  Serial.print  (F("Min Value:   ")); Serial.print(sensor.min_value); Serial.println(F("%"));
  Serial.print  (F("Resolution:  ")); Serial.print(sensor.resolution); Serial.println(F("%"));
  Serial.println(F("------------------------------------"));
  delay(1000);
}
void Humidity()
{
```

```
sensors_event_t event;
dht.temperature().getEvent(&event);
if (isnan(event.temperature)) Serial.println(F("Error reading temperature!"));
else
{
  Serial.print(F("Temperature: "));
  Serial.print(event.temperature);
  Serial.println(F("°C"));
  Blynk.virtualWrite(V1, event.temperature);
}
dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity))
{
  Serial.println(F("Error reading humidity!"));
  delay(1000);
  Count++;
}
else
{
  Serial.print(F("Humidity: "));
  Serial.print(event.relative_humidity);
  Blynk.virtualWrite(V2, event.relative_humidity);
  Serial.println(F("%"));
  Count = 0;
  if(event.relative_humidity <= 80)
  {
    digitalWrite(Relay2, LOW);
    Blynk.virtualWrite(V4, LOW);
  }
  else
  {
      digitalWrite(Relay2, HIGH);
      Blynk.virtualWrite(V4, HIGH);
    }
  }
}
void loop()
{
  Blynk.run();
  Humidity();
  sensorValue = analogRead(Soil);
  SoilValue = map(sensorValue, 0, 4096, 100, 0);
  Serial.println("Soil Level = " + String(SoilValue));
  Blynk.virtualWrite(V0, SoilValue);
  if(SoilValue <= 8)
  {
    digitalWrite(Relay1, HIGH);
    Blynk.virtualWrite(V3, HIGH);
  }
  else
  {
    digitalWrite(Relay1, LOW);
    Blynk.virtualWrite(V3, LOW);
  }
  delay(500);
  if(Count >= 10) ESP.restart();
}
```