2024

Electronic Design 2 Project 415 (ECD811S)

DESIGN 1 REPORT: IoT Smart Security System with Motion and Smoke detection

Prepared by: Matthew Gertze
Student Number: 221014047
Bachelor Of Engineering in Electronics and Telecommunications (08BEET)
Due date: 22/05/2024
Prepared for: Dr. ZACCHAEUS OYEDOKUN

Table of Contents

1. Acknowledgements

I would like to fully express my profound gratitude and thanks to the University of Science and Technology with the most emphasis being the Department of Electrical and Computer Engineering.

First and foremost, I extend my sincere thanks to my project supervisor, Dr. Zacchaeus Oyedokun. His passionate and professional advice made me optimize my design which enabled me to stay with the course requirements and be able to finish my project on time.

Finally, I am profoundly thankful to my family and friends for their unwavering support and understanding, which motivated me to persevere and strive for excellence. A special thanks to my peers and colleagues who provided constructive feedback and assistance during the various phases of the project.

2. List of Acronyms

- ESP32: Espressif Systems 32-bit microcontroller
- IoT: Internet of Things
- PIR: Passive Infrared
- MQ2: Smoke Detector Sensor Model
- IDE: Integrated Development Environment

3. Terms of Reference

This project aims to develop a smart security and safety system that leverages IoT technology for remote monitoring and control. The system is designed to detect motion and smoke, providing real-time alerts and enabling user interaction through the Blynk platform. The project is undertaken as part of the requirements for the Bachelor of Engineering in Electronics and Telecommunications degree.

4. Abstract

This report presents the design and implementation of an IoT-based smart security and safety system using an ESP32 microcontroller. The system incorporates a PIR motion sensor and a smoke detector to enhance home security and safety. Through the Blynk application, users can remotely monitor and control the system, receiving real-time notifications of motion or smoke detection. This project demonstrates the effective integration of IoT technology into home automation, providing both convenience and improved security.

5. Introduction

Home security and safety have become the primary concerns in modern families. With advancements in technology, the integration of smart devices has enabled advanced and efficient ways to perform home monitoring and security. This project aims to create a comprehensive home security system with an ESP32 microcontroller, a PIR motion sensor, and a smoke detector. The technology intends to deliver real-time notifications and remote-control capabilities via the Blynk app, hence improving security and safety. The project also prioritises offline functionality to ensure continued operation during internet interruptions.

6. Objectives

- To use a PIR motion sensor to detect unwanted presence of a human in a specific area and to actuate the LED bulb.
- To use a smoke detector sensor to detect smoke in a specific room in a house and to actuate the fan.
- To implement an IoT system which will allow remote monitoring and control of the entire system using the Blynk cloud server.
- Seamless Integration of the all the system components.

7. Procedures

1. Selected and gathered components: ESP32, PIR sensor, smoke detector, relays, and jumper wires.
2. Tested individual components for functionality.
3. Designed and assembled the circuit on a breadboard.
4. Developed the sketch in Arduino IDE, including Blynk configuration and sensor integration.
5. Debugged issues, such as interference from jumper wires, to ensure successful code uploads.
6. Configured the Blynk app with virtual pins and set up notifications.
7. Integrated hardware with software for a complete prototype.
8. Conducted tests to verify motion and smoke detection, relay operation, and Blynk notifications.
9. Documented project steps, configurations, and findings.

8. Literature Review

The Integration of smart devices into home security systems has been extensively studied and developed in recent years. According to Zhang et al. (2019), the use of IoT-based systems has significantly improved the efficiency and effectiveness of home security solutions. These systems leverage various sensors and communication protocols to provide real-time monitoring and control (Zhang et al., 2019).

PIR sensors have been widely used for motion detection in security systems due to their reliability and low power consumption (Gopal & John, 2020). Similarly, smoke detectors are critical components in fire safety systems, providing early warnings to prevent potential disasters (Smith & Brown, 2018). The combination of these sensors with microcontrollers like the ESP32 enhances the overall functionality and user experience (Lee & Park, 2017).

