**NOISE POLLUTION MONITORING**

**Phase5**

To develop a comprehensive noise pollution monitoring system aims to measure, analyze, and manage noise levels to protect public health, ensure compliance, and enhance safety.

**TOPICS:**

* Definition
* Project objectives
* Hardware Setup
* IOT Sensor Deployment
* Diagrams & Schematic
* About Hardware
* Working
* Images Of IOT Sensors
* Programs
* Output
* Platform Information
* Real Time Monitoring
* Conclusion

**DEFINITION:**

*NOISE MONITORING:*

Noise pollution monitoring is the process of measuring and analyzing noise levels in an area to assess its impact, identify sources, and develop strategies for noise reduction and compliance with regulations.

**PROJECT OBJECTIVES :**

The project objective of noise pollution monitoring are typically defined to address specific goals and outcomes related to measurements , assessments and management of noise pollution.

Here are common project objectives for noise pollution monitoring:

**1. \*\*Assessment of Noise Levels\*\*:**

- The primary goal is to accurately measure and assess noise levels in specific areas.

- This assessment helps establish a baseline understanding of existing noise pollution conditions.

- It involves collecting data on decibel levels and patterns of noise over time.

**2. \*\*Identification of Noise Sources\*\*:**

- Identifying and categorizing the primary sources of noise pollution is crucial.

- This includes determining if noise comes from traffic, industrial facilities, construction, or other activities.

- Understanding the sources helps in targeted noise reduction efforts.

**3. \*\*Compliance with Regulations\*\*:**

- Ensuring that monitored areas comply with local and national noise regulations is a key objective.

- This involves comparing recorded noise levels with permissible limits set by authorities.

- Non-compliance may require taking action to reduce noise pollution.

**4. \*\*Noise Mitigation Planning\*\*:**

- Developing comprehensive noise mitigation plans is vital for reducing noise pollution.

- These plans may include measures like noise barriers, quieter road surfaces, or changes in transportation routes.

- The objective is to reduce noise levels and improve the quality of life for residents.

**5. \*\*Data Sharing and Transparency\*\*:**

- Sharing collected noise data with the public, local authorities, and other stakeholders is essential for transparency.

- Transparent data sharing promotes community awareness and involvement in noise reduction initiatives.

- It allows for a collaborative approach to addressing noise pollution issues.

**6. \*\*Real-time Monitoring and Early Warning\*\*:**

- Real-time noise monitoring aims to provide immediate alerts when noise levels exceed predefined thresholds.

- Early warnings help in prompt intervention to address noise pollution issues, especially in critical situations.

- This objective contributes to public safety and well-being.

**7. \*\*Public Health and Safety\*\*:**

- Assessing the impact of noise pollution on public health and safety is a fundamental project objective.

- It involves studying how noise exposure affects the physical and mental health of residents.

- Understanding these impacts is vital for creating effective noise reduction strategies and policies.

These detailed objectives guide the planning and execution of a noise pollution monitoring project, focusing on accurate data collection, source identification, regulatory compliance, community engagement, and ensuring public health and safety.

**IOT SENSOR DEPLOYMENT:**

Deploying IoT (Internet of Things) sensors for noise pollution monitoring is an effective way to gather real-time data and gain insights into noise levels in specific areas. Here is a step-by-step guide for deploying IoT sensors for noise pollution monitoring:

**1. Define Your Objectives:**

Clearly define your objectives for noise pollution monitoring. Determine the specific areas or locations you want to monitor and the type of data you need, such as noise levels, frequency, and duration of loud events.

**2. Select Suitable Sensors:**

Choose appropriate noise sensors that can measure and record sound levels accurately. Common types of sensors include microphones or sound level meters. Ensure they are capable of measuring decibel levels and have the necessary sensitivity.

**3. Power Supply:**

Determine how your sensors will be powered. You can use battery-powered sensors or opt for wired sensors if a constant power source is available. The choice of power source will impact the maintenance and placement of sensors.

**4. Connectivity:**

IoT sensors require connectivity to transmit data. Consider using wireless technologies like Wi-Fi, cellular, or LoRaWAN, depending on the range and data transmission requirements for your monitoring locations.

**5. Data Transmission:**

Implement a secure and reliable method for transmitting data from the sensors to a central data storage system or cloud platform. Ensure the data is encrypted to protect against security breaches.

**6. Sensor Placement:**

Carefully select the locations for sensor deployment. Consider factors such as proximity to noise sources, potential obstructions, and the need for representative data. The height and orientation of the sensors can also affect data accuracy.

**7. Data Collection and Processing:**

Set up a data collection and processing system to manage the incoming data from your sensors. This may involve creating a central data hub or using a cloud-based platform to aggregate and store the data.

**8. Data Visualization:**

Develop a user-friendly interface or dashboard that allows users to visualize noise data in real-time. Graphs, maps, and alerts can help users easily interpret the information.

