

Présentation de la SAÉ : Concevoir et réaliser un prototype

Table des matières

1	Présentation de la SAÉ	2
1.1	Contexte : le sonomètre	2
2	Description de la SAE	3
3	Cahier des charges techniques	5
3.1	Contraintes de routage	5
3.2	Contraintes mécaniques en mils (millièmes de pouce)	5
3.3	Contraintes de programmation	5
3.3.1	Convention de nommage	5
4	Les ressources mises à votre disposition	7
4.1	Documents	7
4.2	Ressources matérielles	7
4.3	Ressources logicielles	7
5	Les livrables	7
5.1	Livrables CAO	7
5.1.1	Jalon 1 : Schématique (Vérification pendant les cours)	7
5.1.2	Jalon 2 : Typon (Projet complet à rendre avant le 16 décembre 2024)	7
5.2	Livrables informatiques	7
5.2.1	Jalon 1 (TP d'info1 entre le 16/12 et le 20/ 12 selon le groupe)	7
5.2.2	Jalon 2 (lors de l'oral de contrôle de janvier)	8
5.3	Présentation orale (semaine du 13 au 17 janvier 2025)	8
5.4	Rendu final (avant le 13 janvier 2025)	8

1 Présentation de la SAÉ

Le but de cette SAÉ est de vous familiariser avec les notions de conception et de réalisation de cartes électroniques en utilisant un logiciel de CAO et la conception de logiciel avec le langage Python. Le support utilisé dans cette SAE est un sonomètre. Pour développer vos deux compétences **concevoir** et **réaliser** en BUT 1 nous allons travailler à l'élaboration d'un sonomètre.

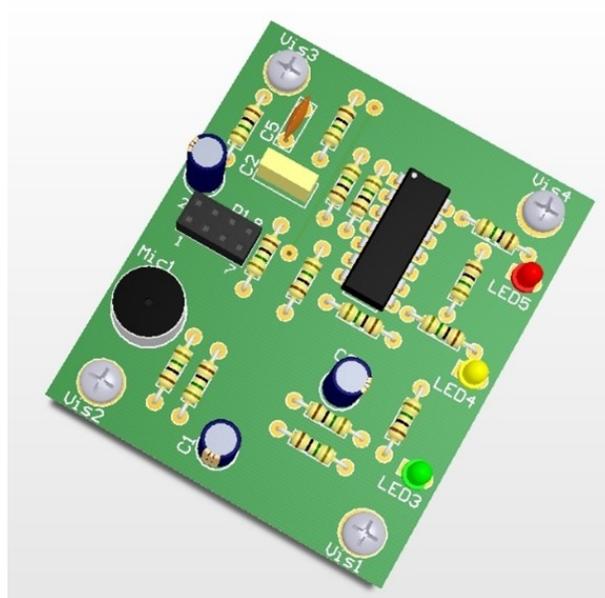


FIGURE 1 – Sonomètre

1.1 Contexte : le sonomètre

Première approche : On comprend mieux une nouvelle notion si l'on part d'éléments connus. Dans votre vie de tous les jours vous avez certainement déjà vu un vumètre. Il permet de visualiser au moyen de leds (autrefois d'afficheurs à aiguille) la mesure électrique d'un son dans un dispositif audio tel qu'une table de mixage.



Le sonomètre :

- **Les points communs aux vumètre et au sonomètre :**
 - Ils travaillent tous les deux dans le domaine des fréquences audibles.
 - Ils disposent tous les deux d'un système d'affichage.
- **Le sonomètre diffère du vumètre par les points suivants :**
 - C'est un appareil industriel.
 - Il dispose de son propre microphone intégré.
 - Il est étalonné pour relever un niveau de pression acoustique (en dB) ou d'intensité acoustique (en W/m^2).



