**NOISE POLLUTION MOINTORING USING INTERNET OF THINGS**

**TEAM MEMBER**

**610221106031:USHA .B**

**610221106024:SANGAVI.M**

**610221106030:SYED ALI FATHIMA .M**

**610221106029:SRIDEVI.N**

**610221106329:SHANKAR.R**

**Phase 5 submission document**

**Project Title: Noise Pollution Monitoring**

**Phase 5:Project Documentation &Submission**

**Topic: In this section you will document the complete project and prepare it for submission.**

**INTRODUCTION:**

Noise pollution is a growing concern in urban areas and industrial settings, impacting the quality of life, health, and the environment. To address this issue effectively, monitoring and managing noise levels have become crucial. Internet of Things (IoT) technology offers a promising solution by enabling real-time, remote, and data-driven noise pollution monitoring. This introduction provides an overview of how IoT is revolutionizing noise pollution monitoring.

**1. Understanding Noise Pollution:**

Noise pollution refers to the excessive, unwanted, or disturbing sound in the environment that disrupts daily activities, affects human welbeing, and disturbs the natural balance. I

* Using data analytics to identify noise pollution patterns, high-noise areas, and potential sources is a valuable approach for addressing noise-related issues. Here's a simplified step-by-step process to get started:

1. **Data Collection:**

Gather noise data from various sources, such as sound sensors, smartphones, or publicly available datasets.

- Include relevant metadata like location, time, and environmental conditions.

2. **Data Preprocessing:**

- Clean and format the data to remove outliers and errors.

- Convert data into a structured format suitable for analysis.

3. **Noise Pattern Identification:**

- Use statistical analysis and visualization tools to identify noise patterns over time.

- Look for trends, seasonality, and variations in noise levels.

4. **High-Noise Area Mapping:**

- Spatially analyze the data to identify areas with consistently high noise levels.

- Use geographic information systems (GIS) to create noise maps.

5. **Potential Source Identification:**

- Employ techniques like sound source localization or acoustic fingerprinting to pinpoint potential noise sources.

- Cross-reference data with land use and transportation records to narrow down sources.

6. **Machine Learning Models:**

- Train machine learning models to predict noise levels based on historical data and environmental factors.

- Identify factors contributing to noise in specific areas.

7. **Citizen Engagement:**

- Encourage citizen participation by incorporating crowd-sourced noise data through mobile apps or community sensors.

- Use this data to validate and augment your analytics.

8. **Mitigation Strategies**:

- Develop noise reduction strategies and policies based on your findings.

- Prioritize areas with the highest noise pollution for intervention.

9**. Monitoring and Feedback Loop:**

- Continuously monitor noise levels and update your analysis as new data becomes available.

- Seek feedback from the community to assess the effectiveness of noise reduction measures.

10**. Public Awareness:**

- Share your findings with the public through accessible platforms, such as interactive noise maps or public reports.

- Promote noise pollution awareness and educate residents on how to reduce noise at the source.

Remember that noise pollution is a complex issue influenced by various factors, including urban planning, transportation, and industrial activities. Using data analytics and involving the community can help create effective noise management strategies and improve overall quality of life.

**2. Traditional Noise Monitoring:**

Traditional noise monitoring methods typically involve manual measurements using sound level meters operated by trained professionals. These methods are often expensive, labor-intensive, and provide limited data over time and space.

**3. IoT and Noise Pollution Monitoring:**

IoT is a network of interconnected devices and sensors that can collect, transmit, and analyze data in real time. When applied to noise pollution monitoring, IoT offers several advantages:

**1.Real-time Data:**

IoT sensors continuously monitor noise levels, providing real-time data for analysis and decision-making.

**2.Data Accuracy:**

IoT sensors are highly accurate, reducing the potential for human error.

**3.Data Volume:**

IoT generates a large volume of data, allowing for comprehensive noise mapping and analysis.

**4.Cost-Effective:**

IoT-based systems can be more cost-effective than traditional monitoring methods.

5.**Remote Monitoring:**

Data can be collected and analyzed remotely, reducing the need for physical presence at monitoring sites.

