

Filière Systèmes industriels

Orientation Infotronics

Diplôme 2015

Daniel Sgroi

Diagnostic de l'algodystrophie

- *Professeur*
Dr Pierre Roduit
- *Expert*
Dr François Luthi
- *Date de la remise du rapport*
10.07.2015

Ce rapport est l'original remis par l'étudiant.
Il n'a pas été corrigé et peut donc contenir des inexactitudes ou des erreurs.

<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / Studienjahr 2014/15	No TD / Nr. DA it/2015/25
Mandant / Auftraggeber <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input checked="" type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i> SUVA	Etudiant / Student Daniel Sgroi <hr/> Professeur / Dozent Pierre Roduit	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja ¹ <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) Dr François Luthi SUVA Av. Grand-Champsec 90 1950 Sion	

Titre / Titel

Diagnostic de l'algodystrophie

Description / Beschreibung

Dans le cadre d'un projet en lien avec l'IRR (Institut de recherche en réadaptation de la SUVA), l'institut Systèmes Industriels est en train de développer des gants bardés de capteurs reliés à un smartphone pour aider au diagnostic de l'algodystrophie. Lors de son projet de semestre, l'étudiant a pu tester différents capteurs qui permettraient d'améliorer le système d'aide au diagnostic développé.

L'objectif du travail de diplôme sera d'ajouter les capteurs choisis durant le projet de semestre et de réaliser un prototype complet (mesure des deux mains, deux gants, smartphone) du système de diagnostic.

Objectifs / Ziele

- Modification de l'électronique de mesure pour incorporer les nouveaux capteurs
- Modification de la mécanique (connectique et boîtier) du capteur
- Implémentation de l'application sur smartphone pour la collecte et l'affichage des mesures.

Signature ou visa / Unterschrift oder Visum

Responsable de l'orientation
Leiter der Vertiefungsrichtung:¹ Etudiant / Student :


Délais / Termine

Attribution du thème / Ausgabe des Auftrags:
11.05.2015Remise du rapport / Abgabe des Schlussberichts:
10.07.2015, 12:00Expositions / Ausstellungen der Diplomarbeiten:
26 – 28.08.2015Défense orale / Mündliche Verfechtung:
Semaine | Woche 36

¹ Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.

Diagnostic de l'algodystrophie

Diplômant Daniel Sgroi

Objectif du projet

L'objectif a été de développer une paire de gants aidant à opérer un diagnostic précoce de l'algodystrophie. Ce système permet de détecter des signes de la maladie qui ne sont pas forcément discernables par un examen médical conventionnel.

Méthodes | Expériences | Résultats

Pour atteindre ce but, différentes recherches sur les capteurs pouvant être utilisés ont été faites. Ensuite chaque principe de mesure a été testé pour confirmer son efficacité. L'étape suivante a été d'embarquer les capteurs validés dans une électronique d'acquisition. Ensuite, les senseurs ont dû être mis en place sur les gants, ce qui a requis une collaboration avec une couturière et des mécaniciens. Une application smartphone assurant l'interface avec le patient et le médecin a finalement été développée. Elle se connecte sans fil afin de recueillir et afficher les mesures.

Une solution pour mesurer l'ampleur du gonflement des tissus a aussi été étudiée et validée, mais n'a pas encore été testée sous sa forme définitive.

Au final le prototype est fonctionnel pour un gant. Ainsi, le système permet de connaître :

- la température émanant de la main au dixième de degré près,
- le pourcentage d'humidité de la main, indiquant le taux de sudation,
- les éventuels tremblements des membres provoqués par la maladie,
- la rougeur de la peau en faisant des photos ayant des réglages fixes.

L'application permet la visualisation de toutes les informations concernant ces différentes mesures.



Travail de diplôme
 | édition 2015 |

Filière
 Système industriels

Domaine d'application
 Infotonics

Professeur responsable
 Pierre Roduit
 pierre.roduit@hevs.ch

Partenaire
 SUVA



Boîtier et électronique d'acquisition



Affichage graphique dans l'application smartphone

Table des matières

1	INTRODUCTION.....	2
1.1	CAHIER DES CHARGES	2
2	ETUDE DE LA PROBLÉMATIQUE	3
2.1	L'ALGODYSTROPHIE.....	3
3	RÉSULTATS DE LA SÉLECTION DES CAPTEURS	4
3.1	CAPTEUR DE FLEXION.....	4
3.2	CAPTEUR D'ÉTIREMENT	5
3.3	PHOTOGRAPHIES	6
4	VALIDATION ET IMPLÉMENTATION DES CAPTEURS.....	7
4.1	CAPTEUR D'ÉTIREMENT	7
4.1.1	<i>Électronique</i>	7
4.1.2	<i>Validation</i>	7
4.1.3	<i>Implémentation</i>	9
4.2	PHOTOGRAPHIES DES MAINS	10
4.2.1	<i>Support mécanique</i>	10
4.2.2	<i>Réglages optiques</i>	10
5	DÉVELOPPEMENT DE L'APPLICATION	11
5.1	FONCTIONNEMENT GLOBAL	11
6	RÉSULTATS.....	13
6.1	VUES GRAPHIQUES DE L'APPLICATION	13
6.1.1	<i>Humidité</i>	13
6.1.2	<i>Température</i>	14
6.1.3	<i>Étirement</i>	15
6.1.4	<i>Mouvements</i>	15
6.2	PHOTOGRAPHIES	17
7	SYNTHÈSE	19
7.1	TRAVAIL RÉALISÉ.....	19
7.2	PERSPECTIVES	20
7.2.1	<i>Travail restant</i>	20
7.2.2	<i>Amélioration possibles</i>	20
7.3	CONCLUSION	21
7.4	ANNEXES	21
7.5	ANNEXES NUMÉRIQUES.....	21
7.6	REMERCIEMENTS.....	22

1 INTRODUCTION

Ce document décrit le déroulement de ce travail de diplôme, s'inscrivant dans la formation de HES-SO dans la filière *Infotronics*.

Dans le cadre d'un projet en lien avec la Clinique Romande de Réadaptation (Suva care), l'institut Systèmes Industriels est en train de développer des gants bardés de capteurs reliés en Bluetooth à un Smartphone pour aider au diagnostic de l'algodystrophie des mains. Ces gants sont déjà équipés de capteurs de température et d'humidité, ainsi que de plusieurs accéléromètres permettant de capter les tremblements des doigts et de la main malade.

L'objectif de ce projet de Bachelor est mettre en place différents autres principes de mesures afin d'accroître le potentiel de ces gants de diagnostic. Cela conduit à la réalisation d'un prototype complet du système comprenant une paire de gants, une électronique d'acquisition connectée à l'application smartphone, et le développement de cette dernière, permettant de visualiser et exploiter les données intéressantes.

Cette tâche se base sur un travail de recherche effectué précédemment dans le semestre. La lecture du document relatif à cette investigation peut permettre d'avoir une meilleure vue d'ensemble de ce projet de diplôme.

Après un rapide survol de la problématique de l'algodystrophie, le présent rapport se découpe en deux grandes parties. La première décrit les deux principes de mesure ajoutés à la version initiale du gant, soit une mesure du périmètre des doigts et une possibilité d'effectuer des photographies des mains. Ces mesures permettent d'évaluer indirectement la présence et l'importance d'un œdème, et respectivement la variation de couleur de la peau engendrée par la maladie. La dernière partie de ce rapport synthétise les résultats obtenus et présente différentes mesures effectuées avec ce système.

1.1 CAHIER DES CHARGES

Dans un souci de structurer le travail de ce projet, des objectifs ont été fixés. Voici les détails du cahier des charges initialement reçu :

- Modification de l'électronique de mesure pour incorporer les nouveaux capteurs
- Modification de la mécanique (connectique et boîtier) du capteur
- Implémentation de l'application sur smartphone pour la collection et l'affichage des mesures.

2 ETUDE DE LA PROBLÉMATIQUE

2.1 L'ALGODYSTROPHIE

L'algodystrophie, aussi désignée par syndrome régional douloureux complexe (SRDC), est une pathologie affectant une articulation en particulier, caractérisée par des douleurs exacerbées au niveau du membre concerné. Les causes peuvent être multiples, allant de l'intervention chirurgicale, aux fractures, luxations, arthrite et autre affection rhumatologique, en passant même par la grossesse (SRDC de la hanche). Le diagnostic de ce syndrome s'avère être compliqué à poser, car il ne peut être vraiment confirmé que par une scintigraphie, examen relativement conséquent. En effet, la rareté des cas d'algodystrophie et l'inefficacité des examens conventionnels (analyses de sang, radiographie,...) ralentissent la pose d'un diagnostic sérieux.

Dans le cadre de ce projet nous allons nous pencher sur l'affection du poignet et de la main, qui est la forme d'algodystrophie la plus courante. La Figure 1¹ montre une photo d'un cas touchant la main droite pendant une phase dite "chaude" de la maladie (gonflement, rougeur, etc.)



Figure 1 : Main droite atteinte d'un SRDC

Le but étant de développer un moyen efficace pour poser un diagnostic précoce à cette maladie, le projet se base sur un gant effectuant plusieurs types de mesures différentes. Le Tableau 1 présente les différents symptômes caractéristiques de l'algodystrophie.

Symptôme	Mesure associée	Faisabilité
Œdème	Mesure de volume, dilatation de la main	Capteur de pression + ballon
		Capteur d'étirement
Rougeur	Mesure colorimétrique de la peau	Capteurs de couleur tels que : TCS34725 HDJD-S822
	Photographie	Appareil photo du Smartphone
Chaleur	Mesure de température	Capteur de température
Hyperhidrose	Mesure d'humidité	Capteur d'humidité
Tremblements	Mesure de mouvements	Accéléromètres
Impotence fonctionnelle	Mesure de l'amplitude des mouvements	Capteur de flexion
Perte de force	Mesure de force	-
Allodynie ²	-	-

Tableau 1 : Présentation des symptômes

¹ Source : <http://algoneurodystrophiecestquoi.e-monsite.com/album/algodystrophie/>, le 30.06.2015

² « L'allodynie est la survenue d'une douleur déclenchée par un stimulus qui est normalement indolore. » <http://fr.wikipedia.org/wiki/Allodynie> (01.05.2015)

3 RÉSULTATS DE LA SÉLECTION DES CAPTEURS

Dans le cadre du projet de semestre effectué plus tôt dans l'année, plusieurs principes de mesures ont été explorés. Le Tableau 2 expose une synthèse des recherches et les résultats obtenus.

Principe de mesure		Validité du principe	Conditions de validité	Avantages	Inconvénients
Volume de la main	Capteur de pression	Valide	<ul style="list-style-type: none"> Bonne adaptation du coussin d'air au gant Gestion des actions extérieures sur le coussin d'air 	<ul style="list-style-type: none"> Bonne précision du capteur 	<ul style="list-style-type: none"> Perturbé si action extérieure sur le coussin Système encombrant
	Capteur d'étirement	Valide	<ul style="list-style-type: none"> Confirmation par des tests ultérieurs 	<ul style="list-style-type: none"> Adaptation facile 	<ul style="list-style-type: none"> Linéarité entre les deux mains pas garantie
	Capteur de force	Non valide	<ul style="list-style-type: none"> Trouver une méthode pour l'adapter 	<ul style="list-style-type: none"> Non déterminés 	<ul style="list-style-type: none"> Non déterminés
Couleur de la peau	Photographies	Valide	<ul style="list-style-type: none"> Respect des réglages Application dédiée 	<ul style="list-style-type: none"> Facile à mettre en œuvre Pas de perte d'information 	<ul style="list-style-type: none"> Fastidieux pour le patient
	Capteur RGB	Non valide	<ul style="list-style-type: none"> Différence de couleur entre les deux mains détectable 	<ul style="list-style-type: none"> Embarqué sur le gant, pas fastidieux 	<ul style="list-style-type: none"> Interface relativement compliquée
Amplitude des mouvements	Capteur de flexion	Valide	<ul style="list-style-type: none"> Bonne adaptation du capteur au gant 	<ul style="list-style-type: none"> Bonne précision du capteur Adaptation facile 	<ul style="list-style-type: none"> Aucun

Tableau 2 : Tableau récapitulatif du travail de recherche

Pour plus de détails sur cette phase de recherche, veuillez vous référer au document relatif au travail de semestre ayant précédé ce projet de Bachelor.

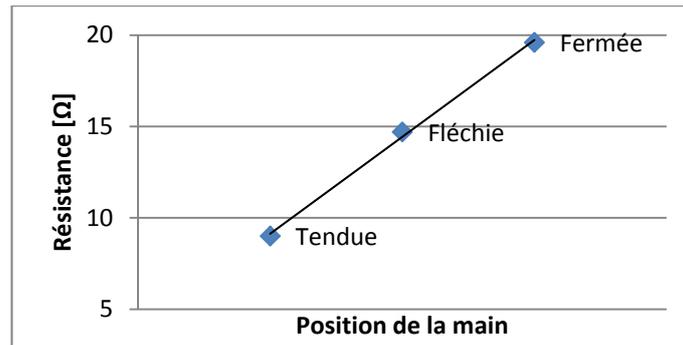
Les sections qui suivent présentent brièvement les procédés retenus suite au travail de recherche.

3.1 CAPTEUR DE FLEXION

Ce capteur aurait servi à mesurer l'amplitude maximale des mouvements de la main, en particulier lors de l'extension des doigts et la fermeture du poing. Sur la Figure 2 on peut discerner le capteur sous le gant, une bande résistive dont la valeur varie avec la flexion appliquée.



Figure 2 : Capteur de flexion



Graph 1 : Mesures avec le capteur de flexion

Les mesures faites avec ce capteur (voir Graph 1), ont permis de confirmer son bon fonctionnement.

Après consultation des médecins collaborant dans ce développement, il a été établi que ce capteur, bien que fonctionnel sur le principe, n'a pas d'utilité dans le cas présent. Les signes cliniques de l'impotence fonctionnelle sont en fait facilement observables lors d'un examen médical standard et ne nécessitent donc pas de mesure électronique.

Le capteur de flexion ne sera donc pas implémenté sur le prototype réalisé dans ce projet.

3.2 CAPTEUR D'ÉTIREMENT

Ce capteur est en fait constitué d'une ficelle de gomme élastique résistive (voir Figure 1Figure 3), dont la valeur varie en fonction de l'étirement qu'elle subit.

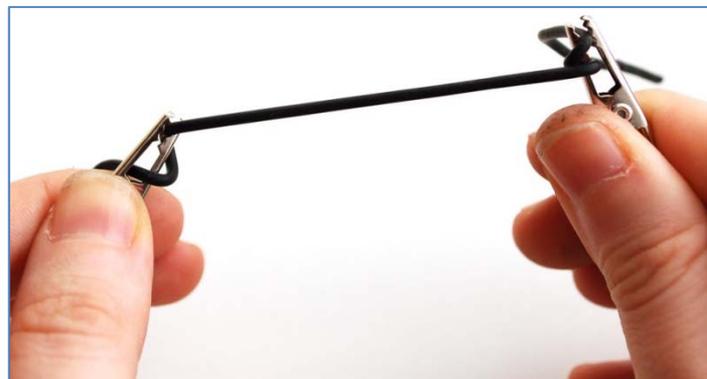


Figure 3 : Capteur d'étirement

En exploitant sa caractéristique, on peut l'utiliser pour évaluer la grandeur de l'œdème en le plaçant sur la main du patient. Plusieurs emplacements ont été étudiés; en effet il est possible de placer ce capteur sur la main comme pour une « mesure en huit³», autour du poignet ou simplement comme un anneau sur un doigt.

