

- E1. Corentin Maillot
- E2. Jean-Baptiste Leduc
- E3. Gaétan Duquesne
- E4. Thomas Franconville



# Rapport de projet ECG/EEG

2<sup>ème</sup> année

**BTS SNEC**



Partenaire du projet : GRAMFC, CHU Amiens  
Année 2016-2017

<b>Lycée des métiers du design et des technologies</b>  70 Bd de saint Quentin  80094 Amiens Cedex 3  Téléphone : 03 22 53 41 03	<b>BTS Systèmes Numériques</b> <b>Option B Électronique et Communication</b>	<b>Session 2017</b>
--	---	---------------------

<b>ECG_EEG</b>
----------------

<i>Partenaire professionnel :</i>  <b>GRAMFC</b>  Adresse : <b>Université de Picardie Jules Verne</b> <b>Chemin du Thil,</b> <b>80000 Amiens.</b>	<i>Étudiants chargés du projet :</i>  <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left;">Noms</th> <th style="width: 50%; text-align: left;">Prénoms</th> </tr> </thead> <tbody> <tr> <td>- MAILLOT CORENTIN</td> <td></td> </tr> <tr> <td>- LEDUC JEAN-BAPTISTE</td> <td></td> </tr> <tr> <td>- DUQUESNE GAÉTAN</td> <td></td> </tr> <tr> <td>- FRANCONVILLE THOMAS</td> <td></td> </tr> </tbody> </table>	Noms	Prénoms	- MAILLOT CORENTIN		- LEDUC JEAN-BAPTISTE		- DUQUESNE GAÉTAN		- FRANCONVILLE THOMAS		<i>Professeurs ou Tuteurs responsables :</i>  <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: left;">Noms</th> <th style="width: 50%; text-align: left;">Prénoms</th> </tr> </thead> <tbody> <tr> <td>-HAESAERT</td> <td>Patrice</td> </tr> <tr> <td>-TOMCZAK</td> <td>Robert</td> </tr> <tr> <td>-</td> <td></td> </tr> <tr> <td>-</td> <td></td> </tr> </tbody> </table>	Noms	Prénoms	-HAESAERT	Patrice	-TOMCZAK	Robert	-		-	
Noms	Prénoms																					
- MAILLOT CORENTIN																						
- LEDUC JEAN-BAPTISTE																						
- DUQUESNE GAÉTAN																						
- FRANCONVILLE THOMAS																						
Noms	Prénoms																					
-HAESAERT	Patrice																					
-TOMCZAK	Robert																					
-																						
-																						

Reprise d'un projet : Oui / Non

## **Présentation générale du système supportant le projet :**

Le GRAMFC (Groupe de Recherches sur l'Analyse Multimodale de la Fonction Cérébrale) est une équipe pluridisciplinaire regroupant neurophysiologistes, pédiatres réanimateurs, et spécialistes du traitement de signal du CHU d'Amiens et de l'Université de Picardie.

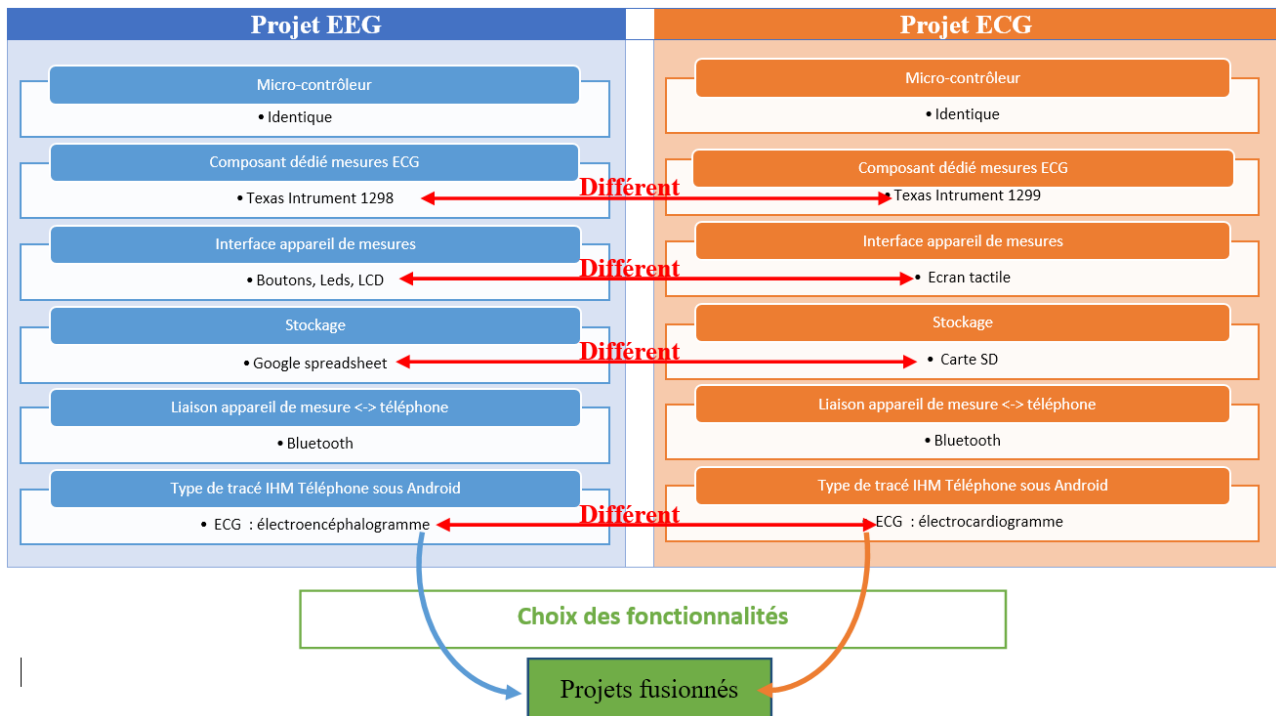
Leurs recherches les amènent à effectuer, très souvent, des examens d'EEG et d'ECG :

- EEG : électroencéphalogramme, est un examen qui repose sur la mesure de l'activité électrique du cerveau. Celle-ci est effectuée par l'intermédiaire d'électrodes placées au contact du cuir chevelu.
- ECG : électrocardiogramme, correspond à l'enregistrement de l'activité électrique du coeur, nécessaire à ses contractions. Pour cela, on dispose des électrodes au niveau des chevilles, des poignets et sur la poitrine des patients

Ces deux examens sont généralement pratiqués soit dans un service de neurologie/cardiologie des hôpitaux soit dans un cabinet de neurologie/ cardiologie en ville, mais très rarement à domicile. En effet, les appareils de mesures sont assez volumineux et non transportables.

Monsieur Ardalan AARABI, Maître de Conférence des Universités en Neurosciences du CURS (Centre Universitaires de Recherches en Santé) aimerait disposer d'un système de mesures facilement transportable chez un patient par exemple, mais aussi dans d'autres services. Il aimerait également disposer d'une visualisation sur un smartphone ou une tablette « Android ».

Le projet initial est séparé en deux : une partie EEG et une partie ECG avec des fonctionnalités différentes. Au final, le donneur d'ordre choisira la solution qui lui convient le mieux et fusionnera les deux parties pour réaliser un seul objet avec les deux fonctionnalités. Beaucoup de tâches étant communes aux deux parties, les étudiants partageront le travail effectué.



## Cette présentation concerne la partie ECG.

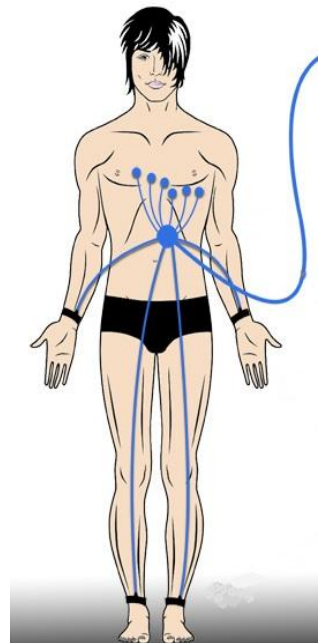
### Qu'est-ce que l'ECG ?

C'est un électrocardiogramme, correspond à l'enregistrement de l'activité électrique du cœur : à chaque battement cardiaque, une impulsion électrique (ou « onde ») traverse le cœur. Cette onde fait contracter le muscle cardiaque afin qu'il expulse le sang du cœur. Un ECG mesure et enregistre l'activité électrique qui traverse le cœur.

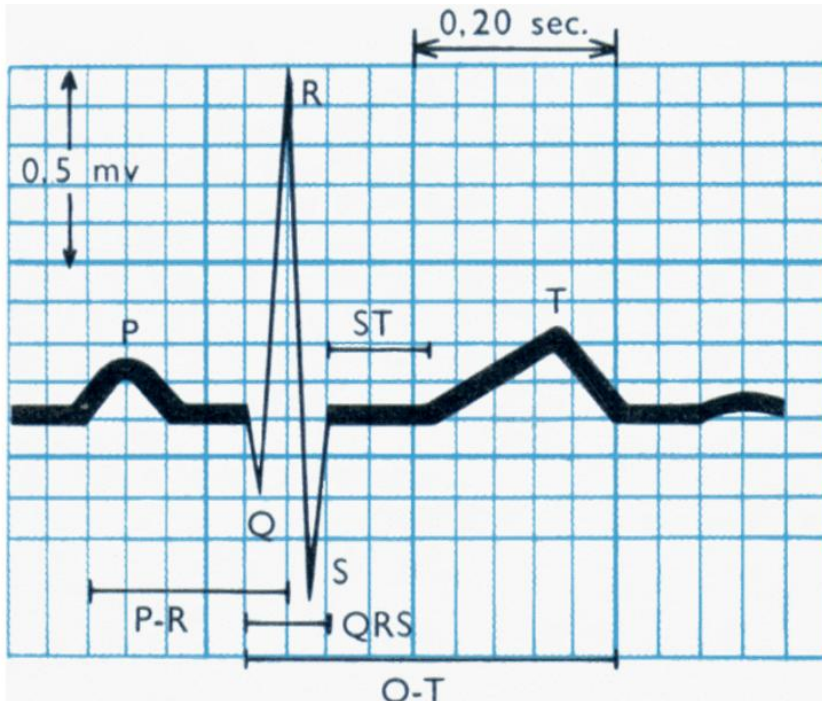
Pour cela, on dispose des électrodes au niveau des chevilles, des poignets et sur la poitrine des patients.

### A quoi sert un ECG ?

- détecter les arythmies (battements irréguliers), qui peuvent avoir favorisé la formation de caillots,
- surveiller des régions endommagées du muscle cardiaque (causées par une crise cardiaque antérieure),
- détecter des troubles non cardiaques comme les maladies pulmonaires,
- surveiller la récupération à la suite d'une crise cardiaque, la progression d'une maladie cardiaque ou l'efficacité de certains médicaments ou d'un stimulateur cardiaque.



## Tracé ECG :



L'onde P représente la contraction des oreillettes. Elle est de forme arrondie, souvent positive et de faible amplitude.

Le complexe QRS correspond à la contraction des ventricules.

L'onde T est la période de repos du cœur.

Le segment ST correspond à la période d'excitation des ventricules jusqu'à la phase de repos.

Un décalage de plus de 1 mm par rapport à la ligne isoélectrique est anormal.

Avec ce système, les examens ECG peuvent se faire à domicile, mais également dans d'autres services que ceux de neurologie. Ils seront pratiqués par une infirmière ou le cardiologue lui-même. Le patient porte des électrodes et l'examen commence : le boîtier proche du patient et de l'infirmière/neurologue récupère les signaux électriques et les envoie vers le smartphone où un tracé est disponible pour une interprétation.

Une source d'énergie est nécessaire pour recharger les batteries du système.

L'utilisateur du système effectue les mesures sur le patient. Au cours de cette étape, les données sont envoyées vers le Smartphone et sont stockées sur la carte SD.

Il est prévu de pouvoir modifier les paramètres de mesures.

Le médecin visualise sur son Smartphone, les tracés ECG qu'il peut interpréter.

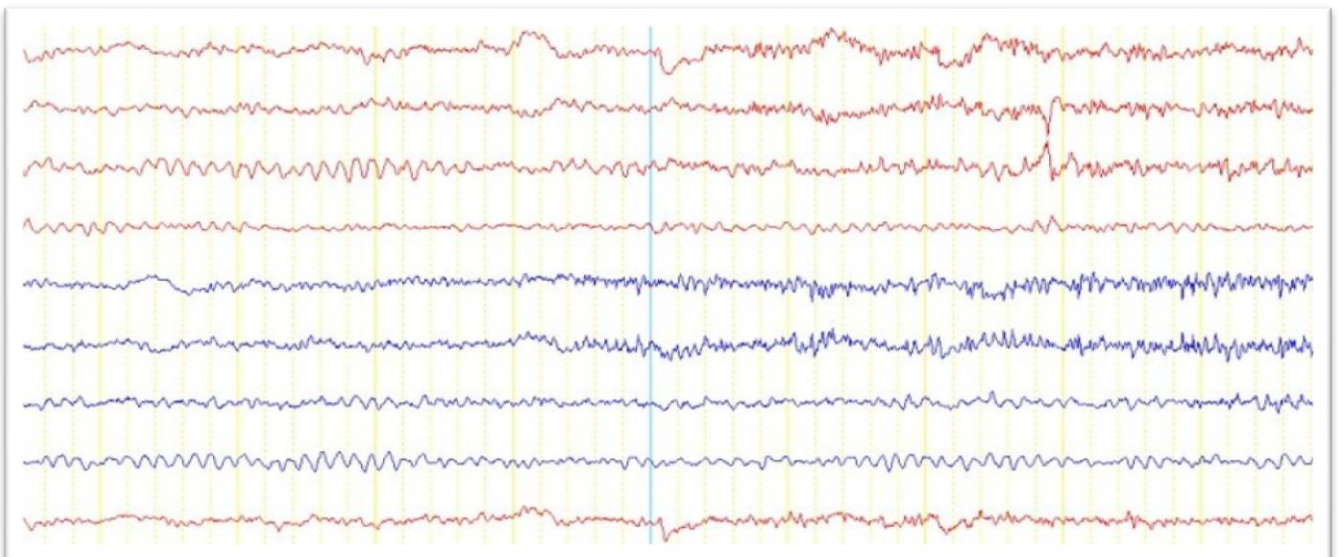
### Les exigences du système :

- On doit pouvoir visualiser les signaux de 8 électrodes
- Les abscisses sont le temps en seconde, et les ordonnées sont les amplitudes
- Huit traces devront apparaître à l'écran
- L'interface pour Android sera réalisée à l'aide d'AppInventor
- Liaison appareil/Smartphone sera en bluetooth
- On doit pouvoir modifier les paramètres pour augmenter la durée d'enregistrement et la fréquence d'échantillonnage. C'est pourquoi il faudra prévoir une interface permettant de modifier ces paramètres, ce paramétrage se fera par l'intermédiaire d'un écran tactile.
- Les informations à mémoriser : outre les informations provenant des électrodes, les numéros d'identification unique, date et heure, nom, prénom et âge du patient.
- Le stockage des données sur une carte SD permettra 2h d'enregistrement.
- On utilisera le composant TI 1298 spécialisé dans ce type de mesures.
- La fréquence d'échantillonnage sera 1000 Hz
- Le nombre d'électrodes est de 8
- Résolution est de 10 bits

### **Cette présentation concerne le projet EEG.**

#### ***Qu'est-ce que l'EEG ?***

L'électroencéphalographie (EEG) est l'enregistrement graphique, au moyen d'électrodes placées sur la surface du cuir chevelu représentée sous la forme d'un tracé appelé électroencéphalogramme qui mesure des différences de potentiel électrique (l'activité électrique) produites au niveau du cerveau. Elle se fait par l'intermédiaire d'un casque d'électrodes.



Tracé des enregistrements de 9 zones du cerveau

## A quoi sert un EEG ?

1. préciser l'état d'éveil d'une personne avec qui la communication est difficile.
2. rechercher des anomalies électriques, notamment en faveur de crises d'épilepsie chez un patient.
3. chez un épileptique connu, l'EEG peut être prescrit pour apprécier l'efficacité du traitement.
4. étudier le fonctionnement du cerveau en cas de troubles cognitifs, difficultés de mémoire, troubles du langage.
5. évaluer systématiquement un nouveau-né prématuré par l'analyse de son EEG afin d'étudier la maturation cérébrale

Avec ce système, les examens EEG peuvent se faire à domicile, mais également dans d'autres services que ceux de neurologie. Ils seront pratiqués par une infirmière ou le neurologue lui-même. Le patient porte un casque à électrodes et l'examen commence : le boîtier proche du patient et de l'infirmière/neurologue récupère les signaux électriques et les envoie vers le smartphone où un tracé est disponible pour une interprétation.

Une source d'énergie est nécessaire pour recharger les batteries du système.

### Les exigences du système :

- On doit pouvoir visualiser les signaux de 8 électrodes
- Les abscisses sont le temps en seconde, et les ordonnées sont les amplitudes
- Huit traces devront apparaître à l'écran
- L'interface pour Android sera réalisée à l'aide d'AppInventor
- Liaison appareil/smartphone sera en bluetooth
- On doit pouvoir modifier les paramètres pour augmenter la durée d'enregistrement et la fréquence d'échantillonnage. C'est pourquoi il faudra prévoir une interface permettant de modifier ces paramètres, sous forme de menu
- L'interface sera à base de boutons poussoirs et d'écran LCD
- Les informations à mémoriser : outre les informations provenant des électrodes, les numéros d'identification unique, date et heure, nom, prénom et âge du patient.
- Le stockage des données sur une carte SD permettra 2h d'enregistrement.
- On utilisera le composant TI 1299 spécialisée dans ce type de mesures
- La fréquence d'échantillonnage sera 1000 Hz
- Le nombre d'électrodes est de 8
- Résolution est de 10 bits

---

### Analyse de l'existant :



Les cartes électroniques avec les circuits spécialisés (ADS1298, ADS1299) ont été réalisées. Ces circuits intégrés sont des convertisseurs analogique->numérique développés spécialement pour les ECG et EEG.

Il faut interfacier ces cartes avec une carte de prototypage rapide à microcontrôleur et développer les routines logicielles permettant de récupérer les résultats de conversion. Ces résultats seront archivés sur une carte SD et envoyés en bluetooth vers un Smartphone ou une tablette.

Il faudra en plus réaliser une IHM pour chacun de ces prototypes.

---

## Détail du travail de chaque étudiant :

### **Etudiant E1 :**

Après avoir étudié le cahier des charges global, l'étudiant E1 devra :

- En collaboration avec les autres étudiants, préparer une présentation Sysml du projet.
- Elaborer un planning des activités pour la durée du projet.
- Etudier les connexions à utiliser pour implanter un module bluetooth sur le prototype final.
- Présenter ce travail aux autres étudiants du groupe qui devront le réutiliser.
- Prendre connaissance des caractéristiques matérielles et logicielles du circuit ADS1299 (ECG) et concevoir l'interface entre la carte conversion et une carte de prototypage rapide (en collaboration avec l'étudiant E3 qui réalise la même tâche pour l'EEG).
- Développer la routine logicielle permettant l'acquisition des résultats de mesure des 8 sondes.
- Réaliser un dossier de fabrication pour la carte d'interface en incluant le travail de l'étudiant E2 pour la partie IHM.
- Réaliser le routage de cette carte
- Faire la mise au point et la recette du prototype final avec l'étudiant E1.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E1.
- Rédiger le rapport final (une partie commune et une partie personnelle).



### **Etudiant E2:**

Après avoir étudié le cahier des charges global, l'étudiant E2 devra :

- En collaboration avec les autres étudiants, préparer une présentation Sysml du projet.
- Elaborer un planning des activités pour la durée du projet.
- Concevoir l'interface entre la carte de prototypage rapide, l'écran tactile et la carte SD.
- Présenter ce travail à l'étudiant E1 qui va réaliser le routage de la carte.
- Réaliser le câblage de cette carte.
- Développer la routine logicielle de l'IHM sur écran tactile.
- Développer la routine logicielle permettant la sauvegarde sur la carte SD.
- Présenter ces routines aux autres étudiants du groupe qui devront les réutiliser.
- Faire la mise au point et la recette du prototype final avec l'étudiant E1.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E1.
- Rédiger le rapport final (une partie commune et une partie personnelle).

### **Etudiant E3:**

Après avoir étudié le cahier des charges global, l'étudiant E3 devra :

- En collaboration avec les autres étudiants, préparer une présentation Sysml du projet.
- Elaborer un planning des activités pour la durée du projet.
- Prendre connaissance des caractéristiques matérielles et logicielles du circuit ADS1298 (EEG) et concevoir l'interface entre la carte conversion, le module bluetooth et une carte de prototypage rapide.
- Développer la routine logicielle permettant la transmission des informations par le module bluetooth, présenter et communiquer ce travail aux autres étudiants du groupe qui vont le réutiliser pour la partie ECG.
- Réaliser un dossier de fabrication pour la carte d'interface en incluant le travail de l'étudiant E4 pour la partie IHM.
- Réaliser le routage de cette carte.
- Faire la mise au point et la recette du prototype final avec l'étudiant E4.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E4.
- Rédiger le rapport final (une partie commune et une partie personnelle).

