

IUT de Nice

# Rapport SAE Son

Schmitt, Galgani,Arbid  
13/06/2024

## Table des matières

Introduction.....	2
Diagramme de cas d'utilisation .....	2
Diagramme d'exigences.....	3
Diagramme de définition de bloc .....	3
Tâche 1 : Conditionnement et Filtrage du signal.....	5
Module Bluetooth : Gestion et Transmission du Signal Audio .....	5
Processus de Connexion et Transmission du Signal .....	5
Étape 1 : Conditionnement du Signal Audio Stéréo.....	6
Observations Initiales .....	6
Problèmes Identifiés .....	7
Besoin de Conditionnement du Signal.....	7
Conversion du signal stéréo en mono.....	7
Amplification du signal.....	9
Remarque : Alimentation du Système.....	13
Étape 2 : Filtrage du Signal .....	13
Tâche 2 : Conditionnement .....	15
Convertisseur analogique numérique (CAN) .....	15
Comparateur et diviseur d'horloge.....	15
Vizualiseur.....	17
Filtre passe-bas .....	19
Banc de test .....	20
Tâche 3 : Réalisation de la carte de l'amplification de classe D.....	23
Fonctionnement de la carte d'amplification de classe d préliminaire.....	23
Confection et vérifications liées à la nouvelle carte .....	26

## Introduction

L'objectif de cette SAE est la création d'un prototype d'enceinte Bluetooth composé d'un amplificateur de classe D. Ce type d'amplificateur fonctionne à l'aide de transistor en position saturé ou bloqué. Cela permet d'éliminer au maximum les pertes et d'avoisiné (en théorie) un rendement de 100%.

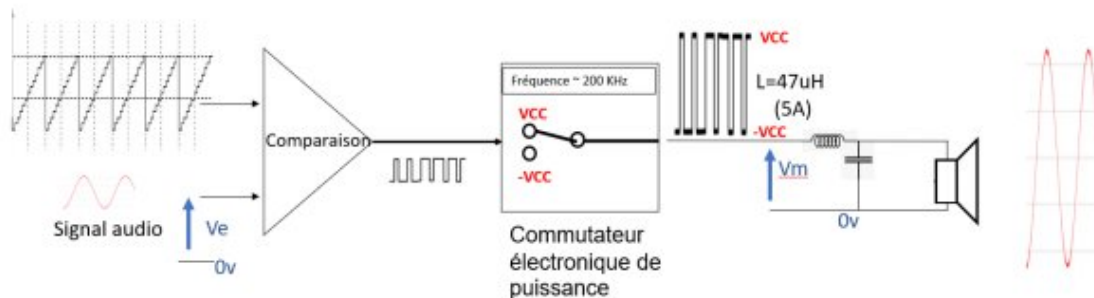


Figure 1 : Schéma du système de l'enceinte

Le signal audio reçu sera transformé en MLI. Cette MLI permettra de contrôler un commutateur électronique de puissance. Ce commutateur permet d'augmenter la tension de la MLI en passant de 3.3-0V à 15- -15V. Ensuite, un filtre passe-bas est afin de restituer le signal audio de ligne mais amplifié.

Pour réaliser cette SAE nous avons créé des équipes de 3 avec, pour chaque membre, un tâche attribué détaillé dans le Diagramme d'exigence. Notre équipe est constitué de Tony Galgani, qui réalisera la tâche 1, Maxime Schmitt, qui s'occupera de la tâche 2 et Mohammed Arbid, à qui la tâche 3 a été confié.

## Diagramme de cas d'utilisation

Ce diagramme permet de représenter l'utilisations du produit en définissant ses relations avec les éléments extérieurs.

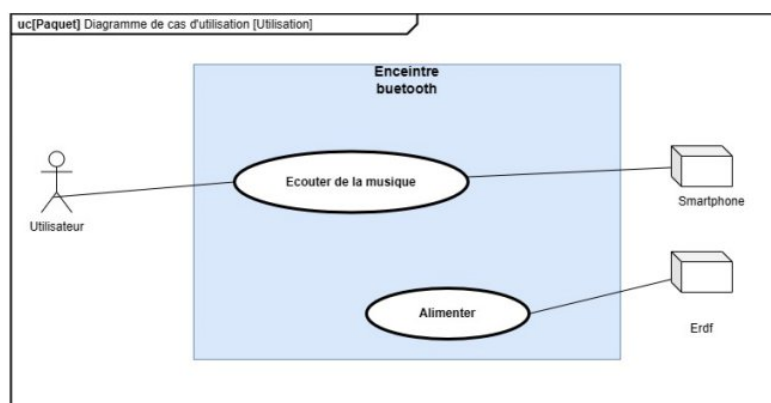


Figure 2 : Diagramme de cas d'utilisations

## Diagramme d'exigences

Ce diagramme permet de définir les nécessités, les contraintes et les capacités du systèmes. Dans notre cas, il permet également des définir chaque mission dans les tâches attribuées. Jaune pour la tâche 1, violet pour la tâche 2 et bleu pour la tâche 3.

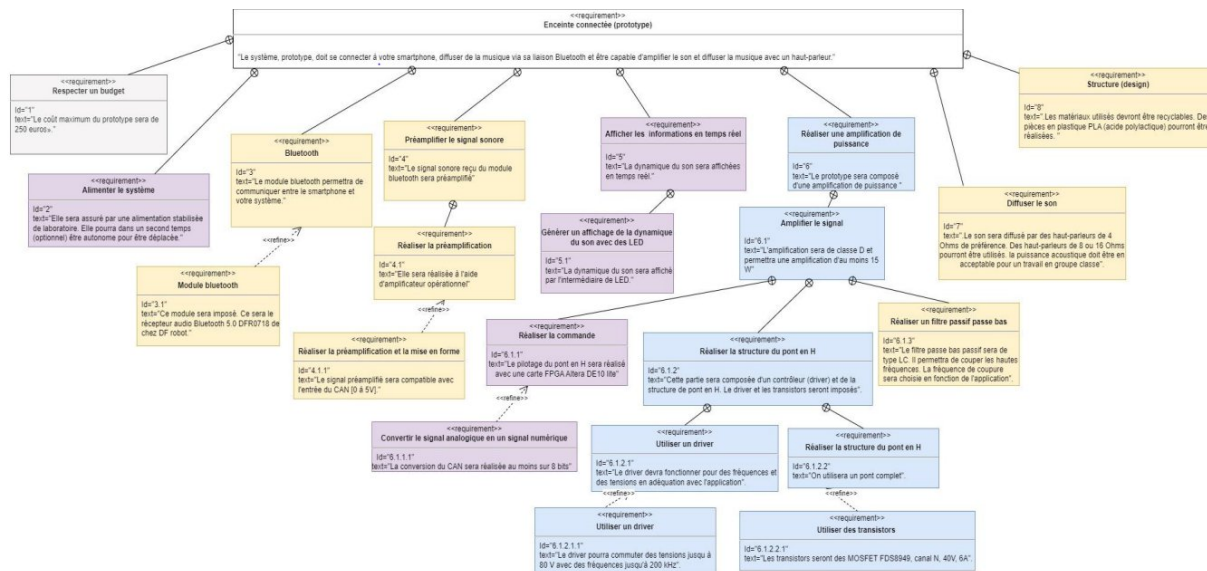


Figure 3 : Diagramme d'exigences

## Diagramme de définition de bloc

Ce diagramme décrit le système via des sous ensemble ainsi que les relations les liants.

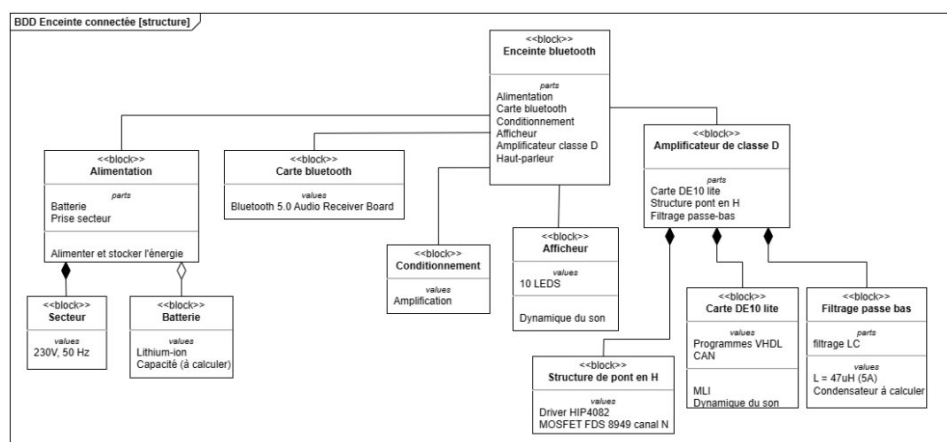


Figure 4 : Diagramme de définition de bloc

Voici un aperçu général des connexions entre les différents composants qui seront utilisés dans notre système :

