JSRT JOURNAL

PAGES: 357-368
24/07/25
JOURNAL OF SCIENTIFIC RESEARCH AND TECHNOLOGY VOLUME-3 ISSUE-7 JULY

A Smart IOT Based Emergency Detection System

Dr. Sujata. Mallapur[1], Ashwini Biradar[2], Shrirakshka Joshi[3], Vinita Chinmalli[4], Alfarha Sultana[5]

[1]Assistant Professor, Department of AIML, Sharnbasva University, Kalaburagi, India.

[2,3,4,5]Students, Department of AIML, Sharnbasva University, Kalaburagi, India

ABSTRACT

This project aims to create a Smart IoT-Based Emergency Detection System that can identify hazardous environmental conditions in real-time. The system is designed to reduce response time during emergencies and enhance safety in residential, industrial, and public environments. The system uses an ESP32 Dev Kit microcontroller with low power consumption and Wi-Fi capabilities. It integrates various environmental sensors to monitor conditions in real-time. Upon detecting abnormal values, the system activates local alerts via a buzzer and LCD display, and sends remote alerts via a GSM module via SMS. Cloud integration using the Blynk IoT platform allows data visualization and control via mobile applications. The system responds accurately to hazardous conditions and reliably sends alerts within seconds. The project extends to smart homes, factories, schools, hospitals, and warehouses, laying the foundation for future AI-based risk prediction and integration with centralized emergency management systems.

Keywords: IOT, Safety, Emergency

I. INTRODUCTION

1.1 INTRODUCTION

In today’s rapidly advancing technological landscape, the integration of smart systems into everyday life has become essential for improving safety, efficiency, and responsiveness—particularly in emergency situations. One of the most impactful innovations in this area is the application of the Internet of Things (IoT) to emergency detection and management systems. IoT enables a network of interconnected devices and sensors that can continuously monitor the environment, detect potential hazards, and transmit data in real-time. These systems have the potential to revolutionize emergency response strategies by automating the process of identifying, communicating, and responding to threats.

The Smart IoT-Based Emergency Detection System presented in this project is designed to autonomously monitor environmental parameters and detect emergency scenarios such as fire outbreaks, gas leaks, abnormal temperature rises, or sudden movements that may indicate accidents. The system utilizes a combination of advanced sensors including fire sensors, gas sensors, motion detectors, smoke detectors, and temperature sensors. These sensors are strategically interfaced with a microcontroller, specifically the ESP32 DevKit, which serves as the central processing unit of the system.

Once an emergency is detected, the system triggers an on-site buzzer and displays the nature of the alert on an LCD screen to immediately notify anyone in the vicinity. Simultaneously, the system employs GSM (Global System for Mobile Communications) or Wi-Fi modules to send automated alert messages or calls to emergency contacts or services. This dual-alert mechanism ensures both local and remote awareness of any critical situation, significantly improving response times and reducing reliance on manual detection or reporting.

This IoT-based solution is highly scalable and versatile. Its applications are far-reaching and include, but are not limited to:

- Industrial environments, where hazardous gas leaks or fires can occur;
- Smart homes, for enhanced residential safety;
- Smart healthcare, to monitor elderly or disabled individuals;

- Public transport systems, for immediate accident or fault detection;
- Educational and commercial buildings, to ensure the safety of occupants during potential threats.

The real-time data collection feature also enables remote monitoring, data logging, and analytics, which can be used for predictive maintenance, incident investigation, and improving emergency protocols. The system can be programmed to prioritize different types of emergencies, escalate alerts based on severity, and even activate automated safety measures like shutting off power or gas supplies.

The significance of this project lies in its ability to bridge the gap between detection and action, providing a proactive approach to emergency management. It reduces human intervention, minimizes errors, and accelerates response time—key factors in mitigating damage and saving lives during emergencies.

