NOISE POLLUTION MONITORING USING IOT

INTRODUCTION:

- Noise pollution can cause health problems for people and wildlife, both on land and in the sea. From traffic noise to rock concerts, loud or inescapable sounds can cause hearing loss, stress, and high blood pressure. Noise from ships and human activities in the ocean is harmful to whales and dolphins that depend on echolocation to survive.

- Sound is measured in decibels. There are many sounds in the environment, from rustling leaves (20 to 30 decibels) to a thunderclap (120 decibels) to the wail of a siren (120 to 140 decibels). Sounds that reach 85 decibels or higher can harm a person’s ears. Sound sources that exceed this threshold include familiar things, such as power lawn mowers (90 decibels), subway trains (90 to 115 decibels), and loud rock concerts (110 to 120 decibels).

- Noise pollution impacts millions of people on a daily basis. The most common health problem it causes is Noise Induced Hearing Loss (NIHL). Exposure to loud noise can also cause high blood pressure, heart disease, sleep disturbances, and stress. These health problems can affect all age groups, especially children. Many children who live near noisy airports or streets have been found to suffer from stress and other problems, such as impairments in memory, attention level, and reading skill.

- Noise pollution also impacts the health and well-being of wildlife. Studies have shown that loud noises can cause caterpillars' dorsal vessels (the insect equivalent of a heart) to beat faster, and cause bluebirds to have fewer chicks. Animals use sound for a variety of reasons, including to navigate, find food, attract mates, and avoid predators. Noise pollution makes it difficult for them to accomplish these tasks, which affects their ability survive.

PROJECT DETAILS:

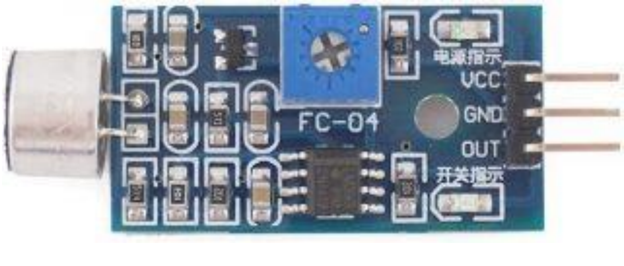
In this IoT project, we will make Decibelmeter using ESP8266 & Sound Sensor.. We will use Nodemcu ESP8266**,** Sound Module **&** Display either 16×2 LCD Display or OLED Display. The Sound Sensor will detect the sound and convert it into an analog voltage which is read by Nodemcu ESP8266. The Nodemcu connects to wifi and uploads the data **to** Thingspeak Server.

COMPONENENTS NEED:

S NO	COMPONENTS NAME	QUANTITY
1	NodeMCU ESP8266	1
2	Sound Sensor	1
3	OLED DISPLAY	1
4	LCD DISPLAY	1
5	CONNECTING WIRES	10
6	BREADBOARD	1

MICRIPHONE SOUND SENSOR:

The microphone sound sensor, as the name says, detects sound. It gives a measurement of how loud a sound is. The sound sensor is a small board that combines a microphone (50Hz-10kHz) and some 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 OUT pin.



