

Année scolaire 2016/2017

**PROJET DE FIN D'ANNÉE IMA4 : REEDUCATION
DE LA POSITION DE LA TÊTE**

INTRODUCTION

Dans le cadre du projet de fin d'année, nous avons choisi de réaliser un dispositif de suivi de la position de la tête. Ce projet a pour but de venir en aide à un kinésithérapeute pour le monitoring d'une position adéquate qui permettrait au patient de diminuer les maux de dos.

Tout au long de ce rendu, nous allons dans un premier temps présenter le cahier des charges afin de pouvoir scinder le projet en plusieurs tâches élémentaires et faire le choix du matériel. Puis, dans un second temps, nous allons expliquer la réalisation de chacune des tâches et tester les parties fonctionnantes avant de discuter finalement des résultats et des difficultés rencontrées.

Sommaire

I Cahier des charges

1 Contexte

2 Besoins

3 Acteurs concernés

4 Analyse fonctionnelle

5 Choix du matériel et méthode de résolution

II Réalisation et résultats

1 Récupération des accélérations

2 Transmission des données au module WIFI

3 Application smartphone

III Discussion des résultats

Conclusion

I. Cahier des charges

Titre abrégé du projet

Rééducation de la position de la tête

I.1 Contexte

L'évolution de la technologie a engendré une modification de la place du salarié dans l'entreprise. Ainsi, nombreux sont ceux qui travaillent dans un bureau face à un ordinateur. Cependant, il a été remarqué que les maux de dos sont en majorité liés à une mauvaise posture au travail.

C'est après différentes études, notamment un bilan postural montrant où se situe la tête par rapport à une position idéale que l'on a remarqué que ces dernières années, 50% des maux de dos sont dus en réalité aux cervicales contre 20 % seulement il y a une dizaine d'années. Du constat précédent, découle ce projet ayant pour but de faciliter aux kinésithérapeutes la rééducation de la position de la tête des patients se plaignants de leur dos.

I.2 Besoins

La rééducation de la posture de la tête chez un kinésithérapeute pourrait être améliorée si le travail pouvait se faire en autonomie.

Le client (ici le kinésithérapeute) souhaite que l'on réalise un système de suivi de la position de la tête d'un patient en rééducation grâce à un monitoring continu de la position de la tête, ainsi qu'un mécanisme pour aider le patient à reprendre une position correcte en lui signalant cette différence. En effet, on lancera le système pendant environ 30 min durant lesquelles nous détecterons dans un premier temps la position idéale que le médecin aura décidé. Ensuite, chaque fois que le patient changera de position, nous le lui indiquerons pour qu'il reprenne sa position idéale.

Ce projet devra ainsi comparer la position actuelle du patient à une position idéale (celle de la verticale) et lui signaler s'il doit ou non s'ajuster.

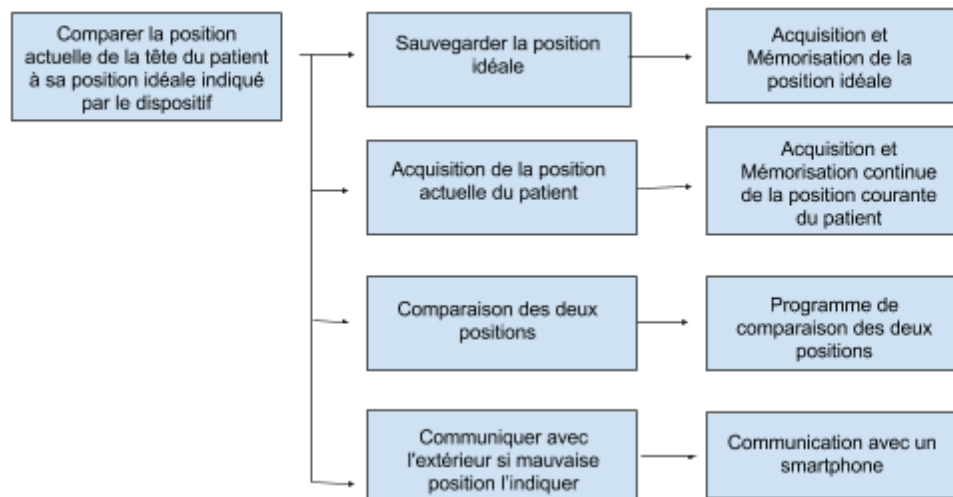
I.3 Acteurs concernés

- > kinésithérapeute (ou le praticien)
- > Le patient
- > le responsable de projet

I.4 Analyse fonctionnelle

Dans ce paragraphe, nous allons poser le problème avec l'aide des certains outils d'analyse fonctionnelle. Nous avons découpé notre fonction principale en plusieurs sous fonctions élémentaires (la colonne toute à droite du schéma ci-dessous) afin d'avoir une approche modulaire pour pouvoir les traiter séparément une par une et de façon indépendante. Voici le schéma.

