Sensor Network Lab (Course Code: ITL603)

PolluSense

T. E. Information Technology

By

Mazin Bangi 08
Tanmay Bhatkar 09
Sahil Bangera 07
Shannen Anthony 04

Under Guidance of
Ms. Akanksha Naik
Assistant Professor

Department of Information Technology
St. Francis Institute of Technology
(Engineering College)

(An Autonomous Institute, Affiliated to University of Mumbai)
AICTE Approved | NAAC A+ grade: 03 programs are NBA Accredited: ISO 9001:2015 Certified

2024-25

CERTIFICATE

This is to certify that the project entitled **"PolluSense"** is a bonafide work of **"Tanmay Bhatkar, Mazin Bangi, Sahil Bangera, Shannen Anthony" Roll Nos. 09, 08, 07, 04** respectively submitted to the University of Mumbai towards completion of mini project work for the subject of **Sensor Lab (Course Code: ITL603).**

Ms. Akanksha Naik
Supervisor/Guide

Dr. Prachi Raut
HOD-INFT

Examiners

1.--

2.--

Date:

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

--------------------------------------	--------------------------------------
Mazin Bangi	**Tanmay Bhatkar**
--------------------------------------	--------------------------------------
Sahil Bangera	**Shannen Anthony**

ABSTRACT

Air pollution poses a significant environmental and public health challenge, particularly in densely populated urban and industrial regions. In India, where monitoring infrastructure is limited and personal air quality tracking is rare, obtaining accurate, localized pollution data remains a critical issue. To bridge this gap, our project, *PolluSense*, offers a cost-effective, real-time air pollution monitoring and alert system that operates independently of traditional communication infrastructure. The system utilizes an Arduino Uno microcontroller integrated with key sensors MQ-7 for Carbon Monoxide (CO), MQ-135 for Volatile Organic Compounds (VOCs) and general air quality, and a Sharp Dust Sensor for PM2.5 and dust density detection. Wireless communication is achieved via the HC-05 Bluetooth module, enabling device-to-device data transmission without relying on internet connectivity. Data from the sensors is processed and transmitted to a mobile application, such as the Serial Bluetooth Terminal by Kai Morich, which displays real-time air quality metrics through a user-friendly interface featuring graphs and AQI indicators. Developed using the Arduino IDE, PolluSense ensures timely alerts and continuous monitoring in both industrial and residential zones. Its portable, scalable, and modular design enhances accessibility and adaptability, making it suitable for deployment in pollution-prone areas and contributing to smart city initiatives and health-centered environmental strategies.

Contents

List of Figures

List of Tables

Chapter 1
INTRODUCTION

1.1 Introduction to Domain/Area and Motivation

The PolluSense project lies within the rapidly evolving domain of IoT-enabled environmental monitoring systems, specifically targeting air quality analysis. With growing industrialization and urbanization, air pollution has become one of the most pressing health and environmental concerns globally, and more severely in India. Major air pollutants like Carbon Monoxide (CO), Volatile Organic Compounds (VOCs), Dust Density, and PM2.5 particles are responsible for serious respiratory and cardiovascular health issues, particularly in areas near factories or congested traffic zones.

Traditional air monitoring methods are dependent on expensive, fixed, and sparsely distributed stations, which are inadequate for hyper-local monitoring. In India, less than 5% of people have access to personal air monitoring devices, and commercial alternatives are both limited in number and expensive. This creates a vast data gap, limiting both public awareness and governmental responsiveness.

PolluSense addresses this challenge by providing a low-cost, real-time, and easily deployable air pollution monitoring system using Arduino and sensor technology. Its goal is to democratize air quality data, allowing individuals, communities, and policy makers to take data-driven decisions. The motivation behind this project stems from the urgent need for an accessible and scalable system that can provide accurate pollution insights without relying on centralized infrastructure. PolluSense, thus aligns with national and global efforts to tackle pollution and promote sustainable urban living.

