



A Reconfigurable Fire Detection Platform with Live Monitoring and Automated Actuation Based on Sensor Data Analytics

Amila Dhanushka Karunanayaka ^{a*}, Sandin Dulsana De Soysa ^b

*^aDepartment of Electrical and Electronic Engineering, Faculty of Engineering, Sri Lanka Institute of
Information Technology, Malabe 10115, Sri Lanka*

*^bDepartment of Mechanical Engineering, Faculty of Engineering, Sri Lanka Institute of Information
Technology, Malabe 10115, Sri Lanka*

^aEmail: karunanayakaamila197@gmail.com

^bEmail: sandindesoysa@gmail.com

Abstract

This paper presents a robust and intelligent fire detection and response system that leverages a multi-sensor configuration that includes DHT11, MQ-2, and LDR with dynamically tunable threshold controls managed through a LabVIEW-based GUI. In contrast to conventional fire alarm systems that rely on rigid, hardcoded thresholds prone to false positives due to environmental variability or industrial maintenance, the proposed system enables real-time threshold reconfiguration without firmware modifications. Sensor data is acquired via an Arduino microcontroller and transmitted to LabVIEW, which applies a threshold-based decision logic to transmit a serial command that actuates a suite of response mechanisms. Upon fire detection, the system activates an auditory buzzer, visual LED indicators, an exhaust fan, and a camera while triggering a servo-operated emergency exit. A real-time camera feed allows human-in-the-loop verification, while a hardware emergency override ensures safe manual reset in the event of false alarms. In non-critical states, an ultrasonic proximity sensor enables automated, contactless door operation. Experimental validation showed an average response time of 2.4 seconds from the moment a sensor threshold was breached to full actuator activation. The standard deviation across trials ranged from 0.2 to 0.5 seconds, reflecting only minor variability due to ambient conditions such as temperature,

humidity, light, and gas concentration. The proposed architecture enhances adaptability, reduces false triggers, and improves operational safety, rendering it highly suitable for dynamic industrial environments and intelligent building systems.

Keywords: Fire Detection System; LabVIEW; Multi-Sensor Integration; Real-Time Threshold Adjustment; Arduino Microcontroller; Industrial Safety; Smart Building Automation.

* Corresponding author. Email address: karunanayakaamila197@gmail.com

1. Introduction

The increasing demand for intelligent safety systems in dynamic industrial and residential infrastructures has underscored the need for advanced fire detection technologies that go beyond traditional rigid configurations. Conventional fire alarm systems, typically reliant on hardcoded thresholds for parameters like temperature, gas concentration, and illumination, often lack adaptability in fluctuating environments [1]. As a result, these systems are prone to false alarms triggered by routine activities such as welding, exhaust emissions, or lighting changes, as well as by sensor drift over time [2], [3]. Such false triggers can lead to operational inefficiencies, resource wastage, and a reduced sense of urgency during real emergencies.

These limitations become even more pronounced in industrial or high-variability settings. For example, temperature and humidity conditions may shift significantly between day and night or across seasons, affecting fire detection accuracy if not dynamically accounted for [4]. Similarly, manufacturing environments with persistent ambient gas emissions may yield false positives or negatives depending on the operating status of nearby machinery [5]. Therefore, there is a growing demand for intelligent systems that can adapt to changing environmental baselines without requiring manual recalibration or firmware modification.

Early fire detection systems relied on standalone smoke detectors, thermistors, or infrared sensors, interfaced with basic logic circuits or microcontrollers using static thresholds [3]. While these systems offered simplicity and low cost, they were not equipped to handle complex, real-time environmental dynamics. As microcontroller platforms such as Arduino [6] and Raspberry Pi [7] became more prevalent, fire detection frameworks began to incorporate multiple sensors such as temperature, gas, flame, and smoke detectors to improve detection accuracy through basic data fusion. However, most of these systems still relied on fixed threshold values, with any adjustment requiring code changes and re-deployment, which is an impractical approach for end-users or dynamic deployments [5].

Recent academic advancements have introduced machine learning algorithms, fuzzy logic, and Internet of Things (IoT)-enabled platforms to enhance the intelligence and accuracy of fire detection systems [8]. These methods, while effective in pattern recognition, often introduce latency due to remote server dependence and may be unsuitable in mission-critical environments that demand fail-safe, autonomous operation. Furthermore, many cloud-based systems require constant internet connectivity, limiting their reliability in offline or isolated deployments [9].

