Documentație

SportWatch

Durla Andreea-Simona

Grupa 30238

1. Introducere

Prin acest proiect se dorește realizarea unui dispozitiv electronic care să contorizeze numărul de pași. Dispozitivul va fi autonom energetic (se va alimenta de la baterii), va afișa numărul de pași realizați și are functie de cronometru.

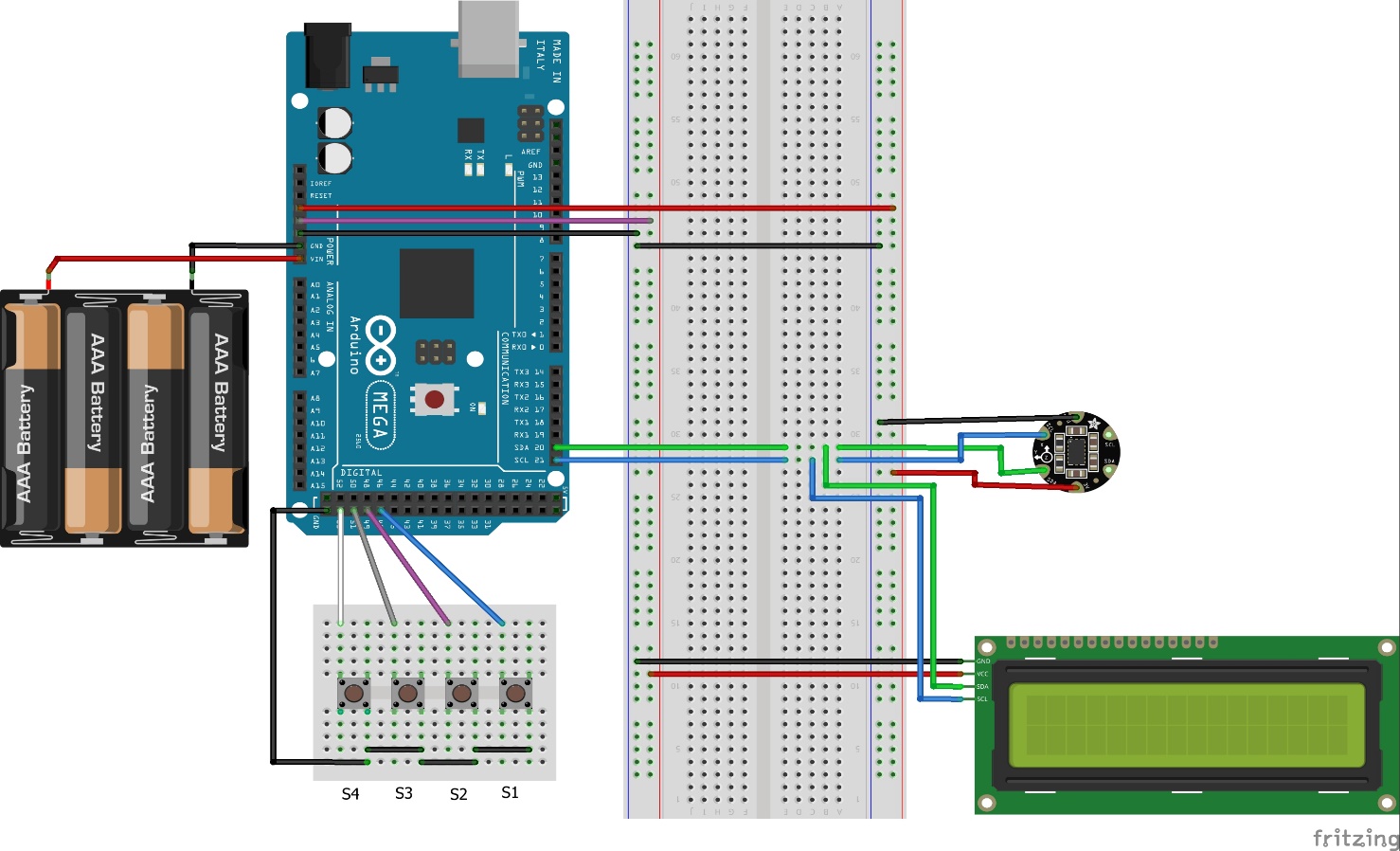
Pentru a realiza acest dispozitiv am ales să utilizez placa de dezvoltare Arduino ATmega 2560, accelerometrul MPU-6050 (inițial) / Adafruit Flora LSM303DLHC (final), display-ul LCD cu 2 linii / 16 caractere pe linie, si un modul de 4 butoane.

