



Aoustin Emma, Denis Angèle, Kouba Simon, Muzard Sarah

Introduction

Le Sonolux est un panneau de LEDs qui réagit en temps réel à la musique. A terme, l'objectif serait de l'installer dans le foyer de l'Institut d'Optique.

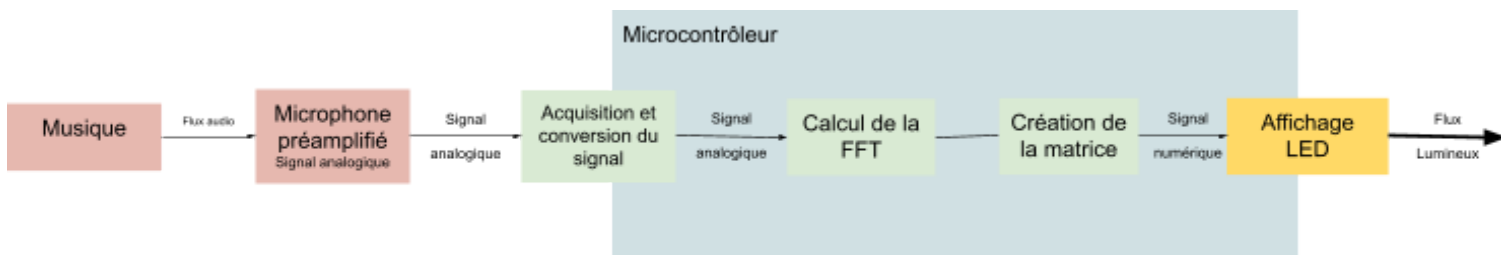
Le système devra être capable de mettre en forme un signal audio et de calculer sa Transformée de Fourier à l'aide d'une carte Nucleo. Ensuite, le système stockera le résultat du calcul de FFT dans une matrice que l'on découpera en plusieurs bandes de fréquences. Après avoir moyenné la Transformée de Fourier sur chaque plage de fréquence, le système devra être capable de l'afficher sur un panneau de Leds. L'aspect esthétique et le choix des couleurs entrent aussi dans notre cahier des charges, le Sonolux doit être agréable et décoratif.

I. Comprendre le projet

Avant de commencer à travailler sur le projet, il a fallu d'abord le comprendre et choisir les caractéristiques que nous voulions donner au Sonolux.

D'abord, nous avons choisi de récupérer le signal audio par un microphone préamplifié ou directement avec l'ordinateur via une prise Jack. Nous avons aussi choisi de découper notre panneau de LEDs en sept bandeaux des sept couleurs de l'arc-en-ciel.

Nous avons résumé le fonctionnement du Sonolux avec le schéma suivant :



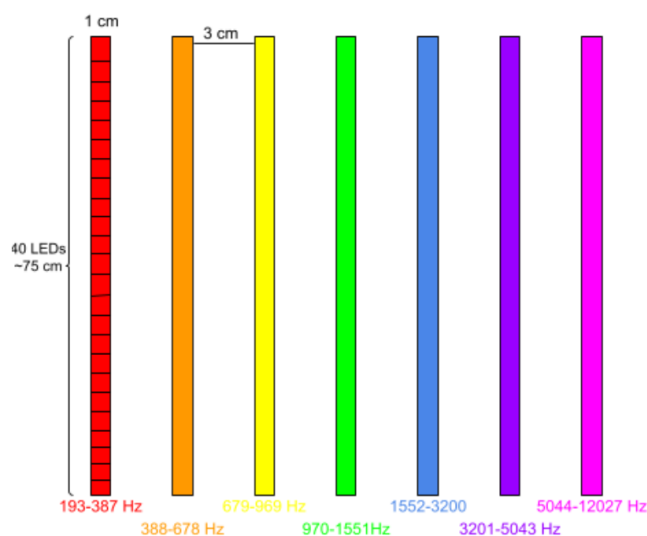
D'abord la musique écoutée au foyer est captée par un microphone préamplifié qui transforme le flux audio en signal analogique. Ce signal est alors envoyé au système électrique décrit plus bas, afin de le rendre compatible à notre microcontrôleur (carte Nucléo). Par un algorithme aussi décrit dans la suite du rapport, le microcontrôleur calcule la Transformée de Fourier du signal analogique et crée une matrice de sept plages de fréquences correspondant aux sept bandeaux de LEDs de notre Sonolux. Cette matrice, qui correspond à un signal numérique, est envoyée au panneau de LEDs afin d'afficher la Transformée de Fourier par un flux lumineux.

