



---

# Projet Ingénieur

## “Réalisation d’un système d’écoutes sonores à base de MEMS”

### Rapport Intermédiaire

---

**Etudiants:** Ali KHALAF, Logan PAQUIN

**Tuteurs:** Alexandre BOE, Thomas VANTROYS

**SE5**

**Promotion 2023**

## **Sommaire**

<b>I - Introduction</b>	<b>3</b>
<b>II - Démarche</b>	<b>4</b>
1 - Etude du projet	4
2 - Prise en main du matériel	5
3 - Acquisition d'une première base de données	6
4 - Réalisation du prototype	8
<b>III - Résultats actuels</b>	<b>9</b>
<b>IV - Travail à effectuer</b>	<b>9</b>
1 - Réalisation matérielle de prototype	9
2 - Contrôle sur l'acquisition	10
3 - Optimisation de l'algorithme d'analyse	10
4 - Utilisation du machine learning	10
<b>V- Conclusion</b>	<b>11</b>

# **I - Introduction**

Les objets connectés se déploient de façon massive dans nos vies personnelles et quotidiennes. La simplicité de création de ces objets permet la mise sur le marché de nombreux nouveaux objets commerciaux, voire artisanaux grâce au mouvement DIY (Do It Yourself). Cette grande rapidité rend l'écosystème des objets connectés très hétérogène et rend ainsi la reconnaissance et l'identification des objets compliquée. Une piste permettant l'identification des objets connectés consiste à s'intéresser à leur signature sonore. Pour cela, on capte à l'aide de microphones les émissions sonores de l'objet en fonctionnement et on extrait une signature sonore. Afin d'améliorer la sensibilité de la captation sonore tout en diminuant la taille et le coût, on peut envisager d'utiliser des microphones basés sur les technologies MEMS. En utilisant de tels microphones il sera possible de multiplier les sources de données.

Le projet consiste en la conception et la réalisation d'un système d'écoute embarquant plusieurs microphones MEMS. La plateforme comportera :

- une partie matérielle avec des microphones MEMS ;
- un protocole de synchronisation des mesures provenant des différents microphones ;
- une partie logicielle permettant de traiter les signaux acquis.

Des expérimentations basiques seront menées afin de valider le prototype et d'estimer les gains en termes de sensibilité notamment.

L'objectif de ce projet est de réaliser un prototype de système d'écoute de signature sonore à base de microphones MEMS permettant de reconnaître et différencier les objets connectés.

Afin de nous assurer du fonctionnement et de la fiabilité de notre prototype, nous réaliserons différentes mesures sur nos smartphones personnels. Nous réaliserons une base de données de référence pour une dizaine de cas différents en prenant plusieurs mesures de chaque cas afin de mesurer un taux d'erreur sur les mesures. Nous pourrions également travailler avec d'autres systèmes tels que des microcontrôleurs.



l'autre. Une étape de demuxage logiciel sépare les signaux des deux microphones et permet un traitement logiciel supplémentaire, une conversion typique de PDM en PCM est effectuée pour transformer les signaux PDM en format PCM, largement adopté et facile à gérer. Les données sont enfin écrites dans la mémoire RAM via un accès mémoire direct (DMA) afin de pouvoir être communiquées à l'ordinateur via USB.

## 2 - Prise en main du matériel

Après nous être familiarisé avec le projet et notre matériel, nous avons commencé par essayer de faire fonctionner les microphones en enregistrant nos voix. Suite à nos recherches sur les cartes d'expansion portant les microphones qui nous ont été confiées, nous avons trouvé que le fournisseur STMicroelectronics propose une bibliothèque de programmes tout fait permettant d'utiliser ces expansions avec différents microcontrôleurs STM32, utilisant la librairie **X-CUBE-MEMSMIC1**.

Cette bibliothèque comprend des pilotes et des middlewares permettant l'acquisition de données provenant des microphones numériques MEMS MP34DT06J et des microphones analogiques MEMS MP23ABS1 en streaming USB.

Après avoir récupéré la librairie, nous avons utilisé un exemple offert par le constructeur permettant de récupérer les acquisitions des deux microphones embarqués sur l'expansion numérique X-Nucleo-CCA02M2 connecté à un microcontrôleur STM32F401RE. Après avoir chargé le fichier binaire dans le microcontrôleur, la carte d'expansion est reconnue par l'ordinateur comme étant un **microphone externe** en mode stéréo.

