

**Laboratoire
ESYCOM**



ESIEE
PARIS



RAPPORT DE STAGE TECHNICIEN

Hugo CARANGEOT ©Année Universitaire 2022-2023

Table des matières

I - Préface	3
II - Institut de recherche : le Laboratoire ESYCOM, UMR 9007	
II - a) Laboratoire ESYCOM : qu'est-ce ?	4
II - b) L'installation progressive du Laboratoire ESYCOM	5
III - Mon stage	
III - a) Rappels sur l'intérêt du stage	6
III - b) Projet ISafe	7
III - c) Inventaire Matériel et Humain	8
III - d) Mes missions	8
IV - Le projet : Participation au développement d'un bracelet connecté	
IV - a) Prise en main	9
IV - b) Connectique	10
IV - c) Code	15
IV - d) Réalisation	
IV - d.1) Matériel	19
IV - d.2) Bracelet connecté et son fonctionnement	24
IV - e) Comparaison	26
IV - f) Limites et Ouverture	28
V - Conclusion	29
VI - Annexes	30

I - PRÉFACE

UN STAGE : POUR QUOI FAIRE ?

Dans le cadre de mes études en école d'ingénieurs à l'**ESIEE PARIS**, je dois réaliser un stage technicien en entreprise. Contrairement au stage de fin de première année, ce stage (E3S1P1) est l'occasion parfaite pour voir **l'ingénieur dans le monde professionnel**. De ce fait, on se rend compte de ce que sont réellement les **attentes** et les **obligations** données par un employeur pour ce vaste métier.

OBJECTIFS DU LIVRET

Ce livret est un rendu de début de première année du cycle ingénieur. Il **retranscrit** une partie de l'**expérience** que j'ai pu acquérir, mais il permet également de partager mon ressenti quant au projet que j'ai pu développer (19 septembre 2022 - 5 novembre 2022, 7 semaines).

Enfin, ce rapport sert également de trace écrite concernant l'avancement de mon projet personnel et professionnel en cet automne 2022.

REMERCIEMENTS :

J'aimerais remercier tout particulièrement ma tutrice de stage, **Gaëlle LISSORGUES**, pour cette proposition de stage et pour sa précieuse aide. Je souhaiterais également remercier **Julien PAGAZANI** et **Carlos PINTO**, qui ont tous deux permis le bon déroulement du projet (respectivement apport de matériel et fabrication de circuits imprimés). J'ai chaleureusement été accueilli par chacun d'entre eux et ce fut très plaisant et enrichissant.

DÉCLARATION ANTI-PLAGIAT

Ce rapport n'est que le fruit de mon propre vécu, de ma pensée ainsi que de mes recherches. Il n'y a donc en aucun cas du plagiat. Je l'ai également entièrement réalisé et conceptualisé grâce à : [Canva.com](#).

Vous pourrez retrouver les informations utilisées pour ce rapport sur internet (voir cf plus bas). Toutes les recherches ont été faites durant la période de mon stage.

Hugo CARANGEOT.

II - Institut de recherche : le Laboratoire ESYCOM, UMR 9007

II - a) Laboratoire ESYCOM, qu'est-ce ?

ESYCOM est un **institut de recherche** français signifiant 'Laboratoire d'électronique, systèmes de communication et microsystèmes' de code 'UMR9007'. Ce dernier, dont le travail est effectué par des enseignants-chercheurs (et des chercheurs) provenant des quatre établissements suivants : CNAM, ESIEE Paris, Université Gustave-Eiffel et CNRS. Le projet que porte le laboratoire est le suivant : « Capteurs et systèmes communicants pour la ville, l'environnement et la personne » et ce projet est scindé en 3 grands thèmes fédérateurs appartenant à l'ingénierie :

- systèmes de communication,
- micro-capteurs,
- micro-énergie.

On dénombre ainsi pas moins de 75 personnes travaillant pour cet organisme en mai 2022.



Logo du laboratoire ESYCOM (2022)



Ce laboratoire se situe dans l'**est Parisien**, il fait partie de la **cité Descartes**. Il dispose également d'un **site internet** :

<https://esycom.cnrs.fr/le-laboratoire/>

et de différents réseaux sociaux pour communiquer.

II - b) L'installation progressive du Laboratoire ESYCOM.



III - Mon Stage

III - a) Rappel sur l'intérêt du stage

Comme dit précédemment, ce stage est effectué en **début de troisième année**, c'est une alternative au départ groupé à l'international. Ainsi, il nous permet [les étudiants] d'acquérir une certaine **expérience professionnelle, d'explorer différents horizons**, tant sur le plan "type d'entreprise", que sur le plan du "domaine d'application".

Nonobstant, ce dernier doit respecter certaines conditions. En effet, il s'agit ici d'un **stage technicien**, il vise donc :

"

- la mise en œuvre ou l'acquisition de connaissances ou de compétences dans un contexte professionnel d'un domaine relevant de la formation du premier cycle ESIEE Paris ;
- le développement des méthodes de travail : travail en équipe, communication, recherche de l'information, auto-formation, rédaction de rapport technique, etc ;
- la connaissance du monde de l'entreprise et une bonne compréhension de son fonctionnement.

Enfin, à l'issue du stage, nous devons avoir :

- produit une réalisation technique, matérielle ou logicielle
 - ou
- pris en main des méthodes ou outils techniques que vous aurez utilisés dans un contexte vous permettant d'acquérir des compétences dans un domaine relevant de votre formation d'ingénieur

(cf : *Présentation aux étudiants du stage technicien de fin de E2, blackboard ESIEE PARIS*)

C'est pourquoi, la possibilité de faire un stage dans des entreprises de domaines différents n'est pas un problème, il ne faut que faire valider un **sujet de stage** auprès de l'établissement scolaire avant de débuter (par exemple, carrefour peut être l'entreprise d'accueil avec un sujet comme "création d'un site internet").

Voici donc mon thème de stage :

"Participation au développement d'un bracelet connecté.",

pour le laboratoire ESYCOM, avec un bureau au sein de l'ESIEE PARIS.

III - b) Projet ISafe

Premièrement, il est primordial de **contextualiser** mon stage. J'ai, pour le développement du bracelet connecté, repris deux prototypes qui ont été faits par un étudiant de 5ème année lors de son stage de fin d'études.

Cet étudiant travaillait sur le **projet ISafe-VH** (*virtual human*, l'homme virtuel).

Pour résumer succinctement, **ISafe-VH** est un projet européen qui a **débuté le 31 mars 2019** et qui s'est **conclu le 31 mars 2022 (3ans)**.



images associées au projet ISafe-VH

"

ISafe consiste à développer un **système d'alerte et d'optimisation des secours** pour les **victimes d'accidents de la route**. Développé en collaboration avec les chercheurs du département transport Santé Sécurité de l'Ifsttar (Laboratoires LBA, LBMC, LMA et UMRESTTE), l'Université de Strasbourg, des médecins et des sapeurs-pompiers de l'ENSOSP, ce système s'appuie sur un ensemble de briques technologiques telles que des enregistreurs embarqués dans le véhicule afin de transmettre des informations sur les conditions du choc. Analysées à distance et en temps réel à l'aide de simulations numériques sur des corps humain virtuels, ces données contribueront à prédire les blessures potentielles et le niveau d'urgence afin d'orienter les décisions des acteurs de la chaîne de secours.

