BLYNK UPLINK SYSTEM DATA COMMUNICATION

A. Objective

1. Understand the basic concepts of Blynk and how the uplink system works on the ESP32.
2. Using the website Blynk to control and monitor devices connected to the ESP32.
3. Implementing an uplink system between ESP32 and Blynk using Wi-Fi connection for device control.
4. Analyze data sent and received via Blynk .

B. Basic Theory

In data communication, uplink is the process of sending data from a device to a server or cloud . Uplink can be defined as sending information, such as sensor data or device status, to a processing platform or application that can be accessed by the user. Blynk is a cloud- based IoT platform that allows users to control and monitor devices in real -time via a mobile or web application . In order for the ESP32 to connect to Blynk , several main components are required, namely:

- Blynk Library for connecting devices to servers.
- Authentication Token as an access key to the project in the Blynk application .
- Wi-Fi SSID & Password so that the device can connect to the internet.

In this jobsheet , monitoring practices will be carried out via Serial Monitor and Blynk , with input in the form of temperature sensors (DHT) and light (LDR).

C. Tools & Materials

a. ESP32
b. DHT Sensor
c. LDR Sensor
d. Breadboard
e. Jumper cables

f. Laptop installed Arduino IDE and libraries

g. USB Type C Cable

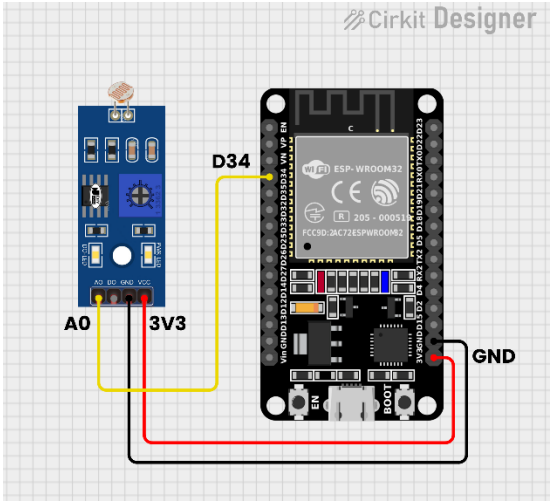
D. Work safety

1. Do or carry out the practicum in a clean and dry place.
2. Using a laptop or computer properly and correctly.
3. Do not directly touch the parts of the lab kit that are electrically active.
4. Make sure the power The practical kit supply is in a dead state during the installation process, or assembly of the practical kit.
5. Always pay attention to the lab kit ports . Do not mix them up, reverse them, or place them incorrectly during installation or assembly of the lab kit.
6. Use the components in the lab kit as intended.
7. If a short occurs circuit , turn off the power immediately supply in the practical kit.
8. Follow the practical steps as stated in the work steps and pay attention to work safety.

E. Work steps

6.1 Monitoring LDR from Serial Monitor

a. Create a circuit like the following image. Use the ADC pin , because this LDR uses analog input .



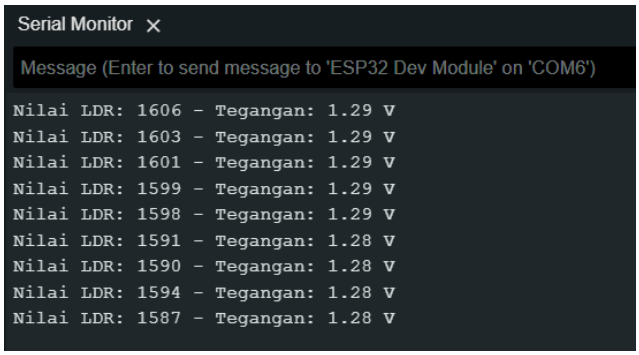
b. Enter the following code.

```
#define LDR_PIN 34 // Gunakan pin analog yang sesuai pada ESP32

void setup() {
  Serial.begin(115200); // Mulai komunikasi serial
}

void loop() {
  int ldrValue = analogRead(LDR_PIN); // Baca nilai dari sensor
LDR
  float voltage = (ldrValue / 4095.0) * 3.3; // Konversi nilai
ADC ke tegangan (3.3V referensi)

  // Tampilkan nilai di Serial Monitor
  Serial.print("LDR value: ");
  Serial.print(ldrValue);
  Serial.print(" - Voltage: ");
  Serial.print(voltage);
  Serial.println(" V");

  delay(1000); // Tunggu 1 detik sebelum pembacaan berikutnya
}
```

c. Connect ESP32 to laptop, before Verify or Compile code make sure the ESP32 Board is connected to the Arduino IDE.

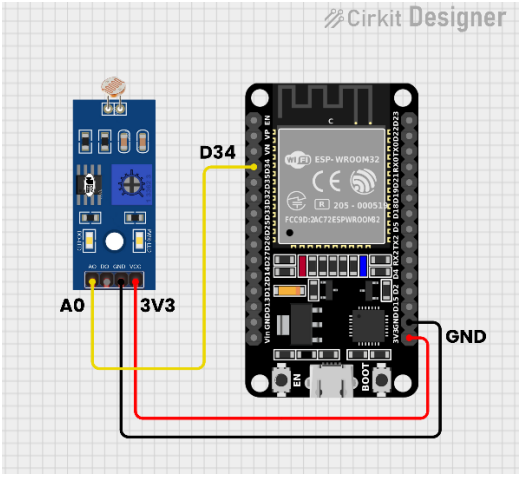
d. If the compilation process is complete and there are no errors , then upload the program, then wait for it to finish.

e. Enter the Serial Monitor as shown in the following image. Then the "LDR Value" data appears in the form of an Analog value, then it is changed into voltage form through the equation `( ldrValue / 4095.0 ) * 3.3 ; // Convert ADC value to voltage.`



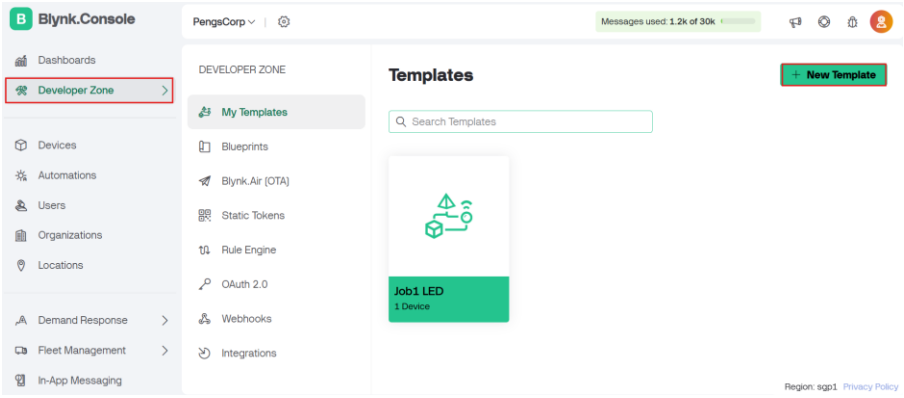
6.2 Monitoring LDR from Blynk

a. Create a circuit like the following image. Use the ADC pin , because this LDR uses analog input .

