



UGC & Govt. Approved  
**Sonargaon University (SU)**  
সোনারগাঁও ইউনিভার্সিটি (এসইউ)

## **ESP32 Multi-Sensor Safety Project**

**Supervised By**

**Md. Ramjan Ali**

Associate Professor

Department of EEE

**Submitted By**

<b>Name</b>	<b>ID</b>
Shahriar Emon	EEE-2203027010
Md. Jibon Mia	EEE-2203027062
Md. Nurul Islam Noyan	EEE-2203027028
Md. Abdul Alim	EEE-2203027052
Md. Jowel Rana	EEE-2203027031

**Department of Electrical & Electronic Engineering**

**Sonargaon University (SU)**

**147/1, Panthapath, Dhaka-1215.**

**Date of Submission: 14 January, 2026**

**DECLARATION**

It is declared hereby that this thesis paper or any part of it has not been submitted to anywhere else for the award of any degree.

---

Shahriar Emon

---

Md. Jibon Mia

---

Md. Nurul Islam Noyan

---

Md. Abdul Alim

---

Md. Jowel Rana

**Under Supervision of**

---

**Md. Ramjan Ali**

Associate Professor

Department of EEE

Sonargaon University (SU)

## **CERTIFICATION**

This is to certify that this thesis entitled “**ESP32 Multi-Sensor Safety Project**” is done by the following students under my direct supervision. This thesis work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Sonargaon University (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering (EEE).

<b>Name</b>	<b>ID</b>
Shahriar Emon	EEE-2203027010
Md. Jibon Mia	EEE-2203027062
Md. Nurul Islam	EEE-2203027028
Md. Abdul Alim	EEE-2203027052
Md. Jowel Rana	EEE-2203027031

### **Supervisor**

Md. Ramjan Ali

Associate Professor

Department of Electrical and Electronic Engineering (EEE)

Sonargaon University (SU)

## **ACKNOWLEDGEMENT**

At first, we started in the name of Almighty Allah. This thesis is accomplished under the supervision of MD RAMJAN ALI, Associate Professor, Department of Electrical and Electronic Engineering, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for this consistent guidance, encouragement, helpful suggestions, constructive criticism, and endless patience throughout the process of this work. The successful completion of this thesis would not be possible without his persistent motivation and continuous guidance.

The authors are also grateful to Md. Rais Uddin Mollah, Head of the Department of Electrical and Electronic Engineering, and all the teachers of the department for their cooperation and significant help in completing the thesis work successful. Finally, we wish to acknowledge those who directly or indirectly helped us in connecting with this work.

## ABSTRACT

The “**ESP32 Multi-Sensor Safety Project**” presents a smart and automated safety monitoring system designed to detect fire-related hazards in real time. The system integrates multiple sensors, including a flame sensor, smoke/gas sensor, temperature sensor, and environmental monitoring units, all controlled by an ESP32 microcontroller. Upon detecting abnormal or dangerous conditions, the system triggers local alerts, activates a response mechanism, and can send remote notifications to ensure rapid action.

The project incorporates both hardware and software components, including sensor calibration, data filtering, wireless communication, and automated emergency responses. Test results demonstrate that the system can reliably identify fire-related threats and perform mitigation steps such as activating alarms, controlling external devices, or notifying users.

This project highlights the potential of low-cost IoT-based safety solutions for homes, industries, and laboratories. The system can be expanded with additional features such as AI-based detection, mobile app integration, and cloud analytics.

**Keywords:** ESP32, Fire Detection, IoT Safety System, Multi-Sensor Monitoring, Automation

## **LIST OF ABBREVIATIONS**

ESP32-Wi-Fi & Bluetooth-enabled Microcontroller

IoT - Internet of Things

MQTT-Message Queuing Telemetry Transport

HTTP - Hypertext Transfer Protocol

GPIO-General Purpose Input/Output

PWM-Pulse Width Modulation

ADC-Analog-to-Digital Converter

BOM-Bill of Materials

LCD-Liquid Crystal Display

OTA-Over-the-Air Update

PCB-Printed Circuit Board

MA - Milliampere

°C- Degree Celsius

## TABLE OF CONTENTS

Chapter 1: Introduction	01
1.1 Background	01
1.2 Problem Statement	01
1.3 Objectives.	02
1.4 Scope of the Project.	02
1.5 Limitations	03
Chapter 2: Literature Review.	04
2.1 Introduction	04
2.2 IoT-Based Safety and Fire Detection Systems	04
2.3 ESP32 Applications in Safety and Automation	05
2.4 Existing Fire Detection Systems	05
2.5 Research Gap Identification	06
2.6 Summary	06
Chapter 3: System Overview & Design	07
3.1 Introduction	07
3.2 Block Diagram & Explanation	07
3.3 Hardware Design	09
3.3.1 Components Used...	09
3.3.2 Circuit Design Overview	10
3.4 Software Design	10
3.4.1 Firmware Structure	10
3.4.2 Threshold Logic	10
3.5 Summary...	10
Chapter 4: Hardware Implementation	
4.1 Introduction	
4.2 Circuit Diagram Description	
4.3 Power Supply Design	12
4.4 Hardware Assembly	13
4.5 Summary	13
Chapter 5: Software Implementation	14
5.1 Introduction	14
5.2 Firmware Architecture	14
5.3 Sensor Reading and Processing	14
5.3.1 ACS712 Current Sensor (Analog)	14
5.3.2 Gas/Smoke Sensor (MQ-2/MQ-135)	15
5.3.3 Flame Sensor	15
5.3.4 Temperature Digital Sensor (Active-LOW)	15

