INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech | LEnsE Laboratoire d'Enseignement Expérimental

Projet SONOLUX

Rapport technique

DE SALEON Victoire
GERENTE Samuel
LAPRAS Albane
REYGROBELLET Alec

Projet d'Ingénierie Electronique
pour le Traitement de l'Information
Institut d'Optique / 1A

Date de rédaction : 10/05/21 – 31/05/21

Sommaire

1. Introduction

1.1 Présentation générale

Notre projet consiste à développer un prototype d'un analyseur de fréquences sonores qui affiche la transformée de Fourier du son capté par un microphone sur des bandeaux de LEDs. Ces bandeaux forment un écran composé de cases colorées qui agrémentera le foyer de SupOptique comme sur l'image ci-dessous.

Problématique : Comment réaliser un système d'affichage en temps réel du spectre audio d'un signal sonore sur une matrice de LEDs ?

1.2 Cahier des charges

Notre dispositif devra réaliser les fonctions suivantes :

- Capter un signal sonore à l'aide d'un microphone
- Mettre en forme le signal : écrêtage de la tension analogique et amplification (gain réglable en fonction de l'amplitude du son écouté)
- Sélectionner une bande de fréquence adaptée
- Acquérir le signal d'entrée analogique à un rythme régulier compatible avec le critère de Shannon. Convertir le signal analogique en données numériques, les stocker dans la mémoire sous forme de tableau
- Calculer et normaliser la Transformée de Fourier FFT
- Mettre en forme des données pour afficher en temps presque réel le spectre audio sur les bandeaux de LEDs.

1.3 Contraintes et performances :

Certaines contraintes et performances sont à respecter :

- Le traitement numérique doit être réalisé à l'aide de cartes Nucléo L476RG.
- Le système doit permettre d'afficher le spectre d'un signal audio, en utilisant une échelle "logarithmique".
- Les cartes Nucléo L476RG n'admettent que des signaux positifs entre 0 et 3,3V et on doit leur imposer des fréquences d'échantillonnage constantes. Une mise en forme du signal sonore en entrée de la première carte Nucléo est donc nécessaire.
- L'affichage doit correspondre à une bande de fréquence (100Hz à 4kHz) incluse dans celle de l'audible.

2. Découpage fonctionnel

Au regard du schéma fonctionnel (figure 1), le projet est découpé en trois parties.

- La première concerne l'acquisition et la mise en forme du signal sonore.
- La deuxième s'intéresse principalement au calcul de la TFD et de l'intensité globale du signal au stockage de ces valeurs dans un tableau et au découpage en 1 tableau de 10 cases (microcontrôleur 1).
- La troisième porte sur la gestion de l'affichage (microcontrôleur 2) et sur l'affichage de la TFD et de l'intensité du signal sonore sur la matrice de LEDs.