### **Etudiant E4:**

Après avoir étudié le cahier des charges global, l'étudiant E2 devra :

- En collaboration avec les autres étudiants, préparer une présentation Sysml du projet.
- Elaborer un planning des activités pour la durée du projet.
- Concevoir l'interface entre la carte de prototypage rapide, l'écran LCD, le mini clavier et la carte SD.
- Présenter ce travail à l'étudiant E3 qui va réaliser le routage de la carte.
- Réaliser le câblage de cette carte.
- Développer une application Appinventor permettant d'afficher les courbes EEG et ECG sur un smartphone.
- Présenter cette application aux autres étudiants du groupe qui devront la réutiliser.
- Faire la mise au point et la recette du prototype final avec l'étudiant E3.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E3.
- Rédiger le rapport final (une partie commune et une partie personnelle).

#### Description structurelle du système :

Principaux constituants :	Caractéristiques techniques :
1 carte « Acquisition des signaux ECG	Voir CDC
1 carte « Acquisition des signaux EEG	Voir CDC
2 modules Bluetooth	Voir CDC
2 lecteurs de carte SD	Voir CDC
1 écran tactile	Voir CDC
1 écran LCD	Voir CDC
1 mini clavier	Voir CDC

#### Inventaire des matériels et outils logiciels à mettre en œuvre par les candidats :

Désignation :	Caractéristiques techniques :
Cartes de prototypage rapide	Teensyduino
Logiciel de saisie de schéma et de routage	Proteus
Plateforme de développement logiciel	Arduino
Matériel d'implantation de CMS	
Four de fusion de CMS	

Tâches	Revues	Contrats de tâche	Compétences	Candidat_1	Candidat_2	Candidat_3	Candidat_4
		Expression fonctionnelle du besoin					
T1.4	R2	Vérifier la pérennité et mettre à jour les informations	C2.1	x	x	x	x
T2.1	R2	Collecter des informations nécessaires à l'élaboration du cahier des charges préliminaire	C2.2				
T2.3	R2	Formaliser le cahier des charges	C2.3 C2.4	x	x	x	x
T3.1	R2	S'approprier le cahier des charges	C3.1	x	x	x	x
T3.3	R2	Élaborer le cahier de recette	C3.5	x	x	x	x
T3.4	R2	Négocier et rechercher la validation du client	C2.4	x	x	x	x
		Conception					
T4.1	R3	Identifier le comportement d'un constituant	C3.1 C4.1	x	x	x	x
T4.2	R3	Traduire les éléments du cahier des charges sous la forme de modèles	C3.1	x	x	x	x
T4.3	R3	Rédiger le document de recette	C3.5	x	x	x	x
T5.1	R3	Identifier les solutions existantes de l'entreprise	C3.1 C3.6	x	x	x	x
T5.2	R3	Identifier des solutions issues de l'innovation technologique	C3.1 C3.6				
T5.4	R2	Sélectionner et/ou adapter une ou des solutions selon le contexte technicoéconomique	C3.8				
T6.1	R3	Prendre connaissance des fonctions associées au projet et définir les tâches	C2.4 C2.5	x	x	x	x
T6.2	R3	Définir et valider un planning (jalons de livrables)	C2.3 C2.4 C2.5	x	x	x	x

T6.3	R3	Assurer le suivi du planning et du budget	C2.1 C2.3 C2.4 C2.5	x	x	x	x
<b>Réalisation</b>							
T7.1	R3	Réaliser la conception détaillée du matériel et/ou du logiciel	C3.1 C3.3 C3.6 C3.8 C3.10	x	x	x	x
T7.2	RF	Produire un prototype logiciel et/ou matériel	C3.9 C4.1 C4.2 C4.3 C4.4 C4.6 C4.7	x	x	x	x
T7.3	RF	Valider le prototype	C3.5 C4.5	x	x	x	x
T8.1	RF	Définir une organisation ou un processus de maintenance préventive	C2.1				
T8.2	RF	Définir une organisation ou un processus de maintenance curative	C2.1	x	x	x	x
T9.2	RF	Installer un système ou un service	C2.5	x	x	x	x
T10.3	RF	Exécuter et/ou planifier les tâches professionnelles de MCO	C2.3				
T11.3	RF	Assurer la formation du client	C2.2 C2.5				
T12.1	RF	Organiser le travail de l'équipe	C2.3 C2.4 C2.5	x	x	x	x
T12.2	RF	Animer une équipe	C2.1 C2.3 C2.5	x	x	x	x
<b>Vérification des performances attendues</b>							
T9.1	RF	Finaliser le cahier de recette	C3.1 C3.5 C4.5	x	x	x	x
T10.4	RF	Proposer des solutions d'amélioration du système ou du service	C3.6	x	x	x	x

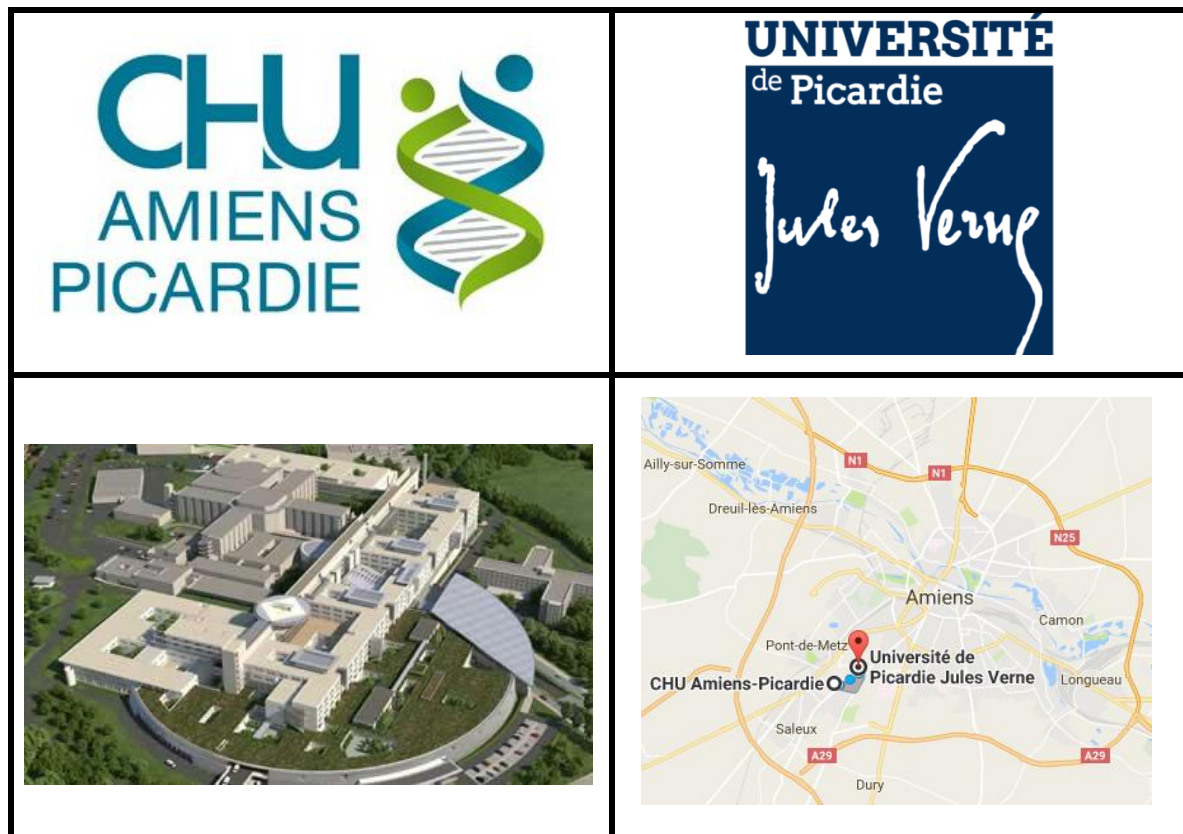
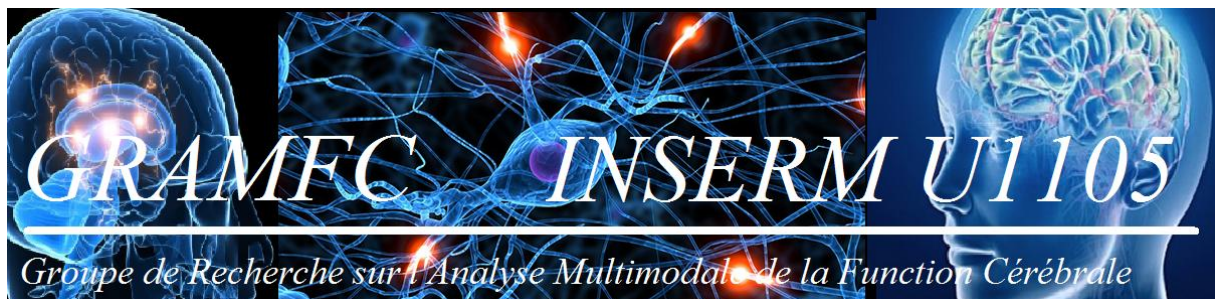
# SOMMAIRE

<b>I. PRESENTATION PROJET</b>	14
A. Présentation du partenaire	14
B. Présentation du projet, expression du besoin	16
C. Analyse du cahier des charges et de ses contraintes	21
D. Répartition des tâches	23
<b>II. PARTIE PERSONNELLE</b>	31
A. Etudiant E1	31
B. Etudiant E2	42
C. Etudiant E3	51
D. Etudiant E4	60
<b>III. CONCLUSION</b>	67

# I. PRESENTATION PROJET

## A. Présentation partenaire :

Le GRAMFC (Groupe de Recherches sur l'Analyse Multimodale de la Fonction Cérébrale) est une équipe pluridisciplinaire regroupant neurophysiologistes, pédiatres réanimateurs, et spécialistes du traitement de signal du CHU d'Amiens et de l'Université de Picardie.





Leurs recherches les amènent à effectuer, très souvent, des examens d'EEG et d'ECG :

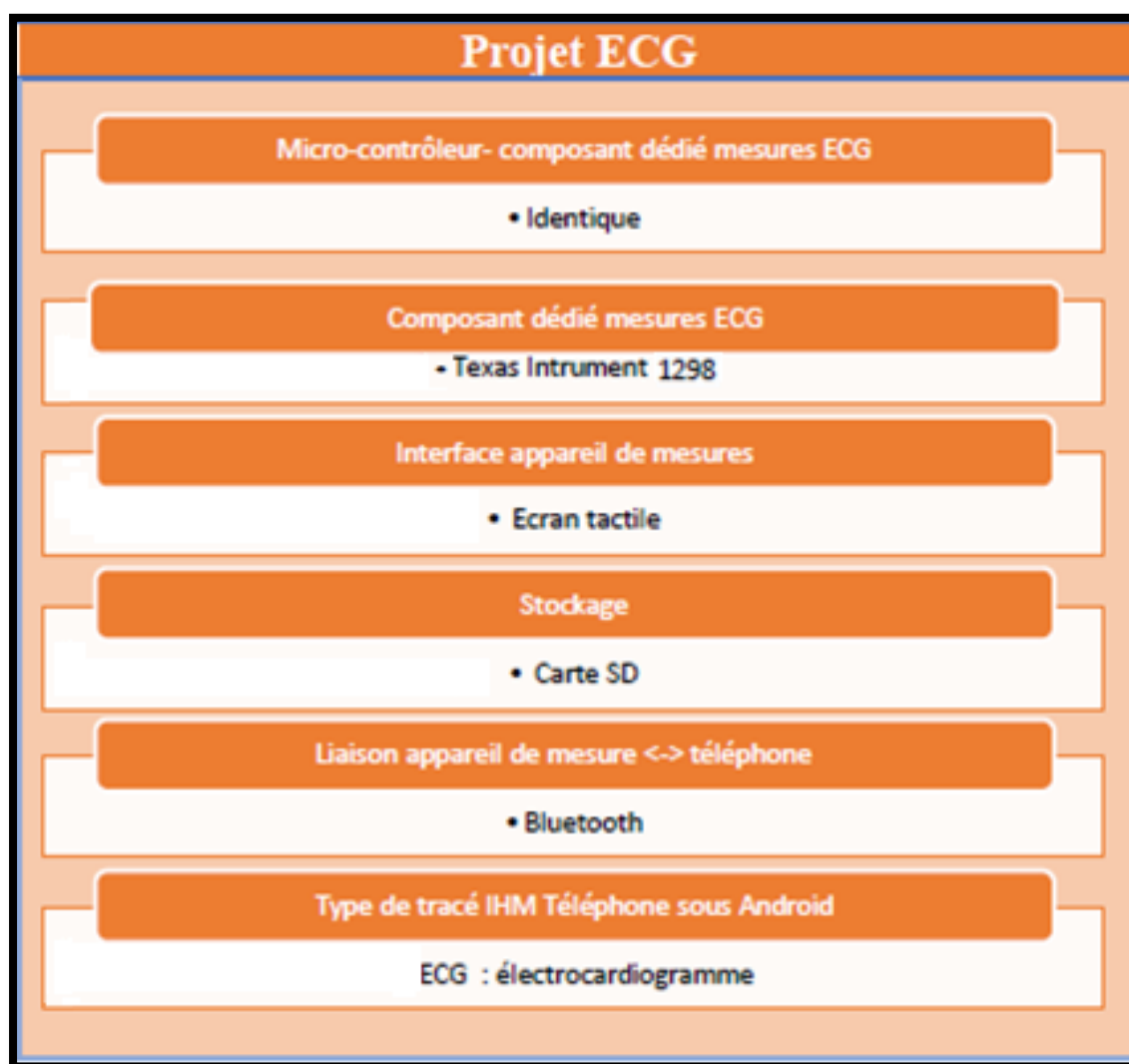
- EEG : électroencéphalogramme, est un examen qui repose sur la mesure de l'activité électrique du cerveau. Celle-ci est effectuée par l'intermédiaire d'électrodes placées au contact du cuir chevelu.
- ECG : électrocardiogramme, correspond à l'enregistrement de l'activité électrique du cœur, nécessaire à ses contractions. Pour cela, on dispose des électrodes au niveau des chevilles, des poignets et sur la poitrine des patients

Ces deux examens sont généralement pratiqués soit dans un service de neurologie/cardiologie des hôpitaux soit dans un cabinet de neurologie/ cardiologie en ville, mais très rarement à domicile. En effet, les appareils de mesures sont assez volumineux et non transportables.

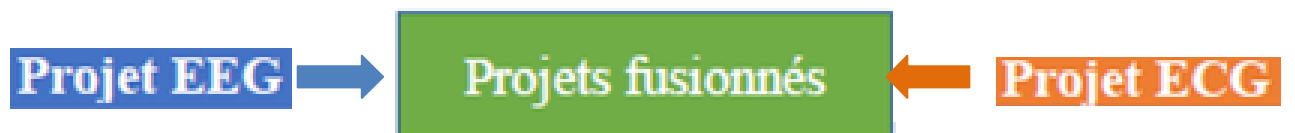
## B. Présentation du projet, expression du besoin :

Monsieur Ardalan AARABI, Maître de Conférence des Universités en Neurosciences du CURS (Centre Universitaires de Recherches en Santé) aimerait disposer d'un système de mesures facilement transportable chez un patient par exemple, mais aussi dans d'autres services. Il aimerait également disposer d'une visualisation sur un smartphone ou une tablette « Android ».

Le projet initial est séparé en deux : une partie EEG et une partie ECG avec des fonctionnalités différentes. Au final, le donneur d'ordre choisira la solution qui lui convient le mieux et fusionnera les deux parties pour réaliser un seul objet avec les deux fonctionnalités. Beaucoup de tâches étant communes aux deux parties, les étudiants partageront le travail effectué.



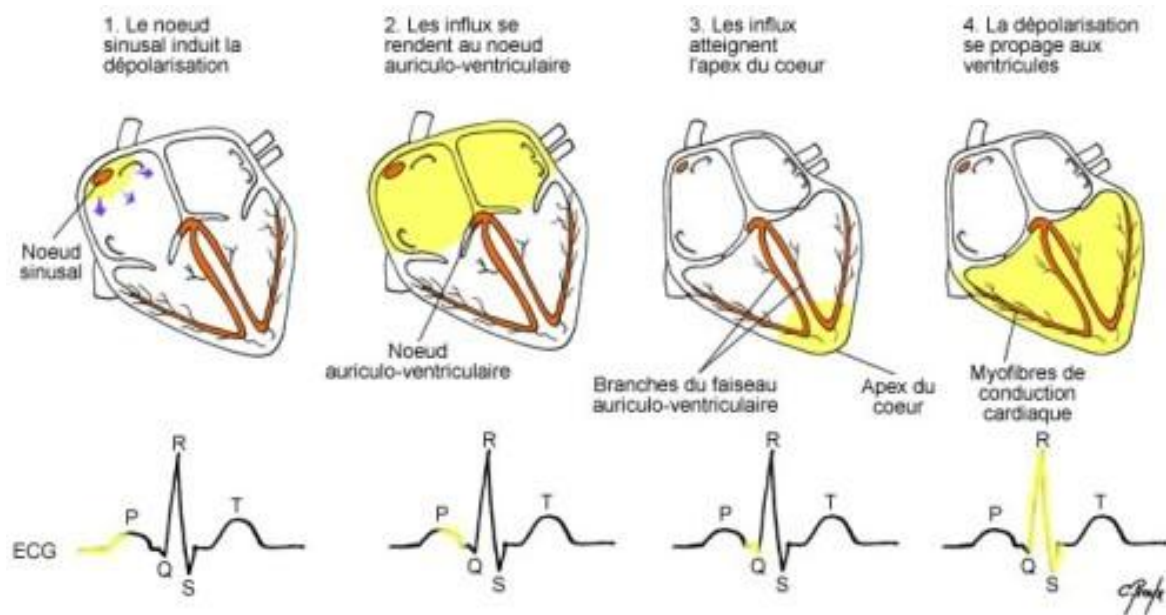




## Qu'est-ce que l'ECG ?

ECG est un acronyme pour électrocardiogramme, correspond à l'enregistrement de l'activité électrique du cœur : à chaque battement cardiaque, une impulsion électrique (ou « onde ») traverse le cœur. Cette onde fait contracter le muscle cardiaque afin qu'il expulse le sang du cœur. Un ECG mesure et enregistre l'activité électrique qui traverse le cœur.

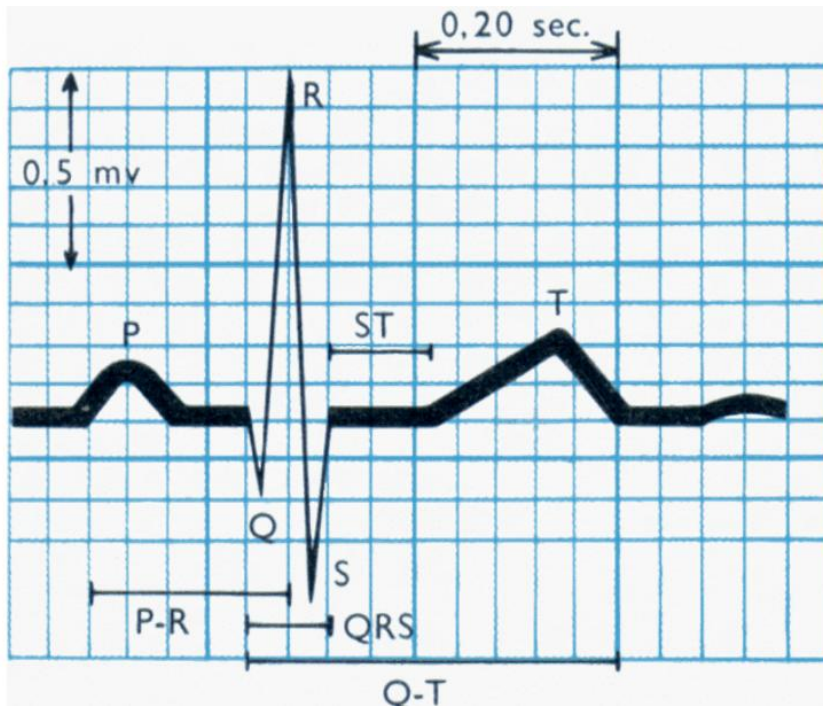
Pour cela, on dispose des électrodes au niveau des chevilles, des poignets et sur la poitrine des patients.