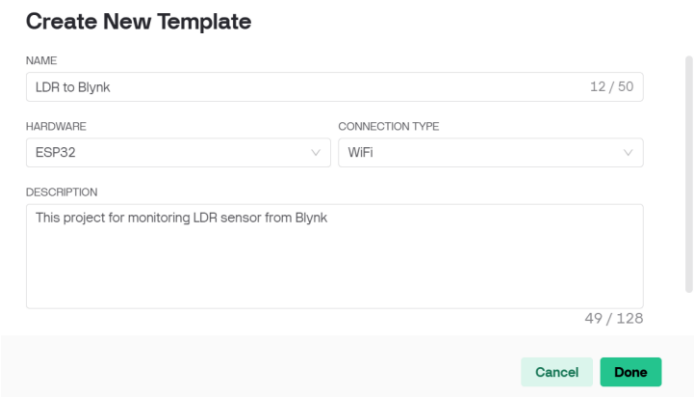


b. A configuration is required first so that Blynk and ESP32 can connect to each other. First, create a Blynk account via the site https://blynk.io/ .

c. After creating an account, you will be directed to the main Dashboard . Go to **the Developer Zone menu** > click **New Template** .



d. Fill in as needed, description is optional.

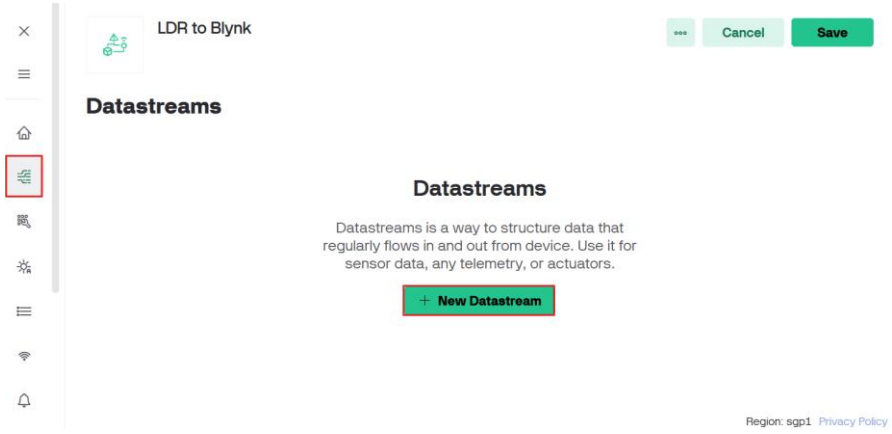


e. Save the code (later) for configuration in the Arduino IDE.

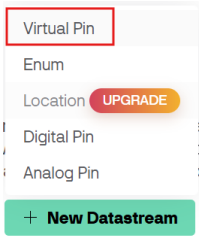
```
#define BLYNK_TEMPLATE_ID "TMPL6_MYpZj-
q"
#define BLYNK_TEMPLATE_NAME "LDR to
Blynk"
```

Region: sgp1

f. Next setting **Datastreams** so that sensor data to actuators can enter or exit Blynk , by clicking **New Datastream** .



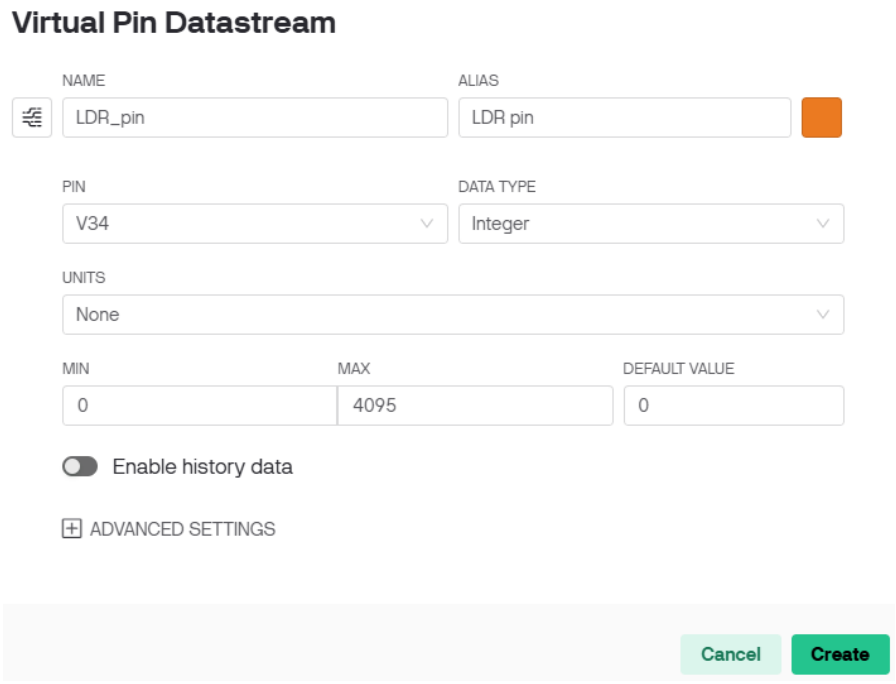
g. Select **Virtual Pin**



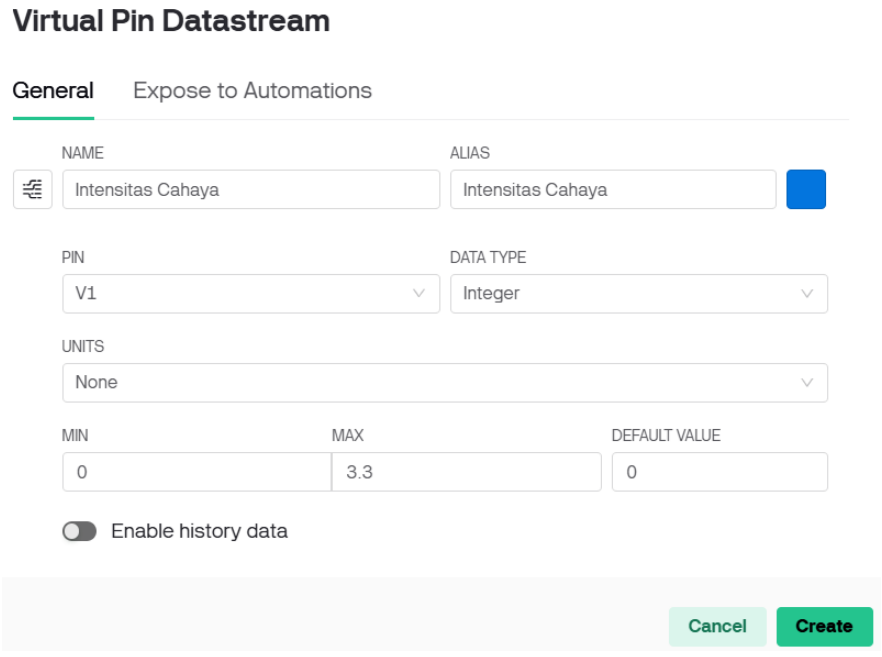
h. Fill in as needed with the following provisions:

