



Institut de Recherche  
en Informatique de Toulouse  
CNRS - INP - UT3 - UT1 - UT2J

Rapport de stage de première année  
Juin/Juillet 2022

**Création d'une GUI d'analyse des ECG/EEG**

# TABLE DES MATIÈRES

- 1 - Les étapes de conception**
- 2 - Les étapes de programmation**
- 3 - Manuel utilisateur**
- 4 - Bilan du projet**
- 5 - Bibliographie**

## Les étapes de conception

Les premiers jours du projet, il a fallut que je me documente sur les travaux déjà réalisés par étudiants de l'ENSEEIHHT qui correspondait à la partie back-end de la synchronisation de signaux EEG/ECG. J'ai aussi passé du temps à étudier des modèles de systèmes de surveillance cardiaque à distance qui existent déjà (<https://tateeda.com/project/ventrilink>).

Ensuite, aux côtés de Mme Ouederni, nous avons listé l'ensemble des éléments graphiques qui devaient être présents sur l'interface de l'application :

### Partie EEG :

- Affichage de l'activité liée à l'EEG la plus récente
- Bouton au-dessus de l'EEG pour avoir accès à un calendrier pour accéder à d'autres jours
- 2 boutons en haut à droite et en haut à gauche pour accéder au jour suivant et au jour précédent
- Possibilité de zoomer sur des parties de l'EEG dans une certaine limite
- Affichage de la date d'activité
- Affichage du timestamp
- Affichage du sampling rate
- Affichage de l'heure locale

### Partie ECG :

- Affichage de l'activité liée à l'ECG la plus récente
- Bouton au-dessus de l'ECG pour avoir accès à un calendrier pour accéder à d'autres jours
- 2 boutons en haut à droite et en haut à gauche pour accéder au jour suivant et au jour précédent
- Possibilité de zoomer sur des parties de l'ECG dans une certaine limite
- Affichage de la date d'activité
- Affichage du timestamp
- Affichage du sampling rate
- Affichage de l'heure locale

### Partie Réseau :

- Affichage du nom de la connexion
- Affichage de l'adresse IP
- Débit de la connexion
- Topologie de la connexion
- Port de la connexion



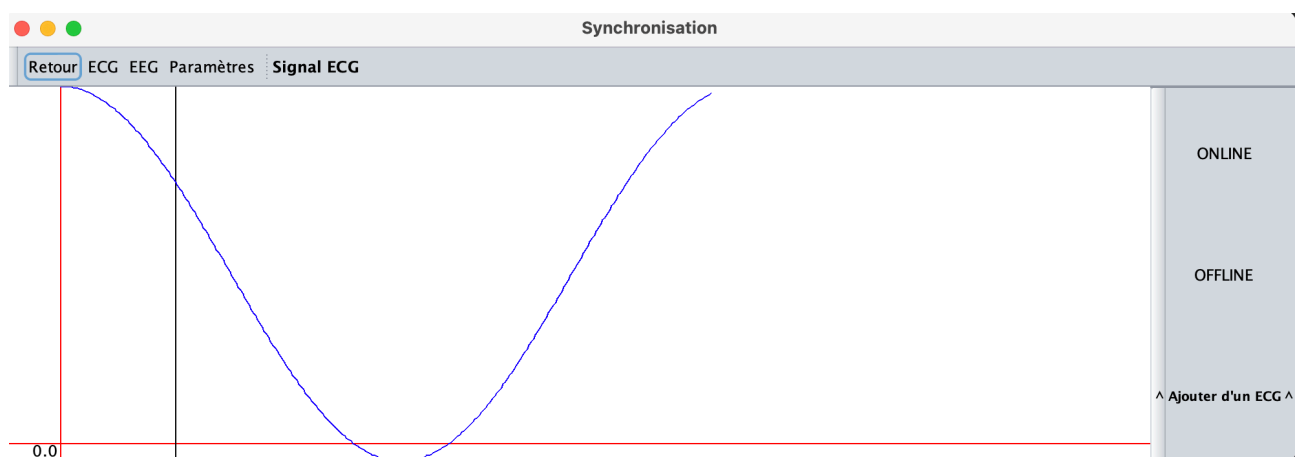
C'est une fois après avoir réalisé une ébauche de l'interface graphique souhaitée que j'ai pu débiter la partie programmation afin de la mettre en place.

## Les étapes de programmation

Le premier défi à réaliser était d'éditer et exploiter les valeurs d'une base de données Excel. En effet, j'ai, au préalable, créé une base de données de signaux afin de réaliser l'affichage de ces-derniers par la suite.

Pour cela, j'ai donc travaillé sur les librairies Swing et la version 3.9 d'Apache. Swing permettait de coder un composant graphique d'affichage de courbes et l'outil Apache a été utilisé afin de créer les attributs contenant les valeurs prises par les signaux à partir des données du fichier Excel de référence (« SIGNAL.xlsx »).

