

CHAPTER 7

BLYNK DOWNLINK SYSTEM DATA COMMUNICATION

A. Objective

1. Understanding the concept of downlink in IoT , namely the process of receiving data or commands from the server to the ESP32 device via the Blynk platform .
2. Implement a downlink system by configuring the ESP32 to receive and execute commands from the Blynk application .
3. Blynk functions and features , such as `Blynk.virtualRead ()` or `Blynk.syncVirtual ()`, to remotely control IoT devices.
4. Analyze the effectiveness of the downlink system in IoT by testing the device's response to commands sent from Blynk.

B. Basic Theory

In data communication, downlink is the process of receiving data or commands from a server to a device. Downlink allows a server or application to send instructions to a connected device, such as an ESP32 to control actuators or change system parameters remotely. To implement a downlink system in Blynk , you need:

- Internet Connection & Blynk Server – The ESP32 device must be connected to the Blynk server using Wi-Fi .
- Authentication Token – Used to connect the ESP32 to a project in the Blynk app .
- Downlink Function in Blynk :
 - `BLYNK_WRITE(Vx)` – Used to read values from Virtual Pins controlled via the Blynk application .
 - `Blynk.syncVirtual (Vx)` – Used to synchronize the last state of the Virtual Pin when the device reconnects.

C. Tools & Materials

- a. ESP32
- b. LED or Buzzer

- c. Relay
- d. Breadboard
- e. Jumper cables
- f. Laptop installed Arduino IDE and libraries
- g. USB Type C Cable

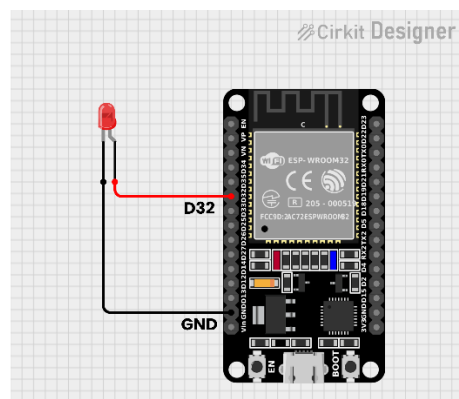
D. Work safety

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 safety.

E. Work steps

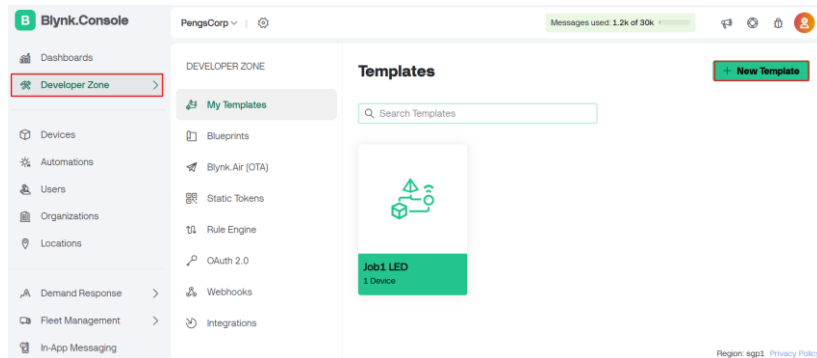
6.1 LED Control from Blynk

- a. Create a circuit like the following image.



- b. Configure Blynk first to create a virtual platform.

c. Go to the **Developer Zone** menu then select **New Template** .



d. Fill in according to your needs, if you have, click Done.

Create New Template

NAME
Control LED 11 / 50

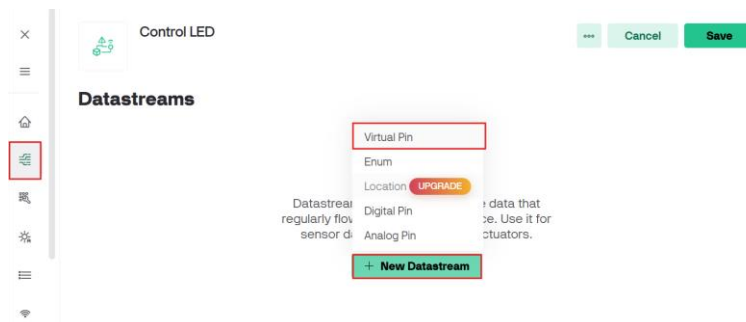
HARDWARE
ESP32

CONNECTION TYPE
WiFi

DESCRIPTION
Description 0 / 128

Cancel Done

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



f. Enter according to your needs, because LEDs work digitally, namely only 0 and 1. So the settings in the MIN (0) and MAX (1) columns are different from analog actuators .

