CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Internet of Things (IoT) is an ideal buzzing technology to influence the Internet and communication technologies. IoT allows people and things to be connected anytime, anyplace, with anything and anyone, by using ideally in any path/network and any service. This project introduces a thought or an idea for home computerization utilizing voice acknowledgment, also the development of a prototype for controlling smart homes devices through IoT and controlling of dumb devices through IoT by the means of Wi- Fi driven chipset solution – ESP8266. This is also acknowledged by the need to give frameworks which offers help to matured and physically impaired individuals, particularly individuals who lives alone. Smart home or home automation can be said as the residential extension of building automation, it also involves the automation and controlling of lightings, ACs, ventilation and security which also includes home appliances such as dryers/washers, ovens or refrigerators/freezers which uses Wi-Fi for monitoring via remote for ease of use. Now a day's speed of the processing and communication through smart mobile devices at very affordable costs, to improve the lifestyle concept relevant to smart life, like smart T.V, Smart cities, smart phones, smart life, smart school and Internet of Things.

1.2 INTRODUCTION OF IoT TECHNOLOGY

- The Internet of Things (IoT) refers to the network of interconnected devices that communicate and share data with each other over the internet. These devices can range from everyday household items to industrial machinery.
- The term "Internet of Things" has come to describe a number of technologies and research disciplines that enable the Internet to reach out into the real world of physical objects. The Internet of Things, also called The Internet of Objects, refers to a wireless network between objects. From any time, any place connectivity for anyone, we will now have connectivity for anything.

- IoT involves embedding sensors, software, and other technologies into physical objects, allowing them to collect and exchange data. This connectivity enables devices to be monitored and controlled remotely, creating a smarter and more automated world.

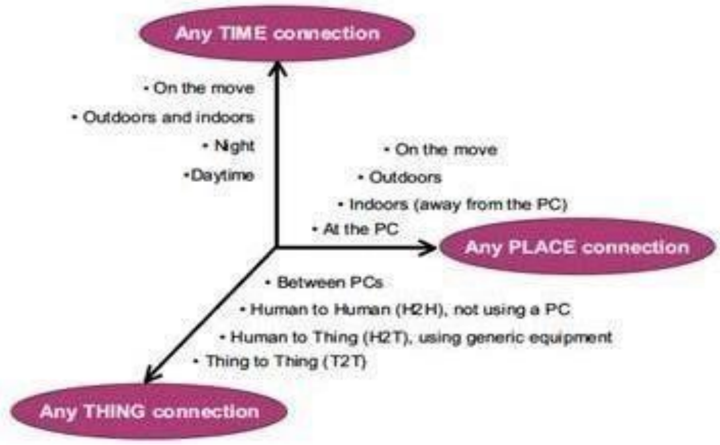


Figure 1: IoT Technology

1.3The Vision

To improve human health and well-being is the ultimate goal of any economic, technological and social development. The rapid rising and aging of population is one of the macro powers that will transform the world dramatically, it has caused great pressure to food supply and healthcare systems all over the world, and the emerging technology breakthrough of the Internet-of-Things (IoT) is expected to offer promising solutions. Therefore, the application of IoT technologies for the food supply chain (FSC) (so-called Food-IoT) and in-home healthcare (IHH) (so called Health-IoT1) have been naturally highlighted in the strategic research roadmaps.

To develop practically usable technologies and architectures of IoT for these two applications is the final target of this work. The phrase "Internet of Things" (IoT) was coined at the beginning of the 21st century by the MIT Auto-ID Centre with special mention to Kevin Ashton and David L. Brock.

As a complex cyber-physical system, the IoT Integrates all kinds of sensing, identification, communication, networking, and informatics devices and systems, and seamlessly connects all the people and things upon interests, so that anybody, at any time and any place, through any device and media, can more efficiently access the information Micro Electro Mechanical Systems (MEMS), mobile internet access, cloud computing, Radio Frequency Identification (RFID), Machine-to-Machine (M2M) communication, human machine interaction (HMI), middleware, Service Oriented Architecture (SOA), Enterprise Information System (EIS), data mining, etc. With various descriptions from various viewpoints, the IoT has become the new paradigm of the evolution of information and communication technology (ICT).

1.4 Definition of Internet of things (IoT)

Today computers and, therefore, the Internet are almost wholly dependent on human beings for information. Nearly all of the roughly 50 petabytes (a petabyte is 1,024 terabytes) of data available on the Internet were first captured and created by human being by typing, pressing a record button, taking a digital picture, or scanning a bar code.

Conventional diagrams of the Internet … leave out the most numerous and important routers of all - people. The problem is, people have limited time, attention and accuracy all of which means they are not very good at capturing data about things in the real world. And that's a big deal. We're physical, and so is our environment … You can't eat bits, burn them to stay warm or put them in your gas tank. Ideas and information are important, but things matter much more. Yet today's information technology is so dependent on data originated by people that our computers know more about ideas than things. If we had computers that knew everything there was to know about things using data they gathered without any help from us we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best. The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so".

"Things are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real/physical world events and influencing it by running processes that trigger actions and create services with or without direct human intervention."

1.5 Key Components of IoT

- **Devices/Sensors:** These are physical objects embedded with sensors and actuators that collect and transmit data. Examples include smart thermostats, wearable fitness trackers, and industrial sensors.
- **Connectivity:** Devices connect to the internet or other networks using various technologies such as Wi-Fi, Bluetooth, Zigbee, or cellular networks.
- **Data Processing:** The data collected by IoT devices is often processed either on the device itself or sent to a cloud-based server. This data can be analyzed to gain insights, make decisions, or trigger actions.
- **User Interface:** Users interact with IoT systems through applications or dashboards, allowing them to monitor and control devices.

1.6 Applications of IoT

- **Smart Homes:** Automate and control home systems like lighting, heating, and security remotely.
- **Healthcare:** Monitor health metrics and track medication adherence with connected devices.
- **Industrial IoT:** Predict equipment failures, optimize manufacturing, and manage supply chains.
- **Agriculture:** Enhance farming with precision tools, automated irrigation, and livestock monitoring.
- **Transportation:** Track and manage vehicles, optimize routes, and support autonomous driving.
- **Smart Cities:** Improve public safety, waste management, and traffic flow with connected infrastructure.
- **Retail:** Manage inventory, personalize customer experiences, and monitor supply chains.
- **Energy Management:** Optimize energy use with smart grids and renewable energy integration.
- **Environmental Monitoring:** Track climate, pollution, and wildlife to better respond to natural events.
- **Education:** Enhance learning environments with smart classrooms and campus security systems.

1.7 Benefits of IoT

- Improved citizen's quality of life Healthcare from anywhere
- Better safety, security and productivity
- IoT can be used in every vertical for improving the efficiency
- Creates new businesses, and new and better jobs
- Economic growth
- Billions of dollars in savings and new services
- Better environment
- Saves natural resources and trees
- Helps in creating a smart, greener and sustainable planet

1.8 Characteristics for Internet of Things:

- Event driven
- Ambient intelligence
- Flexible structure
- Semantic sharing
- Complex access technology

Anyone who says that the Internet has fundamentally changed society may be right, but at the same time, the greatest transformation actually still lies ahead of us. Several new technologies are now converging in a way that means the Internet is on the brink of a substantial expansion as objects large and small get connected and assume their own web identity.

Following on from the Internet of computers, when our servers and personal computers were connected to a global network, and the Internet of mobile telephones, when it was the turn of telephones and other mobile units, the next phase of development is the Internet of things, when more or less anything will be connected and managed in the virtual world.

Smart connectivity with existing networks and context-aware computation using network resources is an indispensable part of IoT. With the growing presence of Wi-Fi and 4G-LTE wireless Internet access, the evolution towards ubiquitous information and communication networks is already evident. However, for the Internet of Things vision to successfully emerge, the computing paradigm will need to go beyond traditional mobile computing scenarios that use smart phones and portables, and evolve into connecting everyday existing objects and embedding intelligence into our environment. For technology to disappear from the consciousness of the user, the Internet of Things demands: a shared understanding of the situation of its users and their appliances, software architectures and pervasive communication networks to process and convey the contextual information to where it is relevant, and the analytics tools in the Internet of Things that aim for autonomous and smart behavior. With these three fundamental grounds in place, smart connectivity and context-aware computation can be accomplished.

CHAPTER II
LITERATURE SURVEY

In Journal titled **Automation of LPG Cylinder Booking and Leakage Monitoring System, P. Bharath et al.** proposed an automated system for gas booking and leakage detection. The system automatically books the cylinder before the LPG supply is fully utilized and alerts the owner in case of a leakage. Continuous monitoring of the cylinder's weight is carried out using a weight sensor, and if the weight decreases, below a preset amount, the technology automatically books the cylinder and notifies the owner via message.

In the research titled **"Kernel Approach on Detection of Ethanol Connection Using ZnO Gas Sensor," L. Shaw et al.** examined the clustering of various ethanol gas concentration levels. It was demonstrated that kernel principal component analysis could be employed to cluster data in this scenario. Different gas concentrations were detected and displayed using the proposed technique on the LabView front panel.

In a project proposal by **V. Ramya et al. the focus is on an embedded system designed for hazardous gas detection and warning.** The project aims to modify the current safety paradigm used in industries and extend its applicability to homes and workplaces. The primary objective of the project is to design a hazardous gas detection and alerting system using a microcontroller. The system includes an LCD display that continuously monitors and displays dangerous gases such as LPG and propane. Additionally, if these gases exceed the normal levels, triggering an alarm, an alert message (SMS) is sent via GSM to the appropriate individual. The automatic detection and warning system offer advantages over the manual approach, including fast response times and precise detection.

