

IoT-Based Early Fire Detection System using Flame and Gas Sensors with Telegram Bot and Blynk Dashboard Integration

Novi Ramadani^{1*}, A M H Pardede², Hermansyah Sembiring³

^{1,2,3}STMIK Kaputama, Binjai

noviramadani2004@gmail.com^{1*}, akimhnp@live.com², hermansyahsembiring240165@gmail.com³

Abstract

Fire incidents continue to pose significant risks to human safety and property due to delayed detection and response. Conventional fire alarm systems generally provide only local alerts, limiting remote monitoring and timely emergency actions. This study presents an Internet of Things (IoT)-based early fire detection system using an ESP32 microcontroller integrated with a KY-026 flame sensor and an MQ-2 gas sensor to detect fire and flammable gas conditions in real time. Sensor calibration was conducted to determine appropriate threshold values, resulting in a flame sensor threshold below 2000 ADC and a gas sensor threshold above 500 ADC. The proposed system integrates the Blynk platform for remote monitoring and Telegram Bot for automatic emergency notifications. To provide an initial fire suppression response, a relay activates a water pump connected to a sprinkler after continuous flame detection for six seconds, while a buzzer immediately alerts nearby occupants. Experimental results demonstrate that all hardware and software components operated reliably and communicated successfully through the Internet. The developed system provides accurate early fire detection, real-time monitoring, rapid warning notifications, and automatic initial fire suppression, making it suitable for residential and small-scale indoor fire safety applications.

Keywords: Blynk; ESP32; Fire detection; Internet of Things; MQ-2 sensor; Telegram Bot

1. Introduction

Fire is one of the most common disasters that can cause severe property damage, environmental losses, and casualties. In many cases, fires are detected only after flames have spread, delaying emergency response and increasing the risk of significant damage. Therefore, an early fire detection system capable of identifying hazardous conditions and delivering immediate warnings is essential to improve safety and minimize fire-related losses [1].

The rapid advancement of Internet of Things (IoT) technology has enabled conventional monitoring systems to evolve into intelligent systems capable of collecting, processing, and transmitting environmental data through the Internet in real time. By integrating sensors, microcontrollers, and wireless communication, IoT allows users to monitor environmental conditions remotely using smartphones or other connected devices. This capability has led to the widespread adoption of IoT in smart homes, industrial automation, environmental monitoring, and disaster mitigation [2].

Several previous studies have developed IoT-based fire detection systems using flame sensors, gas sensors, ESP32, or NodeMCU as the main controller. These systems generally provide early warning notifications through mobile applications or web-based monitoring platforms. However, most existing studies focus primarily on monitoring and notification without implementing an automatic initial fire suppression mechanism. Furthermore, some systems rely on a single sensing parameter, which may reduce detection reliability under different environmental conditions [3], [4], [5], [6].

To address these limitations, this study develops an IoT-based early fire detection system using an ESP32 microcontroller integrated with a KY-026 flame sensor and an MQ-2 gas sensor. The flame sensor is employed to detect infrared radiation emitted by flames, while the MQ-2 sensor continuously monitors smoke and flammable gas concentrations. Before system implementation, both sensors were calibrated to determine appropriate threshold values, thereby improving detection reliability and reducing the possibility of false alarms. Real-time monitoring is provided through the Blynk platform, whereas Telegram Bot delivers automatic warning notifications whenever hazardous conditions are detected [2], [4], [5].

In addition to real-time monitoring and notification, the proposed system incorporates an automatic initial fire suppression mechanism. When a flame is detected continuously for six seconds, the ESP32 activates a relay to operate a water pump connected to a sprinkler, while a buzzer provides an immediate local alarm. This integrated approach combines early detection, remote monitoring, automatic notification, and initial fire suppression within a single IoT-based system. Experimental evaluation demonstrates that the proposed system operates reliably and can provide an effective solution for enhancing fire safety in residential and small-scale indoor environments.