Virtual Pin Datastream

General Expose to Automations

NAME: LED 1 ALIAS: LED 1

PIN: V0 DATA TYPE: Integer

UNITS: None

MIN: 0 MAX: 1 DEFAULT VALUE: 0

Cancel Create

- g. Next, go to **the Dashboard menu** to create a control panel. Select **Switch** > drag and drop it into **the Dashboard area** on the right.

Control LED

Web Dashboard

Widget Box

CONTROL

Switch

Slider

Number Input

Dashboard

Device Name Online

Device Owner Company Name

Show map UPGRADE

1h 6h 1d 1w 1mo 3mo

Switch

- h. Configure the **Switch button** by clicking the settings icon .

Dashboard

Sw

Settings icon

Switch

- i. Configure according to your needs, in **the Datastream column** select the datastream that has been configured earlier. If so, click **Save** .

Switch Settings

TITLE (OPTIONAL): LED 1

Datastream: LED 1 (V0)

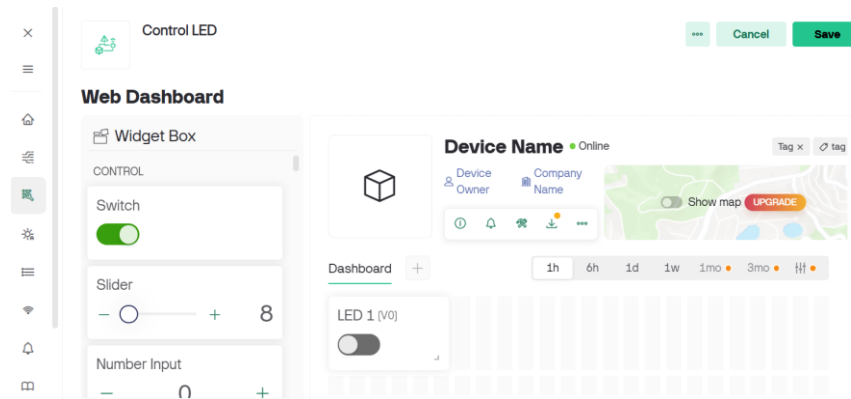
ON VALUE: 1 OFF VALUE: 0

Show on/off labels

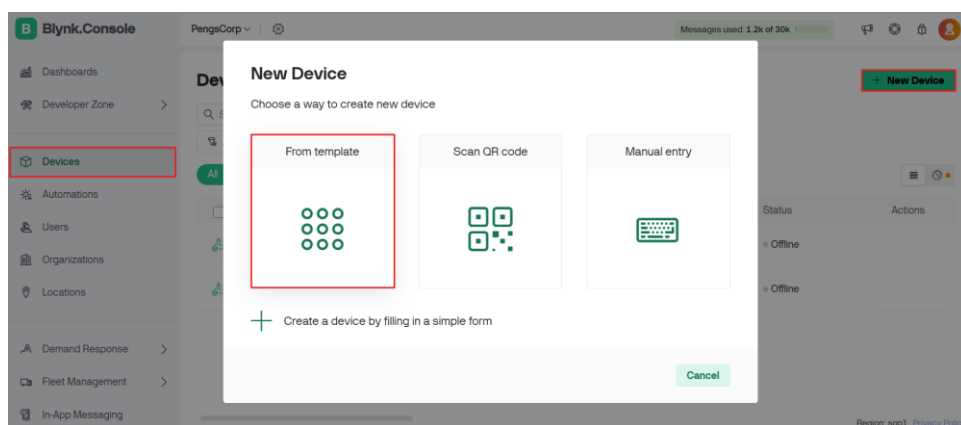
Hide widget name

Cancel Save

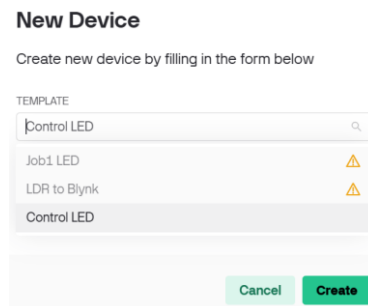
- j. Web Dashboard is now configured. Click Save on the top right.



