SMART BUILDING USING BLYNK

Coincent

SEPTEMBER 28, 2024

KALKI S
126006056@sastra.ac.in

TABLE OF CONTENTS

ABSTRACT

This project demonstrates the implementation of a Smart Building system using the BLYNK IoT platform, ESP32 microcontroller, and Wokwi simulation. The system integrates two LEDs, two buzzers, and a temperature sensor. The LEDs and buzzers are controlled via the BLYNK app, while the temperature sensor continuously monitors and displays the building's temperature on the serial monitor. This system provides an effective way to remotely control and monitor building safety features, showcasing the potential of IoT in home automation.

OBJECTIVE

The objective of this project is to design and implement a Smart Building system that allows remote control and monitoring using the BLYNK platform. The system provides control over lighting (through LEDs) and security (through buzzers) and enables real-time temperature monitoring using a temperature sensor connected to an ESP32 microcontroller.

Introduction

With the rapid growth of IoT, smart buildings are becoming more popular as they enhance automation, energy efficiency, and safety. The BLYNK IoT platform offers a simple and efficient way to manage various smart devices over the internet. This project integrates two LEDs and two buzzers controlled via BLYNK, and a temperature sensor to provide real-time monitoring of building conditions. The ESP32 microcontroller acts as the central unit, connecting sensors and actuators to the BLYNK platform. The Wokwi platform is used to simulate the design, offering an easy testing environment.