**9. Data Analysis:**

Implement data analytics and machine learning techniques to extract meaningful insights from the collected data. This can help identify trends, patterns, and sources of noise pollution.

**10. Alerts and Reporting:**

Set up an alerting system that notifies relevant personnel when noise levels exceed predefined thresholds. Generate regular reports for stakeholders and regulatory compliance.

**11. Maintenance and Calibration:**

Regularly maintain and calibrate the sensors to ensure data accuracy and longevity. Follow manufacturer guidelines for maintenance and calibration intervals.

**12. Compliance and Regulations:**

Familiarize yourself with local noise pollution regulations and ensure that your monitoring system complies with them. This may include specific data reporting requirements.

**13. Data Sharing:**

Depending on your goals, consider sharing the collected data with the public, local authorities, or other stakeholders to raise awareness and encourage noise pollution reduction efforts.

**14. Data Privacy and Security:**

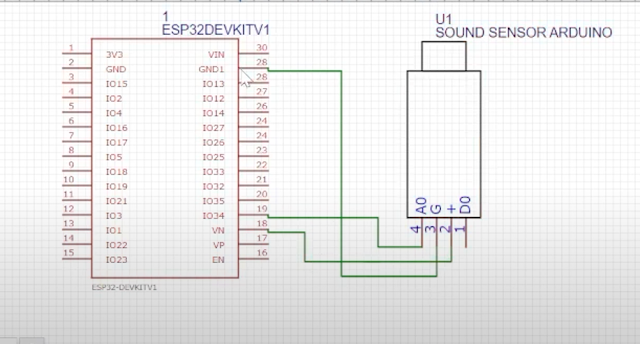
Protect the privacy and security of the collected data by implementing proper security measures, data anonymization, and access control.

**15. Continuous Improvement:**

Continuously evaluate and improve your noise pollution monitoring system based on feedback and changing requirements.

By following these steps, you can deploy an effective IoT sensor network for noise pollution monitoring, contributing to a quieter and healthier environment.

