

Problem statement

The noise pollution is a growing issue these days . As an engineer what is low-cost and effective ways to measure this pollution and encourage community to take certain steps to reduce it to reasonable limit.

In this IoT project, we will create an IoT based noise pollution monitoring system using ESP32, noise module, and Blynk app. We will use the KY-038 microphone sensor to detect noise in decibels and display noise in decibel on an OLED. Additionally, we will link our noise pollution monitor with Blynk application. To make our noise pollution monitor even more practical, users can monitor the sound decibels on the Blynk app. In other words, it becomes extremely handy to monitor the sound from anywhere through a mobile application. We will program our ESP32 board using Arduino IDE.

KY-038 Noise Sensor Module

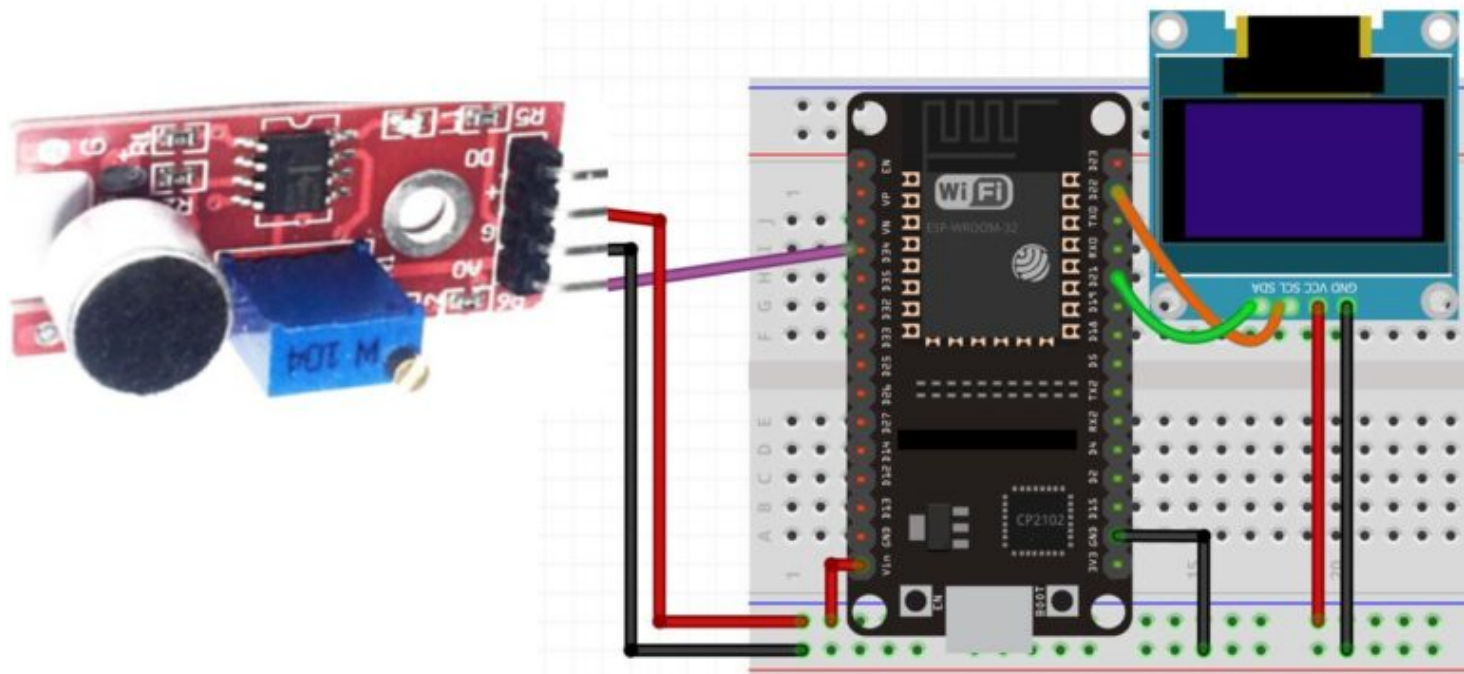
The KY-038 noise sensor module consists of capacitance sensitive microphone (50Hz-10kHz) and an amplification circuit. The module converts noise waves to electrical signals.