Ainsi nous avons réussi à prendre en main les microphones numériques en enregistrant notre voix via le logiciel **Audacity**.

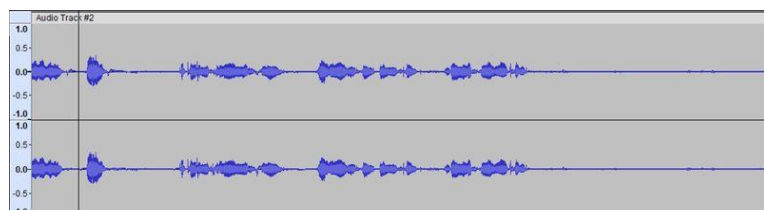
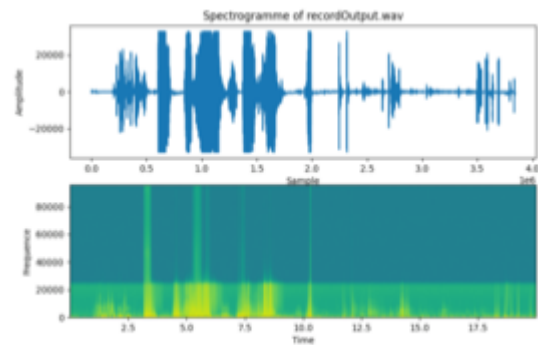


Fig2: Enregistrement de nos voix sur Audacity avec les microphones numériques

Afin de pouvoir utiliser et analyser les échantillons sonores par la suite, nous avons choisi de travailler avec des fichiers d'extension **wave**, qui est une compression sans perte communément utilisée.

Pour acquérir et enregistrer les échantillons sonores sous ce format et par la suite les utiliser, nous avons choisi de travailler avec le langage **Python** qui offre des modules permettant de travailler avec des fichiers wave, ainsi que de nombreuses possibilités concernant le traitement de signal.

Nous avons donc écrit un script Python permettant d'acquérir et d'enregistrer les informations sonores au format wave, qui nous permettrait par la suite de générer notre base de données de référence. Nous avons également réalisé un script permettant d'afficher le **spectrogramme** d'un fichier sonore au format wave et d'en extraire les informations qui nous serviront par la suite à l'analyse.



*Fig3: Spectrogramme d'un enregistrement audio*

Nous pouvons observer que les fréquences supérieures à 25 kHz ne présentent pas de puissance, car cela est au-delà de la bande passante des microphones. Nous ajouterons donc par la suite un filtre afin de ne récupérer que les fréquences inférieures à 25 kHz.

### 3 - Acquisition d'une première base de données

Des premiers tests ont été réalisés sur les chargeurs de nos smartphones (iPhone X et Edenwood W55HD) dans des conditions similaires dans le but de construire une première base de données qui nous servira pour la suite lors de la différenciation de ces deux appareils.

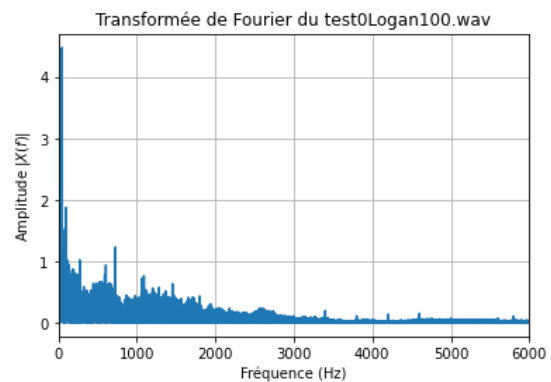
Tout d'abord nous avons effectué plusieurs mesures avec l'un des deux portables était à un niveau de charge de 100% en veille, hors mode avion, connecté à internet. Les sons sont enregistrés grâce au script python dans des fichiers d'extension **.wav**. Ensuite, nous avons pris un autre cas où l'autre smartphone est à un niveau de batterie de 50% dans les mêmes conditions mentionnées ci-dessus. Ces fichiers ont construit une base de données préliminaire.

