**Laurent Tremblay**

*Gr.001*

**Projet final**

*Pédale numérique pour effets de guitare*

Travaille présenté à

**M. Jasmin St-Laurent, M. Claude Barbaud et Mme. Chantal Milette**

dans le cadre du cours

*Projet de fin d’études, 243-631-MA*

**Collège Maisonneuve**

2018-05-11

Table des matières

**Aucune entrée de table des matières n'a été trouvée.**

**Introduction**

Dans le cadre du cours de projet de fin d’études, j’ai réalisé un projet électronique ayant comme fonction de modifié un signal audio provenant d’une guitare à l’aide d’un FPGA. Celui-ci possèderait une interface graphique pour afficher à l’utilisateur l’effet actif ainsi que les paramètres modifiables des effets par des potentiomètres externes. L’utilisateur aurait à sa disposition deux boutons poussoirs pour avancer et reculer entre les multiples effets ainsi qu’un bouton pour activer les effets audios ou non. Parmi ces effets, on peut retrouver un effet de distorsion et un effet de trémolo. L’effet de distorsion consiste à limiter le signal audio pour ajouter des harmoniques à celui-ci, et l’effet de trémolo consiste à moduler le signal audio à l’aide d’une onde de fréquence très basse, ce qui donne l’impression qu’on augmente et réduit automatiquement le volume du son.

J’ai choisi ce projet car je joue de la guitare dans mon temps libre et qu’il est important d’avoir des pédales pour modifier le son de la guitare. Il y a aussi le fait que les pédales à effets pour guitare sont très couteuses, car une pédale peut coûter entre 75 et 500 dollars, et celle-ci ne produit qu’un seul effet. On peut donc s’imaginé qu’on peut se retrouver avec une grosse facture seulement en possédant quelques effets.

L’objectif du projet était donc de pouvoir créer tous les effets voulus pour une fraction du prix que l’ensemble des pédales standard pourraient coutés. De plus, ce projet me donnait raison d’utilisé un FPGA, car ceux-ci sont extrêmement puissant pour le traitement de signaux numériques, et par ce fait approfondir mes connaissances au niveau de la composante et du langage VHDL.

**Bilan des travaux**

\*\* bilan a la fin

**Description générale**

Schéma synoptique

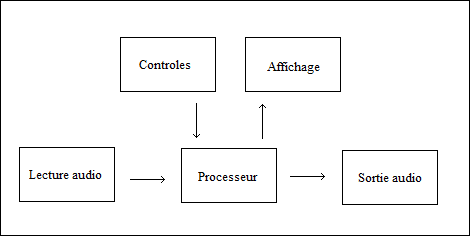
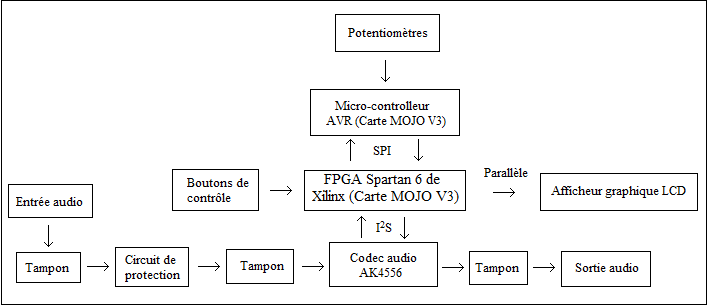