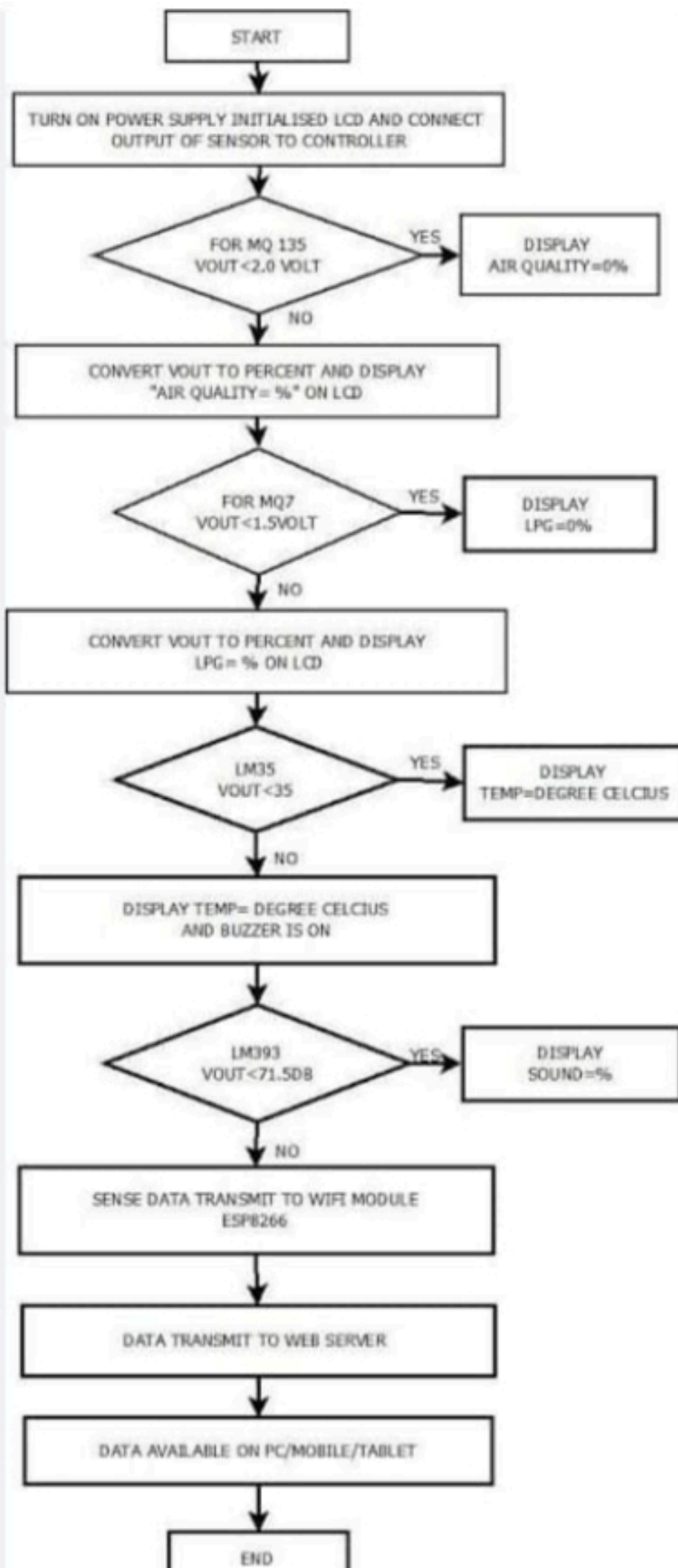
It detects the noise with the help of a microphone and then feeds this noise to processing circuitry which consists of an operational amplifier LM393. It also consists of a potentiometer which is used for setting the noise level and by setting this noise level the output of this noise sensor module could be easily controlled. Similarly, the output of this sensor could be checked by connecting the LED or any other device at output pins.