"
(cf : <https://urbarisklab.org/fr/projets/isafe-virtual-human/>)

Quel intérêt porte donc le bracelet connecté développé par ESYCOM au projet ?

Ici, l'intérêt du bracelet connecté, est de **récolter automatiquement** des données vitales (comme le battement par minute du cœur, la SpO2 : saturation de l'hémoglobine en oxygène par oxymétrie de pouls, la température corporelle, etc {à l'aide de différents capteurs qu'on intègre au dispositif}) **sans impacter le travail des urgences**. De plus, il doit **afficher ces données en temps réel** et de manière embarquée. Il doit également assurer une **sauvegarde de ces dernières** dans l'optique de créer plus tard un algorithme de prédition sur l'évolution de l'état du patient.

III - c) Inventaire Matériel et Humain

Pour faire l'**inventaire matériel**, j'avais en arrivant à ma disposition, deux solutions à l'état de prototype développées par mon prédecesseur : une partant d'un "**Arduino Nano 33 BLE Sense**" et une se basant sur un "**Wio Terminal**". *Note : Pour un objet connecté, il faut un processeur. Ici, j'ai donc deux processeurs différents avec des capteurs et des fonctionnalités différentes.*

De surcroît, je possédais des **codes** dans le **langage de code Arduino** permettant aux processeurs de récolter et traiter les données des capteurs. On m'a également fourni des **documents explicatifs** sur le projet et sur tout l'**aspect médical** : comment interpréter les valeurs...

Enfin, on m'a confié des **locaux** (bureau, laboratoires) et du **matériel** avec une utilité allant de la **création de circuits électroniques** à sa **vérification** (poste à souder, composants, montre médicale commercialisée, multimètre...).

Du **côté humain**, je travaillais à **90% du temps seul**, en totale autonomie. Je n'avais pas "réellement" de collègue avec qui travailler, qui aurait pu m'expliquer ou m'apprendre comment cela fonctionne. C'est donc au travers de l'**autoformation** que j'ai su me construire. J'ai, pour ce fait, passé énormément de temps à faire des recherches sur internet (blogs, vidéos...) et à l'expérimentation, mais cela me convenait personnellement très bien. Néanmoins, madame LISSORGUES me demandait régulièrement des points d'avancement et a pu me mettre en relation avec des personnes pouvant m'aider lorsque j'en avais besoin. J'ai donc tout de même pu bénéficier de l'expertise de ces dernières, comme notamment avec monsieur PINTO concernant la fabrication de PCB ou encore avec les anciens membres du projet pour mieux le comprendre en faisant une vidéo-conférence.

En plus de cela, ma tutrice étant très impliquée, m'a donné son point de vu sur comment avancer lors des **multiples réunions hebdomadaires** que nous nous étions fixées.

III - d) Mes missions

Pour finir cette brève introduction, elle m'a confié un sommaire des tâches à réaliser dès le début :

- prise en main du prototype existant,
- fabrication de nouveaux PCB et tests,
- programmation Arduino,
- campagne de mesures comparatives entre le prototype/montre commerciale.

J'avais comme objectif principal de prouver au travers d'un prototype que j'aurais développé et conçu, qu'il est possible d'avoir un bracelet médical avec l'intérêt vu plus haut à moindre coût.

Cela m'a permis de commencer avec un **fil conducteur** ce stage et ainsi ne pas débarquer sans repère dans un vaste projet.

IV - Le projet : Participation au développement d'un bracelet connecté

Pour décrire de manière simple, concise ainsi que le plus fidèlement possible mon ressenti, j'ai choisi de scinder mon écrit en plusieurs parties bien qu'en réalité, certaines avançait simultanément puisque complémentaires (ex : le code va avec les connectiques).

IV - a) Prise en main

Pour commencer, je suis un élève qui sort tout juste du **cycle préparatoire intégré**. Par conséquent, je n'ai que peu de compétences pratiques et très peu de théorie par rapport au contenu d'une **filière système embarqué**. Hors, ce projet est tout à fait une application qui découle de cette dernière.

C'est pourquoi, lors de ma première semaine je n'ai que seulement lu et compris les documents, comment cela fonctionnait électroniquement, mais également informatiquement. J'ai aussi dû comprendre le raisonnement de l'élève précédent, le matériel, ce qu'il avait fabriqué... J'ai donc assimilé un bon nombre d'informations, que je pourrais regrouper en **trois catégories** comme suit :

- Connaissances théoriques en électronique, informatique (capteurs, codes, etc),
- Connaissances théoriques sur l'aspect médical,
- Avancement du projet récupéré (où cela en est ? pourquoi est-ce conçu de cette façon ?...)

A la fin de cette première semaine, j'ai fait un **état d'avancement global**. J'entends par là que je me suis résumé ce qui était fait ou était à faire.

- Qu'est qui selon moi fait défaut ?
- Quels sont les problèmes déjà identifiés à résoudre ?
- Sur quelle solution se pencher au niveau processeur ?

J'ai également pris en compte les **contraintes** que j'avais. Le matériel étant déjà acheté, il était préférable de le réutiliser, les codes étant en Arduino, il fallait privilégier ce langage...

J'ai également créé un document .docs me permettant de retranscrire chaque jour, si j'en ressentais le besoin, mon avancement, mes problèmes ou encore les questions que je souhaitais poser. Finalement, il me servait un peu de pense-bête mais également de support pour présenter mon travail. Je l'ai partagé à ma tutrice de stage, ce qui laissait une supervision aisée du projet possible à distance.

IV - b) Connectique

Pour rappel, ce bracelet connecté à usage (prévu) médical doit prélever tout un tas de **paramètres physiologiques**, les **afficher** ainsi que les **sauvegarder** (en SD). Pour cela, le bracelet a besoin de plusieurs choses : un **processeur**, un **port de carte SD**, un **écran** (ou une application tierce qui affiche sur un téléphone mobile) et des **capteurs** pour les différents paramètres. Il faut **connecter** tout cela, c'est pourquoi il est nécessaire de réfléchir dans un premier temps à laquelle des deux solutions à l'état de prototype que j'ai récupéré serait la plus pertinente pour répondre à ça.

J'ai dans le cahier des charges un tableau qui fait correspondre un type de capteurs aux données à récupérer (voir ci-dessous).

