

Enregistreur Vocal  
-  
Rapport SAE

## Introduction

Lors de cette SAÉ nous avons réalisé un enregistreur vocal. Le but est d'enregistrer un message venant d'un micro ou d'un ordinateur dans une mémoire RAM et de pouvoir le restituer plus tard.

Pour cela nous avons dû câbler 4 boutons :

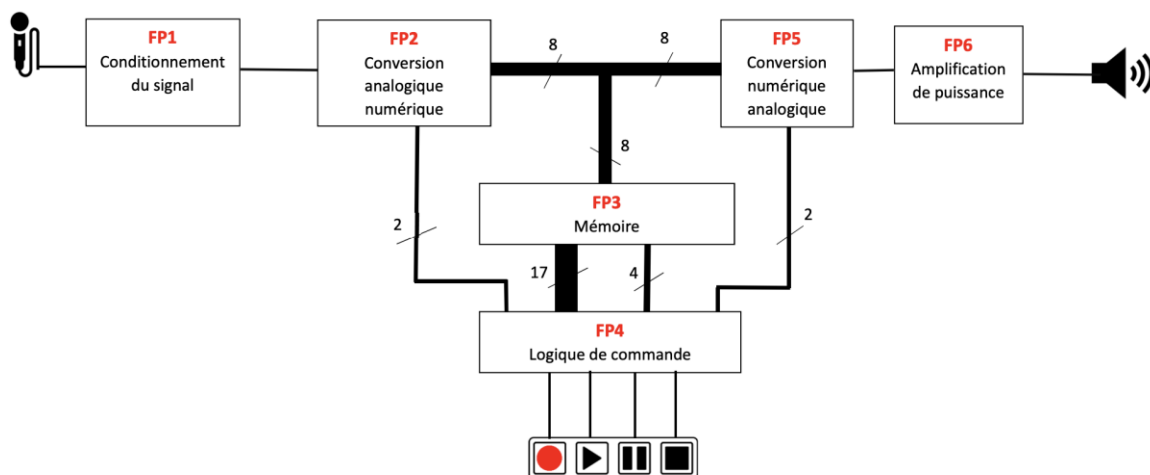
- REC : Mode enregistrement
- PLAY : Mode lecture
- Pause : Met en pause le processus en cours (REC ou PLAY)
- STOP : Arrête le processus en cours

Le montage final étant conséquent, nous avons dû respecter un code couleur pour faciliter la compréhension et la maintenance de notre montage (Rouge pour 3v3, vert pour +15V et jaune pour -15V).

Enfin, nous avons dû reproduire tout le comportement de la RAM et de la logique de commande à l'aide d'une plaque FPGA dans le but d'alléger notre montage.

Pour réaliser l'enregistreur vocal au complet, nous avons fonctionné bloc fonctionnel par bloc fonctionnel en suivant le schéma suivant :

**Schéma fonctionnel global**



Boutons poussoirs : REC, PLAY, PAUSE, RESET

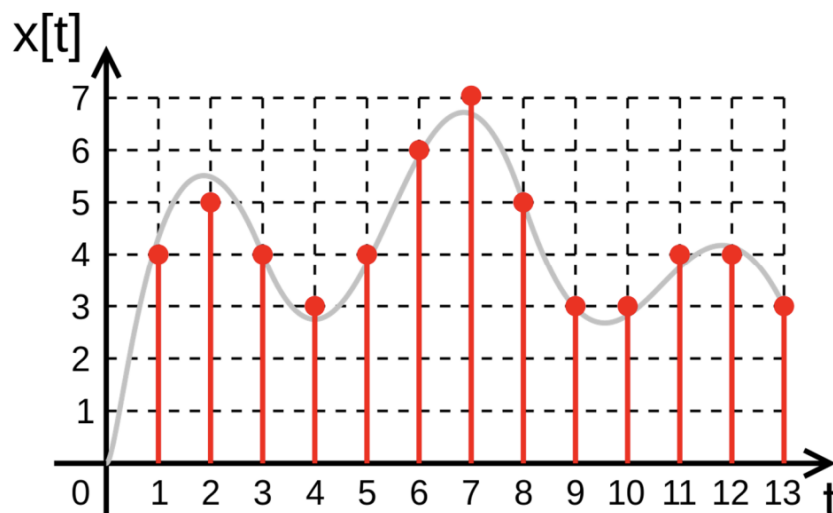
## Principe de fonctionnement

L'utilisateur parle dans un microphone qui transforme le signal sonore en une tension électrique. Cette tension électrique est un signal qualifié d'analogique, car il peut varier à n'importe quel instant et peut prendre une infinité de valeurs comprises entre sa valeur minimale et sa valeur maximale.

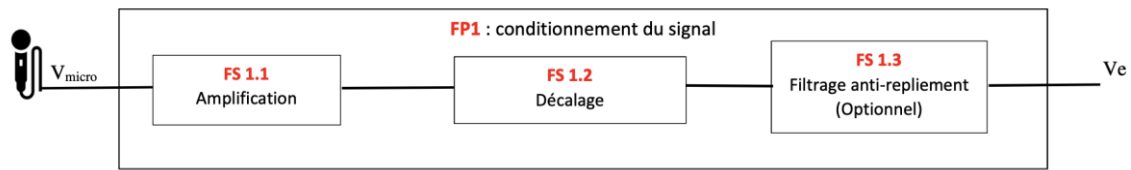
Les cassettes audios à bande magnétique ont servi de support effaçable et ré-enregistrable jusque dans les années 1990. Aujourd'hui, il est préférable d'utiliser une RAM (Random Access Memory), plus légère, plus robuste, plus fiable et moins chère qu'un système à cassette. Une RAM est un système électronique qui ne peut stocker des informations que sous formes numériques, elle mémorise des paquets de 8 bits appelés octets dans des cases mémoires. Chaque bit ne peut prendre que deux valeurs (0 ou 1) et les cases mémoires sont repérées par un numéro appelé « adresse ».

Pour remplir la mémoire, il est donc indispensable de convertir le signal analogique en un signal numérique : c'est le rôle de la fonction principale FP2. Pour réaliser cette fonction, nous utiliserons le composant AD7819.

Ci-dessous, un exemple de chronogramme illustrant le passage d'un signal analogique à un signal numérique. En gris, le signal analogique  $x(t)$  évolue continument entre 0 et 7V et peut prendre toutes les valeurs de l'intervalle [0 V ; 7 V]. En rouge, le signal numérique qui ne peut changer qu'aux instants d'échantillonnage (ici toutes les secondes) et qui ne peut prendre ici que 8 valeurs discrètes (0V ; 1V ; 2V ; ... ; 7V).



## Étape 1



1. En parlant fort dans le micro, on relève une tension de sortie comprise entre [-80mV ; +80mV].

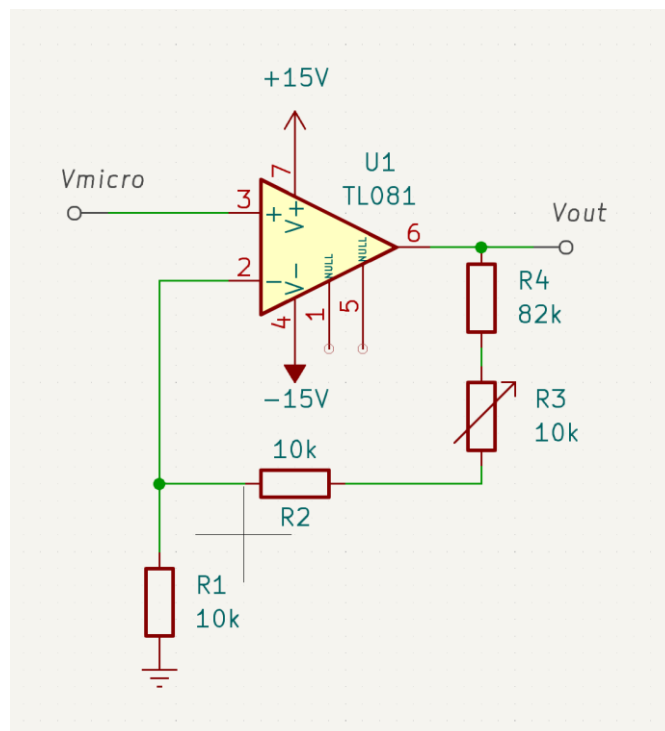
2. On veut que la tension d'entrée  $V_e$  appartienne à l'intervalle [0 ; 3,3V].

On doit donc avoir  $V_e = A \cdot V_{micro} + B$

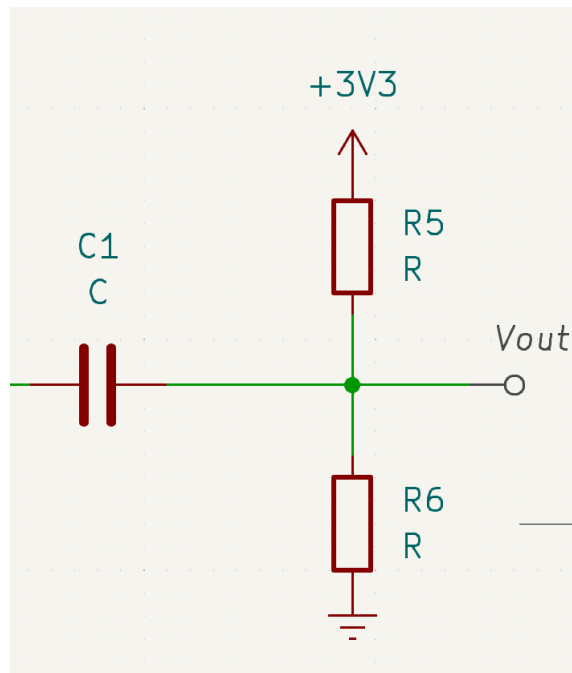
On trouve  $A = 18,8$  et  $B = 1,65$

En mettant un gain de 18,8 et un offset de 1,65V,  $V_e$  sera bien compris entre 0V et 3,3V

3.