## A quoi sert un ECG ?

- détecter les arythmies (battements irréguliers), qui peuvent avoir favorisé la formation de caillots,
- surveiller des régions endommagées du muscle cardiaque (causées par une crise cardiaque antérieure),
- détecter des troubles non cardiaques comme les maladies pulmonaires,
- surveiller la récupération à la suite d'une crise cardiaque, la progression d'une maladie cardiaque ou l'efficacité de certains médicaments ou d'un stimulateur cardiaque.

Tracé ECG :



L'onde P représente la contraction des oreillettes. Elle est de forme arrondie, souvent positive et de faible amplitude.

Le complexe QRS correspond à la contraction des ventricules.

T est la période de repos du cœur.

ST correspond à la période d'excitation des ventricules jusqu'à la phase de repos.

Un décalage de plus de 1 mm par rapport à la ligne isoélectrique est anormal.

Avec ce système, les examens ECG peuvent se faire à domicile, mais également dans d'autres services que ceux de neurologie. Ils seront pratiqués par une infirmière ou le cardiologue lui-même. Le patient porte des électrodes et l'examen commence : le boîtier proche du patient et de l'infirmière/neurologue récupère les signaux électriques et les envoie vers le smartphone où un tracé est disponible pour une interprétation.

Une source d'énergie est nécessaire pour recharger les batteries du système.

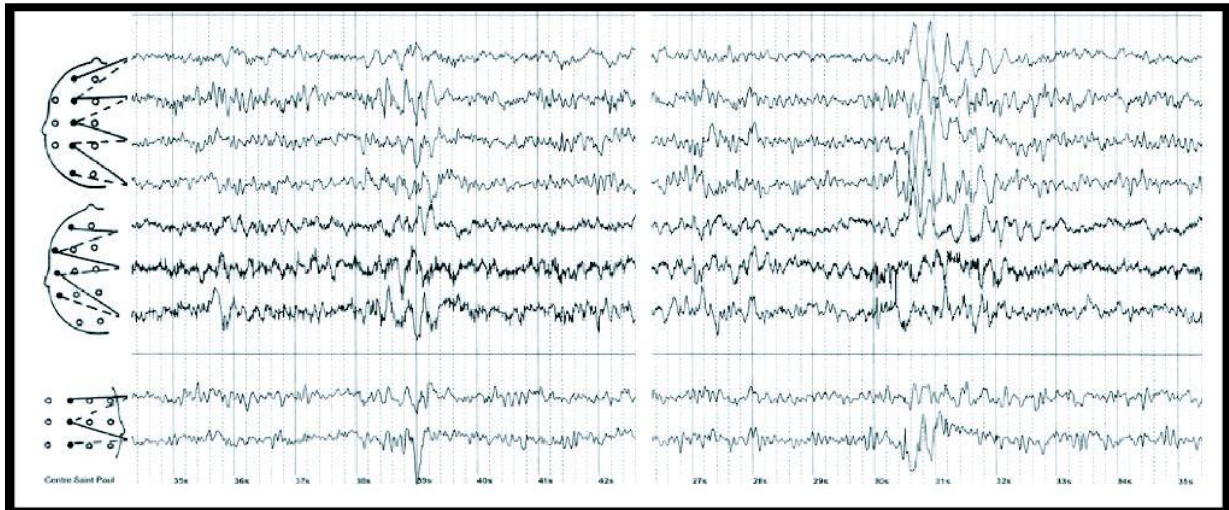
L'utilisateur du système effectue les mesures sur le patient. Au cours de cette étape, les données sont envoyées vers le Smartphone et sont stockées sur la carte SD.

Il est prévu de pouvoir modifier les paramètres de mesures.

Le médecin visualise sur son Smartphone, les tracés ECG qu'il peut interpréter.

## Qu'est-ce que l'EEG ?

L'électroencéphalographie (EEG) est l'enregistrement graphique, au moyen d'électrodes placées sur la surface du cuir chevelu représentée sous la forme d'un tracé appelé électroencéphalogramme qui mesure des différences de potentiel électrique (l'activité électrique) produites au niveau du cerveau. Elle se fait par l'intermédiaire d'un casque d'électrodes.



Tracé des enregistrements de 9 zones du cerveau

## A quoi sert un EEG ?

- préciser l'état d'éveil d'une personne avec qui la communication est difficile.
- rechercher des anomalies électriques, notamment en faveur de crises d'épilepsie chez un patient.
- chez un épileptique connu, l'EEG peut être prescrit pour apprécier l'efficacité du traitement.
- étudier le fonctionnement du cerveau en cas de troubles cognitifs, difficultés de mémoire, troubles du langage.
- évaluer systématiquement un nouveau-né prématuré par l'analyse de son EEG afin d'étudier la maturation cérébrale

Avec ce système, les examens EEG peuvent se faire à domicile, mais également dans d'autres services que ceux de neurologie. Ils seront pratiqués par une infirmière ou le neurologue lui-même. Le patient porte un casque à électrodes et l'examen commence : le boîtier proche du patient et de l'infirmière/neurologue récupère les signaux électriques et les envoie vers le smartphone où un tracé est disponible pour une interprétation.

Une source d'énergie est nécessaire pour recharger les batteries du système.

## C. Analyse du cahier des charges et de ses contraintes :

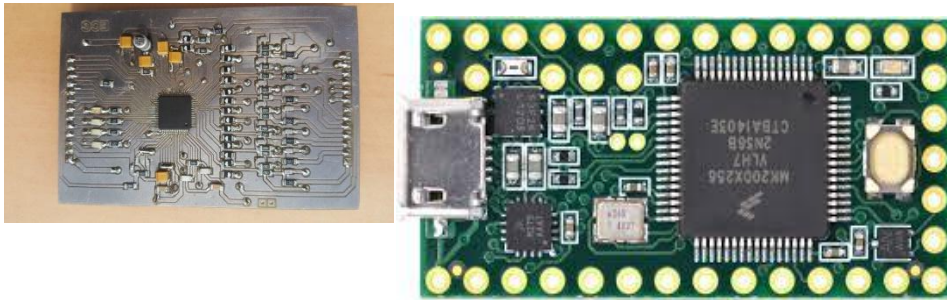
### Les exigences du système : ECG

- On doit pouvoir visualiser les signaux de 8 électrodes
- Les abscisses sont le temps en seconde, et les ordonnées sont les amplitudes
- Huit traces devront apparaître à l'écran
- L'interface pour Android sera réalisée à l'aide d'AppInventor
- Liaison appareil/Smartphone sera en bluetooth
- On doit pouvoir modifier les paramètres pour augmenter la durée d'enregistrement et la fréquence d'échantillonnage. C'est pourquoi il faudra prévoir une interface permettant de modifier ces paramètres, ce paramétrage se fera par l'intermédiaire d'un écran tactile.
- Les informations à mémoriser : outre les informations provenant des électrodes, les numéros d'identification unique, date et heure, nom, prénom et âge du patient.
- Le stockage des données sur une carte SD permettra 2h d'enregistrement.
- On utilisera le composant TI 1298 spécialisé dans ce type de mesures.
- La fréquence d'échantillonnage sera 1000 Hz
- Le nombre d'électrodes est de 8
- Résolution est de 10 bits

### Les exigences du système : EEG

- On doit pouvoir visualiser les signaux de 8 électrodes
- Les abscisses sont le temps en seconde, et les ordonnées sont les amplitudes
- Huit traces devront apparaître à l'écran
- Sucrer des glands
- L'interface pour Android sera réalisée à l'aide d'AppInventor
- Liaison appareil/smartphone sera en bluetooth
- On doit pouvoir modifier les paramètres pour augmenter la durée d'enregistrement et la fréquence d'échantillonnage. C'est pourquoi il faudra prévoir une interface permettant de modifier ces paramètres, sous forme de menu
- L'interface sera à base de boutons poussoirs et d'écran LCD
- Les informations à mémoriser : outre les informations provenant des électrodes, les numéros d'identification unique, date et heure, nom, prénom et âge du patient.
- Le stockage des données sur une carte SD permettra 2h d'enregistrement.
- On utilisera le composant TI 1299 spécialisée dans ce type de mesures
- La fréquence d'échantillonnage sera 1000 Hz
- Le nombre d'électrodes est de 8
- Résolution est de 10 bits

## Analyse de l'existant :



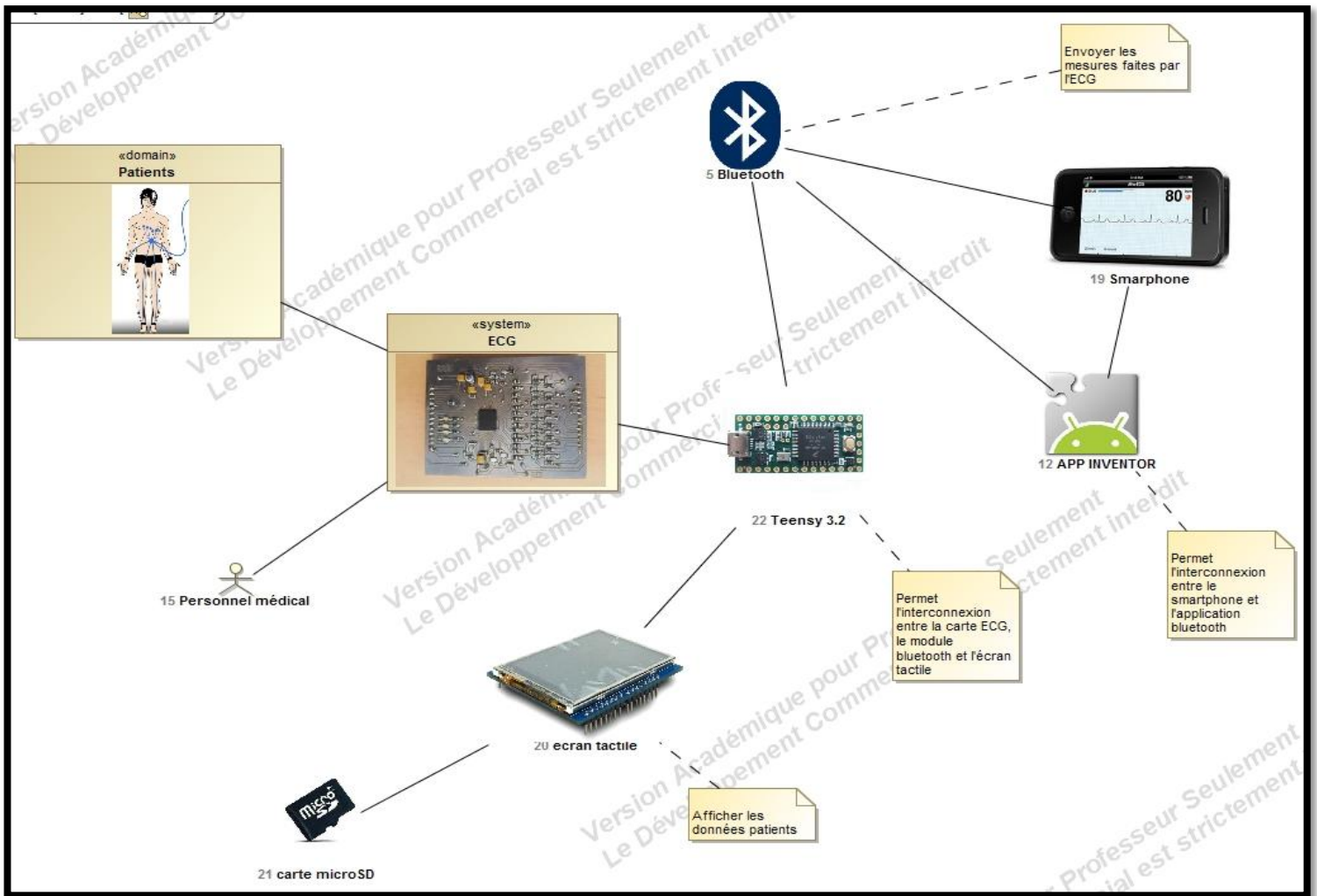
Les cartes électroniques avec les circuits spécialisés (ADS1298, ADS1299) ont été réalisées. Ces circuits intégrés sont des convertisseurs analogique->numérique développés spécialement pour les ECG et EEG.

Il faut interfacer ces cartes avec une carte de prototypage rapide à microcontrôleur et développer les routines logicielles permettant de récupérer les résultats de conversion. Ces résultats seront archivés sur une carte SD et envoyés en bluetooth vers un Smartphone ou une tablette.

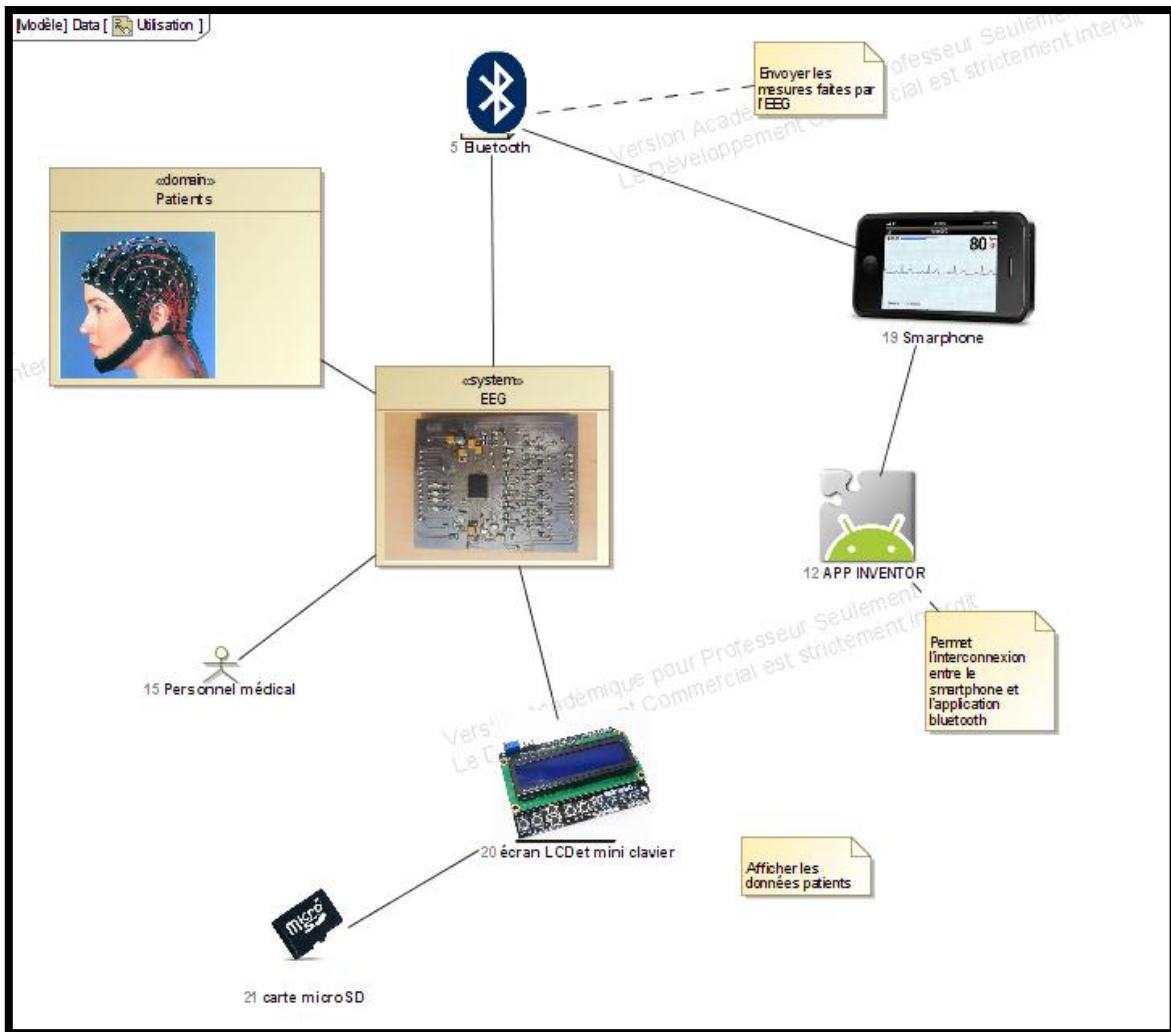
Il faudra en plus réaliser une IHM pour chacun de ces prototypes.



