

Nume student: Dinu Dumitru-Razvan 335CA

Senzor de gaze cu afisaj LED/aplicatie mobila

Introducere

Proiectul prezentat are ca obiectiv inregistrarea nivelului de gaze si afisarea acestuia pe un display led cu 4 cifre. Totodata, afisarea datelor se poate realiza si pe o aplicatie mobile, prin intermediul unui modul Bluetooth, existand si o notificare prin intermediul unui buzzer, ce va reda o melodie in cazul in care nivelul trece de un anumit prag.

Descriere generala

Vor fi colectate date de la senzor, urmand ca mai departe acestea sa fie procesate si date mai departe celorlalte module.

Hardware Design

Componente folosite:

1. Arduino UNO R3
2. Modul sensor de gaz MQ-2
3. Modul display LED cu 4 cifre/Display LCD
4. Modul Bluetooth Master-Slave HC05
5. Breadboard
6. Rezistente
7. Fire

Schema electrica:

Initial mi-am dorit sa o realizez in Tinkercad, insa nu am gasit toate modulele asa ca am preferat sa realizez schema in Fritzing. Dupa cum se poate vedea si in schema, am pus rezistente, acolo unde a fost nevoie, astfel sa nu fie pusa in pericol vreo componenta.

Software Design

Pentru programarea placutei am folosit Arduino 1.8.19, iar schema elctrica, cum am spus si mai sus

am realizat-o in Fritzing. M-am folosit si de biblioteca "ezBuzzer.h", intrucat, initial, am incercat sa folosesc delay in redarea melodiilor, inasa acesta, fiind blocant, imi intarzia transmiterea in aplicatie a valorii si afisarea pe display. Am programat Display-ul led, astfel incat sa-mi afiseze corect valorile primite. Totodata, am cosiderat numarul 700 ca valoare de referinta pentru declansarea buzzerului, astfel, cand senzorul transmite o valoare superioara, noi vom fi notificati prin inceperea melodiei Merry Christmas (am presupus ca vine Craciunul mai repede pentru persoana in cauza :)). Aplicatia mobila am realizat-o pe platforma MIT App Inventor.

```
#include <ezBuzzer.h>
#define Dig1 13
#define Dig2 12
#define Dig3 11
#define Dig4 10
#define DIGIT_ON LOW
#define DIGIT_OFF HIGH
#define DISPLAY_BRIGHTNESS 5000
boolean duiz = false;
boolean hon = false;
#define segA 2
#define segB 3
#define segC 4
#define segD 5
#define segE 6
#define segF 7
#define segG 8