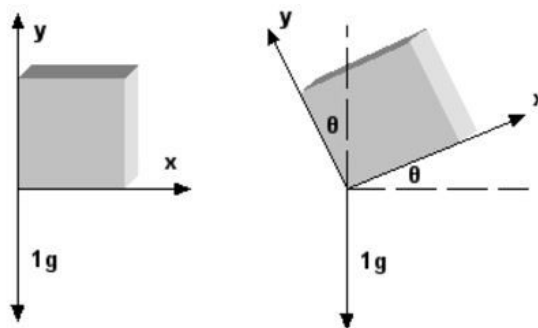
Functional Analysis System Technique



I.5 Choix du matériel et méthodes de résolution

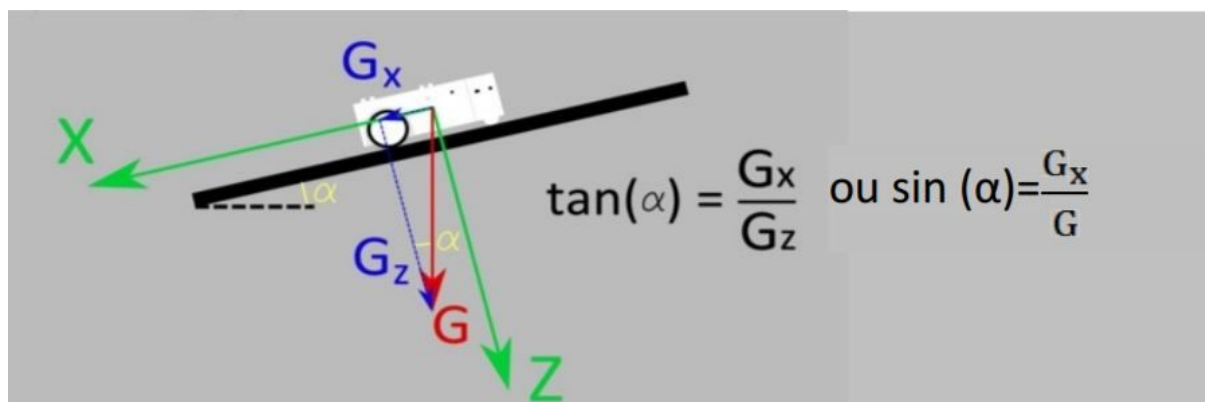
Pour chaque fonction élémentaire, nous allons nous pencher sur le choix du matériel adéquat pour sa réalisation.

Acquisition et mémorisation de la position idéale: le matériel pour cette fonction-ci sera le même que pour la deuxième fonction élémentaire. La mesure de la position idéale se pourrait se définir comme la mesure de l'angle d'inclinaison d'un accéléromètre dû à la force gravitationnelle. en effet, un accéléromètre peut mesurer la force gravitationnelle. Nous allons nous servir de cette mesure pour calculer l'inclinaison de l'accéléromètre selon l'axe x,y et z. L'accéléromètre nous donnera la répartition de la force gravitationnelle (égale à $1g=9.80m/s^2$). Pour calculer l'inclinaison, nous allons utiliser quelques règles de trigonométrie.



calcul des angles:

Partons d'un exemple: Voici un exemple. Prenons une voiture se trouvant sur une route en pente.



Le triangle G, G_x, G_z étant rectangle, nous pouvons donc extraire l'angle α par la formule ci-dessus. Dans notre cas nous avons G_x, G_y et G_z qui sont les valeurs du vecteur gravitationnel \vec{G} sur les trois axes et nous devons déduire α (l'angle que fait G_x avec l'axe x), φ (l'angle que fait G_y avec l'axe y) et ψ (l'angle que fait G_z avec l'axe z). la norme de \vec{G} est

$$|\vec{G}| = \sqrt{(G_x^2 + G_y^2 + G_z^2)}$$

On en déduit donc l'expression des trois angles d'inclinaison

$$\text{roll} = \text{roulis} = \alpha = \arcsin\left(\frac{G_x}{|\vec{G}|}\right)$$

$$\text{pitch} = \text{tangage} = \varphi = \arcsin\left(\frac{G_y}{|\vec{G}|}\right)$$

$$\text{yaw} = \text{verticalité} = \psi = \arccos\left(\frac{G_z}{|\vec{G}|}\right)$$

