

Projet A3 BIOST : ECG (1/3)

Introduction du projet

L'objectif de ce projet est d'élaborer un système de mesure de type électrocardiogramme (ECG), d'obtenir des échantillons, d'en extraire des informations utiles pour finalement les visualiser. Dans un second temps le travail porte sur une comparaison des informations entre individus et des situations différentes.

Note SARS-CoV-2

Le projet porte initialement sur la programmation d'un système embarqué de mesure ECG puis de traitement de l'information issue de ces mesures. La situation actuelle a nécessité la réalisation préalable de l'acquisition des mesures que vous trouverez sous forme de fichiers et la partie embarquée a été retirée.

Règles de programmation

Pour votre second développement logiciel encadré les contraintes sont relâchées (exemple : choix des noms des fonctions, nombre de fichier source...) pour tout ce qui n'est pas précisé par les spécifications se trouvant dans ce document de projet. Les attentes spécifiques de réalisation sont précisées pour chaque partie et l'évaluation se fera uniquement sur ces critères. De manière globale les critères suivants s'appliqueront :

- Le plagiat est interdit
- Les codes sources doivent être commentés selon les mêmes critères que lors de votre premier projet de développement informatique

Recette

La recette est effectuée sur la base du fichier contenant l'ensemble des codes que vous devrez rendre à l'issue du projet dans trois semaines.

La recette est effectuée selon les critères que vous trouverez précisés dans chaque partie du projet.

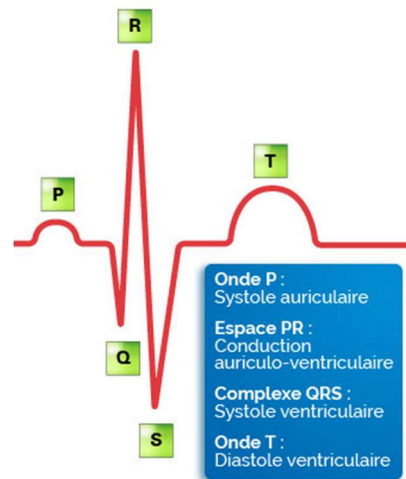
Description de l'environnement du projet

Les mesures

Les mesures sont issues d'ECG qui est un examen médical courant permettant d'enregistrer l'activité cardiaque. Il est basé sur la mesure des courants électriques traversant l'organe à chacune de ses contractions. Pour plus de détails on vous renvoie sur le contenu du cours sur ce sujet.

La lecture et l'interprétation d'un ECG s'acquièrent indubitablement avec l'expérience mais il existe maintenant des logiciels pouvant aider à l'analyse. Ce projet a pour objectif de réaliser certaines de ces tâches automatisées. Pour rappel, on pointe qu'un ECG normal ne signifie en aucun cas l'absence d'une pathologie cardiaque, tout comme un ECG anormal peut être sans gravité.

Un battement cardiaque normal laisse une empreinte typique sur l'ECG décomposée en plusieurs signaux utiles pour un diagnostic médical. On mesure ainsi la durée et l'amplitude de chaque onde constituant un battement cardiaque et celle du temps écoulé entre chaque phase de la contraction. Le temps écoulé entre deux points R permet, par exemple, d'estimer la fréquence cardiaque et diagnostiquer un éventuel trouble du rythme. Autre exemple, une torsade de pointe (anomalie du rythme des ventricules potentiellement mortelle) se traduit sur l'ECG par un allongement de la durée séparant l'onde Q et l'onde T.



Les mesures utilisées dans ce projet ont été rassemblées dans le répertoire *data/*. Les fichiers et leurs spécifications sont décrits dans le fichier *data/description.txt*.

Les fichiers contiennent des données ECG au format texte sur deux colonnes. La première est le temps en secondes (s), la seconde l'amplitude du signal en volt (V).

Les outils de développement

Trois langages et environnement différents vont être utilisés durant ce projet :

- Le langage C pour manipuler les données brutes
- Template de page web (HTML/CSS) + Python ou C pour remplir le template
- Matlab/Octave pour visualiser des figures et réaliser des analyses statistiques sur les informations extraites

Certains de ces outils ont déjà été installés sur vos ordinateurs lors de cours précédents. Sinon, suivez les liens placés en annexe (Divers) pour aider à leur installation.

