TITIE : NOISE POLLUTION MONITORING

Submitted by: Vinotha A
Nithyapriya D
Vasuki M
Arthi R

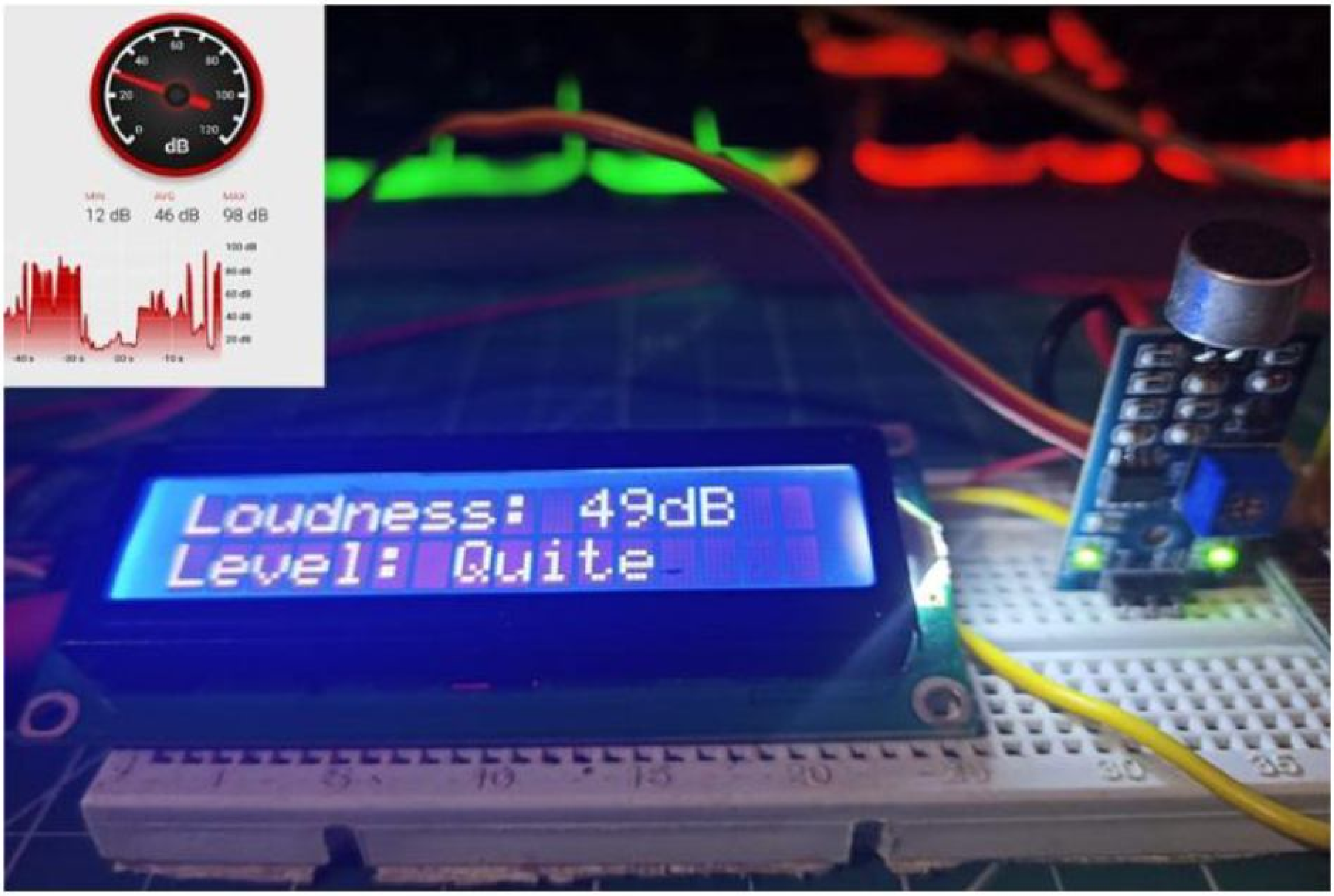
Name of the institution: Ganesh college of engineering