In this context, visual programming platforms like LabVIEW present a compelling alternative. While LabVIEW has been used in previous studies for real-time data acquisition and graphical monitoring [5], its full potential as a runtime-configurable control environment remains underutilised. The platform offers powerful features for real-time sensor monitoring, threshold adjustment, and actuator control through a user-friendly GUI that eliminates the need for code-based reconfiguration and makes the system accessible to non-programmers [8], [10].

The system proposed in this paper addresses the limitations of earlier models by integrating a multi-sensor hardware platform, with comprising DHT11 [11], MQ-2 [11], and an LDR with a LabVIEW-based adaptive control interface [1]. An Arduino microcontroller bridges the sensor array and LabVIEW, enabling real-time data acquisition, graphical behaviour monitoring, dynamic threshold configuration, and actuator activation. Upon detecting fire conditions based on user-defined thresholds, the system initiates a coordinated response including visual and auditory alerts, exhaust fan operation, emergency door opening via servo motor, and camera activation for live monitoring.

A key innovation of the proposed platform lies in its dual-layer safety architecture. Automated decisions based on sensor data are augmented with human-in-the-loop verification via a real-time camera feed [3]. In cases of false alarms, users can override the system using a physical emergency toggle switch. An ultrasonic proximity sensor is also employed for contactless emergency exit control under normal operating conditions while improving daily usability in high-traffic environments.

By combining intelligent sensor integration, real-time threshold control, human-verified decision-making, and a modular hardware-software architecture, this system offers a significant leap forward in the development of adaptive fire safety solutions. Its deployment potential spans smart buildings, industrial facilities, and critical infrastructure, where safety, responsiveness, and configurability are paramount [2], [4].

2. System architecture

The proposed adaptive fire detection and response system is organised into three functional layers that are known as Sensing, Processing, and Actuation, as depicted in Figure 1. This modular structure supports real-time responsiveness, scalability, and simplified maintenance in safety-critical environments.

2.1. Sensing layer

The sensing layer includes four primary sensors, each selected to monitor key environmental indicators related to potential fire events [3]. The **DHT11** sensor measures ambient temperature and relative humidity with a digital output resolution of $\pm 1^\circ\text{C}$ and $\pm 5\%$ RH [4]. Positioned to detect early-stage thermal anomalies, it provides foundational data for identifying heat and moisture variations that may indicate combustion.

Complementing this, the MQ-2 gas sensor detects flammable gases such as methane and liquefied petroleum gas (LPG), helping identify combustible airborne compounds that often accompany fire events [5]. In addition to detecting ignition precursors, the MQ-2 also plays a vital role in characterising airborne particle behaviour before and after a fire event, enabling the system to track the persistence of smoke and gas residues during post-fire recovery phases. A calibrated **LDR** monitors ambient illumination levels to capture visual anomalies such as abrupt changes in light intensity caused by smoke accumulation or flame flickering [2]. Additionally, an **ultrasonic sensor (HC-SR04)** is installed near emergency exits to detect human proximity. It enables automated, contactless door operation under non-critical conditions, enhancing everyday system usability [8].

All sensor data is routed through the **Arduino Uno**, which performs analogue-to-digital conversion (ADC) where necessary. The microcontroller aggregates these inputs into structured data packets and transmits them to the processing layer at one-second intervals over a serial connection. This sampling rate ensures a balance between real-time resolution and processing efficiency [6].

2.2. Processing layer

The processing layer is implemented using **LabVIEW**, which serves as the computational hub for interpreting sensor data from the Arduino. A custom-developed **Virtual Instrument (VI)** parses the incoming data stream and provides a **user-friendly GUI unit** for defining and adjusting threshold values for each parameter, known as temperature, humidity, gas concentration, and illumination [8].

These thresholds are continuously compared against live sensor inputs using a Boolean condition matrix embedded in the VI logic. If any parameter exceeds its defined threshold, LabVIEW interprets this as a fire condition and transmits a control signal ('1') back to the Arduino [3]. If all values remain within safe ranges, a '0' is sent. This lightweight, binary decision scheme minimises communication overhead while ensuring robust, real-time responsiveness, a critical requirement for fire safety systems [2], [4].

2.3. Actuation layer

Upon detection of a fire condition, the actuation layer is engaged to initiate coordinated safety measures. This layer includes a **buzzer** to provide an audible alarm for nearby individuals [2], and an **LED indicator** that remains continuously lit during a fire event or blinks during standby mode for system status feedback [3].