Après plusieurs jours de travail, une première version a pu être conçue :



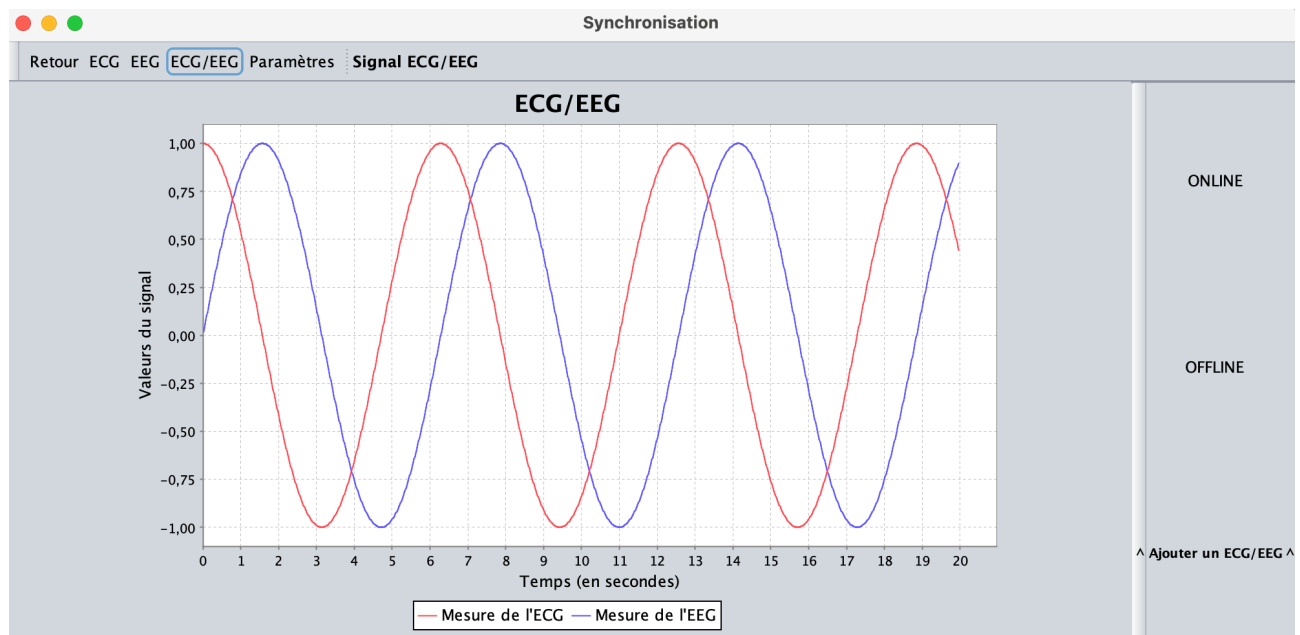
**Image 1 : Itération 1 de l'outil d'affichage**

Ce premier jet permet un affichage peu clair et pas du tout précis. Il a fallu mener des recherches pour un affichage plus pertinent pour des analyses de signaux.

C'est dans cette démarche j'ai découvert une toute nouvelle librairie qui est « JFreeChart » pour un nouvel affichage des graphiques. Il a fallu adapté le code réalisé précédemment à l'utilisation de JFreeChart; la bibliothèque Apache quant à elle reste toujours utile pour l'exploitation du fichier Excel.

## STAGE 1A

Avec l'utilisation de cette nouvelle librairie, j'ai donc réalisé une nouvelle version de l'affichage graphique plus complète et plus lisible :



**Image 2 : Itération 2 de l'outil d'affichage**

Toutefois, malgré l'affichage conforme à l'idée qu'on s'en faisait au tout début du projet, un problème persistait qui était le passage d'un type de signal à un autre. En effet, lorsque je souhaitais passer d'un ECG à un EEG, l'affichage ne suivait pas la sélection du bouton correspondant au signal voulu.

Il a donc fallut réaliser une dernière itération pour régler ce souci et ajouter des éléments d'affichage sur les différentes pages « ONLINE » et « OFFLINE ».

Enfin, un nouveau bouton « Translater » a été ajouté qui mène vers une page permettant en théorie de décaler le signal d'un certain nombre de secondes (nombre à entrer). Le tout n'est pas encore fonctionnel, un problème persiste. En effet, lorsqu'on applique la translation, le fichier Excel devient erroné/inutilisable.

## Manuel utilisateur

Le dossier des fichiers à exécuter est entièrement basé sur les librairies JFreeChart (version 1.0.1), Swing et Apache (version 3.9). Les différents fichiers sont les suivants :

## STAGE 1A

- Signal.java
- Graph.java
- MenuPrincipal.java
- AfficheurSynchronisationManuelle.java
- AfficheurSynchronisation.java
- EditionFichier.java
- Online.java

(*Remarque* : Les boutons « Paramètres » et « OFFLINE » ne sont pas fonctionnels donc ils n'ont aucun fichier Java associé.)

