ECE
ÉCOLE D'INGÉNIEURS
ENGINEERING SCHOOL

Projet d'électronique

Neural Speech

Version 1.3

Objectifs pédagogiques

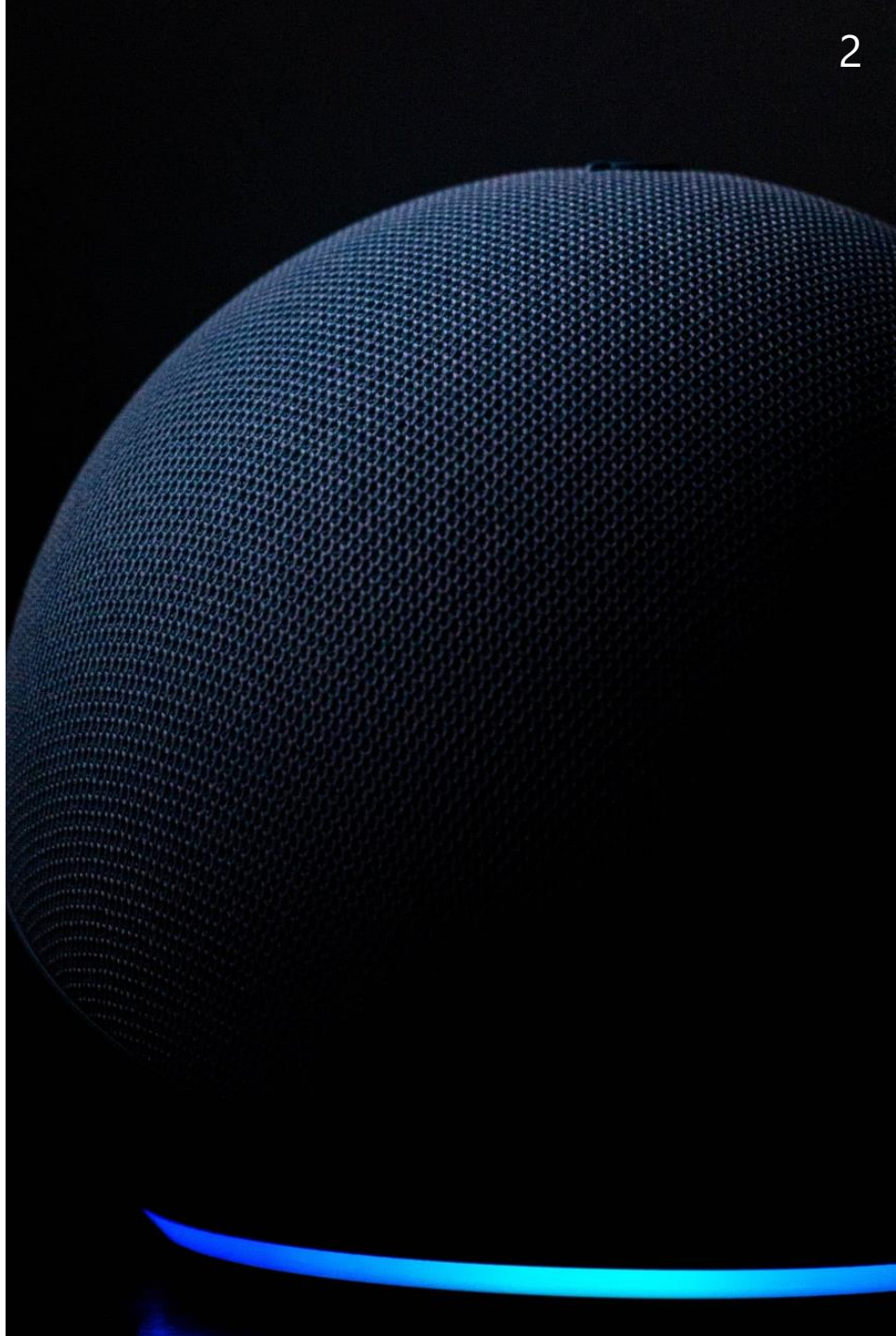
- Appliquer les acquis : Configuration de l'ADC, utilisation d'algorithme de traitement, conception de réseau de neurones.
- Développer de nouvelles compétences

- Méthodologie de projet (cycle en V)
- Documenter un travail technique
- Respecter des deadlines
- Écrire un rapport professionnel

- Travail de groupe (à deux groupes de TP*)
- Présentation professionnelle d'un travail technique

- Créativité

*Pas d'adaptation du barème si jamais un groupe de plus ou moins de 4 étudiants se forme.
Interdiction formelle de remanier les groupes de TP.
Chaque groupe de TP est responsable de son kit prêté en début de semestre.



Neural Speech

L'objectif de ce projet est de combiner les compétences acquises lors du module de calcul embarqué pour réaliser un **projet de reconnaissance vocale basé sur une IA embarquée** selon le use case de votre choix.

Par exemple :

- Commande vocale de l'affichage de la fiche ou de l'image d'un pokémon ;
- Commande vocale pour jouer à « question pour un champion » sur un écran OLED ou sur une interface graphique Processing ;
- Maison connectée ;
- Traducteur dans plusieurs langues ;
- Etc.

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.
Voir le tutoriel sur Audacity dans la Toolbox.

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 pour reconnaitre les mots « bleu » et « rouge » (**GitHub Pôle électronique**). Cela permettra ainsi à la sous-équipe en charge du réseau de neurones de travailler dès le début du projet, sans attendre les avancées de la sous-équipe en charge de l'acquisition. Il faudra bien évidemment adapter le dataset à votre cas d'usage.

