

Pulsoximetru

Autor

Sicoe Petru-Georgian

Introducere

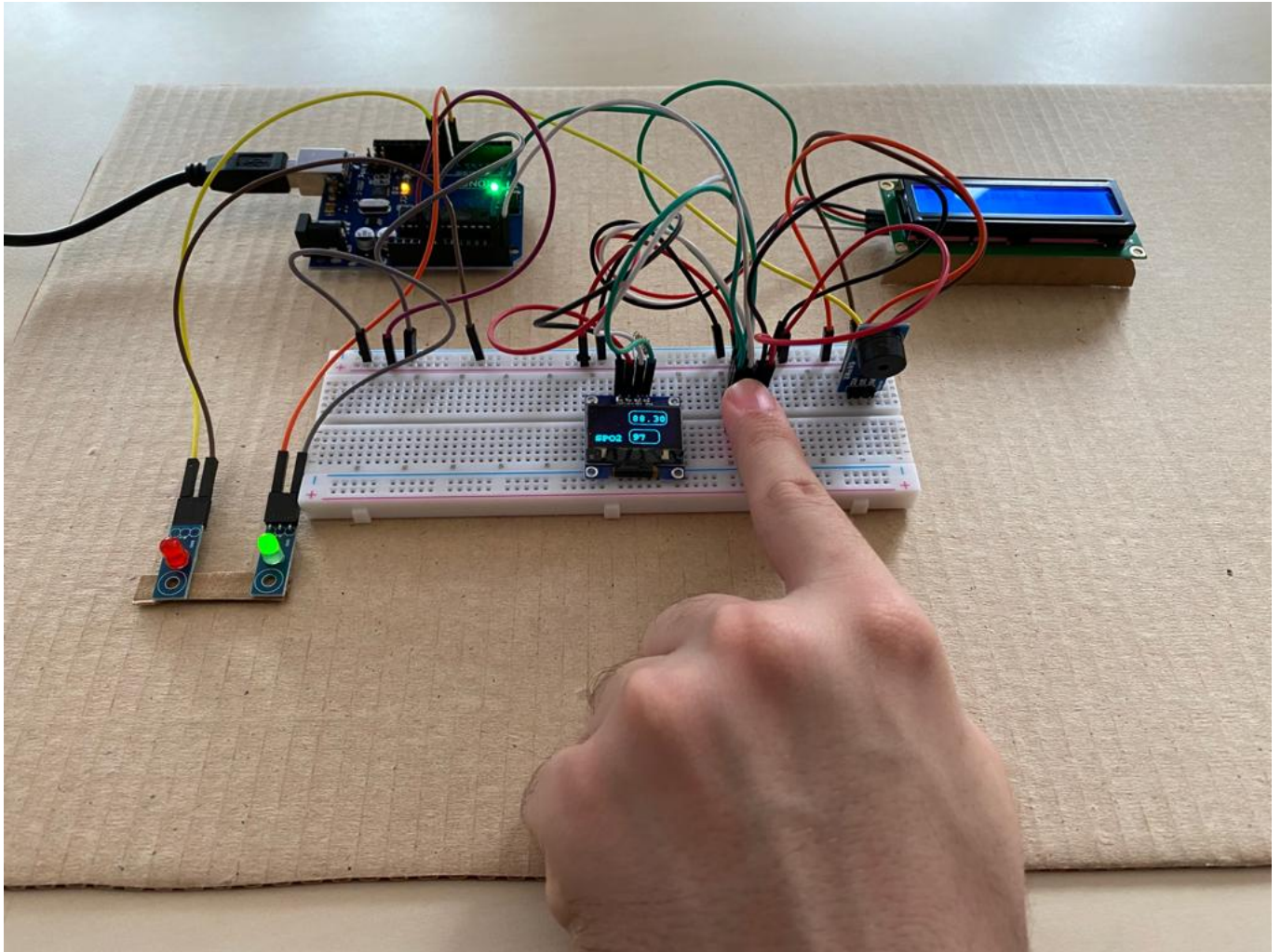
Proiectul presupune implementarea unui pulsoximetru, prin masurarea pulsului si calcularea concentratiei de SPO2 si afisand aceste 2 informatii pe un ecran LCD.

Astfel proiectul poate avea si o utilitate practica, permitand masurarea cu aproximatie a concentratiei de oxigen, nu doar in contextul pandemiei, ci si cu alte diferite ocazii, incurajand utilizatorul sa se mentina sanatos.

