

Seismograf

Autor: Bogdan Ciobanu

Grupa: 335CA

Introducere

Scopul proiectului este implementarea unui senzor de vibratii, optimizat pentru frecventele cutremurelor, care sa afiseze pe un display o seismograma, si sa alerteze sonor in cazul in care se depaseste 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 frecvente foarte joase, specifice cutremurelor. Output-ul modulului de senzor va fi afisat pe un display e-ink, fiind o piesa ce consuma curent putin si nu emite lumina. In ultimul rand, alerta sonora va fi pornita in momentul in care vibratiile trec de un prag predefinit, moment in care se va porni un buzzer pasiv la o frecventa inalta, astfel incat sa 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

Dat fiind numarul relativ mic de conexiuni, nu a fost nevoie sa folosesc un breadboard sau placuta de prototipare, ci am conectat toate piesele direct la Arduino, utilizand majoritatea pinilor disponibili.

Software Design

In `setup()` se initializeaza pinii, timer-ul si obiectul e-paper display. Exista o perioada de initializare a memoriei interne display-ului, care dureaza aproximativ 3 secunde, in care display-ul face flicker.

In `loop()` se verifica daca s-a ajuns in punctul de actualizare a display-ului, caz in care va desena graficul. In afara de asta, va trimite curent catre buzzer in cazul in care bit-ul de `ring_buzzer` este setat pe true.

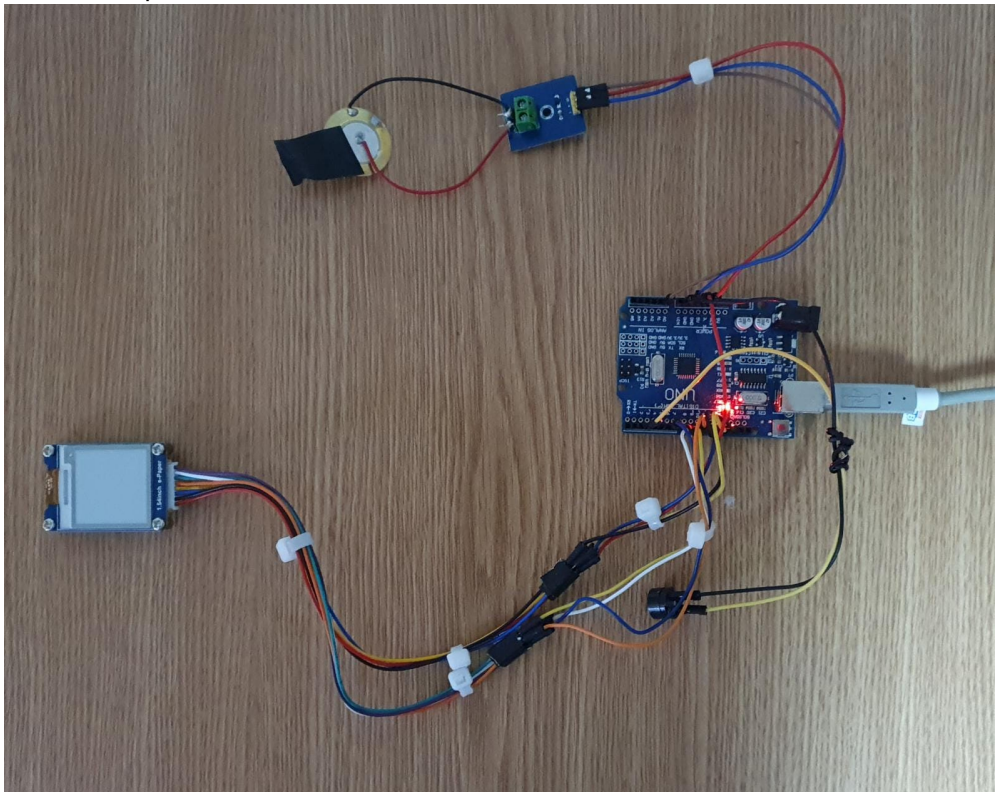
In acest timp, Timer 1 e setat la o frecventa de 21 de hz, frecventa la care am ales sa fac sampling de la senzorul piezo. Aceasta frecventa e data de faptul ca am un buffer de 64 de elemente pentru valorile venite de la senzor, iar in momentul in care buffer-ul se umple, generez graficul si il trimit catre display, fiind cantitatea maxima de date care ar putea incapa pe display care sa fie si lizibila. Asadar, la fiecare 46ms se apeleaza functia `add_value`, care preia valoarea obtinuta de la senzor, calculeaza un offset fata de rolling-average-ul buffer-ului (acest pas este necesar datorita zgomotului