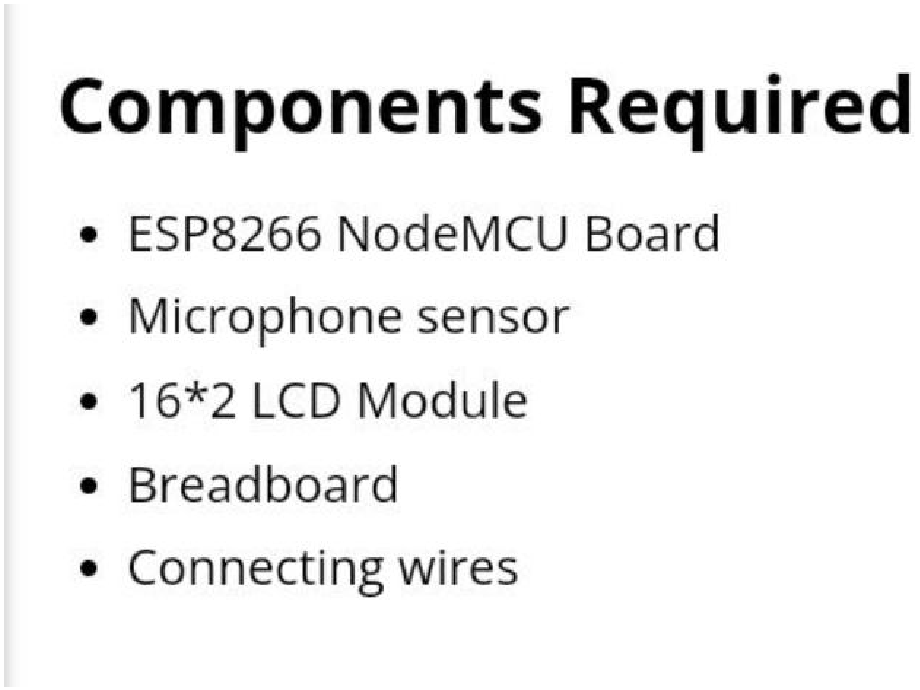
Address of the institution:
Ganesh college of engineering
Attur main road,
Mettupatti(po) valapady(Tk)
Salem Dt

District:salem

State:Tamilnadu pincode:636111

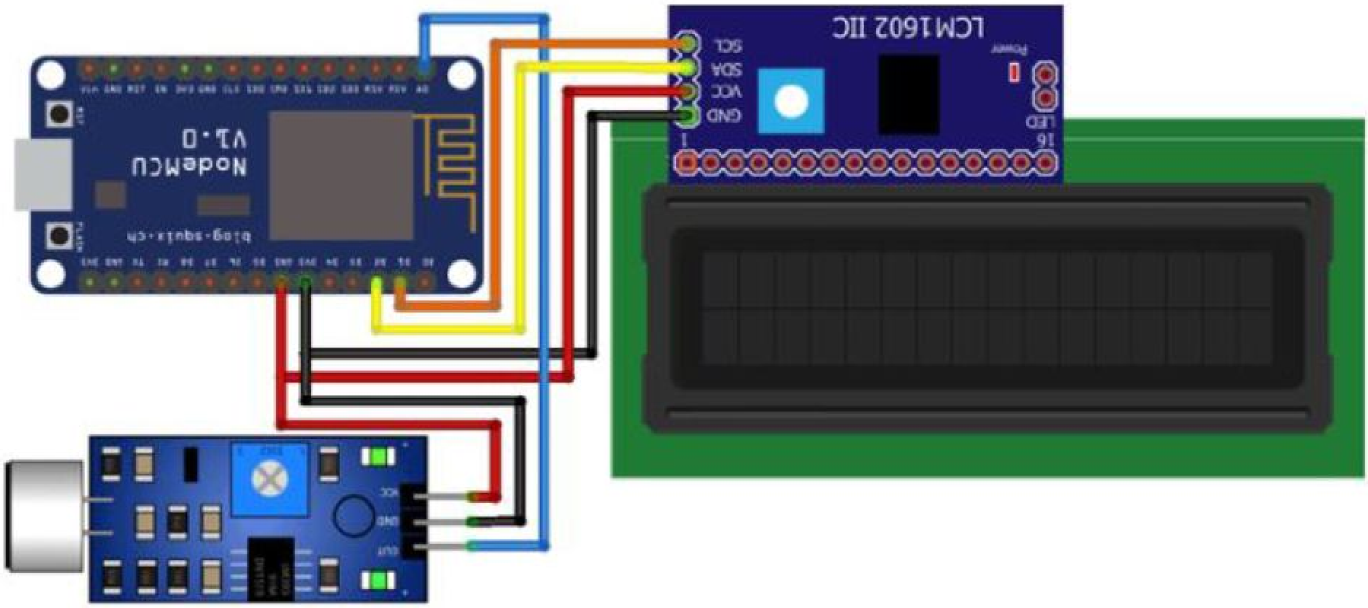
IoT Based Sound Pollution Monitoring System – Measure and Track Decibels (dB) using NodeMCU