In a project proposed by **T. Kiran et al. a "Gas Leakage Detection System" utilizing FPGA and GSM technology was. introduced.** The study suggested a system for detecting gas leaks, wherein the first response team would receive wireless notifications containing information about the leak. This enables prompt preventive measures to be taken, even in the absence of individuals. The detection system automatically places a warning call via GSM to identify the leakage using FPGA.

LPG was utilized to test and develop a prototype of the gas leakage detection system. According to trial findings, the system can detect leaks in less than a minute.

CHAPTER III
OVERVIEW OF THE PROJECT

3.1 INTRODUCTION

Liquefied Petroleum Gas (LPG) is a commonly used fuel in residential, commercial, and industrial environments due to its affordability, efficiency, and ease of use. Despite its advantages, LPG is highly flammable and potentially hazardous. when leaks occur. Gas leaks, if undetected, can lead to catastrophic consequences such as fires, explosions, property damage, and even loss of life. These risks are heightened in enclosed spaces where leaked gas can accumulate without ventilation. Unfortunately, many gas leak incidents go unnoticed in their early stages due to the limitations of traditional detection methods, which often rely solely on manual monitoring or basic alarm systems. These conventional approaches offer limited response time and typically require the presence of someone nearby to intervene, which can delay action during critical moments.

The advancement of Internet of Things (IoT) technology has enabled the development of intelligent, automated systems capable of enhancing safety standards across various domains. IoT facilitates real-time monitoring, automation, and remote communication, making it highly suitable for gas leak detection and control systems. This project aims to design and implement an loT-based gas leakage detection and prevention system that not only identifies the presence of leaked LPG but also initiates immediate safety responses and notifies the user remotely. The system uses an MQ-6 gas sensor to detect gas concentration in the environment and a NodeMCU ESP8266 microcontroller to process sensor data and control connected devices.

Upon detecting a gas leak, the system performs several automated safety actions: it triggers a buzzer and LED to alert people in the vicinity, activates a servo motor to open a nearby window for ventilation, and optionally shuts off the gas valve to prevent further leakage. Moreover, through its IoT integration, the system sends a real-time notification to the user's smartphone, allowing them to monitor the situation and remotely shut off the gas supply if necessary. This proactive approach reduces the dependency on human intervention and ensures that appropriate measures are taken even when no one is present at the site.

By combining sensor-based detection, mechanical automation, and mobile communication, the proposed system offers a comprehensive, reliable, and cost effective solution to mitigate the dangers of gas leakage. Its design emphasizes safety, accessibility, and real-time responsiveness, making it suitable for implementation in homes, restaurants, small-scale industries, and other environments where LPG is used. However, with Intelligent Gas leakage detection system, industries continuously analyzes data and detects potential leaks in real-time In this introduction, we will explore the key features of Intelligent gas detection system, its impact on industrial gas leak detection, and the benefits it brings to various industries. We will also discuss how its innovative use of IOT has transformed the way gas leaks are detected and prevented, making it an essential tool for ensuring safety and efficiency in the workplace. Installing gas leakage detecting equipment is one of the preventative methods to stop accidents caused by gas leakage, which is a serious problem in the industrial sector. Arduino now reads the output from the sensors. It is suggested that physical objects and the internet be connected through the Internet of Things (IOT), a futuristic technology. Safety comes first, thus the suggested gas detection system uses IOT to identify leaks and notify users so they can stop them. Since the gases are dangerous, it is necessary to keep an eye on them in order to spot any changes in their concentration and respond appropriately.

3.2 EXISITING SYSTEM OF THE PROJECT

In current industrial environments, hazardous gas leakage detection systems have evolved into smart, automated solutions that integrate advanced sensors, microcontrollers, and communication technologies. These systems typically employ gas sensors such as the MQ series to continuously monitor the presence of toxic or flammable gases like methane, carbon monoxide, or ammonia. When a leak is detected, the microcontroller processes the sensor data and triggers immediate safety responses, including activating alarms, switching on exhaust fans, and shutting off gas valves through relays or solenoids. Many systems are equipped with GSM or Wi-Fi modules to send real-time alerts via SMS or mobile applications, ensuring rapid response even when personnel are off-site. Some setups also incorporate GPS modules for location tracking and cloud-based dashboards for remote monitoring and data analytics. While these systems significantly enhance industrial safety, they often lack predictive capabilities and integration with broader industrial automation platforms, highlighting opportunities for further innovation.

In addition to basic detection and response, existing systems often include user interfaces such as LCD screens or mobile apps to display gas concentration levels and system status. These systems are designed to be scalable and adaptable to various industrial environments, including chemical plants, oil refineries, manufacturing units, and storage facilities. Despite their effectiveness, many current models are reactive rather than predictive, lacking machine learning algorithms that could forecast potential leaks based on historical data and environmental patterns. Moreover, integration with broader industrial automation platforms like SCADA (Supervisory Control and Data Acquisition) is limited in low-cost implementations. This opens up opportunities for future enhancements, such as AI-driven diagnostics, multi-gas detection arrays, and real-time decision-making systems that not only detect but also anticipate and prevent hazardous incidents.

3.3 PROPOSED SYSTEM

The proposed system aims to enhance industrial safety by developing a smart, real-time hazardous gas leakage detection and automated response mechanism using advanced IoT and AI technologies. It will utilize a multi-gas sensor array capable of detecting a wide range of toxic and flammable gases, integrated with a microcontroller such as ESP32 or Raspberry Pi for efficient data processing. The system will continuously monitor gas concentrations and environmental parameters like temperature and humidity, and employ machine learning algorithms to predict potential leak scenarios based on historical trends and sensor patterns. Upon detecting abnormal gas levels, the system will automatically trigger safety protocols including activating alarms, switching on exhaust fans, and shutting off gas valves using relay-controlled actuators. It will also send instant alerts to safety personnel via SMS, email, or mobile app notifications using GSM or Wi-Fi modules. A cloud-based dashboard will provide remote monitoring, data visualization, and analytics for maintenance planning and incident reporting. Additionally, GPS integration will enable precise location tracking of leak incidents, and the system will be scalable for deployment across multiple industrial zones. This intelligent, proactive approach not only improves response time but also minimizes human intervention and enhances overall workplace safety.

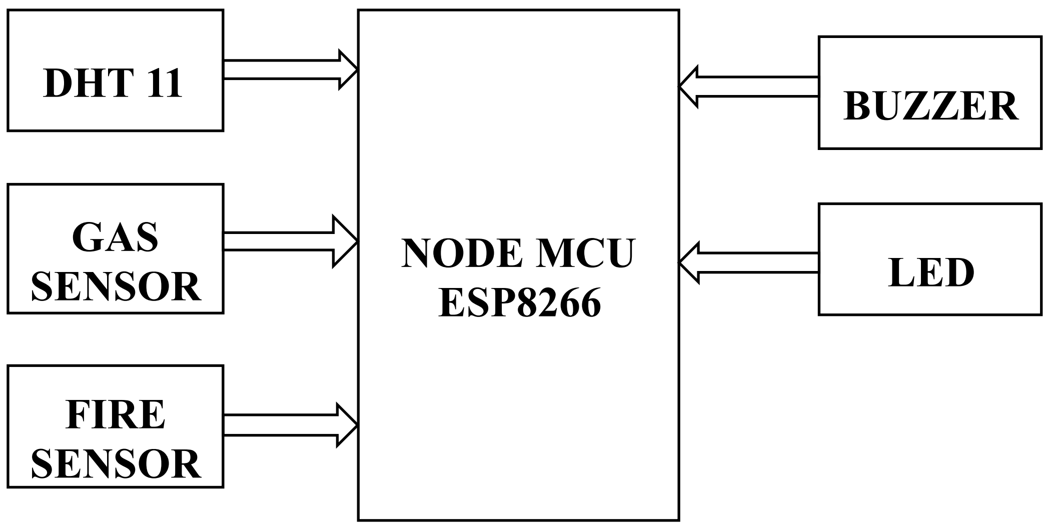
To enhance intelligence and predictive capabilities, the system will incorporate machine learning algorithms trained on historical sensor data to identify patterns and forecast potential leak scenarios before they occur. This proactive approach will allow for early intervention and reduce the risk of catastrophic incidents. Upon detecting abnormal gas levels, the system will automatically initiate a series of safety protocols: activating audible and visual alarms, powering exhaust fans, and shutting off gas supply lines using relay-controlled solenoid valves. These actions will be executed with minimal human intervention, ensuring rapid and reliable containment.

Moreover, the system is designed with scalability and modularity in mind, allowing it to be adapted to various industrial settings such as chemical plants, oil refineries, food processing units, and manufacturing facilities. It supports additional features like battery backup for uninterrupted operation during power outages, tamper detection for enhanced security, and integration with fire suppression systems for comprehensive hazard control. The cloud-based dashboard also provides historical data logs, enabling safety audits, compliance reporting, and predictive maintenance planning. By combining real-time detection, automated response, and intelligent analytics, the proposed system offers a cutting-edge solution that not only mitigates risks but also fosters a culture of proactive safety management in industrial operations.

To enhance accessibility and scalability, the system integrates wireless communication modules like Wi-Fi, GSM, or LoRa, allowing real-time data transmission to cloud platforms such as Firebase, Blynk, or ThingSpeak. These platforms store historical data, generate alerts, and provide analytics dashboards accessible via mobile apps or web interfaces. The dashboards display live sensreadings, system status, and incident history, enabling operators to monitor conditions remotely and make informed decisions. Additionally, the system can be configured to send SMS, email, or push notifications to designated personnel during emergencies.