- **NAME** is used to give a name to the Pin.
- **PIN** is used as a virtual pin in Blynk .
- **DATA TYPE** corresponds to the value data read.
- **UNITS** is a data unit, because it only uses ADC value readings, it is left as **None** .
- The LDR sensor reads the analog value, so the **MIN** and **MAX values** are changed to 0 – 4095, because the ESP32 uses 12-bit resolution.

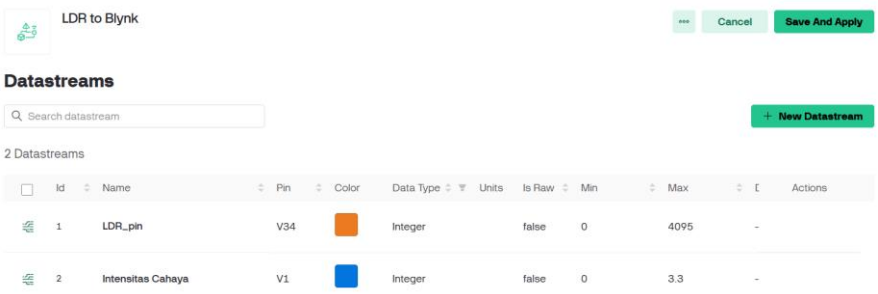
After configuring the LDR data pin above, then click **Create** .



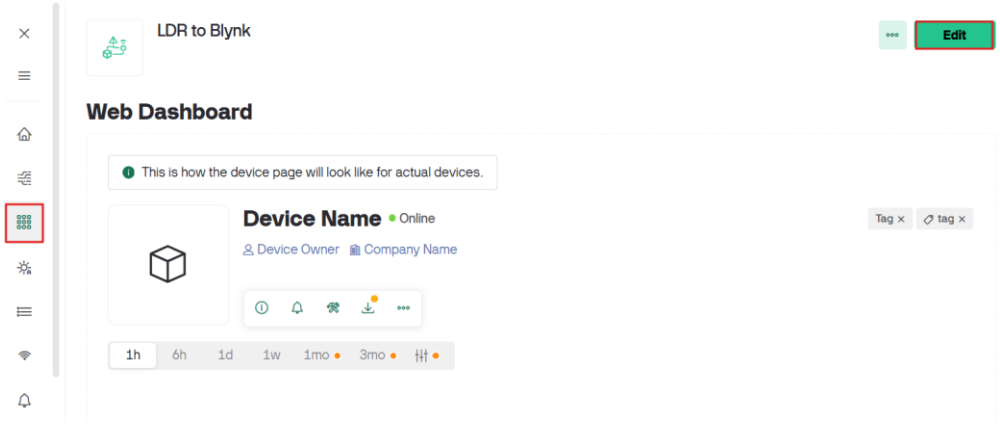
i. Next, configure the Light Intensity pin with the voltage output.



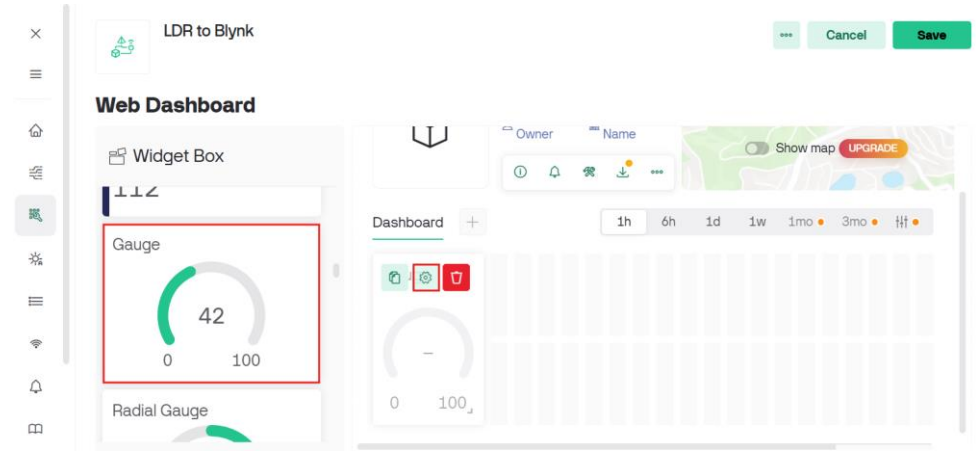
j. After that click **Save** .



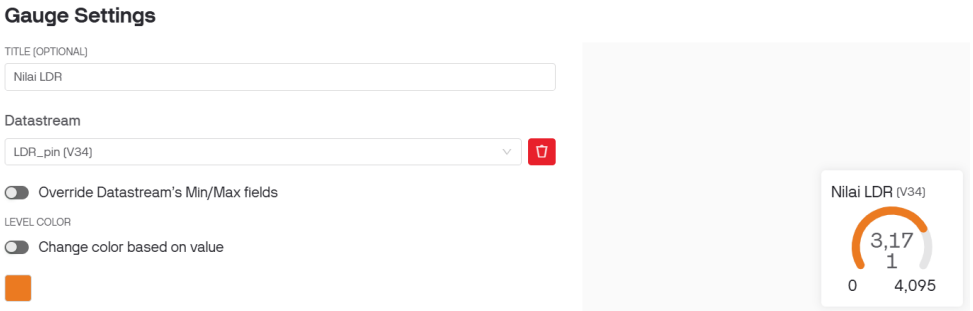
k. Next, configure **the Web Dashboard** by clicking the **Edit menu** .



