

# IOT Based Fire Risk Monitoring and Passenger Evacuation System

T. Sasi Priyanka,  
dept.of Electronics and  
Communication Engineering,  
Annamacharya Institute of Technology  
and Sciences,  
Kadapa, India.  
Sasipriyanka91717@gmail.com

Syed Abdul Subhan,  
dept.of Electronics and  
Communication Engineering,  
Annamacharya Institute of Technology  
and Sciences,  
Kadapa, India.  
syedabdulsubhan7396@gmail.com

Mulla Shahanaj,  
dept.of Electronics and  
Communication Engineering,  
Annamacharya Institute of Technology  
and Sciences,  
Kadapa, India  
mullashahanaj987@gmail.com

Thummaluru Aruneswari,  
dept.of Electronics and  
Communication Engineering,  
Annamacharya Institute of Technology  
and Sciences,  
Kadapa, India  
aruneswari1101@gmail.com

Vadishela Hemanth Kumar Reddy,  
dept.of Electronics and  
Communication Engineering,  
Annamacharya Institute of Technology  
and Sciences,  
Kadapa, India  
vhemanthvhemanth78@gmail.com

K. Lahari,  
dept.of Electronics and Communication  
Engineering, Annamacharya Institute of  
Technology and Sciences,  
Kadapa, India  
laharikdp@gmail.com

**Abstract—** Bus passengers are an important part of public transportation, and their safety is a major concern, especially during fire accidents. Such incidents can lead to serious consequences like injuries, loss of life, and damage to property. These accidents usually happen due to reasons like electrical faults, fuel leakage, overheating of engine parts, or poor maintenance of the vehicle. In many cases, the situation becomes worse because the fire is not detected early and action depends on manual response. In this work, we designed an IoT-based fire monitoring and automatic passenger evacuation system to improve safety inside buses. The system continuously monitors environmental conditions and takes necessary actions when a fire risk is detected. We used a dual-controller setup in our design. The Arduino Uno is responsible for reading sensor data and controlling safety operations, while the ESP32 module is used for wireless communication and sending data to the cloud. To facilitate passenger evacuation, fire and smoke sensors are employed to detect hazardous conditions, subsequently activating alert systems. These systems incorporate reduced motor activity, controlled exhaust airflow, and danger indicators. Should the perilous circumstance persist, a relay-based shutdown is initiated through a time-based escalation protocol. Furthermore, the system incorporates switches designed for failure simulation and manual override, thereby ensuring fail-safe functionality. Data from real-time systems is sent to a Firebase real-time database, allowing for remote monitoring and analysis. This system is well-suited for smart transportation safety applications, providing both adaptability and affordability.

**Keywords—** Internet of Things (IOT), Fire detection system, Passenger safety, Smart transportation, Passenger Evacuation System, Arduino, ESP32, Firebase

## I. INTRODUCTION

Due to their cost and availability, buses are among the most popular modes of transportation in both urban and rural locations. Millions of people depend on buses for daily transportation, especially in developing nations where they are an important part of the

transportation network [1]. While the increasing number of buses, passenger safety remains a major issue, especially when it comes to internal fire incidents. Fire incidents may not occur as much as traffic accidents, but when they do, the results can be very serious, including deaths, injuries, and property damage [2]. These fires can be caused by a number of factors, including electrical short circuits, fuel leaks, engine overheating, battery problems, problems in braking systems, or poor bus maintenance [3]. When a fire breaks out inside a bus, things can quickly get out of hand. Because the area is enclosed, smoke and heat spread quickly, making it difficult for passengers to breathe or see clearly. Evacuation may be delayed as a result, which frequently causes fear [4]. A lot of buses continue to depend on modest alarm systems, fire extinguishers, and driver actions for fire safety. These techniques primarily depend on human awareness and reaction time and are immediate, meaning they are used only after the issue is identified [5]. As a result, early warning indicators like smoke or high heat may not always be recognized right away, increasing the possibility that the problem may escalate. There are also very few conventional fire detection devices. Typically, they are made up of discrete sensors and alarm units that function independently, have set threshold levels, and have minimal capacity for making decisions [6], [7]. They are therefore poor to handle changing emergency situations. The latest developments in embedded systems and the Internet of Things (IoT) have improved fire safety opportunities. In order to continuously monitor the environment and identify threats early on, these modern technologies include sensors, microcontrollers, communication modules, and cloud platforms [8], [9]. In addition, they provide remote monitoring and real-time data sharing, and help in responding to emergencies more quickly and efficiently [10], [11], and [12]. However, a lot of buses still rely mostly on manual actions and simple alarm setups. Once