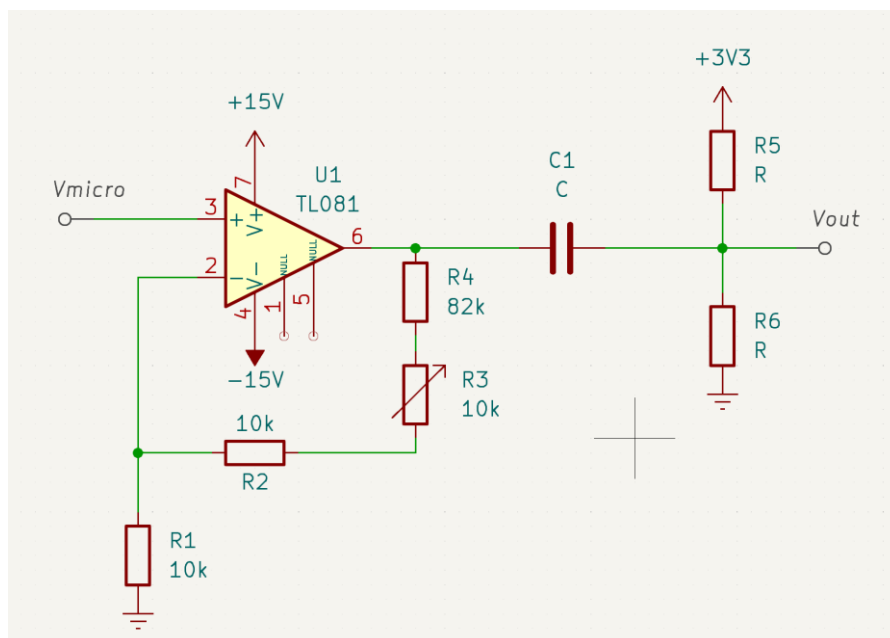
FS 1.1 : Montage amplificateur non-inverseur



FS 1.2 : Offset de 1.65V

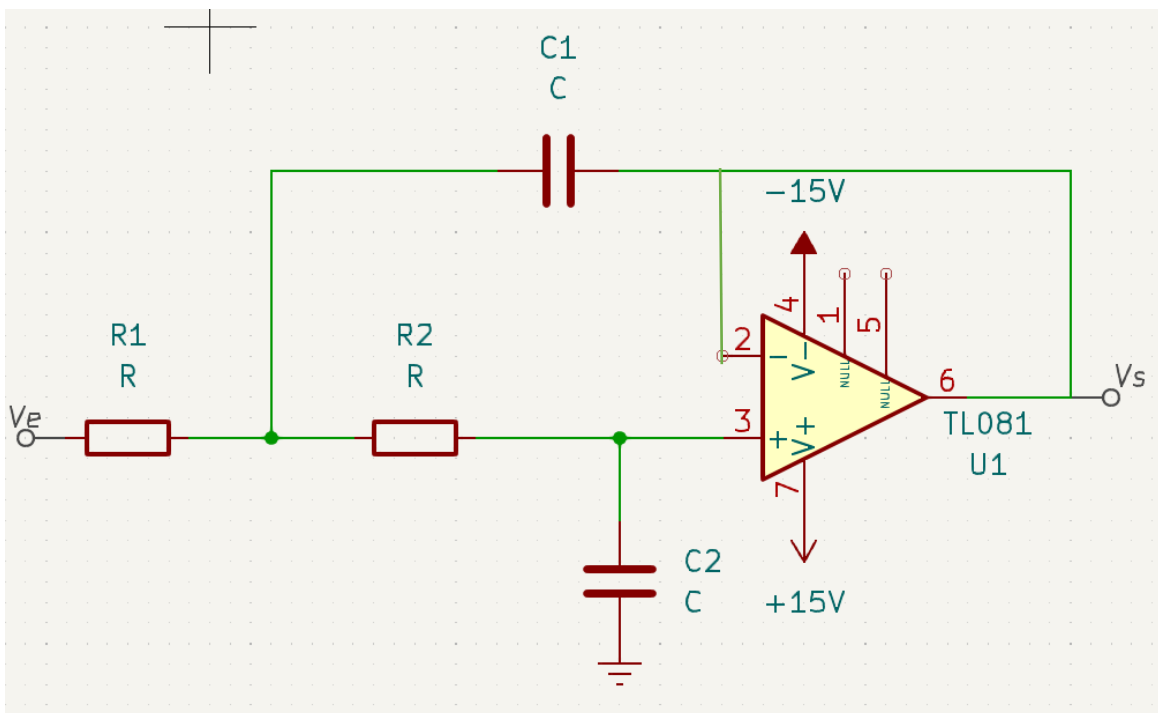
Les deux montages ci-dessus permettent de réaliser les fonctions FS1.1 (amplification) et FS1.2 (Décalage) conditionnement du signal.

En les mettant bout à bout on obtient le montage suivant :



FP1 : Conditionnement du signal

- 4.
- a. Le filtre anti-repliement est un filtre passe-bas du second ordre ou plus qui va atténuer les fréquences trop hautes par rapport au théorème de Shannon-Nyquist. Il empêche donc d'échantillonner les fréquences qui pourraient de replier. Si on échantillonne à 10kHz, la fréquence de coupure idéale du filtre est de 5kHz.
- b. Voir étape 8
- c.

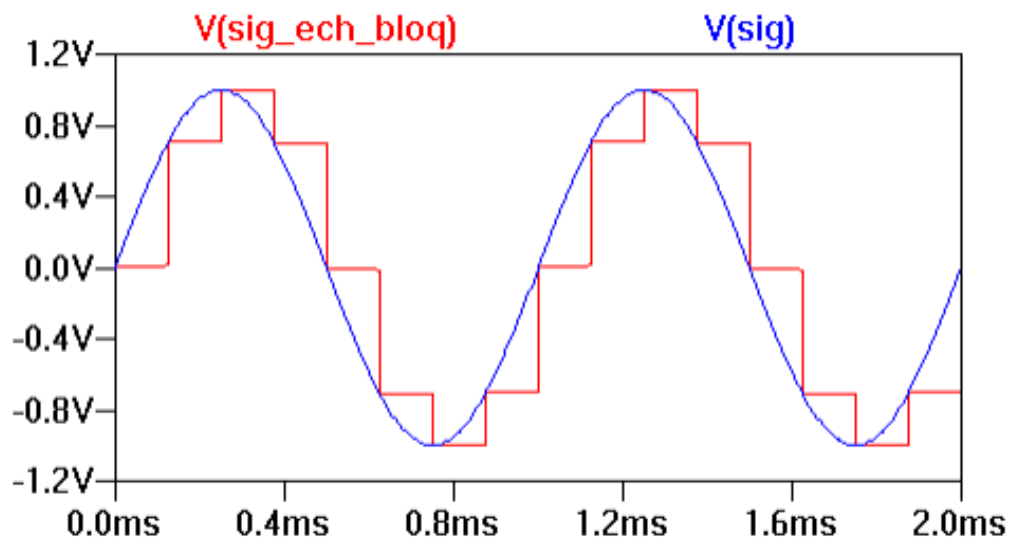


FS 1.3 : Filtrage anti-repliement

- d. Lorsque nous avons câblé le filtre, il y avait plus de bruit dans le signal de sortie que lorsque le filtre n'était pas câblé. Après plusieurs vérifications, il n'y avait aucune erreur de câblage, nous avons donc préféré ôter le filtre anti-repliement de notre montage.

## Étape 2

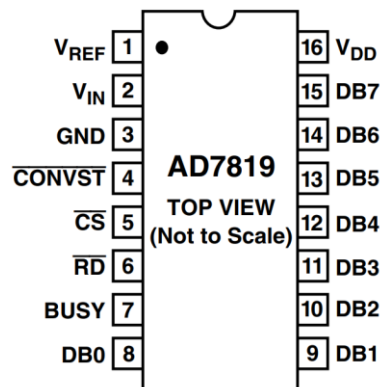
Cette étape va consister à câbler et tester simultanément les fonctions principales FP2 et FP5. En appliquant à l'entrée de FP2 une sinusoïde de fréquence  $f=1\text{kHz}$  (en bleu ci-dessous) et que l'on fixe une fréquence d'échantillonnage  $F_e = 10\text{kHz}$ , nous récupérons un signal échantillonné-bloqué (courbe rouge) en sortie de FP5.



*Signal d'entrée de FP2 (en bleu) et en sortie de FP5 (en rouge)*

## FP2 :

Pour réaliser les conversions analogique-numériques nous avons utilisé le composant AD 7819 donc le brochage est donné ci-dessous :



1. Ce convertisseur est un convertisseur 8 bit.
2. C'est un convertisseur analogique numérique.
3. On peut alimenter ce composant en 3V3 sur la broche 16 car sa plage d'alimentation est : [2.7V ; 5.5V].
4. Sachant qu'on alimente le convertisseur en 3V3, la plage de variation de la tension analogique en entrée sera [0 ; 3V3] si on relie les broches 1 et 16.
5. C'est un front descendant sur la broche CONVST/ qui déclenche une conversion.
6. Les broches 5 et 6 doivent être à l'état bas pour que le résultat de la conversion soit lisible sur les sorties.
7. La broche 7 est la broche BUSY. Elle est à l'état haut lorsque le processus de conversion est en cours. Elle indique donc lorsque le composant est "occupé" ou non.
8.
  - a. Le 7819 est un CAN 8bit, 3V3 représente donc ici la valeur numérique 255. Alors 3V représente :  $\frac{3 \cdot 255}{3.3} = 232$ . L'état des broches de sorties est : (1110 1000)<sub>b</sub>
  - b. En utilisant le même raisonnement : 1V représente :  $\frac{1 \cdot 255}{3.3} = 77$ . L'état des broches de sorties est : (0100 1101)<sub>b</sub>.