La ligne rouge à ne pas franchir (sinon hors sujet) est que la reconnaissance doit se faire sur l'Arduino Due impérativement.

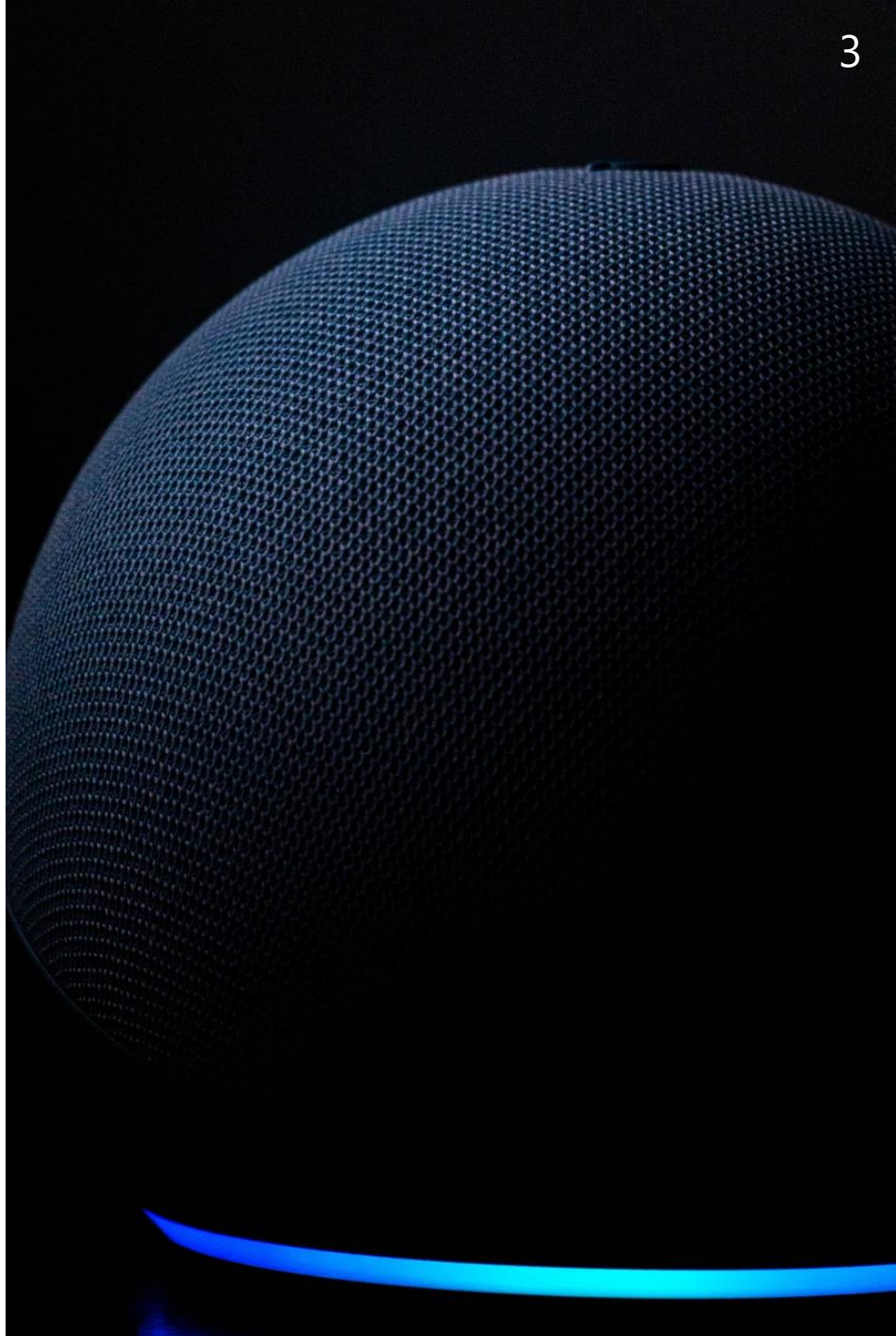
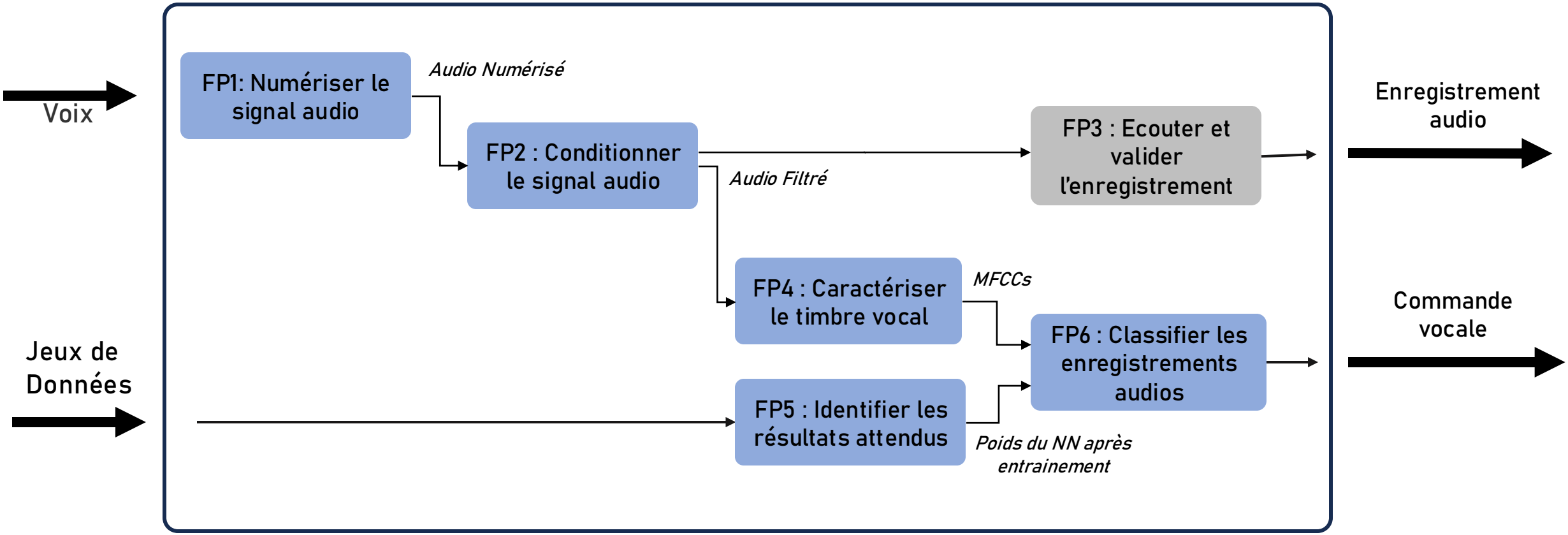