La deuxième étape avant de commencer le projet était de déterminer la fréquence d'échantillonnage de la Transformée de Fourier et les différentes plages de fréquence.

Le calcul de la Transformée de Fourier utilisé au premier semestre avait une fréquence d'échantillonnage de 25 kHz. Nous l'avons gardée car notre projet ne nécessitait pas un calcul plus rapide.

Le découpage en sept plages de fréquence a été effectué à partir de spectres d'une musique quelconque avec des instruments comme la guitare, le clavier ou la trompette. Nous avons ensuite pris soin d'ajuster ces valeurs de telle sorte qu'elles soient des multiples de $\Delta\nu = \frac{25kHz}{256} = 97 Hz$ afin de rendre le code le plus simple possible. $\Delta\nu$ correspond à l'écart de fréquence entre deux cases du tableau dans lequel les calculs de la FFT sont stockées.

Nous sommes arrivés à ce résultat :



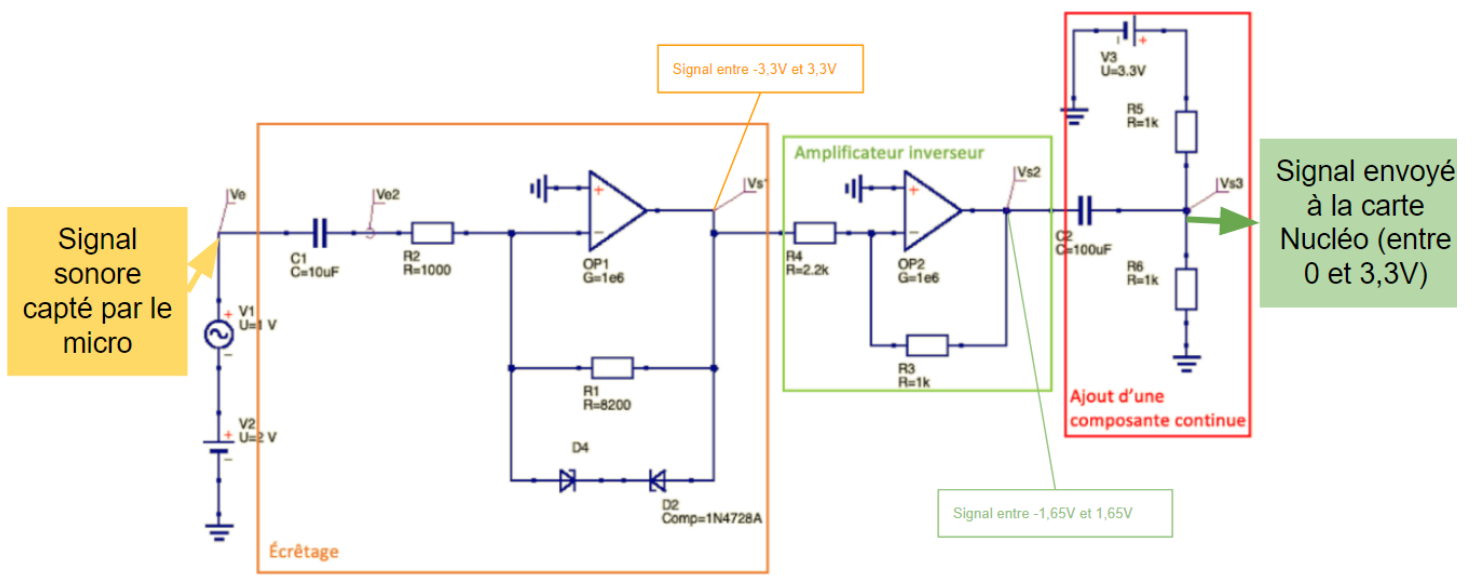
II. Réaliser le prototype

1 - Le schéma électrique

Afin de réaliser la transformée de Fourier du signal audio par la carte Nucléo, nous avons dû utiliser un dispositif électronique qui permet de mettre en forme le signal.