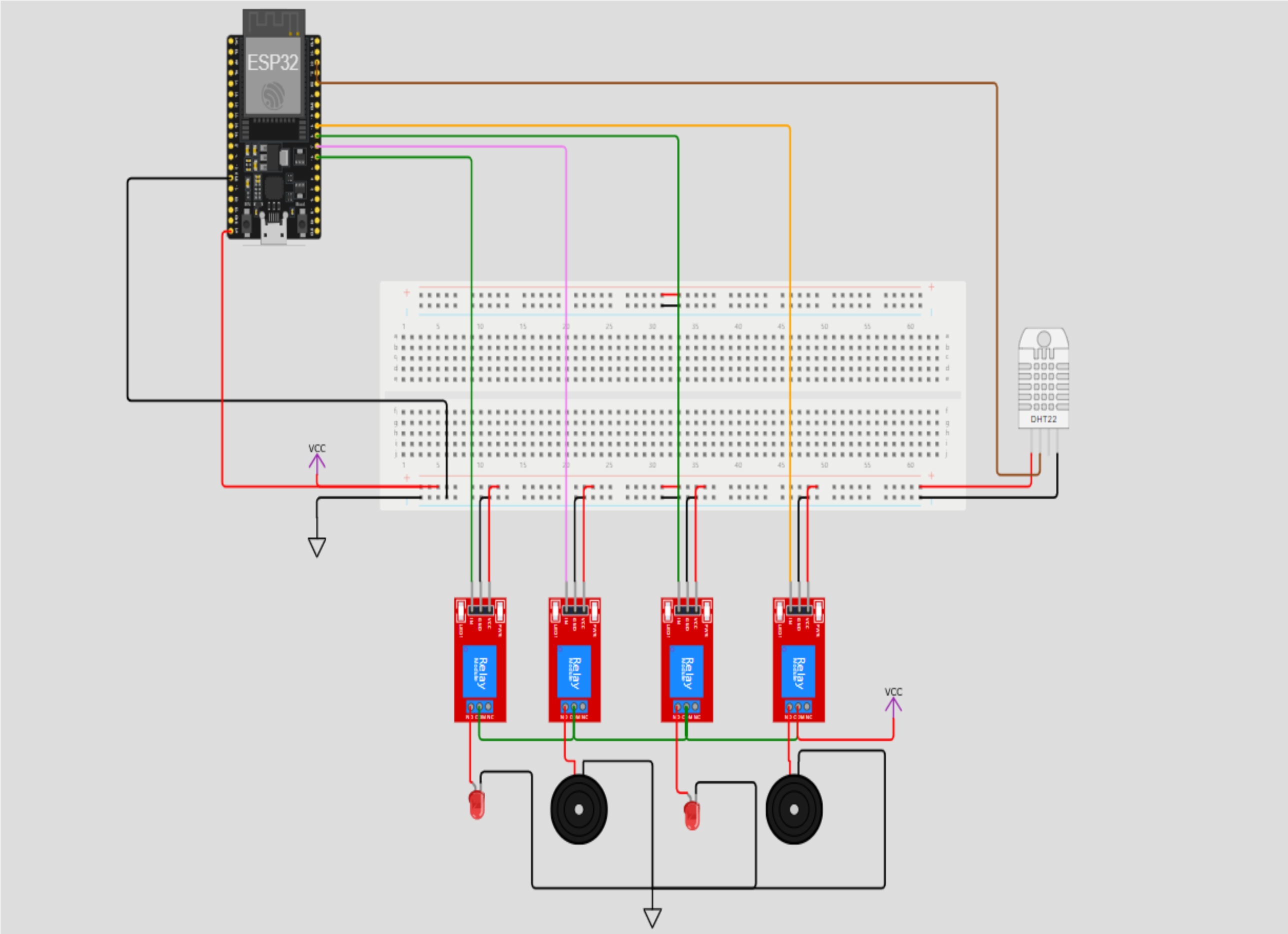
METHADOLOGY

- **Microcontroller**: ESP32 is used to handle the inputs and outputs for the system.
- **Temperature Sensor**: The sensor monitors the ambient temperature and displays it in the serial monitor for real-time tracking.
- **LEDs & Buzzers**: Two LEDs and two buzzers are controlled using the BLYNK app. The LEDs represent building lights, while the buzzers can be used for security alerts.
- **BLYNK Platform**: The mobile app allows remote access to the system, enabling users to switch the LEDs and buzzers on and off via the internet.
- **Simulation Platform**: The Wokwi platform is used for the simulation of the system to ensure that all components work together effectively before deployment.