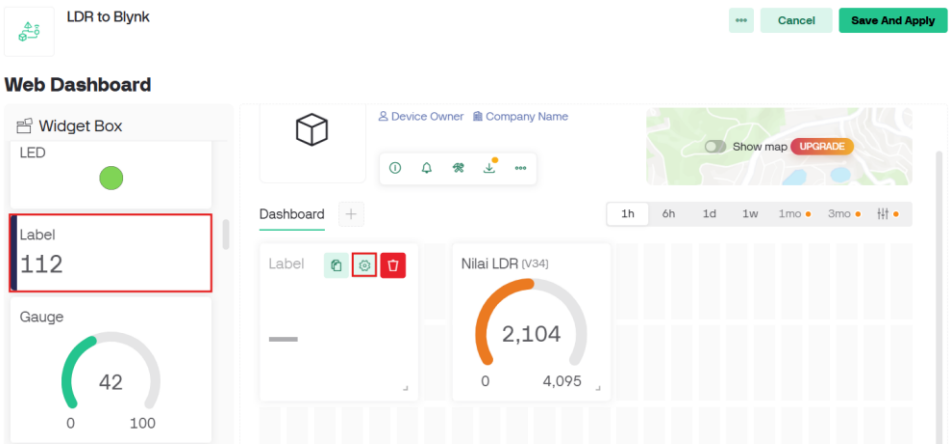
l. Select **the Widget** according to your needs, then drag it to the right as shown in the following image. If you have, click the **settings icon** on the widget .



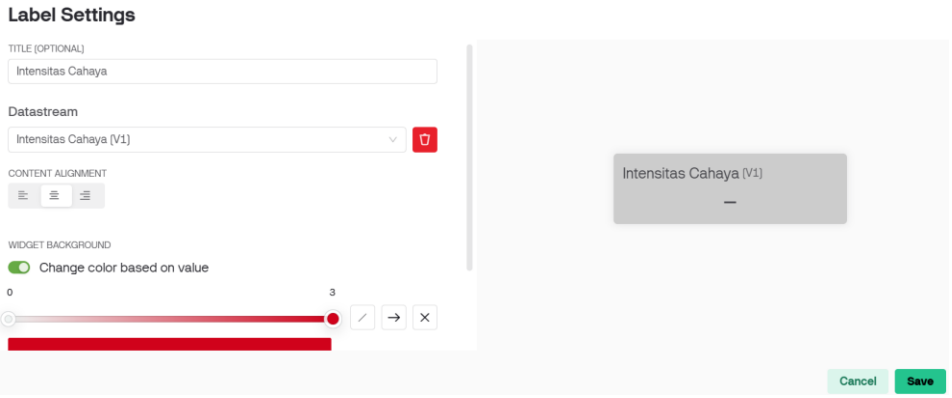
m. Enter **the TITLE** as needed, then select **the Datastreams** that have been created, namely **LDR_pin (V34)** . Then click **Save** .



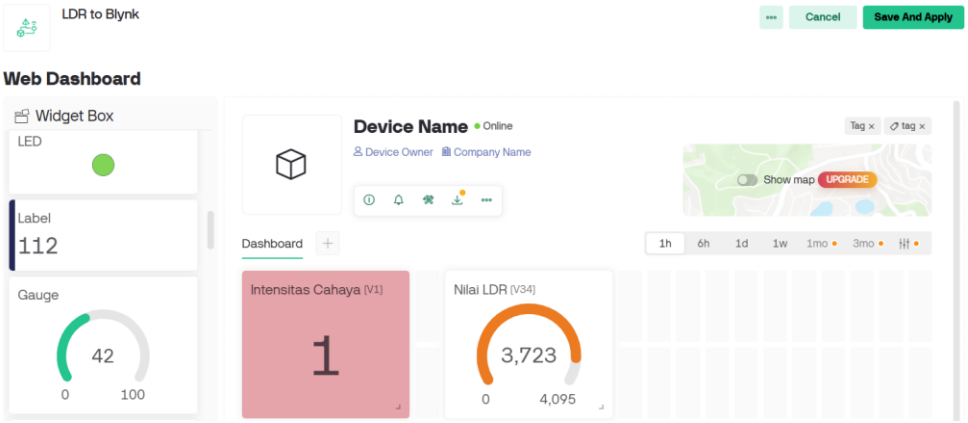
n. Add another Widget for Light Intensity.



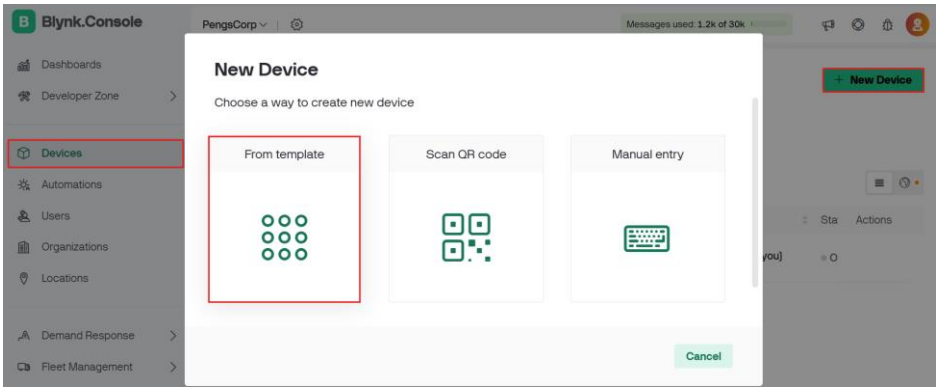
o. Fill the Datastream column with the Light Intensity (V1) that was created earlier.



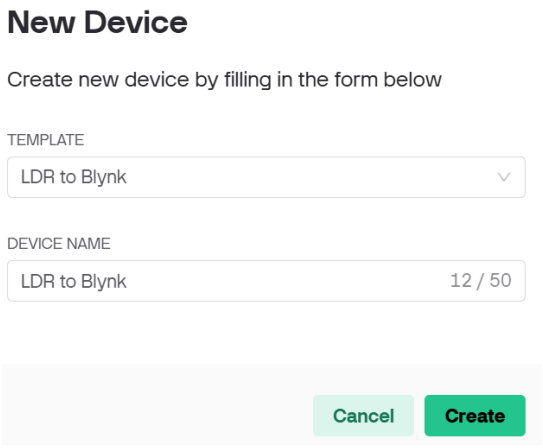
p. When you have finished configuring the **Web Dashboard** , click **Save** .



q. Next, go to the Devices menu to add a virtual device to Blynk . Click **New Device** > then select **From template** .



r. In the **TEMPLATE column** , select the template that has been created. Then enter **the DEVICE NAME** as needed, and click **Create** .



s. Save the following code.

```
#define BLYNK_TEMPLATE_ID "TMPL6_MYpZj-q"
#define BLYNK_TEMPLATE_NAME "LDR to Blynk"
#define BLYNK_AUTH_TOKEN "1SY9-D1bQiV5l-c2NWsvj0dRlLqBKZt7"
```

t. Blynk configuration is complete, next is the program configuration on Arduino IDE. Enter the following code, some things need to be adjusted again:

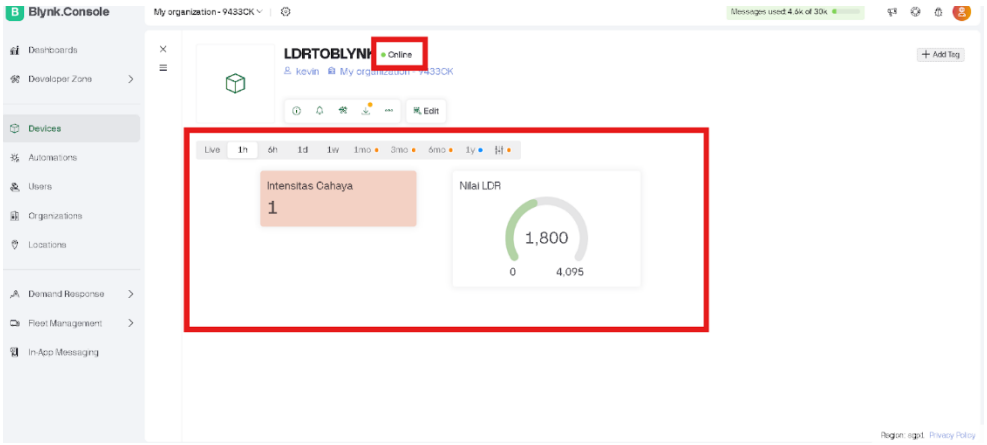
- Blynk IDE template
- Blynk template name
- Blynk device name
- Token authentication Blynk
- WiFi SSID
- Password WiFi , and
- Virtual Pin

```
#define BLYNK_TEMPLATE_ID "TMPL6_MYpZj-q"
#define BLYNK_TEMPLATE_NAME "LDR to Blink"

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

// WiFi Configuration
```

```
char ssid[] = "realme";          // Ganti dengan nama WiFi Anda
char pass[] = "12345678";        // Ganti dengan password WiFi
Anda

// Blynk Auth Token
char auth[] = "1SY9_D1bQiV5l-c2NWsvj0dRlLqBKZt7"; // Ganti
dengan token Blynk Anda

// Pin LDR
#define LDR_PIN 34

void setup() {
  Serial.begin(115200);               // Inisialisasi Serial
Monitor
  Blynk.begin(auth, ssid, pass);      // Inisialisasi koneksi
Blynk
}

void loop() {
  Blynk.run();                        // Jalankan Blynk

  int ldrValue = analogRead(LDR_PIN);  // Baca nilai dari sensor
LDR
  float voltage = (ldrValue * 3.3) / 4095.0; // Konversi nilai
ADC ke tegangan

  // Kirim data ke Blynk Virtual Pins
  Blynk.virtualWrite(V34, ldrValue);   // Nilai LDR
  Blynk.virtualWrite(V1, voltage);     // Tegangan LDR

  // Tampilkan data di Serial Monitor
  Serial.print("LDR value: ");
  Serial.print(ldrValue);
  Serial.print(" | Voltage: ");
  Serial.println(voltage);

  delay(1000);                        // Tunggu 1 detik
}
```

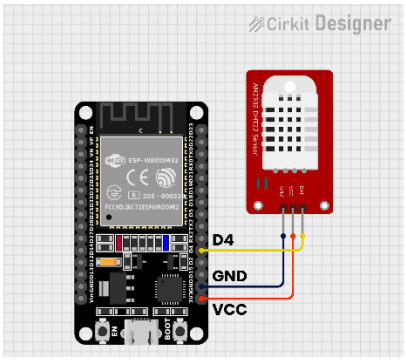
u. Connect ESP32 to laptop, before Verify or Compile code make sure the ESP32 Board is connected to the Arduino IDE, then upload the program.

v. Check the results via Blynk , make sure the device Online status , see Light Intensity value and LDR value in Widget .



6.3 Monitoring DHT from Serial Monitor

a. Create a circuit like the following image. Use Digital pins , because DHT uses digital input .



f. Enter the following code.

```
// Import DHT sensor library
#include <DHT.h>

// Pin dan tipe sensor DHT
#define DHTPIN 4        // Pin data DHT terhubung ke GPIO4 ESP32
#define DHTTYPE DHT11   // Tipe sensor: DHT11

// Inisialisasi objek DHT
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  Serial.println("Starting DHT11 Sensor...");
  dht.begin(); // Mulai sensor DHT
}

void loop() {
  float temperature = dht.readTemperature(); // Baca suhu dalam
°C
  float humidity = dht.readHumidity();       // Baca kelembaban
dalam %

  // Periksa apakah pembacaan valid
```

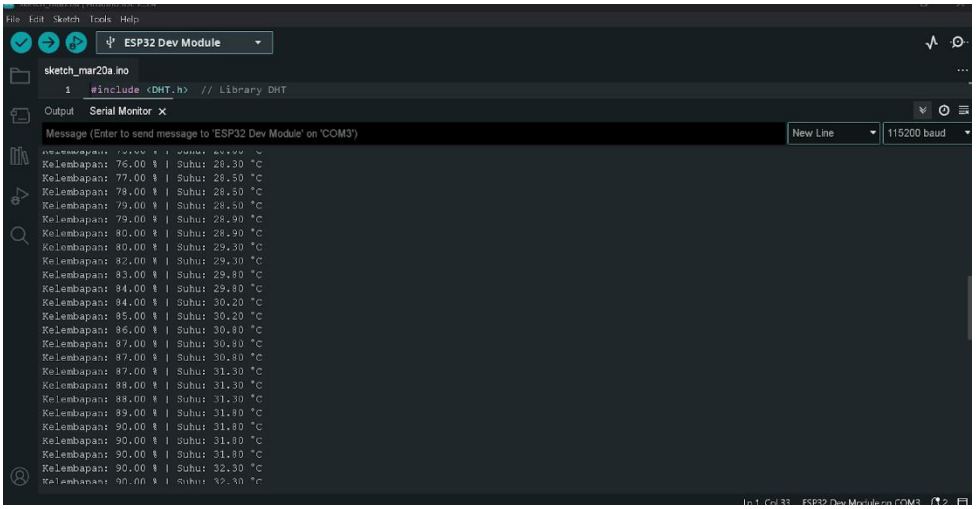
```
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT11!");
  } else {
    // Tampilkan hasil ke Serial Monitor
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C | Humidity: ");
    Serial.print(humidity);
    Serial.println(" %");
  }

  delay(2000); // Tunggu 2 detik sebelum membaca kembali
}
```

g. Arduino IDE requires a DHT22 sensor library to make it easier to read the DHT22 sensor without having to write digital communication code manually.

h. Connect ESP32 to laptop, before Verify or Compile code make sure the ESP32 Board is connected to the Arduino IDE.