Dans un premier temps, nous avons opté pour des accéléromètres analogiques adxl335 qui seraient connectés à un arduino Mega pour la récupération et le traitement des données. Cependant, cette référence là n'était plus en stock et nous avons dû utiliser des accéléromètres digitaux adxl345. Par contre ceux-ci fonctionnent en 3.3V alors que les pins du Mega sont à 5V. Nous avons dû faire une carte destinée à réguler la tension pour pouvoir réguler les différences de tension. Nous avons par la suite remarqué que celle-ci ne fonctionnait pas, Nous avons donc fini par remplacer le Mega pour un arduino mini ayant toutes ses sorties à 3.3V. Nous avons eu également à disposition un accéléromètre 10DOF qui permet d'être plus précis. Celui-ci sera placé sur la tête pour détecter les moindres mouvements tandis que les autres accéléromètres seront placés le long de la colonne vertébrale.

Programme de comparaison des deux positions:

Ce programme sera fait sur le smartphone avec les données reçues à partir du capteur. Nous ne l'avons pas fait sur l'arduino pro mini car la puissance de calcul de celui-ci est trop faible pour pouvoir faire les calculs des angles avec précision. Il consiste à calculer dans un premier temps les angles correspondants à la position idéale, puis, dans un second temps, de mesurer continuellement les angles de la position actuelle du patient durant le monitoring et, si ces angles là ont subi une grande variation par rapport à l'angle idéal correspondant, une notification sera émise.

Communication avec un smartphone:

Pour l'envoi des données du capteur de l'arduino pro mini au smartphone, nous avons le choix entre un module WIFI ESP8266 et un module bluetooth. Sachant que le réseau WIFI peut fonctionner sur une distance plus grande que celle du bluetooth et pour ne pas limiter la distance entre le capteur et le smartphone, nous avons opté pour un module WIFI ESP8266. Ainsi, la distance ne sera pas un facteur contraignant. Par contre, l'utilisation du WIFI cause la décharge rapide de la batterie du smartphone. L'ESP8266 a également besoin de beaucoup de courant pour fonctionner (jusqu'à 250mA). Ceci peut nous amener à augmenter la tension d'alimentation si le besoin se présente.

Traitement et acquisition de donnée:

Au début, nous nous étions posé la question "pourquoi mettre un MEGA plutôt qu'une raspberry Pi". A ce moment là, nous voulions travailler avec les accéléromètres analogiques. La Raspberry ayant une bonne puissance de calcul, n'a cependant pas de pin analogique. Nous étions donc obligées par la nature du matériel à choisir un Mega. Avec 15 sorties analogiques, nous pouvions connecter jusqu'à cinq accéléromètres (5 accéléromètres * 3 (sorties Gx, Gy et Gz)).

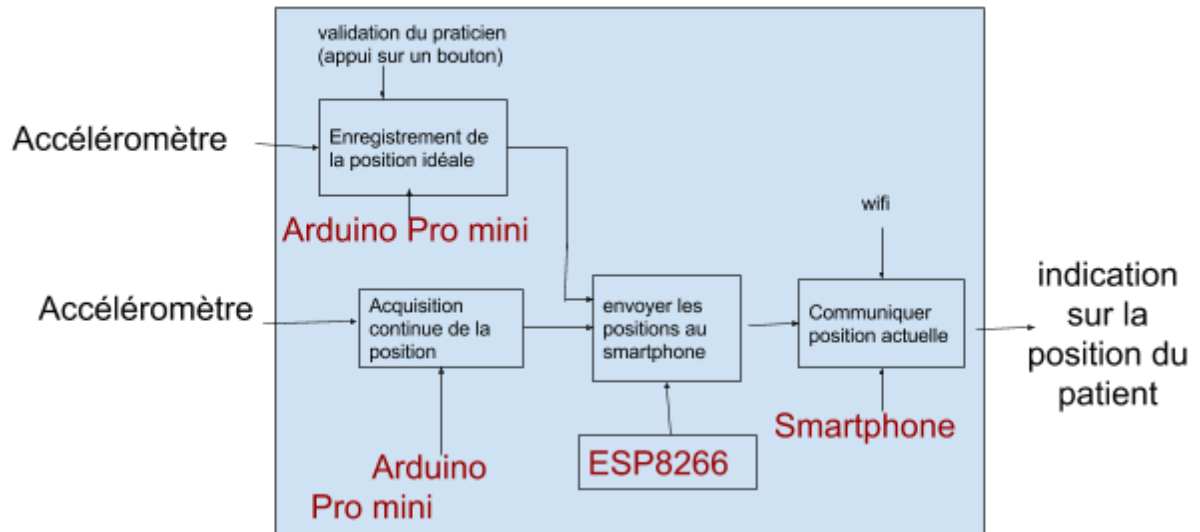
Pour récapituler, nous avons décidé d'utiliser:

- des accéléromètres pour mesurer la force gravitationnelle g
- un arduino pro mini 3.3V
- un module WIFI ESP8266
- un smartphone
- des fils pour connecter ces derniers

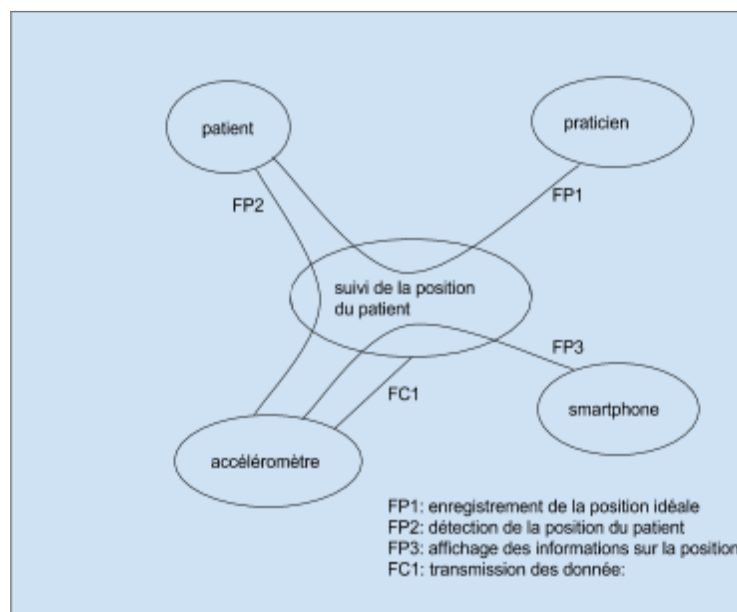
Voici un schéma illustratif avec le matériel utilisé en rouge pour chaque tâche.

structured analysis design technique

dispositif de rééducation



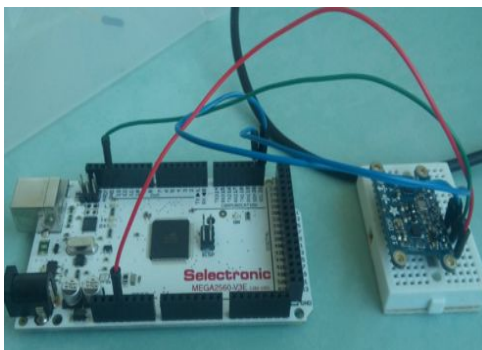
Nous allons à présent donner les tâches élémentaires qu'il faut réaliser. Le schéma suivant est assez illustratif et des décrit les fonctions primaires (FP) qui décrit les tâches élémentaires et les fonctions critiques (FC) qui sont les fonctions à réaliser pour relier les fonctions primaires entre elles.



II. Réalisation et résultats

II.1 Récupération des accélérations

Comme nous avons dit précédemment, nous avons commencé l'acquisition des données avec un arduino Mega. Nous avons commencé avec l'accéléromètre 10DOF et avons réussi à récupérer les données lues à partir de ce dernier.