This section lays the groundwork for understanding the purpose, scope, and importance of implementing a smart IoT-based emergency detection system. It introduces the key components, functionalities, and potential impacts of the system. The chapters that follow will cover an in-depth review of existing related systems, detail the methodology adopted for designing and implementing the prototype, evaluate the performance and outcomes, and discuss the advantages, limitations, and future scope of the project. Through this research and development initiative, we aim to contribute to the growing field of intelligent safety systems and explore practical ways to integrate IoT in everyday emergency management solutions.

II. LITERATURE REVIEW

2.1 BACKGROUND INFORMATION

Emergency detection systems have played a pivotal role in enhancing safety and risk management across homes, industries, and public spaces for decades. Traditionally, these systems included basic sensors and alarm mechanisms such as fire alarms, smoke detectors, and gas leak sensors, often installed as standalone units. Their primary function was to trigger local alerts like sirens, bells, or flashing lights whenever a hazard was detected. While effective to a certain degree, these legacy systems were limited by their inability to communicate with remote users or centralized control systems, often relying on human presence to acknowledge and respond to alerts.

As technology evolved, the introduction of embedded systems and microcontrollers such as Arduino, AVR, and Raspberry Pi brought about the next wave of innovation. These programmable hardware platforms enabled the automation of detection and alert mechanisms, allowing systems to take predefined actions without manual input. However, these early embedded solutions still lacked several crucial capabilities: real-time remote monitoring, internet connectivity, data logging, and scalability. Alerts remained localized, and users had no way of knowing about emergencies when they were not physically present.

The true revolution in emergency detection came with the emergence of the Internet of Things (IoT)—a paradigm shift that allowed everyday devices to communicate intelligently over the internet. IoT introduced the concept of networked sensing and automation, where physical devices embedded with sensors, software, and connectivity could gather and share data in real-time. This advancement led to the development of smart emergency detection systems that could offer remote access, real-time alerting, automation, cloud integration, and mobile interaction.

Modern microcontrollers like the ESP8266 and ESP32, with built-in Wi-Fi and Bluetooth capabilities, have become central to the deployment of such systems. These microcontrollers can interface with multiple sensors (e.g., gas, smoke, flame, temperature, humidity) and communicate data to IoT cloud platforms such as Blynk, Firebase, Adafruit IO, and Thing Speak. These platforms allow developers and users to visualize sensor data, set threshold-based triggers, and receive real-time alerts through mobile applications or web dashboards.

The use of IoT in emergency detection has unlocked several transformational benefits:

- Real-time monitoring: Continuous tracking of environmental conditions without the need for human intervention.
- Instant notifications: Automated alerts sent via mobile apps, SMS, or cloud platforms the moment a threat is detected.

- Remote accessibility: Users can monitor and control the system from anywhere in the world.
- Data analytics: Historical sensor data can be logged and analyzed to detect patterns or recurring risks.
- Affordability and scalability: Low-cost hardware and open-source software make smart systems accessible and adaptable to various settings.

IoT-based emergency detection systems are now increasingly adopted in:

- Smart homes for fire and gas safety.
- Industries for monitoring hazardous gases and overheating machinery.
- Public buildings and transportation hubs for crowd safety and fire prevention.
- Healthcare facilities to safeguard vulnerable populations like the elderly or patients in critical care.

In essence, the integration of IoT into emergency systems represents a significant leap forward from reactive, localized alerts to intelligent, proactive, and connected solutions. These systems are not only more efficient and responsive but also capable of saving lives, minimizing damage, and enhancing public safety in ways that traditional systems simply cannot match.

