Forme

**RAPPORT DE PROJET**

*Projet d’électronique n°5 : Neural Speech*

Forme

|  |  |  |
| --- | --- | --- |
| AUTEURS : | | ENSEIGNANT : |
|  |  | Mme Najma GULDNER |
| AMBROSINI | Quentin |
| CHANZY | Alexandre |
| DAMIANOS | David |
| QIU | Jack |  |

L’objectif de ce projet est de combiner les compétences acquises lors des modules traitement du signal et calcul embarqué pour réaliser une intelligence artificielle embarquée.

A l’aide de la carte Arduino Due, le logiciel sera ainsi capable de numériser le signal audio provenant du microphone MAX9814 puis de le traiter numériquement.

Le résultat du traitement numérique du signal sera ensuite propagé dans un réseau de neurones qui aura pour fonction d’identifier le mot prononcé.

L’entrainement du réseau de neurones pourra se faire dès le début du projet à l’aide d’un data set fourni, permettant ainsi à l’équipe de projet d’être prête dès que l’acquisition et le traitement du signal auront pu être validés

Nous attestons que ce travail est original, qu’il est le fruit d’un travail commun au binôme et qu’il a été rédigé de manière autonome.

 Paris, le 26/05/2024

Table des matières

[**I - Objectif** 3](#_Toc167629576)

[**II - Glossaire** 3](#_Toc167629577)

[A- Termes 3](#_Toc167629578)

[B - Acronymes 4](#_Toc167629579)

[**III - L’équipe** 5](#_Toc167629580)

[A - Présentation de l’équipe 5](#_Toc167629581)

[B - Organisation de l’équipe 6](#_Toc167629582)

[C - Diagramme de GANTT 6](#_Toc167629583)

[**IV – Contexte et problématique** 7](#_Toc167629584)

[A - Contexte 7](#_Toc167629585)

[B - Problématique 7](#_Toc167629586)

[C - Spécifications techniques 8](#_Toc167629587)

[1. Arduino DUE 8](#_Toc167629588)

[2. Microphone MAX9814 9](#_Toc167629589)

[**V - Conception** 10](#_Toc167629590)

[A - Architecture fonctionnelle 10](#_Toc167629591)

[B - Architecture matérielle 10](#_Toc167629592)

[C - Architecture logicielle 10](#_Toc167629593)

[**VI - Développement** 10](#_Toc167629594)

[A - Module 1 : Numériser le signal audio 11](#_Toc167629595)

[B - Module 2 : Conditionnement du signal 12](#_Toc167629596)

[C - Module 3 : Ecouter et valider l’enregistrement 14](#_Toc167629597)

[D - Module 4 : Caractériser le timbre vocal 15](#_Toc167629598)

[**VII - Tests et validation** 15](#_Toc167629599)

[A - Module 1 : Numériser le signal audio 16](#_Toc167629600)

[B - Module 2 : Conditionnement du signal 17](#_Toc167629601)

[C - Module 3 : Ecouter et valider l’enregistrement 20](#_Toc167629602)

[D – Module 4 : Caractériser le timbre vocal 21](#_Toc167629603)

[VIII - Bilan 21](#_Toc167629604)

[A - État d’avancement 21](#_Toc167629605)

[B - Pertinence de la solution technique 21](#_Toc167629606)

[C - Bilan sur le travail d’équipe 21](#_Toc167629607)

[IX - Sources 22](#_Toc167629608)

[X - Annexes 22](#_Toc167629609)

# **I - Objectif**

Ce document a pour objectif de synthétiser un projet qui a eu lieu sur une durée de 6 semaines, en commençant par les hypothèses de base, en explicitant toutes les étapes tout en passant par différents points techniques jusqu’à l’aboutissement complet de ce projet. Les ultimes objectifs sont de mettre en pratique différentes notions vues en cours telles que l’utilisation de l’ADC ou encore du DAC de l’Arduino DUE ainsi que la mise en application d’opérations de filtrage.

Premièrement, les termes et acronymes techniques essentiels à la compréhension de ce dossier seront définis, suivis de la présentation de l’équipe ayant contribué à ce projet. Dans un second temps, ce rapport sera présenté dans son contexte et contiendra les différentes hypothèses, les recherches et l’analyse effectuées avant le commencement du projet. Les différentes étapes pour la réalisation de ce travail seront évoquées et illustrées par différents diagrammes et annexes présents au sein de ce rapport. Ces dernières auront pour but de présenter et renforcer les recherches effectuées pour l’achèvement de ce projet, tel que les documents techniques ou encore le code.

Ainsi, nous pourrons se rendre compte du déroulement et développement de ce projet sur le temps imparti.

# **II - Glossaire**

Ce rapport contient différents termes et acronymes techniques qui nécessitent d'être définis pour une bonne compréhension du rapport.

