

Sergiu WEISZ (67387) - Cheap Guitar Effect

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

Introducere

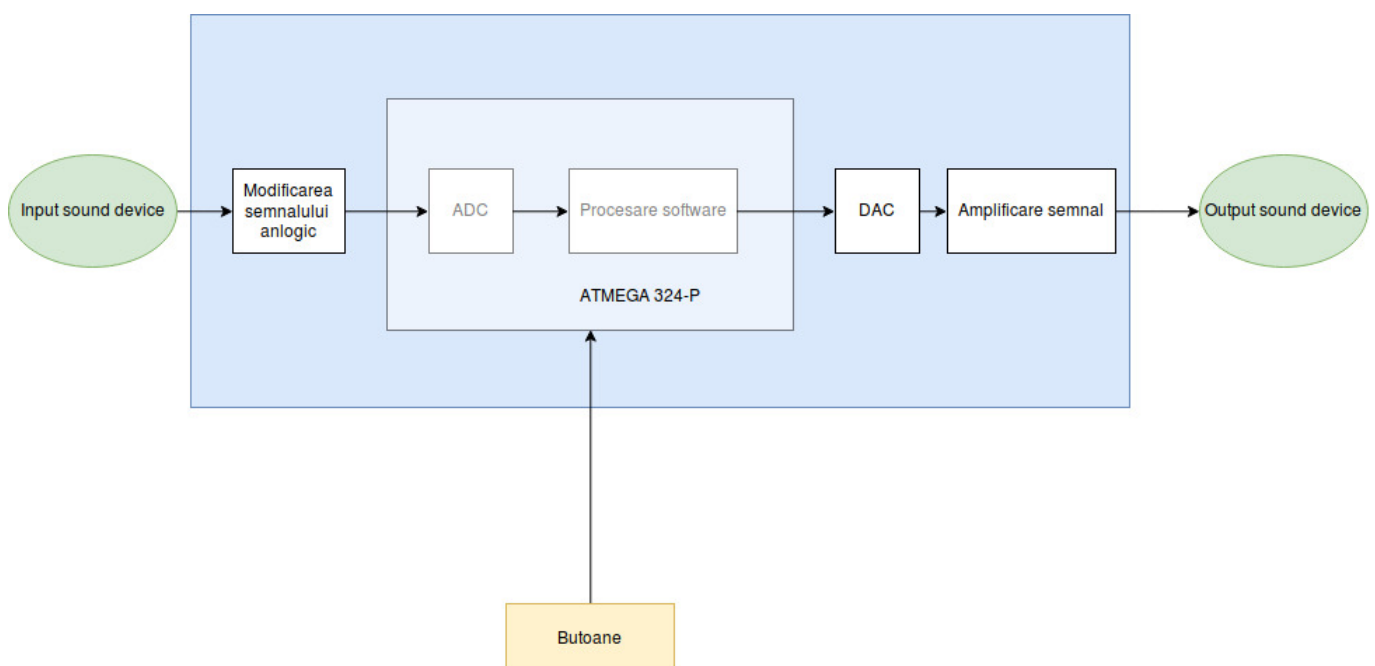
Ce face ?

Este un efect de chitara penit sa manipuleze semnalul primit la input de la chitara (sau orice alt dispozitiv care bagat) si il scoate amplificat la output.

Care este scopul lui si de ce este folositor pentru mine ?

Deoarece efectele de chitara electrica bass sunt scumpe, vreau sa imi fac eu unul, care sa poata fi reprogramat usor de catre utilizator.