III. SYSTEM REQUIREMENTS

3.2.1 Hardware Requirements (Detailed)

Component	Specification / Role
ESP32 Microcontroller	Acts as the central processing unit of the system. It connects to Wi-Fi and reads sensor data, processes it, and sends alerts.
MQ2/MQ5 Gas Sensor	Detects gases like LPG, methane, carbon monoxide. It gives analog/digital output when gas exceeds threshold.
Flame Sensor	Detects infrared light emitted by fire or flame. Used for early fire detection.
DHT11 Sensor	Measures ambient **temperature** and **humidity**. Sends digital output to ESP32.
Smoke Sensor	Detects smoke particles in the air, signaling potential fire. Commonly integrated with MQ series.
Buzzer	Emits sound as a local audible alarm when danger is detected.
LEDs (Red/Green)	Indicate system status – normal (green) or emergency (red).
Power Supply Module	Converts AC or USB power to 3.3V/5V for the ESP32 and sensors.
Breadboard and Jumper Wires	Used for prototyping and connecting various electronic components.
Relay Module (Optional)	Used to control external electrical appliances (e.g., turning off a gas valve or fan).
Wi-Fi Router	Provides internet access to ESP32 for IoT communication.

3.2.2 Software Requirements (Detailed)

Software Tool	Description / Purpose
Arduino IDE	Primary platform for writing, uploading, and debugging the code that runs on ESP32.

Software Tool	**Description / Purpose**
Blynk App (IoT Platform)	Used to create a mobile dashboard for real-time data monitoring and push notifications.
Blynk IoT Cloud	Connects the ESP32 to the mobile app, storing sensor data and controlling outputs remotely.
Blynk Auth Token	A unique identifier that connects the ESP32 to the Blynk cloud and mobile dashboard.
Libraries	ESP32 board package, DHT sensor library, Blynk library, etc., added to the Arduino IDE.
Internet Browser	Used for configuring Blynk dashboard or accessing Firebase if cloud storage is implemented.
Serial Monitor	Tool within Arduino IDE to view live sensor readings and debug output via USB connection.

IV. SYSTEM DESIGN AND IMPLEMENTATION

Microcontroller: Arduino Uno / NodeMCU ESP8266

The microcontroller is the central processing unit of the emergency detection system, acting as the core controller that receives input from sensors and provides corresponding outputs to the alert modules. Arduino Uno, based on the ATmega328P microcontroller, is ideal for beginners and prototype-level deployments due to its simple architecture and stable performance. It provides 14 digital I/O pins and 6 analog inputs, suitable for connecting multiple sensors simultaneously.

The NodeMCU ESP8266 offers built-in Wi-Fi, which makes it a superior choice for IoT-based real-time alert systems. It supports wireless communication out of the box, enabling cloud integration and mobile-based control through applications like Blynk. The NodeMCU also supports Lua scripting and Arduino IDE, offering flexibility in development and deployment scenarios.

Arduino Uno is best suited for local-only monitoring systems that do not require internet-based alerts. It ensures reliability in offline environments and is often used in academic and industrial applications where internet connectivity is limited or unnecessary. It operates at 5V logic and is compatible with a wide range of sensors and modules.

In contrast, the NodeMCU operates at 3.3V logic but is ideal for projects involving remote access and notifications. It supports SPI, I2C, and UART protocols and has onboard flash memory, making it useful for data logging or configuration saving. This allows seamless interaction with cloud platforms, APIs, and mobile apps.

Selection between Arduino Uno and NodeMCU depends on the deployment environment. For real-time remote alerts, NodeMCU is preferred, while Arduino is chosen for low-complexity, isolated monitoring systems. Each has its specific advantages based on technical requirements and the nature of deployment.

Both platforms are cost-effective and widely supported by developer communities. They offer abundant libraries, documentation, and compatibility with numerous hardware components. Their reliability and simplicity make them indispensable in prototyping and full-scale smart detection systems.

Fig.4.1: ESP32 DEVKIT V1

Temperature Sensor: LM35

The LM35 temperature sensor is an analog device used to monitor the ambient temperature in emergency-prone environments. It provides a linear voltage output (10mV/°C), which simplifies signal interpretation for microcontrollers. This sensor plays a crucial role in identifying temperature anomalies that could indicate fire or overheating of electronic equipment.