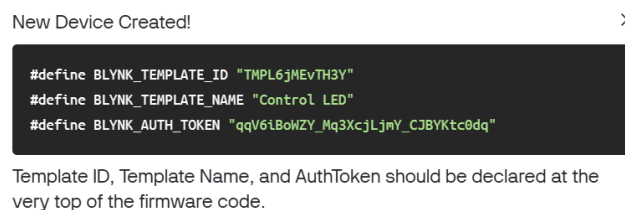
- k. Next, go to the Devices menu to set up the virtual device . Click **New Device** > select **From template** .



- l. Select the template that has been configured, then click **Create** .



- m. So a code will appear, save the code to be configured via Arduino IDE.



- n. Next, configure the program in Arduino IDE. Enter the following code.

```
#define BLYNK_TEMPLATE_ID "TMPL60vDZ2M_G" // Ganti sesuai
Template ID Anda
```

```

#define BLYNK_TEMPLATE_NAME "job2Led"           // Ganti sesuai
nama template
#define BLYNK_AUTH_TOKEN "OZLBErmShKMZD_8AxQV2LfYNpxrIE5No" //
Ganti dengan Auth Token Anda

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

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

// Pin LED di ESP32
#define LED_PIN 32                // Gunakan pin sesuai dengan
wiring Anda

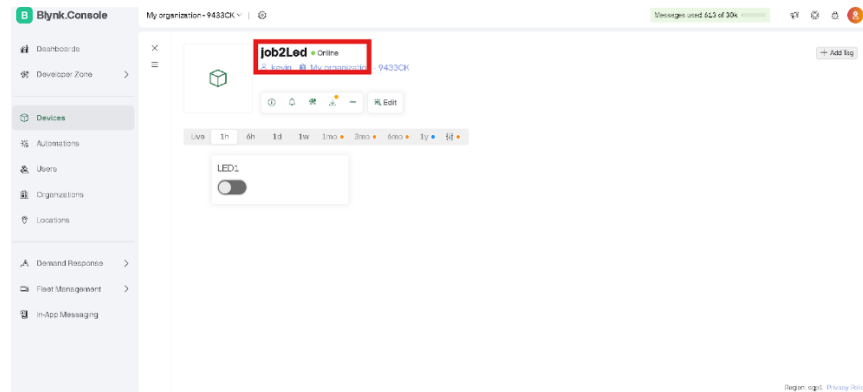
// Fungsi kontrol LED dari Blynk (Virtual Pin V0)
BLYNK_WRITE(V0) {
    int status = param.asInt();    // Baca status tombol (0 =
    mati, 1 = nyala)
    digitalWrite(LED_PIN, status); // Atur LED sesuai status
    Serial.print("LED: ");
    Serial.println(status ? "NYALA" : "MATI");
}

void setup() {
    Serial.begin(115200);
    pinMode(LED_PIN, OUTPUT);      // Set pin sebagai OUTPUT
    digitalWrite(LED_PIN, LOW);    // Pastikan LED mati saat
    boot
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); // Inisialisasi
    Blynk
}

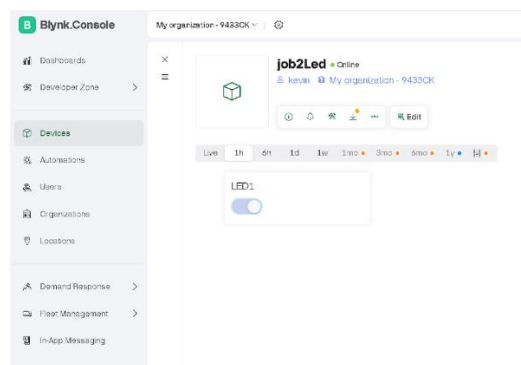
void loop() {
    Blynk.run();                  // Jalankan proses Blynk
}

```

- o. Connect the ESP32 to the laptop, before Verifying or Compiling the code make sure the ESP32 Board is connected in Arduino IDE, then Upload.
- p. After Successful upload, login to Blynk if Already connected then the status will be changed become **On line** .

