Decibel Sound Detector

BECE204L - Microprocessors and Microcontrollers

Team Members:

21BCT0140 – P.V. SATWIK KRISHNA

21BCT0366 – SYED NAVED

21BCT0118 – D. GURU CHARAN

21BCI0247 – VAKA RAMAKRISHNA

21BDS0248 – G VENKATA LIKITH

List of Figures

List of tables

Abstract

On the Indian subcontinent and worldwide, people celebrate a number of happy life occasions that call for the lighting of fireworks; yet, some welfare groups of multiple communities have erected a no crackers-inside restriction. Due to the noise and air pollution cracker popping causes, the government frequently alerts the populace to the health concerns involved. At all costs, seniors and young children should refrain from using firecrackers. In India, it is forbidden to use fireworks between the hours of 10 p.m. and 6 a.m. The selling of firecrackers has drastically diminished, which has had a very favourable influence for the government. For the company, it represents a sizable loss, nevertheless.

Our project aims to reduce the noise pollution caused by firecrackers near hospitals, schools, and other significant public locations, to lower the cost of production and the quantity of chemicals used to make crackers, and to develop a tool that will enable people to live in a secure environment free from harm. All the requirements outlined in the law that the Indian government has enforced regarding firecracker noise standards are to be met by the manufacturer.

NodeMCU ESP8266

The **NodeMCU ESP8266 development board** comes with the ESP-12E module containing the ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.

NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

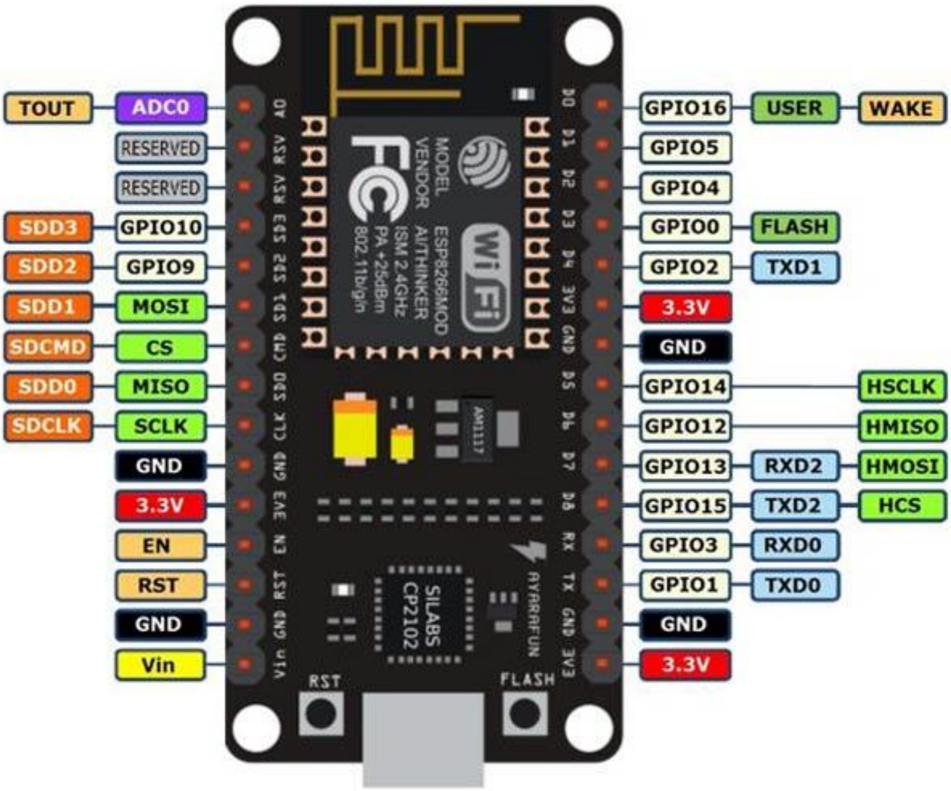


Fig 1: ESP8266 NodeMCU

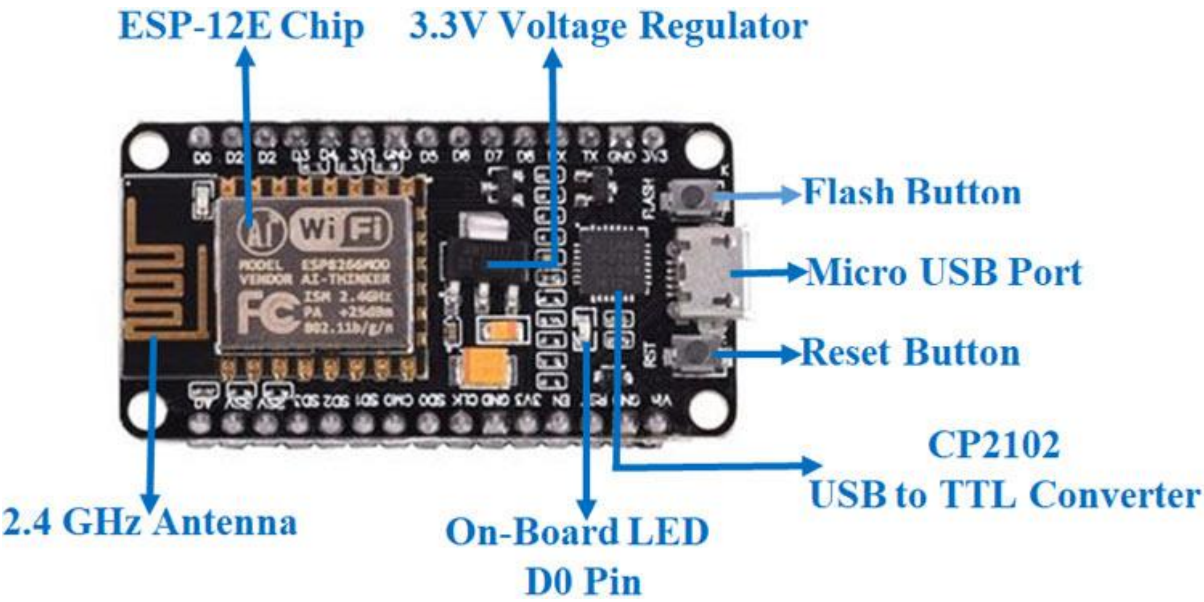


Fig 2: Pins of ESP8266

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

LM393 Sound Detection Sensor Module

LM393 IC

LM393 Comparator IC is used as a voltage comparator in this Sound Detection Sensor Module. Pin 2 of LM393 is connected to Preset (10KΩ Pot) while pin 3 is connected to Microphone. The comparator IC will compare the threshold voltage set using the preset (pin2) and the Microphone pin (pin3).

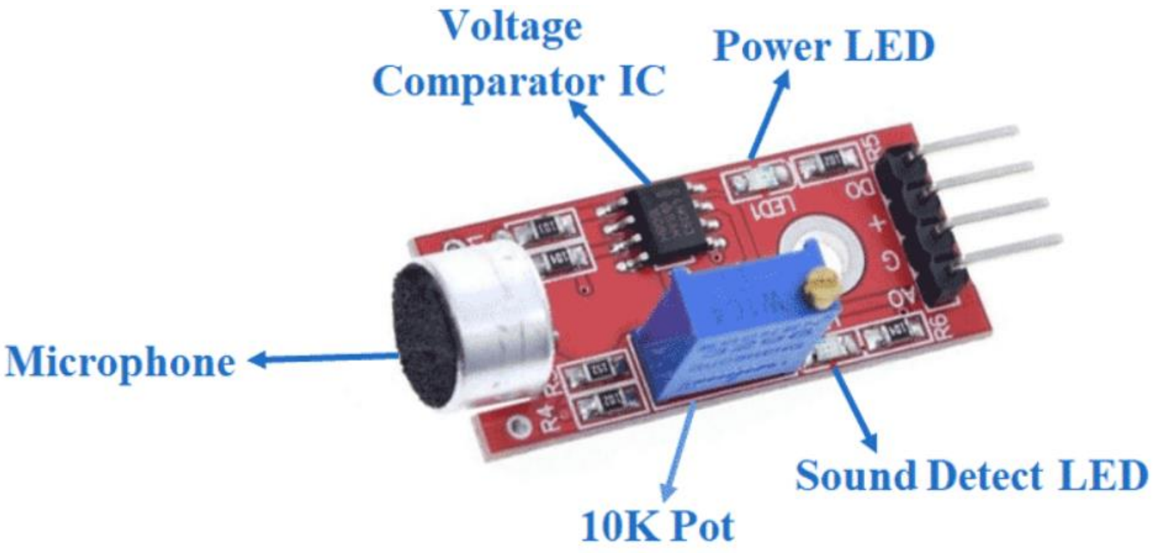


Fig 3: LM393 Sound Sensor

Sound Detection Sensor Module Features & Specifications

- Operating Voltage: 3.3V to 5V DC
- LM393 comparator with threshold preset
- PCB Size: 3.4cm * 1.6cm
- Induction distance: 0.5 Meter
- Operating current: 4~5 mA
- Microphone Sensitivity (1kHz): 52 to 48 dB
- Easy to use with Microcontrollers or even with normal Digital/Analog IC
- Small, cheap and easily available

Decibel Sound Detector

Objective:

Our project is aimed at assisting the manufacturer in meeting all the requirements outlined in the law that the Indian government has enforced regarding firecracker noise standards, to lessen the noise pollution caused by crackers near hospitals, schools, and other significant public locations, to lower the cost of production and the quantity of chemicals used to make crackers, and to develop a tool that will enable people to live in a secure environment free from harm.

Motivation:

The government frequently informs the general public of the health risks associated with popping crackers due to the noise and air pollution they produce. Senior citizens and little children should avoid using firecrackers at all costs. A sudden exposure to loud noises, such as those from fireworks, can result in heart attacks, high blood pressure, sleep difficulties, and temporary or permanent deafness. Fireworks are not permitted in India between the hours of 10 p.m. and 6 a.m. The government has seen a very favourable impact of this restriction, and the sale of firecrackers has significantly decreased. However, it is a significant loss for the firm. We chose this subject for our study in order to assist the government and manufacturers.

Introduction:

People celebrate several joyful life events on the Indian subcontinent and abroad that call for the lighting of fireworks, yet some welfare organizations of numerous communities have put up a no-crackers-inside restriction. People who enjoy popping crackers are controlled. The government frequently informs the general public of the health risks associated with popping crackers due to the noise and air pollution they produce. Senior citizens and little children should avoid using firecrackers at all costs. A sudden exposure to loud noises, such as those from fireworks, can result in heart attacks, high blood pressure, sleep difficulties, and temporary or permanent deafness. Fireworks are not permitted in India between the hours of 10 p.m. and 6 a.m. The government has seen a very positive impact of this restriction, and the sale of firecrackers has significantly decreased. However, it is a significant loss for the firm.

