Department of Electronics and Communication Engineering

National Institute of Technology Rourkela, Odisha 769008

Report

On

GAS KAVACH

(Underground Gas pipeline Fault Detection and Alert system)

By

Satyaprakash Rath (122EI0621)

Abstract

Underground gas pipelines, while essential for energy distribution, pose significant safety hazards if leaks or anomalies remain undetected. This project proposes an IoT-based real-time underground gas pipeline fault detection and alert system that leverages embedded technology and wireless communication for proactive safety management. The system integrates sensors (MQ-6 for gas, DHT11 for temperature/humidity, and a flow sensor) with microcontrollers (Arduino Uno and Node MCU ESP8266) to enable continuous monitoring. When faults are detected, the system triggers a buzzer, sends SMS alerts with GPS-based location via the SIM800L GSM module, and transmits live data to a Thing Speak IoT dashboard. Additionally, a relay-controlled solenoid valve enables automatic shutoff of gas supply to prevent escalation. The system offers a low-cost, scalable, and real-time solution for enhancing pipeline safety and enabling data-driven decision-making.

Introduction

With growing urban infrastructure and dependence on natural gas, underground gas pipelines require robust fault monitoring systems to avoid environmental hazards and ensure public safety. Traditional systems, dependent on manual inspections or isolated sensors, often fall short in providing real-time alerts or precise fault localization.

To bridge this gap, this project presents a smart IoT-based monitoring system using the Node MCU ESP8266, a Wi-Fi-enabled microcontroller that ensures wireless data transmission and remote access. The system employs an MQ-6 gas sensor, DHT11 temperature/humidity sensor, and a flow sensor to detect gas leaks, abnormal temperature rises, and flow anomalies. Data is first processed by an Arduino Uno, then relayed to the ESP8266, which uploads it to the Thing Speak cloud platform for visualization and historical trend analysis.

When a fault is detected, the system sounds an on-site buzzer, sends an SMS alert using the SIM800L GSM module, and includes GPS coordinates from the NEO-6M module for accurate fault localization. Additionally, it automatically shuts off the gas supply via a relay and solenoid valve, preventing further risks. By combining real-time monitoring, automated alerts, and remote visualization, the system ensures a safer and smarter approach to gas pipeline management

Problem Statement

In India, underground gas pipelines are a critical component of urban and industrial infrastructure. However, they are prone to faults such as gas leaks, temperature surges, and flow anomalies, which pose severe threats to human safety, environmental sustainability, and infrastructure integrity. According to reports from various safety audits and news sources, numerous incidents involving gas leaks have led to fatal explosions, environmental pollution, and economic losses, often **exacerbated by delayed fault detection and inadequate response mechanisms.**

Current systems primarily rely on manual inspections, basic alarms, or wired sensor networks, which are not only slow and labor-intensive but also lack real-time responsiveness. Furthermore, many of these setups do not support remote monitoring, precise location tracking, or automated emergency actions. There is a critical need for an intelligent, cost-effective, and scalable solution that can continuously monitor underground pipelines and provide immediate alerts in case of anomalies.

Proposed Solution & Novelty

This project proposes an IoT-based real-time fault detection and alert system for underground gas pipelines, combining multiple sensors and wireless communication for comprehensive monitoring. The system utilizes an MQ-6 sensor for gas leak detection, a DHT11 sensor for temperature and humidity monitoring, and a flow sensor to identify flow irregularities. An Arduino Uno processes sensor data and sends it to a Node MCU ESP8266 module, which uploads the data to the Thing Speak cloud platform for remote visualization and analysis.

In the event of a detected anomaly, the system activates a local buzzer, sends an SMS alert via the SIM800L GSM module, and includes GPS coordinates from the NEO-6M GPS module to accurately identify the fault location. Additionally, it initiates an emergency shutoff by controlling a solenoid valve using a relay module, minimizing the risk of further damage. This integrated approach ensures fast detection, immediate alerts, and real-time accessibility.

The novelty of this project lies in its integration of real-time embedded sensing, wireless communication, and cloud-based analytics in a compact and

deployable unit, specifically tailored for underground gas pipeline applications. Key innovative aspects include:

Wireless, cloud-integrated architecture using ESP8266 and Thing Speak , eliminating the need for expensive and impractical wired networks.

Dual communication mechanism—SMS alerts via GSM and cloud updates via Wi-Fi—ensuring redundancy and reliability.