An **exhaust fan** is activated to ventilate smoke, improving visibility and reducing air contaminants [4]. Simultaneously, a **servo motor** controls the emergency exit mechanism, automatically opening the door during emergencies or when human presence is detected via the ultrasonic sensor [8]. A **camera module** is also activated to provide real-time video streaming of the affected zone, supporting human-in-the-loop verification. Together,

these actuators respond dynamically to fire conditions while offering both autonomous operation and manual override capability via an emergency toggle switch, ensuring operational safety and flexibility [5].

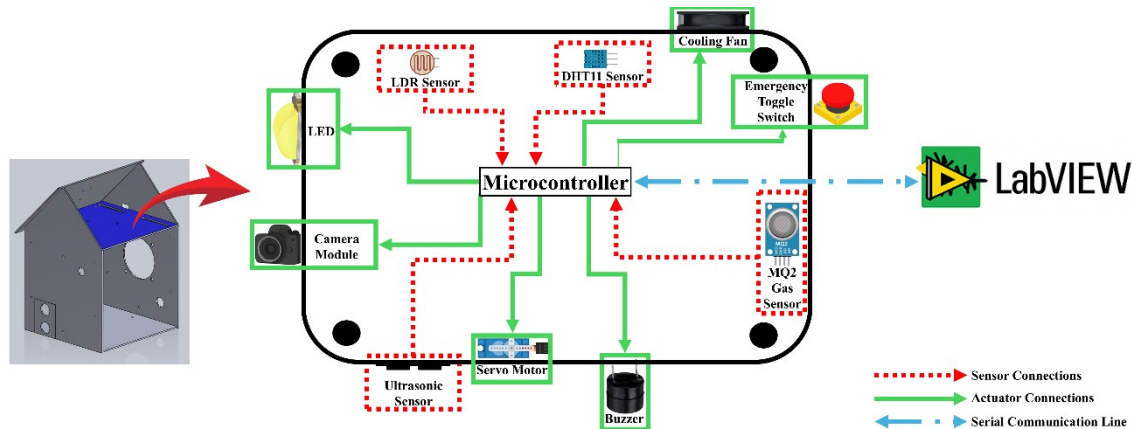


Figure 1: Overview diagram of the proposed solution.

3. Implementation

The implementation of the adaptive fire detection and response system was realised through the seamless integration of hardware components and software developed using Arduino Integrated Development Environment (IDE) and LabVIEW. This stage encompassed hardware interfacing, embedded logic design, graphical interface programming, and system packaging for reliable and user-friendly operation.

3.1. Firmware development

The Arduino Uno was programmed using the Arduino IDE with a structured loop-based approach to continuously read environmental parameters from the connected sensors, which included multiple sensors. These readings were formatted into a comma-separated string and transmitted every second to LabVIEW via a serial connection operating at a baud rate of 115200. Simultaneously, the Arduino listened for incoming control signals from LabVIEW, encoded as binary integers. Upon receiving this signal, the Arduino executed conditional routines that activated corresponding actuators. Servo control was handled using Pulse Width Modulation (PWM) outputs, while digital input polling managed emergency button interrupts, ensuring responsive and safe manual overrides.

3.2. LabVIEW VI design

The graphical programming environment LabVIEW 2021 [8] was employed to develop a VI that served as both a monitoring and control hub for the system [5]. The front panel of the VI was designed to provide real-time graphical plots of all four sensor inputs, enabling users to monitor environmental conditions visually and intuitively [8]. Adjustable threshold fields allowed users to dynamically fine-tune the detection criteria, thereby offering high configurability without requiring changes to the firmware. Additionally, visual status indicators, such as a prominently displayed “FIRE” alert, were used to enhance situational awareness. In the block diagram section of the VI, Virtual Instrument Software Architecture (VISA) [10] communication blocks facilitated serial data exchange, while Boolean logic gates processed incoming sensor values to determine the fire status [3]. This logic yielded a binary decision that was communicated back to the Arduino to trigger appropriate actions in the actuation layer [2]. Figure 2 illustrates the custom-developed LabVIEW VI interface designed for this system, which facilitates real-time monitoring, threshold configuration, and control signal generation based on sensor data analysis.

