

Denis-Adelin FIRU (78323) - Pian

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Proiectul presupune realizarea unui pian cu mai multe functionalitati:

- posibilitatea schimbarii sunetelor, astfel incat pianul sa poata reda si alte instrumente
- schimbarea volumului
- schimbarea octavei
- afisarea pe un ecran lcd a configuratiei actuale (volum + instrument + octava)

Scopul acestui proiect este de a invata sa canti la pian, dar in acelasi timp sa ai posibilitatea sa schimbi sunetul de pian cu un alt instrument. Acest lucru ar ajuta, deoarece nu e nevoie doar sa stii sa canti la pian pentru a reusi sa canti si la celelalte instrumente.

Ideea mi-a venit din dorinta de a vrea sa stiu sa cant la un instrument.

Descriere generală



Ecranul lcd va prelua date din program, in legatura cu volumul, octava si instrumentul. Acesta se va actualiza in momentul in care se apasa pe butoanele de schimbare a uneia din cele 3 informatii afisate.

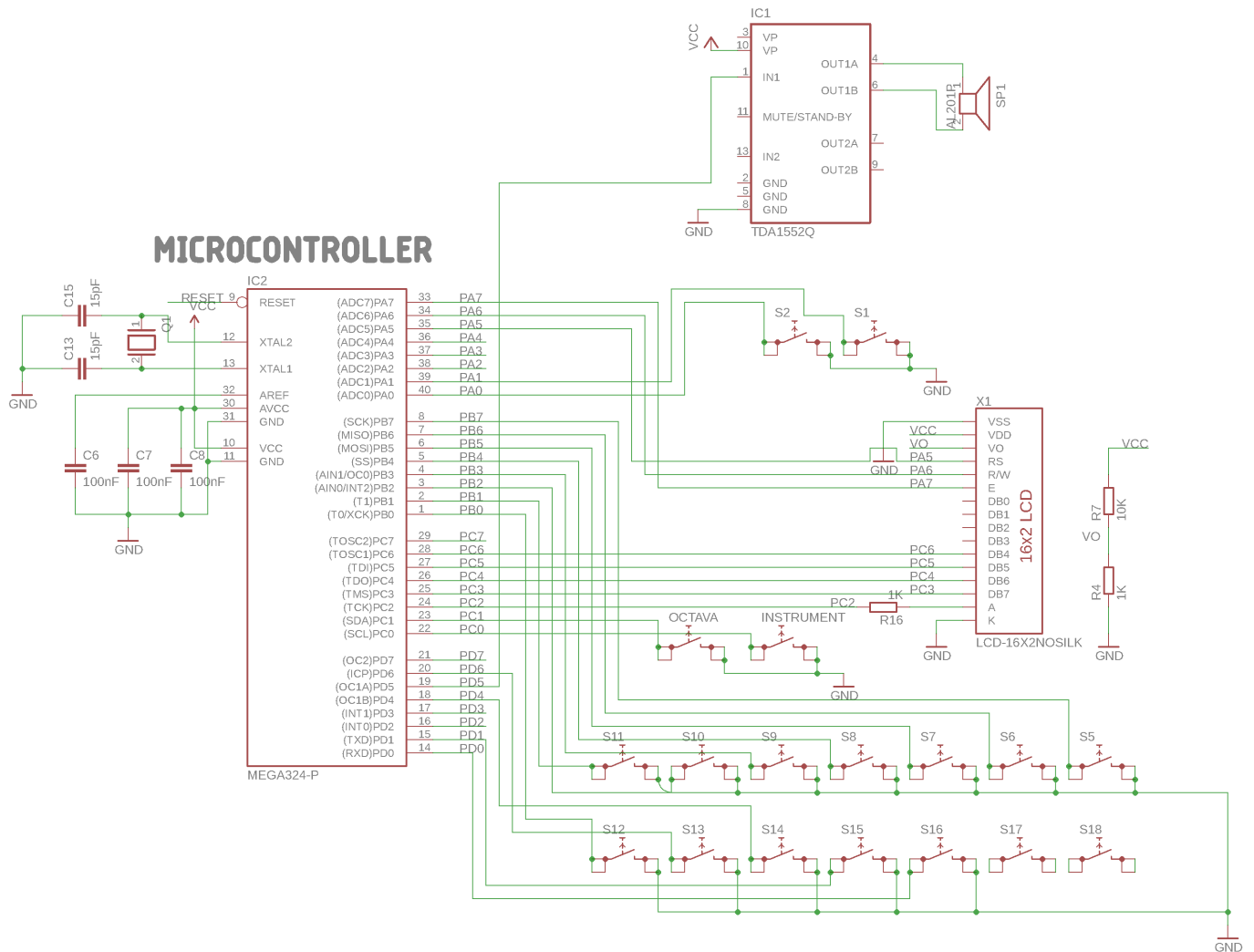
La apasarea unei clape a claviaturii, microcontroller-ul va trimite nota mapata pe clapa respectiva catre difuzor, pentru a reda sunetul.