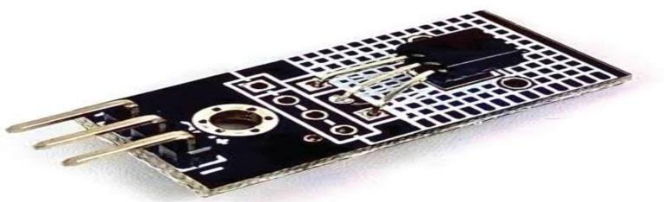

Fig 4.2: Temperature Sensor LM35

Gas Sensor: MQ-2 / MQ-135

Gas detection is vital in emergency systems, especially for identifying hazardous gases like LPG, smoke, CO_2, and ammonia. The MQ-2 is ideal for LPG, smoke, and methane detection, while the MQ-135 is tailored for sensing CO_2 and other toxic gases. Both work on semiconductor gas sensing technology and produce analog signals based on gas concentration.

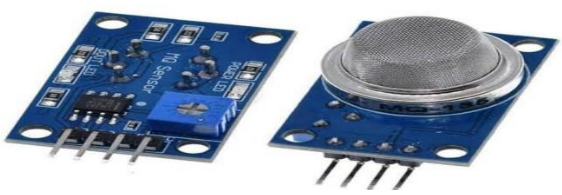

Fig 4.3: Gas Sensor (MQ-135)

Fire Sensor: IR-Based Flame Sensor

The flame sensor detects the presence of fire by identifying infrared light emitted by flames. It operates within a detection angle of 0°–60° and can sense flames up to a certain distance, making it highly effective in identifying fire hazards early. It outputs a digital signal to the microcontroller when it detects a flame within its field of view.

Fig 4.4: Fire Sensor

Ultrasonic Sensor: HC-SR04

The HC-SR04 ultrasonic sensor is primarily used to detect distance by sending ultrasonic pulses and measuring the time it takes for the echo to return. It consists of a transmitter and a receiver and is ideal for applications like obstacle detection, intrusion monitoring, and fall detection in elderly care.

Fig 4.5: Ultrasonic Sensor

Communication Module: GSM Module (SIM800L/SIM900A)

The GSM module allows the emergency detection system to send SMS alerts directly to mobile phones without the need for internet access. This makes it invaluable for areas where Wi-Fi is unreliable or unavailable. Modules like SIM800L and SIM900A are widely supported and cost-effective.

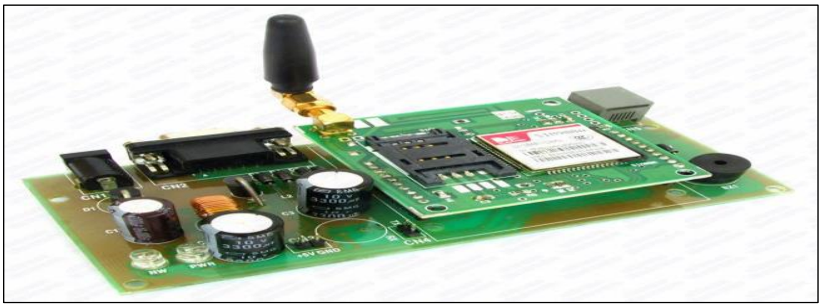

Fig 4.6: GSM Module (SIM900A)

Display: 16x2 I2C LCD

The 16x2 LCD with I2C interface is used to display real-time sensor data, system status, and alerts. It simplifies human interaction by providing textual feedback, which is especially helpful during system setup, monitoring, and debugging.

Fig 4.7: 16x2 I2C LCDDISPLAY

Blynk App

The Blynk App is a mobile platform that allows users to control and monitor IoT devices using a smartphone. It provides an intuitive interface where widgets like buttons, displays, and notifications can be used to interact with the hardware over Wi-Fi or cellular networks. It is particularly suitable for the NodeMCU ESP8266 due to its inbuilt Wi-Fi capabilities.