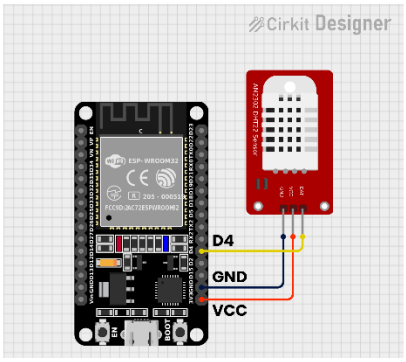
i. If the compilation process is complete and there are no errors , then upload the program, then wait for it to finish.

j. Enter the Serial Monitor as shown in the following image.



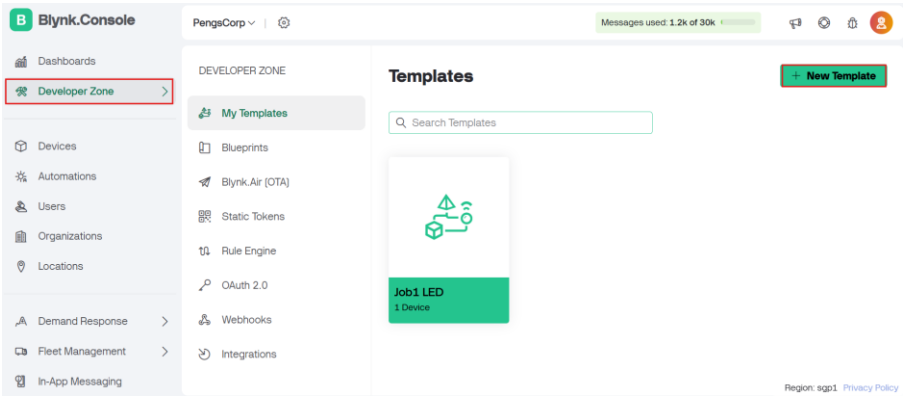
6.4 DHT monitoring from Blynk

a. Create a circuit like the following image. Use Digital pins , because DHT uses digital input .

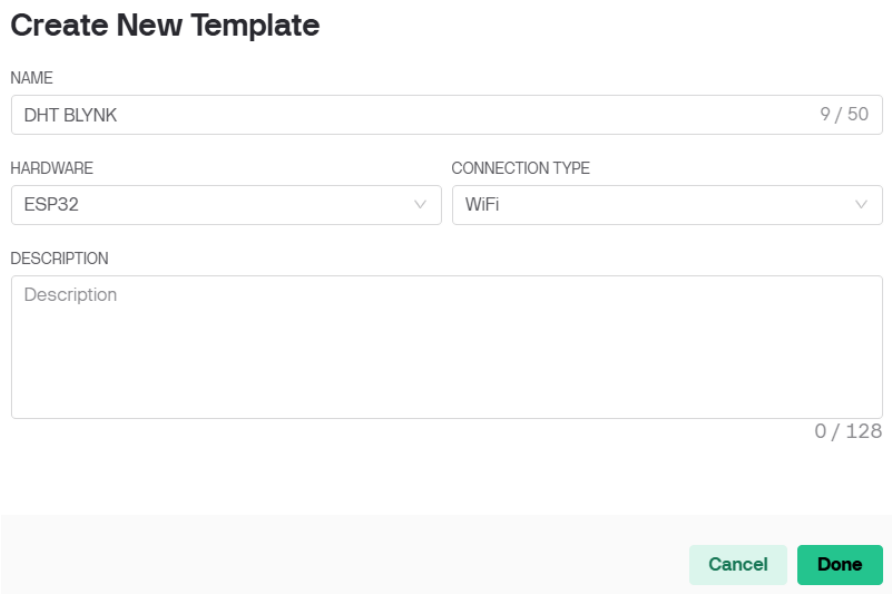


b. Blynk configuration is required so that it can be connected to the ESP32.

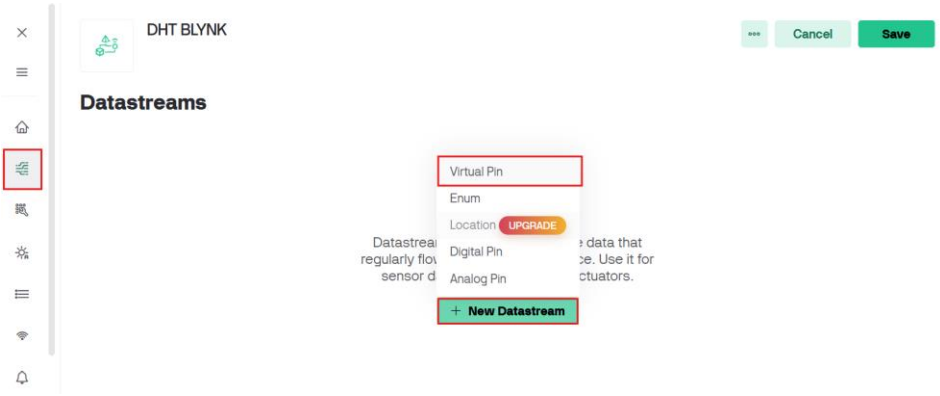
c. Go to the **Developer Zone menu** > click **New Template** .



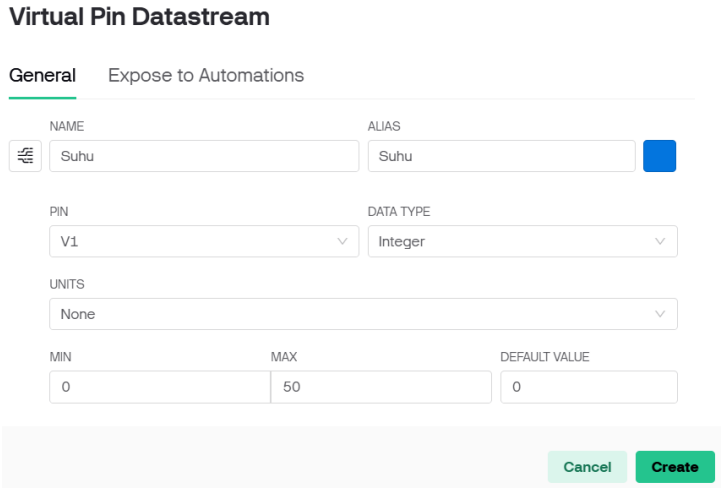
d. Fill in as needed, description is optional.