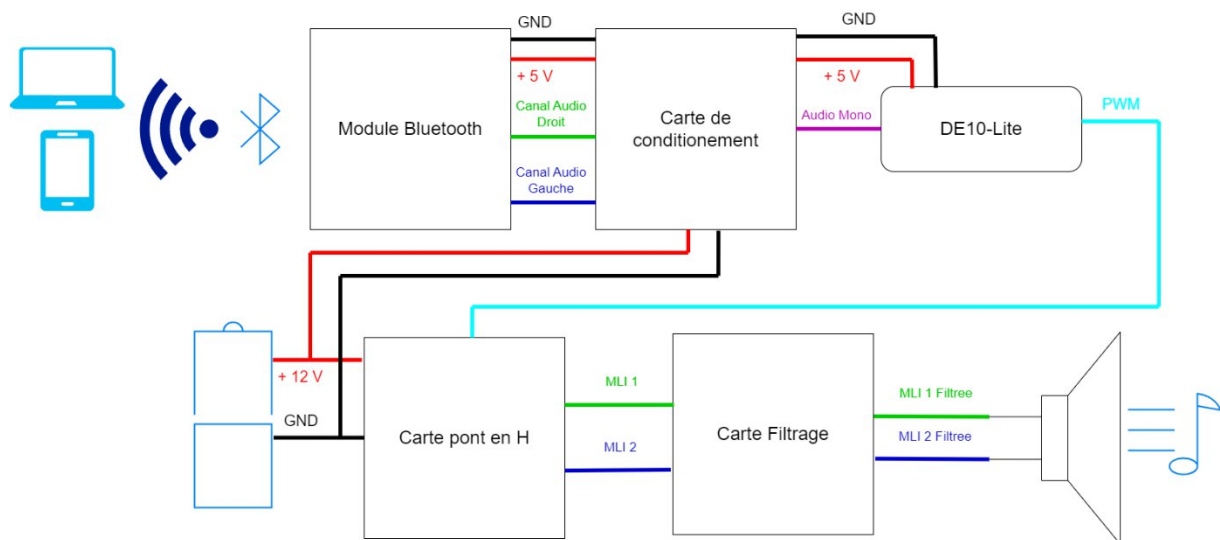


Figure 5: Diagramme de connexions du système

## Tâche 1 : Conditionnement et Filtrage du signal

### Module Bluetooth : Gestion et Transmission du Signal Audio

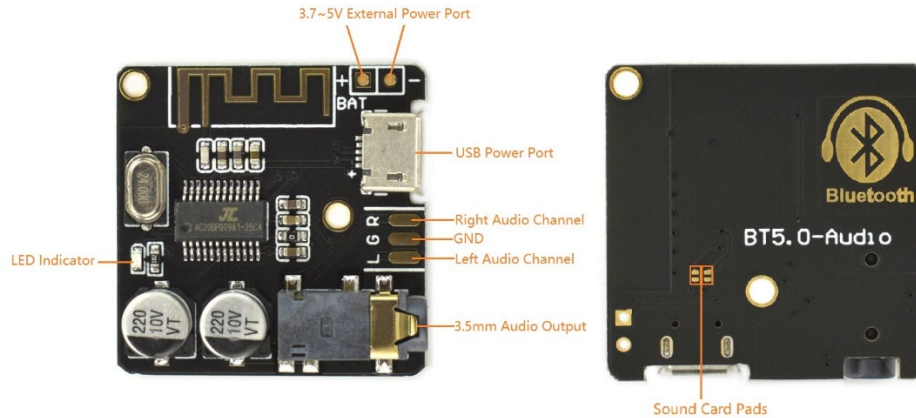


Figure 6: Module Bluetooth

Ce module est responsable de la gestion de toutes les actions liées au Bluetooth, éliminant ainsi la nécessité de développer des solutions spécifiques pour cette technologie. Une fois alimenté, il permet une connexion Bluetooth avec divers appareils tels que smartphones, ordinateurs ou tout autre dispositif équipé d'une sortie audio Bluetooth.

### Processus de Connexion et Transmission du Signal

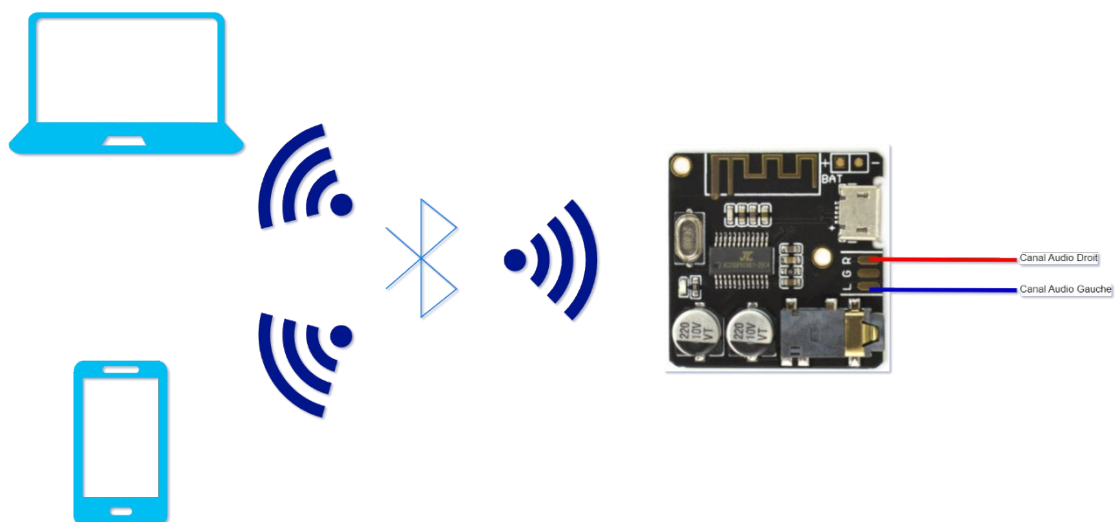


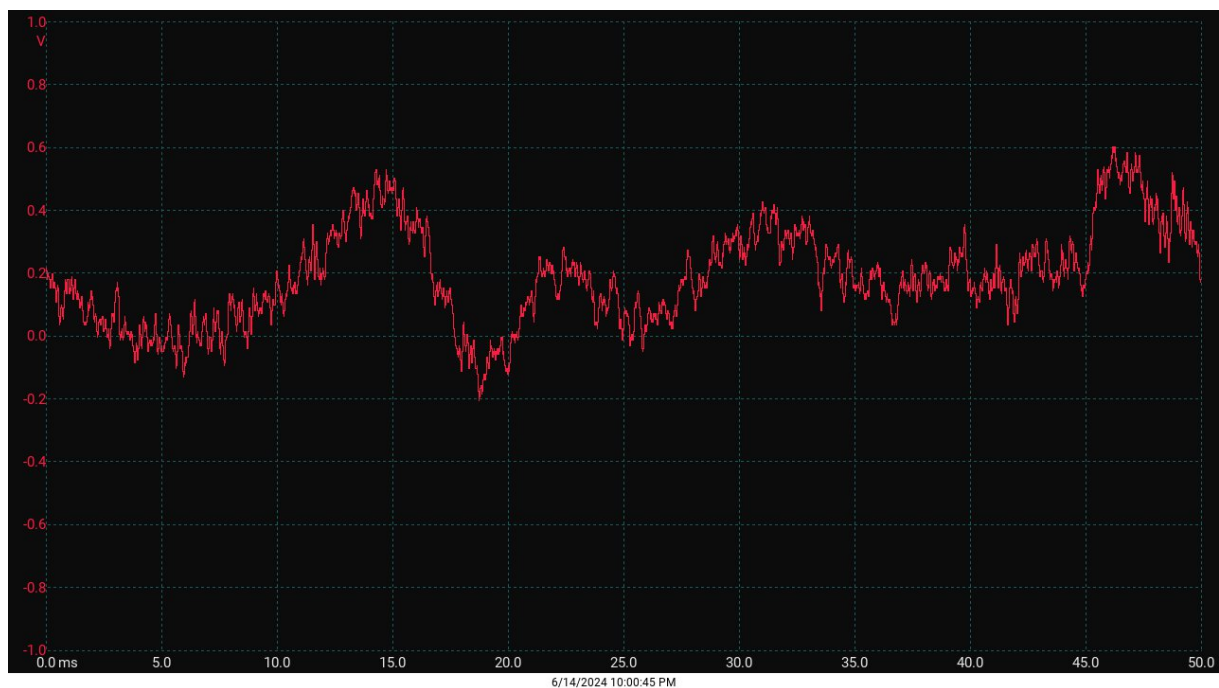
Figure 7: Synopsis de la transmission de signal du module BT5-audio

Le processus de transmission est simple : après alimentation du module, il suffit de le connecter via Bluetooth à l'appareil souhaité. Le signal audio de notre choix peut alors être envoyé au module. Ce dernier convertira le signal reçu en un signal de tension analogique, disponible sur ses sorties « Right Audio Channel » et « Left Audio Channel ».

## Étape 1 : Conditionnement du Signal Audio Stéréo

La première étape de la tâche n°1 consiste à concevoir et réaliser une carte électronique dédiée au conditionnement du signal audio stéréo. Cette carte a pour objectif d'assurer une qualité optimale du signal en le préparant pour les traitements ultérieurs.

La première étape a été de définir l'aspect du signal fourni par le module Bluetooth. Pour cela, nous l'avons visualisé à l'aide d'un oscilloscope.



*Figure 8: Visualisation du signal audio stéréo fournit par le module Bluetooth*

### Observations Initiales

Sur cette visualisation, le signal audio du canal droit est présenté (le signal du canal gauche étant similaire, cette visualisation est également applicable à ce dernier). Les valeurs analogiques observées varient d'environ -0.4 Volt à 0.8 Volt.

## Problèmes Identifiés

À partir de ces observations, deux problèmes principaux peuvent être interprétés :

- Présence de Valeurs de Tension Négatives : Le signal présente des valeurs de tension négatives, ce qui n'est pas compatible avec les exigences de notre système de traitement.
- Amplitude Crête à Crête Insuffisante : L'amplitude du signal, allant de -0.4 Volt à 0.8 Volt, n'est pas suffisamment élevée pour être adéquatement traitée.

## Besoin de Conditionnement du Signal

Pour que le signal audio soit correctement échantillonné par la carte de traitement de la tâche n°2, DE10-Lite, il doit être adapté. Le convertisseur analogique-numérique (CAN) de cette carte permet l'échantillonnage de valeurs de 0 Volt à 5 Volts. Ainsi, un conditionnement préalable du signal est indispensable pour assurer sa compatibilité avec les spécifications du CAN.

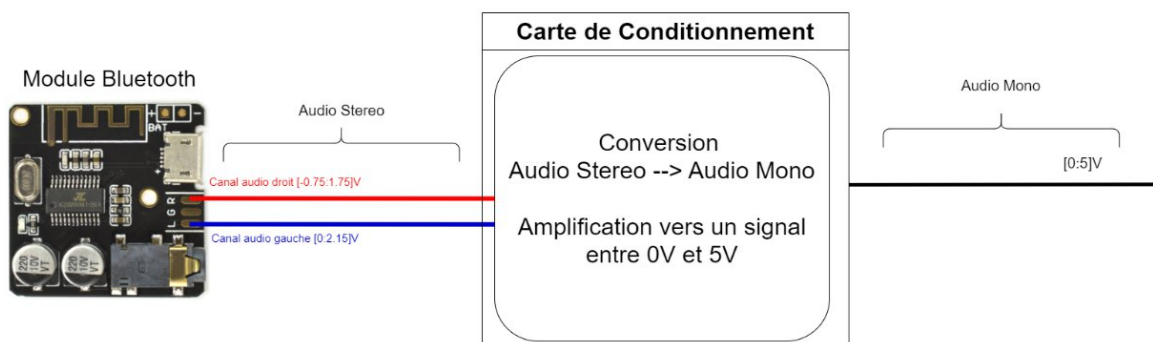


Figure 9: Synopsis entrées - sorties carte de conditionnement

## Conversion du signal stéréo en mono

La première partie du circuit a pour objectif de convertir le signal des deux canaux audio stéréo en un seul signal contenant les données de ces deux canaux. De plus, il est nécessaire d'ajouter un décalage en tension, car actuellement les signaux sont centrés sur 0V. Pour obtenir une bande de tension allant de 0V à 5V, un décalage de 2.5V est optimal pour recentrer le signal au milieu de cette bande de tension.



Pour réaliser cette transformation et décalage, les étapes suivantes sont suivies :

#### *Combinaison des Signaux Stéréo :*

Connexion des deux pistes après des résistances de même calibre. Cette configuration permet de fabriquer un signal mono équivalent à la moyenne des deux signaux, divisée par deux.