2. Research Methods

This section describes the research methodology used to design, develop, and evaluate the proposed Internet of Things (IoT)-based early fire detection system. It covers the overall research procedure, system architecture, hardware and software design, sensor calibration, and threshold determination. These stages were carried out to ensure that the developed system operated reliably in detecting fire hazards, providing real-time monitoring, and performing automatic initial fire suppression.

2.1. Research Methodology

This study employed the Research and Development (R&D) approach to design, develop, implement, and evaluate an Internet of Things (IoT)-based early fire detection system. The research process consisted of several sequential stages, including literature review, system requirement analysis, hardware and software design, sensor calibration, system implementation, and functional testing. The development process aimed to produce a reliable prototype capable of detecting fire hazards, providing real-time monitoring, and initiating automatic preliminary fire suppression.

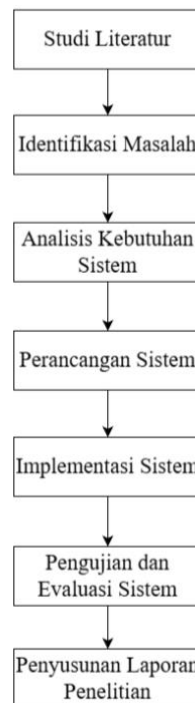


Fig. 1: Research methodology structure

2.2 System Architecture

The proposed system consists of four main components: sensing devices, data processing, communication services, and output devices. The KY-026 flame sensor and MQ-2 gas sensor continuously monitor environmental conditions and transmit analog data to the ESP32 microcontroller. The ESP32 processes the incoming sensor data by comparing each reading with predefined threshold values obtained during the calibration stage.

Once the processed values indicate hazardous conditions, the ESP32 performs several actions simultaneously. Sensor data are transmitted to the Blynk cloud server through a Wi-Fi connection, enabling users to monitor system conditions remotely using the Blynk mobile application. At the same time, warning messages are automatically delivered to users through Telegram Bot. In addition, the buzzer is activated immediately as a local alarm. If flame detection persists continuously for six seconds, the ESP32 energizes a relay module to activate the water pump connected to the sprinkler system, providing an automatic initial fire suppression response.

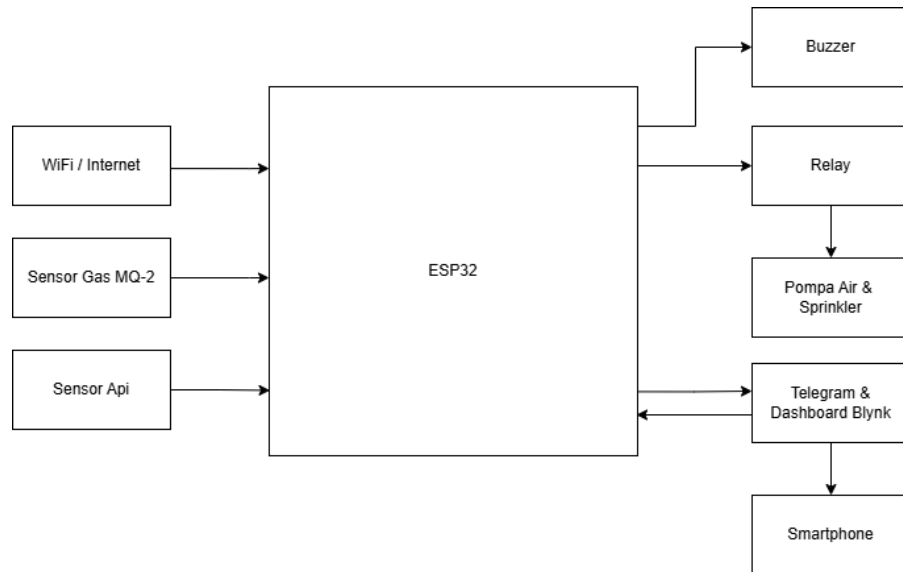


Fig. 2: Block diagram of the proposed monitoring system

2.3 Hardware Design

The hardware configuration was designed to integrate all sensing, processing, communication, and output components into a single embedded system. The ESP32 microcontroller functions as the primary processing unit because it provides integrated Wi-Fi connectivity suitable for IoT applications. The KY-026 flame sensor continuously detects flame radiation, while the MQ-2 gas sensor measures smoke and flammable gas concentrations within the monitored area.