ezBuzzer buzzer(9);
const int nr = A0;

int melody[] = {
  NOTE_C5, //1
  NOTE_F5, NOTE_F5, NOTE_G5, NOTE_F5, NOTE_E5,
  NOTE_D5, NOTE_D5, NOTE_D5,
  NOTE_G5, NOTE_G5, NOTE_A5, NOTE_G5, NOTE_F5,
  NOTE_E5, NOTE_C5, NOTE_C5,
  NOTE_A5, NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_G5,
  NOTE_F5, NOTE_D5, NOTE_C5, NOTE_C5,
  NOTE_D5, NOTE_G5, NOTE_E5,

  NOTE_F5, NOTE_C5, //8
  NOTE_F5, NOTE_F5, NOTE_G5, NOTE_F5, NOTE_E5,
  NOTE_D5, NOTE_D5, NOTE_D5,
  NOTE_G5, NOTE_G5, NOTE_A5, NOTE_G5, NOTE_F5,
  NOTE_E5, NOTE_C5, NOTE_C5,
  NOTE_A5, NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_G5,
  NOTE_F5, NOTE_D5, NOTE_C5, NOTE_C5,
  NOTE_D5, NOTE_G5, NOTE_E5,
  NOTE_F5, NOTE_C5,

  NOTE_F5, NOTE_F5, NOTE_F5, //17
  NOTE_E5, NOTE_E5,
```

```

NOTE_F5, NOTE_E5, NOTE_D5,
NOTE_C5, NOTE_A5,
NOTE_AS5, NOTE_A5, NOTE_G5,
NOTE_C6, NOTE_C5, NOTE_C5, NOTE_C5,
NOTE_D5, NOTE_G5, NOTE_E5,
NOTE_F5, NOTE_C5,
NOTE_F5, NOTE_F5, NOTE_G5, NOTE_F5, NOTE_E5,
NOTE_D5, NOTE_D5, NOTE_D5,

NOTE_G5, NOTE_G5, NOTE_A5, NOTE_G5, NOTE_F5, //27
NOTE_E5, NOTE_C5, NOTE_C5,
NOTE_A5, NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_G5,
NOTE_F5, NOTE_D5, NOTE_C5, NOTE_C5,
NOTE_D5, NOTE_G5, NOTE_E5,
NOTE_F5, NOTE_C5,
NOTE_F5, NOTE_F5, NOTE_F5,
NOTE_E5, NOTE_E5,
NOTE_F5, NOTE_E5, NOTE_D5,

NOTE_C5, NOTE_A5, //36
NOTE_AS5, NOTE_A5, NOTE_G5,
NOTE_C6, NOTE_C5, NOTE_C5, NOTE_C5,
NOTE_D5, NOTE_G5, NOTE_E5,
NOTE_F5, NOTE_C5,
NOTE_F5, NOTE_F5, NOTE_G5, NOTE_F5, NOTE_E5,
NOTE_D5, NOTE_D5, NOTE_D5,
NOTE_G5, NOTE_G5, NOTE_A5, NOTE_G5, NOTE_F5,
NOTE_E5, NOTE_C5, NOTE_C5,

NOTE_A5, NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_G5, //45
NOTE_F5, NOTE_D5, NOTE_C5, NOTE_C5,
NOTE_D5, NOTE_G5, NOTE_E5,
NOTE_F5, NOTE_C5,
NOTE_F5, NOTE_F5, NOTE_G5, NOTE_F5, NOTE_E5,
NOTE_D5, NOTE_D5, NOTE_D5,
NOTE_G5, NOTE_G5, NOTE_A5, NOTE_G5, NOTE_F5,
NOTE_E5, NOTE_C5, NOTE_C5,

NOTE_A5, NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_G5, //53
NOTE_F5, NOTE_D5, NOTE_C5, NOTE_C5,
NOTE_D5, NOTE_G5, NOTE_E5,
NOTE_F5, 0
};

// note durations: 4 = quarter note, 8 = eighth note, etc, also called tempo:
int noteDurations[] = {
4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 4,

