

Tester alcoolemie

Autor: Popa Alexandru-Matei

Introducere

Proiectul constă în implementarea unui aparat, care poate detecta cantitatea de alcool din aerul expirat. Voi folosi un senzor MQ-3 pentru detecția alcoolului și un ecran LCD pentru a afișa cantitatea curentă. De asemenea, voi adăuga un buzzer pentru a semnaliza o posibilă cantitate prea mare de alcool.

Acest aparat este util, deoarece te informează cu privire la cantitatea de alcool din aerul expirat înainte de a te urca la volan. Proiectul poate fi folosit doar ca o simplă curiozitate sau ca o prevenire a unei posibile infracțiuni.

Descriere generală

Senzorul de alcool gazos MQ-3 este un senzor cu un timp de răspuns rapid și cu o sensibilitate înaltă. Acesta este alimentat la breadboard, de unde se preiau datele de ieșire. Placa Arduino transmite ecranului LCD cantitatea de alcool din aerul expirat și în cazul în care este destul de ridicată, buzzer-ul va semnaliza utilizatorul.

Senzorul MQ-3 are 4 pini, dintre care primul se conectează la tensiunea de 5V de pe microcontroller, al doilea la GND, al treilea la D0, dacă se dorește o reprezentare digitală a valorii senzorului, iar al patrulea la A0 la analog. Senzorul MQ3 este de tip MOS (metal oxide semiconductor) și poate detecta concentrații de alcool oriunde începând cu valori de la 25 ppm, ajungând până la 600 ppm (valori corecte, după ce a fost senzorul calibrat).

1 ppm = parts-per-million. O concentrație de 500 ppm de alcool înseamnă ca din 1.000.000 de molecule de gaz, 500 ar fi de alcool.

Modulul de Buzzer utilizează 3 pini: primul se conectează la tensiunea de 5V, al doilea la pinul 8 de la Digital PWM, iar ultimul reprezintă împământarea.

În final, modulul LCD 1602 permite afișarea a câte 16 caractere pe 2 linii, continuând un adaptor I2C. Astfel, se micșorează numărul de pini folosiți la 4. Primul este GND, al doilea tensiunea de 5V, iar al treilea (SDA - serial data) și al patrulea (SCL - serial clock) se duc la analog IN, pe A4, respectiv A5.