The warning subsystem consists of an active buzzer for immediate local alerts and a relay module controlling a water pump connected to a sprinkler. During normal operation, the relay remains inactive. Once continuous flame detection exceeds the predefined delay time, the relay automatically switches on the water pump to perform preliminary fire suppression. This configuration enables the system not only to detect fire hazards but also to initiate an automatic emergency response.

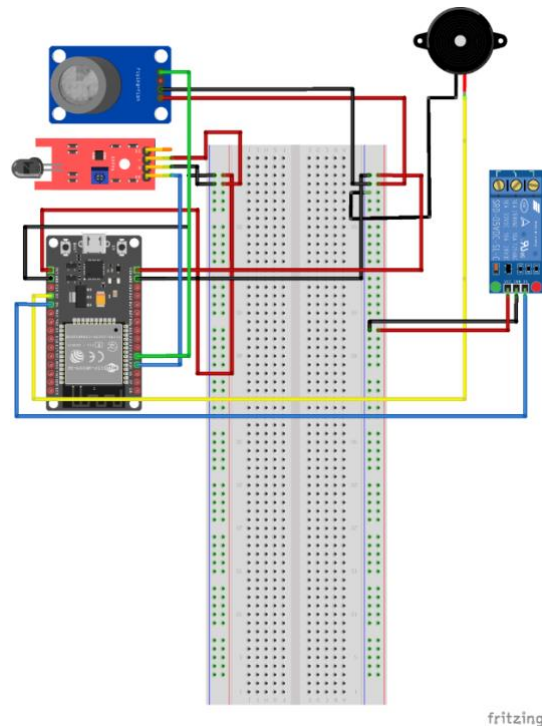


Fig. 3: Hardware design of the proposed monitoring system

2.4 Software Design

The software was developed using the Arduino IDE programming environment. The program continuously reads analog values from both sensors and compares them with predetermined threshold values. Sensor readings are periodically transmitted to the Blynk platform for real-time monitoring. Simultaneously, the system evaluates sensor conditions to determine whether warning notifications or emergency responses should be executed.

When the MQ-2 sensor detects gas concentrations above the threshold, the buzzer is activated and a warning notification is transmitted via Telegram Bot. When the KY-026 flame sensor detects continuous flame exposure, the buzzer immediately generates a local alarm. If the flame persists for more than six seconds, the ESP32 activates the relay to operate the water pump and sprinkler. This sequential control strategy minimizes unnecessary sprinkler activation while maintaining rapid emergency response.

