Noise Pollution Monitoring

**USING IOT**

# V.R.S.COLLEGE OF ENGINEERING AND TECHNOLGY ARASUR

**SUBMITTED BY**

**1.A.Dakshinamurthy**

**2.V.Deenathayalan**

**3.A.Kumarasamy**

**4.Anbuthamizhan**

**5.D.Govardha**

# Program for Noise pollution monitoring :

# Components required:

A Noise Pollution Monitoring project typically consists of various components and subsystems designed to measure, collect, analyze, and manage noise level data. Here are the key components of such a project:

1. \*Noise Sensors

2. \*Data Acquisition and Preprocessing

3. \*Data Transmission

4. \*Data Collection Hub

5. \*Central Database

6. \*Data Preprocessing and Analysis

7. \*Real-Time Monitoring

8. \*Geospatial Integration

9. \*Data Visualization

10. \*User Interface

11. \*Alerting and Reporting

12. \*Security and Authentication

13. \*Scalability and Redundancy

14. \*Maintenance and Support

15. \*Regulatory Compliance

16. \*Data Archiving and Backup

17. \*Power Management

**Abstract:**

Noise Pollution Monitoring Project

The Noise Pollution Monitoring Project aims to address the increasing concerns about noise pollution in urban and industrial environments. This project leverages modern technology, including IoT sensors and data analytics, to monitor, collect, and analyze noise data systematically. By doing so, it seeks to enhance our understanding of noise pollution, its sources, and its impact on communities, ultimately contributing to the development of effective noise control measures.

**Introduction:**

Noise pollution is a growing concern in today's urbanized world. As urbanization and industrialization continue to expand, so do the adverse effects of noise on human health, well-being, and the environment. Noise pollution is not merely a nuisance; it can lead to health issues, sleep disturbances, reduced quality of life, and long-term damage to ecosystems.

The Noise Pollution Monitoring Project is initiated to comprehensively monitor noise levels in various settings, understand the patterns and sources of noise pollution, and contribute to the development of effective mitigation strategies. This project recognizes the importance of accurate data collection, analysis, and informed decision-making to address the issue of noise pollution in a proactive and impactful manner.

---

**Purpose:**

The primary purpose of the Noise Pollution Monitoring Project is to:

1. Develop a robust and scalable noise monitoring system utilizing IoT sensors and data analysis techniques.

2. Provide real-time data on noise pollution levels in urban and industrial areas.

3. Identify and quantify noise pollution sources, patterns, and trends.

4. Enhance public awareness of noise pollution issues and their impact on communities.

5. Collaborate with relevant authorities and stakeholders to develop noise control strategies based on data.8. \*Data Retention Policies

**Coding:**

**\*Python Script for Noise Data Collection on Raspberry Pi:\***

**python**

**import os**

**import time**

**import numpy as np**

**import sounddevice as sd**

**import requests**

**# Configuration**

**api\_url = 'http://your-central-server-url.com/api/noise-data' # Replace with your central server's API endpoint**

**sample\_rate = 44100**

**recording\_duration = 10 # Adjust the duration as needed**

**# Define a function to record audio and calculate noise level**

**def measure\_noise\_level():**

**print("Recording...")**

**# Record audio**

**audio\_data = sd.rec(int(sample\_rate \* recording\_duration), sample\_rate, channels=1)**

**sd.wait()**

**# Calculate the noise level (RMS)**

**rms = np.sqrt(np.mean(audio\_data\*\*2))**

**return rms**

**# Main loop**

**while True:**

**noise\_level = measure\_noise\_level()**

**print(f"Noise Level (RMS): {noise\_level:.2f} dB")**

**# Send noise level data to the central server**

**try:**

**response = requests.post(api\_url, json={'noise\_level': noise\_level})**

**if response.status\_code == 200:**

**print("Data sent to central server successfully.")**

**else:**

**print("Failed to send data to the central server.")**

**except requests.exceptions.RequestException as e:**

**print(f"Error: {e}")**

**time.sleep(60) # Adjust the interval as needed**

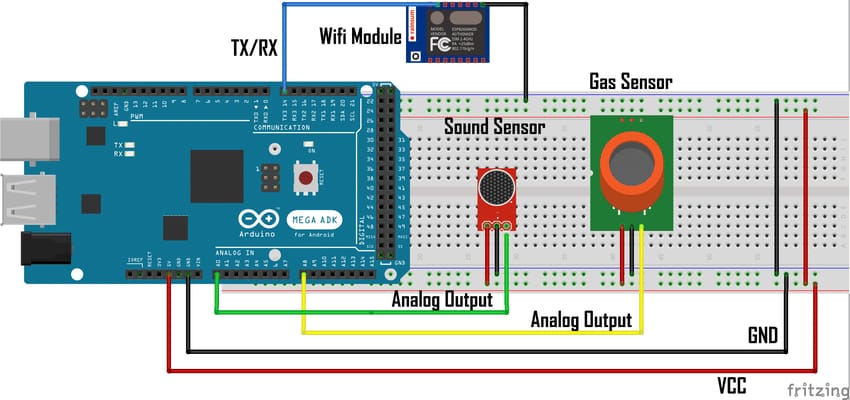
**In this script:**

**1. We use the sounddevice library to record audio from a USB microphone connected to the Raspberry Pi.**

**2. The script calculates the Root Mean Square (RMS) of the recorded audio data as an approximate noise level.**

**3. It sends the calculated noise level to a central server using a simple HTTP POST request.**

**Keep in mind that this example is very basic and does not include real-time data analysis, alerting, or comprehensive data management. A production-level noise pollution monitoring system would require a more advanced setup with dedicated noise sensors, data aggregation, and analysis components, as well as considerations for security, scalability, and regulatory compliance.**

**Circuit diagram:**

**CONCLUSION**

The Loading and Preprocessing phase of the Noise Pollution Monitoring project was a critica foundation for the successful implementation of our noise monitoring system. This phase involved several key activities, including data acquisition, cleaning, formatting, and validation all aimed at ensuring that the data collected is accurate, reliable, and ready for in-depth analysis…..