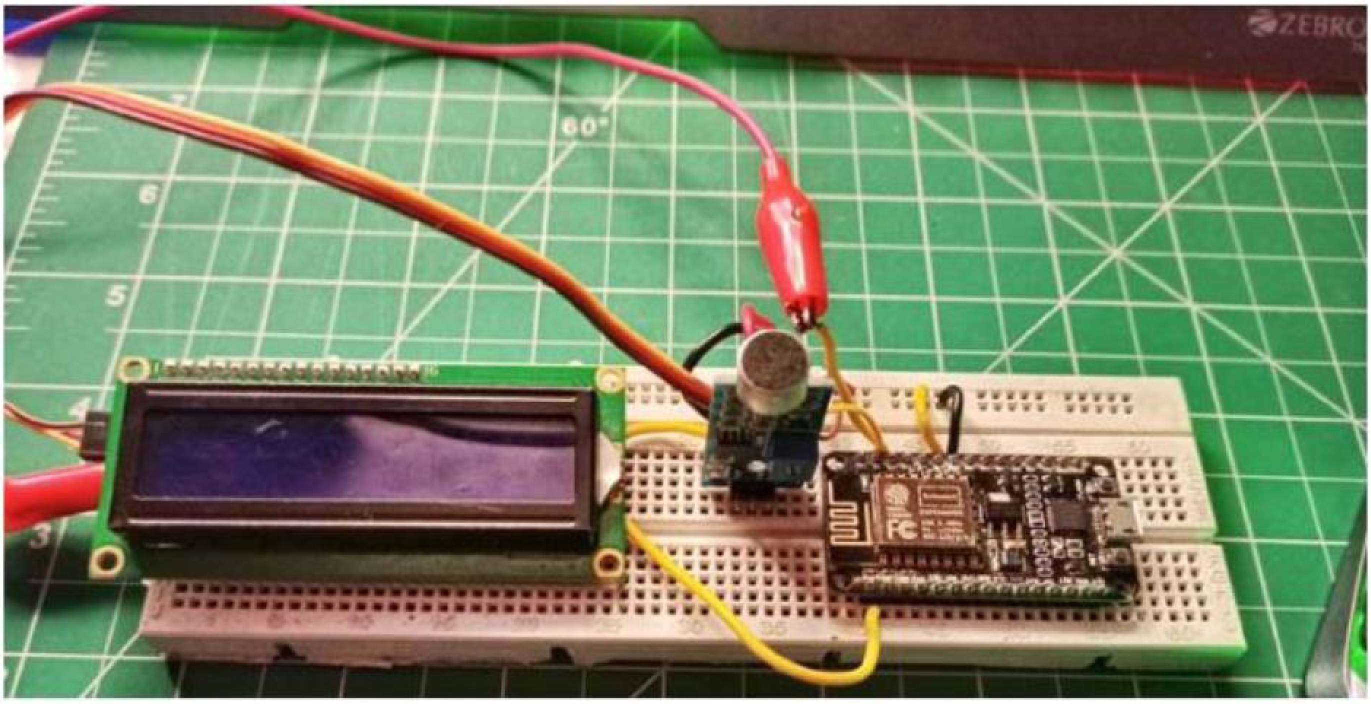




Components Required

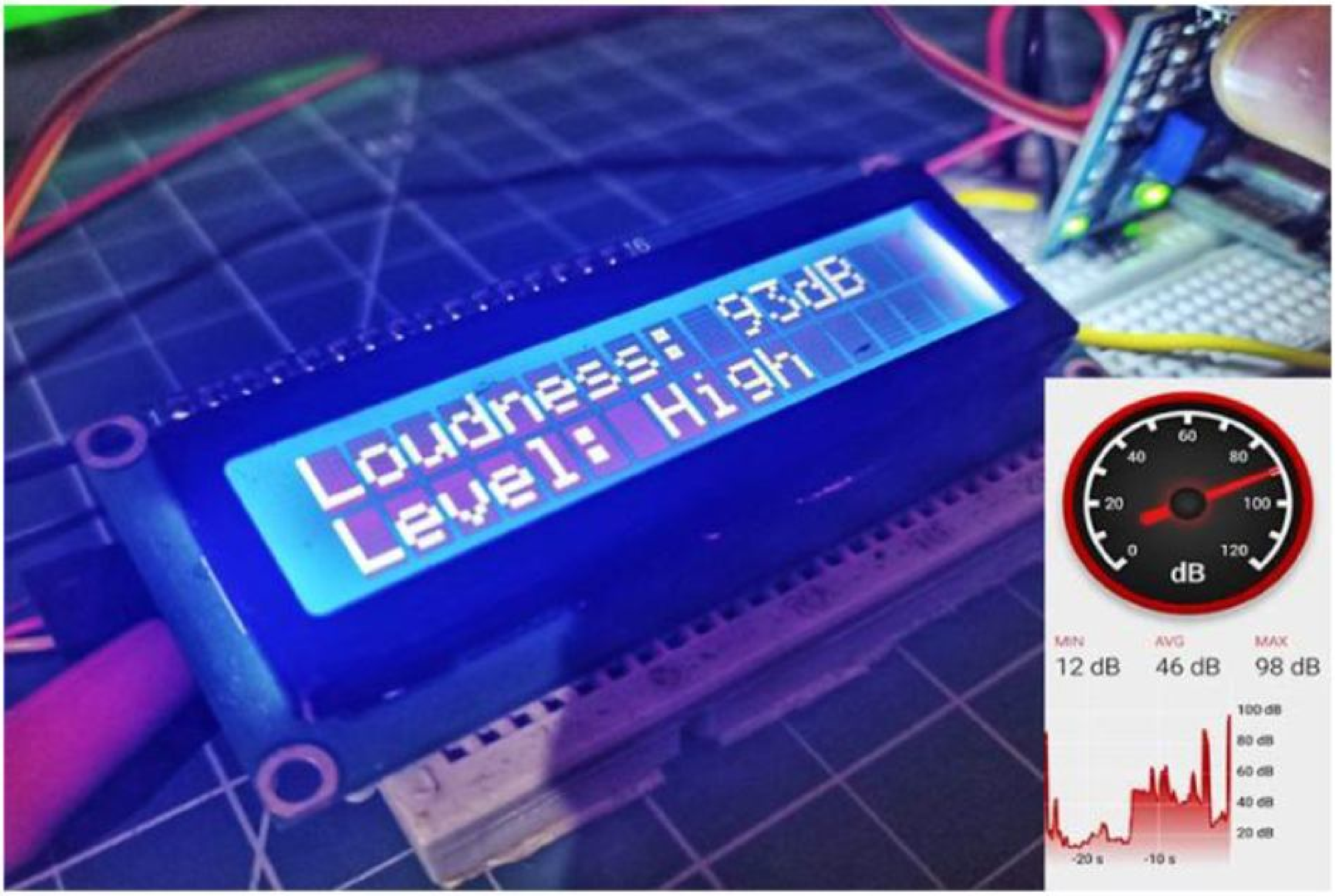
- ESP8266 NodeMCU Board
- Microphone sensor
- 16*2 LCD Module
- Breadboard
- Connecting wires