En effet, la carte Nucléo ne peut réaliser l'acquisition seulement de signaux oscillants de 0 à 3,3V. Nous devons donc obtenir un signal analogique de sortie de même fréquence que le signal analogique d'entrée mais ne variant que de 0 à 3,3V.

Nous réalisons donc un dispositif qui va pouvoir réaliser les objectifs du cahier des charges, on réalise le montage suivant en trois blocs distincts :



Le premier bloc "écrêtage" permet d'amplifier le signal et donc d'avoir une oscillation entre $-3,3V$ et $3,3V$.

Le deuxième bloc est un bloc d'amplification inverse. On envoie un signal en entrée qui oscille entre $-3,3V$ et $3,3V$, on veut en sortie de ce bloc avoir un signal qui oscille entre la moitié de $3,3V$, soit entre $-1,65V$ et $1,65V$.

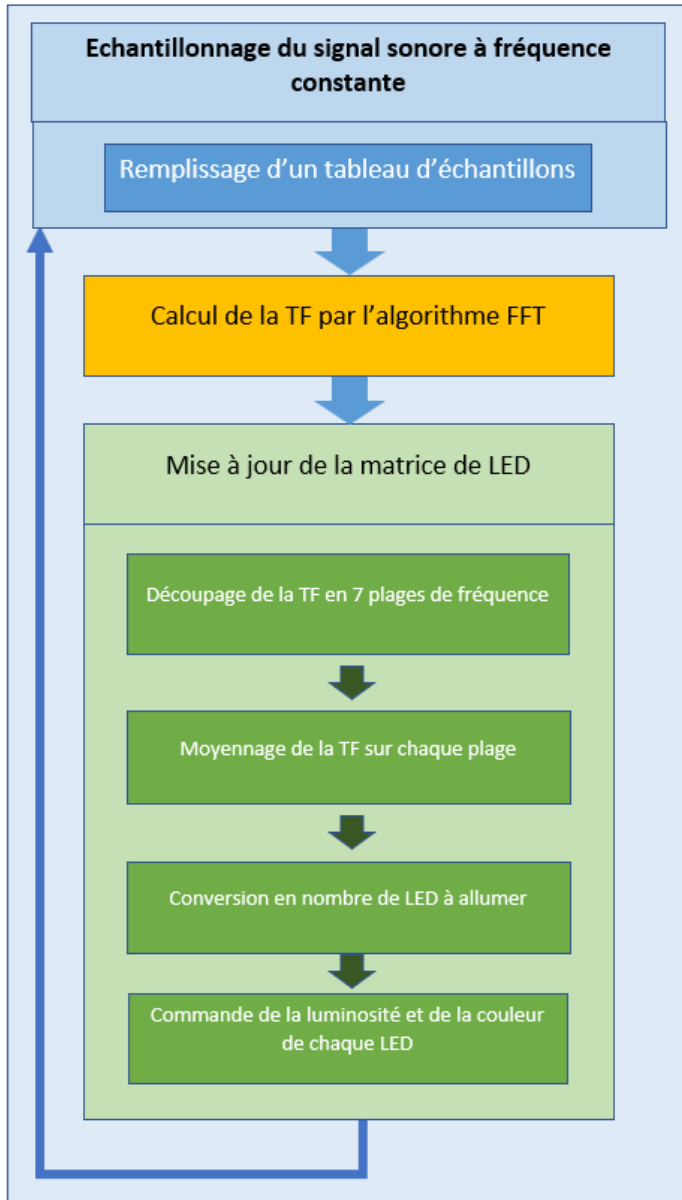
Le troisième bloc a pour objectif d'obtenir le signal final oscillant entre 0 et $3,3V$, à la même fréquence que le signal d'entrée. Tout d'abord, le condensateur rencontré à l'entrée de ce bloc permet de supprimer une éventuelle composante continue qui aurait pu être ajoutée par les amplificateurs opérationnels. On obtient donc un signal oscillant entre $\pm 1,65V$. Le signal continu à $3,3V$ va être coupé en deux par les deux résistances formant un pont diviseur et forme donc une composante continue à $+1,65V$. On ajoute les deux signaux et on obtient un signal oscillant entre 0 et $3,3V$.

