

# Pian electric

Proiect realizat de Radu Andrei - 1220FB

## Introducere

Proiectul meu consta in crearea un pian electric simplu la care poti invata sa canti cateva melodii. Pianul are sapte game muzicale, trecerea de la o gama la alta putand fi facuta cu ajutorul celor doua butoane din stanga. De asemenea, acesta poate reda cateva melodii apasand cate doua butoane concomitent.

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ați pornit
- de ce credeți că este util pentru alții și pentru voi

## Descriere generală

Pianul are sapte butoane pentru notele muzicale: DO, RE, MI, FA, SOL, LA, SI. Alte doua butoane (primele din stanga) ajuta la trecerea de la o gama la alta: primul ajuta la accesul gamelor din ce in ce mai inalte, al doilea la accesul gamelor din ce in ce mai joase. Pianul poate reda cateva melodii cu ajutorul a doua butoane. Prima melodie poate fi redata apasand butonul de schimbare a gamei spre game mai inalte, impreuna cu butonul notei DO. A doua poate fi redata cu acelasi prim buton, iar al doilea fiind butonul notei RE. A treia melodie functioneaza pe acelasi principiu, doar ca al doilea buton este cel al notei MI.

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>


## Hardware Design



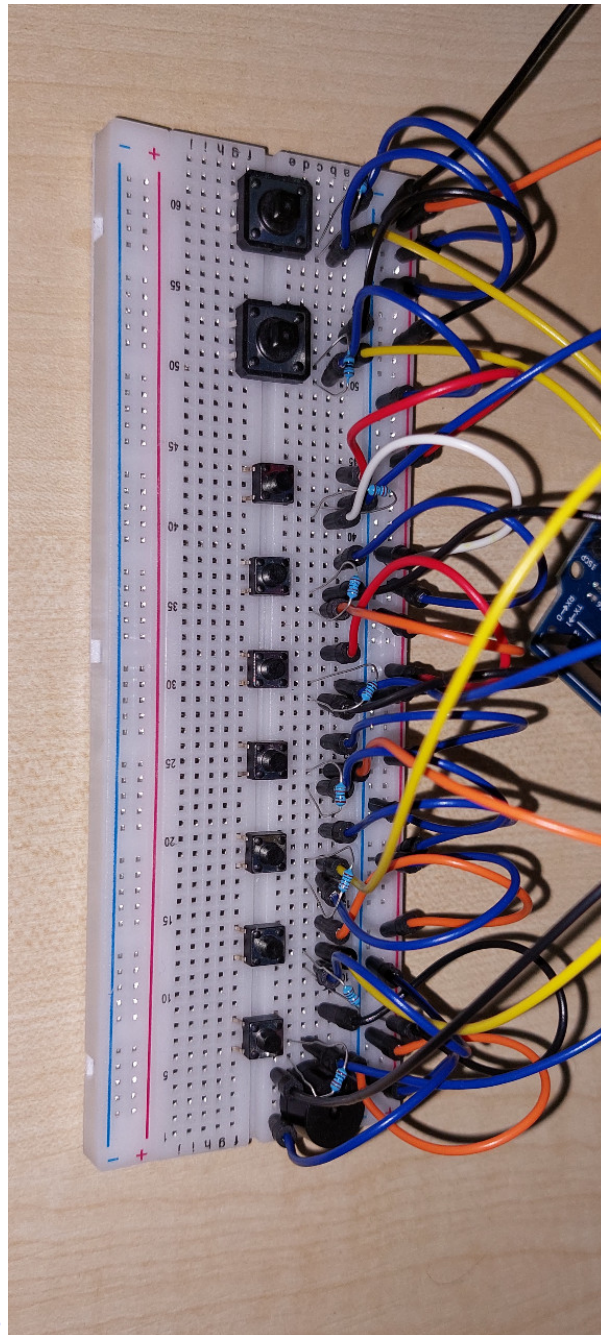
Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