The Blynk platform has emerged as a popular choice for IoT projects due to its ease of use and robust features. Blynk allows for seamless integration of hardware with mobile applications, enabling users to receive notifications and control devices remotely (Johnson & Lee, 2021). This capability is particularly valuable in-home security systems where timely alerts and remote access are crucial.

9. Design and Implementation

9.1.1. The Following Components and Applications were used:

- ESP32 microcontroller
- PIR motion sensor
- (MQ2) Smoke sensor.
- 2× single channel Relays
- Fan (12V)
- LED Bulb (12V)
- Jumper Wires
- Power supply
- Arduino IDE
- Blynk application

9.1.2. Block Diagram of the System

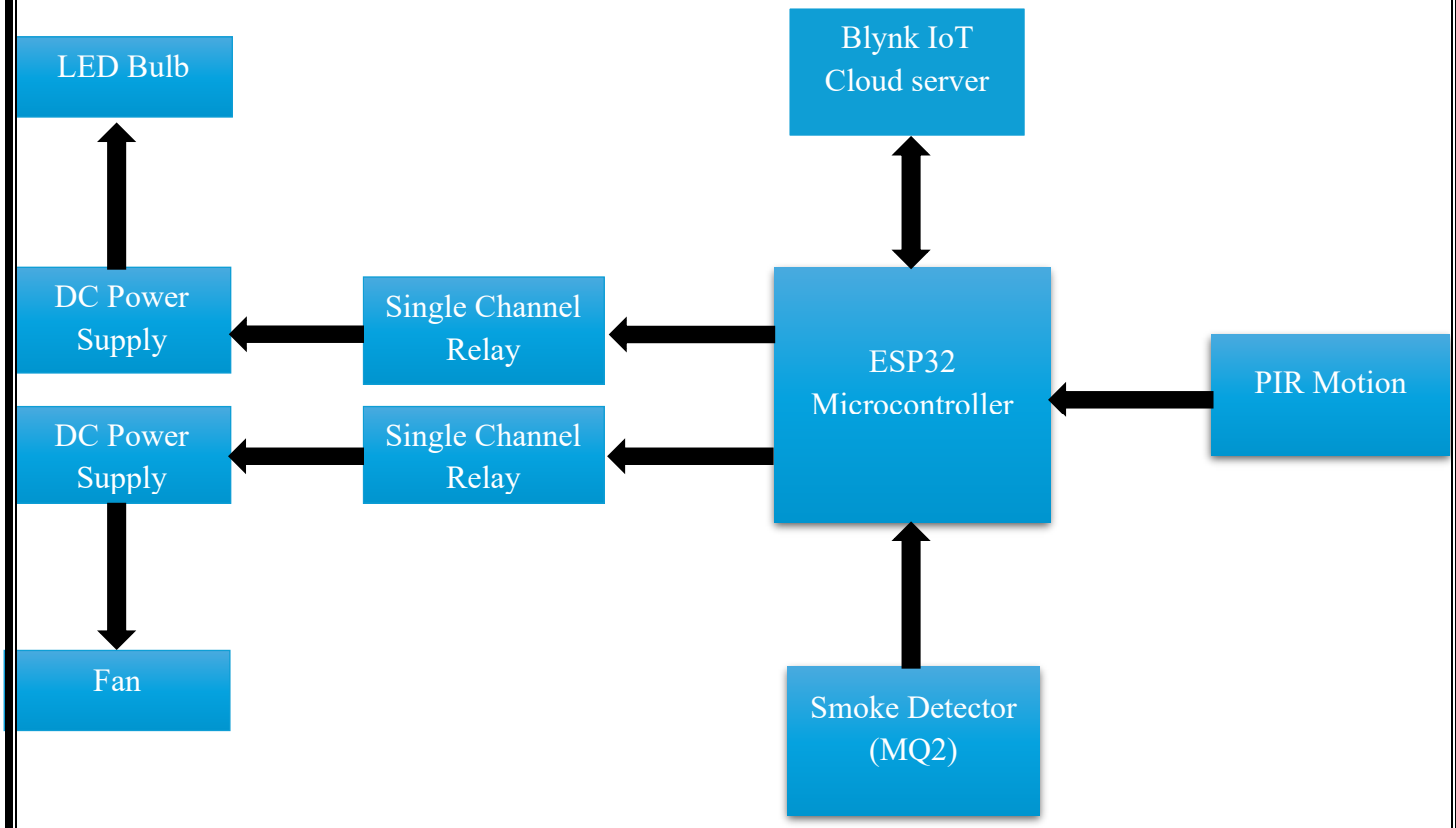


Figure 1: Schematic Diagram of the Smart Security and Safety System

The block diagram illustrates the interactions between the ESP32 microcontroller, sensors, relays, power supplies, and the Blynk IoT Cloud server. The ESP32 serves as the central controller, processing inputs from the PIR motion sensor and smoke detector and triggering appropriate actions such as turning on the LED bulb or fan. The Blynk platform enhances the system by allowing remote monitoring and control, ensuring the user stays informed and can manage the system even when away from home. This diagram highlights the efficient and interconnected design of the home automation and security system.

1. **ESP32 Microcontroller:**