```

4, 8, 8, 8, 8,
4, 4, 8, 8,
4, 4, 4,
2, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 8, 8,
4, 4, 4,
2, 4,
4, 4, 4,
2, 4,
4, 4, 4,
4, 4, 8, 8,
4, 4, 4,
2, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 8, 8,
4, 4, 4,
2, 4,
4, 4, 4,
2, 4,
4, 4, 4,
4, 4, 8, 8,
4, 4, 4,
2, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 8, 8,
4, 4, 4,
2, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 4,
4, 8, 8, 8, 8,
4, 4, 8, 8,
4, 4, 4,

```
2, 4
};
void setup()
{
    pinMode(segA, OUTPUT);
    pinMode(segB, OUTPUT);
    pinMode(segC, OUTPUT);
    pinMode(segD, OUTPUT);
    pinMode(segE, OUTPUT);
    pinMode(segF, OUTPUT);
    pinMode(segG, OUTPUT);
    pinMode(Dig1, OUTPUT);
    pinMode(Dig2, OUTPUT);
    pinMode(Dig3, OUTPUT);
    pinMode(Dig4, OUTPUT);
    Serial.begin(9600);
    pinMode(nr, INPUT);
}
void loop()
{
    buzzer.loop();
    int number;
    int x = analogRead(nr);
    Serial.print(x);

    duiz = false;
    hon = false;
    showNumber(x);
    if (x > 700) {
        if (buzzer.getState() == BUZZER_IDLE) {
            int length = sizeof(noteDurations) / sizeof(int);
            buzzer.playMelody(melody, noteDurations, length); // playing
        }
    }

    if (x <= 700) {
        if (buzzer.getState() != BUZZER_IDLE) {
            buzzer.stop() ; // stop
        }
    }
}

void showNumber(int x) {
    for(int k = 0; k<50; k++){ // for loop to slow it down.

        int figur = x;
        for(int digit = 1 ; digit < 5 ; digit++) { //for loop to place the number
in the right digit
            switch(digit) {
                case 1:
                    if(figur > 999){
```

```
digitalWrite(Dig1, DIGIT_ON);
lightNumber(figur / 1000); // for example 2511 / 1000 = 2
figur %= 1000; // new value of figur = 511 figur = figur %1000

delayMicroseconds(DISPLAY_BRIGHTNESS);
if (figur < 100){
    duiz = true;
    if (figur <10){
        hon = true;
    }

    }else duiz = false;
}

break;
case 2:
if(duiz == true){
    digitalWrite(Dig2, LOW);
    lightNumber(0);
    delayMicroseconds(DISPLAY_BRIGHTNESS);

}if(hon == true){
    break;
}

if(figur > 99 && figur < 1000){
    digitalWrite(Dig2, LOW);
    lightNumber(figur / 100);
    figur %= 100;
    delayMicroseconds(DISPLAY_BRIGHTNESS);
    if (figur < 10){
        hon = true;

    }else hon = false;
}
break;
case 3:
if(hon == true){
    digitalWrite(Dig3, LOW);
    lightNumber(0);
    delayMicroseconds(DISPLAY_BRIGHTNESS);
    break;
}

if(figur > 9 && figur < 100){
    digitalWrite(Dig3, LOW);
    lightNumber(figur / 10);
    figur %= 10;
    delayMicroseconds(DISPLAY_BRIGHTNESS);
}
```

```
    break;
    case 4:
    if(figur < 10){
        digitalWrite(Dig4, LOW);
        lightNumber(figur);
        delayMicroseconds(DISPLAY_BRIGHTNESS);
        break;
    }
}
//Turn off all segments
lightNumber(10);
//Turn off all digits
digitalWrite(Dig1, DIGIT_OFF);
digitalWrite(Dig2, DIGIT_OFF);
digitalWrite(Dig3, DIGIT_OFF);
digitalWrite(Dig4, DIGIT_OFF);

}
}
```

```
void lightNumber(int numberToDisplay) {
    #define SEGMENT_ON LOW
    #define SEGMENT_OFF HIGH
    switch (numberToDisplay){
        case 0:
            digitalWrite(segA, SEGMENT_ON);
            digitalWrite(segB, SEGMENT_ON);
            digitalWrite(segC, SEGMENT_ON);
            digitalWrite(segD, SEGMENT_ON);
            digitalWrite(segE, SEGMENT_ON);
            digitalWrite(segF, SEGMENT_ON);
            digitalWrite(segG, SEGMENT_OFF);
            break;
        case 1:
            digitalWrite(segA, SEGMENT_OFF);
            digitalWrite(segB, SEGMENT_ON);
            digitalWrite(segC, SEGMENT_ON);
            digitalWrite(segD, SEGMENT_OFF);
            digitalWrite(segE, SEGMENT_OFF);
            digitalWrite(segF, SEGMENT_OFF);
            digitalWrite(segG, SEGMENT_OFF);
            break;
        case 2:
            digitalWrite(segA, SEGMENT_ON);
            digitalWrite(segB, SEGMENT_ON);
            digitalWrite(segC, SEGMENT_OFF);
            digitalWrite(segD, SEGMENT_ON);
            digitalWrite(segE, SEGMENT_ON);
            digitalWrite(segF, SEGMENT_OFF);
            digitalWrite(segG, SEGMENT_OFF);
            break;
    }
}
```