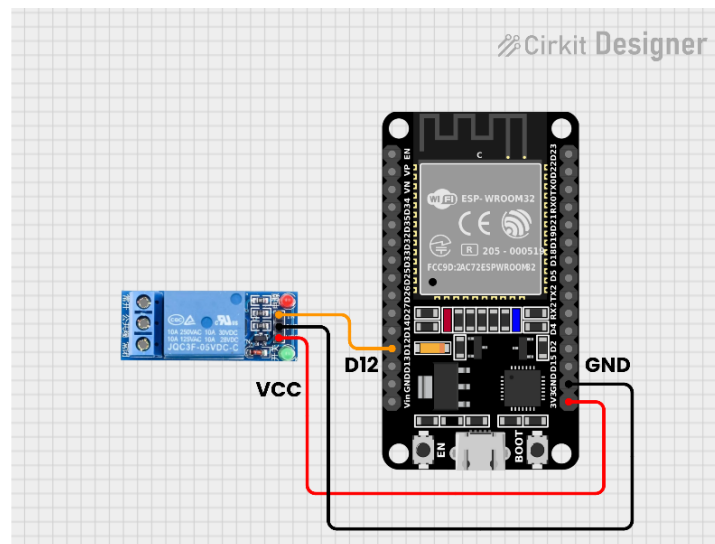


q. Experiment to turn the LED on or off via Blynk by clicking on the Switch button .



6.2 Relay Control from Serial Monitor

a. Create a sequence like the following image.



b. Enter the following code in the Arduino IDE.

```
// Pin Relay
#define RELAY_PIN 4 // Sesuaikan dengan pin yang digunakan

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

```

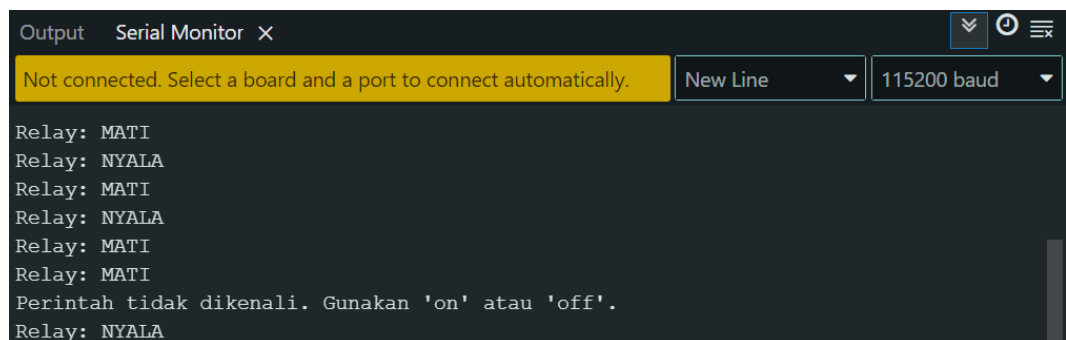
pinMode(RELAY_PIN, OUTPUT);
digitalWrite(RELAY_PIN, LOW); // Matikan relay saat boot
Serial.println("Ketik 'on' untuk menyalakan relay, atau 'off'
untuk mematikan relay.");
}

void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim(); // Hapus spasi dan karakter newline

    if (input.equalsIgnoreCase("on")) {
      digitalWrite(RELAY_PIN, HIGH);
      Serial.println("Relay: ON");
    } else if (input.equalsIgnoreCase("off")) {
      digitalWrite(RELAY_PIN, LOW);
      Serial.println("Relay: OFF");
    } else {
      Serial.println("Perintah tidak dikenali. Gunakan 'on' atau
'off'.");
    }
  }
}