Am ales inițial accelerometrul MPU-6050, deoarece este un circuit integrat care, pe lângă accelerometru, oferă și o funcție de giroscop. Deoarece după efectuarea unor teste accelerometrul a încetat să mai transmită informație, și nu am mai găsit altul în stoc la magazinele online, am optat să îl schimb cu un alt accelerometru - Adafruit Flora LSM303DLHC. Acest accelerometru oferă accelerațiile în sistemul de coordonate atașat cipului; cipul nu oferă informații de tip giroscop.

Ca display, am considerat că este suficienta utilizarea unui ecran LCD cu 2 rânduri / 16 caractere pe rând.

Am optat pentru un modul cu 4 butoane pentru a putea implementa funcții precum alegerea unei opțiuni din meniul principal, start/pauza/stop/revenire în submeniul Cronometru, resetare număr pași/revenire în submeniul Pedometru.

1. Schema conexiunilor electrice



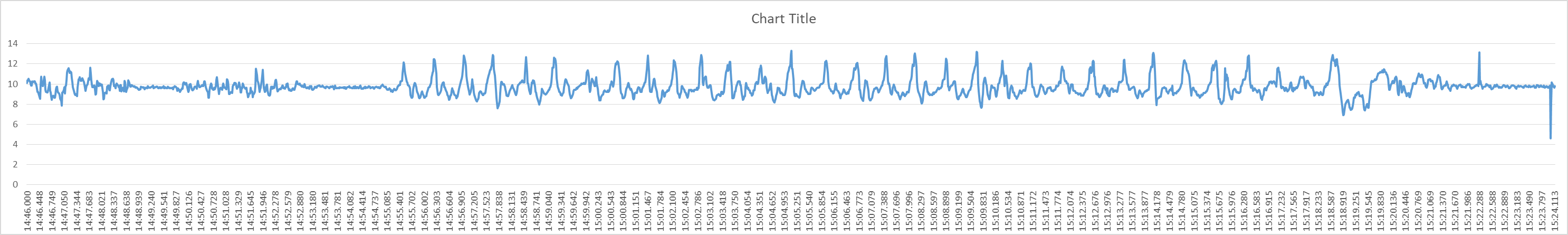
Pinoutul conexiunilor este:

|  |  |  |
| --- | --- | --- |
| Pin Arduino | Pin dispozitiv | Obs. |
| 3.3V | Vcc Adafruit Flora LSM303DLH |  |
| GND | GND Adafruit Flora LSM303DLH |  |
| SDA | SDA Adafruit Flora LSM303DLH | Comunicație I2C, adresa 0x19 |
| SCL | SCL Adafruit Flora LSM303DLH | Comunicație I2C |
| 5V | Vcc display |  |
| GND | GND display |  |
| SDA | SDA display | Comunicație I2C, adresa 0x27 |
| SCL | SCL display | Comunicație I2C |
| GND | Masa modul 4 butoane |  |
| 47 | Intrare digitală comandată de butonul S1 | Pin digital configurat ca PULL-UP |
| 49 | Intrare digitală comandată de butonul S2 | Pin digital configurat ca PULL-UP |
| 51 | Intrare digitală comandată de butonul S3 | Pin digital configurat ca PULL-UP |
| 53 | Intrare digitală comandată de butonul S4 | Pin digital configurat ca PULL-UP |

1. Inregistrari facute pentru a dectata valoarea de prag

Detecția pașilor se realizează prin urmărirea valorilor accelerației și alegerea unei valori prag – atunci când accelerația măsurată va depăși valoarea prag, se va incrementa numărul pașilor.