Nous avons aussi eu besoin de câbler un microphone, et de l'alimenter par une tension continue de $5V$.

Pour le panneau à LEDs, nous avons soudé les différents bandeaux de LEDs en série. Nous avons alors pu utiliser une seule alimentation de $5V$ pour tous les bandeaux.

2 - L' algorithme

Le signal en entrée de la carte Nucléo est traité de sorte que le microcontrôleur en calcule la Transformée de Fourier et commande l'affichage des LEDs en conséquence. L'algorithme utilisé se décompose en deux parties distinctes : le calcul de la Transformée de Fourier puis sa mise en forme (*voir schéma*).



La Transformée de Fourier du signal est calculée avec l'algorithme Fast Fourier Transform (FFT). Il est donc échantillonné à fréquence constante grâce à une commande par interruption. Cette partie de code a été fournie au premier semestre dans le cadre des TP de Céli.

Le tableau contenant les valeurs de la Transformée de Fourier est alors moyenné sur sept plages de fréquences. Ces sept valeurs moyennes sont ensuite multipliées par un coefficient propre aux plages de fréquences pour les convertir en un nombre de LEDs à allumer par bandeau.

La commande est ensuite envoyée aux LEDs dont on modifie les valeurs des proportions RGB. Les couleurs des bandeaux suivent les couleurs de l'arc en ciel, du rouge au violet, le rouge étant associé aux basses fréquences (la moyenne des premières cases du tableau de la Transformée de Fourier) et le violet aux hautes fréquences (la moyenne des dernières cases du tableau).

III. Valider et caractériser le système final

1 - Test de l'algorithme

Pour caractériser les bornes des plages de fréquence des différents bandeaux de LEDs, on utilise un GBF que l'on va brancher en entrée du montage. On envoie un signal sinusoïdal dont on connaît la fréquence et on regarde quel bandeau s'allume et avec quelle hauteur. Aux bords de la plage d'un bandeau, un petit nombre de LEDs est allumé et en milieu de la plage, un grand nombre s'allume.

On peut donc trouver approximativement les fréquences centrales des différents bandeaux de LEDs. Les résultats sont regroupés dans le tableau suivant :

Couleur du bandeau	Rouge	Orange	Jaune	Vert	Bleu	Indigo	Violet
Fréquence centrale (en Hz)	265	500	815	1.2K	2.2K	4.2K	9.2K

Ces résultats correspondent plutôt bien au découpage prévu par l'algorithme. Nous avons fait ces tests avec la petite matrice de LEDs (8x8).

Les valeurs des coefficients propres aux plages de fréquence ont ensuite été trouvées et ajustées par tâtonnement, de sorte que le nombre de LEDs allumées soit harmonieux, ni trop grand (bandeau toujours allumé), ni trop petit (bandeau toujours éteint).