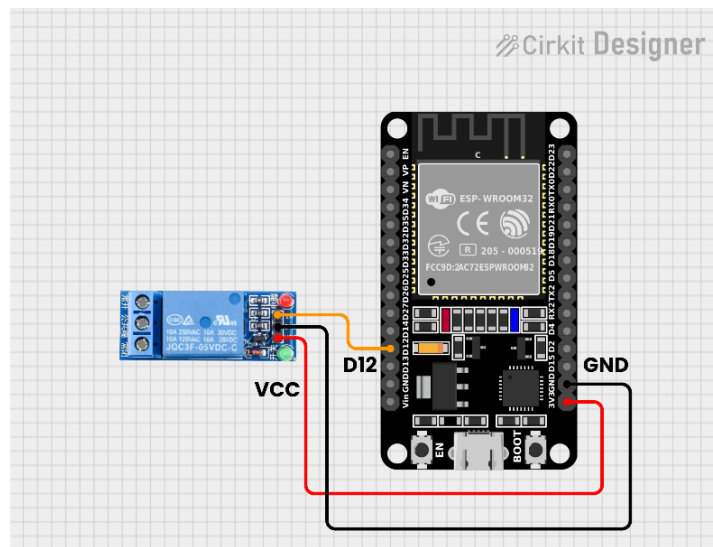
```

- 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 Serial Monitor, then change the baud rate according to what has been programmed.
- f. Type the command “ON” or “OFF” as configured, to turn the LED on or off. After typing the command, click enter .

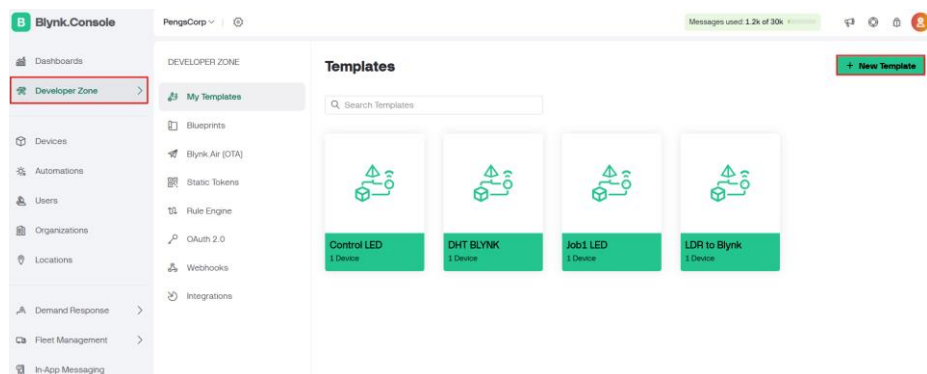


6.3 Relay Control from Blynk

- a. Create a circuit like the following image.



- b. Configure Blynk first to create a virtual platform.
c. Go to the **Developer Zone** menu > click **New Template** .



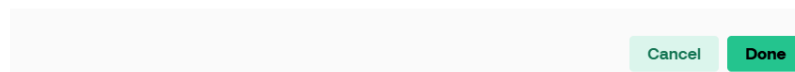
- d. Fill in according to your needs, then click **Done** .

Create New Template

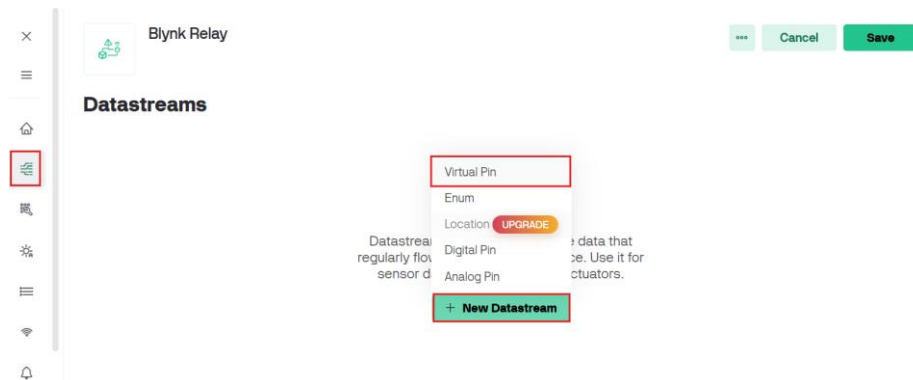
NAME
Blynk Relay 11 / 50

HARDWARE ESP32 CONNECTION TYPE WIFI

DESCRIPTION
Description 0 / 128



- e. Then configure in the **Datastreams** menu , click **New Datastream** > then select **Virtual Pin** .



- f. Fill in according to your needs, because the Relay works digitally, fill in the **MIN (0)** and **MAX (1)** values .

Virtual Pin Datastream

General Expose to Automations

NAME: Relay ALIAS: Relay

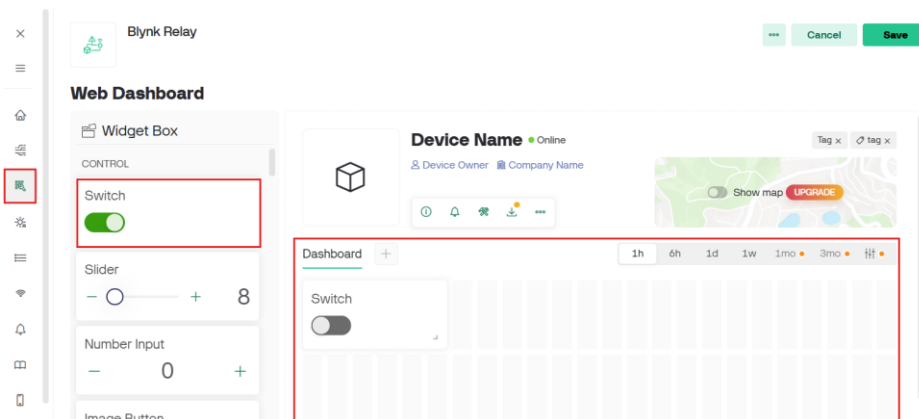
PIN: V1 DATA TYPE: Integer

UNITS: None

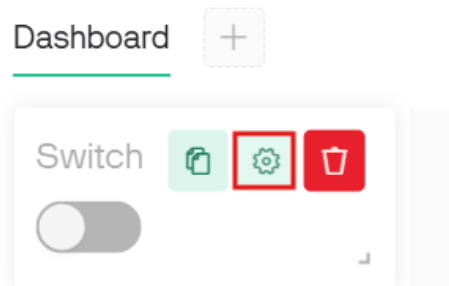
MIN: 0 MAX: 1 DEFAULT VALUE: 0