# Diagramme de contexte (ECG)

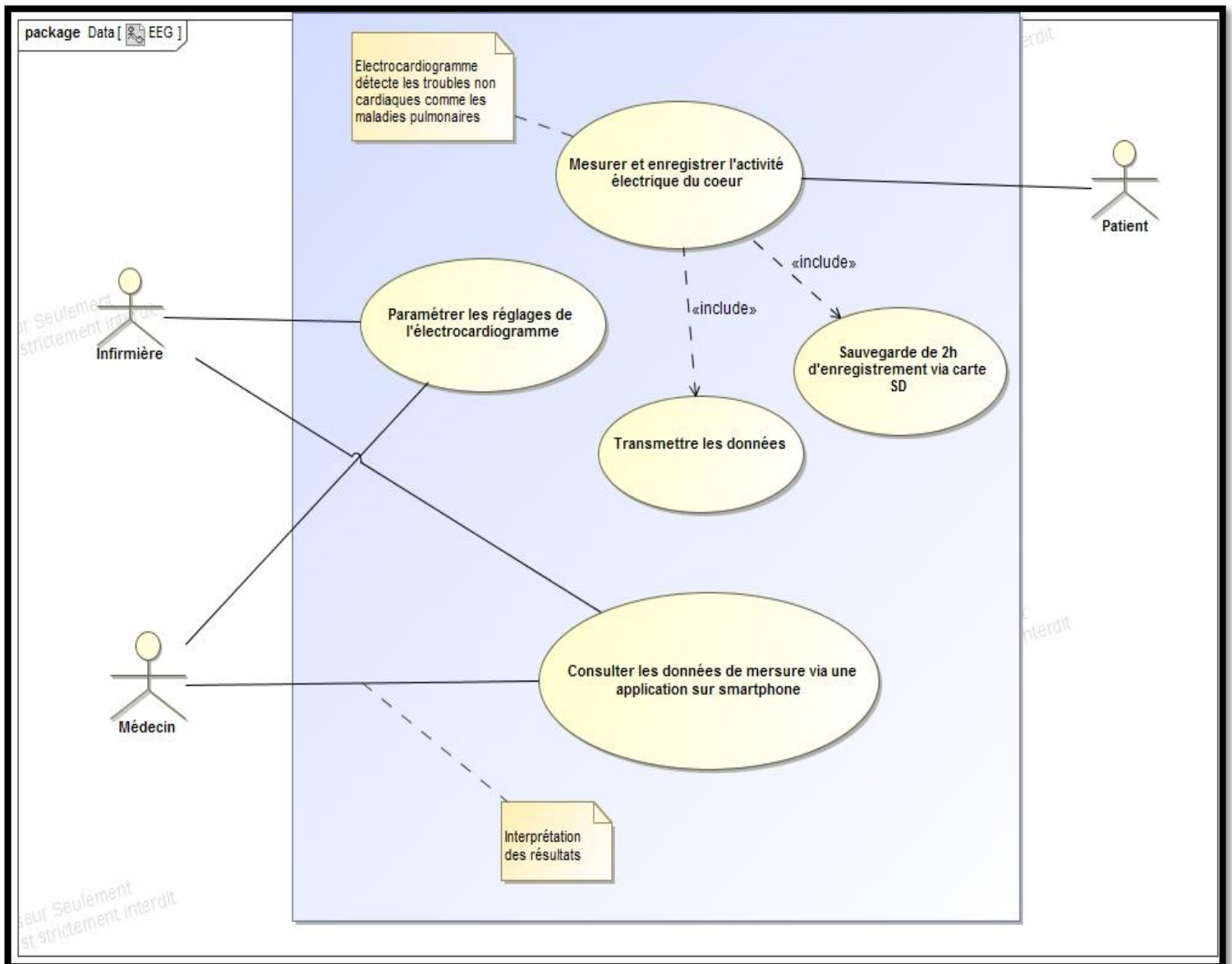


# Diagramme de contexte (EEG)

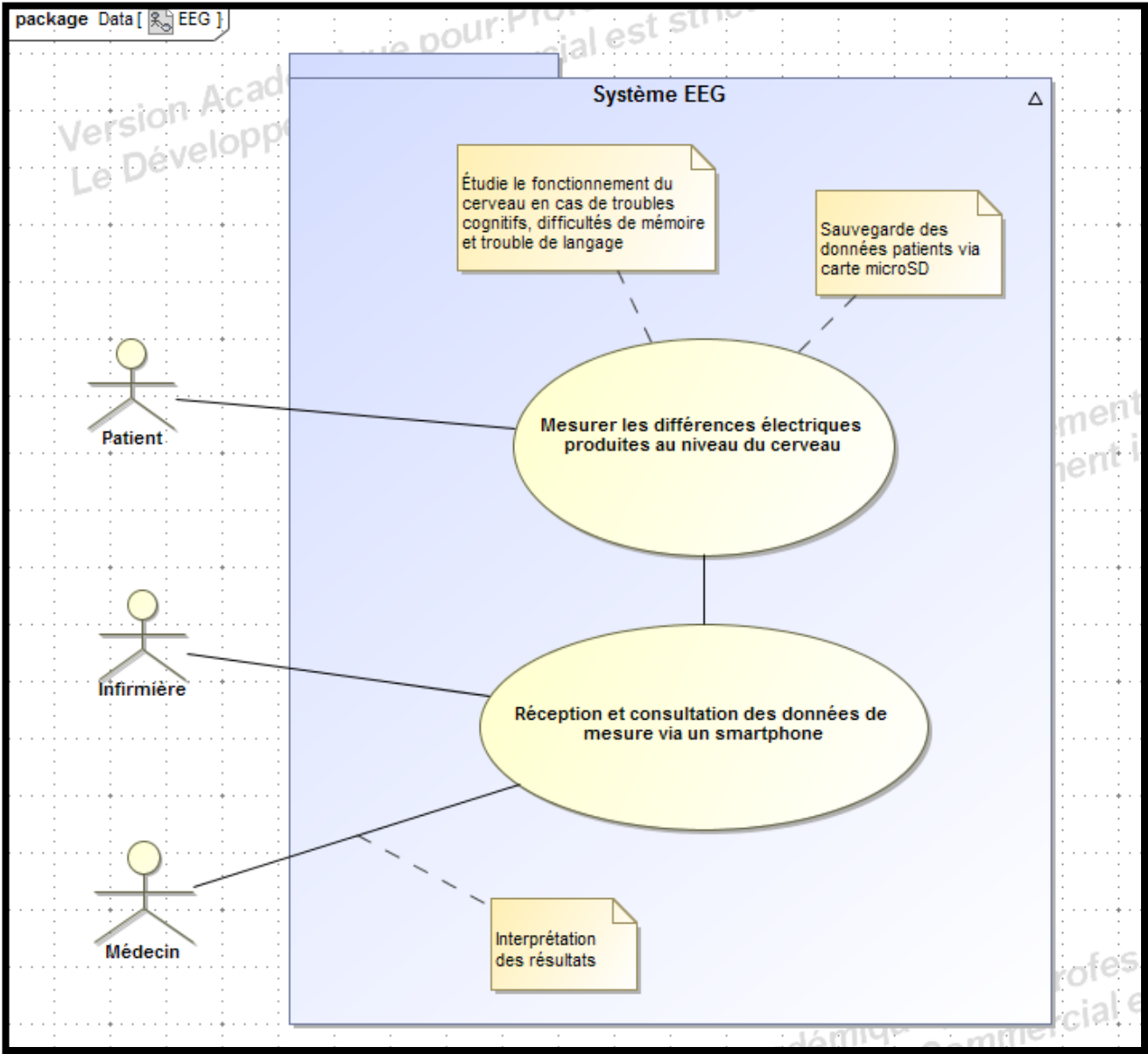




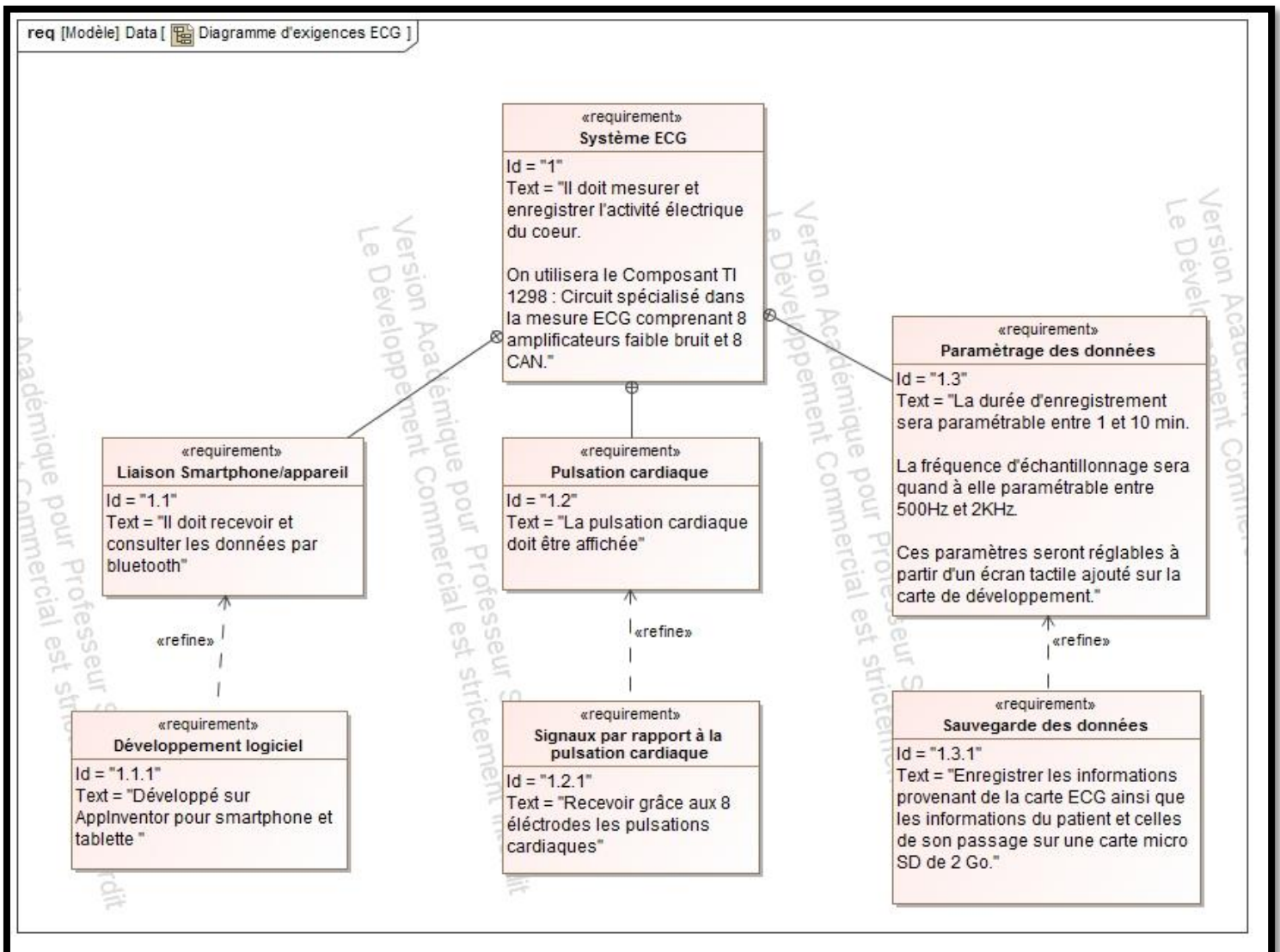
# Diagramme de cas d'utilisation (ECG)



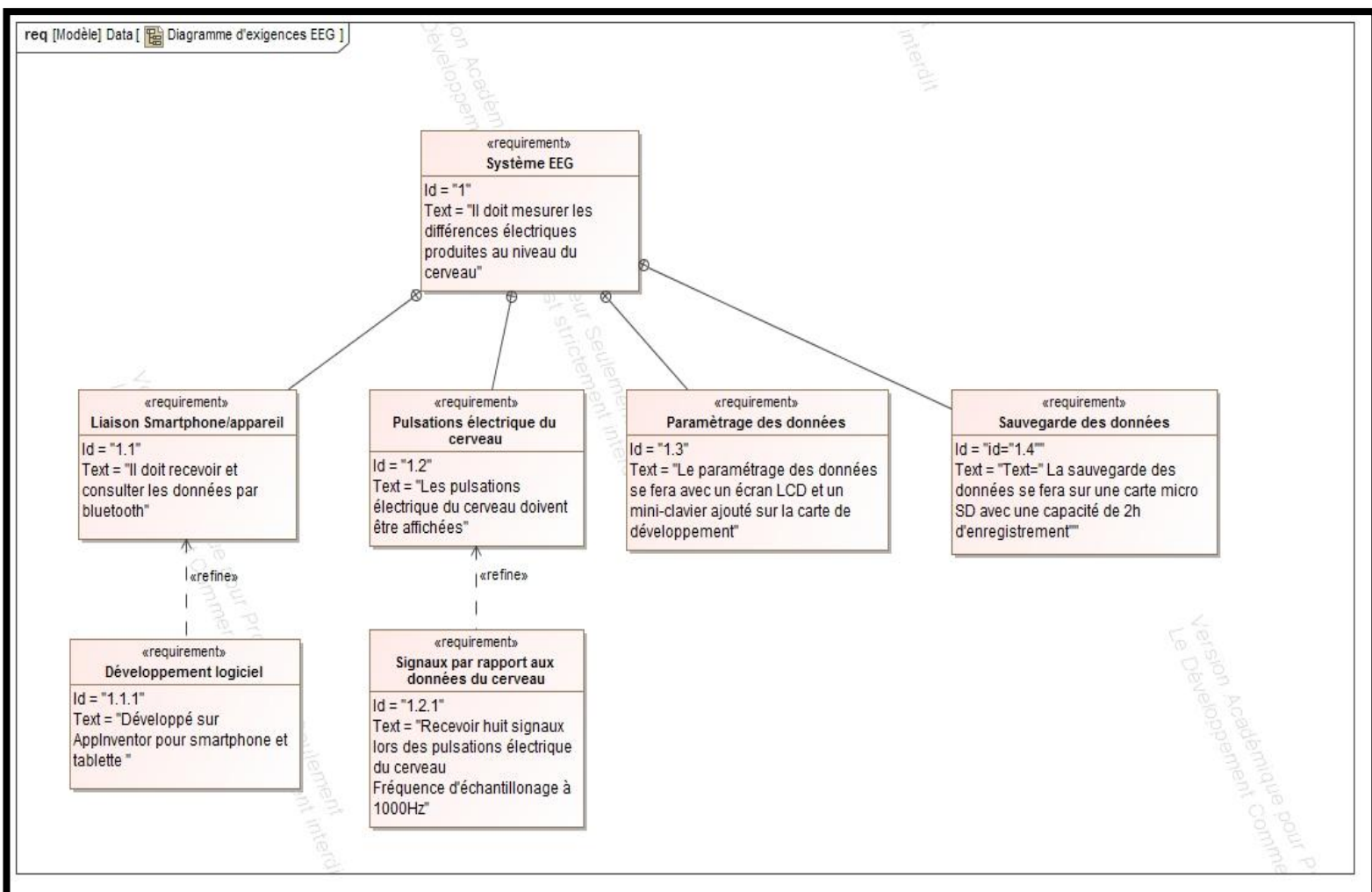
# Diagramme de cas d'utilisation (EEG)



# Diagramme d'exigence (ECG)



# Diagramme d'exigence (EEG)



## D. Répartition des tâches :

### Etudiant E1 (Corentin Maillot) :

- Etudier les connexions à utiliser pour implanter un module bluetooth sur le prototype final.
- Présenter ce travail aux autres étudiants du groupe qui devront le réutiliser.
- Prendre connaissance des caractéristiques matérielles et logicielles du circuit ADS1298 (ECG) et concevoir l'interface entre la carte conversion et une carte de prototypage rapide (en collaboration avec l'étudiant E3 qui réalise la même tâche pour l').
- Développer la routine logicielle permettant l'acquisition des résultats de mesure des 8 sondes.
- Réaliser un dossier de fabrication pour la carte d'interface en incluant le travail de l'étudiant E2 pour la partie IHM.
- Réaliser le routage de cette carte
- Faire la mise au point et la recette du prototype final avec l'étudiant E1.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E1.
- Rédiger le rapport final (une partie commune et une partie personnelle).

### Etudiant E2 (Jean-Baptiste Leduc) :

- Concevoir l'interface entre la carte de prototypage rapide, l'écran tactile et la carte SD.
- Présenter ce travail à l'étudiant E1 qui va réaliser le routage de la carte.
- Réaliser le câblage de cette carte.
- Développer la routine logicielle de l'IHM sur écran tactile.
- Développer la routine logicielle permettant la sauvegarde sur la carte SD.
- Présenter ces routines aux autres étudiants du groupe qui devront les réutiliser.
- Faire la mise au point et la recette du prototype final avec l'étudiant E1.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E1.
- Rédiger le rapport final (une partie commune et une partie personnelle).

### **Etudiant E3 (Gaétan Duquesne) :**

- Prendre connaissance des caractéristiques matérielles et logicielles du circuit ADS1299 (EEG) et concevoir l'interface entre la carte conversion, le module bluetooth et une carte de prototypage rapide.
- Développer la routine logicielle permettant la transmission des informations par le module bluetooth, présenter et communiquer ce travail aux autres étudiants du groupe qui vont le réutiliser pour la partie ECG.
- Réaliser un dossier de fabrication pour la carte d'interface en incluant le travail de l'étudiant E4 pour la partie IHM.
- Réaliser le routage de cette carte.
- Faire la mise au point et la recette du prototype final avec l'étudiant E4.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E4.
- Rédiger le rapport final (une partie commune et une partie personnelle).

### **Etudiant E4 (Thomas Franconville) :**

- Concevoir l'interface entre la carte de prototypage rapide, l'écran LCD, le mini clavier et la carte SD.
- Présenter ce travail à l'étudiant E3 qui va réaliser le routage de la carte.
- Réaliser le câblage de cette carte.
- Développer une application Appinventor permettant d'afficher les courbes EEG et ECG sur un smartphone.
- Présenter cette application aux autres étudiants du groupe qui devront la réutiliser.
- Faire la mise au point et la recette du prototype final avec l'étudiant E3.
- Rédiger une notice de maintenance en collaboration avec l'étudiant E3.
- Rédiger le rapport final (une partie commune et une partie personnelle).

# II. PARTIE PERSONNELLE

## A. Etudiant E1 :

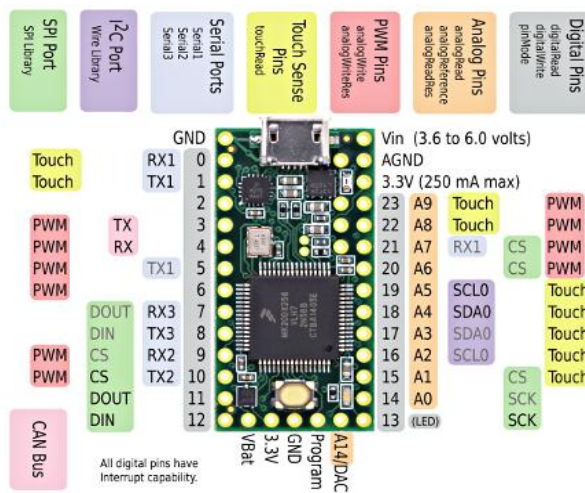
### Elément a disposition :

❖ Module Bluetooth :



- Alimentation de 3.6V à 6V
- 4 Connections

❖ Carte de prototypage (Teensy 3.2) :



- Alimentation de 3.6V à 6V
- microcontrôleur



❖ Composant Ti 1298 :



- Alimentation de 3.3V et 3.7V

❖ Ecran Tactile TFT :



- Alimentation de 3.6V à 5V

- 8 connections

❖ Carte SD:

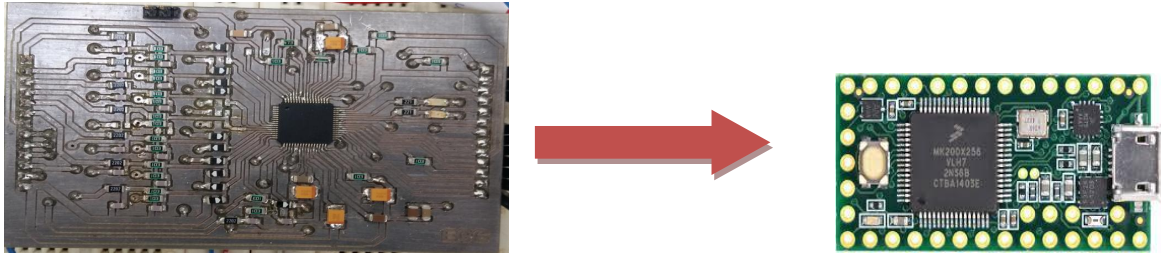


-Carte Micro SD de 2Go

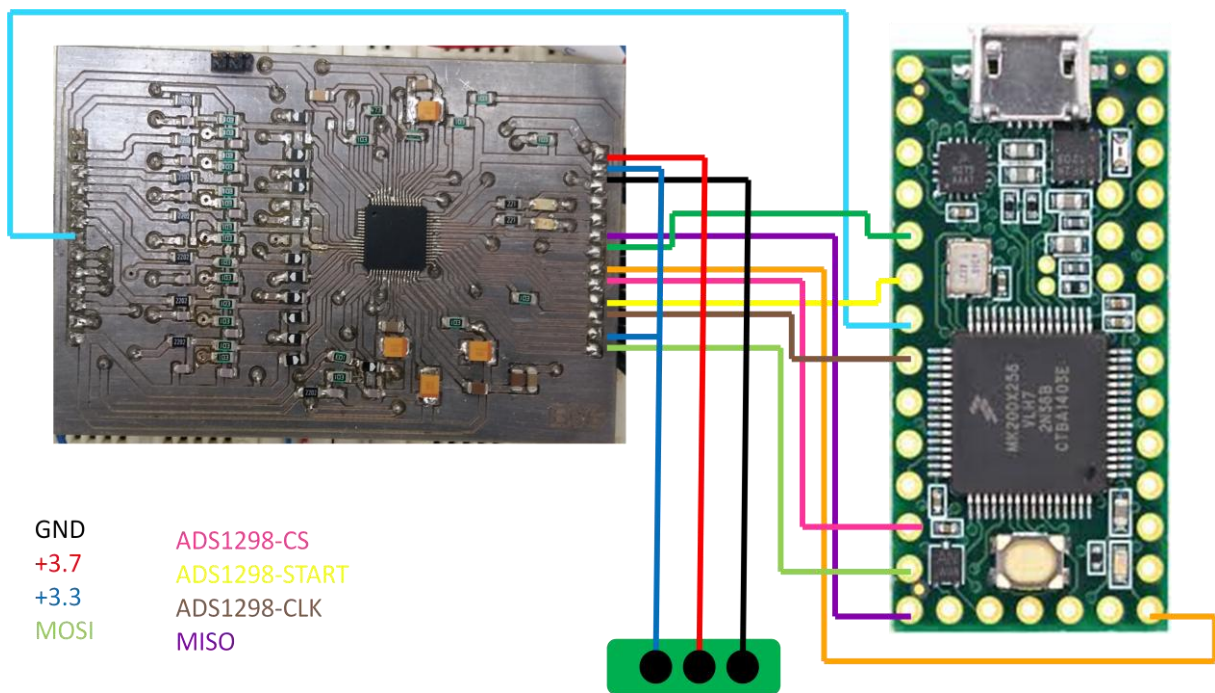


Etudiant E1 :

1) Concevoir l'interface entre la carte conversion et une carte de prototypage rapide :



### Câblage



## Codage

```
#include "ADS129X.h"
#include <SPI.h>

/* ADS129X pins */
const int ADS_RESET = 3;
const int ADS_START = 4;
const int ADS_DRDY = 5;
const int ADS_CS = 10;
byte CONF3;
byte CONF2;
byte CONF1;
byte ID;

/* LEDs */
const int LED1 = 14;

ADS129X ADS = ADS129X(ADS_DRDY, ADS_CS);

void setup() {

  pinMode(ADS_START, OUTPUT);
  pinMode(LED1, OUTPUT);
  digitalWrite(LED1, HIGH);
  delay(100); // wait for PSUs to come UP
  digitalWrite(ADS_START, LOW);
  digitalWrite(ADS_RESET, HIGH);
  delay(100); // delay for power-on-reset (Datasheet, pg. 48)
  // reset pulse
  digitalWrite(ADS_RESET, LOW);
  delay(100);
  digitalWrite(ADS_RESET, HIGH);
  delay(100); // Wait for 18 tCLKs AKA 9 microseconds, we use 1 millisecon
  Serial.begin(115200); // always at 12Mbit/s
  Serial3.begin(115200); // initialisation UART ***Bleutooth***

  ADS.configChannel(1, false, ADS129X_GAIN_1X, ADS129X_MUX_NORMAL);
  //ADS.configChannel(1, false, ADS129X_GAIN_1X, ADS129X_MUX_SHORT);
  // ADS.configChannel(1, false, ADS129X_GAIN_1X, ADS129X_MUX_TEMP);
  ADS.configChannel(2, false, ADS129X_GAIN_1X, ADS129X_MUX_NORMAL
  );
  for (int i = 3; i <= 8; i++) {
    ADS.configChannel(i, false, ADS129X_GAIN_1X, ADS129X_MUX_SHORT);
  }
  ADS.WREG(0x03, 0xC0); //CONFIG3 ref interne
  ADS.WREG(0x02, 0x20); //CONFIG2

  delay(10);
  ID = (ADS.getDeviceId());
  CONF1 = (ADS.RREG(0x01));
  CONF2 = (ADS.RREG(0x20 | 0x02));
  CONF3 = (ADS.RREG(0x20 | 0x03));
  ADS.RDATAC();
  delay(10);
```

```

ID = (ADS.getDeviceId());
CONF1 = (ADS.RREG(0x01));
CONF2 = (ADS.RREG(0x20 | 0x02));
CONF3 = (ADS.RREG(0x20 | 0x03));
ADS.RDATAC();
delay(10);
ADS.START();
delay(10);
}
void loop() {

  long buffer[9];
  static unsigned long tLast;
  digitalWrite(LED1, LOW);
  delay(600);
  if (millis() - tLast > 500) {
    tLast = millis();
  }

  if (ADS.getData(buffer)) {

    digitalWrite(LED1, HIGH);
    delay(500);

    // convert long to bytes
    byte value[3];

    byte value2[3];

    value[0] = (byte) (buffer[1] >> 16);
    value[1] = (byte) (buffer[1] >> 8);
    value[2] = (byte) (buffer[1]);
    value2[0] = (byte) (buffer[2] >> 16);
    value2[1] = (byte) (buffer[2] >> 8);
    value2[2] = (byte) (buffer[2]);

    // Serial.println(ADS.getDeviceId(),HEX);
    delay(100);
    Serial.print("ID: ");

    Serial.println(ID, DEC);
    Serial.print("CONF1: ");

    Serial.println(CONF1, HEX);
    Serial.print("CONF2: ");

    Serial.println(CONF2, HEX);
    Serial.print("CONF3: ");

    Serial.println(CONF3, HEX);