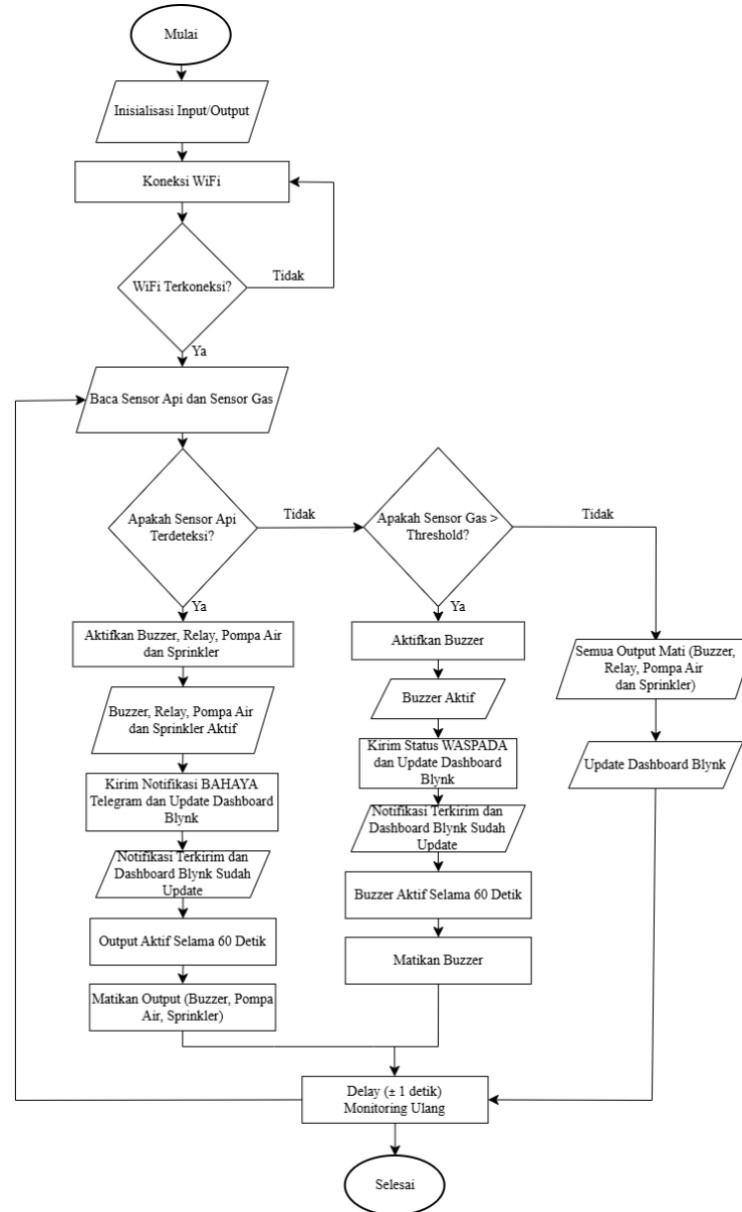


Fig. 4: System flowchart

2.5 Sensor Calibration and Threshold Determination

Sensor calibration was performed before system implementation to evaluate sensor characteristics under normal and hazardous conditions. The KY-026 flame sensor was calibrated by exposing the sensor to a candle flame at various distances, while the MQ-2 gas sensor was calibrated using butane gas generated from a gas lighter. Multiple observations were conducted to obtain stable Analog-to-Digital Converter (ADC) readings for each sensor.

Based on the calibration results, the flame sensor produced significantly lower ADC values when exposed to flames compared with normal environmental conditions. Conversely, the MQ-2 sensor generated higher ADC values as gas concentration increased. Therefore, threshold values of **ADC < 2000** for flame detection and **ADC > 500** for gas detection were selected as the decision parameters used by the ESP32 to classify environmental conditions.

The selected threshold values were subsequently implemented in the control algorithm and validated through functional testing to ensure reliable detection performance while reducing the possibility of false alarms.

Table 1: Flame sensor calibration results

Condition	ADC Value	Description
No flame	2500–3400	Normal condition
Average (No flame)	2750–2850	Characteristic value
Flame detected	0–1300	Flame detected

Table 2: Effect of flame distance on ADC value

Distance	ADC Value	Description
> 100 cm	3000–3400	No flame detected
50–100 cm	2000–3000	Distant flame
20–50 cm	1000–2000	Moderate flame
5–20 cm	500–1000	Close flame
< 5 cm	0–500	Very close flame

Table 3: MQ-2 Sensor Calibration Results

Condition	ADC Value	Description
No gas	270–364	Normal condition
Average (No gas)	304–320	Characteristic value
Low gas concentration	400–600	Gas detected
High gas concentration	600–4095	High gas concentration

Table 4: Threshold Values Used In The Proposed System

Sensor	Threshold (ADC)	Detection Condition
KY-026 Flame Sensor	2000	ADC < 2000
MQ-2 Gas Sensor	500	ADC > 500

3. Results and Discussion

This section presents the implementation and evaluation results of the proposed IoT-based early fire detection system. The discussion includes the hardware implementation, software implementation, and system testing conducted to evaluate the overall performance of the developed prototype. The evaluation focuses on the functionality of each system component, communication reliability, and the effectiveness of the proposed fire detection and initial suppression mechanism.