#### *Ajout du Décalage de Tension :*

Ce décalage est appliqué après une résistance de même calibre que celles utilisées précédemment. La tension de décalage est créée à partir d'un pont diviseur de tension, constitué de deux résistances de  $1\text{k}\Omega$  chacune, avec une tension d'alimentation de  $5\text{V}$ .

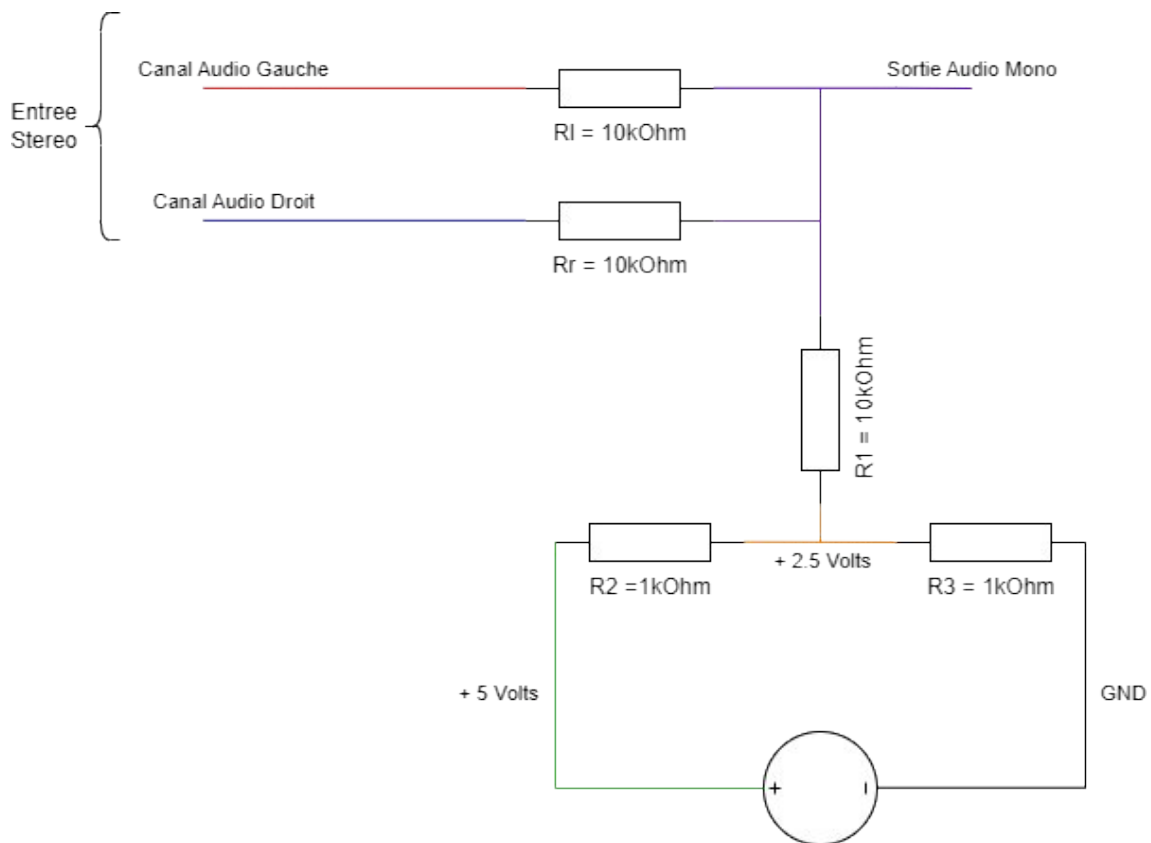


Figure 10: Schéma représentant la transformation stéréo->mono du signal

## Tests pratiques et observations

Après la réalisation du montage sur une plateforme de test, les résultats suivants ont été observés :

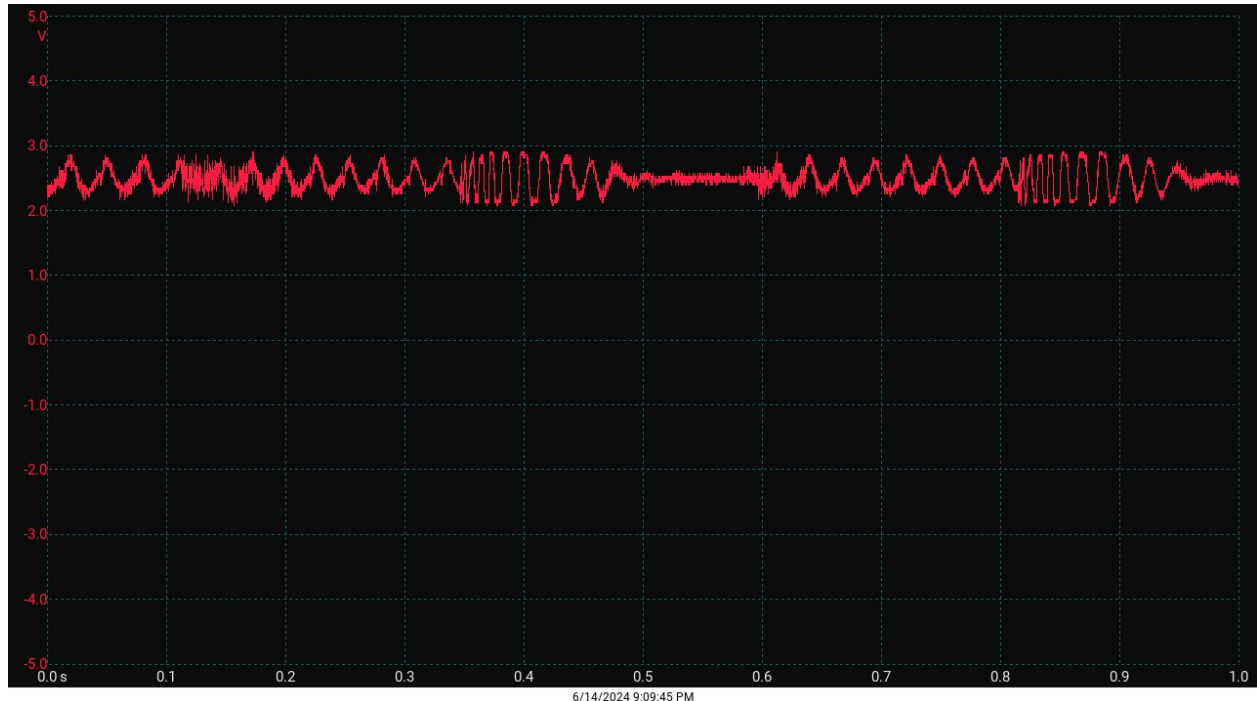


Figure 11: Visualisation du signal audio mono

Les résultats de ce montage sont optimaux. En effet, il produit un seul signal contenant les données des deux canaux audios provenant du module Bluetooth, centré autour de 2,5 V. Bien que ce signal soit exploitable, son amplitude crête à crête d'environ 0,9 V (avec un minimum de 2,1 V et un maximum de 3,0 V) est trop faible pour permettre un échantillonnage du signal précis.

## Amplification du signal

Pour améliorer l'exploitabilité du signal, il est nécessaire de procéder à son amplification. Cela permet d'augmenter l'amplitude crête à crête du signal, facilitant ainsi son échantillonnage.

Pour cette amplification, un amplificateur opérationnel (AOP) sera utilisé. Le modèle choisi est le TL081, sélectionné pour sa fiabilité et notre familiarité avec ses caractéristiques.

Le montage suivant présente le schéma pour cette amplification :

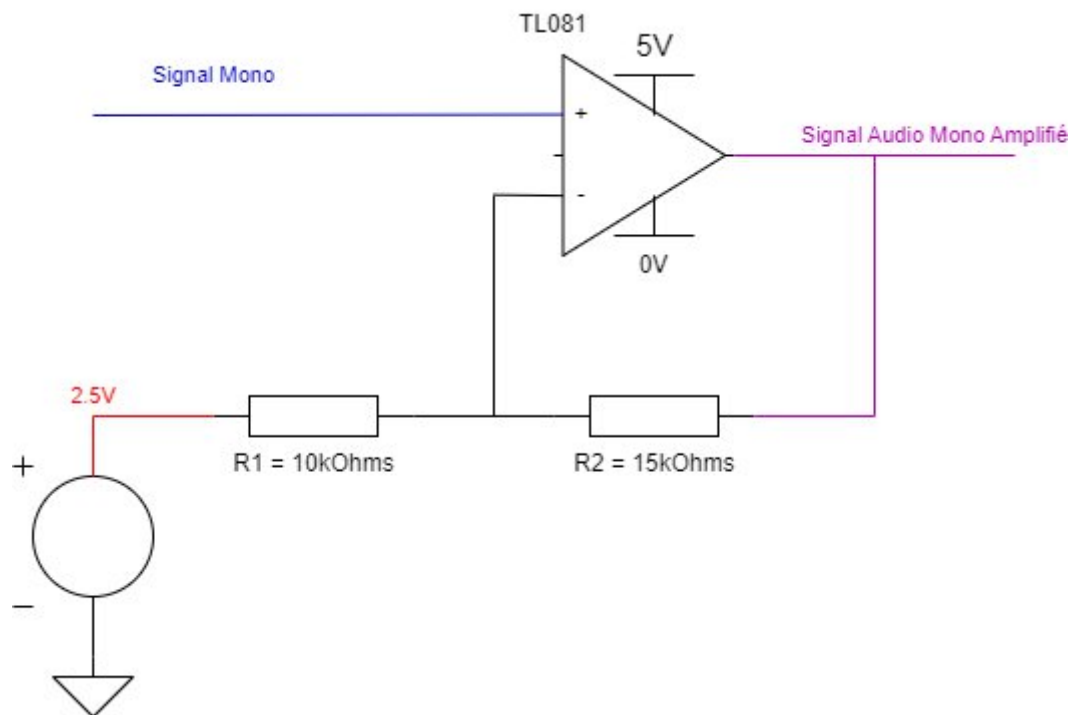


Figure 12: Schéma du montage d'amplification du signal

### *Amplificateur Non-Inverseur*

Le montage présenté est celui d'un amplificateur non-inverseur, dont le gain est calculable par la formule suivante :

$$G = 1 + \frac{R2}{R1}$$

### *Calcul du Gain*

En utilisant les valeurs de résistances suivantes :

$$R1 = 10k\Omega$$

$$R2 = 15k\Omega$$

Le gain obtenu est :

$$G = 1 + \frac{15k}{10k} = 2.5$$

### *Domaine de Contre-Réaction*

Un point crucial de ce montage est son domaine de contre-réaction. Traditionnellement, un amplificateur opérationnel (AOP) fonctionne dans un domaine centré sur zéro, avec une alimentation positive de +5V et une alimentation négative de -5V, et les résistances de contre-réaction branchées sur la terre (ground).

