Industry Monitoring & Security System Using IoT

Mr. Chetan Kidile[1], Mr. Kunal Lende[2], Mr. Prana Karkade[3], Mr. Nrupesh Motghare[4], Ms. Dhanashri Wankhade[5], Mr. Niyajkhan Pathan[6], Ms. Rutuja Fulzele[7]

[1]Assistant Professor, Electrical Engineering Department, G H Raisoni College of Engineering & Management, Nagpur
[2,3,4,5,6,7]Students, Electrical Engineering Department, G H Raisoni College of Engineering & Management, Nagpur

Abstract
This paper presents an IoT-based Industry Monitoring and Security System using the NodeMCU ESP8266, integrating sensors like the MQ2 gas sensor, flame sensor, DHT11 temperature and humidity sensor, LDR module, and PZEM-004T for real-time monitoring and control. The system enhances industrial safety by detecting hazards such as gas leaks, fire, and electrical overloads while providing automated control over lighting and machinery to improve operational efficiency. Alerts are sent via Telegram for immediate response, and a relay module automates industrial load management.
The system is powered by a 5V, 2A SMPS stabilized through a 7805 voltage regulator, ensuring reliable operation. It is scalable, cost-effective, and modular, allowing for easy integration into existing setups and expansion with additional sensors. This system enhances both safety and efficiency, making it a versatile solution for diverse industrial environments.

Keywords:
- Real-time monitoring of industrial parameters.
- Automated safety measures for gas leaks, fires, and electrical anomalies.
- Remote control via IoT integration for improved safety and operational efficiency.
- PZEM-004T Module.
- Telegram Notification.

1. Introduction

1.1 Background:

Industrial environments often present high-risk scenarios involving gas leaks, fire, and electrical anomalies. Traditional monitoring methods may not provide real-time hazard detection, making them insufficient for preventing accidents in large-scale industries. With their capability for real-time data collection and immediate hazard alerts, IoT-based solutions can revolutionize industrial safety by improving response times and operational efficiency.
The core of this system is the NodeMCU ESP8266 microcontroller, which integrates multiple sensors to provide continuous monitoring, automation, and remote control capabilities. This paper presents the

design, implementation, and testing of the IoT-based Industry Monitoring and Security System, ensuring real-time safety monitoring and automated responses in industrial settings.

1.2 Problem Statement:

Industrial environments are prone to hazards like gas leaks, fire, and electrical overloads. Existing monitoring systems lack real-time monitoring and immediate response, increasing the risk of accidents. An automated, real-time system capable of detecting such hazards and sending timely alerts to minimize risks and enhance safety is needed.

1.3 Objectives:

This research aims to:

- Provide real-time monitoring of key industrial parameters such as gas concentration, temperature, humidity, and electrical loads.
- Implement automated safety responses for hazard detection, such as gas leaks and fire.
- Enable remote monitoring and control via IoT integration and real-time notifications.
- Optimize energy efficiency through automated lighting and load control.

2. Literature Review

2.1 IoT in Industrial Automation:

IoT technology has become a crucial component in industrial automation because it provides real-time communication and control across systems. Li et al. (2020) discussed how IoT-enabled systems improve operational efficiency by enabling remote control and monitoring, reducing the need for manual supervision.

2.2 Gas and Flame Detection:

Gas and flame detection is essential for preventing accidents in industrial settings. Recent research by Noor et al. (2021) highlighted the importance of integrating gas sensors like MQ2 for reliable detection of hazardous gases. Similarly, Ahmed et al. (2020) emphasized flame sensors' critical role in early fire detection, contributing to faster response times and reducing potential damages.

2.3 Environmental Monitoring and Energy Efficiency:

Maintaining safe environmental conditions is vital in industries. Ogu et al. (2021) demonstrated the effectiveness of using DHT11 sensors for monitoring temperature and humidity in real time, which is important for maintaining equipment and worker safety. Additionally, as discussed by Singh and Kumar (2021), automated lighting systems significantly contribute to energy savings in industrial settings by using LDR sensors to control lighting based on ambient light levels.