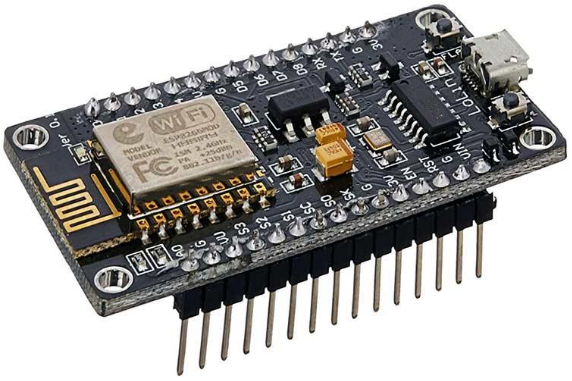
According to the 89th Section of the Environment Protection Act, 1986's firecracker noise guidelines, it is illegal to manufacture, sell, or use firecrackers that produce noise levels of 125 dB (Al) or 145 dB (C) or more at a distance of 4 meters from the place of bursting. The market's selection of firecrackers exceeds the allowed noise limitations. For a single cracker, the noise limit is 125 db. The tested firecracker's maximum noise level was 146.8 db. Even when the relevant officials take action against fire-cracker vendors who set up shop in congested places, they do not monitor the noise level of firecrackers because they lack a suitable system for accomplishing this.

Our project is intended to assist the manufacturer in meeting all the requirements outlined in the law that the Indian government has enforced regarding firecracker noise standards, to lessen the noise pollution caused by crackers near hospitals, schools, and other significant public locations, to lower the cost of production and the quantity of chemicals used to make crackers, and to develop a tool that will enable people to live in a secure environment free from harm.

High sensitivity microphones are used to pick up ambient sound. Voltage is the measurable value. The controller receives the voltage and converts it to the corresponding dB value. Once the conversion is complete, a Bluetooth module sends the value to a nearby mobile device. Here, HC-05 series Bluetooth is being used, allowing for two-way communication. Utilizing IoT technology, data from the mobile device is uploaded to the cloud. The benefit of using Bluetooth is that the user can see the sound value from a distance, preventing them from coming into contact with the cracker as it is bursting. Additionally, anyone can access the readings from the database by using the mobile application.

Hardware Components

Table 1: Hardware components used for decibel sound detector

Components	Image
ESP8266 NodeMCU	
Display	

Microphone Sensor	
Jumper Wires	

ThingSpeak Module

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. ThingSpeak enables sensors, instruments, and websites to send data to the cloud where it is stored in either a private or a public channel. ThingSpeak stores data in private channels by default, but public channels can be used to share data with others. We can send data to ThingSpeak from our devices, create instant visualizations of live data, and send alerts using web services like Twitter and Twilio.

IFTTT Module

IFTTT helps connect different apps and devices. When you sign up for a free account, you can enable your apps and devices to work together to do things they couldn't otherwise do. For example, you can back up your Instagram photos to Dropbox, have your lights turn on when you enter your home, or automatically remind a Slack channel about a meeting.

Components

1. ESP8266 NodeMCU

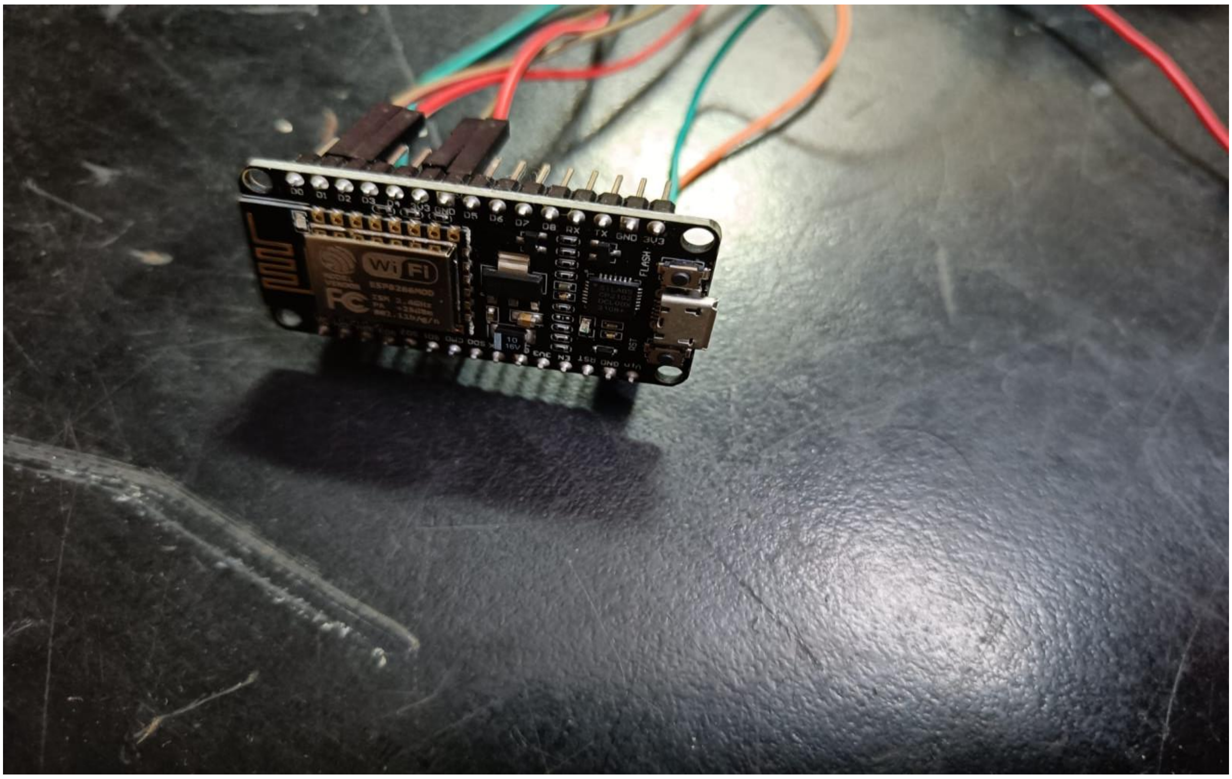


Fig 4: ESP8266

The NodeMCU (*N*ode *M*icro*C*ontroller *U*nit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