The software architecture supports modular expansion, allowing the integration of additional sensors, actuators, and control logic as needed. It also supports machine learning algorithms for anomaly detection and predictive maintenance, enabling the system to learn from historical data and anticipate potential failures before they occur. Furthermore, the system can be linked with fire suppression systems, CCTV networks, and building management systems to create a unified safety infrastructure.

3.4 BLOCK DIAGRAM



3.5 Hardware components

- Node MCU
- DHT 11
- Gas Sensor
- Fire Sensor
- Buzzer
- LED

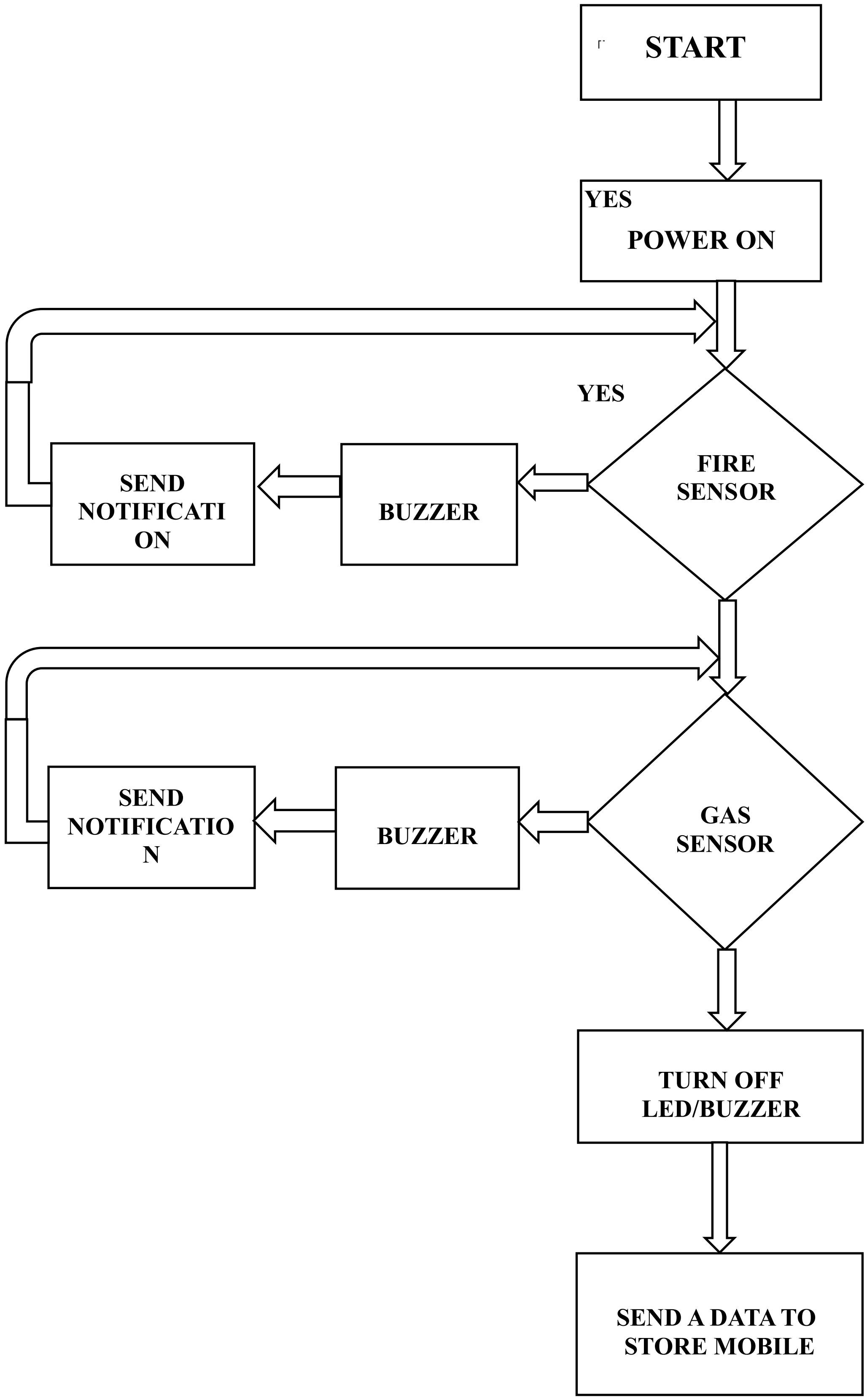
3.6 Software Components

- Arduino IDE
- Thingspeak

3.7 Technology Used

- IoT (Internet of Things)

3.8 WORKING PRINCIPLE



FLOW CHART EXPLANATION:

1.START:

- The system operation begins.
- Power supply and all components are initialized.
- It prepares sensors (fire, gas, temperature, etc.) for detection.
- Microcontroller (like Arduino or NodeMCU) starts executing the main program.
- The system enters into monitoring mode.

2. POWER ON:

- The system gets electrical power from the source.
- All sensors and modules are activated.
- Communication modules (like Wi-Fi or GSM) start working.
- The microcontroller checks if all sensors are connected properly.
- After successful setup, it moves to the sensing process.

3. FIRE SENSOR CHECK:

- The system reads the data from the fire sensor.
- The fire sensor detects flames or high temperature in the environment.
- It continuously monitors for sudden changes in temperature or light (depending on the sensor type).

If Fire Detected (YES):

- The buzzer is turned ON immediately to warn nearby people.
- LED indicators may also glow to give a visual warning.
- A notification or alert message is sent to the user's or control room's mobile through IoT (NodeMCU).
- The alert contains details like " Fire detected at location!".
- The system may also trigger an emergency response (like switching on a sprinkler or ventilation system).
- After sending the alert, the system continues monitoring.

If Fire Not Detected (NO):

- The system moves to check the gas sensor for gas leakage.

4. GAS SENSOR CHECK:

- The gas sensor (like MQ-2 or MQ-135) starts reading air quality.
- It detects harmful gases such as LPG, methane, carbon monoxide, etc.
- It compares the sensor value with a set threshold value.

If Gas Leakage Detected (YES):

- The buzzer starts ringing to alert workers or residents.
- LED light turns ON to show danger visually.
- A notification is sent to the user's mobile, mentioning "⚠ Gas leakage detected!".
- The system keeps sending data until the gas level becomes normal.
- It helps in immediate evacuation and safety measures.

If No Gas Leakage (NO):

- The system goes to the next step to turn off all warning devices.

5. TURN OFF LED / BUZZER:

- When no fire or gas leakage is found, the system goes into a safe state.
- The buzzer is turned OFF to stop the alarm sound.
- LED indicators are turned OFF to show that the environment is safe.
- This helps save energy and avoid false alarms.
- The system remains ready for the next detection cycle.

6. SEND DATA TO STORE / MOBILE:

- The system records or transmits the latest sensor data.
- Data includes temperature, gas levels, and status (safe or alert).

The data can be stored in:

- A cloud database
- Mobile app
- Local memory (SD card)
- Helps in keeping historical records of environmental conditions.
- The user can check the status remotely through IoT.
- Confirms that the system is active and functioning properly.

7. LOOP BACK (Continuous Monitoring):

- After sending data, the system loops back to monitor again.
- It continuously checks the fire and gas sensors.Ensures 24/7 safety monitoring.
- If any new hazard appears, the system reacts immediately.
- This makes the system automatic and smart, reducing the need for manual checking.

CHAPTER IV
HARDWARE COMPONENTS

4.1 NODEMCU ESP8266:

Overview and Architecture:

The NodeMCU (Node MicroController Unit) development board, featuring the ESP8266-12E Wi-Fi System-on-Chip (SoC), acts as the central control unit and the IoT gateway for the project. Its primary function is to bridge the physical world of sensors with the digital world of the cloud. The integrated Tensilica Xtensa LX106 32-bit RISC processor operates at an adjustable frequency of 80 MHz to 160 MHz, providing sufficient computational power to handle three critical, concurrent tasks:

- **Sensor Polling and Data Acquisition**: Reading continuous data from the MQ gas sensor (Analog) and digital signals from the DHT11 and IR Fire Sensor.
- **Alarm Logic Execution:** Processing sensor values against safety thresholds and instantaneously triggering local actuators (LED/Buzzer).
- **Wireless Communication**: Maintaining a connection to the industrial Wi-Fi network and pushing real-time data and critical alerts to the centralized monitoring platform using the integrated TCP/IP protocol stack.

Fig.4.1:Node MCU

Memory, Power, and ADC Constraints:

The ESP8266 module includes 4 MB of Flash Memory for storing the system firmware and configuration, and 64 KB of SRAM for runtime variable storage and managing network communication buffers. This memory allocation is essential for the complexity of the IoT application. In terms of power, the NodeMCU is typically powered by a 5V DC source (via Micro-USB or the VIN pin), which is internally regulated down to the chip's native 3.3V operating voltage and logic level. This 3.3V standard is a key interfacing constraint. Crucially, the NodeMCU possesses only one Analog-to-Digital Converter (ADC) pin, labeled A0. This pin is dedicated to reading the continuous analog voltage output from the MQ Gas Sensor. The ADC is 10-bit, converting the 0V−3.3V input range into a digital value of 0 to 1024. Extreme caution is taken during interfacing to ensure the analog input voltage never exceeds 3.3V to prevent irreversible damage to the chip.