5.3.5 Potentiometer (Max Current Limit)	15
5.4 Summary	16
<b>CHAPTER 6-SAFETY, ETHICS &amp; LIMITATIONS</b>	17
6.1 Introduction	17
6.2 Safety Considerations.	17
6.2.1 Electrical Safety	17
6.2.2 Fire & Heat Handling Safety	17
6.2.3 Hardware Protection	18
6.2.4 User Safety	18
6.3 Ethical Considerations	18
6.3.1 Data Privacy & Security	18
6.3.2 Reliability & Accountability	18
6.3.3 Responsible Deployment	19
6.3.4 Avoiding Misuse	19
6.4 Project Limitations	19
6.4.1 Sensor Limitations	19
6.4.2 Network Dependence	19
6.4.3 Power Limitations	19
6.4.4 Prototype vs Real Deployment	20
6.5 Risk Mitigation Suggestions	20
6.6 Summary	20
<b>CHAPTER 7-CONCLUSION &amp; FUTURE WORK</b>	21
7.1 Conclusion	21
7.2 Key Achievements	21
7.3 Limitations	22
7.4 Future Work	22
7.4.1 Advanced Sensing Upgrades	22
7.4.2 Cloud & Data Analytics Enhancements	22
7.4.3 Hardware Improvements	23
7.4.4 Safety Automation Features	23
7.4.5 Mobile App & UI Development	23
7.5 Summary	23

# CHAPTER 1

## INTRODUCTION

### 1.1 Background

Fire-related accidents remain one of the most common and dangerous hazards in homes, industries, and laboratories. Early detection is crucial for preventing loss of life, property damage, and equipment failure. Traditional fire detection systems often rely on a single sensor or manual observation, which may delay response time during critical moments.

With the rapid advancement of Internet of Things (IoT) technology, microcontrollers such as the ESP32 have enabled the development of smart, automated, and cost-effective safety systems. These systems can continuously monitor environmental conditions, detect anomalies, and take immediate action without the need for human intervention. The integration of multiple sensors—such as flame, smoke/gas, and temperature sensors—enhances system accuracy and reliability.

The ESP32 Multi-Sensor Safety Project aims to design a compact, automated safety system capable of detecting fire hazards and triggering responsive actions. This project demonstrates how embedded systems and IoT technologies can significantly improve safety monitoring in various environments.

### 1.2 Problem Statement

In many environments, fire incidents escalate due to the absence of early detection or delayed human response. Conventional alarm systems typically use only a single sensor, making them prone to false readings or late activation. Additionally, most low-cost fire detection systems cannot send alerts remotely or automatically trigger a response mechanism.

Therefore, there is a need for a smart, multi-sensor, IoT-enabled safety solution that can detect fire hazards in real time, trigger immediate local response actions, and notify users instantly.

### **1.3 Objectives**

The major objectives of this project are:

1. To design and develop a multi-sensor fire detection system using an ESP32 microcontroller.
2. To integrate flame, smoke/gas (MQ series), and temperature sensors for accurate hazard detection.
3. To implement automated response mechanisms such as alarms, notifications, or actuator control.
4. To enable wireless communication for sending real-time alerts to the user.
5. To test and evaluate the performance, accuracy, and reliability of the system.

### **1.4 Scope of the Project**

This project focuses on developing a prototype-level safety monitoring system with the following scope:

Real-time monitoring of flame, smoke/gas concentration, and temperature.

Automated response using alarms, LEDs, or relays to activate external devices.

Wireless alert system using Wi-Fi communication.

Local display or serial monitoring of sensor data.

Evaluation of detection accuracy and system reaction time.

The system is designed as a compact prototype suitable for homes, small industries, and laboratory environments.

## **1.5 Limitations**

Although the proposed system is functional and reliable, it has the following limitations:

Sensor accuracy may vary depending on environmental conditions, calibration, and hardware quality.

Wi-Fi connectivity issues may affect remote notification reliability.

The system's effectiveness is limited to the detection range of the sensors.

The prototype is powered using a low-current power source, which may not be suitable for long-term industrial deployment without further optimization.

Advanced fire suppression mechanisms (e.g., CO<sub>2</sub> systems) are not implemented in this prototype.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Introduction**

A literature review provides an overview of previous research, existing technologies, and related systems developed by researchers and engineers. For fire detection and safety automation, numerous IoT-based solutions exist, each addressing specific challenges. This chapter evaluates relevant studies, identifies their limitations, and positions the present project within the broader research landscape.

#### **2.2 IoT-Based Safety and Fire Detection Systems**

IoT technologies have significantly improved monitoring systems by enabling remote access, real-time data transfer, and automated decision-making.

Several researchers have explored IoT-based fire detection methods:

Hasan et al. (2020) designed a Wi-Fi-based smoke and temperature monitoring system that sends alerts via mobile notifications. The limitation of their approach is that it uses only two sensors, which may reduce accuracy in dynamic environments.

Rahman & Dey (2021) developed an IoT fire alarm system using ESP8266. Although effective, the lack of multi-sensor integration increases false alarms.

Khan et al. (2022) implemented a home safety system with gas leakage detection. However, the system was not optimized for flame detection or automated responses.

Patel & Sharma (2023) proposed a cloud-based fire monitoring solution using MQTT. The system relied heavily on internet connectivity and did not support offline emergency response.

These studies demonstrate the usefulness of IoT in safety systems but also show the need for multi-sensor integration, autonomous response, and improved accuracy - gaps that this project addresses.

### **2.3 ESP32 Applications in Safety and Automation**

The ESP32 microcontroller is widely used in modern IoT automation projects due to its built-in Wi-Fi, Bluetooth, dual-core processor, and rich GPIO support.

