BLYNK UPLINK AND DOWNLINK SYSTEM DATA COMMUNICATION

A. Objective

1. Understanding the concept of *uplink* and *downlink integration* in two-way communication in Blynk and ESP32 based IoT systems.
2. Implementing a combination of *uplink* and *downlink to build a cloud* - based control and monitoring system .
3. Using sensors, actuators and programming functions in one integrated system.
4. Analyze the performance of a two-way IoT system against control commands and the accuracy of sensor data sent to the Blynk server .

B. Basic Theory

In the application of *the Internet of Things Things* (IoT), the ability of devices to communicate two-way becomes important. The combination of *uplink* and *downlink systems* allows devices not only to send data to the server (*uplink*), but also to receive commands from the server or user (*downlink*).

In ESP32 and Blynk based systems , *the uplink* is used to send data from the device to the application, such as sensor value data. While *the downlink* is used to receive commands from the application to the device, such as activating an LED or pump as in Figure 1.

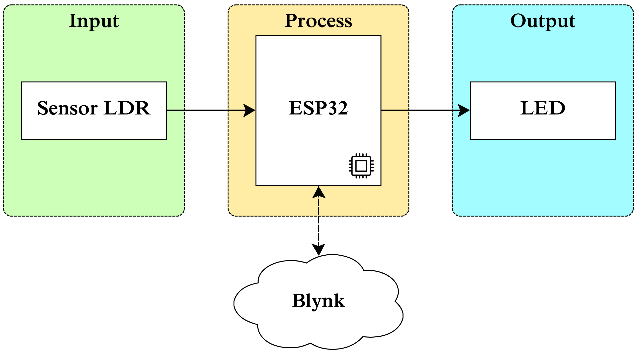


Figure 1. System Block Diagram

Through a combination of *uplink* and *downlink* the system can:

- *Real- time* data monitoring via the Blynk interface .
- Control devices directly based on user commands via Blynk .
- System automation through sensors and actuators , for example turning on a fan if the temperature exceeds a certain threshold.

C. Tools & Materials

a. ESP32
b. LED
c. Relay
d. LDR Sensor
e. Breadboard
f. Jumper cables
g. Laptop installed Arduino IDE and libraries
h. USB Type C Cable

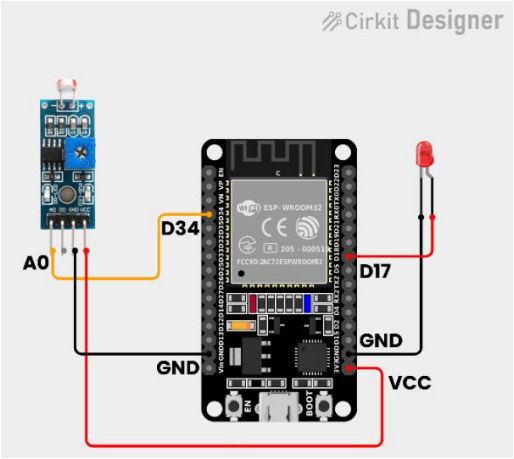
D. Work safety

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

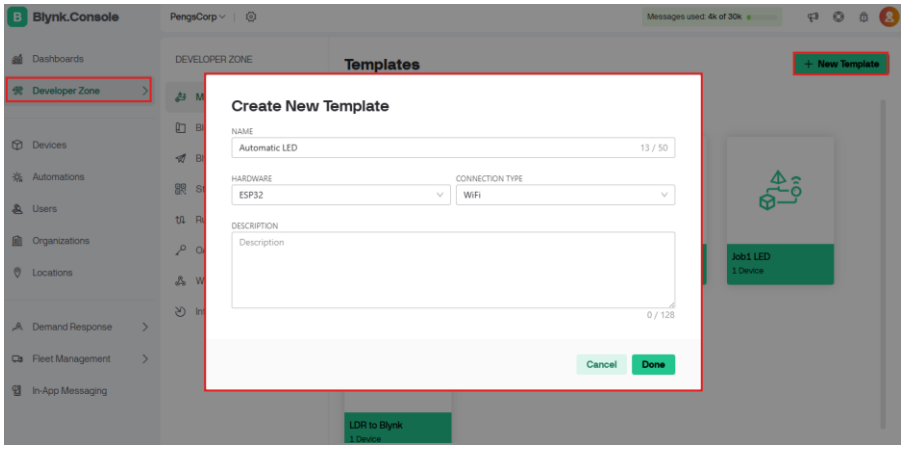
8.1 Automatic Lighting System with LDR Sensor