2. Display

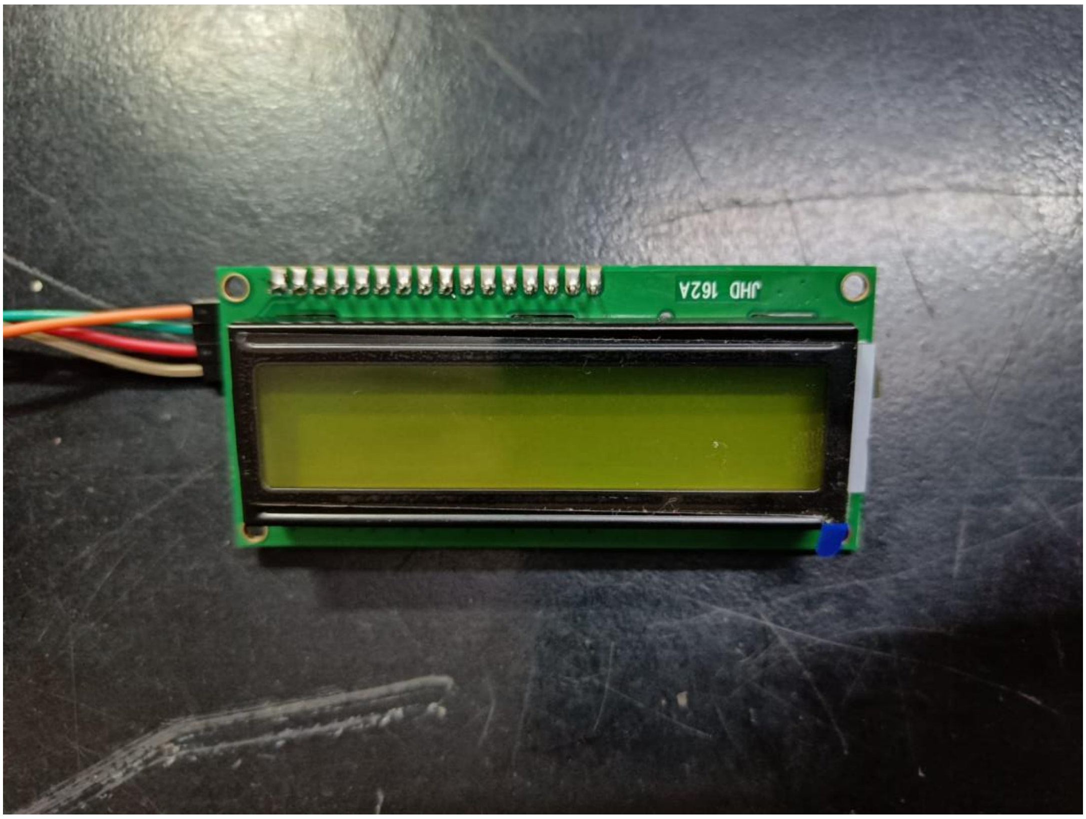


*Fig 5: 16*2 LCD Display*

Display units are the most important output devices in embedded projects and electronics products. The 16x2 LCD is one of the most used display units. 16x2 LCD means that there are two rows in which 16 characters can be displayed per line, and each character takes 5X7 matrix space on LCD.