Interfacing Pins and Project Roles

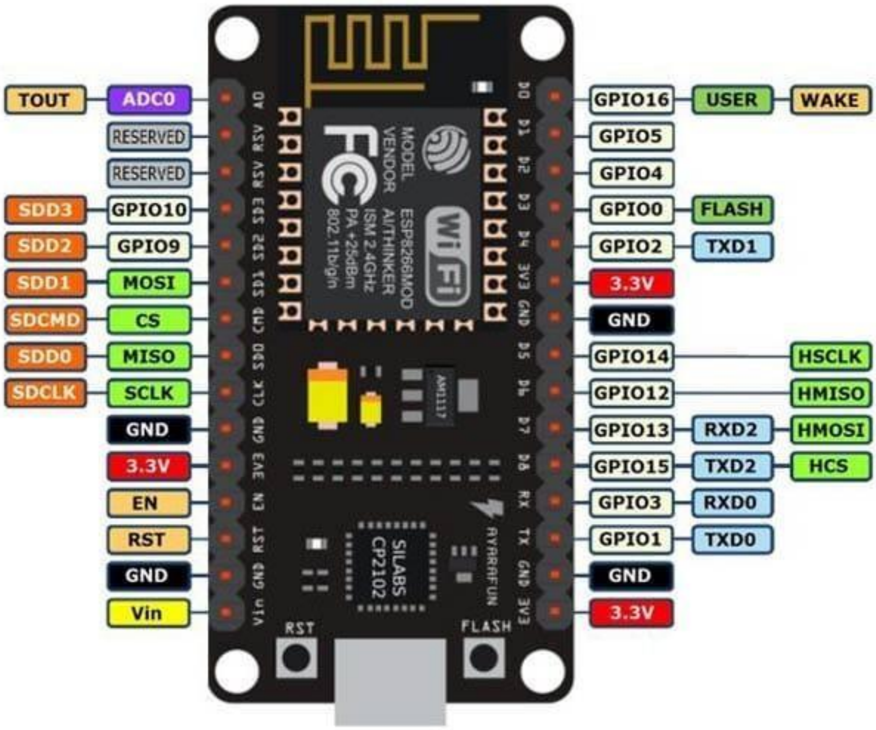


Fig.4.2: Pin Configuration Node MCU ESP8266

The NodeMCU uses GPIO pins for digital communication and control. In this project, specific pins are allocated for critical functions:

- **A0 (ADC):** Receives the analog concentration signal from the MQ Gas Sensor.
- **D1 (GPIO5):** Used for the single-wire digital data exchange with the DHT11 sensor.
- **D2 (GPIO4):** Monitors the digital threshold output (D0) from the IR Fire Sensor.
- **D6 (GPIO12) and D7 (GPIO13):** Used as output control pins for the Buzzer and LED visual indicator, respectivel

Pin Description

Power Pins

There are four power pins namely-a VIN pin and three3.3Vpins. The VIN pin can be used to directly supply ESP8266 and its components if you have a controlled 5V voltage source. The 3.3V pins are the output of the voltage board controller. These pins can be used to supply Power to external parts.

GND- Ground

Itis the ground pin of the ESP8266 Node MCU development board.

I2C Pins

These are used to integrate all types of I2C sensors and parameters in the project. Both I2C Master and I2C Slave are supported. The performance of the I2C optical connector can be systematically detected, and the clock Frequency is 100 kHz at maximum speed. It should be noted that the frequency of the I2C clock should be greater than the frequency of the slowest clock of the slave device.

GPIO Pins

The ESP8266 Node MCU has17GPIO anchors that can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light, and Button respectively. Each GPIO digitally- Enabled can be adjusted to internal drag or drop or set to high intensity. When set as input, it can also Be set to Edge-trigger or level-trigger to produce CPU interference .

ADC channel

Node MCU is embedded 10-bit with SARADC accuracy. These two functions can be performed using the ADC viz. VDD3P3 pin power supply and TOUT pin power input. However, they cannot be used simultaneously

UART

The ESP8266 Node MCU has 2 UART domains, namely UART0 and UART1, which offer Different connections (RS232 and RS485), and can communicate up to 4.5Mbps. UART 0 pins (TXD0, RXD0, RST0 & CTS0) can be used for communication. Supports fluid control.

SPI Pins

ESP8266 incorporates two SPIs (SPI and HSPI) into slave and master modes. These SPIs also support the following SPI features: 4-time modes for SPI format transfer Up to 80MHz with split clocks of 80 MHz Up to 64-ByteFIFO

SDIO Pins

ESP8266 incorporates the Secure Digital Input /Output Interface (SDIO) which is used to connect directly to SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins

The board has 4 Pulse Width Modulation (PWM) channels. PWM output can be programmed and used to drive digital engines and LEDs. The frequency range of PWM ranges from1000µs to10000µs, eg between100Hz and1 kHz.

Control Pins

These anchors are used to control ESP8266. These anchors include Chip Enable pin (EN), Reset pin (RST), and WAKE pin.

ENP in (Enable)

The ESP8266 chip is enabled when the EN pin is pulled INSIDE. When pulled LOW the chip works at low power.

RST Pin (Reset)

The RST pin is used to reset the ESP8266 chip

Wake Pin

The use of Wake pin is used to wake up the chip from a deep sleep.

Features

1Model: ESP8266-12E.

Wireless Standard: 802.11b/g/n.

Frequency range:2.4GHz-2.5GHz (2400M-2483.5M)

Wi-Fi mode: Station/ Soft AP/Soft AP+ station.

Stack: Integrated TCP/IP.

Output power:19.5dBmin802.11bmode.

Data interface: UART/HSPI/I2C/I2S/Ir.

Remote Control GPIO /PWM.

4.2 GAS SENSOR - Chemical Leakage Detector

The MQ-X Gas Sensor (specifying the exact model, such as MQ−2 or MQ−135, in the final report is recommended) is the chemical transducer responsible for quantifying harmful gas concentrations in the air.Operational Principle: Chemiresistance and

Activation The sensor's fundamental operation relies on chemiresistance using a specialized semiconductor material, typically a layer of Tin Dioxide (SnO_2).

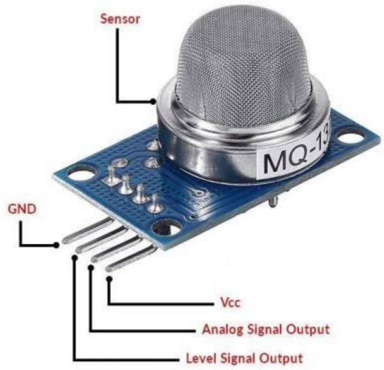


Fig.4.3:Gas Sensor

Thermal Activation: The SnO_2 layer is deposited on a ceramic tube containing an internal heating element (heater coil), usually powered by 5V. The heater's role is to maintain the SnO_2 at a high temperature(200∘C-300∘C). This thermal activation is necessary because the SnO_2 only becomes chemically reactive and sensitive to gases at this elevated temperature.

Sensing Mechanism: Baseline (Clean Air): In the absence of target gases, oxygen molecules (O_2) from the air are adsorbed onto the hot SnO_2 surface. These adsorbed oxygen ions trap electrons from the SnO_2 Conduction band, forming a high potential barrier and resulting in a High electrical resistance (Rs) state.

Detection (Gas Presence): When harmful gas molecules (e.g., LPG, Methane, CO) come into contact with the sensor, they react with the adsorbed oxygen ions. This reaction releases the trapped electrons back into the SnO_2 bulk. The influx of free electrons lowers the potential barrier, causing a significant decrease in the sensor's electrical resistance (Rs).

Signal Output and Quantitative PPM Measurement:

The MQ sensor module converts the change in Rs into a measurable voltage signal.

Analog Output (A0): The sensor is integrated into a voltage divider with a load resistor (RL). As Rs decreases due to gas presence, the voltage across RL(Vout) increases. This continuous analog signal is fed directly to the NodeMCU A0 pin.

PPM Conversion: For industrial monitoring, the raw ADC reading must be mapped to Parts Per Million (PPM), the standard unit for gas concentration. This conversion is achieved in the NodeMCU firmware using the sensor's Characteristic equation: R0/Rs =A.(PPM) −B Where R0 is the sensor resistance in clean air, and A and B are constants derived from the sensor datasheet's concentration curves. This calibrated PPM value is used to set precise safety limits (e.g., Lower Explosive Limit (LEL)). **Module Features and Digital Outpu**t :

The commercial MQ sensor module also includes a Digital Output (D0). This output is generated by an onboard LM393 comparator that compares the V out to a useradjustable reference voltage set by a potentiometer. The D0 pin provides an instantaneous HIGH/LOW signal when the gas concentration crosses the set digital threshold, offering a reliable, pre-filtered trigger for the alarm system.

4.3 IR Flame Sensor

Fire and Combustion Detection The IR Flame Sensor is a dedicated optical detection unit that provides an independent, rapid alert mechanism for the presence of an open flame or fire, which is a common consequence of gas leakage.