Using Blynk, developers can remotely view sensor values such as temperature, gas levels, and fire status in real-time. Customizable dashboards make it easy for users to visualize the data. For emergency detection, Blynk can show warning messages, trigger alarms, and even activate hardware remotely if required.

4.2 DESIGN OF PROPOSED SYSTEM

The Smart IoT-Based Emergency Detection System is a cost-effective, efficient, and scalable solution designed for residential, industrial, and institutional environments. It responds in real time to environmental hazards like gas leaks, fire, smoke, and abnormal temperature or humidity. The system is modular, allowing for future expansion or upgrades without altering the core architecture. It is divided into five functional modules: sensing, processing and decision-making, alerting, communication, and user interface. The sensing module monitors environmental conditions through sensors like MQ2 gas, flame, smoke, and DHT11. The processing and decision module interprets sensor values and compares them with predefined threshold levels, identifying potentially dangerous conditions as emergencies. The alerting module issues local alerts to nearby occupants, while the communication module sends critical information remotely to a cloud-based platform like Blynk. The user interface module, developed using the Blynk mobile app, allows users to monitor and interact with the system. The system is designed to be intelligent, responsive, and user-friendly, balancing performance, affordability, and scalability.

4.3 DATA FLOW DIAGRAM (DFD)

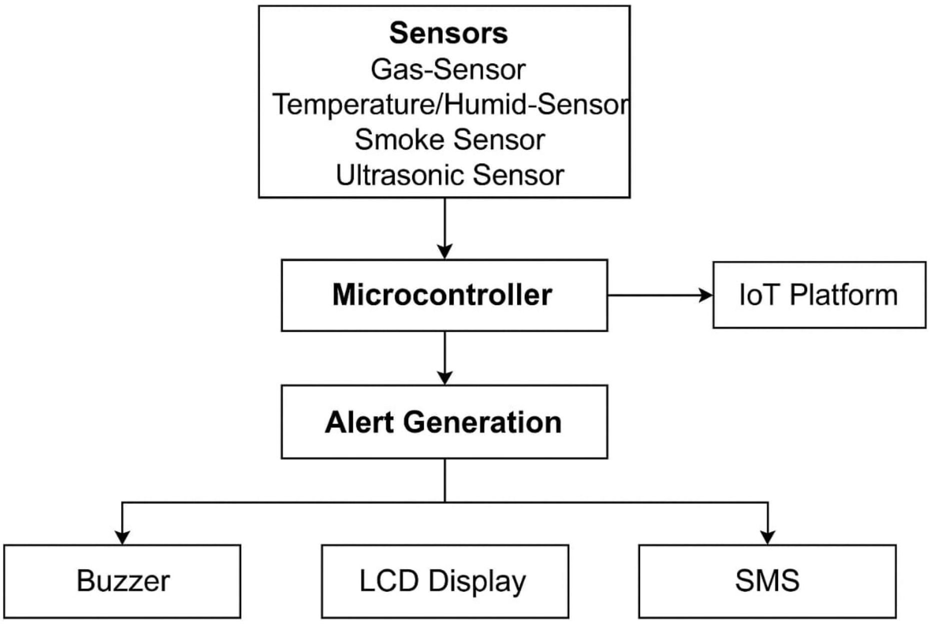

Fig4.8 : Data Flow Diagram

4.4 SYSTEM DESIGN

4.4.1 Block Diagram

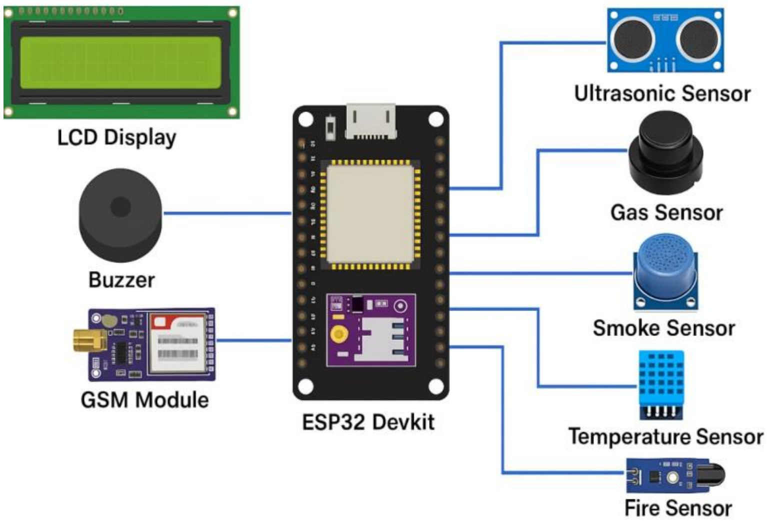

Fig 4.9: Block Diagram of ASmart IOT-Based Emergency Detection System

4.6 TEST CASES

A series of test cases were defined and executed to validate the system's response under various environmental conditions. Each test was performed with the appropriate sensor input, and the output was monitored on both local indicators (LEDs/Buzzer) and the Blynk mobile app.

Test ID	Scenario	Sensor Input	Expected Behavior	Result

Test ID	Scenario	Sensor Input	Expected Behavior	Result
TC-01	Normal environment	All values < threshold	Green LED ON, No alert on app	Pass
TC-02	Gas leakage	MQ2 > 300 ppm	Buzzer ON, Red LED ON, Alert sent to app	Pass