Location-aware alerts using GPS for precise fault localization, a feature often absent in low-cost implementations.

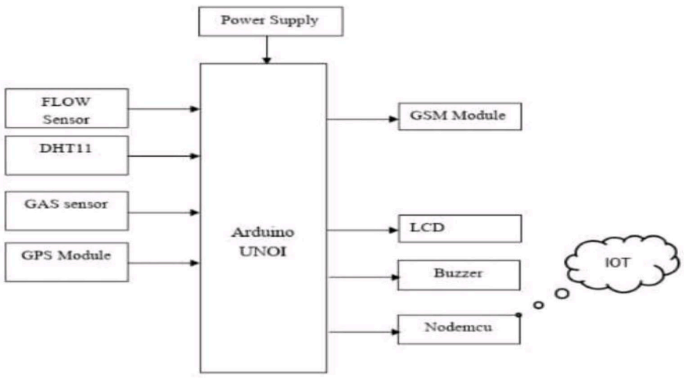
Automatic shutoff feature through a relay-controlled solenoid valve, enabling immediate mitigation of leak-induced hazards.

Modular and scalable design, allowing it to be adapted for various pipeline lengths, environments, and sensor types.

This system offers a cost-effective alternative to industrial SCADA-based solutions and sets the foundation for future enhancements like AI-based predictive maintenance, multi-node WSN, and mobile app integration for alerts

Working Principle

a) Flow chart



b) Hardware Components

Components	Function
Arduino Uno	Central controller for reading sensor data and communicating via UART
Node MCU ESP8266	Wi-Fi-enabled microcontroller to send data to Thing Speak
MQ-6 Gas Sensor	Detects combustible gases like Butane and Propane
DHT11 Sensor	Measures temperature and humidity
Flow Sensor	Detects gas movement; alerts in case of no flow or leakage
NEO-6M GPS Module	Provides real-time location coordinates for SMS alerts
SIM800L GSM Module	Sends SMS notifications using AT commands
LCD Display (16x2)	Displays real-time sensor data on-site
Buzzer	Audible alert for gas leak or high temperature
Power Supply	Provides regulated 5V and 3.3V power to all components

c) **Software used**

Software /Tools	Function
Arduino IDE	Used to program Arduino Uno and NodeMCU ESP8266
Thing Speak IoT Cloud	For online data visualization and historical trend monitoring
Embedded C (in Arduino IDE) Core programming language for hardware logic Serial Monitor	For debugging serial communication and GPS/GSM data
AT Commands	Used for controlling the GSM module to send SMS

Software Sources and Libraries Used

Arduino code Library

Library Used	Function
<DHT.h>	Reads temperature and humidity from the DHT11 sensor.
<LiquidCrystal.h>	Controls the 16x2 LCD.
<SoftwareSerial.h>	Enables UART communication on non-standard pins.
<TinyGPS.h>	Parses and extracts data from GPS module NMEA sentences.

Node MCU ESP8266 Libraries

Library	Function
<ESP8266WiFi.h>	Connects to Wi-Fi network.

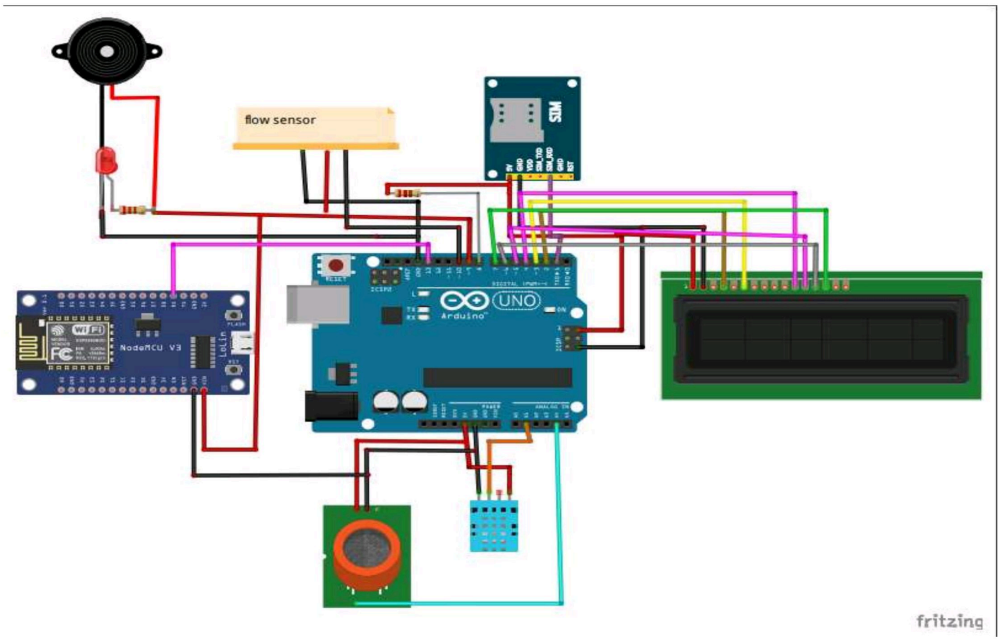
<WiFiClient.h>	Sends HTTP requests to ThingSpeak server.