3.1. Hardware Implementation

The hardware implementation was carried out by integrating all electronic components into a single prototype based on the ESP32 microcontroller. The prototype consists of a KY-026 flame sensor, an MQ-2 gas sensor, an ESP32 development board, a relay module, an active buzzer, and a water pump connected to a sprinkler system. The ESP32 functions as the main controller that receives sensor readings, processes the acquired data, and controls all output devices according to the predefined operating logic.

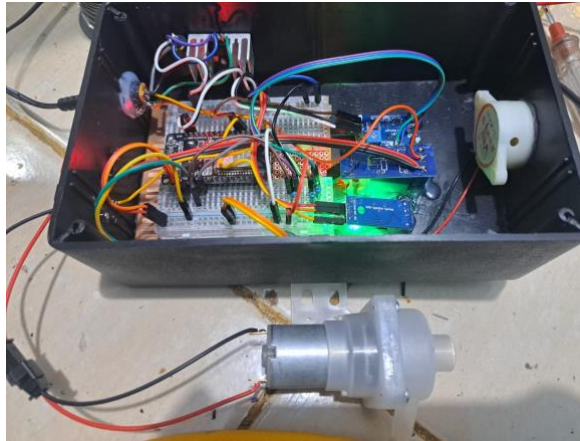


Fig. 5: Hardware Implementation

Figure 5 illustrates the completed hardware prototype developed in this research. The flame sensor continuously detects infrared radiation emitted by flames, whereas the MQ-2 sensor monitors smoke and flammable gas concentrations. Both sensors transmit analog signals to the ESP32 for processing. Under normal conditions, the system continuously monitors environmental conditions without activating any warning devices. When hazardous conditions are detected, the ESP32 immediately initiates appropriate warning and suppression actions according to the programmed control algorithm.

The relay module controls the water pump used for the sprinkler system. Unlike conventional fire alarm systems that only provide warning notifications, the developed prototype performs an automatic initial suppression response after continuous flame detection for six seconds. This mechanism was designed to reduce the possibility of unnecessary sprinkler activation while maintaining rapid emergency response.

3.2. Software Implementation

The software implementation was developed using the Arduino IDE programming environment. The ESP32 continuously acquires analog data from both sensors and compares the measured values with the predefined threshold values obtained during the calibration stage. The software also manages Wi-Fi communication, cloud monitoring through the Blynk platform, Telegram Bot notifications, and relay control for automatic sprinkler activation.

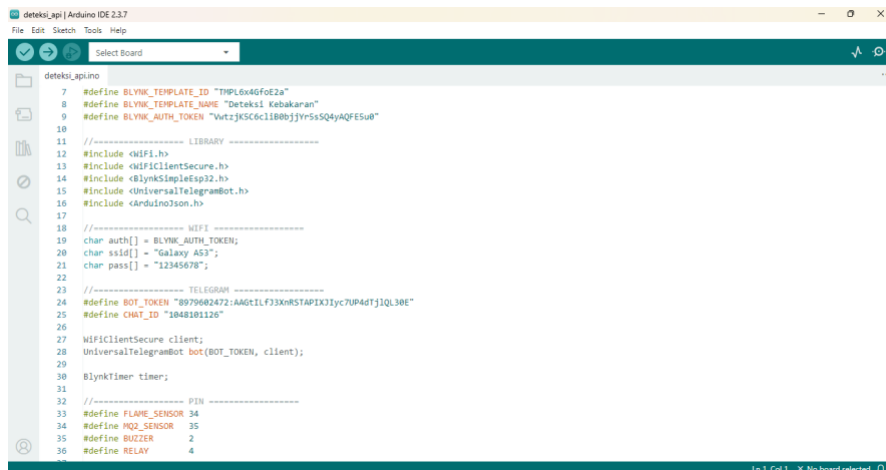


Fig. 6: Software implementation using Arduino IDE

3.2.1 Blynk Dashboard

The Blynk platform provides real-time visualization of sensor measurements obtained from the ESP32. Sensor readings are continuously transmitted through the Internet and displayed on the mobile dashboard, enabling users to monitor environmental conditions remotely.