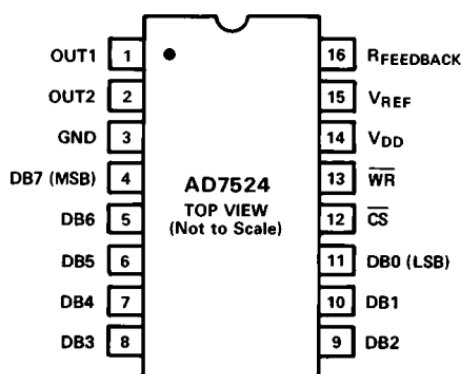
On comprend donc qu'il va être fastidieux de vérifier l'état des sorties du CAN une à une pour plusieurs valeurs analogiques. La solution qui se présente à nous est de relier les 8 sorties numériques au convertisseur numérique-analogique, c'est à dire



FP5. Si le signal obtenu en sortie est similaire au signal d'entrée, notre câblage sera fonctionnel.

FP5 :

Pour réaliser les conversions analogique-numériques nous avons utilisé le composant AD 7819 donc le brochage est donné ci-dessous :



9.

10. Pour qu'un changement des broches numériques affecte la tension de sortie analogique, les broches WR/ et CS/ (12 et 13) doivent être à l'état bas.

11.

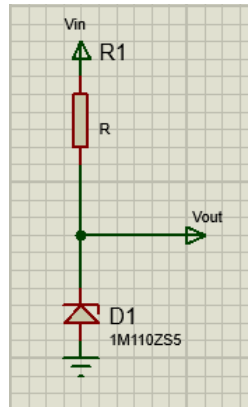
a. On sait que le convertisseur AD7524 est convertisseur 8 bits (256 valeurs). Ici, la valeur max est fixée par Vref (5V). Vref correspond donc à la dernière valeur possible sur 8 bits (255).

Sachant cela, on peut facilement trouver la tension analogique de sortie qui correspond à l'entrée numérique (0111 1010) en binaire à l'aide d'un simple produit en croix.

On trouve donc  $V_{out} = 2,39V$

b. En utilisant la même méthode, on trouve  $V_{out} = 3,27V$

13.



On prend  $R = \frac{10 - V_Z}{I} = \frac{10 - 8.2}{7.75 \cdot 10^{-3}} = 245 \Omega$

14. En remplaçant  $V_{ref} = 3V$  par  $V_{ref} = 6V$ , on remarque que le convertisseur peut travailler sur une plus grande plage de valeur et a moins de chances de saturer.

### **Etape 3**

Dans cette étape, on cherchera à comprendre le fonctionnement de la fonction principale FP6 avant de la câbler.

L'impédance d'un haut-parleur peut être modélisée en première approximation par une résistance de  $4\Omega$  (parfois  $8\Omega$  suivant le haut-parleur utilisé).

Une sinusoïde d'amplitude 6V aux bornes d'un haut-parleur  $4\Omega$  produira une puissance sonore égale à :

$$P = \frac{V_{eff}^2}{R} = \frac{\left(\frac{6}{\sqrt{2}}\right)^2}{4} = 4,5 Watt$$

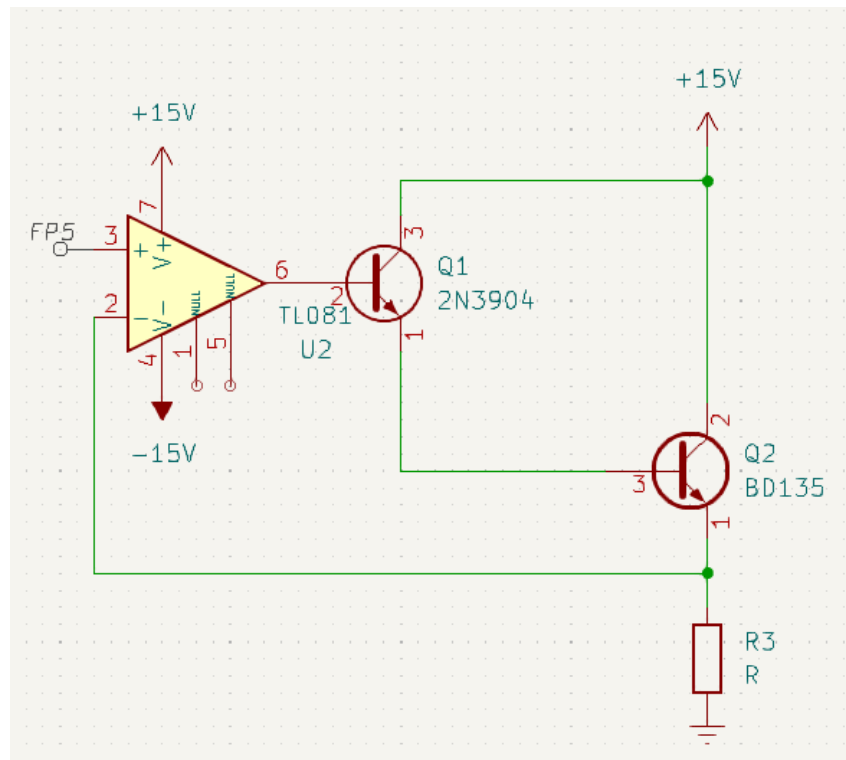
1. Un tel courant (1,5A d'amplitude) ne peut pas être fourni par l'amplificateur opérationnel de la fonction FP5 qui possède une résistance de sortie non nulle.

Le courant maximal que peut fournir un amplificateur opérationnel dont la tension de sortie vide est égale à 13V et dont la résistance de sortie vaut  $350 \Omega$  est :

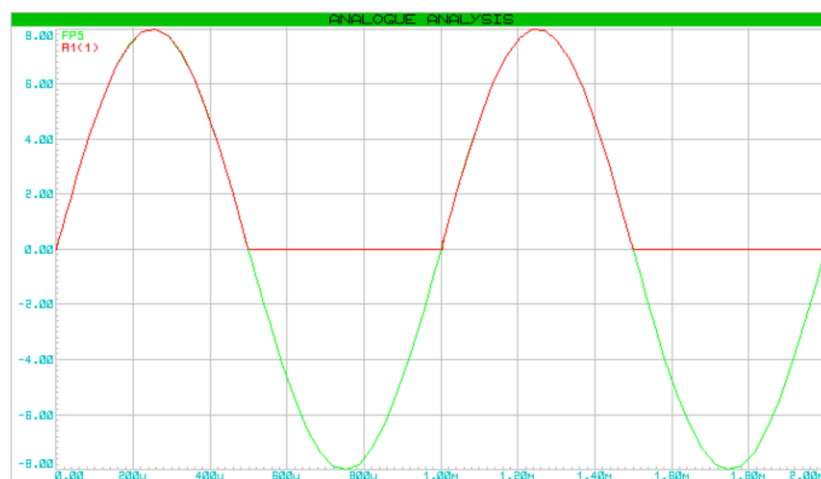
$$I = \frac{U}{R} = \frac{13}{350} = 37mA$$

2. La réponse qui précède met donc en évidence la nécessité d'amplifier le courant

de sortie de FP5. On propose le montage ci-dessous pour amplifier les alternances positives :

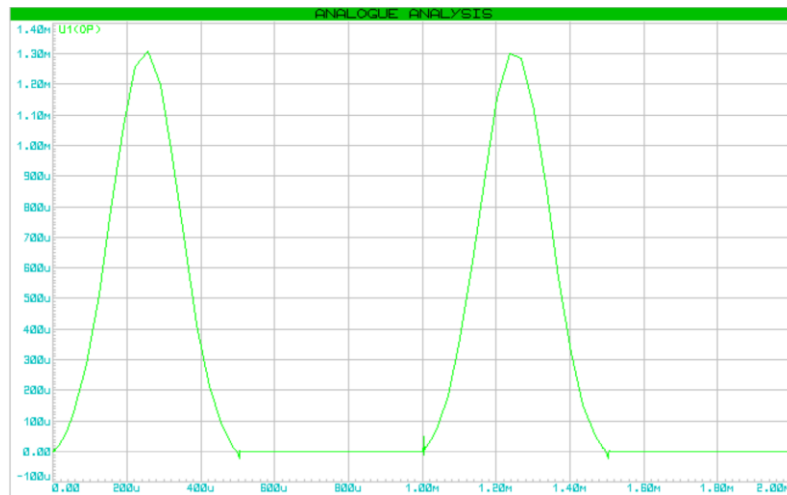


On effectue ensuite des relevés des tensions et courant de sortie du montage :



FP6 : Relevé des tensions  $V_{in}$  (vert) et  $V_{out}$  (rouge)

Grâce à ce premier relevé, on observe bien que ce premier montage conserve uniquement les alternances positives de la tension sinusoïdale venant de FP5



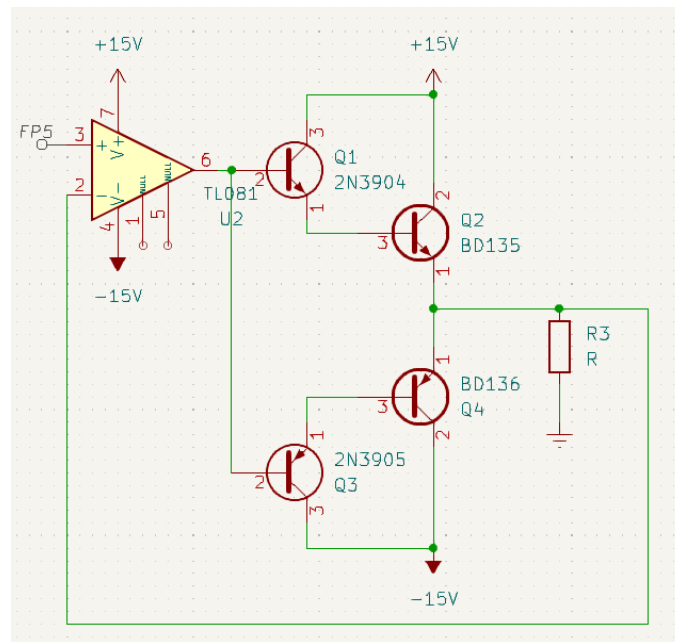
*FP6 : Relevé du courant sortant de l'ampli et entrant dans la base du transistor 2N3904*

Le problème est donc le même pour le courant, ce montage ne permet pas de garder les alternances négatives.