Diagramme Fonctionnel

Afin de répondre aux exigences du projet, il est nécessaire de concevoir tous les modules fonctionnels. Ils doivent chacun répondre à leurs exigences techniques respectives décrites par la suite.



FP1 : Numériser le signal audio

ET1. **L'ADC de l'Arduino due doit être capable d'échantillonner le signal à une fréquence de 32KHz.**

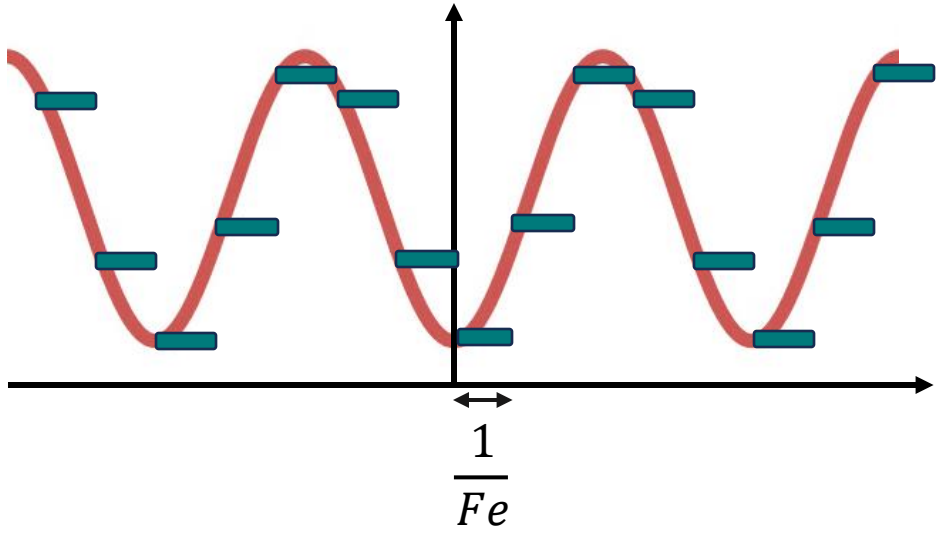
❑ Conception

Pour numériser un signal, il faut absolument avoir un **échantillonnage fixe et précis**. Pour cela l'ADC de l'Arduino Due doit être en mode d'interruption sur timer.
Il est demandé d'échantillonner le signal à 32KHz, afin de pouvoir par la suite appliquer un filtre numérique. Après avoir bien compris l'intérêt d'un tel échantillonnage, configurer l'Arduino Due en conséquence.

❑ Validation

Vérifier la validité de la fréquence échantillonnage à l'aide d'une mesure sur l'oscilloscope en utilisant le DAC de l'Arduino Due pour reconvertir le signal numérisé sous forme de signal analogique.

Il peut être aussi intéressant de vérifier que la condition de Nyquist est bien respectée à la fréquence maximale théorique du signal.



FP2 : Conditionnement du signal

Pour pouvoir traiter l'audio à l'aide du réseau de neurones implémenté sur la carte Arduino DUE, il va être nécessaire de réduire la fréquence d'échantillonnage : procéder à un **down-sampling**.
Il est proposé de réduire la fréquence d'échantillonnage à 8Khz, ce qui est suffisamment élevé pour conserver les fréquences contenues dans tout spectre sonore couvert par la voix humaines et donc la clarté de l'enregistrement. Afin d'atténuer le phénomène de repliement lors du sous échantillonnage, nous allons utiliser un filtre numérique pour retirer les hautes fréquences de l'enregistrement.
Enfin, il devra être possible d'enregistrer 1 seconde d'audio juste après la pression d'un bouton.