Il a été retenu que le capteur serait placé sur un des doigts, les autres placements étant soit non pertinents soit amenant trop d'erreur sur la mesure de par les mouvements des tendons et des muscles.

³ Document présentant les mesures en huit : Geraldine L. Pellecchia, PT, PhD. Figure-of-eight Method of Measuring. Hand Size: Reliability and Concurrent Validity (2003). doi:10.1197/S0894-1130(03)00154-6

3.3 PHOTOGRAPHIES

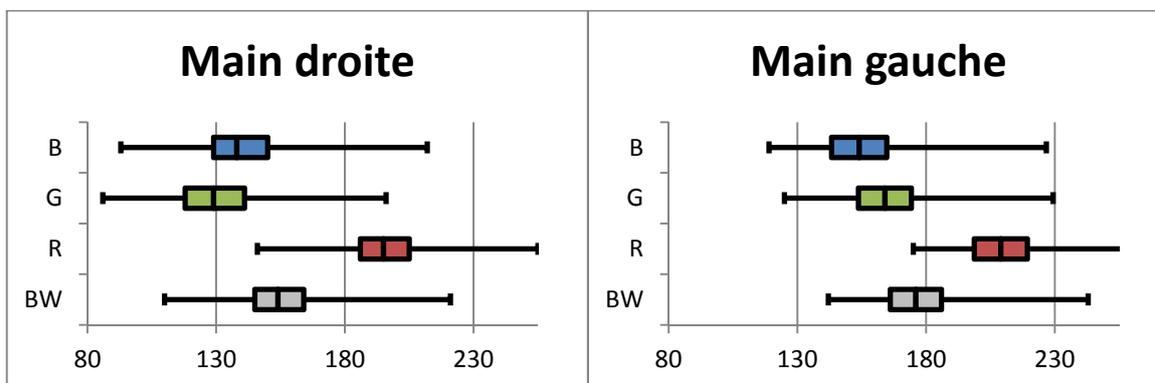
La Figure 4 illustre un exemple d'importants changements de la pigmentation de la peau due à la maladie.



Figure 4 : Mains droite atteinte d'algodystrophie⁴

C'est dans l'optique de pouvoir analyser cette variation de couleur en fonction de l'évolution de la maladie et de ses différentes phases, que l'option de la prise de photo a été étudiée. En effet cela reste le meilleur moyen pour évaluer l'aspect de la main malade en la comparant à celle qui est saine. Cette méthode implique que le patient fasse des photos régulièrement pendant la durée des tests.

Le Graphe 2 montre l'analyse de la photo de la Figure 4 avec un logiciel informatique. Le programme détecte un net décalage de la composante verte ainsi qu'un changement dans la composante bleue.



Graphe 2 : Différences entre les deux mains

Il en ressort que ce procédé est valide, car il permet de détecter une rougeur du membre atteint par la maladie aussi bien de manière triviale en regardant la photo, qu'en faisant appel à un logiciel d'analyse informatique. Rappelons ici que les photos, comme les autres principes de mesure n'ont de réelle valeur que si la mesure est faite de manière différentielle. Sinon, le manque de référence les rendrait nulles.

⁴ Source : <http://algoneurodystrophiecestquoi.e-monsite.com/medias/album/imgp8055.jpg>, le 02.07.2015

4 VALIDATION ET IMPLÉMENTATION DES CAPTEURS

Dans la première partie de ce projet, et suite aux éléments considérés dans le chapitre précédent, les deux derniers principes étudiés ont été repris dans le but de les intégrer au système de diagnostic. En effet, si les différents principes de mesure de ces capteurs sont confirmés, il s'agit maintenant de valider leur intégration réelle au prototype du gant.

4.1 CAPTEUR D'ÉTIREMENT

D'après les médecins, le capteur d'étirement peut se révéler utile si la précision de la mesure du périmètre du doigt peut atteindre 2 à 3 mm. Ainsi pour valider l'utilisation de ce capteur dans le cadre de ce projet, l'électronique d'acquisition a été modifiée pour ensuite pouvoir effectuer plusieurs tests.

4.1.1 ÉLECTRONIQUE

L'électronique de la première version du prototype a été modifiée pour pouvoir intégrer le capteur d'étirement. Elle comprend l'interface qui permet de convertir la faible variation de résistance de la ficelle en tension exploitable par le processeur. Ce dernier enverra ensuite l'information du convertisseur AD par Bluetooth avec un service adapté. La transposition en millimètres est faite par l'application afin d'avoir un affichage final lisible et pertinent.

Pour avoir les détails du développement électronique se référer aux Annexes 1 et 3.

4.1.2 VALIDATION

La Figure 5 illustre le procédé d'un test permettant d'établir un lien entre le diamètre de la gomme élastique et la grandeur électronique (en tension) qui y est associée.

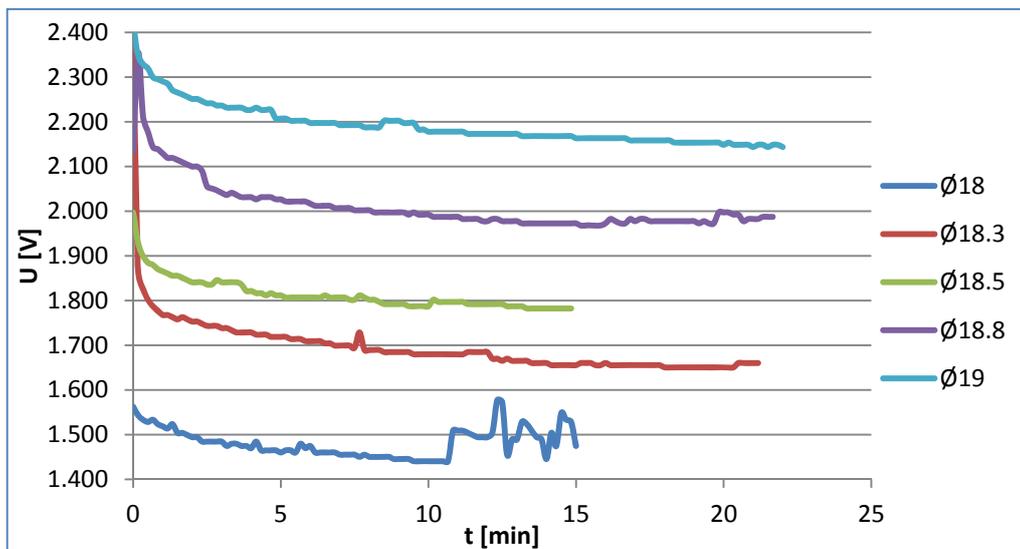


Figure 5 : Procédé de test

Les mesures ont été faites avec la ficelle résistive disposée en forme d'anneau autour d'un tasseau expansible. Cet outil de mesure mécanique permet de faire varier le diamètre du capteur et de pouvoir le mesurer précisément (ici avec un pied à coulisse). Sur l'image précédente, on voit la valeur du diamètre et de la tension qui y est associée.

Le premier graphique ci-après (voir Graphe 3), présente les résultats de cet essai. On y note qu'une variation de 3 dixièmes sur le diamètre mesuré, provoque une variation de tension d'environ 0.2 V.

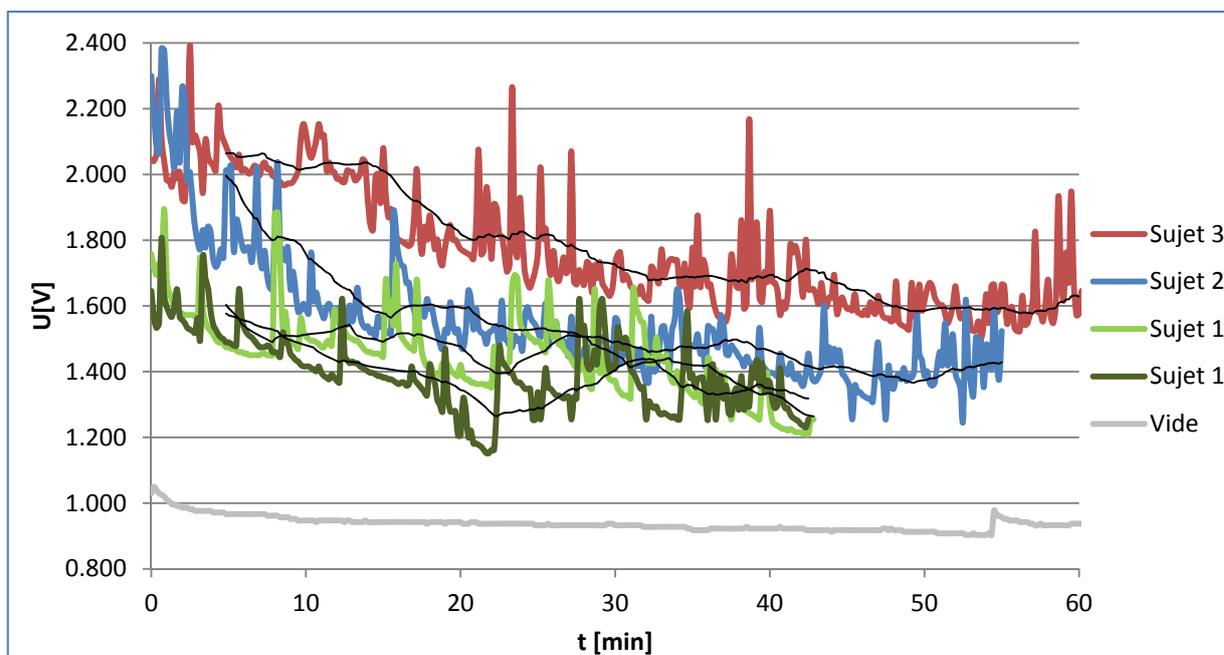
Cela représente une variation du périmètre de 0.94 mm, ce qui correspond aux critères discutés précédemment. Ce test s'est donc révélé concluant.



Graph 3 : Mesure avec tasseau expansible

La forme des courbes met également en évidence une certaine inertie du matériau, la procédure de l'expérience s'étant faite avec des variations rapides de diamètre. La mesure n'est donc pas immédiate. Cependant, dans notre contexte, où la variation du volume des tissus organiques est relativement lente, cela ne pose pas de problème.

Pour compléter cette première mesure, le capteur doit ensuite être mis dans les conditions normales de fonctionnement, c'est-à-dire sur la main d'une personne active. La mesure suivante sert donc à déterminer si le bruit sur la mesure provoqué par des mouvements courants des doigts est assez faible pour ne pas perdre de l'information. Le Graph 4 présente cette mesure.



Graph 4 : Mesure des doigts

Cette deuxième mesure met en scène trois sujets, qui portent le même gant, ainsi qu'une mesure à vide. Les trois sujets sont actifs et effectuent donc les mouvements habituels qu'ils sont appelés

à faire pendant la journée. La taille des doigts va croissant, sachant que le sujet 1 a les doigts les moins volumineux. La différence n'est cependant pas évidente à l'œil nu. Or, avec la mesure ci-dessus on constate une différence exploitable entre les sujets, qui n'est pas perturbée par le bruit sur la mesure de manière significative. En effet, on arrive toujours à distinguer les moyennes des signaux mesurés et seules les courbes moyennes d'un même sujet se croisent, ce qui démontre en outre la répétabilité de la mesure.

Suite à ces mesures, on peut conclure que le principe du capteur d'étirement est aussi applicable sur le gant de diagnostic.

4.1.3 IMPLÉMENTATION

La connectique pour relier le capteur à l'électronique d'acquisition est assurée par des connecteurs à sertir (voir Figure 6). On peut ainsi obtenir une bonne qualité du contact électrique sur la gomme résistive, tout en permettant au capteur d'être démontable.



Figure 6 : Connecteurs

Le contact a toute son importance, en effet la valeur résistive totale du capteur, dépend de la longueur de ficelle entre les deux contacts. Avec ce genre de connecteur, on peut réussir à contrôler les grandeurs des deux capteurs (main gauche et main droite) de manière précise et à l'ajuster de façon à ce qu'elles soient le plus proche possible l'une de l'autre.

La possibilité de démonter le capteur comporte deux avantages. Premièrement, vu que les gants seront donnés à plusieurs patients différents, cela permet d'empêcher que le lavage n'engomme le capteur (déformations, oxydation des contacts, etc.). Deuxièmement, ce capteur doit être adapté à la taille des doigts du patient, il serait en effet inutile sur un doigt moins grand que la taille de la ficelle au repos. Ainsi il est possible de mettre en place plusieurs « anneaux » de tailles différentes, qu'on choisirait en fonction du porteur des gants.

En ce qui concerne la fixation mécanique du capteur sur le gant, des pressions permettent de le maintenir en place. Ces dernières sont plus adaptées que le velcro, car elles permettent d'une part d'assurer un placement précis et elles ne sont pas affectées par la poussière. La ficelle résistive est de plus maintenue à angle droit par un segment de bande élastique, cela facilite le positionnement du capteur et la chausse du gant. La Figure 7 détaille la fixation.



Figure 7 : Pressions et maintien

4.2 PHOTOGRAPHIES DES MAINS

4.2.1 SUPPORT MÉCANIQUE

Le concept de la photographie des mains pour en apprécier la couleur est en soit simple. Le but étant de photographier les deux mains afin de procéder par comparaison, le patient a besoin d'une personne tierce qui puissent prendre la photo ou d'un support pour tenir l'appareil en place.

C'est dans cette optique que la structure mécanique visible en Figure 8 a été réalisée. Elle se compose principalement d'un profil en aluminium auquel viennent se visser le support pour loger le smartphone et les pieds qui équilibrent l'ensemble.

En ajoutant un délai au déclenchement sur l'appareil photo, le patient peut ainsi procéder lui-même à la photographie de ses mains.



Figure 8 : Support photo

4.2.2 RÉGLAGES OPTIQUES

Pour permettre une comparaison optimale de la couleur entre les deux mains il est indispensable d'activer le flash lors du cliché, afin que la lumière ambiante perturbe le moins possible l'image. Dans le même ordre d'idée, il a été établi que tous les réglages de la prise de vue seraient fixes afin de pouvoir aussi comparer les photos entre elles; cela n'est possible qu'avec le contrôle de luminosité donnée par le flash.

Grâce aux fonctionnalités de l'API *Camera2* développée par Android, il est possible d'agir sur tous les paramètres de prise de vue avec l'appareil photo du smartphone. L'application fixe donc les variables de l'appareil photo en termes de temps d'exposition, sensibilité ISO, ouverture et balance des blancs. Cela permet que les photos soient le moins possible faussées par la luminosité ambiante.

5 DÉVELOPPEMENT DE L'APPLICATION

L'application a été développée pour être compatible avec un système Android. Ce choix s'est fait pour une raison de facilité de travail. En effet programmer sur d'autres plateformes aurait non seulement requis des licences spécifiques, mais aussi la maîtrise de langages de programmation moins connus. Cela dit, le développement avec Android reste quelque chose qui demande un temps important, car cela met en jeu beaucoup de bibliothèques et d'interfaces à comprendre pour les utiliser.

5.1 FONCTIONNEMENT GLOBAL

L'application est programmée en Java et est constituée selon le schéma de principe de la Figure 9.