Communication Protocols

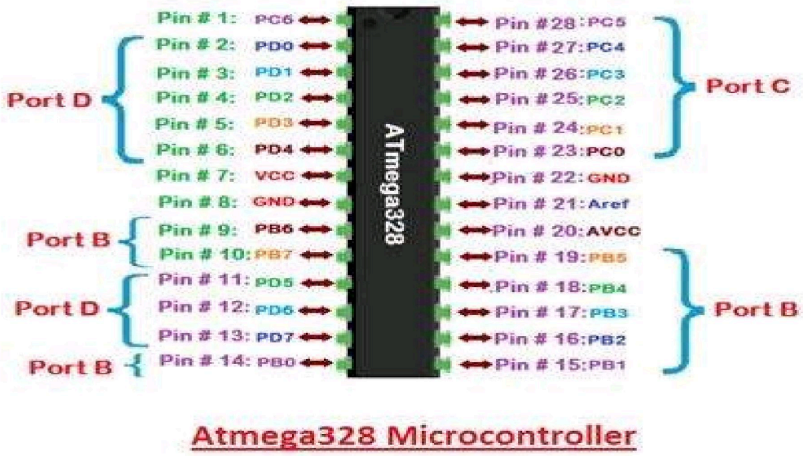
Protocol	**Function**
UART (Universal Asynchronous Receiver/Transmitter)	Used for serial data exchange between: Arduino and Node MCU Arduino and GSM Module Arduino and GPS Module
HTTP POST	Used by Node MCU to push data to ThingSpeak.

Product Integration (Hardware and software)

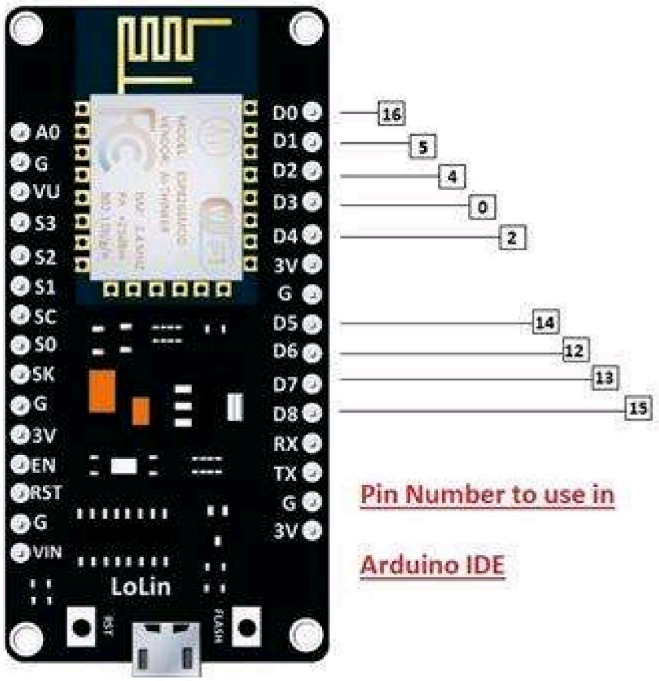
Citrcuit Diagram (Using Tinkercard)