3. Sound Sensor

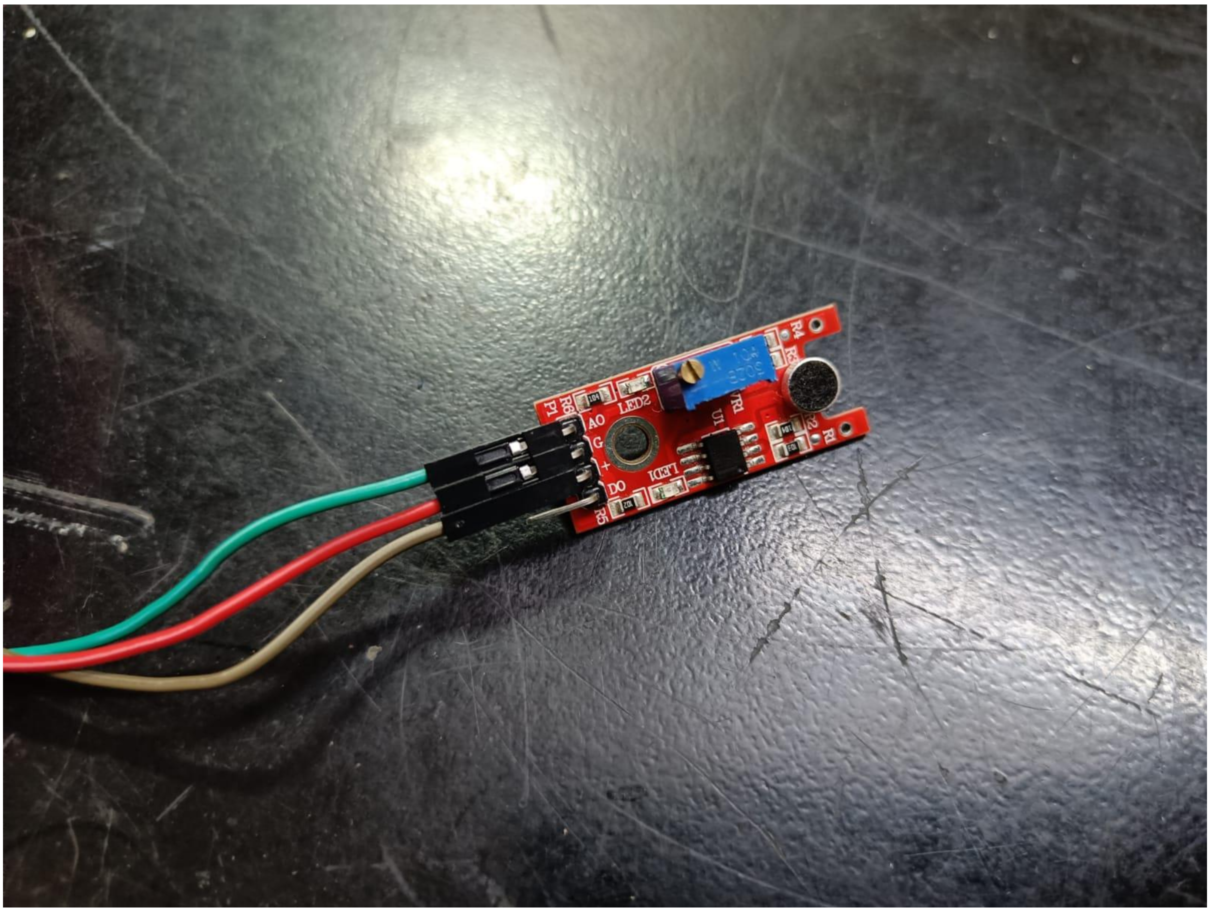


Fig 6: Sound Sensor

Sound detection sensor works similarly to our Ears, having diaphragm which converts vibration into signals. However, what's different as that a sound sensor consists of an in-built capacitive microphone, peak detector and an amplifier (LM386, LM393, etc.) that's highly sensitive to sound.

Circuit diagram

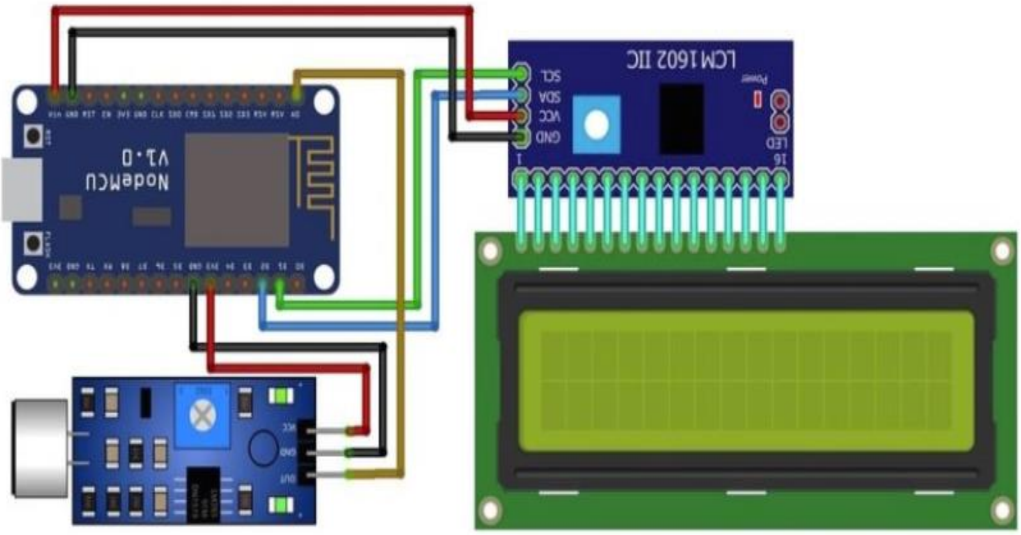


Fig 7: Circuit diagram of decibel sound detector

□ ESP8266 to LCD

i. D1 port of NodeMCU to SCL port in I2C
ii. D2 port of NodeMCU to SDA port in I2C
iii. GND port of NodeMCU to GND port in I2C
iv. Vin port of NodeMCU to Vcc port in I2C

□ ESP8266 to Sound Sensor

i. A0 port in NodeMCU to A0 port in Sensor
ii. GND port in NodeMCU to G port in Sensor
iii. VCC port in NodeMCU to + port in Sesnor

ThingSpeak module Connection

- An ID is created in the ThingSpeak using a registered Gmail ID and password.
- The connection between the thingspeak module and the project is developed through internal code attached in the below part of the document
- A general channel named "dbsounddetector" is then created into our ThingSpeak login.
- As the sound input is received through the mic sensor, the corresponding decibel value travels to the ThingSpeak cloud storage
- This process happens every 10 seconds, and the new decibel value gets updated in the ThingSpeak cloud storage
- A graph is generated for the decibel sound values generated.
- The detailed information about the time and the exact frequency in decibel is obtained by downloading the excel sheet.
- The internet connection required for this process is obtained through the mobile data connection provided to ESP8266

