


# Floppy Orchestra

Contact: [George Mocanu](#)

## Introducere

---

Acest proiect urmărește realizarea unei orchestre muzicale formată din Floppy Driver-e, capabilă să redea un fișier muzical primit ca input (sub forma unui MIDI file).

*Dar cum pot Floppy Driver-e să cânte???* Doar sunt niște dispozitive electronice și nu au voce . Ei bine, fiecare floppy drive are în interiorul său un motor stepper ce poate fi folosit la redarea unor sunete, controlând viteza și sensul mișcării al acestuia la un moment dat.

### How a does stepper motor create music? <sup>1)</sup>

When step motor moves, it creates the vibration with a specific frequency; the vibration makes the surrounding environment oscillate. The oscillation creates a mechanical wave. Frequency value is depended on the speed. When frequency value is between about 20 Hz and 20 kHz, it becomes sound wave and our ear can hear it.

By adjusting the speed, we can create the different frequency of sound. Therefore, we can create a different note of music.

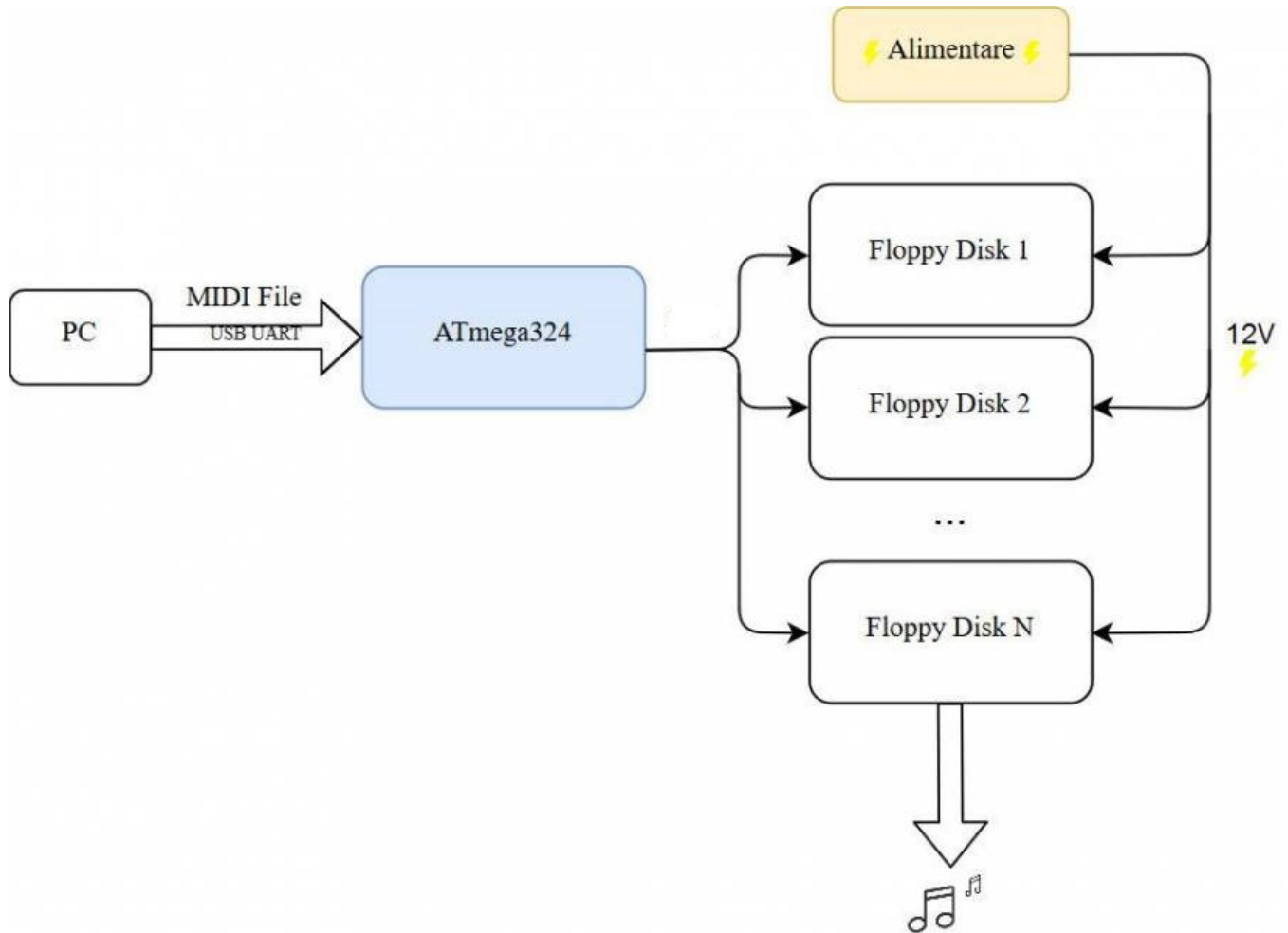
By mapping music notes of a song to speed, moving step motor with the speed values in sequence, step motor can create melody of that song. We can use more than one motor to play a complex song.