### *Adaptation de l'Alimentation pour Éviter l'Amplification de l'Offset*

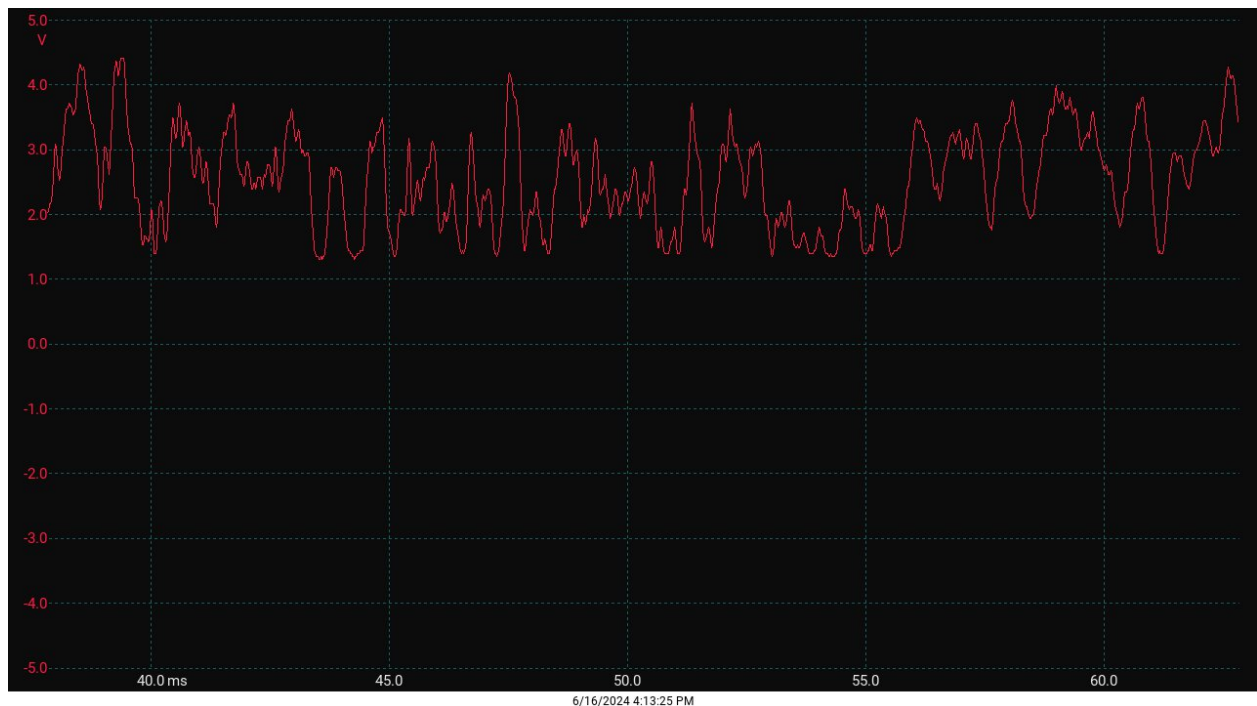
Dans notre application, utiliser l'AOP de cette manière entraînerait l'amplification non seulement du signal audio, mais aussi de l'offset de 2.5V, ce qui n'est pas souhaitable. Cet offset est nécessaire pour recentrer le signal et ne doit pas être amplifié.

### *Configuration de l'Alimentation*

Pour éviter l'amplification de l'offset, notre montage utilise une tension d'alimentation positive de +5V et une tension d'alimentation négative de 0V. De plus, la contre-réaction est branchée sur une tension de +2.5V. Ainsi, tout comme nous avons recentré le signal sur 2.5V, nous recentrons également l'amplification sur ce même potentiel.

### *Tests pratiques et observations*

Après rajout du circuit d'amplification au montage existant, les résultats suivants ont été observés :



*Figure 13: Visualisation du signal amplifié*

Sur cette visualisation, on peut observer une nette augmentation de l'amplitude du signal audio. Désormais, les tensions crêtes du signal sont d'environ 1.5 Volts pour le minimum et 4.5 Volts pour le maximum, ce qui correspond à une tension crête à crête de 3 Volts.

### *Optimisation Potentielle*

Bien que cette configuration soit déjà très performante, une optimisation supplémentaire pourrait être envisagée : en décalant le signal vers une tension légèrement moins élevée et en augmentant par la même occasion un peu le gain, on pourrai permettre un échantillonnage encore plus performant.

Malgré cela, la configuration actuelle reste très efficace. Le signal audio amplifié permet un échantillonnage et une restitution presque parfaits, assurant ainsi une haute qualité de traitement pour les étapes suivante.

## Remarque : Alimentation du Système

Le système est alimenté par une batterie de +12V. Pour obtenir la tension de +5V nécessaire à certaines parties du circuit, un régulateur de tension sera intégré.

### *Utilisation du Régulateur de Tension LM7805BT*

Le régulateur de tension 7805BT sera utilisé pour convertir la tension de +12V de la batterie en une tension stable de +5V.

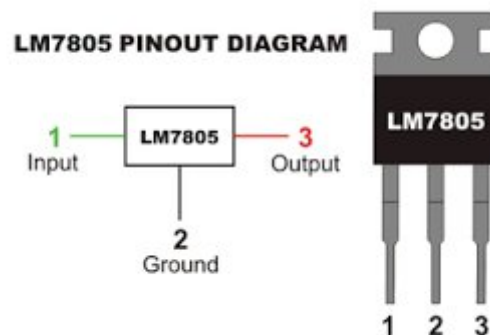


Figure 14: Régulateur LM7805BT Pin Out

## Étape 2 : Filtrage du Signal

La seconde étape, bien plus courte, implique la réalisation d'une deuxième carte électronique. Cette carte est spécifiquement conçue pour filtrer le signal audio stéréo à la fin du système, après qu'il a subi de nombreuses modifications. Ce filtrage est utile pour garantir la pureté et la clarté du signal final.

Le filtre utilisé est un filtre LC, soit incluant une inductance et un condensateur. On choisit une fréquence de coupure de 20kHz pour éliminer tout parasite n'étant pas audible.

Ainsi, avec la valeur de l'inductance imposée à  $47\mu\text{H}$ , nous pouvons déduire la valeur du condensateur par la relation suivante :

$$f_c = \frac{1}{2\pi\sqrt{LC}} \Leftrightarrow C = \frac{\left(\frac{1}{2\pi f_c}\right)^2}{L} = 1.35\mu\text{F}$$

N'ayant pas de condensateur de  $1.35\mu\text{F}$ , on prendra  $C = 1\mu\text{F}$  ce qui nous donne :

$$f_c = 23\text{kHz}$$

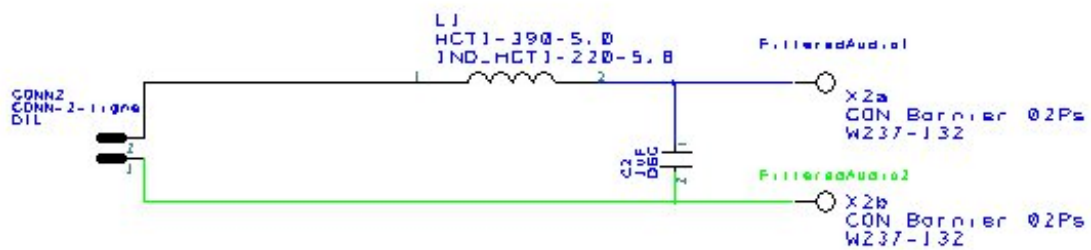


Figure 15: Schéma du Filtre passe-bas

## Tâche 2 : Traitement numérique

### Convertisseur analogique numérique (CAN)

Afin de réaliser la commande en PWM depuis un signal analogique, il va être nécessaire de transformer le signal d'entrée en signal numérique. Pour cela, nous utilisons un Convertisseur Analogique Numérique (CAN) présent dans la carte.

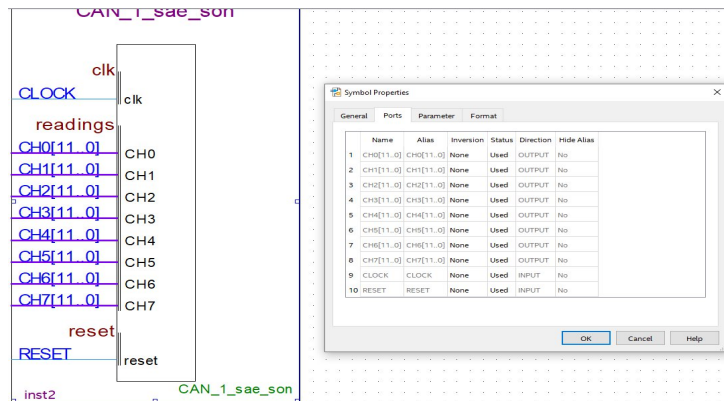


Figure 16 : Bloc CAN

Ce bloc est créé via l'outil de création de bloc IP présent dans le logiciel. Il présente, en entrée, une horloge ainsi qu'un reset, le Reset ne sera pas utilisé. Il permet de convertir jusqu'à 8 signaux. Ce convertisseur permet de transformer le signal analogique en signal numérique codé sur 12 bits et son horloge sera relié à l'horloge interne de 50M Hz de la carte.

### Comparateur et diviseur d'horloge

Le composant qui devra alimenter notre carte via la PWM fonctionne avec un signal pouvant aller jusqu'à 200KHz, nous allons donc diviser notre horloge afin de créer la période de la PWM. De plus, une musique est comprise entre 20 et 20kHz. Le théorème de Shannon est donc respecté car nous avons une fréquence d'échantillonnage plus de 2 fois supérieure à la fréquence du signal. À noter qu'en pratique, nous prenons une fréquence d'échantillonnage 10 fois supérieure.

Théorème de Shannon :

$$F_{\text{signal}} < 2 * F_{\text{échantillonnage}}$$

Calcul du nombre de bits nécessaire :

$$50\,000\,000 = 2^n * 200\,000$$

$$n = \log_2 \frac{50\,000\,000}{200\,000}$$

$$n = 7.965 \text{ bits}$$



Nous pouvons donc prendre un signal numérique de 8 bits car cette valeur répond aux différentes contraintes du système.

Afin de diviser cette horloge et réaliser le compteur, nous ajoutons un compteur d'une valeur de 255.