Schéma de principe

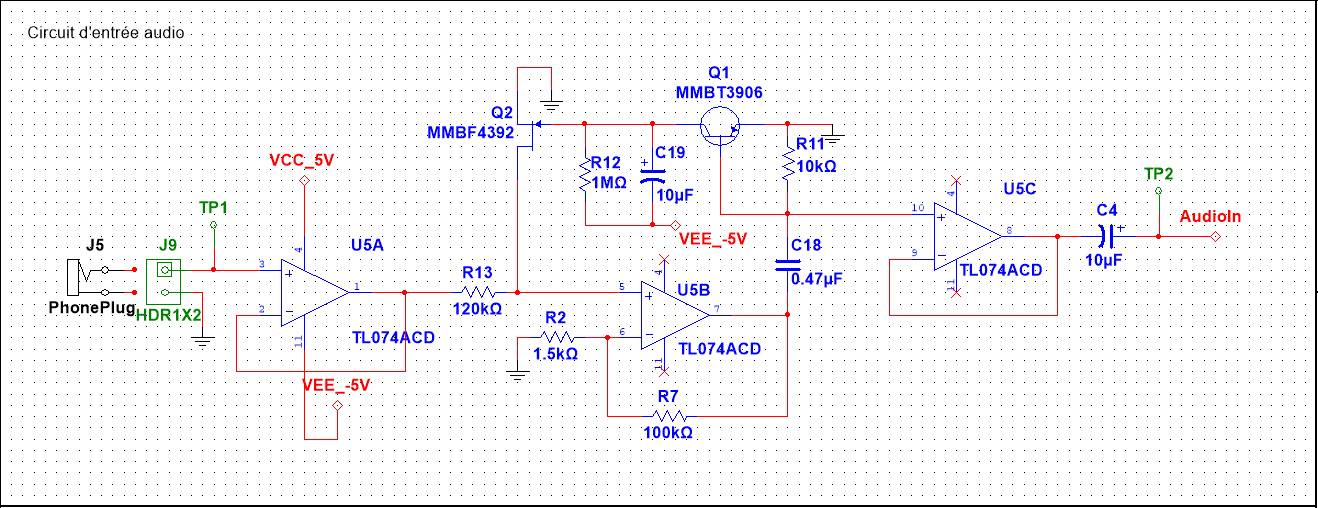


Il y a plusieurs principes utilisés dans ce projet. Ceux-ci peuvent être séparés en deux : soit une partie analogique et une partie numérique. Pour la partie analogique, il y a l’utilisation de l’amplificateur TL04 comme tampon et comme circuit de protection pour transmettre le signal audio au codec audio AK4556. Il est important d’utilisé un circuit tampon pour isoler les impédances, soit celle de l’entrée audio, celle de l’entrée du circuit d’acquisition du codec audio et celle à la sortie du codec audio. De plus, il faut aussi utiliser un circuit de protection pour limiter le niveau de tension de l’entrée audio, car elle pourrait endommager l’entrée du circuit d’acquisition du codec. Dans mon cas, j’ai choisi d’utilisé un circuit de compression audio : ce circuit ajuste automatiquement la tension du signal audio pour qu’elle soit à une tension fixe, prévenant les surtensions. En plus, ce choix permet d’avoir une meilleure résolution de notre signal audio, cependant il vient couper la dynamique du signal audio. Le deuxième aspect de la partie analogique est l’utilisation d’un codec Audio, celui-ci permettant le sur-échantillonnage et le filtrage du signal audio. La communication entre le codec Audio et le FPGA se fait à l’aide du protocole I2S.

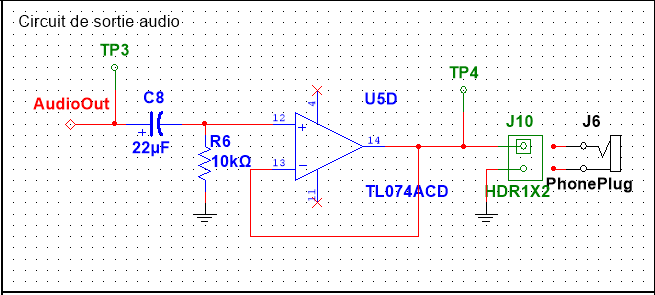
Pour la partie numérique, on peut retrouver une interface pour l’utilisateur ainsi que plusieurs protocoles de communications. L’utilisateur possède trois boutons de contrôles, soit deux pour changer de pages (précédent et prochain) ainsi qu’un bouton activant la pédale. L’utilisateur peut aussi modifier les paramètres des effets à l’aide de trois potentiomètres. Les tensions de ces potentiomètres sont lues à l’aide des convertisseurs analogique à numérique du microcontrôleur AVR de la carte MOJO V3 et sont ensuite transmit via le protocole SPI au FPGA Spartan 6. Le dernier item de l’interface utilisateur est un afficheur graphique LCD. Le FPGA envoi des données en parallèle à l’afficheur étant donné qu’on ne manque pas de broches, que c’est plus simple et que c’est rapide.