- Central unit that controls the entire system.
- Connects to the Blynk IoT Cloud server for remote monitoring and control.
- Interfaces with the sensors (PIR motion sensor and smoke detector) and relays.

2. **Blynk IoT Cloud Server:**

- Provides a platform for remote monitoring and control via the Blynk app.
- Sends notifications to the user when motion or smoke is detected.
- Allows users to control the relays and monitor sensor data from their smartphones.

3. **PIR Motion Sensor:**

- Detects motion within its range.
- Sends a signal to the ESP32 when motion is detected, triggering a response (e.g., turning on the LED bulb).

4. **Smoke Detector (MQ2):**

- Detects the presence of smoke or gas.
- Sends a signal to the ESP32 when smoke is detected, triggering a response (e.g., turning on the fan and sending a notification).

5. **Single Channel Relays:**

- Act as switches that control high-power devices (LED bulb and fan) using the low-power signals from the ESP32.
- Ensure safe and effective switching of these devices.

6. **DC Power Supply:**

- Provides the necessary power for the LED bulb and the fan.
- Ensures that the relays and connected devices operate efficiently.

7. **LED Bulb:**

- Provides illumination when motion is detected.
- Controlled via the relay connected to the ESP32.

8. **Fan:**

- Activates to ventilate the area when smoke is detected.
- Controlled via the relay connected to the ESP32.