Capteur	Paramètres mesurés
Thermomètre	Température
Photopléthysmographe	Rythme cardiaque et variabilité Saturation pulsée en dioxygène À investiguer: rythme respiratoire et onde de pouls
Accéléromètre	État du blessé (coma, état calme ou agité, convulsions, tremblements) Mouvement du véhicule de transport

Extrait du cahier des charges

Voici ce que les processeurs, tous deux codables en langage Arduino, proposent :

Tableau comparatif entre Arduino Nano 33 BLE Sense et WIO Terminal



Arduino Nano
33 BLE Sense



WIO Terminal

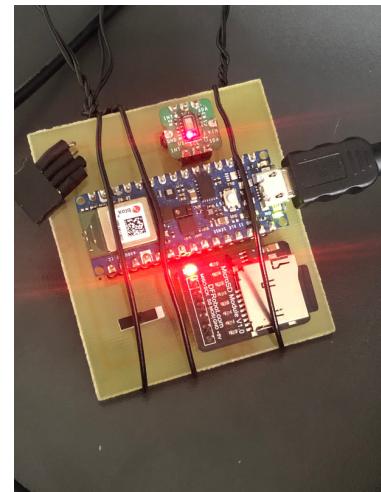
	Arduino Nano 33 BLE Sense	WIO Terminal
Dimensions	45x18mm!(hauteur variable)	72 x 57 x 12 mm wio
Bluetooth	(✓)	(✓)
Wifi	(✓)	(✓)
Accéléromètre	(✓)	(✓)
Température corporelle	(✗)	(✗)
Température ambiante	(✓)	(✓)
SPO2	(✗)	(✗)
BPM	(✗)	(✗)
Support SD	(✗)	(✓)
Interface graphique	(✗)	(✓)

Ce que l'on peut remarquer c'est que le WIO terminal embarque presque tout ce qu'il faut. En se référant à l'extrait du cahier des charges, on déduit qu'il nous faut qu'uniquement adjoindre un thermomètre (car ce qui nous intéresse c'est la température corporelle et non ambiante) et un photopléthysmographe pour subvenir à tous les besoins.

Il est certes, légèrement plus important au niveau de la taille mais il contient un écran et un support pour carte SD. C'est donc un compromis, ici j'ai préféré prendre ce dernier pour une approche plus simple et plus rapide pour mettre à profit le temps récupéré ailleurs. De toute manière, l'objectif

ici était de créer un bracelet connecté (pas forcément optimal) dans le but de prouver que l'on peut fabriquer un tel dispositif à moindre coût. L'intérêt réside donc dans un équipement embarqué fonctionnel et non dans un produit parfaitement pensé.

De plus, on peut voir que dans la solution présentée à droite, l'ajout d'un support de carte micro-SD (composant noir en bas) et du capteur chargé du BPM et de la SPO2 (capteur vert en haut) à la carte Arduino prend finalement plus de place. Sans parler des faux contacts dûs aux mauvaises soudures ou encore le fait que la carte Arduino ne tenait pas dans les slots prévus pour ses pattes (d'où les câbles noirs qui seraient ça). Il aurait donc été nécessaire de repenser un PCB et vérifier que sa conception et réalisation était bonne. On peut également apercevoir qu'il manque un système d'affichage (écran ou via une autre interface comme nos téléphones portables) et un capteur de température corporelle. En effet, l'élève précédent n'avait pas réussi à connecter deux capteurs en même temps sur ses solutions. Ainsi, ma première mission était de trouver un moyen de connecter les deux capteurs. Pour cela, je suis allé voir comment fonctionnait l'**échange d'info** et ces deux capteurs (car l'échange d'informations diffère selon les



Solution Arduino récupérée

Egalement noter que la solution que j'ai privilégié, le WIO Terminal, dispose de pin similaire à l'arduino et repose donc sur le même principe de fonctionnement. Si j'arrive à trouver un schéma de montage pour une carte Arduino alors j'aurais ce même schéma pour mon WIO.

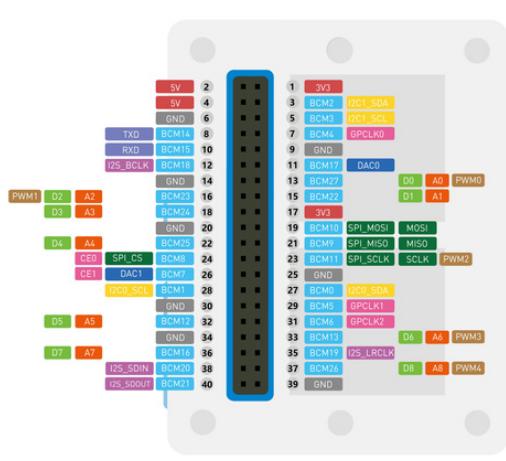


Schéma des pins du WIO Terminal (vue de dos)



image du Mlx90614

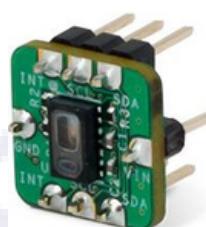


image du Max30102(maxrefdes117)

Dans mon cas, j'utilisais comme capteur de température le **mlx90614** et comme capteur photopléthysmographe le **max30102** (version maxrefdes117) respectivement de société **Melexis** et **Maximintegreated** (cf images sur leur site).

Ces deux capteurs partagent les informations par un principe nommé "**i2c**". L'i2c, "Integrated Circuit", est une technologie existante depuis 1992, créée par Philips pour simplifier la **transmission de données** entre un **microprocesseur** et d'autres **objets électroniques**, c'est donc un bus informatique.

Cette technologie repose sur **4 fils**:

- Vin : apport électrique,
- GND : terre
- SDA (Serial Data Line) : ligne de données bidirectionnelle
- SCL (Serial Clock Line) : ligne d'horloge de synchronisation bidirectionnelle

On peut donc connecter un grand nombre de capteurs utilisant cette technologie à un même microprocesseur, qui lui se chargera de récupérer les informations sans conflit. Il suffit de les **brancher en série**.

Le seul problème que l'on peut rencontrer réside dans le fait que les capteurs doivent avoir des adresses de stockage différentes. En effet, on ne peut pas stocker deux valeurs au même endroit et au même moment (sinon il faut un multiplexeur i2c).