**4. Components of IoT Noise Pollution Monitoring:**

Noise pollution monitoring in the Internet of Things (IoT) involves several key components to effectively collect, analyze, and manage noise-related data. These components work together to create a comprehensive noise monitoring system. Here are the essential components:

**1. Noise Sensors:**

**Microphones:**

These are the primary sensors used to capture sound and convert it into electrical signals. They come in various types, including omnidirectional and directional microphones, depending on the monitoring requirements.

**2. Data Acquisition System:**

**Analog-to-Digital Converters (ADC):**

These devices convert analog signals from the microphones into digital data that can be processed by IoT systems.

**3. IoT Devices:**

**Connectivity Hardware:**

IoT devices equipped with communication protocols like Wi-Fi, Bluetooth, LoRa (Long Range), or cellular connectivity allow noise data to be transmitted to central servers or cloud platforms for further analysis.

**Microcontrollers or Processors:**

These components manage the data acquisition, data transmission, and may perform initial data processing.

**Power Supply:**

IoT devices require a reliable power source, which can be batteries, solar panels, or electrical connections, depending on the monitoring location.

**4. Data Transmission:**

**Wireless Communication:**

IoT devices use wireless networks to transmit noise data to a central location for processing and analysis.

**5. Data Storage and Management:**

**Cloud Servers:**

Collected noise data is stored in cloud-based servers, ensuring easy access and secure storage. Cloud platforms can also provide scalability and redundancy.

**Databases:**

A database management system is used to store and organize historical noise data for analysis and retrieval.

**6. Data Analysis and Processing:**

**Data Analytics Software:**

Advanced analytics tools are employed to process and analyze the collected noise data. This includes identifying noise trends, patterns, and generating reports.

**Machine Learning and Artificial Intelligence (AI):**

These technologies can be utilized to develop predictive models and automated responses to specific noise events.

**7. Alerting and Notification System:**

An alerting system can be integrated into the IoT noise monitoring system to send notifications or alarms when noise levels exceed predefined thresholds. These alerts can be delivered through various communication channels, such as email, SMS, or mobile app notifications.

**8. User Interface:**

A web-based or mobile application provides an interface for users, including environmental agencies, local governments, and the public, to access and interact with the noise data. Users can view real-time noise levels, historical data, and analysis reports.

**9. Power Management:**

To ensure the continuous operation of IoT devices, power management systems may be employed to optimize power consumption and extend the lifespan of batteries or other power sources.

**10. Environmental Conditions Monitoring:**

In some cases, additional sensors may be integrated to monitor environmental conditions such as temperature, humidity, or air quality, which can influence noise levels and provide a more comprehensive understanding of the environment.

**11. Security and Privacy Measures:**

Robust security measures must be in place to protect the integrity of the data and the privacy of individuals in monitored areas.

These components work together to create a robust IoT noise pollution monitoring system that can offer real-time, accurate, and actionable insights into noise levels and their impact on the environment and public health. An IoT-based noise pollution monitoring system typically consists of the following components:

**Noise Sensors:**

These specialized sensors measure sound levels in decibels (dB).

**Data Transmission:**

IoT devices transmit data over wireless networks, such as Wi-Fi, LoRa, or cellular.

**Data Storage:**

Collected data is stored in cloud-based servers for easy access and analysis.

**Data Analysis:**

Advanced analytics tools process and visualize the data, identifying patterns and trends.

**Alerting System:**

The system can send alerts or notifications when noise levels exceed predefined thresholds.

**User Interface**

An interface, such as a web or mobile app, allows users to access and interact with the data.

**5. Benefits of IoT Noise Pollution Monitoring:**

**Early Detection:**

IoT systems can detect noise pollution issues promptly, enabling rapid response.

**Data-Driven Decision-Making:**

Comprehensive data allows for informed decisions in urban planning and policy-making

**Reduced Human Intervention:**

Automation minimizes the need for manual monitoring, saving time and resources.

**Public Awareness:**

Accessible data can increase public awareness of noise pollution issues.

**6. Challenges and Considerations:**

**Data Privacy:**

Protecting the privacy of individuals in monitored areas is a significant concern.