1.2 Problem Statement and Objectives

The aim of this project is to design and develop *PolluSense*, a Bluetooth-based smart air pollution monitoring and alert system capable of continuously tracking, analyzing, and reporting ambient air quality in real time—independent of traditional communication infrastructure. Targeting both industrial and residential zones frequently affected by vehicular emissions, industrial discharge, and domestic pollutants, the system integrates various environmental sensors including MQ-7, MQ-135, and a Sharp Dust Sensor with an Arduino Uno microcontroller. These sensors are used to detect harmful pollutants such as Carbon Monoxide (CO), Volatile Organic Compounds (VOCs), PM2.5 particles, and dust density. Data is transmitted via the HC-05 Bluetooth module to mobile applications like the Serial Bluetooth Terminal, enabling real-time visualization and alerts without the need for internet connectivity. The primary goal is not only to monitor air pollution but also to deliver timely, actionable insights that empower individuals, industries, and authorities to take preventive measures, improve situational awareness, and support informed policy-making in pollution-prone areas.

The objectives of our project are:

A. To design and build an IoT-based system that integrates multiple gas and particulate sensors with Arduino Uno to collect air quality data in real-time.

B. To detect and measure key air pollutants specifically Carbon Monoxide (CO), Volatile Organic Compounds (VOCs), dust concentration, and PM2.5 particles in both residential and industrial zones.

C. To develop a **user-friendly visual dashboard** that displays real-time pollutant levels using charts, graphs, and AQI-based metrics for easy interpretation.

D. To **analyze and compare pollution patterns** across different zones, identifying emission sources like vehicles and factories.

E. To enable **government bodies and health agencies** to make informed decisions by correlating pollution levels with public health data and enforcing appropriate control measures.

F. To lay a foundation for **scaling the system into smart cities**, including integration with health statistics and predictive analytics through machine learning in the future.

1.3 Proposed Solution

The proposed solution, **PolluSense**, is a smart, real-time air pollution monitoring system built on **IoT and embedded system principles**. The system utilizes an **Arduino Uno** as its core microcontroller to integrate a range of sensors like **MQ-7** (Carbon Monoxide), **MQ-135** (air quality and VOCs), and a **Sharp Dust Sensor** (dust and PM2.5 particles).

PolluSense provides **continuous and accurate monitoring** and can be deployed in homes, small factories, or public zones. It offers an ideal combination of **portability, affordability, and scalability**, especially in developing nations where high-end monitoring systems are not feasible for large-scale deployment. Additionally, PolluSense can be placed in **dry, cool environments** and can make use of drying agents to **minimize measurement errors in humid areas**—addressing gaps found in existing sensor technologies.

The system's ability to generate **actionable insights in real-time** makes it a powerful tool for both **individual awareness and large-scale policy formation**. It has future potential to integrate with **smart city infrastructure**, support **health-pollution correlation**, and utilize **machine learning** for pollution prediction and hotspot analysis.

1.4 Organization of the Report

The report is structured in a logical and comprehensive manner to ensure clarity and ease of understanding. The organization is as follows:

- **Chapter 1 – Introduction**: This chapter outlines the problem domain and provides motivation for developing PolluSense. It sets the context for the project by highlighting current limitations in air quality monitoring and introduces the solution framework.

- **Chapter 2 – Literature Survey**: A critical review of existing works and technologies used in air pollution monitoring is presented. This includes an analysis of past projects, sensor types, IoT platforms, their strengths, and identified research gaps such as overestimation in humid conditions and lack of gaseous pollutant sensors.

- **Chapter 3 – Proposed System Design**: This chapter provides a technical breakdown of the system architecture. It includes the block diagram, list of components (like Arduino, MQ sensors, Sharp Dust Sensor), and a detailed circuit design explaining sensor-to-Arduino connections.

- **Chapter 4 – Implementation and Results**: The implementation steps of the project are discussed here, including sensor integration, Arduino programming, and cloud connectivity. It also describes how the system collects and processes data and visualizes it in real-time.

- **Chapter 5 – Conclusion and Future Scope**: This chapter summarizes the project outcomes, reaffirms the usefulness of PolluSense, and outlines future enhancements like integration into smart city networks, predictive analytics using AI, and the development of portable versions for remote or disaster-affected regions.

