Sistem de monitorizare a calitatii aerului

Arghir Monica-Andreea

332CA

Introducere

Considerând lumea modernă în care trăim și aerul pe care îl respirăm zi de zi, sistemul de monitorizare a calității aerului pe care îl construiesc sare în ajutorul locuitorilor României, cu precădere celor din București. Datorită acestui device inteligent, cei care se folosesc de el vor avea o apreciere a aerului care îi înconjoară, afișată pe un ecran. Mai mult decât atât, aparatul va porni o alarmă în momentul detectării unei calități foarte slabe a aerului pentru a avertiza persoanele din jur de pericolul pe care îl respiră. Acest proiect este folositor atât pentru a păstra evidența calității aerului dintr-o încăpere în propria locuință, cât și în spații publice, precum cafenele, restaurante, muzee etc.

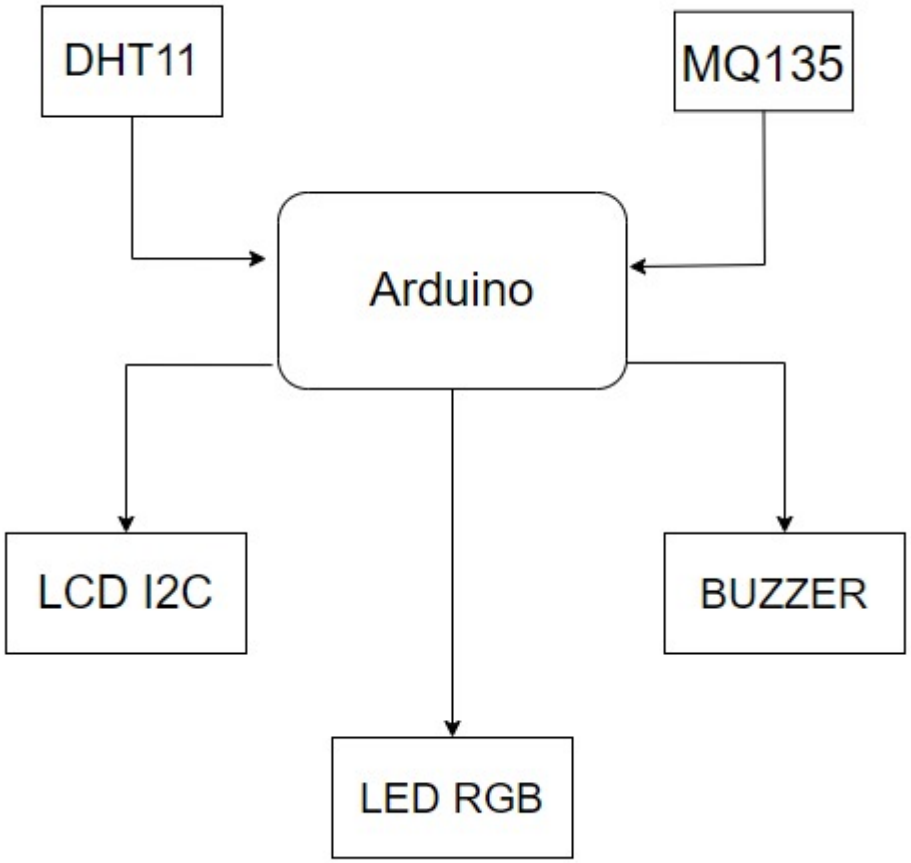
Descriere generală

Pentru implementarea acestui sistem inteligent, am ales să pornesc de la modulul ESP8266 pe care îl voi folosi împreună cu doi senzori ce vor recepționa date despre temperatura si umiditatea din jur, precum și calitatea aerului folosind MQ135. Datele vor fi afișate pe un ecran LCD cu I2C. În funcție de calitate mai bună sau mai slabă a aerului, va fi aprins un led ce va indica:

→ verde: pentru o calitate foarte bună

→ roșu: pentru o calitate mai slabă

Pentru o calitate care ajunge a fi periculoasă pentru cei din jur, va porni o alarmă cu ajutorul unui buzzer pentru a atenționa oamenii. Ulterior, am renunțat la folosirea ESP8266 și am optat pentru folosirea unei plăcuțe Arduino UNO. Motivul pentru care am recurs la această metodă a fost diferența de tensiune de alimentare dintre cele două variante. La încercarea de a conecta ecranul LCD I2C, am observat că acesta are nevoie de 5V, iar ESP8266 putea să ofere doar 3.3V. Pentru a nu complica implementarea HW, am ales să schimb plăcuța.