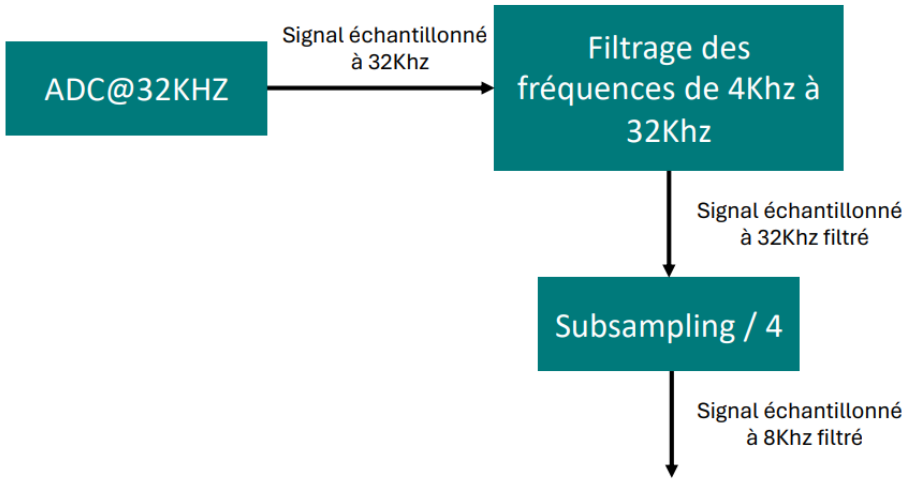
ET2. **Les fréquences supérieures à 4Khz doivent être atténuées d'au moins 30dB**

- **Conception**

  Mettre en œuvre un filtre passe-bas numérique RII ou RIF. Justifier le choix du filtre, puis en fonction du filtre choisi en faire la conception. Attention à la quantité de coefficients nécessaire.

- **Validation**

  A l'aide du spectre du signal, mettez en évidence une réduction du gain après la fréquence de coupure.



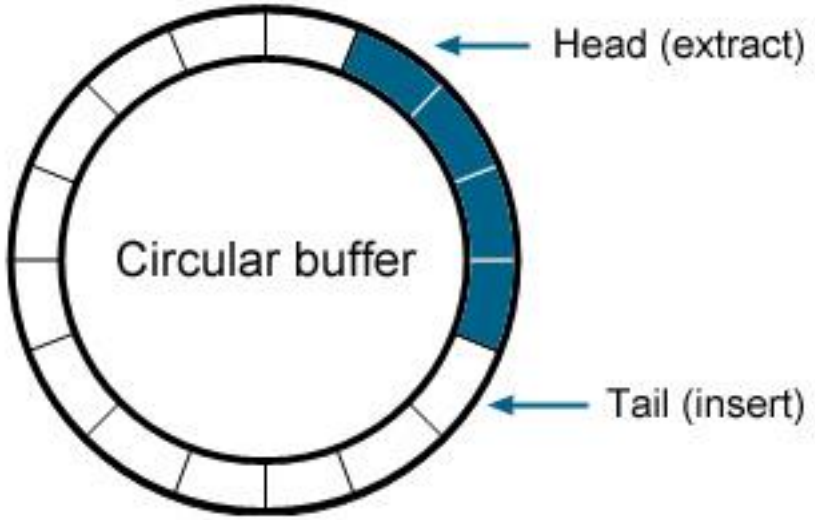
ET3. **Pour chaque échantillon, le filtrage doit être fait en moins de 31 µs**

❑ Conception

La mémoire de la carte Arduino Due étant trop faible pour supporter un enregistrement de plus d'une seconde à 32Khz, il va être nécessaire d'utiliser un **buffer circulaire** pour appliquer le **filtre RIF**. À la suite du filtrage, échantillonner le signal à 8Khz.
Implémenter en un à l'aide des exemples du cours.

❑ Validation

Validation du temps de conversion en l'affichant sur la console (à l'aide de la fonction micros()). Validation du bon fonctionnement du buffer sur la console (affichage du buffer sur 10 cycles par exemple)



FP3 : Ecouter et valider l'enregistrement

Cette fonction est une fonction de validation afin de s'assurer du bon fonctionnement de la FP1 et FP2.
L'Objectif est de transférer à l'aide de la communication série les valeurs du buffer précédemment remplis afin de pouvoir les **écouter**.

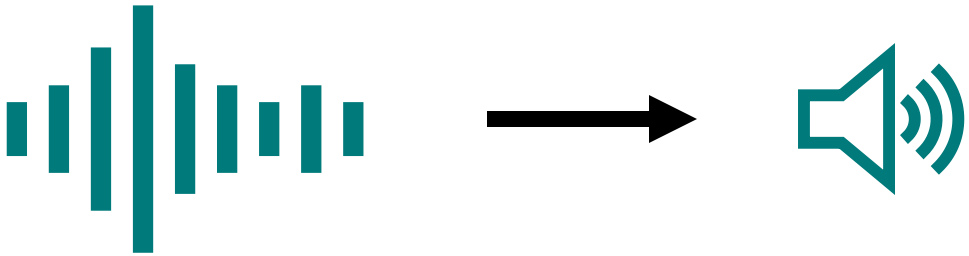
ET4. **Une lecture sur Audacity doit pouvoir restituer de manière claire l'enregistrement du mot « Électronique ».**

❑ Conception

Un tutoriel permettant de transférer vos données sur Audacity est disponible dans la Toolbox sur Boostcamp.

❑ Validation

Après le transfert série et l'utilisation du logiciel Audacity, vous devriez avoir un fichier .wav (ou alors .mp3) avec un enregistrement du mot « Electronique ».