```

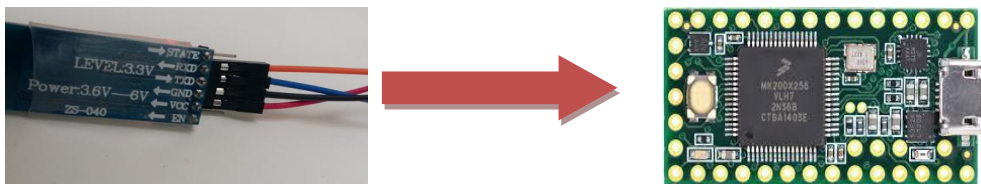
```

Serial.print("Entree1: ");
Serial.println(value[0], HEX);
Serial.print(value[0], HEX);
//Serial.write(value[0]);
Serial.print(value[1], HEX);
//Serial.write(value[1]);
Serial.println(value[2], HEX);
Serial.print("Entree2: ");
Serial.println(value2[0], HEX);
Serial.print(value2[0], HEX);
Serial.print(value2[1], HEX);
Serial.println(value2[2], HEX);
delay(100);
digitalWrite(LED1, LOW);

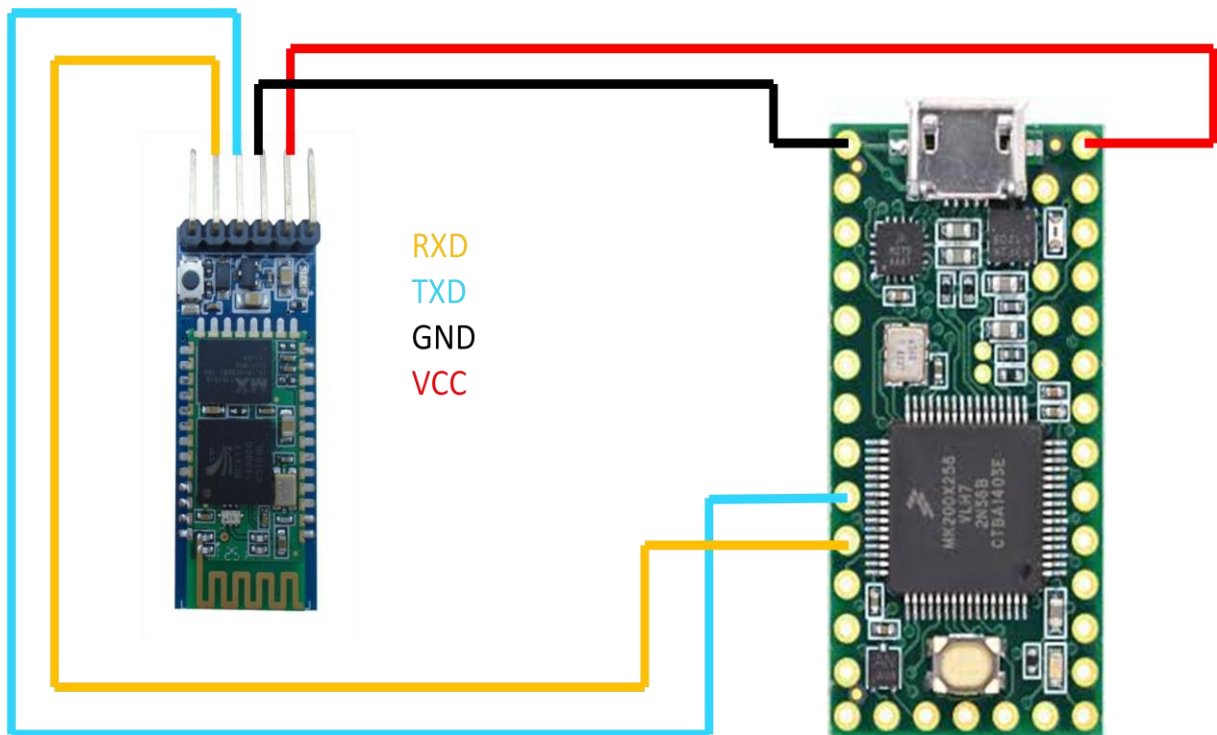
Serial.println("Fin ");
}
}

```

## 2) Implanter le module Bluetooth sur la carte de prototypage rapide



## Câblage



## Codage

Comme nous nous sommes connectés sur les broches 7 et 8 de la carte Teensy ce qui correspond à un port Série en l'occurrence le numéro 3 (TX3 & RX3). Nous devons donc envoyer les données Bluetooth sur le port Série 3.

```
Serial3.begin(115200); // initialisation UART ***Bluetooth***
```

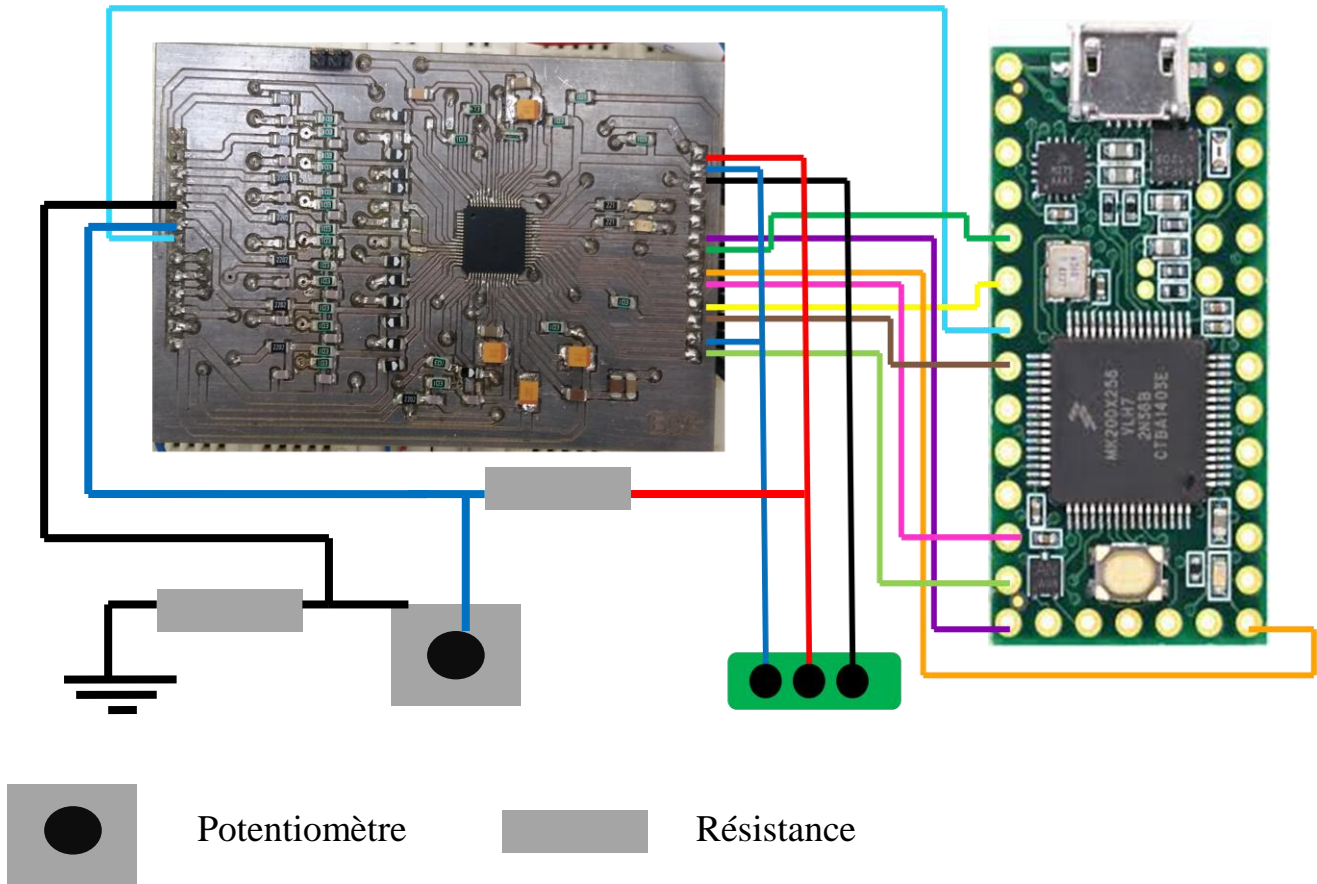
Il faut ensuite placer l'envoi de données dans le loop pour envoyer en boucle.

```
Serial3.print("Entree1: ");  
Serial3.println(value[0], HEX);  
Serial3.print(value[0], HEX);  
Serial3.print(value[1], HEX);  
Serial3.println(value[2], HEX);  
  
Serial3.print("Entree2: ");  
Serial3.println(value2[0], HEX);  
Serial3.print(value2[0], HEX);  
Serial3.print(value2[1], HEX);  
Serial3.println(value2[2], HEX);  
delay(100);
```

Extrait de code

3) Développer la routine logicielle permettant l'acquisition des résultats de mesure des 8 sondes :

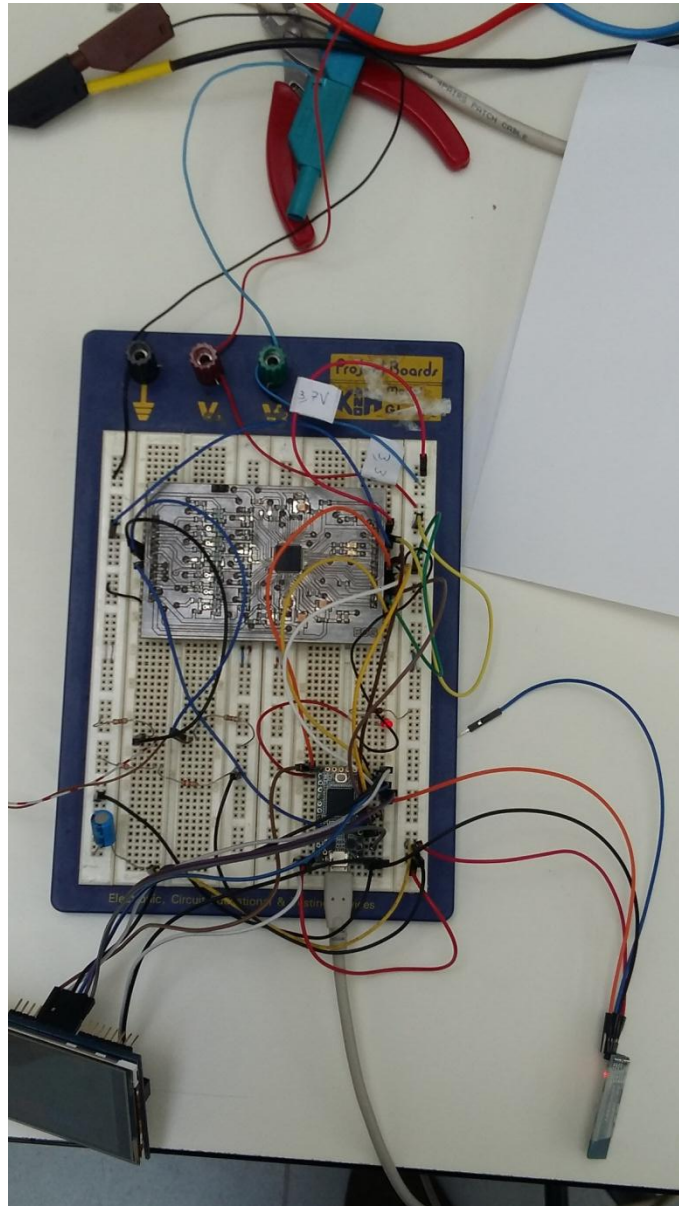
Nous n'avons pas de sonde à disposition pour cela nous utiliserons un potentiomètre qui s'occupera de faire varier les tensions





4) Réaliser un dossier de fabrication pour la carte d'interface et réaliser le routage de cette carte :

Montage actuelle :

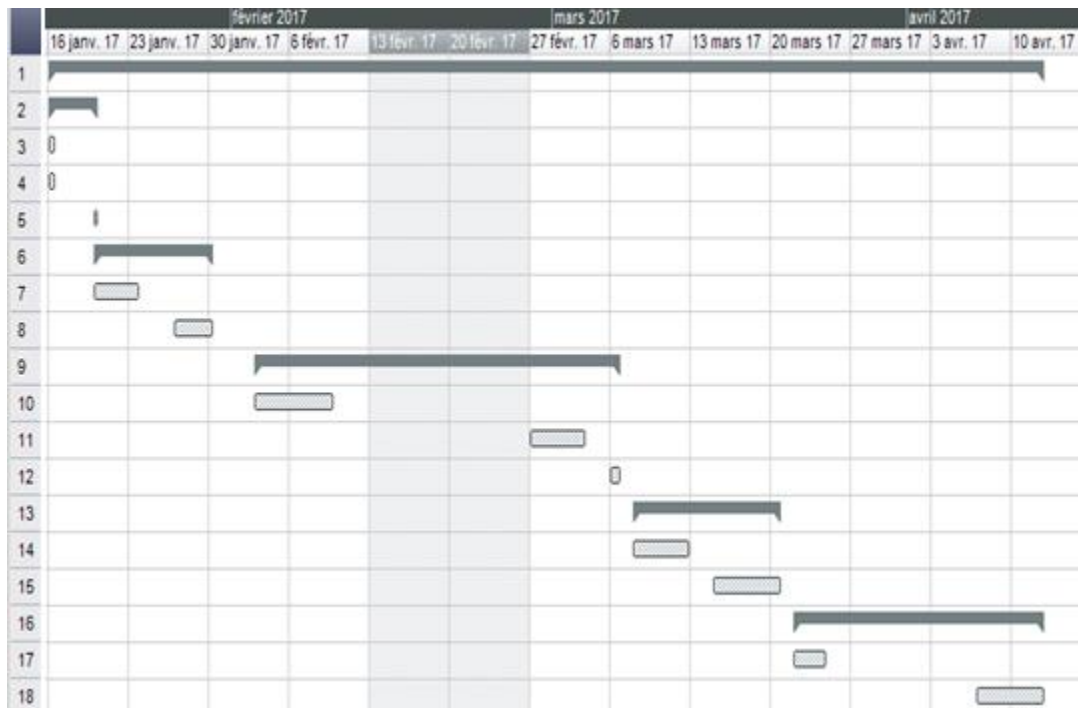


Pour éviter d'avoir trop de fil nous allons créer une carte de support à l'aide de Protéus (logiciel de simulation). Nous devons d'abord réaliser le câblage de tous les composants sur Isis et ensuite le faire sur Ares ce qui va nous permettre de créer des fichiers de fabrication pour réaliser notre carte de support.





	Nom de tâche	Durée	Début	Fin
1	<b>ECG</b>	15,5 jours	16/01/2017	12/04/2017
2	<b>SPECIFICATION / PLANIFICATION</b>	0,75 jour	16/01/2017	20/01/2017
3	Prendre connaissance du support du projet	2 hr	16/01/2017	16/01/2017
4	Décoder le cahier des charges	2 hr	16/01/2017	16/01/2017
5	S'approprier la répartition des tâches communiquées par les enseignants	2 hr	20/01/2017	20/01/2017
6	<b>CONCEPTION PRELIMINAIRE</b>	1,75 jours	20/01/2017	30/01/2017
7	Préparer une présentation Sysml du projet	8 hr	20/01/2017	23/01/2017
8	Elaborer un planning des activités pour la durée du projet	6 hr	27/01/2017	30/01/2017
9	<b>CONCEPTION DETAILLEE</b>	4,5 jours	03/02/2017	06/03/2017
10	Etudier les connexions pour implanter un module bluetooth sur le prototype final.	12 hr	03/02/2017	10/02/2017
11	Présenter ce travail aux autres étudiants du groupe qui devront le réutiliser.	12 hr	27/02/2017	03/03/2017
12	Développer la routine logicielle permettant l'acquisition de mesure des 8 sondes	4 hr	06/03/2017	06/03/2017
13	<b>PROTOTYPAGE / REALISATION</b>	3,5 jours	08/03/2017	20/03/2017
14	Réaliser un dossier de fabrication pour la carte d'interface.	12 hr	08/03/2017	13/03/2017
15	Réaliser le routage de cette carte	12 hr	15/03/2017	20/03/2017
16	<b>QUALIFICATION - INTEGRATION - VALIDATION</b>	5,5 jours	22/03/2017	12/04/2017
17	Faire la recette et la mise au point et la recette du prototype final.	8 hr	22/03/2017	24/03/2017
18	Rédiger le rapport final	12 hr	07/04/2017	12/04/2017




















## B. Etudiant E2 :

Notre projet consiste à la création d'un système comprenant un ECG et un EEG. Nous nous concentrerons sur la partie ECG. Ici notre ECG sera composé d'un écran tactile permettant le paramétrage de la fréquence d'échantillonnage ainsi que la durée d'enregistrement.

Ma partie du projet consiste à la création de l'interface graphique, de la sauvegarde des signaux et des numéros des patients sur une carte micro SD et de la insertion de l'écran tactile au système, au microcontrôleur (Teensy 3.1) .

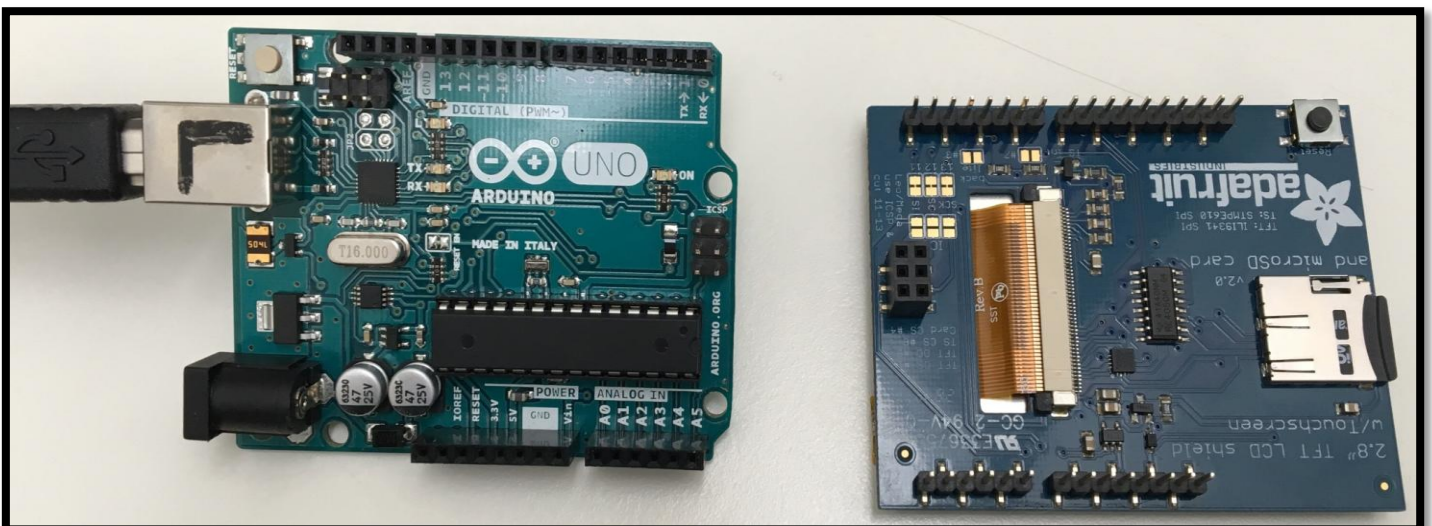
Ci-dessous se trouve mon diagramme de Gantt, c'est la représentation visuelle de l'état d'avancement des différentes activités (tâches) qui constituent le projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.). Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin.

