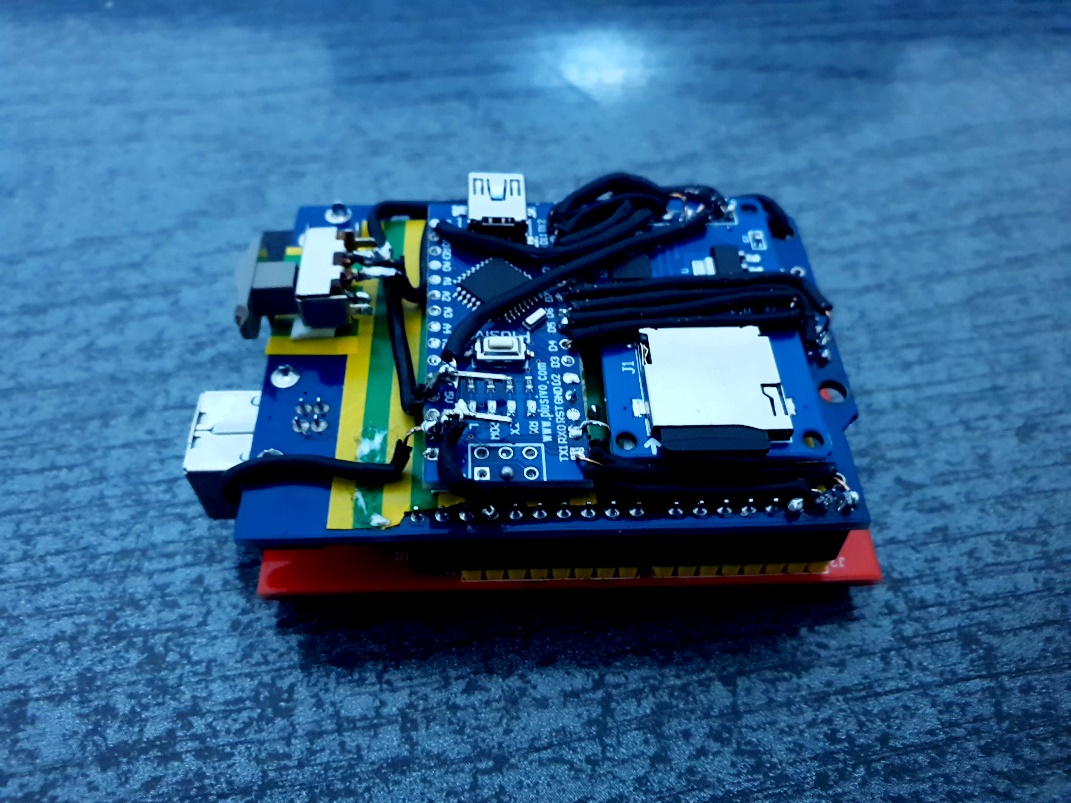
2020

Realizat de:

Hașciar Vlad Andrei

Dunca Ioana Florentina

Pocket Algorithm



# Concept

# Desi informatica si tehnologia capata pe zi ce trece un statut din ce in ce mai important in viata cu zi, resursele de invatare si modul de predare din scoli poate da o impresie plictisitoara si neinteresanta liniilor de cod, informatica fiind perceputa de multi ca o materie grea si mult prea greu de invatat. Poket Algorithms propune o solutie: introducerea unor prezentari simplificate si vizuale a unor algoritme populare le face cu mult mai usor de invatat pentu un amator dar e de folos si pentru un programator experimentat.

# Ideea proiectului este sa cream un dispozitiv compact si portabil menit sa foloseasca ca o librarie pentru algoritmele importante si des intalnite in informatica pentru a le oferi programatorilor incepatori o cale sa invete dar si sa vizualizeze maiusor functionarea acestora. Proiectul e menit sa fie usor de folosit si de a da posibilitatea adaugarii usoare a mai multor proiecte pe aceasi placa.

# Folosind o placa Nano ca programator ICSP putem incarca multiple programe care sa ruleze pe placa Arduino Uno principala. Acest concept ne-a ajutat sa adaugam programe noi mai usor.

