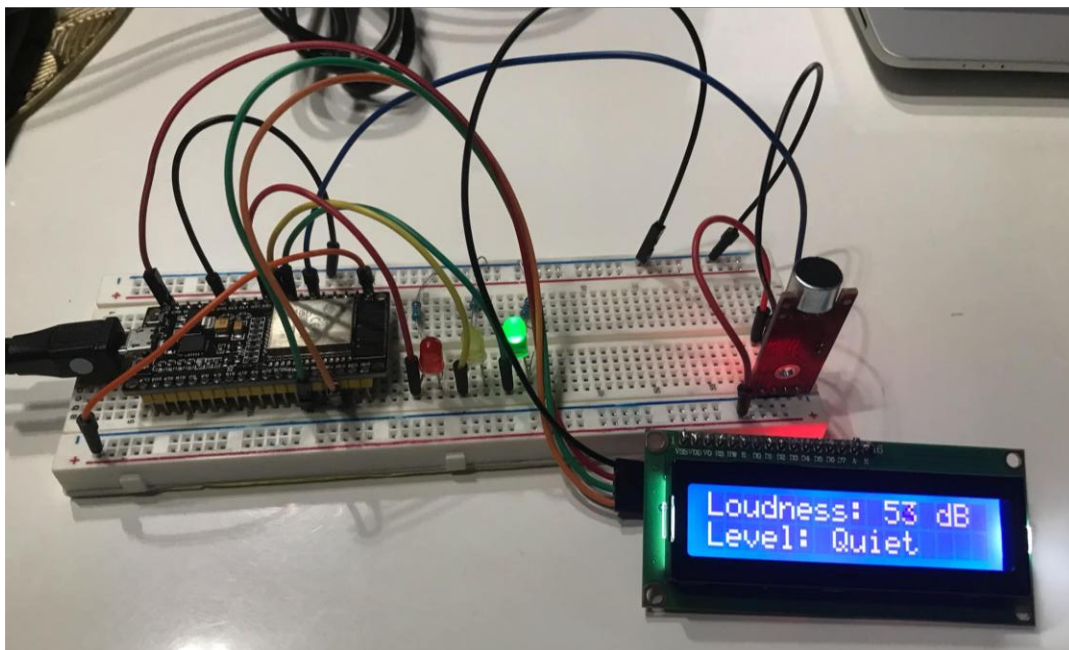


Hlukomer



Hlukomer je pokročilé zariadenie, ktoré je dostupné bežne v predajniach, prípadne e-shopoch, Tento projekt sa však od dostupného prístroja odlišuje aj v tom, že hluk nielen meria v decibeloch, ale úroveň hluku zobrazuje aj prostredníctvom LED diód.

Toto zariadenie bolo prvotne zostrojené s myšlienkou na hluk v materských školách, školských jedálňach, kde sme chceli upozorniť na úroveň hladiny hluku.

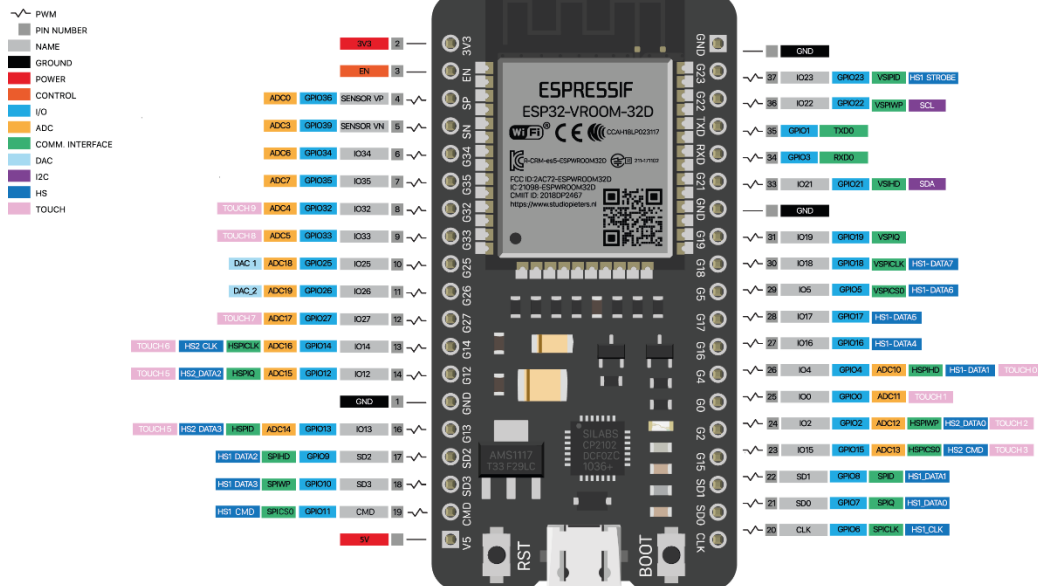
Úvod

V tomto projekte sme sa rozhodli **prepojiť snímač KY-038** s mikrokontrolérom ESP32 a postaviť z neho jednoduchý **decibel meter**, ktorý na základe programu v Arduino IDE detekuje hladinu zvuku, rozsvieti príslušnú diódu a zobrazí nameranú hodnotu v DB na LCD displeji.

