Smart Water Management

Development Part-2

Date	**23-10-2023**
Team ID	**536**
Project Name	**Smart Water Management**

Table of Contents

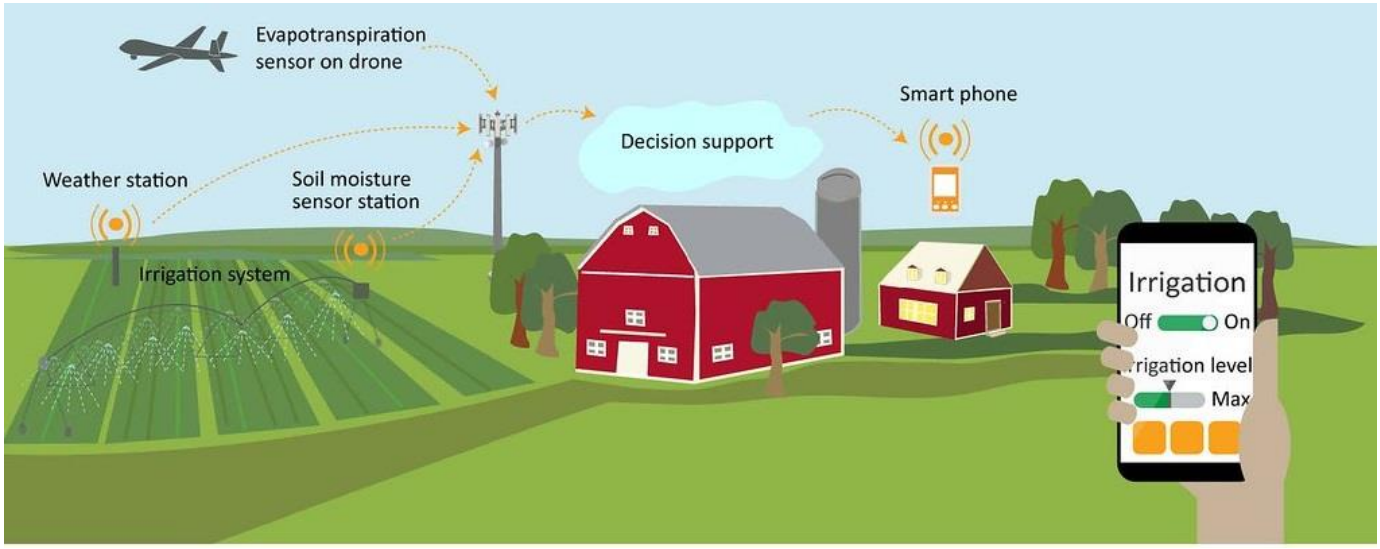
1. Introduction

Smart Water Management (SWM) uses Information and Communication Technology (ICT) and real-time data and responses as an integral part of the solution for water management challenges. SWM is becoming an area of increasing interest as governments from around the world integrate smart principles into their urban, regional and national strategies. The potential application of smart systems in water management is wide and includes solutions for water quality, water quantity, efficient irrigation, leaks, pressure and flow, floods, droughts and much more.

2. Problem Statement

The current water management system in our region suffers from inefficiency, water losses, and a lack of real-time monitoring. The objective is to design, develop, and implement a **"Smart Water Management"** System that addresses these challenges by optimizing water distribution, reducing water wastage, enhancing system resilience, and promoting responsible water use."