TC-03	Fire/Flame detected	Flame Sensor = HIGH	Buzzer ON, Red LED ON, Push alert to mobile	Pass
TC-04	High temperature	Temp > 40°C	App displays overheat warning	Pass
TC-05	Smoke detected	Smoke level > safe	Buzzer + Notification triggered	Pass
TC-06	Network disconnected	Wi-Fi OFF	Buzzer works, but app does not receive new updates	Pass
TC-07	Sensor failure	Disconnected sensor	No data sent; system continues polling remaining sensors	Pass
TC-08	All hazards at once	All sensors triggered	All alerts activated, real-time multi-alert sent to mobile	Pass

Table 4.1: Test Cases

V. RESULTS AND DISCUSSION

Key Observations and Results

1. **Sensor Responsiveness:**
    - The gas sensor (MQ2) effectively detected LPG or CO presence within 2–3 seconds.
    - The flame sensor reliably triggered an alert within milliseconds of flame exposure.
    - The smoke sensor detected burning or vapor-generating materials with good accuracy.
    - DHT11 provided temperature and humidity updates with ±2°C and ±5% RH accuracy, suitable for basic environmental monitoring.
2. **Real-Time Alert System:**
    - Upon detecting abnormal conditions, the ESP32 immediately activated the buzzer and red LED.
    - Simultaneously, real-time alerts were pushed to the user via the Blynk mobile app using Wi-Fi.
    - The app dashboard displayed live values for all sensors, allowing users to take quick action.
3. **Accuracy of Emergency Detection:**
    - The threshold values were carefully calibrated to reduce false positives.
    - No false alerts were triggered during normal conditions.
    - In edge cases (e.g., high humidity with steam), occasional smoke sensor sensitivity was observed, which can be tuned.
4. **Mobile App Functionality:**

   - Blynk dashboard successfully showed live sensor readings and updated values every few seconds.
   - Users received instant notifications when thresholds were crossed.
   - System control features (like reset) were tested and worked as expected.

5. **System Performance and Stability:**
   - The system ran continuously for several hours without crashing or disconnecting from Wi-Fi.
   - Power consumption was low, making the system suitable for battery-powered or backup-supported environments.
   - The ESP32 microcontroller managed multiple sensor inputs efficiently with minimal delay.