a. Create a circuit like the following image.

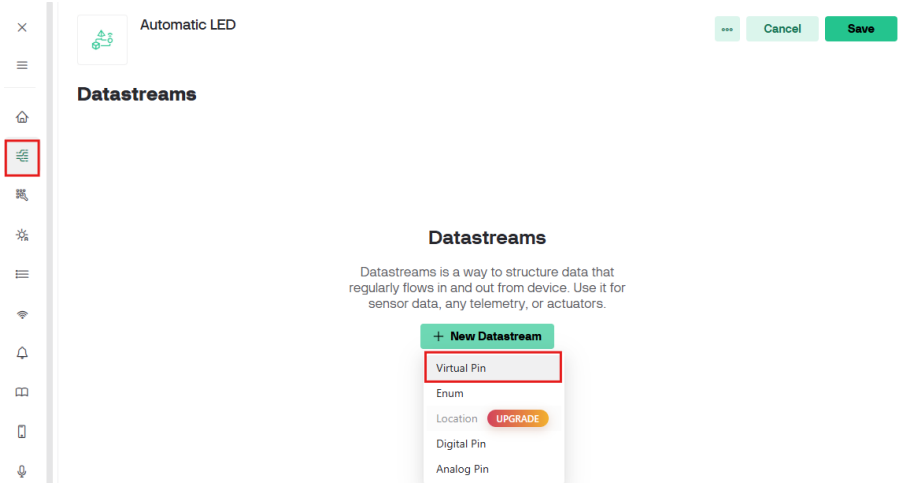


b. Configure Blynk first to create *a virtual* platform.

c. Go to the **Developer Zone menu** then select **New Template** , fill in according to your needs then click **Done** .



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

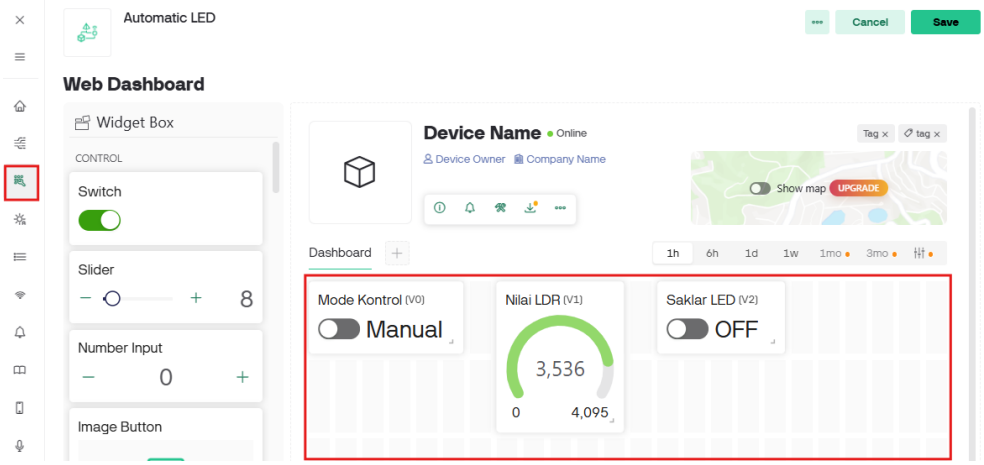


e. Create Virtual Pins according to your needs, for this system three virtual pins are used as shown in Table 1.