Komponenty

1. ESP32: Mikrokontrolér ESP32

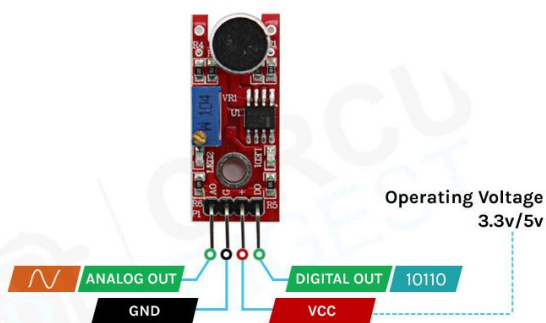
Tento výkonný mikrokontrolér ponúka dvojjadrový procesor, Wi-Fi a Bluetooth konektivitu, čo umožňuje bezdrôtové pripojenie a vzdialené monitorovanie. ESP32 je tiež vybavený množstvom vstupov a výstupov, ktoré umožňujú pripojenie rôznych senzorov a periférií. Konkrétne v tomto projekte využívame ESP 32 Wroom Pinout.



2. Zvukový senzor KY-038

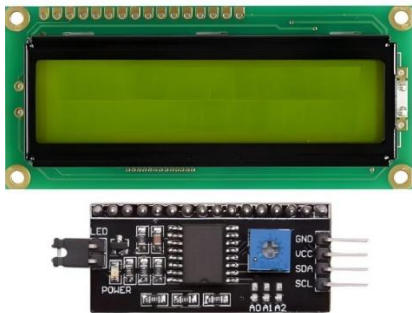
Tento snímač má štyri kolíky, z ktorých dva sú napájacie kolíky vyrovnané VCC a GND a ďalšie dva sú analógové a digitálne kolíky znázornené na obrázku vyššie. Má zabudovanú LED diódu napájania a signálnu LED. LED napájania sa rozsvieti, keď je na dosku privedené napájanie a signálna LED sa rozsvieti, keď je obvod spustený. Táto doska má tiež komparačný operačný zosilňovač, ktorý je zodpovedný za konverziu prichádzajúceho analógového signálu na digitálny signál. Máme aj potenciometer na nastavenie citlivosti; pomocou toho môžeme nastaviť citlivosť zariadenia. Nakoniec tu máme kondenzátorový mikrofón, ktorý sa používa na detekciu zvuku. Všetky tieto dohromady tvoria celkový modul zvukového senzora.

Senzor používa kondenzátorový mikrofón na detekciu zvukových vln. Mikrofón vydáva iba analógové signály, keď zvuková vlna zasiahne membránu snímača. Tento analógový signál spracuje operačný zosilňovač a dostaneme digitálny výstup.



3. 2x16 LCD displej

Displej 2x16 LCD je použitý na zobrazenie nameraných hodnôt v reálnom čase. Tento displej poskytuje jasné a čitateľné zobrazenie údajov, čo umožňuje jednoduché sledovanie hladiny hluku v zadaných jednotkách (dB).