Am utilizat aplicația Cooltermwin, pentru a înregistra valorile transmise de dispozitiv.

****

Pe axa X sunt redate momentele de timp aferente înregistrării.

Pe axa Y este reprezentată evoluția accelerației înregistrare.

Intervalul de timp 14:46.000 – 14:55.085 (in format minute:secunde.milisecunde) reprezintă momentul în care am pornit înregistrarea datelor.

Intervalul de timp 14:55.401 – 15:16.915 reprezintă înregistrările pașilor pe care i-am facut.

Intervalul de timp 14:17.232 – 15:20.466 reprezintă momentul în care m-am așezat la birou.

Intervalul de timp 15:20.466 – final reprezintă momentul în care am oprit înregistrarea datelor.

Din analiza acestor date am concluzionat că pragul de detecție al unui pas este de 11.53 m/s2.

1. Manual de utilizare

Dispozitivul SportWatch este prevăzut cu două funcții: pedometru si cronometru.

Funcția de pedometru este activa din momentul in care dispozitivul este pornit, numărând pașii făcuți de utilizator.

Funcția de cronometru realizează cronometrarea timpului. Cronometrul are 3 opțiuni: start, pauză și stop.

Dispozitivul afișează informația pe un display cu 2 rânduri a câte 16 caractere, și are 4 butoane pentru controlul funcțiilor, notate S1…S4.

După pornirea aparatului, pe display este afișat meniul principal care cuprinde 2 opțiuni:

1. Pedometru
2. Cronometru

Utilizatorul poate accesa funcția Pedometru prin apăsarea butonului S1. Pe display se va afișa mesajul „Nr. de pasi: ” si numărul de pași numărat de la pornirea dispozitivului până în momentul afișării. Utilizatorul poate reseta valoarea numărată prin apăsarea butonului S3. Pentru a se întoarce la meniul principal, se va apăsa butonul S4. Pedometrul va continua să numere pașii în fundal.

Funcția Cronometru se accesează prin apăsarea butonului S2. Pe display se va afișa meniul aferent funcției de cronometru:

1.Start 2.Pauza

3. Stop 4.Inapoi

Prin apăsarea butonului S1 se va porni cronometrul. Pe display se va afișa timpul cronometrat din momentul apăsării Start până în prezent. Dacă utilizatorul dorește, poate întrerupe temporar cronometrarea prin apăsarea butonului S2 – opțiunea pauză. Cronometrarea se poate continua prin apăsarea butonului S1.

Pentru a opri cronometrul (indiferent dacă cronometrul este activ sau este în pauză), se va apăsa butonul S3. Pe display se va afișa valoarea finală cronometrată. O nouă cronometrare se poate realiza prin apăsarea butonului S1.

La orice moment de timp, se poate reveni la meniul principal prin apăsarea butonului S4. Cronometrul va continua să execute în fundal opțiunea activă (va continua să cronometreze / va menține pauza), chiar si atunci când utilizatorul accesează funcția Pedometru - butoanele S1-S4 însă vor executa comenzi aferente funcției Pedometru. La revenirea în meniul 2.Cronometru, se va afișa informația corespunzătoare (timpul cronometrat din momentul apăsării Start până în prezent / timpul unde a fost pusă pauza / meniul aferent funcției de cronometru – dacă înainte de revenirea din meniul 2.Cronometru a fost apăsat butonul S3 – Stop). Pedometrul va continua să măsoare pașii, în paralel cu cronometrul.

