

Projet Innovation¹

Pédale d'effet numérique pour guitare électrique

Rapport 1

AHUJA, Sanju

IGLESIAS MANRÍQUEZ, Esteban Felipe

MARTINEZ ARROYO, Andrea Lorena

PARENT, Nicolas

WIEDEMANN, Andreas

Encadrant : FANTON, Jean-Pierre

YANG, Siyu

¹ Projet Innovation - Pédale d'effet numérique pour guitare électrique - Rapport 1 - Groupe P5C413AG de Sanju, Esteban, Lorena, Nicolas, Nicolas et Siyu est mis à disposition selon les termes de la [licence Creative Commons Attribution - Partage dans les Mêmes Conditions 4.0 International](#).

Plan

Résumé	3
Introduction	4
Travail réalisé	5
Travail à venir	10
Bilan du travail en équipe	11
Annexe 1 : Réalisation des 3 modes	12

Resumé

Le projet "*Pédale d'effet numérique pour guitare électrique*" a été présenté par un étudiant de l'École Centrale Paris pour les projets innovation du Semestre 7 et Semestre 8 pendant de l'année 2014-2015.

Il s'agit de la réalisation d'une pédale d'effet numérique pour guitare électrique, dont les paramètres peuvent être changés suivant les mouvements du corps du musicien en utilisant Arduino. C'est un projet Open Source et Open Hardware.

Dans cette rapport nous présenterons nos premiers pas, c'est-à-dire la mise en place de l'équipe, l'écriture du cahier des charges, l'organisation du travail, la réalisation d'un premier prototype dont nous allons étendre les fonctionnalités afin d'accomplir nos objectifs. Nous détaillerons aussi les prochaines étapes de notre travail ainsi qu'un premier bilan de notre travail en équipe.

Introduction

But:

Réaliser une pédale de guitare permettant au guitariste d'utiliser un mode expérimental dans lequel il contrôle les paramètres sonores grâce aux mouvements de son corps

Motivation:

Du point de vue d'un guitariste, quand il joue de la guitare, il est très incommod de changer les effets sonores avec ses mains. Ce n'est pas le plus efficace pour experimenter avec des effets sonores et pour chercher la meilleure manière de les combiner. Il y avait donc un besoin : construire une pédale qui élimine ce problème en permettant au guitariste de continuer à jouer tout en changeant les effets grâce à une reconnaissance de certains mouvements.

D'un autre coté, les projets *open source* sont très importants pour l'éducation, le partage des idées et l'innovation. Ils permettent que le processus d'invention soit plus court en réduisent le travail des développeurs, tout en partageant la connaissance, et en mettant les idées au service du grand public. Dans ce sens, un projet libre sur Arduino, plateforme standard en matière d'innovations en électronique, peut être l'antécédent d'un projet fructueux pour démontrer aux élèves ingénieurs l'existence et importance de ce mode d'invention ainsi que l'utilité de cette voie alternative pour l'ingénieur.

Objectif:

Notre objectif est la réalisation d'une pédale modifiant le son émis par une guitare électrique en utilisant des effets digitaux, dont les paramètres peuvent être changés suivant les mouvements du corps du musicien.

Le projet sera développé sur les principes des projets *Open Source* et *Open Hardware*.

Moyen:

Nous utiliserons un capteur afin de détecter les mouvements du corps du guitariste. En les utilisant comme données entrantes pour un code enregistré dans l'Arduino, la pédale changera les effets sonores. Il s'agit d'une exploration des nouvelles manières de créer de la musique et une utilisation des technologies digitales pour l'art.

Travail réalisé

Le travail réalisé et les tâches accomplis pendant les cinq séances passées sont décrites ci-dessous.

Séance 0: 16 Septembre: Formation de l'équipe.

Pendant cette séance, l'équipe a fini de se former. Cette journée à été une journée de recrutement, elle nous a permis de nous rencontrer et de commencer à parler du projet.

Nous avons fini la journée de travail avec un rendez-vous entre quelques participants du groupe et le professeur encadrant, Jean-Pierre Fanton.

