International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Real-Time Industrial Hazard Detection System Using Arduino and IoT

[1] Mayur Manikant Patti, [2] Somesh Singh Jagat, [3] Yash Singh, [4] Manoj Kumar Singh

[1 2 3 4] CSE (IOT), SSTC, INDIA.

[1] pattimayur@gmail.com, 2 Someshsingh3738@gmail.com, 3 yashvloggs20@gmail.com, 4 Manoj5682@gmail.com

ABSTRACT –

Factories are at risk of fire outbreaks, gas leaks and increases in temperature. All of these can adversely affect human beings and infrastructure alike. To overcome the problems faced by the industries, a cost-efficient and efficient industrial safety system based on Arduino Uno is presented in this paper. It consists of an LM35 temperature sensor, MQ2 gas sensor, flame sensor, and LDR sensor which monitors parameters inside the industrial environment and generates the alert message to the controller. When critical thresholds are exceeded, it triggers automatic responses like starting exhaust fans and lighting up the lights. The ESP8266 Wi-Fi module allows real-time uploading of data on the cloud, which helps in monitoring and alerting. The system seems very promising to be a scalable and easy-to-implement safety-enhancing mechanism in an industrial setup.

Keywords— Firefighting Robot, IoT, Autonomous Navigation, Fire Extinguishing System, Water Pump, Flame Sensor, Motor Controller (L239D).

I - INTRODUCTION

As industries expand, there is an increasing need for reliable and intelligent safety systems. Standard safety measures depend on the manual control, and because of this, in an emergency, there might be a delay in the response which will pose a danger to the staff and equipment. Fires, gas leaks, and extreme heat conditions are some of the most common hazards in such cases. The Internet of Things (IoT) is evolving at a rapid speed, enabling us to design systems that can automate safety features which are efficient and inexpensive. This paper proposes an IoT-based industrial safety system utilizing the Arduino Uno platform. The system can monitor the surrounding operations by itself, make its own decision and trigger an alarm locally and remotely upon detection of a threat. This makes it a suitable candidate for deployment at small to medium industries, due to its simplicity and modularity.

II - LITERATURE REVIEW

- Simplified Firefighting Robot with Basic Sensors

Tawfiqur Rakib and M A. Rashid Sarkar presented the basic robot using a flame sensor and LM35 temperature sensor. These basic models of firefighting robots are cheap and relatively simple to build but lacks IoT integration and monitoring remotely which our project offers as we use the ESP8266 module which makes it connect to Wi-Fi.

- Automatic Residential Fire Suppression System

An innovative smart home fire suppression system is designed by Ryo Takeuchi who designed it to control it through a web interface and fire is detected using OpenCV. Although more useful for home use the use of remote-control in the work inspired the cloud-based monitoring and alerting of our Arduino-IoT project.

- Sharma and Singh worked on the development of a real-time monitoring system to connect the gas and temperature sensor with the wireless module. It was effective in detection at early stage and laid foundation for low-cost industrial safety.

- Kaur and Verma came up with a GSM Modules with Arduino for gas and flame sensors for alerting system. Their system emphasized how crucial it is to provide local and remote alerts immediately.

- Chen experimented on wireless sensor networks (WSNs) for huge-scale industries and noticed the potential of distributed sensing with real-time visualization. The research stressed on scaling up and cloud integration to emergency response.

III-METHODOLOGY

a. Problem Definition

To tackle the industrial hazards, the system is designed based on an Arduino Uno microcontroller with multiple sensors. These sensors include an LM35 (temperature), MQ2 (gas), flame sensor, LDR for environmental monitoring, through which monitoring can be done.The IoT-based Industrial Safety

System using Arduino aims to address these challenges through a sensor-driven, real-time monitoring system that automatically detects industrial hazards and triggers preventive actions, while also enabling remote access and alerts via IoT technology.Problem planning and Designing the Robot

Design Structure : A relay module is utilized to switch on/off an external device like a fan for fuming gas or high heat and L E D light for LDR.
The ESP8266 Wi-Fi module transmits data in real-time on the cloud for easy access and alerting.

The system follows a four-step design plan

The sensor initialization will start monitoring the parameters continuously.

Comparison of live data with already set values.

Automated Response – Activates fan/light using relays

The Internet of Things sends live data to the cloud and displays it on an LCD.

The modular design of this system makes it easily scalable and affordable for 24/7 industrial safety monitoring.

