SAE22 : mesurer et caractériser un signal

Synthèse

Orlando MURAT et Tommy Malande

Sommaire

Introduction :

Au cours de ce semestre, plusieurs projets nous sont proposés afin d'évaluer nos compétences sur les différents modules que nous avons eu tout au long de l'année. Ces dernières semaines, nous avons travaillé sur le projet « SAÉ22 ». L'objectif de ce projet est de faire passer un message qui sera transmis par les fréquences successives d'un air de musique et déchiffré avec un Arduino. Ce message est un code secret composé de chiffres codés sur 7 bits.

Pour faire ce projet, nous avons utilisé :

- Une carte d'Arduino Uno avec sa plaquette
- Un oscilloscope et GBF
- Des LEDS et des résistances.

Le travail est fait en binôme et le délai pour finir ce projet est estimé à 22 heures.

Travail préliminaire :

Nous avons effectué plusieurs recherches concernant les fréquences des touches d'un piano. Nous avons pris la note LA comme fréquence de référence, c'est-à-dire 440 Hz car elle se situe au milieu du clavier.

	Octave 0	Octave 1	Octave 2	Octave 3	Octave 4	Octave 5	Octave 6	Octave 7	Octave 8	Octave 9	
do	32.70	65.41	130.8	261.6	523.3	1046.5	2093.0	4186.0	8372.0	16 744	do
$do_{\#}$ $ré_{b}$	34.65	69.30	138.6	277.2	554.4	1108.7	2217.5	4434.9	8869.8	17 740	$do_{\#}$ $ré_{b}$
ré	36.71	73.42	146.8	293.7	587.3	1174.7	2349.3	4698.6	9397.3	18 795	ré
$ré_{\#}$ mi_{b}	38.89	77.78	155.6	311.1	622.3	1244.5	2489.0	4978.0	9956.1	19 912	$ré_{\#}$ mi_{b}
mi	41.20	82.41	164.8	329.6	659.3	1318.5	2637.0	5274.0	10 548	21 096	mi
fa	43.65	87.31	174.6	349.2	698.5	1396.9	2793.8	5587.7	11 175	22 351	fa
$fa_{\#}$ sol_{b}	46.25	92.50	185.0	370.0	740.0	1480.0	2960.0	5919.9	11 840	23 680	$fa_{\#}$ sol_{b}
sol	49.00	98.00	196.0	392.0	784.0	1568.0	3136.0	6271.9	12 544	25 088	sol
$sol_{\#}$ la_{b}	51.91	103.8	207.7	415.3	830.6	1661.2	3322.4	6644.9	13 290	26 580	$sol_{\#}$ la_{b}
la	55.00	110.0	220.0	440.0	880.0	1760.0	3520.0	7040.0	14 080	28 160	la
$la_{\#}$ si_{b}	58.27	116.5	233.1	466.2	932.3	1864.7	3729.3	7458.6	14 917	29 834	$la_{\#}$ si_{b}
si	61.74	123.5	246.9	493.9	987.8	1975.5	3951.1	7902.1	15 804	31 609	si

Figure 1: Tableau des fréquences de touche d'un piano

Environnement ARDUINO-UNO :

Pour la suite de ce projet, nous avons utilisé un Arduino Uno. La carte Arduino Uno possède 6 entrées analogiques, reliées à un convertisseur analogique/numérique qui renvoie un code numérique sur 10 bits. Les entrées sont repérées sur la carte par les broches A0 à A5. La masse est noté GND sur l'Arduino Uno. Nous avons ensuite récupéré le programme "Acquisition_v1.ino" sur moodle, ce programme permet de générer un signal sinusoïdal de fréquence d'échantillonnage de 2 kHz.

FFT dans l'environnement ARDUINO :

Nous avons récupéré sur GitHub le programme « FFT_03.ino », ce programme permet d'utiliser la bibliothèque FFT pour calculer la FFT d'un signal échantillonné. Nous avons utilisé la FFT dans ce projet afin de traiter directement les fréquences de la mélodie et non l'intensité. La bibliothèque FFT permet à l'Arduino de faire de l'analyse de fréquence de donnée audio. Dans le programme FFT, plusieurs paramètres doivent être pris en compte, tels que : la fréquence d'échantillonnage, la période d'échantillonnage, les pas d'échantillons. Nous avons modifié notre programme pour obtenir la meilleure fréquence de la note La, nous avons mis une fréquence d'échantillonnage à 3000 Hz. Après avoir fait ceci, nous avons obtenu une fréquence de 443 Hz. Ensuite, nous avons complété le tableau ci-dessous avec les différentes fréquences détectées.

Validation 1

Touche	Fréquence	Fréquence détectée	Broche
Ré2	147	147,25	2
Mi2	165	167,38	3
Fa2	175	180,87	4
Sol2	196	196,52	5
La2	220	226,34	6
Si2	247	247,65	7
Do3	262	265 ,45	8
Ré3	294	295,32	9
Mi3	330	334,30	10

Figure 2: Tableau des fréquences détectées pour chacune des notes

Script MATLAB :

Validation 2

Nous avons fait un script Matlab qui permet de générer la note LA, dans ce script nous avons mit une fréquence d'échantillonnage à 8 kHz. Une période d'échantillonnage 1/8000. Le temps est de 0 à 1 seconde à pas de période d'échantillonnage. Ensuite, nous avons générer un signal sinusoïdal :

*A = sin(2*pi*440*t)*

Pour écouter ce son, il faut utiliser la fonction *sound(A,8000).*

Validation 3

Par la suite, nous avons écrit le programme qui permet de jouer un morceau de musique donné en annexe et de l'enregistrer dans un fichier de type ".wav" dont nous avons défini, la fréquence d'échantillonnage et le temps de référence pris sur la note noire