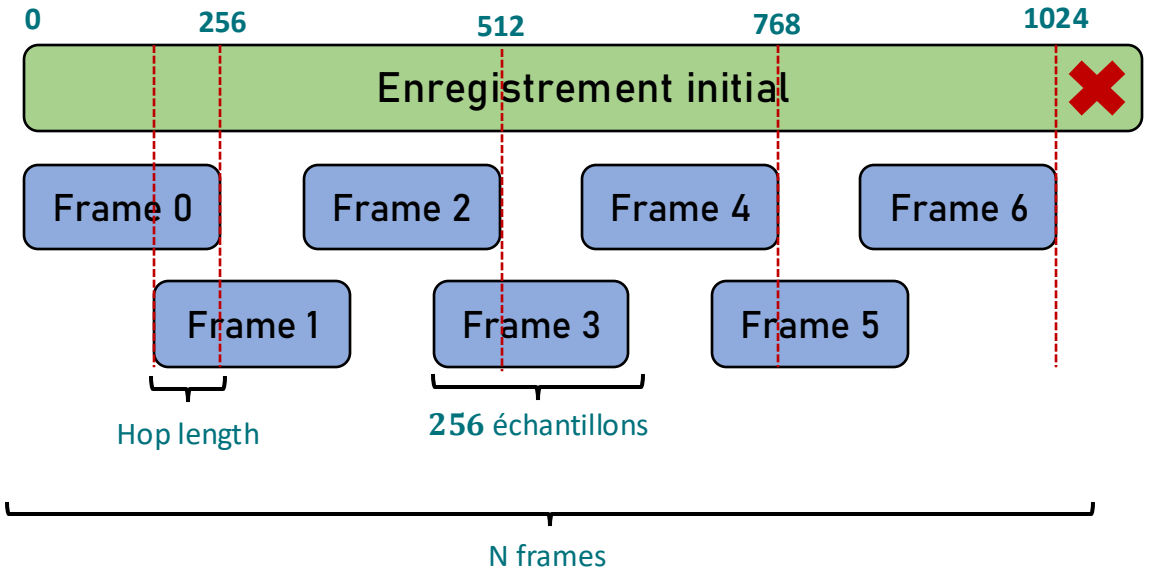
FP4 : Caractériser le timbre vocal

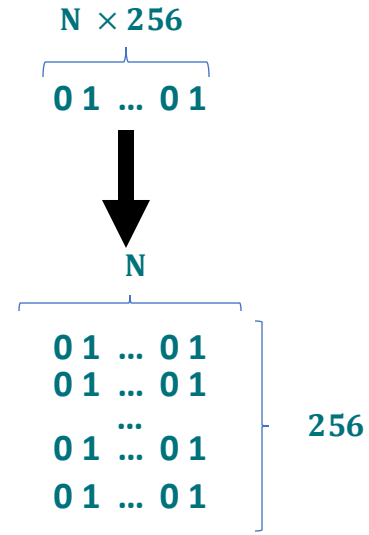
ET5. le signal audio doit être séparé en frames de 256 échantillons

❑ Conception

L'**algorithme MFC** permet de générer un nombre de coefficients à partir d'un audio donné. Afin de prendre en compte l'évolution temporelle du signal audio, séparer l'enregistrement en frame de 256 échantillons qui vont chacun générer des coefficients, appelés MFCC.

En plus de la séparation sous forme de frames il est indispensable de superposer les frames entre elle afin de garantir la continuité de l'application des algorithmes.





Attention :
Bien faire attention ne pas perdre des données à la fin de l'enregistrement à cause d'une mauvaise correspondance

❑ Validation

A l'aide de PuTTY vous pouvez importer le signal audio une fois séparé en frames pour afficher le signal audio d'une frame et de l'ensemble des frames, en mettant en évidence le bon fonctionnement du recouvrement

ET6. Votre Arduino Due doit pouvoir générer 13 MFCCs à partir de d'un enregistrement d'une seconde.

❑ Conception

La dernière étape du conditionnement du signal avant le réseau de neurones est l'application de l'algorithme MFC
L'objectif premier de l'algorithme est de **réduire la taille de la data caractérisant un signal sonore** afin de par la suite pouvoir le reconnaître suffisamment rapidement.

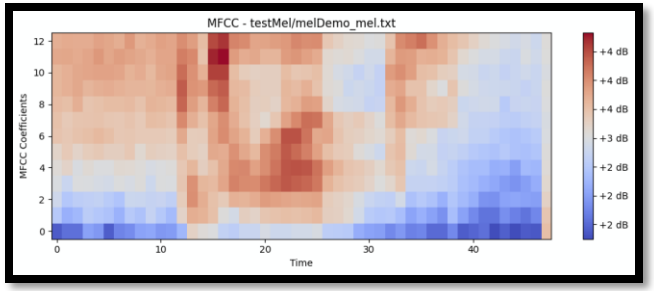
Une implémentation de l'algorithme est proposée sur GitHub du Pole Electronique.
Vous devrez, pour chaque frame appliquer les différentes étapes de l'algorithme.

❑ Validation

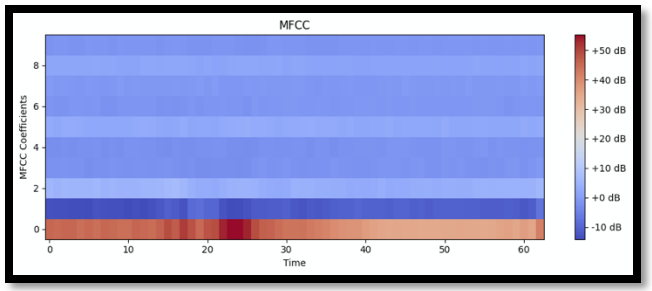
Afin de valider le fonctionnement de la MFC, il faut valider que chaque partie de l'algorithme s'applique correctement.
Générer les graphiques suivants et les analyser finement afin de valider la bonne implémentation de la FP4 :