En tapant sur Google "nomDuCapteur i2c adress", on obtient:

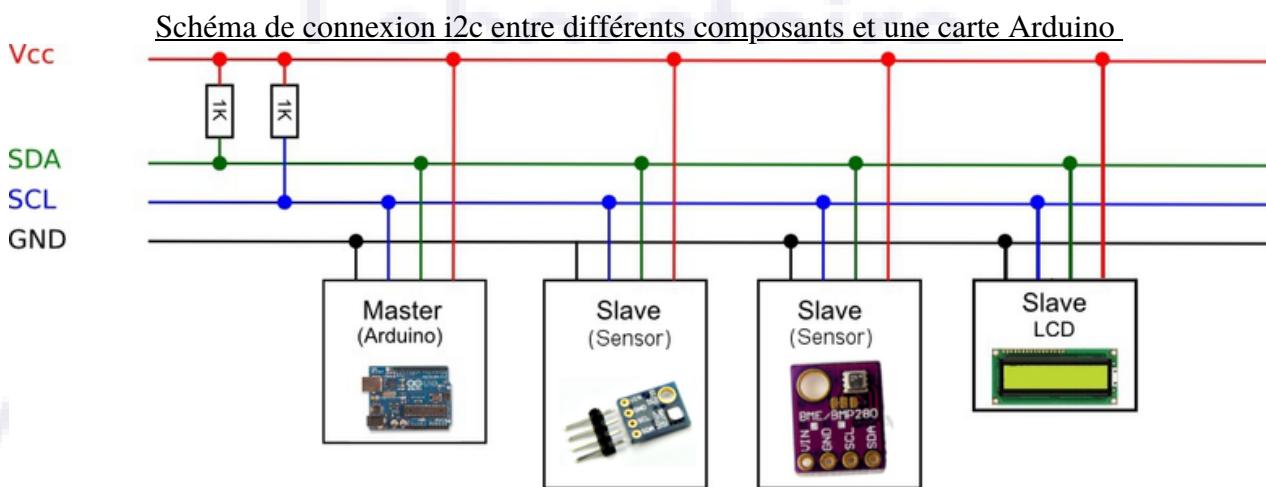
"

- Every MLX90614 has a default I2C address of 0x5A, but that address can be re-written -- one of the major features supported by the device.
- max 30102 : It has a fixed I2C address: 0xAEHEX (for write operation) and 0xAFHEX (for read operation).

"

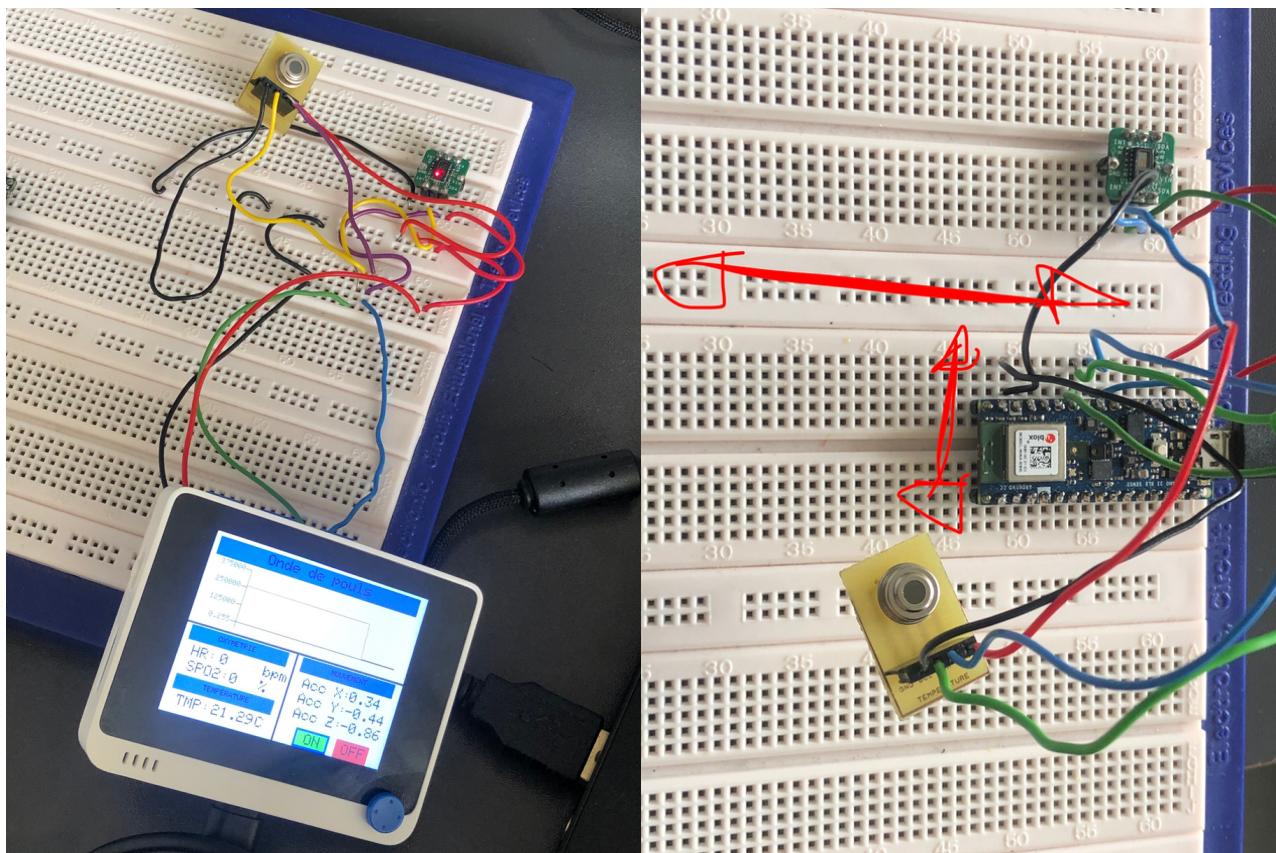
On n'a donc **aucun conflit d'adresse mémoire** pour nos deux capteurs. On peut ainsi faire un montage i2c entre notre WIO Terminal et les deux capteurs.

En **théorie**, cela ressemble à quelque chose comme ci-dessous (résistances non obligatoire):



(cf : <https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2Flearning-by-doing-404%2Fi2c-communication-between-rpi-and-arduino-b7064bffe02a&psig=AOvVaw1PZMxHvLrJYlpIsUDPRp&ust=1669199268746000&source=images&cd=vfe&ved=0CBEQjhxqFwoTCOjE-oPKwfsCFQAAAAAdAAAAABAI>)

Pour vérifier que cela fonctionnait parfaitement, j'ai utilisé une breadboard.



Photos montrant l'utilisation de l'i2c avec d'une part avec le Wio et d'autre part avec l'Arduino

Après quelques courts tests et une implémentation dans le code (voir partie code) ce système a montré ces fruits et j'ai donc pu utiliser mes deux capteurs en même temps. De cette manière, je remplissais tous les critères de prélèvement nécessaires au bracelet connecté.

On souhaitait que les **capteurs** soient sur une **extrémité d'un doigt** tel que l'index pour une **mesure optimale des données**.

Cependant, les capteurs ne sont pour le moment que sur une breadboard et les câbles semi-rigides ne sont pas adaptés pour une utilisation pour le corps.

Il faut donc trouver une alternative qui permet d'avoir des câbles suffisamment fins mais également souples.

Le câble typique qu'on retrouve dans les laboratoires de l'ESIEE (comme ci-dessus) est en fait un unique câble, c'est pour cela qu'il est semi-rigide ! Ce qu'on considère comme étant un câble souple est en fait une multitude de filaments enroulés en hélice formant un câble torsadé. Il existe des **câbles nappes** qui se reposent sur ce principe et qui permettent donc d'établir une connexion électrique souple mais également assez discrète de par leurs tailles (voir à droite le câble nappe gris).