## Termes

|  |  |
| --- | --- |
| **Termes** | **Définition** |
| **Microcontrôleur** | Un gyropode est un moyen de transport électrique offrant une alternative de transport pratique et écologique dans les zones urbaines. C’est un outil à deux-roues permettant de transporter une personne à l’aide d’un système de stabilisation gyroscopique. Il est donc contrôlé par la position du corps de l’utilisateur qui peut s’incliner pour faire avancer ou reculer le gyropode. |
| **Diagramme de GANTT** | Outil utilisé en gestion de projet permettant de visualiser dans le temps les différentes tâches qui composent le projet. Il s'agit de représenter sous forme d’un calendrier les différentes tâches afin d’observer la gestion du temps. Ce genre de diagramme est beaucoup utilisé en entreprise et en études supérieures lors de différents projets. |
| **Microphone** | Appareil électrique qui amplifie les ondes sonores |

## B - Acronymes

|  |  |  |
| --- | --- | --- |
| **Acronyme** | **Signification** | **Explication** |
| ADC | *Analog to Digital Conversion* | C’est un dispositif électronique qui a pour but de traduire une grandeur analogique en une valeur numérique codée sur plusieurs bits (le signal converti est souvent une tension électrique.).  Le processus de conversion consiste à mesurer la tension ou le courant analogique d’entrée à intervalles réguliers, puis à quantifier cette mesure en une valeur numérique.  La qualité de cette conversion dépend de plusieurs facteurs. Pour qu’elle soit optimale, il est préférable que la résolution et le taux d’échantillonnage soient élevés et que la tension de référence soit stable.  On retrouve des CAN dans de nombreux systèmes électroniques tels que les capteurs, les instruments de mesure... |
| Filtre RIF | Filtre à réponse impulsionnelle finie | Un filtre RIF est caractérisé par une réponse impulsionnelle de durée finie. Cela signifie que la réponse du filtre à une est non nulle seulement pendant un nombre fini d'échantillons. Ils sont utilisés dans le traitement audio (suppression des bruitages), le traitement d’images (détection des bords ou lissage des images) ou encore dans la communication numérique (suppression des interférences).  Ce sont des outils puissants et flexibles en traitement du signal numérique, offrant des avantages en termes de stabilité et de conception de phase linéaire. |
| Filtre RII | Filtre à réponse impulsionnelle infinie | Un filtre RII est caractérisé par une réponse impulsionnelle de durée infinie. Cela signifie que la réponse du filtre à une impulsion dure indéfiniment, bien que ses valeurs puissent décroître à zéro sur le long terme. |
| MFC |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **III - L’équipe**

## A - Présentation de l’équipe

Les membres qui constituent cette équipe pour ce projet sont 4 étudiants en 1ere année d’étude en cycle ingénieur à l’ECE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Quentin Ambrosini** | **Jack QIU** | **Alexandre CHANZY** | **David DAMIANOS** |
| **Compétences** | Maitrise des différents **logiciels** tel que Arduino IDE – VS code – Tensorflow etc…  **Débogage** de circuits électronique : utilisation d’oscilloscope et de générateurs de signaux pour identifier et résoudre les problèmes. | * Analyseur logique * Débogage de codes et circuits * Résolution de problème * Algorithmes et structures de données |  | **Gestion de projet :** Suit/planifie/coordine des différentes phases du projet de manière à respecter les délais du projet.  **Test et validation :** Développement de procédures de test et validation pour assurer la qualité et la fiabilité de la DUE ainsi que du micro. |
| **Qualités** | **Précision :** Attention minutieuse aux détails. La patience est un élément clef pour rester concentré lors des débogages. | * Sérieux * Responsable * Efficace * Persévérant * Organisé * Esprit d’équipe | **Esprit analytique :** capaciter à analyser les problèmes complexes et à développer des solutions efficaces.  **Travail d’équipe :** c’est important d’avoir cette qualité pour mieux s’intégrer dans des groupe de projet ou encore pour mieux avancer sur une tâche complexe. | **Leadership :** capacité à guider et motiver l’équipe.  **Communication :** important pour que l’équipe ai une bonne coordination.  **Organisation :** organiser des tâches de manière efficace pour maximiser la productivité de notre travail. |

## B - Organisation de l’équipe

Dès la première semaine du projet, nous nous sommes réunis pour nous organiser et fixer ensemble les objectifs à atteindre à chaque fin de semaine afin d’optimiser au mieux notre temps.

Nous devions faire preuve d’organisation pour rendre à ce projet le meilleur rendu possible. Par la suite, concernant la répartition du travail, nous avions établi une ACD (Analyse Chronologique Descendante) de l’ensemble du projet, afin de nous répartir les différentes tâches. Elles ont été déléguées au membre du groupe selon leur préférence ou selon leur aisance dans certaines taches (Exemple : l’initialisation de l’ADC ou encore la mise en place d’un filtre RII ou RIF).

