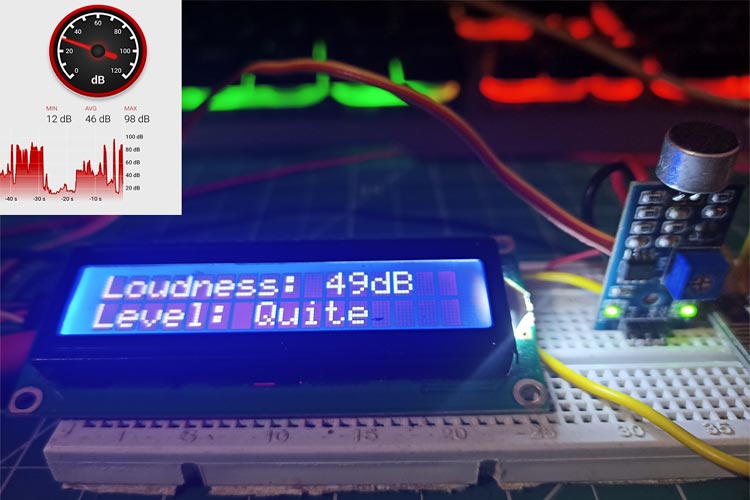
**NOISE POLLUTION**

**Noise pollution refers to the excessive or disturbing noise that may have harmful effects on human health and the environment. It is often considered an underestimated and pervasive problem in many urban areas around the world.**

****

**TEAM MEMBERS:**

|  |  |
| --- | --- |
| **P.THAMARAISELVI** | **732721121056** |
| **M.SHARMILA** | **732721121045** |
| **S.SOWMIYA** | **732721121050** |
| **D.DHARSHSHINI** | **732721121012** |
| **M.MENAKA** | **732721121027** |

**PROBLEM STATEMENT:**

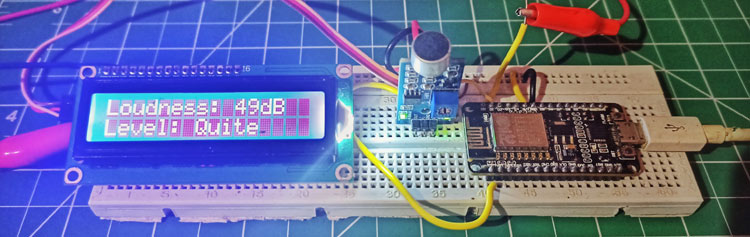
Rapid urbanization and industrialization have led to a significant increase in noise pollution levels in urban areas, posing a serious threat to public health and the well-being of communities. Pervasive sources of noise, including traffic, industrial processes, and recreational activities, have surpassed permissible limits, resulting in chronic exposure that leads to stress, sleep disturbances, hearing impairment, and a range of associated health issues. Moreover, this environmental challenge extends beyond humans, affecting wildlife and disrupting ecosystems. Current mitigation efforts are often fragmented and insufficient to address the multifaceted nature of noise pollution. There is an urgent need for comprehensive strategies, incorporating urban planning, technological innovations, regulatory enforcement, and community engagement, to effectively curb noise pollution and create healthier, more livable urban environments.

INNOVATION:

We surely can’t imagine a world without sound. Sound is one of an integral part of our day to day life, everything just becomes monotonous without the presence of audio. But too much of anything is dangerous, with the advent of automobiles, loudspeakers, etc. sound pollution has become a threat in recent days. So, in this project, we will build an IoT decibel meter to measure sound in a particular place and record the value in a graph using IoT. A device like this will be useful in places like hospitals and schools to track and monitor the sound levels and take action accordingly.

Sound level meters are commonly utilized in sound pollution studies for the quantification of various sorts of noise, especially for industrial, environmental, mining, and aircraft noise. The reading from a sound level meter doesn't correlate well to human-perceived loudness, which is best measured by a loudness meter. Specific loudness may be a compressive nonlinearity and varies at certain levels and certain frequencies.

To make an IoT based decibel meter that will measure the sound in decibels(dB) using a sound sensor and display it to the LCD display along with that, it will also be pushing the readings to the Blynk IoT platform making it accessible from across the world.