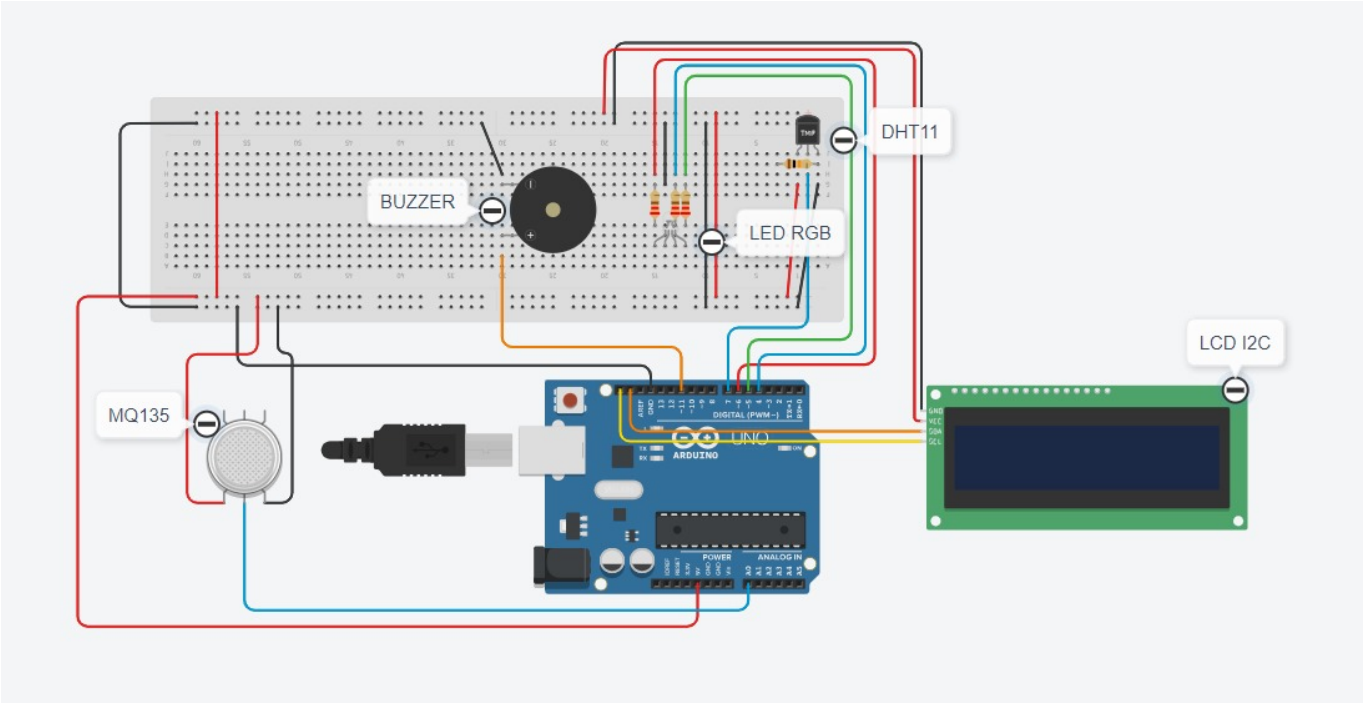


Hardware Design

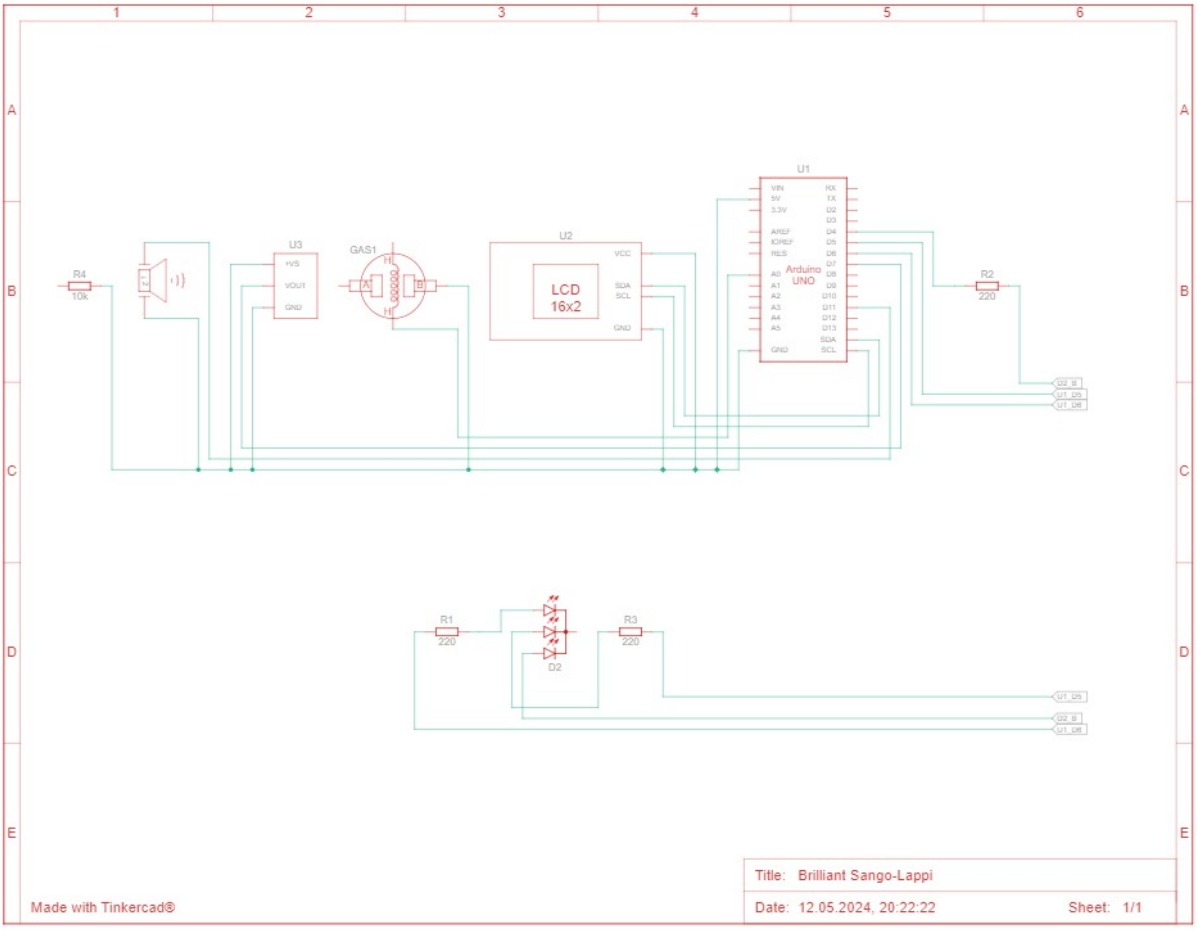
Listă piese:

- Arduino UNO
- Senzor de temperatură și umiditate DHT11
- Senzor de măsurare a calității aerului MQ135
- Ecran LCD I2C
- Buzzer pasiv
- LED RGB
- 1 rezistență 10k
- 3 rezistențe 220
- Breadboard
- Fire

Design Circuit



Schemă Electrică



Software Design

Mediul de dezvoltare pe care l-am folosit este Arduino UNO. Am adăugat următoarele librării:

- Wire.h (comunicația prin I2C)
- LiquidCrystal_I2C.h (pentru LCD)
- dht.h (pentru senzorul de temperatură)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <dht.h>
```

În continuare, se realizează inițializarea obiectelor pentru comunicarea cu afișajul LCD și senzorul DHT. De asemenea, declarăm și inițializăm variabile specifice pinilor folosiți pentru senzorul DHT, buzzer, senzorul de gaz și 3 pini (roșu, verde și albastru) pentru ledul RGB.

```
dht DHT;
LiquidCrystal_I2C lcd(0x27,16,2);
#define ADC_CHANNEL 0 // A0

int dhtPin = 7;
int buzzerPin = 11;

int redPin = 6;
int greenPin = 5;
int bluePin = 4;
```

La pornirea sistemului, se va apela funcția de setup ce are scopul de a inițializa și a activa lumina de fundal a ecranului LCD și de a configura pinii pentru led și pinul pentru buzzer drept output. În plus, se configurează regiștrii specifici realizării convertiri din analog în digital, folosită pentru citirea inputului primit de la senzorul de gaz MQ135.