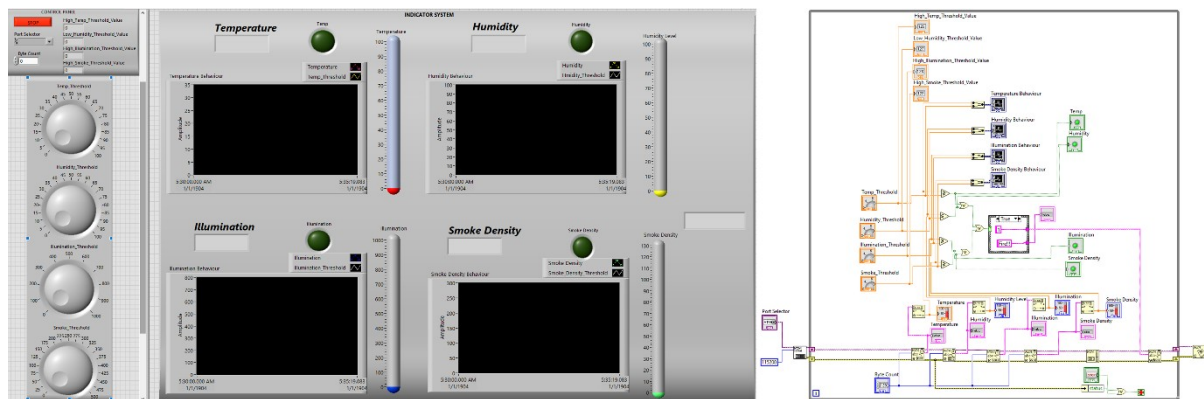


Figure 2: Custom LabVIEW VI for Real-Time Monitoring and Control.

3.3. Custom printed circuit board (PCB) and 3D housing development

To enhance the durability, compactness, and deployability of the system, a PCB was designed to host all essential circuit connections, used during initial prototyping. The PCB layout was optimised to minimise electromagnetic interference and ensure stable signal transmission between the sensors and the Arduino. It included dedicated tracks for power regulation, signal lines, and emergency override circuitry. Additionally, a 3D housing structure in the shape of a miniature house was designed. This housing served both aesthetic and functional purposes, enclosing all electronic components while mimicking the typical environment where the system might be deployed. The design incorporated vents for airflow, cutouts for sensor exposure, and a hinged access door for easy maintenance. The camera module was embedded inside the enclosure, enabling an unobstructed view for surveillance during emergency conditions. This enclosure not only protected the internal components from dust and mechanical damage but also provided a realistic demonstration platform for testing the system's performance in simulated home environments. Figure 3.(a) presents the schematic diagram that formed the foundation for custom-designed PCBs, along with the finalised 3D-printed enclosure modelled as present in Figure 3.(b).