COMPONENTS REQUIRED:

ESP8266 NodeMCU Board

Microphone sensor

16\*2 LCD Module

Breadboard

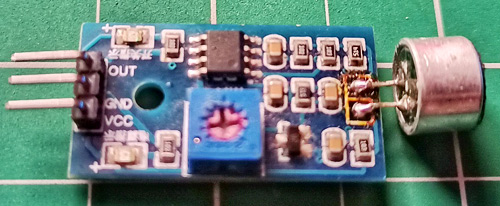
Connecting wires

MICROPHONE MODULE :

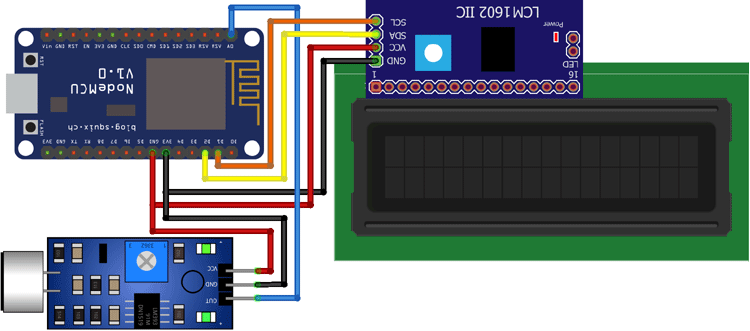
The microphone based sound sensor is used to detect sound. It gives a measurement of how loud a sound is. The sound sensor module is a small board that mixes a microphone (50Hz-10kHz) and a few processing circuitry to convert sound waves into electrical signals. This electrical signal is fed to on-board LM393 High Precision Comparator to digitize it and is made available at the OUT pin.

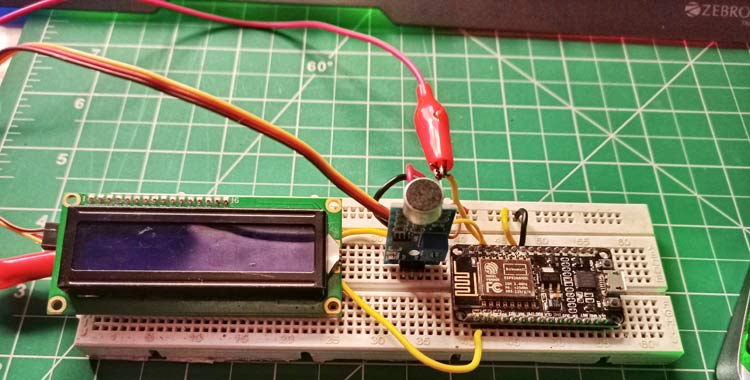
The module features a built-in potentiometer for sensitivity adjustment of the OUT signal. We will set a threshold by employing a potentiometer. So that when the amplitude of the sound exceeds the edge value, the module will output LOW, otherwise, HIGH. Apart from this, the module has two LEDs. The facility LED will illuminate when the module is powered. The Status LED will illuminate when the digital output goes LOW.

The sound sensor only has three pins: VCC, GND & OUT. VCC pin supplies power for the sensor & works on 3.3V to 5V. OUT pin outputs HIGH when conditions are quiet and goes LOW when sound is detected.



CIRCUIT DIAGRAM:





Power pins are connected to the sound sensor and LCD display to 3v3 and GND pin of NodeMCU. Along with that, we have also connected the SCL and SDA pins of the module to D1 and D2 respectively, and the OUT pin of the sound sensor to A0 pin.

PROGRAM:

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <LiquidCrystal\_I2C.h>

#define SENSOR\_PIN A0

LiquidCrystal\_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

const int sampleWindow = 50;

unsigned int sample;

int db;

char auth[] = "IEu1xT825VDt6hNfrcFgdJ6InJ1QUfsA";

char ssid[] = "YourSSID";

char pass[] = "YourPass";

BLYNK\_READ(V0)

{

Blynk.virtualWrite(V0, db);

}

void setup() {

pinMode (SENSOR\_PIN, INPUT);

lcd.begin(16, 2);

lcd.backlight();

lcd.clear();

Blynk.begin(auth, ssid, pass);

}

void loop() {

Blynk.run();

unsigned long startMillis = millis(); // Start of sample window

float peakToPeak = 0; //peak-to-peak level

unsigned int signalMax = 0; //minimum value

unsigned int signalMin = 1024; //maximum value

// collect data for 50 mS

while (millis() - startMillis < sampleWindow)

{

sample = analogRead(SENSOR\_PIN); //get reading from microphone

if (sample < 1024) // toss out spurious readings

{

if (sample > signalMax)

{

signalMax = sample; // save just the max levels

}

else if (sample < signalMin)

{

signalMin = sample; // save just the min levels

}

}

}

peakToPeak = signalMax - signalMin; // max - min = peak-peak amplitude

Serial.println(peakToPeak);

db = map(peakToPeak, 20, 900, 49.5, 90); //calibrate for deciBels

lcd.setCursor(0, 0);

lcd.print("Loudness: ");

lcd.print(db);

lcd.print("dB");

if (db <= 50)