Mais nous n’étions pas dispersés, chacun dans son coin, travaillant chacun pour soi. En effet, la qualité phare de notre équipe était la communication. Nous assistions à l’avancement du projet, étape par étape, ensemble. Cela nous a permis d’augmenter notre qualité de travail et d’acquérir de nombreuses compétences que certains d’entre nous n’avaient pas forcément dans certains domaines au fil du projet.

En résumé, nous étions une équipe soudée, basée sur la communication, l’organisation, l’entraide et l’esprit d'équipe pour tirer profit de ce projet ensemble.

## C - Diagramme de GANTT

Comment est utilisé le temps alloué au projet ?

# **IV – Contexte et problématique**

## A - Contexte

Ce type de projet s’inscrit dans un contexte économique et sociétal marqué par plusieurs tendances et besoins importants. Parmi ces tendances nous pouvons identifier l’automatisation et l’intelligence artificielle. La demande pour des systèmes embarqués capable de percevoir et d’interpréter des informations tirées de l’environnement qui les entourent (comme la reconnaissance vocale par exemple) est en forte croissance. Elles sont utilisées dans des domaines diverses et variés comme les assistants vocaux (Siri, Alexa…). Ces innovations technologiques améliorent l’efficacité et la convivialité des systèmes électroniques automatisé. 

Ces produits illustrent très bien cette tendance en offrant des interfaces naturelles et accessibles. Cette technologie de reconnaissance vocale a des origines remontant jusqu’en 1950 avec le développement des premiers systèmes capables de reconnaitre certains mots. On peut citer par exemple le système « Audrey » développé par Bell Labs en 1952. Audrey était capable de reconnaitre les chiffres de 0 à 9.   
L’avènement du cloud computing et des capacités de traitement de grandes quantités de données au début des années 2000 a marqué une étape cruciale dans l’évolution de cette technologie. Les assistants vocaux cités précédemment utilisent tous des réseaux de neurones profond ce qui permet d’avoir des capacités de reconnaissance vocales très précises.

## B - Problématique

Avec ce projet, nous cherchons à répondre à la problématique suivante :

Quels sont les avantages d'utiliser un réseau de neurones pour la reconnaissance vocale dans des systèmes embarqués par rapport aux méthodes traditionnelles ?

En utilisant notre Arduino DUE et notre microphone MAX9814, nous visons à développer un système capable de numériser le signal audio de notre voix, le traiter numériquement, et enfin de l’analyser à l’aide d’un réseau de neurones pour identifier le mot prononcé. Ce type de systèmes embarqués est très utile et peut être implémenté, par exemple, dans les fonctionnalités mains libres des smartphones (Siri, Google Assistant).

Le projet se déroulera en plusieurs étapes clés. Tout d'abord, le signal audio capté par le microphone sera converti en un signal numérique par l'Arduino DUE. Ensuite, ce signal numérique sera filtré et prétraité pour améliorer la qualité et réduire le bruit.

Après cette étape, le signal sera analysé pour extraire des caractéristiques essentielles, telles que les coefficients cepstraux en fréquences de Mel (MFCCs), qui sont des représentations des propriétés de la voix humaine. Enfin, ces caractéristiques seront introduites dans un réseau de neurones entraîné pour reconnaître et identifier les mots prononcés. Une fois le mot identifié, le système pourra exécuter des commandes spécifiques, comme allumer une LED ou activer d'autres dispositifs.

## C - Spécifications techniques

Pour la réalisation de ce projet, nous avons fait usage du microcontrôleur Arduino DUE, le microphone MAX9814 ainsi que des objets de notre kit d’électronique.

### Arduino DUE

La carte Arduino DUE fait partie des nombreuses cartes créées par Arduino. La carte Arduino Due fait partie des nombreuses cartes créées par Arduino. Elle contient un microprocesseur Atmel SAM3X8E cadencé à 84 MHz avec une mémoire flash de 512 kB, une SRAM de 96 kB et une EEPROM émulée via la mémoire flash. Elle fonctionne à une alimentation de 3,3 volts à courant continu. Ce composant est idéal pour une large gamme de projets électroniques, robotiques, systèmes bouclés ou pour des systèmes embarqués avancés. De plus, il contient un bootloader qui permet de modifier le programme sans avoir à passer par un programmateur externe. Ce composant nous permettra pour notre projet d’implémenter nos codes et nos tests afin de remplir nos différents objectifs.



Figure 1 : Arduino DUE

Les principales caractéristiques de cette carte sont :

* **Microcontrôleur**: Atmel SAM3x8E ARM Cortex-M3
* **Horloge interne** : 84 MHz
* **Mémoire Flash** : 512 kB
* **Mémoire SRAM** : 96 kB
* **Mémoire EEPROM** : Emulée via la mémoire flash
* **54 broches d'E/S numériques** : dont 12 PWM
* **12 broches d'entrée analogique**
* **Interfaces de communication** : UART, I2C, SPI, CAN, USB OTG
* **Alimentation** : 3,3V via source externe ou USB
* **Régulateur de tension intégré**
* **Programmation via l'IDE Arduino**

La puissance de traitement et les capacités étendus de la mémoire de la carte Arduino DUE justifie le choix de l’utilisation de cette carte pour ce projet.