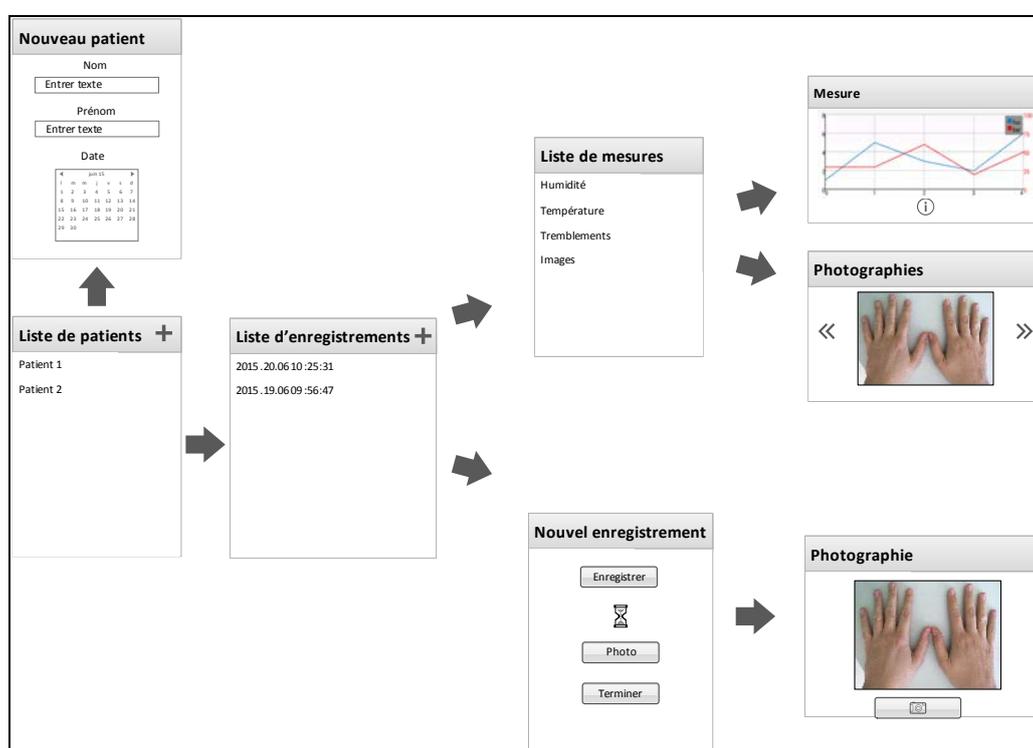


Figure 9 : Fonctionnement global de l'application

L'application se base sur une liste de patients qui peuvent avoir plusieurs enregistrements chacun. Elle contient une petite base de données interne qui stocke les informations du patient et les chemins d'accès aux fichiers de mesures qui lui correspondent.

Le cœur de l'application consiste dans la possibilité de se connecter aux capteurs intégrés sur le gant et de pouvoir procéder à l'enregistrement des données qu'ils émettent. La connexion se fait avec le protocole Bluetooth 4.0, version qui est conçue pour consommer très peu d'énergie lors des échanges de données. Elle fonctionne de manière événementielle, donc évite de maintenir une connexion constante pour demander à l'appareil connecté des valeurs en permanence. La connexion s'active que lorsqu'un événement a lieu.

Lors de cet enregistrement, il est possible d'interrompre le processus pour prendre une photo qui sera ensuite sauvegardée dans les fichiers propres au patient. La photo est ainsi privée à l'application et ne peut être modifiée ou supprimée au travers d'autres applications.

Une fois la compilation de mesures sauvegardée, il est possible de parcourir la liste d'enregistrements et de visualiser les résultats sous forme graphique ou les photographies prises pendant les mesures. Chaque enregistrement contient donc les champs suivants : humidité, température, tremblements, étirement et images.

Chaque mesure est visualisable séparément. On peut aussi visualiser ses détails, c'est-à-dire les valeurs maximale et minimale atteintes et la moyenne pendant la période d'enregistrement.



Figure 10 : Affichage des images dans l'application

Ce rapport ne convenant pas pour s'étendre sur des détails techniques, de plus amples explications sur la programmation elle-même sont rédigées dans le document joint en Annexe 2. Ce dernier précise le rôle de toutes les classes importantes pour les fonctionnalités comme la connexion Bluetooth, la gestion des services liés au gant de diagnostic, le traitement et l'affichage des valeurs, etc.

6 RÉSULTATS

Ce projet s'est en grande partie concentré sur le développement de l'application permettant de visualiser les mesures faites par les capteurs. Ainsi les résultats présentés dans ce chapitre feront essentiellement référence à ce qui est visible pour l'utilisateur du système.

6.1 VUES GRAPHIQUES DE L'APPLICATION

Les images suivantes présentent différentes vues de l'application. En particulier, les graphiques montrant respectivement des mesures d'humidité, de température, et des tremblements des doigts. Les échelles des graphiques s'adaptent automatiquement aux données collectées, ce qui permet une visualisation globale des mesures enregistrées. On peut néanmoins zoomer sur le graphe et le faire défiler pour une lecture plus détaillée.

6.1.1 HUMIDITÉ

La Figure 11 illustre l'enregistrement d'une mesure d'humidité. Ici nous prenons la valeur d'humidité mesurée au poignet (humidité ambiante) comme référence, soit la courbe bleue. La rouge représente l'humidité dans la paume de la main.

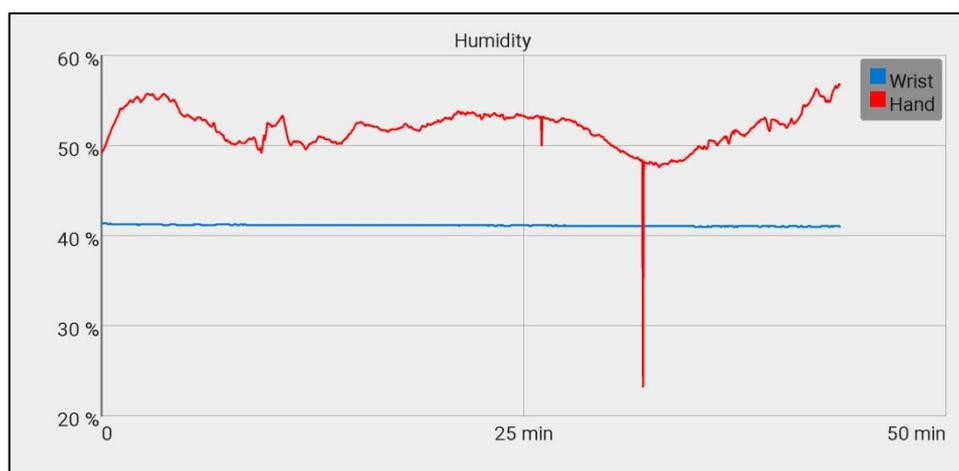


Figure 11 : Graphe d'humidité

Le capteur d'humidité utilisé a une résolution de 0.04%RH avec une précision de $\pm 3.0\%RH$. La mesure présentée plus haut est faite avec une fréquence d'acquisition d'un point de mesure toutes les 3 secondes. Avec une telle fréquence, on peut voir une variation de l'ordre de 17% autour de la moyenne qui est d'environ 52%RH.

On peut expliquer cette variation par le fait que le capteur ne soit pas forcément toujours pressé de la même manière sur la peau à cause des mouvements de la main. Un meilleur placement du capteur ou le choix d'un gant adhérent mieux à la peau, pourrait améliorer la stabilité de la mesure.

La précision du capteur est largement supérieure à celle que cette application requiert. Étant donné que le graphe peut être agrandi, la résolution des axes change en fonction de ce qui est affiché sur le graphe. On peut avoir une résolution de l'axe y de 5%RH ou de 10%RH.

L'application permet aussi de visualiser quelques détails supplémentaires sur la mesure comme on peut le voir sur la Figure 12 (valeur correspondant à la mesure de la Figure 11).

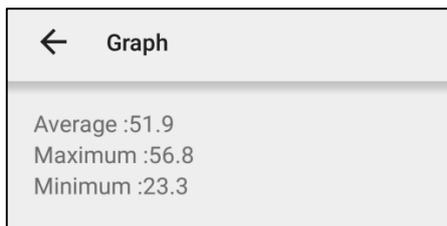


Figure 12 : Détails de la mesure

Afin de constater l'influence de la fréquence sur la précision de la mesure, voici l'exemple d'une mesure faite avec un temps d'attente de 5min entre chaque point (voir Figure 13). On remarque toujours une fluctuation dans la mesure, mais cette fois bien plus lente. Le résultat est plus propre, sans pour autant perdre de sa pertinence.

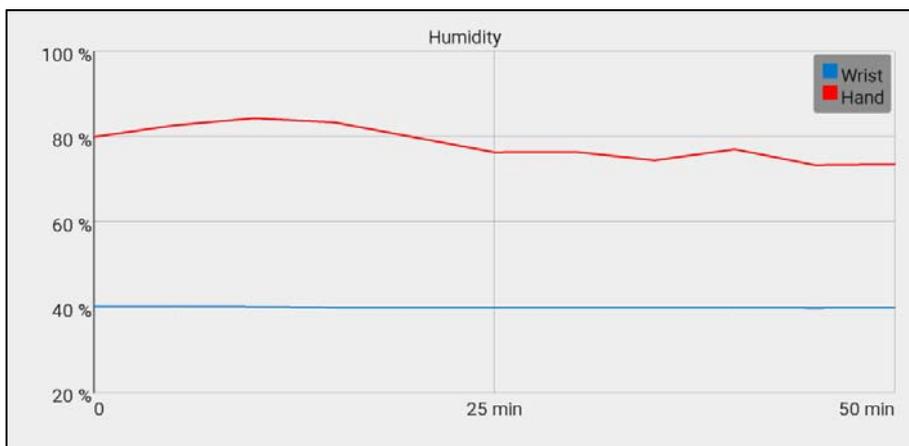


Figure 13 : Mesure d'humidité, 1pt/5min

6.1.2 TEMPÉRATURE

La mesure de la température est liée à celle d'humidité vu que le capteur implémente les deux fonctionnalités dans le même composant. La résolution et la précision pour la température sont néanmoins différentes, soit 0.025°C et respectivement, ±1.0°C. La Figure 14 présente un enregistrement prise avec 1 point de mesure chaque 5min.

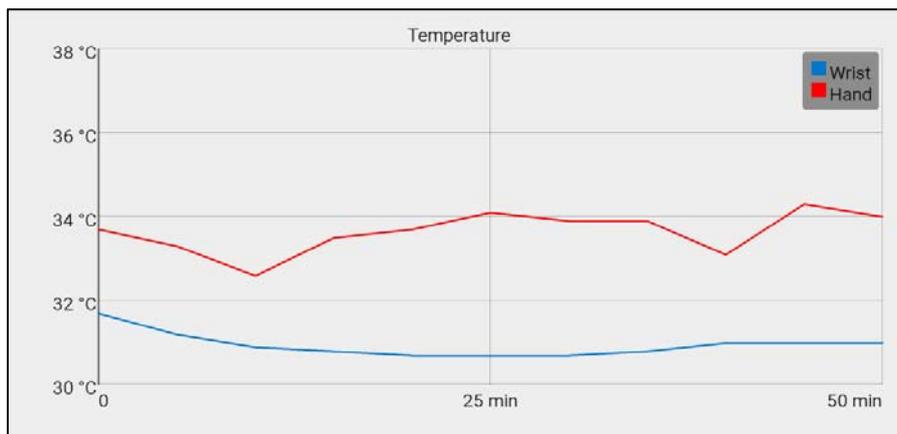


Figure 14 : Mesure de température

La courbe rouge correspond à la mesure faite dans la paume de main, et la courbe bleue à celle de la température ambiante.

De même que pour l'humidité, la température de la main n'est pas constante car le capteur « flotte » quelque peu dans le gant et fonction des mouvements effectués avec la main. Une fréquence plus élevée n'apporterait là aussi pas vraiment d'informations supplémentaires.

Le capteur de température a été comparé avec un thermomètre commercial, pour cela on a placé les deux parties sensible entre le doigt et l'index afin d'avoir une température stable. Voici les résultats de cette comparaison :

Capteur	Thermomètre	Δ calculé
34.5°C	34.0°C	0.5°C

Compte tenu de la précision de $\pm 1.0^\circ\text{C}$ des deux dispositifs, on peut conclure que notre capteur a une justesse tout à fait acceptable.

6.1.3 ÉTIREMENT

La mesure du périmètre des doigts n'a malheureusement pas pu être finalisée du fait du manque de temps. La conception de l'électronique et sa fabrication est terminée, mais il reste une grande partie du logiciel à terminer.

Cela représente en fait l'implémentation du service Bluetooth permettant d'accéder aux données brutes et aussi la conversion et l'affichage de ces valeurs pour qu'elles soient exploitables.

Des tests ont été effectués pendant le développement, leurs résultats sont visibles dans la section 4.1.1 du présent document.

6.1.4 MOUVEMENTS

La mesure des tremblements est faite à l'aide d'un accéléromètre. C'est un capteur miniature (3x3x0.9 mm), très léger (quelques grammes), fonctionnant avec un principe piézoélectrique associé à des jauges de contraintes pour mesurer l'accélération. L'accéléromètre utilisé est tridimensionnel, c'est-à-dire qu'il est sensible aux variations dans les trois dimensions. La mesure de ce composant englobe l'accélération dynamique (mouvements) et l'accélération statique (gravité). Ce qui nous intéresse étant la mesure de l'accélération dynamique liée aux tremblements des doigts, la transformation est directement faite en valeur relative, donc en supprimant la composante de la gravité.

Le capteur est placé au bout de la phalange proximale du majeur.

La procédure de mesure est la suivante : le capteur s'enclenche chaque minute pendant 5 secondes. Durant ces 5 secondes, les mesures sont effectuées avec un intervalle de temps de 200ms, ce qui nous donne 25 mesures à chaque minute. Chaque point d'acquisition contient les composantes des trois axes. L'application calcule alors la norme résultante pour chaque point et détermine la moyenne pour chaque groupe de 25 mesures. Ce résultat est ensuite affiché sur un graphique temporel.

La Figure 15 montre une capture des mouvements, lorsque le porteur du gant est en activité. On distingue les deux mesures des accéléromètres (poignet et doigt).

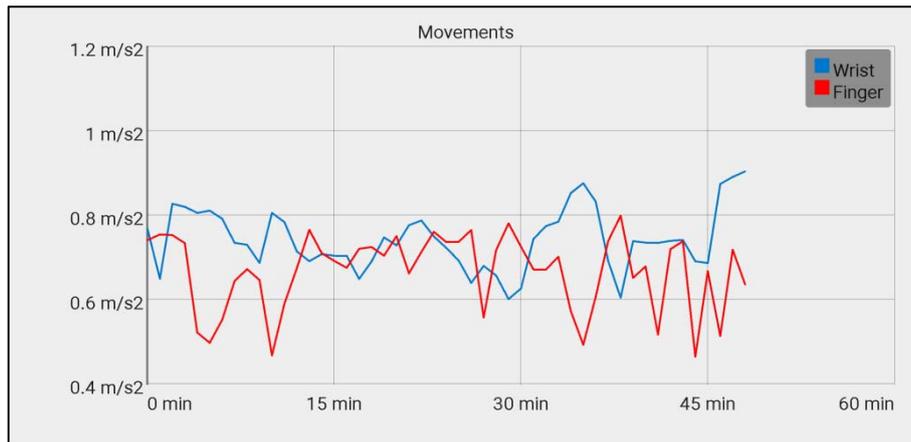


Figure 15 : Mesure des mouvements

Ces valeurs ne sont toutefois pas encore tout à fait exploitables. Comme le montre le raisonnement qui suit, d'ultérieurs essais doivent être faits pour parfaire l'affichage de ces informations.