3. Abbreviations and Acronyms

- **IoT** – Internet of Things
- **NodeMCU** – Node Microcontroller Unit
- **ESP8266** – A Wi-Fi microchip with full TCP/IP stack and microcontroller capability
- **MQ2** – Metal Oxide Semiconductor Gas Sensor (used for detecting combustible gases)
- **DHT11** – Digital Humidity and Temperature Sensor
- **LDR** – Light Dependent Resistor
- **PZEM-004T** – Power, Energy, Voltage, and Current Monitoring Module
- **SMPS** – Switched-Mode Power Supply

- **LCD** – Liquid Crystal Display
- **I2C** – Inter-Integrated Circuit (a communication protocol)
- **UART** – Universal Asynchronous Receiver-Transmitter
- **TTL** – Transistor-Transistor Logic (a digital logic level standard)
- **LED** – Light Emitting Diode

4. Units

- **Voltage (V)** – Volts
- **Current (A)** – Amperes
- **Power (W)** – Watts
- **Energy (kWh)** – Kilowatt-hours
- **Temperature (°C)** – Degree Celsius
- **Humidity (%)** – Percentage
- **Resistance (Ω)** – Ohms
- **Capacitance (µF)** – Microfarad
- **Frequency (Hz)** – Hertz (for AC communication module)
- **Light Intensity (Lux)** – Unit of illuminance (used for the LDR sensor)
- **Gas Concentration (ppm)** – Parts per million (for gas sensor MQ2)

5. Equations

The system's functionality can be modeled using the following equations, which define how sensor data is processed and corresponding actions are triggered.

A. Gas and Flame Detection

For gas detection, the output of the MQ2 sensor is compared against a threshold concentration level T_g:

$$A_{gas} = \begin{cases} 1 \ if\ S > Tg\ (Gas\ detected), \\ 0\ if\ S \leq Tg\ (no\ Gas\ detected). \end{cases}$$

Where:

- A_{gas} = action triggered (buzzer, notification),
- S = sensor reading from the MQ2 gas sensor,
- Tg = predefined threshold for gas concentration (600 ppm).

Similarly, for flame detection:

$$A_{flame} = \begin{cases} 1\ if\ S = 1\ (\text{flame}\ detected), \\ 0\ if\ S = 0\ (no\ \text{flame}\ detected). \end{cases}$$

Where:

- A_{flame} = action triggered (buzzer, notification),
- S = sensor reading from the flame sensor.

B. Relay Control for Automated Lighting and Load Management

The system controls the relay based on the LDR sensor's light intensity and user commands for the industrial load. If the LDR detects low light levels, the system turns on the lighting:

$$Relay_{light} = \begin{cases} ON\ \ if\ S < T, \\ OFF\ if\ S \geq T. \end{cases}$$

Where:

- $Relay_{light}$ = state of the lightning relay,
- S = sensor reading from the LDR,
- T = threshold for ambient light.

The second relay is controlled for managing an industrial load through manual control on the dashboard.

6. Methodology

6.1 Power Supply and Voltage Regulation:

The system uses a 5V, 2A SMPS power supply to provide the necessary power for the NodeMCU and connected sensors. A 7805 voltage regulator is employed to stabilize the voltage, which receives an input of 24V and outputs a regulated 5V supply. The 7805 circuit uses a 1000μF capacitor for input voltage smoothing and a 10μF capacitor for output smoothing. An LED indicator is connected to show when the correct voltage is supplied.

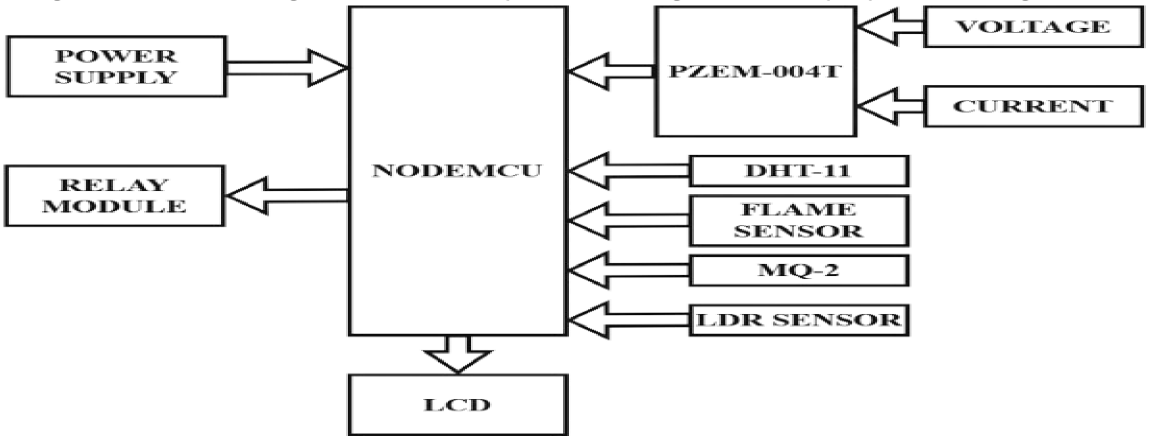
6.2 System Components and Pin Connections:

The NodeMCU serves as the central controller, with the following components connected to it:

- **MQ2 Gas Sensor (A0 Pin):** This sensor continuously monitors gas concentration and sends analog data to the NodeMCU. When gas levels exceed 600 ppm, the system triggers an alert via a buzzer and sends a notification through Telegram.
- **Flame Sensor (D0 Pin):** The flame sensor detects infrared light emitted by a flame. When it detects a fire, the system sends an alert via Telegram and triggers an audible alarm using the buzzer.
- **LCD Display (20x4, SCL on D1, SDA on D2):** The 20x4 LCD provides real-time feedback on gas levels, flame detection status, temperature, and humidity. It is connected to the SCL and SDA pins for I2C communication with the NodeMCU.
- **LDR Module (D3 Pin):** The LDR sensor controls lighting based on ambient light levels. It automates lighting by turning it off during the day and on at night, enhancing energy efficiency.
- **DHT11 Temperature and Humidity Sensor (D4 Pin):** This sensor monitors temperature and humidity levels in the industrial environment. The data is displayed on the LCD and helps ensure that conditions remain within safe operating limits.
- **PZEM-004T AC Communication Module (D5 and D6 Pins):** The PZEM-004T module monitors electrical parameters such as voltage and current. It communicates this data to the NodeMCU using UART communication, allowing operators to track energy consumption and detect potential overloads.
- **Relay Module (D7 and D8 Pins):** Two relays are connected to control external devices. Relay 1 (D7 Pin) controls the automated lighting system based on LDR input, and Relay 2 (D8 Pin) controls the industrial load, allowing for remote switching through the dashboard.
- **Buzzer (D9 Pin):** The buzzer provides an audible alert for gas leaks and fire detection. It operates on a 3.3V to 12V supply.

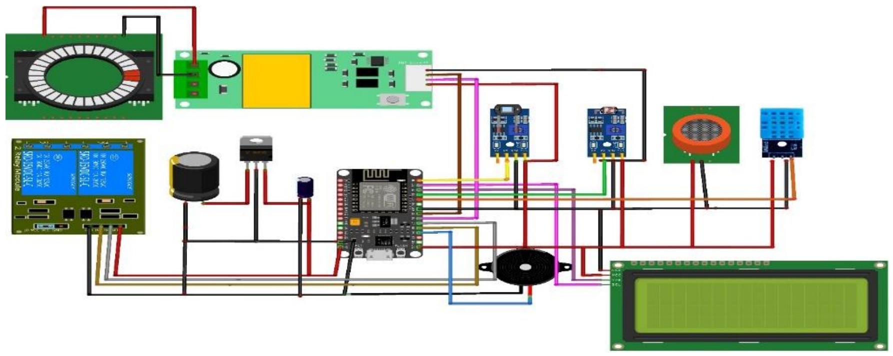
6.3 Block Diagram:

“Figure 1. Block Diagram of Industry Monitoring & Security System Using IoT”



An Industry Monitoring and Security System that integrates multiple sensors and modules, with NodeMCU functioning as the central controller. The system begins with a power supply, which provides the necessary voltage to the NodeMCU and all connected modules. The power is regulated to 5V using a 7805 voltage regulator to ensure stable operation.
At the heart of the system is the NodeMCU, which collects data from various sensors and modules, processes it, and triggers actions based on predefined conditions. The PZEM-004T module monitors voltage and current in the system, sending real-time electrical data to the NodeMCU. This data is vital for monitoring industrial loads and ensuring safety.
The system also includes several environmental sensors. The DHT11 sensor monitors temperature and humidity levels, providing critical environmental data for industrial safety. The flame sensor detects the presence of fire or heat, while the MQ2 gas sensor identifies hazardous gases like methane or propane. These sensors ensure that abnormal conditions are quickly detected, and appropriate actions are taken to mitigate risks. Additionally, the LDR sensor measures ambient light levels, which could be used to automate lighting systems or detect industrial lighting failures.
The relay module acts as a control mechanism for managing high-power electrical loads such as industrial machinery or lighting. Based on the inputs from the sensors, the NodeMCU sends signals to the relay module to either activate or deactivate specific devices, providing real-time automation and safety control. The system also includes an LCD that shows real-time sensor data such as temperature, humidity, gas concentration, voltage, and current. This allows on-site operators to monitor the system’s status directly.

