NOISE POLLUTION MONITORING

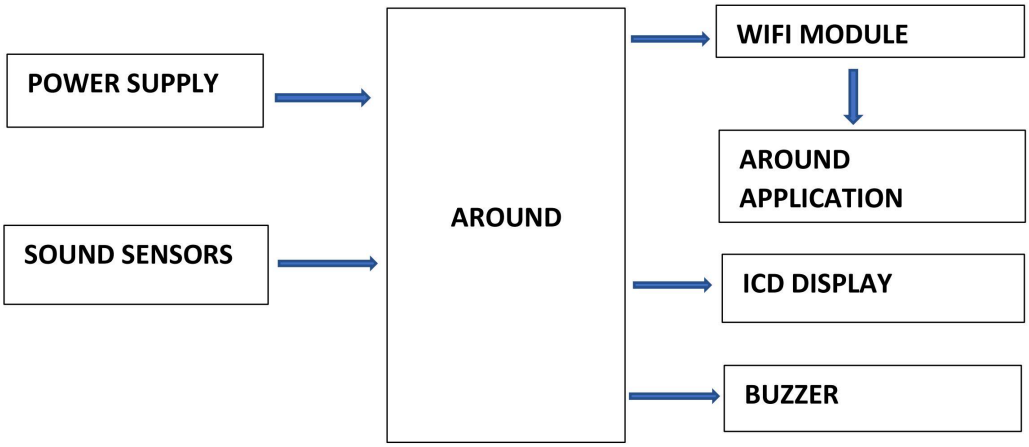
ABSTRACT

Cities and ecosystems worldwide face environmental and technological issues due to sound pollution and a shortage of sound pollution monitoring locations. To solve these issues, the industry has focused its efforts on building a flexible technology solution that allows for improved noise quality assessment and the supply of reference values in network locations where traditional monitoring falls short. Unfortunately, existing items and their outcomes are not low-cost alternatives. IoT has proven itself in a variety of domains by aiding society, including defense, agriculture, safety, comfort, etc. Pollution is steadily increasing these days, having a severe impact on society. Different types of pollution cause different issues.

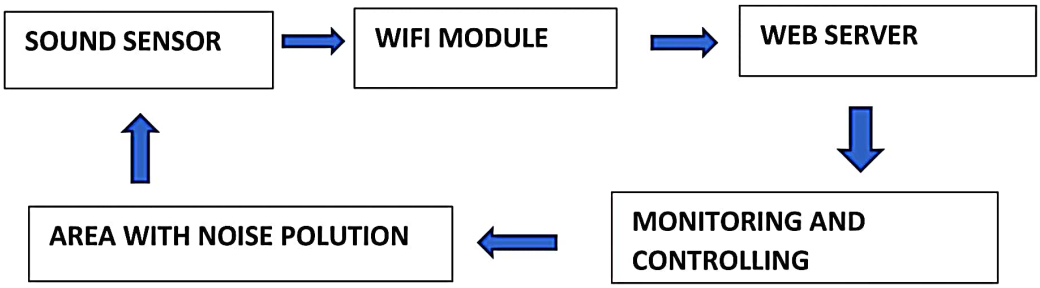
INTRODUCTION

The sensors interact with this data and transmits it over the application. This allows authorities to monitor noise pollution in different areas and act against it. Also, authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas, and if system detects noise issues it alerts authorities so they can take measures to control the issue. Some future consumer applications envisioned for IoT sound like science fiction, but some of the more practical and realistic noise possibilities for the technology include: Receiving warnings on your phone or wearable device when IoT networks detect some physical danger is detected nearby. Self-parking automobiles. The main objective of IOT Noice Monitoring System is that the sound pollution is a growing issue these days. It is necessary to monitor noise quality and keep it under control for a better future and healthy living for all. Due to flexibility and low cost Internet of things (IoT) is getting popular day by day.

BLOCK DIAGRAM:



FLOW CHART:



APPLICATION USING:

- Internet of things
- Sensors application in data collection
- IP network communication

This system is made to fulfill the purpose and need of the society to monitor and check the live sound pollution in an area through IOT. These sensors interact with arduino which processes this data and then transmit it over the mobile application. This system would contribute as a part in the building of a healthy society.

USED COMPONENTS:

- ESP8266 NodeMCU Board
- Microphone sensor
- 16*2 LCD Module
- Breadboard
- Connecting wires

ESP8266 NodeMCU Board:

If you have completed various Arduino projects and are familiar with Arduino, using NodeMCU instead of Arduino Uno is the logical next step if you're looking for a more compact module that encompasses Wi-Fi. NodeMCU is predicated on the Esperessif ESP8266-12E Wi-Fi System-On-Chip. It is based on Lua-based firmware and is open-source.