1. Explicarea codului

Pentru implementarea programului, se utilizează funcții din bibliotecile Wire.h (pentru comunicatie I2C intre accelerometru si Arduino, Arduino și display), Adafruit\_Sensor.h și Adafruit\_LSM303\_U.h (pentru accesarea informației achiziționate de accelerometrul Adafruit Flora LSM303DLHC), LiquidCrystal\_I2C.h (pentru afișarea informațiilor pe ecranul LCD).

Pentru implementarea funcției de pedometru, se citesc de la accelerometru valorile accelerațiilor pe axele X, Y și Z, si se calculează accelerația rezultantă cu ajutorul formulei

Pentru a număra un pas, se pune condiția ca valoarea curentă a accelerației rezultante să fie mai mare decât valoarea de prag stabilită, iar această valoare să fie prima valoare citită peste prag (fapt semnalizat prin valoarea booleană PasActiv; această variabilă va fi resetată atunci când valoarea curentă a accelerației rezultante scade sub valoarea prag. Funcția Pedometru funcționează permanent, începând de la lansarea în execuție a programului. Codul care implementează funcția pedometru se regăsește atât în funcția loop(), cât și în funcțiiile afiseazaPedometru() și afiseazaCronometru().

Pentru afișarea pe ecranul LCD, se folosesc funcțiile clear (pentru stergerea informației afișate pe ecran), print (pentru transmiterea informației noi care va fi afișată) și setCursor (care va fi folosită pentru aranjarea informației afișate pe ecran). Deoarece funcția delay() induce o stare de suspendare totală a microprocesorului (acesta nu executa nici o instrucțiune pe perioada specificată ca parametru al funcției delay), este de evitat utilizarea delay (mai ales pentru programe care trebuie să execute instrucțiuni rapid, cum este cazul acestui program care trebuie să proceseze cât mai rapid informația legată de accelerație). Astfel că se va emula funcția delay() cu ajutorul unei structuri

TimpN2 = millis(); // TimpN2 va fi instanțiat cu valoarea curentă a millis().

if (TimpN2 - TimpN1 >= secunda){ // dacă diferența dintre timpul curent și timpul memorat anterior // este mai mare de o secundă

TimpN1 = TimpN2; // timpul memorat anterior este suprascris cu timpul curent

…… // se executa instrucțiunile de afișare pe ecranul LCD dorite

}

Meniul principal, afișat în corpul loop(), cuprinde 2 opțiuni: 1. Pedometru și 2. Cronometru. La apăsarea butonului S1, se va apela funcția afiseazaPedometru, care va afișa meniul Pedometru și va trata funcțiile sale. La apăsarea butonului S2, se va apela funcția afiseazaCronometru, care va afișa meniul Cronometru și va trata funcțiile sale.

Funcția *afiseazaPedometru()* asigură afișarea numărului de pași efectuați până la momentul respectiv. Codul aferent funcției afiseazaPedometru() rulează într-o buclă while, condiționată de butonul S4 – dacă se apasă butonul S4, se părăsește bucla while.

Prin apăsarea butonului S3 se reseteaza numărul de pași.

Funcția *afiseazaCronometru ()* asigură afișarea submeniurilor aferente cronometrului.

La începutul execuției subrutinei, se setează ca fiind adevărată o variabilă booleană arataCronometru – care marchează faptul că acum se rulează subrutina afiseazaPedometru().

Algoritmul aferent funcției afiseazaCronometru() rulează într-o buclă while, condiționată de valoarea true a variabilei arataCronometru.

Afișarea pe ecranul LCD se face contextual, în funcție de stările variabilelor *meniuCronometruPornit*, *cronometruPauza*, *cronometruStop*, astfel:

* dacă *meniuCronometruPornit* este false, cronometrul nu este pornit in fundal și se afișează meniul principal
* dacă *meniuCronometruPornit* este true, *cronometruPauza* este false și *cronometruStop* este false, cronometrarea este activă și se afișează valoarea cronometrată. Valoarea cronometrată se obține prin citirea timpului curent cu ajutorul millis(), și scăderea momentului de timp când a fost pornit cronometrul, moment memorat în variabila cronometruInit. Deoarece valoarea obținută este exprimată în milisecunde, este necesară convertirea într-un format ore:minute:secunde.milisecunde.