## Planificarea si evolutia conceptului

## Pentru inceput am vrut doar sa cream un mediu in care sa putem pune toate informatiile si programele la un loc. Dupa o scurta perioada de luat in considerare alte alternative, am luat decizia de folosi Arduino pentru a adauga un element fizic proiectului. Nu ne-am gandit insa cata dificultate v-a putea adauga asta realizarii produsului final.

## Ce-a mai grea parte si costisitoare in ceea ce priveste timpul de lucru a fost formarea unei structuri pe care sa putem combina toate elementele pe care ni le doream sa le integram, bateriile fiind in mod special problematice.

## Prioritatea numarul 1 a fost mereu sa tinem toate componentele intr-un mod compact, de asta am vrut sa tinem bateriile intre placa si display, primele incercari fiind cu baterii de telefon, pentru a da optiunea de reincarcare, insa voltajul acestora prea mic s-a dovedit a fi o problema. Astfel am schimbat sursa de incarcare cu niste baterii AAA care pot fi usor inlocuibile.

## Un alt blocaj pe care l-am intalnit a fost cu spatiul de stocare, placa avand doar 32kb de memorie flash, a fost evident de la inceput ca un Arduino Uno normal nu o ne ofere destul si, desi am considerat un Mega, am sfarsit sa prioritizam comoditatea oferita de dimensiunile fizice ale unui Uno oferite unui teoretic utilizator pe langa memoria unui Mega. Astfel ne-am gandit sa incercam reprogramarea placii in timpul utilizarii acesteia folosint un Arduino Nano. Din fericire am gasit, dupa multe cautari, un proiect pentru exact asta din 2012 pe care am reusit sa il adaptam in conceptul nostru.

## Tehnologii folosite:

## Arduino

## Componente:

## Arduino Uno

## Arduino Nano

## Modul SD reader pentru arduino

## Display tactil 2.4’’

## 3 baterii AAA

## 32GB SD card – pentru programe

## 8GB SD card – pentru media

# How it works

Comunicarea intre placi este facuta folosind pinii RX TX pentru serial communication, astfel putem transmite numele fisierului care sa fie incarcat intre placi. Placa programator asteapta pentru mesajul asta, care este trimis o data cu apasarea unui buton din meniu sau butonului de „home” din fiecare program care e menit sa intoarca utilizatorul la meniul principal. Odata receptat semnalul acesta placa v-a lua fisierul .hex de pe card si folosind uploader-ul v-a incarca programul cerut pe placa principala.

Momentan acest proces de incarcare este ince, undeva la 30 de secunde dar speram accelerarea acstuia odata cu adaptarea programului de incarcare strict la nevoile proiectului (uploader-ul hex fiind un proiect mai vechi preluat de pe internet). Insa in ciuda dezavantajului acesta am decis sa ramanem la aceasta abordare datorita flexibitatii oferite de a lucra cu programe care pot fi modificate independent si adaugate usor.

# Produs final

## Dupa toata munca depusa si nenumaratele cunostinte noi adunate pe parcurs, putem zice ca suntem mandrii de produsul final si planuim sa il imbunatatim in viitorul apropiat, adaugand mult mai multe programe si mai ales prezentari vizuale ale acestora, considerand ca acestea ajuta cel mai mult in intelegerea unui algoritm nou.

## Pe partea harware, toate componentele sunt bine prinse impreuna si componentele „removeable” precum cardurile de memorie sunt accesibile.

## Pe partea software, programele sunt stabile, nu credem ca sunt bug-uri majore prezent, daca incep sa apara probleme cu incarcarea anumitor elemente pe display e posibil ca bateriile sa fie descarcate si trebuie inlocuite.

# User Experience

## Interfata este intuitiva, cu butoane care arata clar functionalitatea lor si meniuri simple de navigat.

## La activarea alimentarii, folosind intrerupatorul din laterala, v-a fi incarcat automat programul principal, acesta fiind meniul din care poate fi ales programul pe care utilizatorul doreste sa fie incarcat pe placa.