Pin diagram

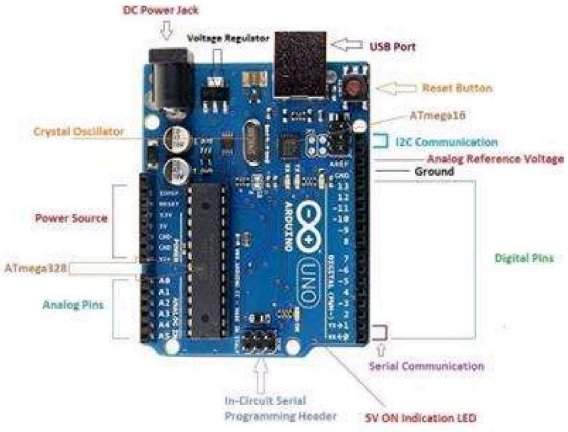


Atmega328 Microcontroller



1. Sensors

- **MQ-6 Gas Sensor** *(Digital Pin 8)*
  ➤ Detects presence of harmful gases such as LPG, Methane, and Propane.
- **Flow Sensor** *(Digital Pin 9)*
  ➤ Monitors gas flow to identify leaks or blockages.
- **DHT11 Temperature & Humidity Sensor** *(Analog Pin A2)*
  ➤ Tracks temperature and humidity in the pipeline area.



Arduino UNO

◇ 2. Microcontrollers

- **Arduino UNO** *(Main Control Unit)*
  - ➤ Reads data from all sensors
  - ➤ Controls alerting and safety mechanisms (Buzzer, LCD, Relay)
  - ➤ Sends formatted data to NodeMCU via Software Serial
- **Node MCU ESP8266**
  - ➤ Receives data from Arduino UNO
  - ➤ Uploads sensor data to **ThingSpeak IoT Dashboard** via Wi-Fi

◇ 3. Communication Modules

- **SIM800L GSM Module**
  ➤ Sends **SMS alerts** to predefined contacts on fault detection
  ➤ Uses **GPS data** from NEO-6M for location tagging
- **NEO-6M GPS Module**
  ➤ Provides accurate **latitude and longitude** of the fault location

4. Alerting Devices

- **Buzzer** *(Digital Pin 10)*
  ➤ Emits sound alerts during gas leaks, temperature anomalies, or flow disruptions
- **16x2 LCD Display** *(Pins 2–7)*
  ➤ Shows real-time sensor readings and alert messages

5. Actuators

- **Relay + Solenoid Valve**
  ➤ Automatically **shuts off gas supply** during hazardous events
  ➤ Controlled by Arduino digital output for safety response

Software Working – Arduino Code (Fault Detection & Alert)

Code working and explanation

Setup Phase

```
void setup() {
Serial.begin(9600);        // Main Serial – GPS & GSM
mySerial.begin(9600);      // Software Serial – NodeMCU
dht.begin();               // Initialize DHT11
pinMode(gas, INPUT);       // MQ-6 Gas Sensor
pinMode(flow, INPUT);      // Flow Sensor
pinMode(buzzer, OUTPUT);   // Buzzer
lcd.begin(16, 2);          // LCD Setup
}
```

➤ Loop Phase – Sensor Data Reading and taking action

```
int flow_value = digitalRead(flow);
int gas_value = digitalRead(gas);
int temperature = dht.readTemperature();
int humidity = dht.readHumidity();
```

🔹 ➤ Conditions & Alerts

- High Temperature (>32°C)
  - ➤ Activates buzzer
  - ➤ Displays alert on LCD
  - ➤ Sends SMS with GPS location
- Flow Sensor = 1 (Leakage)
  - ➤ Indicates pipeline leakage
  - ➤ Activates buzzer, LCD warning
  - ➤ Sends SMS alert
- Gas Sensor = 0 (Gas Detected)
  - ➤ Dangerous gas level
  - ➤ Triggers buzzer
  - ➤ Sends alert SMS

🔹 ➤ GPS Location Parsing

```
gps.f_get_position(&flat, &flon, &age);
```

- Location sent via SMS in Google Maps format: **http://www.google.com/maps/place/{latitude},{longitude}**

🔸 ➤ Send Message Function

```
void SendMessage() {
  Serial.println("AT+CMGS=\"+91xxxxxxxxxx\"");
  Serial.println(msg);
  Serial.println("http://www.google.com/maps/place/" +
flatStr + "," + flonStr);
}
```

- ➤ **Send Data to NodeMCU**

```
uno = "a" + temperature + "b" + humidity + "c" +
flow_value + "d" + gas_value + "e";
mySerial.println(uno);
```

Software Working – Node MCU Code (Cloud Upload to Thing Speak)

- ➤ **Wi-Fi Setup**

```
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
}
```

- ➤ **Main Loop Functionality**

  - Waits data string from Arduino over serial.
  - Parses string to extract:
    - **Temperature**
    - **Humidity**
    - **Flow**
    - **Gas Level**