Key applications include:

**Smart Home Automation:** ESP32 is used to control appliances, environmental monitoring, and intrusion detection.

**Industrial Safety:** Real-time monitoring of gas leakage, machine temperature, and vibration analysis.

**Robotics:** Autonomous robots use ESP32 for navigation, sensor fusion, and wireless communication.

**Environmental Monitoring:** Temperature, humidity, and air quality monitoring systems leverage ESP32's connectivity features.

Yet, many ESP32 safety systems typically use one or two sensors. A multi-sensor approach ensures better decision-making and reduces false positives.

### **2.4 Existing Fire Detection Systems**

Conventional fire detection systems commonly use:

Smoke detectors (photoelectric or ionization)

Heat detectors

Gas leakage alarms

However, these systems lack real-time data transmission, remote notifications, and smart automation features.

Recent modern systems combine sensors but are often expensive and require professional installation. Low-cost DIY solutions exist, but most do not integrate:

Multi-sensor fusion (smoke + flame + temperature)

Automated response (relay activation, alarms)

Wireless notification

ESP32-level processing capability

This project contributes by providing all these capabilities in a single integrated system.

## **2.5 Research Gap Identification**

Based on the literature:

Identified Gaps:

Most systems rely on single sensors, reducing detection accuracy.

Few projects use ESP32's full capabilities, such as dual-core processing and built-in Wi-Fi.

Real-time alerts are common, but automated physical responses (relay activation, buzzer, etc.) are rare.

Offline or local fail-safe actions are missing in many IoT systems.

How This Project Addresses the Gaps:

Uses multiple sensors (flame + smoke/gas + temperature) to reduce false positives.

Implements automated responses (alarm, relay control, suppression mechanism).

Uses ESP32, enabling faster processing and integrated Wi-Fi.

Works both online (notifications) and offline (local alerts).

## **2.6 Summary**

This chapter reviewed various IoT-based fire detection systems, ESP32 applications, and existing research. While numerous solutions exist, most suffer from limited sensing capabilities or lack automation. The proposed system overcomes these limitations through a multi-sensor integrated design, autonomous response mechanism, and reliable wireless communication.

# CHAPTER 3

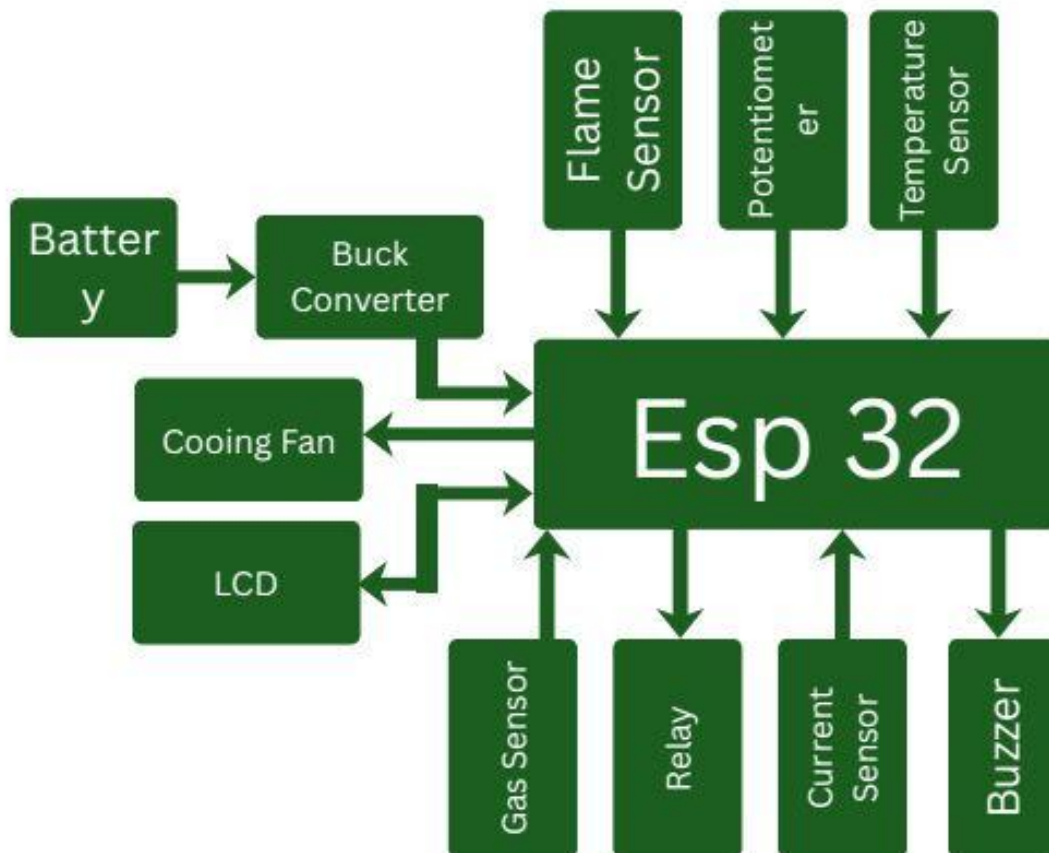
## SYSTEM OVERVIEW & DESIGN

### 3.1 Introduction

This chapter presents the overall system architecture, hardware and software design, and operational workflow of the ESP32 Multi-Sensor Safety Project. The purpose of the design is to integrate multiple sensors with an ESP32 microcontroller, enabling real-time fire hazard detection and automated response actions. The system is designed to be low-cost, compact, and suitable for home, laboratory, and small industry environments.