FIGURE 2 – Exemple d'écrans de sonomètre

2 Description de la SAE

Durant cette situation d'apprentissage, vous allez être amené à réaliser :

- Une carte électronique : sonomètre,
- Un programme informatique : interface homme-machine (IHM)

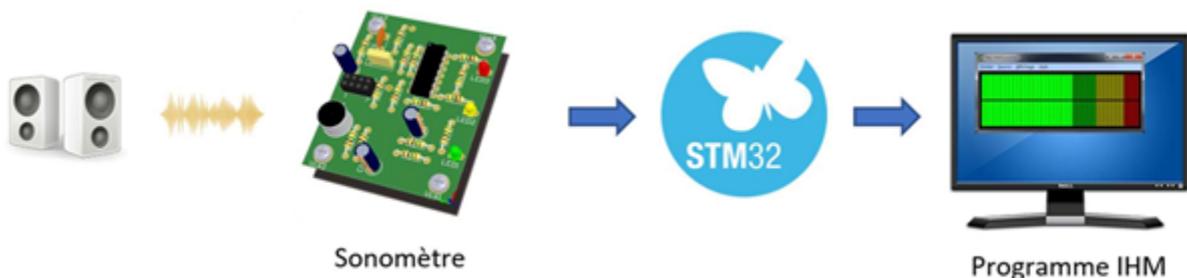


FIGURE 3 – Mise en situation de la SAE

À partir d'une source sonore à mesurer, représentée par les deux enceintes sur la figure précédente, la carte électronique « sonomètre » indiquera par l'intermédiaire de 3 LED (Light-Emitting Diode ou DEL : Diode ÉlectroLuminescente) si le niveau sonore ambiant est confortable (vert), trop fort (jaune) ou beaucoup trop fort (rouge).

Une carte d'interface, fournie et nommée STM32 sur la figure, permettra de convertir les niveaux sonores issus du sonomètre (signal_audio_PC0) en données numériques

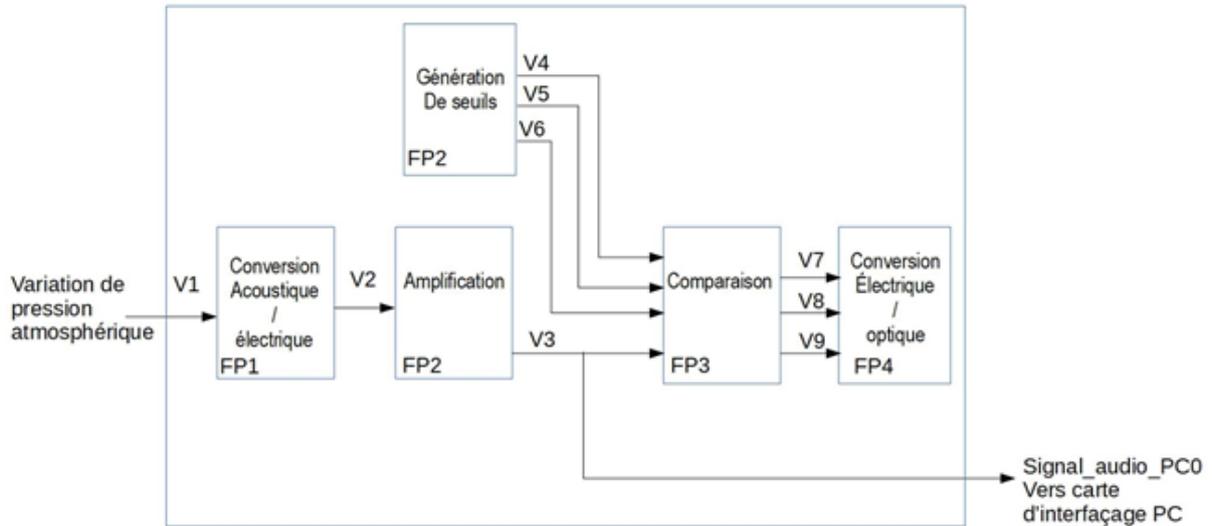


FIGURE 4 – Schéma fonctionnel de la carte "sonomètre" à réaliser

pouvant être traitées et affichées sur un écran d'ordinateur. Cette carte est basée sur un Convertisseur Analogique Numérique (CAN ou ADC : Analog to Digital Converter) qui à intervalle de temps régulier vient prélever la valeur (prendre un « échantillon ») du signal « signal_audio_PC0 » et la convertit en une valeur numérique sur 8 bits dans notre cas. Comme on peut le voir sur la figure ci-dessous (exemple sur 4 bits), à chaque «