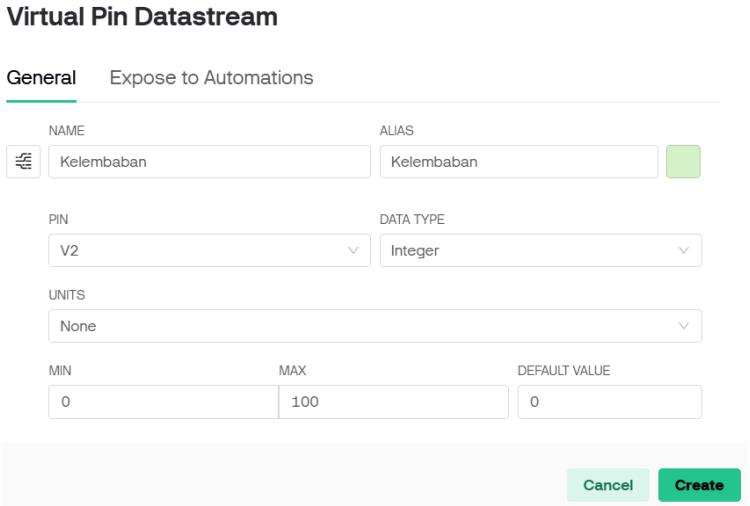
e. Next , go to **the Datastreams menu** > click **New Datastream** > select **Virtual Pin** .



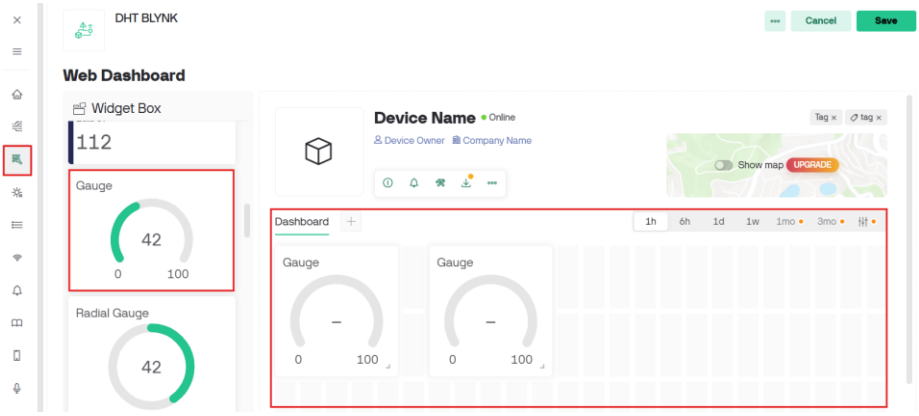
f. Enter the Temperature Datastream according to your needs, for **MIN** and **MAX values** in the form of temperatures from 0 - 50° C.



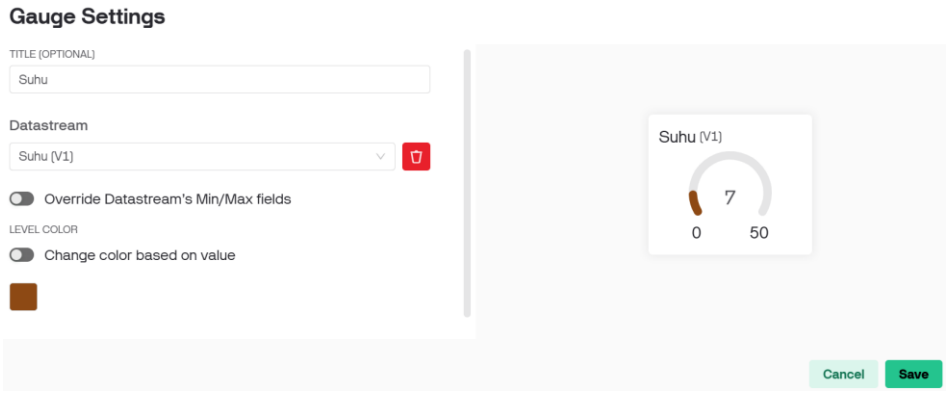
g. Then add another Datastream for Humidity, for MIN and MAX values from 0 – 100%.



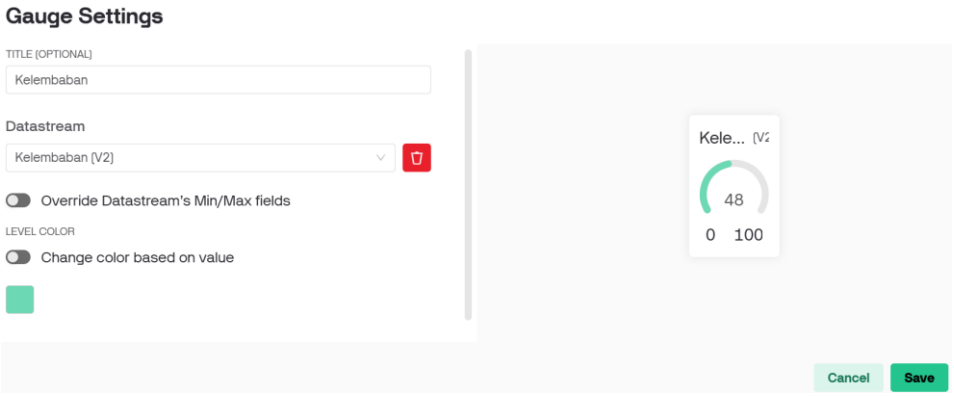
h. Next, configure the **Web Dashboard** menu , drag and drop **the Widget** to the right side **of the Dashboard** , two for Temperature and Humidity.



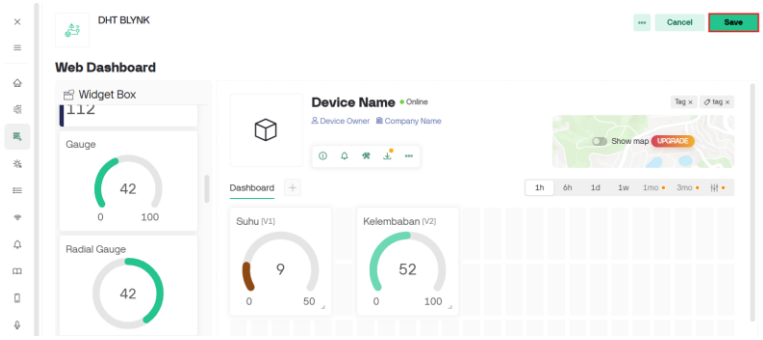
i. Gauge 1 configuration with Datastream source **Temperature (V1)** , then click **Save** .