Connected Circuit diagram

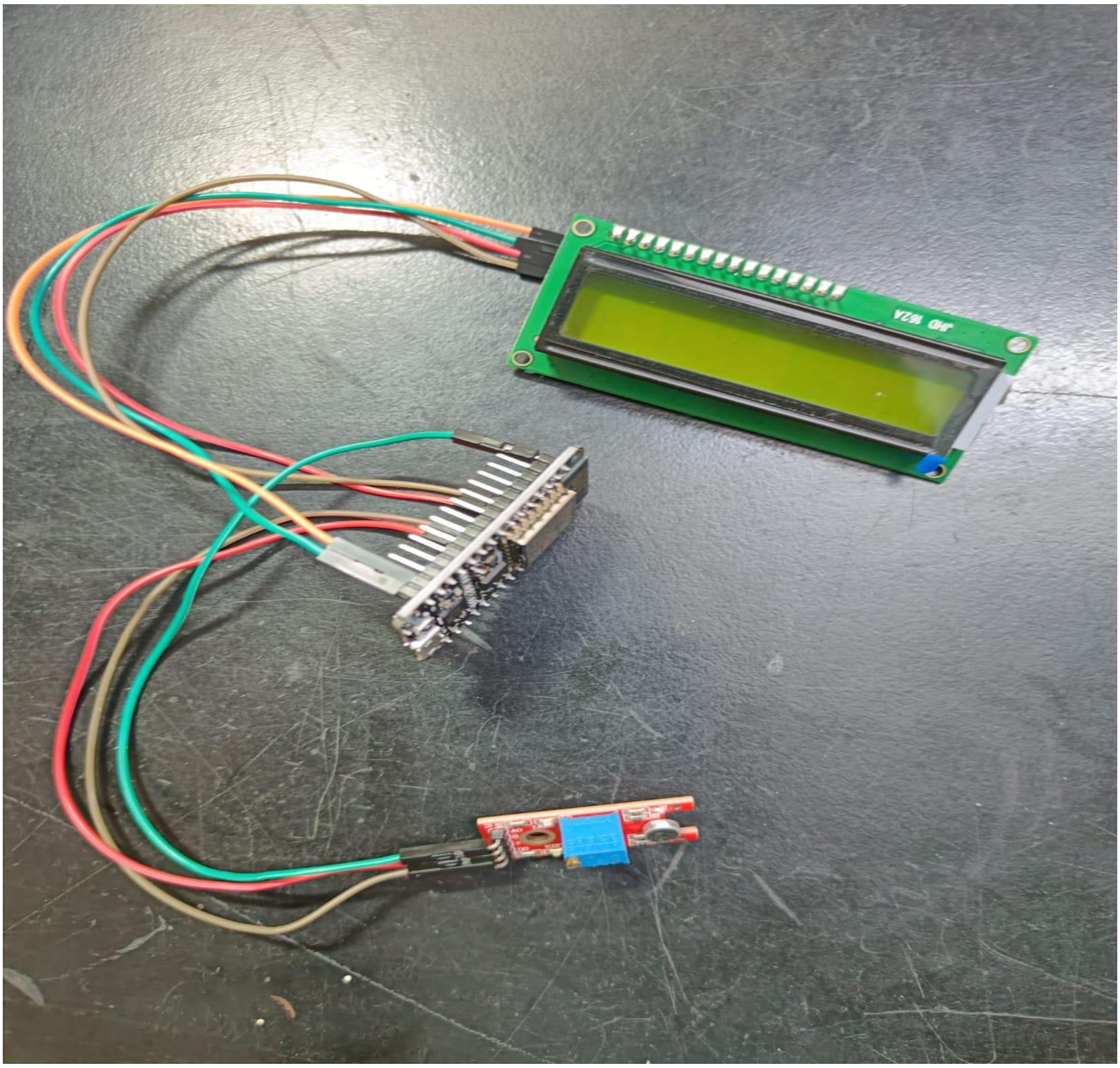