LCM1602 IIC
SCL
SDA
VCC
GND
NodeMCU
V1.0
blog-squix.ch
VCC
GND
OUT

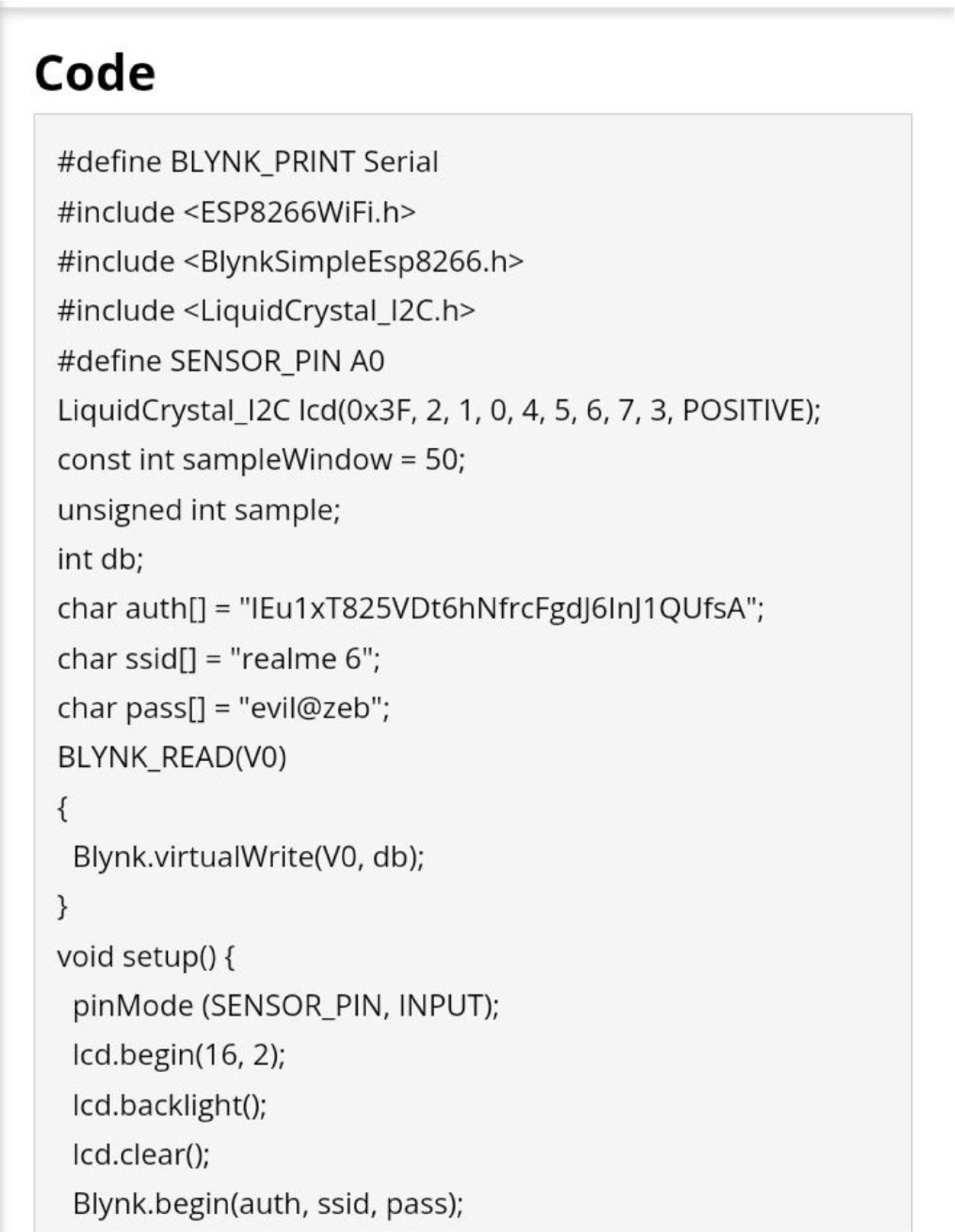


Working of the Project

Now that you have understood the code, you can simply upload it to your NodeMCU board and the project should start working.