3. Block Diagram



4. Code

```
#define BLYNK_TEMPLATE_ID "TMPL3g8NOyuBL"
#define BLYNK_TEMPLATE_NAME "WATER MANAGEMENT"
#define BLYNK_AUTH_TOKEN "keuIMDdFF9hQOue1x8ntz3OuNTInBbD0"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
//Your wifi credentials
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();
```

```
// Update state
  Blynk.virtualWrite(V1, value);
}

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()
{
  // Change Web Link Button message to "Congratulations!"
  Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
  Blynk.setProperty(V3, "onImageUrl",  "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
  Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()
{
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V2, millis() / 1000);
}
#define trigerpin 16
#define echopin 17
#define buzzer 5
float distance,a, duration;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
```

```
  pinMode(trigerpin, OUTPUT);
  pinMode(echopin, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(115200);
}

void loop(){
  digitalWrite(trigerpin, LOW);
  delay(2);
  digitalWrite(trigerpin, HIGH);
  delay(10);
  digitalWrite(trigerpin, LOW);
  duration=pulseIn(echopin,HIGH);
  a=duration/2*0.034;
  Serial.print("distance (in cm)=");
  Serial.println(a);
  Blynk.virtualWrite(V1,a);
  Blynk.virtualWrite(V3, 4);
  if(a>10&&a<30){
    Blynk.virtualWrite(V3, 3);
    Serial.println("level :3");
  }else if(a>30&&a<50){
    Blynk.virtualWrite(V3, 2);
    Serial.println("level :2");
  }else if(a>50&&a<70){
    Blynk.virtualWrite(V3, 1);
    Serial.println("level :1");
  }else if(a>70){
    Blynk.virtualWrite(V3, 0);
    Serial.println("level :0");
```

```
  }


if (a<30){
  digitalWrite(buzzer, HIGH);
  delay(1000);
}
else{
  digitalWrite(buzzer,LOW);
  delay(500);
}
}
```

5. Working Principle

1. Data Collection and IoT Devices:

Set up IoT devices (sensors, flow meters) to monitor water parameters such as water level, quality, and flow rate.

2. Data Processing and Analysis:

Develop software for data processing, analytics, and decision-making algorithms on the server.

Use Python, Node.js, or other server-side languages and libraries to handle data efficiently.

3. Web Development:

Create a web-based platform to monitor and manage the Smart Water Management system.

4. User Interfaces:

Design user-friendly web interfaces for different stakeholders, including administrators, maintenance personnel, and end-users.

Implement responsive web design to ensure usability on various devices.

5. Data Visualization:

Use web-based data visualization libraries such as D3.js, Chart.js, or Plotly to display real-time and historical data through interactive charts and graphs.

Display water quality, consumption trends, and equipment health.

6. Alerts and Notifications:

Set up alert mechanisms to notify stakeholders via web notifications or email about critical water parameters, equipment issues, or consumption anomalies.

7. Remote Monitoring:

Implement remote monitoring of the water management system through a web dashboard, enabling stakeholders to check system status and make adjustments as needed.

8. User Authentication and Security:

Implement secure user authentication mechanisms to control access to system data.Use HTTPS for secure data transmission and ensure that user data remains private.

9. Database Management:

Set up a database system (e.g., MySQL, MongoDB) to store historical data for analysis and reporting.