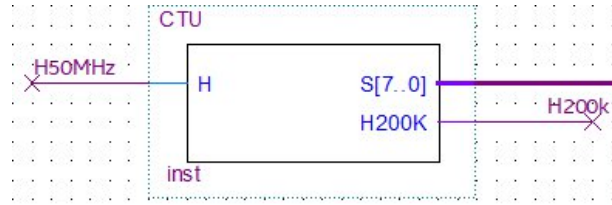


Figure 17 : Horloge et Compteur

Ce bloque est réalisé via une programmation VHDL :

```

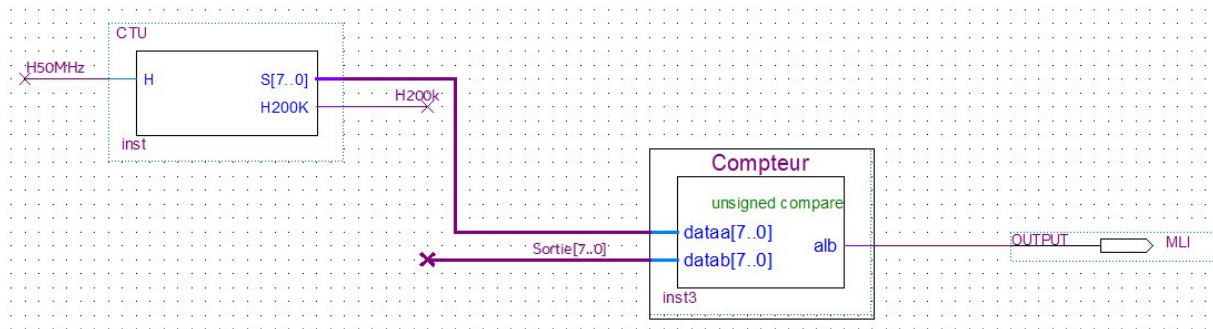
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4  ENTITY CTU IS
5  PORT (
6      H : IN STD_LOGIC;
7      S : BUFFER INTEGER range 0 to 255;
8      H200K : OUT STD_LOGIC
9  );
10 END CTU;
11 ARCHITECTURE ARCHI OF CTU IS
12 BEGIN
13     PROCESS (H)
14     BEGIN
15         IF (H'EVENT and H='1') THEN --Front montant d'horloge
16             IF (S=255) THEN
17                 S<=0; --RAZ du compteur
18                 H200K<='1'; --Horloge 200k à 1
19             else
20                 S<=S+1; --Incrémentation du compteur
21                 H200K<='0'; --Horloge 200k à 0
22             END IF;
23         END IF;
24     END PROCESS;
25 END ARCHI;
26

```

Figure 18 : Programme compteur

Dans ce programme, nous réalisons un compteur qui s'incrémente à chaque front montant de notre horloge de référence (50M Hz) jusqu'à une valeur de 255 que nous remettons à 0 lorsque celle-ci est dépassée. Cette valeur est présente en permanence sur la sortie S de notre bloc. De plus nous réalisons un diviseur d'horloge pour créer une horloge de 200kHz

Cette sortie est ensuite comparée avec la sortie de notre CAN afin de créer une PWM qui dépendra de notre signal d'entrée analogique.



De plus, nous adaptons le signal de 12 bits en sortie de CAN à un signal 8 bits. Nous gardons les bits de poids fort afin de conserver les informations les plus importantes. La comparaison réalisée est la suivante :

$alb = 1$  Si  $Data A < Data B$

Cela nous crée une PMW avec une valeur haute en début de période. Ce résultat est ensuite envoyé dans une sortie de la carte afin de contrôler le pont en H.

Nous avons ensuite réalisé une simulation afin de vérifier le bon fonctionnement du programme.

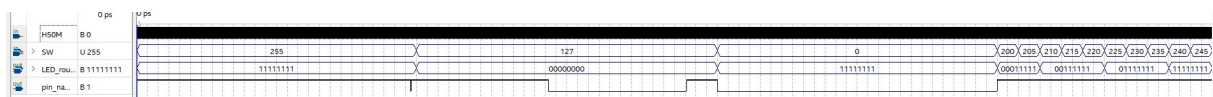


Figure 20 : simulation de la PMW

La simulation est donc bonne, la valeur de la PMW est à 1 lorsque notre valeur de data b est à 255, et la période lorsque notre valeur est à 127 est équitablement partagé en 2.

## Vizualiseur

Nous avons ensuite réalisé l’affichage dynamique sur les leds. Pour cela, nous avons créé un bloc programmé en VHD.

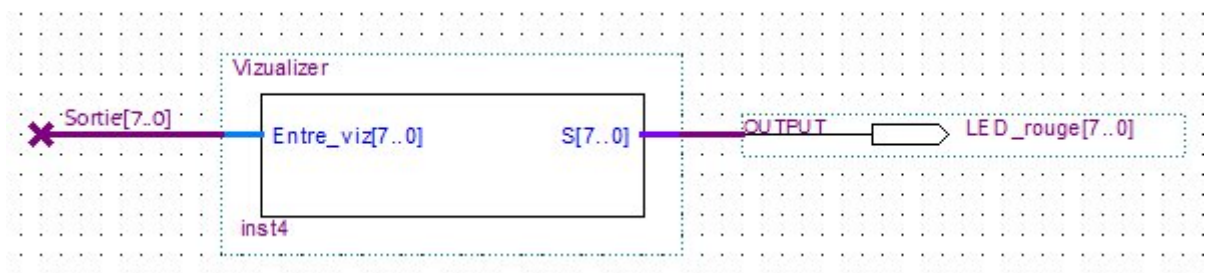


Figure 21 : Liaison bloc Visualiseur

Etant donné que notre valeur « 0 » équivaut à une valeur de (127 sur 8 bits) sur notre CAN, il est nécessaire de « redresser » le signal pour que la valeur « 0 » soit égale à la valeur « 255 » sur notre visualiseur.

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4  ENTITY vizualizer is
5  PORT(
6      Entre_viz : IN INTEGER range 0 to 255;
7      s : out std_logic_vector (7 downto 0));
8  END vizualizer;
9  ARCHITECTURE archi OF vizualizer IS
10     signal tempo : INTEGER range 0 to 255;
11     BEGIN
12     PROCESS (Entre_viz)
13     BEGIN
14         IF (Entre_viz>=127) THEN
15             tempo<=Entre_viz;
16         Else -- Si Entre_viz<127 Redressage |
17             tempo<=255-Entre_viz;
18         End if;
19         --IF (tempo> 127) and (tempo<143) THEN S<="00000001";end if;
20         --IF (tempo>=143) and (tempo<159) THEN S<="00000011";end if;
21         --IF (tempo>=159) and (tempo<175) THEN S<="00000111";end if;
22         --IF (tempo>=175) and (tempo<191) THEN S<="00001111";end if;
23         --IF (tempo>=191) and (tempo<207) THEN S<="00011111";end if;
24         --IF (tempo>=207) and (tempo<223) THEN S<="00111111";end if;
25         --IF (tempo>=223) and (tempo<239) THEN S<="01111111";end if;
26         --IF (tempo>=239) and (tempo<=255) THEN S<="11111111";end if;
27
28         IF (tempo> 127) and (tempo<131) THEN S<="00000001";end if;
29         IF (tempo>=131) and (tempo<135) THEN S<="00000011";end if;
30         IF (tempo>=135) and (tempo<139) THEN S<="00000111";end if;
31         IF (tempo>=139) and (tempo<143) THEN S<="00001111";end if;
32         IF (tempo>=143) and (tempo<147) THEN S<="00011111";end if;
33         IF (tempo>=147) and (tempo<151) THEN S<="00111111";end if;
34         IF (tempo>=151) and (tempo<155) THEN S<="01111111";end if;
35         IF (tempo>=155) and (tempo<=159) THEN S<="11111111";end if;
36
37
38
39         IF (tempo=127) THEN S<="00000000";end if;
40     End process;
41 end archi;

```

Figure 22 : Programme bloc Visualiseur

Ce bloc permet donc de réaliser ce redressement et retourne en sortie un mot en 8 bits permettant d'allumer les leds choisi sur le pin planner. Nous avons choisi de réaliser l'affichage sur 8 leds, nous avons donc divisé 127 par 8 afin de connaître le pas entre chaque leds.

$$\frac{127}{8} = 15.875$$

Mais étant donné que notre signal d'entrée de CAN varie entre 2.9 et 2.1V, nous n'atteignons jamais ces valeurs, nous avons donc modifié les valeurs des leds afin d'avoir l'affichage de l'amplitude sur toutes les leds.

Nous avons ensuite testé cette partie du programme via une simulation.

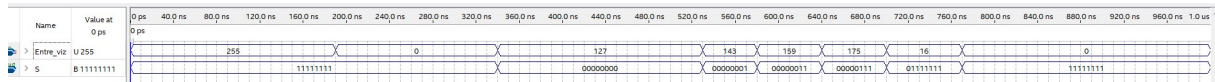


Figure 23 : Simulation Visualiseur

Nous pouvons constater le bon fonctionnement du redresseur et de l'affichage.

### Filtre passe-bas

En supplément, nous avons réalisé un filtre passe-bas. Ce filtre a pour but de modifier la musique. Afin d'avoir un changement facilement audible, nous avons choisi une fréquence de coupure de 1kHz

Afin de réaliser ce filtre numérique, nous avons utilisé la formule suivante :

$$S(t) = a * S(t - 1) + (b - a) * E(t - 1)$$

Cette formule crée un retard, mais ce retard n'est pas perceptible à l'oreille.

Afin de trouver les coefficients inconnus a et b, nous avons utilisé un site ([Biquad calculator v3 | EarLevel Engineering](#)) en insérant nos fréquences.

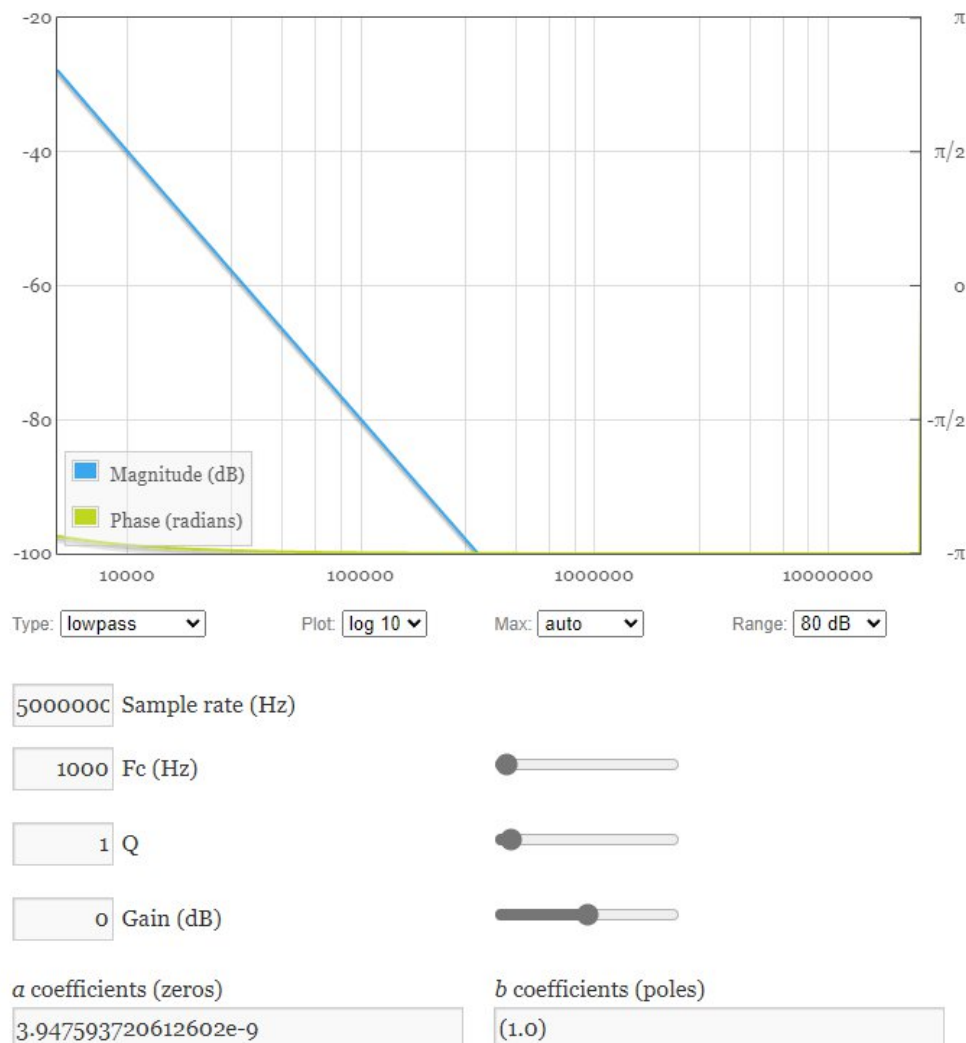


Figure 24 : coefficient et fréquence du passe\_bas



Une fois ces coefficients obtenus, nous avons calculé le nombre de bits nécessaire afin de programmer le coefficients A. Pour cela, nous multiplions la valeur de A par 2 jusqu'à trouver un résultat proche d'une valeur entière. Dans notre cas, nous avons trouvé ce résultat :

$$3.9476 * 2^{28} = 1.05$$

Nous avons ensuite réalisé le bloc de programmation :