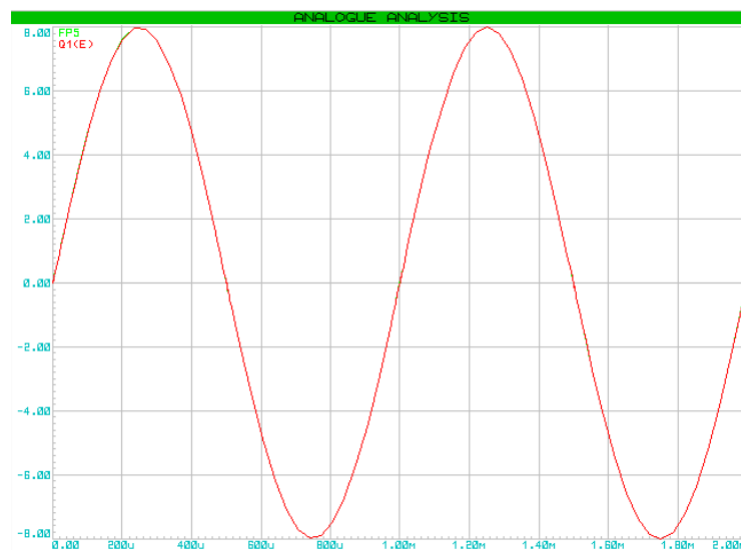
- a. Dans R1, le courant max est de 2A
- b. L'amplificateur opérationnel fourni un courant max de 1.3mA.
- c. Le montage fourni donc un gain en courant de 1538.

Désormais, nous souhaitons que notre montage amplifie les alternances positives ET négatives du courant du signal d'entrée venant de FP5.

Pour amplifier les alternances négatives, on réutilise le même principe que pour le premier montage avec les transistors bipolaires.



On observe ensuite la tension de sortie de ce nouveau montage :



FP6 : Relevé des tensions  $V_{in}$  (vert) et  $V_{out}$  (rouge)

On remarque que les tensions d'entrée et de sortie du montage se superposent parfaitement, le montage est fonctionnel.

Enfin, il faut encore ajouter quelques améliorations pour obtenir le montage final :

- [illegible]

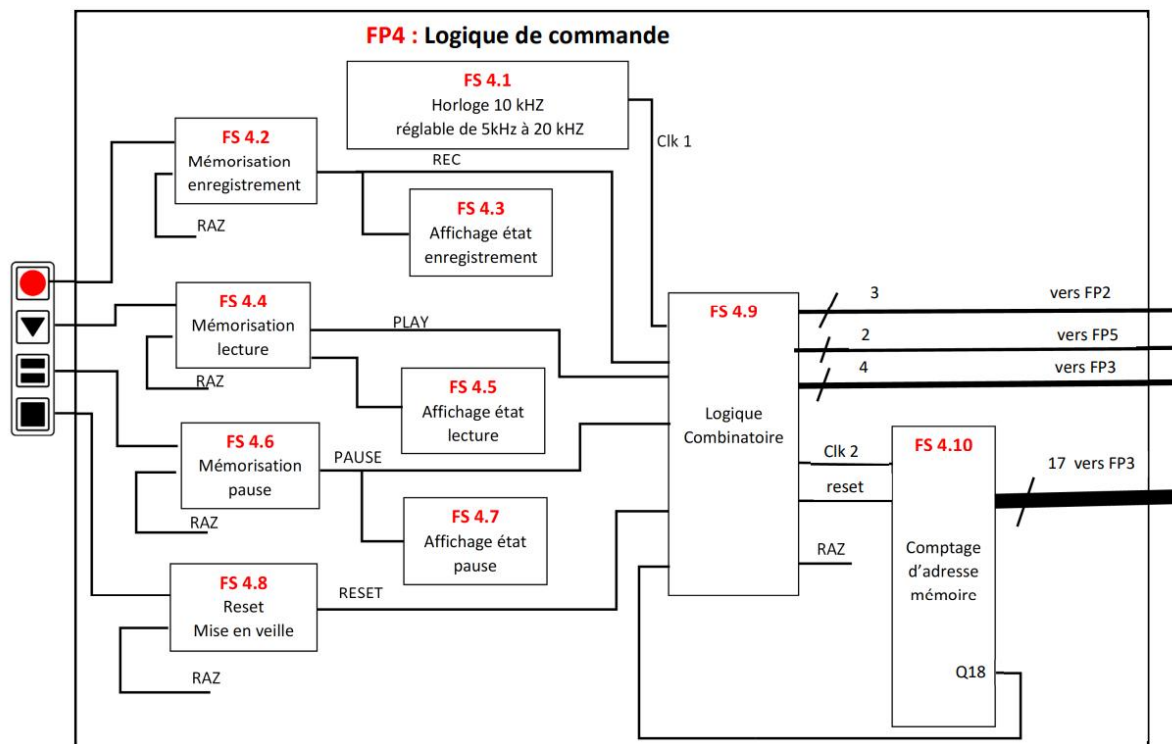
3. La fonction de transfert de ce filtre est :  $H(j\omega) = \frac{-jR3C5\omega}{1+jR3C6\omega} \cdot \frac{1}{1+jR3C5\omega}$

14

## Etape 4

Lors de cette étape, nous allons étudier et câbler la logique de commande de notre enregistreur vocal.

Nous allons mettre en place un système de boutons (REC, PLAY, PAUSE, RESET) pour contrôler le fonctionnement de notre montage.



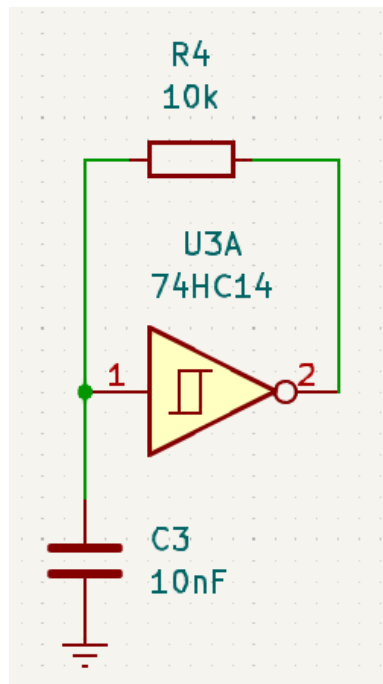
*FP4 : Logique de Commande*

- Le bouton REC permet de lancer un enregistrement
- Le bouton PLAY permet de lire un message enregistré
- Le bouton PAUSE permet de mettre en pause la lecture d'un message et de reprendre la lecture lors d'un second appui sur le bouton
- Le bouton RESET permet de remettre à zéro le montage et de désactiver les actions qui ont été démarrées par les autres boutons

### A) Fonction secondaire FS4.1

Cette fonction permet de générer un signal d'horloge. Ce signal d'horloge définira la fréquence d'échantillonnage de notre montage.

A.1)



FS4.1 : Signal d'horloge

a) Après avoir câbler ce montage, on mesure une fréquence d'oscillation de  $f = \frac{k}{RC}$  On en déduit donc que  $k = f \cdot R \cdot C$

On fixe  $R = 10\text{k}\Omega$  et  $C = 10\text{nF} \implies$  On mesure une fréquence de 10kHz

$$\implies k = \frac{1}{fRC} = \frac{1}{10^4 \cdot 10^4 \cdot 10^{-8}} = 1$$

b) Donc pour obtenir 20kHz : on fixe  $R = 50\text{k}\Omega$  et  $C = 1\text{nF}$ .

c) Pour pouvoir régler la fréquence d'oscillation on ajoute un potentiomètre.



## **B] Fonction secondaire FS4.10**

B.1) Sachant que la RAM possède une capacité de 128kO, 17 bits d'adresses seront nécessaires car  $2^{17} = 131\ 072$ .

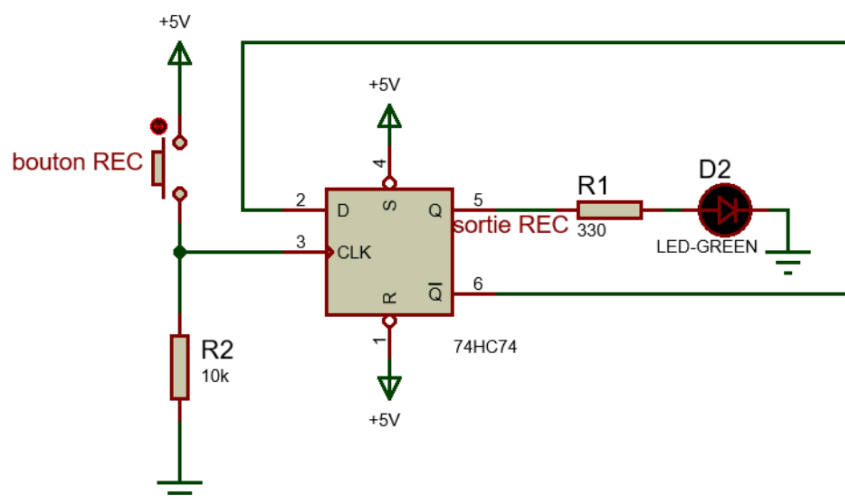
B.2) La logique synchrone incorpore une horloge qui sert à enclencher les actions alors que la logique asynchrone n'en possède pas.

### **C] Fonction secondaire FS4.2 à FS4.8**

Pour mémoriser la pression que l'utilisateur exerce sur un bouton poussoir, il est courant d'utiliser une bascule D (74HC14 ou CD4013).

C.1) La sortie Q d'une bascule D recopie l'entrée de la bascule à chaque front d'horloge.