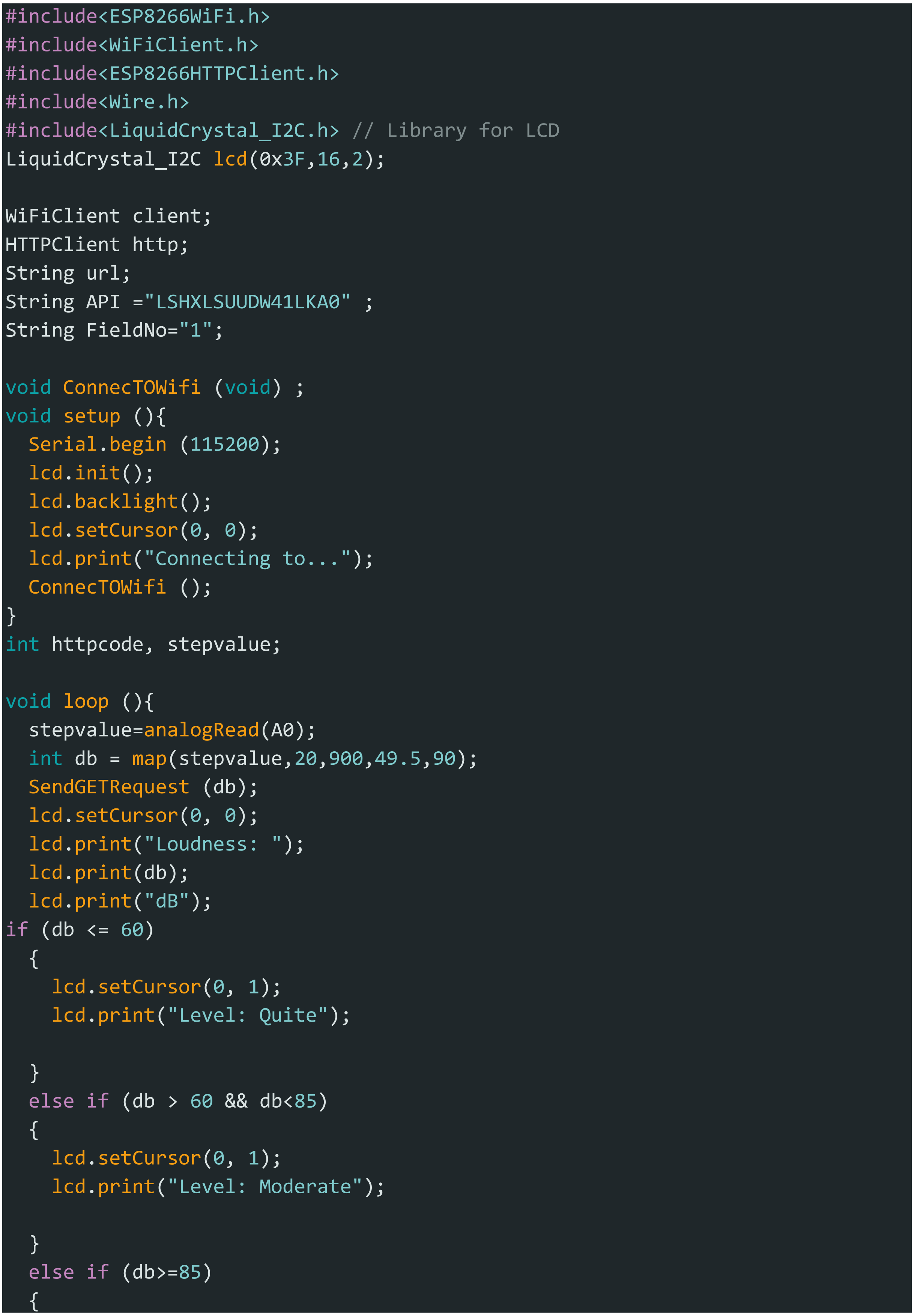
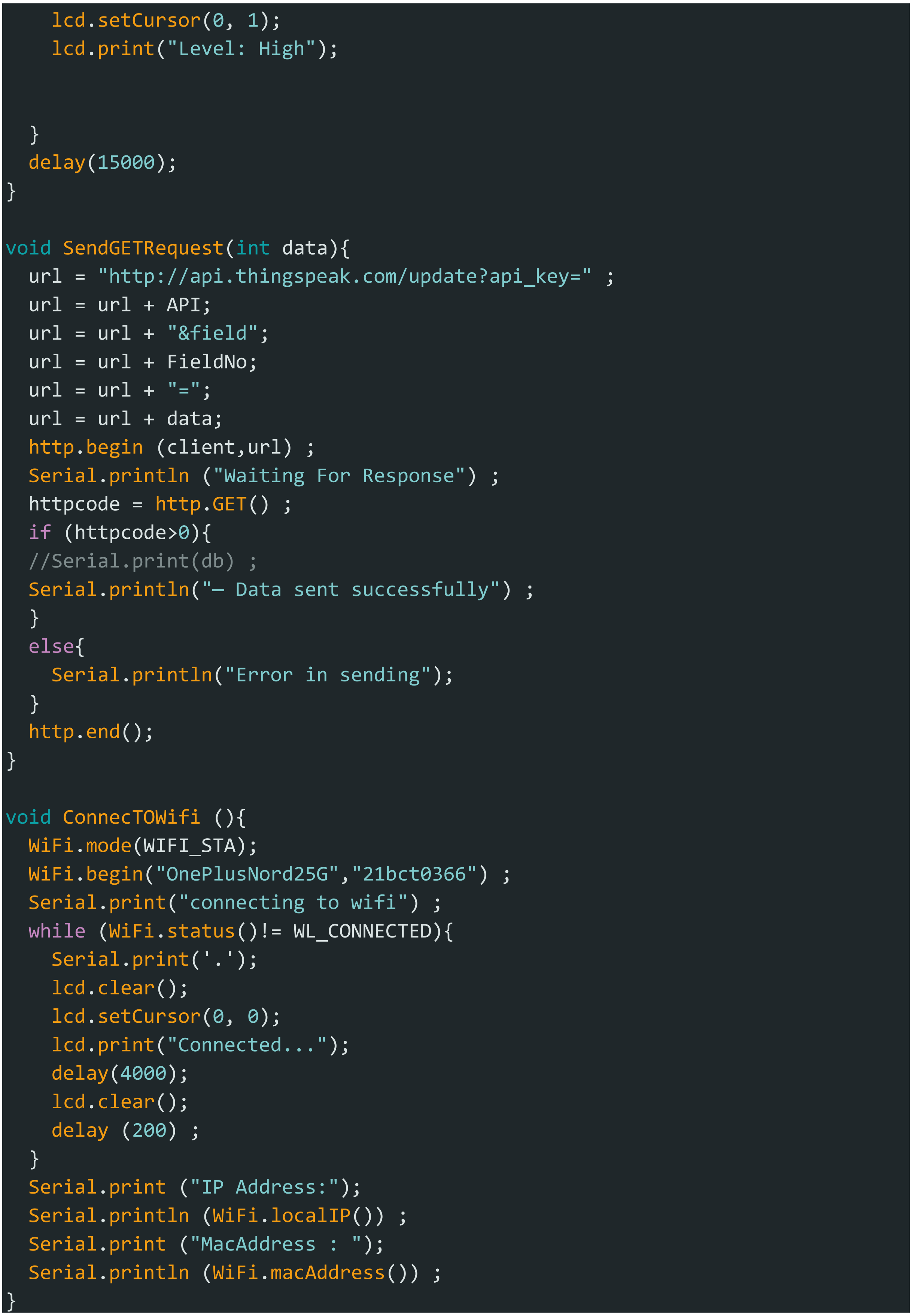