more, these methods depend on an individual's reaction time, and they could miss early warning indicators. Also, older systems with different components and set limitations lack the flexibility to handle complex situations. Newer methods are being investigated to

conventional systems [10]. Similarly, advanced systems that utilize a number of sensors and advanced algorithms can decrease false alarms and increase system dependability, particularly in safety-critical scenarios [13], [4]. These systems were helped from cloud platforms like Firebase, which enable data to be updated in real time from a central location and monitored remotely [14]. This makes it easier to continuously monitor system conditions. However, many of the current solutions still have certain drawbacks even these advancements. The majority of them primarily concentrate on fire detection and alerting, however they don't offer enough help for passenger safety measures or evacuation during emergencies, especially in bus settings [15], [16]. This work suggests an IoT-based fire risk monitoring system and an automatic passenger evacuation method to close these gaps. Using a variety of sensors, the system is built to continuously monitor environmental conditions and detect possible dangers early on. Important safety functions are managed by an Arduino-based controller, and cloud connectivity and communication are handled by the ESP32 module. To make sure the system is dependable even in the event of unforeseen circumstances or component failures, additional safety features like human override and fault simulation are added in addition to detection and monitoring.

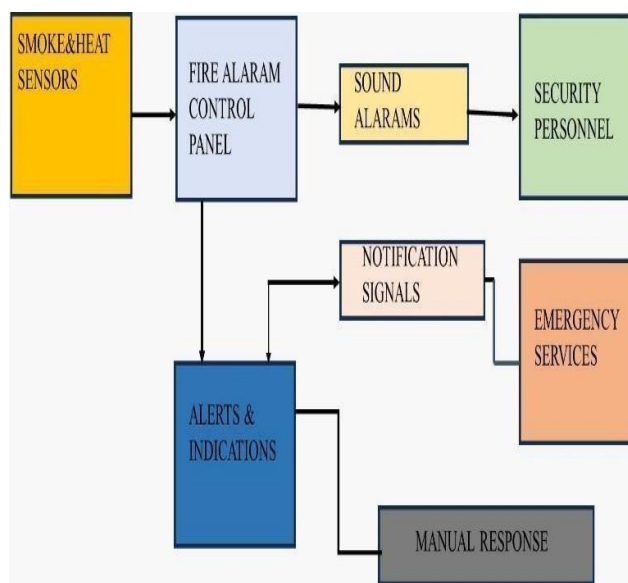
## II. RELATED WORK

Several The use of IoT and embedded technologies to enhance safety monitoring and fire detection has been extensively researched. These devices enable quicker emergency response times and aid in early detection of possible risks. Smoke levels, humidity, and the existence of flames within enclosed spaces can all be continuously monitored by linking sensors with microcontrollers and communication modules [1], [2]. IoT-based fire detection systems have gained increased attention in recent years due to their capacity to enable remote monitoring and offer real-time notifications. These systems often track changes in the environment and identify abnormal conditions using sensors that are integrated with microcontrollers. Alerts are sent using standard wireless communication techniques when any unusual situation is detected [3], [4]. They function better than conventional fire alarm systems because they can identify threats more rapidly and respond more quickly. Additionally, studies on the safety of smart vehicles have shown the need of integrating fire detection systems with automated safety measures. System reliability has been enhanced by the employment of numerous sensors working together, particularly in scenarios where safety is crucial. At the same time, real-time data collection and processing from sensors placed at various locations is made possible by cloud technology. Monitoring system status and emergency scenarios from a central place has

address these problems. To increase detection accuracy and overall system reliability, researchers are using machine learning algorithms and smart sensors. IoT-based solutions can identify fire hazards more quickly and react successfully than

gotten simpler because to real-time databases. To increase detection accuracy and lower false alarms, modern fire monitoring systems also make use of cognitive algorithms and networked sensors. However, a lot current systems primarily concentrate on detecting fire hazards and producing alarms in spite of these developments. They frequently do not have strong fail-safe mechanisms, which are crucial in emergency situations, and poorly support safe passenger evacuation. Likewise, certain systems are highly dependent on access to networks, which could compromise system dependability in a moment of need.[15], [16]. The suggested remedy combines automated evacuation assistance, a fail-safe mechanism, IO T-based inspection, and fire and smoke detection to overcome these constraints. The system attempts to offer robust fire detection, evacuation support, and distant surveillance for commuter security apps by fusing local embedded control with connections to the cloud.