Une mesure à vide, c'est-à-dire sans que les accéléromètres ne soient soumis à des mouvements, a été effectuée pour déterminer la précision des senseurs. Elle est illustrée par la Figure 16.

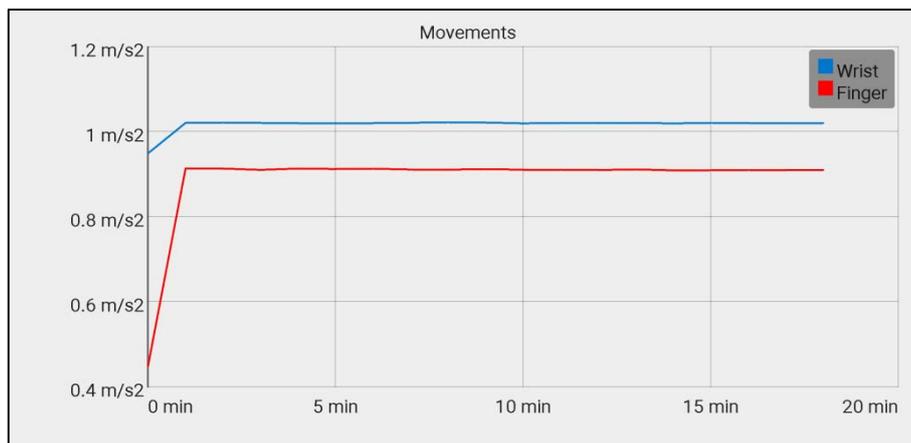


Figure 16 : Mesure à vide

On note que les accéléromètres n'ont pas une caractéristique identique et qu'il faut ajouter un facteur de constant pour corriger leur référence. Afin de définir s'il y a des rectifications dynamiques à faire sur la mesure, on a comparé les résultats affichés par l'application avec un accéléromètre de référence. Dans les deux images de la Figure 17, on peut voir le signal appliqué au dispositif pendant plusieurs minutes et le résultat fourni par l'application.

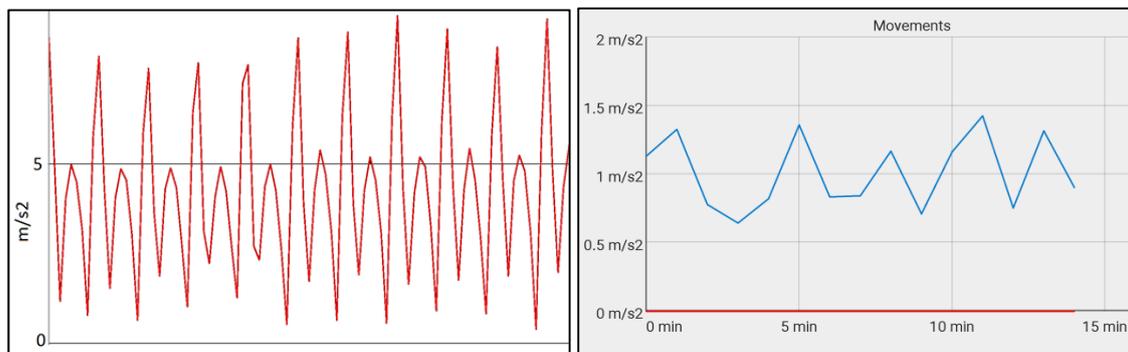


Figure 17 : Comparaison avec une référence

Vu qu'on affiche des valeurs moyennes, le résultat devrait être constant, compte tenu de la stimulation d'entrée. Or, on constate au contraire des variations d'une grande amplitude, relativement à la mesure. Cela est certainement dû à la fréquence d'acquisition des mesures qui n'est pas idéal. Il faut donc déterminer la fréquence des tremblements à mesurer pour pouvoir échantillonner à la bonne fréquence. Ensuite le signal ne donne pas une valeur correcte car il devrait varier autour des 4 m/s^2 au lieu des 1 m/s^2 affichés. On en conclut que d'autres modificateurs doivent être appliqués pour cette mesure.

6.2 PHOTOGRAPHIES

Comme expliqué dans les chapitres précédents, la prise de photographies est utilisée pour déterminer la couleur de la peau. C'est en effet le moyen le plus direct d'effectuer une telle mesure.

L'application intègre la possibilité de faire des photos pendant l'enregistrement des données, ce qui permet de continuer les mesures une fois le cliché terminé.

Ci-dessous, quelques photos illustrent les résultats qu'on peut obtenir en utilisant le mode photo. Rappelons que les réglages des prises de vue ont été fixés manuellement de manière logicielle, ainsi rien n'est automatique dans le processus d'évaluation des paramètres d'exposition.

La Figure 18 montre deux photos, prises au même endroit, l'une avec la lumière (artificielle) de la pièce allumée et l'autre avec la lumière éteinte.



Figure 18 : Gauche avec lumière ; Droite sans lumière

On peut voir que la luminosité de l'image change légèrement mais la teinte des couleurs restent, bien que plus sombres, les mêmes. Ainsi ces deux clichés peuvent être comparés.

La Figure 19 présente deux photos faites dans des pièces sans éclairage artificiel mais exposées différemment au soleil et donc avec deux luminosités distinctes. Les mêmes remarques que pour la Figure 18 s'appliquent ici, donc que les couleurs sont comparables malgré la différence de luminosité.



Figure 19 : Eclairage naturel dans deux pièces différentes

La Figure 20 montre les limites du systèmes avec deux photos l'une sur-exposée et l'autre respectivement sous-exposée. La première est prise à côté d'une fenêtre (projetant la lumière sur la droite de l'image) lors d'une journée ensoleillée. La deuxième est prise dans un couloir sombre sans aucun éclairage. Certainement les mains restent discernables mais ces deux photos ne sont pas comparables entre elles.



Figure 20 : Photos sur/sous-exposées

Sur ces derniers clichés, on remarque que la lumière du flash n'est pas suffisante pour éclairer la scène de manière à ce que les couleurs ne soient pas faussées. Donc ces photos ne pourraient pas être utilisées pour effectuer des comparaisons avec d'autres clichés. Si les réglages avaient été faits à partir de telles images, alors les autres photos auraient affiché une exposition inadéquate. On suppose cependant que le patient fera toujours les photos dans un endroit avec un éclairage minimal et en faisant attention à ne pas se mettre à proximité directe de la lumière du jour.

Répetons que le fait de comparer deux photos entre elles n'est qu'un aspect accessoire; ce qui est réellement important est de pouvoir comparer les deux mains sur un seul et même cliché. Or cela est possible dans tous les cas à une exception près; si les deux mains ne sont pas éclairées de la même manière et que la luminosité est trop forte pour être compensée par le flash.

7 SYNTHÈSE

Ce chapitre expose la synthèse du travail réalisé et les capacités du système développé au stade actuel.

7.1 TRAVAIL RÉALISÉ

Le prototype dans son état actuel est complet en terme de matériel, tout est à disposition pour que le système fonctionne correctement. Le design du gant est terminé dans les limites de ce qu'il a été possible de faire avec des gants déjà fabriqués (voir Figure 21). Il est fonctionnel et ses performances seront considérablement améliorées en fabricant des gants sur mesure (cf. 7.2.2).



Figure 21 : Prototype du gant de diagnostic

Le boîtier et la structure mécanique servant de trépied au smartphone sont également prêts à l'usage, ils sont conçus de manière simple, avec le moins de pièces possible.

La partie électronique de l'appareil est complètement achevée; elle permet l'interface entre les cinq capteurs et le téléphone portable, traitant les valeurs brutes mesurées sur le poignet et sur la main.

Le dispositif permet au patient de s'enregistrer dans la base de données de l'application et d'effectuer sur sa main les mesures listées ci-dessous. Ces mesures ont pour référence les senseurs placés sur le poignet.

- Mesure d'humidité au niveau de la paume, avec une précision de $\pm 3.0\%RH$
- Mesure de température au niveau de la paume, avec une précision $\pm 1.0^{\circ}C$
- Mesure des tremblements du majeur (précision restant à déterminer)
- Photographie des mains en vue de comparer leur couleur

Chacun des enregistrements est ensuite répertorié par patient et par date.

L'application Android consent finalement au médecin de visualiser les mesures prises antérieurement à l'aide de graphiques temporels, en plus des photographies faites par chaque patient.

La mesure avec le capteur d'étirement permet de distinguer jusqu'à 2mm de différence entre les diamètres. Elle n'est cependant pas encore accessible dans l'application smartphone.

7.2 PERSPECTIVES

Différents éléments seraient à ajouter pour que le prototype soit totalement opérationnel. Cette section propose également quelques options supplémentaires pouvant être ajoutées dans le développement futur du dispositif.

7.2.1 TRAVAIL RESTANT

Pour répondre au cahier des charges, l'application doit principalement pouvoir se connecter avec deux appareils simultanément et ainsi pouvoir collecter les mesures provenant des deux mains. L'affichage se ferait alors en superposant les deux mesures. Sans cela, le prototype est peu efficace car il n'implémente pas la possibilité de comparer une main malade avec une main saine. En d'autres termes, il n'y a pas de référence sur laquelle s'appuyer pour poser un diagnostic.

La mesure de périmètre qui permet de déterminer la présence d'un œdème reste à finaliser. Il s'agit de la conversion de la valeur numérique mesurée en son équivalent en millimètres et l'affichage du résultat. Une partie de la programmation est déjà prête pour ajouter cette fonctionnalité, elle devrait donc être opérationnelle lors de la défense orale du projet en septembre prochain.

Dans la mesure où tous les capteurs (pour chaque type de mesure) n'ont pas exactement le même comportement, des mesures à vide supplémentaires seraient utiles afin de déterminer les rectifications à apporter. Cela peut être fait par la suite de manière logicielle en ajoutant un offset ou un facteur correctif, permettant ainsi d'avoir des mesures aussi justes que possible.

7.2.2 AMÉLIORATION POSSIBLES

Le choix d'un gant commercial ne conviendra jamais aux critères nécessaires pour ce type d'application, de plus un gant déjà fabriqué est très difficile à retravailler et à modifier. Il faut donc faire fabriquer des gants sur mesure, et pour cela prévoir une liste d'exigences complète concernant les tissus à utiliser, leur quantité et disposition sur le gant, la fixation des capteurs, etc. Tout cela en collaboration avec une personne compétente dans le domaine, afin d'associer au mieux confort et fonctionnalité des gants.

Il serait judicieux d'ajouter une notification ou une alarme pour indiquer au patient le moment pour prendre la photo, cela le tranquillise, prévient tout oubli de sa part et assure une régularité dans la prise des photos.

Pour l'instant l'application est programmée pour se connecter automatiquement, mais exclusivement à dispositif en particulier. À l'avenir, la possibilité de lier des appareils en sauvegardant ensuite leur adresse serait intéressante, cela éviterait de faire dysfonctionner l'application en la connectant à des appareils inadéquats tout en gardant le caractère automatique de la connexion.

Il est possible d'imaginer augmenter le nombre d'informations liées à chaque graphe et ajouter des indications telles que la normalité de la mesure, une comparaison avec un enregistrement antérieur, des données statistiques de l'ensemble des enregistrements, etc.

7.3 CONCLUSION

Au terme du temps imparti pour ce projet, le travail est presque terminé. En l'état actuel, le prototype est fonctionnel pour une seule main, et permet au médecin d'avoir des informations complémentaires aux signes cliniques visibles pendant la visite médicale. De ce fait, le système peut déjà aider dans une certaine mesure le médecin pour la pose du diagnostic.

Pour le patient, il est possible d'effectuer les mesures en plusieurs fois durant la journée en procédant à plusieurs enregistrements différents.

La synchronisation des mesures entre les deux mains et la finalisation de la mesure de périmètre du doigt sont encore à compléter pour pouvoir prétendre d'avoir un prototype qui remplisse vraiment l'objectif fixé au début du projet. Suite à cela, plusieurs tests et ajustements seront encore nécessaires pour parfaire l'efficacité de l'appareil.

Le but n'a pas pu être totalement atteint à cause de la coordination difficile du travail entre les collaborateurs, occasionnant des retards importants. En effet le projet mêle plusieurs domaines d'activité différents allant de la couture, à la programmation haut niveau d'une application, passant par du développement électronique et de la mécanique. Tous ces éléments n'ont pas été évidents à concilier avec mon manque d'expérience dans ce type de projet complexe.

7.4 ANNEXES

1. Détails de l'électronique
2. Détails de programmation
3. Schématique de l'électronique d'acquisition
4. Liste de composants
5. Mode d'emploi

7.5 ANNEXES NUMÉRIQUES

Les documents suivants sont des fichiers informatiques qui sont indispensables pour la réalisation et la suite du projet. Ils seront donc placés sur le support CD-ROM rendu en parallèle.

1. Projet java complet programmé sous Android Studio
2. Schématique et PCB (fichiers Altium)

7.6 REMERCIEMENTS

Plusieurs personnes ont apporté leur aide précieuse durant tout le déroulement de ce projet de diplôme. C'est pourquoi je tiens à remercier chaleureusement les personnes suivantes.

Le Dr. Pierre Roduit pour ses conseils relatifs à l'organisation du projet, son suivi semaine après semaine a été très important pour maintenir le bon cap. Il a également toujours fait preuve de disponibilité en cas de questions ou de problèmes relatif au projet.

M. Patrice Rudaz pour sa collaboration durant le projet, ses conseils concernant les services Bluetooth, technologie que je n'avais encore jamais eu l'occasion d'employer de cette manière.

Pour la conception de la nouvelle carte électronique et le montage du PCB, M. Steve Gallay. Je le remercie pour son travail et sa disponibilité.

M. Pascal Sartoretti que je remercie pour ses conseils dans la partie électronique, dans la mise à disposition de tout le matériel nécessaire au développement et la gestion des impressions 3D.

Merci à M. Nohen Deulieutraz qui, travaillant dans le domaine de la mécanique, m'a grandement aidé pour la conception mécanique du trépied photo.

Toute ma gratitude encore à Mlle. Jennifer Amacker pour la réalisation d'une grande partie des coutures et des fixations sur le gant lui-même.

Je remercie finalement tous mes proches pour leur soutien et encouragements durant ces neuf semaines.

ANNEXE 1

DÉTAILS DE L'ÉLECTRONIQUE

1 AVANT-PROPOS

Ce document présente les modifications apportées à l'électronique d'acquisition en vue d'ajouter le capteur d'étirement. Étant donné qu'il s'agit d'une mesure résistive, il est nécessaire d'avoir un étage de conversion et d'amplification du signal.

2 CAPTEUR D'ÉTIREMENT

Pour intégrer le capteur dans le système global, il faut donc une conversion en tension. La mesure est réalisée avec un montage en pont de Wheatstone, afin de mesurer des petites variations de résistance, et un amplificateur pour que la chute de tension soit exploitable de manière numérique par la suite. La Figure 1 représente le schéma utilisé pour nos mesures.