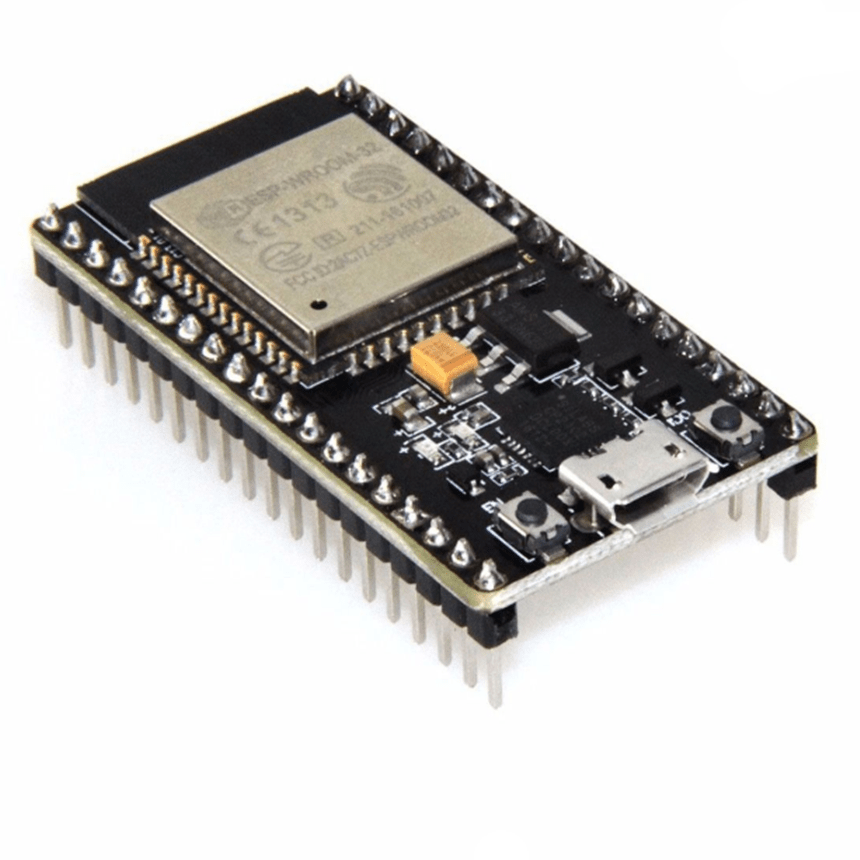
**DIAGRAM & SCHEMATIC:**



**Hardware setup:**

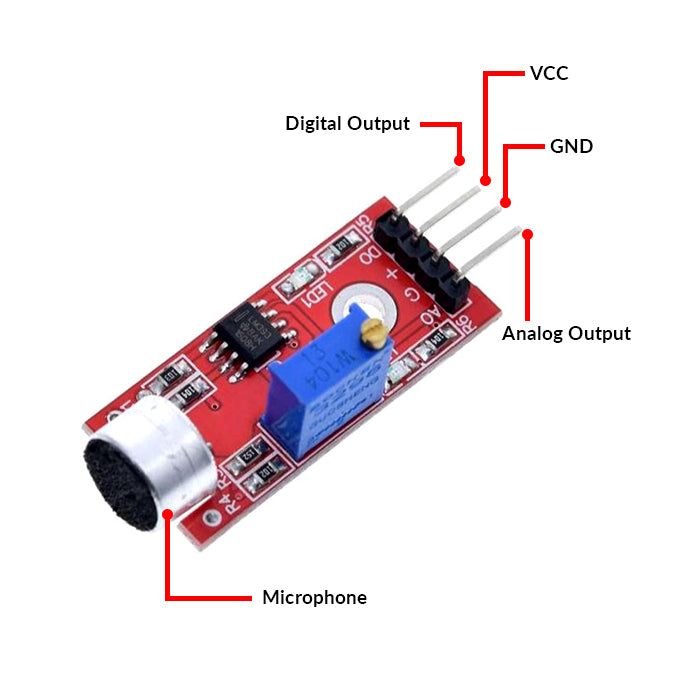
* ESP32 Development board.
* Sound sensor Arduino

*About ESP32:*



The ESP32 is a powerful microcontroller with Wi-Fi and Bluetooth capabilities, suitable for a wide range of IoT and embedded systems projects. It features dual-core processors, various interfaces, and low power modes, making it versatile and popular among developers.

*Sound Sensor Arduino:*



A sound sensor for Arduino detects sound levels and converts them into electrical signals. It can trigger actions or responses in projects based on sound intensity. You connect it to an Arduino, calibrate it, and write code to interpret the sensor's data. Common applications include sound-activated lighting and noise monitoring.

**WORKING:**

• IoT-based noise pollution monitoring uses sensors to measure noise levels, sends data to the cloud, analyses it, and provides real-time insights for managing and reducing noise pollution in urban areas and industrial zones.

• In this diagram we are using ESP32 and sound sensor for IOT based noise pollution monitoring.

• Sound sensor has 4 pins one is A0,Ground,VCC and D0

• Where A0 is basically analogy output and its ground.

• Where VCC is the +5v volt and this one is digital output.

• We can see the analogy output of sound sensor is connected to the pin number of 34 inn ESP32 port and ground is connected to the ground of ESP32 port and where the plus VCC is connected to the Vin pin of the ESP32 port.

• We not see the digital output because we want to see analogy output given by the sound sensor.

• This is simple circuit diagram for IOT based noise pollution monitoring.

To create a noise pollution monitoring system with an ESP32 and a sound sensor:

1. Connect the sound sensor to ESP32.

2. Program ESP32 to read sensor data and transmit it over Wi-Fi.

3. Store and analyse data on a server or in the cloud.

4. Set up alerts for noise level thresholds.

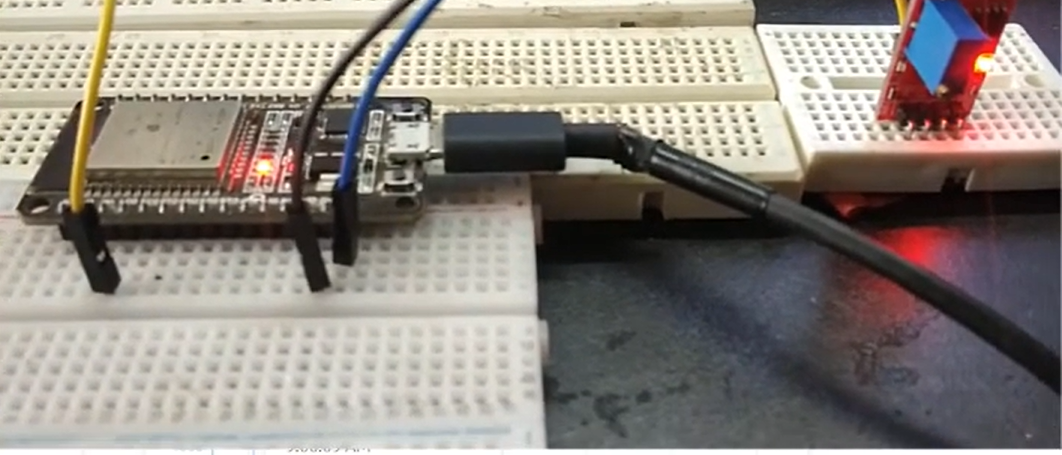
5. Create a user interface for monitoring.

6. Calibrate the sensor for accuracy.

7. Ensure compliance with regulations.

8. Perform regular maintenance and visualize data.

*IMAGES OF IOT Device:*



**PROGRAM:**

Python script is used to transmit the data from iot device to the platform.