L'encadrement en rouge montre à quelle étape du projet je me trouve lors de la rédaction de ce dossier.

			Nom de tâche
1			<b>Sujet</b>
2			<input type="checkbox"/> <b>1. Spécification/planification</b>
3			1.1 Elaborer un planning des activités pour la durée du projet.
4			1.2 En collaboration avec les autres étudiants, préparer une présentation Sysml du projet.
5			<input type="checkbox"/> <b>2. Conception préliminaire</b>
6			2.1 Revue de projet 1
7			2.2 Revue de projet 2
8			2.3 Revue de projet 3
9			<input type="checkbox"/> <b>3. Conception détaillée</b>
10			3.1 Réaliser le câblage de cette carte.
11			3.2 Présenter ce travail à l'étudiant E1 qui va réaliser le routage de la carte.
12			<input type="checkbox"/> <b>4. Prototypage/réalisation</b>
13			4.1 Développer la routine logicielle de l'IHM sur écran tactile.
14			4.2 Développer la routine logicielle permettant la sauvegarde sur la carte SD.
15			4.3 Présenter ces routines aux autres étudiants du groupe qui devront les réutiliser.
16			<input type="checkbox"/> <b>5. Idée</b>
17			5.1 Idée
18			5.2 Rédiger une notice de maintenance.
19			5.3 Rédiger le rapport
20			5.4 Finitions et présentation du projet

	Durée	Début	Fin	Prédécesseurs	T1 2017			T2 2017		
					Janv. 17	févr. 17	mars 17	avr. 17	mai 17	juin 17
1	54 jours	16/01/2017	21/06/2017		[Bar chart showing a long duration across the period]					
2	10 jours	16/01/2017	27/02/2017		[Bar chart showing duration in Jan/Feb]					
3	1 jour	16/01/2017	16/01/2017		[Single bar in Jan]					
4	10 jours	16/01/2017	27/02/2017		[Bar chart showing duration in Jan/Feb]					
5	33 jours	23/01/2017	15/05/2017		[Bar chart showing duration from Jan to May]					
6	1 hr	23/01/2017	23/01/2017		[Single bar in Jan]					
7	2 jours	06/03/2017	08/03/2017	6	[Bar chart showing duration in Mar]					
8	2 jours	12/05/2017	15/05/2017	7	[Bar chart showing duration in May]					
9	9 jours	27/02/2017	15/03/2017		[Bar chart showing duration in Feb]					
10	3 jours	27/02/2017	03/03/2017		[Bar chart showing duration in Feb]					
11	4 hr	15/03/2017	15/03/2017	10	[Single bar in Mar]					
12	31,12 jours	08/03/2017	02/06/2017		[Bar chart showing duration from Mar to Jun]					
13	15 jours	08/03/2017	07/04/2017		[Bar chart showing duration in Mar]					
14	9 jours	12/05/2017	31/05/2017	13	[Bar chart showing duration in May]					
15	4 hr	31/05/2017	02/06/2017	14	[Single bar in May]					
16	28 jours	24/04/2017	21/06/2017		[Bar chart showing duration from Apr to Jun]					
17	3 jours	24/04/2017	28/04/2017		[Bar chart showing duration in Apr]					
18	5 jours	24/04/2017	10/05/2017		[Bar chart showing duration from Apr to May]					
19	3 jours	15/05/2017	19/05/2017	18	[Bar chart showing duration in May]					
20	17 jours	22/05/2017	21/06/2017	19	[Bar chart showing duration from May to Jun]					

Dans un premier temps j'ai travaillé sur un écran tactile 2.8" TFT Touch Shield V2 et carte Arduino Uno afin de créer le code de l'interface graphique.



Veillez trouver ci-dessous, le code de l'écran tactile.

Code temporaire et dont la version définitive sera présenter durant l'oral de projet.

```

1. //Utilisation des librairies permettant le fonctionnement de l'écran
   tactile
2. #include <Adafruit_GFX.h>
3. #include <SPI.h>
4. #include <Wire.h>
5. #include <Adafruit_ILI9341.h>
6. #include <Adafruit_STMPE610.h>
7. #include <SD.h>
8.
9. // calibration des coordonnées de l'écran
10.     #define TS_MINX 150
11.     #define TS_MINY 130
12.     #define TS_MAXX 3800
13.     #define TS_MAXY 4000
14.
15.     //Définition des broches de connexion du microcontrôleur
16.     #define STMPE_CS 8
17.     Adafruit_STMPE610 ts = Adafruit_STMPE610(STMPE_CS);
18.     #define TFT_CS 10
19.     #define TFT_DC 9
20.     #define CARD_CS 4
21.     Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);
22.
23.     boolean RecordOn = false;
24.
25.     //Coordonnées du bouton 1
26.     #define FRAME_X 60
27.     #define FRAME_Y 180
28.     #define FRAME_W 200
29.     #define FRAME_H 40
30.
31.     //Coordonnées du bouton 2
32.     #define FRAME_X2 10
33.     #define FRAME_Y2 40
34.     #define FRAME_W2 100
35.     #define FRAME_H2 20
36.
37.     #define REDBUTTON_X FRAME_X
38.     #define REDBUTTON_Y FRAME_Y
39.     #define REDBUTTON_W (FRAME_W/2)
40.     #define REDBUTTON_H FRAME_H
41.
42.     #define REDBUTTON_X2 FRAME_X2
43.     #define REDBUTTON_Y2 FRAME_Y2 // test
44.     #define REDBUTTON_W2 (FRAME_W2/2)
45.     #define REDBUTTON_H2 FRAME_H2
46.
47.     #define GREENBUTTON_X (REDBUTTON_X + REDBUTTON_W)

```

```

48.     #define GREENBUTTON_Y FRAME_Y
49.     #define GREENBUTTON_W (FRAME_W/2)
50.     #define GREENBUTTON_H FRAME_H
51.
52.     #define GREENBUTTON_X2 (REDBUTTON_X2 + REDBUTTON_W2)    // test
53.     #define GREENBUTTON_Y2 FRAME_Y2
54.     #define GREENBUTTON_W2 (FRAME_W2/2)
55.     #define GREENBUTTON_H2 FRAME_H2
56.
57.     void drawFrame()
58.     {
59.         tft.drawRect(FRAME_X, FRAME_Y, FRAME_W, FRAME_H,
ILI9341_BLACK);
60.         tft.drawRect(FRAME_X2, FRAME_Y2, FRAME_W2, FRAME_H2,
ILI9341_BLACK);
61.     }
62.
63.     //Création du bouton "+"
64.     void rougeBtn()
65.     {
66.         tft.fillRect(REDBUTTON_X2, REDBUTTON_Y2, REDBUTTON_W2,
REDBUTTON_H2, ILI9341_RED);
67.         tft.fillRect(GREENBUTTON_X2, GREENBUTTON_Y2, GREENBUTTON_W2,
GREENBUTTON_H2, ILI9341_BLUE);
68.         drawFrame();
69.         tft.setCursor(GREENBUTTON_X2 + 6 , GREENBUTTON_Y2 +
(GREENBUTTON_H2/2));
70.         tft.setTextColor(ILI9341_WHITE);
71.         tft.setTextSize(2);
72.         tft.println(" +");
73.         RecordOn = false;
74.     }
75.
76.     //Création du bouton "-"
77.     void vertBtn()
78.     {
79.         tft.fillRect(GREENBUTTON_X2, GREENBUTTON_Y2, GREENBUTTON_W2,
GREENBUTTON_H2, ILI9341_GREEN);
80.         tft.fillRect(REDBUTTON_X2, REDBUTTON_Y2, REDBUTTON_W2,
REDBUTTON_H2, ILI9341_BLUE);
81.         drawFrame();
82.         tft.setCursor(REDBUTTON_X2 + 6 , REDBUTTON_Y2 +
(REDBUTTON_H2/2));
83.         tft.setTextColor(ILI9341_WHITE);
84.         tft.setTextSize(2);
85.         tft.println(" -");
86.         RecordOn = true;
87.     }

```

```

88.
89.     //Création du bouton "START"
90.     void redBtn()
91.     {
92.         tft.fillRect(REDBUTTON_X, REDBUTTON_Y, REDBUTTON_W,
REDBUTTON_H, ILI9341_RED);
93.         tft.fillRect(GREENBUTTON_X, GREENBUTTON_Y, GREENBUTTON_W,
GREENBUTTON_H, ILI9341_BLUE);
94.         drawFrame();
95.         tft.setCursor(GREENBUTTON_X + 6 , GREENBUTTON_Y +
(GREENBUTTON_H/2));
96.         tft.setTextColor(ILI9341_WHITE);
97.         tft.setTextSize(2);
98.         tft.println(" START");
99.         RecordOn = false;
100.    }
101.
102.    //Création du bouton "OFF"
103.    void greenBtn()
104.    {
105.        tft.fillRect(GREENBUTTON_X, GREENBUTTON_Y, GREENBUTTON_W,
GREENBUTTON_H, ILI9341_GREEN);
106.        tft.fillRect(REDBUTTON_X, REDBUTTON_Y, REDBUTTON_W,
REDBUTTON_H, ILI9341_BLUE);
107.        drawFrame();
108.        tft.setCursor(REDBUTTON_X + 6 , REDBUTTON_Y +
(REDBUTTON_H/2));
109.        tft.setTextColor(ILI9341_WHITE);
110.        tft.setTextSize(2);
111.        tft.println(" OFF");
112.        RecordOn = true;
113.    }
114.    void setup(void)
115.    {
116.        int frequence = 100;
117.        int seconde = 100;
118.        Serial.begin(9600);
119.        tft.begin();
120.        if (!ts.begin()) {
121.            Serial.println("Unable to start touchscreen.");
122.        }
123.        else {
124.            Serial.println("Touchscreen started.");
125.        }
126.
127.        tft.fillScreen(ILI9341_BLACK);
128.        // origin = left,top landscape (USB left upper)
129.        tft.setRotation(1);

```

```

130.     redBtn();
131.     rougeBtn();
132.
133.     //Création d'un cadre de couleur blanc
134.     tft.drawRect(0,0,319,240,ILI9341_WHITE);
135.
136.     //Affiche le texte : "Fréquence :"
137.     tft.setCursor(5,2);
138.     tft.setTextColor(ILI9341_WHITE);
139.     tft.setTextSize(2);
140.     tft.print("Freq : ");
141.
142.     //Affiche le texte : "Durée :"
143.     tft.setCursor(150,2);
144.     tft.setTextColor(ILI9341_WHITE);
145.     tft.setTextSize(2);
146.     tft.print("Duree :");
147.
148.     //Affichage de la variable temps (Durée)
149.     tft.setCursor(230,2);
150.     tft.setTextColor(ILI9341_YELLOW);
151.     tft.setTextSize(2);
152.     tft.print(seconde);
153.
154.     //Affichage de la variable frequence (Fréquence)
155.     tft.setCursor(73,2);
156.     tft.setTextColor(ILI9341_YELLOW);
157.     tft.setTextSize(2);
158.     tft.print(frequence);
159.
160.
161. void loop()
162. {
163.     //Analyse si des données tactile sont reçus
164.     if (!ts.bufferEmpty())
165.     {
166.         // Retrouve un point
167.         TS_Point p = ts.getPoint();
168.         // Echelle à l'aide de la calibration reçu plus haut
169.         // Et rotation des coordonnées du système
170.         p.x = map(p.x, TS_MINY, TS_MAXY, 0, tft.height());
171.         p.y = map(p.y, TS_MINX, TS_MAXX, 0, tft.width());
172.         int y = tft.height() - p.x;
173.         int x = p.y;
174.
175.         if (RecordOn)
176.         {

```



```

177.         if((x > REDBUTTON_X) && (x < (REDBUTTON_X +
REDBUTTON_W))) {
178.             if ((y > REDBUTTON_Y) && (y <= (REDBUTTON_Y +
REDBUTTON_H))) {
179.                 Serial.println("Red btn hit");
180.                 redBtn();
181.             }
182.         }
183.     }
184.     else //Record est off (RecordOn == false)
185.     {
186.         if((x > GREENBUTTON_X) && (x < (GREENBUTTON_X +
GREENBUTTON_W))) {
187.             if ((y > GREENBUTTON_Y) && (y <= (GREENBUTTON_Y +
GREENBUTTON_H))) {
188.                 Serial.println("Green btn hit");
189.                 greenBtn();
190.             }
191.         }
192.     }
193.     if (RecordOn)
194.     {
195.         if((x > REDBUTTON_X2) && (x < (REDBUTTON_X2 +
REDBUTTON_W2))) {
196.             if ((y > REDBUTTON_Y2) && (y <= (REDBUTTON_Y2 +
REDBUTTON_H2))) {
197.                 Serial.println("Rouge btn hit");
198.                 rougeBtn();
199.             }
200.         }
201.     }
202.     else //Record est off (RecordOn == false)
203.     {
204.         if((x > GREENBUTTON_X2) && (x < (GREENBUTTON_X2 +
GREENBUTTON_W2))) {
205.             if ((y > GREENBUTTON_Y2) && (y <= (GREENBUTTON_Y2 +
GREENBUTTON_H2))) {
206.                 Serial.println("vert btn hit");
207.                 vertBtn();
208.             }
209.         }
210.     }
211.     Serial.println(RecordOn);
212. }
213. }

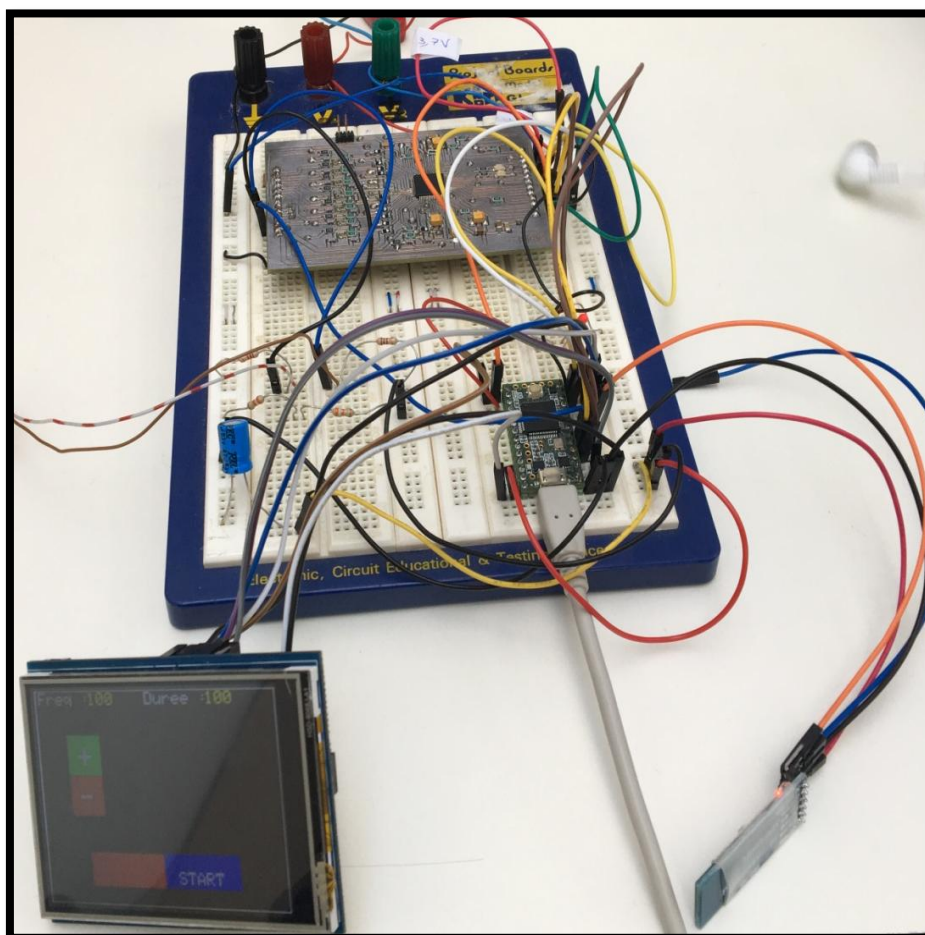
```

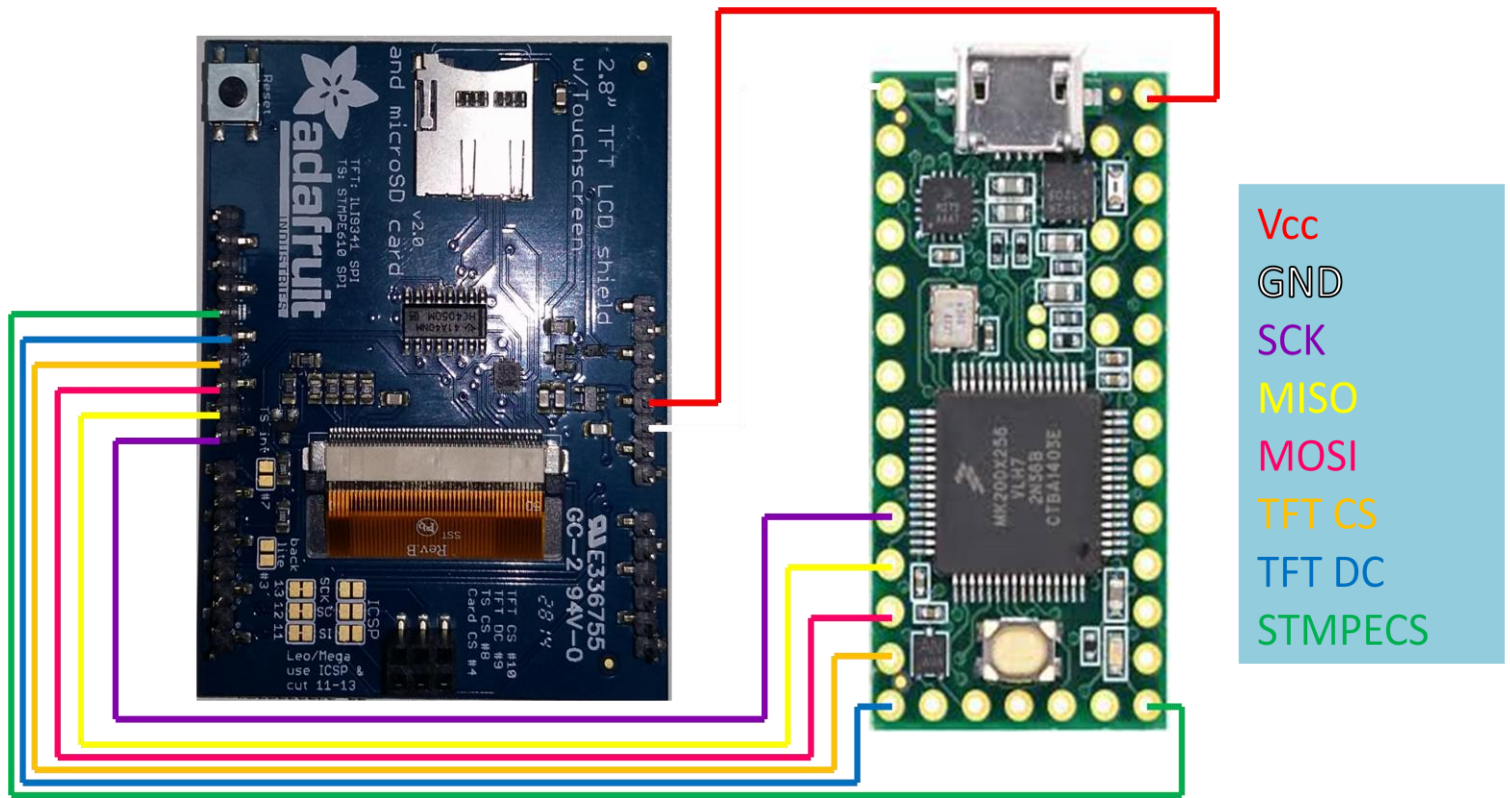
Voici le résultat :



A noter que le code réalisé sur Arduino est compatible avec la carte Teensy.

- **Implantation de l'écran TFT à la carte Teensy**





Ci dessus, se trouve un schéma montrant les liaisons entre l'écran tactile et la carte Teensy.