{

lcd.setCursor(0, 1);

lcd.print("Level: Quite");

}

else if (db > 50 && db < 75)

{

lcd.setCursor(0, 1);

lcd.print("Level: Moderate");

}

else if (db >= 75)

{

lcd.setCursor(0, 1);

lcd.print("Level: High");

}

delay(600);

lcd.clear();

}

DISADVANTAGES OF IOT BASED NOISE POLLUTION MONITORING SYSTEM:

While an IoT-based Sound Pollution Monitoring System using NodeMCU has several advantages, it also comes with its share of disadvantages. Here are some potential drawbacks:

Cost: Implementing an IoT-based sound pollution monitoring system can be expensive. It involves the purchase of hardware components like NodeMCU modules, microphones, sensors, and other equipment. Additionally, there may be ongoing costs for maintenance and data storage.

Complexity of Setup: Setting up an IoT system requires technical expertise. Configuring the NodeMCU, connecting sensors, and ensuring proper data transmission can be challenging for individuals without a background in electronics or programming.

Power Consumption: NodeMCU devices require a power source to operate. Depending on the deployment location, providing a continuous power supply may be a challenge. Battery-powered solutions may require frequent recharging or replacement.

Data Security and Privacy Concerns: IoT devices collect and transmit data, which raises concerns about privacy and security. It's crucial to implement strong encryption and security measures to protect sensitive information.

Reliability and Connectivity Issues: The reliability of an IoT system is dependent on network connectivity. If there are network disruptions or signal interference, it can lead to data loss or inaccurate readings. Additionally, if the monitoring location has poor or no internet connectivity, data transmission may be compromised.

Calibration and Maintenance: Sensors used to measure sound levels require periodic calibration to ensure accuracy. Additionally, the equipment may need maintenance or replacement over time, which can incur additional costs.

Limited Coverage: The range of the IoT devices and sensors may be limited, potentially leading to gaps in the monitoring coverage. This is especially relevant in large or complex environments.

Environmental Factors: Environmental conditions, such as extreme temperatures, humidity, or exposure to elements, can affect the performance and lifespan of the monitoring equipment.

Data Overload and Processing: Continuous monitoring generates a large volume of data. Processing and analyzing this data can be resource-intensive, requiring robust backend systems and storage solutions.

Regulatory Compliance: Depending on the location and purpose of the monitoring system, there may be legal and regulatory requirements to meet. Ensuring compliance with local laws and standards can be a complex process.

Limited Customization: Off-the-shelf IoT solutions may have limited customization options. This can be a disadvantage if specific features or functionalities are required for a particular application.

OVERCOME METHOD:

Cost:

Opt for cost-effective components and sensors. Consider open-source alternatives and explore bulk purchasing options. Additionally, seek funding or grants for environmental monitoring projects.

Complex Setup:

Provide comprehensive documentation and tutorials. Offer training for users and technicians. Utilize user-friendly software and interfaces for configuration.

Power Consumption:

Implement power-saving features in the NodeMCU code. Consider using energy-efficient components, and explore alternative power sources like solar panels or rechargeable batteries with energy harvesting capabilities.

Data Security and Privacy Concerns:

Employ strong encryption protocols for data transmission. Implement secure authentication and access controls. Regularly update and patch software to address security vulnerabilities.

Reliability and Connectivity Issues:

Use reliable communication protocols and consider redundancy in data transmission. Employ error-checking mechanisms to ensure data integrity. Implement backup data storage solutions in case of connectivity disruptions.

Calibration and Maintenance:

Develop a regular maintenance schedule for sensor calibration and equipment checks. Provide remote diagnostic tools for troubleshooting. Include self-check mechanisms in the monitoring system.

Limited Coverage:

Strategically position multiple monitoring nodes to ensure adequate coverage. Use mesh networking or repeaters to extend the range of the IoT network. Consider using long-range communication modules or technologies.

Environmental Factors:

Encase sensitive components in weatherproof enclosures. Use sensors and materials rated for the environmental conditions of the deployment location. Implement temperature and humidity monitoring to detect adverse conditions.

Data Overload and Processing:

Employ edge computing techniques to perform initial data processing on the Node MCU itself. Use data aggregation and compression methods to reduce the volume of transmitted data. Implement cloud-based solutions for scalable data storage and processing.

Regulatory Compliance:

Conduct thorough research on local regulations and standards. Ensure that the monitoring system meets or exceeds these requirements. Maintain proper documentation and records for compliance reporting.

Limited Customization:

Consider using a modular design approach, allowing for the integration of custom sensors or components. Use open-source platforms that allow for code customization to meet specific requirements.