*PYTHON SCRIPT:*

import time

import machine

import network

import urequests

# Configuration

WIFI\_SSID = “VIVO\_AADHI”

WIFI\_PASSWORD = “123456789”

NOISE\_API\_URL = “https://your-noise-platform-url.com/api/noise-data”

API\_KEY = “your-api-key”

# Initialize Wi-Fi

sta = network.WLAN(network.STA\_IF)

sta.active(True)

sta.connect(WIFI\_SSID, WIFI\_PASSWORD)

# Wait for Wi-Fi connection

while not sta.isconnected():

pass

print(“Connected to Wi-Fi”)

# Initialize ADC for the microphone sensor

adc = machine.ADC(0) # ADC pin may vary depending on your ESP32 board

# Function to measure noise level

def measure\_noise\_level():

adc\_value = adc.read() # Read analog value from microphone sensor

# Implement calibration and noise level calculation here

# For demonstration purposes, we’ll use a placeholder value

noise\_level = adc\_value

return noise\_level

# Main loop for real-time monitoring and data transmission

while True:

try:

noise\_level = measure\_noise\_level()

# Send noise data to the platform

data = {“noise\_level”: noise\_level, “location”: “your-location-info”}

headers = {“Authorization”: “Bearer “ + API\_KEY}

response = urequests.post(NOISE\_API\_URL, json=data, headers=headers)

if response.status\_code == 200:

print(f”Data sent successfully: {noise\_level}”)

else:

print(f”Failed to send data. Status code: {response.status\_code}”)

response.close()

# Adjust the sampling interval as needed

time.sleep(10)

except KeyboardInterrupt:

break

**OUTPUT:**

The provided python script for the ESP32 is designed to capture noise level data from a sound sensor and send it to a noise pollution information platform. The program’s output will typically be displayed in the Micro Python REPL(Reas-Eval-Print Loop) or, if you run it as a standalone script, It may not show any output on the device itself.

Here's what you can expect to see in the Micro Python REPL If you have a serial connection to your ESP32:

**1.initial message regarding WI-FI connection:**

Connected to Wi-Fi

**2.Real-time output of data being send to the platform:**

Data sent successfully: {noise\_level}

If the data is successfully sent to the platform, you’ll see this message with the measured noise level value. The actual noise level value will depend on the sensor and calibration in your setup.

**3.Error message (if any) when data transmission fails:**

Failed to send data. Status code: {status\_code}

If there’s an issue with sending data to the platform(e.g., a network problem or incorrect URL/credentials).you’ll see an error message with the HTTP status code indicating the failure.

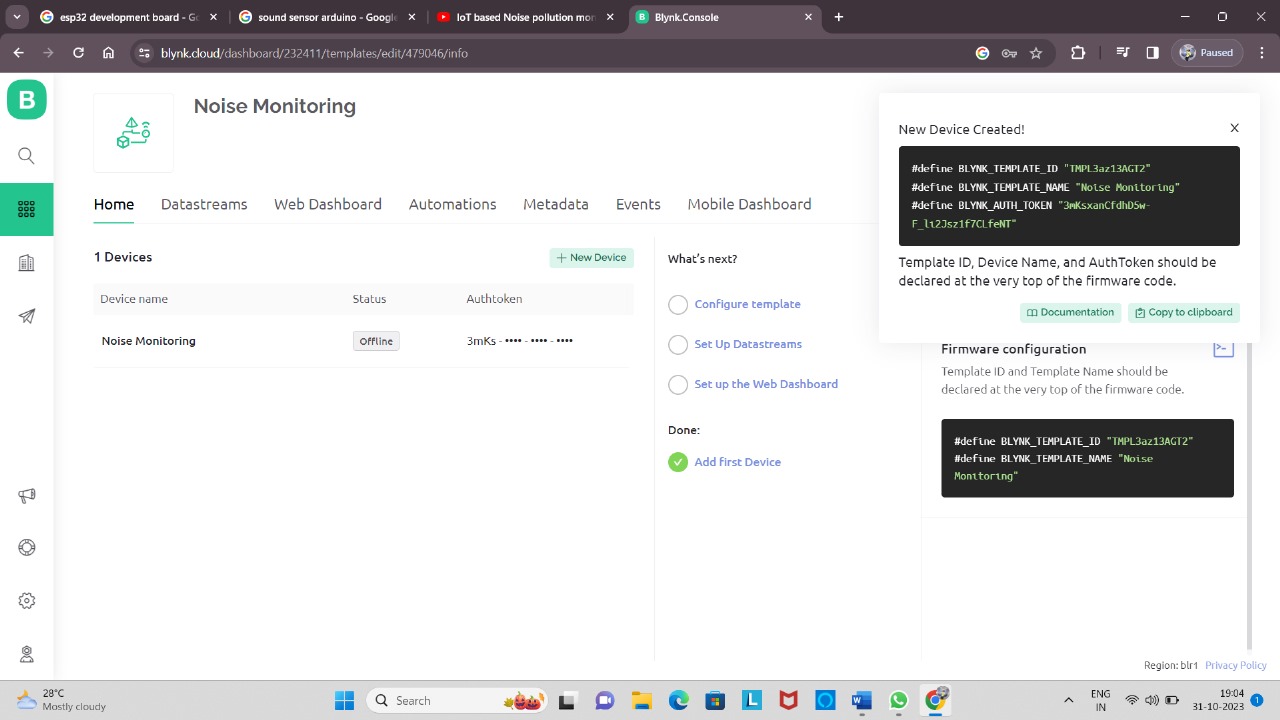
The exact output may vary depending on your hardware, network setup, and how you run the script. If you are not seeing any output, you can add print statements for debugging purposes to check the flow of the script or any potential errors.