10. Mobile App Integration :

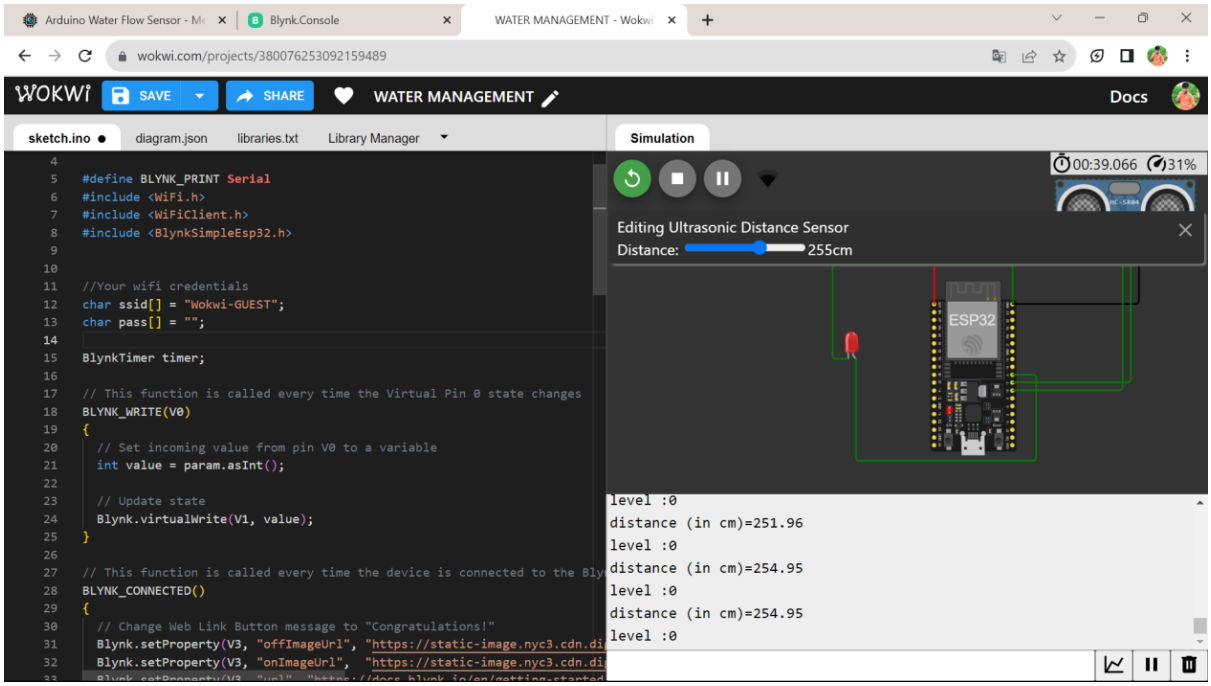
Develop mobile applications for Android and iOS platforms to enable on-the-go monitoring and control of the Smart Water Management system.

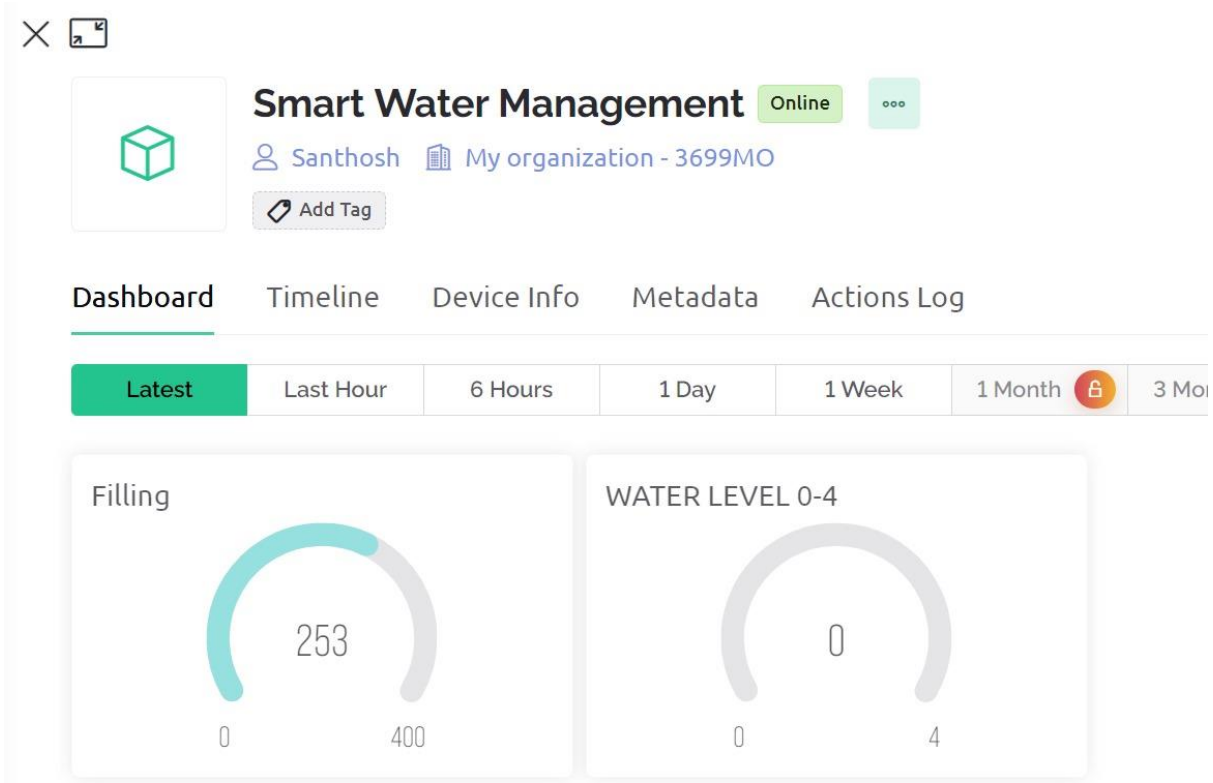
Utilize web technologies, like React Native or Flutter, to build cross-platform mobile apps.