## Odata incarcat, un program va incepe ori cu prezentarea vizuala a algoritmului, ori, in caz in care aceasta nu a fost inca implementata, direct cu ecranul de informatii despre algoritmul respectiv. Pentru programele cu prezentare vizuala sau interactiva o sa fie si un meniu pentru parametrii de unde pot fi ajustate anumite variabile (Ex. Numarul de linii, si nu numai, la algoritmele de sortare). Astfel proiectul incearca sa explice functionarea algoritmelor in mod vizual si inacelasi timp sa ii ofere un nivel de interactivitate pentru a putea testa aceasta functionare in mai multe cazuri de parametrii initiali.

## Fiind menit sa ofere si niste informatii noi pentru utilizator am rezervat multa atentie la ecranele de informatii, cu informatii din mai multe surse si cu ilustatii realizate in photoshop. Am incercat sa alegem informatii de baza pentru a da o idee buna despre algoritm fara sa incarcam cu prea multe informatii neimportante.

# Why arduino?

# Platforma arduino are avantajul accesibilitatii si a pretului redus, chiar daca timpul de incarcare e putin mai ridicat am preferat aceasta varianta din dorinta de a avea un dispozitiv independent care sa poata sa contina toate programele care dorim sa le contina si de a oferii o experienta mai interactiva decat o simpla aplicate.

# Continut

**Selection Sort –** informatii si vizualizare a procesului de sortare si multipli parametrii ajustabili

**Bubble Sort –** informatii si vizualizare a procesului de sortare si multipli parametrii ajustabili

**Quick Sort –** informatii despre algoritm

**Euclidean Algorithm –** informatii despre algoritm si vizualizare a procesului de determinare a celui mai mare divizor comun

**Knight’s Tour –** informatii despre algoritm

**Raycasting –** informatii despre algoritm si vizualizare interactive a procesului, calculate folosind line-line intersection

**Towers of Hanoi –** informatii despre algoritm

**Prim’s Algorithm –** informatii despre algoritm si vizualizare proces de generare labirint folosind algoritmul

**Dijkstra’s Algorithm –** informatii despre algoritm si vizualizare process de generare a celei mai scurt drum

**A\* Algorithm –** informatii despre algoritm

**Minimax –** informatii despre algoritm

**Line Clipping –** informatii despre algoritm

**Flood Fill–** informatii despre algoritm

## Informatii algoritmi

Fiecare algoritm are informatii de baza care explica functionarea acestuia precum ilustrari, descriere, cod sau pseudocode, time complexity si alte informatii despre algoritmele descrise.

## Echipa

Partea de programare a fost realizata de Hasciar Vlad in timp ce Dunca Ioana s-a ocupat de adunarea informatiilor si realizarea a tot ce tine de media (butoane, ecrane) dar si de partea de testare a programelor. La partea de hardware am lucrat impreuna lipind toate firele si componentele impreuna de mai multe ori si schimband si placa principala la un moment dat din diverse motive, cum ar fi cand am schimbat placa principala datorita unor componente pe aceasta care blocau plasarea bateriilor.

Pentru inceput am pornit de la ideea de a ne testa amandoi cunostintele si in acelasi timp de a invata chestii noi, astfel am incercat sa ne iesim din zona de confort si sa incercam sa abordam solutii cu care nu am fost asa de familiari la incepera proiectului, astfel acesta a evoluat pe parcurs ce am lucrat la el, am planificat diferite aspecte ale lui impreuna si a trebuit sa facem anumite compromisuri in ceea ce considera produsul final. La inceput progresul a fost incet din cauza asta, consistand mai mult din starea unei temelii si aranjarea unui plan de lucru, dar am reusit sa terminam in asa fel incat dupa incepera carantinei am avut fiecare ceva la care sa putem lucra in mod individual.