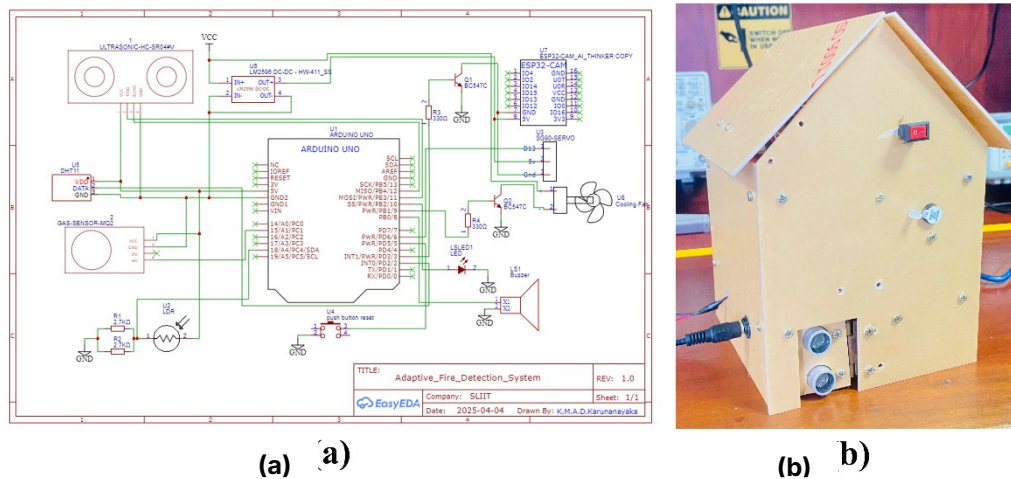
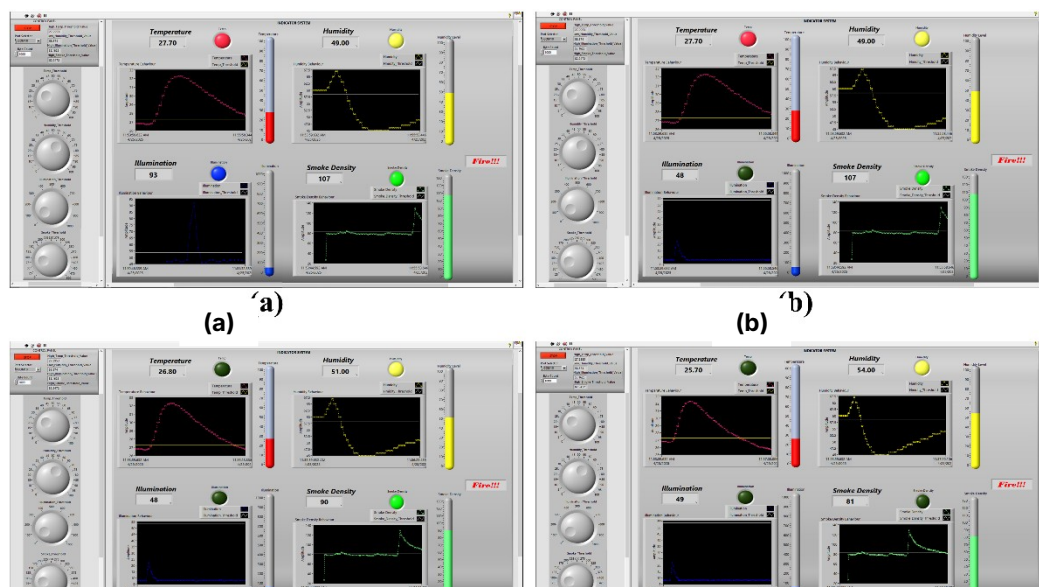


Figure 3: (a) Schematic diagram that formed the foundation for the custom-designed PCBs;

(b) 3D-printed enclosure designed to house the fire detection system components.

4. Experimental results

The effectiveness of the proposed system was validated through a staged fire simulation, during which all sensor readings, actuator responses, and environmental recovery were monitored and documented in sequential phases, as illustrated in Figure 4(a–f). The integrated camera module provided continuous visual verification of environmental changes and actuator behavior, corroborating sensor-based decisions throughout the experiment.



In the initial phase as shown in Figure 4.(a), the simulated fire event caused all four sensor parameters, which are temperature, humidity, illumination, and smoke density, to exceed their defined thresholds. The DHT11 sensor detected a rapid rise in both temperature and relative humidity, the MQ-2 sensor reported elevated gas concentrations, and the LDR registered increased illumination levels corresponding to the visible flame [1]. LabVIEW processed these anomalies and generated a high-level (1) control signal, initiating full system actuation: the buzzer emitted an audible alert, yellow LEDs illuminated, the exhaust fan was activated, real-time camera streaming commenced, and the servo-controlled emergency door was opened. As the exhaust fan continued operating, the intensity of the fire diminished, resulting in normalised illumination levels according to Figure 4.(b). This was reflected by the LDR readings returning below the defined threshold [1]. However, residual heat, moisture, and airborne gases kept the temperature, humidity, and smoke density above safe limits, maintaining the system in an active alert state [4].

With continued ventilation, the ambient temperature was successfully reduced to within acceptable limits, as indicated by the DHT11 sensor, as shown in Figure 4.(c). Illumination remained stable, confirming the absence of flames. Despite this, elevated humidity and gas concentrations persisted due to lingering vapours and smoke particles from the extinguished fire [8]. Subsequently, in Figure 4.(d), the MQ-2 readings fell below the safety threshold, indicating effective removal of combustible gases and particulate matter by the exhaust fan. The visual feed confirmed an increasingly clear environment. However, relative humidity remained marginally high due to water vapour trapped in the enclosure, ultimately delaying system reset.