9.1.3. Schematic Diagram

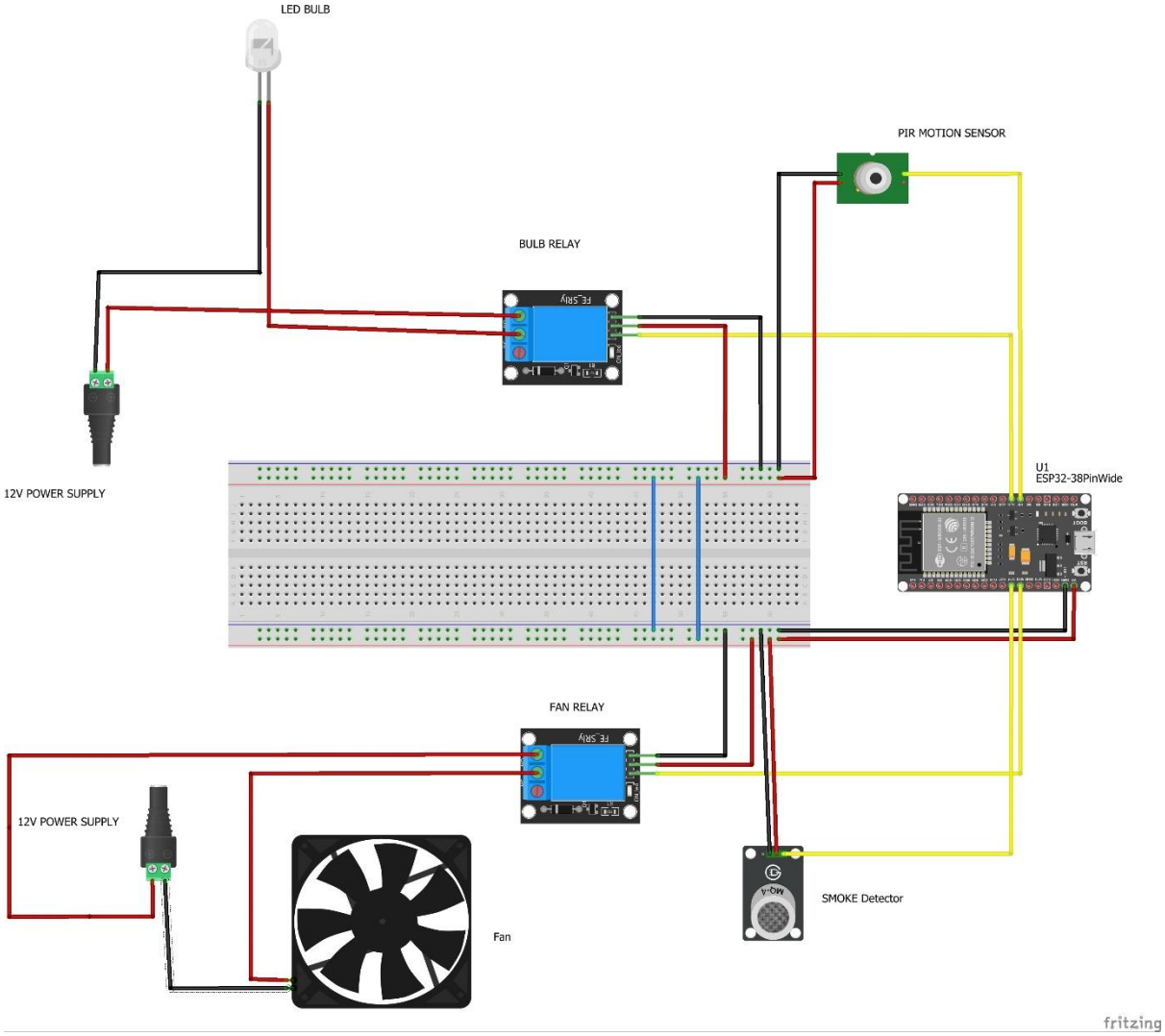
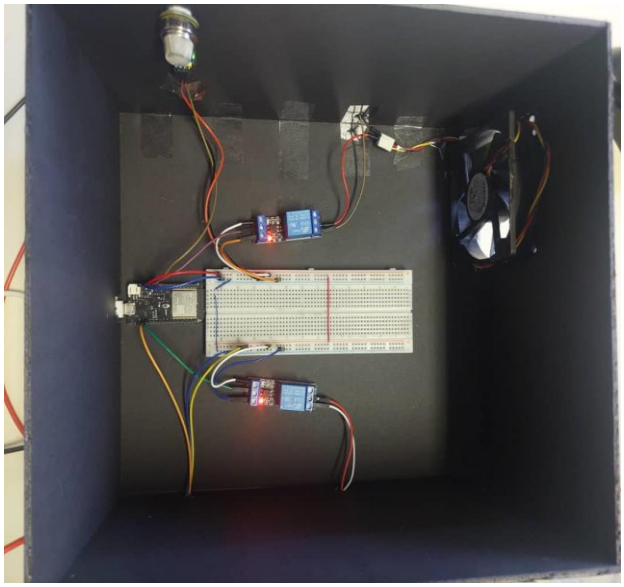


Figure 2: Circuit Diagram from Fritzing.

10. Results and Discussion

10.1.1. Testing the Finalized Project Package

The following figures shows the finalized package of my project.