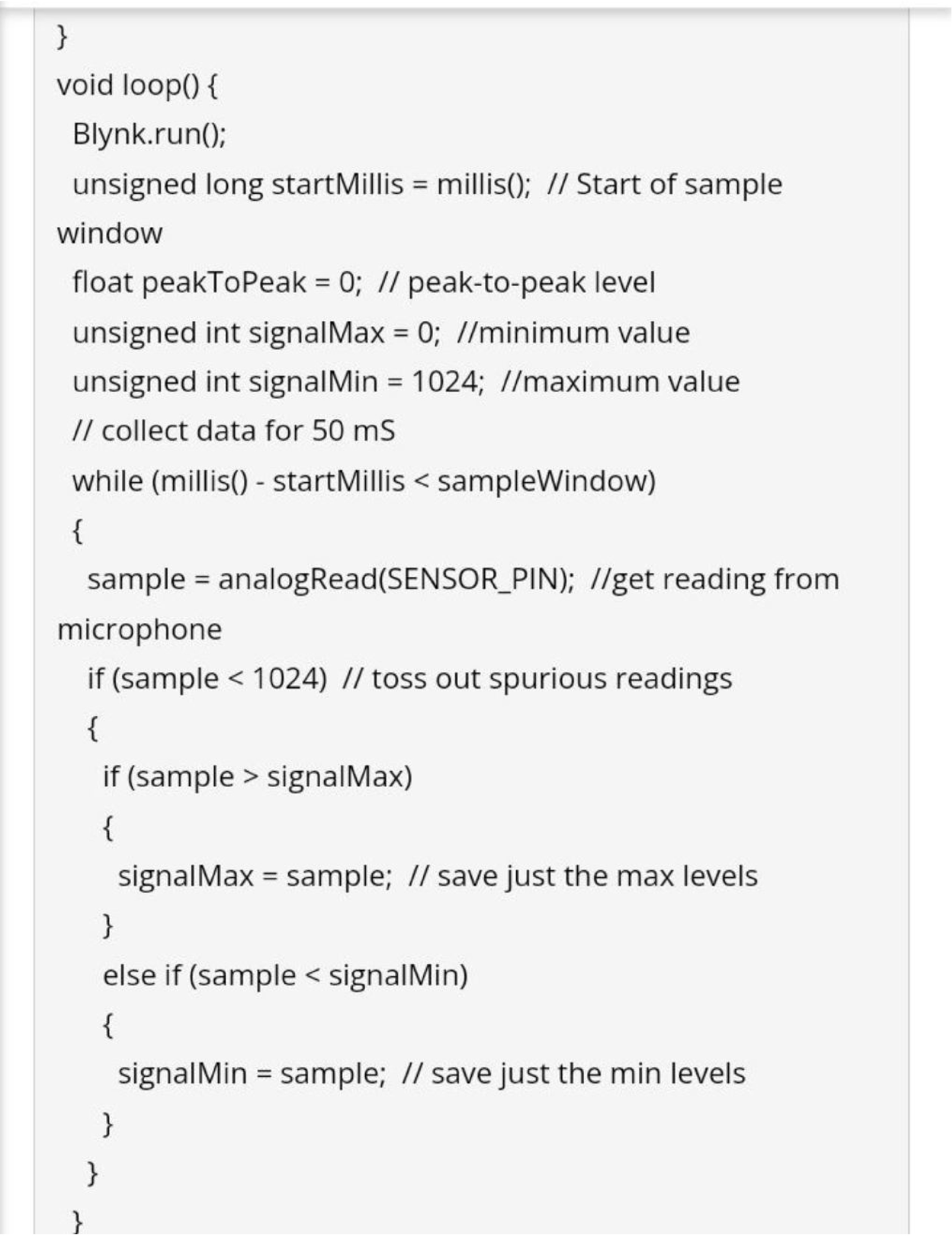


To make sure the values are correct, I compared them to an android application on my phone that could measure sound. As you can see from the pictures, the results were quite close.