Schemă bloc



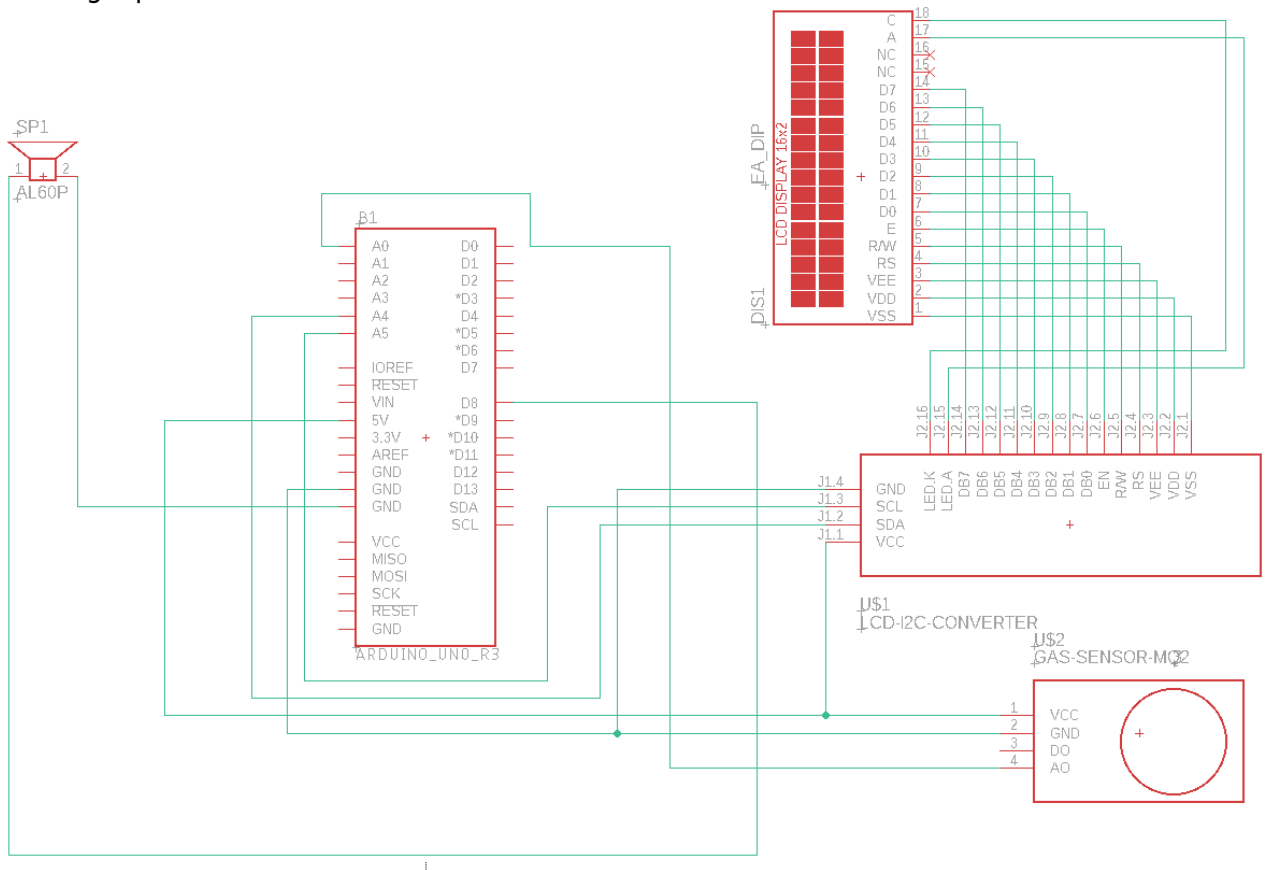
Hardware Design

Piese necesare

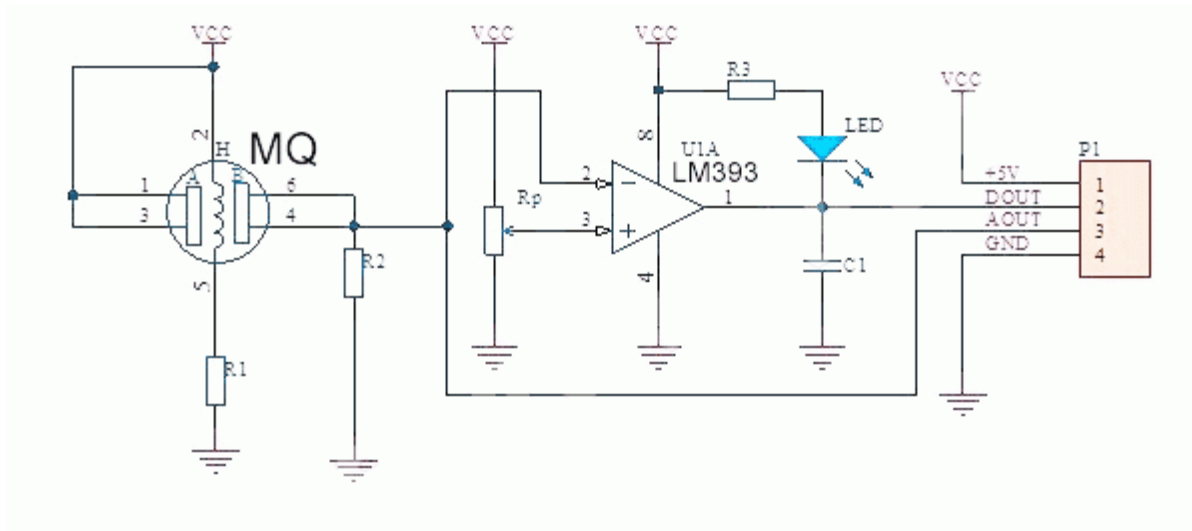
- placă Arduino
- senzor de gaze MQ-3
- breadboard
- ecran LCD 1602 cu I2C
- fire mamă-tată
- fire jumper
- modul buzzer

Schemă electrică

Am utilizat Eagle pentru schema electrică.