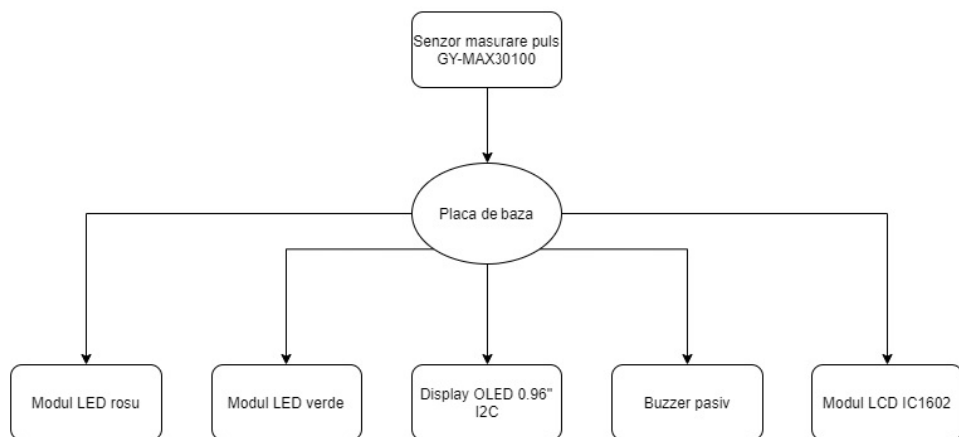
Descriere generală

Senzorul de ritm cardiac preia pulsul de la deget si trimite datele catre placa de baza, care le prelucreaza si care afiseaza pe un display OLED la final numarul de batai pe minut si concentratia de SPO2. Astfel, utilizatorul pune degetul pe senzor si va vedea pe ecran masuratorile dorite.

Pulsoximetrul este un dispozitiv care masoara saturatia de oxigen din sange. Oxigenul, dupa ajunge in plamani, este transportat prin corp cu ajutorul hemoglobinei din sange. Astfel, saturatia de oxigen se refera la procentul de hemoglobine care transporta oxigen din numarul total de hemoglobine. Senzorul folosit se bazeaza pe urmatoarea proprietate, numita si legea lui Beer: cantitatea de lumina absorbita este direct proportionala cu concentratia de substanta care absoarbe lumina, iar in cazul nostru hemoglobina absoarbe lumina. De asemenea, hemoglobina transportatoare de oxigen absoarbe mai mult lumina infrarosie decat lumina rosie, iar hemoglobina care nu transporta oxigen absoarbe mai mult lumina rosie decat cea infrarosie. Astfel, cu 2 LED-uri si un fotodetector se poate masura saturatia de oxigen din sange.