- **Les limites des sauvegardes de la carte micro SD**

Fréquence d'échantillonnage entre 500Hz - 2000Hz : (1000Hz prédéfini)

1000Hz = 1000 mesures

10 bits/mesure \* 8 électrodes = 80 \* 1000 (mesures) = 80 000 bit/s = **0.01 Mo**

Temps d'enregistrement entre 1min et 10min :

1min = 0.01 Mo \* 60 = **0.6 Mo**

10min = 0.01 Mo \* 600 = **6 Mo**

1H = 0.01 \* 3600 = **36 Mo**

Carte micro SD de 2Go (2000 Mo)

2000 / 36 = **56 heures et 30 minutes**

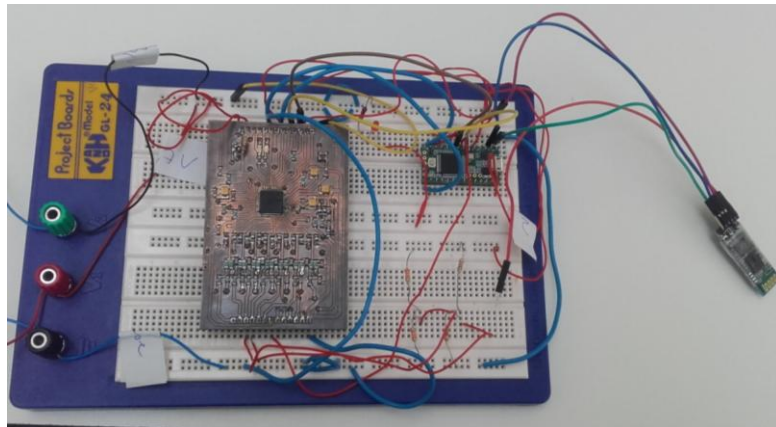
## *C. Etudiant E3 :*

### I/ Présentation de la partie

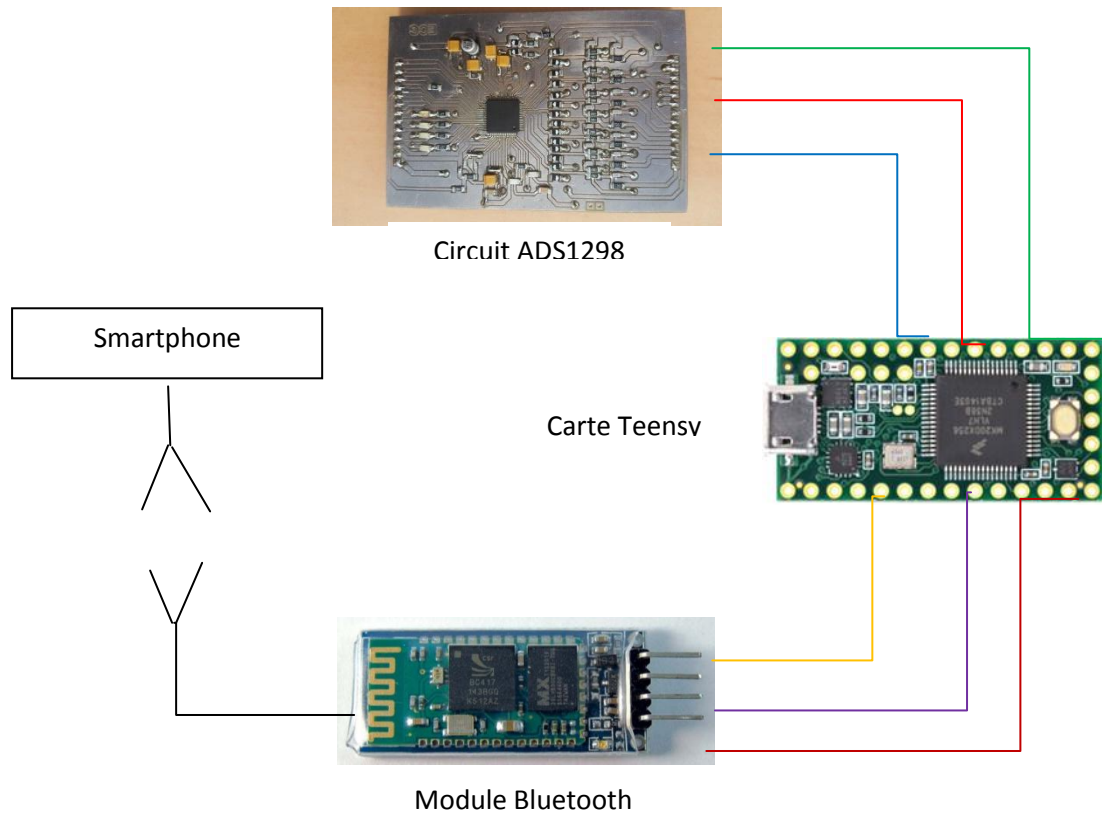
Dans ce projet EEG, ma partie consiste à élaborer les interfaçages des différentes cartes utilisées. Elle met en jeu le circuit ADS1298, la carte Teensy et le module Bluetooth.

Une fois les données converties par le circuit ADS1298, les données sont envoyées par la carte Teensy grâce au module Bluetooth.

Le circuit ADS1298 permet de faire les conversions analogiques-numériques des signaux émis pour l'électroencéphalogramme.



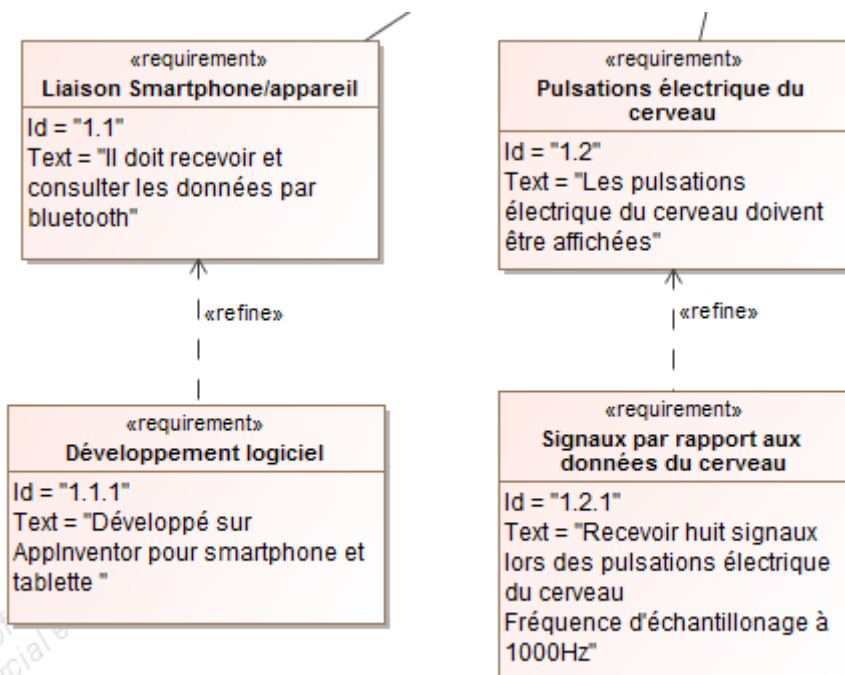
## Schéma fonctionnel



Cette partie consiste à envoyer les données du circuit ADS1298 vers le module bluetooth via la carte Teensy puis permettre la transmission des données en Bluetooth vers le smartphone.

## II/ Diagrammes

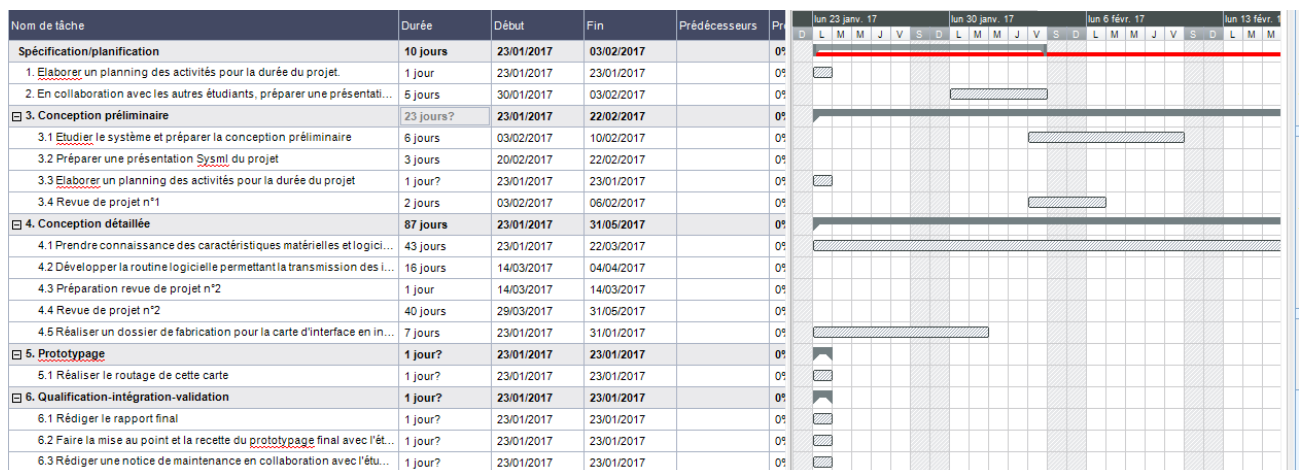
# Diagramme d'exigences



Ce diagramme permet d'établir les différentes contraintes que doit respecter le projet. Il traduit le cahier des charges sous forme d'une décomposition en exigences. Dans ce diagramme, on peut voir les différentes parties et sous-parties qui détaillent les exigences que je dois satisfaire durant mon projet.



# Diagramme de Gantt



Le diagramme de Gantt permet de préparer et suivre l'avancement du projet. En haut, on peut voir les différentes tâches à réaliser accompagnées d'une estimation de leurs durées. On y voit également les personnes concernées par les tâches.

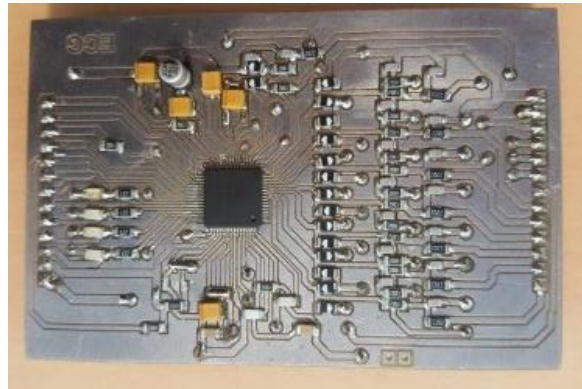
Le diagramme de Gantt n'a pas été respecté, on a pris plus ou moins notre temps pour travailler sur ce projet.



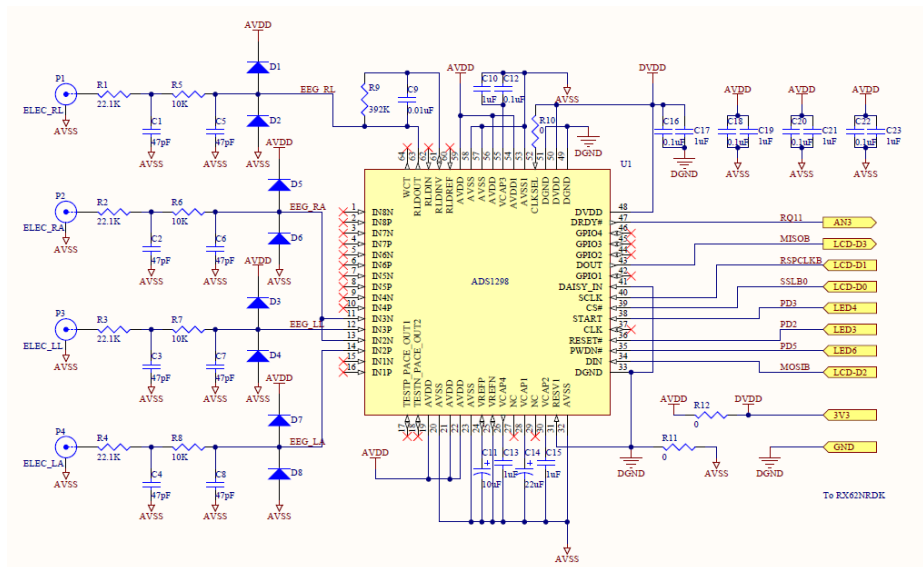
### III/ Ressources et tests

# Matériels nécessaires

Circuit ADS1298 :



Le circuit ADS1298 dispose de huit amplificateurs à gain programmables à faible bruit et huit convertisseurs analogique-numérique à haute résolution (convertisseur CAN). Ces différents CAN sont exigés pour les applications médicales ECG et EEG.



## Carte Teensy :

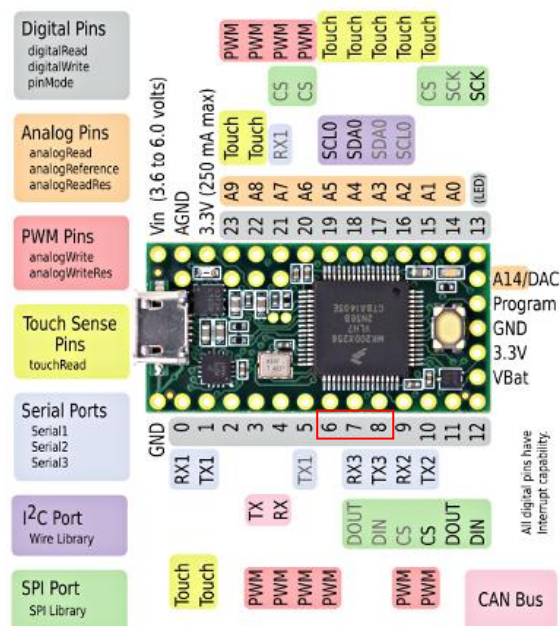


La carte Teesy est une petite carte électronique programmable du style Arduino qui présente certains avantages :

- elle est plus petite,
- elle est plus puissante,
- elle peut être contrôlée en tant que souris, clavier etc...

Mais elle présente un inconvénient, un bouton Mode programme permettant de réinitialiser le programme à chaque fois (la réinitialisation ne se déclenche pas automatiquement).

### Schéma entrée/sortie du brochage de la carte



Module bluetooth :



Le module Bluetooth est un module de communication spécialisé pour des cartes programmables (Arduino, Teensy et Raspberry).

Branchements :

- +5V/-5V
- GND
- RX/Tx

# Tests

## Programmes :

```
const int ADS_DRDY = 5;
const int ADS_CS = 10;
byte CONF3;
byte CONF2;
byte CONF1;
byte ID;

/* LEDs */
const int LED1 = 14;

ADS129X ADS = ADS129X(ADS_DRDY, ADS_CS);

void setup() {

  pinMode(ADS_START, OUTPUT);
  pinMode(LED1, OUTPUT);
  digitalWrite(LED1, HIGH);
  delay(100); // wait for PSUs to come UP
  digitalWrite(ADS_START, LOW);
  digitalWrite(ADS_RESET, HIGH);
  delay(100); // delay for power-on-reset (Datasheet, pg. 48)
  // reset pulse
  digitalWrite(ADS_RESET, LOW);
  delay(100);
  digitalWrite(ADS_RESET, HIGH);
  delay(100); // Wait for 18 tCLKs AKA 9 microseconds, we use 1 millisecc
  Serial.begin(115200); // always at 12Mbit/s

  digitalWrite(ADS_RESET, HIGH);
  delay(100); // Wait for 18 tCLKs AKA 9 microseconds, we use 1 millisecc
  Serial.begin(115200); // always at 12Mbit/s
  Serial.begin(115200); // initialisation UART
  Serial.print("test ");
  ADS.configChannel(1, false, ADS129X_GAIN_1X, ADS129X_MUX_NORMAL);
  //ADS.configChannel(1, false, ADS129X_GAIN_1X, ADS129X_MUX_SHORT);
  // ADS.configChannel(1, false, ADS129X_GAIN_1X, ADS129X_MUX_TEMP);
  ADS.configChannel(3, false, ADS129X_GAIN_1X, ADS129X_MUX_NORMAL);
  for (int i = 4; i <= 8; i++) {
    ADS.configChannel(i, false, ADS129X_GAIN_1X, ADS129X_MUX_SHORT);
  }
  ADS.WREG(0x03, 0xC0); //CONFIG3 ref interne
  ADS.WREG(0x02, 0x20); //CONFIG2

  delay(10);
  ID = (ADS.getDeviceId());
  CONF1 = (ADS.RREG(0x01));
  CONF2 = (ADS.RREG(0x20 | 0x02));
  CONF3 = (ADS.RREG(0x20 | 0x03));
  ADS.RDATAC();
  delay(10);
  ADS.START();
  delay(10);

void loop() {
  long buffer[12];
```

```

long buffer[12];
static unsigned long tLast;
digitalWrite(LED1, LOW);
delay(600);
if (millis() - tLast > 500) {
    tLast = millis();
}

if (ADS.getData(buffer) ) {
    digitalWrite(LED1, HIGH);
    delay(500);

    digitalWrite(LED1, HIGH);
    delay(500);

    // convert long to bytes
    byte value[3];
    byte value2[3];

    value[0] = (byte) (buffer[1] >> 16);
    value[1] = (byte) (buffer[1] >> 8);
    value[2] = (byte) (buffer[1]);
    value2[0] = (byte) (buffer[3] >> 16);
    value2[1] = (byte) (buffer[3] >> 8);
    value2[2] = (byte) (buffer[3]);

    Serial.println(ADS.getDeviceId(), HEX);
    delay(100);
    Serial.print("ID: ");

    Serial.println(ID, DEC);
    Serial.print("CONF1: ");

    Serial.println(CONF1, HEX);
    Serial.print("CONF2: ");

    Serial.println(CONF2, HEX);
    Serial.print("CONF3: ");

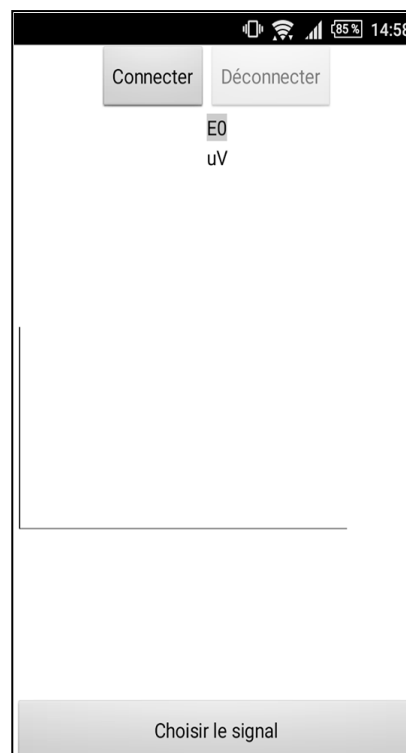
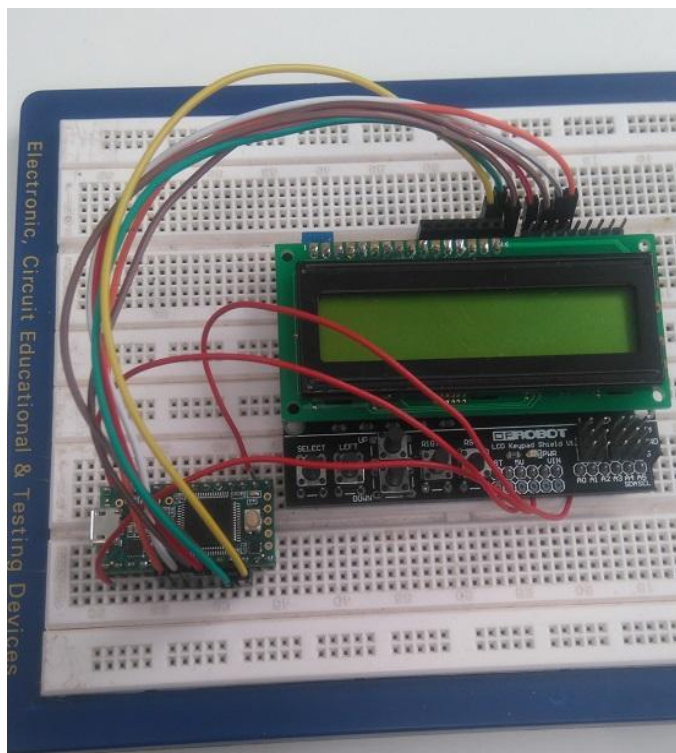
    Serial.println(CONF3, HEX);

    Serial.print("Entree1: ");
    Serial.println(value[0], HEX);
    Serial.print(value[0], HEX);
    //Serial1.write(value[0]);
    Serial.print(value[1], HEX);
    //Serial1.write(value[1]);
    Serial.println(value[2], HEX);
    Serial.print("Entree3: ");
    Serial.println(value2[0], HEX);
    Serial.print(value2[0], HEX);
    Serial.print(value2[1], HEX);
    Serial.println(value2[2], HEX);
}

```

## D. Etudiant E4 :

### I. Présentation de la partie



Ma partie consiste à travailler sur l'Electroencéphalogramme, mais une partie servira tout de même pour l'Electrocardiogramme. Pour le projet je devrais réaliser plusieurs tâches comprenant :

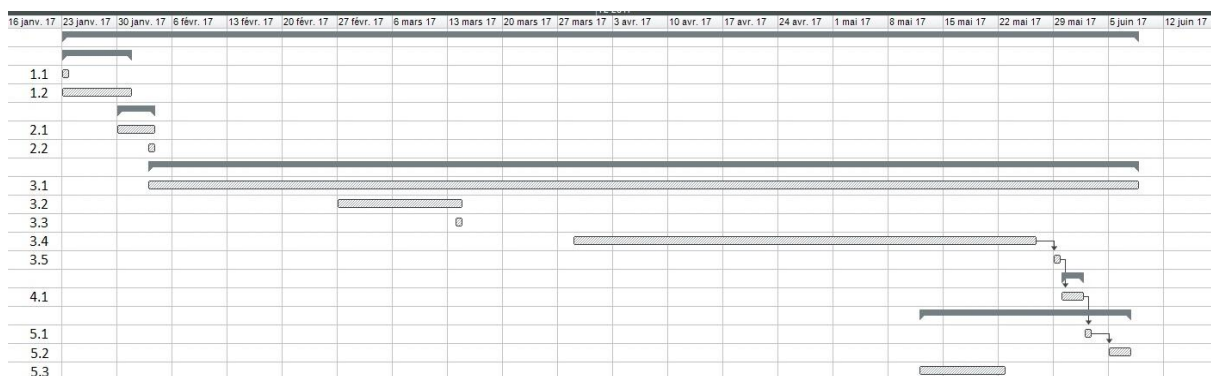
- La conception d'une interface de paramétrage via un Shield LCD et un module Teensy
- L'adaptation d'une carte micro SD permettant l'enregistrement des données
- La conception de l'application Android permettant la réception et la visualisation des signaux EEG et ECG

La partie interface de paramétrage devra être capable de s'intégrer au reste du système, par rapport à la partie code ainsi qu'à la partie physique. C'est-à-dire qu'il faudra utiliser les bonnes broches vis-à-vis du module Teensy, de l'utilisation de la micro SD et de la carte de développement.