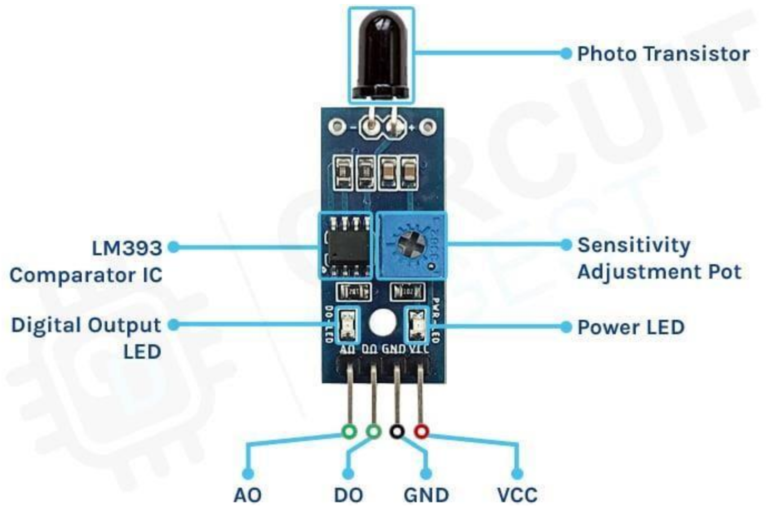


Fig.4.4:Fire Sensor

Flames do not emit uniform light; they have a distinct spectral signature. The IR Flame Sensor is designed to be highly sensitive to the Infrared (IR) region of the electromagnetic spectrum, specifically wavelengths between 760 nm and 1100 nm. This range corresponds to the thermal IR radiation produced by the rapid combustion process.

- **Sensing Element**: The primary component is an IR Photodiode which acts as a transducer, converting the intensity of the received IR light into a proportional electrical current.

- **Rapid Response**: Unlike slower smoke detectors, the IR photodiode provides an instantaneous optical response upon sighting a flame, making it ideal for safety-critical applications where milliseconds matter. The sensor requires a direct line-of-sight to the fire source for effective operation.

Signal Processing and Digital Thresholding :The sensor module integrates an LM393 Voltage Comparator to process the photodiode's signal and produce a clear digital output.

- **Comparator Function**: The amplified signal from the IR photodiode is fed into one input of the comparator. A reference voltage, which determines the sensitivity threshold, is set on the other input using an onboard potentiometer.
- **Output Logic:** When the detected IR intensity (signal voltage) exceeds the reference voltage, the LM393's output instantly switches state, providing a clear digital signal. The output logic is typically configured to switch to LOW when a flame is detected and remain HIGH in a normal state.
- **Interfacing:** The digital output pin (D0) of the module is connected to a GPIO pin on the NodeMCU (e.g., D2). The system firmware continuously polls this pin, and a LOW reading triggers the fire alarm sequence.

Advantages for Safety The inclusion of the IR flame sensor provides sensor redundancy and complementary detection. It confirms the presence of combustion, which, when coupled with a high PPM reading from the gas sensor, elevates the alarm status to a maximum urgency level, minimizing false positives and accelerating emergency response.

4.4 DHT11 - Temperature and Humidity Sensor

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness

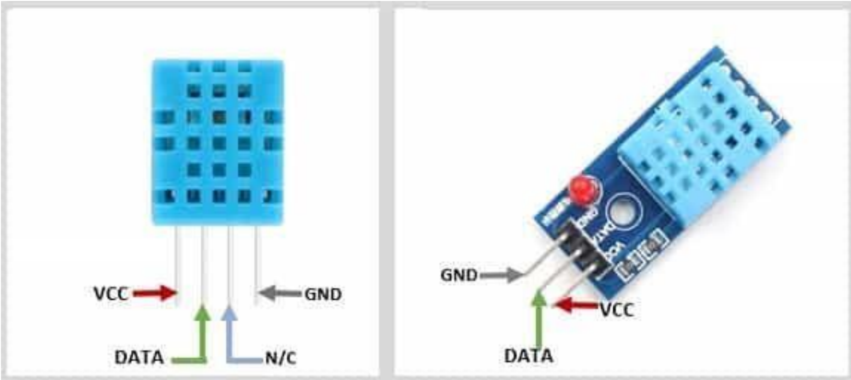


Fig.4.5:DHT.Sensor

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4pin single row pin package. It is convenient to connect and special packages can be provided according to users' request

Dual - Sensing Mechanism

- The DHT11 combines two distinct sensing principles within a single unit: Humidity **Sensing (Resistive Type):** This component consists of two electrodes with a moisture-holding substrate. As the substrate absorbs or releases water vapor based on ambient humidity, its electrical resistance changes. This resistance change is measured and mapped to Relative Humidity (RH).
- **Temperature Sensing (NTC Thermistor):** An integrated Negative Temperature Coefficient (NTC) thermistor measures the air temperature. The thermistor's inherent property is that its resistance decreases as the temperature increases.

Single-Wire Digital Protocol and Data Integrity:

The DHT11 employs a proprietary single-wire digital serial interface. This protocol is the most critical aspect of its operation:

- **Simplified Wiring:** Only one data pin is required for bidirectional communication between the sensor and the NodeMCU.

- **Timing Requirement:** The protocol requires precise timing control from the microcontroller to initiate and read the data stream. The NodeMCU sends a start signal, and the sensor responds with a 40-bit data frame.
- **Data Structure**: The 40 bits are structured as: 8 bits Humidity Integer, 8 bits Humidity Decimal, 8 bits Temperature Integer, 8 bits Temperature Decimal, and 8 bits Checksum.
- **Checksum Validation**: The final 8 bits (Checksum) are the sum of the preceding four data bytes. The NodeMCU calculates the sum and compares it with the received Checksum. If they match, the data is considered valid, ensuring data integrity against transmission errors.

Specifications and Project Context:

The DHT11 provides data within the range of 0∘C to 50∘C (±2∘C accuracy) and 20% to 90% RH (±5% RH accuracy). For the project, a sudden temperature spike captured by the DHT11 serves as a tertiary alarm indicator, supporting the primary gas and secondary fire detection.

Features:

1. Temperature and Humidity Measurement:
   - Both sensors measure ambient temperature and relative humidity.
   - DHT11 is suitable for basic applications, while DHT22 offers higher accuracy and a wider measurement range.
2. Digital Output:
   - Outputs calibrated digital signals, eliminating the need for analog-to-digital conversion.
   - Data is transmitted via a single-wire serial interface.

3. Factory Calibrated: o Both sensors are pre-calibrated, ensuring plug-and-play functionality with minimal setup.

4. Low Cost and Compact Size: o DHT11 is ultra-low-cost and smaller, ideal for hobbyist projects.

- DHT22 is slightly more expensive but still compact

5.Single-Wire Communication:

- Uses a proprietary single-wire protocol for communication with microcontrollers, simplifying wiring.

Specifications

- **Operating Voltage**: 3.3V to 5V DC.
- **Current Consumption:**
  - Maximum during conversion: 2.5mA.
  - Standby: 0.5mA.
- **Humidity Measurement:**
  - Range: 20% to 90% RH.
  - Accuracy: ±5.0% RH.
  - Resolution: 1% RH.
- **Temperature Measurement:**
  - Range: 0°C to 50°C.
  - Accuracy: ±2.0°C.

4.5 Output Actuators: LED and Buzzer

Local HMI The LED and Buzzer form the essential local Human-Machine Interface (HMI), providing immediate, non-network dependent warnings to personnel.

Buzzer (Active Type):

Auditory Alert The Active Buzzer is used to generate a loud, attention-demanding auditory alarm.

Fig.4.6:Buzzer

Principle: Unlike a passive buzzer, the active type contains an internal electronic oscillation circuit (oscillator and driver). It requires only a stable DC voltage across its terminals to produce a continuous, high-pitch tone.

Interfacing: The buzzer is connected to a GPIO pin on the NodeMCU (e.g., D6). The alarm is triggered simply by setting the GPIO pin to HIGH (3.3V), simplifying the alarm code structure.

Project Role: Provides the first, most immediate warning of a hazard, essential for rapid safety response and evacuation before cloud notifications are processed. 5.5.2

LED (Light Emitting Diode)

Visual Status The LED serves as the primary visual status indicator for the system. **Principle:** A semiconductor junction device that emits light (electroluminescence) when a current passes through it in the forward direction.

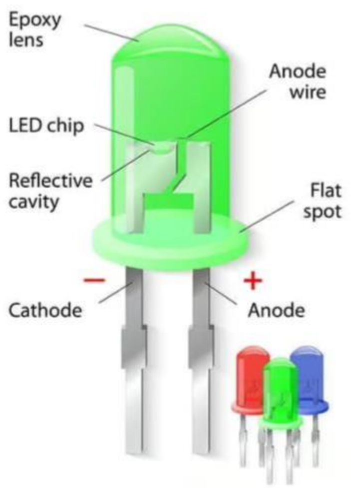


Fig.4.7:LED

Safety and Interfacing: The LED must be connected in series with a current-limiting resistor (typically 220Ω to 330Ω). This prevents the LED from drawing excessive current (overcurrent) from the GPIO pin, which could damage both the LED and the NodeMCU chip.

Project Role: The system uses LED colors to represent operational states: a GREEN LED confirms the system is powered and monitoring (safe state), while a RED LED is activated and often flashed by the Node MUC.

CHAPTER V
SOFTWRAE

5.1 Introduction to Arduino IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are

Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, nurning LED on/off. connect to the cloud and many other actions.

You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).

Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.

Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

Finally, Arduino provides a standard form factor that breaks the functions of the micro- controller into a more accessible package.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

5.1.1 Arduino data types:

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

The following table provides all the data types that you will use during Arduino programming.

Void:

The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called.

Example:

```
Void Loop ()
{
// rest of the code
}
```

Boolean: A Boolean holds one of two values, true or false. Each Boolean variable occupies one byte of memory.

