

Breathalyzer (Etilotest)

Introducere

- * This project presents the development of a portable Breathalyzer device designed to estimate blood alcohol content (BAC) by analyzing a person's breath.
- * The main goal is to create an affordable and accessible solution that helps individuals assess their sobriety, promoting responsible behavior and improving public safety.
- * The idea came from myself :) Being a person who consumes alcohol (rather) often, I thought it would be a cool party idea to create such a device to know whether me or my friends drank too much.
- * I believe it's useful for me as well as for others because this way we can be sure whether we are allowed to drive or not. (maybe at a party you may have accidentally consumed alcohol, or someone pulled a nasty prank on your road-legal juice :)))

Descriere generală

Software:



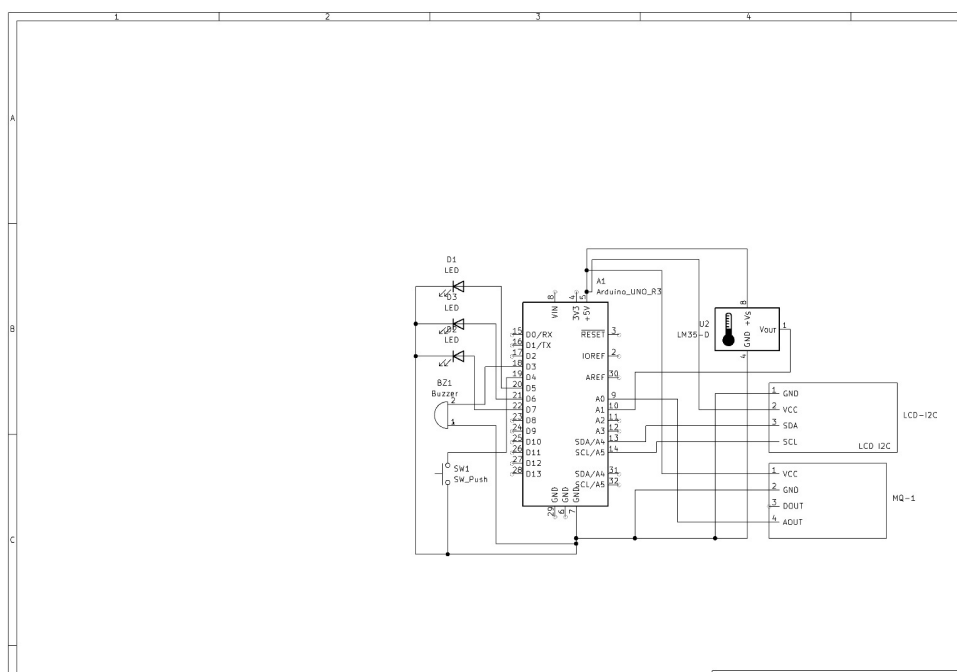
Hardware:



Hardware Design

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării



List of components:

- Arduino UNO R3 + cablu
- fire
- breadboard
- LCD 1602 I2C
- senzor MQ-3 alcool gaz
- senzor digital DS18B20 temperatura
- buton
- buzzer pasiv
- LED-uri (rosu, galben, verde pt. stari)

Pini utilizati:

- 5V: pentru alimentari LCD, senzori, LED-uri, buzzer
- GND: ground
- SDA si SCL: pentru I2C LED
- A5: pin analog MQ-6
- D4, D10, D5: LED-uri verde, galben (cu PWM), rosu
- D3: buzzer (PWM)
- D2: buton (cu intrerupere)

Software Design

Link GitHub: <https://github.com/NicuPolonicu/Breathalyzer>

Codul a fost dezvoltat in Arduino IDE. Biblioteci folosite:

- LiquidCrystal_I2C: pentru LCD pe interfata I2C

- DallasTemperature: pentru functiile legate de senzorul de temperatura
- Wire.h: clasica biblioteca Arduino :)

Funcțiile Arduino precum pinMode, delay, analogRead, folosire PWM, enableInterrupt au fost implementate manual prin AVR. (manipulare de registri)

Este important de subliniat faptul ca am folosit Timer1 pentru PWM si Timer0 pentru custom delay. Din cauza utilizarii Timer0, functii precum delay si millis (built-in Arduino) nu mai sunt utilizabile. (din moment ce se bazeaza pe Timer0 :)) Asadar, debouncing-ul pentru buton este implementat fara delay-uri, doar cu un simplu check in loop.

