

Fitness Tracker

Introducere

Prezentarea pe scurt a proiectului:

- numara pasii parcursi
- masoara pulsul (bataile inimii)
- sincronizeaza datele cu telefonul prin Bluetooth si aplicatie mobila

Pentru ce e util?

- motiveaza sa faceti mai multa miscare
- va ajuta sa aveti grija de sanatatea dvs.

Descriere generală

Schema Bloc:



Arduino primește constant datele de la accelerometru și le trimite la modulul Bluetooth care le trimite la telefon pentru sincronizare și afișarea numărului de pași și temperatura. În interfața aplicației mobile există un buton, care porneste algoritmul de măsurare a bătăilor inimii pe minut. Acest algoritm rulează 6 secunde, ceea ce provoacă blocarea celorlalte calculări. După ce se termină măsurarea BPM se afișează rezultatul pe telefon și se întoarce la celelalte calcule.

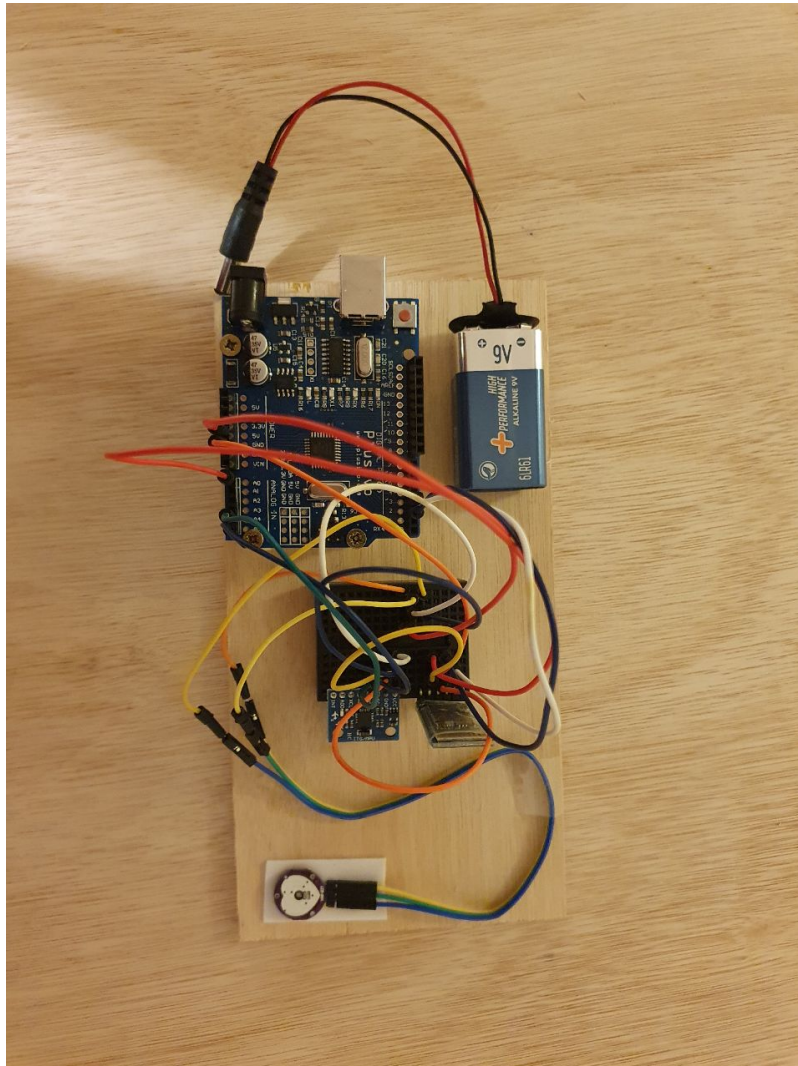
Hardware Design

Lista de piese:

- Arduino UNO
- Bluetooth HC-05
- Pulse XD-58C

- Accelerometer - MPU6050
- Fire

Design



Conexiunile si schema electrica



Software Design

Codul aplicatiei (firmware) consta din:

- mediu de dezvoltare Arduino IDE
- [librăria RemoteXY](#) instalata din Arduino Library Manager
- [libraria MPU6050](#)
- algoritm de a calcula bataile inimii pe minut
- algoritm de a detecta pasul

Descrierea codului

setup()

Initializeaza structurile de date, varaibilele si libraria mpu (initializeaza comunicatia si seteaza senzorul).

compute_pulse()

Functia timp de 6 secunde citeste datele de la senzor si detecteaza daca s-a efectuat o bataie completa. Aceste 6 secunde au fost alese pentru ca rezulta o valoare mai exacta si e timp destul de mic ca sa nu se piarda conexiunea Bluetooth cu telefonul. Ca sa putem asocia un rezultat al pulsului cu o bataie de inima, am luat doua valori de referinta 550 si 520, daca e mai mica de 520 atunci a fost efectuata o bataie, iar in momentul cand am citit o valoare mai mare de 550, calculam timpul scurs intre aceste doua citiri si calculam numarul de batai pe minut folosind aceste date. Dupa ce s-au scurs 6 secunde se calculeaza media acestor BPM, acesta va fi rezultatul final si cel trimis pentru sincronizare pe telefon.

compute_steps()

Pentru a deduce algoritmul de numarare a pasilor, am conectat arduino la laptop cu firmware care afiseaza valorile X, Y, Z citite de la accelerometrul MPU6050. Dupa am deschis Serial Plotter si ma plimbam prin camera cu laptop si arduino conectat. De aici am dedus ca la fiecare pas valoarea Z depaseste 19500 (presupun ca aceasta valoare e doar pentru inaltimea mea si pasul meu). Algoritmul consta in verificarea daca variabila Z e mai mare ca 19500, atunci se incrementeaza numarul de pasi si se calculeaza ce procentaj din scopul zilnic (pentru testare 2000) sunt parcursi.

get_temp()

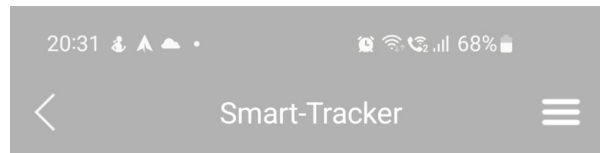
Se citeste temperatura de la senzorul MPU6050.

loop()

Aceasta e functia care ruleaza la infinit cat timp placuta Arduino e conectata la tensiune. Se verifica daca a fost apasat butonul pentru masurarea BPM si se apeleaza functia `compute_pulse()`. La infinit se apeleaza si functiile `compute_steps()` si `get_temp()` fara nicio verificare.

Rezultate Obținute

Interfata grafica a aplicatiei pentru sincronizare cu tracker:



214 pasi



84 BPM

28.4 °C

Calculeaza pulsul

Demo:

[YouTube](#)

Concluzii

Acest proiect m-a invatat cum e sa proiectezi un device de la zero, care include si hardware si software. Am invatat cum sa deduc pattern-uri din datele citite din lumea reala si sa le utilizez in proiect. Am invatat cum comunica Arduino prin Bluetooth cu alte device-uri. Consider ca acest "fitness-tracker" isi atinge scopurile: calculeaza bine pulsul cu o eroare mica si numara bine pasii.

Download

[Arhiva cu surse si scheme](#)

Bibliografie/Resurse

Link-uri utile:

- [Milestone 1](#)
- [Library Example](#) folosit pentru intelegerea a cum se citesc date de pe MPU6050
- [Heart Beat Sensot ECG Display](#) algoritm folosit pentru masurarea BPM, putin update-at
- [RemoteXY](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/amocanu/smart-tracker>

Last update: **2022/05/23 00:22**