Séance 1 : 23 septembre: Vision Commune du Projet et rédaction du cahier des charges.

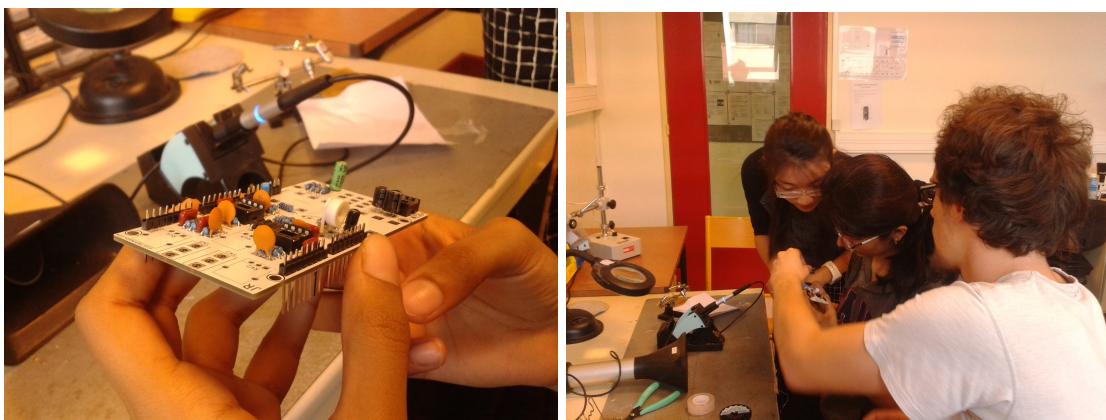
Cette séance était la première séance où la totalité du groupe était présente. Nous avons donc commencé avec l'explication de l'idée par l'étudiant qui a présenté le projet et nous avons construit notre propre vision commune du projet. La rédaction du cahier des charges a été proposé à tout le groupe, document qui a été terminé pendant la semaine. Voici le lien vers notre cahier des charges disponible sur le dépôt GitHub de l'équipe.

<https://github.com/pedaleECP/pedalSensors/blob/master/Documentation/P5C413AG%20-%20Pedale%20-%20Cahier%20de%20charge%20-%20V1.pdf?raw=true>

Séance 2 : 30 septembre: Réalisation du premier prototype de pédale

Le projet est basé sur le projet pedalSHIELD, une pédale numérique d'effet pour Arduino. Pendant cette séance nous avons fabriqué la pedale pedalSHIELD,dont nous allons étendre les fonctionnalités afin accomplir nos objectifs.

Nous avons aussi reçu la visite du professeur encadrant Jean-Pierre Fanton puis fini la journée en faisant des essais dans la salle de musique (salle M) (lien sur vidéo de l'essai: <http://goo.gl/LMfHhU>)



Fabrication du pedalSHIELD dans le laboratoire LISA

Séance 3 : 7 Octobre. Division du travail en tâches et attribution des dates limites

Pendant cette séance nous avons commencé à travailler avec Asana, logiciel en ligne de gestion des projets. Nous avons donc discuté pour diviser le projet en différentes tâches principales, que nous avons ensuite divisée en actions simples.

Finalement, nous nous sommes mis d'accord au niveau des dates limites pour chaque tâche et de ses tâches filles dans le but d'arriver à un planning du groupe visible. nous avons fini par la répartition des premières tâches en différents équipes. Nous avons décidé de ne pas nous diviser en groupes permanents, mais en groupes par tâches, ce qui permettra à chaque un d'expérimenter dans les différentes composantes du projet (informatique, électronique...)