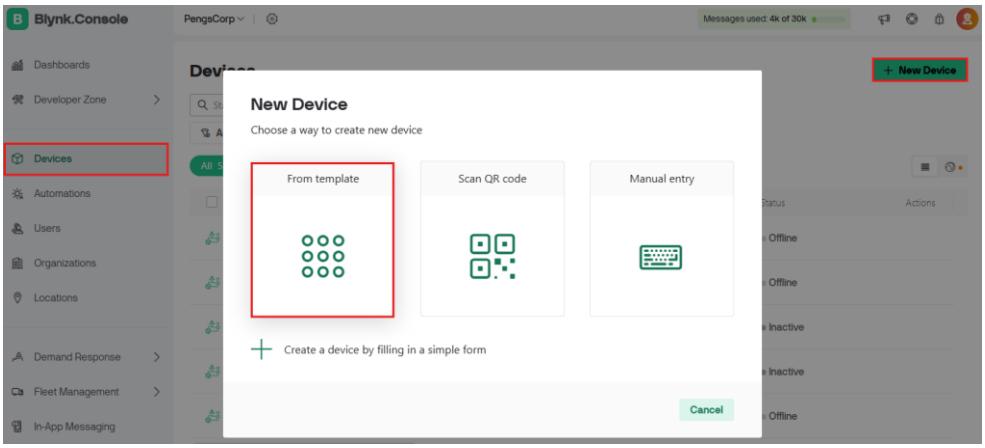
Table 1Virtual pin configuration blink

Name	Pin	Data Type	Value	Function
Control Mode	V0	Integer	0 - 1	Select the control mode (manual or automatic)
LDR Value	V1	Integer	0 - 4095	Taking LDR sensor data
LED Switch	V2	Integer	0 - 1	Manual LED control

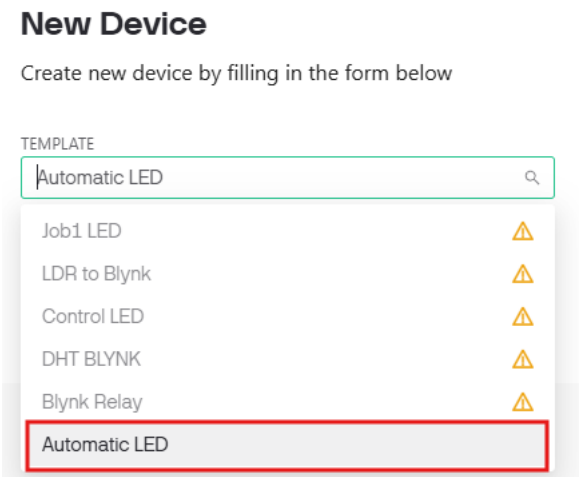
f. Next , go to **the Dashboard menu** to create a *monitoring* and control panel. Select according to your needs as set *through* Datastreams , if so, click **Save** .



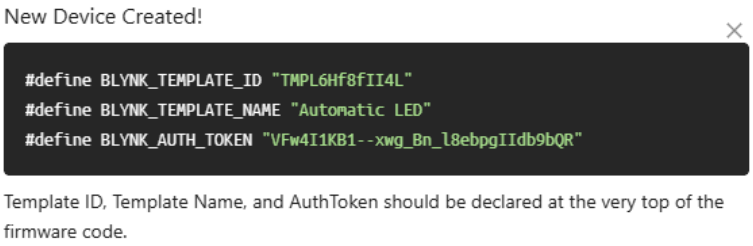
g. Furthermore enter to the Devices menu to arrange device *virtual* . Click **New Device** > select **From template** .



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



i. Save the code to be configured via Arduino IDE.