### Microphone MAX9814

Le microphone MAX9814 est un composant clé de notre projet de reconnaissance vocale embarquée. Il s'agit d'un capteur audio de haute performance conçu pour capturer les signaux audios avec une grande précision et une faible distorsion. Ce microphone est particulièrement adapté aux applications nécessitant une sensibilité élevée et une large plage dynamique, comme la reconnaissance vocale. Le MAX9814 est équipé d'un amplificateur à gain automatique (AGC), ce qui simplifie le traitement du signal en ajustant automatiquement le gain pour obtenir une qualité audio optimale. Ce composant, utilisé en conjonction avec la carte Arduino Due, nous permettra de capturer et de traiter le signal audio de manière efficace pour notre projet.

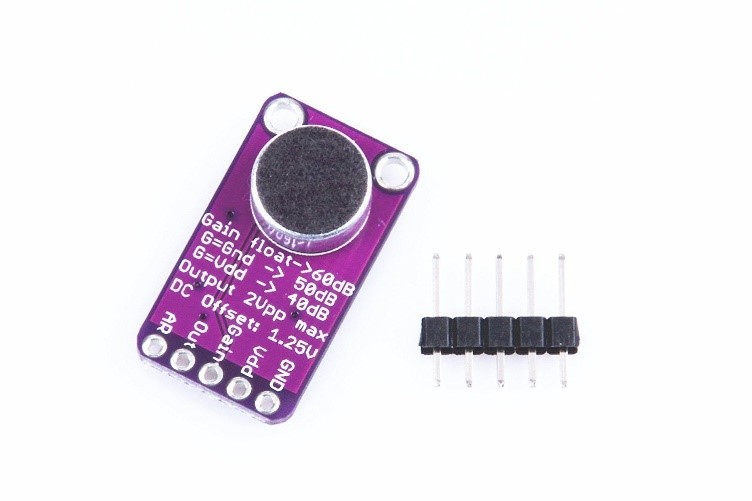


Figure 2 : Microphone MAX9814

Voici les principales caractéristiques du microphone MAX9418 :

* **Amplificateur à Gain Automatique (AGC)** :

Ajuste automatiquement le gain pour une capture audio optimale.

* **Réponse en fréquence étendue** :

Conçue pour des applications audios de haute qualité.

* **Faible bruit** : Réduit le bruit de fond pour une meilleure clarté audio.
* **Interface de communication** :

Sortie analogique facile à interfacer avec l’Arduino DUE.

* **Tension d'alimentation** : 3,3 V
* **Faible consommation d'énergie** : Idéal pour les applications sur batterie.
* **Sensibilité élevée** : Permet de détecter les sons faibles
* **Petit facteur de forme** : Facilite l'intégration dans des systèmes compacts
* **Applications supplémentaires** :

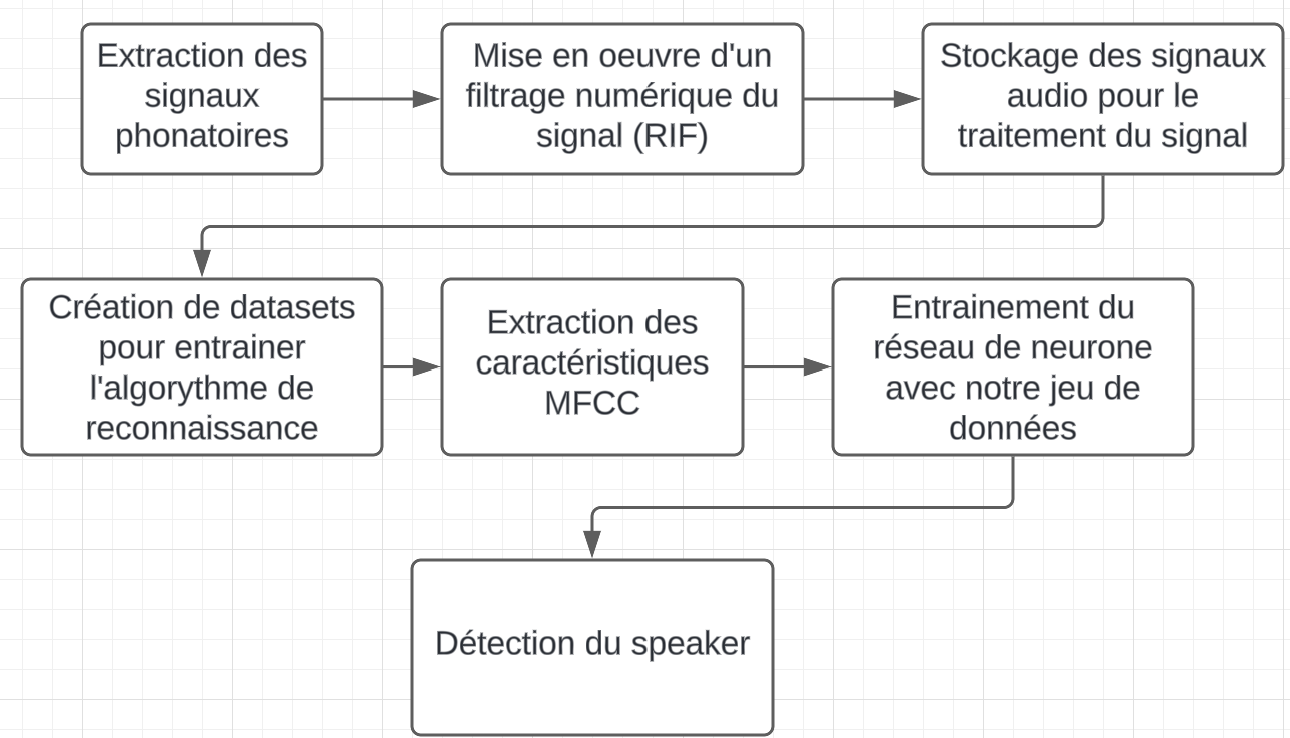
Compatible avec les applications de détection de voix, d'enregistrement audio, et de reconnaissance vocale