Char: A data type that takes up one byte of memory that stores a character value. Character literals are written in single quotes like this: 'A' and for multiple characters, strings use double quotes: "ABC"

However, characters are stored as numbers. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic operations on characters, in which the ASCII value of the character is used. For example, "A'+ 1 has the value 66, since the ASCII value of the capital letter A is 65.

Unsigned char is an unsigned data type that occupies one byte of memory. The unsigned char data type encodes numbers from 0 to 255.

Byte: A byte stores an 8-bit unsigned number, from 0 to 255.

Int: Integers are the primary data-type for number storage, int stores a 16-bit (2-byte) value. This yields a range of 32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15)-1).

The int size varies from board to board. On the Arduino Due, for example, an int stores a 32-bit (4-byte) value

Unsigned int: Unsigned ints (unsigned integers) are the same as int in the way that they store a 2byte value. Instead of storing negative numbers, however, they only store positive values, yielding a useful range of 0 to 65,535 (2016) 1). The Due stores a 4 byte (32-bit) value, ranging from 0 to 4,294,967,295 (2^32-1).

Word: On the Uno and other ATMEGA based boards, a word stores a 16-bit unsigned number. On the Due and Zero, it stores a 32-bit unsigned number.

Long: Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from 2,147,483,648 to 2,147,483,647,

Unsigned long: Unsigned long variables are extended size variables for number storage and store 32 bits (4 bytes). Unlike standard longs, unsigned longs will not store negative numbers, making their range from 0 to 4,294,967,295 (2-32-1).

Short: A short is a 16-bit data-type. On all Arduinos (ATMega and ARM based), a short stores a 16-bit (2-byte) value. This yields a range of-32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15)-1).

Float: Data type for floating-point number is a number that has a decimal point. Floating-point numbers are often used to approximate the analog and continuous values because they have greater resolution than integers.

Double: On the Uno and other ATMEGA based boards, Double precision floating-point number occupies four bytes. That is, the double implementation is exactly the same as the float, with no gain in precision. On the Arduino Due, doubles have 8-byte (64 bit) precision.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1: First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanoye. Nano, Arduino Mega2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Step 2: Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3: Power up your board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable.

The green power LED (labeled PWR) should glow.

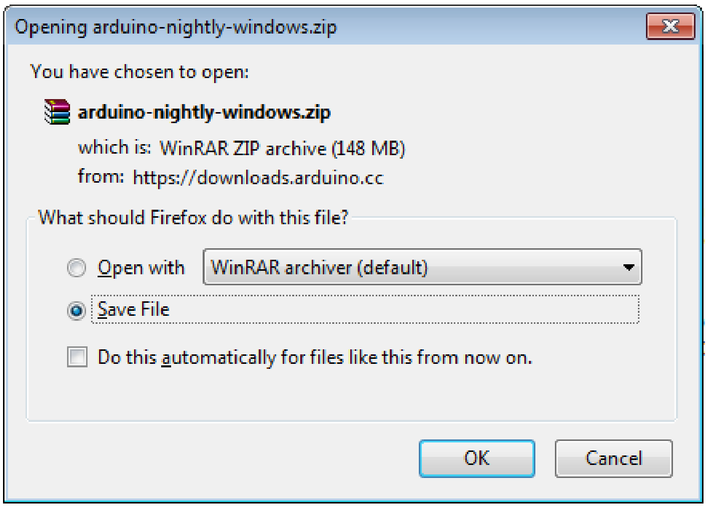


Figure 5.1: Arduino ide for node mcu

Step 4: Launch Arduino IDE.

SMART OFFICE AUTOMATION USING ESP

After your Arduino IDE software is downloaded, you need to unzip the folder.

Inside the folder, you can find the application icon with an infinity label (application.exe). Double click the icon to start the IDE.

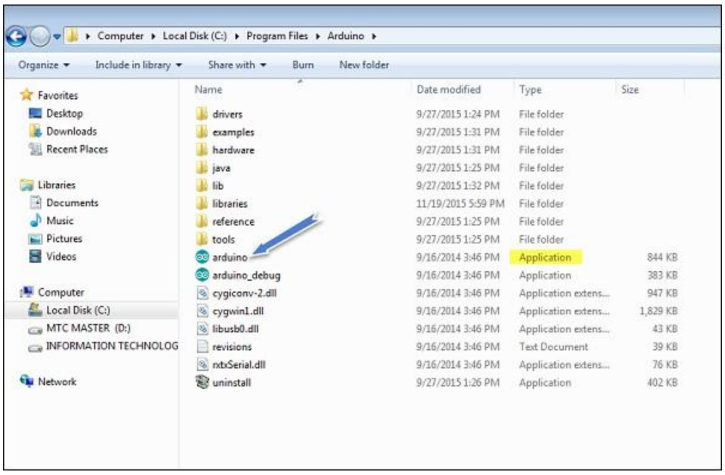


Figure 5.2: Install arduino

Step 5: Open your first project.

Once the software starts, you have two options:

Create a new project.

Open an existing project example.

To create a new project, select File-à New, To open

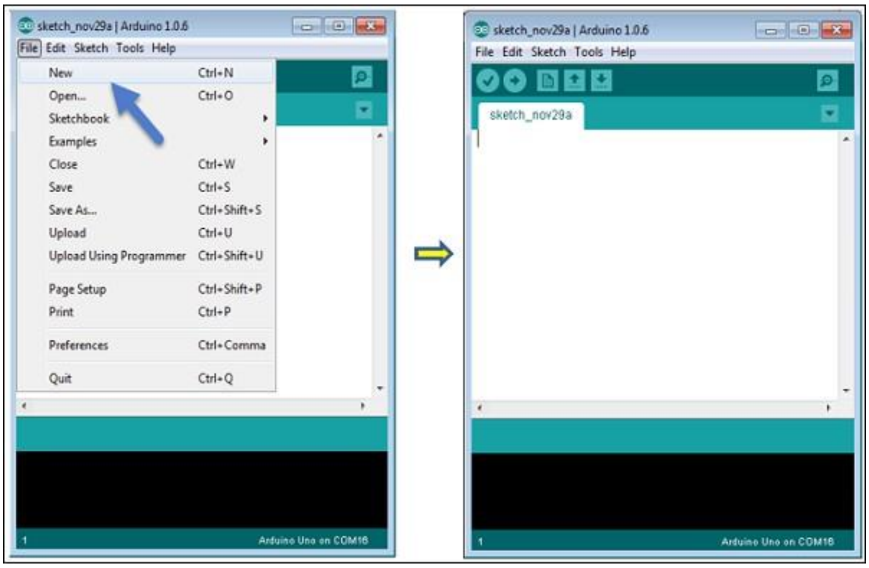


Figure5.3: Creating new project

To open an existing project example, select File Example Basics -> Blink.

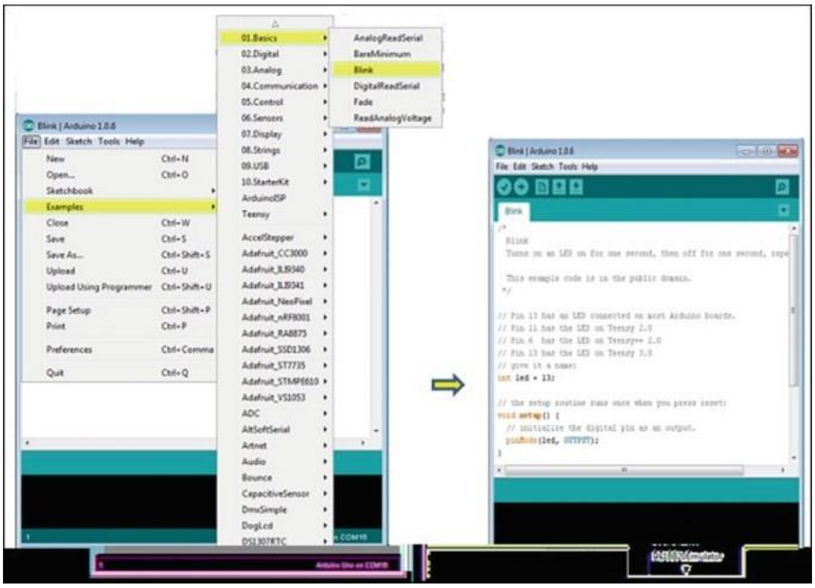


Figure 5.4: Installed boards under tools

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with

Sometimes delay. You can select any other example from the list.

Step 6: Select your Arduino board to avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools-à Board and select your board SMART OFFICE AUTOMATION USING ESP

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using

Step 7: Select your serial port.