Nous avons pensé à une structure de classement des fichiers, qui nous permettra facilement d'accéder à nos données lors de la partie du traitement de signal et de regrouper les différentes mesures concernant un même cas, pour cela nous avons mis en place une architecture de dossier qui contiendra pour chaque appareil la mesure réalisée, ainsi que la condition dédiée à cette mesure. Pour l'instant, cette architecture contient un dossier racine du projet, incluant les dossiers de chaque téléphone, qui contient les mesures à 50% et à 100%.

Une fois que la base de données est construite, nous sommes passés à l'analyse logicielle dans le but de réaliser des tests sur la partie de la PCA. PCA (Principal Component Analysis), est une méthode de la famille d'analyse des données et plus généralement de la statistique multivariée, qui consiste à transformer des variables liées entre elles dites corrélées (dans notre cas c'est la puissance à chaque fréquence en fonction du temps), extraites du spectrogramme associé aux fichiers sonores qui correspondent aux mesures faites sur les objets étudiés, en nouvelles variables décorréées les unes des autres qui seront nommées **composantes principales**. Cette méthode va nous permettre de résumer

l'information en réduisant le nombre de variables, et ce en perdant un minimum d'information. Ici, les variables sont représentées sur un plan de deux axes [composante principale 1; composantes principale 2] qui ne possèdent aucune signification physique.

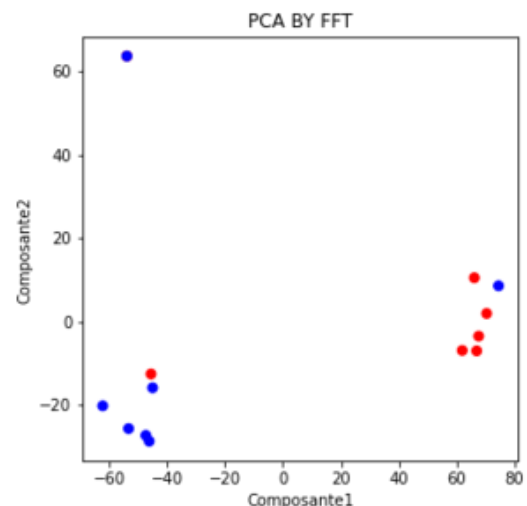
Pour appliquer la PCA sur nos données, nous nous sommes basés en premier temps sur la FFT des signaux sonores obtenues, cela a été fait grâce aux fonctionnalités offertes par python.



*Fig4: FFT d'une mesure de chargeur*

Suite à ça, nous avons utilisé les valeurs d'amplitude à chaque fréquence comme features nécessaires aux méthodes de PCA présente en python. Voici le résultat obtenu suite à la méthode de la FFT.

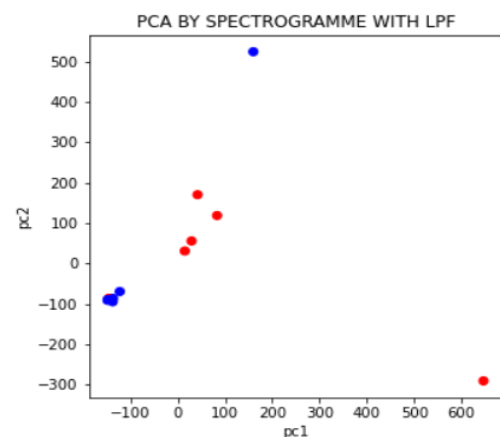
Les points en rouge représentent les échantillons d'un iPhone X, alors que les bleus celui d'un Edenwood W55HD.



*Fig5 : PCA sur les valeurs provenant de la FFT*

Ensuite nous avons utilisé les valeurs de puissance provenant du spectrogramme, dont nous avons filtré les fréquences hautes, à la place de celles provenant de la FFT, ce qui nous a donné le modèle ci-contre.

On remarque qu'avec cette méthode, les points sont bien séparés, ce qui indique que la distinction des objets a été faite correctement.