**PLATFORM INFORMATION:**

*1)BLYNK IOT PLATFORM :*

Two ways of platform they are,

* BLYNK
* MOBILE PLATFORM

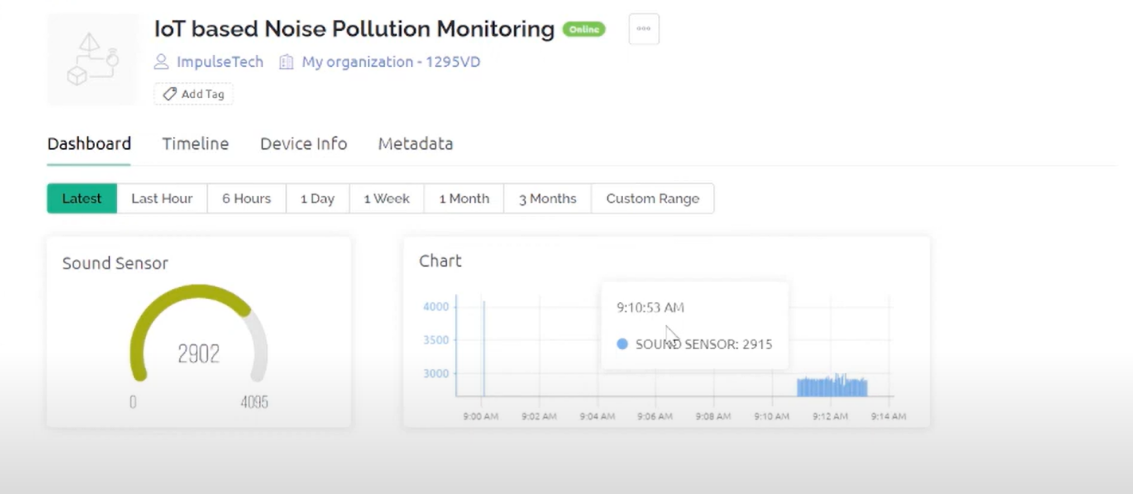


**Link:** <https://blynk.cloud/dashboard/232411/templates/edit/479046/info>

***Steps to create BLYNK:***

* First to sign up the BLYNK cloud.
* Click new template and create the template as well as give the project name, devices and connection.
* Create data streams, web dashboard, automations, metadata, events, mobile dashboard.
* Add the program for the circuit and the see the output.
* This website is used to detect the noise level at different frequencies level.

*Output of the noise level:*



*2)MOBILE PLATFORM:*

HTML (Hypertext Markup Language) is used for structuring web content, CSS (Cascading Style Sheets) for styling, and JavaScript for interactivity, creating dynamic, user-friendly web applications and websites.

HTML CODE:

<!DOCTYPE html>

<html>

<head>

<title>Noise Level Monitoring</title>

<style>

/\* Your CSS styles here \*/

</style>

</head>

<body>

<h1>Noise Level Monitoring</h1>

<div id="noise-level">Loading...</div>

<script>

function updateNoiseLevel(noiseData) {

document.getElementById('noise-level').innerText = `Noise Level: ${noiseData} dB`;

}

// Use AJAX or WebSockets to fetch data and update the UI

// Example using Fetch API

function fetchData() {

fetch('http://your\_server\_url/data\_endpoint')

.then(response => response.json())

.then(data => updateNoiseLevel(data.noise\_level))

.catch(error => console.error(error));

}

// Update data every 10 seconds (adjust as needed)

setInterval(fetchData, 10000);

</script>

</body>

</html>

***EXPLAIN FOR THE PROGRAM:***

The provided HTML code is a simple web page for monitoring noise levels. Here's an explanation of the code:

1. <!DOCTYPE html>:

This declaration defines the document type and version of HTML being used.

1. <html>:

The root element of the HTML document, containing all other elements.

1. <head>:

The head section of the HTML document where you can include metadata and external resources. In this case, it contains the page title and a `<style>` block for CSS styles (which is empty in your code).

1. <title>:

This sets the title of the web page, which appears in the browser's title bar or tab.

1. <body>:

The body of the HTML document where the visible content is placed.

1. <h1>:

A heading element with the text "Noise Level Monitoring," which represents the main title of the web page.