- Signal audio avant / après le filtre de Pre-emphasis
- Affichage de la FFT calculée par la carte pour une fréquence de 1Khz (utilisation de l'outil « Tracer le spectre » d'Audacity)
- Affichage des coefficients juste après la convolution avec le banc de filtre MEL sous forme de spectre (script python fourni sur Github). Tous les coefficients de chaque frame doivent être affichés :

Exemple de MFCC (12 coefficients) :

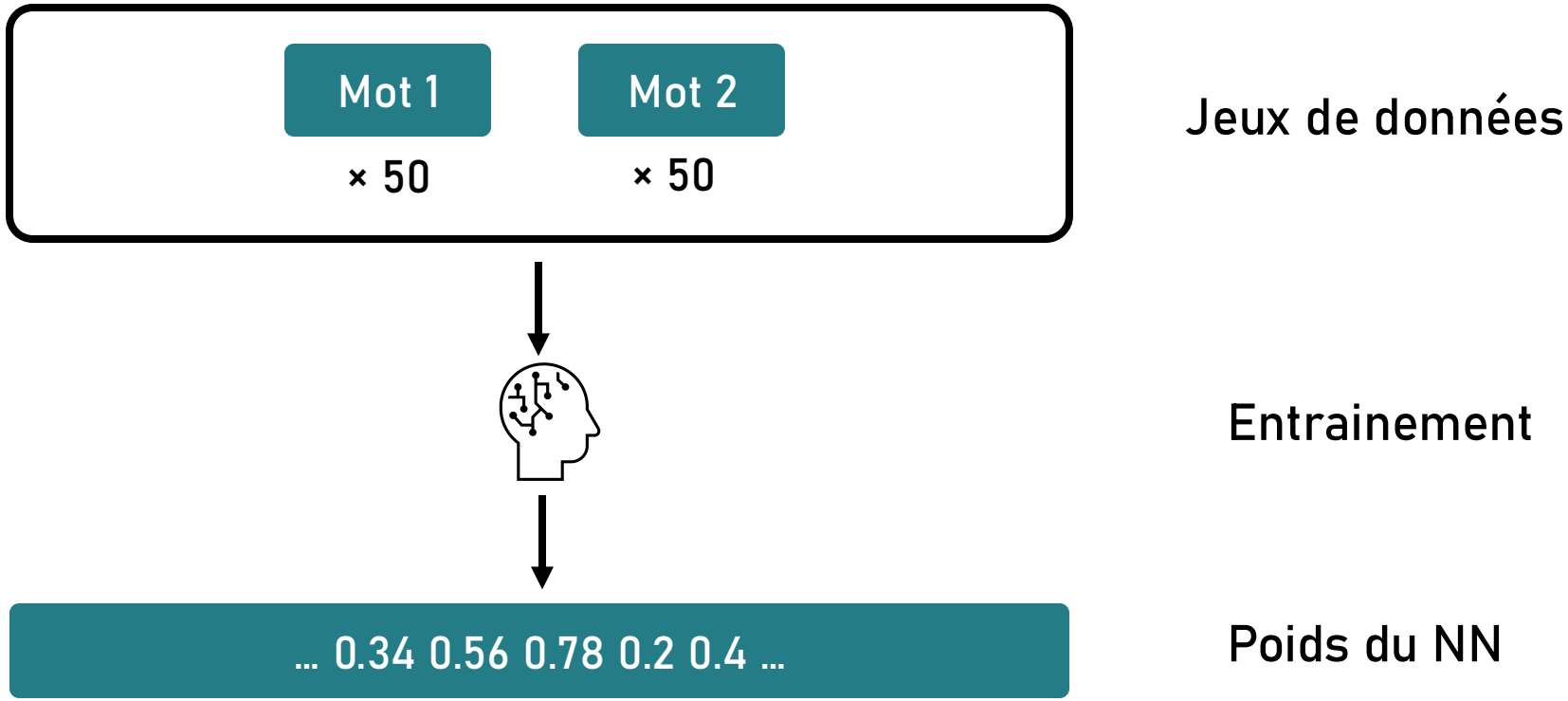


Après le calcul de la DCT :



FP5 : Identifier les résultats attendus

Le projet doit être capable d'identifier les différents enregistrements en fonction du mot qui a été prononcé. Pour cela, il est proposé d'utiliser un **réseau de neurones convolutionnel** (CNN) dont l'architecture est à concevoir.
Dans un premier temps, un dataset est fourni avec le sujet afin de faciliter le développement de l'architecture du réseau de neurones.
Un jeu de données différents sera à créer après la validation de la fonction FP4.
Lorsque l'erreur du réseau de neurones sera suffisamment faibles les poids du réseau pourront être réutilisés dans le programme afin de faire de la détection.



ET7. La MSE des données de test fournis doit être inférieur à 0,05 après entrainement

❑ Conception

Vous disposez de deux ***datasets*** sur le GitHub du Pôle Électronique, avec les MFCC liés aux mots « Bleu » et « Rouge ».
Le premier *dataset* de 100 tableaux de coefficients devra être utilisé pour entraîner votre réseau de neurones.
Le second *dataset* de 10 tableaux de coefficients devra être utilisé pour valider l'entraînement du réseau de neurones.
Vous devrez concevoir l'architecture d'un réseau de neurones de type CNN (à vous de définir les paramètres du réseau) capable de classifier les deux mots qui ont été prononcés.

