

# Electronic Keyboard

## Introducere

Proiectul presupune realizarea unui mini pian digital care poate schimba între mai multe tipuri de sunete: pian de concert, flaut, chitară, vioară. De asemenea, pianul trebuie să fie polifonic (permite redarea mai multor note în același timp).

Am vrut să creez un proiect pe o temă care mă interesează în mod deosebit, văzându-l ca o șansă de a aprofunda un subiect pe care nu l-aș fi abordat în mod normal. Îmi place să cânt la pian, dar nu am căutat până acum și nu m-am gândit la ce piese sunt în interior și cum duc la producerea sunetului.

Proiectul realizat în sine nu cred că este foarte util, fiind mult mai limitat în numărul de note și moduri în care poți să cânti decât un pian online care poate fi accesat cu foarte mare ușurință, dar poate fi o ocazie de descoperire a componentelor hardware și software care duc la redarea sunetelor unui pian.

## Descriere generală

### Schema bloc



Modulul **Switch** are un buton care controlează modul de sunet al pianului. Prin apăsarea lui, se ciclează între tipurile diferite de instrumente/sunete.

Modulul **Keyboard** cuprinde 8 butoane pentru cele 8 tonuri naturale, de bază ale pianului. Apăsarea lor va determina difuzorul să emită o notă muzicală.

Modulul **SD Card** presupune prezența unui SD Card Holder și al unui SD Card pe care sunt încărcate *sample-uri* de la diferite instrumente (note izolate) care vor fi cântate la apăsarea "clapei" corespunzătoare.

Modulul **Speaker** se referă la difuzorul prin intermediul căruia se vor emite sunetele la apăsarea butoanelor.

## Hardware Design

## Lista de piese

- Placa de bază - ATmega328



- 8 Butoane (switch + keyboard)



- Difuzor

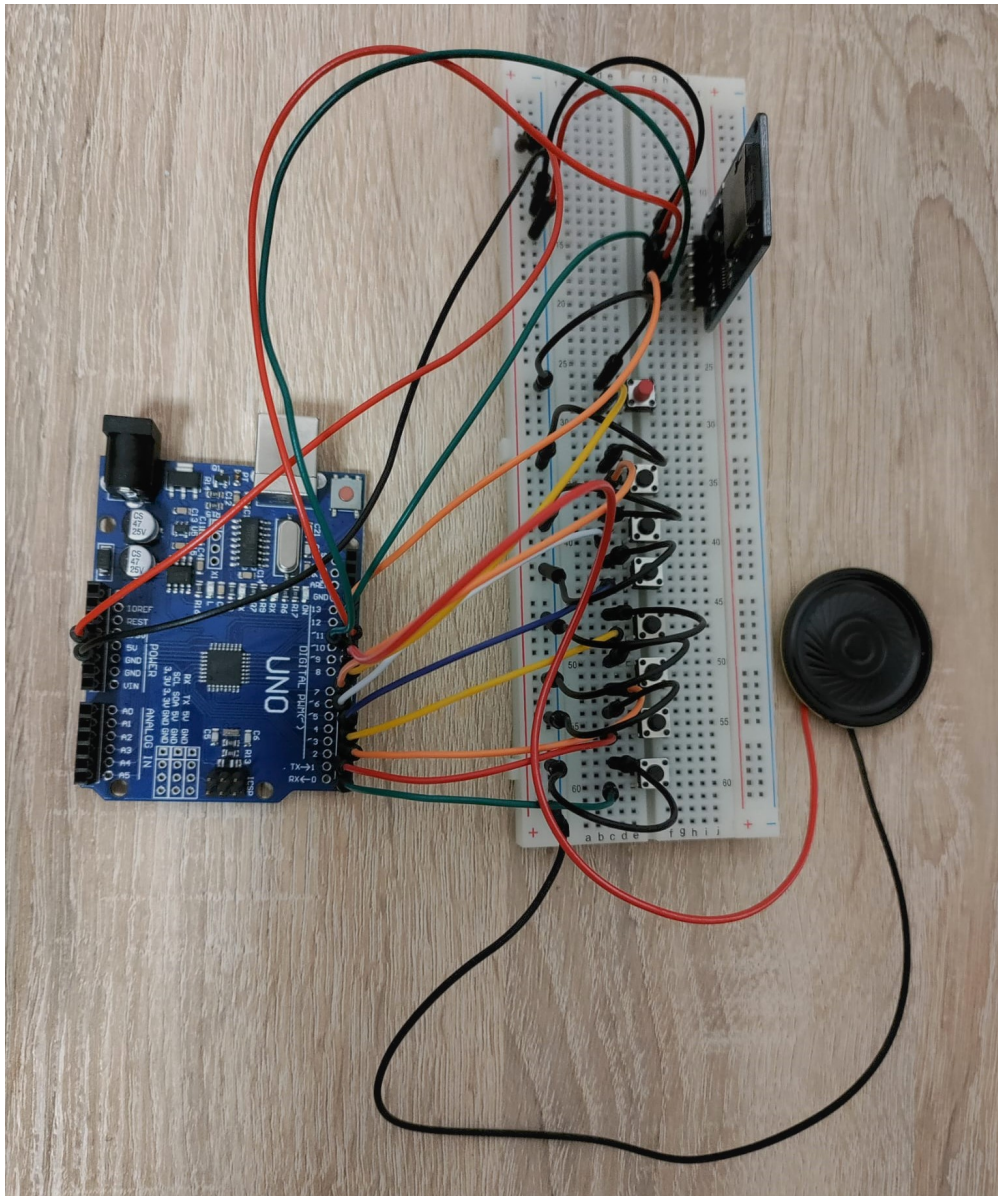


- SD Card Holder + SD Card



## Schema electrică





## Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare: Arduino IDE
- bibliotecă 3-rd party: TMRpcm pentru redarea fișierelor WAV din cardul SD (folosește și biblioteca standard Arduino SD)

Implementarea constă în principal din codul pentru diferite întreruperi (butonul de switch, “clapele” și timer-ul pentru difuzor).