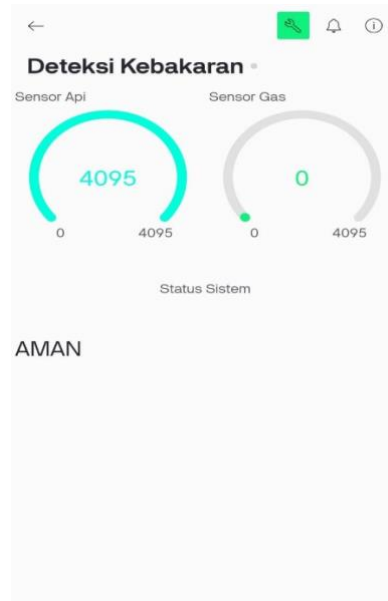


Fig. 7: Blynk Dashboard Interface.

The dashboard displays the current flame sensor value, MQ-2 gas sensor value, and overall system status. This interface allows users to observe sensor behavior continuously and verify whether the monitored environment remains under safe operating conditions.

3.2.2 Telegram Bot Notification

Telegram Bot was implemented as the remote notification service for emergency situations. Whenever hazardous conditions are identified, the ESP32 automatically sends warning messages through the Telegram application.

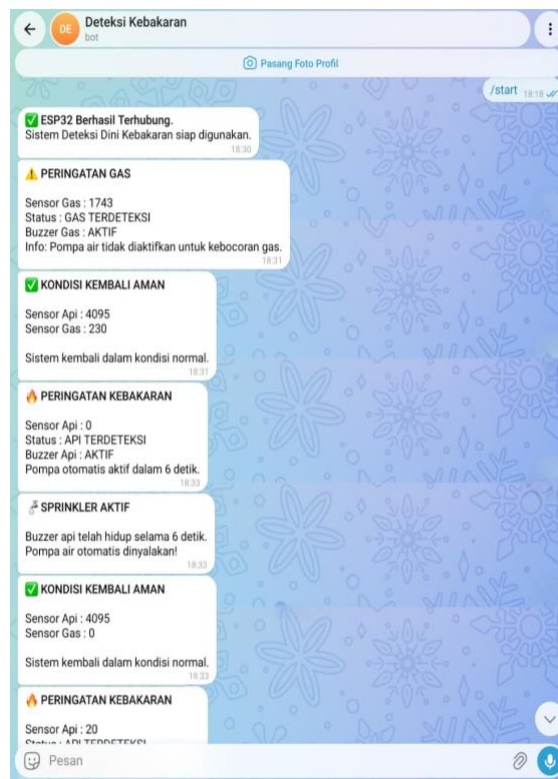


Fig. 8: Telegram Bot Notification Interface.

The notification system provides immediate information regarding detected fire or gas hazards. Consequently, users receive warning messages even when they are not physically present at the monitored location, thereby improving response time during emergency situations.

3.3. System Testing

The developed prototype was evaluated through a series of functional tests to verify the performance of each hardware and software component. The evaluation included flame sensor testing, MQ-2 gas sensor testing, relay activation testing, Blynk dashboard testing, Telegram Bot notification testing, and overall system testing. Each experiment was performed under predefined conditions to ensure that the proposed system operated according to the intended design specifications.

3.3.1 Flame Sensor Testing

The flame sensor was tested using a candle flame placed at various distances from the sensor. During the experiment, the ESP32 continuously recorded the Analog-to-Digital Converter (ADC) values generated by the KY-026 sensor and compared them with the predefined threshold value.

Table 5: Flame Sensor Testing Results

No.	Test Condition	Flame Sensor Value (ADC)	System Status	Buzzer	Relay	Water Pump	Result
1	No flame detected	4095	SAFE	OFF	OFF	OFF	Successful
2	Flame detected	1729	FIRE DETECTED	ON	OFF	OFF	Successful
3	Flame detected (> 6 s)	0	FIRE DETECTED	ON	ON	ON	Successful

The experimental results indicate that the flame sensor successfully detected flame presence whenever the measured ADC value decreased below the threshold value of 2000. Under normal environmental conditions, the recorded ADC values remained above the threshold, indicating stable sensor performance without false flame detection. These results demonstrate that the selected threshold effectively distinguishes normal conditions from hazardous conditions and enables reliable flame detection.