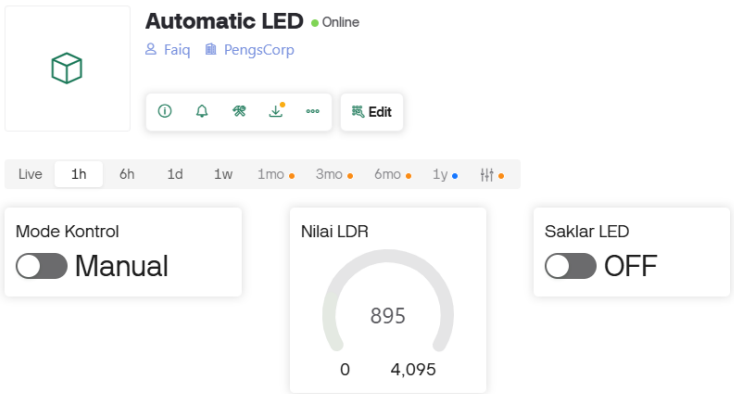
j. Blynk configuration is complete, the next step is to configure the program in the Arduino IDE, enter code following .

```
#define BLYNK_TEMPLATE_ID "TMPL6Hf8fII4L"
#define BLYNK_TEMPLATE_NAME "Automatic LED"
#define BLYNK_AUTH_TOKEN "VFw4I1KB1--xwg_Bn_l8ebpgIIdb9bQR"

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

// Ganti dengan WiFi Anda
char ssid[] = "fxx";
char pass[] = "11111111";

const int ldrPin = 34;
const int ledPin = 17;

int ldrValue = 0;
bool ledState = false;
bool autoMode = true;        // Default: mode otomatis
int manualControl = 0;       // Nilai kontrol manual dari Blynk

#define VPIN_MODE V0         // Virtual pin untuk switch mode
(auto/manual)
#define VPIN_LDR V1          // Virtual pin untuk menampilkan
nilai LDR
#define VPIN_MANUAL V2       // Virtual pin untuk tombol manual
LED

void setup() {
  Serial.begin(9600);
  pinMode(ldrPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

// Mode otomatis/manual dari Blynk (Switch)
BLYNK_WRITE(VPIN_MODE) {
  autoMode = param.asInt();
}

// Kontrol LED manual dari Blynk
BLYNK_WRITE(VPIN_MANUAL) {
```

```
  manualControl = param.asInt();
}

void loop() {
  Blynk.run();

  ldrValue = analogRead(ldrPin);

  // Tampilkan nilai LDR ke Serial dan Blynk
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);
  Blynk.virtualWrite(VPIN_LDR, ldrValue);

  // Logika kontrol LED
  if (autoMode) {
    // Mode otomatis: LED nyala jika terang
    if (ldrValue > 2500) {
      digitalWrite(ledPin, HIGH);
      ledState = true;
    } else {
      digitalWrite(ledPin, LOW);
      ledState = false;
    }
  } else {
    // Mode manual: kontrol LED via tombol Blynk
    digitalWrite(ledPin, manualControl ? HIGH : LOW);
    ledState = manualControl;
  }

  // Tampilkan status LED ke Serial
  Serial.print("LED Status: ");
  Serial.println(ledState ? "ON" : "OFF");

  delay(1000);
}
```

k. Connect the ESP32 to the laptop, before Verifying or Compiling the code make sure the ESP32 Board is connected in Arduino IDE, then Upload.

l. After Successful upload, login to Blynk if Already connected then the status will be changed become **On line** .



F. Question

1. Show output based on the Work Step instructions! Add analysis and working principles to the lab report!
2. Circuit modification using DHT sensor input and output LED actuator (3 LEDs in total), assume the LED is a fan speed according to the following scenario:
   - Temperature <29 output 1 LED
   - Temperature 29 – 34 output 2 LED
   - Temperature >34 output 3 LED

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/1aK1wROwYA5LkRB38jrDirvBtd6uQlPt3?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