

Sistem hibrid de control programabil pentru efecte de chitara

Introducere

Este un dispozitiv compact care contine intern mai multe efecte audio. Acesta ofera doua moduri de operare:

1. Modul individual: Utilizatorul poate activa sau opri fiecare efect integrat folosind comutatoare dedicate (ca si cum ar avea pedalele integrate).
2. Modul de Preset-uri: Microcontrolerul ESP32 coordoneaza aceste efecte, permitand utilizatorului sa ruleze configuratii deja programate (de exemplu, o setare specifica de 'Solo' care activeaza instantaneu distorsia si reverb-ul la parametri stabiliti).

Scop

Scopul este de a oferi un sistem complet de procesare a sunetului intr-o singura unitate. Elimina nevoia de a cumpara, alimenta si conecta mai multe pedale separate, oferind în schimb o solutie integrata, usor de transportat si programabila.

Ideea de la care am pornit

Am pornit de la dorinta de a eficientiza setup-ul unui muzician. In loc de un pedalier voluminos si plin de cabluri care se pot defecta, am vrut sa integram cele mai importante efecte intr-o singura carcasa inteligenta, unde controlul analogic se întalneste cu flexibilitatea digitala oferita de ESP32.

De ce este util

1. Portabilitate: Ai toate efectele necesare intr-un singur dispozitiv compact.
2. Simplitate: Nu mai exista probleme de conexiune intre pedale sau zgomot de fond cauzat de cabluri multiple.
3. Versatilitate: Posibilitatea de a schimba radical sunetul printr-un singur preset programat.

Descriere generală



Modul de functionare si interactiune

1. Sunetul chitarei intră în TL-072, unde este amplificat, trece prin etajul de Diodes Gain unde primește caracterul analogic de distorsiune
2. Semnalul analogic este transformat în flux digital de către PCM1802 și trimis prin protocol I2S către ESP32
3. ESP32 verifică starea comutatoarelor, dacă este selectat un preset, microcontrolerul aplică transformări matematice asupra fluxului audio digital în timp real
4. Datele procesate sunt trimise către PCM5102, care recrează unda sonoră analogică
5. Semnalul trece prin releul G5V-2 către mufa de ieșire, ESP32 dictează mixul dintre semnalul curat și cel procesat

Hardware Design

Idei de baza:

Pentru ca datele / semnalele sa fie procesate in acelasi timp, la aceeasi frecventa al ceasului, am folosit 2 pini de GPIO (IO25 si IO27) al ESP32-ului pentru a face conexiunea fizica intre PCM1802 si PCM5102a.

Fiindca folosim footswitch latching, trebuie sa il actionam pentru a schimba configurarea sistemului. Pentru a sti in ce preset ne aflam in orice moment, doar citim valoarea globala "preset" pentru a verifica starea pedalei.

LM2596 este folosit pentru alimentarea (aproape) intregului sistem. Acesta converteste cei 9V care vin din cablul de alimentare conectat la priza si ii transforma in 5V. Cei 9V sunt folositi pentru a alimenta amplificatoarele operationale.

Lista de componente

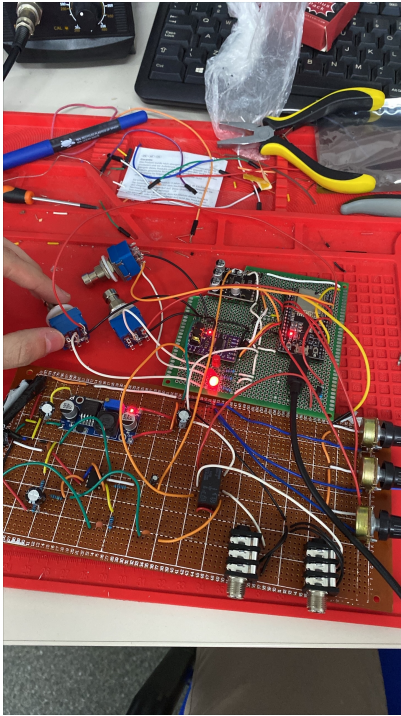
- ESP32-WROOM-32U
- PCM1802 24bit
- PCM5102
- 2x TL-072
- Omron G5v-2 5V DC releu
- 2x 6.35mm jacks
- 3x potentiometre
- 3x footswitches

- Rezistente, condensatoare
- Breadboard (mai mult pentru testate), fire
- Placa prototip
- LM2596 DC step-down
- mufa alimentare DC
- PN2222 tranzistor