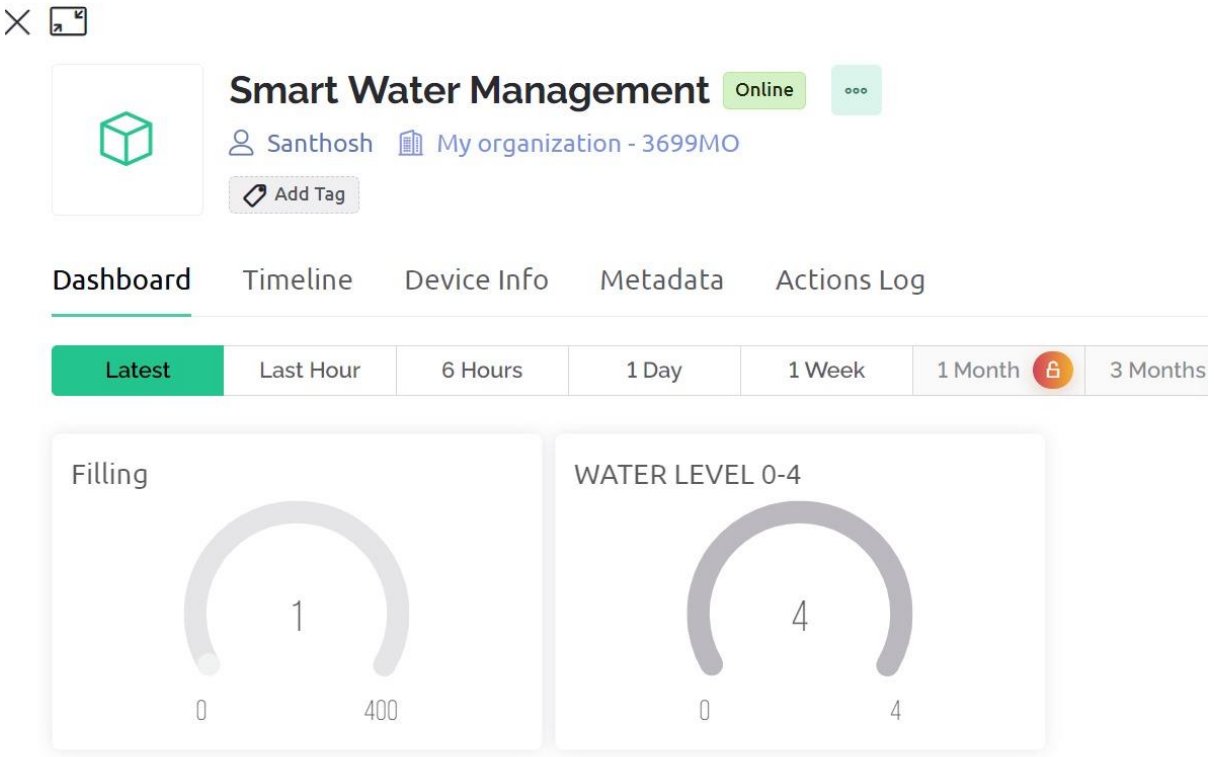
Using C/C++ & BLYNK

The Blue LED indicated the functioning of the Fountain and the Red LED indicates the functioning of refill motor which fills the reservoir.