Select the serial device of the Arduino board. Go to Tools Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports), To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port

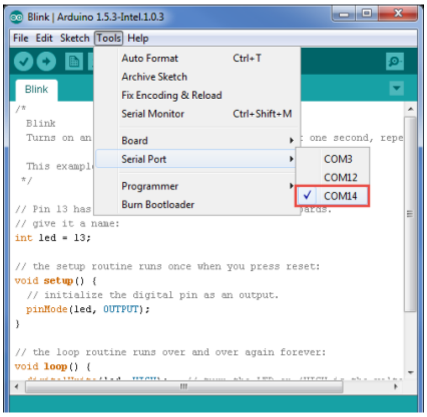


Figure 5.5: Selecting serial port

Step 8: Upload the program to your board. Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

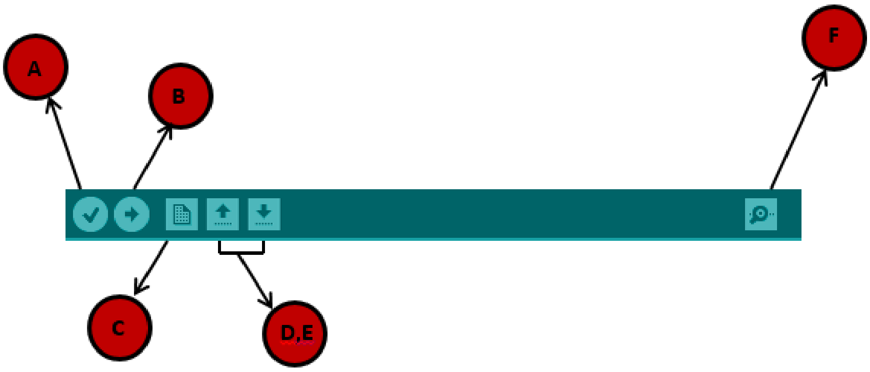


Figure5.6: upload the program

A- Used to check if there is any compilation error.

B- Used to upload a program to the Arduino board.

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketch.

E- Used to save your sketch.

5.2 Thing speak

Thing Speak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. You can send data to Thing Speak from your devices, create instant visualization of live data, and send alerts. It provides instant visualizations of data posted by your devices to Thing Speak

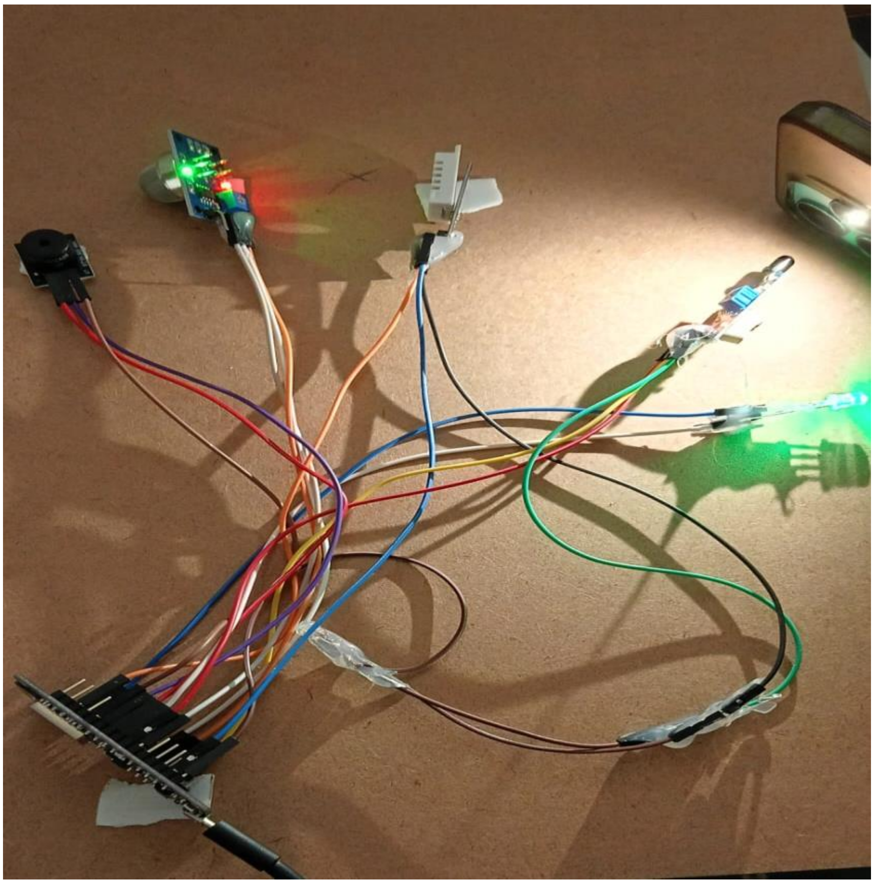
Figure 5.7 :Thing speak interface

Thing Speak is an IoT platform that allows users to collect, analyze, and visualize data from sensors and other devices. A Thing Speak channel can be used to monitor garbage levels in real- time, enabling efficient waste management.

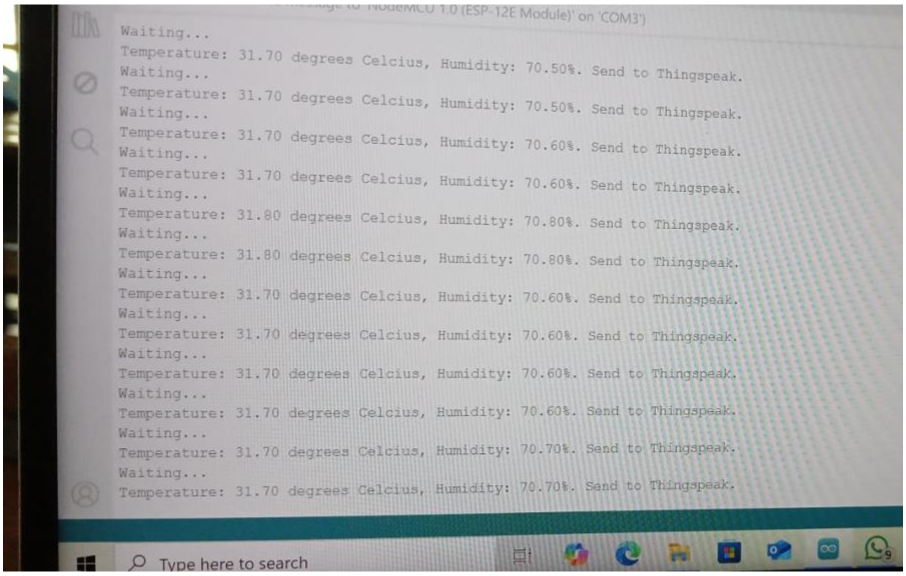
CHAPTER VI
RESULTS AND DISCUSSIONS

6.1 RESULTS

The implementation of the smart industrial hazardous gas leakage detection and automated safety response system yielded promising results across multiple testing scenarios. The system was deployed in a controlled industrial-like environment using MQ-series gas sensors (MQ-2, MQ-6, MQ-135) interfaced with NodeMCU ESP8266 microcontroller. The sensors successfully detected the presence of gases such as methane, LPG, and carbon monoxide with high sensitivity and accuracy.



During simulation tests, the system demonstrated rapid response times, with gas detection and alert generation occurring within 2–3 seconds of exposure. Upon detecting gas concentrations above the safety threshold, the system automatically triggered alarms, activated exhaust fans, and sent real-time notifications to connected mobile devices via the Blynk IoT platform. The automated shut-off mechanism for simulated machinery was also successfully engaged, confirming the system’s ability to initiate safety protocols without human intervention.



Data was logged continuously to the cloud (Thing Speak), allowing for real-time monitoring and historical analysis. The dashboard displayed live sensor readings and system status, which helped operators visualize trends and identify potential risks. The system maintained stable performance over extended periods, with minimal false positives and consistent connectivity.

6.2 DISCUSSIONS

A smart industrial hazardous gas leakage detection and automated safety response system is a critical innovation aimed at enhancing workplace safety and environmental protection in industrial settings. This project integrates advanced sensors, IoT technology, and AI-driven analytics to continuously monitor the presence of toxic or flammable gases such as methane, ammonia, or carbon monoxide. Upon detecting abnormal concentrations, the system triggers immediate alerts and activates automated safety protocols—such as ventilation control, area isolation, and emergency shutdowns—to prevent accidents and minimize exposure risks. The system can also notify personnel and emergency services in real time, ensuring rapid response and evacuation if necessary. By combining real-time data acquisition with intelligent decision-making, this solution not only safeguards human lives but also helps industries comply with safety regulations and reduce operational downtime.

Expanding further, the smart industrial hazardous gas leakage detection and automated safety response system represents a transformative leap in industrial safety management. Traditional gas detection methods often rely on manual monitoring or periodic inspections, which can delay response times and increase the risk of exposure. In contrast, this intelligent system operates continuously and autonomously, using a network of strategically placed sensors to detect even minute concentrations of harmful gases. These sensors are calibrated to recognize specific gas signatures and are integrated with microcontrollers and wireless communication modules to transmit data in real time. The system's software component analyzes this data using predefined thresholds and machine learning models to identify potential leaks or unsafe conditions. Upon detection, it initiates a cascade of safety responses—activating alarms, shutting down equipment, sealing off affected zones, and alerting personnel through mobile and desktop notifications. This proactive approach not only protects workers and assets but also ensures compliance with stringent industrial safety standards. Moreover, the system's ability to log incidents and generate analytical reports supports continuous improvement and risk assessment. As industries move toward smarter, more connected operations, such systems are becoming essential for sustainable and resilient infrastructure.

CHAPTER VII

ADVANTAGES AND APPLIACTIONS

7.1 ADVANTAGES

Enhanced Safety

- **Immediate Leak Detection**: Identifies hazardous gas leaks in real time, reducing the risk of exposure.
- **Automated Emergency Response**: Activates alarms, ventilation, and shutdown protocols instantly without human intervention.
- **Worker Protection**: Minimizes health risks for employees working in high-risk environments.

Operational Efficiency

- **Reduced Downtime**: Early detection prevents equipment damage and production halts.
- **Predictive Maintenance**: Data analytics help forecast potential failures before they occur.
- **Streamlined Compliance**: Automatically logs incidents and safety actions for regulatory audits.

Environmental Protection

- **Emission Control**: Prevents uncontrolled release of toxic gases into the atmosphere.
- **Sustainable Operations**: Supports eco-friendly practices by minimizing industrial pollution.