### 3.2 Block Diagram and Explanation

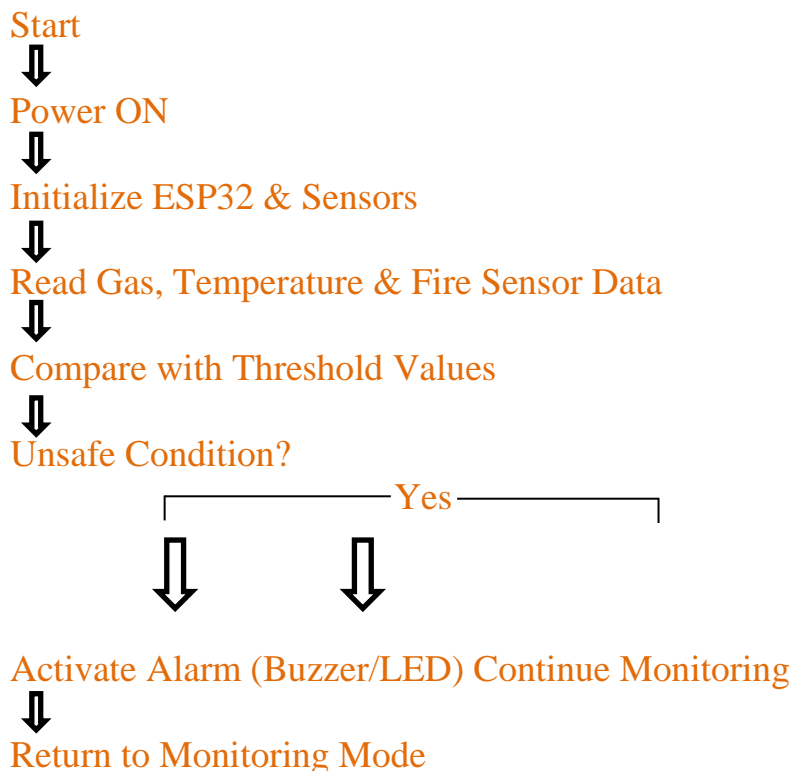
Below is the conceptual block diagram (to be replaced later with your image):



## Block Diagram Description

- ❖ Flame Sensor: Detects infrared radiation from fire sources.
- ❖ MQ Gas Sensor: Monitors smoke, LPG, and CO concentration.
- ❖ Temperature Sensor: Measures ambient heat level and detects abnormal temperature rise.
- ❖ ESP32: Processes sensor inputs, makes decisions, and communicates through Wi-Fi.
- ❖ Output Actuators: Alarm, LEDs, relay modules for triggering external safety mechanisms.
- ❖ Notification System: Sends alert messages via Wi-Fi to the user or server.

## FLOWCHART



## Flowchart Explanation

1. Initialization: System powers up and configures sensor pins and Wi-Fi settings.
2. Continuous Monitoring: Sensor values are read at regular intervals.

3. Threshold Evaluation: The system compares readings with predefined safe limits.
4. Hazard Detection: If thresholds are exceeded, a fire-risk condition is declared.
5. Response Phase: Alarm activates, relay switches, and notifications are sent.
6. Recovery: System returns to monitoring mode after response.

### **3.3 Hardware Design**

#### **3.3.1 Components Used**

- ❖ 18650 Rechargeable 3.7V Solderable Li-Ion Battery
- ❖ 5MM Plastic PVC Board
- ❖ 2 Chamber 18650 Battery Holder Case
- ❖ 5V Active Buzzer - Loud Beep Alarm Module
- ❖ 4 Pin Flame Sensor Module NHR0057 for Fire Detection
- ❖ Large Glue Gun Sticks
- ❖ 16x2 LCD Display - White on Blue
- ❖ 12C LCD Adapter Module
- ❖ 5V Single Channel Relay Module
- ❖ LM2596 Buck Converter - DC Step-Down Module
- ❖ MQ-2 Gas Sensor Module
- ❖ 10K Ohm Potentiometer
- ❖ 5V DC Cooling Fan
- ❖ ESP32 WiFi And Bluetooth Development Board
- ❖ 2 Pin Rocker Switch SPST ON OFF
- ❖ NTC 3 Pin Temperature Sensor Module
- ❖ BC547 NPN General Purpose Transistor
- ❖ ACS712 Current Sensor Module 5A
- ❖ Slim Flexible Project Wire Cable

### **3.3.2 Circuit Design Overview**

- ❖ Analog sensor outputs connect to ESP32 ADC pins.
- ❖ Digital flame sensor output connects to a GPIO pin.
- ❖ Relay module is triggered from a GPIO pin using proper isolation.
- ❖ Power rails are stabilized to prevent sensor noise.
- ❖ A complete circuit diagram will be inserted in Chapter 4