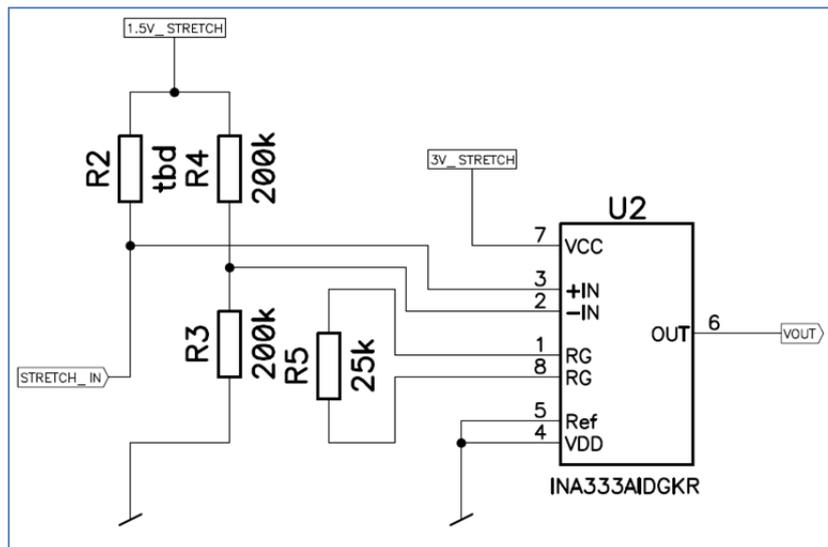


Figure 1 : Acquisition de la mesure

La résistance R5 permet de régler le gain selon les valeurs de la Table 1, tirée du datasheet du composant INA333. Le choix est ici de 25kΩ afin de fixer l'amplification à un facteur 5.

DESIRED GAIN	R _G (Ω)	NEAREST 1% R _G (Ω)
1	NC ⁽¹⁾	NC
2	100k	100k
5	25k	24.9k
10	11.1k	11k
20	5.26k	5.23k
50	2.04k	2.05
100	1.01k	1k
200	502.5	499
500	200.4	200
1000	100.1	100

Table 1 : Gains couramment utilisés et valeur de résistances

Le calcul de la résistance R2 est quant à lui dépendant de la résistance du capteur, de l'amplification appliquée et de la plage de sortie désirée. L'alimentation étant assurée par une pile 3V, la plage d'entrée du convertisseur AD du processeur NRF peut être au maximum de 0-3V. Par souci d'économie d'énergie la tension d'alimentation de cette partie a été diminuée de moitié. Afin de rester dans la bonne plage avec une marge d'erreur commençons par fixer le seuil minimal à 0.2V. On utilise ensuite la formule suivante pour le calcul de R2, avec R_{stretch}, la résistance du capteur au repos :

$$V_{out_{min}} = \left(\frac{V_+ * R_{stretch}}{R2 + R_{stretch}} - \frac{V_+}{2} \right) * A$$

Comme expliqué dans le rapport (voir point 4.1.2), le capteur peut au final prendre plusieurs valeurs différentes, du fait qu'il doit s'adapter à plusieurs tailles de doigts différentes. Ainsi, on prend la valeur minimale, en l'occurrence 105Ω pour fixer R2. Cette valeur a été déterminée pour une taille de doigt d'enfant, correspondant donc à 40mm de longueur de ficelle. En manipulant la formule précédente on obtient :

$$R2 = \frac{V_+ * R_{stretch}}{\frac{V_{out_{min}}}{A} + \frac{V_+}{2}} - R_{stretch} = \frac{1.5 * 200}{\frac{0.2}{5} + 0.75} - 200 = 179.7\Omega$$

Suite à ce calcul la valeur standard de 180 Ω a été choisie. Avec cette valeur on obtient un seuil minimal V_{out_{min}} de :

$$V_{out_{min}} = \left(\frac{V_+ * R_{stretch}}{R2 + R_{stretch}} - \frac{V_+}{2} \right) * A = \left(\frac{1.5 * 105}{100 + 105} - 0.75 \right) * 5 = 0.09V$$

La valeur du seuil minimal (0.2V) sera retenue, pour ensuite appliquer un offset dans la partie programmation, afin de bien régler le zéro.

Si on veut utiliser toute la plage de conversion, on pourrait donc mesurer des valeurs de résistances R2 jusqu'à :

$$V_{out_{max}} = 1.5V = \left(\frac{V_+ * R_{stretch}}{R2 + R_{stretch}} - \frac{V_+}{2} \right) * A \Rightarrow R_{stretch} = 420\Omega$$

Le capteur le plus grand sera d'une longueur de 10 cm de long pour une résistance de 250 Ω au repos, et qu'avec une déformation supplémentaire de 5mm on arrive à une valeur ohmique de 400Ω. On remarque que les valeurs correspondent.

Lorsque le capteur va augmenter de taille, sa résistance va grandir de manière proportionnelle. Par exemple, pour une variation de 2mm il a été mesuré un ΔR d'environ 100Ω. Par conséquent, la variation de tension de sortie sera de :

$$\Delta V_{out} = \left(\frac{V_+ * R_{stretch}}{R2 + R_{stretch}} - \frac{V_+}{2} \right) * A - 0.2V = \left(\frac{1.5 * 280}{280 + 180} - 0.75 \right) * 5 - 0.2 = 0.615V$$

L'utilisation d'un pont de Wheatstone pose un problème au niveau de la consommation de courant. En effet le courant consommé est ici directement proportionnel aux résistances présentées dans le pont. La résistance du capteur étant faible (quelques centaines de Ohms), l'intensité du courant n'en est que plus grande. Pour pallier à ce problème, les résistances R3 et R4 ont été choisies avec une grande valeur ohmique; le courant qu'elles consomment est de :

$$i = \frac{1.5V}{400k\Omega} = 3.75 \mu A$$

En outre, des modifications ont été apportées à la partie alimentation du circuit afin de diminuer encore la consommation.

3 ALIMENTATION

La partie gérant l'alimentation de l'électronique a subi deux modifications soit, l'ajout d'un mosfet contrôlant le courant alimentant le capteur d'étirement et l'insertion d'un composant permettant de diviser la tension fournie par la batterie. La Figure 2 montre les éléments venu s'ajouter à la schématique de base.

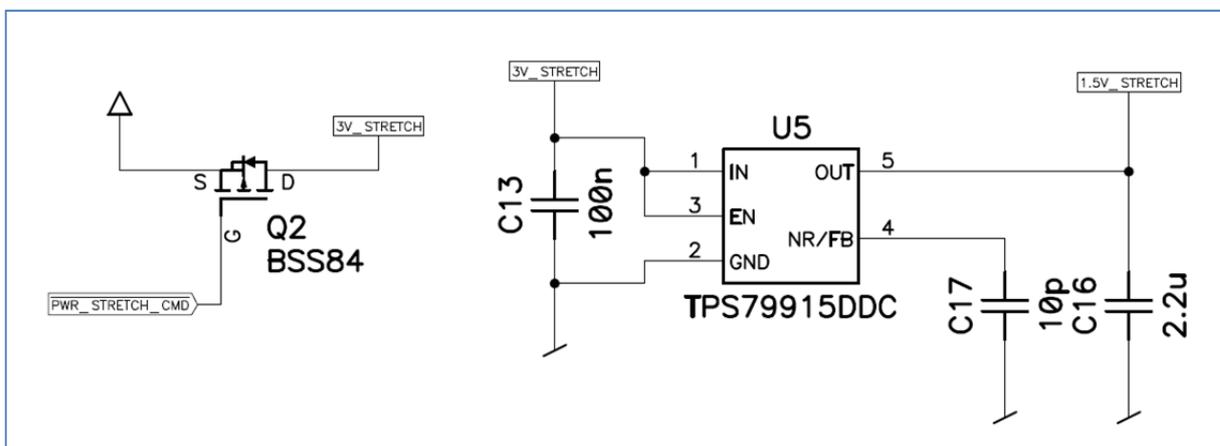


Figure 2 : Partie alimentation du capteur

Q2 fonctionne donc comme un interrupteur activé seulement lorsque les mesures sont activées, cela réduit de manière drastique la consommation de courant, vu qu'il n'est consommé que périodiquement. Le composant U5 est quant à lui un abaisseur de tension (3V vers 1.5V). Sachant que le pont de résistance mentionné dans le chapitre précédent est très gourmand en énergie, cela permet encore de réduire d'un facteur 4 la puissance absorbée par le capteur.

Étant donné que la création du service Bluetooth correspondant à ce capteur n'a pas pu être terminée à temps, aucune mesure de consommation du circuit total n'a pu être réalisée.

ANNEXE 2

DÉTAILS DE PROGRAMMATION

1 AVANT-PROPOS

Ce document est un complément au rapport du Travail de diplôme intitulé « Diagnostic de l'algodystrophie ». Il a pour but d'offrir une vue du travail réalisé du côté de la programmation et d'expliquer les parties du code les plus importantes.

La base du code utilisé a été reprise d'un autre projet qui implémentait plus ou moins les mêmes fonctionnalités, des capteurs connectés en Bluetooth pour mesurer la déviation du genou. J'ai repris ce code créé par Fabian Imsand pour l'adapter à mes besoins.

La programmation de l'application smartphone a été faite sur le système d'exploitation mobile Android. L'environnement de développement est Android Studio, outil très performant du fait qu'il est l'environnement officiel prévu par Android pour le développement d'applications mobiles.

L'architecture d'Android combine plusieurs composants constituant le système d'exploitation (voir Figure 1). Les composants avec lesquels on a principalement à faire dans notre application sont le *View System*, le *Resource Manager*, le *Content Providers* et l'*Activity Manager*. Ce dernier gère les activités, classes principales faisant tourner l'application. Chaque activité peut aussi avoir un ou plusieurs *fragments* qui permettent de définir des parties d'interfaces, un ensemble de vues, qu'il est possible de contrôler et de réutiliser pour gérer par exemple des écrans de taille différentes. Le langage de programmation utilisé est le Java.

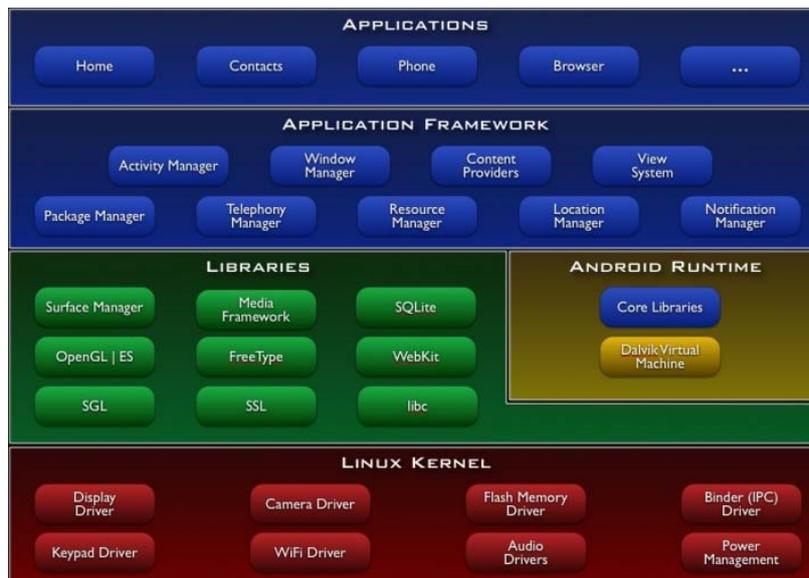


Figure 1 : Architecture d'Android¹

La suite de ce document détaille les activités principales de l'application Android développée.

¹ Source de l'image : <https://upload.wikimedia.org/wikipedia/commons/thumb/a/af/Android-System-Architecture.svg/800px-Android-System-Architecture.svg.png>, le 09.07.2015

2 DEVICE SCAN ACTIVITY

Laissant de côté les activités s'occupant de l'ajout de patients dans la base de donnée de l'application et la visualisation de leurs enregistrements respectifs, passons directement à cette activité qui sert à scanner les environs en Bluetooth pour trouver des appareils compatibles. Cette activité est basée sur le code d'exemple donné par Android dans la gestion de dispositifs Bluetooth. Ces exemples sont disponibles dans les « samples » données directement par sur le site *developer.android.com*, plus précisément dans le projet nommé BluetoothLeGatt. Toutes les sources de ce projet d'exemple sont disponibles sous le lien suivant :

<https://developer.android.com/samples/BluetoothLeGatt/project.html>

Avec cette classe il est possible de visualiser tous les appareils Bluetooth à proximité pour demander ensuite la connexion à l'un d'eux. Cette classe est a été conservée dans le but de lier des appareils à l'application, mais elle a été dupliquée et modifiée pour se connecter automatiquement aux appareils dont l'adresse est déjà connue.

2.1 KNOWN DEVICE SCAN ACTIVITY

Ainsi cette classe a été renommée et transformée pour enlever la possibilité de choix par l'utilisateur. Cela évite de se connecter par erreur à des appareils non-compatibles avec l'application. L'activité est appelée au moment où l'utilisateur veut ajouter un nouvel enregistrement (*Record*) pour le patient sélectionné.

La partie importante de ce code se trouve dans la fonction *scanLeDevice()* (visible sous le Code 1), elle définit le mode de scan en Low Power Mode, ce qui permet de ne détecter que les appareils compatibles BLE, et lance le scan si l'adresse a le bon format (par exemple "00:43:A8:23:10:F0").

```
if(mBluetoothAdapter.checkBluetoothAddress(mDeviceAddress)) {
    ScanSettings.Builder bSettings = new ScanSettings.Builder();
    bSettings.setScanMode(ScanSettings.SCAN_MODE_LOW_POWER);
    bSettings.setReportDelay(0);
    ScanSettings settings = bSettings.build();
    [...]
    mScanning = true;
    mBluetoothScanner.startScan(null, settings, mScanCallback);
}
```

Code 1 : Scan

Ce qui a été modifié est le callback (voir Code 2). Les callbacks sont des fonctions appelées de manière asynchrone lors d'évènements survenant dans la classe à laquelle ils sont liés. En l'occurrence la découverte d'un nouvel appareil compatible. À ce moment-là, si ce dernier est l'appareil déjà enregistré par l'application, une nouvelle activité est démarrée pour commencer le nouvel enregistrement.

L'*Intent* est une description abstraite d'une opération à faire. Il peut être utilisé pour démarrer une activité (avec *startActivity*) ou communiquer avec des services (avec *startService* et *bindService*). Il est possible d'y ajouter des arguments pour faire passer des valeurs d'une activité à l'autre ou vers un fragment.

```
// Device scan callback.
private ScanCallback mScanCallback = new ScanCallback() {
    @Override
    public void onScanResult(int callbackType, ScanResult result) {
        super.onScanResult(callbackType, result);
        BluetoothDevice device = result.getDevice();
        if(device.getAddress().equals(mDeviceAddress))
        {
            final Intent intent = new Intent(getApplicationContext(),
                RecordAddActivity.class);
            [...]

            startActivity(intent);
        }
    }
};
```

Code 2 : Scan callback

3 RECORD ADD ACTIVITY

Cette activité permet d'ajouter un enregistrement, elle représente le cœur de l'application. C'est en effet à partir de là que le gros des fonctions de l'application s'enchaînent. Cette activité gère deux fragments utilisés pour la connexion à l'électronique d'acquisition et la récupération des données qu'elle envoie.

