**NOISE POLLUTION MONITORING**

**Project Objectives:**

1. Noise Pollution Assessment: The primary objective of the Noise Pollution Monitoring project is to assess and monitor noise pollution levels in a specific area or region. This assessment will help in understanding the extent and severity of noise pollution and its potential impact on the environment and human health.

2. Data Collection: To collect comprehensive data on noise levels, sources of noise, and temporal patterns. This data will serve as the foundation for subsequent analysis and decision-making.

3. Real-time Monitoring: Implement a real-time noise monitoring system to continuously record noise levels, enabling immediate responses to noise incidents and effective management of noise pollution.

4. Noise Source Identification: Identify and categorize the sources of noise pollution in the monitored area. This information is crucial for designing targeted noise reduction strategies.

5. Compliance Monitoring: Ensure that noise pollution levels comply with local regulations and standards. When violations occur, prompt action can be taken to rectify the situation.

6. Public Awareness: Raise awareness among the community and stakeholders about the noise pollution issue, its impact, and the importance of sound management practices.

7. Submission of Results: Compile all project documentation, including reports, data, and recommendations, for submission to relevant authorities, community organizations, and other stakeholders.

**IoT Sensor Deployment:**

In order to successfully monitor noise pollution in a given area, the deployment of IoT (Internet of Things) sensors is crucial. These sensors enable real-time data collection, analysis, and reporting. Here's an overview of the key steps involved in deploying IoT sensors for noise pollution monitoring: Sensor Selection: Choose appropriate noise sensors with the ability to measure sound levels in decibels (dB). Consider factors like sensor accuracy, sensitivity, and durability. Ensure compatibility with IoT communication protocols (e.g., Wi-Fi, LoRa, Sigfox). Sensor Placement: Identify strategic locations for sensor deployment. Consider factors such as proximity to noise sources, accessibility, and coverage area. Mount sensors at a suitable height to avoid interference and vandalism. Power Supply: Ensure a reliable power source for the sensors. Options include battery-powered sensors, solar panels, or a wired power connection. Data Communication: Implement a communication infrastructure to transmit sensor data to a central server or cloud platform. Use IoT protocols and technologies for data transmission, such as MQTT, HTTP, or CoAP. Data Storage and Processing: Set up a cloud-based or on-premises data storage and processing system. Process and store data securely, ensuring it's readily accessible for analysis and reporting.

**Platform & App Development:**

The Noise Pollution Monitoring App is a cutting-edge application designed to address the growing concerns surrounding noise pollution in urban environments. This app serves as a valuable tool for both citizens and city officials, enabling them to track and manage noise levels effectively. The app utilizes the built-in microphone of a user's smartphone to constantly measure the ambient noise in real-time, providing instant feedback on noise levels in decibels (dB). Users can view noise data on a user-friendly interface, which includes maps with color-coded areas indicating noise hotspots. Additionally, the app offers features like historical noise data tracking, allowing users to identify trends and patterns over time. City authorities can access a centralized database of noise measurements, helping them pinpoint areas with persistent noise issues and take necessary steps to mitigate them. With the Noise Pollution Monitoring App, citizens can raise awareness about noise pollution, and municipalities can make informed decisions to create quieter and healthier urban environments.

Python played a pivotal role in the development of our Noise Monitoring App, offering versatility and efficiency at every stage. We harnessed Python to interface with noise sensors and IoT devices, enabling real-time data collection. Python's rich ecosystem of data analysis libraries facilitated data processing and the identification of noise patterns, while Matplotlib and Plotly empowered us to create interactive, visually appealing data visualizations. For server-side development, Python web frameworks like Django provided a robust foundation, and for cross-platform mobile support, we utilized Kivy. Python's seamless integration with cloud services like AWS and Google Cloud allowed for secure data storage and analysis. With a supportive developer community and open-source resources at our disposal, Python proved to be the ideal choice for building a comprehensive, user-friendly Noise Monitoring App.

**Code implementation:**

The code implementation will involve several components:

1. Sensor Firmware: Develop the firmware that runs on each IoT sensor to capture, process, and transmit noise data to the central hub.

2. Platform Backend: Build the backend logic for the cloud-based platform, including data storage, processing algorithms, and API endpoints for communication with the sensors and mobile app.