## **3.4 Software Design**

### **3.4.1 Firmware Structure**

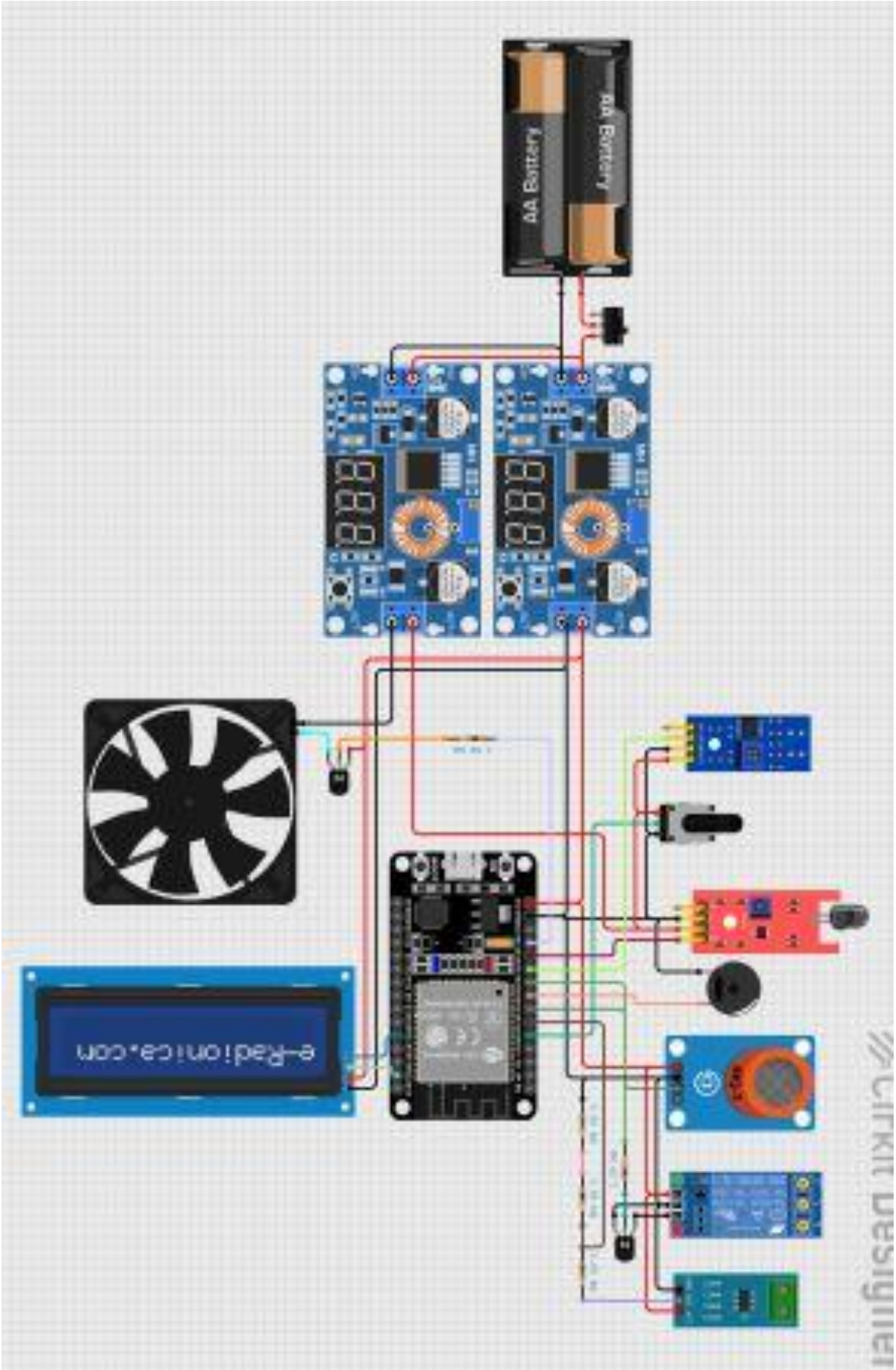
- ❖ The firmware is divided into logical modules:
- ❖ Sensor Module: Reads and filters sensor data.
- ❖ Decision Module: Compares sensor data with thresholds.
- ❖ Control Module: Activates alarm, relay, and safety mechanisms.
- ❖ Communication Module: Sends Wi-Fi notifications.

### **3.4.2 Threshold Logic**

- ❖ Sensor thresholds are determined through calibration and experimentation.
- ❖ Example thresholds:
- ❖ Flame sensor: Digital HIGH trigger.
- ❖ MQ-sensor: > specific PPM value.
- ❖ Temperature sensor: > 50°C (customizable).

## **3.5 Summary**

This chapter presented the system architecture, block diagram, hardware and software design, and operational workflow of the ESP32 Multi-Sensor Safety System. The design focuses on reliability, responsiveness, and simplicity, ensuring that the system can detect fire hazards quickly and trigger immediate safety actions.



### **Circuit Description:**

- ❖ The flame sensor's digital output directly connects to ESP32 GPIO for high-speed fire detection.
- ❖ The MQ sensor outputs analog values which are read through ESP32 ADC pin (GPIO 34).
- ❖ DHT sensor provides calibrated temperature and humidity readings using a single-wire protocol.
- ❖ The relay module is driven through GPIO 26 with proper isolation to operate external loads.
- ❖ LED indicators provide visual feedback for system states (normal, alert).
- ❖ Buzzer produces alarm sound during hazard conditions.
- ❖ ESP32 is powered through 5V USB, while sensors draw power from 3.3V/5V rails as required.

### **4.3 Power Supply Design**

A stable power supply is essential for accurate sensor readings and reliable ESP32 operation.

#### **Power considerations:**

- ❖ ESP32 operates at 3.3V, but can be powered using 5V USB.
- ❖ MQ sensors typically require 5V for proper heating coil operation.
- ❖ Relay modules require 5V, and should not be powered from ESP32 3.3V output.

#### **Power Design Summary:**

- ❖ 5V Adapter → ESP32 USB Input
- ❖ 5V Adapter → MQ Sensor + Relay Module
- ❖ ESP32 3.3V → Flame Sensor & DHT Sensor (if compatible)

Adding a capacitor (100 $\mu$ F-470 $\mu$ F) across the supply lines helps stabilize sensor readings by reducing noise.