Le Code 3 montre les actions effectuées lors de la création de l'activité. L'appel du *KeyguardManager* sert à éviter que l'application ne se mette en pause si l'écran est verrouillé lorsqu'elle est dans cette activité. Cela permet de continuer les mesures avec la connexion Bluetooth tout en ayant l'écran verrouillé. Ensuite le gestionnaire de fragments lance celui qui va justement gérer cette connexion et collecter les mesures.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_record);

    KeyguardManager keyguardManager = (KeyguardManager)
        getSystemService(Activity.KEYGUARD_SERVICE);
    KeyguardManager.KeyguardLock lock =
        keyguardManager.newKeyguardLock(KEYGUARD_SERVICE);
    lock.disableKeyguard();

    getWindow().addFlags(WindowManager.LayoutParams.FLAG_SHOW_WHEN_LOCKED);
    [...]

    mContent = DeviceControlFragment.newInstance();
    getFragmentManager().beginTransaction().replace(R.id.fragmentContainer,
        mContent).commit();
}
```

Code 3 : Création de l'activité d'enregistrement

Cette activité contient un deuxième fragment qui implémente l'appareil photo. C'est pourquoi les deux fonctions dans le Code 4 sont ajoutées, pour prendre en charge le passage d'un fragment à un autre.

```
public void openCamera() {  
  
    [...]  
  
    Camera2Fragment fragment = Camera2Fragment.newInstance();  
    getFragmentManager().beginTransaction().replace(R.id.fragmentContainer,  
        fragment).commit();  
}  
  
public void closeCamera() {  
    mContent = DeviceControlFragment.newInstance();  
    [...]  
    getFragmentManager().beginTransaction().replace(R.id.fragmentContainer,  
        mContent).commit();  
}
```

Code 4 : Passage d'un fragment à un autre

La fonction la plus importante est celle présentée dans le Code 5, elle sert à sauvegarder l'enregistrement. Pour cela on procède avec un SharedPreferences, système de stockage de données primitives dans des paires clé-valeur, les données reçues étant simplement des grandes chaînes de caractères. On sauvegarde donc seulement les clés correspondantes à chaque capteur et les chemins d'accès aux fichiers contenant les photos prises.

```
public void saveRecord(Record record, long timestamp)  
{  
    record.setAllPictures(mPicString);  
    record.setSensor(0, timestamp + ".WristHumidity");  
    record.setSensor(2, timestamp + ".WristTemperature");  
  
    [...]  
    SharedPreferences sharedPreferences = getApplicationContext()  
        .getSharedPreferences("ch.hevs.algodiag.algodystrophy",  
        Context.MODE_PRIVATE);  
    SharedPreferences.Editor editor = sharedPreferences.edit();  
  
    for (int i = 0; i < 10; i++)  
        editor.putString(record.getAllSensors().get(i),  
            mMeasurementsArray[i]);  
  
    editor.commit();  
    RecordDataSource recordDataSource =  
        new RecordDataSource(getApplicationContext());  
    recordDataSource.createRecord(record, mPatientId);  
}
```

Code 5 : Sauvegarde d'un enregistrement

3.1 DEVICE CONTROL FRAGMENT

Ce *Fragment* est basé sur la classe *DeviceControlActivity* donnée en exemple sur le site officiel d'Android. La classe a été convertie en fragment afin de pouvoir passer du mode d'acquisition de valeurs au mode photo plus facilement tout en séparant adéquatement ces classes.

C'est dans cette classe que la connexion proprement dite avec l'appareil est lancée. Après qu'elle ait été établie, l'utilisateur peut enclencher la mesure à l'aide d'un bouton, entraînant l'initialisation de l'électronique. Malheureusement le service BLE n'implémente pas de queues pour les requêtes faites aux différentes caractéristiques. Après de nombreux dysfonctionnements du programme suite à cela, j'ai établi un système de FIFO.

Pour que le processus fonctionne correctement il faut suivre la procédure suivante : enclencher les notifications, changer les paramètres et activer la mesure. Sans cet ordre, certaines requêtes sont simplement ignorées par le service.

```
private void enableNotifications(){
    [...]

    // Wrist Temperature and Humidity notification enable
    characteristic = gatt.getService(UUID_ADMS_SERVICE).
        getCharacteristic(UUID_WRIST_TH);
    mNotifQueue.add(characteristic);
    [...]

    // Queue is full of requests => write them on the gatt
    writeCharacteristics();
}
/*****
private void enableMeas(boolean enable){
    [...]

    enableVal = new byte[]{0x01};
    TAndHSettings = new byte[]{0x05,0x00,0x02,0x00};

    // Temperature and Humidity polling delay set to 1 sec
    characteristic = gatt.getService(UUID_ADMS_SERVICE).

        getCharacteristic(UUID_TH_DELAY);
    characteristic.setValue(TAndHSettings);
    mCharQueue.add(characteristic);

    // Enable measurements
    characteristic = gatt.getService(UUID_ADMS_SERVICE).
        getCharacteristic(UUID_ENABLE_MEASURES);
    characteristic.setValue(enableVal);
    mCharQueue.add(characteristic);

    // Queue is full of requests => write them on the gatt
    if(mNotifQueue.isEmpty())
        writeCharacteristics();
}
}
```

Code 6 : Initialisation des paramètres de l'appareil

Dans la fonction *enableNotifications()* on voit une caractéristique mise dans la queue dans l'attente de son traitement. Ensuite on en récupère son descripteur pour activer les notifications.

Le descripteur est un élément d'une caractéristique permettant d'agir sur les notifications et les indications.

De même une queue pour modifier les paramètres a été établie.

Les canevas à suivre pour écrire des paramètres dans les différentes caractéristiques est le suivant :

- On utilise des blocs de 32bits en big-endian séparés en deux parties chacun. Les 16 premiers bits sont pour la valeur et les 16 suivants pour l'unité. Les unités sont définies ainsi :
 - milliseconde : 0x00
 - seconde : 0x01
 - minute : 0x02
 → Exemple : 0x05 00 02 00, signifie 5 secondes
- ADMS_ACCELERO_SETTINGS_CHAR 0x1732:
 Caractéristique composée de trois blocs. Le premier définit la fréquence d'échantillonnage, le deuxième le temps de mesure et le dernier fixe le temps entre deux mesures.
 - Exemple : 0x 64 00 00 00 88 13 00 00 D8 D6 00 00, signifie : échantillonnage à 100 ms, durée de mesure de 5 s, temps d'attente de 55 s
 - 0x 64 00 00 00 05 00 01 00 37 00 01 00, donne le même résultat
- ADMS_TEMP_HUMIDITY_SETTINGS_CHAR 0x1735:
 Comprend un seul bloc permettant de définir la fréquence de mesure
 - Exemple : 0x 05 00 02 00, donne une mesure toute les 5 minutes

Dans le Code 6, le *BroadcastReceiver* sert à récupérer les diffusions faites par le service Bluetooth Low Energy (voir le chapitre 4). On voit quelques exemples des actions qui en découlent. C'est ici qu'on est notifié à chaque fois qu'une caractéristique ou un descripteur a été écrit ou encore que le contenu d'une caractéristique a changé (indiquant qu'un nouveau point a été mesuré).

```
private final BroadcastReceiver mGattUpdateReceiver = new
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        final String action = intent.getAction()

        [...]
        if (BluetoothLeService.ACTION_CHAR_WRITTEN.equals(action)) {
            writeCharacteristics();
        } else if (BluetoothLeService.ACTION_DESCR_WRITTEN.equals(action)) {
            writeCharacteristics();
        } else if (BluetoothLeService.ACTION_CHAR_CHANGED.equals(action)) {
            if (mRecording)
                addDataPoint(
                    intent.getStringExtra(BluetoothLeService.EXTRA_DATA),
                    intent.getStringExtra(BluetoothLeService.EXTRA_UUID));
        }
    }
};
```

Code 7 : Réception des actions

L'ajout d'un point de mesure doit se faire en récupérant la donnée diffusée par le service. Elle est ensuite traitée selon la caractéristique dont elle est originaire. Chaque point sauvé est un String contenant la date de la mesure et sa valeur. On peut voir cela au Code 8.

```
private void addDataPoint(String dataString, String serviceUuid) {

    Date xValue = new Date();

    if(serviceUuid.equals(GattAttributes.ADMS_TEMP_HUMIDITY_CHAR))
    [...]
    String hString = (dataString.substring(0, 8));
    String tString = (dataString.substring(8, 16));
    if(serviceUuid.equals(GattAttributes.ADMS_TEMP_HUMIDITY_WRIST_CHAR))
    {
        mHumStringW = mHumStringW+xValue.toString()+","+hString+";";
        mTempStringW = mTempStringW+xValue.toString()+","+tString+";";
    }
    else[...]
}
else if(serviceUuid.equals(GattAttributes.ADMS_ACCELERO_CHAR))
{[...]
String xString = (dataString.substring(0, 8));
String yString = (dataString.substring(8, 16));
String zString = (dataString.substring(16, 24));
if(serviceUuid.equals(GattAttributes.ADMS_ACCELERO_WRIST_CHAR))
{
    mAccXStringW = mAccXStringW+xValue.toString()+","+xString+";";
    mAccYStringW = mAccYStringW+xValue.toString()+","+yString+";";
    mAccZStringW = mAccZStringW+xValue.toString()+","+zString+";";
}
else [...]
}
}
```

Code 8 : Récupération des données

3.2 CAMERA2 FRAGMENT

Avec les nouvelles versions du système Android, l'API de la camera a été refait, *camera2* remplaçant *camera*. Dans le code d'exemple mis à disposition par Android, tout le processus de la prise de photo est détaillé et découpé en plusieurs phases très spécifiques (prévisualisation, attente de la mise au point, attente de la précapture, etc.).

Notre application ne nécessitant pas toutes les fonctionnalités de l'appareil photo, la classe a été simplifiée et adaptée aux besoins du cas considéré.

De ce fait la capture de l'image advient 3 secondes après que le patient l'ait déclenchée et se fait avec des réglages fixes visibles dans le Code 9. On effectue un réglage automatique et continu de la netteté, désactive le mode d'exposition automatique et on le règle à des valeurs fixes (ISO, ouverture, temps d'exposition), on active le flash à chaque cliché et finalement on désactive la balance des blancs.

```
private void captureStillPicture(CaptureResult result) {
    try {
        [...]

        captureBuilder.set(CaptureRequest.CONTROL_AF_MODE,
            CaptureRequest.CONTROL_AF_MODE_CONTINUOUS_PICTURE);
        captureBuilder.set(CaptureRequest.CONTROL_AE_MODE,
            CaptureRequest.CONTROL_AE_MODE_OFF);
        captureBuilder.set(CaptureRequest.SENSOR_SENSITIVITY,
            ISO_SENSIBILITY);
        captureBuilder.set(CaptureRequest.SENSOR_EXPOSURE_TIME,
            EXPOSURE_TIME);
        captureBuilder.set(CaptureRequest.LENS_APERTURE, APERTURE);
        captureBuilder.set(CaptureRequest.SENSOR_FRAME_DURATION, null);

        captureBuilder.set(CaptureRequest.FLASH_MODE,
            CaptureRequest.FLASH_MODE_SINGLE);

        captureBuilder.set(CaptureRequest.CONTROL_AWB_MODE,
            CaptureRequest.CONTROL_AWB_MODE_OFF);
        captureBuilder.set(CaptureRequest.CONTROL_AWB_LOCK, true);
        [...]
    }
}
```

Code 9 : Réglages de la prise de vue

Dès que la photographie est terminée, l'image est stockée dans la mémoire réservée à l'application. Le caractère privé de la sauvegarde permet de rendre les fichiers inaccessibles pour d'autres applications. Le nom du fichier contient la date de la prise de vue, la rendant unique. Le chemin d'accès est sauvegardé dans une chaîne de caractères spécifique au client.

```
private String saveToInternalStorage(Image image){
    mTimeStamp = String.valueOf(System.currentTimeMillis());
    String filename = mTimeStamp + "pic.jpg";
    ContextWrapper cw = new ContextWrapper
        (getActivity().getApplicationContext());
    File directory = cw.getDir("images", Context.MODE_PRIVATE);
    File mFile = new File(directory, filename);

    ByteBuffer buffer = image.getPlanes()[0].getBuffer();
    byte[] bytes = new byte[buffer.remaining()];
    buffer.get(bytes);

    FileOutputStream output = null;
    try{
        output = new FileOutputStream(mFile);
        output.write(bytes);
        [...]
    }
    return mFile.getAbsolutePath();
}
```

Code 10 : Sauvegarde de l'image

4 BLUETOOTH LE SERVICE

Cette classe a été prise des exemples d'Android, et n'a pas été modifiée. Elle gère la connexion avec le Gatt (voir Figure 2) de l'appareil. C'est ici que sont diffusés les différents états de la connexion et des caractéristiques. Un exemple en est donné avec le Code 11.

```

@Override
public void onCharacteristicChanged(BluetoothGatt gatt,
    BluetoothGattCharacteristic characteristic)
{
    broadcastUpdate(ACTION_CHAR_CHANGED, characteristic);
}
    
```

Code 11 : Exemple de diffusion

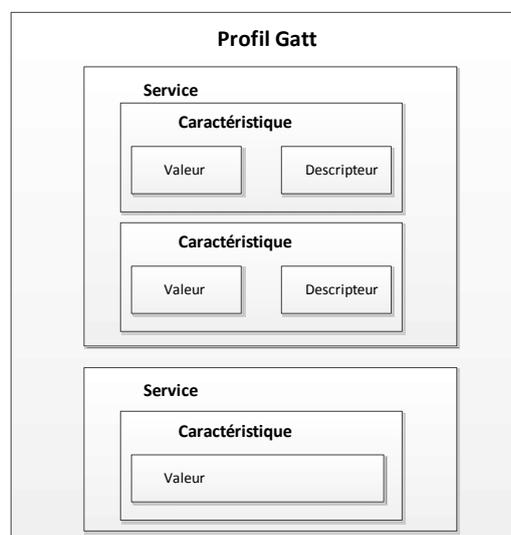


Figure 2 : Contenu du Gatt

5 RECORD DETAIL ACTIVITY

Cette activité affiche les résultats des valeurs mesurées durant les enregistrements, sous forme de graphiques temporels et aussi avec une section affichant des détails (extrémums et moyenne). On fait usage de la librairie open source nommée Graphview pour le dessin des graphes.

Le Code 12 montre toute la démarche faite pour y parvenir.

On crée premièrement chaque liste qui contiendra les séries de points pour chaque type de mesure. Ensuite chaque point de mesure est retiré du *SharedPreferences* afin d'être ajouté dans sa série. L'extraction des informations à partir de String est quelque peu laborieuse, mais fonctionne bien. Rien de compliqué n'est à faire pour les mesures d'humidité ou de température. Les séries peuvent ensuite être ajoutées à la vue correspondant au type de mesure choisi. On peut agir sur le format des axes, des étiquettes, des couleurs des séries, etc.

L'affichage des données des accéléromètres est plus compliqué vu la mesure faite. Il a été nécessaire d'utiliser un vecteur à 4 dimensions pour retrouver l'information intéressante. En effet ce qui doit être extrait est la moyenne de la norme des points mesurés chaque tant de temps. Par

exemple admettons que les mesures soient faites toutes les minutes pendant 5 secondes chaque 200ms. Cela fait un nombre indéterminé de groupes de 25 points eux-mêmes constitués de 3 composantes dimensionnelles, d'où l'utilisation d'un vecteur de cette taille.

```

LineGraphSeries<DataPoint>wHSerie= new LineGraphSeries<DataPoint>();

    [...]

    DataPoint dp = new DataPoint(i, dataVal);
    switch (sensor) {
        case 0:
            wHSerie.appendData(dp, true, 86400);
            break;

        [...]