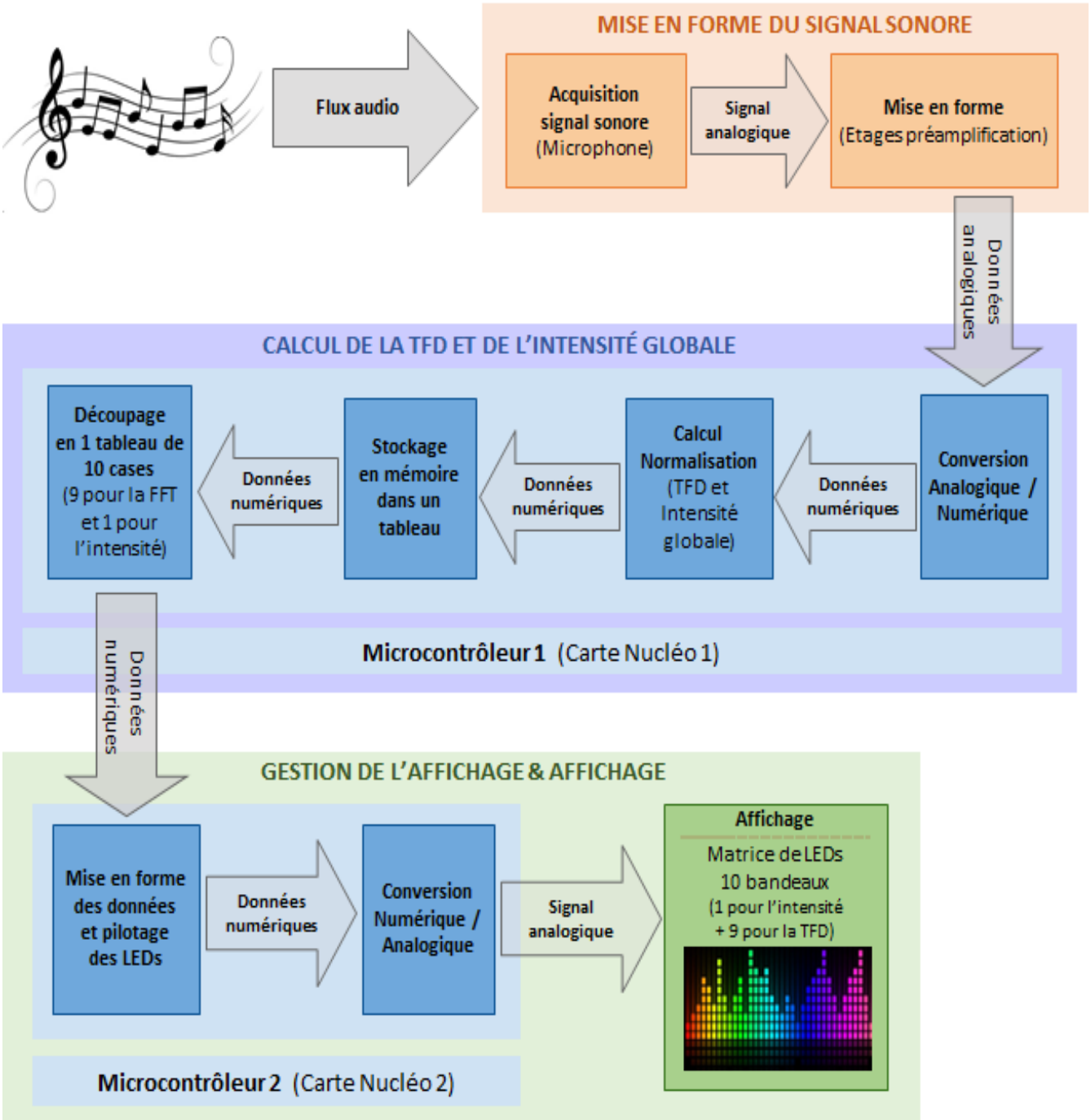


Figure 1 : *Découpage fonctionnel du SONOLUX*

3. Conception et réalisation

3.1 Bloc acquisition et mise en forme du signal sonore

a) Schéma électrique

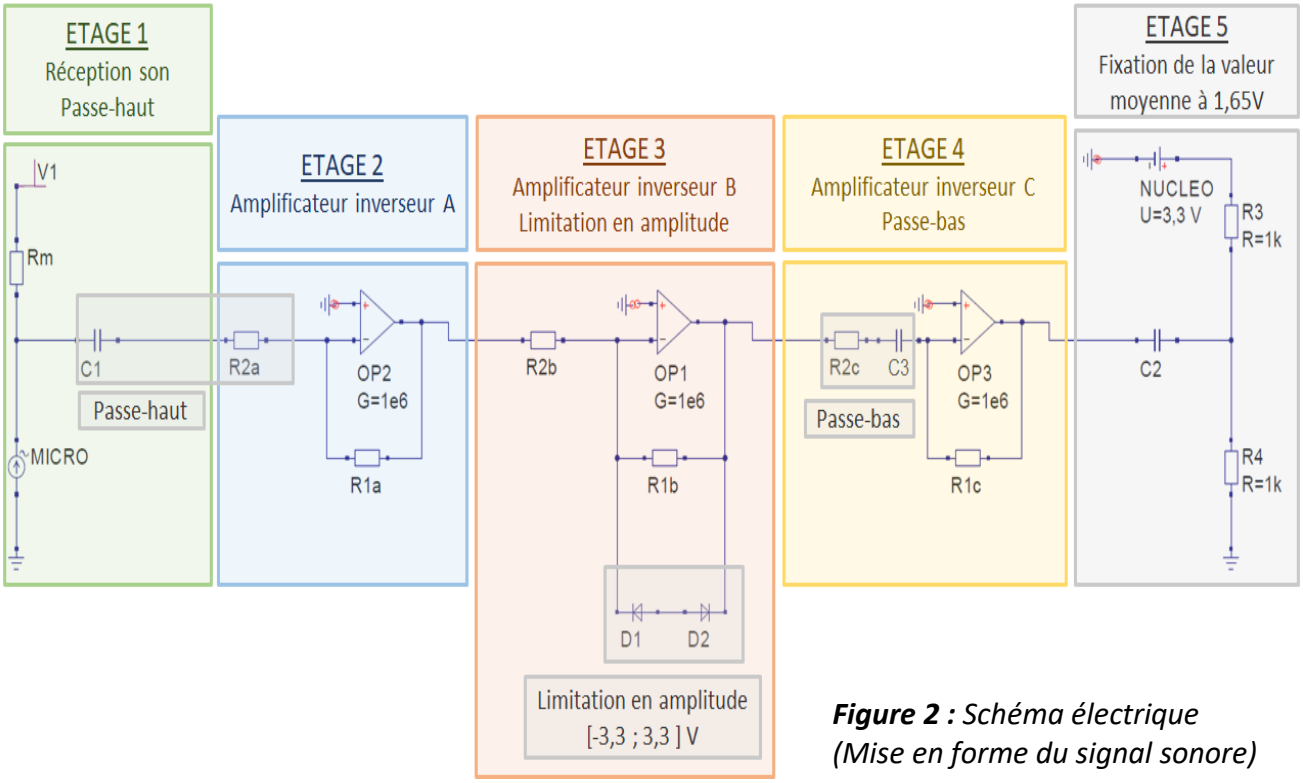


Figure 2 : *Schéma électrique (Mise en forme du signal sonore)*

ETAGE 1 **Réception son** **Passe-haut**	**ETAGE 2** **Amplificateur inverseur A**	**ETAGE 3** **Amplificateur inverseur B** **Limitation en amplitude**	**ETAGE 4** **Amplificateur inverseur C** **Passe-bas**	**ETAGE 5** **Fixation de la valeur moyenne**
V_1 = 12 V C_1 = 4,7 µF R_m =22 kΩ	R_{1a} = [0,1000] kΩ R_{2a} = 10 kΩ	R_{1b} = 1 kΩ R_{2b} = 100 Ω	R_{1c} = 1 kΩ R_{2c} = 2,1 kΩ C_3 = 5 nF	C_2 = 100 µF R_3 = R_4 = 10 kΩ
Micro : RS PRO n° 724-3134	OP2 : TL081	OP1 + OP3 : TL082 (qui intègre 2 ALIs)		

b) Description des différents étages

Afin de mettre au point le bloc mise en forme du signal sonore, nous nous sommes basés sur le schéma électrique vu en TP électronique « Thème 2 Bloc 2 » que nous avons adapté à nos contraintes en ajoutant notamment un étage supplémentaire (Amplificateur inverseur A).

- **Etage 1 : réception du signal sonore et filtrage passe-haut**

La réception du signal sonore s'effectue grâce au microphone RS PRO n° 724-3134 (diamètre 9,7mm, sensibilité -41dB).

La tension d'opération du microphone étant de 1,5V avec un courant maximal de 0,5A, nous avons donc choisi pour valeur de la résistance R_m = 22 kΩ en série avec la source continue V_1 = 12 V.
Ceci permet de fournir un courant suffisant pour alimenter le microphone :
$\mathbf{I_{MICRO}} = \frac{\mathbf{V_1 - V_0}}{\mathbf{R_m}} = \frac{\mathbf{12-1,5}}{\mathbf{22.10^3}} \approx 0{,}48\ \text{mA}$.
Cette configuration produit une composante continue de 1,5V en sortie du bloc microphone.
Afin de couper cette composante continue et ainsi de récupérer les variations de la tension (proportionnelle aux vibrations de la membrane du microphone causées par les ondes sonores) autour de cette tension continue, nous plaçons un filtre passe-haut.
Pour que le microphone fonctionne correctement, C_1 ne doit pas être trop élevé. De plus, le choix de C_1 influe sur le choix de R_{2a} de l'amplificateur inverseur A.
La fréquence de coupure du filtre passe-haut est alors : $\mathbf{f_c} = \frac{\mathbf{1}}{\mathbf{2\pi R_{2a} C_1}} \approx 3{,}4\ \text{Hz}$

- **Etage 2 : amplificateur inverseur A**

Comme les variations de tension créées par le microphone sont relativement faibles, il nous faut donc amplifier ce signal grâce à un amplificateur linéaire intégré (OP2) TL081, en mode amplificateur inverseur.
Comme on souhaite obtenir un gain variable : $\boldsymbol{G_a} = \left|-\frac{\boldsymbol{R_{1a}}}{\boldsymbol{R_{2a}}}\right| \in [0\ ;\ 100]$, on choisit les valeurs de résistances : $R_{1a} \in [0\ ;\ 1000]$ kΩ et R_{2a} = 10 kΩ.

- **Etage 3 : amplificateur inverseur B et limitation en amplitude**

Cet étage permet d'amplifier le signal en le faisant passer d'une d'amplitude maximale de 330mV à une amplitude de 3,3V centrée autour de 0V, soit une amplitude de 6,6V crête à crête.
On utilise ici un amplificateur linéaire intégré de type TL082 qui intègre 2 ALIs (celui de l'étage 3 et celui de l'étage 4).
Comme on souhaite obtenir un gain : $\boldsymbol{G_b} = \left|-\frac{\boldsymbol{R_{1b}}}{\boldsymbol{R_{2b}}}\right|$ = 10, on choisit donc les valeurs de résistances : R_{1b} = 1kΩ et R_{2b} = 100 Ω.
Les deux diodes Zener (3,3V) en contre-réaction permettent de limiter la tension à [-3,3V ; 3,3V].
Nous avons choisi une valeur de R_{1b} pas trop élevée afin que l'intensité passant dans les diodes Zener soit suffisante pour qu'elles coupent bien à 3,3V.
(Nous avions fait un essai avec R_{1b} = 100 kΩ et R_{2b} = 10 kΩ et les diodes Zener coupaient à 1,84V).

- **Etage 4 : amplificateur inverseur C (atténuation, inversion et filtrage passe-bas)**

Cet étage est nécessaire pour atténuer et inverser le signal provenant de l'étage 3 (qui peut aller jusqu'à 6,6V d'amplitude crête à crête) en réduisant son amplitude par 2 de façon à obtenir 3,3V crête à crête autour de 0V.
Ce montage amplificateur linéaire inverseur réalise donc un gain : $\boldsymbol{G_c} = \left|-\frac{\boldsymbol{R_{1c}}}{\boldsymbol{R_{2c}}}\right| = \frac{\mathbf{1}}{\mathbf{2}}$.
Ici, on a choisi R_{1c} = 1 kΩ et R_{2c} = 2,1 kΩ pour obtenir un gain voisin de 0,5.

Par ailleurs, on a ajouté un condensateur C_3 = 5nF en entrée de l'amplificateur C, afin de réaliser un filtre passe-bas de façon à couper les fréquences ajoutées par l'écrêtage et les fréquences audio élevées. Sa fréquence de coupure est : $\mathbf{f_c} = \frac{\mathbf{1}}{\mathbf{2\pi R_{2c} C_3}} \approx 10\ \text{kHz}$

- **Etage 5 : adaptation de l'amplitude à 1,65V (ou $\frac{1}{2} * 3,3V$)**

En sortie de l'étage 4, le signal a déjà l'amplitude adéquate : il est étalé sur 3,3V et peut alors bénéficier du maximum de conversion de la carte Nucléo. On peut ainsi avoir la meilleure résolution possible. Cependant, la carte Nucléo n'accepte pas les tensions négatives donc le signal qui sort de l'étage 4 ne peut pas arriver directement dans la carte au risque de l'endommager.
Ainsi il est nécessaire de mettre un offset pour obtenir un signal (Vs) positif centré autour de $1{,}65V = \frac{3{,}3}{2} V$ et variant de 0V à 3,3V.
L'étage 5 permet donc de fixer la valeur moyenne du signal à 1,65V en le rendant positif.
On utilise un pont diviseur de tension avec 2 résistances $R_3 = R_4 = 1k\Omega$ connecté à la broche 3V3 de la carte Nucléo et à la broche physique PA_1 correspondant à la broche Arduino A1 (entrée analogique), les masses de tous les étages étant reliées entre elles et reliées à la broche GND de la carte Nucléo.
En sortie de l'étage 5, nous disposons donc d'un signal audio mis en forme de façon à être compatible avec les restrictions de la carte Nucléo (signal à valeurs uniquement positives) et suffisamment amplifié sans dépasser 3,3V.