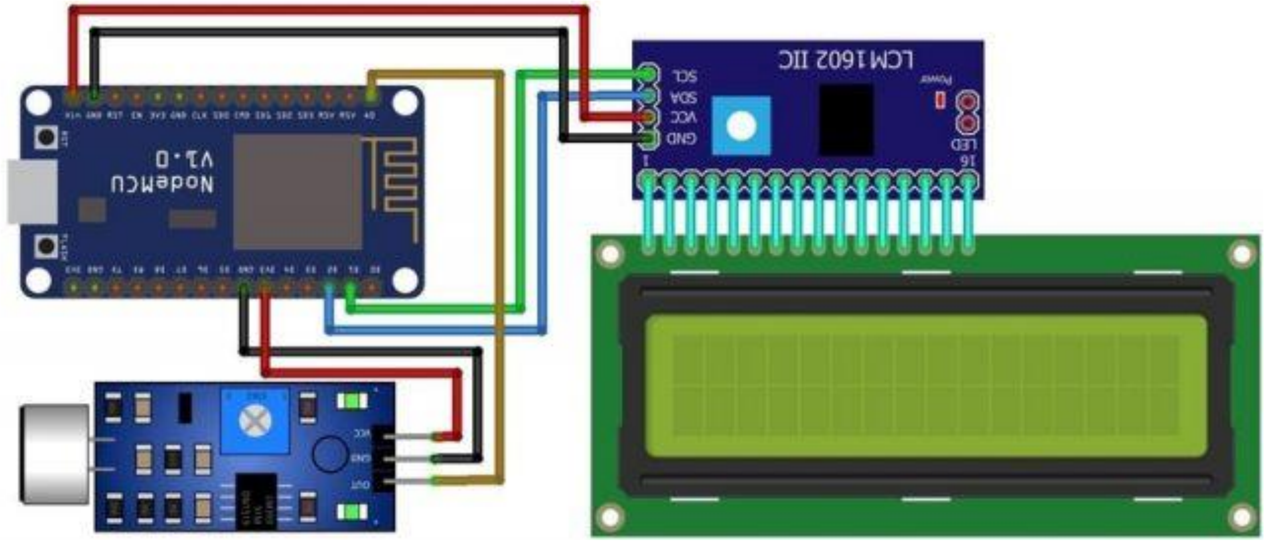
- The module has a built-in potentiometer for sensitivity adjustment of the OUT signal. We can set a threshold by using a potentiometer. So that when the amplitude of the sound exceeds the threshold value, the module will output LOW otherwise HIGH.

- Apart from this, the module has two LEDs. The Power LED will light up when the module is powered. The Status LED will light up 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.

IoT Decibelmeter with ESP8266 & 16×2 I2C LCD Display:

Let us first make Decibelmeter by interfacing Sound Sensor with Nodemcu ESP8266 & I6x2 I2C LCD Display. The connection diagram is given below.



- Connect the Analog output pin of Sound Sensor to ESP8266. Similarly connect the I2C Pins (SDA, SCL) of LCD Display to D2 & D1 of ESP8266. Supply the LCD Display with 5V through Vin Pin. Similarly, supply the Sound Sensor with 3.3V supply through 3.3V Pin. We can also add 3 LED of a different color to Nodemcu D3, D4 & D5 Pins. This LED glows on the basis of different sound intensity.

EXAMPLE PROGRAM:

```
"your_SSID";
const char* password = "your_PASSWORD";
// ThingSpeak Settings
const char* server = "api.thingspeak.com";
const char* writeAPIKey = "your_API_KEY";
// Noise Sensor Pin
const int noiseSens #include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>
// Replace with your network credentials
const char* ssid = orPin = A0;
// Variables
int noiseLevel = 0;
void setup() {
  // Initialize serial communication
  Serial.begin(115200);
  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to WiFi");

  // Initialize ThingSpeak

  ThingSpeak.begin(client);

  // Pin setup

  pinMode(noiseSensorPin, INPUT);

}

void loop() {

  // Read noise level from sensor

  noiseLevel = analogRead(noiseSensorPin);

  // Print noise level to Serial Monitor

  Serial.print("Noise Level: ");

  Serial.println(noiseLevel);

  // Send data to ThingSpeak

  ThingSpeak.setField(1, noiseLevel); // Field 1 is used for noise level

  int response = ThingSpeak.writeFields(writeAPIKey);

  if (response == 200) {

    Serial.println("Data sent to ThingSpeak successfully.");
  } else {
    Serial.println("Error sending data to ThingSpeak. Check your configuration.");
  }
  // Delay for a period before taking another reading
  delay(10000); // 10 seconds
}
```