Schema internă a modului MQ-3



Software Design

Am utilizat Arduino IDE pentru partea de software și două librării:

- **Wire.h**: pentru a putea comunica cu I2C
- **LiquidCrystal_I2C.h**: pentru accesul la funcțiile necesare folosirii ecranului LCD

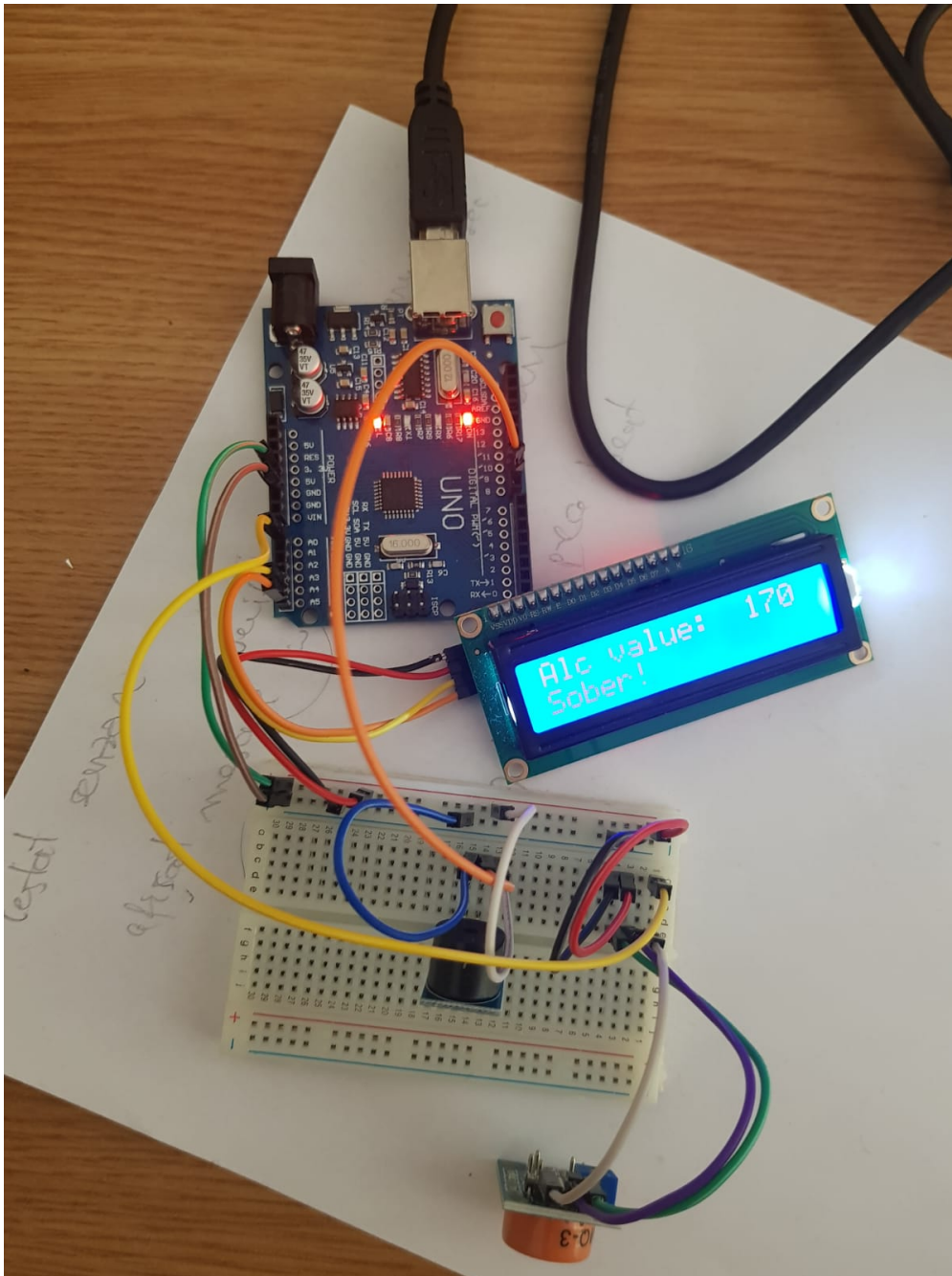
Funcțiile principale sunt **setup()** și **loop()**, în încercarea de a face codul cât mai modular.