Meniul principal al subrutinei afișează cele 4 opțiuni ale funcției Cronometru: 1.Start 2.Pauza 3.Stop 4.Inapoi.

La apăsarea butonului S1, aferent funcției Start, se pornește cronometrul. Se memorează într-o variabilă *cronometruInit* timpul inițial obținut cu ajutorul funcției millis(). Se setează ca adevărate variabilele boolene *cronometruPornit*, pentru a marca faptul că cronometrul a fost pornit, *meniuCronometruPornit* pentru a activa submeniul subrutinei afiseazaCronometru() – care afișează timpul scurs de la pornirea cronometrului, și se setează ca fals variabila booleană *cronometruStop*.

Pentru a pune pauză funcției cronometru, trebuie să fie simultan adevărate următoarele condiții:

* să fie apăsat butonul S2,
* să fie pornit cronometrul (*cronometruPornit*=true),
* cronometrul să nu fie pus deja pe pauză (*cronometruPauza*=false)
* să fie activ meniul *arataCronometru*=true.
* deoarece dacă se părăsește meniul cronometru, și apoi se revine, revenirea se face tot prin apăsarea butonului S2, este posibil ca la apăsarea acestuia să se acceseze meniul cronometru dar simultan să se pună accidental și pauză. De aceea, se utilizează o variabilă activeMenu, care se instanțiază cu ajutorul funcției millis(), de fiecare dată când se accesează subrutina *afiseazaCronometru().* Pentru a activa pauza cronometrului, se adaugă o condiție suplimentară ca activeMenu să fi fost instanțiată cu cel puțin 500 milisecunde (meniul să fi fost accesat deja de minim 500 milisecunde), înainte de apăsarea butonului S2 – pauză. Se evită astfel dubla apăsare și pauzarea accidentală a cronometrului la reaccesarea meniului acestuia.

După validarea activării pauzei, se setează următoarele variabile:

* Variabila booleană *cronometruPauza* se setează ca adevărată
* Într-o variabilă globală *momentPauza*, se memoreaza momentul de timp cand s-a pus pauza

Pentru a opri cronometrul, trebuie să fie simultan adevărate următoarele condiții:

* Să fie apăsat butonul S3
* Cronometrul să fie pornit deja (*cronometruPornit*=true)
* să fie activ meniul *arataCronometru*=true.

După validarea comenzii, se setează următoarele variabile:

* *cronometruPauza*=false
* *cronometruPornit*=false
* *cronometruStop*=true; se marchează faptul că s-a oprit cronometrul. Există două scenarii posibile: utilizatorul pornește o nouă cronometrare, și atunci se va afișa noua valoare cronometrată; utilizatorul se întoarce în meniul principal SportWatch, iar la întoarcerea în opțiunea de Cronometru, se va afișa meniul principal al acestuia, cu cele 4 opțiuni.

Dacă utilizatorul dorește să se întoarcă în meniul principal al SportWatch, indiferent de starea în care se află cronometru (start, pauză, sau stop), se apasă butonul S4.

După validarea comenzii, se setează următoarele variabile:

* *arataCronometru*=false – nu se mai afișează meniul principal sau submeniurile Cronometru
* dacă s-a apăsat butonul Stop, *cronometruStop* devine false și *meniuCronometruPornit* devine false – deoarece butonul Stop a fost apăsat, nu mai este nici o cronometrare pornita și se invalidează afișarea submeniului care arată timpul cronometrat; la revenirea în Cronometru, se va afișa meniul principal cu cele 4 opțiuni (1.Start 2.Pauza 3.Stop 4. Inapoi).