The screenshot shows the Asana web interface for the 'Pédale' project. The left sidebar includes links for 'My Tasks', 'My Inbox', 'Pédale', 'PROJET INNO - PEDALE', 'Team Calendar', 'PROJECTS +', 'Rapports', 'Open Source Projet', 'New Project', and 'MORE TEAMS IN STUDENT.ECP.FR'. The main area displays a task list with the following tasks:

- 2 Contacter pedalSHIELD team Oct 25 >
- 3 Lire le code Oct 6 >
- 4 Réaliser l'état de l'art Oct 10
- 5 ✓ Fabriquer pedalSHIELD Sep 30
- 6 ✓ Experimenter avec pedalSHIELD et des guitares Sep 30 >
- 7 ✓ Documenter

Below this, under 'Créer notre pédale:', there is a sub-task list:

- 9 3 Modes Yesterday >
- 11 NP Changer les boutons Nov 4 < (highlighted)
- 12 SA Intégration des Capteurs Nov 10 >
- 13 SA Écran LCD Nov 10 >
- 14 NP Sauvegarde des paramètres Nov 10 >
- 15 NP Make capteurs wireless Nov 17 >
- 16 SA Display-Button interaction Nov 25 >
- 17 Sensibilité Controlée Nov 25 >
- 18 Multi-Effets Dec 2 >
- 19 Présentation de la pédale Jan 23, 2015

At the bottom, there is a 'Videos' section with links to 'Intro to Asana', 'Teamwork Without Email', 'Set Goals with Calendars', 'Plan Your Day in Asana', 'Plan & Run Meetings in Asana', and 'Capture Ideas in Asana'. A right-hand sidebar shows a detailed view of the 'Changer les boutons' task, which has the following sub-tasks:

- ✓ étude des boutons alternatifs et choisir un Oct 9
- ✓ Contacter Didier Oct 9 NP
- make 1 trial with Arduino Nov 4
- ✓ code the "Hello World" Yesterday
- ✓ present to the team Today
- création de schéma électrique Nov 3
- ✓ change the buttons Nov 3
- Demander des nouveaux pot 10k ou voir une alternative Nov 3
- Debug Nov 6

Comments from users Esteban, Nicolas, and Sanju are listed at the bottom of the sidebar.

Planning dans le logiciel Asana.com

Les tâches attribuées lors de cette séance étaient:

Trois modes pour la pédale (Software): Andreas et Siyu.

Pour programmer le micro-contrôleur on utilise le logiciel Arduino en version 1.5.8 Beta, car la 1.0.6 ne supporte pas Due Boards. Normalement Arduino se programme en C, cependant Assembler est aussi supporté.

Notre réalisation des 3 modes se trouve en Annexe 1. Le but est de permettre de passer de la marche/effet/mode expérimental de manière simple.

Changement des boutons:

Pour avoir une interface avec des capteurs sans fils (contrôles à distance) il faut changer les potentiomètres de la pédale pour lire la position relative.

Après une discussion sur comment remplacer nos potentiomètres, nous avons divisé le groupe en deux:

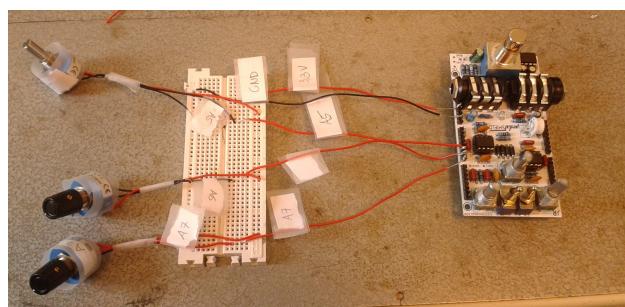
Changement des boutons: (Software): Sanju et Nicolas.

Comment Arduino va récupérer les données des potentiomètres à vis sans fin?

Comment résoudre le problème de position relative?

Changement des boutons(Hardware): Esteban et Lorena.

Comment connecter ces nouveaux potentiomètres à notre pédale pour qu'ils puissent être lus par l'Arduino et le code de Sanju et Nicolas?



Prototype des nouvelles potentiomètres.

Séance 4: 14 Octobre GitHub, Rapport et Bilan du travail en équipe

On a commencé cette réunion en testant l'installation de GitHub pour chacun de nous.

Désormais, nous pouvons donc partager les programmes et sauvegarder leurs changements sur GitHub, permettant de travailler en simultané sans conflits grâce à leurs outils de *Fork* et *Merge*.