- **References** and **Application Areas** are included at the end, citing all sources used and listing possible use cases beyond air pollution, such as automated lighting and room occupancy control using similar sensor-based technology.

Chapter 2
LITERATURE REVIEW

In our effort to make this project, we read and reviewed multiple research papers across publications and identified the following technical paper to be most relevant to our project. We collected the methodology used by the said research papers in effort to derive at a conclusion or provide novel solutions to problem statements. Advantages and limitations of every research paper was critically evaluated and correspondingly gaps in it were identified and are attempted to be fixed in our solution

Table 1: Literature Review

REF. NO.	METHODOLOGY	ADVANTAGES	GAPS IDENTIFIED
[1]	The methodology involved deploying a low-cost NOVA SDS011 sensor near industrial zones to monitor PM2.5 and PM10 levels. The sensor transmitted real-time air quality data to a scalable cloud-based IoT platform where it was processed and stored. Machine learning classifier algorithms were used to categorize the data based on Air Quality Index (AQI) values. Additionally, the system integrated blockchain technology to ensure secure and tamper-proof data logging.	• Low Cost,Mobile Sensors • Use of Scalable IOT platform • Use of Machine Learning (ML) Classifier algorithms to classify the data into multiple classes based on its AQI (Air Quality Index) values	• 1)Overestimation due to unnecessary moisture in high humidity environments • 2) No tool present to measure gaseous pollution, especially VOCs ,NOx and CO.
[2]	This paper reviewed the shift from traditional fixed monitoring stations to portable, low-cost air quality sensors. It examined various technologies such as electrochemical cells, metal oxide semiconductors, NDIR, UV absorption, and MEMS-based light scattering sensors. The methodology focused on real-time, high-resolution data collection for improved spatial and temporal pollution mapping,	• Real-time, portable, and low-cost monitoring. • Increased spatial/temporal data resolution. • Supports compliance monitoring and personal exposure assessment.	• 1) Few sensors for PM mass or hazardous air pollutants (HAPs). • 2) Challenges in translating sensor data to health benchmarks. • 3) No FEM/FRM certification for low-cost sensors.

	enabling better personal exposure tracking and community-level air monitoring.		
[3]	The methodology in this commentary focused on analyzing the performance and reliability of low-cost personal air pollution sensors. It highlighted the need for standardization, proper calibration, and validation of sensors like metal-oxide detectors, electrochemical sensors, and optical particle counters. The paper called for academic and regulatory involvement to ensure that such sensors produce trustworthy data suitable for health research and policy-making.	• Empowers public engagement and personal exposure tracking. • Supports health research (e.g., asthma management). • Potential for real-time policy integration (e.g., traffic management). • Addresses data gaps in developing nations.	● - Lack of standardization and independent validation. ● - Limited peer-reviewed studies on real-world performance. ● - Risks of misuse in legal/policy decisions without quality assurance.

Chapter 3
SYSTEM DESIGN

The block diagram depicted in Fig 3.1 presents the system design which shows the transmitter and receiver sections of the proposed system.