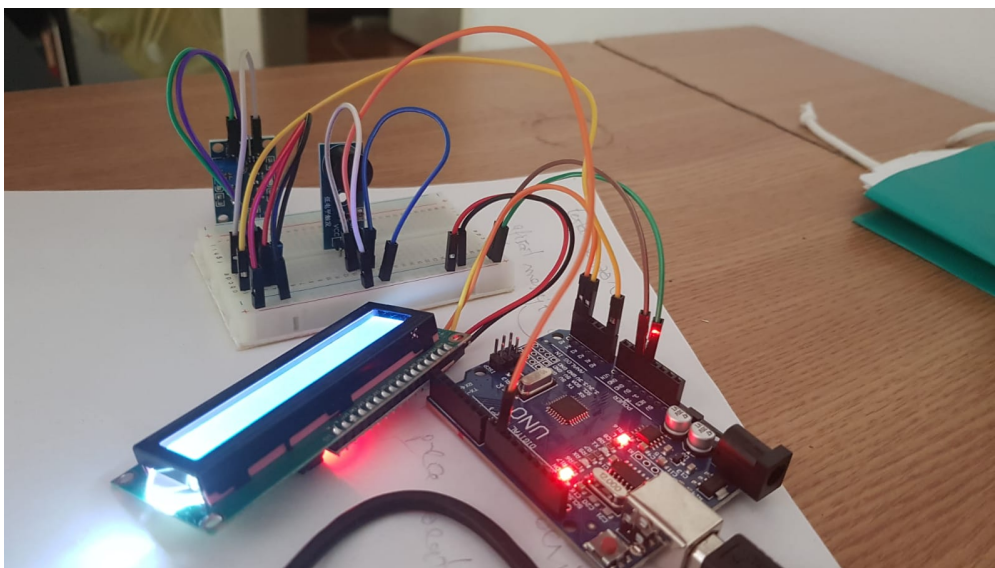
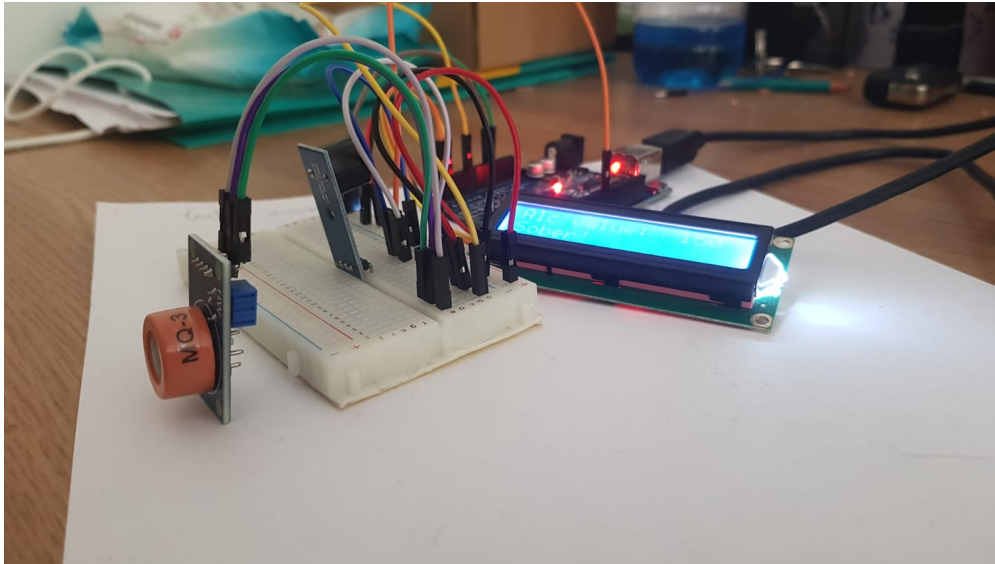
- În funcția **setup**, inițializez ecranul LCD, configurez pin-ul pentru buzzer și afișez pe ecran un mesaj corespunzător, lăsând la încălzit senzorul pentru câteva secunde.
- În funcția **loop**, calibrez senzorul, iau valoarea pe care acesta o transmite și afișez un mesaj specific în funcție de mai multe intervale.
 - *calibrateSensor()*: citesc valoarea data de senzor de pe pin-ul 0 cu `analogRead`. Fiind o valoare prea mare, a trebuit să calibrez sensibilitatea rezistenței potentiometrului din modul, pe care am setat-o la o altă valoare, după mai multe încercări. `RS_gas` este rezistența senzorului, care se schimbă în funcție de concentrația gazului, iar `R0` este rezistența la o valoare știută fără prezența altor gaze în jur. După o serie de ecuații, pe care le-am explicat mai jos, am aflat valoarea minimă și maximă, iar pentru a afișa o valoare cât mai ușor de recunoscut, am schimbat capetele intervalului prin funcția `map`.
 - *alcoholLevel()*: se face media a două valori, cu un anumit `delay` între ele, pentru o valoare cât mai corectă
 - *printLCD()*: se afișează un mesaj pentru o anumită valoare data de senzor. Dacă valoarea este prea mare, buzzer-ul scoate un sunet.