## III. EXISTING



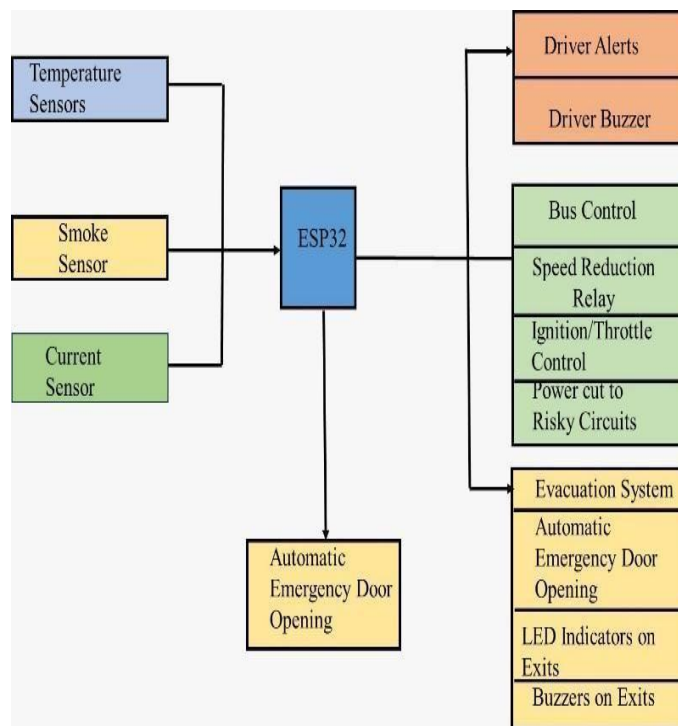
*Fig.1. Existing Block Diagram*

Since many years, conventional fire detection systems have been extensively utilized in commercial, industrial, and residential settings. These devices are made to identify situations that could result in a fire and sound local alarms to notify residents. Even though they offer rudimentary fire safety, current systems are inappropriate for contemporary smart settings because to their lack of intelligence, automation, and connectivity. Conventional systems mostly rely on independent sensing devices that are linked to a local alarm unit. Each component functions independently utilizing predetermined threshold values and a small amount of data processing power. The image below depicts the general layout of the current system. Among the most

often utilized parts of traditional fire detection systems is the smoke sensor block. Devices use either photoelectric or ionization-based sensing methods to find smoke particles in the air [14]. The sensor generates a signal indicating a possible fire condition when the concentration of smoke beyond a predetermined threshold. Smoke sensors are useful for spotting smoldering fires, however they are extremely sensitive to smoke from causes other than fires, like steam, dust, and cooking smells. This frequently results in false alarms, which can lower system trust. The main working concepts of a smoke sensor are ionization or light scattering. A photodiode and light source are positioned inside a sensing chamber in photoelectric sensors. The photodiode is not reached by light under normal circumstances. When smoke particles enter the chamber, they scatter light in the direction of the photodiode, creating an electrical signal that, when it surpasses a predetermined threshold, triggers the alarm. An ionization chamber in ionization sensors permits an electric current to pass through ionized air; when smoke particles reach the chamber, this current is disrupted, setting off the alarm. Temperature Sensor Block Heat sensors track variations in the surrounding temperature to identify fires [15]. When the temperature climbs above a predetermined threshold or when the rate of temperature rise surpasses a particular threshold, these sensors sound an alarm. Fixed temperature sensors and rate-of-rise sensors are the two main categories of heat sensors. Heat sensors have delayed detection even though they work well in challenging conditions where smoke sensors might not. Usually, fire is discovered only after a major rise in temperature, which permits the fire to spread before it is discovered. Flame Sensor Block Flame sensors use infrared (IR) or ultraviolet (UV) radiation from flames to detect fire. In industrial settings where open flames are anticipated, these sensors are frequently utilized. The warning is triggered by the sensor when radiation intensity surpasses a predetermined threshold [16]. However, when the flame is obscured or not clearly seen, flame sensors are useless. Additionally, artificial lighting or sunshine might cause them. Block of Control and Alarm Units In traditional systems, an alarm unit with a buzzer or siren is directly connected to sensor outputs. The alarm unit makes a loud noise to warn occupants when any sensor detects a dangerous situation [17]. The device has basic ON/OFF logic and does not interpret data or make decisions, therefore it is unable to assess the severity of a fire or integrate data from several sensors. Block of Power Supply Sensors and alarm devices receive electrical power from the power supply block. Conventional systems usually have a limited battery backup and rely on mains power. The backup battery is the only source of system reliability during power outages.