3.2 Bloc calcul de la TFD et de l'intensité globale

Ce bloc utilise une 1ère carte Nucléo L476RG qui est une carte de développement fabriquée par STMicroelectronics autour d'un microcontrôleur 32 bits. Cette carte est reliée à l'ordinateur via un câble USB.
La tension d'alimentation de la carte (et du microcontrôleur) se fait via la prise USB et vaut 3,3V.

Le programme informatique codé en langage C/C++ **(cf. Annexe 1)** compilé à l'aide du compilateur MBed (en ligne) est téléversé sur la 1ère carte Nucléo via le câble USB (liaison série - protocole RS232) pour être exécuté par le microcontrôleur de type STM32 qui impose des contraintes sur les signaux analogiques qu'il peut traiter.
Pour notre prototype, nous avons utilisé deux cartes Nucléo qui communiquent entre elles selon un protocole série. (Figure 3)

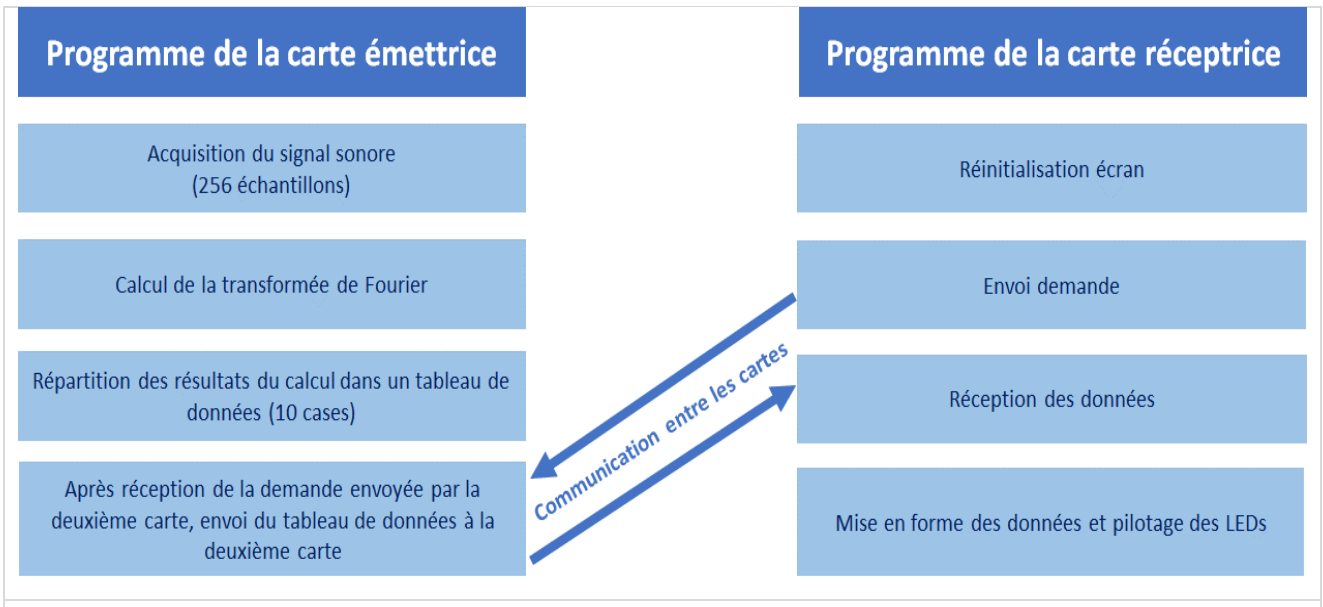


Figure 3 : *Structure des deux programmes des cartes Nucléo*

La 1ère carte Nucléo réalise donc l'acquisition du signal via l'entrée analogique et son convertisseur analogique/numérique, puis elle calcule la TFD sur 256 échantillons et elle répartit les résultats dans un tableau de données de 10 cases (9 pour la FFT et 1 pour l'intensité).
Après réception de la demande envoyée par la 2ème carte Nucléo (carte réceptrice), cette 1ère carte envoie un tableau de données à la 2ème carte. Cette configuration a permis de régler tous les problèmes de synchronisation entre les deux cartes.

Le programme téléversé dans la 1ère carte Nucléo comprend 4 étapes : (cf. Annexe 1)

a) <u>Une acquisition du signal d'entrée analogique, à un rythme régulier, compatible avec le critère de Shannon :</u>

Les systèmes échantillonnés seront ainsi capables de restituer le signal analogique si la fréquence d'échantillonnage est strictement supérieure au double de la fréquence maximale présente dans le spectre du signal à échantillonner. Le spectre sera alors contenu dans l'intervalle de fréquence [0 ; Fe/2].
La FFT s'effectue sur 256 échantillons y_j convertis et stockés dans le tableau **Input** à l'aide de la fonction Sample. Une fois ces échantillons récupérés, on affecte la valeur 0 à la variable trig ce qui permet déclencher la boucle du main. Les capacités d'échantillonnage (temps de conversion autour de 25µs) sont limitées par la programmation de ces cartes le système d'exploitation embarqué MBED. Par conséquent, on utilise un Ticker « timer » qui appelle la fonction sample à intervalle régulier.

b) <u>Calcul de la FFT :</u>

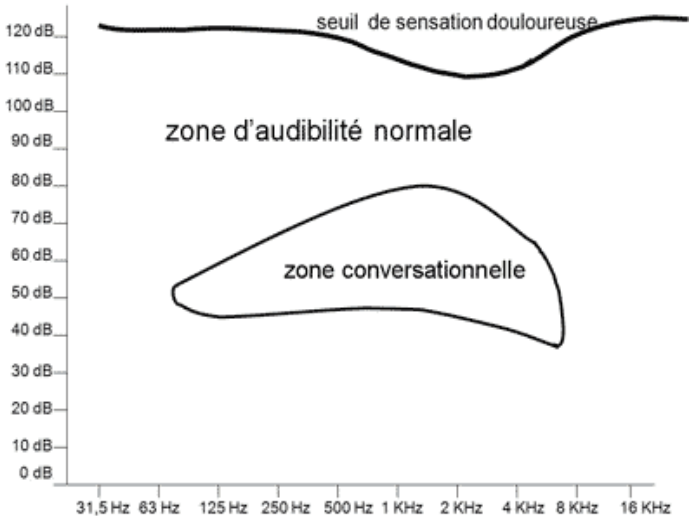
La transformée de Fourier FFT est calculée à l'aide d'un algorithme déjà implémenté dans la bibliothèque dsp.h de MBED. Ce calcul s'effectue dans la partie principale du code mais n'est exécuté que lorsque le sémaphore trig vaut 0. Puis la variable trig est remise à 1 et un appel à la fonction « sample » permet d'effectuer à nouveau un échantillonnage du signal d'entrée.
Une fois le calcul terminé, les 256 valeurs $\widetilde{y_k} = \sum_{j=0}^{255} y_j \exp\left(-2i\pi j \frac{k}{256}\right)$ avec k ϵ [0,255]
sont stockées dans le tableau **Output**.

c) <u>Répartition des résultats du calcul dans un tableau de données :</u>

➢ **Choix techniques pour le découpage fréquentiel**

- <u>Fréquences minimales et maximales :</u> nous avons choisi un intervalle de fréquences cohérent avec le domaine de l'audible : de 100 Hz à 4 kHz.

- Le découpage des fréquences doit être exponentiel / linéaire sur une échelle logarithmique.
  Ainsi on peut obtenir les valeurs de la colonne **« fréquences parfaites »** (Figure 4) en réalisant l'opération :
  $f_k = f_{min} \times 10^{\frac{(k-1)}{9} \times (\log(f_{max}) - \log(f_{min}))}$
  avec f_k : fréquence du bandeau $k \in [\![1,9]\!]$



- Fréquence d'échantillonnage :

Nos premiers tests ont été réalisés avec une **fréquence d'échantillonnage de 25 kHz**.