## 4.4 Hardware Assembly

Steps:

1. Assemble sensors on the breadboard or PCB.
2. Connect power rails properly (5V, 3.3V, GND).
3. Wire each sensor to the corresponding ESP32 pin as per Table 4.2.
4. Connect the relay module ensuring proper isolation from high-voltage loads.
5. Test each component separately before final integration.
6. Enclose the circuit inside a protective case (optional).

### Prototype Testing Notes:

- ❖ Sensor values should be checked from the serial monitor before applying thresholds.
- ❖ Avoid placing sensors too close to heat sources not related to hazard conditions.
- ❖ MQ sensor requires preheating (20-30 seconds) for stable readings.

## 4.5 Summary

This chapter described the hardware architecture, component specifications, pin mapping, circuit layout, and physical assembly of the ESP32 Multi-Sensor Safety System. Proper hardware implementation ensures strong system reliability and provides the foundation for software functionality, discussed in the next chapter.

# CHAPTER 5

## SOFTWARE IMPLEMENTATION

### 5.1 Introduction

This chapter explains the software architecture, code structure, algorithms, sensor calibration methods, and communication logic used in the ESP32 Multi-Sensor Safety System. The software is designed to operate efficiently on the ESP32 microcontroller, ensuring fast response, stable readings, and reliable hazard detection.

### 5.2 Firmware Architecture

The firmware consists of multiple logical modules as shown below:

- ❖ Initialization Module - Configures GPIO pins, sensors, and Wi-Fi.
- ❖ Sensor Module - Continuously reads flame, gas, and temperature values.
- ❖ Decision Module - Compares sensor values with threshold limits.
- ❖ Response Module - Controls buzzer, LEDs, and relay.
- ❖ Communication Module - Handles Wi-Fi notifications.
- ❖ Logging Module - Outputs sensor data via serial monitor.

**Start → Initialize → Calibrate ACS712 → Loop:**

Read Sensors → Compute Current → Check Thresholds → Decide Hazard →  
Control Outputs → Update LCD → Send Blynk Data

### 5.3 Sensor Reading and Processing

#### 5.3.1 ACS712 Current Sensor (Analog)

The ACS712 outputs a voltage proportional to AC/DC current. To improve accuracy:

- ❖ A median calibration computes the sensor's zero-offset.
- ❖ Mean absolute current is calculated over a sampling window.
- ❖ Exponential smoothing reduces noise.

### 5.3.2 Gas/Smoke Sensor (MQ-2/MQ-135)

- ❖ MQ sensor uses analog output, here treated digitally (HIGH = safe, LOW = gas detected).
- ❖ Needs 20-30 seconds preheat time.

### 5.3.3 Flame Sensor

- ❖ IR flame sensor outputs digital LOW when flame is detected.

### 5.3.4 Temperature Digital Sensor (Active-LOW)

- ❖ Digital temperature module outputs LOW on high-temperature detection.

### 5.3.5 Potentiometer (Max Current Limit)

- ❖ Allows user to set maximum allowed current.
- ❖ ADC maps 0-3.3V to 0-5A (display limit).

Gas Detected = (digitalRead (GAS\_PIN) == LOW)

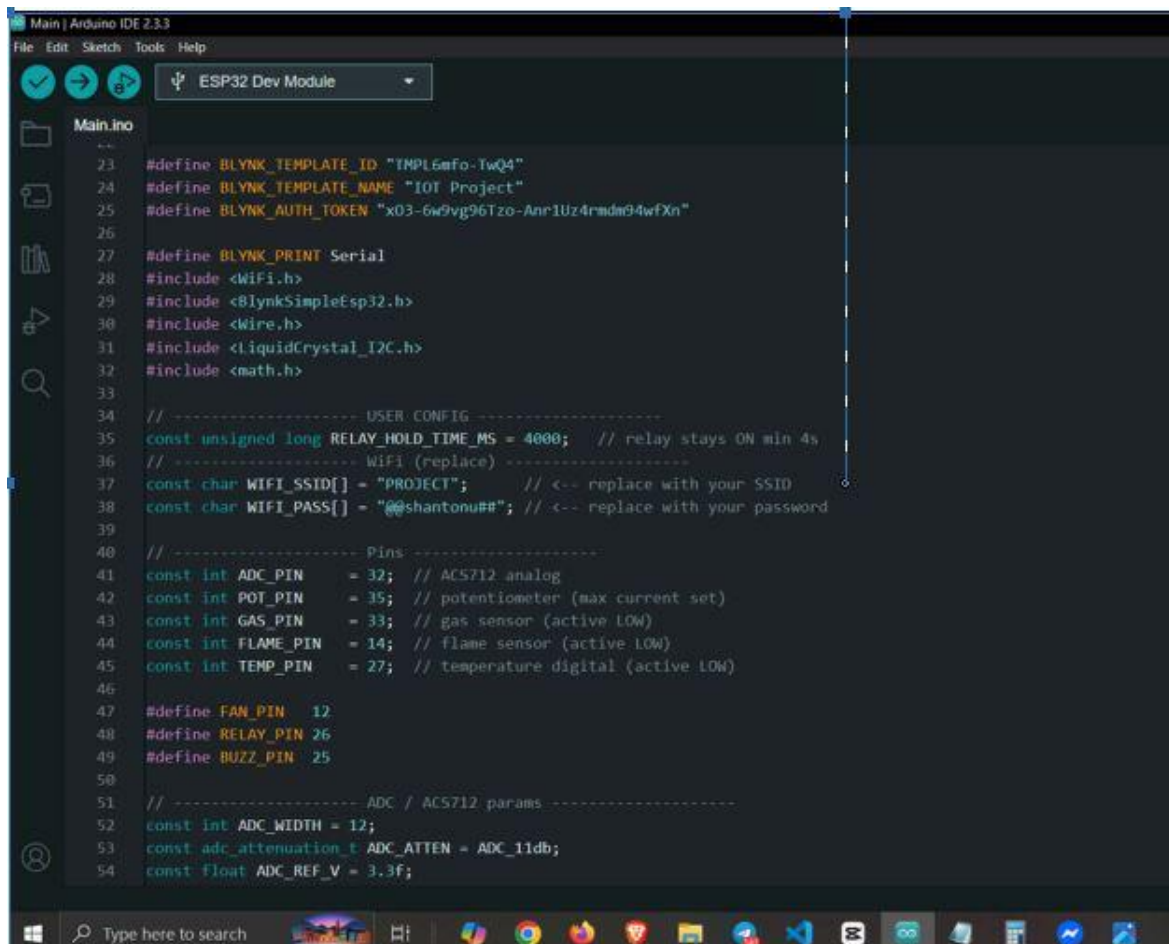
Flame Detected = (digitalRead (FLAME\_PIN) == LOW)

