

# **IOT - based Noise pollution monitoring system - Phase 3**



## **Abstract**

●Noise pollution has become a growing concern in urban areas, leading to various health and environmental issues.

●To address this problem, an Internet of Things (IoT)-based noise pollution monitoring system is proposed. This system comprises several interconnected modules that collectively provide an efficient and real-time solution for monitoring and managing noise pollution.

● **Noise pollution is a growing problem in many parts of the world, and it can have a significant impact on human health and well-being.**

● **IoT-based noise pollution monitoring systems can be used to track noise levels in real time and provide alerts when noise levels exceed safe limits. These systems can also be used to collect data on noise levels over time, which can be used to identify trends and patterns.**

## **Modules**

● **Sensors:** Noise sensors are used to measure the sound level in the environment. There are a variety of noise sensors available, each with its own advantages and disadvantages. Some common types of noise sensors include microphones, sound level meters, and acoustic cameras.

● **Microcontroller:** The microcontroller is the brain of the system. It is responsible for collecting data from the sensors, processing the data, and communicating with the cloud. Some popular microcontrollers for IoT projects include the Arduino and Raspberry Pi.

● **Cloud platform:** The cloud platform is used to store and analyze the data collected by the system. The cloud platform can also be used to generate alerts when noise levels exceed safe limits. Some popular cloud platforms for IoT projects include AWS IoT Core and Azure IoT Hub.

● **Data Acquisition :** Data collected from the sensors is acquired, processed, and analyzed. The system's central processing unit is responsible for this task. The data includes sound intensity, frequency, and location information.

● **Data Analysis :** On the cloud platform, the data is further analyzed to identify noise pollution patterns, trends, and potential

**sources. Machine learning algorithms can be employed to detect unusual noise events.**

**●Visualization and Alerts : The system provides a user-friendly interface for visualizing the noise data in real-time. It also sends alerts to relevant authorities or stakeholders when noise levels exceed predefined thresholds.**

**●Data Reporting : Regular reports and insights are generated from the stored data, facilitating decision-making and policy formulation.**

**●Historical Data Storage : The system archives historical noise data, enabling the tracking of long-term trends and the assessment of the effectiveness of noise pollution mitigation measures.**