Advantages Noted During Testing:

- **Portability and compact size**, allowing easy installation in different environments.
- **Low cost**, due to the use of affordable and widely available components.
- **Scalability**, since more sensors or modules (like relays or exhaust fans) can be added.
- **Remote access**, enabling user monitoring even when far from the installation site.

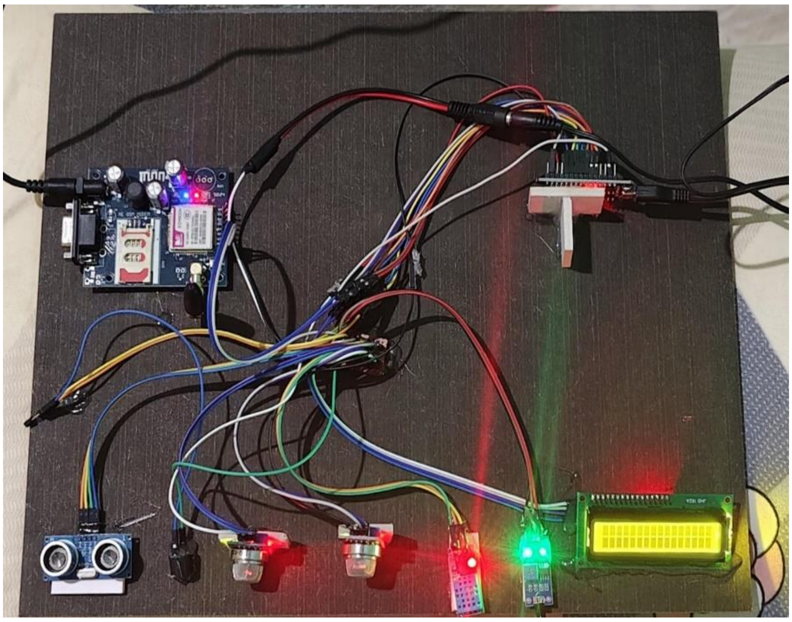

Fig 5.1: Emergency Detection System

Fig 5.2: Alert Message

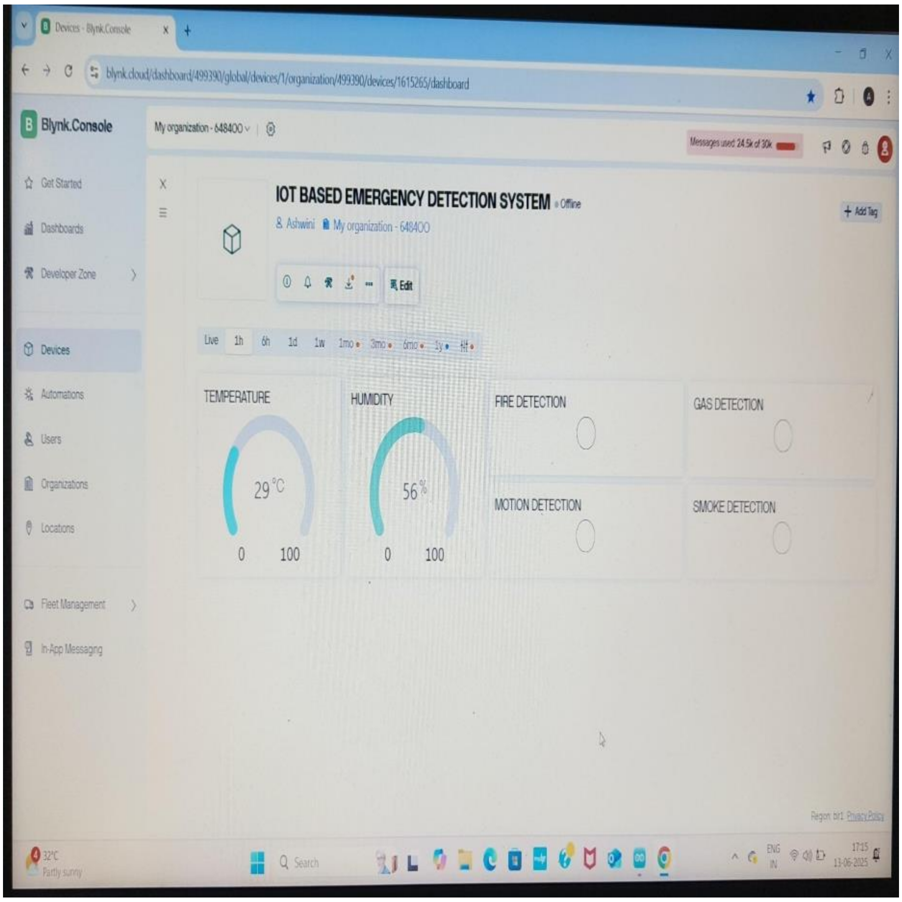

Fig5.3: Snapshot of Blynk

VI. CONCLUSIONAND RECOMMENDATIONS

6.1 CONCLUSION

In this project, a Smart IoT-Based Emergency Detection System was successfully designed, developed, and tested. The primary objective was to create a reliable, low-cost, and efficient system capable of detecting emergency conditions such as gas leaks, fire, smoke, and extreme temperature or humidity levels, and alerting users both locally and remotely.

By leveraging the ESP32 microcontroller, various environmental sensors, and the Blynk IoT platform, the system achieved real-time data acquisition, cloud-based monitoring, and instant user notifications via a smartphone app. The integration of these technologies resulted in a scalable, responsive, and user-friendly solution suitable for smart homes, schools, small industries, and offices.

Key features achieved:

- Real-time monitoring of environmental hazards.
- Push notifications and mobile dashboard using Blynk.
- Immediate local alerts through buzzer and LEDs.
- Modular hardware structure allowing easy expansion.

The system performed effectively during testing and validation, accurately detecting hazardous conditions and responding in real-time without significant delays. It proved to be stable, accurate, and practical for real-world use in safety-critical environments.