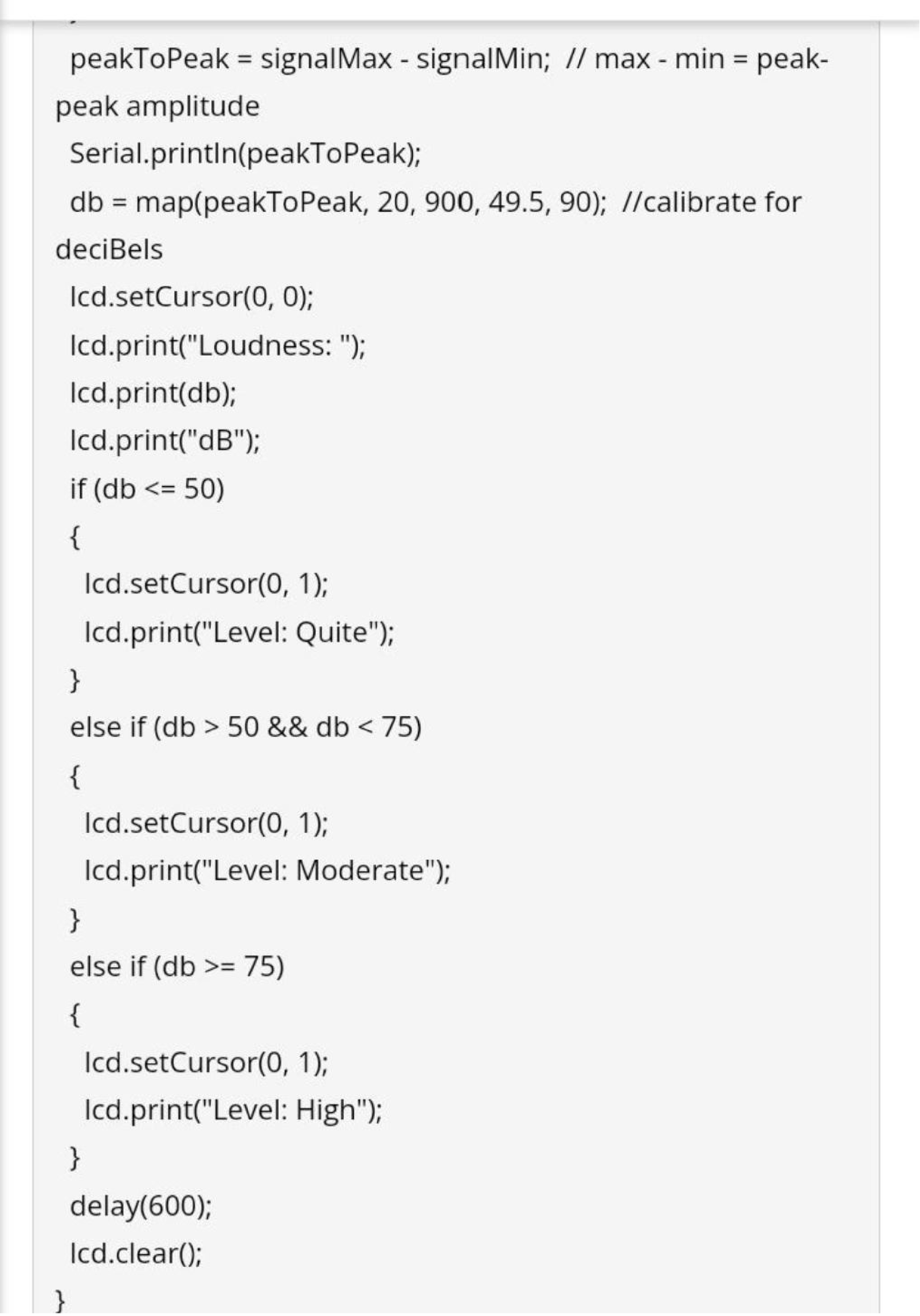


Code

```
#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();
}
```