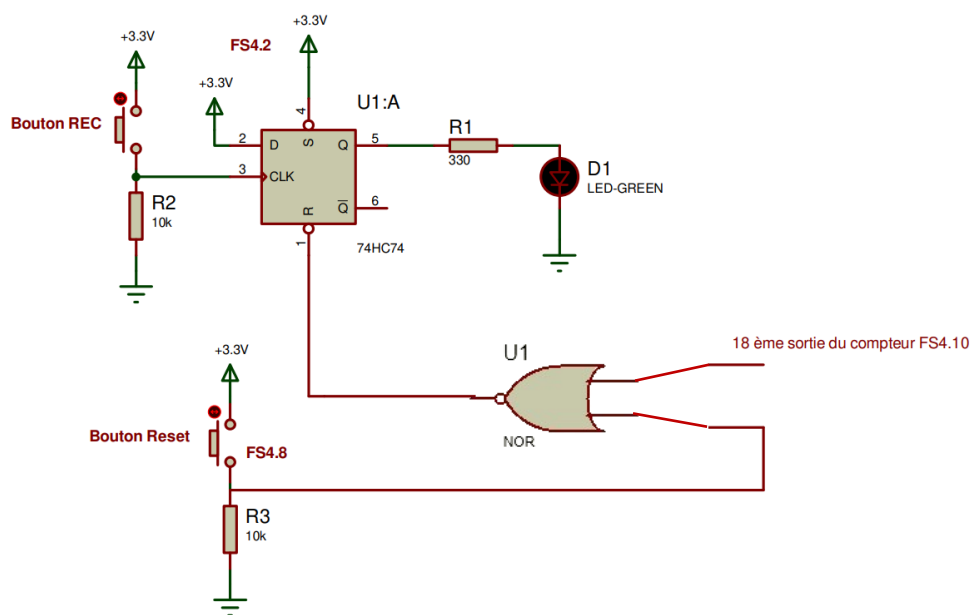
C.2) Pour mémoriser l'appui sur le bouton poussoir, nous réalisons le montage ci-dessous.



*FS4.2 : Mémorisation d'enregistrement*

En simulant ce montage sur Proteus, on observe le bon fonctionnement de la bascule D. A chaque pression sur le bouton poussoir, l'état de la LED (ici D2) est inversé.

C.3) Lors d'un enregistrement, la bascule D mémorise l'appui sur le bouton REC. La sortie Q de cette bascule reste à l'état haut pendant toute la séquence d'enregistrement. La séquence d'enregistrement prend fin lorsque la mémoire est pleine ou lorsque l'utilisateur appui sur le bouton RESET.



a) La porte logique à rajouter est une porte NOR.

b) La fonction FS4.3 est la fonction secondaire qui permet l'affichage de l'état d'enregistrement. Cette fonction est réalisée par une LED (rouge dans notre cas). Cette LED est accompagnée d'une résistance pour limiter le courant et donc éviter tout risque de dégradation

#### **FS4.9 : Logique Combinatoire**

La logique combinatoire est la partie la plus importante du projet car elle nécessite d'avoir parfaitement compris le fonctionnement et les datasheets de FP2 (CAN), FP3 (mémoire) et FP5 (CNA).

Liste des entrées et sortie de FS4.9 :

- Entrées:
  - Le signal d'horloge de fréquence réglable clk1
  - Les sorties des 4 bascules D : REC, PLAY, PAUSE, RESET
  - La 18<sup>ème</sup> sortie du compteur d'adresse.
  
- Sorties:
  - 3 sorties vers FP2 : CS\, RD\ et CONVST\
  - 2 sorties vers FP5 : CS\ et WR\
  - 4 sorties vers FP3 : CE1\, CE2, WR\ et OE\ (parfois appelé RD\)
  - 1 ou 2 sorties vers FS4.10 : MR\ du compteur d'adresse et éventuellement clk2 si différente de clk1

Pour chaque séquence de fonctionnement, nous avons réalisé les chronogrammes des sorties de FS4.9 en fonction des différentes entrées.

Les séquences possibles si vous câblez toutes les options sont :

-enregistrement en cours => (REC, PLAY, PAUSE, RESET) = (1,0,0,0)

-lecture en cours => (REC, PLAY, PAUSE, RESET) = (0,1,0,0)

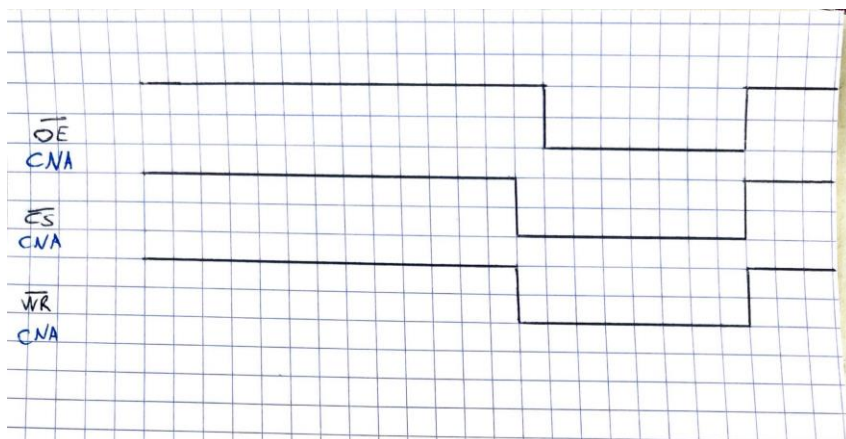
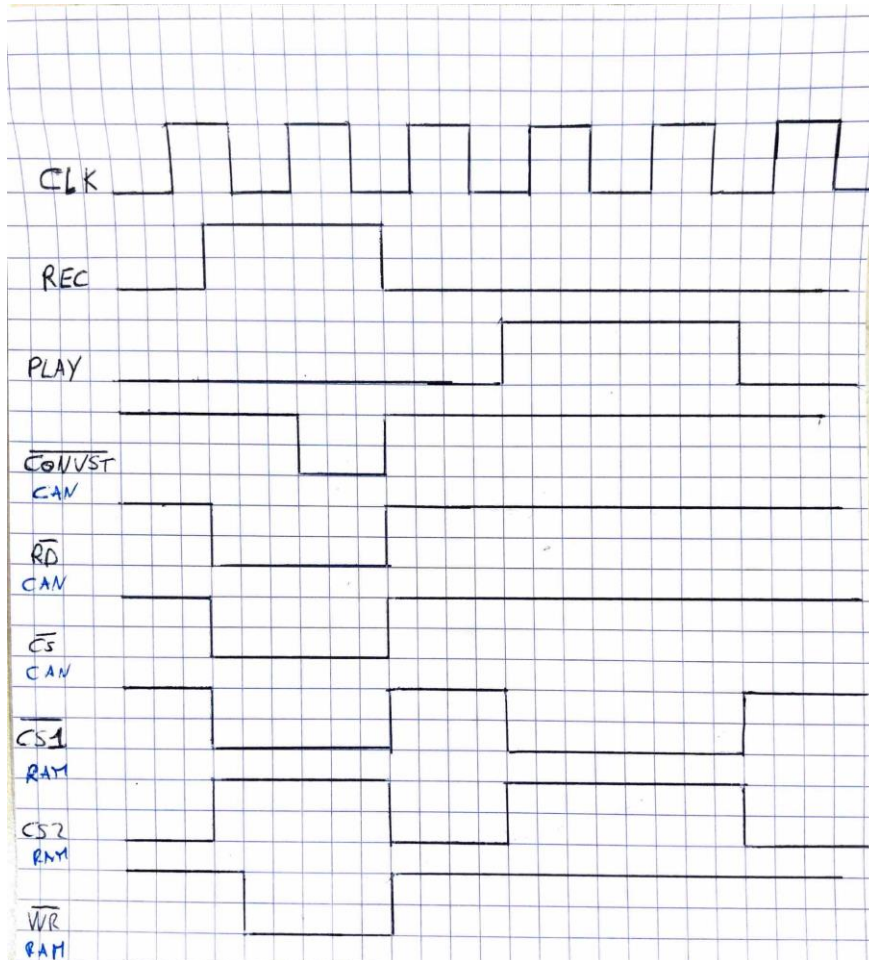
-lecture et enregistrement simultané => (REC, PLAY, PAUSE, RESET) = (1,1,0,0)

-enregistrement en pause=> (REC, PLAY, PAUSE, RESET) = (1,0,1,0)

-lecture en pause=> (REC, PLAY, PAUSE, RESET) = (0,1,1,0)

-enregistrement lecture simultané en pause=> (REC, PLAY, PAUSE, RESET) = (1,1,1,0)

D.1)



D.2) A partir des chronogrammes précédents, nous pouvons établir les équations logiques de toutes les sorties :

FP2 (CAN) :

- $\overline{CS} = \overline{REC}$
- $\overline{RD} = \overline{REC}$
- $\overline{CONVST} = \overline{REC \& CLK}$

FP5 (CNA) :

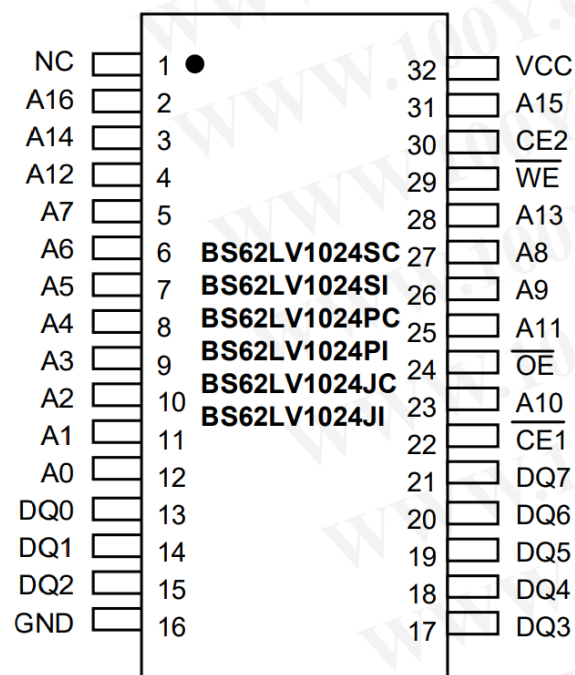
- $\overline{CS} = \overline{PLAY}$
- $\overline{WR} = \overline{PLAY} \& \overline{CLK}$

### FP3 (mémoire) :

- $\overline{CE1} = \overline{REC \text{ OR } PLAY}$
- $\overline{CE2} = \overline{REC \text{ OR } PLAY}$
- $\overline{WR} = \overline{REC \& CLK}$
- $\overline{OE} = \overline{PLAY \& CLK}$

## Etape 5

Cette étape consiste à réaliser le câblage de la RAM dans notre montage.