PIN Configuration

1. G5V-2 5V DC Releu
 1. 5V (ESP32) → PIN1
 2. PIN16 → GPIO19 (ESP32)
 3. PIN8 → PIN9
 4. LROUT (DAC converter) → PIN11
 5. PIN13 → PIN T port jack output
 6. PIN4 → PIN T port jack input
 7. VCC_9V (DC port) → PIN16
2. ESP32 DevKitC V4 → PCM5102a
 1. GND3 → GND
 2. 5V → VIN_5V
 3. GPIO25 → LCK
 4. GPIO26 → DIN
 5. GPIO27 → BCK
 6. UNCONNECTED → SCK
3. ESP32 DevKitC V4 → PCM1802
 1. 5V → VIN
 2. GND → GND
 3. GPIO27 → BCK
 4. GPIO25 → LRCK
 5. GPIO32 ← DATA (OUT)
 6. GPIO0 → SCK
 7. DISCONNECTED → FSX
 8. LEFT OUT → MODE0
 9. LEFT OUT → MODE1
 10. SOLDERED PINS → FMT0
 11. LEFT OUT → FMT1
 12. LEFT OUT → OSR
 13. LEFT OUT → BYPAS
 14. (TO ITSELF)3v3 → PDWN





Software Design

Mediul de dezvoltare: am folosit VSCode IDE cu extensia PlatformIO

Algoritmi si structuri implementate:

- Aplicarea efectului de overdrive → algoritmul rotunjeste forma undei. Este ca o forma de distorsie, mult mai clean decat distorsia propriu-zisa, inasa ceea ce face este sa treaca de limita producerii sunetului “curat”. Pentru a obtine un astfel de efect, mai intai amplificam semnalul cu valoarea citita din potentiometru, iar apoi “rotunjim” semnalul. Aceasta este formula folosita:

```
semnalAmplificat = semnal * valoareCititaPotentiometru  
semnalOut = semnalAmplificat / (1 + abs(semnalAmplificat))
```

- Aplicare efectul de distortion → aplicam o amplificare asupra semnalului de input si apoi limitam semnalul in functie de anumite threshold-uri setate manual.

```
semnalDistorsionat = semnal * valoareCititaPotentiometru *  
constantaAmplificare  
  
if (semnalDistorsionat > threshold1) semnalDistorsionat = threshold1  
else if (semnalDistorsionat < threshold2) semnalDistorsionat = threshold2  
else ...
```

- Aplicare efect de Fuzz → functioneaza in 2 faze:
 1. In prima faza, daca nu este actionata o coarda sau sunetul curent nu trece de pragul de 0.05, setam automat semnalul la 0 pentru a evita zgomotul inutil
 2. In a doua faza, am lovit una dintre corzi, iar semnalul este limitat de 2 praguri, nivelul superior > 0.8, iar nivelul inferior poate sa ajunga la valori negative, precum -0.4

```
semnalFuzz = semnal * valoarePotentiometru * amplificareFuzz

// corzi lasate liber, neactionate

if (abs(semnalFuzz) < 0.05) semnalFuzz = 0

// corzi actionate

if (semnalFuzz > threshold1) semnalFuzz = threshold1
else if (semnalFuzz < threshold2) semnalFuzz = threshold2
```

Notiuni din laboratoare folosite in timpul implementarii proiectului:

1. Laborator GPIO → pentru setarea bitilor de intrare / iesire, pentru citirea datelor primite de la PCM1802 (convertorul ADC extern) si pentru trimiterea datelor catre convertorul DAC (PCM5102a), cat si pentru footswitch-uri si potentiometre
2. Laboratorul de ADC → pentru citirea tensiunii din potentiometre → in functie de valoarea citita din potentiometru, putem sa aplicam unul dintre efectele: Distorsie, Overdrive sau Fuzz. Pentru citirea semnalului analogic, folosesc PCM1802 extern dedicat, care se ocupa de transformarea semnalului analogic in digital
3. Laborator UART → in principal, il folosesc doar in faza testarea → verificarea valorilor citite din potentiometre pentru a vedea ce valori obtin in urma aplicarii efectului de Fuzz
4. Laborator de Intreruperi → in acest proiect, footswitch-urile functioneaza ca niste butoane pentru preset-uri, ESP-ul trebuie sa receptioneze apasarea butonului pentru a seta variabile globale "preset"

Arhiva

[Arhiva proiectului](#)

Bibliografie/Resurse

- [ESP32-DevKitC V4 \(ESP32-WROOM-32U\)](#)
- [PCM1802 ADC Converter](#)
- [PCM5102a Stereo DAC I2S](#)
- [TL072cp Op Amp](#)
- [LM2596 StepDown Buck Converter](#)
- [G5V-2 5V DC 2 Pole Relay](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/florin.stancu/razvan.bobes>

Last update: **2026/05/26 16:53**