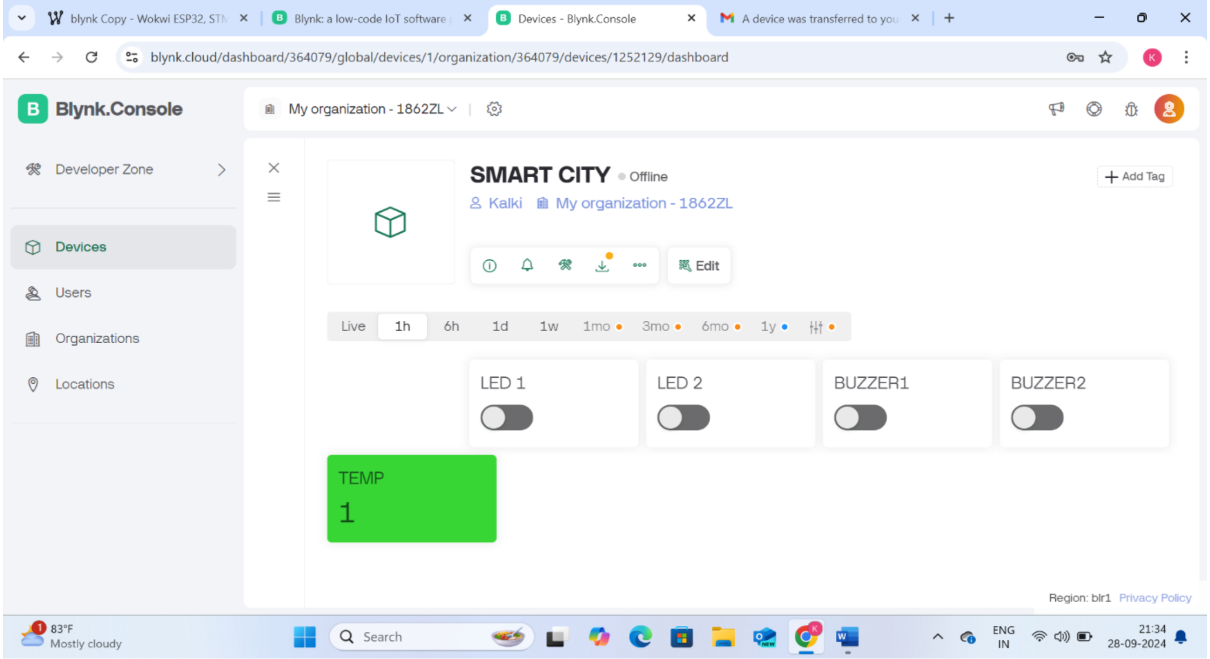
PROJECT FILE

https://wokwi.com/projects/410006857085570049

C:\Users\kalki\Downloads\blynk Copy.zip



BLYNK CONSOLE



CODE

```
#define BLYNK_TEMPLATE_ID "TMPL3Cb2EdMV9"
#define BLYNK_TEMPLATE_NAME "SMART CITY"
#define BLYNK_AUTH_TOKEN "eyHaj_k9cPZAF-iPC2HSBONRyOKooUW0"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHTesp.h>
// WiFi credentials
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
BlynkTimer timer;
DHTesp dhtSensor;
const int dev1 = 16;  // GPIO for dev1 (LED)
const int dev2 = 17;
const int dev3 = 5;
const int dev4 = 18;
const int dhtPin = 22; // GPIO pin where DHT22 is connected
int relay_s_dev1 = 0;  // Variable to track the state of dev1 (LED)
int relay_s_dev2 = 0;
int relay_s_dev3 = 0;
int relay_s_dev4 = 0;
// Virtual pins for relay controls and DHT22
#define vpin1 V0
```

```
#define vpin2 V1

#define vpin3 V2

#define vpin4 V3

#define vpin5 V4

BLYNK_CONNECTED() {

 // Sync virtual pins with the hardware when Blynk connects

 Blynk.syncVirtual(vpin1);

 Blynk.syncVirtual(vpin2);

 Blynk.syncVirtual(vpin3);

 Blynk.syncVirtual(vpin4);

 Blynk.syncVirtual(vpin5);

}

BLYNK_WRITE(vpin1) {

 // Update the state of dev1 (LED) based on the value from Blynk

 relay_s_dev1 = param.asInt();

 digitalWrite(dev1, relay_s_dev1);  // Set dev1 (LED) to HIGH or LOW

}

BLYNK_WRITE(vpin2) {

 relay_s_dev2 = param.asInt();

 digitalWrite(dev2, relay_s_dev2);

}

BLYNK_WRITE(vpin3) {

 relay_s_dev3 = param.asInt();

 digitalWrite(dev3, relay_s_dev3);

}

BLYNK_WRITE(vpin4) {

 relay_s_dev4 = param.asInt();

 digitalWrite(dev4, relay_s_dev4);

}
```

```
// Function to send DHT22 temperature data to Blynk
void send_dht_val() {
 TempAndHumidity data = dhtSensor.getTempAndHumidity();

 if (isnan(data.temperature) || isnan(data.humidity)) {
  Serial.println("Failed to read from DHT sensor!");
  return;  // Skip if the sensor read fails
 }

 float temp = data.temperature;  // Get the temperature value
 Serial.println("Temperature: " + String(temp)); // Debug: Print to Serial monitor
 Blynk.virtualWrite(V4, temp);  // Send temperature to Blynk app
}
void setup() {
 // Initialize device pins as outputs
 pinMode(dev1, OUTPUT);
 pinMode(dev2, OUTPUT);
 pinMode(dev3, OUTPUT);
 pinMode(dev4, OUTPUT);
 // Set initial states of all devices to LOW (OFF)
 digitalWrite(dev1, LOW);
 digitalWrite(dev2, LOW);
 digitalWrite(dev3, LOW);
 digitalWrite(dev4, LOW);
 // Initialize DHT22 sensor
 dhtSensor.setup(dhtPin, DHTesp::DHT22);
 // Initialize serial communication
 Serial.begin(115200);
 // Connect to Blynk using WiFi credentials
```

```
  Blynk.begin(auth, ssid, pass);
  // Set the interval for sending DHT sensor values every 5 seconds
  timer.setInterval(5000L, send_dht_val);
}

void loop() {
  Blynk.run();  // Run Blynk
  timer.run();  // Run Blynk timer
}
```

CONCLUSION

The Smart Building system using BLYNK and ESP32 provides a functional and scalable IoT solution for remote building management. By integrating LEDs, buzzers, and a temperature sensor, the system demonstrates the capability to automate and monitor critical building features through a user-friendly mobile interface. The use of Wokwi as a simulation platform also ensures that the system can be thoroughly tested and debugged before deployment in a real-world scenario.

This project highlights the feasibility of using IoT technologies in smart home and building automation, offering insights into energy management and security systems.