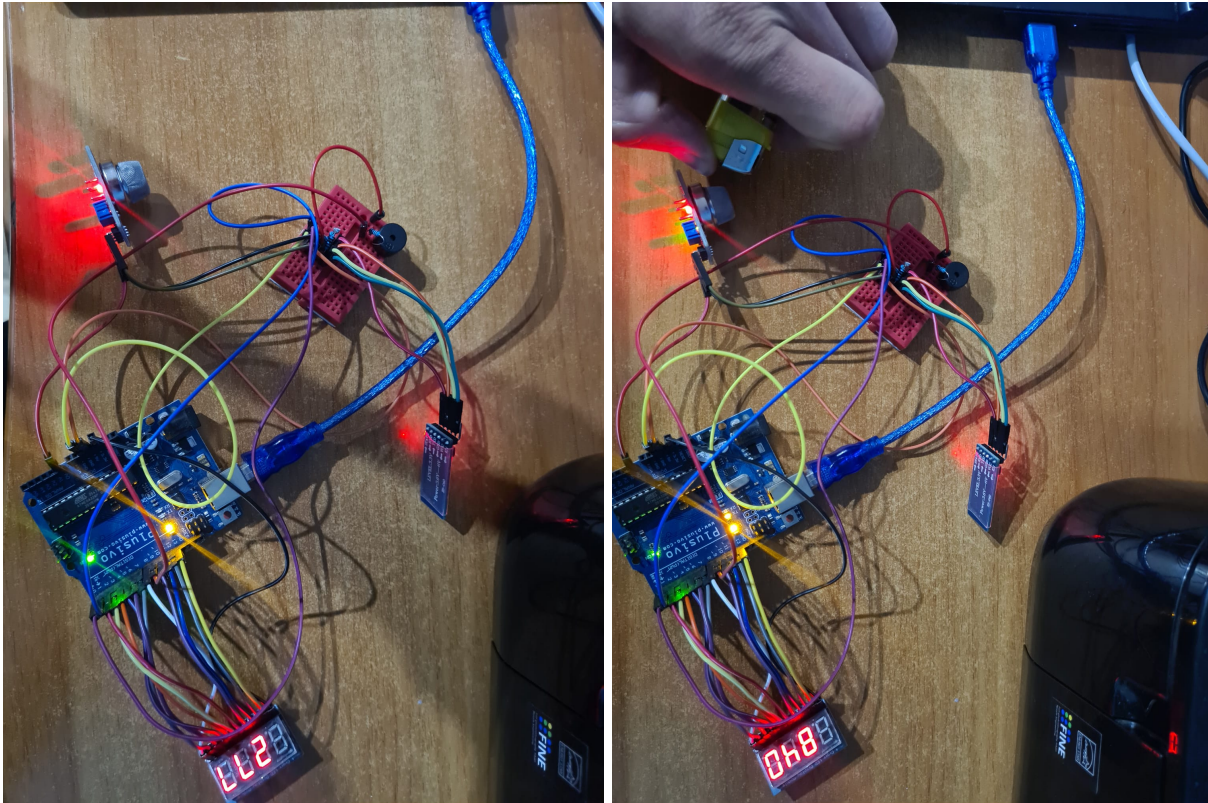
```
digitalWrite(segG, SEGMENT_ON);
break;
case 3:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_ON);
break;
case 4:
digitalWrite(segA, SEGMENT_OFF);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_OFF);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;
case 5:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_OFF);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;
case 6:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_OFF);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_ON);
digitalWrite(segE, SEGMENT_ON);
digitalWrite(segF, SEGMENT_ON);
digitalWrite(segG, SEGMENT_ON);
break;
case 7:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
digitalWrite(segD, SEGMENT_OFF);
digitalWrite(segE, SEGMENT_OFF);
digitalWrite(segF, SEGMENT_OFF);
digitalWrite(segG, SEGMENT_OFF);
break;
case 8:
digitalWrite(segA, SEGMENT_ON);
digitalWrite(segB, SEGMENT_ON);
digitalWrite(segC, SEGMENT_ON);
```

```
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_ON);  
digitalWrite(segF, SEGMENT_ON);  
digitalWrite(segG, SEGMENT_ON);  
break;  
case 9:  
digitalWrite(segA, SEGMENT_ON);  
digitalWrite(segB, SEGMENT_ON);  
digitalWrite(segC, SEGMENT_ON);  
digitalWrite(segD, SEGMENT_ON);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_ON);  
digitalWrite(segG, SEGMENT_ON);  
break;  
case 10:  
digitalWrite(segA, SEGMENT_OFF);  
digitalWrite(segB, SEGMENT_OFF);  
digitalWrite(segC, SEGMENT_OFF);  
digitalWrite(segD, SEGMENT_OFF);  
digitalWrite(segE, SEGMENT_OFF);  
digitalWrite(segF, SEGMENT_OFF);  
digitalWrite(segG, SEGMENT_OFF);  
break;  
}  
}
```



Rezultate Obținute

Foto proiect:



Aplicatie:



Concluzii

A fost un proiect interesant, m-am distrat realizandu-l. Mi-as fi dorit sa il aduc, estetic, intr-o stare putin mai buna, insa niciodata nu e prea tarziu:). Am avut cateva probleme cu placuta, dar s-au rezolvat in final. Per total sunt multumit de ce a iesit, dat fiind ca deviceul are si o utilitate in viata de zi cu zi.

Download

[dinu_dumitru-razvan.zip](#)

Jurnal

22.04.2022 - Realizare pagina wiki ce contine descrierea proiectului

08.05.2022 - Realizare schema bloc

27.05.2022 - Finalizare proiect + documentatie

Bibliografie/Resurse

Resurse Software:

<https://arduinogetstarted.com/faq/how-to-use-buzzer-without-blocking-other-code>

<https://arduinogetstarted.com/tutorials/arduino-buzzer-library>

<https://github.com/robsoncoto/arduino-songs/blob/master/merrychristmas/merrychristmas.ino>

Resurse Hardware:

<https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/>

<https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/>

<https://osoyoo.com/2017/08/08/4-digit-7-segment-led-display/>

Export to PDF

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/arosca/senzorgaze>



Last update: **2022/05/29 19:37**