

# Arduino Guitar Pedal

Cristina Grigore

## Introducere

Ideea proiectului este de a implementa o pedala cu efecte pentru chitara cu diferite efecte programabile. Pedala va prelua input-ul generat de dozele chitarei si le va aplica diverse distorsiuni/efecte. Semnalul va trece printr-un ADC (convertor analog-digital), va fi modificat si transmis inapoi printr-un DAC (convertor digital-analog).

AM ales acest proiect deoarece (daca va fi functional 😊) imi va fi extrem de folositor atat in practica cat si pentru a intelege mai bine cum functioneaza procesoarele de sunet din comert.

## Descriere generală



## Preamplificare

Primul amplificator operational de pe TL082 realizeaza o amplificare de tensiune destul de mare, aduce semnalul de la chitara la "line level" si il inverseaza. Cand iese semnalul din op amp este impartit in 2 intre inputul Arduino-ului si butonul de clean/semnal ne-distorsionat.

## Input Arduino

Inputul de la Arduino ia semnalul audio de la chitara si il comprima la aprox 1.2V pt ca asa este configurat pinul AREF. Dupa asta, semnalul e trimis la pinul 0 si e convertit la digital prin ADC-ul inclus in Arduino.

## Arduino

La pinul analog A3 e conectat un potentiometru de 100k, iar la A2 switch-ul rotativ cu 10 pozitii. Switch-ul rotativ cu 6 pozitii care functioneaza similar cu un potentiometru, dar fiecare pozitie are o rezistenta discreta asociata cu el. In functie de ce pozitie selectez, se va crea un divizor de tensune diferit.

Pentru crearea efectelor: outputul de la Arduino se leaga cu un potentiometru conectat la al 2-lea amplificator operational de pe TL082 care se ocupa de mixatul semnalului si il inverseaza din nou. De aici semnalul mai trece printr-un condensator si dupa la jack-ul de output.

## Output Mixer

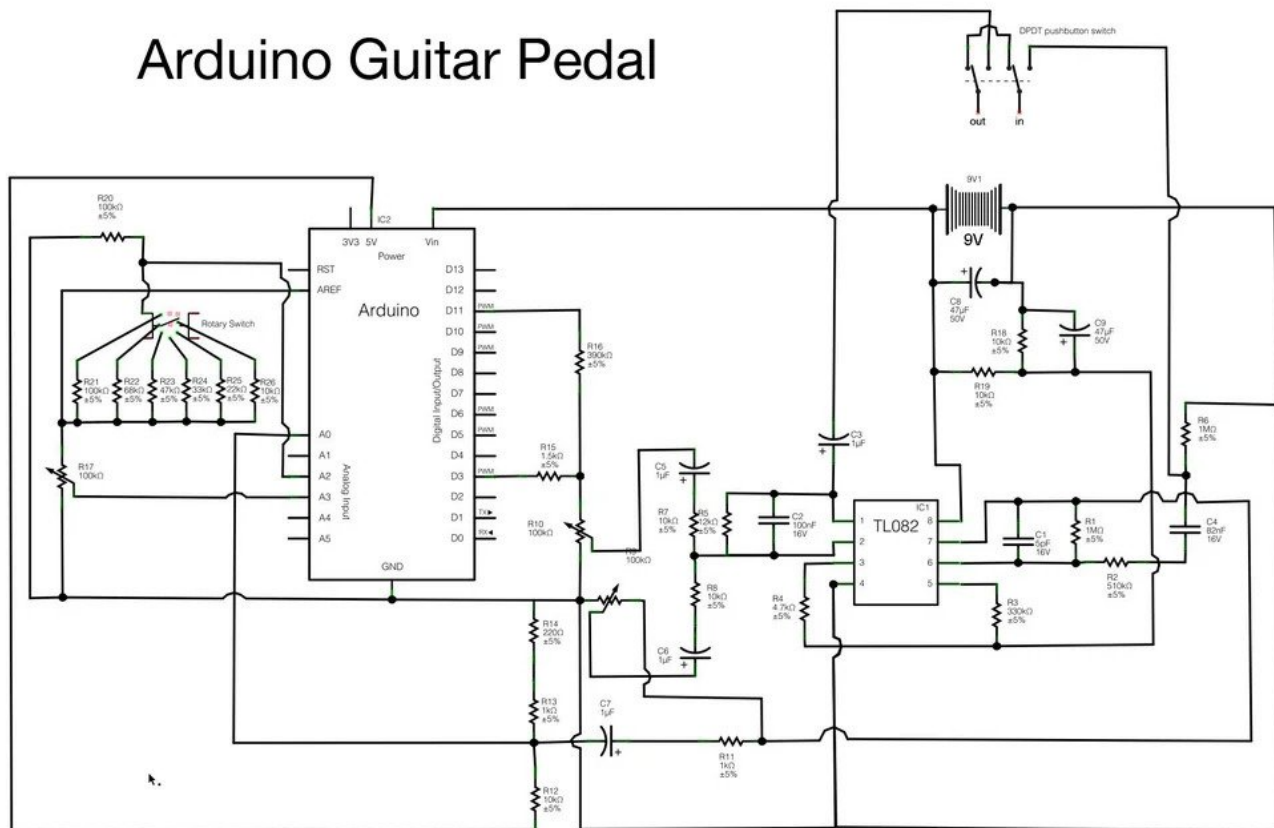
Output-ul Arduino-ului e conectat cu un potentiometru de 100k care intra in cel de-al 2-lea amplificator operational din TL082. Acest potentiometru este folosit impreuna cu semnalul clean de la celalalt potentiometru 100k. Acesta mixeaza cele 2 semnale intr-unul singur pe care il inverseaza pentru a-l aduce in faza cu semnalul original provenit de la chitara. De aici semnalul trece printr-un condensator de 1uF si catre jack-ul de output.