FP3 : Pinout de la RAM

1) Montage et fonctionnement validés par l'enseignant

2) Lorsque l'on fait varier la fréquence d'échantillonnage avec le réglage potentiométrique, on constate que l'on peut enregistrer un long message de faible qualité audio lorsque la fréquence d'échantillonnage est réglée au plus bas (5kHz). Lorsque l'on règle la fréquence d'échantillonnage au maximum (20kHz) la qualité audio est beaucoup plus importante mais la durée d'enregistrement est plus courte.

3) Lorsque  $f_e = 20\text{kHz}$ , la qualité audio de notre enregistrement est bonne, le son est clair et compréhensible. A l'inverse, lorsque l'on règle  $f_e = 5\text{kHz}$ , la qualité audio est fortement dégradée, le son est beaucoup moins clair et peu compréhensible.

Cette différence de qualité audio est simplement dû au fait que l'on échantillonne beaucoup plus sur une même période lorsque  $f_e = 20\text{kHz}$  que lorsque  $f_e = 5\text{kHz}$ . Par conséquent, à  $f_e = 20\text{kHz}$ , le signal restitué est plus complet et plus proche du signal d'origine que lorsque  $f_e = 5\text{kHz}$ .

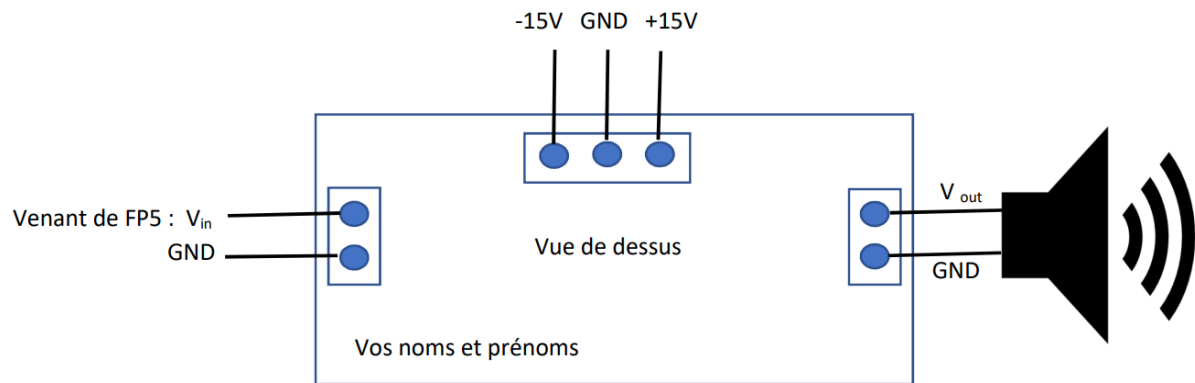
## **Etape 6**

A ce stade, nous disposons d'un montage fonctionnel sur breadboard, capable d'enregistrer et restituer des signaux sonores de quelques secondes.

Nous avons ensuite conçu une carte électronique de dimension maximale 8cm\*6cm réalisant la fonction FP6 (amplification audio).

Pour cela, nous avons utilisé le logiciel Kicad qui est un logiciel libre pour la conception de circuit imprimés.





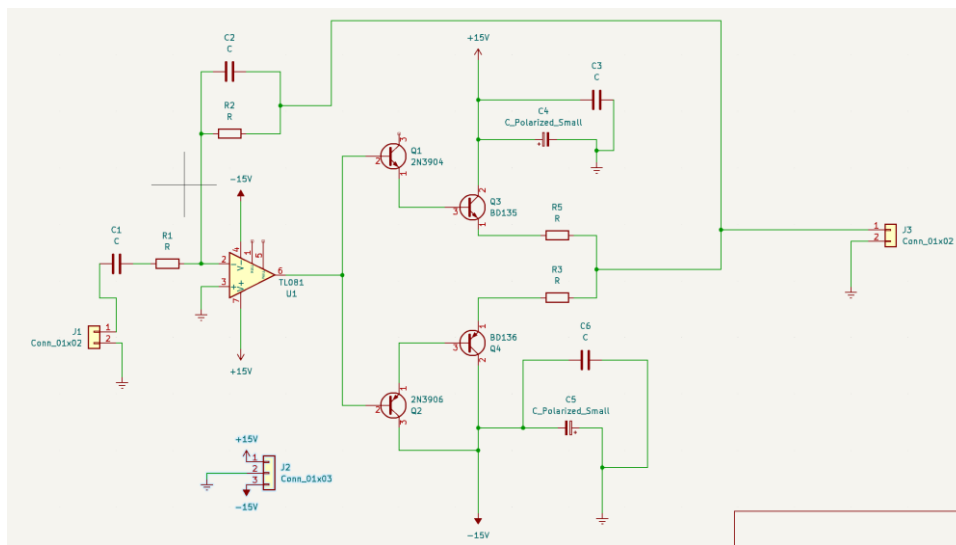
*FP6 : Amplification audio*

Comme nous pouvons le voir ci-dessus, notre carte possède les 3 connecteurs suivants :

- 1 connecteur 2 broches à gauche :  $V_{in}$  et GND
- 1 connecteur 3 broches en haut : -15V, GND et +15V
- 1 connecteur 2 broches à droite :  $V_{out}$  et GND

## A] Schéma électrique

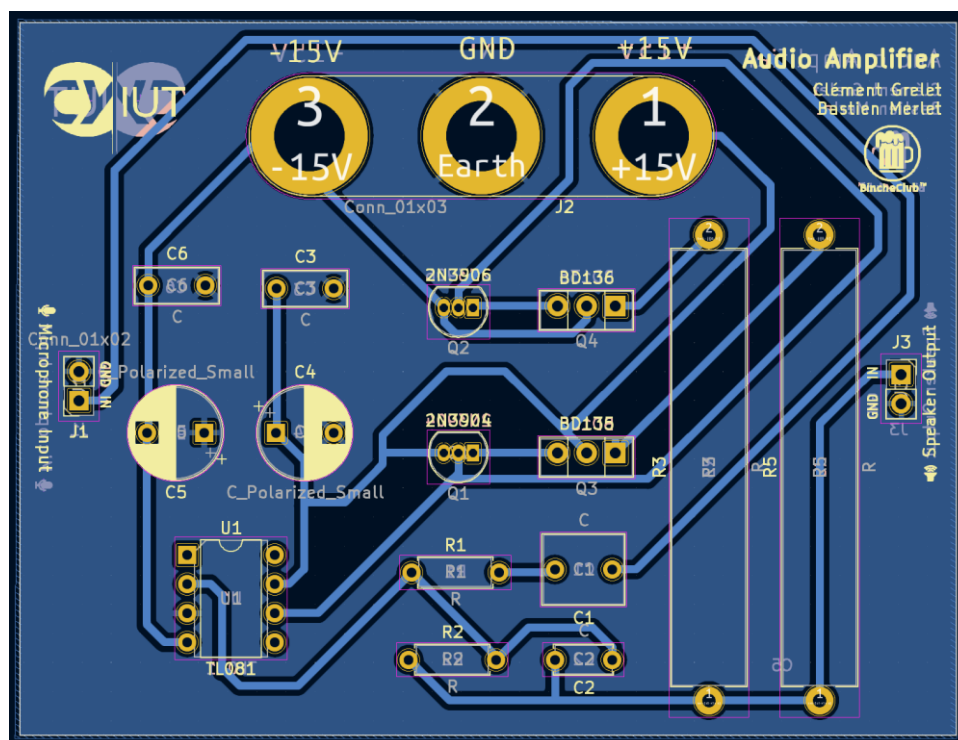
La première étape dans la conception de notre PCB est la schématisation électrique du montage. Cette étape est cruciale car c'est ici que l'on définit tout le câblage de notre ampli audio.



Comme le montre l'image ci-dessus, le montage que l'on réalise sous Kicad pour réaliser notre PCB est le même que nous avons simulé sous proteus lors de l'étape 3.

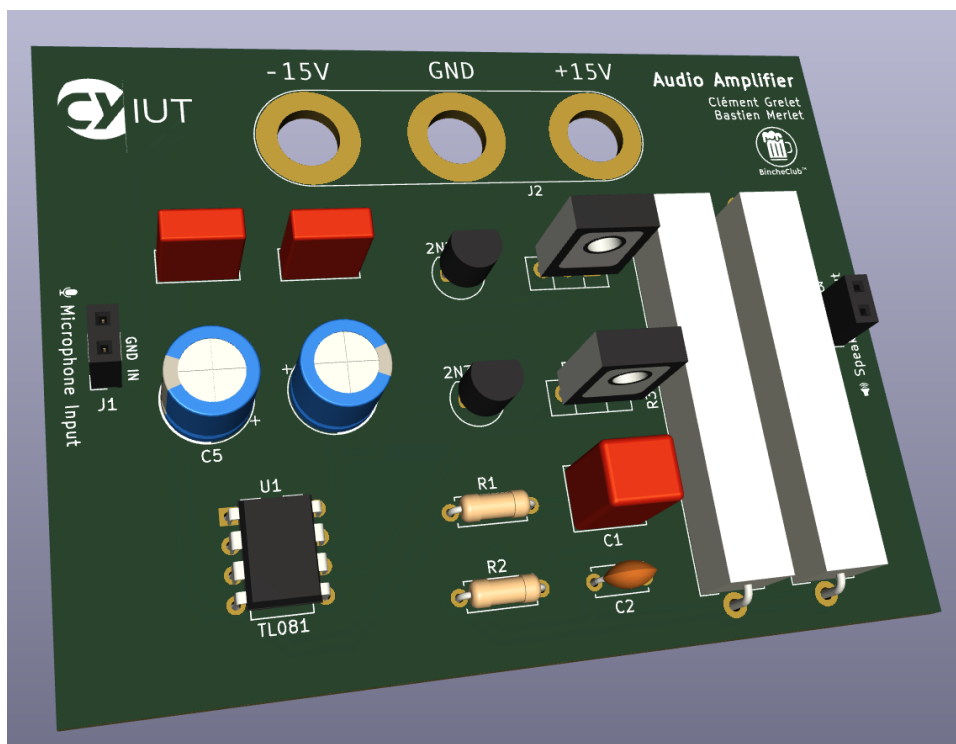
## B] Réalisation du PCB

Le routage de notre carte est la deuxième grande étape de sa réalisation. C'est à ce moment que nous choisissons le placement de nos composants sur la carte ainsi que d'autres éléments comme la taille des pistes, le nombre de couches, les dimensions du PCB...