This project highlights the potential of IoT-based embedded systems to improve emergency management, reduce human error, and enhance response time during critical situations. It also lays the foundation for future advancements in smart safety systems using cloud computing, AI, and wireless communication technologies.

REFERENCES

[1]. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). *Internet of Things: A survey on enabling technologies, protocols, and applications*. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[2]. Arduino. (n.d.). *Arduino IDE Official Documentation*. Retrieved from: https://www.arduino.cc/en/software

[3]. Espressif Systems. (2023). *ESP32 Technical Reference Manual*. Retrieved from: https://www.espressif.com/en/products/socs/esp32/resources

[4]. Blynk IoT Platform. (n.d.). *Blynk Documentation*. Retrieved from: https://docs.blynk.io

[5]. MQ Gas Sensor Series Datasheet. (n.d.). *MQ-2/MQ-5 Sensor Overview*. Retrieved from: https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-2.pdf

[6]. DHT11 Temperature & Humidity Sensor Datasheet. (n.d.). Retrieved from: https://www.electronicwings.com/nodemcu/dht11-temperature-and-humidity-sensor-interfacing-with-nodemcu

[7]. ThingSpeak. (n.d.). *IoT Analytics with MATLAB &*ThingSpeak. Retrieved from: https://thingspeak.com/

[8]. Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). *Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges*. 2012 10th International Conference on Frontiers of Information Technology. https://doi.org/10.1109/FIT.2012.53

[9]. Firebase Google. (n.d.). *Realtime Database and Cloud Functions for IoT*. Retrieved from: https://firebase.google.com/docs.

[10]. Jain, A., & Bagga, S. (2020). *IoT-based Smart Fire Detection and Alert System*. International Journal of Advanced Research in Computer Science, 11(2), 1-5.