

Seismograf

Autor: Bogdan Ciobanu

Grupa: 335CA

Introducere

Scopul proiectului este implementarea unui senzor de vibratii, optimizat pentru frecvențele cutremurelor, care să afiseze pe un display o seismogramă, și să alerteze sonor în cazul în care se depășeste un anumit prag.

Descriere generală



Senzorul de vibratii va fi un senzor piezoelectric ceramic, lipit pe capacul unei sticle. Scopul este de a amplifica vibratiile de frecvențe foarte joase, specifice cutremurelor. Output-ul modulului de senzor va fi afisat pe un display e-ink, fiind o piesă ce consumă curent puțin și nu emite lumina. În ultimul rand, alerta sonora va fi pornita în momentul în care vibratiile trec de un prag predefinit, moment în care se va porni un buzzer pasiv la o frecvență înaltă, astfel încât să alerteze oamenii din jur.

Hardware Design

Lista de componente:

1. 1xArduino UNO R3
2. 1xBuzzer
3. 1xSenzor piezoelectric
4. 1xDisplay e-ink 1.54inch
5. Fire

Dată fiind numărul relativ mic de conexiuni, nu a fost nevoie să folosesc un breadboard sau placă de prototipare, ci am conectat toate piesele direct la Arduino, utilizând majoritatea pinilor disponibili.

Software Design

În `setup()` se initializează pinii, timer-ul și obiectul e-paper display. Există o perioadă de initializare a memoriei interne display-ului, care durează aproximativ 3 secunde, în care display-ul face flicker.

În `loop()` se verifică dacă s-a ajuns în punctul de actualizare a display-ului, caz în care va desena graficul. În afara de asta, va trimite curent către buzzer în cazul în care bitul de `ring_buzzer` este setat pe true.

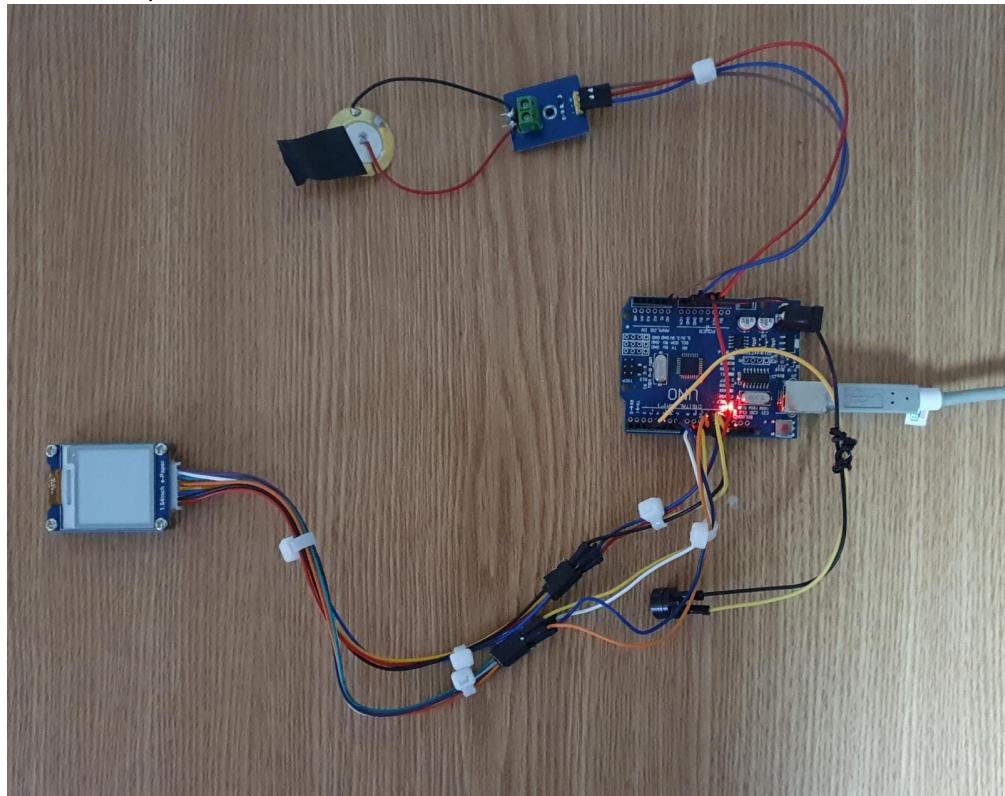
În acest timp, Timer 1 este setat la o frecvență de 21 de hz, frecvență la care am ales să fac sampling de la senzorul piezo. Aceasta frecvență este data de faptul că am un buffer de 64 de elemente pentru valorile venite de la senzor, iar în momentul în care buffer-ul se umple, generez graficul și îl trimit către display, fiind cantitatea maximă de date care ar putea încăpea pe display care să fie să liștibila. Asadar, la fiecare 46ms se apelează funcția `add_value`, care preia valoarea obținută de la senzor, calculează un offset față de rolling-average-ul buffer-ului (acest pas este necesar datorită zgromotului