De plus, le processus de calcul de la FFT et la commande des LEDs doivent impérativement prendre moins de temps que T_{ech} , la période d'échantillonnage, ce qui impose une limite à l'actualisation des bandeaux de LEDs. Nous avons utilisé $T_{ech} = 40 \mu s$ mais les LEDs s'actualisent seulement toutes les 1 ms car bien que nous pourrions actualiser plus fréquemment, la vitesse de clignotement des LEDs devient désagréable.

2 - Test du panneau à LEDs

Nous avons soudé en série 8 bandeaux de 40 LEDs. Nous avons dans un premier temps branché cette matrice à notre système exactement comme nous l'avons fait pour la petite matrice. Nous avons rencontré un problème avec l'alimentation : la carte nucléo ne peut envoyer plusieurs ampères. Nous avons donc alimenté la matrice avec un générateur à 5V et de plusieurs ampères.

Cependant les couleurs sur les bandeaux ne correspondaient dans un premier temps pas à celles que nous voulions à cause de faux contacts aux niveaux des branchements. Il a donc fallu boucher tous les câbles afin de localiser le problème et de rebrancher proprement les câbles.

Nous avons ensuite refait la recherche des valeurs des coefficients propres aux plages de fréquences de la même manière que précédemment, car nous avons changé significativement le nombre de LEDs.

Nous avons ensuite cherché la bonne intensité à envoyer dans la matrice de LEDs, car la luminosité des LEDs augmente avec l'intensité jusqu'à un certain stade où les LEDs se mettent à afficher des couleurs qui ne sont pas voulues.

IV. Comprendre les étapes de réalisation

1 - Planning et rétro-planning

Séance 1 :

- ❖ Emma et Angèle : redécouverte du code de la FFT
- ❖ Simon et Sarah : Compréhension du code des bandeaux de leds + prise en main du bandeau led

Séance 2 :

- ❖ Emma et Angèle : Schéma de principe et finir le cahier des charges
- ❖ Simon et Sarah : Découpage des fréquences pour les bandeaux

Livrable intermédiaire

Séance 3 :

- ❖ Sarah et Simon : Mettre le résultat de la FFT dans un tableau
- ❖ Angèle et Emma : Séparer le tableau en 7 parties en fonction du découpage des fréquences et moyenner la FFT sur une plage de fréquence donnée et pertinente

=> en plus, test des plages de fréquences avec un bandeau de LEDs, mesure des limites des plages de fréquences expérimentales (on regarde quand les leds commencent à s'allumer et quand elles s'éteignent en fonction des différentes fréquences) : test avec un GBF et pas le micro

Séance 4 :

- ❖ Emma : Câbler le micro les deux cartes NUCLEO aux bandeaux
- ❖ Simon et Angèle : Code qui sépare dans une matrice (7 lignes pour les 7 bandeaux et 3 colonnes pour le RGB) pour allumer les sept bandeaux simultanément -on prendra une moyenne de référence de test-
- ❖ Sarah : dégradé de couleurs pour chaque bandeau

=> câblage simplifié avec le son qui sort de l'ordinateur et un bandeau à led (le micro ne fonctionnait plus)

câblage avec le montage en entier et nouveau micro

insertion d'une matrice de LEDs 8x8

test avec le micro et la matrice et définition des coefficients pour chaque bande de fréquences

Séance 5 :

- ❖ Simon et Emma : Choix des fréquences d'échantillonnage pour les 2 cartes NUCLEO et du nombre de points pour optimiser le temps de calcul et choix de la fréquence où l'on change l'affichage sur les bandeaux
- ❖ Sarah et Angèle : Commencer à commander les bandeaux avec les moyennes de la FFT en temps réel

=> passer de la matrice de led à 7 bandeaux de 40 leds, branchées en série (soudage)

finalement, on ne prend qu'une seule carte Nucléo, car on se rend compte qu'on arrive bien à commander les différentes plages de fréquences

Séance 6 :