It's perfect for IoT projects, especially other Wireless connectivity projects as Arduino does not work wirelessly. We either need to connect it to a Bluetooth or nRF module This chip has a great deal in common with the Arduino – they're both microcontroller-equipped prototyping boards that can be programmed using the Arduino IDE. The ESP8266 is more updated and younger than Arduino, and therefore the ESP has stronger specifications than Arduino.

Microphone sensor:

microphone is a sensor or transducer that converts sound (acoustic energy) into electrical energy that we can amplify, digitize, display, record, and more. As with other sensors, there are several types of microphones that are commonly used in sound and noise-measuring applications.

16*2 LCD Module:

An LCD screen is an electronic display module that uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates a display of 16 characters per line in 2 such lines. In this LCD, each character is displayed in a 5×7 pixel matrix.

Breadboard:

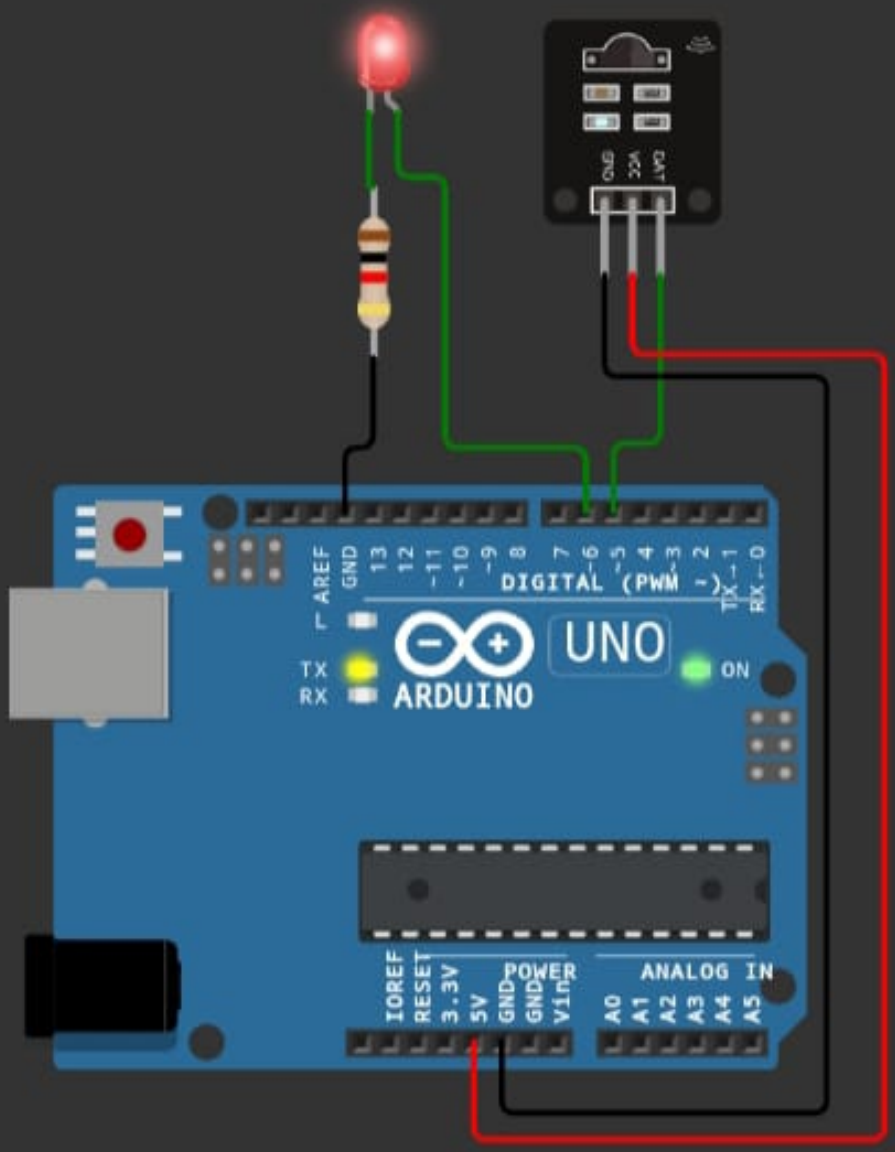
A breadboard (sometimes called a plugblock) is used for building temporary circuits. It is useful to designers because it allows components to be removed and replaced easily. It is useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit.

Connecting wires:

Connecting wires are one of the most important components in an electrical circuit because these are the components through which electricity flows from one electrical component to another. It is with the help of wire that electricity flows from cell to light bulb.