In Figure 4.(e), all environmental readings returned to safe operational thresholds. The LabVIEW interface automatically issued a low-level (0) control signal to the Arduino, triggering the deactivation of all actuators. The emergency door closed, LEDs reverted to standby blinking mode, and the system transitioned into idle. Throughout the experiment, the integrated camera module confirmed each phase of environmental recovery, validating sensor accuracy and system autonomy. The manual override feature remained unused during the experiment, underscoring the system's reliability in fully automated operation; however, it remains available as a critical safeguard in the event of false alarms or unexpected environmental anomalies [2]. Finally, Figure 4.(f) illustrates the system's emergency shutdown procedure activated after a false alarm is detected. Once the camera module confirms the absence of an actual fire, the user can press the emergency override button to manually halt all actuator functions and reset the system. This safety feature ensures that false positives do not cause unnecessary disruption, allowing the system to safely return to standby mode while maintaining readiness for future detection events.

To thoroughly evaluate the responsiveness and operational consistency of the proposed adaptive fire detection and response system, an extensive series of structured experimental trials was carried out under a wide range of simulated environmental conditions. These trials were meticulously designed to replicate real-world scenarios, thereby ensuring that the system's behaviour could be tested in both controlled and dynamically changing environments. Key environmental factors such as ambient temperature, relative humidity, light intensity, and the concentration of combustible gases were systematically varied during each experimental session. These variations

were achieved by altering the physical testing location, the time of day, and the presence or absence of environmental stimuli (such as artificial lighting or smoke sources), thereby simulating practical deployment settings like industrial kitchens, warehouses, residential interiors, and outdoor semi-enclosed spaces. Throughout the testing process, the system's sensors operated in continuous monitoring mode, vigilantly tracking environmental parameters in real time. When sensor readings surpassed their predefined threshold values, either due to gradual changes or abrupt anomalies, the corresponding actuator modules, such as exhaust fans or alert mechanisms, were automatically and promptly triggered. Importantly, each experimental setup included multiple repetitions and cycles of detection and response to capture the consistency, stability, and repeatability of system performance over time.

The primary performance indicator used to assess the system's effectiveness was its response time, defined as the elapsed duration between the moment a sensor threshold was breached and the point at which the relevant actuator achieved full operational activation. This metric was carefully recorded and analysed for each detection-response event across all trials and repeated cycles. The collected data allowed for the calculation of the average response time, as well as the standard deviation associated with each test condition, thereby offering a robust quantitative evaluation of both the system's speed and its consistency in reacting to hazardous environmental changes. These measurements are summarized in Table 1, which presents a comparative view of response times across differing environmental profiles. The dataset serves as a critical benchmark for validating the system's reliability, stability, and readiness for deployment in variable real-life contexts where timely fire detection and rapid actuation are essential for minimizing damage and ensuring safety.

Table 1: System response time across ten trials.

Trial number	Average response time (s) – Per trial	Std. deviation (s)
1	2.3	0.2
2	2.6	0.3
3	2.2	0.2
4	2.5	0.4
5	2.4	0.3
6	2.7	0.5
7	2.3	0.2
8	2.5	0.3
9	2.4	0.2
10	2.6	0.4

5. Discussion

The adaptive fire detection and response system developed in this research presents a highly effective synergy between cost-efficient sensor hardware and an advanced software-driven control interface, designed and implemented using LabVIEW. This integrated system harnesses the capabilities of several key environmental sensors, including the DHT11 sensor to measure both temperature and humidity, the MQ-2 sensor to detect the presence of smoke and flammable gases, a Light Dependent Resistor (LDR) for monitoring light intensity levels, and an ultrasonic sensor that determines the proximity of objects [1]. The strategic deployment of these diverse sensors allows the system to collect extensive and multifaceted environmental data. This comprehensive, multi-sensor approach improves the accuracy of fire detection while significantly lowering the risk of generating false alarms, which are more likely to occur in systems that rely solely on a single detection parameter [4]. The integration with LabVIEW greatly enhances the system's performance, as it enables the seamless transformation of raw sensor data into meaningful, actionable insights in real time. LabVIEW's graphical programming interface allows for intuitive implementation of complex logic, facilitating real-time monitoring, control, and adaptive decision-making. Through this dynamic interface, the system remains highly responsive to environmental changes and maintains effective operation across varying conditions. It also supports efficient visualization and synchronization of sensor data, enabling quick interpretation and timely responses. Furthermore, LabVIEW's real-time processing capabilities ensure minimal latency between detection and actuation, which is critical for effective fire response.