Découpage du projet

Le projet va se découper maintenant en trois étapes successives qui feront l'objet de 3 évolutions de ce document :

1. TRAITEMENT du SIGNAL : La lecture des fichiers de mesure, la détection de ses caractéristiques et la construction d'informations pour le diagnostic.
2. INTERFACE UTILISATEUR : L'utilisation des informations de diagnostic pour construire une fiche récapitulative sur le patient sous forme d'une page web statique générée à partir d'un template
3. ANALYSE : L'utilisation de l'ensemble des informations de toutes les mesures pour extraire des informations discriminantes entre individus.

Le paquet de démarrage

Les données nécessaires au démarrage du projet se trouvent dans le fichier compressé fourni. Une fois décompacté vous devez obtenir un répertoire racine (ECG-SW-Project-BIOST3/). Cette racine contient un répertoire de données (data/). La racine contient aussi un squelette pour le premier développement en C du projet. Les autres informations vous seront fournies en cours de la période du projet avec les évolutions du document.

1- Traitement du signal de mesure (2/4)

L'objectif de cette étape est de lire un fichier de mesure pour en extraire l'ensemble des informations suivantes avant de les sauvegarder ensuite dans un fichier de résultat dont la forme est décrite plus loin :

- Les positions et amplitudes des ondes P,Q,R,S,T,U
- La fréquence et la variation de rythme
- Les délais moyens QRS, QT, PQ

Le programme, son exécution et son évaluation

Le programme est à réaliser en langage C sur la base du squelette (makefile, common.h/c, process-data.c) que vous trouverez dans la racine de ce qui vous a été fourni. Vos choix de programmation sont libres dans la limite de ce qui est dans les règles de programmation globale et dans le cahier des charges qui suit.

A noter que common.h/c propose une solution de mémorisation dans un tableau de structure pour le stockage des données que vous allez lire dans le fichier ECG. Toutefois, il s'agit juste d'une proposition et son utilisation n'est pas une obligation.

Le programme à exécuter doit se nommer process-data (process-data.exe dans certaines configurations de l'environnement de développement), être localisé dans la racine du projet et la commande d'exécution pour l'utilisateur doit prendre la forme suivante :

```
./process-data <le numéro du fichier de données>
```

Lors de l'exécution un affichage doit informer l'utilisateur de l'évolution du programme lors des phases de détection et de calcul.

A l'issue de cette exécution un fichier d'information est généré dans le même répertoire que les données. Son nom est de la forme : <nom fichier données>-infos.txt (exemple : pour data/ECG1.txt on obtient data/ECG1-infos.txt) . Le format du fichier de données enregistré est le suivant :

```
<Nom du fichier ECG traité>
<frequence moyenne pulsation cardiaque>
<temps moyen QRS>
<temps moyen QT>
<temps moyen PQ>
###BEGIN DATA WAVES
<temps onde P> <amplitude onde P> <tps Q> <ampl. Q> <tps R> <ampl. R> <tps S> <ampl. S> <tps T> <ampl. T> <tps U> <ampl. U>
...
###END DATA WAVES
###BEGIN DATA HEARTBEAT
<temps taille fenêtre de mesure>
<temps début de fenêtre 1>
<fréquence moyenne fenêtre 1>
```

```
<fréquence moyenne fenêtre 2>
```

```
...
```

```
###END DATA HEARTBEAT
```

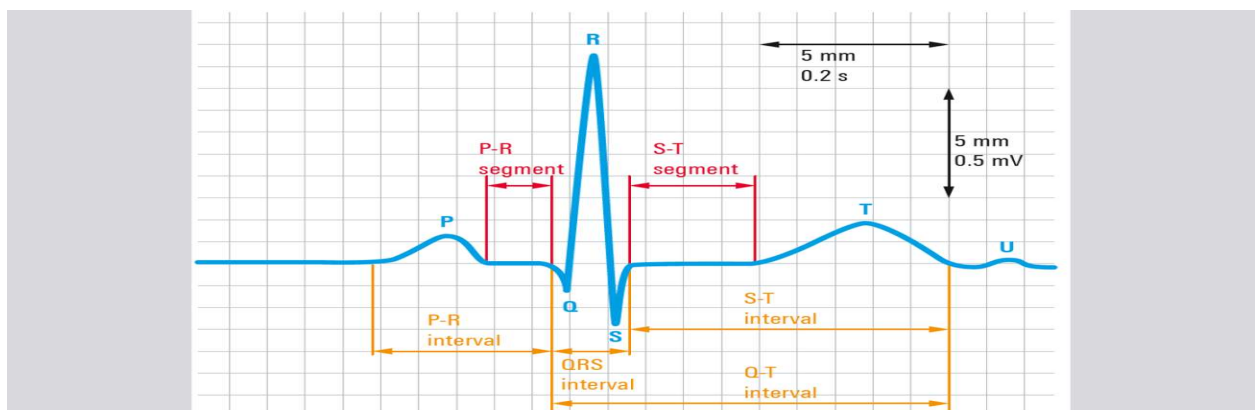
Les temps seront donnés en millisecondes (ms), les amplitudes en millivolts (mV), les fréquences en Hertz (Hz).