#### IV. PROPOSED SYSTEM

##### A. Block Diagram



*Fig.2. Proposed Block Diagram*

This project proposes a smart fire warning and passenger evacuation support system for sightseeing buses to improve safety during fire emergencies. It combines fire sensors, control units, evacuation support features, and IoT technology to build a reliable safety system. This system's primary goals are early fire threat detection, automated safety action, and real-time monitoring. Two controllers—the Arduino Uno and the ESP32—are needed to do this. The Arduino Uno serves as the main control unit, processing sensor data and supervising various safety functions. Conversely, the ESP32 module is in charge of supplying internet access through data transmission to a cloud platform, which allows remote monitoring and quick alert messages. Smoke and fire sensors are put inside the bus to continuously monitor the surroundings. These sensors can identify the presence of heat or flame in addition to smoke. The Arduino controller processes the signals from the sensors and determines if the situation is dangerous or normal by comparing them with predetermined threshold values. The device immediately changes from regular mode to emergency mode upon detection of any abnormal state. In order to facilitate passenger evacuation, several safety measures are activated during this phase. Visual indicators like LEDs and warning messages on an LCD screen provide information to both the driver and the passengers. Concurrently, an exhaust system is turned on to clear the bus of smoke and enhance breathing and visibility. Additionally, the system is built to continue operating even if certain components are malfunctioning, guaranteeing dependability in dire situations. IoT-based

remote monitoring is used in addition to local safety procedures. A cloud-based Firebase Realtime Database receives system status data from the ESP32 module, including information about smoke detection, fire status, and system shutdown conditions. This makes it possible for transportation authorities to save incident logs for additional study and remotely monitor bus safety conditions. All things considered, the suggested system offers an integrated safety solution that incorporates fail-safe design principles, automatic evacuation aid, early fire detection, and real-time IoT-based monitoring for enhanced transportation safety. The suggested system's block diagram shows how the cloud communication modules, computing units, actuator mechanisms, and sensor components interact. It is a high-level representation of the control logic and data flow needed to identify fire risks and start an emergency reaction. At the input stage, environmental conditions inside the bus passenger are continuously monitored by fire detection sensors such as the smoke and flame sensors. While the smoke sensor detects an amount of flammable gases and smoke particles, the flame sensor finds infrared radiation emitted by fire. When dangerous conditions are present, these sensors provide electrical signals. The Arduino Uno microcontroller, which serves as the system's central processing unit, is connected to the sensor outputs. The Arduino triggers the relay to cut off possible dangerous electrical loads when a risky state lasts for a specified period of time. When it comes to remote monitoring, the Serial connection is used by Arduino and an ESP32 module. Using Wi-Fi connectivity, the ESP32 serves as a wireless communication gateway that links the system to a cloud platform. Recording data and remote monitoring are made possible by uploading stored system status data to a Fire Realtime Database. All things considered, the block diagram shows how sensor, processing, actuation, and communication modules are integrated to produce a planned fire monitoring and evacuation support system. hazardous state persists for a predefined duration, the Arduino activates the relay to disconnect potentially dangerous electrical loads. For remote monitoring, the Arduino communicates with an ESP32 module using serial communication. The ESP32 functions as a wireless communication gateway that connects the system to a cloud platform through Wi-Fi connectivity. Processed system status information is uploaded to a Firebase Realtime Database, enabling remote monitoring and data logging. Overall, the block diagram demonstrates the integration of sensing, processing, actuation, and communication modules to create a coordinated fire monitoring and evacuation support system.