# **V - Conception**

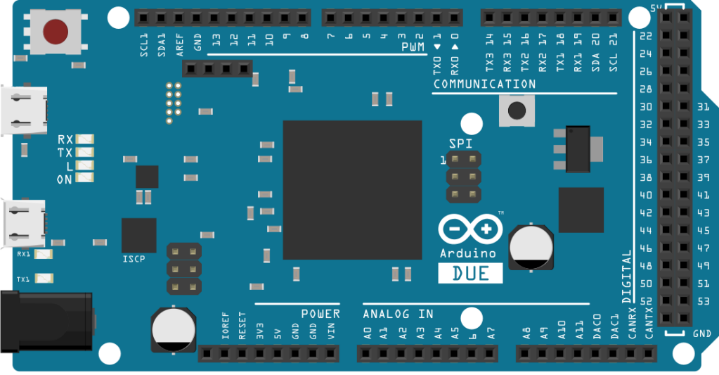
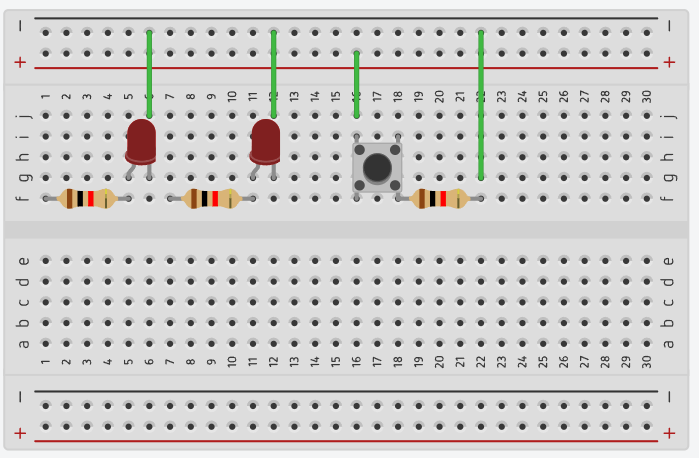
## A - Architecture fonctionnelle

L’architecture fonctionnelle est une des étapes clé pour mener à bien ce projet. Elle permet une représentation sous forme de diagrammes, pour simplifier un procédé relativement complexe. Elle a aussi pour but d’illustrer les différentes fonctions qui sont définies en français, sans termes techniques, afin d’avoir une vision globale sur le sujet.

Voici, ci-dessous, le digramme fonctionnel du Neural Speech :



## B - Architecture matérielle



Nous avons ci-dessus notre schématique du circuit électronique.

**Nous avons utilisé :**

- Arduino DUE

- Un microphone MAX9814(relié à la pin A0)

- 2 leds (une rouge et une verte aux pin D5 et D9)

- 2 résistances

- Un bouton (relié au pin D2)

- une Breadboard

## C - Architecture logicielle

Comment fonctionne le programme

**NB** : Présenter un algorigramme de votre code si vous en avez-un.

# **VI - Développement**

## A - Module 1 : Numériser le signal audio

L’ADC de l’Arduino DUE est un composant qui va être crucial pour numériser les signaux analogiques captés par le microphone. Pour capter ces signaux, il faut pouvoir réaliser un échantillonnage fixe et précis. Pour ce faire, l’ADC doit être réglé en mode interruption sur Timer, avec une fréquence d’échantillonnage de 32 kHz. Pour utiliser cette fréquence d’échantillonnage, nous devons régler le paramètre RC de l’ADC grâce à la formule suivante :