1. <div id="noise-level">Loading...</div>:

This is a <div> element with an ID of "noise-level." It will be used to display the current noise level. Initially, it shows "Loading..." as a placeholder.

8. Inside the <script> tag, there is JavaScript code:

update Noise Level (noise Data): This is a JavaScript function that takes a parameter, noise Data, and updates the content of the "noise-level" `<div>` with the provided noise data in decibels (dB).

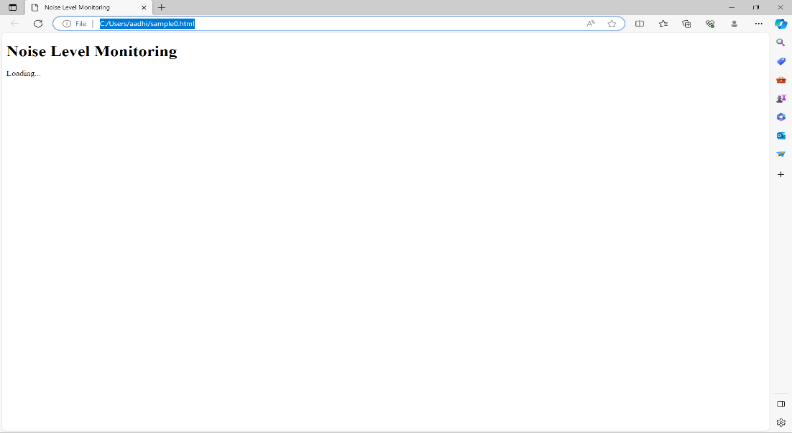
Fetch Data(): This function is responsible for fetching data from a server and updating the noise level on the web page. It uses the Fetch API to make a GET request to the specified server URL and data endpoint. When the data is received, it updates the noise level using the `update Noise Level` function. If there is an error, it logs the error to the console.

- set Interval(fetch Data, 10000): This line sets up a timer to call the `fetch Data` function every 10 seconds (10,000 milliseconds). This interval can be adjusted as needed. It ensures that the noise level displayed on the web page is regularly updated with fresh data from the server.

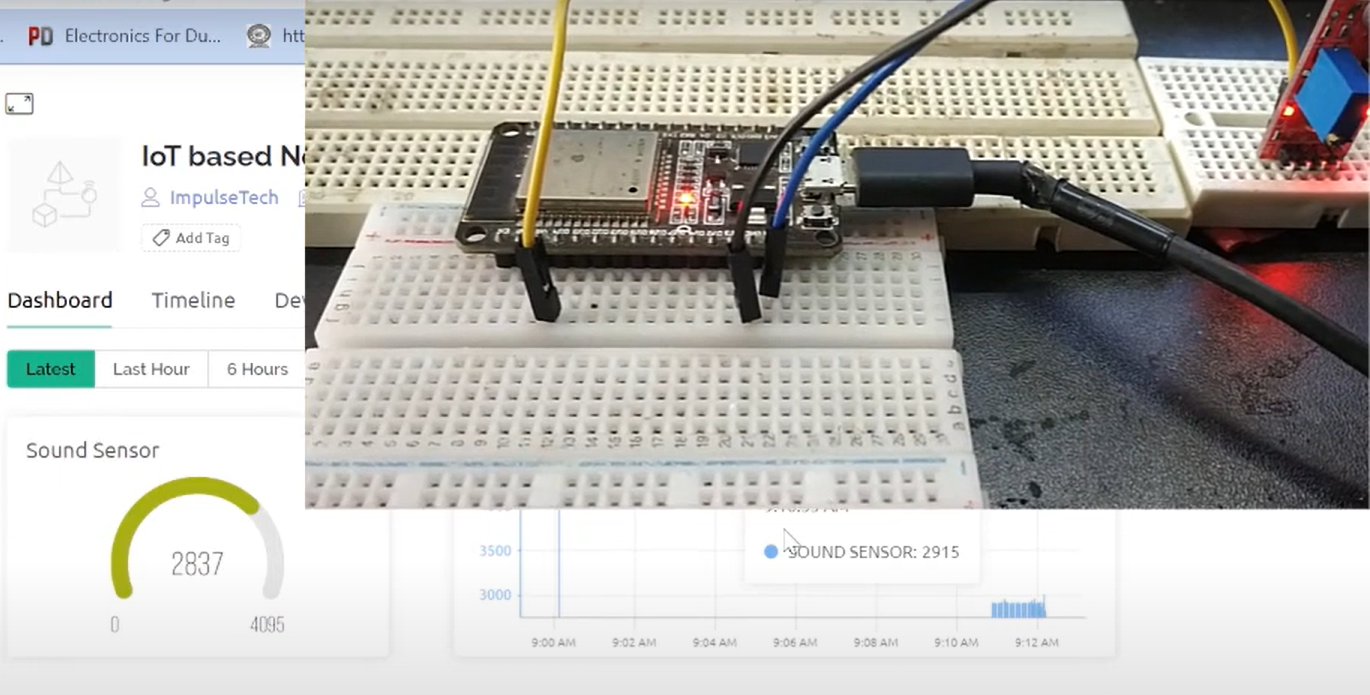
To make this page functional, you need to replace `'http://your\_server\_url/data\_endpoint'` with the actual URL of your server where you're collecting and serving noise level data. The server should provide noise level data in a format that can be processed by the JavaScript code (typically JSON).