Cependant cette valeur de fréquence nous impose un $\Delta f = \frac{f_e}{256} = 98\ Hz$ (intervalle entre deux échantillons). Or le premier bandeau de LEDs, selon le découpage parfait devrait être affecté aux fréquences comprises entre 100 et 151 Hz.

Donc dans le **découpage 1**, le premier bandeau est au mieux forcément affecté aux fréquences comprises entre 100 et 198 Hz.

Nous avons donc décidé de passer à une **fréquence d'échantillonnage de 12,8 kHz** qui nous permet d'avoir un $\Delta f = \frac{f_e}{256} = 51\ Hz$ et donc un **découpage 2** plus juste.

				découpage 1		découpage 2
Projet SONOLUX découpage des fréquences			NB échantillons	256	NB échantillons	256
			fe	25 000	fe	12 800
			fmax	12 500	fmax	6 400
			delta f	97,66	delta f	50
			fmin	100	fmin	100
			fmax	4 000	fmax	4 000
n° bandeau	**freq parfaite**	**delta f**	**nbre ech 1**	**freq**	**nbre ech**	**freq**
1	100	51	1	100	1	100
2	151	76	1	198	1	150
3	227	115	1	295	3	200
4	342	173	1	393	3	350
5	515	261	3	491	5	500
6	776	393	4	784	8	750
7	1 170	593	6	1 174	12	1 150
8	1 762	893	9	1 760	18	1 750
9	2 655	1 345	14	2 639	27	2 650
max	4 000			4 006		4 000
		total :	40		78	

***Figure 4** : Découpage des fréquences [découpage 1 (en rose) & découpage 2 (en vert)]*

La figure 5 ci-dessous permet de comparer les deux découpages avec le découpage parfait.

On remarque que pour les fréquences les plus basses, l'écart n'est pas négligeable pour le découpage 1. En effet, un son de fréquence 250 Hz devrait apparaître sur le bandeau 3, or avec le découpage 1, il apparait dans le bandeau 2. Ce problème est réglé dans le découpage 2.

Par exemple, si on émet un son pur de 440 Hz (La 440), le 4ème bandeau s'allume. La droite en bleu représente la fréquence minimale de chacun des bandeaux (découpage parfait).

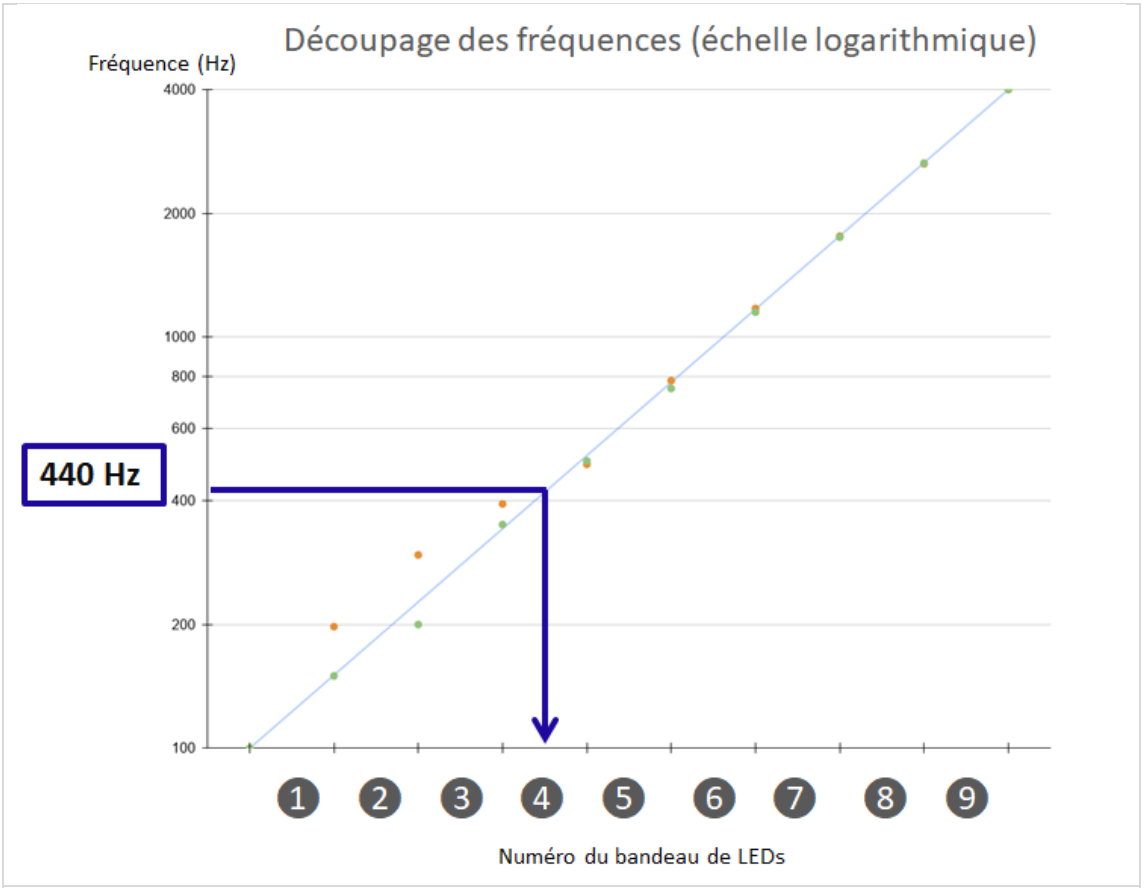


***Figure 5** : Le « La 440 » s'affiche sur le bandeau 4*
[découpage 1 (en orange) & découpage 2 (en vert)]

- **Création du tableau à 10 cases Output2**

A l'aide de la colonne « nombre d'échantillons », on crée le tableau **Output2.**
On fait la moyenne des valeurs de **Output** concernées par la plage de fréquences

Lignes de code	Nombre d'échantillons
Output2[0]=moyenne(Output,2,2);	1
Output2[1]=moyenne(Output,3,3);	1
Output2[2]=moyenne(Output,4,6);	3
Output2[3]=moyenne(Output,7,9);	3
Output2[4]=moyenne(Output,10,14);	5
Output2[5]=moyenne(Output,15,22);	8
Output2[6]=moyenne(Output,23,34);	12
Output2[7]=moyenne(Output,35,52);	18
Output2[8]=moyenne(Output,53,79);	27

***Figure 6** : Création du tableau de données*

d) Envoi du tableau de données à la deuxième carte

Lorsque la Carte 1 (micro) reçoit la demande de la Carte 2 (LEDs), la Carte 1 envoie le tableau **Output2** à la carte 2.

3.3 Bloc gestion de l'affichage et affichage sur la matrice de LEDs

a) Explication du programme de la carte Nucléo 2 : (cf. Annexe 2)

Le programme de la Carte 2 (LEDs) commence par la réinitialisation de l'écran. Puis après avoir envoyé la demande à la carte 1, elle reçoit le tableau de données.
Pour chaque bandeau, on a fait le choix d'éclairer le nombre correspondant de LEDs dans la couleur souhaitée (grâce à la documentation des LEDs, en s'aidant des codes RVB). Les couleurs des bandeaux ont été choisies en cohérence avec notre logo du projet SONOLUX.
En ce qui concerne le premier bandeau relatif à l'intensité du signal sonore, les LEDs s'éclairent en faisant un dégradé du vert (intensité plus faible) vers le rouge (intensité plus forte).

b) Conception et réalisation de la matrice de LEDs :

Notre matrice de LEDs est constituée de 10 bandeaux verticaux de LEDS. Le premier correspond à l'intensité globale du signal sonore et les 9 bandeaux suivants affichent la TFD des fréquences les plus basses aux fréquences les plus élevées (de gauche à droite). (Figure 7 et 8)

- Chaque bandeau comprend 40 LEDs espacées entre elles d'un peu moins de 2 cm.
- Ces bandeaux ont été collés sur une plaque en carton de 71 cm de haut et 60 cm de large.
- Les bandeaux sont espacés entre eux de 5 cm et les 2 bandeaux aux extrémités sont à 2,5 cm du bord.
- Tous les bandeaux sont reliés entre eux : la broche D_{out} du nième bandeau est reliée à la broche D_{in} du n+1ième bandeau.
- Toutes les masses et les sorties 5V de chacun des bandeaux sont reliées à un même potentiel sur la carte à souder.
- Les 7 premiers bandeaux sont reliés à une première carte à souder sur laquelle est soudé un connecteur, lui-même relié à l'alimentation en 5V.
- De même, les 3 derniers bandeaux sont connectés à une autre carte à souder, reliée également à une alimentation en 5V.
- La connexion avec la deuxième carte Nucléo (réception) s'effectue grâce à 2 fils (masse + signal) qui arrivent sur le premier bandeau de LEDs.