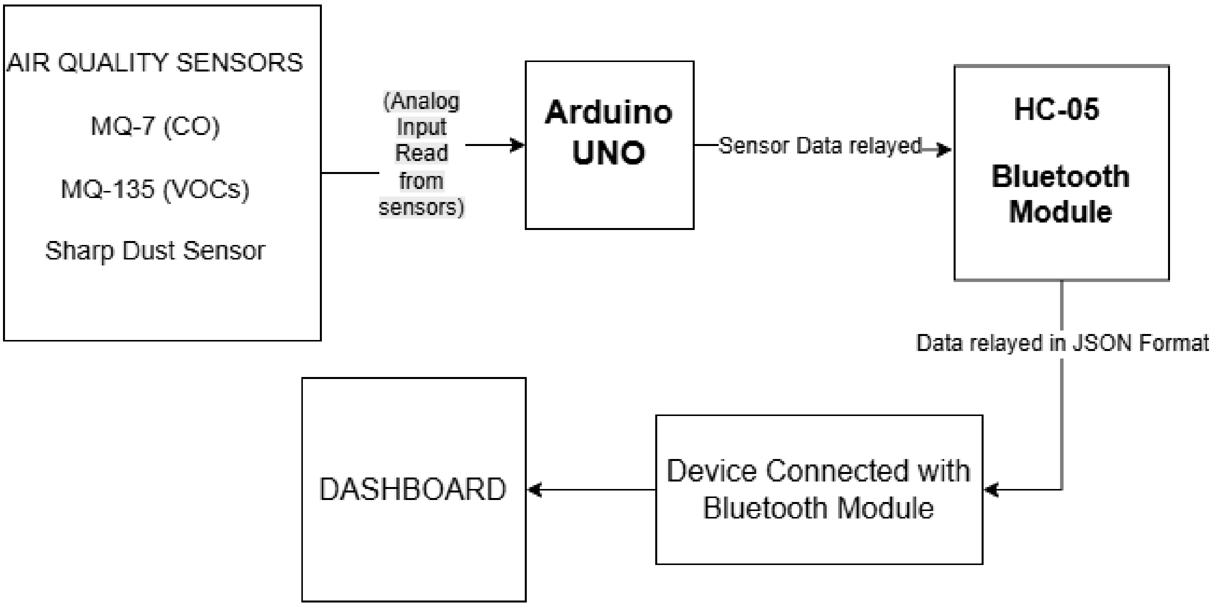

Fig 3.1: Block diagram

The diagram shows a system where air quality sensors (MQ-7 for CO, MQ-135 for VOCs, and a Sharp Dust Sensor) send analog signals to an Arduino UNO. The Arduino processes this data and relays it to an HC-05 Bluetooth module. The Bluetooth module transmits the sensor data in JSON format to a connected device, which then displays the information on a dashboard for monitoring air quality.

Hardware and Software Requirements

Hardware:

Table 2: Component list

S. No.	Name of Component	Price per unit (INR)	No. of units	Total Price (INR)
1	Arduino Uno	-	1	500
2.	MQ-7 Sensor	-	1	105
3.	MQ-135 Sensor	-	1	180
4.	Sharp Dust Sensor	-	1	550
5	Jumper Wires	-	multiple	125
6	Breadboard	-	1	160
7	Power Supply (5V Adapter)	-	1	100
8	HC-05 Bluetooth Module		1	250
	TOTAL	-		-

Software:

- Arduino IDE for programming the microcontroller
- Bluetooth Terminal application for mobile devices