6.4 Circuit Diagram:



“Figure 2. Circuit Diagram of Industry Monitoring & Security System Using IoT”

The circuit in the diagram illustrates an IoT-based Industry Monitoring System with multiple sensors and modules controlled by a NodeMCU. Here's a step-by-step explanation of the connections and the working of the circuit based on the provided description:

Power Supply and Voltage Regulation:

- A 5V, 2A SMPS power supply is connected to a 7805 voltage regulator. The regulator ensures a stable 5V output for powering the NodeMCU and the other components.
- A 1000µF capacitor (24V) is used at the regulator's input to filter noise, and a 10µF capacitor (16V) is used at the output for voltage smoothing.
- An LED is connected to the power circuit as an indicator. When the LED is on, it confirms that the correct voltage (5V) is supplied to the circuit.

NodeMCU and Sensor Connections:

The NodeMCU is the main controller that coordinates the functioning of all sensors and modules.

1. Gas Sensor (MQ2):

- The A0 pin of the NodeMCU is connected to the MQ2 gas sensor. This sensor monitors the gas concentration in the environment (up to 600 ppm) and provides an analog signal to the NodeMCU.

2. Flame Sensor:

- The D0 pin of the NodeMCU is connected to the flame sensor, which detects the presence of fire. A notification is sent via Telegram, and a dashboard alert is triggered if a flame is detected.

3. LCD Display (20x4):

- The D1 (SCL) and D2 (SDA) pins of the NodeMCU are connected to the I2C interface of the 20x4 LCD. This display shows real-time data from the sensors (e.g., temperature, humidity, gas concentration, etc.).

4. LDR Module:

- The D3 pin is connected to the LDR module, which detects the ambient light level. When the LDR senses low light (indicating nighttime), it automatically triggers the connected relay to turn on the lighting or load.

5. DHT11 (Temperature and Humidity Sensor):

- The D4 pin is connected to the DHT11 sensor, which provides real-time temperature and humidity data to the NodeMCU. This data is displayed on the LCD and can trigger alerts if abnormal conditions are detected.

PZEM-004T Communication Module:

- The NodeMCU's D5 and D6 pins are connected to the PZEM-004T communication module, which measures voltage and current. This module sends data to the NodeMCU using UART communication (TTL logic).
- The PZEM-004T allows the system to monitor the load connected via Relay 2 and send voltage and current data to the dashboard for real-time monitoring.

Relay Module:

- The system uses a relay module to control high-power loads (such as lighting or industrial equipment).
  - D7 pin: Controls Relay 1, connected to the LDR module. It turns the load on or off based on light conditions (e.g., nighttime or daytime).
  - D8 pin: Controls Relay 2, connected to a load monitored by the PZEM-004T module. This ensures the load's voltage and current are continuously monitored.

Buzzer and Notifications:

- A buzzer is connected to the D9 pin of the NodeMCU. The buzzer is triggered when gas levels exceed 600 ppm or when a flame is detected. The buzzer operates between 3.3V and 12V and serves as an audible alert in addition to the notifications sent via Telegram.

Working Overview:

Using the respective sensors,

- The system continuously monitors gas levels, flame detection, temperature, humidity, and ambient light. The data from the PZEM-004T module provides real-time feedback on voltage and current consumption, allowing for effective monitoring of connected loads.
- The NodeMCU processes data from these sensors and takes actions such as turning on/off the relays (for lighting control) and activating the buzzer for alarms. It also communicates with the dashboard and Telegram for remote monitoring and control.
- The real-time data is displayed on the LCD screen for local monitoring, making this system highly suitable for industrial environments that require constant supervision and automated control for safety and efficiency.

6.5 Software Development:

The system's software was developed using the Arduino IDE. The code handles the initialization and communication of all sensors, processes the incoming data, and controls outputs based on predefined conditions. The NodeMCU also communicates with Telegram to send real-time notifications of detected hazards.

