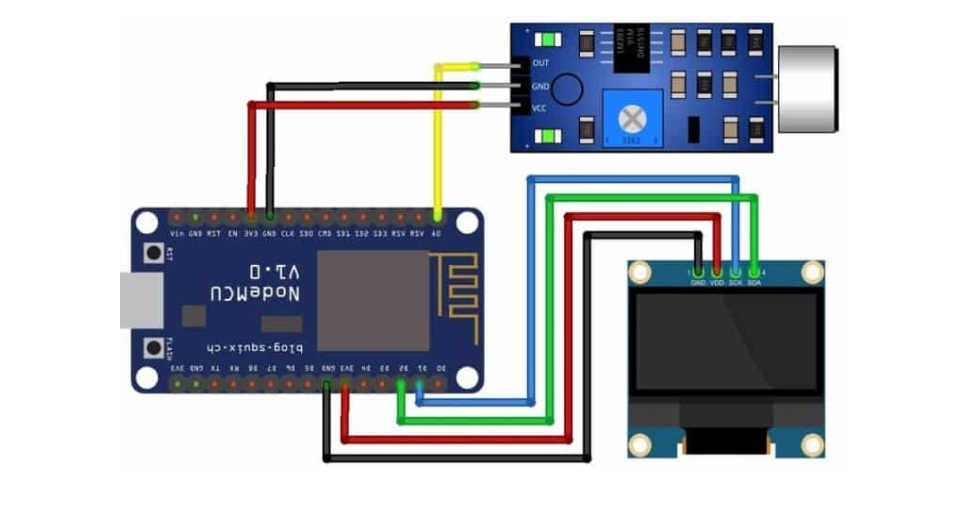
**IoT Decibelmeter with ESP8266 & OLED Display**

The Decibelmeter circuit

The Decibelmeter circuit diagram for Interfacing Sound Sensor with NodeMCU ESP8266 & OLED Display is given below. The connection is fairly simple.



The sound sensor

The sound sensor and OLED display both need 3.3V power supply. So connect their VCC & GND terminal to 3.3V & GND of NodeMCU respectively. Connect the output pin of the sound sensor to the Nodemcu A0 pin. Connect the I2C pin, i.e SDA & SCL pin of OLED Display to Nodemcu D2 & D1 pin respectively.

**Source Code: Decibelmeter with ESP8266 & OLED Display**

Copy the complete

Copy the complete code from below and upload it to the Nodemcu board. But before uploading make sure to change the wifi SSID, Password & Thingspeak API Key.

<https://thingspeak.com/>

**String apiKey = "14K8UL2QEK8BTHN6"; // Enter your Write API key from ThingSpeak**

**const char \*ssid = "Alexahome"; // replace with your wifi ssid and wpa2 key**

**const char \*pass = "12345678";**

**code :** [**Download**](https://github.com/vishal-ravi/projects/blob/main/sound)

#include <ESP8266WiFi.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

#define OLED\_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

String apiKey = "14K8UL2QEK8BTHN6"; // Enter your Write API key from ThingSpeak

const char \*ssid = "Alexahome"; // replace with your wifi ssid and wpa2 key

const char \*pass = "12345678";

const char\* server = "api.thingspeak.com";

const int sampleWindow = 50; // Sample window width in mS (50 mS = 20Hz)

unsigned int sample;

WiFiClient client;

void setup()

{

Serial.begin(115200); //Serial comms for debugging

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); //OLED display start

display.display(); //show buffer

display.clearDisplay(); //clear buffer

display.setTextSize(1); //Set text size to 1 (1-6)

display.setTextColor(WHITE); //Set text color to WHITE (no choice lol)

display.setCursor(0,0); //cursor to upper left corner

display.println("Decibelmeter"); //write title

display.display(); //show title

delay(2000); //wait 2 seconds

WiFi.begin(ssid, pass);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

display.clearDisplay();

display.setCursor(0,0);

display.setTextSize(1);

display.setTextColor(WHITE);

display.print("WiFi connected");

display.display();

delay(4000);

display.clearDisplay();

}

//--------------------------------------------------------------------------------------------

// MAIN LOOP

//--------------------------------------------------------------------------------------------

void loop()

{

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(0); //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

float db = map(peakToPeak,20,900,49.5,90); //calibrate for deciBels

display.setCursor(0,0); //cursor to upper left

display.setTextSize(2); //set text size to 2

display.print(db); //write calibrated deciBels

display.print(" dB"); //write units

for(int x =5;x<114;x=x+6){ //draw scale

display.drawLine(x, 32, x, 27, WHITE);

}

display.drawRoundRect(0, 32, 120, 20, 6, WHITE); //draw outline of bar graph

int r = map(db,0,120,1,120); //set bar graph for width of screen

display.fillRoundRect(1, 33, r, 18, 6, WHITE); //draw bar graph with a width of r

display.display(); //show all that we just wrote & drew

display.clearDisplay(); //clear the display

if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com

{

String postStr = apiKey;

postStr += "&field1=";

postStr += String(db);

postStr += "r\n";

client.print("POST /update HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n");

client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

client.print("Content-Type: application/x-www-form-urlencoded\n");

client.print("Content-Length: ");

client.print(postStr.length());

client.print("\n\n");

client.print(postStr);

}

client.stop();

delay(150);

}