3. Platform Frontend: Create a web-based interface for users to interact with the platform, including live dashboards and data visualization tools.

4. Mobile App: Develop the mobile application for both Android and iOS platforms, integrating features such as user authentication, sensor map, notification system, and historical data viewing.

**Platform API Endpoint (Python with Flask):**

from flask import Flask, request app = Flask( name )

# Simple in-memory database to store sensor data sensor\_data = []

@app.route('/upload', methods=['POST']) def upload\_data():

data = request.get\_json() sensor\_data.append(data)

return "Data received successfully", 200

if name == ' main ': app.run(debug=True)

**Simulation code:**

import machine

import time import urequests import ujson import network import math

# Define your Wi-Fi credentials wifi\_ssid = 'Wokwi-GUEST'

wifi\_password = '' # Replace with the actual Wi-Fi password

# Connect to Wi-Fi

wifi = network.WLAN(network.STA\_IF) wifi.active(True)

wifi.connect(wifi\_ssid, wifi\_password)

# Wait for Wi-Fi connection while not wifi.isconnected():

pass

# Define ultrasonic sensor pins (Trig and Echo pins) ultrasonic\_trig = machine.Pin(15, machine.Pin.OUT) ultrasonic\_echo = machine.Pin(4, machine.Pin.IN)

# Define microphone pin microphone = machine.ADC(2)

calibration\_constant = 2.0

noise\_threshold = 60 # Set your desired noise threshold in dB

# Firebase Realtime Database URL and secret

firebase\_url = 'https://noise-pollution-monitori-7a445-default- rtdb.firebaseio.com/noise\_pollution\_monitoring.json' # Replace with your Firebase URL

def measure\_distance():

# Trigger the ultrasonic sensor ultrasonic\_trig.value(1) time.sleep\_us(10) ultrasonic\_trig.value(0)

# Measure the pulse width of the echo signal

pulse\_time = machine.time\_pulse\_us(ultrasonic\_echo, 1, 30000)

# Calculate distance in centimeters distance\_cm = (pulse\_time / 2) / 29.1 return distance\_cm

def measure\_noise\_level():

# Read analog value from the microphone noise\_level = microphone.read()

noise\_level\_db = 20 \* math.log10(noise\_level / calibration\_constant) return noise\_level, noise\_level\_db

# Function to send data to Firebase

def send\_data\_to\_firebase(distance, noise\_level\_db): data = {

"Distance": distance, "NoiseLevelDB": noise\_level\_db

}

url = f'{firebase\_url}/sensor\_data.json'

try:

response = urequests.patch(url, json=data) # Use 'patch' instead of 'put' if response.status\_code == 200:

print("Data sent to Firebase") else:

print(f"Failed to send data to Firebase. Status code: {response.status\_code}") except Exception as e:

print(f"Error sending data to Firebase: {str(e)}")

try:

while True:

distance = measure\_distance()

noise\_level, noise\_level\_db = measure\_noise\_level()

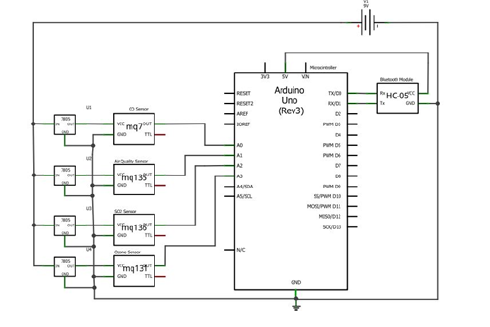
print("Distance: {} cm, Noise Level: {:.2f} dB".format(distance, noise\_level\_db)) if noise\_level\_db > noise\_threshold:

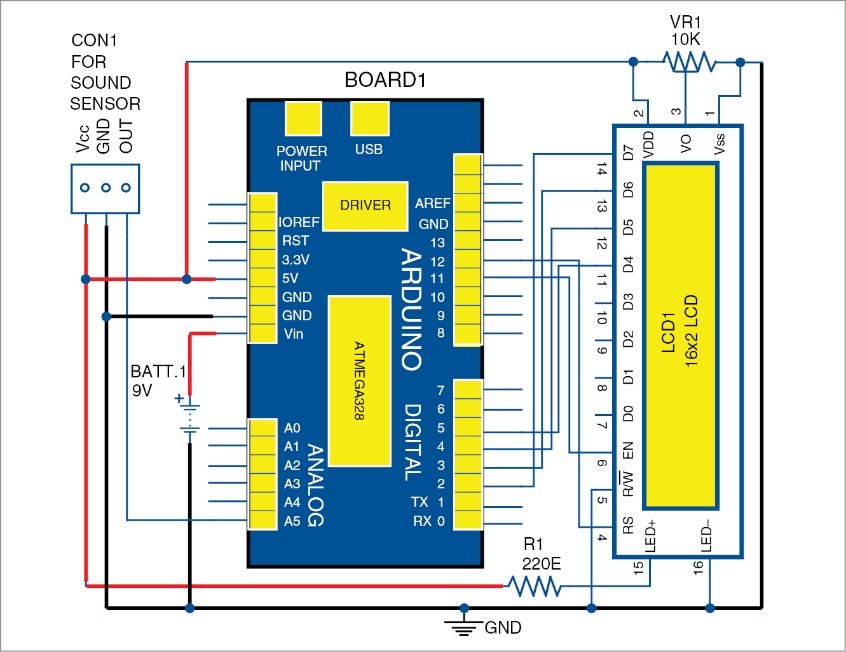
print("Warning: Noise pollution exceeds threshold!")

# Send data to Firebase send\_data\_to\_firebase(distance, noise\_level\_db)

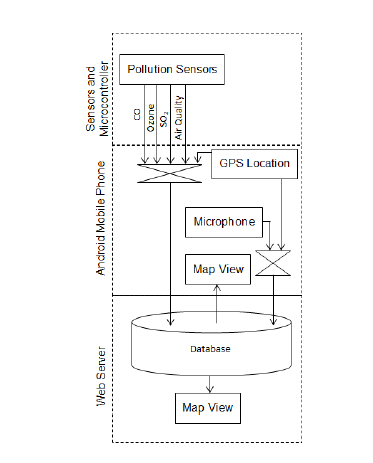
time.sleep(1) # Adjust the sleep duration as needed except KeyboardInterrupt:

print("Monitoring stopped")