Descriere generală

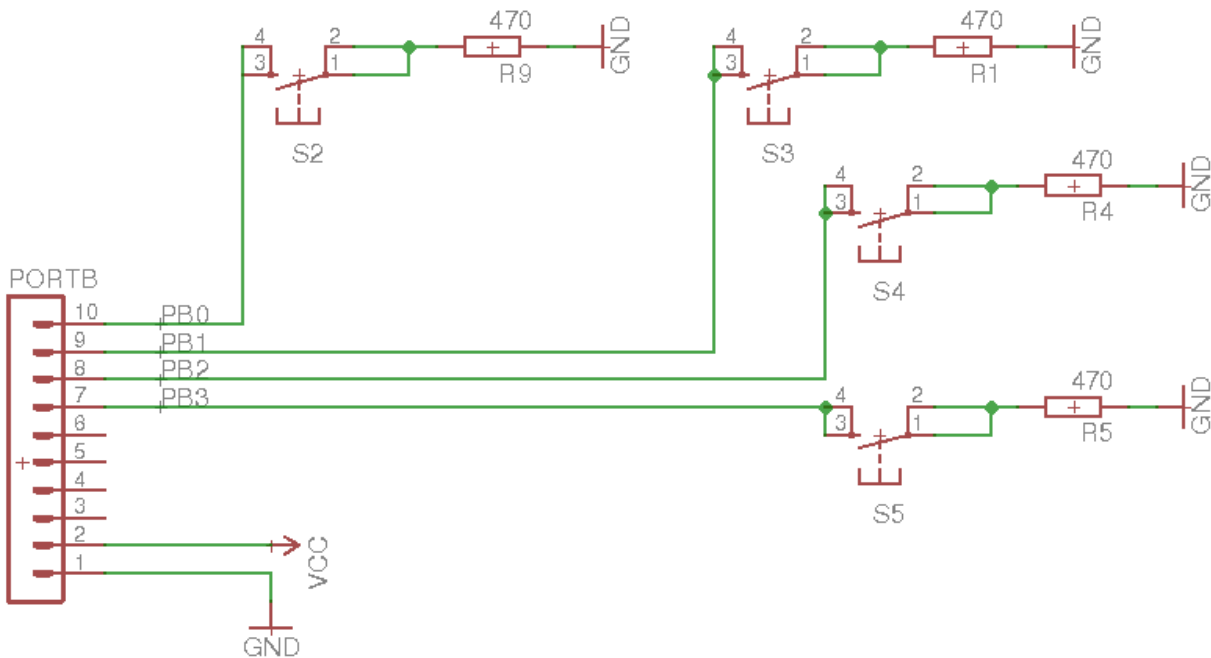


Semnalul o sa fie primit de la un input device, o sa fie filtrat astfel in cat sa fie primit corect la input de catre procesor pe PORTA. Pe port-urile respective se afla input-ul DAC-ului. Semnalul este procesat folosind un algoritm. Initial vreau sa fac numai un efect de distorsionare a semnalului, dar in functie de rezultatele obtinute, as vrea sa incorporez si un efect de ecou si unul de intarziere. Dupa ce semnalul a fost procesat acesta va fi trimis la output si va fi convertit din semnal digital in semnal analogic si va fi amplificat si trimis la iesire. Pentru a modifica efectul, voi folosi mai multe butoane.