    }

    [...]
    // 2x3 axes for the accelerometers
    Vector<Vector<Vector<Double>>> accData = new
        Vector<Vector<Vector<Double>>>(6);
    [...]
    // Compute the norm of each point of a bunch for the three axes
    squareSumF = Math.pow(accData.get(axe+3).get(bch).get(pt), 2);
    [...]
    // Compute the average of each bunch
    wristAvg.add(getAvg(normW.get(bch)));
    fingerAvg.add(getAvg(normF.get(bch)));
    [...]
    if(sectionNb == 1) {
        switch (mItemId) {
            case 0:
                wHSerie.setTitle("Wrist");
                hHSerie.setTitle("Hand");
                hHSerie.setColor(Color.RED);
                graph.addSeries(wHSerie);
                graph.addSeries(hHSerie);
                graph.setTitle("Humidity");
                graph.getGridLabelRenderer()
                    .setLabelFormatter(new HumidityLabel());
                break;
            [...]
        }
        graph.getGridLabelRenderer().setNumHorizontalLabels(5);
        graph.getLegendRenderer().setVisible(true);
        graph.getLegendRenderer()
            .setAlign(LegendRenderer.LegendAlign.TOP);
        graph.getViewport().setScalable(true);
    }

    [...]
}
    
```

Code 12 : Formatage des points de mesure et affichage graphique

Les données envoyées par le Bluetooth suivent le standard ISO de type *Float IEEE11073* et doivent donc être reconverties pour être lisibles. Le Code 13 est utilisé pour effectuer cette conversion.

```
private static double ieee11073ToDouble(String dataString) {
    double result = 0;
    long mantissa;
    int exponent;
    if(dataString.length() == 8)//32bit format
    {
        exponent = Integer.parseInt(dataString.substring(6), 16);
        if(exponent > 127)
            exponent = exponent-256;
        dataString = dataString.substring(0,6)+"00";
        mantissa = swapEndian(Long.parseLong(dataString, 16));
        result = (double) ((double)mantissa *(Math.pow(10,exponent)));
    }
    return result;
}

private static long swapEndian(long i) {
    return (i&0xff)<<24 | (i&0xff00)<<8 | (i&0xff0000)>>8 | (i>>24)&0xff;
}
```

Code 13 : Conversion des valeurs

6 PIC SELECT ACTIVITY

Les photos peuvent être affichées avec une animation de défilement grâce à un `FragmentPagerAdapter` (voir Code 14). On y ajoute le nombre de pages correspondant au nombre de photos prises par le patient afin d'avoir un rendu intuitif et agréable.

```
public class SectionsPagerAdapter extends FragmentPagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return PlaceholderFragment.newInstance(position + 1);
    }

    [...]
}
```

Code 14 : Programmation de sections dynamiques

Le segment de code présenté ci-dessous (voir Code 15) assure l'affichage des images. `mPitures` est un tableau contenant tous les chemins d'accès aux photos du patient. Ils sont ensuite récupérés indépendamment, les fichiers correspondants sont ouverts, les données converties grâce à la classe `BitmapFactory` et finalement affichées dans la vue du fragment.

```
if(mPictures != null) {
    String imagePath = mPictures[sectionNb-1];

    String[] temp = imagePath.split("/");
    String tempStr = temp[temp.length-1];
    long d = Long.parseLong(tempStr.substring(0,tempStr.length()-7));
    Date date = new Date(d);
    String title = "Picture taken " +(date.toString());

    picLabel.setText(title);

    File f=new File(imagePath);
    if(f.exists()){
        Bitmap myBitmap = BitmapFactory.decodeFile(f.getAbsolutePath());
        myBitmap.setDensity(480);
        ImageView imageView = (ImageView) view.findViewById(R.id.imgPicker);
        imageView.setImageBitmap(myBitmap);
        imageView.setRotation((float) 270);
    }
}
else
    picLabel.setText("No pictures taken...");
```

Code 15 : Affichage des photos

7 PAQUET GRAPHVIEW

Étant donné que la gestion des dates n'est plus fonctionnelle avec la librairie graphview, le problème a été contourné en créant des *Labels* personnalisés. Cela permet de formater la notation des deux axes. Dans l'exemple du Code 16, la mesure d'humidité intervenant toutes les trois secondes, l'affichage des secondes est inutile; ceux des minutes et des heures doivent être adaptés d'un facteur 3.

La programmation de ce formatage n'a pas été optimisée, c'est-à-dire qu'il ne changera pas si des nouveaux paramètres de mesure sont fixés dans les capteurs.

Cheque type de mesure à son Label spécifique.

```

public class HumdityLabel extends DefaultLabelFormatter{
    @Override
    public String formatLabel(double value, boolean isValueX) {
        if (isValueX) {
            // show x as a time
            if(value == 0)
                return super.formatLabel(value, isValueX);
            else if(value < 12)
                return super.formatLabel(value*5, isValueX)+" min";
            else
                return super.formatLabel(value/12, isValueX)+" hours";
        } else {
            // show percents for y values
            return super.formatLabel(value, isValueX) + " %";
        }
    }
}
    