## Hardware Design

Listă de piese

- Arduino Uno
- Mufa Jack Mama
- Switch rotativ
- buton cu retinere
- potentiometre
- fire
- condensatoare
- rezistente

# Arduino Guitar Pedal



## Software Design

Mediu de dezvoltare: Arduino IDE, Visual Studio Code  
 Librarii utilizate: Arduino.h

## Setup

In etapa de setup, configurez Timer-ul 0 si Timer-ul 2 pe fastPWM 8-bit inverting, prescaler 2. Configurez ADC-ul: tensiunea de referinta la 1.1V si prescaler-ul la 32.

```
void waveformGenerationMode(int pin) {
  int timer = getTimer(pin); //select Timer 2/0
  //setting the waveform generation mode bits WGM to 011 selects fast PWM
  int wgm = B011;

  //set WGM2:0 = 3 so that counter counts to 0xFF instead of OCR0A
  if(timer == 0) {
```

```
TCCR0B &= ~(B1 << 3); // clear WGM02
TCCR0A &= ~B11; // clear WGM01 and WGM00
TCCR0A |= wgm; // set WGM01 and WGM00
} else if(timer == 2) {
  TCCR2B &= ~(B1 << 3); // clear WGM23
  TCCR2A &= ~B11; // clear WGM21 and WGM20
  TCCR2A |= wgm; // set WGM21 and WGM20
}
}
```

```
analogReference(INTERNAL);
//set the ADC's prescaler to a low value (32)
ADCSRA &= ~B111; // clear analog prescale
ADCSRA |= B101;
```

## Input Switch

Pentru citirea input-ului de la utilizator citesc valoarea de pe pinul A2 pentru pozitia switch-ului rotativ si obtin o valoare discreta(variabila mode). pentru fiecare dintre cele 9 pozitii, desi am ales sa implementez doar 2 efecte pentru 2 pozitii din considerente de lipsa de memorie.

Variabila fx contine inputul de pe potentiometrul P1 ce ajusteaza intensitatea efectului.

```
void readKnobs(){
  mode = analogRead(2);
  //dividing by 75 ensures proper discrete values
  //for if statements above
  mode = mode / 75;

  fx = analogRead(3);
}
```