Cost Savings

- **Lower Insurance Premiums**: Improved safety can lead to reduced liability and insurance costs.
- **Minimized Damage**: Prevents costly repairs and replacements due to gas-related accidents.
- **Efficient Resource Use**: Optimizes energy and safety system usage through smart automation.

Smart Integration

- **IoT & Cloud Connectivity**: Enables remote monitoring and control from anywhere.

- **Scalability**: Can be expanded across multiple zones or facilities.
- **Real-Time Alerts**: Sends notifications to mobile devices, control rooms, and emergency teams.

7.2 APPLICATIONS

Industrial Safety & Monitoring

- **Chemical Plants**: Detect leaks of toxic gases like chlorine, ammonia, or hydrogen sulfidic to prevent accidents.
- **Oil & Gas Refineries**: Monitor flammable gases such as methane and propane to avoid explosions and fires.
- **Pharmaceutical Manufacturing**: Ensure clean air standards and prevent contamination from volatile organic compounds.

Underground & Confined Spaces

- **Mining Operations**: Identify buildup of dangerous gases like methane or carbon monoxide in tunnels and shafts.
- **Sewage Treatment Facilities**: Detect hydrogen sulfide and other harmful emissions in enclosed areas.

Smart Buildings & Infrastructure

- **Industrial Warehouses**: Integrate with building management systems for automated ventilation and safety alerts.
- **Laboratories & Research Centre's**: Maintain safe working conditions by detecting chemical vapors and gas leaks.

Transportation & Logistics

- **Hazardous Material Transport**: Monitor gas levels in cargo containers and tanker trucks during transit.
- **Fuel Stations & Depots**: Detect leaks from fuel storage and dispensing systems.

Environmental & Public Safety

- **Air Quality Monitoring Stations**: Track industrial emissions and support environmental compliance.
- **Disaster Response Units**: Deploy portable systems in emergency zones to assess gas hazards quickly.

CHAPTER VIII

CONCLUSION AND SCOPE

8.1 CONCLUSION

The Smart Industrial Hazardous Gas Leakage Detection and Automated Safety Response System stands as a cornerstone of modern industrial safety and automation. In an era where industries are increasingly adopting digital transformation and smart technologies, this system offers a proactive, intelligent solution to one of the most critical challenges—gas leak hazards. By combining real-time sensing, IoT connectivity, and automated control mechanisms, it ensures that even the slightest trace of toxic or flammable gases is detected and addressed instantly. This not only protects human lives and valuable assets but also reinforces environmental responsibility and regulatory compliance.

The system's modular design and scalability make it adaptable across a wide range of industrial sectors—from chemical plants and oil refineries to underground mining and smart buildings. Its ability to integrate with existing infrastructure and provide remote monitoring through cloud platforms enhances operational transparency and decision-making. Furthermore, the data analytics and reporting capabilities support continuous improvement, predictive maintenance, and strategic planning.

In essence, this project is not just a safety tool—it's a comprehensive risk management framework that embodies the principles of Industry 4.0. It empowers industries to move beyond reactive safety measures toward intelligent, automated, and resilient safety ecosystems. As industrial environments grow more complex and demanding, the deployment of such smart systems will be pivotal in shaping a safer, smarter, and more sustainable future.

8.2 FUTURE SCOPE

The future scope of the **Smart Industrial Hazardous Gas Leakage Detection and Automated Safety Response System** is vast and promising, especially as industries continue to embrace digital transformation and prioritize safety and sustainability. With advancements in sensor technology, artificial intelligence, and the Internet of Things (IoT), this system can evolve into a more intelligent, adaptive, and predictive safety infrastructure.

In the near future, integration with **edge computing** will allow faster local decision-making, reducing latency in emergency responses. The use of **machine learning algorithms** can enhance the system's ability to predict potential gas leaks based on historical data, environmental conditions, and equipment behavior, enabling preventive maintenance and reducing unplanned downtimes. Additionally, the system can be expanded to support **multi-site monitoring**, allowing centralized control rooms to oversee safety across multiple facilities in real time.

Another significant development lies in **drone-based gas detection**, where autonomous drones equipped with gas sensors can patrol large or hard-to-reach industrial zones, providing aerial surveillance and leak detection. Integration with **augmented reality (AR)** and **wearable devices** can further empower workers by providing real-time gas concentration data and safety alerts directly through smart helmets or glasses.

Moreover, as environmental regulations become stricter, the system can be enhanced to generate **automated compliance reports**, helping industries meet legal standards and avoid penalties. With the rise of **smart cities and Industry 5.0**, this system could also be adapted for urban infrastructure, public transportation hubs, and residential complexes to ensure broader public safety.

In essence, the future of this system lies in its ability to become more autonomous, intelligent, and interconnected—shaping a safer, more responsive, and environmentally responsible industrial landscape.

RERERENCES

1.Gas Detection and Identification Using Multimodal Artificial Intelligence Based Sensor Fusion

- Parag Narkhede et al. (2021)
- Presents a multimodal sensor fusion method (gas sensors + thermal camera) for detection/identification of gases. Achieved ~96% accuracy for fused data compared to single sensor modalities. arXiv
- Good for: detection/identification layer; combining modalities improves robustness in industrial settings.

2.Infrared Imaging Detection for Hazardous Gas Leakage Using Background Information and Improved YOLO Networks

- Jue Wang et al. (2024)
- Proposes an infrared-imaging approach + improved YOLO model (BBG-FA-YOLO) to detect gas plumes in outdoor/industrial settings. MDPI
- Good for: vision-based detection component; especially for visible plume/leak detection in large industrial sites.

3.Design and Development of IoT based Industrial Gas Leakage Monitoring System

- Mourya Sirapu (2022)
- Focused on an IoT system for industrial gas leak monitoring: sensor inputs, microcontroller, Wi-Fi alerts & cloud monitoring. IJRASET
- Good for: IoT architecture; remote monitoring, alerting & data visualization.

4.Mobile Detection and Alarming Systems for Hazardous Gases and Volatile Chemicals in Laboratories and Industrial Locations

- Mohammed Faeik Ruzaij Al-Okby et al. (2021)
- Review article summarizing mobile detection/alarming systems, sensor technologies (MOX, MEMS, etc.), comms (Bluetooth, Wi-Fi, LoRa) in lab/industrial sites. MDPI
- Good for: survey/background; selecting sensor types and communication tech; understanding state-of-the-art.

APPENDIX

```
#include <DHT.h>  // Including library for dht

#include <ESP8266WiFi.h>

String apiKey = "I3JG2AQ6EOWV4B40";     //  Enter your Write API key from
ThingSpeak

const char *ssid =  "batch13";     // replace with your wifi ssid and wpa2 key
const char *pass =  "batch1313";
const char* server = "api.thingspeak.com";
int val = 0; // variable for reading the IR pin status
int fval =0 ;

#define DHTPIN 4 //pin where the dht22 is connected D2
#define LED 2 // LED pin to D4
#define IR 13 // GAS pin to D7
#define FIRE_LED 14      // Fire alert LED connected to D5 (GPIO14)
#define FIR 12 // D6

DHT dht(DHTPIN, DHT22);

WiFiClient client;

void setup()
{
    Serial.begin(115200);
    delay(10);
    dht.begin();

    pinMode(LED, OUTPUT);    // LED pin as output.
    pinMode(FIRE_LED, OUTPUT);    // LED pin as output.
```

```
    digitalWrite(LED, HIGH);
    digitalWrite(FIRE_LED, LOW);
    pinMode(IR, INPUT); // declare Infrared sensor as input
    pinMode(FIR, INPUT);

    Serial.println("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, pass);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

}

void loop()
{

    float h = dht.readHumidity();
    float t = dht.readTemperature();
    val = digitalRead(IR); // read IR input value
    fval = digitalRead(FIR); // read IR input value



        if (isnan(h) || isnan(t))
          {
            Serial.println("Failed to read from DHT sensor!");
             return;
```

```
        }

          if (client.connect(server,80))   //   "184.106.153.149" or
api.thingspeak.com
         {

             String postStr = apiKey;
             postStr +="&field1=";
             postStr += String(t);
             postStr +="&field2=";
             postStr += String(h);
             postStr += "\r\n\r\n";

             client.print("POST /update HTTP/1.1\n");
             client.print("Host: api.thingspeak.com\n");
             client.print("Connection: close\n");
             client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
             client.print("Content-Type: application/x-www-form-
urlencoded\n");
             client.print("Content-Length: ");
             client.print(postStr.length());
             client.print("\n\n");
             client.print(postStr);

             Serial.print("Temperature: ");
             Serial.print(t);
             Serial.print(" degrees Celcius, Humidity: ");
             Serial.print(h);
             Serial.println("%. Send to Thingspeak.");

             if (val== 1)
             {
              digitalWrite(LED, LOW); // ON
```

```
                     digitalWrite(FIRE_LED, LOW); // ON
                    }
                   else
                   {
                    digitalWrite(LED, HIGH); // OFF
                    digitalWrite(FIRE_LED, HIGH); // OFF
                   }
                   delay(50);


                  if (fval== 1)
                   {
                    digitalWrite(FIRE_LED, LOW); // ON
                    }
                   else
                   {
                    digitalWrite(FIRE_LED, HIGH); // OFF
                   }
                   delay(50);
              }
      client.stop();

      Serial.println("Waiting...");

  // thingspeak needs minimum 15 sec delay between updates
  delay(1000);
}
```