

FitBox

Autor: [Ioana Laura Stăncioiu](#)

Grupa: 336CA

Introducere

Proiectul constă în dezvoltarea unui tracker pentru activitatea fizică. Dispozitivul te ține la curent cu numărul de pași, distanța parcursă și kaloriile consumate și te motivează în încercarea de a deveni o persoană mai activă. Pentru ca informațiile să fie ușor de vizualizat acestea sunt afișate în timp real pe telefon, printr-o conexiune de tip bluetooth. De asemenea, trackerul îți reamintește să fii activ prin intermediul unui semnal sonor.

Descriere generală

Componentele principale ale dispozitivului sunt:

- accelerometru, folosit pentru calcularea numărului de pași
- modul bluetooth, prin intermediul căruia se realizează conexiunea dintre Arduino și aplicația mobilă
- buzzer, emite un semnal sonor dacă au trecut 10 secunde de când utilizatorul s-a mișcat ultima oară, de asemenea te felicită din 10 în 10 pași

Aplicația mobilă ¹⁾ afișează următoarele informații:

- Numărul de pași
- Caloriile consumate
- Distanța parcursă(km)
- Activitate (mesaj sugestiv despre starea utilizatorului)



Schemă bloc



Hardware Design

Componente

Nume	Număr Piese
Arduino Nano	1
Modul Bluetooth HC-05	1
Piezo Buzzer	1
Accelerometru ADXL345	1
Perfboard 5×7	1
Rezistență 100 Ω	1
Rezistență 1 kΩ	1
Rezistență 2 kΩ	1
Baterie 9V	1
Conector baterie 9V	1
Întreprător	1
Baretă 40 pini mamă-tată	2

Schema electrică



Modulul bluetooth HC-05 suportă tensiuni de maxim 3.3V pe pinul RX, astfel a fost nevoie să adaug un divizor de tensiune pentru a limita tensiunea de 5V de la pinul TX de pe placa Arduino.

Software Design

Mediul de dezvoltare

- **Arduino IDE** → dezvoltarea codului și încărcarea acestuia pe plăcuța Arduino
- **Autodesk Eagle** → realizarea schemei electrice
- **draw.io** → realizarea schemei bloc

Biblioteci folosite

- **Wire** (I2C communication)²⁾
- **Adafruit_ADXL345** (ADXL345 accelerometer)³⁾
- **Adafruit Sensor** (ADXL345 accelerometer)⁴⁾
- **Adafruit BusIO** (I2C communication)⁵⁾

Structura codului

Algoritm numărare pași^{6) 7) 8)}

Pentru calculul accelerației, se face o medie pe cele 3 axe, pe un sampling size de 100. Apoi, se calculează magnitudinea accelerației, aceasta fiind folosită în calcule ulterioare. Pentru a detecta un pas se verifică dacă diferența dintre două magnitudini succesive este mai mare ca un threshold, caz în care se înregistrează un pas.

Starea utilizatorului

Pentru interactivitatea aplicației, odată cu semnalul sonor, utilizatorul primește și un mesaj sugestiv. Valorile posibile sunt:

- Move! → au trecut 10 secunde de când a fost înregistrat un pas
- One more! → încurajează utilizatorul să se miște în continuare
- Good job! → utilizatorul s-a deplasat 10 pași

Valorile de 10 secunde și 10 pași sunt alese în scop pur demonstrativ. Într-un context din viața reală acestea ar fi mult mai mari, pentru o apreciere corectă a activității.

Funcții

- *setup()* → inițializează accelerometrul și apelează funcția *readAvgAccel* pentru calibrarea sa
- *loop()* → logica programului, apelează în ordine *countSteps*, *checkNumSteps*, *checkMovement* și *sendData*
- *readAvgAccel()* → calculează accelerația media pe axele x, y și z, folosită ca un punct de referință pentru calcule ulterioare
- *countSteps()* → face diferența dintre două accelerații succesive, dacă aceasta este mai mare decât o limită prestabilită, se incrementează numărul de pași
- *getAccel()* → calculează accelerația
- *checkMovement()* → verifică dacă utilizatorul s-a mișcat în ultimele 10 secunde, dacă îl atenționează printr-un semnal sonor
- *checkNumSteps()* → felicită utilizatorul din 10 în 10 pași printr-un semnal sonor
- *sendData()* → trimite datele aplicației de pe telefon prin interfața serială UART

Rezultate Obținute

Produs final



Circuit final



Demo

[Youtube Link](#)

Concluzii

Mi-a plăcut foarte mult să lucrez la acest proiect și consider ca am învățat multe despre cum pot face un dispozitiv funcțional cu o utilitate practică. Experiența nu ar fi fost completă fără lipirea componentelor pe perfboard, proces, care deși a durat mai mult decât aproximaseam inițial, m-a ajutat să dobândesc skill-uri noi și a fost pretty fun per total .

Cu toate că algoritmul de pași mai poate fi îmbunătățit, nefiind suficient de precis, sunt destul de mulțumită de produsul finit.

Download

- [Proiect Github](#)
- [fitbox.zip](#)

Jurnal

- **21.04.2022:** alegere temă proiect
- **26.04.2022:** publicare descriere wiki
- **02.05.2022:** comandă piese
- **05.05.2022:** finalizare circuit hardware
- **07.05.2022:** finalizare software
- **15.05.2022:** lipire componente pe perfboard
- **25.05.2022:** finalizare pagină wiki

Bibliografie/Resurse

[Export to PDF](#)

- ¹⁾ <https://play.google.com/store/apps/details?id=com.frederikhauke.ArduTooth&hl=ro&gl=US>
- ²⁾ <https://www.arduino.cc/en/reference/Wire>
- ³⁾ https://github.com/adafruit/Adafruit_ADXL345
- ⁴⁾ https://github.com/adafruit/Adafruit_Sensor
- ⁵⁾ https://github.com/adafruit/Adafruit_BusIO
- ⁶⁾ http://web.cs.wpi.edu/~emmanuel/courses/cs528/F20/slides/papers/deepak_ganesan_pedometer.pdf
- ⁷⁾ <https://www.analog.com/media/en/technical-documentation/technical-articles/pedometer.pdf>
- ⁸⁾ <https://www.engineersgarage.com/arduino-based-walking-steps-distance-calculator-adxl345/>

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2022/avaduva/activity_tracker



Last update: **2022/05/27 21:28**