```

1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4  ENTITY passe_bas IS
5  PORT (
6    E : IN STD_LOGIC_VECTOR (7 downto 0);
7    SW : IN STD_LOGIC;
8    H : IN STD_LOGIC;
9    S : BUFFER STD_LOGIC_VECTOR (7 downto 0)
10 );
11 END passe_bas;
12 ARCHITECTURE ARCHI OF passe_bas IS
13 signal tempo13SP : STD_LOGIC_VECTOR (35 downto 0) := "00000000000000000000000000000000";
14 signal tempo13S : STD_LOGIC_VECTOR (64 downto 0);
15 signal tempo13E : STD_LOGIC_VECTOR (35 downto 0) := "00000000000000000000000000000000";
16 signal a : STD_LOGIC_VECTOR (28 downto 0);
17 signal b : STD_LOGIC_VECTOR (28 downto 0);
18 BEGIN
19   PROCESS (H)
20   BEGIN
21     IF (H'EVENT and H='1') THEN
22       IF (SW='1') THEN
23         a<="00000000000000000000000000000001";--Definition
24         b<="10000000000000000000000000000000";
25         tempo13S<=(a*tempo13SP)+((b-a)*tempo13E);--Calcul de la nouvelle sortie
26         tempo13SP<=S & "00000000000000000000000000000000";--Enregistrement des valeurs t-1
27         tempo13E<=E & "00000000000000000000000000000000";
28         S<=tempo13S(64 downto 57);
29       else
30         S<=E;
31         tempo13SP<=S & "00000000000000000000000000000000";--Enregistrement des valeurs t-1
32         tempo13E<=E & "00000000000000000000000000000000";
33       END IF;
34     END IF;
35   END PROCESS;
36 END ARCHI;
37
38

```

Figure 25 : Programmation passe-bas

## Banc de test

Nous avons ensuite réalisé les tests des programmes de l'altera en pratique à l'aide d'un GBF paramétré sur une fréquence de 1kHz ainsi qu'une amplitude de 0 à 5V.



Figure 26 : Fréquence du GBF

Nous avons ensuite relié le GBF ainsi que la sortie de la carte DE10 Lite à un oscilloscope

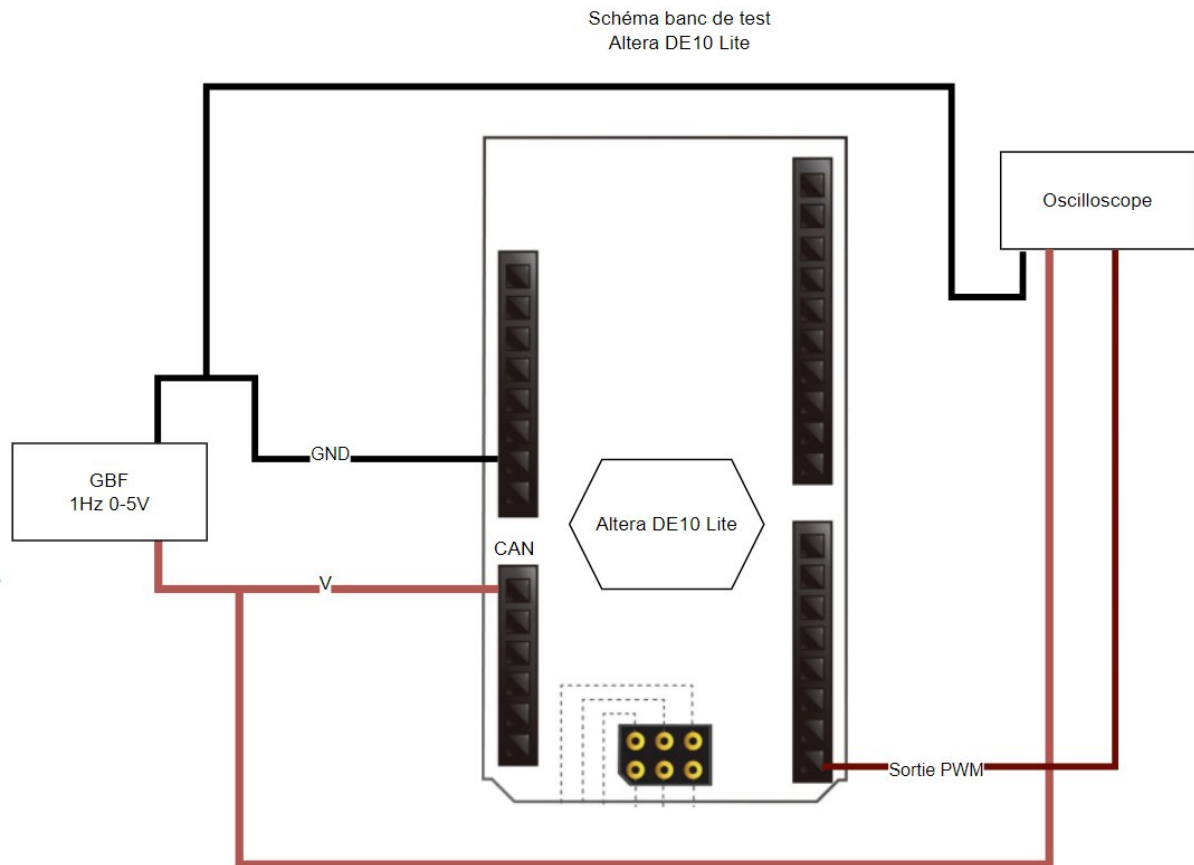


Figure 27 : Schéma banc de test

Ce banc de test nous a permis de tester la PWM créée ainsi que le visualiseur :

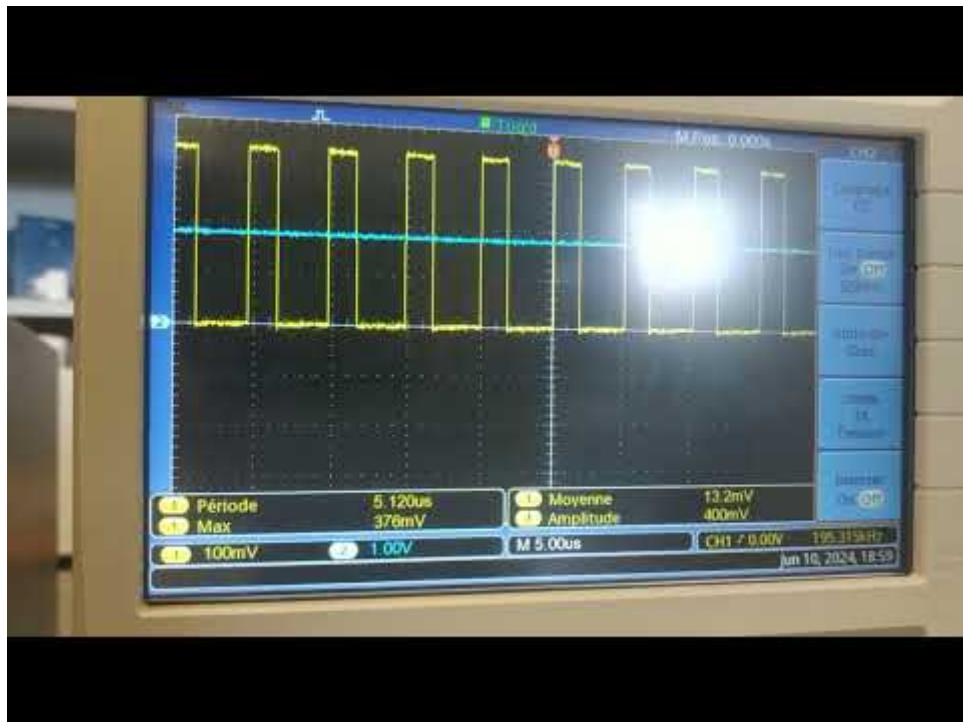


Figure 28: Test de la PWM sur banc de test

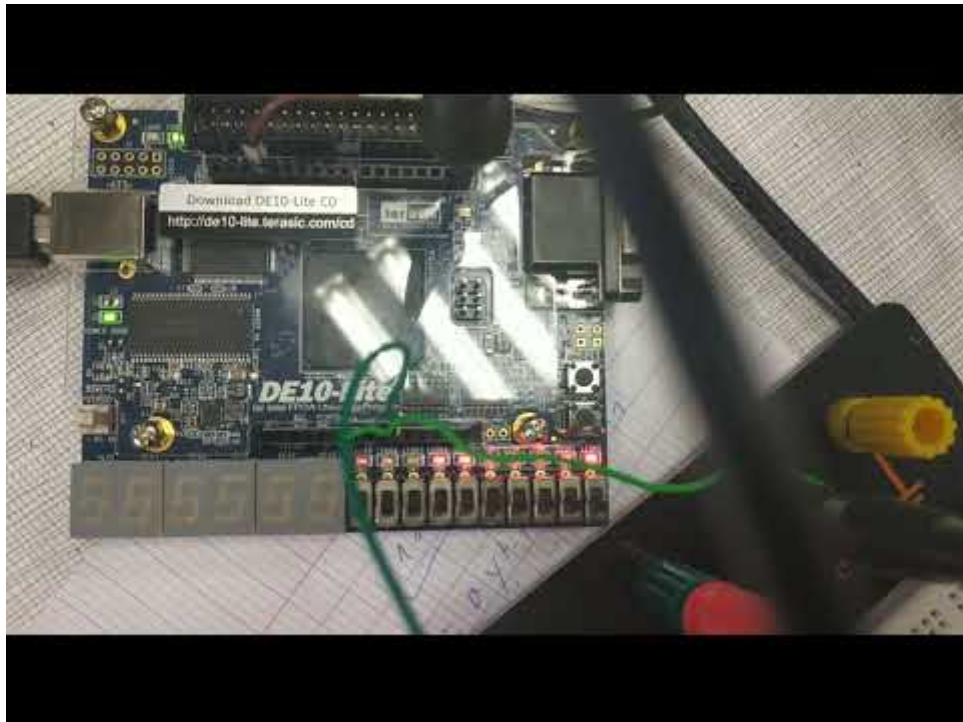


Figure 29 : Test du vizualiseur sur banc de test

### Tâche 3 : Réalisation de la carte de l'amplification de classe D

#### Fonctionnement de la carte d'amplification de classe d préliminaire

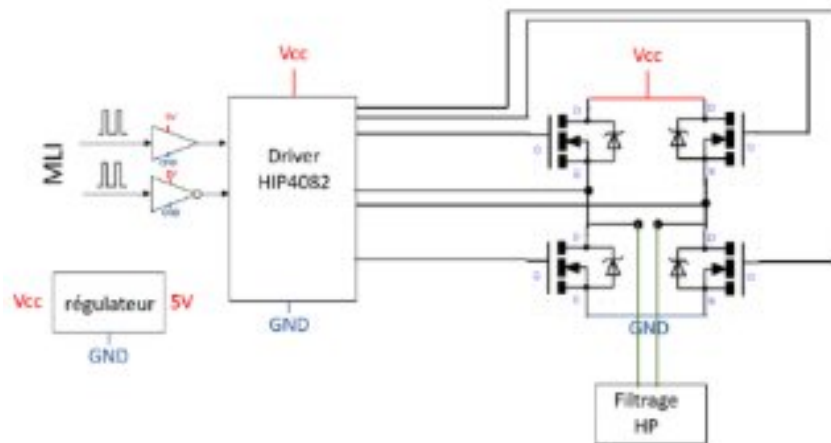


Figure 30 : Schéma du pont en H

Lorsque le signal PWM est envoyé par le microcontrôleur, les transistors du pont en H vont commuter de manière à reproduire fidèlement le signal. Les transistors peuvent être des MOSFET, des IGBT ou des bipolaires en fonction des exigences de puissance et de rendement.

L'avantage de l'amplificateur de classe D est son rendement élevé, qui peut atteindre jusqu'à 90 %, ce qui signifie que peu d'énergie est dissipée sous forme de chaleur. Cela en fait un choix idéal pour les applications où la consommation d'énergie est un enjeu important, comme dans les systèmes audios haut de gamme ou dans les amplificateurs de puissance pour les véhicules.