## Efecte

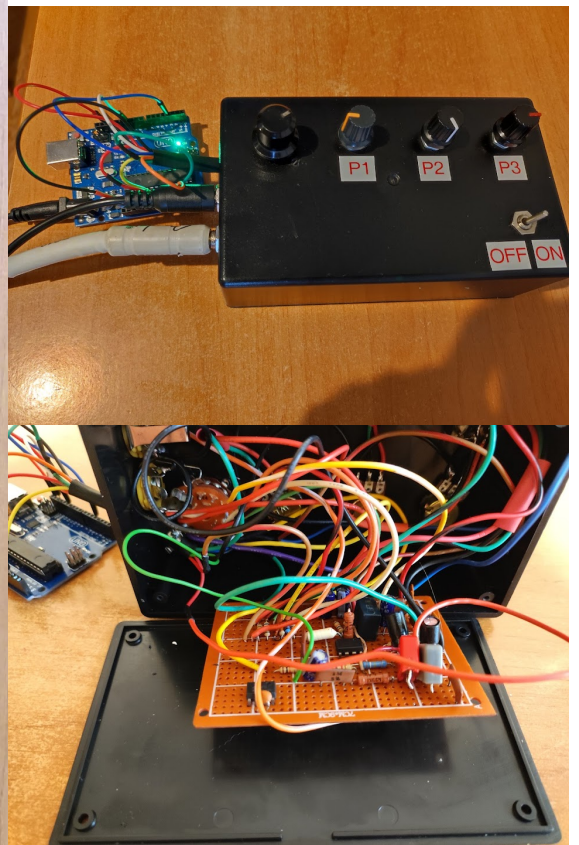
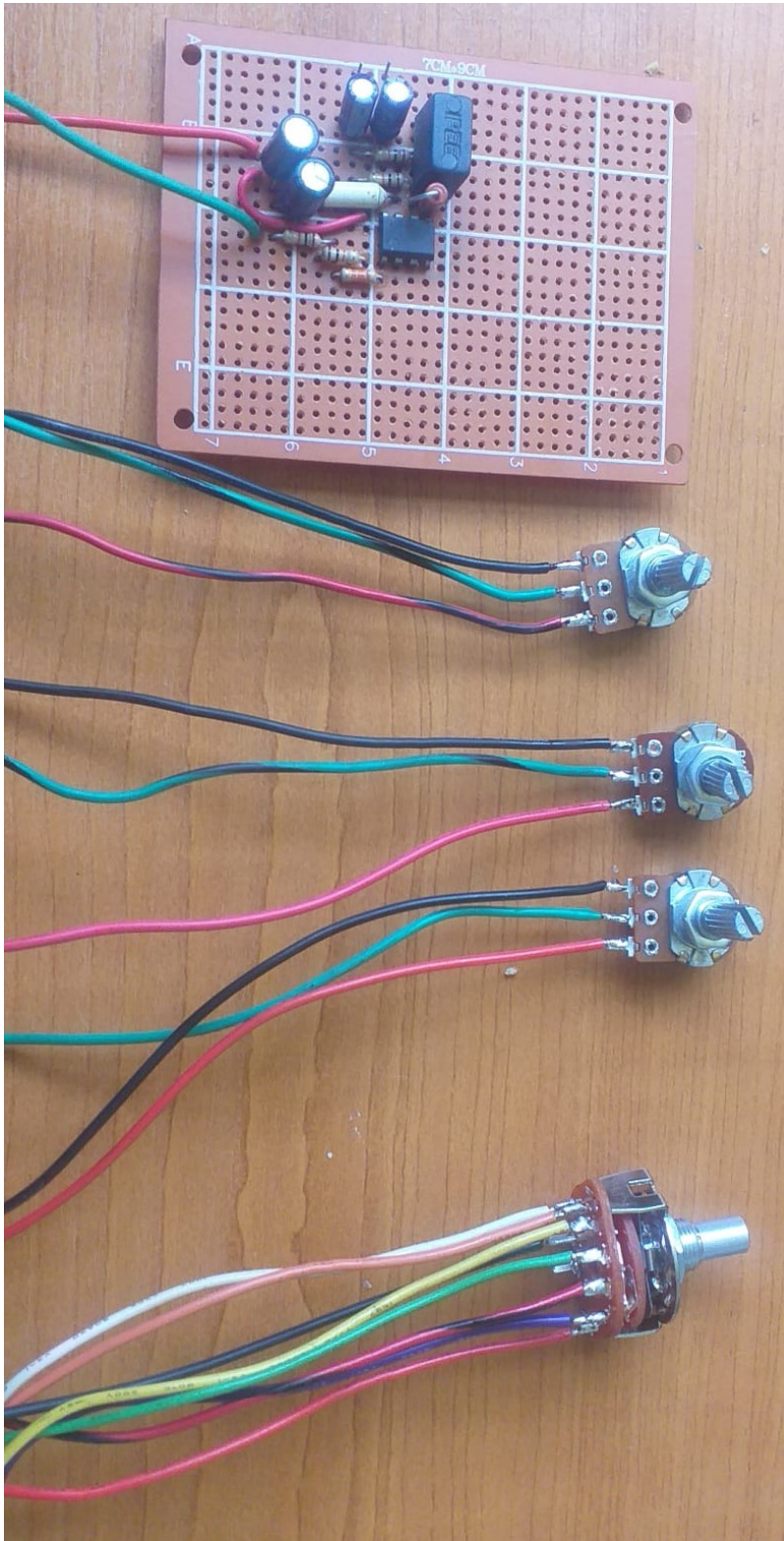
Am ales sa implementez 2 efecte: un delay si un bitcrusher.

```
// *****
// ***Overdrive***
// *****
if(mode == 13){
  value50 = 1 + ((float) fx / (float) 20);
  byte input = analogRead(left);
  input = (input * value50);
  output(input);
}
```

```
}
```

```
// *****  
// ***bitcrush**  
// *****  
if(mode == 6){  
  value300 = 1 + ((float) fx / (float) 3);  
  if(delayed > value300) {  
    byte input = analogRead(left);  
    input = (input >> 6 << 6);  
    output(left, input);  
    delayed = 0;  
  }  
  delayed++;  
  
}
```

## Rezultate Obținute



**Concluzii**

**Download**

[arduino\\_guitar\\_pedal.zip](#)

## Jurnal

08.05.2022 - Creare wiki

## Bibliografie/Resurse

<https://www.instructables.com/Arduino-Guitar-Pedal/> <https://www.electrosmash.com/pedalshield-uno>

[Export to PDF](#)

From:

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

Permanent link:

[http://ocw.cs.pub.ro/courses/pm/prj2022/sgherman/guitar\\_pedal](http://ocw.cs.pub.ro/courses/pm/prj2022/sgherman/guitar_pedal)

Last update: **2022/06/01 23:50**