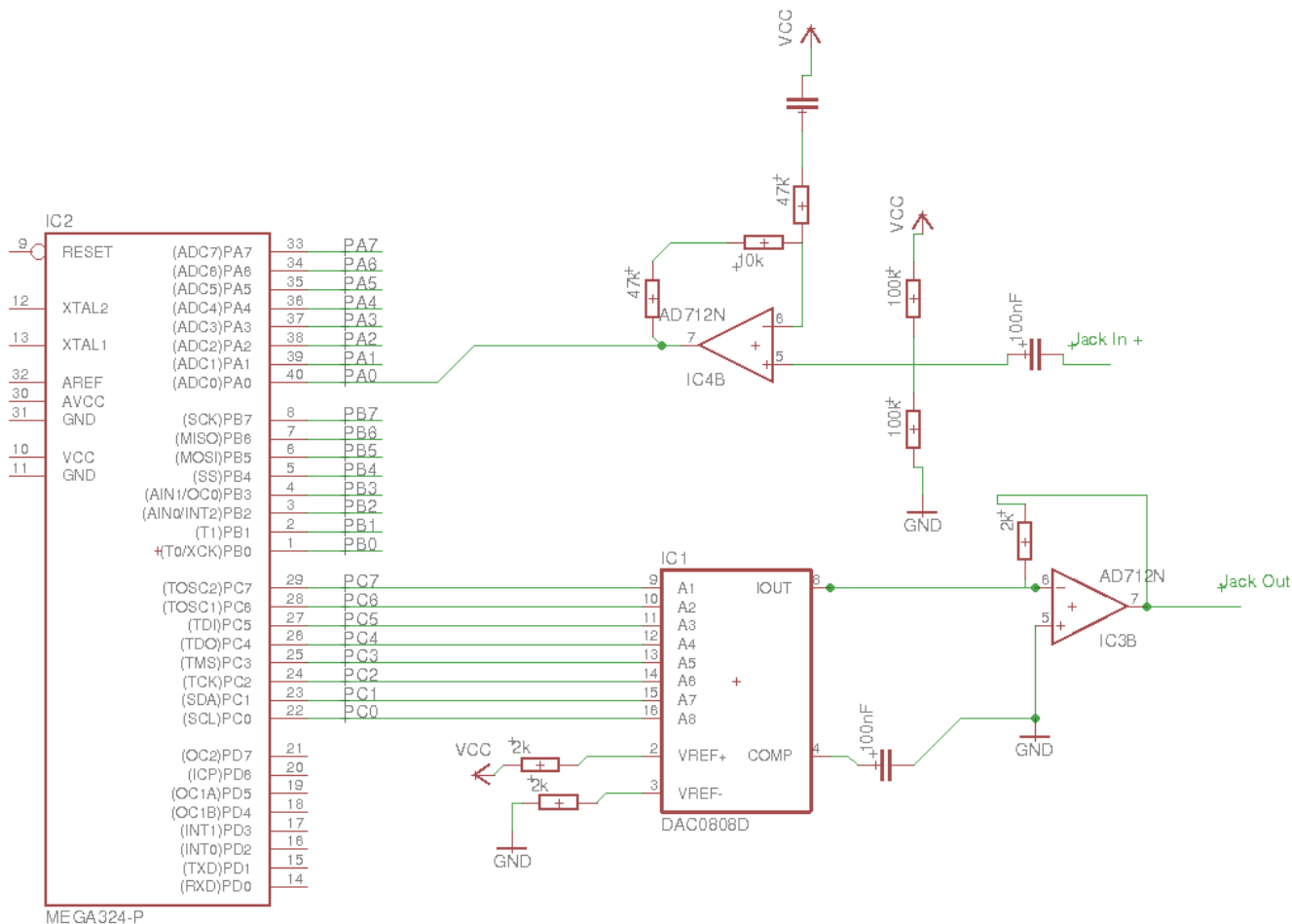
Hardware Design

- Conector jack 6.25mm
- Convertor digital analogic 8080
- 3 butoane
- 2 amplificatoare operationale TL074CNE4
- 2 rezistente 47K
- 2 condensatoare 100nF
- 2 rezistoare 2K (nu am avut asa ca am folosit 4 de 1K)
- 4 butoate (nu am mai legat rezistente la ele, deoarece pot sa le activez pe cele din microcontroller)

Input



Output



Software Design

Acum sunt la faza de scris cod. Ma astept ca joi seara sa termin aceasta faza, daca tot ansamblul electronic functioneaza ok. Acum il testez. Am lasat aceasta faza atat de tarziu, deoarece tot ce trebuie sa fac este sa copiez niste algoritmi pentru efecte de pe internet, sa implementez 4 stari (cate una pentru fiecare efect) si sa citesc input-ul si sa scriu output-ul, lucru pe care l-am facut deja in laborator.

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 plănuieți să le implementați
- (etapa 3) surse și funcții implementate

Rezultate Obținute


Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Deoarece proiectul nu este încă gata, nu pot trage niste concluzii solide, cu excepția faptului că amplificatoarele operationale sunt o durere de cap. În momentul de față, cel mai mult îmi este teama de sample rate-ul procesorului, deoarece nu este foarte rapid, deci voi pierde foarte mult din calitate. Pe lângă asta DAC-ul ales nu este ideal, deoarece acesta are o rezoluție pe 8 biti.

Download

<https://drive.google.com/drive/folders/0BtxpF6iLleRbWVMSzVqWEZuWmM?usp=sharing>

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:prj2017:avoinescu:dumitru_alin**.

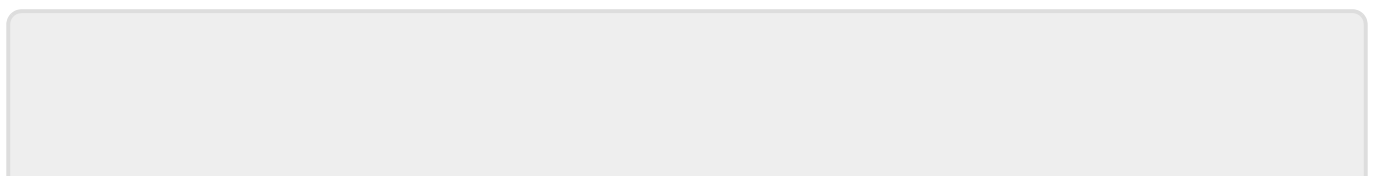
Jurnal

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**.

- Documentația în format [PDF](#)



From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/ddragomir/sergiuw>



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