The OLED display has 4 terminals which we will connect with the ESP32 board. As the OLED display requires an operating voltage in the range of 3.3-5V hence we will connect the VCC terminal with 3.3V which will be in common with the ESP32 board. SCL of the display will be connected with the SCL pin of the module and the SDA of the display will be connected with the SDA of the module. By default, the I2C pin in ESP32 for SDA is GPIO21, and for SCL is GPIO22.

Installing Required Arduino Libraries for IoT noise Pollution Monitoring System

We will use Arduino IDE to program our ESP32 development board. Make sure your Arduino IDE already has the ESP32 plugin installed. To program our ESP32 board for this sound monitor we will be required to install three libraries: BlynkSimpleEsp32.h, Adafruit_SSD1306.h and Adafruit_GFX.h





```
#define BLYNK_PRINT Serial

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>
```

```
#define AO 34
```

```
Adafruit_SSD1306 display = Adafruit_SSD1306(128, 64,
&Wire, -1);
```

```
unsigned int output;
```

```
int Decibels;
```

```
char auth[] = "eO3YD5N52-kdPn3-Ttqu6AfnG0lk****";
```

```
char ssid[] = "YOUR_SSID";
```

```
char pass[] = "YOUR_PASSWORD";
```

```
BLYNK_READ(V0)
```

```
{

  Blynk.virtualWrite(V0, Decibels);

}
```



```
void setup() {  
    Serial.begin(115200);  
    pinMode (AO, INPUT);  
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  
    if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {  
        Serial.println(F("SSD1306 allocation failed"));  
        for(;;);  
    }  
    delay(2000);  
    display.clearDisplay();  
    display.setTextColor(WHITE);  
    Blynk.begin(auth, ssid, pass);  
}  
  
void loop() {  
    Blynk.run();  
    unsigned long start_time = millis();  
    float PeakToPeak = 0;  
    unsigned int maximum_signal = 0; //minimum value  
    unsigned int minimum_signal = 4095; //maximum  
    value
```

```
while (millis() - start_time < 50)
{
    output = analogRead(A0);
    if (output < 4095)
    {
        if (output > maximum_signal)
        {
            maximum_signal = output;
        }
        else if (output < minimum_signal)
        {
            minimum_signal = output;
        }
    }
}
```



```
void loop() {  
  Blynk.run();  
  
  unsigned long start_time = millis();  
  
  float PeakToPeak = 0;  
  unsigned int maximum_signal = 0; //minimum value  
  unsigned int minimum_signal = 4095; //maximum  
  value  
  
  while (millis() - start_time < 50)  
  {  
    output = analogRead(A0);  
    if (output < 4095)  
    {  
      if (output > maximum_signal)  
      {  
        maximum_signal = output;  
      }  
      else if (output < minimum_signal)  
      {  
        minimum_signal = output;  
      }  
    }  
  }  
}
```

```
}display.setTextSize(2);  
display.setCursor(0,10);  
display.print(Decibels);  
display.setTextSize(2);  
display.setCursor(40,10);  
display.print("db");  
display.display();  
if (Decibels <= 50)  
{ display.setTextSize(2);  
  display.setCursor(0,30);  
  display.print("LOW");  
  display.display();  
}
```

```
,  
else if (Decibels > 50 && Decibels < 75)  
{  
    display.setTextSize(2);  
    display.setCursor(0,30);  
    display.print("Moderate");  
    display.display();  
}  
else if (Decibels >= 75)  
{  
    display.setTextSize(2);  
    display.setCursor(0,30);  
    display.print("HIGH");  
    display.display();  
}  
delay(1000);  
display.clearDisplay();
```

THAKING YOU