La recette de votre travail sera effectuée sur vos codes sources (les codes *.c/h de la racine) avec :

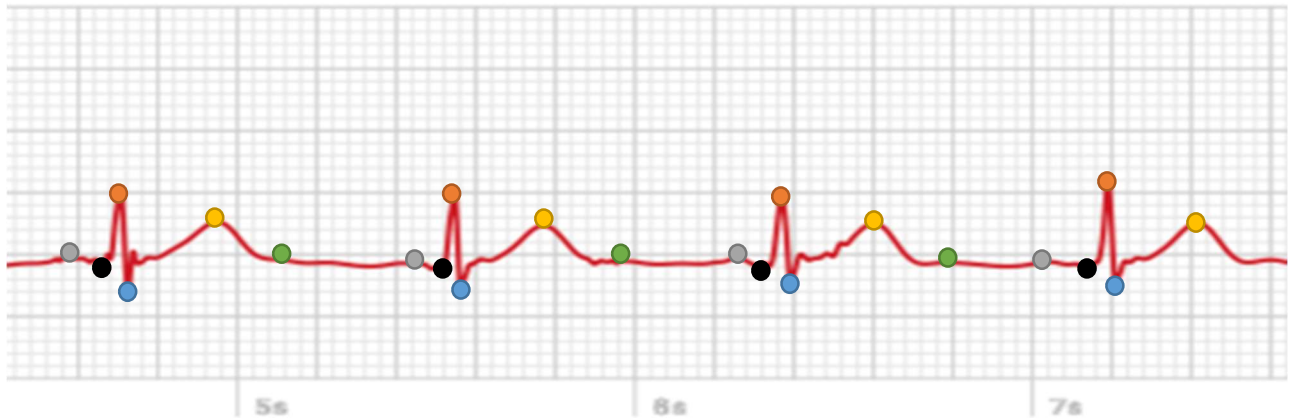
- Test de résultat (2/3 - oui/non) : Compilation, lecture d'un fichier de données, détection des ondes et sauvegarde dans fichier
- Qualité des commentaires (1/3 - oui/non) : pourcentage de commentaires dans le code.

Détection des formes d'onde

La détection a pour objectif de retrouver les points caractéristiques d'une onde ECG (cf figures ci-dessous). Vous noterez que l'ensemble d'une impulsion de signal ECG se trouve localisée généralement sur une fenêtre de temps de l'ordre d'une seconde et demi. Les algorithmes de recherche d'information que vous allez mettre en place pour détecter les paramètres vont donc utiliser les caractéristiques de l'ECG (par exemple : le pic P se trouve avant le R à un délai inférieur à 0.2s. Il est aussi positif.). Grâce à ces informations a priori et des recherches de maximum ou minimum vous devez essayer d'obtenir les informations de votre signal et pour chaque pulsation.



La première étape est donc une recherche de la dépolarisation ventriculaire R. Elle sera à réaliser en recherchant les pics positifs principaux de votre signal. Ils serviront ensuite de référence pour identifier les autres composantes.



Les résultats sont à sauvegarder dans le fichier d'information préalablement décrit et entre les bornes

```
###BEGIN DATA WAVES
```

```
..
```

```
###END DATA WAVES
```

Chaque point caractéristique est enregistré avec sa position temporelle et sa valeur en amplitude. L'ordre des points est celui de leur apparition dans l'impulsion (PQRSTU).

Calcul des moyennes

A l'issue de votre détection vous devrez calculer les moyennes suivantes en utilisant l'ensemble des informations détectées précédemment :

- La fréquence de pulsation
- La durée du complexe QRS (délai entre Q et S)
- Le délai entre Q et T
- Le délai entre P et Q

Les résultats sont à sauvegarder en début du fichier d'information.

Calcul de l'évolution du rythme cardiaque

Cette évolution se calcule sur des fenêtres de temps de taille à fixer (on définira dans le programme une variable globale modifiable qui identifiera la durée de la fenêtre. Dans un premier temps on prendra 5s). Dans ces fenêtres la fréquence de pulsation moyenne est calculée. La variation de rythme correspond à l'évolution de cette moyenne dans le temps. Les fenêtres de moyennage sont consécutives et ne se chevauchent pas.