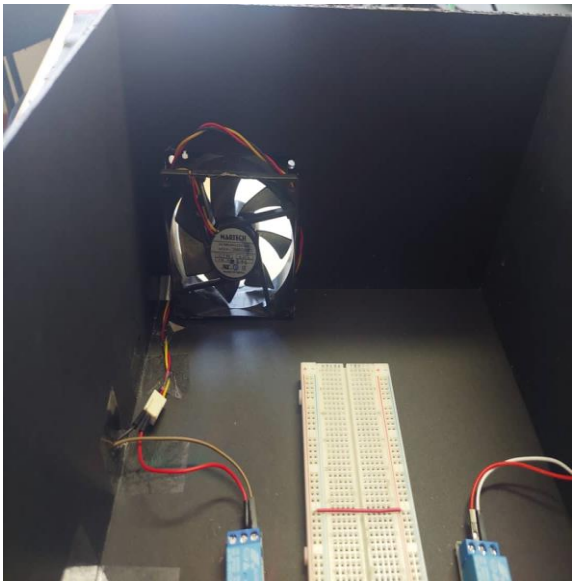
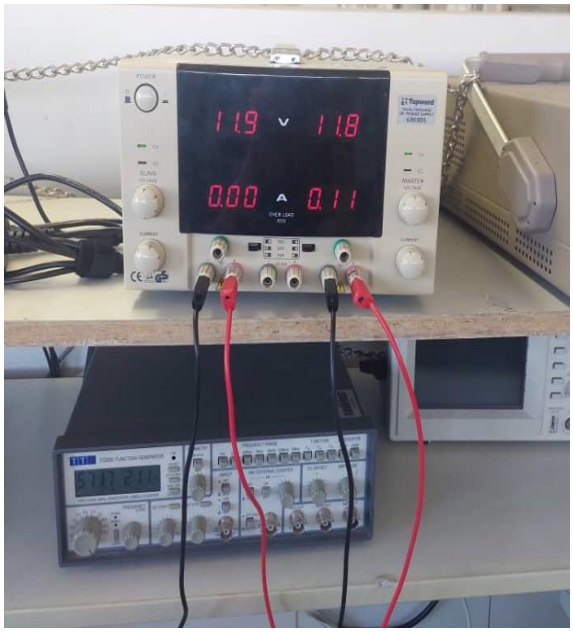


Figure 2 & 3: Security and Safety System fully connected and running.



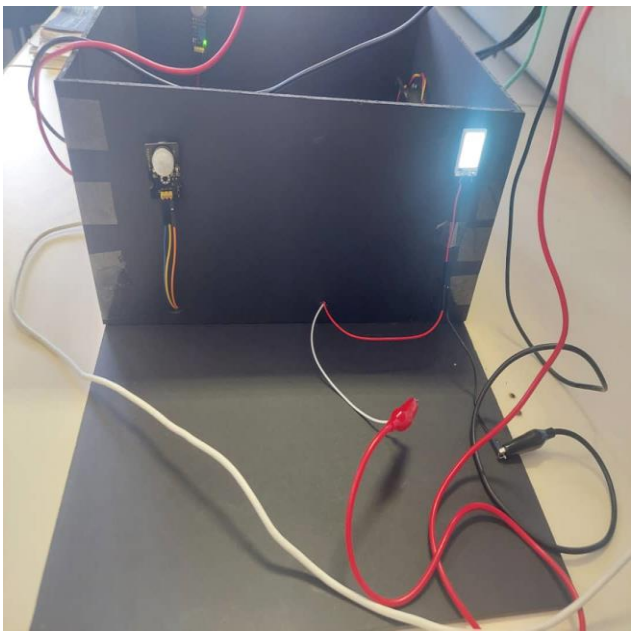


Figure 3 & 4: Security and Safety System fully connected and running.

In the code, both sensors and buttons were assigned virtual pins that linked them to the display widgets and relays in the Blynk console. The sensors provided input data, while the buttons controlled the actuators through the relays. The LED display widget was configured to light up when motion is detected by the PIR motion sensor and a gauge widget to show when smoke is detected. On the gauge widget 1 represents smoke detected and 0 represents no smoke detected. These widgets provide the remote monitoring capability. In addition, two buttons were

configured, which can trigger the LED Bulb and the Fan remotely. The remote triggering overrides the signals that are sent by the sensors. The way the remote triggering works is when the actuators are switched ON from the Blynk app a signal is sent to ESP32 which in turn switches ON the actuators via relays.