Environnement ARDUINO : Visualisation des notes

Nous avons branché LEDs sur les sorties numériques 2 à 10. Tout en respectant les broches données dans le tableau de la validation 1. Après nous avons modifié le programme précédent afin que les LEDs s'allument selon la note jouée :

Validation 4

```
for (int i=2; i<11; i++){
digitalWrite(i, LOW);              Permet d'éteindre toutes les LED's
}


if (x > 145.0 && x < 148.0) {
  digitalWrite(2, HIGH);    Permet d'allumer la LED de la branche 2 si la fréquence entre 145 et 148 Hz
} else if (x > 160 && x < 170){
  digitalWrite(3, HIGH);    Permet d'allumer la LED de la branche 3 si la fréquence entre 160 et 170 Hz
} else if ( x > 178 && x < 182) {
  digitalWrite(4, HIGH);
} else if ( x > 194 && x < 198) {
  digitalWrite(5, HIGH);
} else if (x > 224 && x < 230){
  digitalWrite(6, HIGH);
} else if (x > 245 && x < 250){
  digitalWrite(7, HIGH);
} else if (x > 265 && x < 280){
  digitalWrite(8, HIGH);
} else if (x > 293 && x < 296) {
  digitalWrite(9, HIGH) ;
} else if (x > 330 && x < 336) {
  digitalWrite(10, HIGH);
}
```

Figure 3: Modification apporter au programme afin que les LEDs s'allument selon la note jouée

Validation 5 et 6

Nous avons vérifié avec le morceau de musique créé sur Matlab si notre programme fonctionne correctement. Ensuite, nous avons modifié le programme pour que la Built-in LED s'allument lorsque les notes DO3, MI3 et RE3 sont reconnues :

```
} else if (x > 265 && x < 280){
 digitalWrite(LED_BUILTIN, HIGH); Permet d'allumer la Built-in LED lorsque la note Do3 est reconnue
} else if (x > 293 && x < 296) {
 digitalWrite(LED_BUILTIN, HIGH); Permet d'allumer la Built-in LED lorsque la note Ré3 est reconnue
} else if (x > 330 && x < 336) {
 digitalWrite(LED_BUILTIN, HIGH); Permet d'allumer la Built-in LED lorsque la note Mi3 est reconnue
}
```

Figure 4: Modification apporter au programme afin que la Built-in LED s'allume lorsque les notes DO3, MI3 et RE3 sont reconnues

Transmission d'un message :

Validation 7

Notre but est de transmettre 5 chiffres codés en ASCII à travers une mélodie. L'*American Standard Code for Information Interchange*, de son petit nom le code ASCII ou Ascii (prononcez aski), est une norme informatique pour le codage des caractères. En adoptant le même codage, les systèmes informatiques conçus par n'importe quel fabricant savent ainsi échanger du texte, des nombres, des signes de ponctuation et bien d'autres symboles.

Validation 8

Pour notre cas, nous avons affecté à chaque fréquence un nombre, allant de 0 à 9.

```
if (x > 145.0 && x < 148.0) {
  digitalWrite(2, HIGH);
  Serial.print(1);  Permet d'associer le code 1 de la fréquence 145 Hz à la fréquence 148 Hz
} else if (x > 160 && x < 170){
  digitalWrite(3, HIGH);
  Serial.print(2); Permet d'associer le code 2 de la fréquence 160 Hz à la fréquence 170 Hz
} else if ( x > 178 && x < 182) {
  digitalWrite(4, HIGH);
  Serial.print(3); Permet d'associer le code 3 de la fréquence 178 Hz à la fréquence 182 Hz
```

Figure 5: Un bout du programme pour attribuer aux fréquences un code secret

Nous avons rajouté ces lignes de code jusqu'à la dernière fréquence. Ce programme récepteur permet de convertir une mélodie en code secret.

Ensuite, nous avons écrit un script Matlab qui permet de convertir un code secret en mélodie :

```
function convertisseur(A,B,C,D,E)

     %Les différents fréquences du tableau de la validation 1
     a1=147;
     a2=165;
     a3=175;
     a4=196;
     a5=220;
     a6=247;
     a7=262;
     a8=294;
     a9=330;

     clc %Permet d'effacer l'écran
     Fe=8000; %Fréquence d'échantillonnage
     Ts=1/Fe; %Période d'échantillonnage
     t=[0:Ts:2]; %le tempo entre les notes


     %Tableau des notes
     notes = [A ; B; C; D; E];
     x = cos(2*pi*notes*t);

     %Redimensionnement du tableau avec la fonction RESHAPE
     sig = reshape(x',5*length(t),1);
     soundsc(sig,1/Ts)
end
```

Figure 6: Script Matlab qui permet de convertir un code secret en mélodie

Conclusion :

Durant ce projet nous voulons remercier Mme Cothenet pour sa disponibilité et ses conseils cela nous a permis de bien mener notre projet.

Ce projet a durée au total 22 heures de travail (TD et TP). Au début du projet, nous ne savions pas comment commencer. Au fil du temps, nous avons répartie les tâches en fonctions des compétences de chacun. Nous avons rencontré quelques difficultés, notamment au niveau des LEDs car elles n'ont pas une bonne intensité lumineuse.

Ce projet permettra à une personne d'envoyer un message sans que d'autres personnes n'arrivent pas à découvrir les informations contenues dans le message car ce message sera transformé en une mélodie.

Enfin, ce projet nous a permis de prendre un peu de recul par rapport au cours et il nous a permis de découvrir d'autres horizons.