```
void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);

  pinMode(buzzerPin, OUTPUT);

  lcd.init();
  lcd.backlight();

  // Configure the ADC
  // Clear registers
  ADCSRA = 0;
  ADCSRB = 0;
  ADMUX  = 0;

  // AVCC with external capacitor at AREF pin
  ADMUX |= (1 << REFS0);
```

```
  ADMUX |= ADC_CHANNEL;

  // Set ADC prescaler to 128 (16MHz/128 = 125kHz)
  ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);

  // Enable the ADC
  ADCSRA |= (1 << ADEN);
}
```

Următoarea funcție este și cea mai importantă. În funcția de loop, se pornește conversia ADC, se așteaptă ca aceasta să se realizeze, după care se citește valoarea digitală.

Pentru început, dacă valoarea citită este mai mică decât 400, vom aprinde ledul în culoarea verde și vom afișa pe ecran un mesaj corespunzător: “Good air quality”. De asemenea, vom afișa și temperatura resimțită la momentul respectiv, citită de pe pinul DHT.

Dacă valoarea citită de MQ135 se află între 400 și 700, anunțăm utilizatorii dispozitivului că aerul devine dăunător printr-un mesaj afișat pe ecran și modificăm culoarea ledului în roșu. În plus, vom apela o funcție ce va porni buzzerul, scoțând un sunet întrerupt pentru a atenționa utilizatorii de posibilul pericol.

Totuși, dacă valoarea depășește 700, vom porni alarma buzzerului, un sunet ce se va auzi neîntrerupt pâna la revenirea unei calități mai bune ale aerului. De asemenea, se va apela o funcție menită să afișeze un mesaj corespunzător și vom avea culoarea roșie pentru led.

```
void loop() {

  // Start the ADC conversion
  ADCSRA |= (1 << ADSC);

  // Wait for the conversion to complete
  while (ADCSRA & (1 << ADSC));

  // Read the ADC value
  uint16_t adcValue = ADC;

  if (gasVal >= 700) {
    analogWrite(buzzerPin, 200);
    printWarningMessage();

  } else {
    if (adcValue >= 400) {
      setColor(255, 0, 0); // Red Color
      printGettingBad(adcValue);
      buzzerDelay(buzzerPin);

    } else {
      setColor(0, 255, 0); // Green Color

      int chk = DHT.read11(dhtPin);
```

```
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Good air quality!");

      lcd.setCursor(0, 1);
      lcd.print("Temperature = ");
      lcd.print(DHT.temperature);
      delay(1000);
    }
  }
}
```

Pentru aprinderea unei anumite culori a ledului RGB, am folosit funcția setColor. Aceasta primește 3 valori, câte una pentru fiecare culoare ce va fi folosită în crearea culorii dorite, după care aceste valori sunt scrise analog în pinii specifici.

```
void setColor(int redValue, int greenValue, int blueValue) {
  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}
```

Funcția buzzerDelay va porni buzzerul atunci când calitatea aerului începe să scadă. Aceasta va primi ca parametru pinul specific buzzerului. Ne dorim să auzim un sunet întrerupt de o mică pauză astfel:

```
void buzzerDelay(int buzzerPin) {
  analogWrite(buzzerPin, 127);
  delay(500);
  analogWrite(buzzerPin, 0);
  delay(500);
}
```

Avem, desigur, și un mesaj ce se va afișa în cazul menționat precedent: “Air quality is getting bad”, urmat de valoarea citită de senzor. Funcția primește ca parametru valoarea ce trebuie afișată.

```
void printGettingBad(int adcValue) {
  lcd.clear();

  lcd.setCursor(0, 0);
  lcd.print("Air quality is ");