Gas Detection Algorithm:

- **Trigger Condition:** If the MQ2 gas sensor detects a gas level greater than 600 ppm.
- **Response:** Sound the buzzer, & Send a notification to Telegram: "Smoke Detected".
- **Code Explanation:** The gas sensor is monitored by reading the analog value from the MQ2 sensor pin. If the reading exceeds 600 ppm, the system sends a message to the user via Telegram and triggers the buzzer for audible alerts.
- **Code:**

```
server.on("/gas-sensor", []()
{
int gas = analogRead(gasSensor); // Reading the gas sensor value
gasValue = String(gas); // Convert value to string for display
server.send(200, "text/plain", gasValue);
lcd.setCursor(10,1); // Displaying the gas value on LCD
lcd.print("G=");
lcd.print(gasValue);
lcd.print("ppm");
if(gas >= 600) { // Trigger if gas level is above 600 ppm
bot.sendMessage(CHAT_ID, "Smoke Detected", ""); // Send Telegram alert
tone(buzzer, 500, 2000); // Activate buzzer
}
});
```

Flame Detection Algorithm:

- **Trigger Condition:** If the flame sensor detects fire (value is LOW).
- **Response:** Trigger the buzzer, & Send a notification to Telegram: "Fire Detected".
- **Code Explanation:** The flame sensor monitors for fire conditions. When a flame is detected, the system triggers a notification to Telegram and an alarm through the buzzer. Lighting Control
- **Code:**

```
server.on("/flame sensor", []()
{
int flame = digitalRead(flamesensor); // Read flame sensor value
flameValue = String(flame);
server.send(200, "text/plain", flameValue);
if(flame == 0) { // Flame detected (LOW)
lcd.setCursor(0, 2); // Display message on LCD
lcd.print("Fire=");
lcd.print("Fire Detected");
bot.sendMessage(CHAT_ID, "Fire Detected", ""); // Send Telegram alert
tone(buzzer, 500, 2000); // Activate buzzer
} else {
lcd.setCursor(0, 2);
lcd.print("Fire=");
lcd.print("Normal       "); // Normal condition
}});
```

Lighting Control Algorithm:

- **Trigger Condition:** The LDR sensor detects ambient light conditions.
- **Response:** If it is dark (LDR reads HIGH), turn on lighting via Relay 2, & If it is bright (LDR reads LOW), turn off lighting.
- **Code Explanation:** The LDR sensor is monitored using a digital input. Based on the value (HIGH for the night, LOW for the day), the system automatically controls Relay 2 to switch on/off lighting.
- **Code**:

```
server.on("/ldr-sensor", []()
{
int ldr = digitalRead(ldrSensor); // Read LDR sensor value
ldrValue = String(LDR);
server.send(200, "text/plain", ldrVvalue);
if(ldr == 0) { // Day condition
lcd.setCursor(0, 3); // Display on LCD
LCD.print("LDR=Day  ");
digitalWrite(Relay2, HIGH); // Turn off lighting
} else { // Night condition
lcd.setCursor(0, 3);
LCD.print("LDR=Night") ;
digitalWrite(Relay2, LOW); // Turn on lighting
}
});
```

Electrical Load Monitoring:

- **Functionality:**
  - Monitor the voltage and current values using the PZEM-004T module.
  - Display these values on the LCD and make them accessible via the web server.
  - If any anomalies in voltage or current are detected, a message is sent via Telegram.
- **Code Explanation:** Voltage and current are measured using the PZEM-004T module. The values are displayed on the LCD and can be accessed through the web server.
- **Code:**

```
server.on("/voltage", []()
{
float voltage = pzem.voltage(); // Get voltage reading
voltageValue = String(voltage); // Convert to string
server.send(200, "text/plain", voltageValue); // Send to web server
lcd.setCursor(0, 0); // Display on LCD
lcd.print("V:");
lcd.print(voltage);
lcd.print("V |");
});
server.on("/current", []()
float current = pzem.current(); // Get current reading
currentValue = String(current); // Convert to string
server.send(200, "text/plain", currentValue); // Send to web server
lcd.setCursor(11, 0); // Display on LCD
lcd.print("I:");
lcd.print(current);
lcd.print("A");
});
```