➤ **Construct POST Request to Thing Speak**

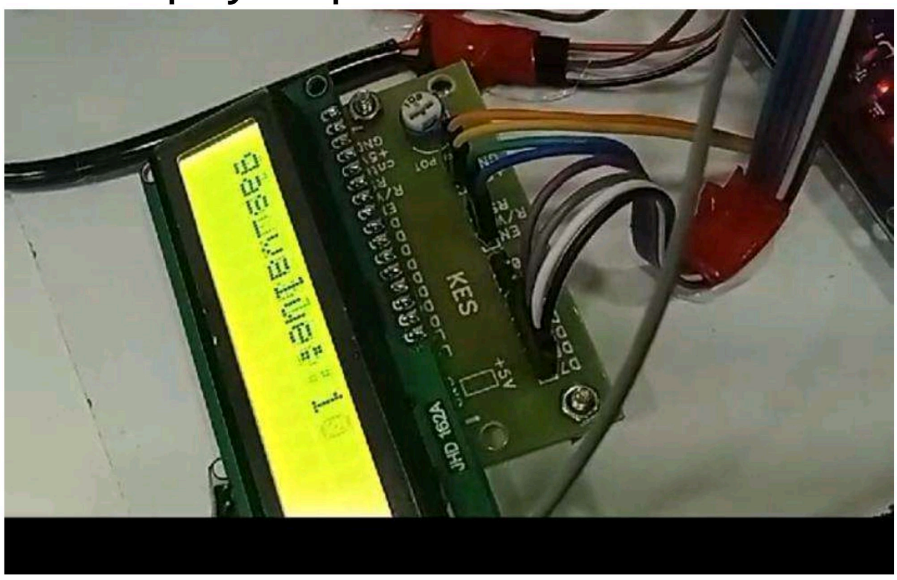
```
postStr += "&field1=" + valueA; // Temp
postStr += "&field2=" + valueB; // Humidity
postStr += "&field3=" + valueC; // Flow
```

```
postStr += "&field4=" + valueD; // Gas
client.connect(server, 80);
```

Project Output Results – Sensor-Wise

1. Harmful Gases Detected – MQ-6 Sensor

LCD Display Output:



harm ful gases :message sent
detected

Buzzer Output:

➤ Activates for **4 seconds** (HIGH), then turns off

📨 **SMS Alert Sent:**

harmful gases detected - MQ6
Location: Click to view
http://www.google.com/maps/place/0.000000,0.000000

ThingSpeak Output:

➤ **Field 4 (Gas Value) = 0**

→ Indicates **hazardous gas level**

◇ **2. Pipe Leakage Detected – Flow Sensor**

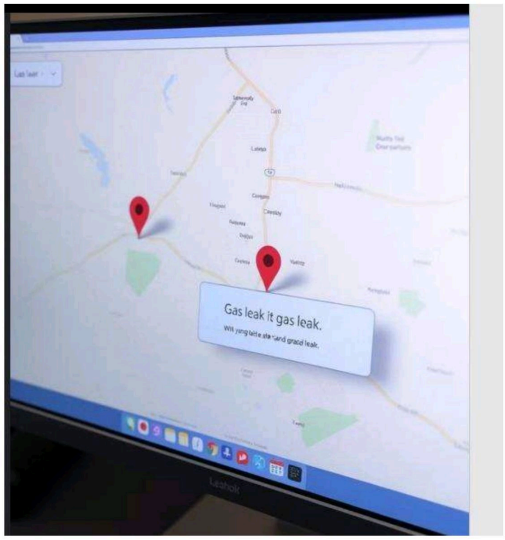
LCD Display Output:

pipe leakage :Message sent
detected

Buzzer Output:

Rings for **4 seconds** to alert nearby personnel

SMS Alert Sent:



pipe leakage detected - FLOW SENSOR
Location: Click to view
http://www.google.com/maps/place/0.000000,0.000000

ThingSpeak Output:

➤ **Field 3 (Flow Value) = 1**

→ Indicates **abnormal flow** or **leakage**

◇ 3. Temperature Threshold Exceeded – DHT11 Sensor

LCD Display Output:

Temperature limit Exceeded:With Value

Buzzer Output:

➤ Activates for **4 seconds** on temperature > **32°C**

SMS Alert Sent:

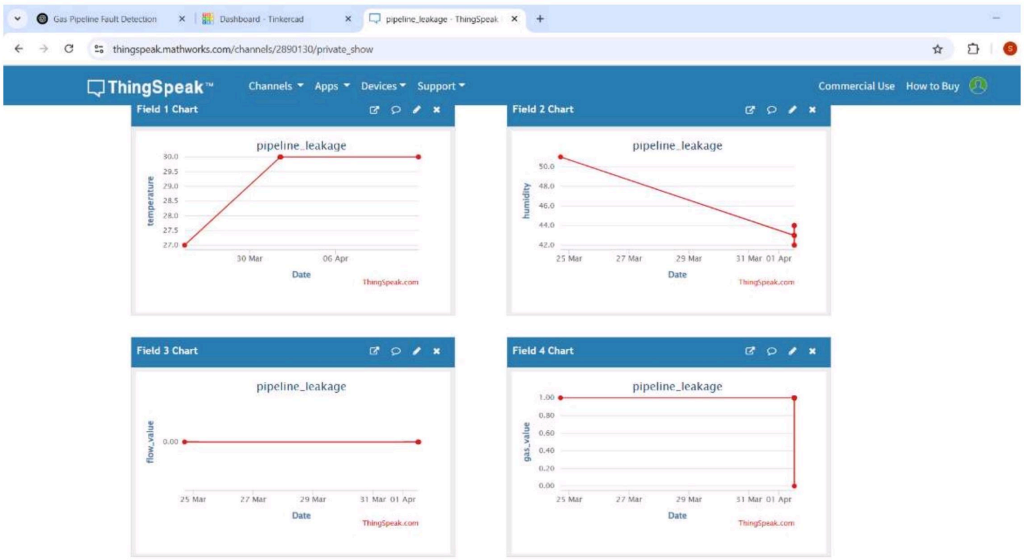
temperature value exceeded - DHT11
Location: Click to view
http://www.google.com/maps/place/0.000000,0.000000

ThingSpeak Output:

➤ **Field 1 (Temperature)** = Value above threshold (e.g., **33°C**)

ThingSpeak Dashboard Output (Field Mapping)

Field	Sensor	Parameter	Example values
Field 1	DHT11	Temperature (°C)	27.0 → 33.0
Field 2	DHT11	Humidity (%)	50% → 42%
Field 3	Flow Sensor	Flow Value	0 (Normal), 1 (Leak)
Field 4	MQ-6 Gas Sensor	Gas Detection	1 (Safe), 0 (Leak)



Overall Software Flow Summary

- **Arduino UNO**:
- Reads values from **Gas**, **Flow**, and **DHT11** sensors
- If a fault is detected:
    - Activates **buzzer**
    - Displays **message on LCD**
    - Sends **SMS with GPS location**
    - Sends **formatted sensor data** to NodeMCU

- **Node MCU ESP8266**:
- Parses incoming string from Arduino
- Uploads real time sensor values data to **Thing Speak IoT Cloud**
- Enables real-time monitoring and visualization

♦ Prototype Costing

Product Cost Breakdown

Component	Quantity	Unit Price (INR
Arduino Uno	**1**	**550**
Node MCU ESP8266	**1**	**400**
MQ-6 Gas Sensor	**1**	**200**
DHT11 Temperature Sensor	**1**	**120**
Flow Sensor	**1**	**300**
SIM800L GSM + NEO-6M GPS (1 set)	**1**	**2500**
LCD Display (16x2 with I2C)	**1**	**250**
Relay Module (Single channel)	**1**	**100**
Solenoid Valve (12V)	**1**	**400**
Buzzer	**1**	**50**
Power Adapter (12V 1A)	**1**	**250**
Breadboard + Jumper Wires	-	**200**
Soldering Supplies & Buffer	-	**230**
Total		**5500**

Practical Deployment Cost Budget (Approx.)

Category	Cost(INR)
Electronic Components (MCUs, Sensors, etc.)	5000

Power Supply (Battery + Solar Panel)	1500
Enclosure (Waterproof, IP65-rated)	800
PCB Fabrication & Circuit Integration	600
Wiring, Connectors & Mounting Parts	300
Assembly, Testing & Labor	1000
Miscellaneous & Packaging	800
Total	10000

- 🔧 **Prototype Cost**: ~₹6,000
- 🏭 **Field-Ready Product Cost Estimate**: ~₹10,000 per unit (approx.)