- ❖ Emma et Sarah Choix de la moyenne de référence optimale ou on change à chaque fois
- ❖ Simon et Angèle : Finir commande des bandeaux avec les moyennes de la FFT en temps réel (si on finit en avance on pourra commencer à réfléchir à comment récupérer le volume sonore pour régler la luminosité)

=> audit

câbler le montage en entier avec notre version de led 40x7

voir si on peut améliorer les leds en faisant un dégradé de couleur pour chaque bandeau et en réglant la luminosité en fonction du volume sonore

Séance 7 : 2 options (tous ensemble)

- ❖ Rattrapage de retard
- ❖ Bonus : luminosité en fonction du volume sonore

=> refaire fonctionner le montage, le brancher à la matrice de 7 bandeaux de 40 leds. Nous avons renoncé à faire varier la luminosité en fonction du volume sonore. Début de rédaction des livrables.

Séance 8 : Présentation du projet

Livrables finaux

2 - Difficultés rencontrées

Durant toute la durée du projet, nous avons rencontré diverses difficultés.

Dans un premier temps, la complexité du montage électronique nous a parfois posé problème. En effet, dès qu'il y avait un problème de branchement, de faux contacts, de composants qui ne fonctionnait plus, c'était assez difficile de trouver où était exactement le problème. Dans ces cas, nous prenions le temps d'observer à l'oscilloscope le signal après chaque étape du montage, afin d'identifier dans quelle partie était le problème.

De même, la longueur de l'algorithme pouvait parfois poser problème pour identifier une erreur.

Lorsque nous avons créé le panneau à LEDs final, nous avons dû faire des soudures entre chaque bandeau de LEDs, malheureusement certaines de nos soudures avaient des faux contacts. C'est pourquoi quand nous avons branché ce panneau, le premier bandeau ne fonctionnait pas toujours, nous avons dû refaire les soudures.

Enfin, lors du passage de la matrice de 7x8 LEDs à 7x40 LEDs, nous avons eu un problème d'alimentation. En effet, nous avons beaucoup plus de LEDs à alimenter et l'alimentation de 5 V de la carte Nucléo ne convenait plus, il fallait passer à une alimentation externe.

3 - Analyse du travail d'équipe

Le travail en équipe s'est très bien déroulé, dès le début nous avons réussi à nous séparer équitablement le travail. Simon et Sarah se sont concentrés sur comprendre le fonctionnement des bandeaux à LEDs et modifier l'algorithme de Transformée de Fourier pour rajouter les bandeaux de LEDs. Nous avons tous travaillé en même temps pour séparer la Transformée de Fourier en 7 planches de fréquences, que ce soit les fréquences à choisir ou la modification de l'algorithme. Angèle et Emma se sont plutôt concentrées sur le montage électronique et la conception du panneau final.

Ce projet nous a permis d'être de plus en plus autonome dans notre travail, nous avons aussi dû être à l'écoute de chaque membre du groupe, attentif aux différentes remarques et ainsi gérer la communication à la fois à l'intérieur du groupe et avec les autres groupes qui travaillaient sur le même projet que nous.

Conclusion / points à améliorer :

Nous avons réussi à la fin de ces semaines de projet à créer et à faire fonctionner le Sonolux. Il remplit sa fonction principale qui est d'afficher la Transformée de Fourier du signal sonore en temps réel sur le panneau de Leds.

Cependant, le projet n'est pas réellement utilisable. Il faudrait améliorer sa portabilité pour pouvoir l'utiliser dans le foyer. De plus, nous utilisons un GBF pour l'alimentation, ce qui n'est pas non plus pratique pour le déplacer. Finalement, nous avons pensé à des améliorations possibles pour notre projet : nous pourrions par exemple créer des dégradés de couleurs sur chaque bandeau de Leds ou encore régler l'intensité lumineuse en fonction du volume sonore.