To evaluate the performance of the system under controlled but varied conditions, a series of ten independent fire simulation experiments were conducted. Each experiment included ten repeated detection and response cycles, allowing for detailed performance assessment and improved reliability of results. The findings, shown in Table 1, revealed that on average, the system achieved a response time of approximately 2.4 seconds from the moment any sensor threshold was exceeded to the point at which the system's actuator became fully operational. The observed standard deviation in response time across all experiments ranged between 0.2 and 0.5 seconds, indicating only minimal fluctuations. These slight variations were attributed to changing ambient parameters such as temperature, humidity, light intensity, and gas levels. This consistent performance strongly indicates that the system is both rapid and dependable, meeting the fundamental requirements for early detection and swift action expected in industrial fire safety systems. The narrow range of response time variability further highlights the robustness of the system's adaptive thresholding mechanism and decision-making process, which function effectively even when surrounding environmental conditions fluctuate.

LabVIEW serves not only as a visualisation and monitoring platform but also as the core control hub for the entire system. Utilising the VISA protocol [10], LabVIEW establishes a reliable, low-latency communication link with the Arduino microcontroller. Through a custom-developed GUI, users can observe real-time sensor values, view system status through visual indicators, and initiate automatic control operations such as exhaust fan activation. These operations are triggered based on dynamically adjustable threshold values. The seamless communication and interaction between the software and hardware components create a highly user-friendly and efficient interface, enabling real-time control with minimal latency and maximum reliability.

One of the most significant advantages offered by the LabVIEW platform is its capacity to support dynamic threshold adjustment for each sensor, directly through the GUI. This flexibility allows users to tailor the system's sensitivity to match the specific characteristics of various environments. For example, in areas such as industrial kitchens or production floors where moderate heat or smoke may be a normal part of daily operations, the threshold values can be set higher to prevent nuisance alarms. In contrast, in more sensitive zones such as chemical storage units or server rooms, thresholds can be lowered to ensure quicker responses to potential fire risks. LabVIEW also enables manual override controls and supports calibration modes, facilitating easier testing, maintenance, and fine-tuning of system behaviour to suit user requirements.

The underlying software architecture, built on LabVIEW's modular VI framework, provides significant flexibility for future expansion and functional upgrades. The system can be scaled to include additional features such as time-stamped data logging for historical analysis, remote monitoring through LabVIEW's IoT toolkits like ThingSpeak or NI SystemLink, and integration of wireless communication modules such as ESP32 for enhanced connectivity. Moreover, LabVIEW's compatibility with Artificial Intelligence (AI) and machine learning toolkits opens up possibilities for incorporating real-time anomaly detection algorithms and predictive analytics. These enhancements would transform the current setup from a reactive standalone system into an intelligent, connected, and proactive fire safety solution.

Despite the many strengths of the existing system, some limitations must be acknowledged. Currently, it functions as a single-node solution without the ability to store data persistently or access it remotely. In addition, the system relies on continuous connectivity to a Personal Computer (PC), which restricts its deployment in mobile, remote, or distributed environments [5]. To address these limitations, future development should explore transitioning the user interface to LabVIEW NXG (Next Generation) Web Modules or LabVIEW Real-Time, which would allow access through web browsers and improve platform independence. Ultimately, LabVIEW acts as much more than a simple monitoring tool; it is the foundation upon which the system's adaptability, responsiveness, and configuration flexibility are built. As such, the developed system is exceptionally well-suited to meet the evolving demands of modern intelligent fire detection applications, whether in industrial or residential settings.

6. Conclusion

This paper has introduced a highly adaptable fire detection and response system that leverages a combination of multiple environmental sensors and a LabVIEW-based control interface with real-time configurability. Unlike traditional fire detection systems that operate on fixed, hard-coded thresholds which often result in false alarms or delayed responses, this system allows users to dynamically adjust sensor thresholds through an intuitive and interactive GUI, ensuring adaptability to varying operational environments. The inclusion of diverse sensors, namely for temperature, humidity, gas, light intensity, and proximity, enables the system to capture a comprehensive and multi-dimensional view of the monitored environment. This multi-sensor approach, when coupled with integrated actuator responses such as automated fan control and optional live camera feeds, provides an intelligent, context-aware detection mechanism capable of operating reliably across a wide range of indoor scenarios. Experimental trials, comprising ten simulated fire events with ten detection-response cycles each, validated the system's performance. Results showed a consistent average response time of 2.4 seconds from the