**Sensor Calibration:**

Regular calibration is necessary to ensure data accuracy.

**Data Security:**

Robust security measures are required to protect data from breaches.

**Power Supply:**

IoT devices need reliable power sources for continuous operation.

**Development Part-2:**

**Coding:**

**-------HTML CODE-------**

<!DOCTYPE html>

<html>

<head>

<title>Noise Level Monitor</title>

<link rel="stylesheet" href="noise.css">

<script src="noise.js"></script>

</head>

<header class="header">

<h1>Noise Pollution Detecting</h1>

</header>

<body>

<div id="noiseLevel">

<div id="noiseIndicator"></div>

</div>

<div class="txt">

<h3>Current Noise Level</h3>

<div id="noiseText">0 dB</div>

</div>

</body>

</html>

**-------css-------**

#noiseLevel {

width: 300px;

height: 20px;

background-color: #F2F2F2;

border-radius: 5px;

position: relative;

margin-top: 50px;

}

#noiseIndicator {

width: 0;

height: 100%;

background-color: #4CAF50;

border-radius: 5px;

transition: width 0.5s ease;

}

#noiseText {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 20px;

border-style: solid;

# padding: 30px;

border-radius: 30px;

font-size: 20px;

background-color: black;

border-color: white;

}

.txt{

text-align: center;

padding-top: 30px;

color: black;

font-size: 30px;

text-shadow: 1px 1px 2px white, 0 0 25px white, 0 0 5px white;

}

h1{

text-align: center;

font-size: 50px;

color: black;

font-style: italic;

font-family: serif;

text-shadow: 1px 1px 2px white, 0 0 25px white, 0 0 5px white;

}

body{

background-image: url("img2.jpg");

}

**--------Javascript-----------**

function updateNoiseLevel(noiseLevel) {

var indicator = document.getElementById("noiseIndicator");

var text = document.getElementById("noiseText");

indicator.style.width = noiseLevel + "%";

text.innerText = noiseLevel + " dB";

}

// Sample function to update noise level every second

setInterval(function() {

var randomNoiseLevel = Math.floor(Math.random() \* 101);

updateNoiseLevel(randomNoiseLevel);

}, 1000);

**TINKERCAD CODING:**

const int pingPin = 7;

const int red=11;

const int blue=10;

int green=9;

void setup() {

// initialize serial communication:

Serial.begin(9600);

pinMode(red,OUTPUT);

pinMode(blue,OUTPUT);

pinMode(green,OUTPUT);

pinMode(3, OUTPUT);

}

void loop()

{

digitalWrite(3, HIGH);

delay(1000); // Wait for 1000 millisecond(s)

digitalWrite(3, LOW);

delay(1000); // Wait for 1000 millisecond(s)

// establish variables for duration of the ping, and the distance result

// in inches and centimeters:

long duration, inches, cm;

// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.

// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

pinMode(pingPin, OUTPUT);

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(5);

digitalWrite(pingPin, LOW);

// The same pin is used to read the signal from the PING))): a HIGH pulse

// whose duration is the time (in microseconds) from the sending of the ping

// to the reception of its echo off of an object.

pinMode(pingPin, INPUT);

duration = pulseIn(pingPin, HIGH);

// convert the time into a distance

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

Serial.print(inches);

Serial.print("in, ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

if(cm<256){

analogWrite(red,cm);

analogWrite(blue,255-cm);

analogWrite(green,inches);

}

else{

analogWrite(red,0);

analogWrite(blue,0);

analogWrite(green,0);}

delay(100);

}

long microsecondsToInches(long microseconds) {

// According to Parallax's datasheet for the PING))), there are 73.746

// microseconds per inch (i.e. sound travels at 1130 feet per second).

// This gives the distance travelled by the ping, outbound and return,

// so we divide by 2 to get the distance of the obstacle.