Fig. 1 Basic prototype of the system IoT-based Industrial Safety System

Hardware Implementation :

The hardware configuration is a critical aspect of the BlazeControl firefighting robot, enabling reliable sensing, navigation, and fire suppression capabilities. The system incorporates an ESP microcontroller as the central control unit, offering enhanced processing power and compatibility with IoT functionalities for future scalability. The robot includes three IR flame sensors placed strategically to detect fire sources in multiple directions (left, center, and right), facilitating accurate fire localization. To support autonomous movement, BO motors coupled with rubber wheels provide mobility, while an L293D motor driver controls and powers the motors effectively. The fire suppression mechanism uses a submersible water pump, and the nozzle direction is controlled using servo motors, allowing targeted extinguishing based on fire position. All components are mounted on a chassis and interconnected via a mini breadboard for compact assembly. Figure 2 illustrates the block diagram of the BlazeControl system, highlighting the core input-output flow: flame detection, processing via ESP controller, motion control, and fire extinguishing.

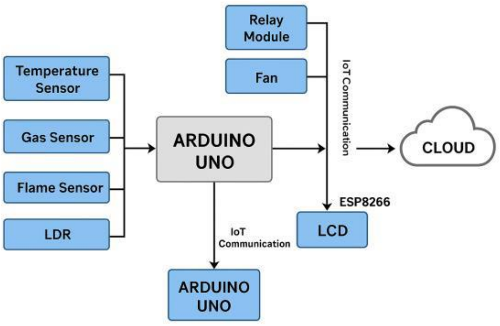

Fig. 2 Block diagram of the IoT-based Industrial Safety System

b. Hardware Used

ESP32 : Fig 3. shows the ESP32 development board, which serves as the central microcontroller for the BlazeControl firefighting robot. The ESP32 is a powerful dual-core microcontroller with integrated Wi-Fi and Bluetooth capabilities, making it ideal for real-time robotics and IoT-based applications. It is capable of handling multiple inputs and outputs simultaneously, allowing efficient processing of sensor data and motor control. The ESP32 board features a rich set of digital and analog GPIO pins that can interface with various components such as flame sensors, motor drivers, and servo motors. Its high processing speed and built-in timers make it suitable for time-critical operations like fire detection and suppression. The board contains all essential features required to support the microcontroller, including voltage regulation, USB connectivity, and flash memory, making it a compact and efficient choice for autonomous fire-fighting systems

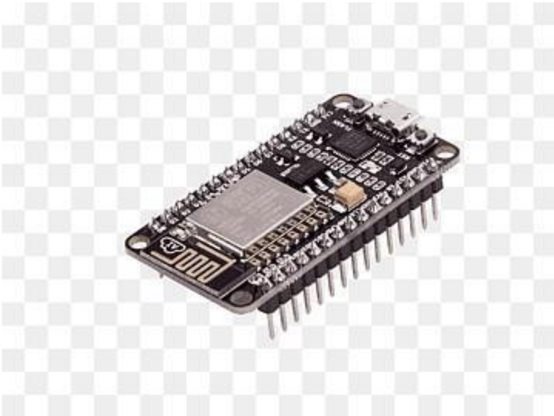

Fig 3. ESP32 development board

ESP8266 Wi-Fi Module : This module allows the system to connect to the internet It sends real-time information from sensors to a cloud platform Blynk or ThingSpeak. With this we can Continuously monitor and get alerts on things remotely so that we can act quickly in case of any harmful situation..

Fig 4. ESP8266 Sensor

Arduino Uno : At the heart of the system is the Arduino Uno microcontroller board. The threshold values are fixed which the sensor needs to read so that output can be controlled as per it**.**

Fig 5. Arduino Uno

LM35 Temperature Sensor : The LM35 is a type of sensor that measures temperature in Celsius. The system communicates with Arduino to detect unusual temperature rise. It enables the fan when the temperature reaches a specified level, and the IoT will send alerts notification

Fig 6. LM35 Temperature Sensor

MQ2 Gas Sensor : MQ2 is used to detect gases like LPG, methane, and smoke It keeps checking the air quality and sends analog values to Arduino. Switch on the fan and raise an alert if the gas concentration is above the safety limits.

Fig 7. :MQ2 Gas Sensor

Flame Sensor : The flame sensor is used to detect the infrared light from fire of the flame. It signals to the Arduino using the digital pin in the presence of flame. This turns on the fan, alerts the user, and sends the information to the cloud in an emergency response..