Annexe

Le code commenté suivant a été utilisé avec Mbed 5 pour la version finale de notre projet.

```
/* Bibliothèques utiles */
```

```
#include "mbed.h"  
#include "arm_math.h"  
#include "math.h"  
  
#include "dsp.h"  
#include "arm_common_tables.h"  
#include "arm_const_structs.h"  
  
#include "PixelArray.h"  
#include "WS2812.h"
```

```
/* Définition des variables utiles */
```

```
#define NB_LED 280 // Nombre de LEDs en cascade  
#define L 40 // Hauteur d'une bande de LED  
// Variables globales  
PixelArray px(NB_LED); // Tableau de pixels / Nb de LEDs en cascade  
  
// For Nucleo F476 : 3, 12, 9, 12  
WS2812 ws(D9, NB_LED, 3, 12, 9, 12); // Lien vers la série de LEDs / Sortie utilisée-Nb de LEDs  
// NE PAS MODIFIER LES 4 DERNIERS PARAMETRES  
  
#define SAMPLES 512  
// 256 échantillons pour la FFT : 256 parties réelles et 256 parties imaginaires  
#define FFT_SIZE SAMPLES / 2  
// FFT_SIZE a toujours la même taille que le nombre d'échantillons : 256 ici  
#define T_ech 40  
// T_ech correspond à la période d'échantillonnage !\ T_ech est en µs  
  
float32_t Input[SAMPLES]; // tableau des échantillons  
float32_t Output[FFT_SIZE]; // tableau de la TF  
bool trig=0;  
int indice = 0;  
  
double r = 1 / (T_ech*FFT_SIZE);  
// résolution en fréquence: intervalle de fréquence entre 2 cases consécutives du tableau Output : environ 97 Hz ici  
  
int tab_divise[7]; // tableau des différentes plages de fréquence  
  
DigitalOut myled(LED1);  
AnalogIn myADC(A0);  
Serial pc(USBTX, USBRX);  
Ticker timer;
```

```
/* Fonction d'échantillonnage, appelée à une fréquence constante T_ech */
```

```
void sample(){  
    myled = 1;  
    if(indice < SAMPLES){  
        Input[indice] = myADC.read() - 0.5f; // La partie réelle de l'échantillon auquel on enlève l'offset
```

```

    Input[indice + 1] = 0;           // La partie imaginaire est nulle pour le signal en entrée
    indice += 2;
}
else{ trig = 0; }
myled = 0;
}

```

/ Fonction principale */*

```

int main() {

    float maxValue; // maxValue contient la valeur maximale de la FFT (utile pour une normalisation)
    uint32_t maxIndex; // Indice du tableau Output contenant maxValue
    int i;
    ws.useI(WS2812::GLOBAL); // Initialisation de la liaison avec les LEDs
    ws.setI(0); // Luminosité maximale initialisée à 0/255

    /* Réinitialisation de l'écran : toutes les LED s'éteignent */

    for (i = 0; i < NB_LED; i++) {
        px.Set(i, 0);}
    ws.write(px.getBuf()); // Mise à jour de la matrice
    ws.setI(120); // Luminosité maximale à 30/255

    /* Boucle infinie : coeur du programme */

    while(1) {
        if(trig == 0){
            timer.detach();

            /* Calcul de la FFT avec les fonctions de la bibliothèque arm_math.h */

            // Init the Complex FFT module, intFlag = 0, doBitReverse = 1
            //NB using predefined arm_cfft_sR_f32_lenXXX, in this case XXX is 256
            arm_cfft_f32(&arm_cfft_sR_f32_len256, Input, 0, 1);

            // Complex Magnitude Module put results into Output(Half size of the Input)
            arm_cmplx_mag_f32(Input, Output, FFT_SIZE);
            Output[0] = 0;
            //Calculates maxValue and returns corresponding value
            arm_max_f32(Output, FFT_SIZE/2, &maxValue, &maxIndex);

            /* Découpage du tableau Output en 7 plages de fréquence et moyennage sur les plages */

            int rang=1;
            int rang_suivant;
            int tab_decoupage[8] = {193,387,678,969,1551,3200,5043,12027};
            // Les différentes plages de fréquence : [193 Hz, 387 Hz], ..., [5043 Hz, 12027 Hz]
            int tab_coef[7]={40,20,40,40,50,80,250}; // tableau des coefficients
            for (int k = 0; k < 7; k++){
                double s=0;
                rang_suivant = rang + (tab_decoupage[k+1]-tab_decoupage[k]) /97;
                for (int i = rang ;i < rang_suivant ; i++){
                    s = s + Output[i]*tab_coef[k];
                }
                rang = rang_suivant ;
                tab_divise[k] = (int)(s/((tab_decoupage[k+1]-tab_decoupage[k])/ 97));
            }