Project based in pedalSHIELD by ElectroSmash

22 commits 1 branch 0 releases 3 contributors

branch: master / pedalSensors / +

	SiyuYang authored 43 minutes ago	latest commit 21b8dfdb ea
	Documentation Photos	43 minutes ago
	Effects Putting effects in one folder	4 hours ago
	Our code Andi's nouvelle code de 3Mode	an hour ago
	.gitignore Initial commit	10 months ago
	LICENSE Initial commit	10 months ago
	README.md check from Esteban	5 hours ago
	README.md~ hola	4 hours ago

pedalSHIELD

Check from Esteban

pedalSHIELD is an Arduino Open Source guitar pedal made for guitarists, hackers and programmers

Depôt GitHub de l'équipe.

Ensuite, chaque petit groupe a parlé du travail qu'ils ont fait. Andi et Siyu ont présenté le programme pour réaliser les trois modes (Annexe 1) et la méthode de basculement; Andrea Lorena et Esteban ont montré le prototype des trois boutons à vis sans fin; Nicolas et Sanju ont parlé de leur programme pour les nouveaux boutons. Après d'avoir travaillé pour le présent rapport nous avons fini avec une discussion en groupe pendant le déjeuner concernant nos progrès, nos motivations individuelles pour faire ce projet et de nos intérêts, ainsi que de notre avis sur le travail en équipe.

Logiciels Utilisés:

Nous avons commencé à utiliser les logiciels suivants :

- **Asana:** une plate-forme en ligne de gestion des projets. Elle permet à nous de voir les tâches et ses dates-limites, leurs descriptions, faire des commentaires sur le progrès, designer des tâches, voir un calendar et télécharger des documents.
- **Git/ GitHub:** Git est un logiciel de gestion de versions décentralisé, cependant la plateforme GitHub provisionne un service web pour que le groupe ait accès au même dépôt. Ça nous permet de partager programmes et fichiers ,de travailler individuellement sur le code en même temps, de sauvegarder les changements et de les combiner après.
- **Arduino:** un éditeur en C pour programmer le microcontrôleur Arduino. De plus, Assembler est supporté.
- Documentations avec des **GoogleDocs:** un éditeur en ligne pour écrire les rapports

Nous pensons utiliser les logiciels suivants :

- **LTS spice:** une application pour construire des schémas électroniques et les simuler
- **EAGLE:** une application pour dessiner le plan de PCBs, au cas où nous voudrions changer le PCB de PedalShield

Travail à venir

Le tableau suivant présente notre planning pour les semaines à venir:

Date	Lieu	Tâches à réaliser.
4 novembre	LISA	Continuer à changer les boutons: demander à M. Didier Coudray de nouveaux potentiomètres 10kΩ ou voir une alternative; créer un schéma électrique et le tester avec l'Arduino.
8 novembre		Choisir le logiciel pour créer le schéma électrique du PCB et du circuit.
10 novembre	LISA	Intégrer un capteur de mouvement qui interagit avec les 3 paramètres de la pédale.
10 novembre	LISA	Commander, installer et programmer un écran LCD pour montrer le nom de l'effet et du mode courant.
10 novembre	LISA/Salles de projets	Programmer notre 'master program' afin de sauvegarder les paramètres.
17 novembre	LISA	Connecter le capteur de mouvement sans utiliser de fils.
25 novembre	LISA	Créer l'interface entre l'écran et les boutons.
25 novembre	LISA	Programmer un bouton pour ajuster la sensibilité du capteur de mouvement.
30 novembre	Salles de projets	Écrire le deuxième rapport.
2 décembre	Salles de projets	Programmer les effets.
15 janvier (provisoire)	Salles de projets	Écrire le troisième rapport.
20 janvier	Salles de projets	Préparer la présentation finale.

Bilan du travail en équipe

Jusqu'à présent notre stratégie a été de se répartir en groupes petits (deux ou trois) en fonction des tâches à venir. Comme ça on a une organisation libre et on ne se restreint pas à un type de tâche spécifique (hardware/software, par exemple). Nous avons trouvé cette stratégie efficace: programmation, assemblage des composants, soudure et recherche sur internet sont plus efficaces en petits groupes tout en ayant assez de place pour s'exprimer, sans avoir à faire trop de coordination ce qui permet un travail à la fois rapide et efficace.