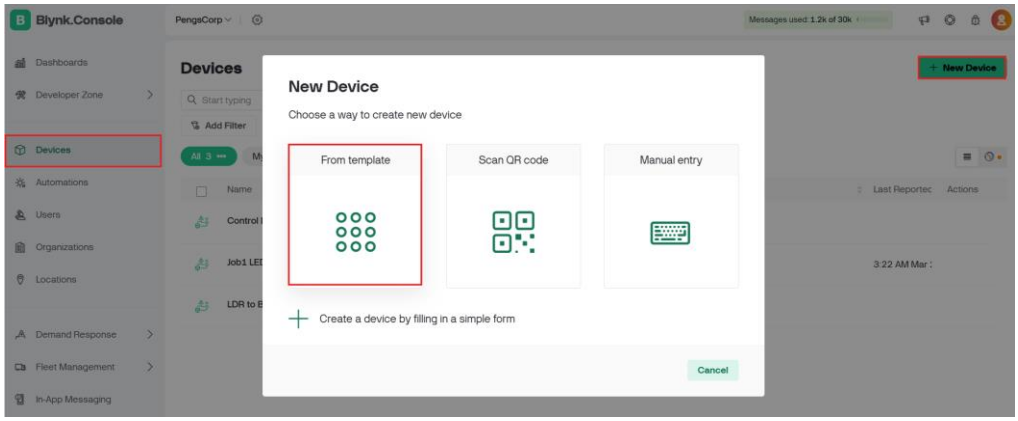
j. Next configure Gauge 2 with Datastream source **Humidity (V2)** , then click **Save** .



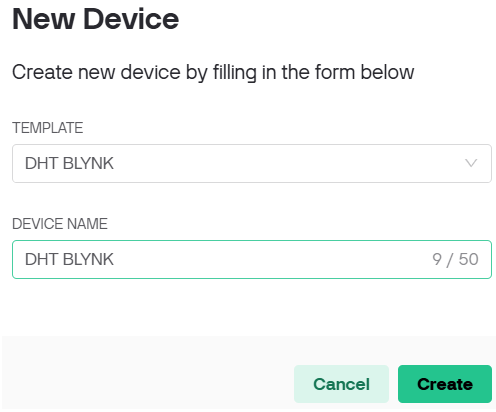
k. When you have finished configuring the **Web Dashboard ,** click **Save** .



l. Next, go to **the Devices menu** to set up the virtual device. Click **New Device** > select **From template** .



m. In the **TEMPLATE column,** select the template that was created in the initial stage, namely **DHT BLYNK** , if so, click Create .



n. Save the code for configuration in Arduino IDE.

o. The next step, configure the program via Arduino IDE, enter the following code.

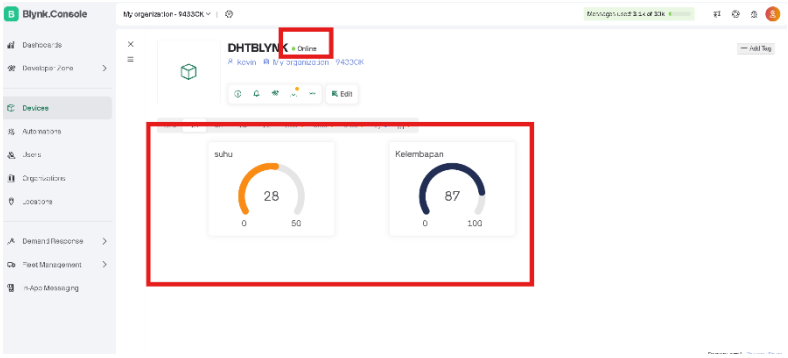
```
#define BLYNK_TEMPLATE_ID "TMPL6ubmFc1C1"
#define BLYNK_TEMPLATE_NAME "DHT BLYNK"
#define BLYNK_AUTH_TOKEN "qDD37SGObXCRpfPMwRi7B3eFMBw6FMDe"

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

// Konfigurasi pin dan tipe sensor DHT
#define DHTPIN 4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);  // Inisialisasi objek DHT

// Konfigurasi WiFi
char ssid[] = "fxx";
char pass[] = "11111111";

// Timer untuk pengiriman data berkala
BlynkTimer timer;

void setup() {
  Serial.begin(115200);
  Serial.println("Starting DHT Sensor...");

  dht.begin();  // Mulai sensor DHT
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); // Mulai koneksi ke
Blynk
  timer.setInterval(2000L, sendDataToBlynk); // Kirim data
setiap 2 detik
}

void loop() {
  Blynk.run();
  timer.run();
}

void sendDataToBlynk() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

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

  // Kirim data ke Virtual Pin di Blynk
  Blynk.virtualWrite(V0, temperature); // Suhu ke V0
  Blynk.virtualWrite(V1, humidity);    // Kelembaban ke V1

  // Tampilkan di Serial Monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
```

```
  Serial.print(" °C | Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");
}
```

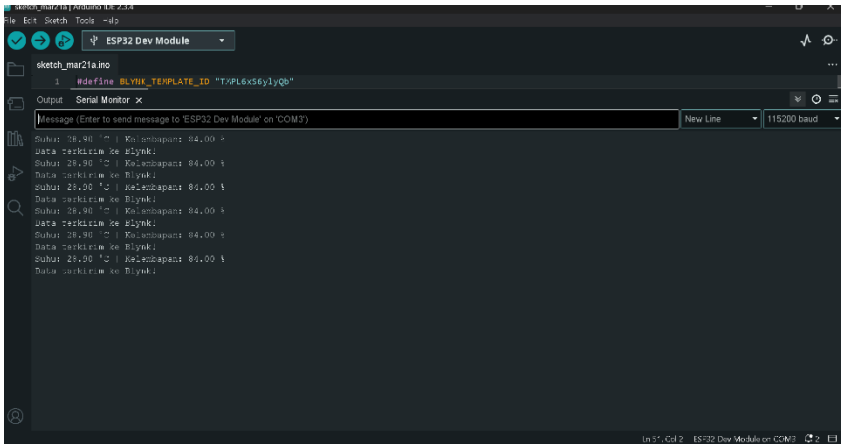
p. Connect ESP32 to laptop, before Verify or Compile code make sure the ESP32 Board is connected to the Arduino IDE.

q. If the compilation process is complete and there are no errors , then upload the program, then wait for it to finish.

r. To see the results, check on the website Blynk . If it is connected then the status will be Online.



s. You can also check the results via Serial Monitor, by setting the baud rate to 115200.



F. Question

1. Show output based on Work Step instructions! Add analysis and working principle!
2. Change the widget on the DHT or LDR sensor monitoring to your liking, use the resources on Blynk Widget Box!
3. Document the results of the work on the Practical Worksheet. Include a video of the practical results and source the code !
4. Upload the Practical Report worksheet (in PDF format) and video documentation on Google Drive . https://drive.google.com/drive/folders/1QWANpOpyaCtYx6A11s2VY5WKJCsV4Sbj?usp=sharing
5. The conditions for collecting drives are as follows:
   - UNNES student account
   - Create a folder with the format " NIM_Full Name "
   - Upload the Practical Report worksheet and video documentation