```

```
/* On plafonne le nombre de LED à allumer à L (40) sur une bande */
```

```
for (int l=0; l<7; l++){  
    if (tab_divise[l]>L) {  
        tab_divise[l]=L;  
    }  
}
```

```
/* On allume le premier bandeau en rouge */
```

```
for(int j = 0; j < tab_divise[0]; j++){  
    px.SetR(j, 255); px.SetG(j, 0); px.SetB(j,0);}  
for (i = tab_divise[0]; i < L; i++){  
    px.Set(i, 0);} 
```

```
/* On allume le deuxième bandeau en orange */
```

```
for(int j = L; j < L+tab_divise[1]; j++){  
    px.SetR(j, 255); px.SetG(j, 50); px.SetB(j,0);}  
for (i = L+tab_divise[1]; i < 2*L; i++){  
    px.Set(i, 0);} 
```

```
/* On allume le troisième bandeau en jaune */
```

```
for(int j = 2*L; j < 2*L+tab_divise[2]; j++){  
    px.SetR(j, 255); px.SetG(j, 210); px.SetB(j,0);}  
for (i = 2*L+tab_divise[2]; i < 3*L; i++){  
    px.Set(i, 0);} 
```

```
/* On allume le quatrième bandeau en vert */
```

```
for(int j = 3*L; j < 3*L+tab_divise[3]; j++){  
    px.SetR(j, 0); px.SetG(j, 190); px.SetB(j,10);}  
for (i = 3*L+tab_divise[3]; i < 4*L; i++){  
    px.Set(i, 0);} 
```

```
/* On allume le cinquième bandeau en bleu */
```

```
for(int j = 4*L; j < 4*L+tab_divise[4]; j++){  
    px.SetR(j, 0); px.SetG(j, 255); px.SetB(j,255);}  
for (i = 4*L+tab_divise[4]; i < 5*L; i++){  
    px.Set(i, 0);} 
```

```
/* On allume le sixième bandeau en indigo */
```

```
for(int j = 5*L; j < 5*L+tab_divise[5]; j++){  
    px.SetR(j, 130); px.SetG(j, 0); px.SetB(j,255);}  
for (i = 5*L+tab_divise[5]; i < 6*L; i++){  
    px.Set(i, 0);} 
```

```
/* On allume le septième bandeau en violet */
```

```
for(int j = 6*L; j < 6*L+tab_divise[6]; j++){  
    px.SetR(j, 255); px.SetG(j, 0); px.SetB(j,175);}  
for (i = 6*L+tab_divise[6]; i < 7*L; i++){  
    px.Set(i, 0);} 
```

```
ws.write(px.getBuf()); // On met à jour l'allumage des LED  
wait_us(1000); //toutes les 1000 µs
```

```
/* liaison PC série */
```

```
pc.printf("Output[3] = %lf, tab_divise[0] = %d \r\n", Output[3], tab_divise[0]) ;
```

```
trig = 1;  
indice = 0;
```

```
timer.attach_us(&sample,T_ech);// Interruption pour un échantillonnage à fréquence constante  
}  
}  
}
```