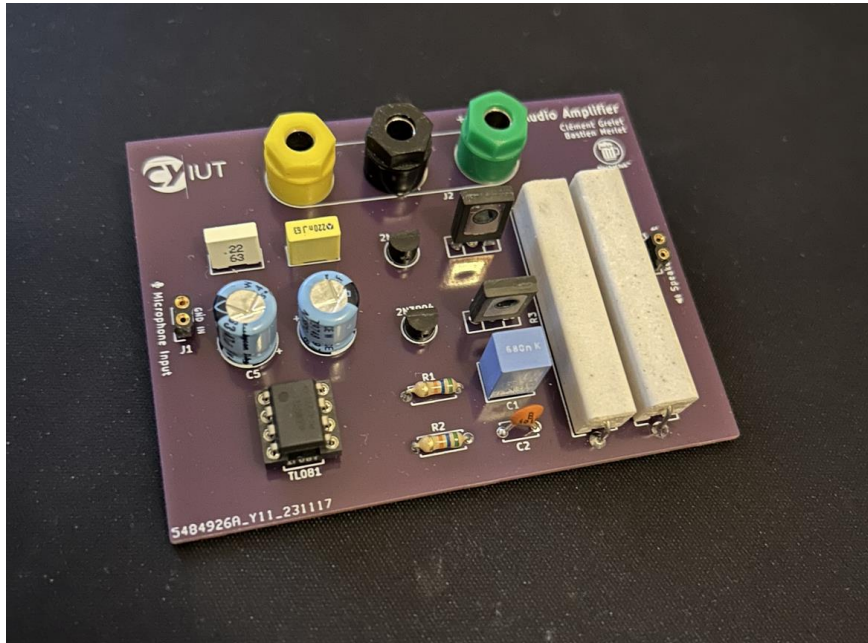


### C] Vue 3D

Dernière étape de la conception de notre PCB, la vue 3D nous permet d'avoir un premier aperçu du rendu final de notre carte.



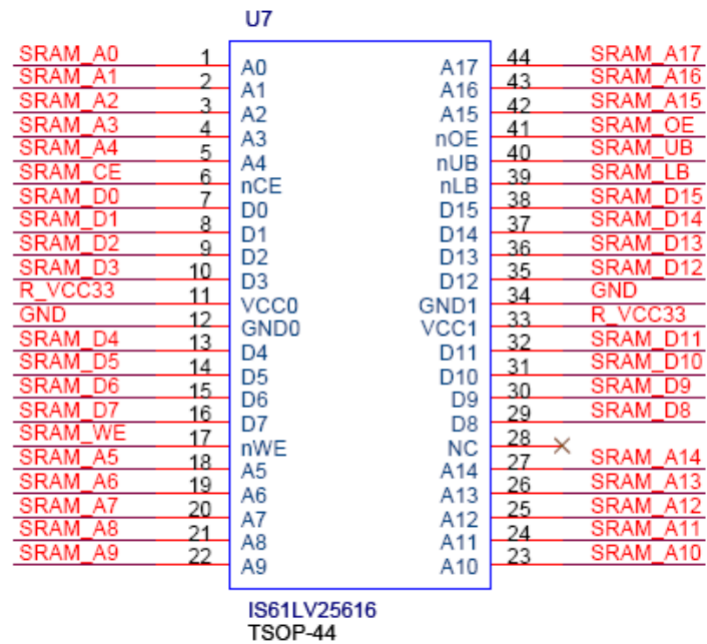
## Photo du PCBA



## Etape 7

Notre montage fonctionne désormais parfaitement sur breadboard. Nous allons donc maintenant utiliser une carte ALTERA DE1 (FPGA) pour réaliser les fonctions FP3 (mémoire) et FP4 (logique de commande). Nous utiliserons le logiciel Quartus 13 pour programmer cette carte.

Sur la carte ALTERA DE1, nous disposons d'une SRAM 512kO, organisée en 256k \* 2 Octets, ce qui signifie que cette RAM possède 256k=218 adresses mémoires et qu'à chaque adresse mémoire, il est possible de stocker 2 octets : Lower data pour l'octet de poids faible et Upper data pour l'octet de poids fort. Puisque nous souhaitons remplacer la SRAM 128kO câblée sur notre plaque lab, nous n'utiliserons que 128k adresses soit 17 broches et un seul octet de données (fixons le LOWER data).



*Pinout SRAM*

1)

Broche d'alimentation	Broches de la RAM
VCC	GND
11,33	12,34

NOM DE LA BROCHE	BROCHE DE LA RAM	BROCHE DU FPGA
WRRAM	17	AA10
/CSRAM	6	G16
/OERAM	41	T8
D0	7	AA6
D1	8	AB6
D2	9	AA7
D3	10	AB7
D4	13	AA8
D5	14	AB8

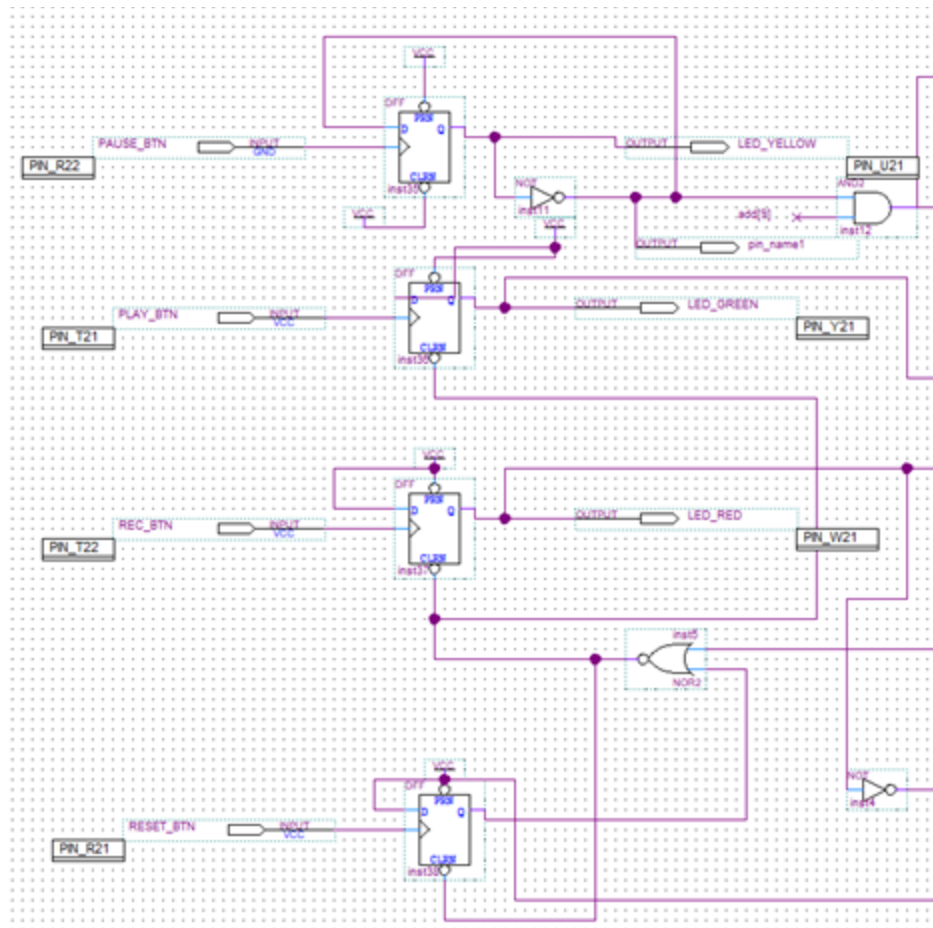
D6	15	AA9
D7	16	AB9
A16	43	Y6
A15	42	T7
A14	27	R10
A13	26	U10
A12	25	Y10
A11	24	T11
A10	23	R11
A9	22	W11
A8	21	V11
A7	20	AB11
A6	19	AA11
A5	18	AB10
A4	5	AA5
A3	4	AB4
A2	3	AA4
A1	2	AB3
A0	1	AA3
LOWBYTE	39	Y7
HIGHBYTE	40	W7

Nous fixons /RAMLOWBYTE au GND et /RAMHIGHBYTE à VCC afin de ne commander que les bytes inférieurs de la RAM (soit sa première moitié)

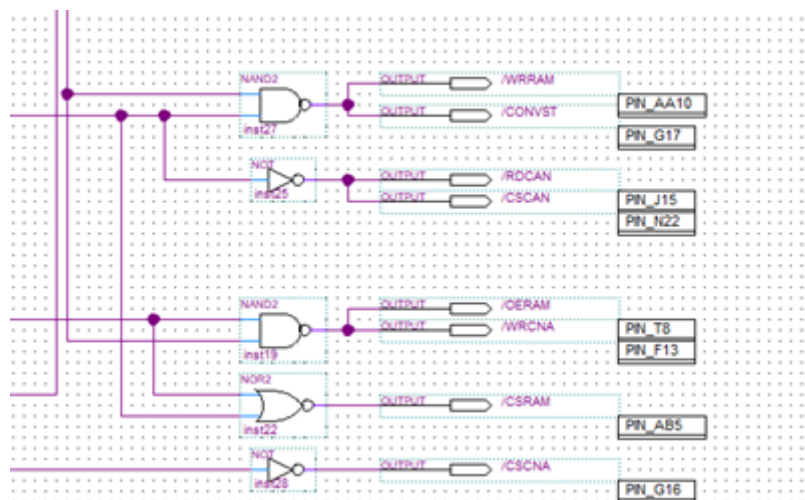
2) Ici, nous utilisons l'horloge du PIN A12 qui est une horloge 24MHz.







Pour FS4.2,4,6 et 8, nous reproduisons à l'identique la logique de commande que nous avons préalablement recablée sur proteus. Le processus est bien plus rapide que le câblage physique, et plus intuitif.