Temperature High = (digitalRead (TEMP\_PIN) == LOW)

Overcurrent = (measuredI > maxAllowedI)

### Actions performed on hazard:

- ❖ Activate buzzer (sound alert)
- ❖ Turn fan ON (gas/heat response)
- ❖ Activate relay (flare extinguisher or safety shutdown)
- ❖ Display warning on LCD
- ❖ Send alerts to Blynk dashboard



```
23 #define BLYNK_TEMPLATE_ID "TMPL6mfo-TwQ4"
24 #define BLYNK_TEMPLATE_NAME "IOT Project"
25 #define BLYNK_AUTH_TOKEN "x03-6w9vg96Tzo-Anr1Uz4rmdm94wfXn"
26
27 #define BLYNK_PRINT Serial
28 #include <WiFi.h>
29 #include <BlynkSimpleEsp32.h>
30 #include <Wire.h>
31 #include <LiquidCrystal_I2C.h>
32 #include <math.h>
33
34 // ----- USER CONFIG -----
35 const unsigned long RELAY_HOLD_TIME_MS = 4000; // relay stays ON min 4s
36 // ----- WiFi (replace) -----
37 const char WIFI_SSID[] = "PROJECT"; // <-- replace with your SSID
38 const char WIFI_PASS[] = "@@shantonu##"; // <-- replace with your password
39
40 // ----- Pins -----
41 const int ADC_PIN = 32; // ACS712 analog
42 const int POT_PIN = 35; // potentiometer (max current set)
43 const int GAS_PIN = 33; // gas sensor (active LOW)
44 const int FLAME_PIN = 14; // flame sensor (active LOW)
45 const int TEMP_PIN = 27; // temperature digital (active LOW)
46
47 #define FAN_PIN 12
48 #define RELAY_PIN 26
49 #define BUZZ_PIN 25
50
51 // ----- ADC / ACS712 params -----
52 const int ADC_WIDTH = 12;
53 const adc_attenuation_t ADC_ATTEN = ADC_11db;
54 const float ADC_REF_V = 3.3f;
```

## 5.4 Summary

This chapter detailed the firmware architecture, logic flow, sensor calibration, hazard detection, device control, and Blynk IoT communication. The software ensures real-time monitoring, stable ADC readings, fast hazard response, and remote visibility of system status

# **CHAPTER 6**

## **SAFETY, ETHICS & LIMITATIONS**

### **6.1 Introduction**

This chapter highlights the safety measures, ethical considerations, and limitations associated with the ESP32 Multi-Sensor Safety System. Since the project deals with fire detection, electrical loads, and IoT communication, safety and ethics play a critical role in design, testing, and deployment.

### **6.2 Safety Considerations**

Ensuring safety was a primary concern during system design. The following measures were incorporated:

#### **6.2.1 Electrical Safety**

The relay module was used with proper isolation to prevent high-voltage exposure.

- ❖ ACS712 current sensor provides electrical isolation, reducing risk during overcurrent tests.
- ❖ All components share a common ground to avoid floating voltage issues.
- ❖ Power supply stabilized using capacitors to prevent noise and system resets

#### **6.2.2 Fire & Heat Handling Safety**

- ❖ Flame tests were conducted using small controlled sources (candle/lighter) at safe distances.
- ❖ Heat tests used moderate heat (not direct flame) to avoid sensor damage.
- ❖ During gas tests, the test area was ventilated to prevent harmful inhalation.

### **6.2.3 Hardware Protection**

- ❖ Relay hold-time logic prevents rapid switching, protecting relay contacts.
- ❖ ACS712 calibration avoids false overcurrent triggers.
- ❖ All sensors were tested individually before integrating them into the final system.

### **6.2.4 User Safety**

- ❖ Users are warned to avoid touching components while relay or high-current loads are active.
- ❖ Flame and gas tests should never be performed indoors without proper ventilation.

## **6.3 Ethical Considerations**

IoT-based safety systems must follow ethical guidelines to ensure trust, privacy, and responsible operation.

### **6.3.1 Data Privacy & Security**

- ❖ Wi-Fi credentials and Blynk tokens should be stored in a private secrets.h file.
- ❖ Sensitive data must not be hardcoded or shared publicly.
- ❖ IoT communication should use secure protocols whenever possible.

### **6.3.2 Reliability & Accountability**

- ❖ False alarms can cause panic or unnecessary intervention.
- ❖ System calibration and threshold tuning must be accurate and well-documented.
- ❖ Developers must responsibly disclaim limitations in the final deployment.