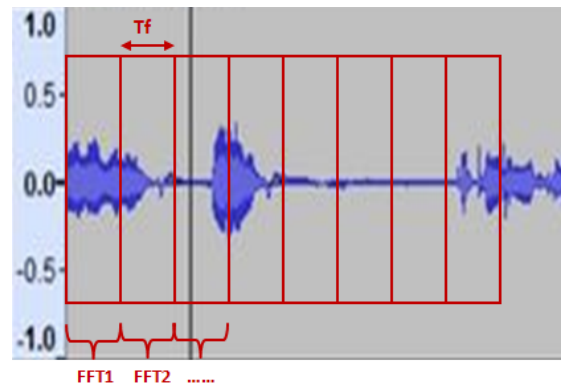
*Fig6 : PCA sur les valeurs provenant du spectrogramme*

## 4 - Réalisation du prototype

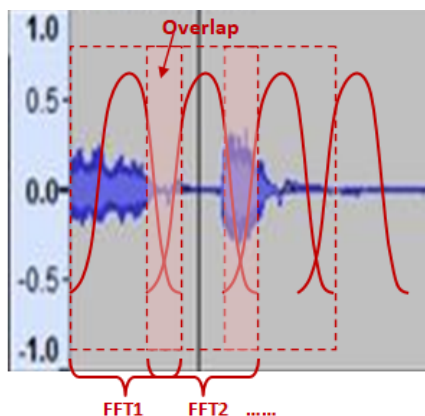
Après avoir réalisé ces premiers essais de l'analyse avec l'algorithme de PCA, nous avons pu identifier différents facteurs qui nous permettraient d'affiner la différenciation et d'obtenir des résultats plus facilement exploitables.

Premièrement, nous pouvons choisir le **jeu de données** que nous donnons à l'algorithme de PCA. Nous avons réalisé des essais en utilisant les données du spectrogramme et avec les données d'une simple FFT réalisée sur tout le fichier. Dans la théorie, le jeu de données provenant du spectrogramme devrait permettre de mieux différencier les différents groupes de données car il contient plus d'information, notamment concernant la disposition temporelle des fréquences présente.

Ensuite, dans le cas du spectrogramme, nous pouvons modifier plusieurs paramètres nécessaires au calcul de celui-ci. Le calcul du spectrogramme est en réalité composé de nombreuses FFT réalisés sur des fenêtres étroites et mises bout-à-bout. Sur celui-ci nous pouvons donc modifier la **taille de la fenêtre**  $T_f$  sur laquelle sera effectuée la FFT, la **forme de la fenêtre** (rectangulaire, Bartlet, Hamming, ...).



*Fig7: Principe de calcul du spectrogramme*



*Fig8: Overlap dans le spectrogramme*

Nous pouvons également modifier la valeur de l'**overlap**, qui correspond à une durée sur laquelle deux fenêtres de FFT consécutives se superposent, les valeurs concernées seront donc utilisées deux fois. Cela a pour principal intérêt de compenser l'affaiblissement des valeurs aux extrémités dûes à la forme de la fenêtre.

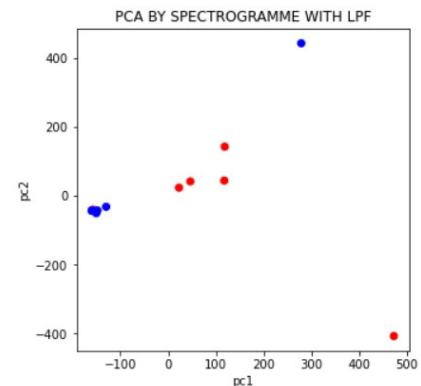
De plus, nous pouvons également travailler sur la gestion des micros afin de pouvoir récupérer les informations de chaque microphone individuellement afin de connaître la disposition spatiale du son et d'augmenter la sensibilité et la précision de l'analyse qui suivra. Cela nécessite de modifier le code existant provenant de l'exemple fourni par STMicroelectronics avec la librairie X-CUBE-MEMSMIC1 afin de contrôler la lecture des informations de chaque microphone dans les registres de la mémoire RAM.

Nous sommes actuellement en train de travailler sur cette étape, en nous ayant réparti les rôles de tels sortes que Ali s'occupe de l'optimisation de l'analyse logicielle à base de PCA et Logan de la lecture des données de chaque microphone individuellement.



### III - Résultats actuels

Actuellement, à partir de notre base de données nous parvenons à distinguer deux portables différents utilisant des chargeurs différents. De plus, en modifiant les paramètres de calcul du spectrogramme, nous sommes parvenus à séparer un peu mieux les deux groupes de points.



*Fig9: Overlap dans le spectrogramme*

Cependant, étant donné que ces mesures ne sont pas effectuées dans des conditions optimales (aucune isolation au bruit extérieur), nous nous sommes contentés de travailler avec deux cas très différents, ce qui nous a permis de définir les procédés de calculs que nous utiliserons par la suite, ainsi que de déterminer les paramètres sur lesquelles nous pourrions jouer afin d'affiner notre analyse.