  lcd.setCursor(0,1);
  lcd.print("getting bad: ");
  lcd.print(adcValue);
  delay(500);
}
```

Ultima funcție folosită în cadrul acestui proiect este cea care afișează mesajul: “WARNING”, urmat de “DANGEROUS GAS”. Acest mesaj va fi afișat și va dispărea la un mic interval de timp pentru a atrage atenția.

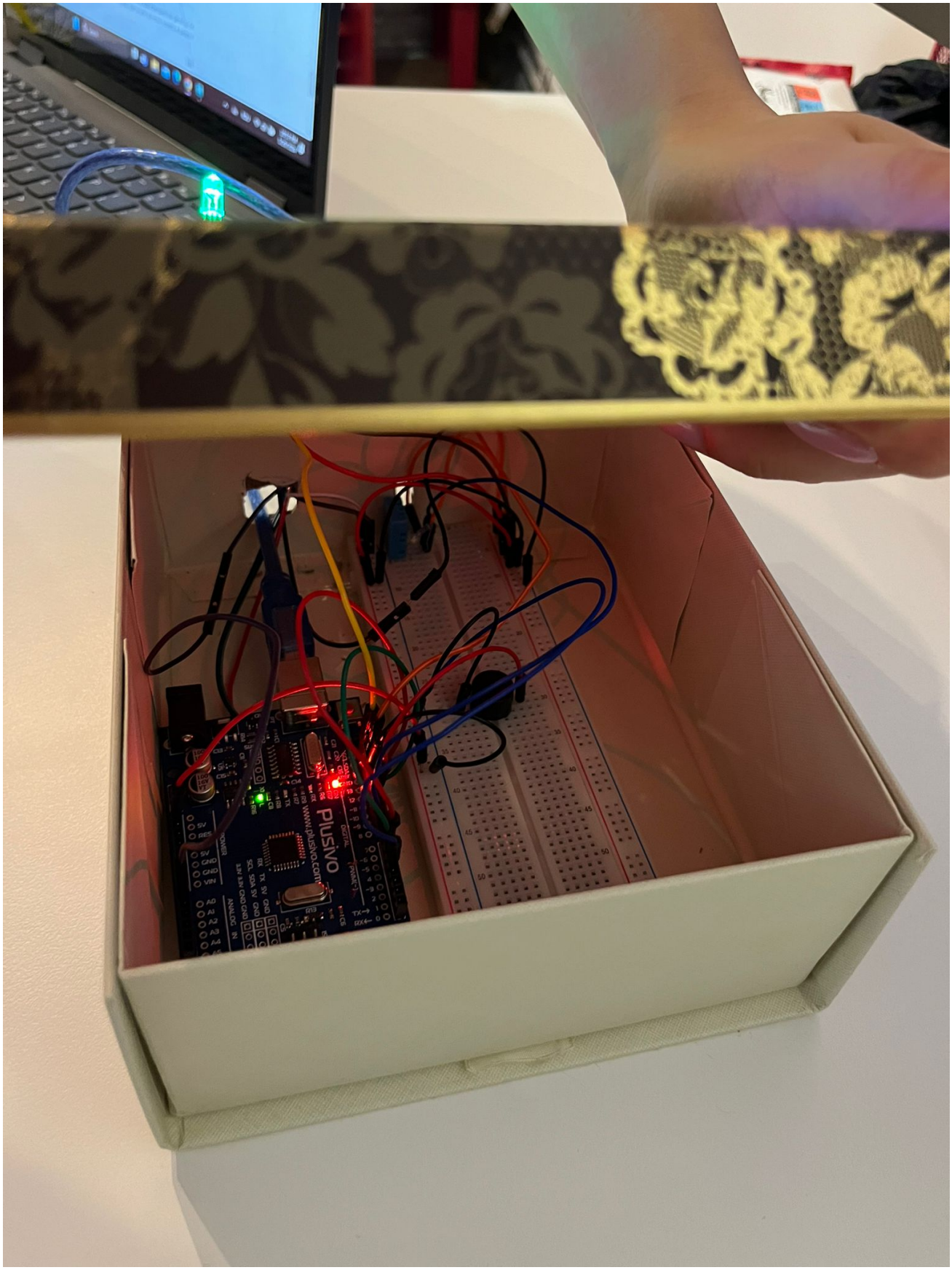
```
void printWarningMessage() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("    WARNING   ");

  lcd.setCursor(0, 1);
  lcd.print("  DANGEROUS GAS  ");
  delay(1000);