Target Market (B2B) – Deployment Scope

1. Oil & Natural Gas Corporations

- **Use Case**: Underground pipeline leakage detection and auto shutoff
- **Why**: Prevents hazardous incidents, supports preventive maintenance
- **Examples**: ONGC, Indian Oil, GAIL, BPCL, HPCL, Reliance Industries

2. City Gas Distribution (CGD) Companies

- **Use Case**: Monitoring gas pipelines for urban/domestic supply
- **Why**: Critical for safety in **densely populated residential zones**
- **Examples**: Adani Gas, Mahanagar Gas, Indraprastha Gas

3. Industrial Gas Supply & Chemical Plants

- **Use Case**: Leak detection in industrial gas distribution pipelines
- **Why**: Reduces **shutdown costs** and ensures **regulatory compliance**
- **Examples**: Tata Steel, L&T, JSW, Essar, etc.

🔹 4. Smart City Infrastructure Projects

- **Use Case**: Integration with smart utility and safety systems
- **Why**: Supports **Govt. Smart City Initiatives** with advanced monitoring
- **Opportunity**: Collaborate with urban planners and civic bodies

🔹 5. Disaster Management Agencies & Safety Consultants

- **Use Case**: Deployment in **high-risk zones** for early warning
- **Why**: Ensures **rapid emergency response** and public safety
- **Users**: NDRF, Civil Defence, Private Safety Firms

🔹 6. Public-Private Partnerships (PPP)

- **Use Case**: Large-scale infrastructure safety implementation
- **Why**: Enables **scalable deployment** with potential **government funding**
- **Focus**: Expansion into **semi-urban and rural** gas networks

▪ Why This Product Fits B2B Market Needs

✓ **Affordable, scalable, and easy to install**
✓ **Real-time alerting** via SMS and cloud dashboard
✓ **GPS-based localization** for fault pinpointing
✓ Enables **predictive maintenance** and minimizes downtime
✓ Aids in **regulatory compliance** for gas safety standards

♦ Conclusion

The proposed IoT-Based Real-Time Underground Gas Pipeline Fault Detection and Alert System successfully demonstrates the integration of sensors, microcontrollers, GSM, GPS and cloud platforms to deliver a cost-effective, reliable, and smart safety solution by integrating:

- MQ-6 Gas Sensor, Flow Sensor, DHT11 Temp & Humidity Sensor
- Arduino UNO for control & fault logic
- SIM800L GSM for SMS alerts
- NEO-6M GPS for accurate location tracking
- Node MCU ESP8266 for cloud connectivity via ThingSpeak

Key Capabilities:

- **Real-time monitoring** of underground pipelines
- **Instant alerts** with GPS-based fault localization
- **Data logging & dashboard visualization**
- **Automatic shutoff** using relay-actuated solenoid valves

This solution is ideal for **smart cities**, **industrial plants**, and **critical safety zones**, effectively reducing risk and enabling **proactive gas leak detection**.

Future Scope

The current prototype presents **strong potential for scaling** and **technological upgrades** to align with modern industrial needs and smart infrastructure. Key enhancement areas include:

1. Scalability for Large-Scale Deployment

- Enable deployment across **large pipeline networks** (2–5 km segments)
- Integration with **centralized monitoring systems** for utility providers

2. Advanced Wireless Communication

- Upgrade communication modules to **LoRa**, **NB-IoT**, or **5G** for:
  - **Longer range**
  - **Lower power consumption**
- Adopt **MQTT-based secure protocols** for seamless real-time IoT communication

3. Predictive Maintenance with AI

- Apply **Machine Learning** to ThingSpeak historical data for:
  - **Early fault prediction**

    - **Anomaly detection**
- Automate **inspection schedules** using predictive analytics

4. Improved Power Efficiency

- Implement **solar-powered modules** with battery backups in remote zones
- Use **ultra-low-power microcontrollers** to maximize battery life

5. Web & Mobile App Interface

- Develop a **mobile and web application** for:
    - **Live monitoring**
    - **Instant alerts**
    - **Sensor history dashboard**
- Add **role-based access** for technicians, engineers, and admins

6. Security Enhancements

- Incorporate **end-to-end encryption** for data integrity and privacy
- Design **tamper-proof enclosures** and **anti-sabotage mechanisms**

7. Environmental Monitoring Integration

- Add sensors for:
    - Air Quality (e.g., PM2.5, CO2)
    - Pressure
- Analyse pollution impact due to gas leaks in real time