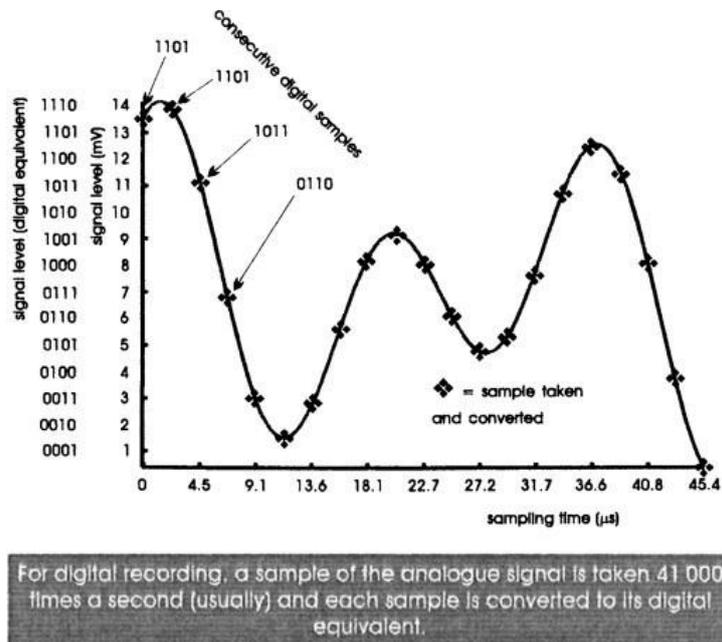


FIGURE 5 – Principe de conversion numérique

échantillonnage » correspond une valeur numérique image de la valeur analogique (courbe en trait continu).

À cette information, la carte d'interface ajoutera également une information sur l'état des LED (allumée ou éteinte).

Afin de savoir où se situe chaque information dans le signal envoyé au PC, la carte STM32 émettra périodiquement une trame (succession d'information) de 4 octets dont le

format est le suivant :

Entête ADC	Valeur ADC sur 8 bits						Entête Leds	Leds On/Off								
(AB)	LSB						MSB	(CD)	0	0	0	0	0			

- **Entête ADC** est un octet qui vaudra toujours AB en hexadécimal. Ainsi dès que vous recevez AB vous savez que l’octet suivant correspond à la valeur issue d’un Convertisseur Analogique Numérique (CAN ou ADC : Analog to Digital Converter).
- **Entête Leds** vaudra toujours CD en hexadécimal. L’octet qui suivra sera un « miroir » de l’état des 3 leds rouge, jaune et verte de votre sonomètre. Il indique l’état (on / off de chacune des 3 leds) suivant le code :
 - Si aucune led n’est allumée la valeur de l’octet sera (0000 0000) = (00)
 - Si la led verte est allumée la valeur de l’octet sera (0000 0001) = (01)
 - Si les leds orange et verte sont allumées la valeur de l’octet sera (0000 0011) = (03)
 - Si les leds rouge, orange et verte sont allumées la valeur de l’octet sera (0000 0111) = (07)

Vous aurez également à concevoir la partie logicielle de visualisation du niveau sonore sur l’écran de cet ordinateur. Pour cela vous devrez :

- Acquérir la trame émise par la carte d’interface STM32,
- Exploiter la trame reçue, c’est-à-dire extraire le niveau sonore et l’état des 3 leds,
- Mettre à jour l’affichage.

3 Cahier des charges techniques

3.1 Contraintes de routage

- Le routage sera réalisé en simple face (Bottom Layer).
- Le nombre de straps (horizontal et/ou vertical) autorisés est de 4.
- Nom d’un membre du groupe sur la face cuivre.
- Utilisation des règles de routage du fichier : `Regles_Routage_IUT.RUL` (rappelées en annexes).