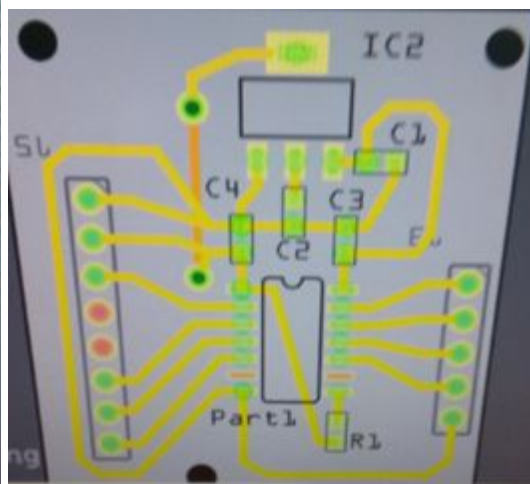
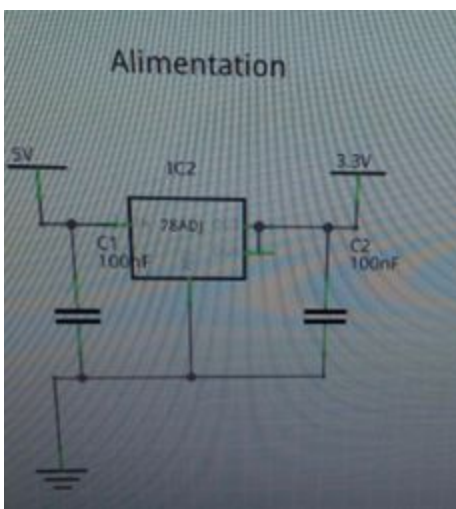
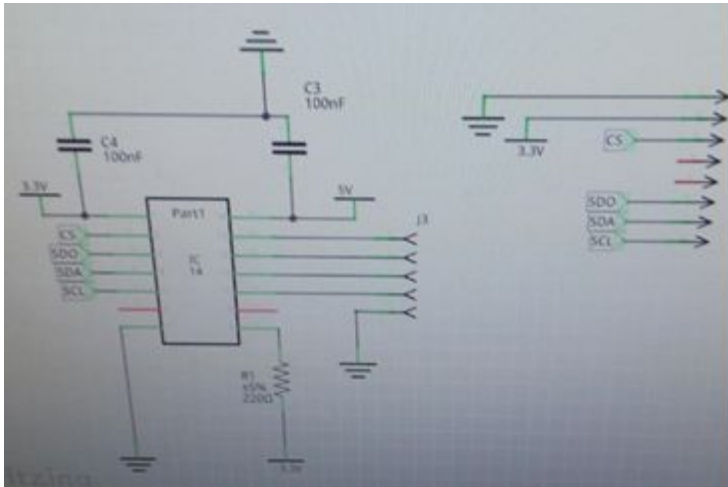


```
AX: -24 AY: 144 AZ: 988  
total: 120 Angle x: -2147483648 Angle y: 0 Angle z: 0  
AX: -20 AY: 140 AZ: 984  
total: 72 Angle x: -2147483648 Angle y: 0 Angle z: 0  
AX: -20 AY: 140 AZ: 988  
total: 114 Angle x: -2147483648 Angle y: 0 Angle z: 0  
...
```

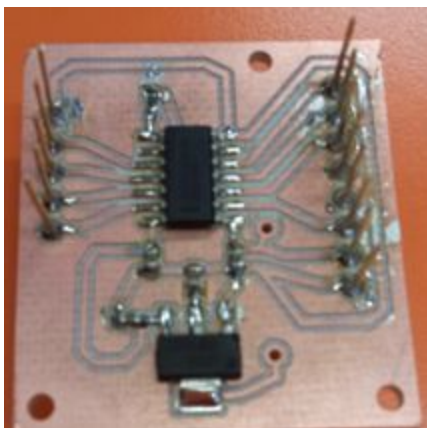
Lorsque nous effectuons les calculs d'angle à partir du MEGA, nous avons pu observer que les valeurs calculées étaient différentes de ceux que l'on calculait avec la calculatrice. Cela est due au fait que la puissance de calcul du MEGA n'est pas assez grande. Nous avons donc décidé de réaliser les calculs sur le smartphone.

Seulement, une fois les accéléromètres reçus, nous nous sommes rendues compte que ces derniers étaient des 3.3V digitaux alors que le MEGA envoyait 5v en sortie des pins digitaux du MEGA. Nous avons donc essayé de réaliser une carte électronique permettant de fournir une tension de 3.3V en sortie lorsqu'elle en reçoit une de 5V.

Nous avons donc réalisé la carte grâce au logiciel fritzing.



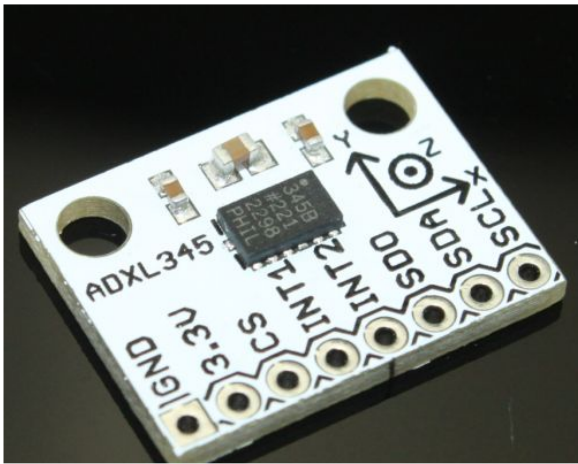
Nous avons ensuite fait graver la carte et avons soudé chaque composant.