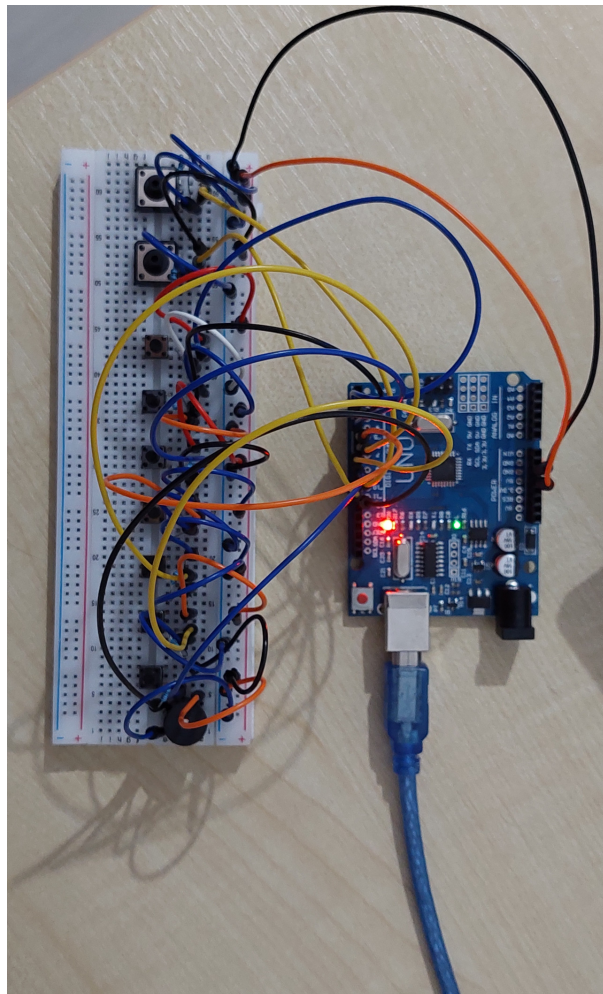
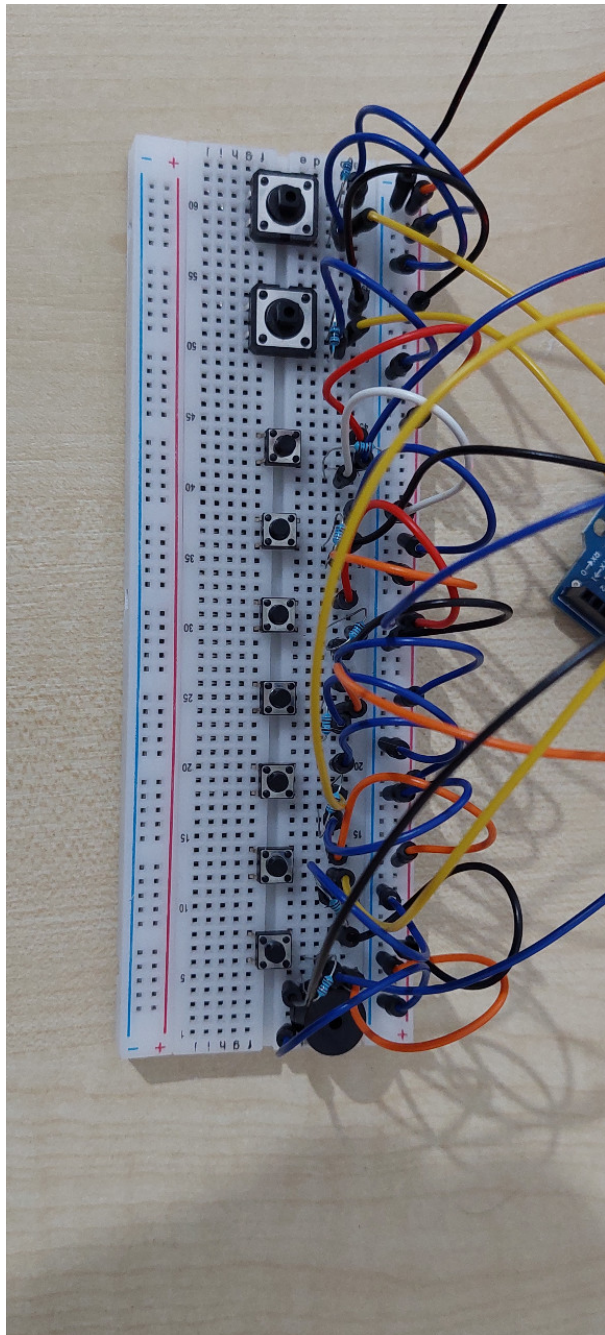
## Software Design

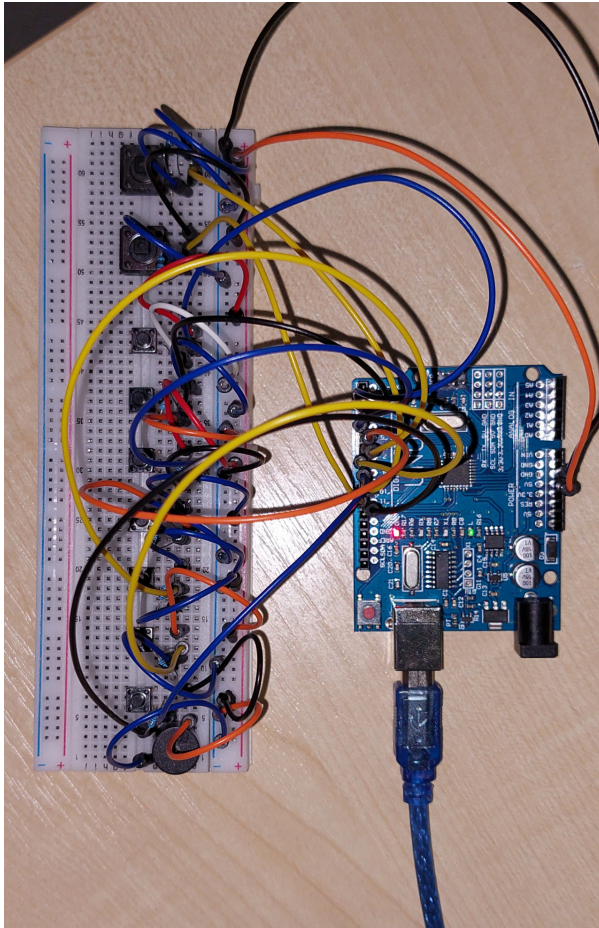
 **Mediu de dezvoltare:** Arduino IDE. **Componente:**

1. Arduino Uno R3
2. 9xPushbutton
3. 9x1 k $\Omega$  Resistor
4. 1xBuzzer



[andrei\\_radu-pm\\_2022.rar](#)





Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

==== Rezultate Obținute ====

Care au fost rezultatele obținute în urma realizării proiectului vostru.

==== Concluzii ====

==== Download ====

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

===== Jurnal =====

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

===== Bibliografie/Resurse =====

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/avaduva/profesoruldepian>



Last update: **2022/06/02 11:20**