Montage final

Enfin le dernier objectif en terme de connectique est comment relier les capteurs ? puisque pour ma phase de test ils étaient sur une breadboard, on se doute que ce n'est pas pensé pour un équipement amovible.

En électronique, pour faire des circuits plus complexes ou plus optimisés, on crée des **PCB** (circuits imprimés). On s'intéressera plus bas à comment cela fonctionne réellement et comment je l'ai conçu/créé. Ce qu'il faut comprendre ici, c'est l'intérêt de ce dernier.

C'est un **gain certain de place** et le fait que les deux capteurs soient ensemble sur une plaque rigide **facilite l'installation** des capteurs sur le doigt.



Pour faire simple, les circuits imprimés peuvent être fabriqués grâce à différents procédés mais le principe demeure identique. Il ne s'agit que de pistes formées dans un conducteur qui relie les différents composants entre eux (ici 4 pistes puisque l'i2c avait 4 fils !).

Circuit imprimé réalisé par gravure anglaise

Pour conclure cette partie, il y a eu une **évolution de la connectique** tout au long du projet pour répondre à de **nouveaux besoins**. On est ainsi passé d'une connectique "pour tester" à une connectique adaptée à un objet transportable.

Laboratoire
ESYCOM

IV - c) Code

Le **code** est **complémentaire** à la **connectique** ! En effet, sans connexion électrique, le code ne peut opérer (et sans code les capteurs ne font rien !). Il a donc fallu répondre à certaines problématiques pour que l'addition des deux donne un résultat.

Dans le cadre du stage, j'ai majoritairement programmé en **Arduino** mais également légèrement en **C**. En réalité, l'Arduino permet l'utilisation de fichiers C, ce qui est très pratique pour importer des librairies de capteurs et donc grandement facilité l'utilisation de ces derniers.

En fait, le C, inventé en 1970, est l'un des premiers langages de programmation, il est donc par conséquent un des plus utilisés. Beaucoup de langages découlent de ce dernier et ne sont que des évolutions. Ainsi, on retrouve traditionnellement le C pour être employé dans les systèmes embarqués (qui sont déjà conçus depuis plusieurs années !). Il permet de manipuler aisément et comme on le désire nos composants

électroniques et notamment de jouer avec les mémoires (très pratique pour récupérer des informations par exemple). Il rend également possible plein d'autres concepts intéressants, comme le multi-threading et le multi-programming. Comme il est l'un des codes phares puisqu'il est énormément utilisé et depuis longtemps, il est presque totalement universel. Sa portabilité est très agréable pour un développeur.

Bien qu'il ait beaucoup d'avantages, il a forcément des inconvénients. On ne s'intéressera ici qu'au temps de développement et à sa difficulté, puisque ce sont les raisons pour lesquelles l'étudiant précédent n'a pas pris ce langage.

Le langage Arduino, quant à lui est apparu en 2005. Son utilité, à sa création, est pour simplifier la compréhension à la programmation aux microcontrôleurs pour les étudiants. On comprend donc qu'il est plus rapide et facile d'utilisation. Cela se fait au détriment des possibilités qu'offrait le C. Il permet donc de manipuler moins de systèmes (ils doivent être compatibles Arduino). Ce langage n'est pas non plus un langage d'optimisation et ne permet pas de faire tout de ce que proposait le C comme le multi-threading.

Avec ce langage, on peut programmer des microcontrôleurs tels que la carte Arduino Nano et le WIO Terminal vu précédemment.



Logos Arduino et C (2022)

Pour pouvoir utiliser les codes .ino (Arduino) il m'a fallu installé l'**IDE Arduino** (windows 10, 2022 : <https://docs.arduino.cc/software/ide-v1/tutorials/Windows>). Une fois dans l'IDE il faut préciser que l'on code sur un WIO Terminal. Par ailleurs, je n'ai malheureusement pas reçu de fichier .zip contenant toutes les librairies utilisées dans les codes. J'ai, par conséquent, passé du temps à trouver ces dernières sur internet (et plus particulièrement sur github.com) qui n'étaient parfois plus d'actualité...

Finalement après avoir, avec quelques difficultés, réuni toutes les librairies nécessaires, j'ai pu réellement m'attaquer au code.

Arduino étant un langage nouveau pour moi, je me suis formé grâce à des vidéos Youtube d'un créateur de contenu français (je ne me souviens malheureusement plus du nom...).

Dès le début du stage, je me suis dit qu'il y avait une énorme charge de travail sur cette partie. Les codes que j'ai reçu du dernier élève étaient très désordonnés, il y avait également beaucoup de problèmes de logique et ils contenaient également beaucoup de bugs.

Je dirais avoir fourni un gros travail sur les codes. En effet, j'ai pu acquérir au cours de mes deux premières années beaucoup de notions informatiques et appris un certain nombre de conventions de code. J'ai eu la chance de programmer en JAVA, en C mais également beaucoup en Python (langage avec lequel j'ai le plus d'expérience). Ces notions retenues grâce à ces cours ont pu être de nouveau appliquées mais cette fois en Arduino.

Le **code** n'étant **pas fonctionnel au début** (il ne se passait rien), j'ai dû chercher la/les raison(s). Mais dans ce cafouillage cela était impossible !

```

198 > int bpmCalculator(long ir) {-
224 }
225
226 > void createSavingfile() { ...
239 }
240
241 > void LineGraph(long ir) { ...
279 }
280
281 > void drawData(float x, float y, float z, int bpm, int spo2, float temperature) { ...
308 }
301
302 > void drawEnd() { ...
320 }
321
322 > void drawRecordingBackground() { ...
368 }
369
370 > void drawWaitingScreen() { ...
381 }
382
383 > void eraseWriting() { ...
395 }
396
397 void initializeSensors() {
398     /*fonction qui initialize tous nos capteurs*/
399     //Initialize the tft
400     tft.begin();
401     tft.setRotation(3);           //Defining the rotation
402     tft.fillRectScreen(TFT_WHITE); //Black background
403     spr.createSprite(320, 120); //Create a Sprite for line chart IR (width 320, length 120)

```

Code provenant de mon IDE Arduino

Mon premier réflexe a alors été de **compartimenter** le code, pour y voir plus clair (l'oeil y dissocie mieux les différentes parties) mais cela a également eu comme avantage de rendre indépendantes ces parties. Dans les codes que j'ai reçu, il n'avait pas de fonctions (par exemple pour initier les différents capteurs) et toutes les instructions étaient mélangées... C'était donc un problème puisque je veux que mes traitements, mes sauvegardes ainsi que mes affichages ne soient pas liés ! Quand j'enlevais une partie d'affichage ça plantait tout le code... un réel casse-tête pour **déboguer** !