En résumé, l'amplificateur de classe D est une excellente solution pour amplifier des signaux PWM avec un rendement élevé, ce qui en fait un choix privilégié pour de nombreuses applications. On place deux portes non, la première pour inverser et préamplifier le signal, le deuxième ré inverse le signal (on transforme le signal reçu ,0-3V en -5V +5V, pour ne pas perdre d'information). Pour alimenter ses AOP nous devons ajouter un régulateur de tension qui permet de créer du 5V (la tension d'alimentation des portes) à partir du 15V.



Ces AOP sont connectées à un Driver, le HIP4082. Le Driver est utile car si on utilise les transistors seuls, ils déformeraient le signal (les transistors agissent comme un condensateur qui se charge et qui se décharge), ce composant agit comme un amplificateur de courant et permet d'avoir le signal voulu. Ce driver est composé de deux circuits, un circuit en haut et un circuit bas, qui sont eux-mêmes composés de transistors. Dans le circuit bas le montage est classique : un transistor MOSFET NPN et un transistor PNP, ainsi lorsque  $V_{gs} = V_{cc}$  la tension de sortie est nulle, et inversement. Le montage du haut est plus complexe, il est composé de 2 transistors MOSFET N. Ainsi lorsqu'il y a une tension  $V_{gs}$ , à la sortie on a une tension de  $V_{cc} + 5V$ , ce n'est pas ce que l'on veut. Pour supprimer ce problème on ajoute un condensateur et une diode (voir schéma). C'est un montage Bootstrap. D'après la datasheet du composant on se doit d'utiliser un condensateur de 1000pF (soit 1nF).

Ce driver va commander le Pont en H qui sert à amplifier. Ce pont en H est composé de transistors MOSFET cms (FDS8949), ils permettent de faire passer le courant dans les deux sens, sans trop de perte.

Pour que le pont en H fonctionne parfaitement et sans discontinuité, on va devoir réaliser un décalage temporel, c'est-à-dire laisser un temps très court à 0V au moment du changement de signe de la PWM. Pour ce faire on utilise une résistance de 3KOhms, que l'on a choisie en fonction du graphique proposé sur la datasheet du driver. Cette résistance permet un décalage temporel de X s. Pour finir, on place des condensateurs de découplage sur l'adaptateur (1uF) et de puissances sur la piste d'alimentation (47uF). Puis un connecteur CIL 6 points pour l'alimentation, le GND, l'arrivée du signal MLI et les sorties qui iront sur le filtre HP.

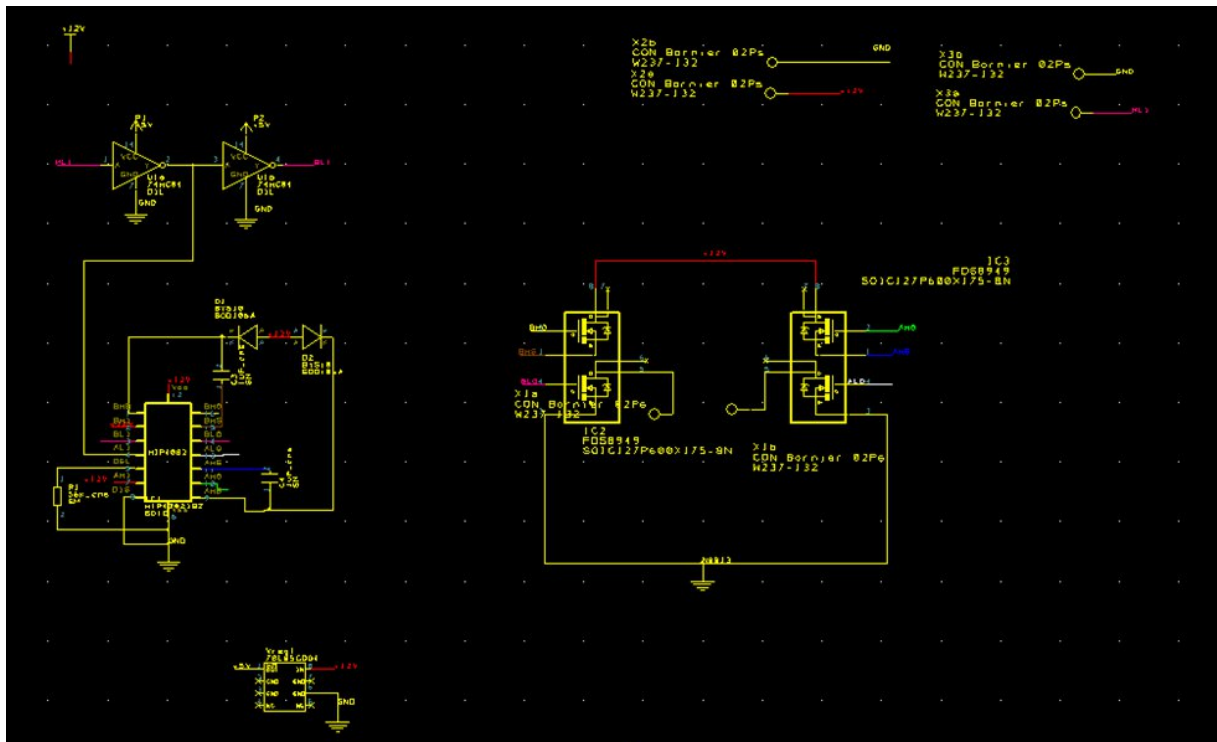


Figure 31 : Schematic de la carte préliminaire d'amplification de classe d

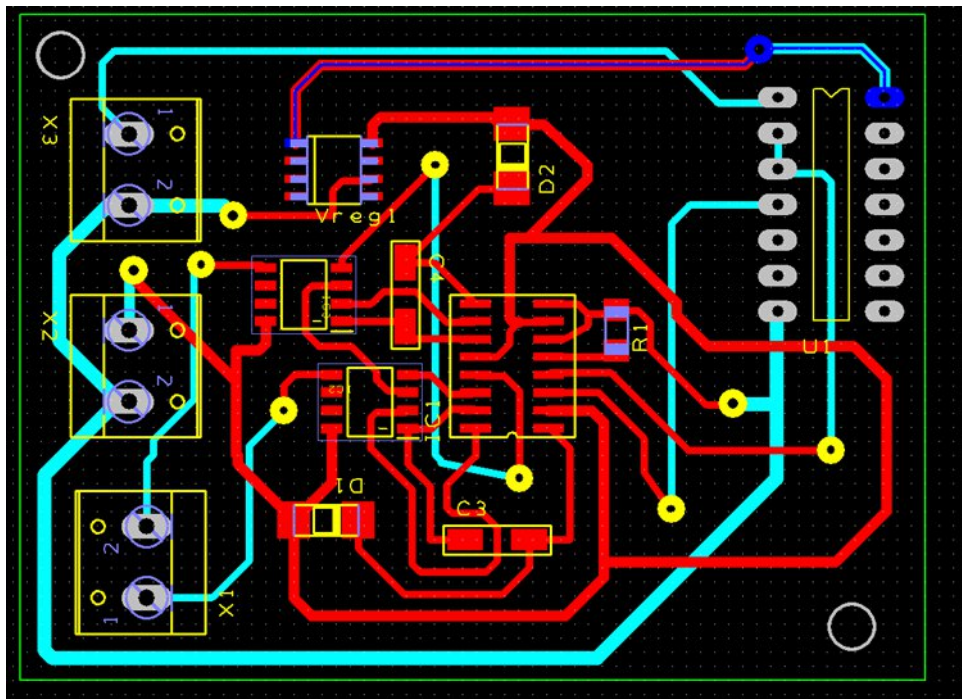


Figure 32 : PCB de la carte préliminaire d'amplification de classe d

## Confection et vérifications liées à la nouvelle carte

Il est important de s'assurer que tous les composants sont correctement positionnés sur la carte et bien alignés. Ensuite, on applique la pâte à souder sur les pattes des composants et on place la carte dans le four à souder. Une fois que le four est bien chaud, on enfourne la carte et on laisse chauffer pendant quelques minutes, le temps nécessaire pour que la pâte fonde et soude les composants. Il est important de surveiller la