❑ Validation

L'entrainement du réseau de neurones doit permettre d'avoir une erreur quadratique moyenne sur les données de test inférieure à 0,05. Attention, il est interdit de d'appliquer une *back-propagation* sur les données de test !
Noter les données importantes liées à l'entrainement : durée, MSE des données d'entrainement, nombre de cycles. Expliquer les actions mises en œuvre pour optimiser le réseau.

ET8. Votre jeu de données d'entrainement doit contenir au moins 100 éléments

❑ Conception

Faire plus de **50 enregistrements** pour chaque mot (il faut a minima 2 mots donc 50 x 2 = 100 éléments). Un bouton pourra être utilisé afin de démarrer un enregistrement d'une durée fixe. Tous les enregistrements doivent être mis sous la même forme que les données d'entrainement. Pour cela : transférer sur un PC, valider avec Audacity, puis les convertir sous forme de tableau C.
NB : Pensez à normaliser vos données (cf TP sur les réseaux de neurones)

❑ Validation

Fournir un tableau sample[100][256][N] avec N en fonction du nombre de frames. Ce tableau contient les MFCCs qui servirons de données d'entrainement pour le réseau de neurones.

Il peut aussi être intéressant de créer des enregistrements supplémentaires qui vous permettrons de tester votre détection sur des mots qui ne sont pas dans le jeu de données d'entrainement (Attention, un réseau avec une MSE faible sur les données de test n'est pas forcement efficace sur des nouvelles données !).

FP6 : Classifier les enregistrements audios

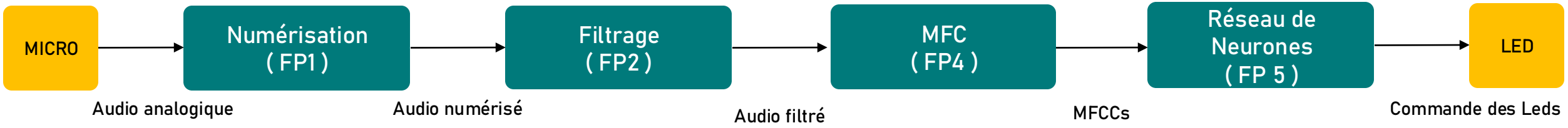
ET9. **Le réseau de neurones doit pouvoir classifier les 10 mots prononcés avec moins de 5% d'erreur.**

❑ Conception

A l'aide uniquement des poids du réseaux entrainé précédemment, mettre en place un code permettant de **classifier un enregistrement** directement à partir du microphone.
Utiliser un bouton afin de démarrer un enregistrement d'une durée fixe. Ainsi, lorsque l'utilisateur appuie sur le bouton d'enregistrement, des LEDs indiquent le mot détecté.

❑ Validation

Après entrainement, le réseau doit être capable de reconnaître des mots prononcés avec moins de 5% d'erreur (au moins 10 mots).
Montrer que votre réseau est robuste à des perturbations comme du bruit de fond.



Fonctionnalités avancées

Deux branches de fonctionnalités avancées vous sont proposées.
Vous êtes libres de choisir l'un ou l'autre (ou les deux, avec saturation des points à 4). Vous serez évalués sur la pertinence des solutions techniques mises en œuvre. Pour valider le moindre point de Fonctionnalité Avancée, vous devez présenter des courbes de résultat montrant les performances / précisions avant et après amélioration.

Performance

3 pts

- L'objectif est de réduire la durée entre l'enregistrement de l'audio et la détection par le réseau afin d'avoir une détection de mots en temps réel sans bouton (type Alexa)

Vous présenterez les techniques mises en œuvre, voici des exemples :
- DMA pour enregistrer l'audio pendant le traitement du signal
- Détection automatique du début d'enregistrement
- Autres …

Précision

3 pts

- L'objectif est d'améliorer la précision de votre détection ou d'ajouter plus de deux mots à la classification

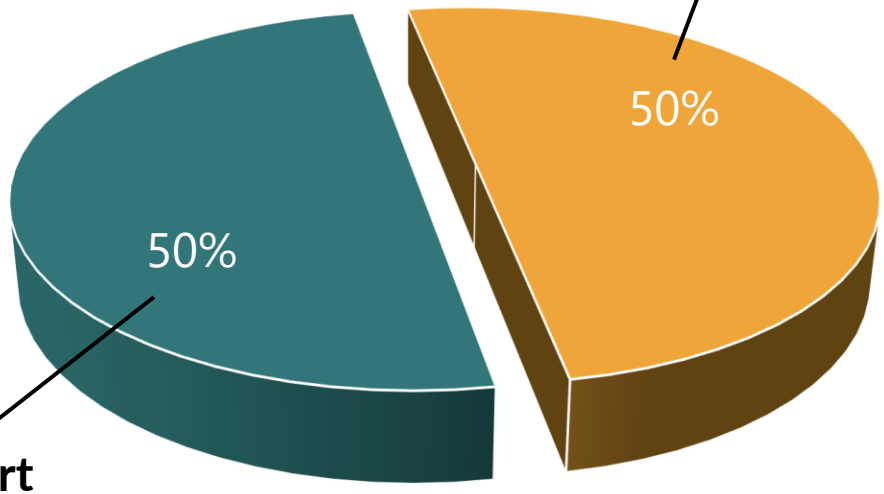
Vous présenterez les techniques mises en œuvre, voici des exemples :
- Entrainement de votre IA avec tensorFlow (exemple de code).
- Reconnaissance du locuteur.
- Autres …

Dans les deux cas, une étude paramétrique sera attendue pour voir l'impact du nombre de MFCC.