Les résultats sont à sauvegarder dans le fichier d'information préalablement décrit et entre les bornes

```
###BEGIN DATA HEARTBEAT
```

```
...
```

```
###END DATA HEARTBEAT
```

La première valeur correspond à la taille des fenêtres, la seconde à la position du début de la première fenêtre de mesure, la troisième et suivante sont les moyennage de pulsation dans chaque fenêtre jusqu'à la fin du signal échantillon.

2- Interface utilisateur (1/4)

A venir document 2/3

3- Analyse (1/4)

A venir document 3/3

Divers

Installation environnement de développement C sous Windows 10

Depuis Windows 10, il est possible de faire tourner une distribution Ubuntu nativement mais sans interface graphique (il y a des "hack" pour avoir tout de même une interface graphique)

Pour son installation suivez le tuto de Korben : <https://korben.info/installer-shell-bash-linux-windows-10.html>

Une fois l'installation réalisée vous pouvez lancer Bash depuis le menu Windows. Un terminal s'ouvre, il s'agit d'un environnement Linux dans lequel vous pouvez rentrer la commande usuelle et notamment la commande gcc.

A noter que vous pouvez accéder à votre arborescence Windows (par exemple lecteur D:), il suffit de rentrer la commande unix : `cd /mnt/d` (tous vos disques sont montés en /mnt)

Pour l'édition du code, vous pouvez utiliser les logiciels Windows suivants :

- Notepad++ (<https://notepad-plus-plus.org/fr/>)
- Sublimetext (<https://www.sublimetext.com/>)
- VisualStudio(<https://visualstudio.microsoft.com/fr/vs/getting-started/web-install/>)
- Code:Blocks(<http://www.codeblocks.org/>)
- ...

Note pour développement C sous Mac et Linux

Ces environnements possèdent par défaut une base du type Linux et vous devez avoir les outils de développement nécessaires déjà installés.

Rappel de commandes Unix

Changer de répertoire : `cd <chemin répertoire>`

Créer un répertoire : `mkdir <chemin répertoire>`

Effacer un fichier ou un répertoire : `rm -rf <chemin>`

Déplacer un fichier ou un répertoire : `mv <chemin source> <chemin destination>`

Copier un fichier ou un répertoire : `cp -R <chemin source> <chemin destination>`

Référence pour le langage C

Vous trouverez des informations sur les commandes standards en langage C à l'adresse suivante et en utilisant leur formulaire de recherche :

<http://www.cplusplus.com/reference/>

The screenshot shows the cplusplus.com website. The browser address bar displays www.cplusplus.com/reference/. The page has a dark blue header with a search bar and navigation links like 'register' and 'log in'. A left sidebar contains a 'C++' menu with links to 'Information', 'Tutorials', 'Reference', 'Articles', and 'Forum'. The main content area is titled 'Reference' and 'Standard C++ Library reference'. It includes a 'C Library' section with a list of C standard headers and their descriptions.

C library:	C Library:
<code><cassert></code> (<code>assert.h</code>)	C Diagnostics Library (header)
<code><cctype></code> (<code>ctype.h</code>)	Character handling functions (header)
<code><cerrno></code> (<code>errno.h</code>)	C Errors (header)
<code><cfenv></code> (<code>fenv.h</code>)	Floating-point environment (header)
<code><float></code> (<code>float.h</code>)	Characteristics of floating-point types (header)
<code><stdint.h></code> (<code>inttypes.h</code>)	C integer types (header)
<code><iso646></code> (<code>iso646.h</code>)	ISO 646 Alternative operator spellings (header)
<code><limits></code> (<code>limits.h</code>)	Sizes of integral types (header)
<code><locale></code> (<code>locale.h</code>)	C localization library (header)
<code><math></code> (<code>math.h</code>)	C numerics library (header)
<code><setjmp></code> (<code>setjmp.h</code>)	Non-local jumps (header)

Références pour Python

Vous trouverez les informations d'installation et la documentation sous <https://www.python.org/>

Références pour Octave/Matlab

Vous trouverez la documentation sur le langage dans les onglets documentation

- soit du site d'Octave [<https://www.gnu.org/software/octave/>] (> Docs)
- soit de Matlab [<https://fr.mathworks.com/products/matlab.html>] (> Support > Documentation)