3.3.2 MQ-2 Gas Sensor Testing

The MQ-2 gas sensor was evaluated using butane gas generated from a gas lighter. Gas exposure was gradually increased while the corresponding ADC values were continuously monitored by the ESP32.

Table 6: MQ-2 Gas Sensor Testing Results

No.	Test Condition	Gas Sensor Value (ADC)	System Status	Buzzer	Relay	Water Pump	Result
1	No gas detected	0	SAFE	OFF	OFF	OFF	Successful
2	Gas detected	2899	GAS DETECTED	ON	OFF	OFF	Successful

The test results show that the MQ-2 sensor consistently detected increased gas concentrations by producing ADC values above the predefined threshold of 500. During normal operation without gas exposure, sensor readings remained below the threshold value. This behavior confirms that the selected threshold provides reliable discrimination between safe and hazardous environmental conditions while maintaining stable monitoring performance.

3.3.3 Relay and Sprinkler Testing

The relay module was evaluated to verify its capability to control the water pump connected to the sprinkler system. The relay was programmed to activate only after continuous flame detection for six seconds.

Table 7: Relay and Sprinkler Testing Results

No.	Test Condition	Relay	Water Pump	Result
1	Normal condition	OFF	OFF	Successful
2	Flame detected (< 6 s)	OFF	OFF	Successful
3	Flame detected (> 6 s)	ON	ON	Successful
4	Gas detected	OFF	OFF	Successful

Experimental observations demonstrate that the relay remained inactive during normal conditions and was successfully activated after continuous flame detection exceeded the programmed delay period. This delay mechanism effectively prevents unnecessary sprinkler activation caused by temporary flame exposure while maintaining an appropriate emergency response time for actual fire events.

3.3.4 Blynk Dashboard Testing

The Blynk dashboard was tested to evaluate real-time communication between the ESP32 and the cloud monitoring platform.

Table 8: Blynk Dashboard Testing Results

No.	Test Condition	Flame Sensor (ADC)	Gas Sensor (ADC)	System Status	Result
1	Normal condition	4095	0	SAFE	Successful
2	Flame detected	0	0	FIRE DETECTED	Successful
3	Gas detected	4095	2899	GAS DETECTED	Successful

The testing results indicate that sensor measurements were transmitted successfully to the Blynk platform through the Wi-Fi network. The dashboard continuously displayed updated flame sensor values, MQ-2 sensor readings, and system status with stable communication performance. Consequently, users were able to monitor environmental conditions remotely using a smartphone.

3.3.5 Telegram Bot Notification Testing

Telegram Bot functionality was evaluated by generating several fire and gas detection scenarios and observing notification delivery performance.

Table 9: Telegram Bot Notification Testing Results

No	Test Condition	Telegram Notification	Result
1	ESP32 connected	ESP32 Connected Successfully	Successful
2	Flame detected	Fire Warning	Successful
3	Gas detected	Gas Leakage Warning	Successful
4	Flame detected (> 6 s)	Sprinkler Activated	Successful
5	System returned to normal	System Returned to Normal	Successful

The notification system successfully transmitted warning messages immediately after hazardous conditions were detected. Users received notification messages containing information about flame detection or gas leakage in real time. These results confirm the effectiveness of Telegram Bot as a reliable remote warning mechanism for the proposed system.

3.3.6 Overall System Testing

Comprehensive system testing was performed to evaluate the interaction among all hardware and software components under different operating conditions. Each scenario involved simultaneous observation of sensor detection, alarm activation, cloud monitoring, Telegram notification, and automatic sprinkler control.