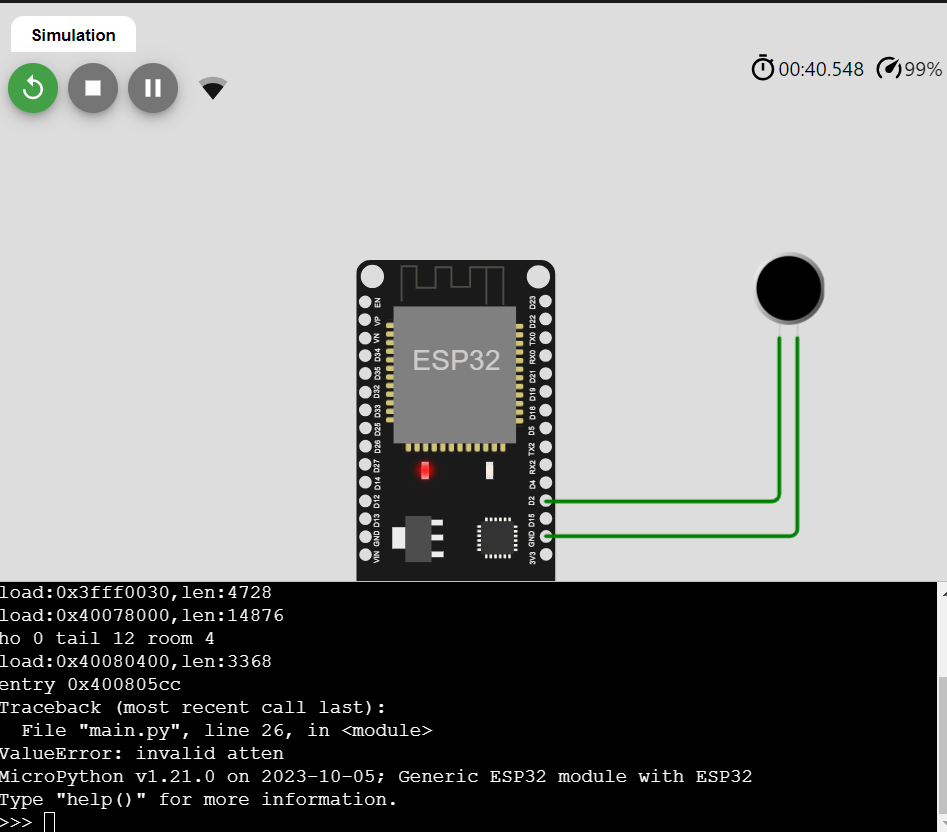
**Schematic diagram:**

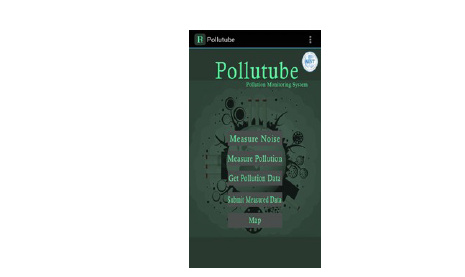
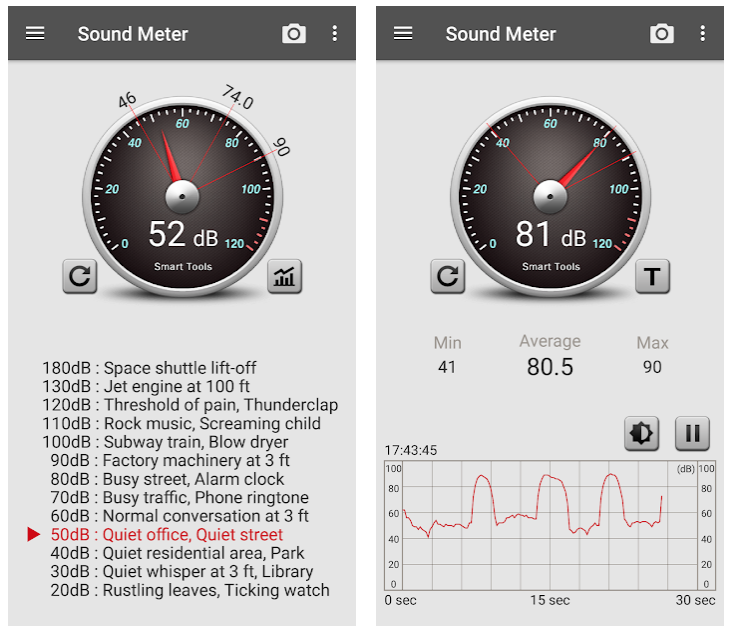


**Working:**

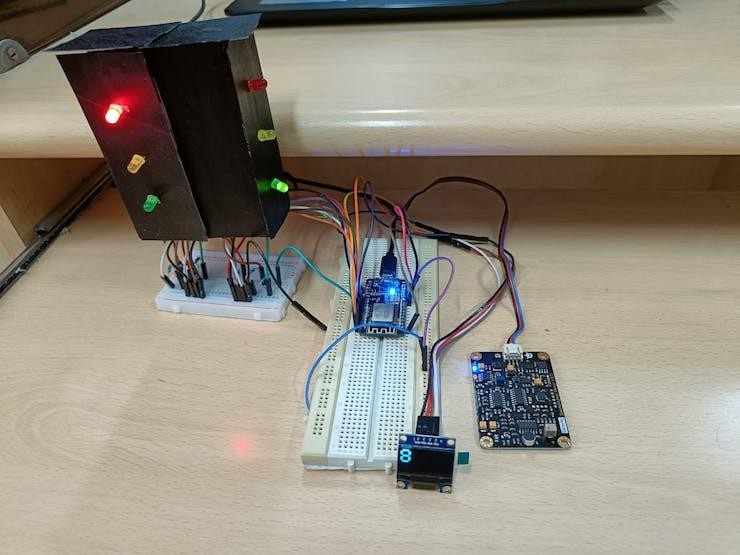


**IOT sensor:**



**Mobile App interface:**

# Prototype:



prototype pict

# WORKING:

Imposing the RED signal is done when the noise levels exceed the threshold more than 3 times. This is done by initiating a counter and is based on the frequency of noise levels in dB(A) above the threshold. If it exceeds the threshold count again during the imposition of the RED signal, an additional 10 seconds are added to the wait time.