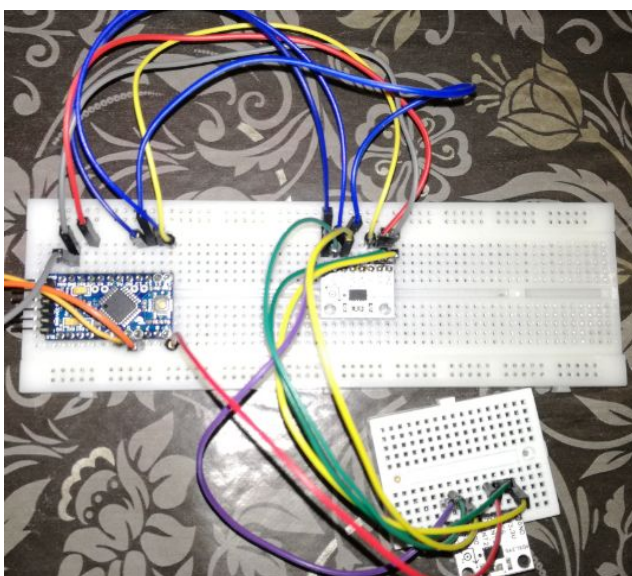
Nous avons testé la carte en alimentant les pins d'entrée par des 5v et avons observé la sortie du circuit. Malheureusement, seulement un pin en sortie était en mesure de fournir une tension de 3.3V.

Malgré les nombreuses tentatives, nous n'avons pas pu réussir à avoir 3.3V sur toutes les sorties. Nous avons donc changé de microcontrôleur et pris un arduino mini qui était plus adapté car ses sorties sont de 3.3v. Le premier montage avec un accéléromètre a été un succès et nous avons réussi à récupérer les données à partir de ce dernier.

Il a été plus difficile de rajouter un second accéléromètre car nous devons tenir compte du fait que lorsque l'on lit les valeurs de ce derniers grâce à la fonction `adxl.readAccel()` procuré par la librairie `<SparkFun_ADXL345.h>`, on lit les valeurs reçu par les pins SDA et SCL.



Nous devons donc considérer ces pins comme étant liés à des bus de données. Il est donc impossible de lire les valeurs simultanément, nous déclarons donc le second accéléromètre comme étant une entité différente du premier en prenant soin de relier son chip select (CS) à un pin disponible et différent de celui du premier accéléromètre. Une fois cela fait, on peut ajouter d'autres accéléromètres beaucoup plus facilement.



```

accelero 1
4
31
258
accelero 2
18
28
236
accelero 1
0
1
10
accelero 2
0
1
9

```

Les valeurs du haut sont celle lues sans aucun traitement du mini et celle du bas sont celles ramenées à 1g (9.8). En effet, les valeurs de l'accélération de +/-2g sont codées sur 10 bits (-512 et 512). Sachant que nous relevons l'accélération gravitationnelle, les valeurs recueillies seront toujours entre +/-1g donc les valeurs du registre de données seront dans la plage [-256,256]. Nous avons également arrondi la valeur à 10 pour pouvoir envoyer des valeurs entières. Une fois que cela est fait, transformer les valeurs du registre de sortie en une valeur significative de force de gravitation devient facile.

II.2 Transmission des données au module WIFI

Nous avons également implémenté le module Wifi de manière à ce que l'on puisse envoyer les données des accélérations reçus par l'arduino mini.

Nous avons dans un premier temps réalisé la liaison série entre l'arduino et l'esp. Pour cela, nous avons utilisée la bibliothèque SoftwareSerial d'arduino et avons défini deux liaisons série. Une du côté du mini et l'autre implémenté dans l'esp pour permettre ainsi d'assurer la communication entre les deux. Le mini commençait par envoyer un "a" pour indiquer que l'on commence une nouvelle transmission, ensuite on ramène les valeurs de l'accéléromètre sur l'échelle de g (gravitation terrestre). Pour prendre en compte les chiffre négatif bien que le softwareserial ne transmet pas de nombre signé on ajoute 25 à chaque valeur lue négative. Pour finir, nous envoyons les données grâce à un serial write qui écrit sur les pins de l'esp (pin 14 RX et 12 TX).

Du côté de l'esp, on définit une liaison du nom de swSer au même baudrate que le précédent. On attend la réception de 72 (l'équivalent de 'a' en integer) indiquant que l'on commence à transmettre des données de l'accéléromètre. Une fois ce nombre reçu on incrémente un compteur à chaque réception d'un chiffre et à partir de la valeur de ce compteur nous plaçons la donnée dans la variable qui la définit (la première donnée est celle de x1 donc si compteur est à 1 on place la valeur lue par swSer.read dans la variable x1)

```
x1 : 0  
y1 : 0  
z1 : 9  
x2 : 0  
y2 : 0  
z2 : 8
```