Scopul proiectului este punerea la încercare a noțiunilor acumulate (sper eu) până acum, dar și dobândirea unor alte cunoștințe din domeniul muzical.

## Descriere generală

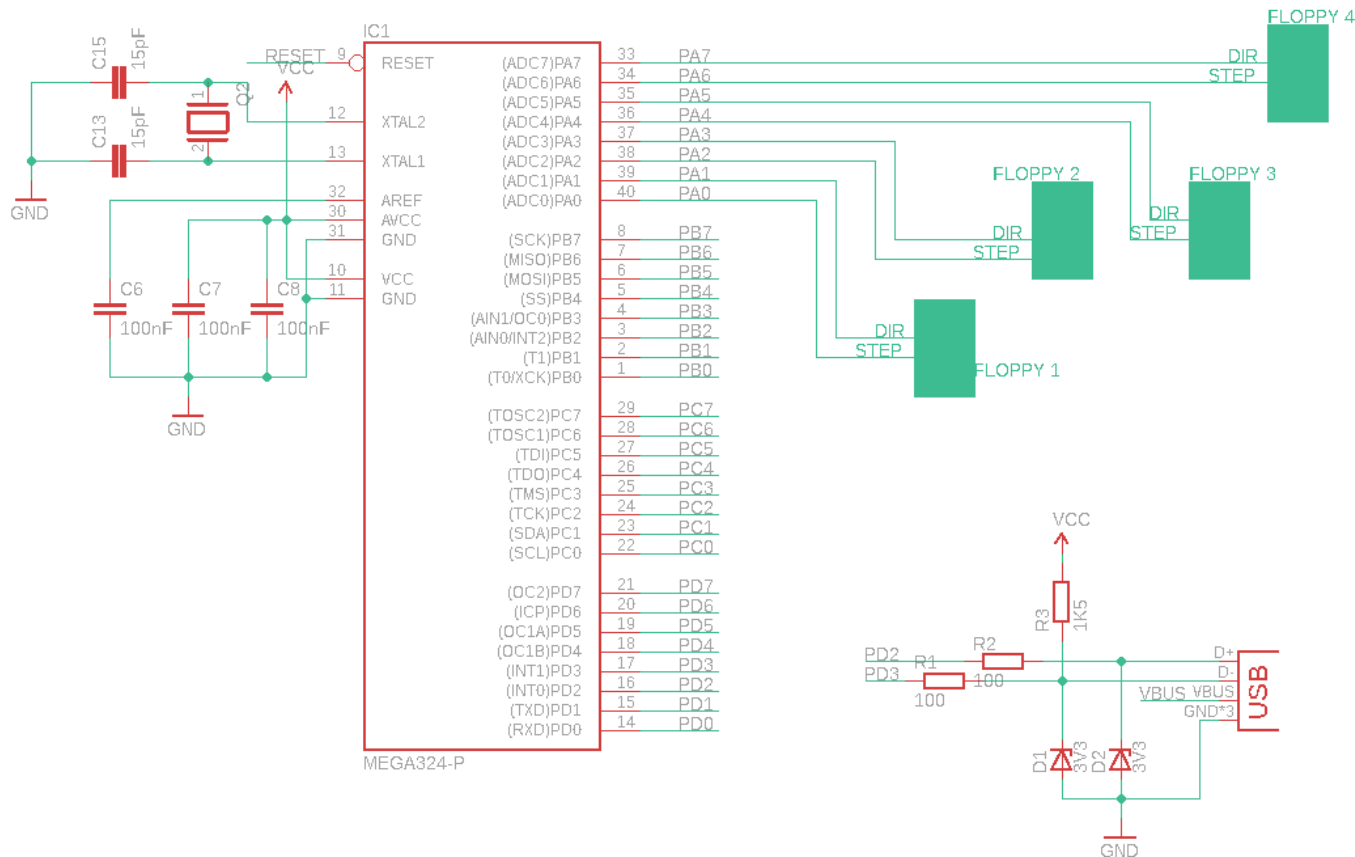
---

### Schema bloc:



Microcontroller-ul ATmega324 de pe plăcuța de bază va controla Floppy Driver-ele (mai exact motorul stepper din interiorul fiecăruia), pentru a recrea o anumită notă muzicală, conform fișierului MIDI ce se dorește a fi redat.