Afin de réaliser des mesures plus fiables, nous travaillons sur la réalisation d'un prototype qui permettra de nous assurer que chacune sera prise dans des conditions semblables, que ce soit quant à l'isolation par rapport au bruit extérieur ou bien la disposition spatiale des microphones par rapport à l'objet à écouter.

A l'heure actuelle, nous utilisons seulement les microcontrôleurs STM32F401RE ainsi que la carte d'expansion numérique X-NUCLEO-CCA02M2 qui nous ont été fournis.

### IV - Travail à effectuer

#### 1 - Réalisation matérielle de prototype

Pour l'instant, les mesures et les expériences sont réalisées dans des conditions où des bruits extérieurs interviennent, ce qui peut jouer sur la qualité des résultats obtenues pour la suite, d'où l'intérêt d'utiliser une boîte spécifique pour ce type de mesures de son, qui contient une isolation acoustique.

D'autre part, et comme que le bruit est un élément incontrôlable, nous avons pensé à ajouter sur notre réalisation matérielle, des microphones à l'extérieur de la boîte de mesure, dans le but de détecter les sons provenant de l'extérieur, et essayer de les soustraire de l'enregistrement à possession. Cela sera fait grâce au microphone analogique (**STEVAL-MIC004V1**) et numériques (**STEVAL-MIC001V1**) fournis avec le matériel du projet et placés à l'extérieur.

En plus nous devrions prendre en compte la fixation des microphones à l'intérieur et à l'extérieur de la boîte, pour avoir toujours les même prise des mesures pour chaque cas d'étude, ce qui rend nos prélèvement fiable et correcte d'une mesure à l'autre.

## **2 - Contrôle sur l'acquisition**

Afin de connaître l'origine des informations sonores que nous utiliserons, il serait préférable de récupérer les informations de chaque microphone indépendamment, en étant capable de savoir quelle information provient de quel microphone.

Afin de réaliser cela, nous récupérons le code fourni en exemple par STMicroelectronics, que nous modifierons afin de lire dans la mémoire RAM les informations qui nous intéressent, et créer un fichier d'extension wave pour les données de chaque microphone.

## **3 - Optimisation de l'algorithme d'analyse**

Lorsque nous aurons une base de données conséquente obtenue par des mesures effectuées dans de bonnes conditions (avec le prototype fini), nous essaierons d'optimiser les paramètres de calcul du spectrogramme afin de faciliter la différenciation des cas.

Puis nous utiliserons un algorithme de segmentation de l'espace type SVM afin de définir des champs qui correspondent à chaque cas. Ces champs permettent de définir à quel cas correspondent chaque nouvelle mesure. Nous pourrons étudier l'influence des différents algorithmes de segmentation sur le taux d'erreur.

## **4 - Utilisation du machine learning**

Par la suite, nous aurons la possibilité d'introduire un modèle machine learning, grâce au logiciel **NanoEdge**, fourni par STMicroelectronics, qui nous permettra de faire une classification sur n-class. Nous pensons récupérer les données provenant du spectrogramme du signal (fréquence, puissance et temps), afin de les fournir au modèle qui nous enverra le cas le plus probable ainsi que la probabilité associé à ce cas.

Cette étape pourra remplacer l'analyse par la méthode de PCA et les algorithmes de segmentation par SVM.

## V- Conclusion

Pour conclure, après avoir étudié ce projet, nous avons réalisé une première base de données et de premiers tests, lesquels nous ont permis d'observer la faisabilité de ce projet ainsi que de nous éclairer concernant toutes les possibilités à notre disposition afin de réaliser un prototype fiable et sensible. Nous avons également pu réfléchir à la démarche à suivre par la suite.

Au fil de ce projet, des étapes supplémentaires que nous n'avions pas prévu à l'origine sont apparues, mais nous ne sommes actuellement pas en retard sur notre planning.

Actuellement, nous sommes capables de différencier des objets connectés bien distincts, ce qui permet de répondre au besoin principal du cahier des charges. Nous allons maintenant travailler sur la fiabilité et la sensibilité de notre prototype afin de proposer un système le plus performant possible, capable d'observer de légères variations sur les cas connus et de les caractériser précisément.