Circuit Diagram

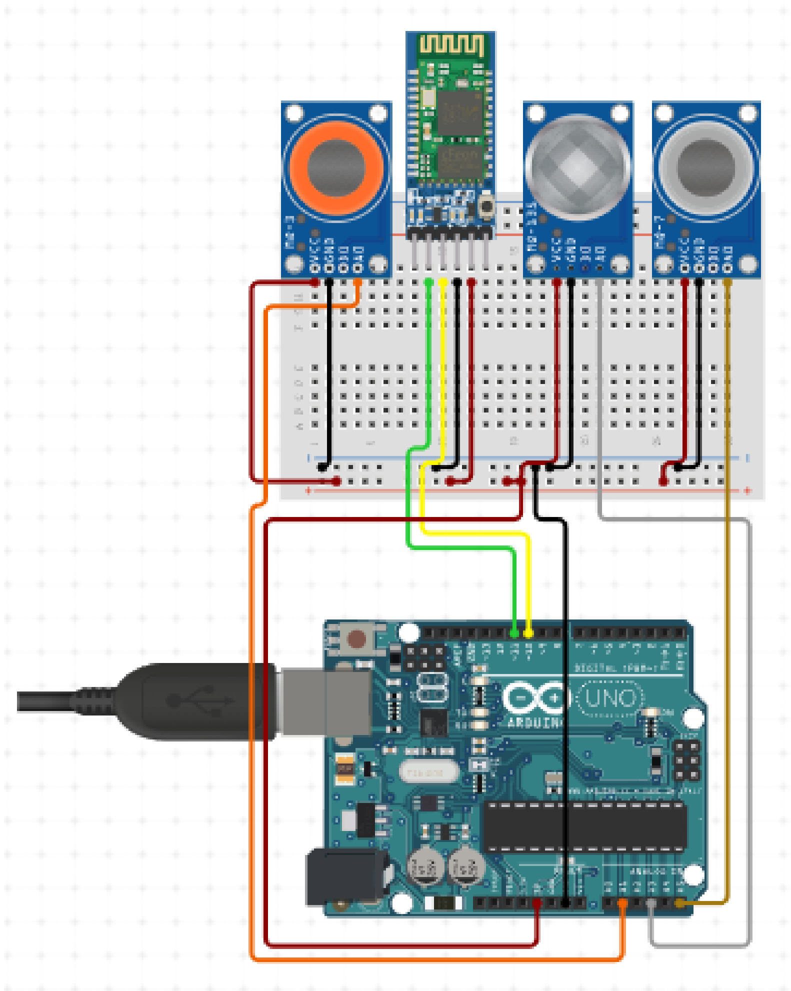

Fig 3.2: Circuit diagram

This circuit connects three sensors — MQ-135 (air quality), MQ-7 (carbon monoxide), and Sharp GP2Y1010AU0F (dust) — to an Arduino UNO using a breadboard. Each sensor's VCC and GND are linked to the Arduino's 5V and GND, while their analog outputs connect to pins A0, A1, and A2 respectively. An HC-05 Bluetooth module is connected to digital pins 10 and 11 via SoftwareSerial for wireless data transmission. The Arduino, powered by a 5V adapter, reads sensor values and sends the data in JSON format over Bluetooth to a connected device.

Application Areas

1. Any Area with unregulated lightning such as parks,empty roads,etc.
2. It can be used in parking lots to chain the lights on one trigger.
3. It can also be used as an alarm mechanism to prevent intrusion as the lights will light up upon detecting the presence of a human.
4. Ultrasonic Based Automatic Operation of Room Lights and Appliance Control.

Chapter 4
IMPLEMENTATION AND RESULTS

Flowchart

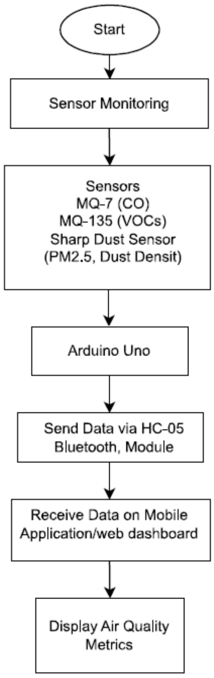

Fig 4.1: Flowchart of the system

The flowchart shows the process of air quality monitoring starting with sensor activation. Sensors (MQ-7 for CO, MQ-135 for VOCs, and Sharp Dust Sensor for PM2.5) collect data, which the Arduino UNO reads. The Arduino then sends the data via the HC-05 Bluetooth module to a mobile app or web dashboard, where air quality metrics are displayed in real-time.

RESULTS:

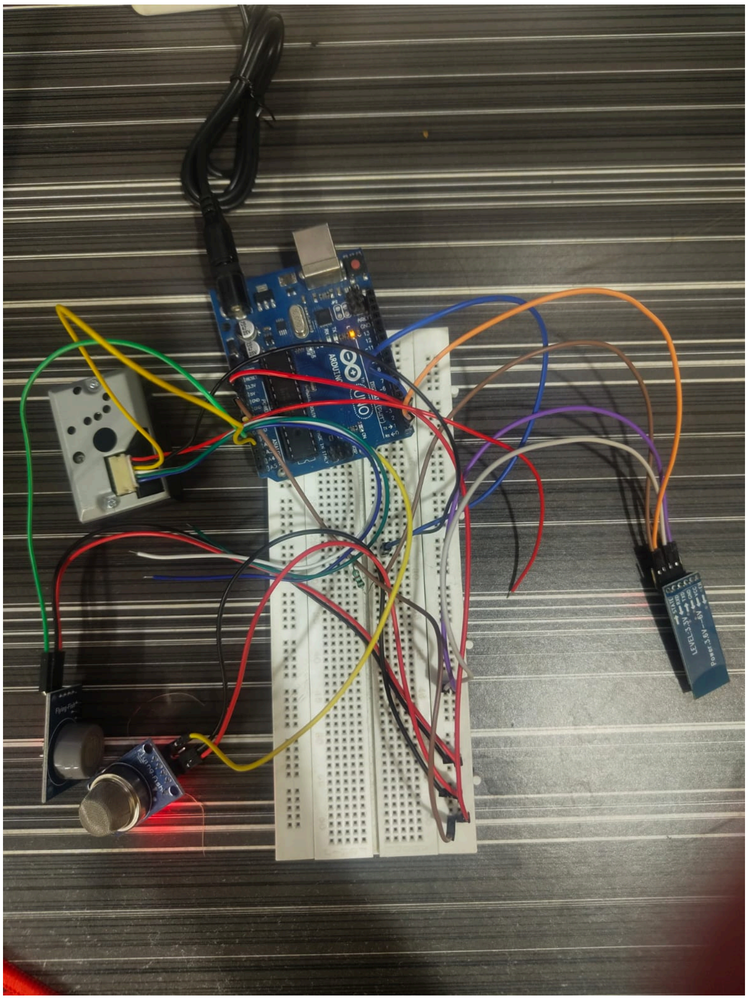

Fig 4.2: Physical Sensor connections.