Întreruperea pentru **butonul de switch** presupune ca la apăsarea acestuia să se modifice starea lui, ciclând între 3 moduri: pian, flaut, vioară.

```
ISR(INT0_vect) {  
    // Increment switchState and wrap around if necessary  
    if (millis() - lastDebounceTime >= debounceDelay) {  
        if ((PIND & (1 << Switch)) == 0) {
```

```
    lastDebounceTime = millis();
    switchState = (switchState + 1) % WAV_TYPE_COUNT;
    Serial.println(switchState);
  }
}
```

Pentru **clape**, la apăsarea lor se dă play la audio-ul corespondent butonului și activează variabila *soundEnabled* care este folosită pentru redarea sunetului de către difuzor.

```
ISR(PCINT2_vect) {
  for (int i = 0; i < sizeof(buttonPins) / sizeof(buttonPins[0]); i++) {
    if ((PIND & (1 << buttonPins[i])) == 0) {
      // Button is pressed, play the corresponding WAV file
      if (switchState == 0) {
        playWavFile(pianoFiles[i]);
      } else if (switchState == 1) {
        playWavFile(fluteFiles[i]);
      } else if (switchState == 2) {
        playWavFile(violinFiles[i]);
      }
      soundEnabled = true;
      return;
    }
  }
  soundEnabled = false;
}
```

Timer-ul **difuzorului** este de tipul Fast PWM și generează sunet atunci când primește semnal de la clape că una dintre ele este apăsată.

```
ISR(TIMER1_COMPA_vect) {
  if (soundEnabled) {
    // Generate sound
    PORTB |= (1 << PB1);
  } else {
    // Stop sound
    PORTB &= ~(1 << PB1);
  }
}
```

## Rezultate Obținute

Cum TMRpcm nu are suport pentru redarea sunetului concurent și, în general, Arduino Uno este limitat în această privință din cauza memoriei necesare stocării mai multor fișiere audio în același timp și complexității mai mari a programului, se poate reda o singură notă în același timp.

Dacă nu aş fi folosit fişiere audio, problema ar fi fost rezolvată. În loc ca fiecare buton să genereze un sunet la o anumită frecvenţă, aş fi adunat frecvenţele tuturor pentru a compune o frecvenţă finală pe care o reda difuzorul.

O altă problemă întâmpinată a fost faptul că anumite întreruperi interferau unele cu altele. Astfel, în loc să folosesc PCINT0 pentru întreruperea specifică butonului de Switch, din cauză că nu funcţiona corect împreună cu TIMER1, folosit la difuzor, am modificat la o întrerupere externă, INTO.

Rezultatele sunt cele aşteptate: la apăsarea butoanelor se redă sunetul corespunzător şi ciclează printre fişierele audio prin folosirea butonului de Switch. Totuşi, având în vedere difuzorul folosit, sunetele sunt neclare. Acesta este un lucru pe care l-aş schimba cu siguranţă pentru a îmbunătăţi proiectul.

## Concluzii

Proiectul demonstrează utilizarea modulului de card SD pentru a stoca si a prelua fişiere audio, biblioteca TMRpcm pentru redarea audio şi utilizarea întreruperilor pentru a gestiona evenimentele de intrare.

Poate fi folosit în diverse aplicaţii în care este necesară redarea audio, cum ar fi playere muzicale, sisteme de efecte sonore, instalaţii interactive şi proiecte educaţionale.

Realizând acest proiect am învăţat multe lucruri pe care nu le cunoşteam înainte şi am întâmpinat probleme atât hardware, cât şi software pe care nu le anticipasem. M-a făcut să am mai multă apreciere şi respect pentru persoanele care lucrează cu astfel de module.

## Download

Arhiva conţine codul sursă şi sample-urile de note.

[piano.zip](#)

## Jurnal

- 7.05.2023 - Introducere + Descriere generală
- 21.05.2023 - Hardware
- 28.05.2023 - Software

## Bibliografie/Resurse

Resurse Hardware:

- Set-up inițial al unui mini-pian:  
<https://projecthub.arduino.cc/rahulkhanna/arduino-tutorial-mini-piano-0c7ec5>
- Laboratoare de PM (în special SPI pentru conectarea corectă a modulului SD)

#### Resurse Software:

- Sample de note pentru vioară:  
<https://depositonce.tu-berlin.de/items/f81ba73c-4d9b-48de-9fbc-31cb03d5b9bc>
- Sample de note pentru flaut: <https://freesound.org/people/MTG/packs/20209/>
- Sample de note pentru pian: <https://pixabay.com/sound-effects/search/piano-notes/>
- Biblioteca TMRpcm: <https://www.arduino-libraries.info/libraries/tm-rpcm>
- Laboratoare de PM
- Forumuri Arduino
- Multe conversații cu ChatGPT

[Export to PDF](#)

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

Permanent link:  
[http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/electronic\\_keyboard](http://ocw.cs.pub.ro/courses/pm/prj2023/apredescu/electronic_keyboard)



Last update: **2023/05/30 06:08**