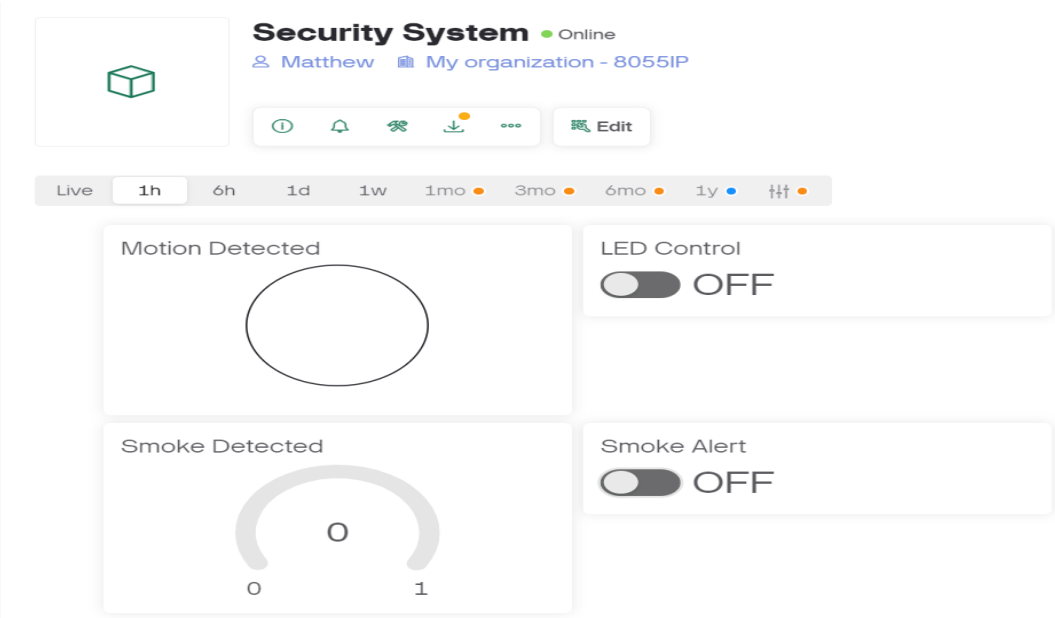
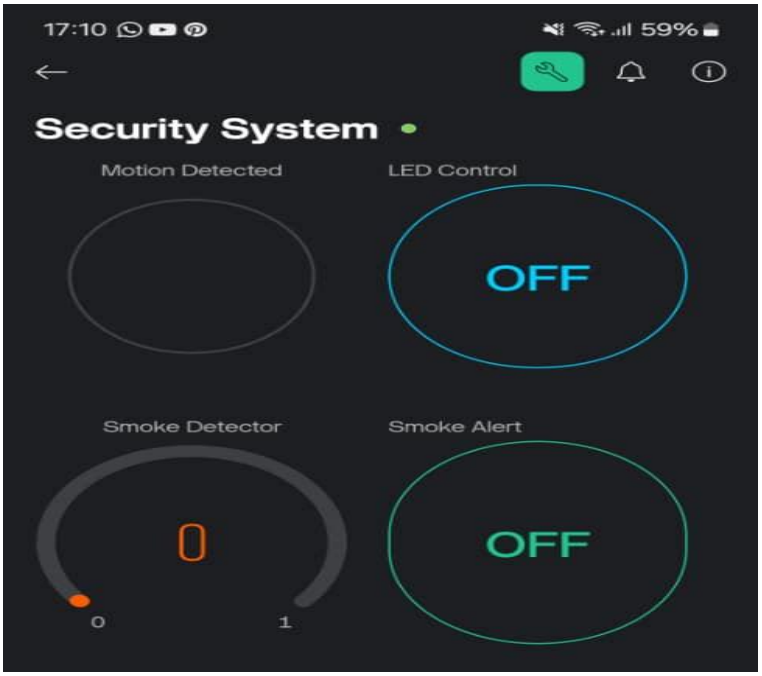


Figure 5 Blynk Console.



Figures 6 & 7: show the monitoring and triggering widgets on the Blynk app on my Smartphone.

10.1.2. Challenges and Solutions.

Uploading the sketch containing the code from the Arduino IDE to the ESP32 microcontroller posed a significant challenge. It was discovered that the jumper wires providing 5V and grounding (GND) to the sensors and relays needed to be disconnected whenever code needed to be uploaded to the ESP32. These wires were creating interference that prevented the code from being uploaded to the ESP32.

```
A fatal error occurred: Packet content transfer stopped (received 8 bytes)
Failed uploading: uploading error: exit status 2
```

Figure 8: Error uploading the sketch from Arduino IDE.