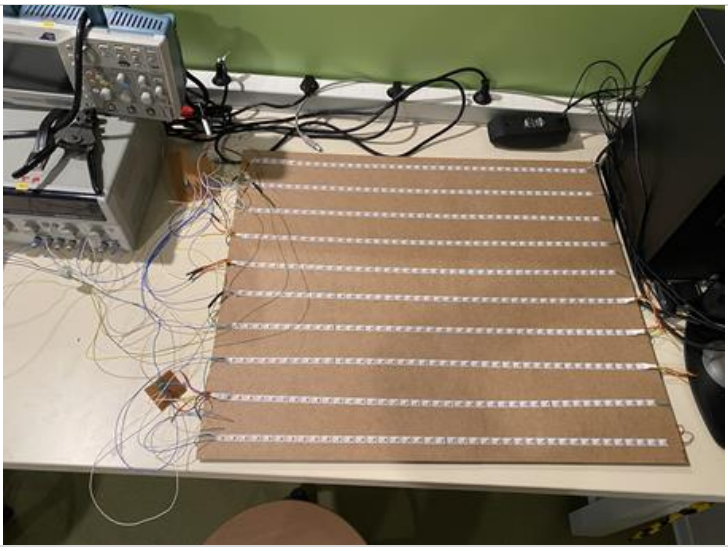


Figure 7 : *Photo de la matrice de LEDs et de ses connexions*

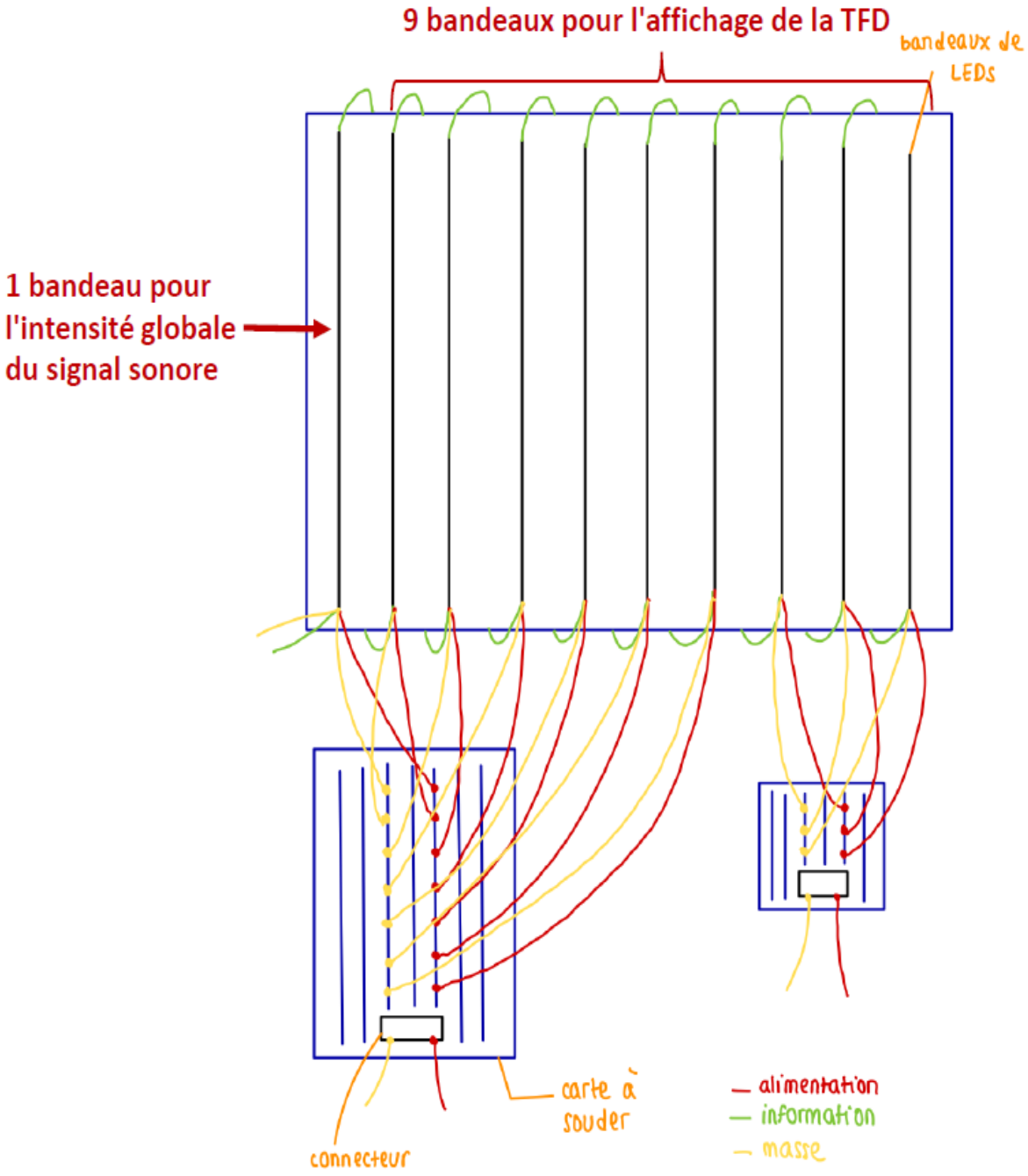
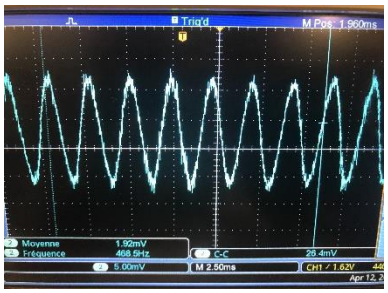


Figure 8 : *Schéma de la matrice de LEDs et de ses connexions*

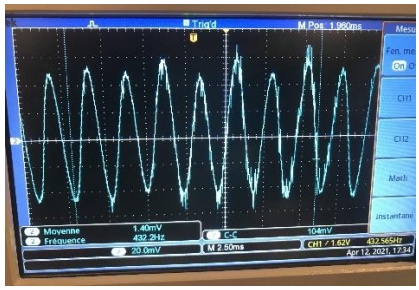
4. Tests et Validation

4.1 Tests du bloc mise en forme du signal sonore

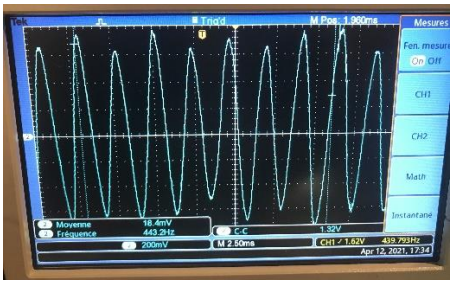
Afin de tester les différents étages de pré-amplification, nous avons visualisé sur l'oscilloscope les différents signaux obtenus en sortie de chacun des étages pour une émission d'un son pur de fréquence 440Hz (La 440). (Figure 9) Ceci nous a permis d'ajuster les valeurs de certains composants afin d'obtenir en sortie de l'étage 5, un signal positif suffisamment amplifié sans dépasser 3,3V.
En particulier, nous avons fait varier le gain de l'amplificateur inverseur B en modifiant les valeurs de R_{1b} et R_{2b} et nous avons bien constaté que lorsque le gain était trop élevé, les diodes Zener écrêtaient le signal. Nous avons donc bien choisi nos résistances de façon à obtenir un gain qui permette d'étaler le signal sur les 6,6V d'amplitude crête à crête sans les dépasser pour ne pas perdre de l'information.