### B. Methodology

The suggested system employs a methodical approach that blends automated emergency response with ongoing environmental monitoring. It is intended to assist with passenger evacuation and identify dangerous situations early on while continuing to function dependably. The first step in the process is to use smoke and fire sensors to continuously monitor the bus interior. Based on the amount of smoke and the detection of flame radiation,

these sensors produce signals. These signals are routinely read and processed by the Arduino controller to determine the current state. The system stays in monitoring mode under typical circumstances. The bus runs normally throughout this phase, and sensor values are regularly monitored. The LCD display shows the system's present state; no emergency actions are initiated. At this point, the driver and passengers are alerted by warning indicators like buzzers and LEDs. In order to lower the amount of smoke inside the bus cabin, the exhaust ventilation system is also turned on. Simultaneously, the device starts tracking how long the dangerous situation lasts. The system goes into shutdown mode if the abnormal condition continues longer than a predetermined amount of time. To lower the danger of electrical risks, the relay module disconnects critical electrical circuits at this point. While circulation keeps running to keep smoke out of the air, the evacuation mechanism is turned on to help passengers leave. In order to ensure safe functioning in the event of a component failure, the system additionally includes reliability mechanisms. In the event that the automated system malfunctions, an extra switch enables the driver to manually operate the evacuation system. During testing, a fault simulation switch is utilized to confirm the dependability and fail-safe behavior of the system.

### C. Hardware Implementation

Techniques and communication modules to achieve efficient evacuation support and fire monitoring. The system's primary control unit is the Arduino Uno microcontroller. It handles sensor data, runs algorithms for making decisions, and manages the functioning of output devices. Embedded firmware is used to program the controller, which controls system states like shutdown mode, alert situations, and regular operation. Two main sensing devices are used to detect fires. The bus room's combustible gases and smoke are detected by the MQ-2 smoke sensor. The infrared radiation that fire sources emit is detected by the flame sensor. The Arduino controller continuously monitors the sensor outputs to detect potentially dangerous situations. Real-time system information is shown on a 20x4 I2C LCD display. Operational messages such as system shutdown status, smoke detection alerts, and fire detection alerts are displayed on the screen. The driver and passengers can rapidly comprehend the state of the system thanks to this interface. The system's motor-based air and evacuation systems are controlled by an L298N motor driver module. The evacuation door is operated by one motor, while the exhaust fan that removes smoke is driven by another motor. Using PWM signals, the Arduino gives the motor driver direction and speed control signals. There is a relay module to disconnect electrical circuits in an emergency. The relay reduces the risk of fire escalation by disconnecting potentially dangerous electrical loads when it detects extreme fire conditions. An ESP32 module serves as the system's wireless communication interface for Internet of Things connectivity. In order to post system status data to the Firebase Realtime Database, the ESP32 connects to a

Wi-Fi network and uses serial communication with the Arduino. The system is powered by a separate power source for motor operation and a regulated power supply for control electronics. This arrangement provides enough power for mechanical actuation while ensuring the control system operates continuously. In order to provide an accurate and scalable fire monitoring and passenger evacuation support system, the hardware architecture includes sensing, processing, action, and communication elements.