We have used Blynk app to get live


Arduino Water Flow Sensor - M
Blynk.Console
WATER MANAGEMENT - Wokwi
wokwi.com/projects/380076253092159489
WOKWI
SAVE
SHARE
WATER MANAGEMENT
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
//Your wifi credentials
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
BlynkTimer timer;
// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
// Set incoming value from pin V0 to a variable
int value = param.asInt();
// Update state
Blynk.virtualWrite(V1, value);
}
// This function is called every time the device is connected to the Bly
BLYNK_CONNECTED()
{
// Change Web Link Button message to "Congratulations!"
Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.di
Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.di
Simulation
00:39.066
31%
Editing Ultrasonic Distance Sensor
Distance:
255cm
ESP32
level :0
distance (in cm)=251.96
level :0
distance (in cm)=254.95
level :0
distance (in cm)=254.95
level :0



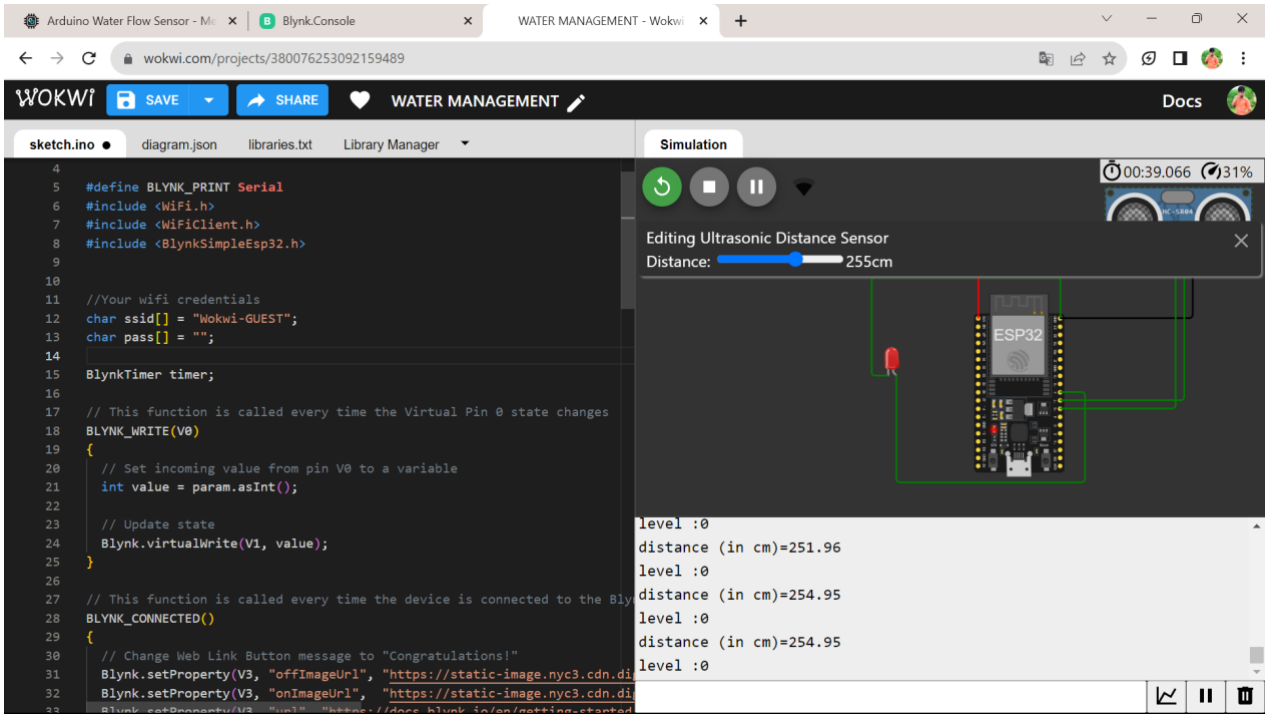


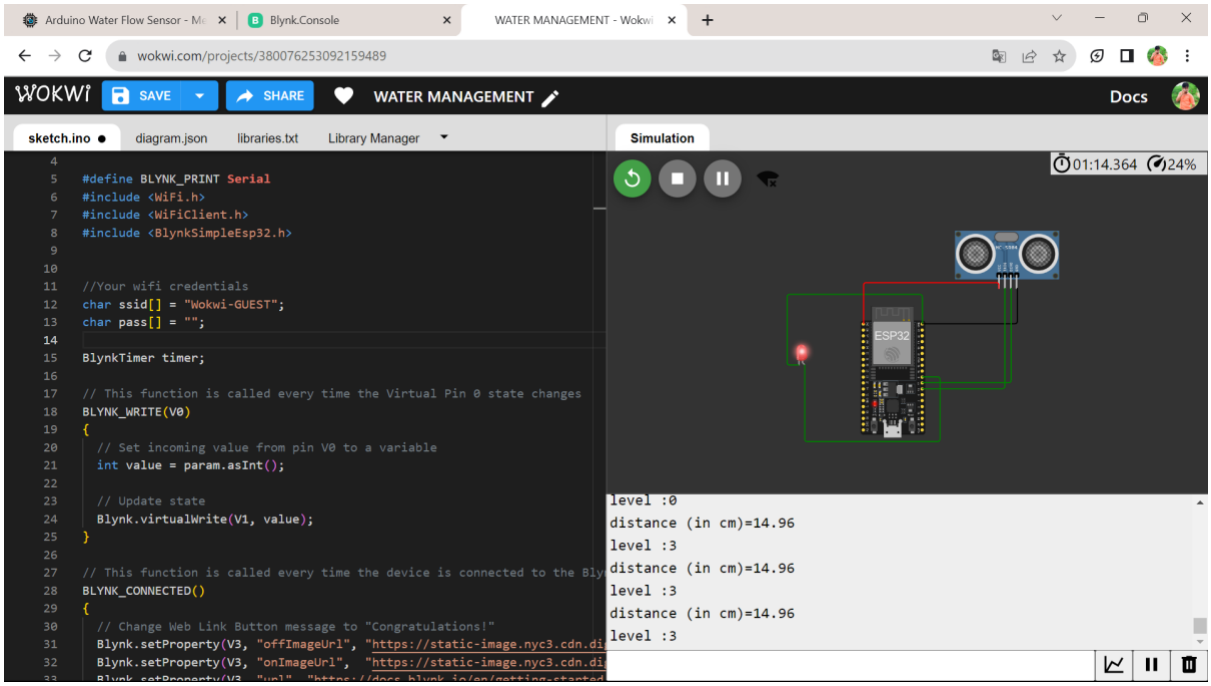
The Above processes can be monitored and controlled using BLYNK

6. Conclusion

Smart water management is the use of digital technologies to improve the efficiency and sustainability of water management. It involves the use of sensors, data analytics, and artificial intelligence to monitor and control water systems in real time. smart water management can also have a broader impact on the environment and society. Smart water management is a vital tool for addressing the global water crisis. By investing in smart water management systems, we can improve the efficiency and sustainability of water management and ensure that everyone has access to clean water.

Output of project in wokwi




Arduino Water Flow Sensor - M
Blynk.Console
WATER MANAGEMENT - Wokwi
wokwi.com/projects/380076253092159489
WOKWI
SAVE
SHARE
WATER MANAGEMENT
Docs
sketch.ino
diagram.json
libraries.txt
Library Manager
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
//Your wifi credentials
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
BlynkTimer timer;
// This function is called every time the Virtual Pin 0 state changes
BLYNK_WRITE(V0)
{
// Set incoming value from pin V0 to a variable
int value = param.asInt();
// Update state
Blynk.virtualWrite(V1, value);
}
// This function is called every time the device is connected to the Bly
BLYNK_CONNECTED()
{
// Change Web Link Button message to "Congratulations!"
Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.di
Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.di
Simulation
01:14.364
24%
ESP32
level :0
distance (in cm)=14.96
level :3
distance (in cm)=14.96
level :3
distance (in cm)=14.96
level :3