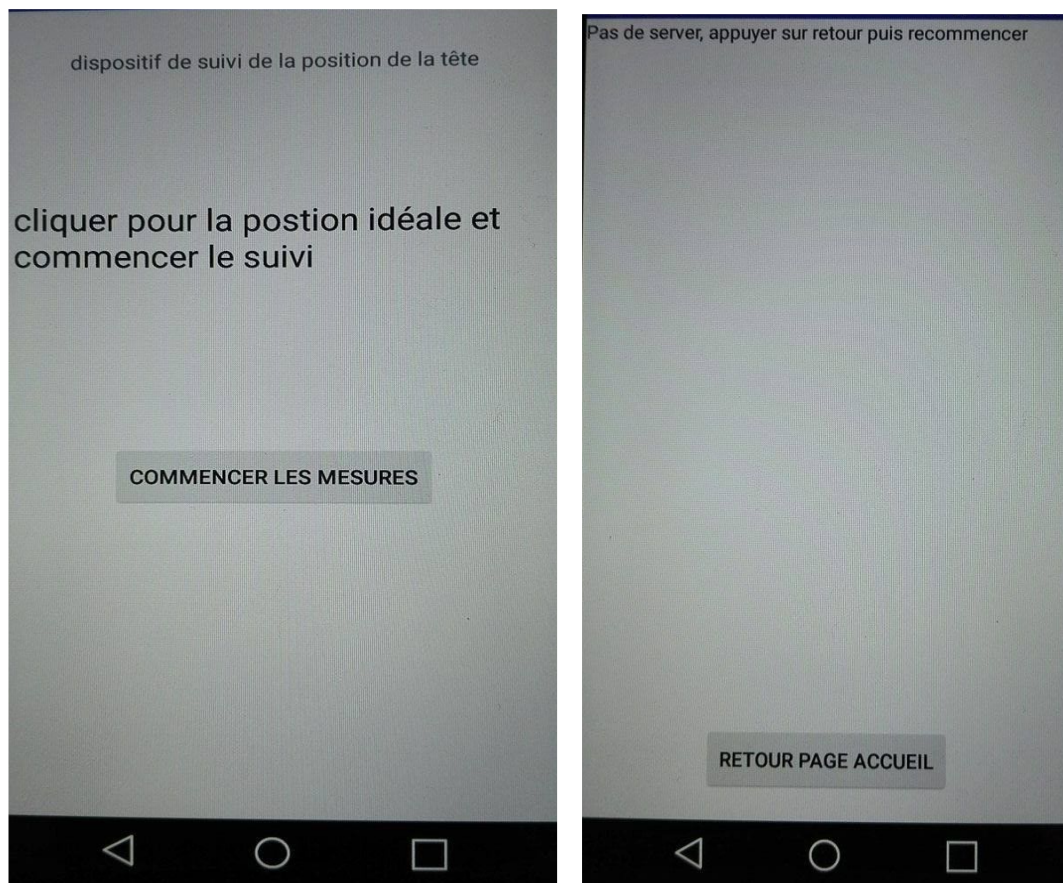
L'esp doit également pouvoir communiquer avec le smartphone pour l'envoi des données de l'accéléromètre. Pour ce faire, nous avons implémenté des sockets dans l'esp et dans l'application smartphone. Ces sockets vont réaliser des transmissions de paquets UDP de part et d'autre. L'ESP sera le serveur qui sera toujours connecté au réseau WIFI et, une fois que l'application smartphone veut récupérer les données, il envoie un message broadcast à toutes les stations connectées au réseau. L'ESP va par la suite répondre et commencer à envoyer les données au smartphone. Celui-ci les récupère, exécute l'algorithme de calcul des angles et affiche, pour chaque accéléromètre, l'angle idéal qui correspond à la position idéale décidée par le médecin et l'angle actuel qui traduit la position actuelle du patient.

L'union de ces deux parties n'est pas un succès dans le sens où la connexion n'arrive pas à aboutir et nous ne parvenons pas à tester les deux parties mises en commun. En effet, l'esp reste bloqué dans la boucle `while(WiFi.status() != WL_CONNECTED)` et ne parvient pas à dépasser cette étape.

II.3 Application smartphone

L'application a été implémentée pour des smartphones sous Android avec l'aide du logiciel android studio. Nous avons décidé de faire deux pages. Une servant de page d'accueil avec le but de l'application et une autre pour l'affichage des angles. La page d'accueil contient un bouton que le kinésithérapeute devra appuyer après avoir positionné le patient à la position idéale. Une fois que ce bouton sera appuyé, la transmission commencera. Le smartphone va dans un premier chercher l'ESP à l'aide d'un message broadcast qu'il va envoyer. S'il n'y a pas de connexion, l'utilisateur en sera informé par l'apparition d'un message lui disant de vérifier la connexion. Si également le paquet n'a pas

été bien envoyé, l'utilisateur en sera informé et il n'aura qu'à retourner à la page d'accueil et réappuyer sur le bouton de démarrage.