$RS = [(V_{in} \times R / V_{out}) - R]$, unde `R` este rezistența potentiometrului.

Pentru a calcula `R0` trebuie să aflăm valoarea lui `RS_gas` în aerul curat. Acest lucru se poate face luând valorile date de senzor și convertindu-le la tensiuni. Apoi se calculează `R0` cu formula $R0 = RS / 60$, unde 60 este raportul dintre rezistențe în aerul curat.

Rezultate obtinute





Demo

Am testat proiectul folosind spirt, caz în care valoarea arătată ar trebui să se apropie de maxim, ceea ce se și întâmplă. De asemenea, am mai testat proiectul și cu câteva beri și arată rezultatele dorite.

[Demo](#)

Concluzii

A fost un proiect interesant și mi-a plăcut să lucrez la el. Fiind primul proiect hardware pe care l-am făcut, am învățat lucruri noi și utile. Am întâmpinat câteva dificultăți în a calibra senzorul. În final, am reușit să-l fac să dea rezultate cât mai corecte, mai aproape de realitate. Ceea ce este important de știut, este că nu te poți baza pe valorile oferite de acest senzor, deoarece este unul ieftin, care nu oferă rezultate atât de exacte.

Download

- [schema_bloc_etilotest.png](#)
- [schema_el_etil.png](#)
- [poza_1_etilotest.jpeg](#)
- [poza_2_etilotest.jpeg](#)
- [poza_3_etilotest.jpeg](#)

[arhiva_cod.zip](#)

Jurnal

- **25 aprilie 2021:** crearea paginii de wiki, completarea temei proiectului, descrierea proiectului și adăugarea schemei bloc
- **19 mai 2021:** testarea componentelor, calibrarea senzorului MQ-3, crearea hardware a proiectului
- **21 mai 2021:** completarea descrierii, adăugarea schemelor electrice
- **26 mai 2021:** finalizarea părții software

Bibliografie/Resurse

- <https://lastminuteengineers.com/mq3-alcohol-sensor-arduino-tutorial/>
- https://wiki.seeedstudio.com/Grove-Gas_Sensor-MQ3/
- <https://components101.com/sensors/mq-3-alcohol-gas-sensor>
- <https://www.instructables.com/LCD-1602-With-Arduino-Uno-R3/>
- <https://thetempedia.com/tutorials/interfacing-mq-3-gas-sensor-with-evive/>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/avaduva/alcoholtester>



Last update: **2021/05/27 13:08**