**Description précise**

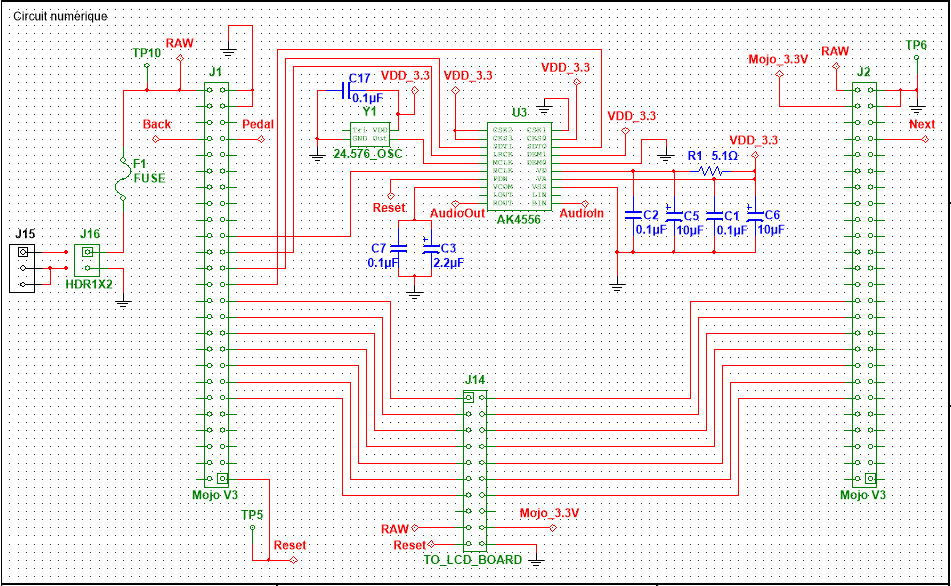
**Schéma de montage**

Plaquette principale

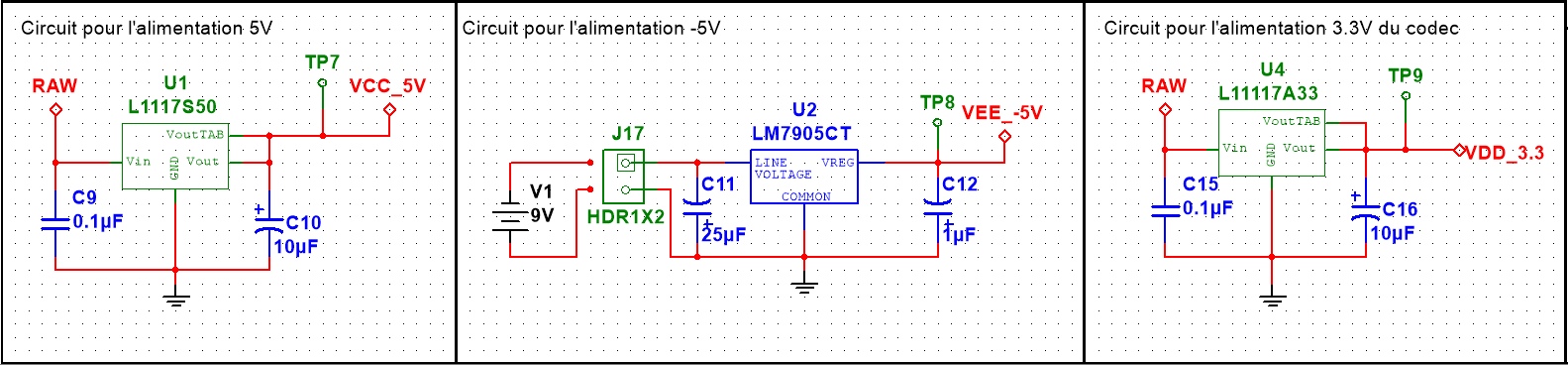
Le circuit d’entrée audio comporte trois étages. Le premier étage est un suiveur qui isole l’entrée l’impédance de la guitare et celle du circuit. Le deuxième étage est un circuit de compression audio, régulant le signal d’entré pour qu’il soit toujours à 1.31 volts. Ce circuit est important pour protéger l’entrée du convertisseur analogique à numérique du codec audio AK4556. De plus, il permet d’utiliser une plus grande résolution du convertisseur analogique à numérique. Le troisième étage est un autre suiveur qui isole l’impédance du circuit de compression et l’impédance du convertisseur analogique à numérique du codec audio. Il possède un condensateur à sa sortie pour couper le décalage courant continue.



En contrepartie, le circuit audio comporte un seul étage, soit un suiveur qui vient isoler la sortie du convertisseur numérique à analogique du codec audio AK4556. Il possède aussi un filtre passe haut à son entrée pour couper le décalage courant continue.

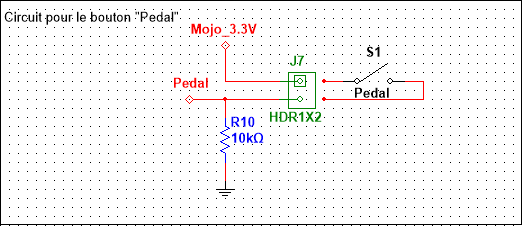


Le circuit principal du projet se retrouve ici. Le connecteur J15 est une entrée pour le bloc d’alimentation 9V. Le connecteur J14 permet le lien entre la plaquette principale et la plaquette pour l’écran graphique. Les connecteurs J1 et J2 font le lien entre la carte Mojo et les circuits du projet. Dans ce circuit, on retrouve le codec AK4556, qui lui est responsable pour l’échantillonnage du signal audio.