### Schema electrică:



## Hardware Design

### Lista de piese & materiale:

Piesă	Scop	Modalitate de obținere
Placa de bază PM 2019	...	Lipită în laborator, după posibilități
6 x floppy drive (doar 4 functionale)	Dânsele or să cânte	2 din calculatoare vechi, 4 cumparate
Sursă ATX	Folosită la alimentarea floppy-urilor	Furată dintr-un PC vechi
Breadboard	Pentru a pune la comun gnd-ul placii cu cel al floppy-urilor	Imprumutata de la un prieten
Convertor USB-USART	Trimiterea notelor pe seriala	Cumparat de la optimusdigital
Multe fire mama-mama si mama-tata	Conexiuni placa - floppy-uri	Achizitie proprie + imprumut

Deocamdată dețin 2 floppy drivere, urmând să fac rost și de altele, în funcție de posibilitățile hardware ale plăcuței, dar și a flerului meu .

Lista nu este încă completa. O voi completa pe parcurs în funcție de necesități.

## Software Design

---

Implementarea s-a realizat pe Fedora 28 (orice distributie de Linux ar trebui sa fie capabila sa compileze codul sursa), folosind compilatorul avr-gcc. Intai s-a implementat logica miscarii capurilor de citire pentru generarea sunetelor - acestea se vor misca la o perioada de timp stabilita pentru a genera o anumita frecventa (s-a folosit un timer de 40us pentru a "masura" un tick  $\leftrightarrow$  la inregistrarea unui numar de tick-uri pentru un anume floppy, acesta va face un step in directia potrivita si va genera nota dorita), urmand si partea cu transmiterea fisierelor peste seriala. Fiindca informatiile primite pot fii multe, am ales sa folosesc o biblioteca deja implementata pentru comunicarea peste UART<sup>2)</sup>, ce foloseste un buffer circular (in arhiva se regaseste si o versiune "basic" pe care am folosit-o la inceput, dar nu a dat randamentul dorit). Transmiterea informatiilor peste seriala de la calculator este destul de complicata, astfel ca am ales sa folosesc un tool deja existent ce citeste fisierele .midi si trimite informatiile necesare (a trebuit doar sa fac ca interpretarea mesajelor primite pe UART sa fie compatibila). Pentru testare s-a utilizat programul amintit <sup>3)</sup>, iar rularea acestuia s-a facut pe Windows, folosind NetBeans, si incarcand diferite fisiere .midi<sup>4)</sup> (cu numarul de canale cat mai mic, nu am decat 4 floppy-uri).

## Rezultate

---



Din pacate am reusit sa utilizez doar 4 floppy-uri, avand 2 ce par sa fie defecte (una este reparabila, fiind necesare doar cateva lipituri).

Un video deomonstrativ: <https://www.youtube.com/watch?v=iA3PDe6U8fA>

## Concluzii

---

Per total, a fost un proiect interesant, am vazut ce inseamna sa creezi un produs cap-coada, nu doar din punct de vedere Software, ci si Hardware. A fost o adevarata provocare dar si satisfactia este pe masura rezultatului.

## Download

---

[floppy\\_orchestra\\_src.zip](#)

## Bibliografie/Resurse

---

Inspirație: [https://www.youtube.com/watch?v=KcIRiK-L-\\_4](https://www.youtube.com/watch?v=KcIRiK-L-_4)

<https://github.com/SammyIAm/MoppyClassic>

<https://github.com/andygoock/avr-uart>

Documentația în format PDF

1)

<https://www.hackster.io/45036/making-music-using-a-step-motor-01f8e6#toc-how-a-does-stepper-motor-create-music-1>

2) <https://github.com/andygoock/avr-uart>

3) <https://github.com/SammyIAm/MoppyClassic/tree/moppy-advanced/Java>

4) <https://github.com/coon42/Floppy-Music--midis->

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2019/mlungoci/floppy\\_orchestra](http://ocw.cs.pub.ro/courses/pm/prj2019/mlungoci/floppy_orchestra)



Last update: **2021/04/14 15:07**