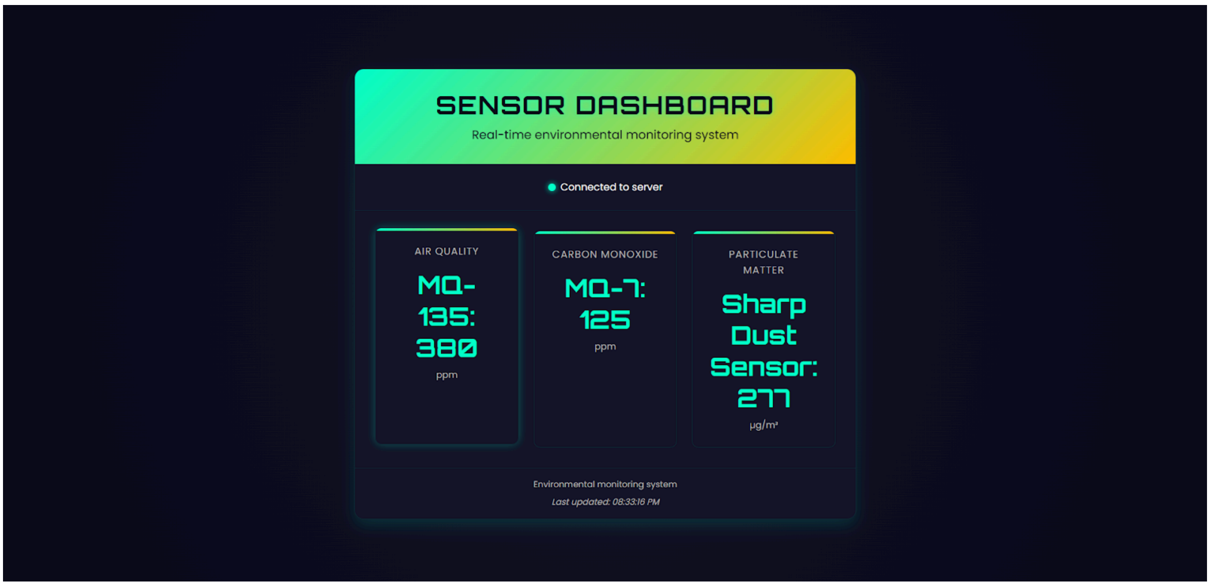

Fig 4.3: Desktop Dashboard

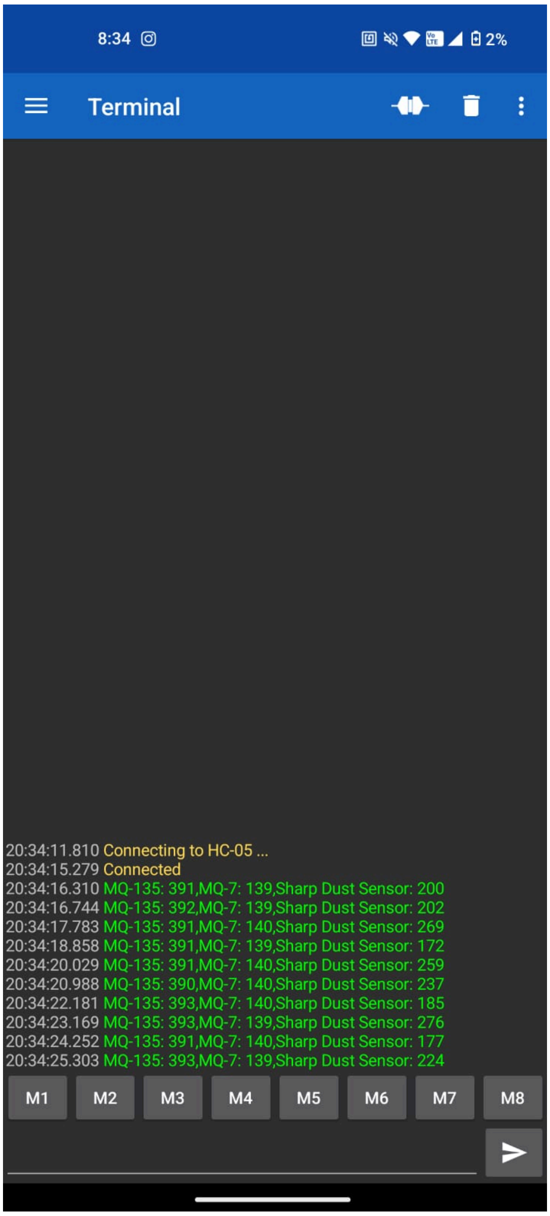

Fig 4.4: Mobile Serial monitor

Code

```
const express = require('express');
const http = require('http');
const { SerialPort } = require('serialport');
const { ReadlineParser } = require('@serialport/parser-readline');
const socketIo = require('socket.io');

const app = express();
const server = http.createServer(app);
const io = socketIo(server);

// Console log to see when server starts
```

```
console.log('Starting server...');

// Serve static files from the public directory
app.use(express.static('public'));

// Serial port setup - Wrapped in try/catch to handle errors
try {
  const port = new SerialPort({
    path: 'COM6', // Make sure this is the correct port
    baudRate: 9600,
  });

  const parser = port.pipe(new ReadlineParser({ delimiter: '\r\n' }));

  parser.on('data', (data) => {
    console.log('Received data:', data);
    try {
      const sensorValues = data.split(',');
      if (sensorValues.length === 3) {
        const [sensor1, sensor2, sensor3] = sensorValues;
        console.log('Emitting sensor data:', { sensor1, sensor2, sensor3 });
        io.emit('sensor-data', { sensor1, sensor2, sensor3 });
      } else {
        console.log('Invalid data format received:', data);
      }
    } catch (e) {
      console.error('Error processing sensor data:', e);
    }
  });

  port.on('error', (err) => {
    console.error('Serial port error:', err.message);
    startDemoMode();
  });
} catch (e) {
  console.error('Failed to open serial port:', e.message);
  startDemoMode();
}

// Add a demo mode function to send test data if serial fails
function startDemoMode() {
  console.log('Starting demo mode with simulated data');
  setInterval(() => {
    const sensorData = {
      sensor1: (Math.random() * 100).toFixed(1),
```

```
    sensor2: (Math.random() * 10).toFixed(1),
    sensor3: (Math.random() * 150).toFixed(1)
   };
   console.log('Emitting demo data:', sensorData);
   io.emit('sensor-data', sensorData);
  }, 2000);
}

// Socket.IO connection handling
io.on('connection', (socket) => {
  console.log('New client connected:', socket.id);

  socket.on('disconnect', () => {
    console.log('Client disconnected:', socket.id);
  });
});

// Start the server
const PORT = process.env.PORT || 3000;
server.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
```