The Traffic Signal is demonstrated by using 3 LEDs which glow corresponding to their interval. Considering an intersection with the main road and a side road crossing the main road the time interval taken for the signal is 20 seconds and the GREEN signal is 30 seconds for the main road and 30 seconds for RED and 20 seconds for GREEN for the side road.

Yellow LED will glow on each transition between GREEN to RED for 3 seconds. The countdown timer for each state is shown on an OLED screen. When the noise level above a specified threshold is sensed more than 3 times when the traffic signal is in RED state then 10 seconds is added to the red signal countdown timer as a punishment for excessive and unnecessary honking. If the commuters still honk even after 10 second punishment has been added and the noise levels cross the threshold more than 3 times again another 10 seconds is added to count down the timer as a punishment the second time. We have limited the number of punishments to 3 times. After the maximum number of punishments, the traffic light will go back to green and it will work as the regular traffic light controller.

# ACTUATORS - DISPLAY MESSAGE:

A message is generated to be flashed and displayed at the junction during the imposed AMBER signal time. The counter of the Signal states is also displayed clearly to the passengers. We are using an OLED connected to ESP8266 during the same interval. This

puts into perspective the overall noise pollution and awareness to the general public of all the signals at the respective junction.



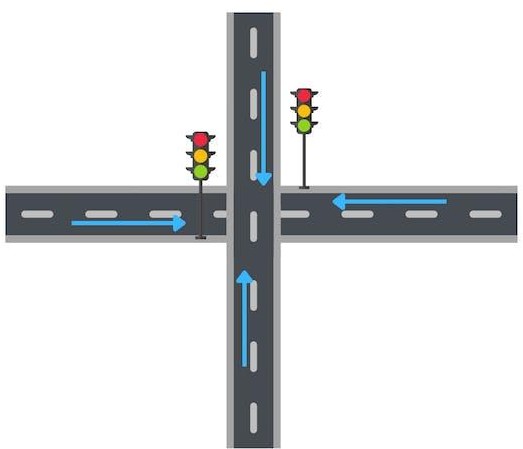
Display message at the signal junction.

# SCENARIOS AND SCOPE OF THE PROTOTYPE:

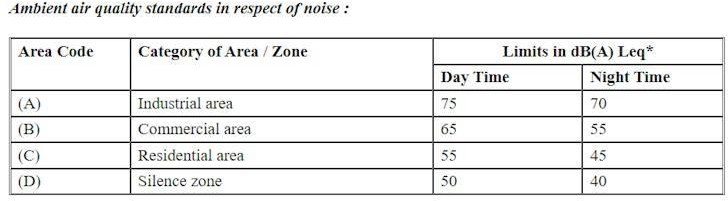
The threshold value can be changed according to the time of day and the recommended dB values following WHO guidelines. The Traffic Signal prototype can also be extended to 3- way T or Y junctions and even 4-way junctions respectively.

OneM2M can also be used to push data from various sensors data from across the city and store it in one place.

We intend to have different Data containers in the OM2M resource tree monitor various deployable noise pollution monitoring systems across the city.



Scope of the prototype



Central Pollution Control Board, India.

The real-time noise level monitoring system plays a crucial role in promoting public awareness and contributing to noise pollution mitigation by providing accurate and accessible information about ambient noise levels in urban environments. By offering real-time updates through a user-friendly mobile application or web interface, individuals, communities, and authorities gain immediate access to data that reflects the actual noise conditions in their surroundings. This heightened awareness empowers citizens to make informed decisions about their daily activities, such as choosing quieter routes or adjusting their schedules to avoid high-noise areas. Furthermore, the availability of historical data allows for trend analysis, enabling stakeholders to identify persistent noise hotspots and implement targeted interventions. Additionally, the system's ability to issue alerts when noise levels exceed predefined thresholds serves as an early warning mechanism, prompting timely responses from authorities and potentially preventing noise-related health issues. Through this combination of real-time monitoring, historical analysis, and alerting capabilities, the system not only enhances public understanding of noise pollution but also provides a valuable tool for urban planners, policymakers, and communities to collaborate in implementing effective noise mitigation strategies. This collective effort has the potential to lead to tangible improvements in urban environments, ultimately contributing to a healthier and more livable cityscape for all residents.