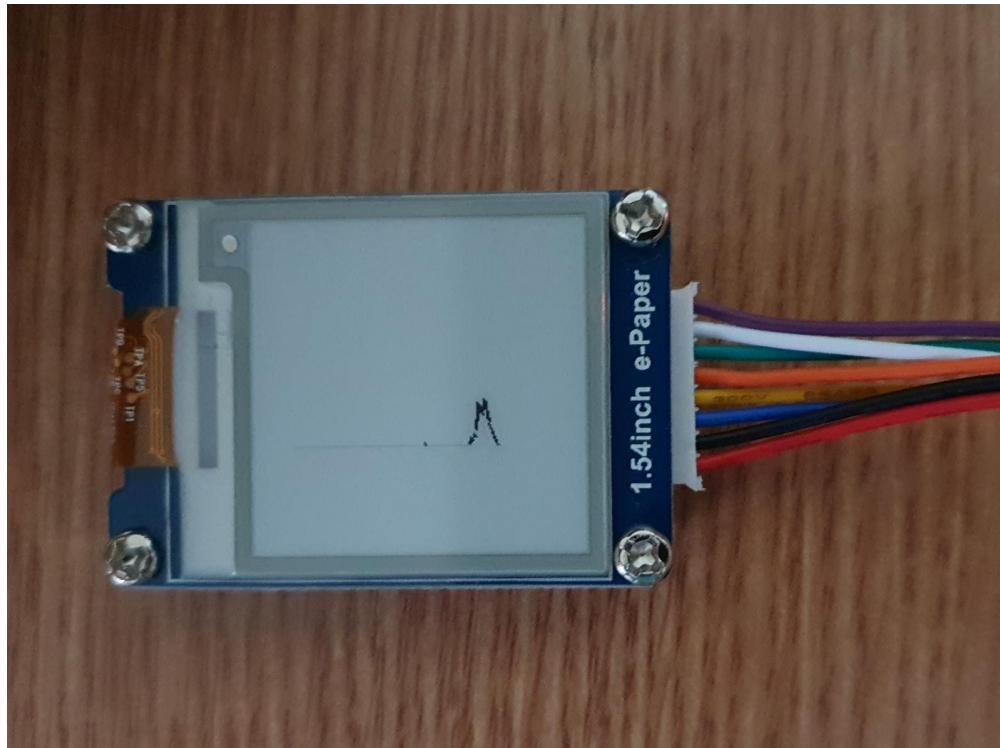
inerent senzorului, acesta fiind destul de sensibil). Dupa acesta, se verifica daca acest offset trece de un prag prestabil, caz in care `ring_buzzer` e setat pe true. La final, se verifica daca s-a facut loop-around pe buffer, caz in care se seteaza si flag-ul `draw_pending`.

Pe partea de display, acesta are o biblioteca cu functionalitati limitate pentru Arduino, data fiind memoria mica de care dispune microcontroller-ul. Aproximativ jumata din memoria RAM e utilizata de obiect-ul `epaper-display`, cu ajutorul caruia se poate comunica prin SPI cu display-ul. In momentul in care flag-ul `draw_pending` e setat, se apeleaza functia `draw_graph`. Aceasta va updata o arie de 50×136 pixeli de pe display cu graficul aferent buffer-ului de la senzor. Deoarece display-ul are un DPI destul de mare, ca liniile graficului sa fie clar vazute, acestea sunt practic cvadruplicate pe toate 4 directiile, un pixel din grafic fiind inconjurat mereu de alti 4 pe directiile sus-jos-stanga-dreapta. Acest QoL trick vine totusi cu un cost al numarului de sample-uri care incape pe display, intrucat daca acesta nu ar fi fost aplicat, am observat ca empiric ar fi incaput aproximativ 150 de sample-uri in acea arie de pe display, la costul vizibilitatii.

Rezultate obtinute

Toate componentele asamblate arata in acest fel:





La acest [link](#), se poate vizualiza un demo al functionalitatii proiectului.

Concluzii

A fost o experienta foarte buna sa lucrez hands-on pe Arduino, si am avut ocazia sa imi consolidez informatiile obtinute la curs si laborator. Probabil cea mai complicata parte a proiectului a fost lucrul cu display-ul, data fiind resursele reduse ale microcontroller-ului. De asemenea, o alta provocare a fost gasirea unei solutii pentru eliminarea zgomotului de la senzor, care sa fie memory and CPU friendly.

Download

[Cod Sursa](#)

Jurnal

25 Apr 2021: Created page, added hardware design image

20 Mai 2021: Finalizare proiect fizic

31 Mai 2021: Adaugat sectiune Hardware Design, adaugat schema EAGLE

31 Mai 2021: Adaugat sectiune Software Design

31 Mai 2021: Adaugat sectiuni Rezultate & Concluzii

Bibliografie/Resurse

[Seismograf](#)

[Documentatie Display](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2021/alazar/seismograf> 

Last update: **2021/05/31 18:31**