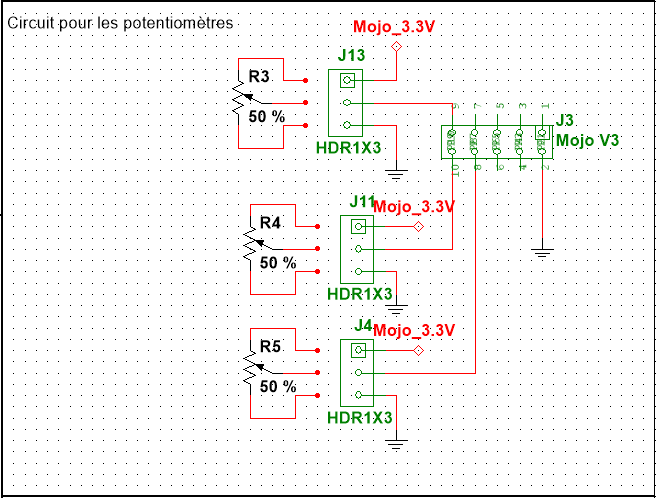


La plaquette utilise trois alimentations, soit 5 volts et -5 volts pour l’alimentation de l’amplificateur opérationnel TL074 utilisé dans les circuits audios, ainsi que 3.3V pour alimenter le codec audio. Il est important de noter que la carte Mojo utilise l’alimentation fourni à la carte pour produire ses propres tensions.

\*\*\*\*\* MODIFICATION -> LM7905 C11 = 22uF \*\*\*\*\*\*\*\*\*



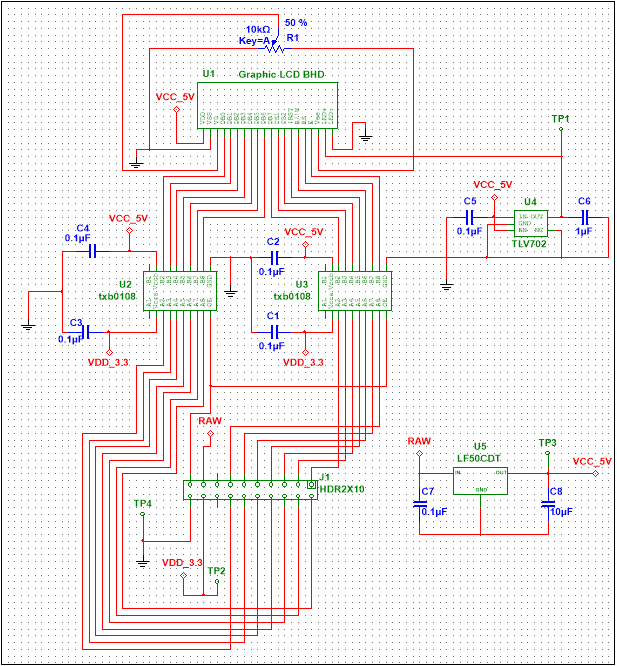
Le même circuit est utilisé pour le les boutons « Pédale », « Suivant » et « Précédent ».



Ce circuit est utilisé pour pouvoir lire la tension des potentiomètres utilisés pour modifier les paramètres des effets. Ceux-ci sont connectés au connecteur J3, qui lui est directement lié aux convertisseurs analogiques à numériques du microcontrôleur Atmel de la carte Mojo.

\*\*Le circuit principal complet peut être trouvé à l’annexe.

Plaquette pour l’écran graphique



Ce circuit est le circuit complet pour interfacer l’afficheur LCD avec le FPGA. Le connecteur J1 permet la liaison entre les deux plaquettes. Le circuit possède deux régulateurs : un LF50CDT régule la tension à 5V, soit l’alimentation nécessaire pour la logique de l’écran, et un TLV02 qui régule la tension à 3.5V, soit l’alimentation nécessaire pour alimenter le rétro-éclairage de l’écran. Le circuit possède aussi deux TXB0108, qui eux convertissent la tension provenant des tampons de sortie du FPGA, soit 3.3V, à la tension nécessaire pour la logique de l’écran, soit 5V.

**Circuit imprimé**

**Câbles**

Ribbon cable -> 20, AWG,

Crimped cables -> AWG, lenght

**Programmes**

**Boîtier**

**Caractéristiques et limites de l’appareil**

Consommation d’énergie : Alimentation positive : 1,35 Watts (9v x 150mA)

Alimentation négative : 0,045 Watts (9v x 5mA)

Autonomie : 100 heures avant de changer la batterie (~550mAh /5mA)

Type d’alimentation : Murale et batterie 9V

Milieu :

Température :

Plage de fonctionnement :

Vitesse d’acquisition du signal audio : 48kHz

**Liste de matériel**

Partie 1



Partie 2



\*\* Ajouter : Régulateur 5V DPAK pour écran LCD

**Évaluation des coûts de production**

**Procédure d’assemblage**

**Procédure de calibration**

La seule calibration nécessaire pour le projet est d’ajuster le potentiomètre « R1 » sur la carte de l’écran LCD pour ajuster le rétro-éclairage de l’écran pour avoir la meilleure image.

**Manuel de l’utilisateur**

**Conclusion**

**Annexes**