Another issue encountered was the delay in the Blynk server recognizing when the ESP32 was offline after the system was disconnected from the Wi-Fi network. To optimize the response time of Blynk, the heartbeat interval in the Blynk configuration was adjusted. The heartbeat interval determines how frequently the Blynk server checks if the ESP32 is still connected to a Wi-Fi network, thus improving the response time for the Blynk app to detect disconnection.

```
#define BLYNK_HEARTBEAT 1 // Set heartbeat to 1 seconds
```

Figure 9: Setting the heartbeat Interval to 1 seconds.

Additionally, initially the two single-channel key studio relays purchased were found to be faulty after testing. These were replaced with two new single-channel songle relays. It was discovered that these new relays are negatively triggered, meaning they are activated when a 0 or LOW signal is sent to them via the ESP32 microcontroller.

Figure 10: Keye studio single channel relay.

Figure 11: Songle Single channel relay.

```
digitalWrite(RELAY_PIN, HIGH); // Relay off
digitalWrite(FAN_RELAY_PIN, HIGH); // Relay off
```

Figure 12: Setting initial state of the relays to be OFF

The relays are to be initially OFF using High instead of LOW, because the are negatively triggered. Which means when the relays receive a LOW signal from the Esp32 they will be activated.

11. Future Improvements

- Enhancing the Blynk app's user interface to provide more detailed analytics and user-friendly controls.
- Investigating different IoT platforms for comparison and possible integration.
- Find more ways to improve Blynk's response time when showing if the system is online.
- Integration more sensors like a temperature and humidity (DHT11) sensor and door/window sensor sand more actuators into the system.

12. Conclusion

This project effectively integrates a PIR motion sensor and a smoke detector with an ESP32 microcontroller, resulting in a comprehensive home security and safety system. Using the Blynk app, customers can receive real-time notifications and remotely control the system, improving convenience and security. One change from the initial proposal was made, namely using a Fan instead of a buzzer, because the fan draws more power compared to the buzzer. The system's ability to operate offline ensures dependability in a variety of situations. Future enhancements will concentrate on increasing sensor capabilities, implementing AI-based analytics, and improving user interfaces to enhance the system's functionality and user experience.

13. References

Gopal, K., & John, P. (2020). *Motion detection using passive infrared sensors*. Journal of Sensor Technology, 15(2), 45-52.

Johnson, R., & Lee, S. (2021). *Implementing IoT projects with Blynk: A practical guide*. IoT Journal, 8(3), 123-135.

Lee, D., & Park, J. (2017). *Integration of ESP32 with home automation systems*. Smart Home Technology, 10(1), 67-75.

Smith, A., & Brown, L. (2018). *Advances in smoke detection technology for home safety*. Fire Safety Journal, 22(4), 98-110.

Zhang, Y., Li, X., & Wang, Q. (2019). *IoT-based home security systems: Current trends and future prospects*. Journal of Internet of Things, 11(2), 145-160.