J'ai donc créé une multitude **fonctions**, qui sont toutes **indépendantes**, me ne créant ainsi plus d'erreur fatale lorsque je retirais une partie du code.

On peut voir ci-dessus, un extrait de mon code final, présentant différentes fonctions.

Un autre exemple concret se situe à droite. Dans la fonction setup (une des deux fonctions de base Arduino), on comprend facilement son contenu et si l'on

```

116
117 //#####
118 void setup() {
119   pinMode(buttonOn, INPUT);
120   pinMode(buttonOff, INPUT);
121
122   initializeSensors();
123   drawWaitingScreen();
124   createSavingFile();
125 }
126
127 //#####
128 void loop() {

```

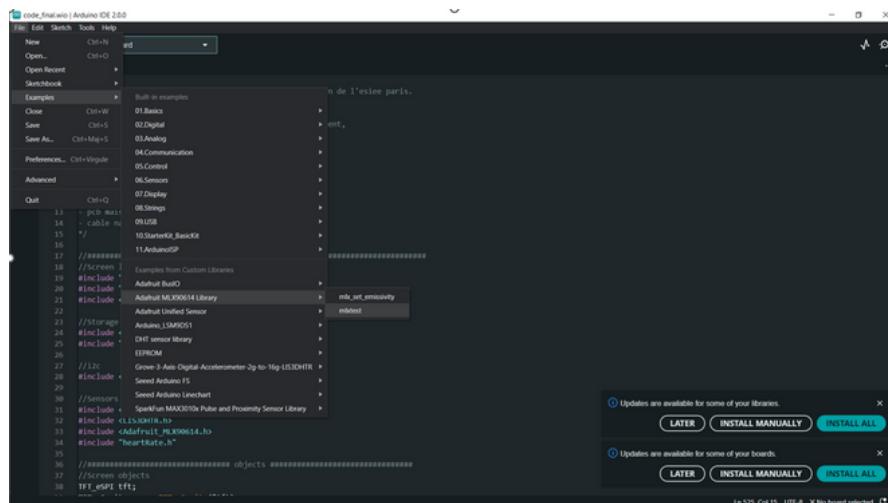
veut changer

Code provenant de mon IDE Arduino

l'initialisation d'un capteur on va dans initializeSensors. Ainsi, je suis passé d'un setup illisible et bugué de 100 lignes à un setup de 6 lignes compréhensible par tout le monde (le nom des fonctions étant parlant...).

J'ai, par ailleurs, rapidement pris la peine de **renommer** toutes les **variables** et également de les **classifier** pour qu'elles soient repérables et rapidement interprétables (elles n'avaient jusqu'alors que des noms compliqués ne signifiant pas grand chose).

Une fois le code récupéré de base, à peu près fonctionnel avec toutes les modifications que j'ai déjà apporté, j'ai continué de le retoucher en repensant la logique du code. En effet, il y avait de **grosses erreurs de logique** : des while à la place de if, des inégalités dans le mauvais sens, pour ne citer que quelques exemples.



Code provenant de mon IDE Arduino

J'ai donc **modifié** la **structure** du code, toujours dans le but de le faciliter, tant pour la compréhension que pour la maintenance. Après cela, j'ai obtenu un code bien plus lisible et j'ai pu y **ajouté** une **fonction** pour mon nouveau **capteur de température** qui n'était pas implémenté jusqu'ici ! (mlx90614)

Un des moyens rapides pour utiliser un capteur en Arduino est de chercher une librairie en C sur internet et de l'installer dans l'IDE. L'avantage, c'est que le capteur est déjà prêt à l'utilisation avec les fonctions créées dans ce fichier. De plus, on retrouve bien souvent des codes Arduino "exemples". Comme pour mon capteur de température, il y a un code qui permet d'afficher la donnée (voir ci-dessus). De cette manière, on n'a pas besoin de réinventer la roue et on a juste à modifié le code pour obtenir ce que l'on désire, c'est l'avantage de l'**open source**.

Cependant, cette méthode a ses limites. Dans un premier temps, il faut vérifier pour quelle utilisation les codes sont autorisés. De plus, on ne trouve pas toujours de codes exemples, surtout lorsque les programmes deviennent complexes. Et puis, il faut vérifier que les codes donnés fonctionnent ! Ce n'est pas toujours le cas. Notamment pour mon calcul de la SPO2. Je n'ai pas pu trouver de code qui fonctionnait correctement ! Ils me donnaient tous des valeurs ahurissantes. Donc pour palier à ce problème, je suis allé **chercher des revues scientifiques** sur internet, des études me permettant de comprendre exactement comment fonctionne et se calcule la SPO2. J'ai finalement trouvé un document qui m'a permis de comprendre tout cela (voir feuille annexe 1 si besoin).

Ainsi, j'ai **entièremment créé la fonction** Arduino permettant de calculer cette donnée :

```

495 int SpO2Calculator(long ir, long red) {
496     /*fonction/algorithme qui calcule le spo2 */
497     //Serial.println("spo2");
498     if (ir > 30000) { //If a finger is detected
499         tableIR[i] = ir;
500         tableRed[i] = red;
501         i++;
502         if (i == tableSize) {
503             irValueAverage = 0, redValueAverage = 0;
504             irMin = tableIR[0], irMax = tableIR[0];
505             redMin = tableRed[0];
506             redMax = tableRed[0];
507             i = 0;
508             for (int j = 0; j < tableSize; j++) {
509                 if (tableRed[j] > redMax) redMax = tableRed[j];
510                 if (tableRed[j] < redMin) redMin = tableRed[j];
511                 if (tableIR[j] > irMax) irMax = tableIR[j];
512                 if (tableIR[j] < irMin) irMin = tableIR[j];
513                 irValueAverage += tableIR[j];
514                 redValueAverage += tableRed[j];
515                 tableIR[j] = 0;
516                 tableRed[j] = 0;
517             }
518             redValueAverage /= tableSize;
519             irValueAverage /= tableSize;
520             ratio = ((redMax - redMin) * irValueAverage) / ((irMax - irMin) * redValueAverage);
521             spo2 = (-45.060 * ratio + 30.354) * ratio + 94.845;
522         }
523     } else spo2 = 0;
524     if (spo2 < 0 || spo2 > 100) spo2 = 0;
525     return spo2;
526 }
```

Code provenant de mon IDE Arduino

Pour conclure cette partie, je n'ai rencontré qu'un soucis majeur concernant le code que j'expliquerai plus bas dans les limites du projet.

On ne rend pas forcément compte de la charge de travail fournie pour le code. Ce n'est pas tellement visuel et dire que l'on crée des fonctions semble rapide mais il faut d'abord réfléchir à comment organiser tout cela, puis le faire, le tester... Cela m'a donc pris beaucoup de temps pour comprendre, déboguer, implémenter et réorganiser tout le code qui fait tout de même 526 lignes.

IV - d.1) Réalisation de matériel

Comme vu précédemment, pour faire évoluer la connectique afin qu'elle soit adaptée pour un bracelet, il a fallu concevoir et produire du matériel.

