

Pulsoximetru

Introducere

M-am gandit sa fac un pulsoximetru pentru a-mi masura pulsul si nivelul de saturatie de oxigen din organism

Descriere generală

Pulsoximetrul măsoară modificările de absorbție a luminii de către cele două forme de hemoglobină: oxigenată și neoxigenată.

Hemoglobina ce transportă oxigenul (oxihemoglobina) absoarbe radiații infraroșii (800-940 nm), iar hemoglobina fără oxigen absoarbe lumină roșie din spectrul vizibil (600-700 nm). Măsurătorile au la bază un principiu optic, senzorii putând fi plasați la nivelul degetului, lobului urechii, cât și antebrățului, având două surse de emisie, una de lumină roșie în spectrul vizibil cu lungime de undă de 660 nm (650 nm după unii autori) și una invizibilă în spectrul infraroșu cu lungime de undă de 940 nm (950 nm după unii autori), precum și cel puțin un fotodetector. În infraroșu coeficientul de absorbție al oxihemoglobinei este mai mare de cât al hemoglobinei reduse (neoxigenate). Partea electronică măsoară raportul lumină pulsatilă/lumină nepulsatilă și îl exprimă în lungimi de undă, în spectrul roșu și infraroșu, și ulterior îl transpune în procent de saturație în oxigen.

Hardware Design

Aceste componente au fost utilizate in dezvoltarea pulsoximetrului:

- Regulator de Tensiune de 5V L7805 TO-220
- Arduino Pro Mini
- Ecran OLED 0.96" comunicare I2C
- Conector Baterie 9V Modul Adaptor Micro USB

- Senzor max30102

- Buton

Schema functionala: <https://imgur.com/a/qKSKUsM>

Poza cu pulsoximetrul (partea hardware): <https://imgur.com/a/Zr9ZHea>

Software Design

Descrierea codului aplicației (firmware):

mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR) librării și surse 3rd-party (e.g. Procyon AVRlib) algoritmi și structuri pe care plănuieți să le implementați (etapa 3) surse și funcții implementate

Modulul senzorial max30102 integrează un LED roșu și un LED infraroșu, un fotodetector, un dispozitiv optic și un circuit electronic cu zgomot redus cu suprimarea luminii ambientale. Odată pornit pulsoximetrul, se pune un deget pe senzor și se apasă ușor (pentru a nu pune presiune pe senzor să se strice) pentru a se citi ritmul cardiac și concentrația de hemoglobină oxigenată în timp real. În timp ce degetul stă amplasat, se poate apăsa degetul pentru a arăta valorile curente. După o perioadă de inactivitate, pulsoximetrul se va stinge automat.

```
#include "ssd1306h.h"
```

```
#include "MAX30102.h"
```

```
#include "Pulse.h"
```

Acestea sunt header-ele pentru clasele care au fost integrate pentru componentele circuitului

`#include "avr/pgmspace.h"`: Această bibliotecă este specifică pentru platforma AVR și oferă funcții și macro-uri pentru lucrul cu memorie programabilă (flash) pe microcontrolerul AVR. Memoria programabilă este utilizată pentru a stoca codul program al microcontrolerului. Biblioteca "avr/pgmspace.h" oferă funcții și macro-uri pentru a accesa și manipula datele stocate în memoria program.

`#include "EEPROM.h"`: Această bibliotecă oferă funcții pentru lucrul cu memoria EEPROM (Electrically Erasable Programmable Read-Only Memory) pe microcontrolerul AVR. EEPROM este o memorie non-volatilă utilizată pentru stocarea datelor care trebuie menținute chiar și atunci când dispozitivul este alimentat. Biblioteca "EEPROM.h" oferă funcții pentru a scrie și citi date din memoria EEPROM a microcontrolerului AVR.

`#include "avr/sleep.h"`: Această bibliotecă oferă funcții și macro-uri pentru gestionarea modului de somn (sleep mode) pe microcontrolerul AVR. Modul de somn permite microcontrolerului să intre

Într-o stare de economisire a energiei prin reducerea consumului de curent la nivel minim. Biblioteca "avr/sleep.h" oferă funcții pentru a controla modul de somn al microcontrolerului, cum ar fi setarea modului de somn, activarea sau dezactivarea întreruperilor în timpul somnului și altele.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Videoclipul cu pulsoximetrul functional: <https://youtu.be/p2UwBuCtMiE>

Concluzii

1. Merge ca uns. Poate ar fi mers sa folosesc un senzor mai bun, dar isi face treaba.
2. Mi-am redescoperit pasiunea pentru asamblare si inginerie. Partea de hardware a fost cea mai interesanta si placuta, de asemenea si cea mai frustranta, etapa a proiectului.

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.

Cod Arduino (fara clasele in C++ pentru fiecare componenta): [nanopulseppg.zip](#)

Jurnal

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

05.05 - ideea de pulsoximetru imi vine in minte

07.05 - research si planificare

08.05 - lista de cumparaturi facuta

11.05 - ajung piesele. partea de hardware se face

13.05 - s-a ars un regulator :(fac switch de la alimentare cu baterie la cablu micro usb

16.05 - cursa pana la HobbyMarket Berceni sa iau adaptorul pentru micro usb

18.05 - retusare hardware

20.05 - implementare software

Bibliografie/Resurse

<https://iotprojectsideas.com/pulse-oximeter-using-arduino-max30100/>

[Export to PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2023/amocanu/denis_marius.vladu



Last update: **2023/05/28 15:23**