NOISE POLLUTION MONITORING USING IOT

INTRODUCTION

Noise pollution is one of the major environmental problems that affects the health and well-being of humans and animals. It can cause hearing loss, stress, hypertension, sleep disturbance, and reduced productivity. Therefore, it is important to monitor and control the noise level in different environments, such as urban areas, industrial zones, schools, hospitals, and airports.

One of the possible solutions to monitor noise pollution is to use the Internet of Things (IoT) technology. IoT is a network of interconnected devices that can collect, process, and transmit data over the internet. By using IoT devices, such as microphones, microcontrollers, and Wi-Fi modules, we can measure the noise level in real time and send the data to a cloud server for analysis and visualization. We can also generate alerts and notifications when the noise level exceeds a certain threshold or violates the regulations.

In this paper, we present a noise pollution monitoring system using IoT that can detect and report the noise level in different locations.

COMPONENTS REQUIRED

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

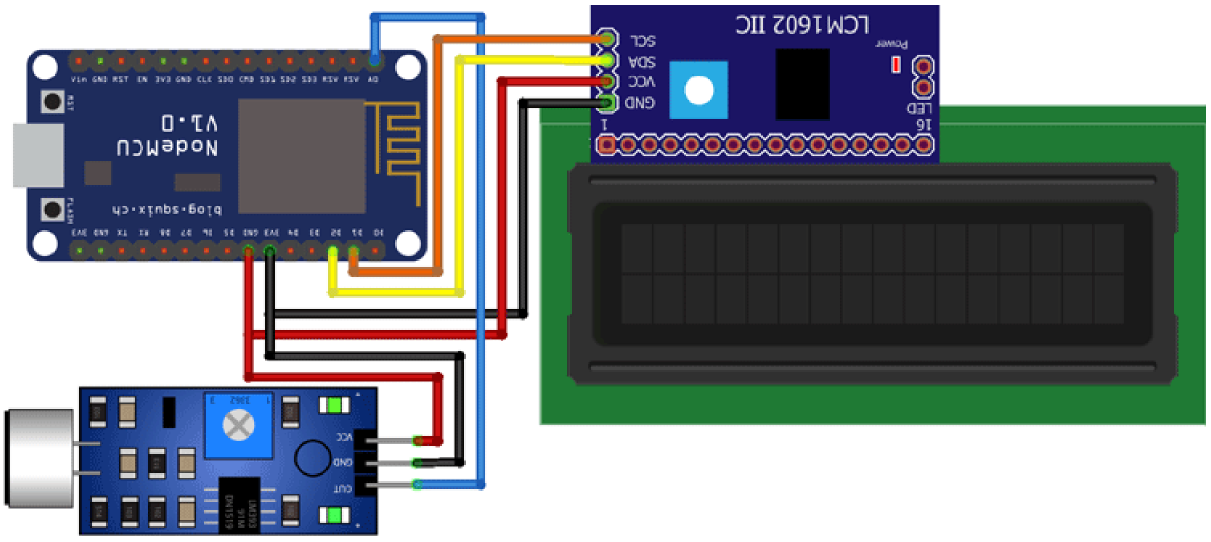
WORKING OF MICROPHONE MODULE

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.

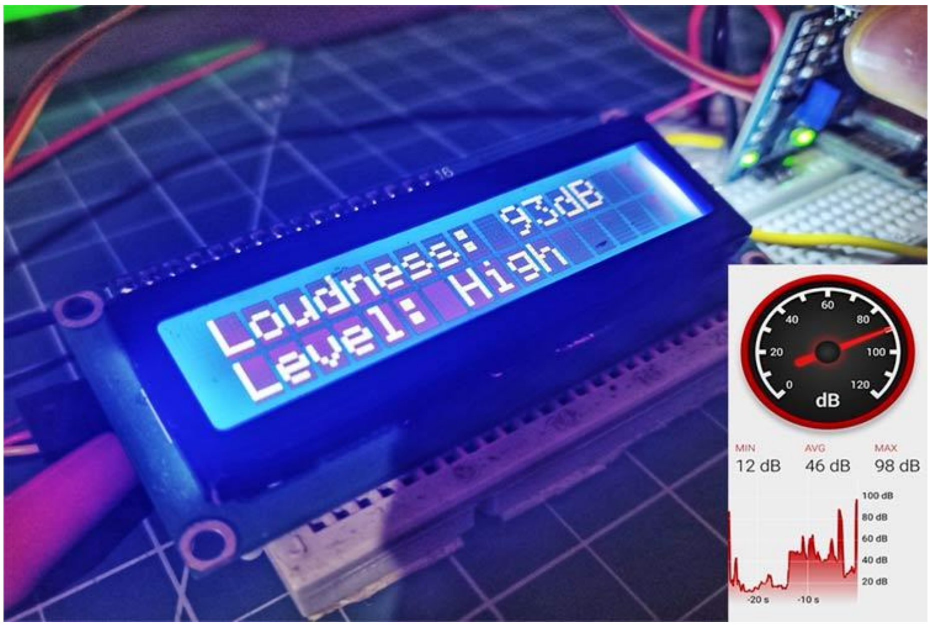
CIRCUIT DIAGRAM

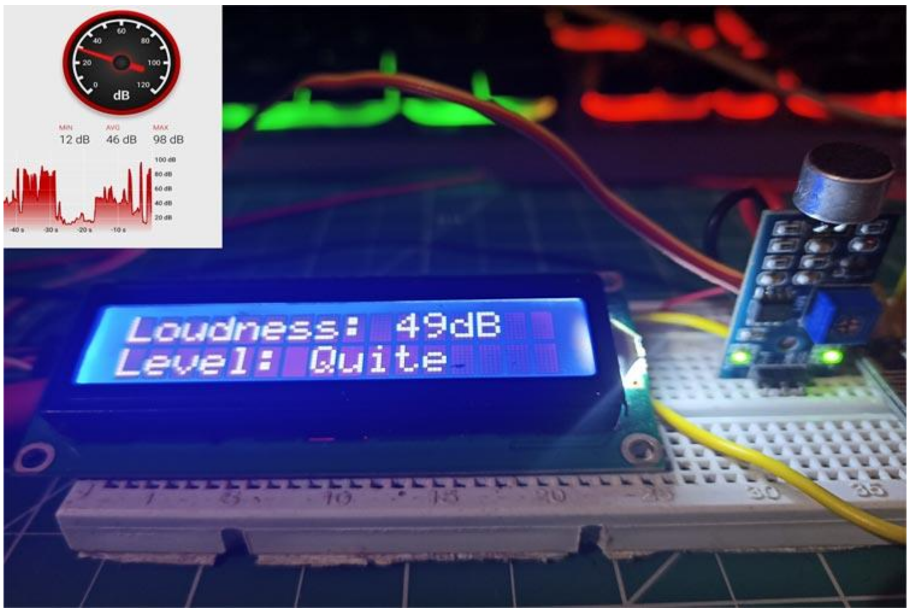


CODE

```
#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[] = "realme 6";
char pass[] = "evil@zeb";
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();
}
```

SAMPLE OUTPUT




dB
MIN
12 dB
AVG
46 dB
MAX
98 dB
Loudness: 49dB
Level: Quite