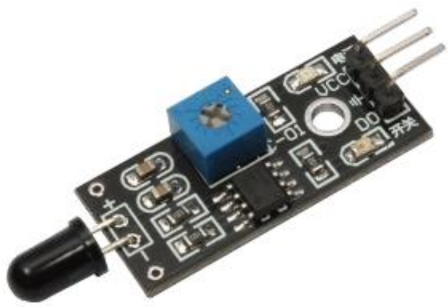

Fig 8. Flame Sensor

16*2 LCD Display : The LCD module is used to display live values like Temperature, Gas , and Flame . It provides quick feedback on the system , helping the workers to get the notification instantly without using the internet.

Fig 9. LCD Display

230V AC Bulb : The relay is attached by the Arduino to control the bulb based on the LDR . It switches on when lighting is low so the area can be seen.This allow workers to react better during any power failures.

Fig 10. Chargeable Battery

Jumper Wires : Fig 11. shows the Jumper Wires. These are used for establishing electrical connections between various components and the ESP32 on the breadboard, enabling signal transmission and power distribution.

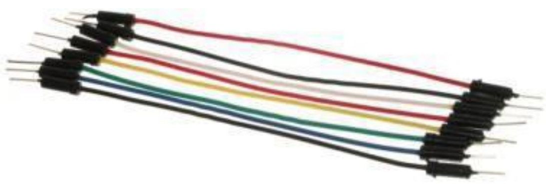

Fig 11. Jumper Wires

DC Fan : The activation of the 12V fan is ordered via the relay through Arduino in case of high temperatures or gas level. It helps workers to face emergency or power failure situations better at the industrial site. Arduino switches on the 12V fan using relay when high temperatures or gas comes.

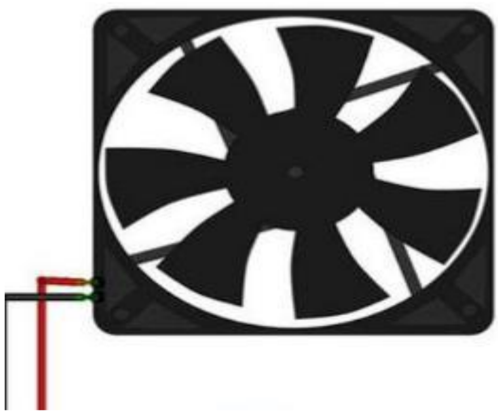

Fig 9. DC Fan

c. Programming

The main program of the IoT-based industrial safety system is programmed on Arduino IDE. The Arduino IDE is easy to use software for writing code that is loaded onto the Arduino Uno microcontroller. C and C++ programming language are supported by it. It is used to handle the sensor data and make the decision in real-time.

The main logic with the help of input values from various sensors such as the MQ-2 gas sensor, LM35 temperature sensor, and flame sensor will process the data and provide alerts through the buzzer, LCD, and ESP8266. The Wi-Fi module is programmed to send updates over cloud platforms like ThingSpeak or mobile apps via Blynk immediately.

The Arduino IDE is free, cross-platform software available for Windows, OSX, and Linux. Its versatility and extensive community support make it very effective for developing embedded safety applications in industrial environments.

d. Working

In this system, sensors detects any hazard in working environment and send the information for corrective action taken. The basic component is the microcontroller that is a UNO Arduino and takes input from different sensors. These do multiple works. The first one is the MQ-2 gas sensor used for gas leakage. The second is the LM35 temperature sensor that detects the temperature in the atmosphere. The last is the flame sensor that can detect if there is a flame or fire in the industrial area.

Each sensor continuously feeds data to the Arduino If any of the readings of the sensor exceed the safety thresholds, then alarm and turn on the fan. Safety Thresholds Monitor gas concentration levels. Monitor temperature levels. To start, a buzzer and LCD get activated to alert employees on-site. At the same time, the data is sent to IoT cloud platforms like ThingSpeak or Blynk by the ESP8266 Wi-Fi module from where it can be remotely accessed via smartphones or web dashboards by safety managers and supervisors.

The system has been designed to function all by itself. Future upgrades can link it to automated shutoff devices or a fire sprinkling system. The sensors and elements are installed in a compact and sturdy casing for the rugged industrial environment. This safety mechanism guarantees hazard detection, alert generation, and data transmission within seconds in a closed loop. It drastically reduces response time while avoiding accidents.