Slave Node

```
#include <SoftwareSerial.h>

// HC-05 Bluetooth module connected to pins 10 & 11
SoftwareSerial BT(10, 11); // RX, TX

void setup() {
  // Initialize serial communication with laptop
  Serial.begin(9600);

  // Initialize Bluetooth communication
  BT.begin(9600);

  Serial.println("Arduino ready to relay data to Bluetooth");
}

void loop() {
  // Read sensor data
  int sensor1 = analogRead(A0);
  int sensor2 = analogRead(A1);
  int sensor3 = analogRead(A2);
```

```
  // Send sensor data to Bluetooth (for your phone)
  BT.print("For Bluetooth Device:  ");
  BT.print("MQ-135: ");
  BT.print(sensor1);
  BT.print(",");
  BT.print("MQ-7: ");
  BT.print(sensor2);
  BT.print(",");
  BT.print("Sharp Dust Sensor: ");
  BT.println(sensor3);

  // Also send to Serial (for debugging on laptop)
  Serial.print("MQ-135: ");
  Serial.print(sensor1);
  Serial.print(",");
  Serial.print("MQ-7: ");
  Serial.print(sensor2);
  Serial.print(",");
  Serial.print("Sharp Dust Sensor: ");
  Serial.println(sensor3);

  // Wait a bit before next reading
  delay(1000);
}
```

Chapter 5
Conclusion

PolluSense offers a practical, affordable, and user-friendly solution for real-time air pollution monitoring. By integrating multiple gas and dust sensors with an Arduino Uno and enabling wireless data transmission, the system provides live insights into key pollutants like CO, VOCs, PM2.5, and dust concentration. The data is visualized through a mobile-compatible dashboard, making it accessible to non-technical users, especially small-scale factory owners and local authorities. Unlike traditional, complex setups, PolluSense allows for easy deployment and informed decision-making, helping users monitor air quality without the high costs or technical burden.

Future Scope

1. **Smart City Integration:**
   PolluSense can be embedded at micro levels within smart city networks for area-specific pollution tracking in traffic zones, parks, and residential areas.

2. **AI-Based Predictive Analytics:**
   By integrating machine learning, the system can forecast pollution patterns and suggest actions like traffic control or public alerts.

3. **Health Data Correlation:**
   With added health data, the system could help link pollution levels with respiratory illnesses, strengthening environmental policy decisions.

4. **Portable Deployment:**
   A compact, portable version of PolluSense could be used in remote or disaster-affected areas to support real-time environmental assessments.

References

RESEARCH PAPER

[1] X. Li, Y. Zhang, and L. Wang, "A review of low-cost particulate matter sensors from the perspectives of data calibration and quality assurance," *International Journal of Advanced Technology and Engineering Exploration*, vol. 10, no. 3, pp. 123–135, 2023. doi: 10.1016/j.ijatee.2023.01.001.

[2] E. G. Snyder *et al.*, "The Changing Paradigm of Air Pollution Monitoring," *Environmental Science & Technology*, vol. 47, no. 20, pp. 11369–11377, 2013. doi: 10.1021/es4022602.

[3] A. Lewis and P. Edwards, "Validate personal air-pollution sensors," *Nature*, vol. 535, pp. 29–31, 2016. doi: 10.1038/535029a.

ONLINE REFERENCES:
https://www.sciencedirect.com/science/article/abs/pii/S2214785323014128
https://pubs.acs.org/doi/epdf/10.1021/es4022602?ref=article_openPDF

Acknowledgment

We are thankful to our college **St. Francis Institute of Technology** for giving us this chance to gain exposure in solving real world problems and acquire practical knowledge and skill sets that will prove to be very crucial to our long-term career prospects. We would take this opportunity to express our sincerest gratitude to our esteemed Director **Bro. Shantilal Kujur**, our Principal **Dr. Sincy George** and our **HOD, Dr. Prachi Raut** for their encouragement, the direction that they give to our college and us students, and the facilities provided to us.
This project, and the research that we undertook, could not have been realized without the utmost support of our project guide **Ms. Akanksha Naik**, who guided us every step of the way, starting from the conception of the project, right up to the execution of the finished solution.