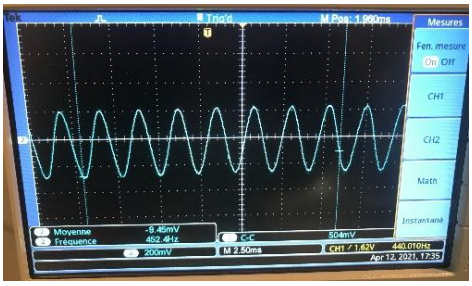
(a) *en sortie du microphone*



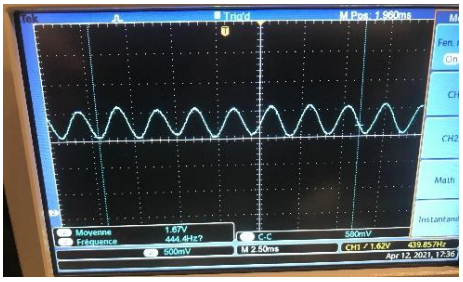
(b) *après l'amplificateur inverseur A*



(c) après l'amplificateur inverseur B



(d) après l'amplificateur inverseur C

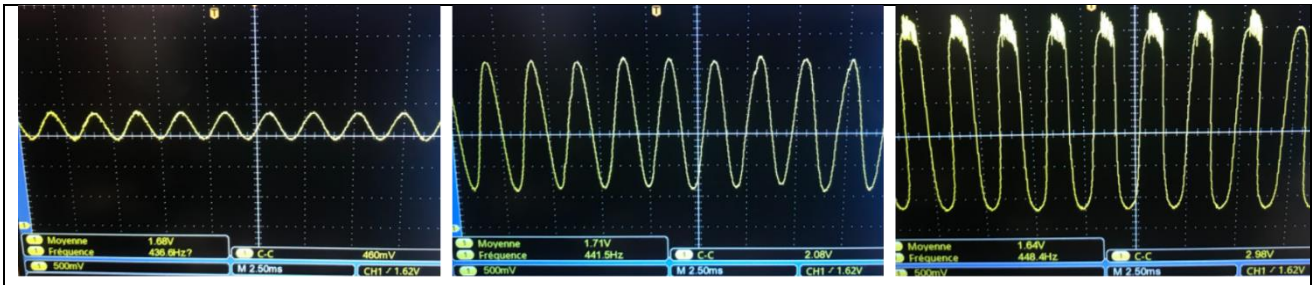


(e) en sortie de l'étage 5

Figure 9 : *Forme du signal après chaque étage*

Récapitulatif des tests des blocs de pré-amplification (tensions de sortie)				
Bloc microphone (réception du son)	Amplificateur inverseur A	Amplificateur inverseur B	Amplificateur inverseur C	En sortie du montage
Valeurs attendues				
	Moyenne : 0	Moyenne : 0 Crête à crête max : 6,66 V	Moyenne : 0 Crête à crête max : 3,33 mV	Moyenne : 1,5 V Crête à crête max : 3,33 mV
Valeurs affichées par l'oscilloscope				
Moyenne : 1,92 mV Crête à crête : 26,4 mV	Moyenne : 1,40 mV (proche de 0) Crête à crête : 104 mV	Moyenne : 18,4 mV (proche de 0) Crête à crête : 1,32 V	Moyenne : -9,54 mV (proche de 0) Crête à crête : 504 mV	Moyenne : 1,67 V Crête à crête : 580 mV

Comme expliqué précédemment, nous avons souhaité que le gain de l'amplificateur A soit réglable, afin que notre Sonolux puisse s'adapter à différentes ambiances musicales (petite musique sur le téléphone, ou gros concert dans le foyer). On visualise sur l'oscilloscope le signal en sortie de l'ampli A pour un son d'intensité constante et on fait varier la valeur de la résistance R_{1a} pour observer les situations auxquelles on peut être confronté si le Sonolux n'a pas été correctement réglé.



gain trop faible	gain correctement réglé	gain trop élevé (saturation)

Figure 10 : *Modification du gain de l'amplificateur A*

4.2 Tests du bloc calcul de la TFD et de l'intensité

Avant de visualiser notre TFD sur la matrice de LEDs, nous avons commencé par afficher dans Tera Term les données qui circulent entre les deux cartes Nucléo pour faire nos premiers tests du bloc de calcul de la TFD et de l'intensité. On constate bien que les valeurs reçues par la carte Nucléo réceptrice correspondent aux valeurs envoyées par la carte Nucléo émettrice.

Le nombre de LEDs par bandeaux pour un LA 440 est représenté ci-dessous (Figure 11). La 1ère valeur du tableau de données correspond au nombre de LEDs allumées pour le bandeau intensité globale et les 9 valeurs suivantes aux 9 bandeaux de fréquences. On observe que le nombre maximal de LEDs s'allument pour le 4e bandeau « fréquence » (5e valeur du tableau de données envoyé à la carte Nucléo « réceptrice » : 40), ce qui est cohérent avec la théorie.

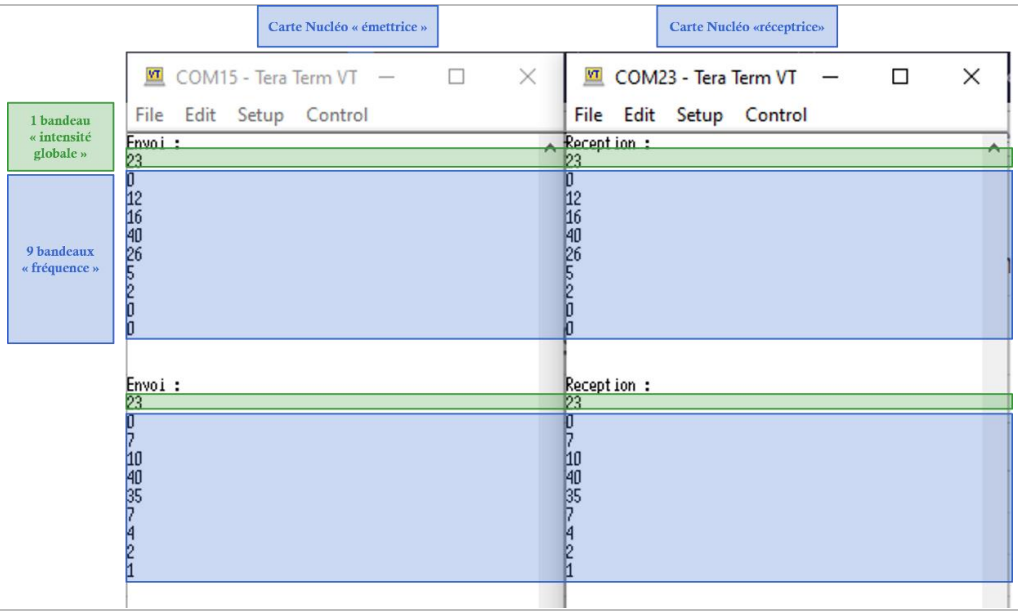


Figure 11 : *Valeurs affichées dans Tera Term lorsqu'on émet un LA 440*

4.3 Test du bloc gestion de l'affichage et visualisation sur la matrice de LED

Avant d'afficher la TFD sur la matrice de LEDs, nous avons testé quelques petits programmes simples de gestion d'affichage sur un bandeau de LEDs (consistant par exemple à allumer une LED sur 10 dans différentes couleurs) afin de prendre en main ces nouvelles commandes.
En raison de faux contacts et de connexions mal soudées, nous avons dans un premier temps rencontré quelques difficultés à visualiser les figures attendues.

4.4 Validation globale du prototype

fréquences test (Hz)
130
210
320
440
700
1000
1500
2100
2800

Pour tester l'exactitude de nos programmes, nous avons émis des sons purs à des fréquences précises (1 par bandeau de fréquence : tableau ci-contre), pour vérifier que les bonnes LEDs s'allumaient (cf figure 12).

C'est lors de ces tests que nous avons constaté le problème du « mauvais » découpage en fréquence évoqué dans le 3.2 c) lié à la fréquence d'échantillonnage.

Enfin nous avons pu tester notre prototype sur différentes musiques. (cf figure 13)

Figure 12 : *Photo de la matrice de LEDs pour un son de fréquence 1500 Hz*

Figure 13 : *Photo de la matrice de LEDs analysant la musique Bohemian Rapsody de Queen*

Remarque : Nous n'avons pas rencontré de problèmes de vitesse de transmission de données au niveau des LEDs car nous avons observé une vitesse de rafraichissement des LEDs assez importante et car nous avons choisi un fonctionnement adéquat de communication entre les deux cartes.