Notation

Soutenance

Voir le document « **Comment préparer une soutenance** » sur « La Toolbox »



Rapport

Voir le document « **Comment rédiger un rapport** » sur « La Toolbox »

Obligation d'utiliser le template de rapport de la page Moodle « La Toolbox »
Voir le document « Le pôle électronique » pour le plagiat et la gestion des retards.

Déroulement de la soutenance

8 min

❶ Présentation orale

But : être capable de synthétiser 4 semaines de travail sur le projet

- Diaporama obligatoire
- 8 min pour convaincre
- **Pas un résumé du rapport**
- **Présenter des schémas électriques, des alogorigrammes, des courbes de résultats et les tests.**

4 min

❷ Démonstration

But : s'assurer des performances du système

- Démonstration de la maquette
- Validation par le jury

6 min

❸ Discussion

But : s'assurer de la bonne compréhension des briques du projet

- Questions sur le projet
- Anticiper des questions et prévoir des slides de back up

- **Tester le projecteur la veille**
- **S'entrainer pour être pertinent et tenir les temps**
- **Apporter un adaptateur USB C – HDMI si nécessaire**
- **Être prêt à commencer en entrant dans la salle (ordinateur allumé et prêt à être branché)**

Un projet parfaitement fonctionnel n'aura la note maximale que s'il est correctement présenté et chaque notion maîtrisée. Voir ppt « Comment préparer une soutenance »

Évaluation de la soutenance

Intitulé	Note maximale		Critères de réussite
Forme	**2 point**		Qualité générale des slides Orthographe, alignements des figures, homogénéité (police, taille de caractère, etc.) -0,25 par erreur Nom des axes sur les graphiques, graduations, etc.
Fluidité de la soutenance	**-1 à 0 pt**		Présentation claire et efficace
Maitrise de la technique	**2 points**		Qualité des explications techniques Explications claires (protocoles de mesure notamment), maitrisées et véridiques
Use case	**1 point**		Présentation du use case choisi et diagramme fonctionnel associé
Diagrammes techniques (HW et SW)	**1 point**		Schématique sur KiCAD et Logigramme, chacun notés sur 0,5
Numérisation	**2 points**		**ET1.** L'ADC de l'Arduino due doit être capable d'échantillonner le signal à une fréquence de 32KHz.
Conditionnement	**2 points**		**ET2.** Les fréquences supérieures à 4Khz doivent être atténuées d'au moins 30dB **ET3.** Pour chaque échantillon, le filtrage doit être fait en moins de 31 µs
Caractérisation	**2 points**		**ET4.** Une lecture sur Audacity doit pouvoir restituer de manière claire l'enregistrement du mot « Électronique ». **ET5.** le signal audio doit être séparé en frames de 256 échantillons **ET6.** Votre Arduino Due doit pouvoir générer 13 MFCCs à partir de d'un enregistrement d'une seconde.
Entraînement	**2 points**		**ET7.** La MSE des données de test fournis doit être inférieur à 0,05 après entrainement **ET8.** Votre jeu de données d'entrainement doit contenir au moins 100 éléments
Classification	**2 points**		**ET9.** Le réseau de neurones doit pouvoir classifier les 10 mots prononcés avec moins de 5% d'erreur.
Aboutissement du use case choisi	**2 points**		Le use case choisi est abouti et fonctionnel Prise en compte ici de la difficulté technique du use case choisi Éventuelle maquette, cable management, fluidité de l'interface graphique s'il y en a une, etc.
FA : Performance	**max 4 points**	**3 points**	Réduction de la durée entre l'enregistrement et la détection Solution(s) et résultat(s)
FA : Précision		**3 points**	Amélioration de la précision de la détection Solution(s) et résultat(s)
IA non embarquée	**0 ou -12**		Une IA développée sur ordinateur mais non fonctionnelle sur Arduino vous expose à un malus forfaitaire de -12 (hors sujet) pour non suivi de la méthodologie. Il vaut mieux ne rien présenter

Évaluation du rapport

Intitulé	Note maximale	Critères de réussite
Forme	**2 pts**	Figures légendées et renvoi aux figures Orthographe Qualité des figures Axes et graduations claires sur les graphiques Pas de code mais des algorigrammes
Objectif, contexte, problématique, sources, annexes	**-1 à 0 pt**	Voir *template*
L'équipe et diagramme de GANTT	**1 pt**	Les tâches sont bien réparties et bien ventilées dans le temps Diagramme clair et commenté
Conception	**3 pts**	Architectures fonctionnelle (pas de nom de composant !), matérielle et logicielle commentées et digne d'un professionnel – chaque diagramme noté sur 1 point
Développement	**5 pts**	Qualité des explications techniques pour chaque étape / modules : comment cela fonctionne et comment cela a été fait.
Tests et validation	**8 pts**	Tests unitaires (courbes) pertinents et fonctionnels Qualité des analyses Conclusion pour chaque résultat
Bilan	**1 pts**	État d'avancement du projet Pertinence et limites de la solution technique Bilan sur le travail d'équipe
Non utilisation du template	**-2 pts**	-2 pts pour non utilisation du *template* ou modification trop importante (police trop fantaisiste, taille de caractère trop grande, plan trop modifié, etc.)
Format .PDF	**-2 pts**	Rapport remis dans un autre format que .PDF
Rapport rendu à temps et sur Boostcamp	**-2 pts**	-2 pts si le rapport est rendu en retard ou par mail dans les 6h qui précèdent la fermeture du dépôt.

ECE
ÉCOLE D'INGÉNIEURS
ENGINEERING SCHOOL

À vous de jouer.