température pour éviter de brûler les composants. Une fois le processus de soudure terminé, on laisse la carte refroidir avant de tester son bon fonctionnement. Pour cela, on peut utiliser un multimètre pour vérifier la continuité des circuits et s'assurer qu'il n'y a pas de court-circuit. Il est également recommandé de vérifier visuellement que toutes les soudures sont bien réalisées et qu'il n'y a pas de défauts apparents. Si tout est en ordre, on peut alors considérer que la carte est prête à être utilisée dans notre projet. Un schématique nous a été donné, grâce à lui on saura où placer les composants sur notre carte.

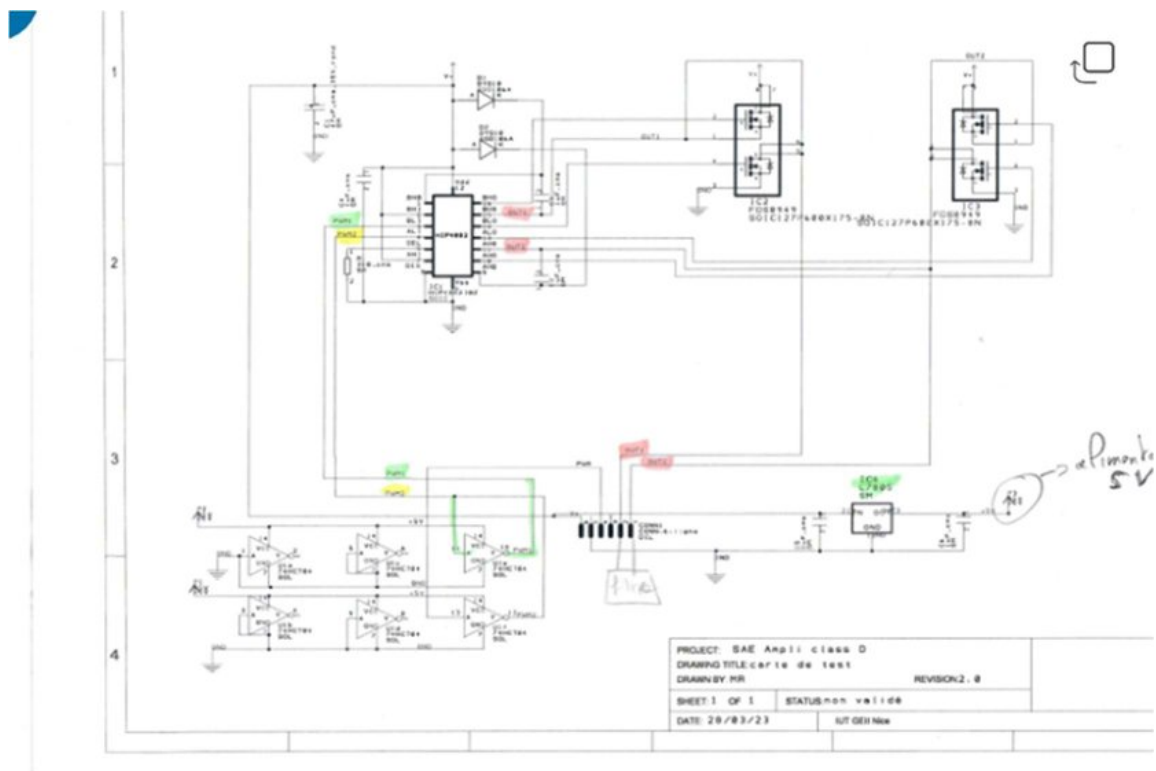


Figure 33 : Schematic de la carte d'amplification de classe d finale



Figure 34 : Test de continuité



Figure 35 : Test de tension

Ensuite, nous avons vérifié les tensions de V+ en alimentant la carte. Nous avons constaté une tension de +15V entre l'alimentation et le GND. Ensuite, nous avons mesuré la tension de sortie du régulateur pour m'assurer qu'il délivre bien du 5V.

Ensuite, nous avons examiné les signaux de sortie en envoyant un signal carré de 0 à 3V (représentant une modulation de largeur d'impulsion) sur la sortie PWM à l'aide d'un générateur de fonctions. Nous avons ensuite utilisé un oscilloscope pour observer les signaux de sortie.

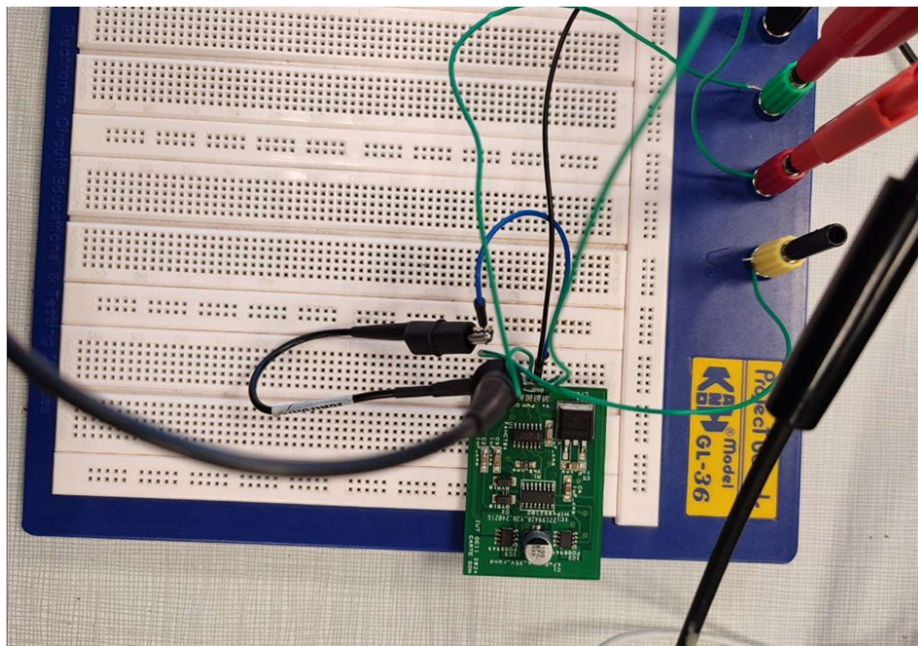


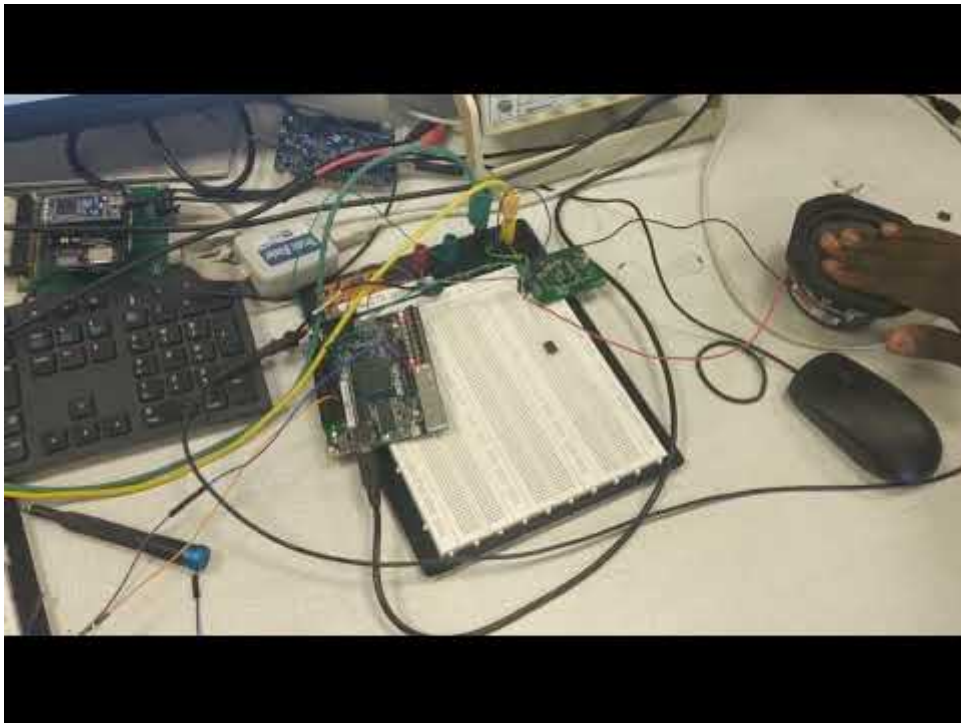
Figure 36 : Banc de test

## Résultats finaux

Une fois que chacun a terminé sa tâche, nous avons pu rassembler les différentes parties pour assembler le système complet.

Dans un premier temps, les cartes de conditionnement et de filtrage n'étant pas encore imprimées, nous avons testé le bon fonctionnement du système avec le montage de conditionnement réalisé sur une plateforme de test, sans le filtrage.

Voici la vidéo de ce premier test :



*Figure 37 : Vidéo de test de l'enceinte sur plateforme de test*

On constate qu'avec ces conditions de test, on obtient un son clair. Ce qui valide le travail réalisé par chacun des étudiants.



Une fois que toutes les cartes ont été imprimées et soudées, nous avons tout assemblé dans une boîte servant de caisse de résonance pour le haut-parleur. Le support du haut-parleur a été réalisé en impression 3D, et du bois de récupération a été utilisé pour le reste de la boîte afin d'offrir une meilleure résonance sonore.



*Figure 21 : Enceinte Bluetooth au complet et fonctionnelle*

Le système produit un son clair et une dynamique sonore fonctionnelle, attestant du caractère concluant de notre projet.

En conclusion, bien que notre système soient satisfaisants, des améliorations sont nécessaires. Par manque de temps, nous avons alimenté le système avec une batterie de 12V sans mettre en place de dispositif de protection adéquat avant la présentation. Il serait donc recommandé de l'ajouter. De plus, l'intégration d'un montage permettant de varier le volume du son serait également bénéfique pour optimiser le fonctionnement global du système.