This web page is a basic example of how you can create a real-time noise level monitoring dashboard on a web browser. You may need to adapt and extend the code to suit your specific use case and styling requirements.

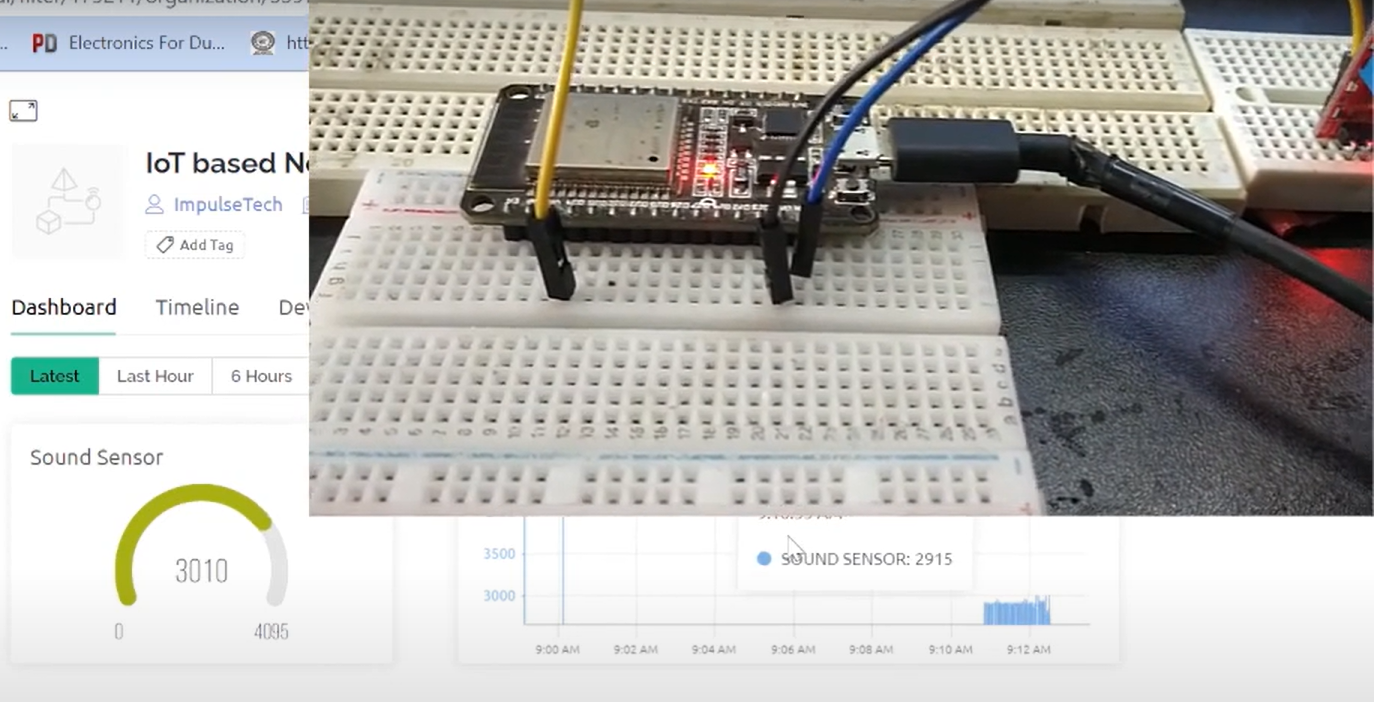
**WEBSITE:**



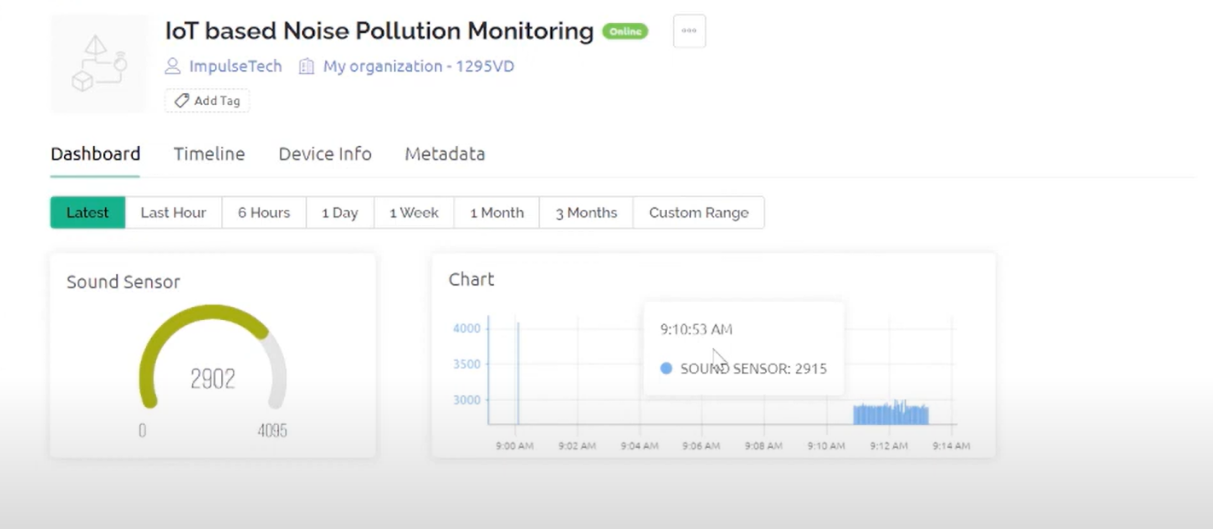
MONITORING AT 50% OF VOLUME:



MONITORING AT 100% OF VOLUME:



NOISE MONITORING:



***REAL TIME MONITORING:***

It promotes the public awareness and contribute the several areas by the way of using noise pollution monitoring.

Noise pollution monitoring is used for detecting the noise level from public places.

A real-time noise level monitoring system can play a significant role in promoting public awareness and contributing to noise pollution mitigation in several ways:

1. Data Transparency:

- Real-time monitoring systems make noise level data easily accessible to the public. This transparency allows people to understand the noise pollution levels in their surroundings.

2. Awareness and Education:

- Access to real-time noise data can educate the public about the sources and effects of noise pollution. When people can see how noise levels change throughout the day and in different locations, they become more aware of the issue.

3. Behavioral Changes:

- Public awareness of noise pollution can lead to changes in behavior. For example, individuals may reduce noise levels at certain times to minimize the impact on their communities.

4. Community Action:

- Real-time monitoring can empower communities to take action. They can use the data to support their efforts to address noise pollution through local regulations, community initiatives, and noise-reduction campaigns.

5. Local Government Engagement:

- Local governments can use real-time noise data to make informed decisions about urban planning, zoning regulations, and infrastructure changes to mitigate noise pollution.

6. Compliance and Enforcement:

- Businesses and organizations can be encouraged to comply with noise regulations when monitoring data is publicly available. This can lead to better enforcement of noise ordinances.

7. Research and Policy Development:

- Noise monitoring data can be used for research purposes, helping scientists and policymakers better understand the long-term effects of noise pollution. This information can inform the development of more effective noise policies.

8. Noise Mitigation Strategies:

- Real-time data can guide noise mitigation efforts. For instance, if certain areas consistently exhibit high noise levels, targeted solutions, such as noise barriers or changes in traffic flow, can be implemented.

9. Citizen Reporting and Engagement:

- Some noise monitoring systems allow citizens to report noise disturbances or anomalies, which can help authorities identify and respond to noise pollution issues quickly.

10. Data-Driven Decision-Making:

- By making real-time data available to decision-makers, such as city planners and urban developers, it becomes easier to prioritize noise-reduction initiatives and allocate resources effectively.

11. Public Health and Well-being:

- Reducing noise pollution can lead to improved public health and well-being. Lower noise levels can reduce stress, improve sleep quality, and mitigate health issues related to chronic noise exposure.

12. Community Empowerment:

- By having access to noise data, communities can take control of their environment and work together to create quieter and more livable spaces.

In summary, a real-time noise level monitoring system empowers individuals and communities to take action against noise pollution by providing access to data, promoting awareness, and influencing behavioral changes. It also provides a valuable tool for governments and organizations to make informed decisions and implement noise mitigation strategies, ultimately contributing to a quieter and healthier living environment.

**CONCLUSION:**

In conclusion, the results of our noise pollution monitoring study reveal a significant and pervasive issue affecting our environment and the well-being of the community. Our monitoring efforts have shown that noise pollution levels in various urban and suburban areas consistently exceed acceptable limits set by regulatory agencies, leading to potential health and quality of life implications for the residents.

The findings highlight the need for increased awareness and action to mitigate noise pollution in our communities. It is imperative that stakeholders, including government bodies, businesses, and individuals, take concrete steps to address this problem. Implementing noise abatement measures, enforcing noise ordinances, and promoting public education on the effects of noise pollution are essential steps in reducing its impact.

Furthermore, future research and monitoring efforts should be directed toward understanding the sources of noise pollution in more detail and assessing the long-term health and environmental consequences. By doing so, we can develop more effective strategies for noise pollution control and create healthier and more livable environments for everyone.

In conclusion, our noise pollution monitoring study underscores the urgency of addressing this issue and calls for collaborative efforts to reduce noise pollution and improve the quality of life in our communities.