5. Gestion du projet

5.1 Planning

Quand	Principales étapes	Tutoriaux acquis
25/01/21 **(Séance 1)**	• Découverte du sujet et prise en main du matériel • Création d'un espace partagé pour nos documents • Ecriture du 1er cahier des charges • Réflexion sur les premières briques à mettre en place • Réalisation des premiers blocs du prototype	• Connecter une source sonore (2) • Faire une action à intervalle régulier (2) • Caractériser un traitement numérique (2)
01/02/21 **(séance 2)**	• Calcul et affichage de la FFT sur l'oscilloscope • Ecriture du code pour tester l'affichage de l'intensité globale du son sur un bandeau de LEDs • Modification des étages de préamplification pour que le gain soit réglable • Finalisation du cahier des charges	• Régler la luminosité d'une LED (2) • Calculer le spectre d'un signal en temps réel (4) • Faire des actions à intervalle régulier (2)
10/03/21 **(séance 3)** 24/03/21 **(séance 4)**	• Finalisation et tests des étages de préamplification • Choix d'utiliser 2 cartes Nucléo (la 1ère, dédiée à la réception du signal et au calcul de la TFD et de l'intensité globale et la 2ème dédiée au pilotage des 10 bandeaux de LEDs) • Ecriture du code (carte Nucléo1) pour un affichage sur 10 bandeaux de LEDs verticaux [9 bandeaux pour la TFD (chacun affecté à une plage de fréquence différente) + 1 bandeau affecté à l'intensité globale] • Conception de la matrice de LEDs (choix de l'écartement entre bandeaux…) • Choix du découpage de fréquences pour l'écriture du code de réception (2ème carte Nucléo)	• Calculer le spectre d'un signal en temps réel (4) • Filtrer une bande fréquentielle / spectrale (3) • Filtrer une bande fréquentielle plus efficacement (3) • Configurer une communication point à point RS232 (2)
12/04/21 **(séance 5)**	• Ecriture du code de la carte Nucléo 2 & correction des bugs • Réalisation de la matrice de LEDs	
10/05/21 **(séance 6)**	• Finalisation de la matrice de LEDs • Finalisation des codes • Phase de tests	
19/05/21 **(séance 7)**	• Ajout de la gestion du bruit de fond dans le code de 2ème carte Nucléo • Validation et tests finaux du prototype • Finalisation du support de présentation et rédaction du rapport technique	
26/05/21 **(séance 8)**	• Forum avec démonstration et présentation orale	

5.2 Difficultés rencontrées

Au cours du projet, nous avons rencontrés différents problèmes que nous avons réussi à régler.

Par exemple, nous avons eu quelques soucis pour trouver une bonne manière pour faire communiquer les deux cartes sans bugs et en restant synchronisé. A un moment donné, alors que tout fonctionnait sauf la ligne de code qui commande les LEDs, lorsqu'on rajoutait cette ligne de code plus rien ne marchait. On a mis un peu de temps à trouver que ce dysfonctionnement venait en fait d'un problème d'indice qui faisait un reset des cartes.

Par ailleurs, lors de la réalisation de la matrice de LEDs, nous avons eu des soucis de faux contacts en particulier pour le 1 er bandeau dont les LEDs clignotaient et ne s'allumaient pas de la façon attendue. Nous avons remédié à ce problème, en refaisant toutes les soudures de ce bandeau.
Par ailleurs, les 3 derniers bandeaux étaient allumés en permanence car les 2 masses des alimentations n'étaient pas correctement reliées entre elles.

D'autre part, en raison du 1er découpage en fréquence, les bandeaux de LEDs qui s'allumaient n'étaient pas ceux attendus. Nous avons donc trouvé un 2ème découpage plus adéquat.

5.3 Analyse du travail d'équipe

Lors des premières séances du projet, nous avons beaucoup travaillé tous ensemble : notamment, sur le cahier des charges, les étages de pré-amplification, puis les premiers codes. Cette étape a été nécessaire pour bien définir notre projet, nos objectifs communs et pour mettre en place une communication transparente et partagée dans un répertoire GoogleDrive.
Ensuite, nous avons plus fonctionné par binôme pour plus d'efficacité, par exemple tandis qu'un groupe travaillait sur la conception et réalisation de la matrice de LEDs, l'autre binôme s'employait à débugger le code, en particulier la communication entre les deux cartes. Comme nos profils étaient assez complémentaires, nous avons rapidement trouvé les bonnes associations pour avancer dans la même direction.
Après chaque séance, nous rédigions un petit compte-rendu sur un document partagé, ce qui permettait de faire un bilan de nos principales réalisations en séance et d'anticiper le travail à faire sur les prochaines étapes, que ce soit pendant les séances au LEnsE ou en dehors.
Globalement, nous avons eu une bonne ambiance de travail au sein de notre groupe sans doute car nous avons établi une bonne communication et nous avons su être très transparents les uns avec les autres.
Lorsqu'un problème délicat se présentait, nous avons su réfléchir ensemble et nous entraider.
Nous ne sommes pas enfermés dans des rôles prédéfinis et une répartition des tâches trop rigide, nous avons préféré l'union, l'adaptabilité, la flexibilité et une grande transparence pour plus de créativité et d'efficacité.
Ce projet nous a permis d'être de plus en plus autonomes et à l'écoute les uns des autres, attentifs aux différentes remarques de chacun pour converger vers une solution optimale.
Ainsi, nous n'avons pas été freinés par nos différences mais nous avons réussi à nous coordonner et à avancer dans la même direction en unissant nos forces, nos aptitudes et nos compétences.
Nous sommes très fiers de ce que nous avons accompli ensemble.

6. Conclusion

Ce projet s'est avéré instructif à la fois en termes de travail de groupe et d'innovation (gestion de la communication entre les 2 cartes Nucléo, conception et réalisation d'une matrice de LEDs,…). Nous avons atteint l'objectif fixé tout en gagnant peu à peu en confiance et en autonomie.

Par ailleurs, sur la matrice de LEDs de notre prototype, le bandeau d'intensité globale du signal sonore peut permettre à l'utilisateur de régler le gain en modifiant la résistance R_{1a} de façon à éclairer suffisamment de LEDs (pour arriver au maximum de LEDs éclairées dans le 1er bandeau).

Si on devait améliorer notre prototype, on pourrait le rendre plus esthétique en ajoutant par exemple un panneau diffuseur devant notre matrice de LEDs. On pourrait également réaliser une communication sans fil entre les 2 cartes Nucléo pour avoir un affichage déporté par rapport au microphone. On pourrait peut-être aussi avoir une interface pour les utilisateurs afin qu'ils puissent modifier la plage de fréquences à afficher.

Enfin, nous tenons à remercier les équipes du LEnsE pour leur aide et pour nous avoir permis de mener à bien ce projet au cours duquel nous avons tant appris.

7. Annexes

Annexe 1 : Code du microcontrôleur 1 (Carte Nucléo 1)

« Calcul de la TFD et de l'intensité globale »

Annexe 2 : Code du microcontrôleur 2 (Carte Nucléo 2)

« Gestion de l'affichage sur la matrice de LEDs »