```

Code 16 : Formatage des axes

8 PAQUET SQLITE

Le paquet sqlite est lui-même composé selon la hiérarchie suivante (voir Figure...) :

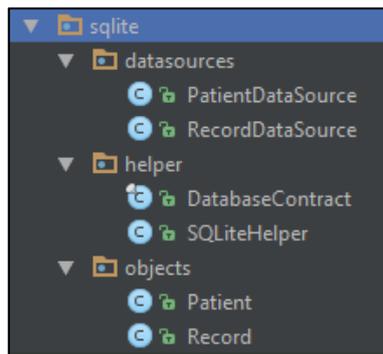


Figure 3 : Paquet sqlite

Helper est un paquet qui gère la base de données SQL, constituée par plusieurs tables de patients et d'enregistrements.

Le paquet *datasources* contient deux classes qui prennent en charge la manipulation de la table SQL pour en extraire les données correspondantes soit au patient soit à l'enregistrement. Elles contiennent par exemple des méthodes telles que *createPatient(...)*, *deleteRecord(...)*, *upDatePatients(...)*, *getRecordbyId(...)*, etc.

Enfin, le paquet *objects* rassemble les deux classes contenant les données du patient, respectivement de l'enregistrement. C'est là que sont sauveés les toutes les clés permettant de trouver les valeurs dans la *SharedPreferences* lors de la restitution des résultats d'un enregistrement.

INDEX OF PAGES FOR ALGODYSTROPHY SYSTEM

NR	NAME	DESCRIPTION	COMMENTS
1	title	this page	describe the schematic sheets
2	power	power supply control user input/output	battery, power switch user switch, user led
3	sensors	all system sensors	the sensors's interface and the external connector
4	bluetooth	bluetooth parts	all bluetooth parts (module, antenna, prog. connector)
5			
6			

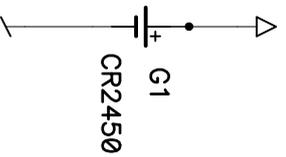
SPECIAL GENERAL COMMENTS

all resistors are 1% tolerance (unless otherwise noted)

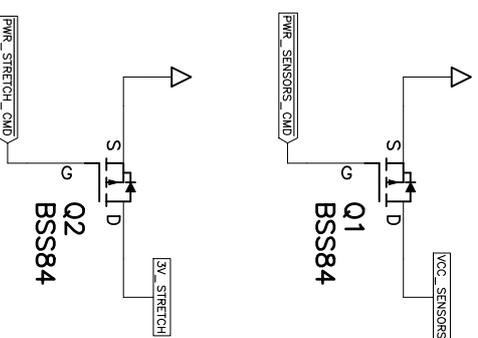
REVISION: 1.1

Algodystrophy strap system		DES 30.03.15
title		REV V1.1
HAUTE ECOLE VALAISANNE		1/4 bracelet_v11.sch

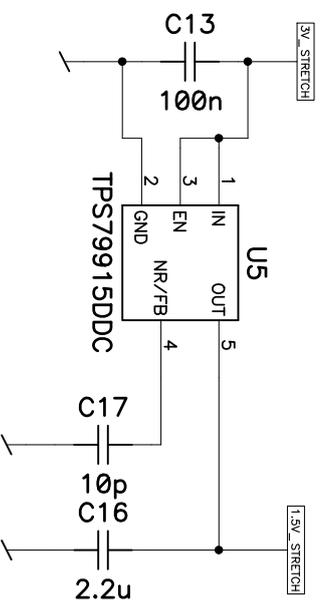
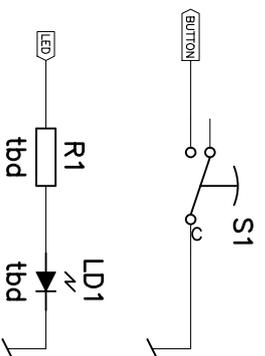
Battery



Sensors power



User input / output



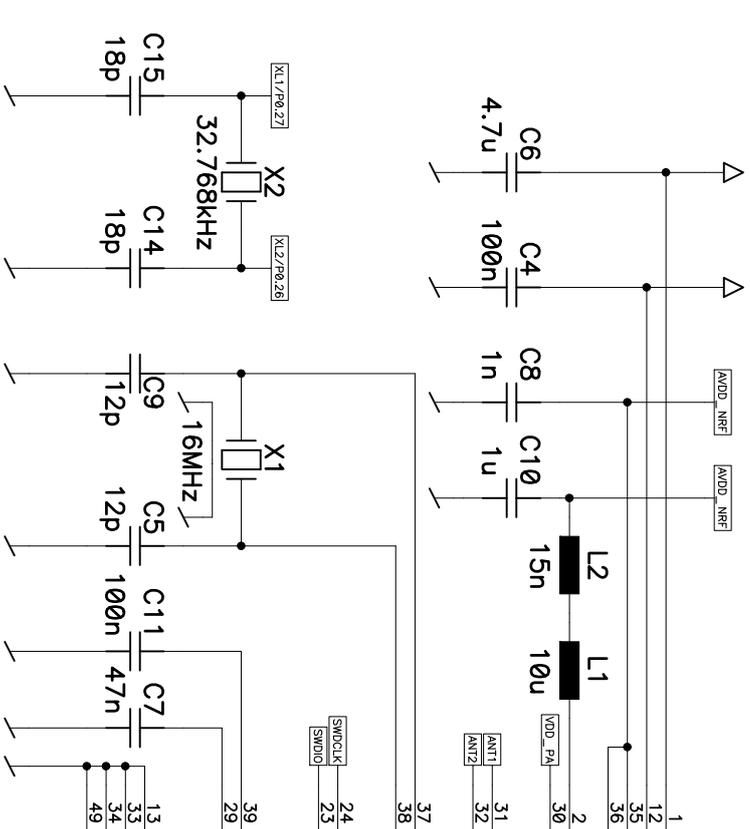
Algodystrophy
strap system
HAUTE ECOLE VALAISANNE

power

DES 30.03.15

REV V1.1

2/4 bracelet_v11.sch



Bluetooth

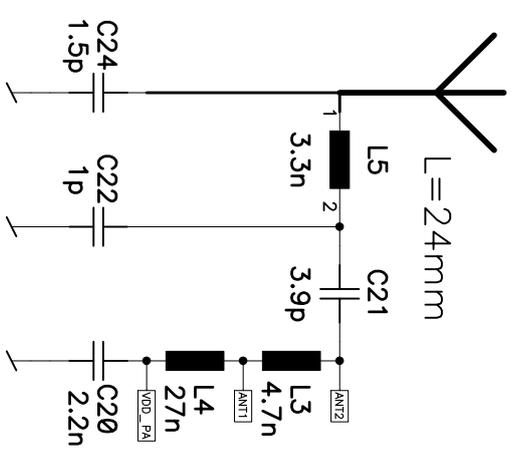
U3

P0.00	4	X	
P0.01	5	X	
P0.02	6		SCK
P0.03	7		MOSI
P0.04	8		MISO
P0.05	9	X	
P0.06	10	X	YSTRETCH
P0.07	11	X	
P0.08	14		CS_HXT1
P0.09	15		CS_HXT0
P0.10	16		CS_ACC1
P0.11	17		CS_ACC0
P0.12	18	X	
P0.13	19	X	
P0.14	20	X	
P0.15	21	X	
P0.16	22		BT_DEBUG_RX
P0.17	25		BT_DEBUG_TX
P0.18	26	X	BT_DEBUG_RX
P0.19	27		BT_DEBUG_TX
P0.20	28		BT_DEBUG_RX
P0.21	40		BT_DEBUG_TX
P0.22	41	X	
P0.23	42	X	
P0.24	43	X	
P0.25	44	X	
P0.26	45		
P0.27	46		
P0.28	47		
P0.29	48		
P0.30	3	X	

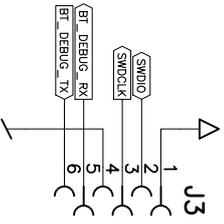
39	DEC1		
29	DEC2		
24	SWDCLK		
23	SWDIO		
13	VSS		
33	VSS		
34	VSS		
49	EPAD		
39	DEC1		
29	DEC2		
24	SWDCLK		
23	SWDIO		
13	VSS		
33	VSS		
34	VSS		
49	EPAD		

nRF51822 Rev3

RF antenna



Programming



Algodystrophy strap system	DES	30.03.15
	REV	V1.1
HAUTE ECOLE VALAISANNE	4/4	brocelet_v11.sch

Annexe 4 - Liste de composants

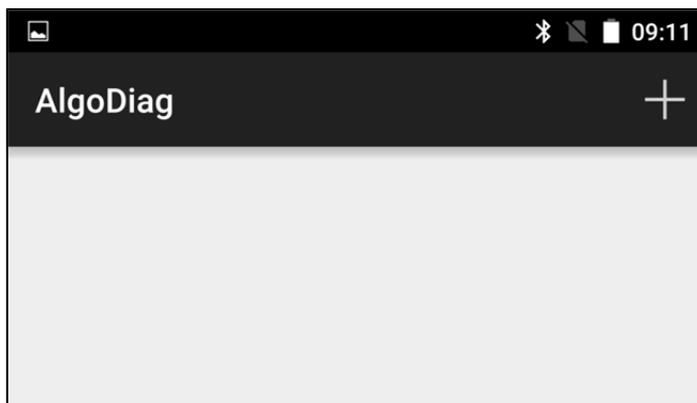
#	Mouser	Farnell	Distrelec	Référence	qté	Dénomination	prix/u	Total
1				Rutronik BTOOTH1325	1	nRF51822 Rev3	fr. 2.18	fr. 2.18
2	x			234-6876	1	Condensateur 4.7uF	fr. 0.43	fr. 0.43
3	x			247-0461	6	Condensateur 100nF 0603	fr. 0.11	fr. 0.67
	x			810-C1005X5R1A104K	2	Condensateur 100nF 0402	fr. 0.10	fr. 0.20
4	x			175-8911	2	Condensateur 1nF	fr. 0.01	fr. 0.02
5	x			175-9056	2	Condensateur 18pF	fr. 0.01	fr. 0.02
6		x		831814	2	Condensateur 12pF	fr. 0.02	fr. 0.05
7	x			77-VJ0402V473ZXCBC	1	Condensateur 47nF	fr. 0.06	fr. 0.06
8		x		831808	1	Condensateur 3.9pF	fr. 0.02	fr. 0.02
9		x		221-0768	1	Condensateur 1.5pF	fr. 0.09	fr. 0.09
10		x		221-8842	1	Condensateur 1pF	fr. 0.04	fr. 0.04
11		x		221-0834	1	Condensateur 2.2nF	fr. 0.11	fr. 0.11
12	x			963-HK160815NK-T	1	Inductance 15nH	fr. 0.11	fr. 0.11
13	x			963-LBMF1608T100K	1	Inductance 10uH	fr. 0.19	fr. 0.19
14		x		134-3068	1	Inductance 3.3nH	fr. 0.06	fr. 0.06
15		x		171-1735	1	Inductance 4.7nH	fr. 0.03	fr. 0.03
16		x		151-5383	1	Inductance 27nH	fr. 0.15	fr. 0.15
17				Rutronik QUARZ5910	1	Quartz 16MHz	fr. 0.28	fr. 0.28
18	x			815-ABS07-32.7K-1-T	1	Quartz 32.768 kHz	fr. 1.28	fr. 1.28
19			x	121631	1	Connecteur de debug/programmation	fr. 2.05	fr. 2.05
20			x	970877	1	Batterie 3V CR2450	fr. 1.60	fr. 1.60
21			x	973583	1	Support de batterie	fr. 0.56	fr. 0.56
22		x		242-9857	1	Button poussoir	fr. 3.50	fr. 3.50
23		x		233-1716	1	Résistance 180Ω	fr. 0.02	fr. 0.02
24		x		211-2098	1	Led jaune	fr. 0.27	fr. 0.27
25		x		105-6526	2	Mosfet BSS84	fr. 0.11	fr. 0.23
26		x		118-0158	1	Convertisseur de tension TPS79915DDC	fr. 0.69	fr. 0.69
27		x		175-9053	1	Condensateur 10pF	fr. 0.02	fr. 0.02
28		x		221-1185	1	Condensateur 2.2uF	fr. 0.20	fr. 0.20
29			x	121633	1	Connecteur 14 pôles	fr. 2.85	fr. 2.85
30	x			912-KXTIA-1006	2	Accelerometre	fr. 4.11	fr. 8.22
31	x			785-HIH-7131-000-001	2	Capteur d'humidité	fr. 15.49	fr. 30.98
32		x		234-6906	2	Condensateur 220nF	fr. 0.13	fr. 0.26
33		x		190-3402	1	Ampli. d'instrumentation INA333AIDGKR	fr. 4.90	fr. 4.90
34		x		933-0780	2	Résistance 200kΩ	fr. 0.01	fr. 0.03
35		x		117-0926	1	Résistance 25kΩ	fr. 0.01	fr. 0.01
36		x		933-0755	1	Résistance 200Ω	fr. 0.01	fr. 0.01
37				Adafruit Article #519	1	Capteur d'étirement	fr. 9.39	fr. 9.39

TOTAL fr. 71.78

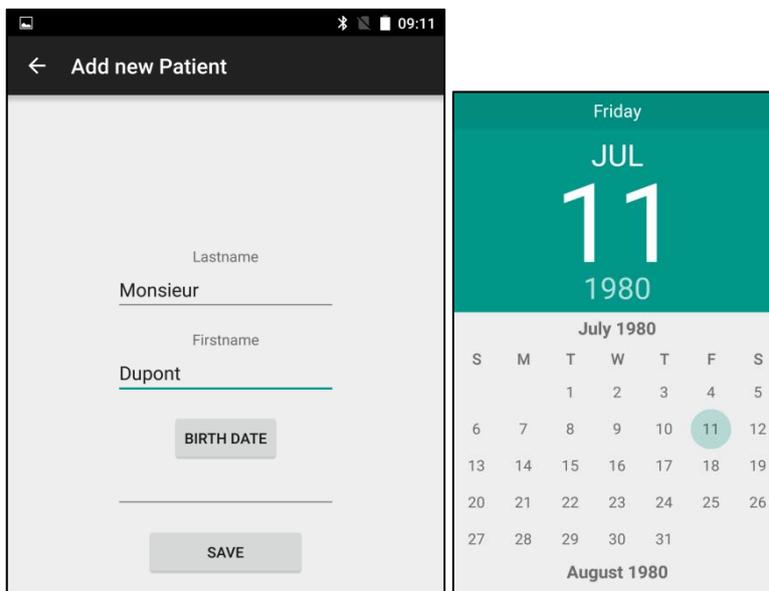
ANNEXE 5

MODE D'EMPLOI

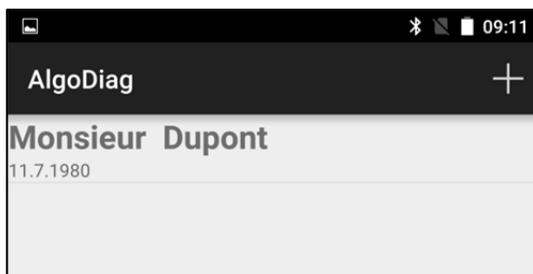
La première page de l'application montre une vue de la liste des patients. Pour ajouter un patient appuyer sur l'icône .



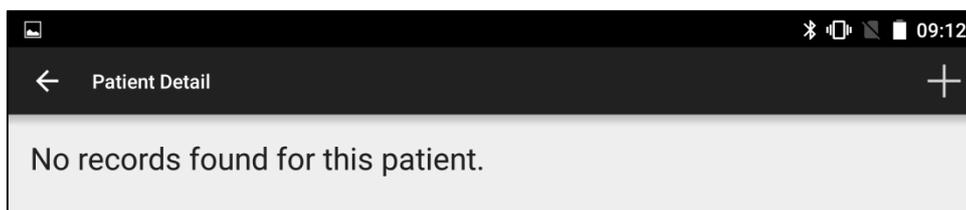
Cela ouvre une nouvelle fenêtre permettant de s'enregistrer. Entrer votre nom, votre prénom et appuyer ensuite sur le bouton "Birthdate" pour choisir la date de naissance. Pour sélectionner la bonne année appuyer sur l'année (1980 sur l'image ci-dessous).



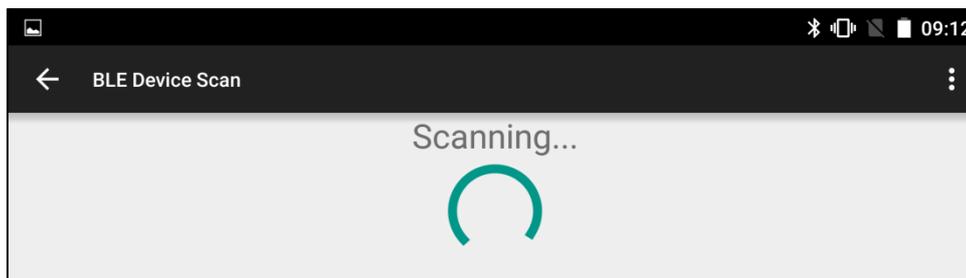
Le patient a été ajouté dans la liste. La suppression d'un patient peut se faire en appuyant de manière prolongée sur le nom concerné. Cela ouvre une icône prévue pour la suppression dans le menu.



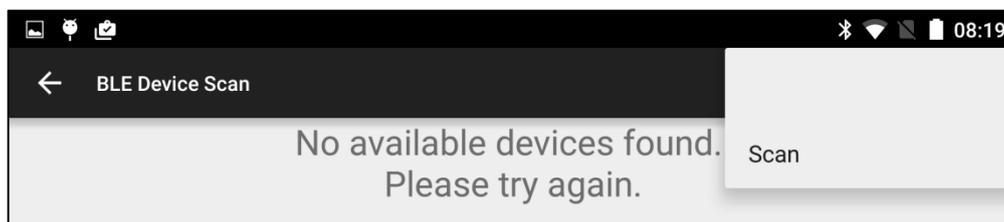
Pour ajouter un nouvel enregistrement appuyer sur l'icône



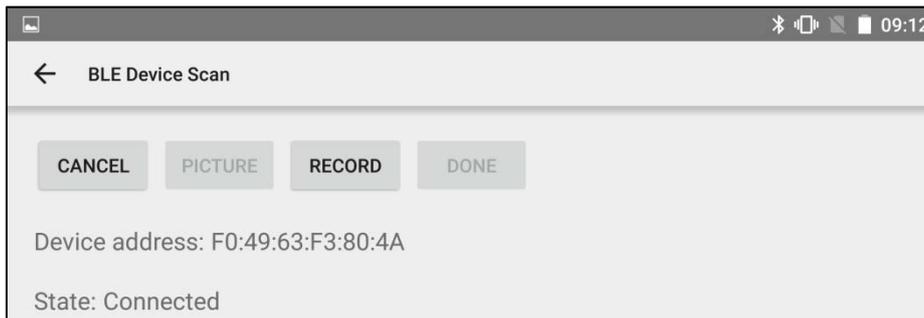
L'appareil va être scanné et se connecter automatiquement



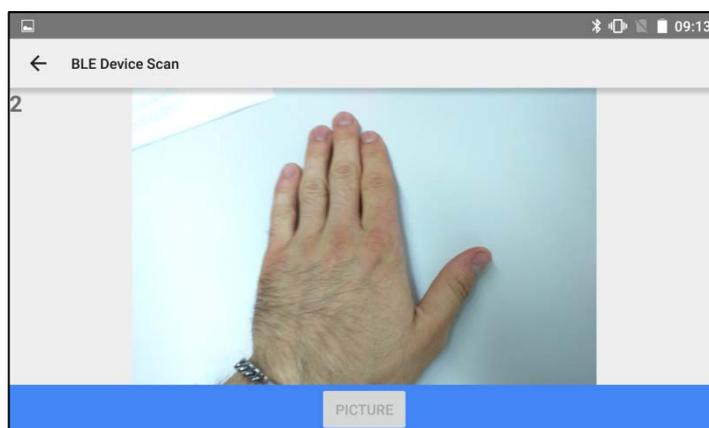
Si aucun appareil n'est trouvé, appuyer le bouton dépassant du boîtier de l'appareil. Ensuite appuyer sur le menu et appuyer sur "Scan" pour relancer le processus.



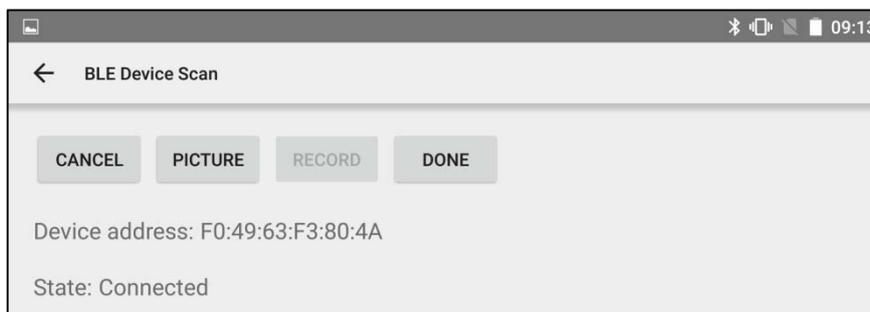
La fenêtre pour ajouter un enregistrement est maintenant visible. Appuyer sur le bouton "Record" pour commencer.



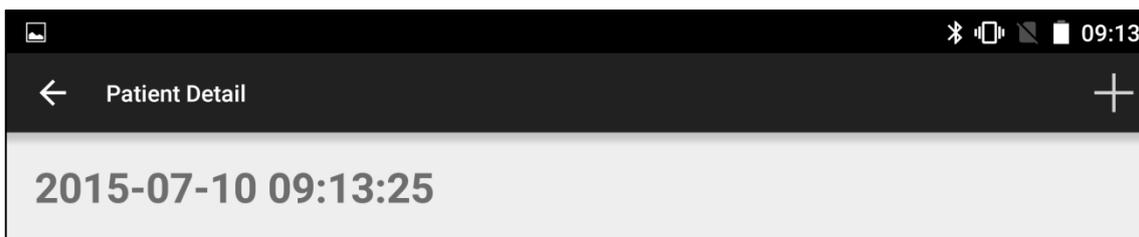
L'enregistrement se déroule. Pour faire une photo pendant les mesures appuyer sur le bouton "Picture". Cela ouvre l'appareil photo. Placer le smartphone et appuyer sur "Picture" pour amorcer le cliché. Après trois secondes la photo sera prise. S'assurer que les deux mains seront bien visibles sur la photo.



Appuyer sur le bouton "Done" lorsque les mesures doivent être arrêtées.



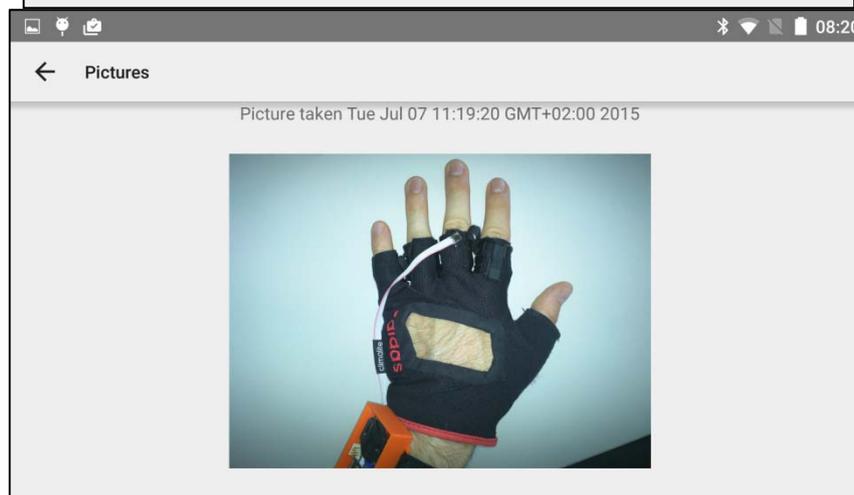
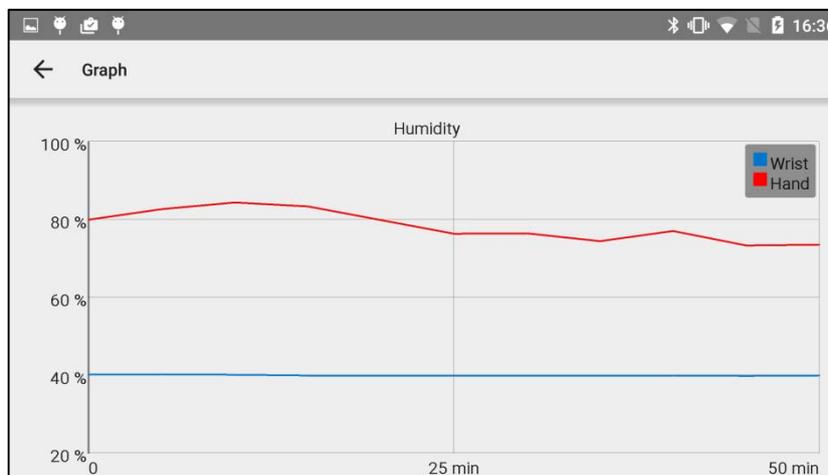
Le nouvel enregistrement a été ajouté. Il peut être supprimé de la même manière que pour un patient. Pour visualiser les résultats appuyer sur l'enregistrement intéressant.



Cela ouvre une liste des mesures pouvant être affichées.



Appuyer sur une des valeurs pour afficher le graphe correspondant ou pour visualiser les photographies prises pendant les mesures.



Glisser le doigt sur les côtés pour passer d'une photo à l'autre ou pour voir les détails de la mesure. Le graphe peut être agrandi en effectuant un mouvement de pincement des doigts. Le défilement se fait avec des mouvements en diagonale.