Hardware Design

Lista de piese:

- PCB 2018
- Kit de baza
- Butoane
- LCD
- Claviatura

- Fire de legatura
- Rezistoare
- Condensatori
- Difuzor
- Amplificator audio GF1002 + potentiometru



Pentru afisaj folosesc un LCD cu un controller Hitachi 44780.

Difuzorul este de 8 ohmi.

Pentru butoanele de control, cat si cele ale claviaturii nu a mai trebuit sa pun rezistente, deoarece microcontroller-ul are rezistente de pull-up integrate.

Software Design

Ca mediu de dezvoltare am folosit Clion ca IDE, iar hex-ul l-am generat cu un Makefile. Pentru librarii am folosit:

- <util/delay.h> - pentru functia _delay_ms()
- <avr/interrupt.h> pentru intreruperi

Pentru functionarea pianului am folosit 2 timere:

Timer 0 verifica la fiecare 10 ms printr-o intrerupere, daca s-a apasat un buton de pe claviatura

Timer 1 il folosesc pentru a genera sunet prin pwm la speaker

Funcții implementate sunt câteva funcții ajutatoare:

- `init_timer` - funcție care îmi initializează cele 2 timere:
- `push` - verifică dacă s-a apăsător un buton
- `pwm_freq` - setează ocr pentru timer1 pentru a putea genera sunete
- `pwm_on` - porneste pwm
- `pwm_off` - opreste pwm
- `check` - această funcție verifică ce butoane s-au apasat și în funcție de asta ia frecvența și face media în caz că s-au apasat mai multe

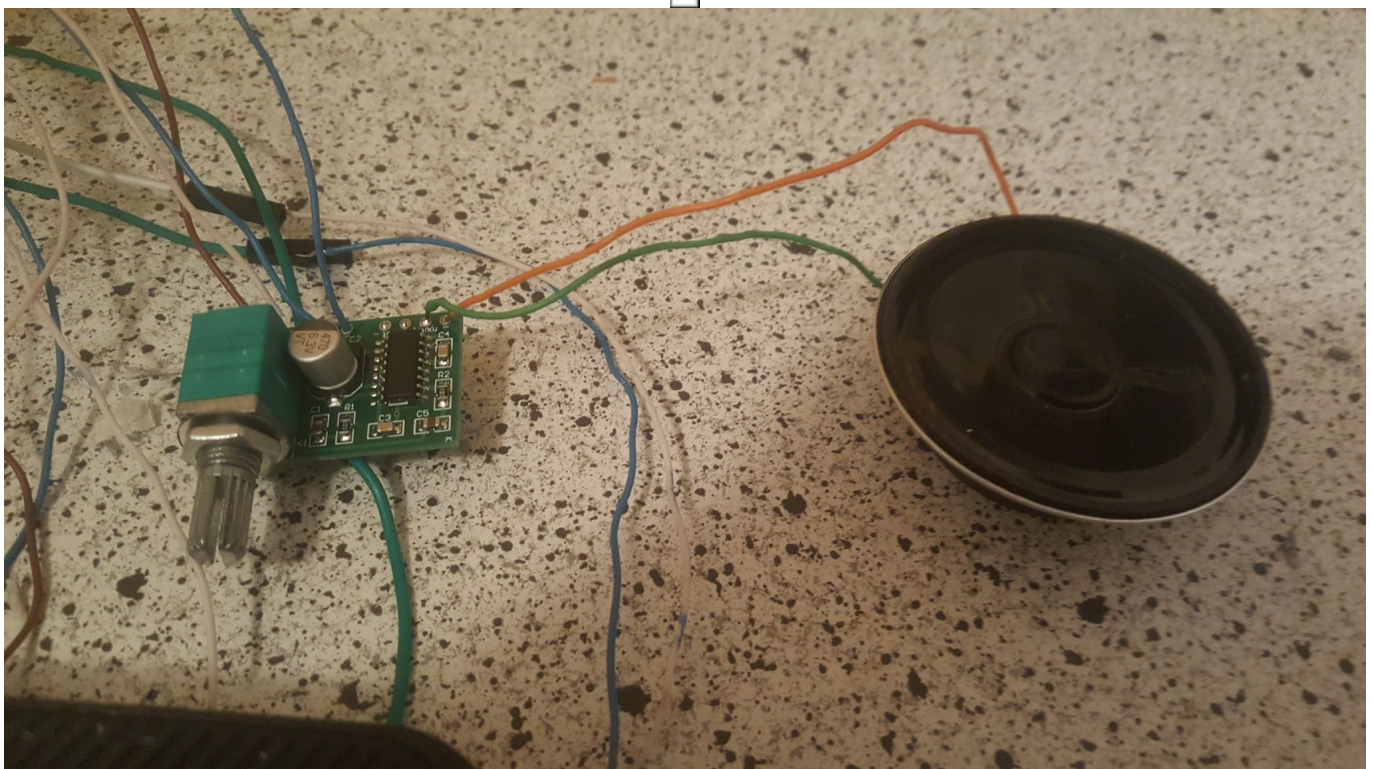
Timer 0 , mod CTC , prescaler 1024 cu intreruperile activate Timer 1 , pwm activat, cu prescaler 1024