Annexe 1 : Code du microcontrôleur 1

```
#include "mbed.h"
#include "arm_math.h"
#include "dsp.h"
#include "arm_common_tables.h"
#include "arm_const_structs.h"
#define SAMPLES             512             /* 256 real party and 256 imaginary parts */
#define FFT_SIZE            SAMPLES / 2     /* FFT size is always the same size as we have samples,
                                               so 256 in our case */

#define NB_LED  40
#define intensite_mini 10
#define intensite_max 10

float32_t Input[SAMPLES];
float32_t Output[FFT_SIZE];
float32_t Output2[9];
double maxi = 0;
bool     trig=0;                            // sémaphore de blocage de l'échantillonnage
int      indice = 0;
AnalogIn   myADC(A1);
Serial     s(A4, A5);
Serial pc(USBTX, USBRX);
Ticker     timer;
char chc;
int led[9];



void sample(){
   if(indice < SAMPLES){

      Input[indice] = myADC.read() - 0.5f;          //Real part NB removing DC offset
      Input[indice + 1] = 0;                        //Imaginary Part set to zero
      if(Input[indice]>maxi) maxi=Input[indice];

      indice += 2;

   }
   else{ trig = 0; }
}

float32_t moyenne(float32_t t[], int i0, int i1)   {
   float32_t sum=0;
   int i;

   for(i=i0; i<=i1; i++) {  sum=sum+t[i];  }

   return sum/(i1-i0+1);
   }
```

```
int main() {
    float maxValue;                         // Max FFT value is stored here
    uint32_t maxIndex;                      // Index in Output array where max value is
    pc.baud(115200);


    while(1) {
        if(trig == 0){
            timer.detach();
            // Initialisation du calcul de la FFT
            // NB utilisation de fonctions prédéfinies arm_cfft_sR_f32_lenXXX, où XXX est le nombre
               d'échantillons, ici 256
            arm_cfft_f32(&arm_cfft_sR_f32_len256, Input, 0, 1);

            // Calcul de la FFT et stockage des valeurs dans le tableau Output
            arm_cmplx_mag_f32(Input, Output, FFT_SIZE);
            Output[0] = 0;
            // Calcul la valeur maximale du spectre - maxValue - et son indice - maxIndex
            arm_max_f32(Output, FFT_SIZE/2, &maxValue, &maxIndex);

            //s.printf("%lf ", maxi);
            //pc.printf("maxi : %lf\n", maxi);


           if (s.readable()) {
               /*Output2[0]=moyenne(Output,1,1);
               Output2[1]=moyenne(Output,2,2);
               Output2[2]=moyenne(Output,3,3);
               Output2[3]=moyenne(Output,4,5);
               Output2[4]=moyenne(Output,6,8);
               Output2[5]=moyenne(Output,9,12);
               Output2[6]=moyenne(Output,13,18);
               Output2[7]=moyenne(Output,19,27);
               Output2[8]=moyenne(Output,28,41); //ancien découpage pour fe=25kHz*/

               Output2[0]=moyenne(Output,2,2);
               Output2[1]=moyenne(Output,3,3);
               Output2[2]=moyenne(Output,4,6);
               Output2[3]=moyenne(Output,7,9);
               Output2[4]=moyenne(Output,10,14);
               Output2[5]=moyenne(Output,15,22);
               Output2[6]=moyenne(Output,23,34);
               Output2[7]=moyenne(Output,35,52);
               Output2[8]=moyenne(Output,53,79);


               chc=s.getc();  // Lecture dans stockage du caractère envoyé comme marqueur par la carte de
                                 commande

               s.putc(char(NB_LED*maxi*2)); // maxi est un float entre 0 et 0.5 correspond à l'intensité sonore
                                               maximale
```

```
            for(int i=0; i<9; i++) {
                led[i] = int(Output2[i]*2);  // Le *2 permet un ajustement du niveau maximal de l'affichage des
                                                fréquences. Si on diminue le facteur multiplicatif, on affiche moins
                                                de LEDs
                if(led[i]>=40)
                  s.putc(char(40));
                else
                  s.putc(char(led[i]));
              }

        /*  Envoi des données dans la liaison avec le PC pour debugage   */
            /*pc.printf("Envoi :\n");
            pc.printf("%d\n", int(NB_LED*maxi*2));
            for(int i=0; i<9; i++) {
                if(led[i]>=40)
                  pc.printf("%d\n", 40);
                else
                  pc.printf("%d\n", led[i]);

              }

            pc.printf("\n\n");*/

            maxi=0;
          }

          trig = 1;
          indice = 0;
          //timer.attach_us(&sample,40);     //40us 25KHz sampling rate
          timer.attach_us(&sample,78);     //78us 12.8KHz sampling rate
        }
      }
    }
```

Annexe 2 : Code du microcontrôleur 2

```
/* Déclaration des ressources externes */
#include "mbed.h"
#include "PixelArray.h"
#include "WS2812.h"

#define NB_LED_colonne  40
#define NB_bandeau      10
#define NB_LED          NB_LED_colonne*NB_bandeau

/* Déclaration des variables globales */

/* Déclaration des entrées/sorties */
DigitalOut  myled(LED1);
Serial      s(A4, A5);
Serial      pc(USBTX, USBRX);

PixelArray px(NB_LED);          // Tableau de pixels / Nb de LEDs en cascade
// For Nucleo F476 : 3, 12, 9, 12
WS2812 ws(D7, NB_LED, 3, 12, 9, 12);    // Lien vers la série de LEDs / Sortie utilisée-Nb de LEDs
// NE PAS MODIFIER LES 4 DERNIERS PARAMETRES

int LED_maxi;
int LED[9];
char chc;

//Timer t;  Timer utilisé pour mesurer la fréquence d'actualisation des LEDs



void SetColor(int pixel, int colonne) {
   if(colonne == 1) {
      px.SetR(pixel, 255);
      px.SetG(pixel, 209);
      px.SetB(pixel, 0);
      }
   if(colonne == 2) {
      px.SetR(pixel, 190);
      px.SetG(pixel, 255);
      px.SetB(pixel, 0);
      }
   if(colonne == 3) {
      px.SetR(pixel, 58);
      px.SetG(pixel, 255);
      px.SetB(pixel, 0);
      }
   if(colonne == 4) {
      px.SetR(pixel, 0);
      px.SetG(pixel, 255);
      px.SetB(pixel, 135);
      }
```

```
  if(colonne == 5) {
      px.SetR(pixel, 0);
      px.SetG(pixel, 240);
      px.SetB(pixel, 255);
      }
   if(colonne == 6) {
      px.SetR(pixel, 85);
      px.SetG(pixel, 0);
      px.SetB(pixel, 255);
      }
   if(colonne == 7) {
      px.SetR(pixel, 178);
      px.SetG(pixel, 0);
      px.SetB(pixel, 255);
      }
   if(colonne == 8) {
      px.SetR(pixel, 255);
      px.SetG(pixel, 0);
      px.SetB(pixel, 131);
      }
   if(colonne == 9) {
      px.SetR(pixel, 255);
      px.SetG(pixel, 0);
      px.SetB(pixel, 0);
      }
   }



/* Fonction principale */
int main(){

   ws.useII(WS2812::GLOBAL);          // Initialisation de la liaison avec les LEDs
   ws.setII(20);                       // Luminosité maximale à 20/255



   // Reinitialisation ecran
   for (int i = 0; i < NB_LED; i++) {
      px.Set(i, 0);
   }
   ws.write(px.getBuf());             // Mise à jour de la matrice

   pc.baud(115200);
```

```
while(1){

    /*  Réception des données    */
    s.putc('a');    //envoie un caractère pour demander à la carte d'acquisition d'envoyer des données

    while(s.readable()==0) {__nop();}      //attend de recevoir des données

    chc=s.getc();
    LED_maxi=int(chc); //recupère le 1er caractère et le stock

    for (int i=0; i<9; i++) {
       while(s.readable()==0) {__nop();}    //attend de recevoir des données
       chc=s.getc();
       LED[i]=chc;                          // récupère les caractères suivant et les stocke
    }



    /* Envoi des données dans la liaison avec le PC pour debugage   */

    /*pc.printf("R :\r\n");
    pc.printf("%d\t", LED_maxi);
    for (int i=0; i<9; i++) {
       pc.printf("%d\t", LED[i]);
    }
    pc.printf("\r\n");*/


    /* Commande du 1er bandeau de LED correspondant à l'intensité sonore globale    */

    //allume les LEDs jusqu'à LED_maxi
    for(int j = 0; j < LED_maxi; j++){
       if(j<7) {

          px.SetR(j, int(255*j/6));                 //permet de faire un dégradé du vert au rouge
          px.SetG(j, 255);
          px.SetB(j, 0);
          }
       else {
          px.SetR(j, 255);
          px.SetG(j, int(255-(j-7)*255/32));        //permet de faire un dégradé du vert au rouge
          px.SetB(j, 0);
          }
    }

    //éteint les LEDs après LED_maxi
    for (int j = LED_maxi; j < NB_LED_colonne ; j++) {
       px.Set(j, 0);
    }
```

```
/* Commande du 1er bandeau de LED correspondant à l'intensité sonore globale    */


      for(int i=1; i<10; i++) {
         for(int j = 0; j < LED[i-1]; j++){
            SetColor(j+i*NB_LED_colonne, i);
         }
         for (int j = LED[i-1]; j < NB_LED_colonne ; j++) {
            px.Set(j+i*NB_LED_colonne, 0);
         }
      }


      //t.stop();
      //pc.printf("%d\n", t.read_ms());
      //t.reset();
      ws.write(px.getBuf());  //Mise à jour de la matrice
      //t.start();

   }
}
```