Potrebné diely

ESP 32

Snímač KY-038

Display 2X16 LCD so zbernicou I2C

Prepojovacie vodiče

LED diódy

odporové rezistory 220Ω

Bredboard

Vzájomné prepojenie dielov/komponentov

Senzor KY-038 - ESP 32

GND – čierny vodič – GND

VCC – červený vodič – 3V3

AO – modrý vodič – GPIO35

LED displej – ESP 32

VCC - červený vodič - 3V3

GND - čierny vodič - GND

SDA – zelený vodič – GPIO21

SCL – oranžový vodič – GPIO22

LED diódy – ESP 32

Zelená led dióda – zelený vodič – GPIO33

Žltá led dióda – žltý vodič – GPIO25

Zelená led dióda – zelený vodič – GPIO26

Zelená dióda – normálna hladina hluku $\leq 60\text{dB}$

Žltá dióda – zvýšená hladina hluku $60\text{dB} \geq 85\text{dB}$

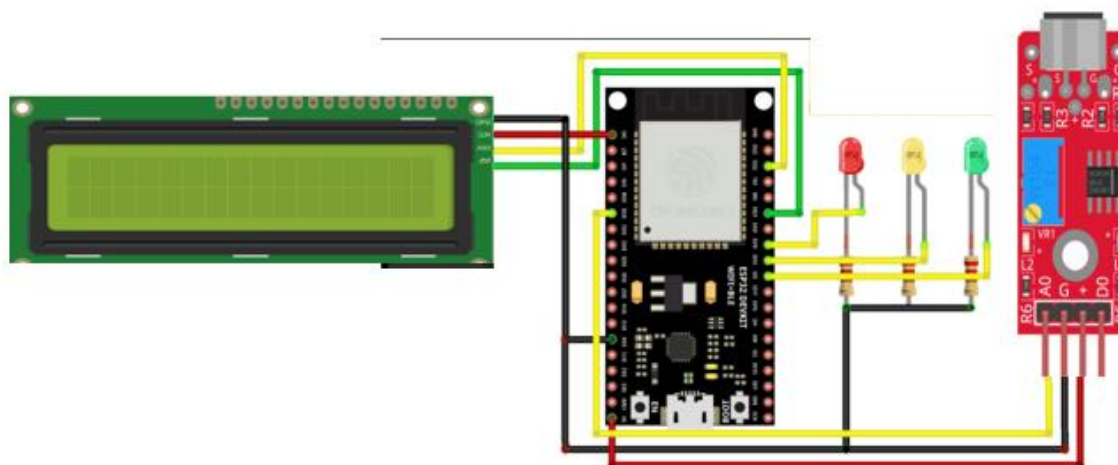
Červená dióda – vysoká hladina hluku $\leq 85\text{dB}$

Vývojové prostredie

Kód pre HLUKOMER založený na Arduino je veľmi jednoduchý a ľahko pochopiteľný. V kóde len zmeriame zvukový šum prichádzajúci z mikrofónu a prevedieme ho na digitálny signál so vzorkou ADC a zistíme decibelový šum.

Spustíme náš kód vrátane všetkých požadovaných knižníc; sú to knižnice **Wire.h** ,
LiquidCrystal_I2C.h.

Schéma zapojenia



Zdrojový kód

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#define SENSOR_PIN 35

#define PIN_QUIET 33

#define PIN_MODERATE 25

#define PIN_LOUD 26


LiquidCrystal_I2C lcd(0x27, 16, 2);


const int sampleWindow = 50;

unsigned int sample;


void setup() {

    pinMode(SENSOR_PIN, INPUT);

    pinMode(PIN_QUIET, OUTPUT);

    pinMode(PIN_MODERATE, OUTPUT);

    pinMode(PIN_LOUD, OUTPUT);


    digitalWrite(PIN_QUIET, LOW);

    digitalWrite(PIN_MODERATE, LOW);

    digitalWrite(PIN_LOUD, LOW);


    Serial.begin(115200);


    lcd.init();

    lcd.backlight();

    lcd.setCursor(0, 0);

    lcd.print("Loudness Meter");

}


void loop() {
```

```
unsigned long startMillis = millis();
```

```
float peakToPeak = 0;
```

```
unsigned int signalMax = 0;
```

```
unsigned int signalMin = 1024;
```

```
while (millis() - startMillis < sampleWindow) {
```

```
    sample = analogRead(SENSOR_PIN);
```

```
    if (sample < 1024) {
```

```
        if (sample > signalMax) {
```

```
            signalMax = sample;
```

```
        } else if (sample < signalMin) {
```

```
            signalMin = sample;
```

```
        }
```

```
    }
```

```
}
```

```
peakToPeak = signalMax - signalMin;
```

```
int db = map(peakToPeak, 0, 900, 49, 90);
```

```
Serial.print("\t");
```

```
Serial.println(db);
```

```
lcd.clear();
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Loudness: ");
```

```
lcd.print(db);
```

```
lcd.print("dB");
```

```
if (db <= 55) {
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("Level: Quiet");
```

```
    digitalWrite(PIN_QUIET, HIGH);
```

```
    digitalWrite(PIN_MODERATE, LOW);
```

```
    digitalWrite(PIN_LOUD, LOW);
```

```
} else if (db > 60 && db < 85) {
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print("Level: Moderate");

    digitalWrite(PIN_QUIET, LOW);

    digitalWrite(PIN_MODERATE, HIGH);

    digitalWrite(PIN_LOUD, LOW);

} else if (db >= 85 && db <= 90) {

    lcd.setCursor(0, 1);

    lcd.print("Level: High");

    digitalWrite(PIN_QUIET, LOW);

    digitalWrite(PIN_MODERATE, LOW);

    digitalWrite(PIN_LOUD, HIGH);

} else {

    digitalWrite(PIN_QUIET, LOW);

    digitalWrite(PIN_MODERATE, LOW);

    digitalWrite(PIN_LOUD, LOW);

}

delay(200);

}
```

Zdroje

<https://circuitdigest.com/microcontroller-projects/interface-ky038-sound-sensor-with-esp32>

https://www.hwpro.cz/oc/index.php?route=product/product&product_id=201

<https://uniot.sk/Teoria?id=20>

<https://domoticx.net/webshop/esp32-microcontroller-wifi-bluetooth-38-pins-esp-wroom-32-met-cp2102-usb-chip-usb-micro/?add-to-cart=33302>