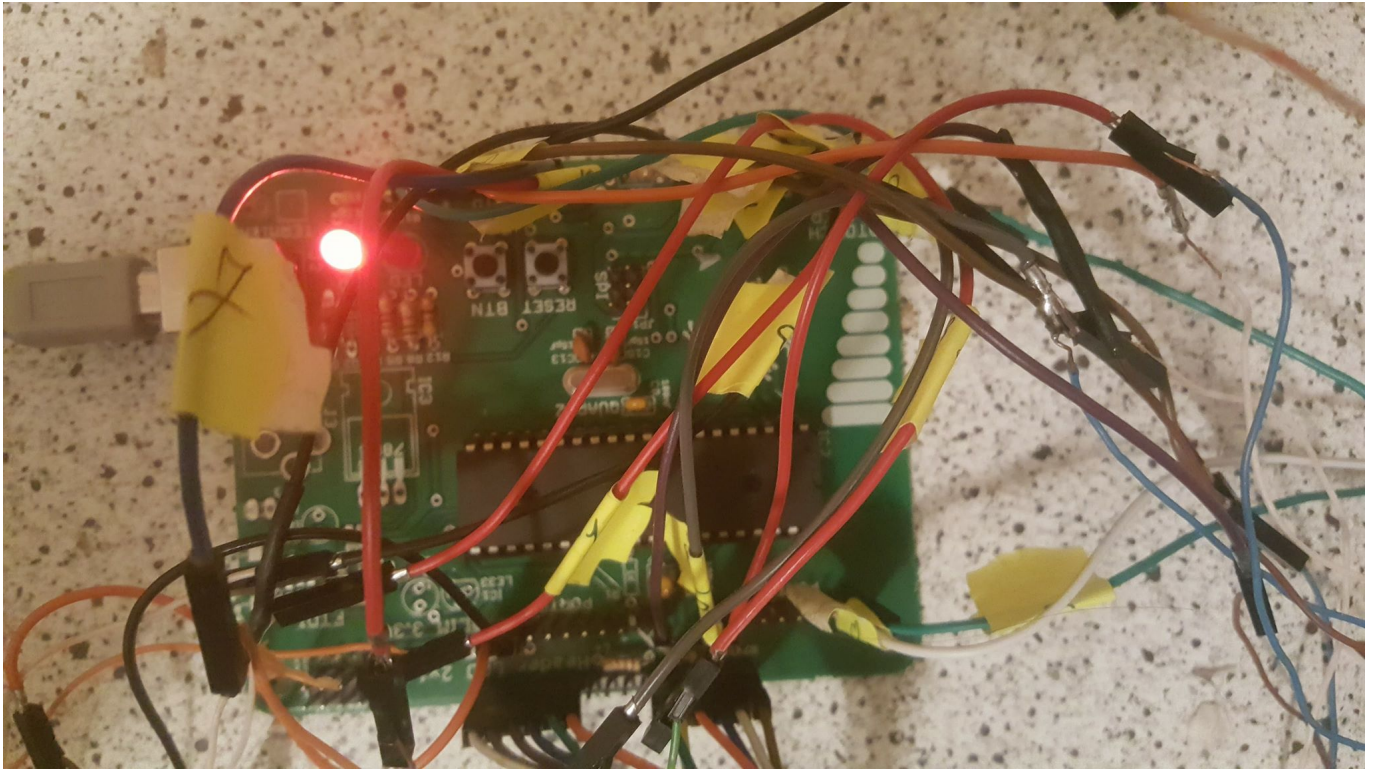
Rezultate Obținute

În momentul de față pianul are implementate:

- schimbarea de octavă
- modificarea volumului

Încă nu am reușit să modific sunetul pe care îl scoate, pentru a simula mai multe instrumente





Concluzii

A fost un proiect interesant , in care am invatat sa lucrez mai bine si sa inteleg timerele si pwm.

De asemenea, am descoperit ca imi place sa programez placute si sa vad cum obtin anumite rezultate

Download

Arhiva cu codul sursa : [denis_adelin.firu_pian.zip](#)

Eagle: [denis_adelin.firu_sch.sch](#)

Jurnal

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

Bibliografie/Resurse

Laboratoare PM

Youtube

Datasheet ATmega324 [doc8272.pdf](#)

- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/avoinescu/pian>



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