Schema bloc

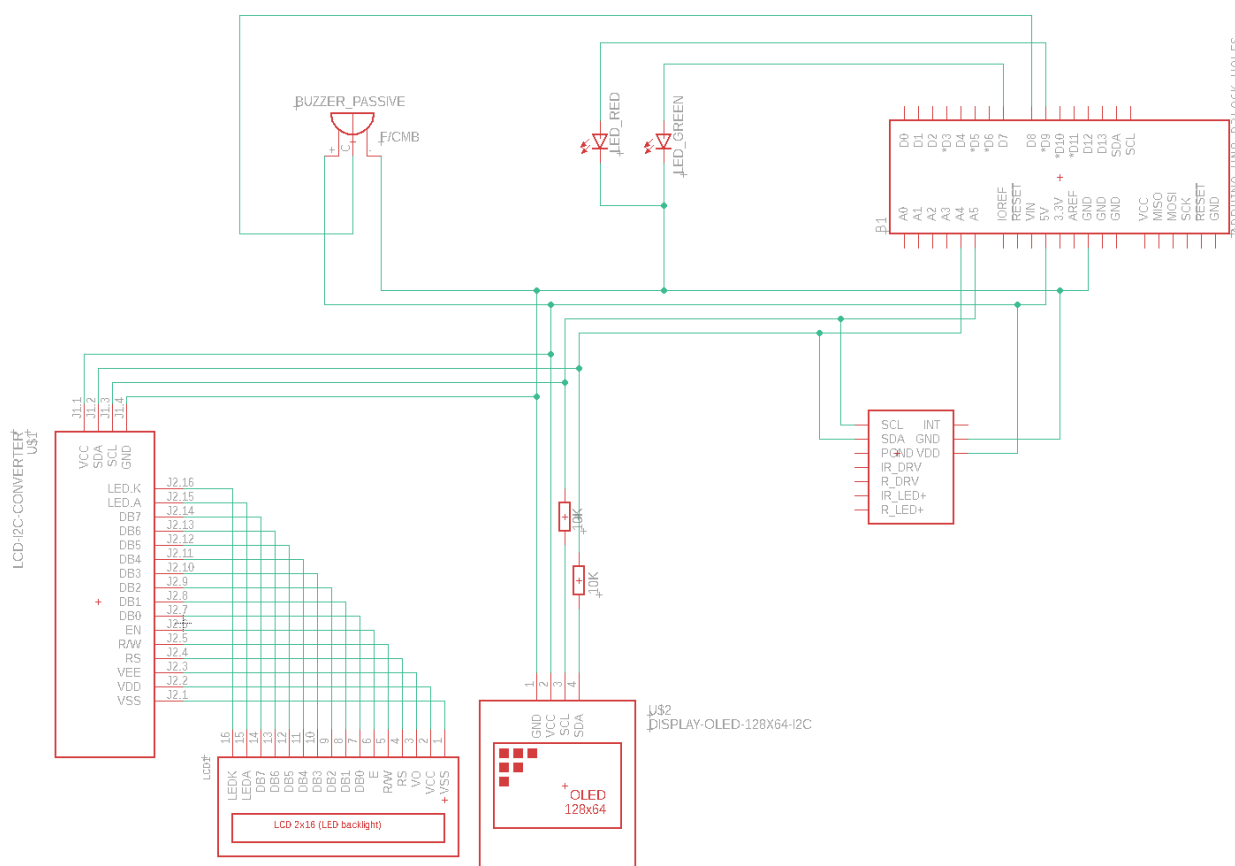


Hardware Design

Lista de piese:

- Arduino Uno
- Breadboard
- Display OLED 0.96 I2C
- Modul senzor ritm cardiac GY-MAX30100
- Modul LCD tip IC1602
- Buzzer pasiv
- Led rosu
- Led verde
- Fire de legatura
- 2 rezistente 10 KOhm

Schema electrica



Descrierea codului:

- mediu de dezvoltare: Arduino IDE
- Biblioteci folosite:
 - OLED display: `OakOLED.h`, `Adafruit_SSD1306.h`, `Fonts/FreeMonoBold9pt7b.h`
 - LCD: `LiquidCrystal_I2C.h`, `Wire.h`
 - Senzor: `"MAX30100_PulseOximeter.h"`
- Algoritmi și structuri de date:
 - logică simplă (if, while, for, etc)
 - structuri de bază C (variabile, constante, etc)
- Functii implementate
 - **setup():** sunt initializate LCD-ul si display-ul OLED, iar pe LCD este printat titlu proiectului. Dupa aceea este verificat daca senzorul este initializat corect. In cazul in care este, se afiseaza mesajul 'SUCCESS', este setat fontul pentru OLED, este pornit senzorul si este setata functia de callback pentru fiecare bataie detectata. Daca initializarea esueaza, pe OLED va aparea mesajul 'FAILED' si se intra intr-un loop. La final mai sunt setate ledurile GREEN si RED si buzzerul ca OUTPUT.
 - **loop():** masurarea propriu-zisa incepe abia dupa 2 secunde de la terminarea functiei de setup. Dupa aceea la fiecare secunda vor fi facute masuratorile dorite. Cu functiile de biblioteca `getHeartRate()` si `getSPO2()` aflu pulsul, respectiv concentratia de SPO2 din sange. Atunci cand nu este detectat niciun deget pe senzor (cand SPO2 masurat este zero), se aprinde ledul rosu si se stinge ledul verde. Analog, atunci cand degetul este detectat pe senzor, ledul verde se aprinde si ledul rosu se stinge. Daca degetul este detectat, valorile masurate sunt printate pe display.

Altfel, va fi printat mesajul 'SPO2...', sugerand faptul ca degetul nu este detectat sau ca nu s-a terminat masurarea valorilor dorite.

- **onBeatDetected()**: functie care este apelata de fiecare data cand este detectata o bataie a inimii. Aceasta afiseaza pe display un bitmap cu o inima si apeleaza functia buzz().
- **buzz()**: functie care face buzzerul sa sune. Sunetul este produs prin apelarea de 80 de ori a digitalWrite de HIGH/LOW pe portul 8 la o frecventa mare.

Rezultate obtinute

- [Cod sursă](#)
- [Video Demo](#)

Concluzie

Proiectul la PM a fost pentru mine in mod cert o experienta interesanta care mi-a oferit multa satisfactie la final. M-a ajutat sa imi solidific cunostiintele dobandite in cadrul laboratorului intr-un mod practic si faptul ca la final am ceva care chiar merge imi ofera multa incredere si motivatie sa incerc pe viitor ceva mai complex. Greutatile intampinate la mine au fost sa gasesc o biblioteca potrivita cu tipul senzorului pe care il am si sa integrez functionalitatea senzorului in cadrul proiectului. Cel putin la functia de setup am petrecut ceva timp pe partea de debug, deoarece senzorul nu putea fi initializat.

Bibliografie/Resurse

- <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>
- <https://github.com/netguy204/OakOLED>
- https://github.com/adafruit/Adafruit_SSD1306
- <https://www.electronicclinic.com/wp-content/uploads/2020/02/MAX30100.zip>
- https://www.howequipmentworks.com/pulse_oximeter/

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/pulsoximetru>



Last update: **2021/06/02 21:45**