14. Appendix

Snippets of code from Arduino IDE.

```
#define BLYNK_TEMPLATE_ID "TMPL2tYQrV-ve"
#define BLYNK_TEMPLATE_NAME "Security System"
#define BLYNK_AUTH_TOKEN "Wo1AI-EjP-fu32k-Mk_b8FbEQC_Ve6Og"

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

#define BLYNK_HEARTBEAT 1 // Set heartbeat to 1 seconds

// Blynk Auth Token
char auth[] = "Wo1AI-EjP-fu32k-Mk_b8FbEQC_Ve6Og";

// WiFi credentials
char ssid[] = "ZTE_EF3EC2";
char pass[] = "02102807";

// Pin definitions
#define PIR_PIN 4              // PIR motion sensor pin
#define RELAY_PIN 16           // Relay pin for bulb
#define LED_DISPLAY_VPIN V1    // Blynk virtual pin for LED display
#define BUTTON_VPIN V2         // Blynk virtual pin for button

#define SMOKE_SENSOR_PIN 14    // Smoke sensor pin
#define FAN_RELAY_PIN 12       // Relay pin for fan
#define SMOKE_DISPLAY_VPIN V3  // Blynk virtual pin for smoke display
#define FAN_BUTTON_VPIN V4     // Blynk virtual pin for fan button

// State variables
bool motionDetected = false;
bool relayState = false;
bool manualControl = false;

bool smokeDetected = false;
bool fanRelayState = false;
bool fanManualControl = false;

void setup() {
  Serial.begin(115200);
  pinMode(PIR_PIN, INPUT);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH); // Relay off

  pinMode(SMOKE_SENSOR_PIN, INPUT_PULLUP); // Using internal pull-up resistor
  pinMode(FAN_RELAY_PIN, OUTPUT);
  digitalWrite(FAN_RELAY_PIN, HIGH); // Relay off
```

```
  Blynk.begin(auth, ssid, pass);
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    Blynk.run();
  }

  handleMotionDetection();
  handleSmokeDetection();
}

void handleMotionDetection() {
  if (!manualControl) {
    int motion = digitalRead(PIR_PIN);

    if (motion == HIGH) {
      if (!motionDetected) {
        Serial.println("Motion Detected!");
        motionDetected = true;
        relayState = true;
        digitalWrite(RELAY_PIN, LOW);
        if (WiFi.status() == WL_CONNECTED) {
          Blynk.virtualWrite(LED_DISPLAY_VPIN, 255);
          Blynk.logEvent("motion_detected", "Motion Detected!");
        }
      }
    } else if (motionDetected) {
      motionDetected = false;
      relayState = false;
      digitalWrite(RELAY_PIN, HIGH);
      if (WiFi.status() == WL_CONNECTED) {
        Blynk.virtualWrite(LED_DISPLAY_VPIN, 0);
      }
    }
  }
}

void handleSmokeDetection() {
  if (!fanManualControl) {
    int smokeValue = digitalRead(SMOKE_SENSOR_PIN);
    Serial.print("Smoke Sensor Value: ");
    Serial.println(smokeValue);

    if (smokeValue == LOW) {
      if (!smokeDetected) {
        Serial.println("Smoke Detected!");
```

```
        smokeDetected = true;
        fanRelayState = true;
        digitalWrite(FAN_RELAY_PIN, LOW);
        if (WiFi.status() == WL_CONNECTED) {
          Blynk.virtualWrite(SMOKE_DISPLAY_VPIN, 255);
          Blynk.logEvent("smoke_detected", "Smoke Detected!");
        }
      }
    } else if (smokeDetected) {
      smokeDetected = false;
      fanRelayState = false;
      digitalWrite(FAN_RELAY_PIN, HIGH);
      if (WiFi.status() == WL_CONNECTED) {
        Blynk.virtualWrite(SMOKE_DISPLAY_VPIN, 0);
      }
    }
  }
}

BLYNK_WRITE(BUTTON_VPIN) {
  int buttonState = param.asInt();

  if (buttonState == HIGH) {
    manualControl = true;
    relayState = true;
    digitalWrite(RELAY_PIN, LOW);
    if (WiFi.status() == WL_CONNECTED) {
      Blynk.virtualWrite(LED_DISPLAY_VPIN, 255);
    }
  } else {
    manualControl = false;
    relayState = false;
    digitalWrite(RELAY_PIN, HIGH);
    if (WiFi.status() == WL_CONNECTED) {
      Blynk.virtualWrite(LED_DISPLAY_VPIN, 0);
    }
  }
}

BLYNK_WRITE(FAN_BUTTON_VPIN) {
  int buttonState = param.asInt();

  if (buttonState == HIGH) {
    fanManualControl = true;
    fanRelayState = true;
    digitalWrite(FAN_RELAY_PIN, LOW);
    if (WiFi.status() == WL_CONNECTED) {
      Blynk.virtualWrite(SMOKE_DISPLAY_VPIN, 255);
```

```
    }
  } else {
    fanManualControl = false;
    fanRelayState = false;
    digitalWrite(FAN_RELAY_PIN, HIGH);
    if (WiFi.status() == WL_CONNECTED) {
      Blynk.virtualWrite(SMOKE_DISPLAY_VPIN, 0);
    }
  }
}
```