Cancel Create

- g. Next, configure the **Web Dashboard** menu , select the **Widget** according to your needs > then drag and drop to the **Dashboard** on the right.



- h. Widget Configuration **Switch** by clicking the settings icon as shown in the picture.



- i. Enter **the TITLE** as needed, change **the Datastream source** to **Relay (V1)** as configured. Change the **ON VALUE column** to 1 and the **OFF VALUE column** to 0. If so, click **Save** .

Switch Settings

TITLE (OPTIONAL)
Relay

Datastream
Relay (V1)

ON VALUE
1

OFF VALUE
0

☒ Show on/off labels

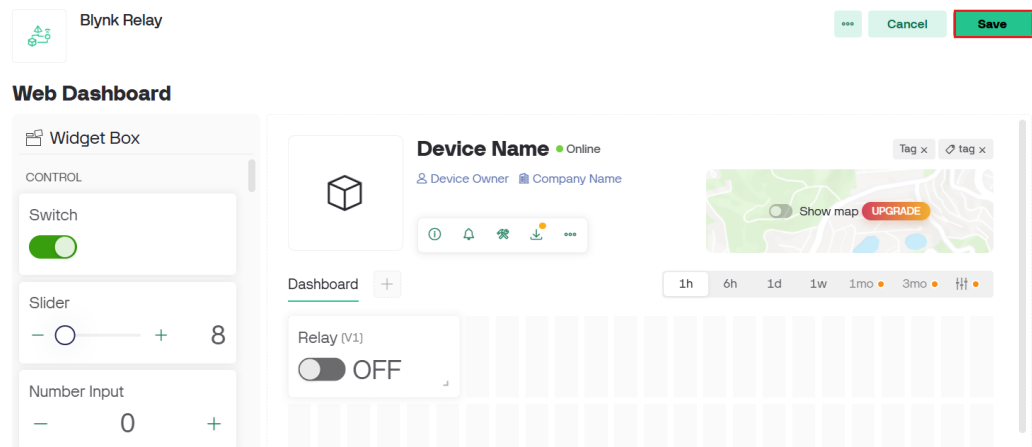
ON LABEL
ON

OFF LABEL
OFF

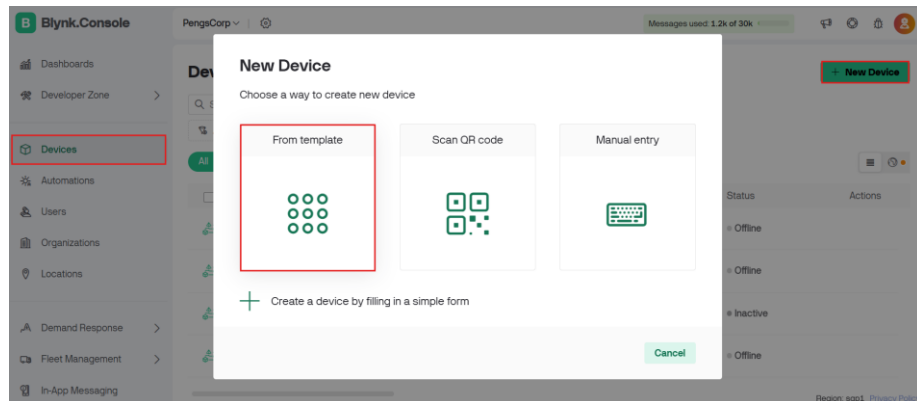
LABEL POSITION

Cancel Save

- j. **Web Dashboard** configuration is complete, click **Save** .



- k. After configuring the Web Dashboard , the next step is to add a Device .
Go to **the Devices menu** > click **New Device** > select **From Templates** .



- l. In the **TEMPLATE** column, select the template that was configured earlier (Blynk Relays).

New Device

Create new device by filling in the form below

TEMPLATE
Blynk Relay

DEVICE NAME
Blynk Relay 11 / 50

Cancel Create

- m. Save the following code to configure in Arduino IDE.

New Device Created!