𝑅𝐶 = 𝑀𝐴𝑆𝑇𝐸𝑅 𝐶𝐿𝑂𝐶𝐾𝑃𝑟𝑒𝑠𝑐𝑎𝑙𝑒𝑟/(𝐹𝑟é𝑞𝑢𝑒𝑛𝑐𝑒 𝑑′é𝑐ℎ𝑎𝑛𝑡𝑖𝑙𝑙𝑜𝑛𝑛𝑎𝑔𝑒)

L’ADC étant réglé en mode interruption sur Timer, c’est-à-dire qu’il génère des interruptions à intervalle régulière, nous pouvons calculer la période d’échantillonnage pour mieux comprendre :

𝑇 = 1/𝑓𝑒 = 1/32000 = 31.25 𝜇𝑠

Le timer génère des interruptions toutes les 31.25 microsecondes. C’est-à-dire que toutes les 31.25 microsecondes, nous allons effectuer des mesures par le biais de filtre numériques. Pour s’assurer de nos mesures, il faut également configurer le DAC de l’Arduino DUE. Il est utilisé pour reconvertir notre signal numérisé en un signal analogique pour pouvoir procéder à des vérifications sur oscilloscope.

Une image contenant ligne, diagramme

Description générée automatiquement

Figure 3 : Signal théorique censé s'afficher à l'oscilloscope

## B - Module 2 : Conditionnement du signal

Cette partie consiste à mettre conditionner le signal pour qu’il puisse être traité par un réseau de neurones. Pour cela, il est nécessaire de réaliser une opération  de down sampling. Il faut passer la fréquence d’échantillonnage de 32 kHz à 8 kHz. Cela permet de conserver les fréquences fondamentales de la voix humaine tout en réduisant le nombre d’informations à traiter par le réseau de neurones.

Tout d’abord, il faut éviter le phénomène de repliement. Ce phénomène se produit généralement lorsqu’un signal analogique est échantillonné à une fréquence d’échantillonnage insuffisante. Plus précisément, cela se produit lorsque les composantes de fréquence du signal dépassent la moitié de la fréquence d’échantillonnage, cela ne respecte pas la condition de Shannon qui énonce que, la fréquence maximale du signal échantillonné ne doit pas dépasser la moitié de la fréquence d’échantillonnage.

Dans notre cas, la fréquence d’échantillonnage étant « downsampler » de 32 kHz à 8 kHz, on doit utiliser un filtre numérique passe bas pour atténuer les fréquences supérieures à 4 kHz avant cette opération de down sampling pour éliminer toute possibilités de recouvrement spectral.

Comment avons-nous choisi le type de filtre à utiliser ? Nous avions le choix entre deux types de filtre numériques passe bas : un filtre RIF (Réponse impulsionnelle finie) et un filtre RII (Réponse impulsionnelle infini). Nous avons fait le choix d’utiliser un filtre RIF pour plusieurs raisons :

* **Stabilité** : Le filtre RIF est toujours stable car il n’a pas de rétroaction (la rétroaction désigne un mécanisme ou la sortie du filtre est utilisée par le système comme une partie de l’entrée) contrairement au filtre RII qui peut s’avérer instable dans le cas où les coefficients de rétroaction ne sont pas correctement choisis.
* **Facile à concevoir**: Nous avons besoin de confectionner des coefficients pour le filtre RIF. Nous pouvons utiliser des outils pour générer ces coefficients de manière automatique avec notamment, l’outil que nous avons utilisé : Tfilter.

Une image contenant texte, capture d’écran, Tracé, ligne

Description générée automatiquement

Figure 4 : Gain vs Fréquence générée sur Tfilter

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

Figure 5 : Quelques coefficients générée par TFilter

A l’opposé, un filtre RII demande des calculs beaucoup plus complexes pour garantir sa stabilité.

**Implémentation sur l’Arduino DUE facile :** Son implémentation sur notre carte Arduino est plus facile. On peut aussi prévoir le délai de traitement du filtre contrairement à un filtre RII ou on ne peut pas prévoir précisément son délai puisqu’elle varie à cause de la rétroaction.

**Mémoire**: Le filtre RIF ne présente pa que des avantages. Comme sa mémoire. Il est possible que nous ayons besoin de plus de mémoire pour stocker les coefficients contrairement au filtre RII qui peut être plus efficace en termes de mémoire de calcul et de calculs pour un filtrage équivalent.

Pour palier à ces soucis de mémoire, nous devons implémenter un système de buffer circulaire. Ce système de buffer circulaire va nous permettre de de gérer les données de manière continue tout en minimisant l’utilisation de la mémoire.

Concrètement, comment fonctionne ce système de buffer circulaire ?

Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquementLe buffer circulaire est un tableau de taille fixe. Lorsqu’on ajoute un élément dans le buffer. Il permet de stocker chaque nouvel échantillon à l’emplacement de l’échantillon le plus ancien. Contrairement au buffer linéaire qui fonctionne comme une file (principe FIFO : premier arrivé, premier sorti).

Figure 6 : Schéma expliquant le fonctionnement du buffer circulaire

Lorsque la nouvelle position d’échantillon atteint la fin du buffer, on revient au début de la mémoire allouée : le prochain emplacement mémoire sera le premier emplacement du buffer. Le cheminement est donc circulaire.

## C - Module 3 : Ecouter et valider l’enregistrement

Nous devons maintenant nous assurer du bon fonctionnement de ces deux premières fonctions. Dans ce module, nous cherchons a transféré les valeurs du buffer vers Audacity pour l’enregistrement du mot « électronique ». Cette partie est relativement courte mais essentiel pour la suite du projet. Voici les étapes à suivre pour valider le bon fonctionnement de l’échantillonnage ainsi que le conditionnement du signal :

* **Numérisation du signal** : on doit d’abord numériser le signal audio avec l’ADC à la fréquence requise de 32 kHz. On stock ensuite les échantillons numériser dans notre buffer circulaire.
* **Récupération des données**: pour récupérer les données, nous allons utiliser le logiciel Putty. Grâce à ce logiciel, nous pouvons enregistrer le flux de données série dans un fichier binaire.
* **Importer les données sur Audacity**: Après, nous avons choisi d’importer les données sauvegarder grâce à Putty puis configurer les différents paramètres d’importation comme la fréquence d’échantillonnage, le nombre de canaux, le format de données etc…
* **Vérification de l’enregistrement** : Si tout s’est bien passé, nous allons pouvoir écouter l’enregistrement. Il faut que le mot « électronique » soit entendu clairement pour valider le fonctionnement.

## D - Module 4 : Caractériser le timbre vocal

Dans ce module, notre objectif est de caractériser le timbre vocal en utilisant l’algorithme MFC. On va d’abord effectuer une opération de fenêtrage en séparant l’enregistrement en différentes frames.

Une image contenant texte, capture d’écran, affichage, Police

Description générée automatiquement

Figure 7 : explication de notre procédé de fenêtrage

Nous allons segmenter le signal audio en plusieurs frame de 256 échantillons chacune. Nous devons recouvrir les frames pour garantir la continuité des informations temporelles. Ensuite pour vérifier, on peut exécuter la même procédure en utilisant le logiciel Putty pour afficher le signal audio de l’ensemble des frames et mettre en évidence le recouvrement.

# **VII - Tests et validation**

## A - Module 1 : Numériser le signal audio

Une image contenant texte, Appareils électroniques, écran, ordinateur

Description générée automatiquement

1 carreau représente 50 μs

Figure 8 : affichage du signal du DAC sur l'oscilloscope

En jaune nous avons un signal sinusoïdale correspondant à la fréquence max théorique selon la condition de Shannon c’est-à-dire 16 kHz. En violet, nous avons le signal du DAC.

Déjà, nous observons qu’a intervalle régulière, le signal jaune est échantillonné. Nous pouvons voir une continuité entre les interruptions de l’ADC. Les segments horizontaux violets montrent donc que le signal jaune est échantillonné à intervalle régulière et continue.

Une image contenant texte, Appareils électroniques, écran, ordinateur

Description générée automatiquement

Environ 30 μs

Figure 9 : Analyse de la période d'échantillonnage

L’oscilloscope est réglé de manière à ce qu'un carreau corresponde à 50 microsecondes. Nous pouvons estimer que la durée d’un échantillon, c’est-à-dire sa période, est d’environ 30 microsecondes. Théoriquement, nous devrions avoir une période d’échantillonnage égale à 31,25 microsecondes.

Nous avons donc tous les éléments pour valider l’échantillonnage de notre signal !

## B - Module 2 : Conditionnement du signal

Une image contenant Appareils électroniques, texte, personne, multimédia

Description générée automatiquement

Figure 10 : spectre du signal sans réduction de gain

Une image contenant texte, Appareils électroniques, multimédia, écran

Description générée automatiquement

Figure 11 : spectre du signal avec réduction du gain

En comparant les deux signaux à l’oscilloscope, nous observons que le gain après filtrage a bien diminué considérablement après la fréquence de coupure qui est de 4 kHz. Nous avons une atténuation supérieure à 30 dB ce qui correspond bien aux attentes du cahier des charges.