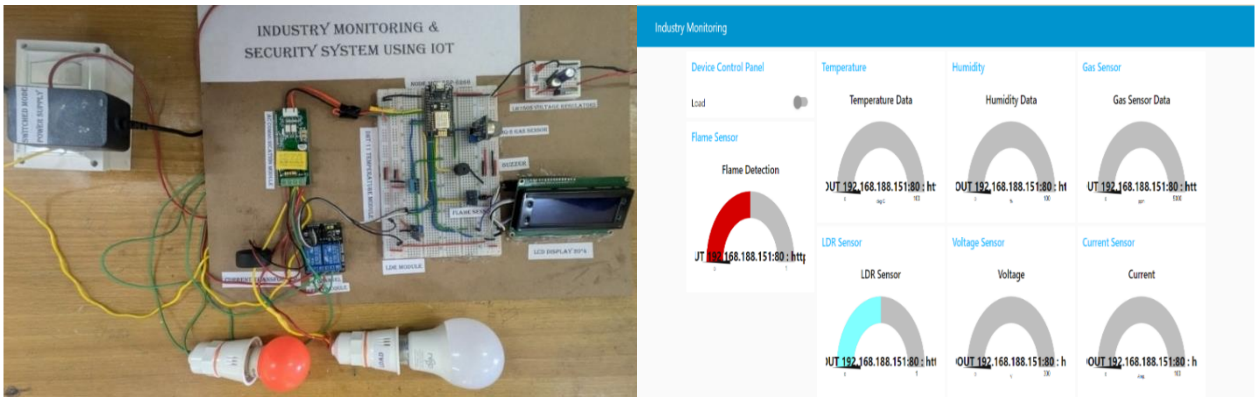
- **System Initialization:** When the system is initialized, a message is sent to Telegram indicating that the system is ready.
  bot.send Message(CHAT_ID, "System is Ready", "");
- **Relay Control (Manual via Web Server): Relay 1** can be controlled manually through the web server, allowing for external control of connected devices.
- **Code:**

```
server.on("/relay1/on", []()
{
digitalWrite(Relay1, HIGH); // Turn on Relay 1
server.send(200, "text/plain", "Relay 1 turned ON");
});
server.on("/relay1/off", []()
{
```

```
digitalWrite(Relay1, LOW); // Turn off Relay 1
server.send(200, "text/plain", "Relay 1 turned OFF");
});
```

7. Results and Discussions

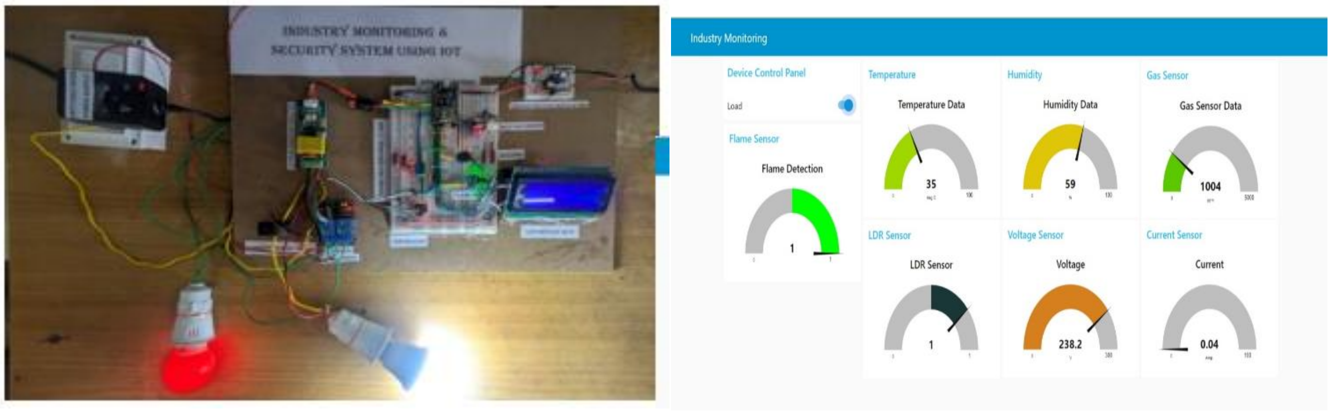
Before:



"Figure 3. System Is In OFF Condition"

An Industry Monitoring and Security System using IoT is designed to enhance safety, efficiency, and real-time oversight in industrial environments through the integration of connected devices, sensors, and analytics. In the "Before" state, when the system is off, critical safety parameters like flame detection, temperature, humidity, and gas concentration remain unmonitored. The dashboard, which visualizes real-time data, displays no information, creating vulnerabilities such as undetected gas leaks or equipment malfunctions. Without active monitoring, any potential hazards can escalate unchecked, jeopardizing both safety and operational efficiency.