```
#define BLYNK_TEMPLATE_ID "TMPL6z9CHX-h7"
#define BLYNK_TEMPLATE_NAME "Blynk Relay"
#define BLYNK_AUTH_TOKEN "86_7DH38A9CmLvLY1cWavD8tU400iY-j"
```

- n. Next, configure the program in Arduino IDE, enter the following code.

```
#define BLYNK_TEMPLATE_ID "TMPL6j44liV1G" // Template
ID dari Blynk
#define BLYNK_TEMPLATE_NAME "blynkrely" // Nama
template Blynk
#define BLYNK_AUTH_TOKEN "9xtjeZJ-82CiVSKjWNDAXirniylY8ebv" //
Token autentikasi Blynk

#include <WiFi.h> // Library koneksi WiFi untuk
ESP32
#include <BlynkSimpleEsp32.h> // Library Blynk untuk ESP32

// Konfigurasi WiFi
char ssid[] = "Jackfruit"; // Ganti dengan SSID WiFi Anda
char pass[] = "dikap999"; // Ganti dengan password WiFi
Anda
```

```

// Pin Relay
#define RELAY_PIN 12 // Gunakan GPIO12 untuk relay

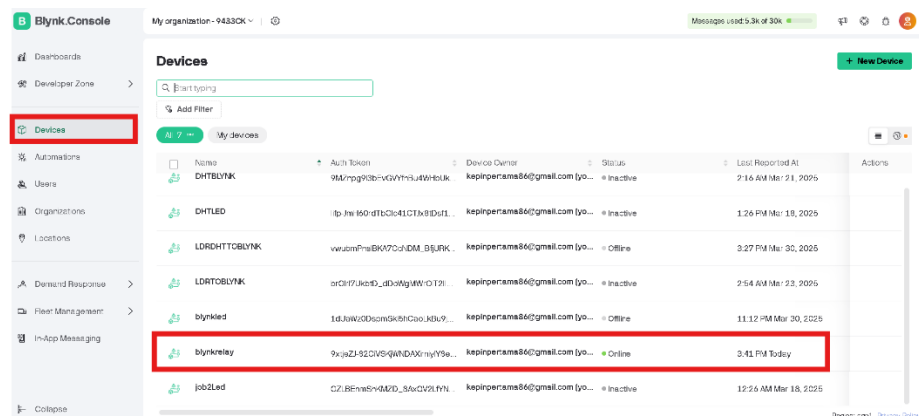
void setup() {
    Serial.begin(115200); // Inisialisasi
    komunikasi serial
    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); // Hubungkan ke
    WiFi dan Blynk
    pinMode(RELAY_PIN, OUTPUT); // Atur pin relay
    sebagai output
    digitalWrite(RELAY_PIN, LOW); // Matikan relay
    saat awal
}