Code Blame

```
import requests
import time
import random


channel_id = "2310756 "
write_api_key = "XQP9UCOC8P7L0YS8 "
thing_speak_url = f"https://api.thingspea


def sound_data():

    return random.uniform(0, 10)
while True:
    try:

      sound = sound_data()


        data = {
            "field1":sound
        }

        response = requests.post(thing_sp
        print("Data sent to ThingSpeak. S

    except Exception as e:
        print("Error:", str(e))

    time.sleep(15)
```


GND
VCC
OUT
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM ~)
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN

Server Side code

```
const express = require('express');
const path = require('path');
const mongoose = require('mongoose');

const app = express();
const port = 3000;

app.use(express.static(path.join(__dirname, 'client', 'public')));

app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'client', 'public', 'index.html'));
});

mongoose.connect('mongodb://localhost:27017/noisedb', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

app.listen(port, () => {
  console.log(`Server is running on)
import React from 'react';

const App = () => {
  return (
    <div>
      {/* Your React components here */}
    </div>
  );
};
export default App;
const mongoose = require('mongoose');
const NoiseSchema = new mongoose.Schema({
  name: String,
```

```
  status:String,
});
constNoise=mongoose.model('Noise',NoiseSchema);

module.exports=Noiseisonlevel;
```

WebPagecode

```
<!DOCTYPEhtml>
<html>
<head>
  <title>Noiselevel</title>
  <style>
    body{
      font-family:Arial,sans-serif;
      background-color:#f5f5f5;
      text-align:center;
      margin:0;
      padding:0;
    }

    h1{
      background-color:#333;
      color:#fff;
      padding:20px0;
      margin:0;
    }

    .container{
      max-width:800px;
      margin:0auto;
      padding:20px;
      background-color:#fff;
      border-radius:5px;
      box-shadow:0010pxrgba(0,0,0,0.1);
    }

    .status{
      display:flex;
      justify-content:space-between;
      align-items:center;
      margin:10px0;
    }

    .label{
      font-size:18px;
      color:#333;
    }

    .availability{
      font-size:18px;
      font-weight:bold;
      color:green;/*Defaultcolorfor'Available'*/
    }
```

```
    .info-box{
      background-color: #f9f9f9;
      padding: 20px;
      border: 1px solid #ddd;
      border-radius: 5px;
      margin-top: 20px;
    }

    .feedback-button{
      background-color: #007BFF;
      color: #fff;
      padding: 10px 20px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
      margin-top: 20px;
    }

    .feedback-button:hover{
      background-color: #0056b3;
    }
  </style>
</head>
<body>
  <h1>Noise level</h1>
  <div class="container">
    <div class="status">
      <div class="label">Noise 1:</div>
      <div class="availability" id="Noise1Status">Loading...</div>
    </div>

    <div class="status">
      <div class="label">Noise2:</div>
      <div class="availability" id="Noise2Status">Loading...</div>
    </div>

    <div class="status">
      <div class="label">Noise 3:</div>
      <div class="availability" id="Noise3Status">Loading...</div>
    </div>

    <!-- Additional content -->
    <div class="info-box">
      <h2>Information</h2>
      <p>This place have is under monitoring by the noise pollution monitoring sesor. So keep your ear buds on your ear
and avoid getting something by the noise pollution</p>
    </div>

    <!-- Feedback button -->
    <button class="feedback-button" onclick="openFeedbackForm()">Provide Feedback</button>
  </div>

  <!-- Firebase setup -->
  <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-database.js"></script>
  <script>
```

```
    var firebaseConfig = {
      apiKey: "YOUR_API_KEY",
      authDomain: "YOUR_AUTH_DOMAIN",
      databaseURL: "YOUR_DATABASE_URL",
      projectId: "YOUR_PROJECT_ID",
      storageBucket: "YOUR_STORAGE_BUCKET",
      messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
      appId: "YOUR_APP_ID"
    };
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);

    // Get a reference to the Firebase Realtime Database
    var db = firebase.database();

    // Function to update the noise status
    function updateNoiseStatus() {
      // Reference to the 'noise' node in your Firebase database
      var noiseRef = db.ref('noise');

      NoiseRef.on('value', function(snapshot) {
        var NoisecontrolData = snapshot.val();
        if (NoosecontrolData) {
          document.getElementById('Noise1Status').textContent = Noise1Data.Noise1;
          document.getElementById('Noise2Status').textContent = NoiseData.Noise2;
          document.getElementById('Noise3Status').textContent = NoiseData.Noise3;
        }
      });
    }

    // Function to open the feedback form
    function openFeedbackForm() {
      document.getElementById("feedback-form").style.display = "block";
    }

    // Periodically update the Noisecontrol status
    updateNoisecontrolStatus(); // Initial update
  </script>
</body>
</html>
```

Webpage:

Conclusion:

In conclusion above program is used to monitor the noise and notify it to the person around the place.