Concernant le temps de filtrage de notre filtre RIF :

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Figure 12 : temps de filtrage d'un échantillon

Nous avons programmé une série de tests afin d’afficher le temps de traitement du filtre RIF. Comme nous pouvons le voir sur la figure 12, le temps de conversion (affiché en microsecondes) est bien inférieur à 31 microsecondes. Nous devons également vérifier que le buffer circulaire fonctionne correctement.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 13 : données enregistrer dans le buffer circulaire

En affichant les valeurs du buffer circulaire, nous obtenons les valeurs suivantes présentées sur la figure 13. L’élément le plus récent est indiqué par une étoile \* (par exemple : 38681\*). Comme nous pouvons le voir sur la toute première ligne, un seul élément est inséré. Sur la 2e ligne, un deuxième élément est inséré et ainsi de suite jusqu’à ce que le buffer soit complet. Une fois le buffer complet, le nouvel élément inséré est censé remplacer l’élément le plus ancien.

Nous observer cette opération à partir de la ligne 14. L’élément le plus ancien « 38681 » est remplacé par l’élément le plus récent « 31446 ». Ces opérations sont effectuées indéfiniment. Nous pouvons alors valider le bon fonctionnement de notre buffer avec ce chemin « circulaire » caractéristique d’un buffer circulaire.

## C - Module 3 : Ecouter et valider l’enregistrement

Après transfert de nos données sur Audacity, nous avons pu écouter le mot « électronique ».

Une image contenant ligne, Tracé, logiciel, capture d’écran

Description générée automatiquement

Figure 14 : spectre du signal audio "électronique" reconstituée sur Audacity

Voici le spectre correspondant au signal audio du mot « électronique ». Nous apercevons plus précisément à quel moment nous pouvons entendre le mot sur le spectre. Cette zone est encadrée en rouge.

## D – Module 4 : Caractériser le timbre vocal

Une image contenant texte, ligne, capture d’écran, logiciel

Description générée automatiquement

Figure 15 : Spectre du signal audio "rouge" après assemblage des frames sur Audacity

Une image contenant texte, Tracé, capture d’écran, diagramme

Description générée automatiquement

Figure 16 : spectre du signal audio "rouge" après assemblage des frames

Après avoir assemblé les frames du signal audio récupérer grâce à PuTTY. En écoutant le signal sur Audacity, Nous pouvons entendre le mot rouge, ce qui nous permet de valider que le recouvrement fonctionne correctement et que nous n’avons perdu aucunes données.

Par la suite, nous avons les spectres du signal audio sur les figures 15 et 16.

Nous effectuons la même opération avec le signal audio « bleu » et nous pouvons conclure que le recouvrement fonctionne pour ce signal également. Les spectres sont observables sur les figures 17 et 18.

Une image contenant logiciel, texte, ligne, Logiciel multimédia

Description générée automatiquement

Figure 17 : spectre du signal audio "bleu" après assemblage des frames sur Audacity

Une image contenant texte, capture d’écran, Tracé, diagramme

Description générée automatiquement

Figure 18 : spectre du signal audio "bleu" après assemblage des frames

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 19 : Signal audio avant et après le filtre de preemphasis

Une image contenant texte, Tracé, capture d’écran, diagramme

Description générée automatiquement

Figure 20 : FFT calculée par la carte pour une fréquence de 1 kHz

Une image contenant capture d’écran, Rectangle, carré, Caractère coloré

Description générée automatiquement

Figure 21 : Coefficients MFCC sans le calcul de la DCT

Une image contenant capture d’écran, texte, ligne, Tracé

Description générée automatiquement

Figure 22 : Coefficients MFCC après calcul de la DCT

# VIII - Bilan

## A - État d’avancement

Où en est le projet ? A-t-on atteint les objectifs ?

Quels modules restent à finaliser (ou à perfectionner pour être en accord avec les spécifications techniques) ?

## B - Pertinence de la solution technique

Quelles sont les limites techniques de la solution développée ?    
Quelles sont les possibilités d’évolution ou de poursuite ?

## C - Bilan sur le travail d’équipe

Qu’avez-vous appris individuellement ? Quelles compétences vont pouvoir être mises en avant lors de votre prochaine recherche de stage ?

Comment l’équipe aurait pu mieux s’organiser ? Proposer un plan d’action pour le prochain projet.

Ce projet a été une expérience à la fois riche et inspirante pour notre équipe. Nous avons beaucoup appris, tant sur l e plan technique que pratique. Ce projet nous a également permis de nous dépasser et de faire preuve d'ingéniosité, nous incitant à explorer de nouvelles perspectives. Nous avons cherché à tirer le maximum de bénéfices des enseignements tirés de cette expérience. De plus, ce projet nous a aidés à développer une méthode de travail en équipe efficace, en respectant un cahier des charges précis et en produisant de nombreux résultats documentés de manière rigoureuse. En somme, cette expérience nous sera précieuse pour nos futurs projets et dans notre carrière d'ingénieur.

# IX - Sources

Documents utilisés et sites internet consultés pour développer le projet.

NB : voir le document « comment rédiger un rapport » sur la page Moodle La Toolbox pour la syntaxe à utiliser pour vos citations.

# X - Annexes

Documents annexes, éventuels codes (**pas de code dans le rapport**).