Pour rappel, le câble nappe est un regroupement de filaments enroulés en hélice. On ne peut donc brancher ce câble directement au port du WIO Terminal. Pour cela, il a fallu souder des pattes de composants sur les extrémités du câble.

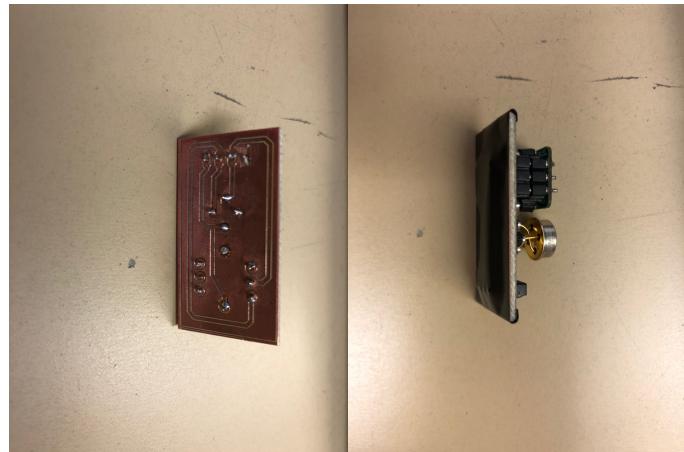
Mais la **réalisation principale** est le **circuit imprimé** présenté à droite. En effet, l'intérêt de ce dernier est de simplifier la connexion électrique entre les deux composants, de réduire la place que le dispositif prend mais également de faciliter l'installation des capteurs sur l'extrémité du doigt.

Cependant, produire un tel support se fait en plusieurs étapes et à l'aide de logiciels permettant ce genre de réalisations. Durant mon stage, c'est le **logiciel EAGLE** que j'ai utilisé. C'est une application simple de prise en main puisque assez intuitive et l'on peut retrouver facilement des tutoriels d'explication sur internet. Son seul "défaut" réside dans le fait que l'application est payante. On peut tout de même en bénéficier gratuitement si l'on est étudiant.

Pour commencer la réalisation, la première question qui se pose est : que désire-t-on mettre sur notre PCB ? Ici, je voulais intégrer mon capteur de température (mlx90614) et le photoplésthymographe (max30102).

Il est fondamental de **bien choisir le composant** ! En effet, chaque composant a une empreinte qui lui est unique (fichier qui permet de savoir les différentes entrées électriques, qu'est-ce qu'il y a dans le composant...). Ainsi, pour mon max30102, bien faire attention à prendre la déclinaison maxrefdes117 et pas une autre.

L'avantage de travailler avec un logiciel de renom comme EAGLE, c'est qu'il existe des sites **open source** comme : <https://www.snapeda.com/home/> permettant d'installer des fichiers adaptés à ces derniers important des empreintes déjà créées, c'est un mécanisme relativement similaire aux librairies avec l'IDE Arduino. C'est un gain de temps énorme, et un moyen qui permet de réduire les erreurs de développement également. Au lieu de devoir recréer les empreintes de chaque composants, de les enregistrer dans l'application pour les réutiliser... on installe directement cette empreinte depuis le site vu précédemment. C'est pour cela qu'il faut faire attention à la référence de notre composant. Si l'on recherche le mauvais composant sur le site et qu'on fait notre PCB à partir de ce dernier, il n'est pas fonctionnel car non adapté.



Photos du PCB sans et avec composants

On peut voir sur ce premier schéma les empreintes des déclinaisons dont je dispose du mlx90614 et du max30102. On se rend aussi compte qu'il y a un port connectique de 4 branches sur la gauche. Pour permettre de **changer aisément les capteurs et les fils**, ma tutrice me demandé de souder des **connecteurs** et non pas nos capteurs. Cela permet d'**intervertir nos capteurs** ou juste de les **récupérer** si ça ne fonctionne pas. On s'aperçoit que j'ai aussi nommé des fils électriques pour établir une connexion entre mes ports connectiques et mes deux capteurs (puisque les pistes doivent relier les ports correspondant entre eux). On en déduit donc que mon fil SDA reliera tous les ports SDA, le SCL tous les ports SCL...

D'autre part, on remarque que dans les rectangles du max30102 (maxrefdes117) il y a une entrée VIN et une GND et dans le mlx90614 ces entrées sont nommées VDD et VSS, ce sont des conventions :

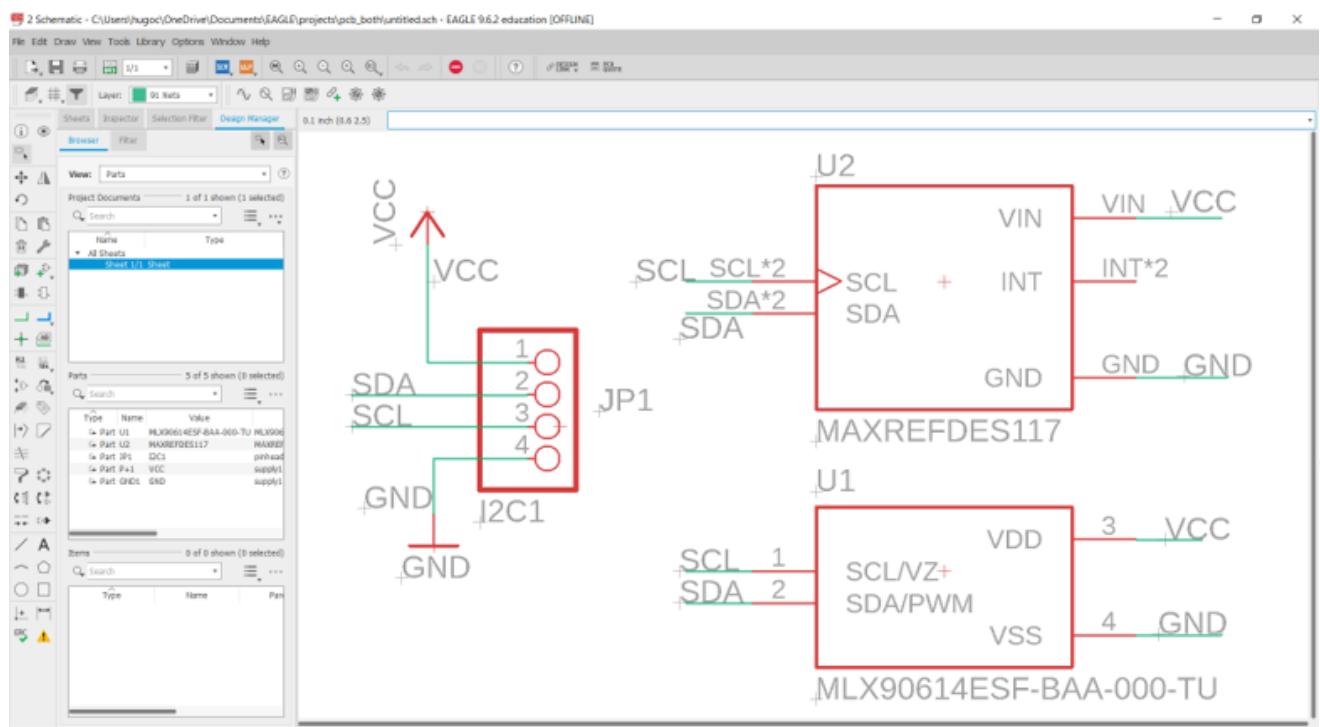
VCC : alimentation ou niveau logique haut pour un circuit TTL

VDD : alimentation ou niveau logique haut pour un circuit CMOS

GND : masse ou niveau logique bas pour un circuit TTL

VSS : masse ou niveau logique bas pour un circuit CMOS

Si le schéma comporte des TTL et des CMOS, on utilise souvent les notations VCC et VSS mais ce ne sont que des notations.