inerent senzorului, acesta fiind destul de sensibil). Dupa acesta, se verifica daca acest offset trece de un prag prestabilit, 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`.

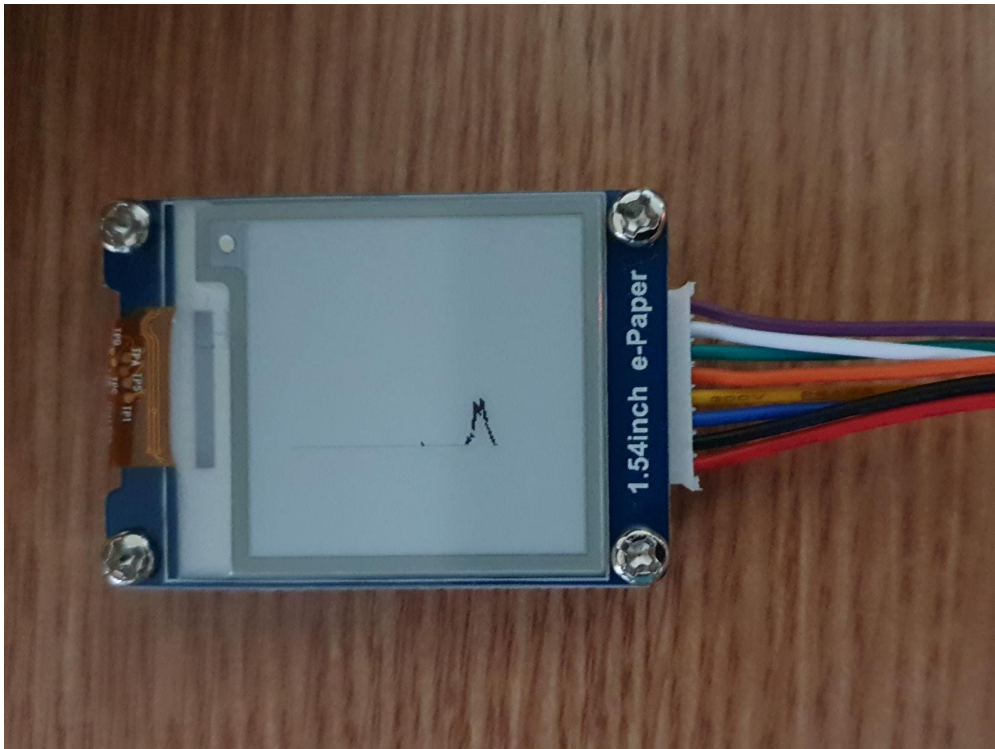
Pe partea de display, acesta are o biblioteca cu functionalitati limitate pentru Arduino, data fiind memoria mica de care dispune microcontroller-ul. Aproximativ jumatate din memoria RAM e utilizata de obiect-ul `epaper-display`, cu ajutorul caruia se poate comunica prin SPI cu display-ul. In momentul in care flag-ul `draw_pending` e setat, se apeleaza functia `draw_graph`. Aceasta va update 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 cvadruplate 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 incapa 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 lizibilitatii.

Rezultate obtinute

Toate componentele asamblate arata in acest fel:



Graficul pe display:



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**