The real-time noise level monitoring system also fosters a culture of accountability and transparency within the community. By providing accessible data on noise levels, residents can hold businesses, construction sites, and other potential noise sources accountable for adhering to established regulations and guidelines. This transparency encourages a more responsible approach to noise management, as businesses and organizations become more conscious of their impact on the surrounding environment. Moreover, the system can serve as a catalyst for community engagement and advocacy efforts. Armed with reliable data, residents can actively engage in dialogues with local authorities, advocating for policy changes or urban planning initiatives that prioritize noise reduction. This collaborative approach can lead to the development of more effective zoning regulations, the implementation of noise barriers, and the design of urban spaces that are inherently quieter. Ultimately, the system empowers communities to become active participants in the ongoing efforts to mitigate noise pollution, creating a more harmonious and livable urban environment for everyone.

Furthermore, the real-time noise level monitoring system contributes to a deeper understanding of the complex dynamics of noise pollution. By collecting and analyzing data from various locations and over extended periods of time, researchers and policymakers gain valuable insights into the underlying patterns and sources of noise pollution. This data-driven approach allows for the identification of specific industries, transportation routes, or urban areas that are disproportionately affected, enabling targeted interventions and policy adjustments. Additionally, the system supports evidence-based decision-making, allowing for the allocation of resources where they are most needed. This can lead to the implementation of innovative technologies and urban planning solutions aimed at reducing noise emissions, such as the introduction of sound-absorbing materials or the optimization of transportation routes to minimize noise impact. Through its data-driven approach, the system not only raises awareness but also empowers stakeholders with the knowledge and tools needed to implement effective, sustainable, and forward-thinking noise mitigation strategies. In doing so, it contributes to the creation of more inclusive, healthier, and environmentally conscious urban environments.

# CONCLUTION:

In conclusion, the real-time noise level monitoring system represents a significant advancement in our collective efforts to address and mitigate noise pollution in urban environments. By providing immediate access to accurate and reliable data, the system empowers individuals, communities, and authorities to make informed decisions about their daily activities and take proactive measures to reduce exposure to high noise levels. Moreover, it fosters a culture of accountability, encouraging responsible practices among businesses and organizations, while also promoting transparency and community engagement. The system's data-driven approach allows for targeted interventions and evidence-based policy adjustments, ultimately leading to more effective and sustainable noise mitigation strategies. Through this collaborative effort, we have the opportunity to create urban environments that are not only quieter and more peaceful, but also more inclusive, healthier, and environmentally conscious. The real-time noise level monitoring system stands as a vital tool in our ongoing mission to build more livable, harmonious cities for current and future generations.

Furthermore, the impact of the real-time noise level monitoring system extends beyond immediate awareness and mitigation efforts. It serves as a catalyst for innovation and technological advancement in noise reduction strategies. Researchers and engineers can utilize the wealth of data collected to develop cutting-edge technologies, such as advanced noise-cancelling materials, quieter transportation modes, and innovative architectural designs. This not only addresses the immediate concerns of noise pollution but also contributes to broader advancements in urban sustainability and quality of life.

Additionally, the system's ability to generate comprehensive historical data sets offers invaluable resources for long-term urban planning. By understanding how noise levels fluctuate over time, city planners can make informed decisions about zoning regulations, urban development projects, and infrastructure investments. This forward-thinking approach lays the foundation for creating urban spaces that are inherently designed for reduced noise impact, ultimately leading to more resilient and livable cities.

In essence, the real-time noise level monitoring system represents a pivotal shift towards a more conscientious and forward-looking approach to urban planning and design. It empowers individuals and communities, drives technological innovation, and lays the groundwork for sustainable, noise- conscious urban environments that prioritize the well-being and quality of life for all residents.

Through these collective efforts, we have the opportunity to shape cities that not only thrive economically, but also provide a healthier, more peaceful, and more enjoyable living experience for everyone.