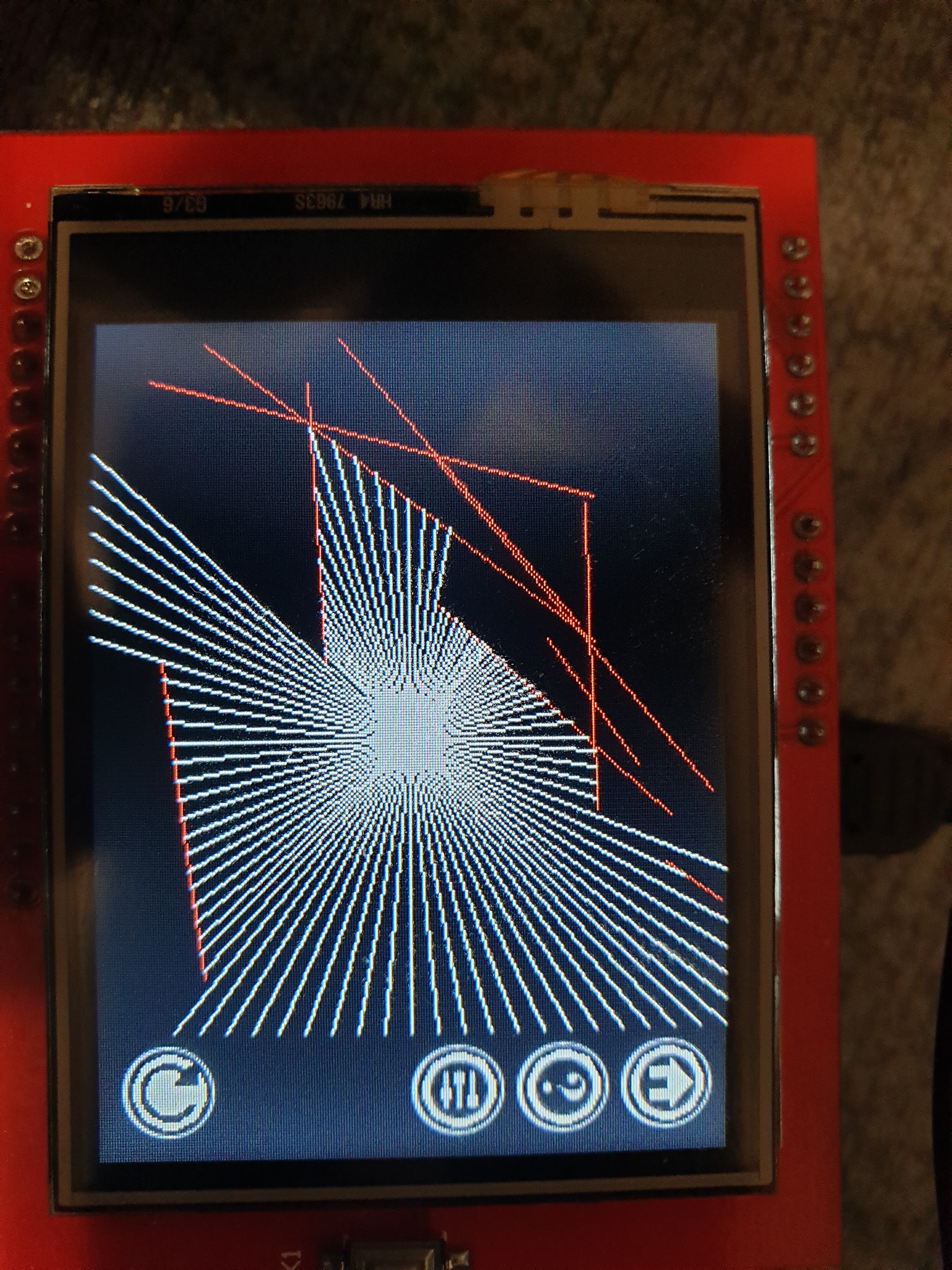
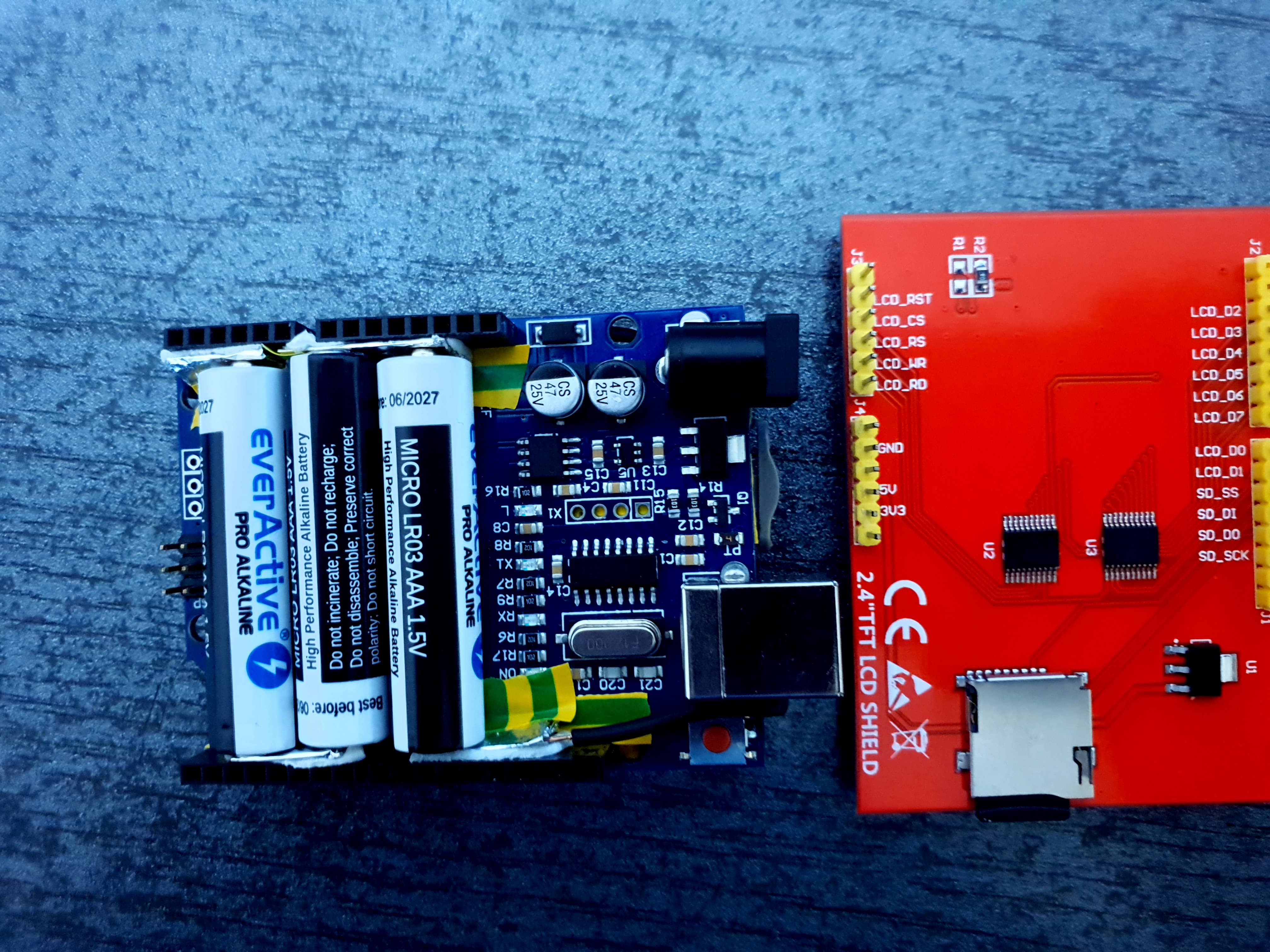
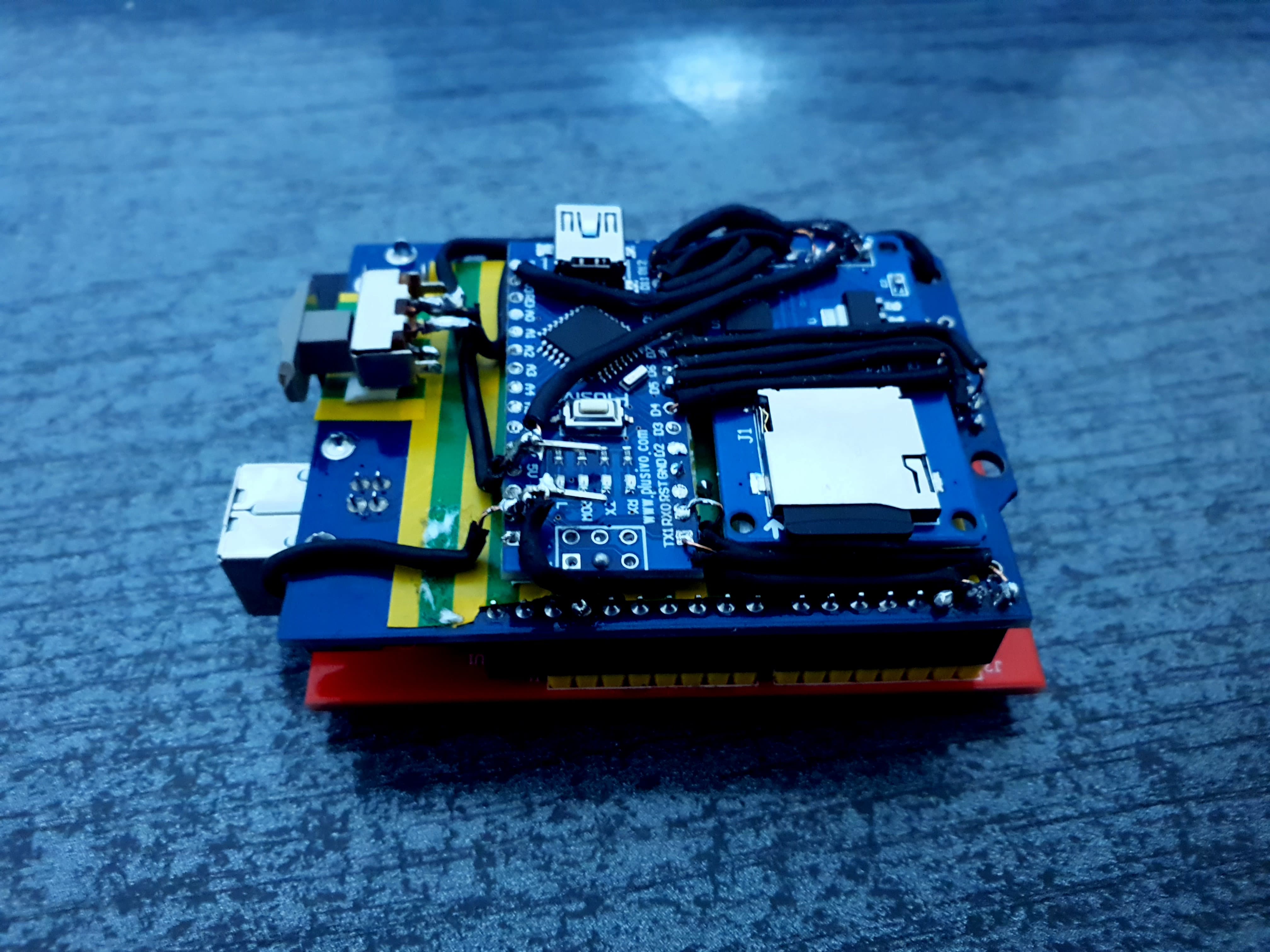
Astfel am lucrat atat impruna cat si in paralel pe parcursul dezvoltarii proiectului si am comunicat constant progresele fiecaruia si discutat planurile si ideile implementabile si desi a trebuit sa luam multe compromisuri in ceea ce consista „features”. Amandoi putem zice ca suntem multumiti de rezultat.

Pentru organizare ne-am folosit de github, trello dar si planificari facute impreuna pe hartie. Pe github se poate gasii tot codul sursa cat si toate celalalte fisiere folosite.

<https://github.com/Vlasciar/InfoEdu>

## Galerie:

## 



## Resurse externe:

## Programare:

## Arduino .hex uploader <http://www.gammon.com.au/uploader>

## Informatii:

## <https://www.geeksforgeeks.org>

## <https://en.wikipedia.org/>

## <https://www.educative.io/>

## [www.edureka.co](http://www.edureka.co)

## [www.programiz.com](http://www.programiz.com)

## rosettacode.org

## [www.techopedia.com](http://www.techopedia.com)

## docs.unity3d.com

## 