**V. RESULTS AND DISCUSSION**

The suggested IoT-based fire monitoring and passenger evacuation system was created and tested under controlled circumstances to assess how well it can identify fire threats and assist emergency evacuation. An Arduino Uno, ESP32 communication module, MQ-2 smoke sensor, flame sensor, motor driver, relay unit, and a 20x4 LCD display were among the parts of the system. The system continuously tracked environmental factors, such as flame presence and smoke levels, while it was operating normally. Smoke and flame sources were placed close to the sensing area for testing reasons. These unusual circumstances were correctly identified by the sensors, who then reacted appropriately. To assess its efficacy in identifying fire danger and facilitating emergency evacuation, the suggested IoT-based Fire Risk Monitoring and Automatic Passenger Evacuation System was created and placed through controlled testing. An Arduino Uno, ESP32 communication module, MQ-2 smoke sensor, flame sensor, motor driver module, relay unit, and a 20x4 LCD display were used to create the initial system. The system continuously recorded environmental parameters like smoke density and flame detection signals while it was operating normally. When smoke or flame sources were placed close to the detecting region, the sensors were able to identify aberrant situations. The system switched from regular mode to alert mode as soon as the sensor values above the predetermined threshold levels. In a copied bus area environment, replicating controlled evacuation conditions, the motor speed was lowered. This phase gave people a heads-up and gave them enough time to prepare for an evacuation. The system went into shutdown mode if the dangerous situation continued for a predetermined amount of time. At this point, the relay module maintained ventilation support while isolating electrical circuits to stop additional risks. In order to facilitate passenger departure, the evacuation system was also triggered. The reliance on individuals was greatly decreased by these automated replies. To confirm the ability to monitor remotely, the IoT feature was also tested. Through Wi-Fi connectivity, the ESP32 module received system status parameters to the Firebase Realtime Database, including the shutdown state, smoke level status, and fire detection state. Effective cloud connectivity was confirmed by the ability of remote monitoring interfaces to show system updates in real time. Overall, the experimental results show that the suggested system may initiate automated evacuation support mechanisms, detect fire dangers early, and offer remote monitoring capabilities. For emergencies, the

incorporation of emergency methods like failure simulation and manual override improves system reliability even more.

TABLE – I

Comparison of Fire Detection System

Feature	Conventional Fire Detection Systems	Proposed IoT-Based System
Fire Detection	Basic detection using alarms	Multi-sensor detection (smoke + flame)
Response Mechanism	Manual response required	Automated alert and safety response
Passenger Evacuation Support	Not available	Motor-assisted evacuation mechanism
Smoke Ventilation	Not included	Automated exhaust ventilation
Electrical Safety	Limited	Relay-based shutdown protection
Remote Monitoring	Not available	IoT-based real-time monitoring
Fail-Safe Mechanisms	Rarely implemented	Manual override and fault simulation
Data Logging	Not supported	Cloud-based event logging

**VI. CONCLUSION**

This In order to increase bus safety, this project describes the design and implementation of an Internet of Things-based fire risk monitoring and passenger evacuation system. The system combines remote monitoring, automatic reaction, and sensing into one framework. Reliability and real-time connectivity are guaranteed by the system's use of ESP32 for communication and Arduino for local control. System efficiency is increased by adding various operating modes, such as shutdown, alert, and normal. Reliability is further improved by features like fault simulation and human override, which enable safe operation even in the event of a breakdown. Through cloud integration, the system effectively displays early fire detection, automated safety response, and real-time monitoring. This strategy can help create safer and more intelligent transportation systems with additional enhancements and practical application.

**REFERENCES**

- [1] Smart Transportation,," IEEE Sensors Journal, 2024.
- [2] Google Firebase Documentation, "Firebase Realtime Database," 2024.
- [3] World Bank, "Public Transport Systems in Developing Countries," 2022.
- [4] National Fire Protection Association (NFPA), "Vehicle Fire Statistics," 2023.



- [5] R. Mehta and A. Sharma, "Analysis of Vehicle Fire Accidents," Safety Science Journal, 2022.
- [6] Elhanashi, "Early Fire and Smoke Detection Using Deep Learning," Applied Sciences, 2025.
- [7] A. Kumar and S. Gupta, "Conventional Fire Safety Mechanisms in Public Transport," IJRASET, 2023.
- [8] J. Brown et al., "IoT Applications in Transportation Safety," IEEE Internet of Things Journal, 2024.
- [9] Safety Journal (MDPI), "Comprehensive Review of IoT Enabled Fire Safety Systems," 2025.
- [10] H. Smith, "Smoke Detection Techniques and Technologies," Fire Safety Journal, 2022.
- [11] P. Zhao, "Heat Detection Systems for Fire Safety Applications," IEEE Sensors Conference, 2023.
- [12] TWang, "Flame Detection Methods in Industrial Environments," IEEE Transactions on Industrial Electronics, 2021.
- [13] S. Gupta and R. Verma, "Alarm Systems for Fire Detection," International Journal of Electronics, 2020.
- [14] R. Kumar, "Power Management in Safety Systems," Electronics World Journal, 2021.
- [15] JAZ India Journal, "Emergency Evacuation by IoT in Private Bus," 2024.
- [16] CEUR Workshop Proceedings, "Fire Safety Monitoring System Based on IoT," 2025.