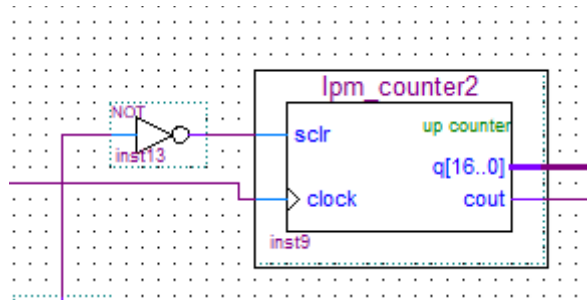
Nous reproduisons les équations logiques que nous avons décrites pour le contrôle de la RAM, du CAN et du CNA.

4) Comme observé sur la figure 1 de la question précédente, nous connectons simplement les LED de cette manière.

LED	ROUGE	VERT	JAUNE
PIN FPGA	W21	Y21	U21

5)

Pour le compteur d'adresse, nous réutilisons un bloc `lpm_counter`.



Notre horloge sera connectée à la broche clock.

La broche sclr sera connectée à FS4.2 et 4.4 (play et pause) , afin de démarrer le compteur.

les broches q sont les broches d'adresses qui seront reliées à la RAM.

Pour finir, cout signifie la fin du décompte et est reliée à la fonction RESET qui remettra à zéro et arrêtera le compte.

6&7) Test en simulation:

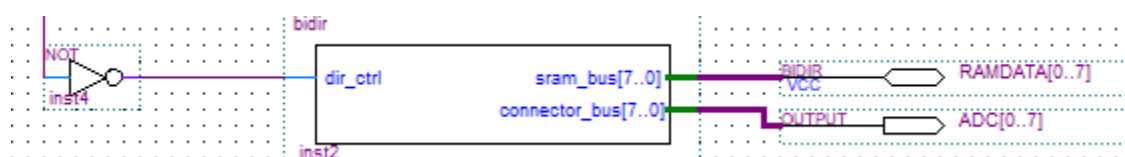
Une fois la logique de commande entièrement complétée, nous simulons, avant de faire un test fonctionnel réel.

La simulation se montre concluante, néanmoins, ne nous permet pas de tester le fonctionnement sur longue durée et donc de voir l'arrêt du comptage au bout de X secondes.

Nous passons donc à un test physique qui nous confirme le bon fonctionnement du montage.

Remarques Une des particularités du montage sur Quartus a été de faire fonctionner le bus de données de la RAM dans les deux sens, écriture et lecture.

Pour cela, nous avons utilisé un composant codé en vhdl qui dicte le sens de la marche pour le bus:



Symbole du composant

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity bidir is
4  |  port
5  |  (
6  |    sram_bus : inout std_logic_vector(7 downto 0);
7  |    connector_bus : inout std_logic_vector(7 downto 0);
8  |    dir_ctrl : in std_logic
9  |  );
10 |  end bidir;
11 |  architecture rtl of bidir is
12 |  begin
13 |
14 |    -- Tri-State Buffer control
15 |    sram_bus <= connector_bus when (dir_ctrl = '0') else (others=>'Z');
16 |    connector_bus <= sram_bus when (dir_ctrl = '1') else (others=>'Z');
17 |  end rtl;

```

Code VHDL du composant

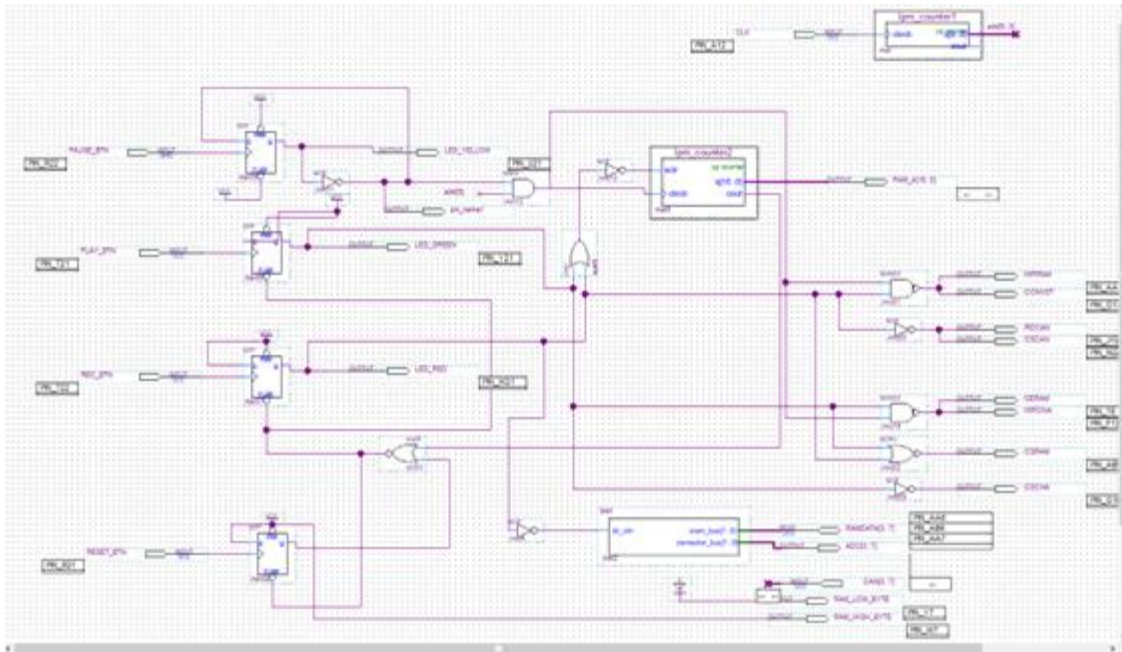
Le sens du bus sera commandé par un signal sur la broche dir\_ctrl.

Pour cela, nous devons décrire quand le bus doit être en lecture ou en écriture.

Nous remarquons que le bus doit être en lecture uniquement quand play est appuyé.

L'équation sera donc "NON PLAY".

Après cela, nous obtenons le circuit complet:



Tests physique du circuit complet:

Les tests se montrent très concluants. Après la connexion de notre amplificateur audio, du CAN et du CNA au circuit sur plaque LAB, nous obtenons un circuit plus compact, plus compréhensible et plus fiable que notre projet précédent.

De plus, l'enregistrement est fonctionnel, montrant moins d'interférences/de bruit lors de la lecture, qui peuvent être dû à la limitation des interférences générées par les nombreux câbles qu'il nous avait fallu utiliser.

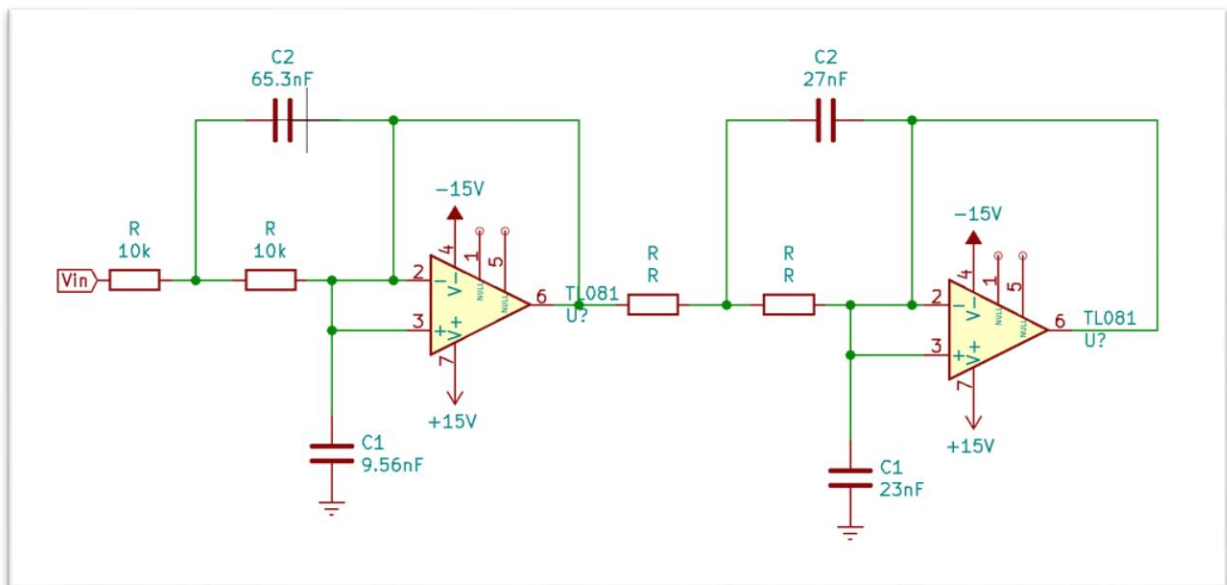
L'utilisation d'un FPGA où d'un microcontrôleur semblerait donc plus judicieuse pour la conception d'un tel système.

## **Etape 8**

a) La Fonction de transfert du Butterworth d'ordre 4 est :

$$F(P) = \frac{1}{(1 + 0,765P + P^2)(1 + 1,847P + P^2)}$$

b)



Comme on peut le voir ci-dessus, en fixant 10k pour toutes les résistances et par identification :

- C1 (cellule 1) : 9.56nF
- C2 (cellule 1) : 65.3nF
- C1 (cellule 2) : 23nF
- C2 (cellule 2) : 27nF

Nous n'avons pas pu effectuer les tests car ce filtre rajoutait beaucoup de bruit à notre montage. Les valeurs sont correctes et le montage l'était aussi (vérifié par l'enseignant).

L'analyse de nos tests n'aurait pas été pertinente au vu du bruit que générait ce filtre.

## **Conclusion**

Lors de cette SAE, nous avons pu mettre en pratique toutes les connaissances théoriques acquises durant le semestre comme l'électronique, les maths ou encore l'anglais.

Nous avons pu acquérir des connaissances et de l'expérience dans le domaine hardware mais aussi software avec le FPGA.

C'est une SAE qui nous a plus nous en sommes satisfaits car nous avons pu compléter le sujet dans son entièreté.