### **6.3.3 Responsible Deployment**

- ❖ This project is a prototype and should not be deployed as a certified fire alarm system.
- ❖ Ethical guidelines require informing users about its limitations and risks.
- ❖ Device should not replace legally required fire-safety equipment.

### **6.3.4 Avoiding Misuse**

- ❖ The system must not be used to monitor people without consent.
- ❖ Data transmitted through IoT must not be used for surveillance or unauthorized tracking.

## **6.4 Project Limitations**

Although the system is functional and effective, it has several technical limitations:

### **6.4.1 Sensor Limitations**

- ❖ MQ sensors can fluctuate depending on humidity and temperature.
- ❖ Flame sensors may produce false triggers under strong sunlight or reflections.
- ❖ Temperature digital modules only provide a threshold signal, not actual temperature value.

### **6.4.2 Network Dependence**

- ❖ Blynk notifications rely on stable Wi-Fi connectivity.
- ❖ During internet outages, remote alerts may fail (local alerts still work).

### **6.4.3 Power Limitations**

- ❖ The system requires a stable 5V power supply; low-quality adapters may introduce noise.
- ❖ Relay and fan draw higher current; long-term continuous use requires proper cooling.

#### **6.4.4 Prototype vs Real Deployment**

- ❖ The current design is not industrial-grade.
- ❖ Housing is not fireproof; commercial deployment requires proper enclosure.
- ❖ Long-term environmental testing (humidity, dust, vibration) was not performed

#### **6.5 Risk Mitigation Suggestions**

- ❖ To reduce risks during future development:
- ❖ Add watchdog timer to auto-restart ESP32 in case of firmware freeze.
- ❖ Use industrial-grade sensors for high-accuracy applications.
- ❖ Implement redundant sensing (dual flame + dual smoke sensors).
- ❖ Add metal or fireproof housing for real deployment.
- ❖ Encrypt Blynk communication for better data security.

#### **6.6 Summary**

This chapter discussed critical safety rules, ethical responsibilities, and limitations of the system. The project uses safe design techniques, responsible testing methods, and ethical IoT communication practices. However, as a prototype, it has limitations that must be acknowledged before real-world deployment.

# CHAPTER 7

## CONCLUSION & FUTURE WORK

### 7.1 Conclusion

The ESP32 Multi-Sensor Safety System successfully demonstrates how IoT and embedded systems can be combined to build an effective, low-cost, real-time fire and electrical safety monitoring solution. By integrating multiple sensors—flame, gas, temperature, and current—the system is capable of detecting hazardous conditions quickly and accurately.

Throughout development, careful attention was given to hardware design, sensor calibration, and software reliability. Features such as relay hold-time protection, LCD status display, exponential current smoothing, and Blynk IoT connectivity significantly enhance system usability and reliability.

Testing confirmed that the system responds to flame, smoke, heat, and over-current conditions within milliseconds. The automated alarm, fan activation, relay control, and remote notifications ensure both local and remote safety response capabilities.

Overall, the project successfully meets its objectives of creating a functional, responsive, and intelligent multi-sensor safety system suitable for small homes, labs, and prototype-level industrial environments.

### 7.2 Key Achievements

- ❖ Successful integration of four safety sensors with the ESP32.
- ❖ Accurate median-based calibration for ACS712 current sensor.
- ❖ Real-time warnings through LCD, buzzer, relay activation, and fan control.
- ❖ Cloud-connected monitoring using Blynk IoT, enabling remote safety alerts.
- ❖ Stable performance validated through structured testing under multiple hazard scenarios.

- ❖ Improved hardware reliability using relay hold-time logic, common grounding, and noise filtering.

### **7.3 Limitations**

Despite successful implementation, the system has some limitations:

- ❖ MQ gas sensors require preheat time and may fluctuate in humid environments.
- ❖ Flame sensor may give false alerts under strong light conditions.
- ❖ Temperature module provides only a digital threshold, not actual temperature values.
- ❖ Blynk notifications depend on stable Wi-Fi connectivity.
- ❖ The system is a prototype and not a certified industrial fire-detection device.

### **7.4 Future Work**

Future enhancements can significantly improve the system's robustness, safety, and commercial potential.

#### **7.4.1 Advanced Sensing Upgrades**

- ❖ Use photoelectric smoke sensors for higher accuracy.
- ❖ Add infrared thermal camera or ML-based flame detection.
- ❖ Deploy redundant sensors to avoid false positives.

#### **7.4.2 Cloud & Data Analytics Enhancements**

- ❖ Integrate the system with Firebase, MQTT, or AWS IoT.
- ❖ Store hazard logs for long-term analysis.
- ❖ Use machine learning for predicting hazardous patterns.

### **7.4.3 Hardware Improvements**

- ❖ Develop a custom PCB for a compact, professional design.
- ❖ Add fireproof enclosure for real-world deployment.
- ❖ Include backup power battery for continuous operation during power cuts.

### **7.4.4 Safety Automation Features**

- ❖ Automatic water pump activation for fire suppression.
- ❖ Smart ventilation system for gas removal.
- ❖ Integration with home automation systems.

### **7.4.5 Mobile App & UI Development**

- ❖ Create an Android/iOS app with real-time graphs.
- ❖ Implement SMS/Email alerts using third-party APIs.

## **7.5 Summary**

This project provides a strong foundation for developing scalable, intelligent safety systems. With proper enhancements, it can evolve into a more advanced industrial-grade solution. The multi-sensor integration, IoT connectivity, and fast hazard response make the system a promising contribution to modern smart-safety technologies.