IV-RESULTS AND EVALUATION

System Performance.-It was tested on many simulated industrial situations to see reliability and response. It was able to detect dangerous conditions and take the right corrective actions almost instantaneously.

S.No	Temperature	Gas	Fire	LDR	Date
1	58	OFF	OFF	Light	2025-04-11 17:46:07
2	32	OFF	OFF	Light	2025-04-11 17:44:38
3	31	OFF	OFF	Light	2025-04-11 17:43:08
4	31	OFF	OFF	Light	2025-04-11 17:41:39
5	31	OFF	ON	Light	2025-04-11 17:41:10
6	31	ON	OFF	Light	2025-04-11 17:40:20
7	31	ON	OFF	Light	2025-04-11 17:40:03
8	31	OFF	OFF	Light	2025-04-11 17:39:19
9	31	OFF	OFF	Light	2025-04-11 17:37:50
10	31	OFF	OFF	Light	2025-04-11 14:09:21
11	31	ON	OFF	Light	2025-04-11 14:08:56
12	31	OFF	OFF	Light	2025-04-11 14:07:35
13	31	OFF	OFF	Light	2025-04-11 14:06:06
14	31	OFF	OFF	Light	2025-04-11 14:04:36
15	31	OFF	OFF	Light	2025-04-11 14:03:07
16	31	OFF	ON	Light	2025-04-11 14:01:53
17	31	ON	OFF	Light	2025-04-11 14:01:34
18	32	ON	OFF	Light	2025-04-11 14:01:07
19	32	ON	OFF	Light	2025-04-11 14:00:33
20	32	ON	OFF	Light	2025-04-11 14:00:15

Observations and Outputs.

i. The LCD screen showed correct sensor values on real-time basis.
ii. The IoT dashboard via smartphone has live graph and notifications.
iii. Relays triggered the actuators reliably without delay

V-DISCUSSION

The proposed system effectively shows how industries can use IoT and embedded systems for safety measures. With real-time response, cost-effective and easy-to-install solution for industries, it helps manage hazards without burning a hole in the pocket.

Advantages

i. Cost-effective hardware and easy scalability
ii. On-site as well as cloud dashboard monitoring
iii. Safety controls auto, less need for human response.

Challenges

i. Real-time monitoring needs to have stable Wi-Fi connectivity.
ii. Current implementation doesn't predict nor analyze historical data.

Future Scope

i. Integrating AI and ML for spotting abnormalities.
ii. GSM/LoRa modules to facilitate communication in weak or no-Wi-Fi areas.
iii. Data logging and analytics for long-term system assessment
iv. SMS or email alerts so that staff can be notified immediately without using the app.

VI-CONCLUSION

System provides real-time detection of hazards and responds to them through the use of a sensor network automated and cloud-based monitoring. This system can help to avoid accidents that might be industrial type. Especially relevant in setups where resources are constrained. A full-fledged intelligent safety management system may be developed This paper presents a smart and practical IoT-based industrial safety solution using the Arduino platform. The through future improvements.

REFERENCES :

1. Rakib and Sarkar (2016) developed a low-cost Arduino-based fire-fighting robot. It utilized LM35 and flame sensors for automatic extinguishing through a water pump. [ResearchGate] https://www.researchgate.net/publication/309672911

2. Created an IoT-enabled system to detect faults in industrial areas which make use of gas sensor, fire sensor, GSM, and camera. [Asian Journal of Computer and Technology] https://asianssr.org/index.php/ajct/article/view/1253

3. Kumaratharan and colleagues 2023 suggested a fire accident tracking system using arduino uno esp8266 and cloud monitoring for safety in industry [AmritaVishwaVidyapeetham] https://www.amrita.edu/publication/industries-fire-accident-tracking-system-based-on-internet-of-things-using-arduino-uno/

4. According to Vishwakarma et al., 2023, a fire-fighting robot was created with sensor-based detection and an onboard water spraying system. [IJRASET]https://www.ijraset.com/research-paper/design-and-fabrication-of-automated-fire-fighting-robot

5. Basha et al. (2021) developed an Arduino + IoT based system which uses gas sensor, flame sensor and LDR sensor to avoid fire accidents and send alerts in real-time to Blynk. [MATJournals] https://matjournals.co.in/index.php/JOCEI/article/view/3154