Cependant, il est important d'avoir des réunions ensemble, donc nous avons des réunions tous ensemble chaque mardi, comme ça nous pouvons rester informés des progrès de notre projet entier, et nous pouvons aussi discuter et attribuer les tâches à venir et demander de l'aide si un petit groupe ne peut pas résoudre un problème dans le planning. Nous espérons d'améliorer la ponctualité pour les réunions. En même temps, nous voulons maintenir une atmosphère détendue et pas stressante.

Nous sommes très heureux d'avoir un groupe extraordinairement divers: chacun de nous a une nationalité différente (Chine, Inde, Mexique, Allemagne, France et Chili). Le plupart d'entre nous sont étudiants en échange ce qui permet de partager les techniques et des plate-formes que nous utilisons dans nos universités (cf Asana, GitHub, dynamiques de travail en équipe). Après ce semestre, nous les apporterons chez nous et les utiliserons dans nos futurs projets. Il y a donc un enrichissement personnel et culturel.

En outre, l'idée d'un projet Open Source nous enthousiasme, ainsi que le fait de contribuer à la communauté en ligne et d'exposer l'idée aux étudiants à l'Ecole Centrale Paris.

Bibliographie

pedalSHIELD Arduino Guitar Pedal, ElectroSmash [Online]
<http://www.electrosmash.com/pedalshield>

Software: PedalShield, ElectroSmash [Online]
<http://www.electrosmash.com/forum/software-pedalshield>

Annexe 1 : Réalisation des 3 modes

```

int in_ADC0, in_ADC1; //variables for 2 ADCs values (ADC0, ADC1)
int POT0, POT1, POT2, out_DAC0, out_DAC1; //variables for 3 pots (ADC8, ADC9, ADC10)
const int LED = 3;
const int FOOTSWITCH = 7;
const int TOGGLE = 2;
int upper_threshold, lower_threshold;

int footswitch_detect;
int footswitch_detect_last;

const int STANDBY_MODE = 0;
const int BUTTON_MODE = 1;
const int SENSOR_MODE = 2;

int footswitch_mode = STANDBY_MODE;

void setup()
{
    // initialize the footswitch as an input
    pinMode(FOOTSWITCH, INPUT);

    // initialize the LED as an output
    pinMode(LED, OUTPUT);

    // record the state of the footswitch when turned on
    footswitch_detect_last = digitalRead(FOOTSWITCH);

    // initialize serial communication (for edge (on/off) detection)
    Serial.begin(9600);

    //ADC Configuration
    ADC->ADC_MR |= 0x80; // DAC in free running mode.
    ADC->ADC_CR = 2; // Starts ADC conversion.
    ADC->ADC_CHER = 0x1CC0; // Enable ADC channels 0,1,8,9 and 10

    //DAC Configuration
    analogWrite(DAC0, 0); // Enables DAC0
    analogWrite(DAC1, 0); // Enables DAC1
}

```
Program for testing the different modes. In the standby mode the LED is switched off, in the button mode the led blinks ! In the case of an indeterminate mode a text is displayed on the computer monitor.
```
void loop() {

    // It must be considered that pushing the foot switch toggles the state of FOOTSWITCH (LOW->HIGH and HIGH->LOW).
    // Using the variable footswitch_detect_last enables the comparison of FOOTSWITCH to its previous state.
    footswitch_detect = digitalRead(FOOTSWITCH);

    if (footswitch_detect != footswitch_detect_last) {
        footswitch_mode = (footswitch_mode + 1) % 3;
        footswitch_detect_last = footswitch_detect;
    }

    switch (footswitch_mode) {
        case STANDBY_MODE:
            digitalWrite(LED, LOW);
            break;

        case BUTTON_MODE:
            digitalWrite(LED, HIGH);
            delay(1000);
            digitalWrite(LED, LOW);
            delay(1000);
            break;

        case SENSOR_MODE:
            digitalWrite(LED, HIGH);
            break;

        default:
            Serial.print("Invalid state");
    }
}

```