### **Signal.java**

Cette classe permet de créer un signal, les valeurs de temps et les valeurs prises par le signal sont dans des tableaux parmi les attributs de la classe (« temps » et « valeurs »).

La création de l'élément « Signal » se fait à partir du nom du fichier Excel et du nom de la feuille souhaitée.

### **Graph.java**

Cette classe utilise les bibliothèques Swing et JFreeChart. Elle permet de dessiner le composant graphique correspondant au tracé des deux signaux ECG et EEG. Les deux éléments de type « Signal » sont dans les attributs de classe et sont obtenus à partir des chemins des 2 fichiers Excel qui correspondent.

De plus, dans le constructeur de classe, l'attribut « type » permet de choisir le/les signal/signaux à afficher. En effet :

- Pour type = « ECG », seul le signal ECG (signal) sera affiché.
- Pour type = « EEG », seul le signal EEG (signalbis) sera affiché.
- Pour un autre type, les deux signaux ECG/EEG seront affichés sur le composant graphique.

### **MenuPrincipal.java (LANCER LE MAIN)**

Cette classe est un élément « JFrame », il correspond à la page principal. Le bouton « Synchronisation » présent sur cette page permet d'afficher/synchroniser les signaux souhaités.

## **AfficheurSynchronisationManuelle.java**

Cette classe correspond elle aussi à un élément « JFrame », c'est la page dédiée à la synchronisation manuelle des signaux (qui n'est pas fonctionnelle comme précisé précédemment). Elle contient les éléments graphiques permettant une telle synchronisation :

- Un **JTextArea** (« *temps* ») pour la zone de texte à compléter avec le nombre de secondes dont on veut traduire les signaux.
- Un **JComboBox<String>** (« *jComboBox* ») qui est le menu déroulant pour sélectionner le signal à traduire.
- Un **JButton** (« *btn* ») pour appliquer la traduction.

La classe permet d'éditer le fichier Excel correspondant au signal sélectionné et permet de traduire toutes les valeurs directement sur ce fichier.

## **AfficheurSynchronisation.java**

Cette classe est elle aussi un élément « JFrame » et c'est la page d'affichage des signaux. Six composants de type « JButton » sont présents sur la barre d'outils supérieure :

- *btnECG* : Permet d'afficher le signal ECG sur le centre de la page
- *btnEEG* : Permet d'afficher le signal EEG sur le centre de la page
- *btnECGEEG* : Permet d'afficher les signaux ECG/EEG sur le centre de la page
- *btnRetour* : Permet de retourner sur le menu principal
- *btnTranslator* : Mène vers la page « AfficheurSynchronisationManuelle »
- *btnParametres* : Permet d'afficher la page des paramètres (*non fonctionnelle*)

De plus, sur la barre d'outils de droite, deux boutons sont présents :

- *btnONLINE* : Mène vers une page ONLINE (précisée par la suite)
- *btnOFFLINE* : *Non fonctionnelle*

## **EditionFichier.java**

Le constructeur de la classe prend en paramètres le nom d'un fichier Excel et la feuille correspondant au signal qu'on souhaite éditer.

Dans l'attribut de classe « *body* » est présent l'ensemble du contenu du fichier Excel.



## **Online.java**

Cette page permet en théorie d'ajouter un signal à partir d'une adresse réseau, d'un numéro de port et d'un nom de signal. Pourtant, je n'ai pas pu aller plus loin étant donné que la base de données des signaux était un ensemble de fichiers Excel.

## **Bilan du projet**

Ce projet m'a permis de consolider mes bases en Java, m'a fait découvrir de nouvelles bibliothèques telles que JFreeChart ou encore Apache et surtout, cela m'a appris à développer une méthodologie afin de mener mes recherches lorsque je rencontre un problème technique.

Le fait d'avoir été complètement autonome sur ce projet a été formateur et le sujet traité m'a beaucoup passionné.

Je remercie Mme Ouederni de m'avoir accompagné sur toutes les étapes de ce projet ainsi que les différents membres de l'IRIT avec qui j'ai pu discuter sur des plans du sujet.

## **Bibliographie**

- *Documentation sur un modèle de système de surveillance cardiaque à distance qui existe déjà* : <https://tateeda.com/project/ventrilink>
- *Vidéo de découverte sur la bibliothèque JFreeChart* : [https://youtu.be/cw31L\\_OwX3A](https://youtu.be/cw31L_OwX3A)
- *Vidéo sur la bibliothèque Swing* : <https://youtu.be/l2gzJGd8i3s>
- *Vision du Modèle - Vue - Contrôleur* : <https://www.lemagit.fr/definition/modele-vue-controleur-MVC>