Fig 8: Connected circuit diagram

Code:



```
#include<ESP8266WiFi.h>
#include<WiFiClient.h>
#include<ESP8266HTTPClient.h>
#include<Wire.h>
#include<LiquidCrystal_I2C.h> // Library for LCD
LiquidCrystal_I2C lcd(0x3F,16,2);

WiFiClient client;
HTTPClient http;
String url;
String API ="LSHXLSUUDW41LKA0" ;
String FieldNo="1";

void ConnecTOWifi (void) ;
void setup (){
  Serial.begin (115200);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Connecting to...");
  ConnecTOWifi ();
}
int httpcode, stepvalue;

void loop (){
  stepvalue=analogRead(A0);
  int db = map(stepvalue,20,900,49.5,90);
  SendGETRequest (db);
  lcd.setCursor(0, 0);
  lcd.print("Loudness: ");
  lcd.print(db);
  lcd.print("dB");
if (db <= 60)
  {
    lcd.setCursor(0, 1);
    lcd.print("Level: Quite");

  }
  else if (db > 60 && db<85)
  {
    lcd.setCursor(0, 1);
    lcd.print("Level: Moderate");

  }
  else if (db>=85)
  {
```



```
    lcd.setCursor(0, 1);
    lcd.print("Level: High");


  }
  delay(15000);
}

void SendGETRequest(int data){
  url = "http://api.thingspeak.com/update?api_key=" ;
  url = url + API;
  url = url + "&field";
  url = url + FieldNo;
  url = url + "=";
  url = url + data;
  http.begin (client,url) ;
  Serial.println ("Waiting For Response") ;
  httpcode = http.GET() ;
  if (httpcode>0){
  //Serial.print(db) ;
  Serial.println("– Data sent successfully") ;
  }
  else{
    Serial.println("Error in sending");
  }
  http.end();
}

void ConnecTOWifi (){
  WiFi.mode(WIFI_STA);
  WiFi.begin("OnePlusNord25G","21bct0366") ;
  Serial.print("connecting to wifi") ;
  while (WiFi.status()!= WL_CONNECTED){
    Serial.print('.');
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Connected...");
    delay(4000);
    lcd.clear();
    delay (200) ;
  }
  Serial.print ("IP Address:");
  Serial.println (WiFi.localIP()) ;
  Serial.print ("MacAddress : ");
  Serial.println (WiFi.macAddress()) ;
}
```

Output

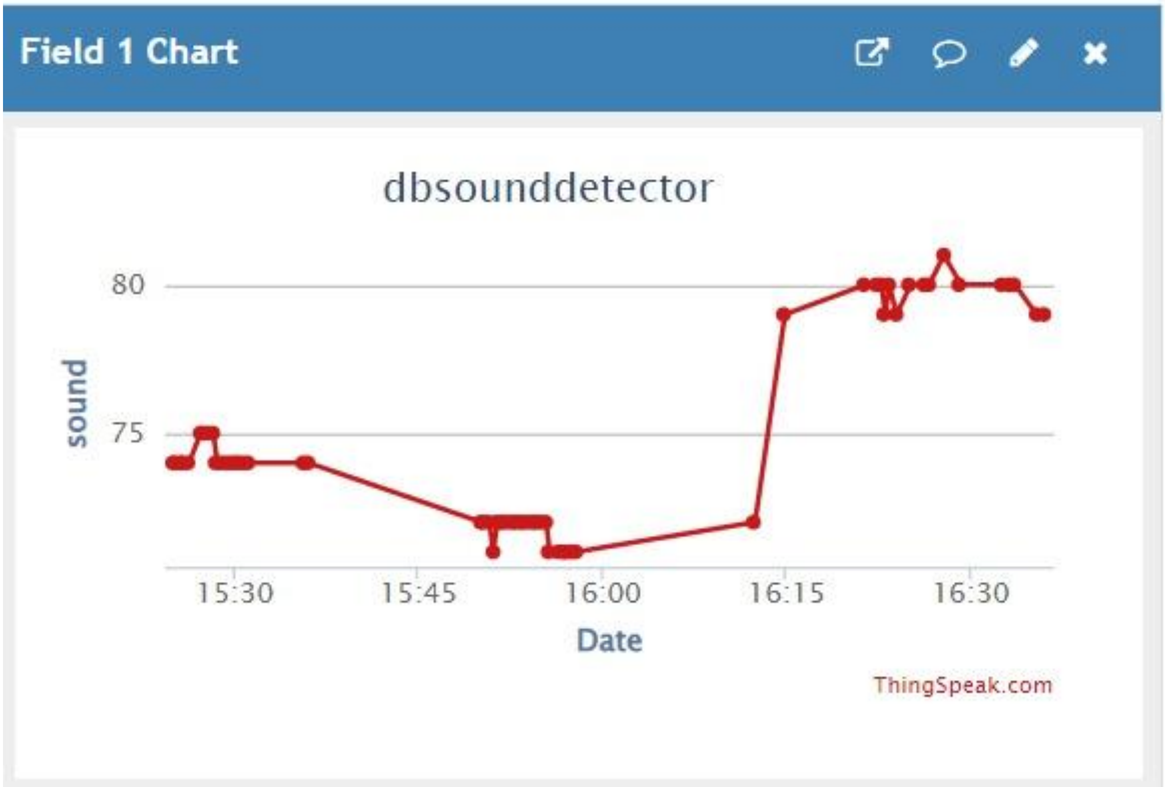


Fig 9: ThingSpeak Graph Output

Conclusion:

The aim to reduce noise pollution caused by firecrackers near hospitals, schools, and other significant public locations, to lower the cost of production and the quantity of chemicals used to make crackers, and to develop a tool that will enable people to live in a secure environment free from harm is achieved through this device. The sound from the environment is detected through Mic Sensor and given to ESP8266 NodeMCU board. The decibel value calculated is displayed on the screen. For every 10 secs, the value gets uploaded to ThingSpeak website through Wi-Fi. In ThingSpeak website, the user can view the graph and download the data in Excel format, also a message notification is sent to registered smart phones which gives a high sound decibel level alert. This way the production company can keep check on the sound pollution.

References:

1. *https://store-usa.arduino.cc/products/arduino-uno-rev3*
2. *https://en.wikipedia.org/wiki/Arduino_Uno*
3. *https://docs.arduino.cc/hardware/uno-rev3*
4. *https://www.javatpoint.com/arduino-blinking-anled#:~:text=It%20is%20the%20simple%20basic,are%20used%20in%20various%20applications*.
5. *https://www.tutorialspoint.com/arduino/arduino_blinking_led.htm*
6. *https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/*
7. *https://thingspeak.com/*