Table 10: Overall System Testing Results

No	Test Scenario	System Response	Status
1	Normal condition	System operates under safe conditions	Successful
2	Flame sensor detects fire	Buzzer activated, Telegram notification sent, Blynk status updated	Successful
3	MQ-2 sensor detects gas	Buzzer activated, Telegram notification sent, Blynk status updated	Successful
4	Flame detected for more than 6 seconds	Relay activated, water pump turned ON, sprinkler notification sent	Successful
5	System returned to normal	Buzzer and relay turned OFF, Blynk status updated to SAFE, recovery notification sent	Successful



Fig. 9 Implementation of The Iot-Based Monitoring System

Overall experimental results demonstrate that all system components operated according to the proposed design. The flame sensor and MQ-2 gas sensor successfully detected hazardous conditions, while the ESP32 accurately processed sensor data and executed the corresponding control logic. The buzzer provided immediate local warnings, Blynk enabled continuous remote monitoring, Telegram Bot delivered automatic notifications, and the relay-controlled sprinkler performed the intended initial fire suppression after the specified delay period. These findings indicate that the developed IoT-based early fire detection system achieved reliable real-time monitoring, rapid emergency notification, and automatic initial fire suppression for residential and small-scale indoor environments.

4. Conclusion

This study successfully designed, implemented, and evaluated an Internet of Things (IoT)-based early fire detection system using an ESP32 microcontroller integrated with a KY-026 flame sensor and an MQ-2 gas sensor. The developed system was capable of detecting flame and gas conditions in real time based on predefined threshold values obtained through sensor calibration. Experimental results demonstrated that the flame sensor reliably detected fire when the ADC value decreased below 2000, while the MQ-2 gas sensor successfully identified gas leakage when the ADC value exceeded 500. These threshold values enabled accurate environmental condition classification and improved system reliability.

Furthermore, the integration of the Blynk platform and Telegram Bot enabled continuous remote monitoring and automatic warning notifications, allowing users to receive real-time information regarding fire hazards. The relay-controlled water pump and sprinkler also functioned as intended by providing automatic initial fire suppression after continuous flame detection for six seconds. Overall, the proposed system successfully combined early fire detection, real-time monitoring, remote notification, and automatic emergency response into a single IoT-based solution suitable for residential and small-scale indoor environments.

References

- [1] M. Wahidin, A. Elanda, and S. S. Lie, "Implementasi Sistem Pendeteksi Kebakaran Berbasis IoT dan Telegram Menggunakan NodeMCU pada Kantor Notaris Leodi Chanda Hidayat, S.H., M.Kn.," *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 16, no. 2, pp. 1–8, 2021, doi:10.35969/interkom.v16i2.104.
- [2] H. Haryanto, P. Paryanta, A. Aji, and M. Sulistyani, "Sistem Monitoring Kebakaran Rumah Berbasis Internet of Things," *Go Infotech: Jurnal Ilmiah STMIK AUB*, vol. 29, no. 2, pp. 147–158, 2023, doi:10.36309/goi.v29i2.212.
- [3] W. Kuncoro, J. Maulindar, and R. P. Indah, "Monitoring Peringatan Dini Kebakaran pada Sistem Smart Home Menggunakan NodeMCU Berbasis IoT," *Generation Journal*, vol. 7, no. 2, pp. 105–115, 2023, doi:10.29407/gj.v7i2.20015.
- [4] A. U. Khasanah, N. Azizah, N. Ilmah, T. B. Bahtiar, R. Riswira, and A. A. Putri, "Perancangan Sistem Deteksi Kebakaran Menggunakan ESP32 dengan Notifikasi Blynk," *Joule (Journal of Electrical Engineering)*, vol. 5, no. 2, pp. 46–53, 2024.
- [5] J. T. Simanungkalit and B. M. Rambe, "Penerapan IoT dalam Pendeteksi Gas (CO) dan Kebakaran dengan Notifikasi Aplikasi Telegram," *Jurnal Minfo Polgan*, vol. 14, no. 1, pp. 691–701, 2025, doi:10.33395/jmp.v14i1.14843.
- [6] C. Tri Ardiyanto, C. Supriadi, and Priyadi, "Rancang Bangun Sistem Monitoring Deteksi Dini Kebakaran Ruang Server Berbasis IoT," *Jurnal Ilmiah Sistem Informasi*, vol. 3, no. 3, pp. 35–45, 2025, doi:10.51903/432zjf74.