

```
#define BLYNK_TEMPLATE_ID "Temp_ID"
#define BLYNK_TEMPLATE_NAME "Temp_name"
#define BLYNK_AUTH_TOKEN "Auth_Token"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <ThingSpeak.h> // Include ThingSpeak library

char ssid[] = "Your Wifi username";
char pass[] = "Wifi Password";

// DHT settings
#define DHTPIN 15
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// Relay pins
#define FAN_RELAY 23 // IN1
#define BULB_RELAY 22 // IN2
#define ALARM_RELAY 21 // IN3

// Blynk virtual pins
#define FAN_VPIN V0
#define BULB_VPIN V1
#define TEMP_VPIN V2
#define ALARM_VPIN V3

bool fanState = false;
bool bulbState = false;

unsigned long lastAlarmTime = 0;
const unsigned long alarmInterval = 5000;

WiFiClient client;

// ThingSpeak settings
unsigned long channelID = 123456; // Your ThingSpeak Channel ID
const char *writeAPIKey = "XXXXXXXXXXXXXX"; // Your ThingSpeak API Key

void setup() {
```

```
Serial.begin(115200);
dht.begin();

pinMode(FAN_RELAY, OUTPUT);
pinMode(BULB_RELAY, OUTPUT);
pinMode(ALARM_RELAY, OUTPUT);

digitalWrite(FAN_RELAY, HIGH); // OFF
digitalWrite(BULB_RELAY, HIGH); // OFF
digitalWrite(ALARM_RELAY, HIGH); // OFF

Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

// Initialize ThingSpeak
ThingSpeak.begin(client);
}

// Manual fan control
BLYNK_WRITE(FAN_VPIN) {
fanState = param.asInt();
digitalWrite(FAN_RELAY, fanState ? LOW : HIGH);
}

// Manual bulb control
BLYNK_WRITE(BULB_VPIN) {
bulbState = param.asInt();
digitalWrite(BULB_RELAY, bulbState ? LOW : HIGH);
}

void loop() {
Blynk.run();

float temp = dht.readTemperature();
if (isnan(temp)) {
Serial.println("Failed to read from DHT sensor!");
delay(2000);
return;
}

// Display temperature on Serial Monitor
Serial.print("Temperature: ");
```

```
Serial.print(temp);
Serial.println(" °C");

// Send temperature data to ThingSpeak
ThingSpeak.setField(1, temp); // Field 1 corresponds to Temperature

// Write data to ThingSpeak
int writeSuccess = ThingSpeak.writeFields(channelID, writeAPIKey);
if (writeSuccess == 200) {
    Serial.println("Data successfully uploaded to ThingSpeak");
} else {
    Serial.print("Error writing to ThingSpeak. HTTP error code: ");
    Serial.println(writeSuccess);
}

bool shouldRingAlarm = false;

if (temp > 30 && !fanState) {
    shouldRingAlarm = true;
} else if (temp < 20 && !bulbState) {
    shouldRingAlarm = true;
}

if (fanState || bulbState) {
    shouldRingAlarm = false;
}

if (shouldRingAlarm) {
    if (millis() - lastAlarmTime >= alarmInterval) {
        lastAlarmTime = millis();
        digitalWrite(ALARM_RELAY, LOW); // Alarm ON
        delay(2000); // Ring for 2 seconds
        digitalWrite(ALARM_RELAY, HIGH); // Alarm OFF
    }
} else {
    digitalWrite(ALARM_RELAY, HIGH); // Always OFF if not needed
}

delay(2000); // Delay before the next reading
}
```