After:-



"Figure 4. System Is In ON Condition"

In the "After" state, the system's activation marks a significant transformation. Once powered on, sensors including flame detectors, temperature sensors like the DHT11, and gas sensors such as the MQ2 begin collecting and transmitting data to a NodeMCU microcontroller. This data is processed and displayed in real time on the dashboard. For instance, the flame sensor actively scans for fire, issuing alerts if a flame is detected, while the DHT11 continuously monitors temperature and humidity to ensure safe conditions. The MQ2 sensor is vital for detecting harmful gases, alerting the management team if concentrations exceed safe thresholds.

To optimize operational efficiency, the system features an LDR sensor that automates lighting based on ambient light levels. It turns on lights during low-light conditions and switches them off in daylight, enhancing safety and conserving energy. Additionally, the PZEM-004T module tracks real-time voltage and current consumption, providing insights into electrical performance and helping operators manage loads to prevent overloads. Continuous energy monitoring helps identify inefficiencies and optimize power usage, leading to potential cost savings.

7.1. Result:

- **Gas Detection:** The MQ2 sensor reliably detected gas levels above 600 ppm and triggered the buzzer. The notification system worked flawlessly, sending alerts to Telegram in real time.
- **Flame Detection:** The flame sensor detected fire within 50 cm, immediately triggered the alarm, and sent a notification.
- **Temperature and Humidity Monitoring:** The DHT11 sensor provided accurate readings that were displayed in real-time on the LCD, ensuring optimal environmental conditions.
- **Lighting Automation:** The LDR module successfully automated lighting, turning the lights on when ambient light levels dropped and off during daylight.
- **Electrical Load Monitoring:** The PZEM-004T module accurately tracked voltage and current, providing real-time data for load management.

7.2. Performance Data:

“Table 1. Showing Performance of the System”

Parameter	Sensor Used	Measured Value	Accuracy
Gas Concentration	MQ2	600 ppm	±50 ppm
Flame Detection	Flame Sensor	Detected	100%
Temperature	DHT11	25°C	±2°C
Humidity	DHT11	45%	±5%
Voltage	PZEM-004T	230V	±1V
Current	PZEM-004T	10A	±0.1A

7.3. Discussion:

The system effectively monitored environmental and electrical conditions and responded to real-time hazards. Integrating multiple sensors with NodeMCU allowed for seamless communication, and the Telegram notifications ensured timely responses to critical events. However, Future improvements to the system could focus on enhancing network reliability by implementing WiFi reconnection logic and adding backup connectivity options. Expanding capabilities with advanced CO2, vibration, or weather sensors would enable more precise monitoring.

8. Conclusion

This research demonstrates the potential of IoT-based systems to enhance industrial safety through real-time monitoring and automation. Integrating sensors such as MQ2 for gas detection, flame sensors, and the DHT11 for temperature and humidity enables continuous monitoring and automated responses to hazards, improving safety and operational efficiency. NodeMCU serves as the core, processing sensor data and sending real-time alerts via platforms like Telegram. In addition to hazard detection, the system automates lighting based on ambient light and controls industrial loads, optimizing energy use. Voltage and current monitoring through the PZEM-004T module ensures safe load management. Future improvements could strengthen network reliability and expand capabilities by integrating more advanced sensors, such as smoke detectors, to enhance safety features further. Overall, the system provides a scalable, efficient industrial safety and energy management solution.

9. References

1. X. Li, S. Wang, Y. Chen, "IoT-Based Industrial Monitoring System for Real-Time Data Collection and Control," *Journal of Industrial Electronics*, 2020, 45 (3), 120–130.
2. A. Noor, I. Ahmad, S. Rahman, "Gas Leak Detection in Industrial Environments Using IoT-Based Systems," *IEEE Access*, 2021, 9, 18947–18956.
3. Z. Ahmed, A. Khan, S. Malik, "Fire Detection Using IoT and Cloud-Based Monitoring in Industrial Settings," *Sensors*, 2020, 20 (9), 2754.
4. M. N. Ogu, A. O. Nwankwo, "Environmental Monitoring in Industrial Plants Using IoT Sensors," *Sustainability*, 2021, 13 (2), 843–860.
5. R. Singh, V. Kumar, "IoT-Based Smart Lighting Control for Energy Efficiency in Industrial Settings," *Journal of Energy Systems*, 2021, 26 (1), 25–35.