Algoritm detectie expiratie: detecteaza expiratia utilizatorului prin fluctuatiile in temperatura. Expiratia umana, fiind mai calda ca temperatura camerei, (24-25 grade Celsius vs. 30-32 grade Celsius) poate fi usor detectata daca temperatura inregistrata sare suficient de mult. (minim 2 cuantile de temperatura, adica 0,30 grade Celsius in cazul senzorului meu) Astfel, daca diferenta dintre temperatura trecuta si cea curenta este mai mare sau egala cu acest prag, se considera ca am detectat expiratie, si incep masurarea alcoolemiei.

Flow-ul codului:

- se initializeaza LCD-ul, LED-urile si pull-up-ul pentru buton
- se inregistreaza o temperatura initiala cu senzorul de temperatura (pentru a nu intra automat in masurare alcoolemie in loop)
- se initializeaza ADC pentru senzorul de alcool. Din moment ce acest senzor in particular are un heating time destul de mare (mai mult de o ora), un warmup pe loc nu este suficient. Cu toate acestea, am inclus si un scurt timp de warm-up pentru senzor in initializare, chiar daca senzorul este incalzit de dinainte (de acasa :))
- intram in loop: senzorul de temperatura masoara odata la ~0.4s temperatura camerei. (Nota: senzorul, cu rezolutia default de 12 biti, completeaza o conversie odata la ~1s, ceea ce era destul de incet pentru aplicatia mea. Asadar, am redus rezolutia ADC la 11 biti, reducand timpul intre read-uri la ~0.4s si avand o rezolutie de ~0.15 grade Celsius, mai mult decat suficient pentru aplicatie)
- daca se detecteaza respiratie (prin algoritmul simplu mentionat mai sus) si a trecut suficient timp de la ultima masurare de alcoolemie, incepe masurarea alcoolemiei - alternativ, o apasare de buton nu are acest cooldown intre masurari de alcoolemie, insa inceperea masurarii poate sa nu fie intocmai instantana, datorita delay-ului implicit pentru masurarea temperaturii in loop.

Flow masurare alcoolemie: - buzzer-ul va bipai pentru a semnifica pregatirea persoanei pentru suflat, iar un mesaj va aparea pe LCD

- odata ce buzzer-ul se opreste, incepe masurarea. LED-ul galben va lumina in functie de cat de mult timp mai are persoana sa sufle.

- dupa ce s-a terminat timpul de suflat (4s), se verifica valoarea maxima detectata de senzor.

*Daca depaseste un anumit prag, se considera ca persoana a consumat alcool. LED-ul rosu se aprinde, iar buzzer-ul bipaie puternic.

*Daca nu se depaseste pragul, LED-ul verde va lumina.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

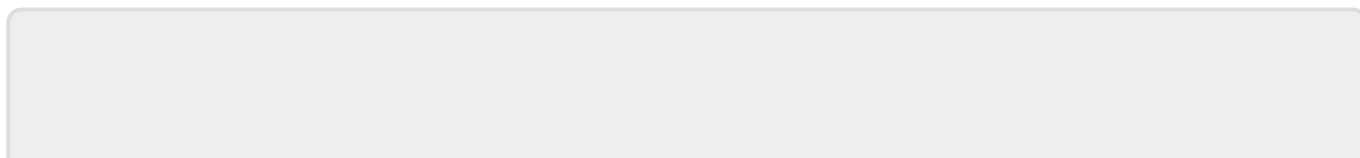
Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)



From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2025/eradu/dan.nicula>



Last update: **2025/05/23 11:19**