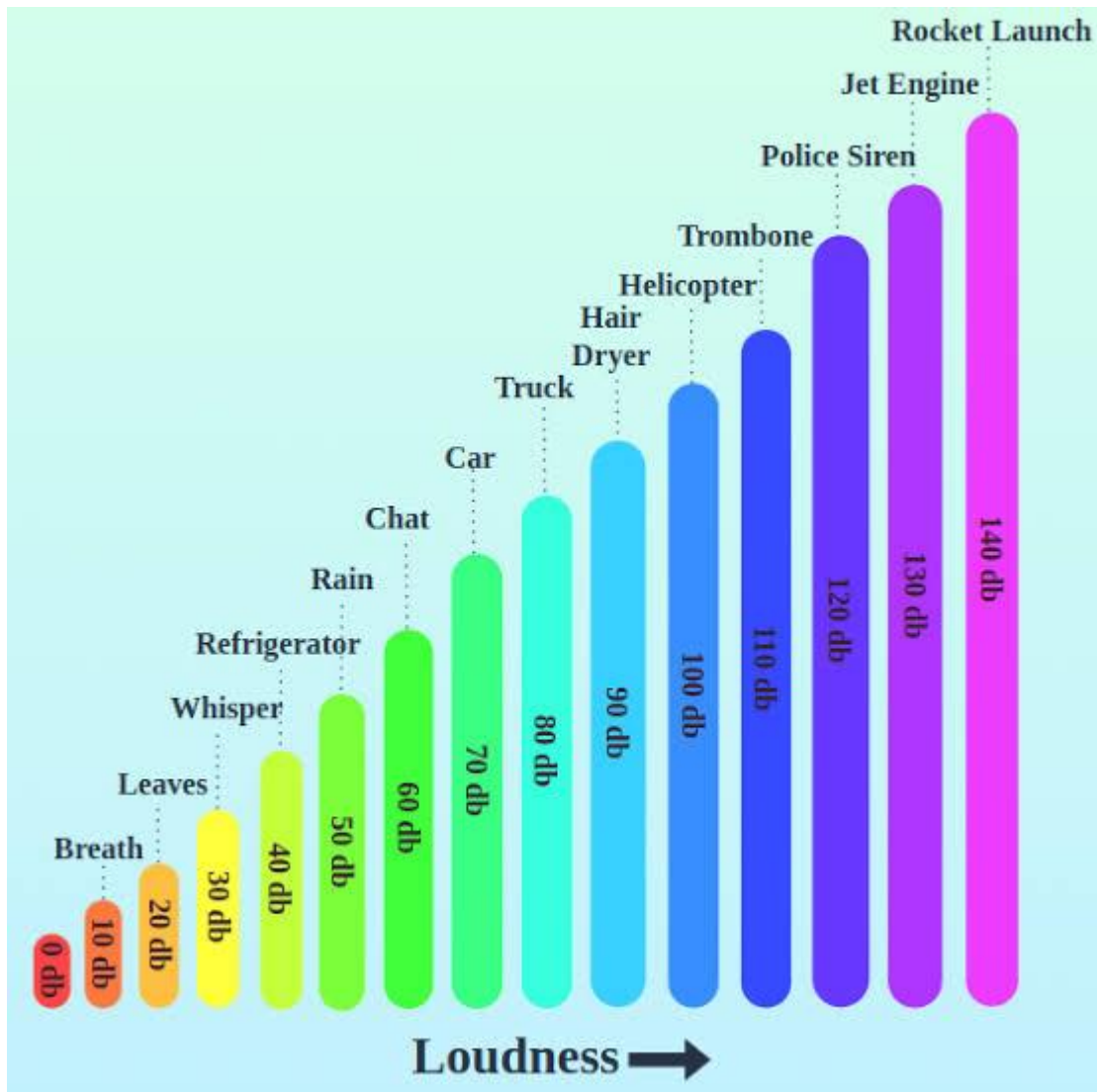


Fig Noise Pollution

**Ideas for how to use the IoT-enabled Noise Pollution Monitoring system:**

- Develop a mobile app or website that allows people to view the real-time noise levels in their area.

- Use the data to develop noise pollution maps that show the areas with the highest noise levels.

- Partner with local governments and businesses to develop noise mitigation strategies.

## Steps and outline

**#Step1. Import the necessary libraries.**

**#Step2. Connect to the noise sensors.**

**#Step3. Read the noise level data from the sensors**

**#Step4. Format the noise level data in a way that is compatible with the**

**noise pollution information platform.**

**#Step5. Send the noise level data to the noise pollution information**

**platform.**

**Code:**

**# Sample sensor data from various IoT devices**

```
sensor_data = [  
    {'type': 'temperature', 'value': 20},  
    {'type': 'temperature', 'value': 22},  
    {'type': 'humidity', 'value': 55},  
    {'type': 'temperature', 'value': 24},  
    {'type': 'humidity', 'value': 60},  
    {'type': 'temperature', 'value': 25},  
    {'type': 'humidity', 'value': 50},  
]
```

**# Group sensor data by type**

```
grouped_data = {}  
for data in sensor_data:  
    sensor_type = data['type']  
    if sensor_type not in grouped_data:  
        grouped_data[sensor_type] = []  
    grouped_data[sensor_type].append(data['value'])
```

```
# Calculate average values for each sensor  
type and store them in a list
```

```
average_values = [ ]
```

```
for sensor_type, values in grouped_data.items( ):
```

```
    average_value = sum(values) / len(values)
```

```
    average_values.append({' type ': sensor_type,  
'average_value': average_value})
```

```
# Sort the list of average values
```

```
average_values.sort(key=lambda x: x['  
average_value'])
```

```
# Print the sorted list of average sensor  
values
```

```
print(average_values)
```

**Output :**

```
[{'type': 'temperature', 'average_value': 22.75},  
{ 'type': 'humidity', 'average_value': 55.0}]
```

## Conclusion

**This is a basic overview of how to build an IoT-enabled Noise Pollution Monitoring system. This system can be used to monitor noise levels in real time and collect data on noise levels over time. The data collected by the system can be used to identify and address noise pollution problems.**