point of threshold violation to the activation of the actuating mechanisms. A standard deviation of 0.2 to 0.5 seconds across variable ambient conditions confirmed the system's operational stability and responsiveness, even in fluctuating environmental states. The hardware and software components were designed with modularity and maintainability in mind, featuring a custom-designed PCB and a compact, durable 3D-printed enclosure, which enhance ease of deployment, reconfiguration, and maintenance. Additionally, the system's architecture supports both automated actuation and manual emergency overrides, offering users greater control and safety assurance during critical events. LabVIEW's scalable programming environment further allows for future upgrades, such as AI-based anomaly detection, remote monitoring via IoT integration, wireless communication modules, and browser-based access through LabVIEW NXG. This forward-compatible design sets the stage for transforming the system into a proactive, intelligent fire safety solution. Overall, the proposed system marks a significant advancement toward flexible, robust, and user-friendly fire detection architectures suitable for implementation in smart buildings, industrial zones, high-risk facilities, and any environment where early, reliable fire detection is paramount.

References

- [1] Department of Electrical Engineering, Faculty of Engineering, Technology, Applied Design and Fine Art, Kabale University, Kabale, Uganda and N. Alice, "Industrial Monitoring System with Real-time Alerts and Automated Protection Mechanisms," *IJEM*, vol. 15, no. 2, pp. 56–67, Apr. 2025, doi: 10.5815/ijem.2025.02.05.
- [2] National fire alarm and signaling code, 2019, 2019 edition. Quincy, MA: National Fire Protection Association, 2018.
- [3] R. C. Luo, K. L. Su, and K. H. Tsai, "Fire detection and isolation for intelligent building system using adaptive sensory fusion method," in *Proceedings 2002 IEEE International Conference on Robotics and Automation* (Cat. No.02CH37292), Washington, DC, USA: IEEE, 2002, pp. 1777–1781. doi: 10.1109/ROBOT.2002.1014799.
- [4] F. Khan, Z. Xu, J. Sun, F. M. Khan, A. Ahmed, and Y. Zhao, "Recent Advances in Sensors for Fire Detection," *Sensors*, vol. 22, no. 9, p. 3310, Apr. 2022, doi: 10.3390/s22093310.
- [5] F. Idris, N. Hashim, A. F. Kadmin, and L. B. Yee, "Intelligent fire detection and alert system using labVIEW," *IJECE*, vol. 9, no. 3, p. 1842, June 2019, doi: 10.11591/ijece.v9i3.pp1842-1849.
- [6] B. Srujana, K. Chenchireddy, M. Sujatha, W. Sultana, S. Ahmed Sydu, and E. Giddalur, "Battery temperature monitoring system using Arduino," *IJAAS*, vol. 14, no. 1, p. 53, Mar. 2025, doi: 10.11591/ijaas.v14.i1.pp53-59.
- [7] R. Ubale, Y. Khandge, and N. Kulkarni, "ENERGY-EFFICIENT LIGHTING CONTROL SYSTEM BASED ON MQTT AND RASPBERRY PI," *ShodhKosh J. Vis. Per. Arts*, vol. 5, no. 7, July 2024, doi: 10.29121/shodhkosh.v5.i7.2024.4724.

- [8] V Anupriya, Dr. N. Sathish Kumar, Dr. M Kasiselvanathan, S Sudharsan, Kb Sujeeth, and M Vigneshwar, "Intelligent Factory Safety and Emergency Response System Using LabVIEW," *Int Res J Adv Engg Hub*, vol. 3, no. 05, pp. 2032–2039, May 2025, doi: 10.47392/IRJAEH.2025.0296.
- [9] "An IoT-Based Fire Safety System with Automated Alerts and Live Video Streaming Using ESP32 in Bangladesh," ResearchGate. Accessed: Aug. 09, 2025. [Online]. Available: https://www.researchgate.net/publication/388491924_An_IoT-Based_Fire_Safety_System_with_Automated_Alerts_and_Live_Video_Streaming_Using_ESP32_in_Bangladesh
- [10] C. Barnes, L. Henke, L. Henke, I. Krukov, and O. Hildreth, "SwiftVISA: Controlling Instrumentation with aSwift-based Implementation of the VISA Communication Protocol," *JOSS*, vol. 8, no. 83, p. 4752, Mar. 2023, doi: 10.21105/joss.04752.
- [11] J. Kalunga, S. Tembo, and J. Phiri, "Incorporating Environmental Protection Requirement in Industrial IoT Access Control Security Using Arduino Technology MQ2 and DHT11 Sensor Networks," *IJASRE*, vol. 08, no. 04, pp. 97–122, 2022, doi: 10.31695/IJASRE.2022.8.4.9.