3.2 Contraintes mécaniques en mils (millièmes de pouce)

Nous considérons ici que votre carte devra s’intégrer dans un boîtier déjà existant de l’entreprise. La position des 4 vis, des 3 leds ainsi que les dimensions de la carte sont donc figées.

3.3 Contraintes de programmation

3.3.1 Convention de nommage

Les noms de variables devront avoir un préfixe (voir ci-dessous) suivi d’un nom qui permet de décrire le rôle de cette variable. Chaque nouveau mot du nom commencera par une majuscule. La convention des préfixes sera la suivante :

- `int` pour les variables entières
- `float` pour les variables réelles

fonction valable.

4 Les ressources mises à votre disposition

4.1 Documents

- La schématique du sonomètre ([Sonometre.pdf](#))
- La liste des footprints des composants ([BOM.pdf](#))
- Le tutoriel KiCad (logiciel de CAO dédié à la réalisation de cartes électroniques)
- Les tutoriels sur l'utilisation des bibliothèques Python
- Les cours du semestre 1

4.2 Ressources matérielles

- Plaque de test de type Labdec.
- Composants électroniques disponibles dans le département GEII.
- Matériel de laboratoire, banc de test dans chaque salle et ordinateurs de l'IUT.

4.3 Ressources logicielles

- KiCad avec la bibliothèque de composants de l'IUT « LIB_IUT »
- Python sous toutes ses formes : NoteBook Jupyter, Divers IDE (Thonny, VS code...)
- VSPE (<https://eterlogic.com/products.vspe.html>) pour créer des ports virtuel et tester votre programme avant d'avoir fini sa carte.

5 Les livrables

5.1 Livrables CAO

5.1.1 Jalon 1 : Schématique (Vérification pendant les cours)

- Une schématique KiCad par étudiant, incluant la schématique du sonomètre refaite sous KiCad et l'assignation des footprints.

5.1.2 Jalon 2 : Typon (Projet complet à rendre avant le 16 décembre 2024)

Les fichiers nécessaires au tirage de la carte devront être validés lors de la dernière séance de SAÉ encadrée et conformes au cahier des charges, en incluant le nom du réalisateur.

5.2 Livrables informatiques

5.2.1 Jalon 1 (TP d'info1 entre le 16/12 et le 20/ 12 selon le groupe)

- Chaque groupe passera à l'oral pendant ce TP et devra présenter :
- La répartition du travail entre chaque membre du groupe.

- Une fenêtre Tkinter, avec les widgets Tkinter nécessaires donnant le visuel du sonomètre (Les boutons et autre visuels seront présents mais pas forcément encore actifs)
- Un programme console permettant simuler 400 bits sortant du STM32 au bon format (comme décrit plus haut) et de transformer les données en une chaîne de caractères pour extraire la valeur maximale après le bit ‘CD’.
- Un programme console qui ouvre deux ports virtuels (créés avec VSPE (option ‘pair’)) et qui permet d’écrire des données sur un port et de les recevoir sur l’autre.

5.2.2 Jalon 2 (lors de l’oral de contrôle de janvier)

1. Programme fenêtré permettant de se connecter à un port virtuel qui recevra des données simulées et qui affichera le niveau sonore (LEDs allumées) ou en mode VU-mètre en fonction de ces données simulées.
2. Programme fenêtré permettant de se connecter à la carte et d’afficher en temps réel le niveau sonore (LEDs allumées) ou en mode VU-mètre.

5.3 Présentation orale (semaine du 13 au 17 janvier 2025)

Prévue pour la semaine du 13 au 17 janvier 2025. La présentation inclura une démonstration de la carte opérationnelle et du logiciel développé, avec 5 minutes d’exposé suivies de 10 minutes de questions.

5.4 Rendu final (avant le 13 janvier 2025)

Chaque groupe devra rendre un rapport finale se composant :

1. Le programme final commenté en .py
2. Un carnet de bord (Répartition des tâches, recherches effectuées, difficultés rencontrées, solutions envisagées, qui a fait quoi à quel moment et pendant combien de temps, dates et heures passées en concertation)