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

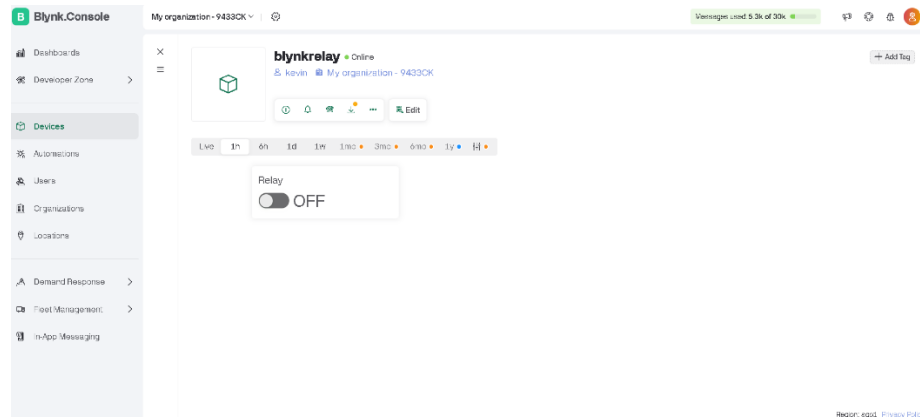
// Fungsi kontrol relay dari aplikasi Blynk (melalui Virtual Pin
V1)
BLYNK_WRITE(V1) {
    int status = param.asInt(); // Baca status
    tombol (0 = mati, 1 = nyala)
    digitalWrite(RELAY_PIN, status); // Kontrol relay
    sesuai input
    Serial.print("Relay: ");
    Serial.println(status ? "ON" : "OFF"); // Tampilkan status
    relay di Serial Monitor
}

```

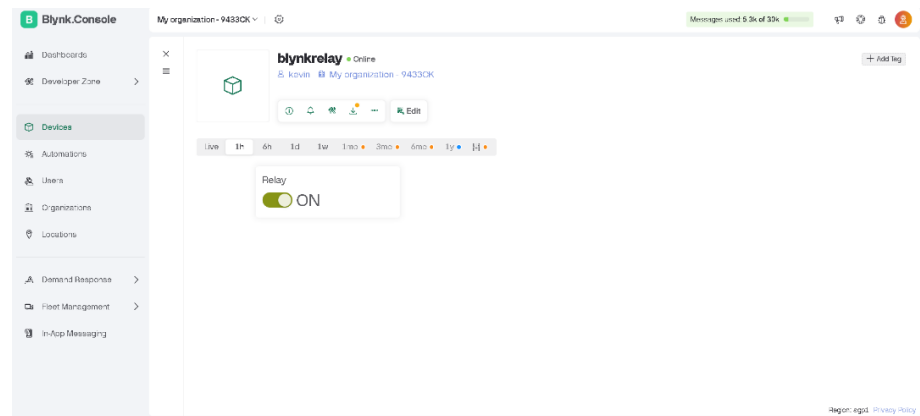
- o. Connect ESP32 to laptop, before Verify or Compile code make sure the ESP32 Board is connected to the Arduino IDE, then Upload .
- p. Check the results via Blynk , go to the **Devices menu** then select **Relay Blynk** .



- q. Once connected, the status will be Online.

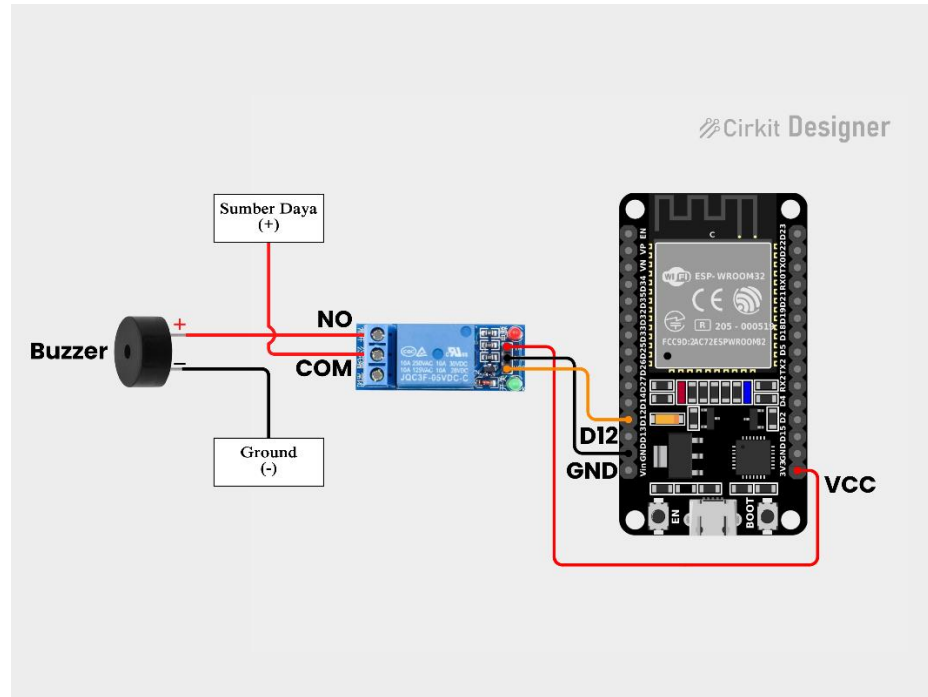


- r. When the switch button is changed to ON, the relay will be active and vice versa.



F. Question

1. Show output based on the Work Step instructions! Add analysis and working principles to the lab report!
2. Modify the relay circuit with a combination of actuators . So that it can control the actuator through the relay ! Use LED or Buzzer actuators !



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/1X2R15dGXmTuoX5a81Ebzub6cuEzUz1AB?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