Le programme de connexion au server et de récupération de données est réalisé par un thread qui opère en background. Ceci lui permettra d'afficher les données en temps réel sur l'écran, permettant ainsi à l'utilisateur de pouvoir se corriger sur le champ. Voici une photo de l'affichage (la partie transmission de donnée n'étant pas achevée, nous avons donné nous même des valeurs d'accélération dans l'esp pour qu'il l'envoie au smartphone).

```
angle idéal sur x accel1 90.0
angle idéal sur y accel1 0.0
angle idéal sur z accel1 90.0
angle idéal sur x accel2 0.0
angle idéal sur y accel2 90.0
angle idéal sur z accel2 90.0
angle sur x accel1 0.0
angle sur y accel1 -90.0
angle sur z accel1 90.0
angle sur x accel2 0.0
angle sur y accel2 0.0
angle sur z accel2 180.0
```

RETOUR PAGE ACCUEIL

III. Discussion des Résultats

Après avoir implémenté et testé, nous nous sommes demandées si les résultats correspondaient vraiment aux spécifications faites dans le cahier des charges.

Commençons par les accéléromètres: leur sensibilité et leur précision à première vue permettent de suivre les mouvements du patients et de relever sa position. Cependant, il faudrait trouver un dispositif qui permettent de les rattacher au patient de sorte à pouvoir être encore plus précis et de détecter plus précisément les mouvements du patient. L'arduino mini utilisé a été une solution au fait que les accéléromètres ne supportait que 3.3V. Cependant, il nous contraint à faire les calcul sur le smartphone. Mais ceci ne ralentit en rien la transmission des données puisque ça faisait partie de programme établie. Pour pouvoir faire les calculs de l'angle au niveau de la partie électronique, nous aurions pu remplacer l'arduino par une raspberry.

En ce qui concerne l'algorithme de transmission de données entre module WIFI et le smartphone, l'utilisation des paquets UDP pourrait être contesté dans la mesure où Udp ne s'assure pas que les paquets ont bien été transmis (nous ne vérifions pas si la trame a des erreurs ou si le paquet est bien reçu). Il aurait fallu utiliser des sockets TCP pour cela. Nous nous sommes dit que dans notre cas, l'affichage des données doit se faire en temps réel,

nous avons donc décidé d'implémenter des sockets UDP et de gérer nous même l'éventuel manque d'envoi des donnée par le biais d'un time-out sur le smartphone.

Cependant, nous n'avons pas réussi la transmission des données entre l'arduino pro mini et l'esp. Pour faire l'envoi via un softwareSerial, l'entier doit être converti en octet non signé, donc nous ne pouvons envoyer ni de valeurs négatives ni de valeurs qui dépassent 255. La valeur du registre qui contient l'accélération est codée sur 16 bits. Donc, une fois cette valeur obtenue , nous la séparons en 2 octets et nous les envoyons. A la réception, l'esp doit les reconstruire. Lorsque nous exécutons, nous voyons que la valeur obtenue après reconstruction n'est pas la même que celle envoyée.

Au final , nous pensons que nous devrions nous pencher plus sur la transmission des données entre l'arduino et l'esp. Et au pire, si nous ne trouvons pas de solution, utiliser un raspberry Pi qui peut supporter une connexion wifi et, par conséquent, l'utiliser pour pour l'envoi des données au smartphone. Mais, avec l'utilisation d'une raspberry, nous ne pourrons pas utiliser l'accéléromètre 10DOF.

CONCLUSION

Ce projet nous a permis de pouvoir nous familiariser avec les différentes phases de conception d'application, en partant de la spécification des besoins jusqu'au test et à la vérification du projet. La division du projet en tâches élémentaires nous a aidé à nous focaliser sur la résolution d'un problème à la fois. Ainsi, la réalisation de notre projet, qui comporte des tâches et des composants assez variés, nous a permis de nous enrichir encore plus dans le domaine de l'électronique et de l'informatique grâce aux fiches techniques.

Cependant, des améliorations peuvent encore être faites dans la vue d'une personnalisation du projet à des cas spécifiques (par exemple implémenter directement l'adresse ip de l'esp si on connaît le réseau et qu'on sait qu'on n'utilisera que celui-ci. Comme ça nous n'aurons pas envoyer un message broadcast) ou tout simplement pour apporter des solutions aux problèmes qui persistent.