Pour l'application Android, elle devra fonctionner pour l'Electroencéphalogramme et l'Electrocardiogramme, étant donné que le principe des systèmes est sensiblement le même, l'application sera facilement adaptable aux deux systèmes.

## II. Diagramme de Gantt

	Nom de tâche	Début	Fin	Prédécesseu...	Ressources
1	ECG/EEG	23/01/2017	08/06/2017		
2	☐ 1. Spécification/Planification	23/01/2017	31/01/2017		
3	1.1 Elaborer un planning des activités pour la durée du projet.	23/01/2017	23/01/2017		Corentin, Jean-Baptiste, Gaetan, Thomas
4	1.2 En collaboration avec les autres étudiants, préparer une prés...	23/01/2017	31/01/2017		Corentin, Jean-Baptiste, Gaetan, Thomas
5	☐ 2. Conception préliminaire	30/01/2017	03/02/2017		
6	2.1 Etudier le système et préparer la conception préliminaire	30/01/2017	03/02/2017		Thomas
7	2.2 Revue de projet n°1 (en groupe)	03/02/2017	03/02/2017		Corentin, Jean-Baptiste, Gaetan, Thomas
8	☐ 3. Conception détaillée	03/02/2017	08/06/2017		
9	3.1 Concevoir l'interface entre la carte de prototypage rapide, l'é...	03/02/2017	08/06/2017		Thomas
10	3.2 Préparation revue de projet n°2	27/02/2017	14/03/2017		Thomas
11	3.3 Revue de projet n°2	14/03/2017	14/03/2017		Thomas
12	3.4 Développer une application Appinventor permettant d'afficher...	29/03/2017	26/05/2017		Thomas
13	3.5 Présenter cette application aux autres étudiants du groupe q...	29/05/2017	29/05/2017	12	Thomas
14	☐ 4. Prototypage	30/05/2017	01/06/2017		
15	4.1 Réaliser le prototype du système complet	30/05/2017	01/06/2017	13	Corentin, Jean-Baptiste, Gaetan, Thomas
16	☐ 5. Qualification-intégration-validation	12/05/2017	07/06/2017		
17	5.1 Rédiger une notice de maintenance.	02/06/2017	02/06/2017	15	Corentin, Jean-Baptiste, Gaetan, Thomas
18	5.2 Faire la mise au point et la recette du prototype final.	05/06/2017	07/06/2017	17	Corentin, Jean-Baptiste, Gaetan, Thomas
19	5.3 Rédiger le rapport	12/05/2017	22/05/2017		Corentin, Jean-Baptiste, Gaetan, Thomas



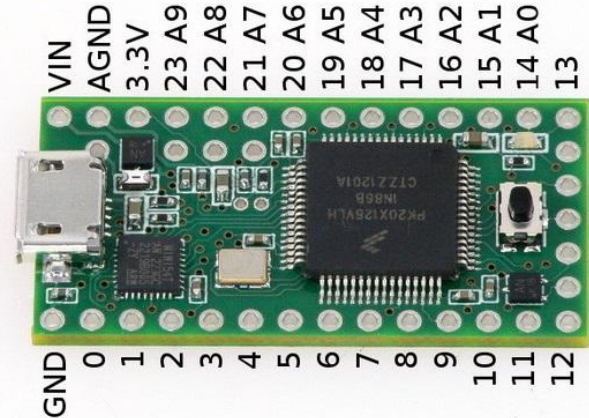
Un diagramme de Gantt a été planifié afin d'estimer le temps que prendra chaque tâche tout au long du projet. Certaines tâches sont réalisées en même temps, car elles n'ont pas besoin de se suivre pour poursuivre le projet. Le diagramme a été changé afin de correspondre aux réalisations réelles, étant donné que certains problèmes peuvent être rencontrés, influençant donc l'avancée du projet. Pour le moment les temps sont respectés mais il est possible que la réalisation soit retardée pour cause techniques.

Les deux tâches principales (interface de paramétrage et application Android) se font tout au long du projet, l'important sera d'avoir fini ces parties pour la fin du projet, c'est-à-dire pour la recette et la notice de maintenance.



### III. Tâches réalisées

#### Recherches préliminaires :



Module Teensy

Pin	Fonction
Analogique 0	(Select, haut, droite, bas et gauche)
Numérique 4	DB4
Numérique 5	DB5
Numérique 6	DB6
Numérique 7	DB7
Digital 8	RS (Sélection de l'affichage de données ou de signaux)
Numérique 9	Permettre
Numérique 10	Contrôle rétro-éclairé

Pour l'interface de paramétrage, on utilisera un Shield LCD, qui est une carte d'extension permettant d'y mettre un écran LCD, et comprenant plusieurs boutons qui sont intégrés.

Pour le Shield LCD, l'écran utilisera différentes broches numériques mais pour les boutons nous n'aurons besoin que d'une broche analogique.

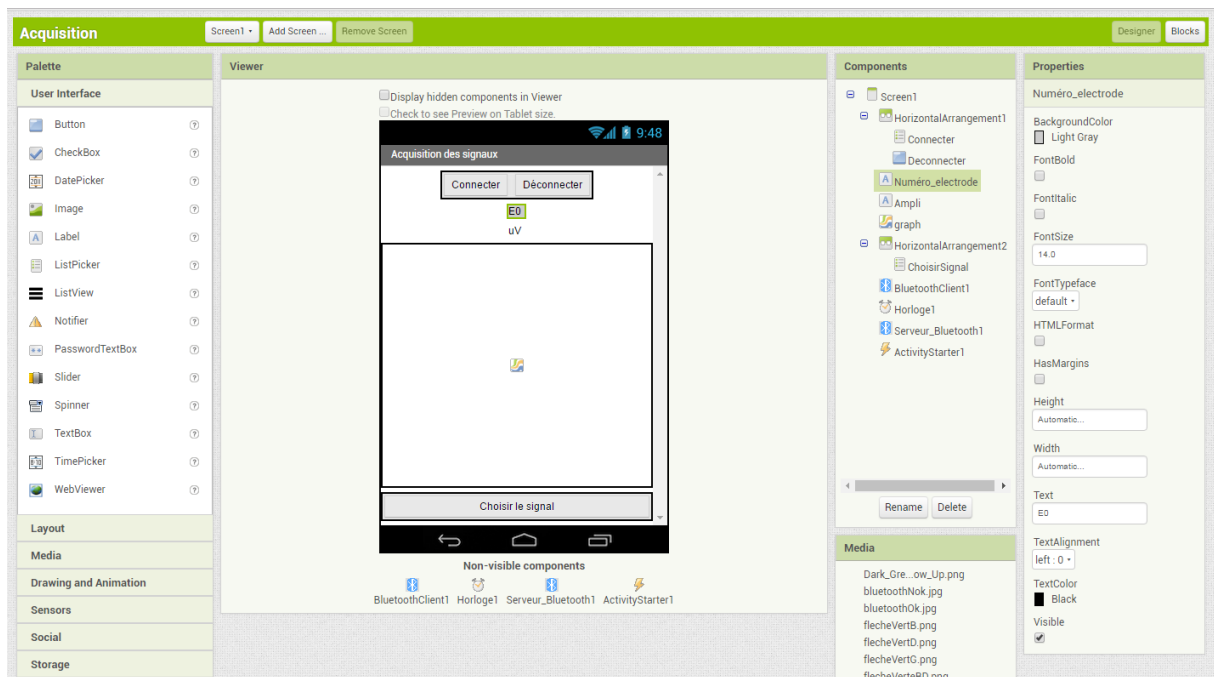
Pour ce montage nous utiliserons un microcontrôleur Teensy, il remplacera l'Arduino, il est plus petit et plus performant et marche tout de même sous



Pour intégrer la carte micro SD au système nous utiliserons tout simplement un Lecteur de carte adaptable à Arduino, permettant de lire et d'écrire sur la carte. Elle fonctionnera selon le code C++ du programme, qui utilisera la librairie SD.

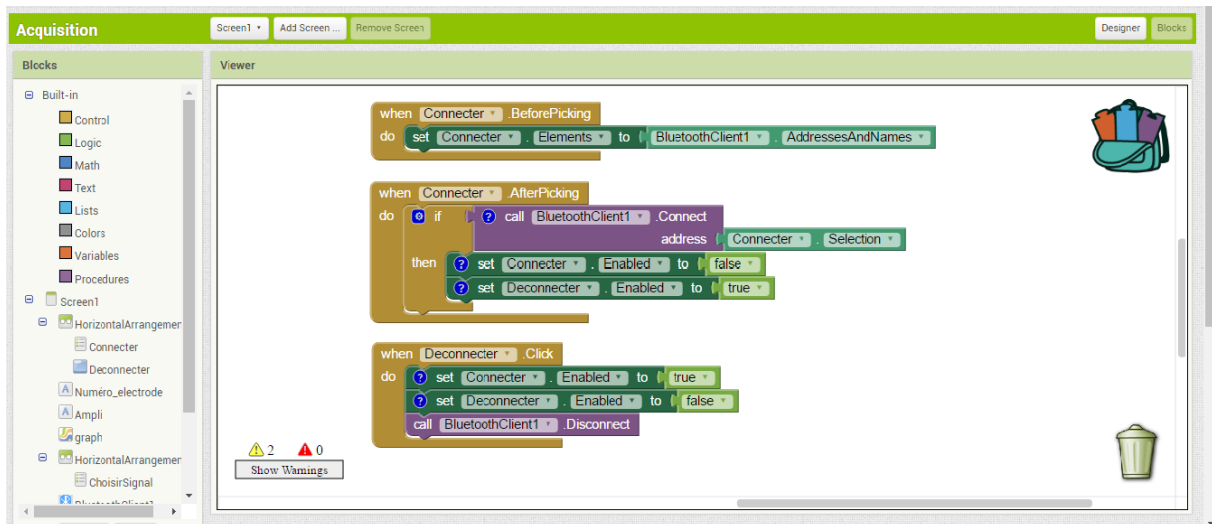
Cette interface va servir à paramétrer la fréquence d'échantillonnage de l'Electroencéphalogramme, ainsi qu'augmenter ou diminuer la durée d'enregistrement sur la carte micro SD. Elle permettra également de déterminer de quel patient proviendra les données, pour connaître son numéro d'identification, la date et l'heure, son nom, son prénom ainsi que son âge.

## Application APP Inventor :

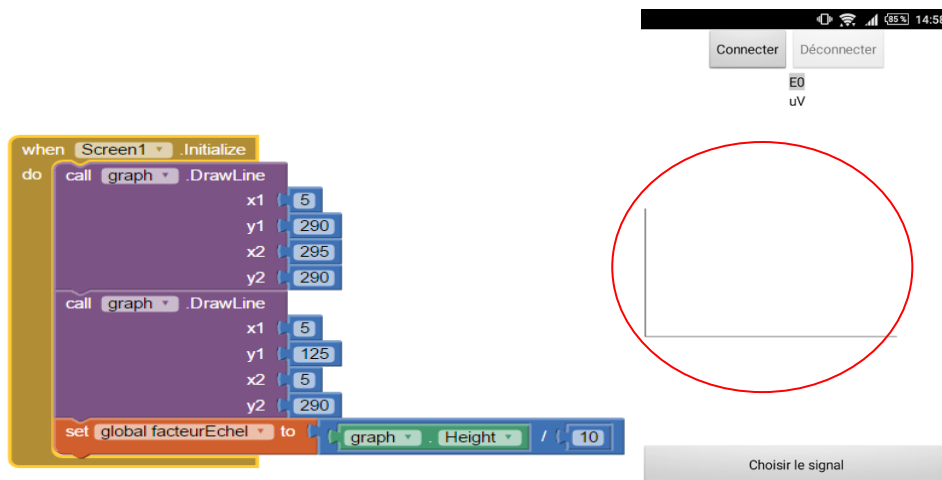


Pour le projet il nous est donc demandé de visualiser les signaux sur un Smartphone Android ou une Tablette Android. Pour réaliser l'application nous utiliserons donc APP Inventor, qui permet de développer assez simplement les applications Android. Dans APP Inventor on retrouve deux parties pour concevoir une application, la partie Designer (qui permet de gérer la partie visuelle de l'app) et la partie Blocks (qui est en réalité la partie code de l'app, qui se crée par l'intermédiaire de blocs d'algorithmes).

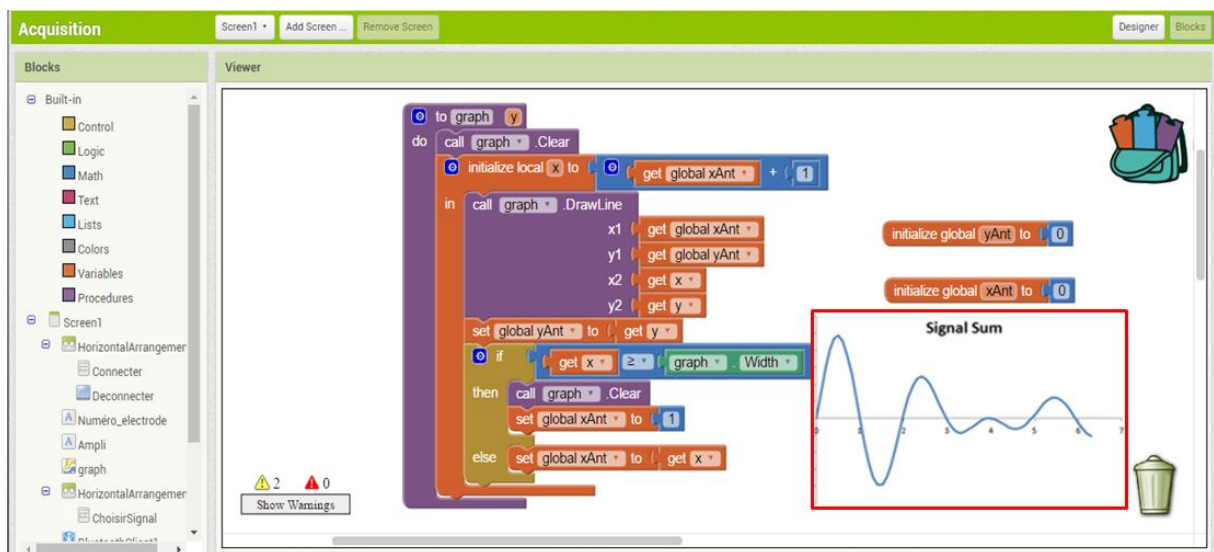
Étant donné la taille de l'écran et la capacité de réalisation nous ne pourrions visualiser qu'un signal à la fois. Pour la partie visuelle j'ai donc mis deux boutons Connecter/Déconnecter afin de se connecter en Bluetooth au module relié à la TI 1298 et 1299. Un afficheur qui affichera le numéro du signal visualisé (numéro de l'électrode), ainsi qu'un deuxième pour afficher la tension du signal. En dessous il y a une « toile » qui permettra d'y dessiner le signal grâce à une fonction réalisée dans la partie Blocks. Il y aura un dernier bouton permettant de choisir quelle électrode visualiser.



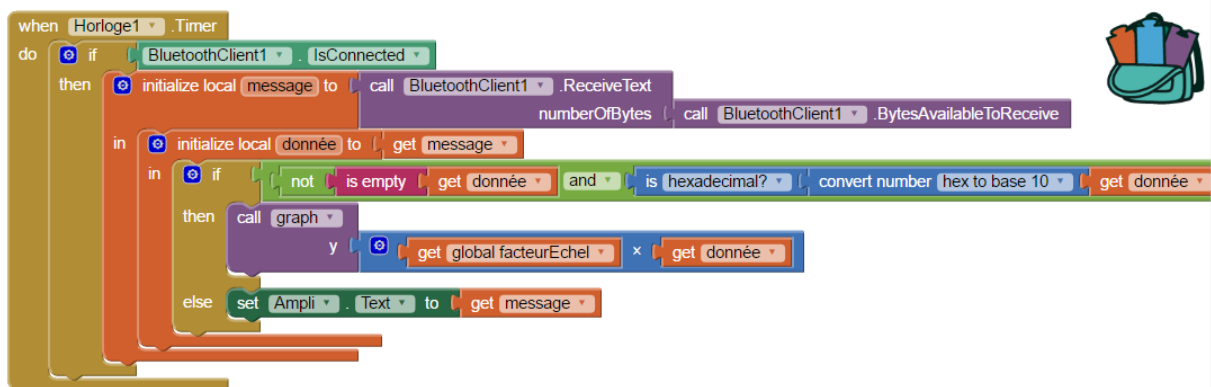
Dans la partie Blocks, nous avons ici la partie qui servira à choisir l'adresse Bluetooth à laquelle se connecter et se déconnecter. Le bouton *Connecter* va ouvrir une liste, ou on retrouvera toutes les adresses Bluetooth répertoriés dans l'appareil. Après le choix de l'appareil auquel se connecter (dans notre cas ce sera un module Bluetooth HC-05), il sera sélectionné et le lien Bluetooth s'effectuera, jusqu'à ce que le bouton *Déconnecter* soit activé.



Cette partie va simplement servir à dessiner deux axes (X et Y) sur la toile, cela servira de repère pour le signal. Cette fonction utilise la position que l'on donne à X et Y, ici on trace un trait entre les deux positions données.



Cette partie servira à dessiner le signal sur la « toile » de la partie visuel. Sur celle-ci nous allons utiliser un graph d'axe X et Y. X représentera le temps et Y la tension du signal. Donc dans le code, X sera incrémenté de 1 toute les secondes, et Y sera égale aux données reçut par le module Bluetooth. Un point suivra X et Y et laissera apparaitre la courbe.



Cette partie reste à travaillée, mais elle va en faite permettre de donner à Y la valeur des données reçut. Le texte qui sera réceptionné va devenir une variable, la fonction va vérifier si le texte est vide, et si ce n'est pas le cas va envoyer le texte (chiffres) sur Y, qui servira à dessiner le signal. Les chiffres reçut seront également multipliés par la variable Facteur Echelle, car la tension sera minime et la taille du graph sera trop grande par rapport à la tension, on va donc « amplifier » la tension afin d'avoir une visualisation plus propre.

Il faut savoir que les données envoyées par le système seront en Hexadécimale, cette partie va donc également convertir les données en Décimale, afin de les lises sur le graphique.

## Pour la suite du projet :

Pour la suite du projet, il me reste à terminer l'application Android, des problèmes ont été rencontrés au niveau de la connexion Bluetooth, qui suite aux tests, viendrait de la réception du nombre de Bits reçut.

Il reste également l'interface de paramétrage à mettre en place. Étant donné que l'application servait pour les deux parties (EEG et ECG), elle devait être faite en priorité. Pour l'écran LCD j'ai récupéré un code permettant d'afficher un menu ainsi que des sous menus, qui devra être modifié afin de convenir à notre utilisation.

### **III. CONCLUSION**

A la conclusion de ce dossier, chacun des élèves estiment que leurs parties seront complétées et finies à temps pour la présentation oral et les démonstrations du projet.