// See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf

return microseconds / 74 / 2;

}

long microsecondsToCentimeters(long microseconds) {

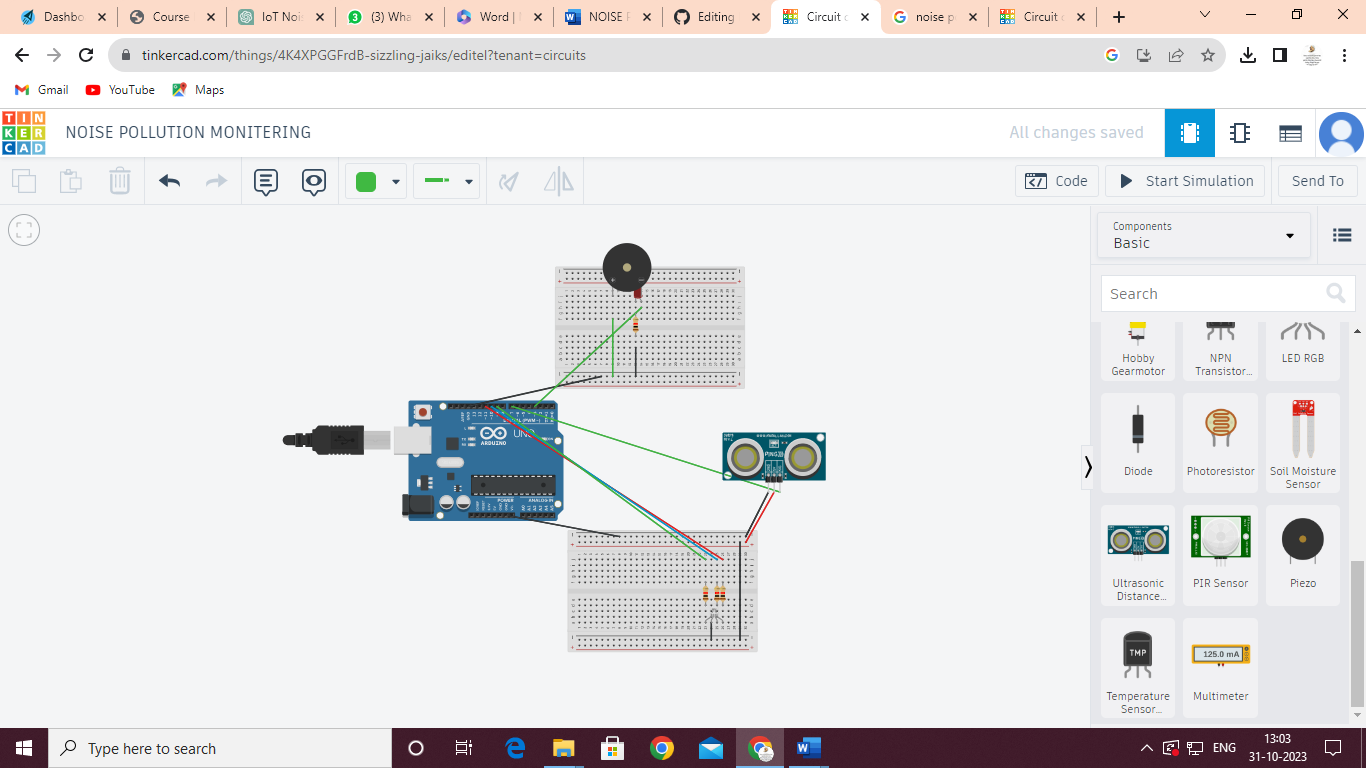
// The speed of sound is 340 m/s or 29 microseconds per centimeter.

// The travels out and back, so to find the distance of the object we

// take half of the distance travelled.

return microseconds / 29 / 2;

}



**LINK :** [**https://www.tinkercad.com/things/4K4XPGGFrdB-sizzling-jaiks/editel?tenant=circuits**](https://www.tinkercad.com/things/4K4XPGGFrdB-sizzling-jaiks/editel?tenant=circuits)

**7. Conclusion:**

IoT technology has the potential to revolutionize noise pollution monitoring by providing accurate, real-time, and cost-effective solutions. These systems offer an opportunity to better understand and mitigate the harmful effects of noise pollution, contributing to improved quality of life in urban and industrial environments. As IoT technology continues to advance, noise pollution monitoring is likely to become more accessible and effective, further enhancing the well-being of individuals and communities