  lcd.noDisplay();
  delay(500);
  lcd.display();
  delay(1000);
}
```

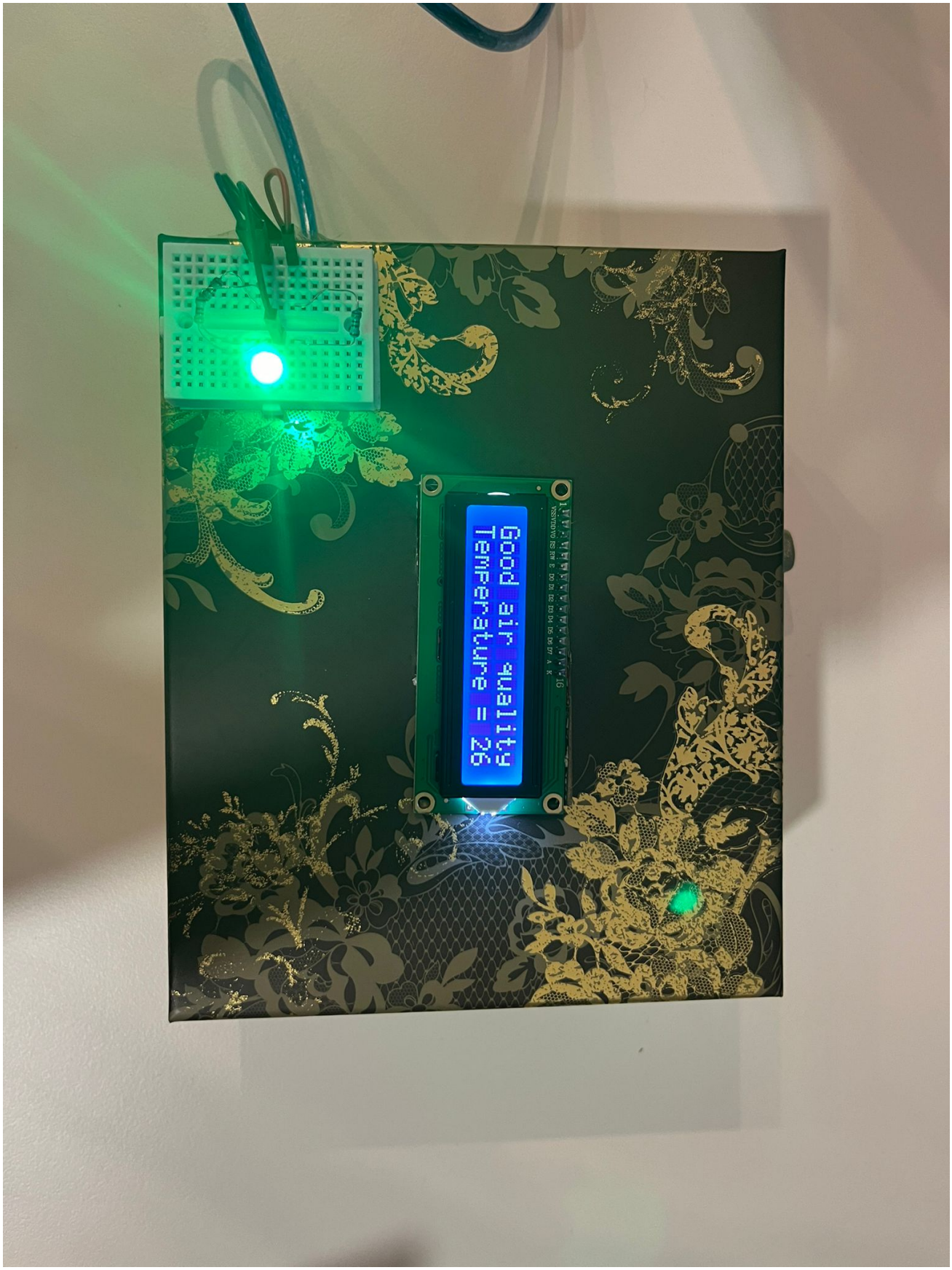
Rezultate Obţinute

Puteți viziona următorul videoclip pentru a vedea întreaga funcționalitate a proiectului: https://youtube.com/shorts/CB_Y99LJ6v8.

Avem următorul produs:


Good air quality
Temperature = 26
SISTEM DE
MONITORIZARE
A CALITĂȚII
AERULUI


Plusivo
WWW.PLUSIVO.COM



Concluzii

Consider acest proiect ca fiind unul foarte folositor în multe situații. Proiectul a fost interesant de

gândit și de creat, prin încercarea inițială de a folosi ESP8266 în loc de Arduino, dar și prin punerea laolaltă a pieselor pentru a crea un sistem estetic și eficient.

Download

Puteți obține codul prezentat mai sus prin descărcarea următoarei arhive: pm-project.zip

Bibliografie/Resurse

- Senzor DHT:
  - https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/
  - https://projecthub.arduino.cc/arcaegecengiz/using-dht11-12f621
- Senzor de gaz MQ135:
  - https://www.electrovigyan.com/arduino/mq135-air-quality-sensor/
  - https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024
- Buzzer pasiv:
  - https://medium.com/@arduinounomagic/buzzer-with-arduino-c06b64f010df
  - https://www.circuitgeeks.com/arduino-buzzer-tutorial/
  - https://community.element14.com/products/arduino/b/blog/posts/controlling-buzzer-using-analog-input
- LED RGB
  - https://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/
- LCD cu interfața I2C:
  - https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/
  - https://www.instructables.com/How-to-Connect-I2C-Lcd-Display-to-Arduino-Uno/
  - https://www.arduino.cc/reference/en/libraries/liquidcrystal/nodisp

Export to PDF