Logo et images tirées du logiciel
EAGLE

Une fois après avoir informé le logiciel de nos différentes connexions, il nous crée un autre fichier où l'on doit placer nos différents composants. Lorsque qu'ils sont placés, on doit faire le **routage**. Cela signifie qu'il faut créer nos pistes entre nos différents ports. Ici, le fichier n'est pas à jour puisque pour nos pistes on **cassera les angles** (pas d'angles droits) mais plutôt des angles à 45° en raison de la **possibilité d'interférences électromagnétiques** (EMI) bien que négligeables pour notre application pratique.

On remarquera que ce n'est pas une "piste" pour notre masse. C'est en réalité un **plan de masse**, en plus d'avoir moins de travail à faire (car on enlève de la matière lorsque l'on crée un PCB, donc plus de matière moins de travail), il a des avantages notables :

"

Retour de tension : la plupart des composants de la carte sont connectés à l'alimentation, puis la tension de retour revient à travers la masse électrique. Sur les cartes à une ou deux couches, les masses sont acheminées à l'aide de pistes plus larges. Dans le cas d'une carte multicouches, dédier une couche entière au plan de masse, permet de simplifier la connexion de chaque composant à la masse.

Retour de signal : les signaux classiques doivent également revenir à la masse, et dans le cas des cartes high-speed, il est très important qu'ils empruntent un chemin clair et le plus directe possible vers la masse. Sans ce retour clair, ces signaux peuvent générer beaucoup d'interférences sur l'ensemble ou partie du PCB.

Réduire le bruit et les interférences : avec l'augmentation de la vitesse des signaux, les changements d'états augmentent sur les circuits numériques. Cela engendre des pics de bruit sur la masse, qui peuvent affecter d'autres parties du circuit. Un grand plan de masse permet de réduire les perturbations en diminuant l'impédance plus qu'une piste d'alimentation pourrait le faire.

"

(cf : <https://www.artedas.fr/Blog/Articles/article-plan-de-masse-PCB.php>)

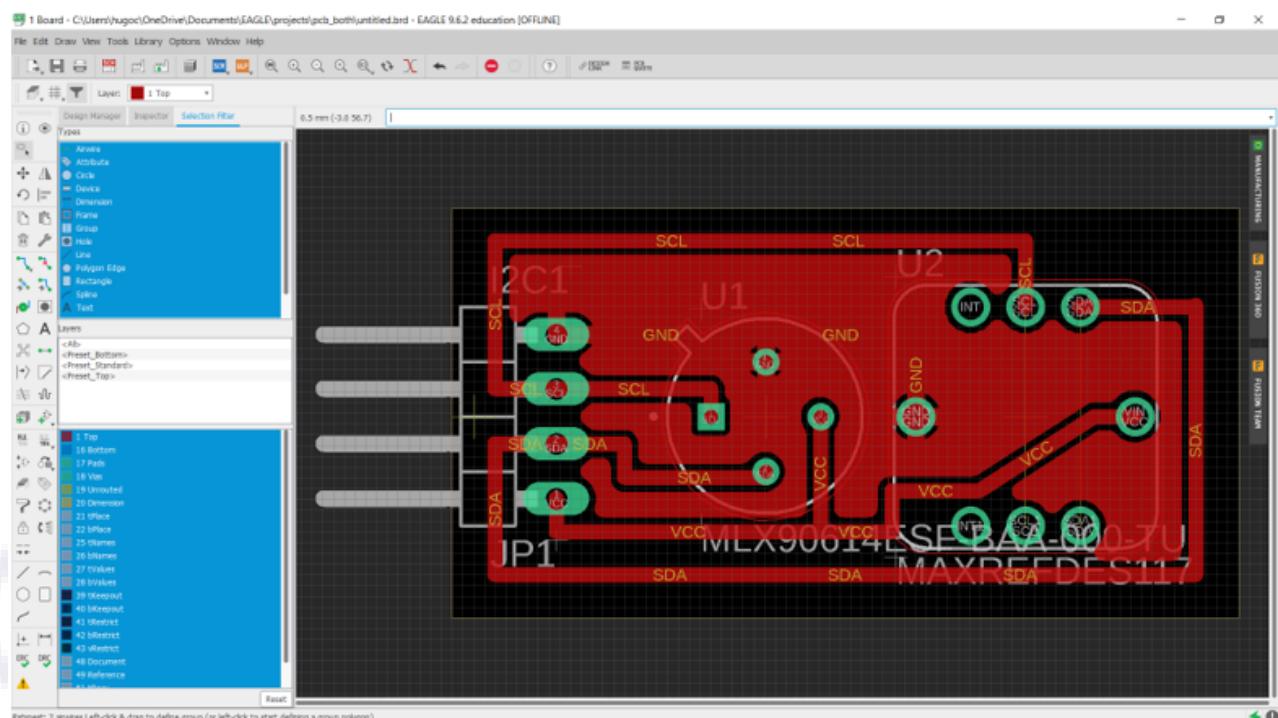


Image tirée du logiciel EAGLE

Si le fichier n'a pas de problème comme par exemple des routes qui se croisent alors on peut l'exporter.

On peut créer des PCB de plusieurs manières. Dans notre cas, je ne nécessitais pas d'une précision extrême et mon circuit imprimé faisait quelques dizaines de millimètres. Pour éviter d'aller en salle blanche et de créer des PCB par gravure à l'acide (plus coûteux et long) on m'a redirigé vers un membre de l'établissement, Carlos PINTO, en charge des **gravures de PCB à l'anglaise**. C'est un procédé simple, rapide et relativement peu onéreux.

On prend une plaque d'époxy sur laquelle on pose une très fine **couche de cuivre** (de l'ordre du micromètre) et on va **enlever** ce cuivre par **voie mécanique**. Cela va donc **créer les pistes**. Une fois les gravures faites on percera aux bons endroits la plaque pour mettre nos composants (ici nos connecteurs)

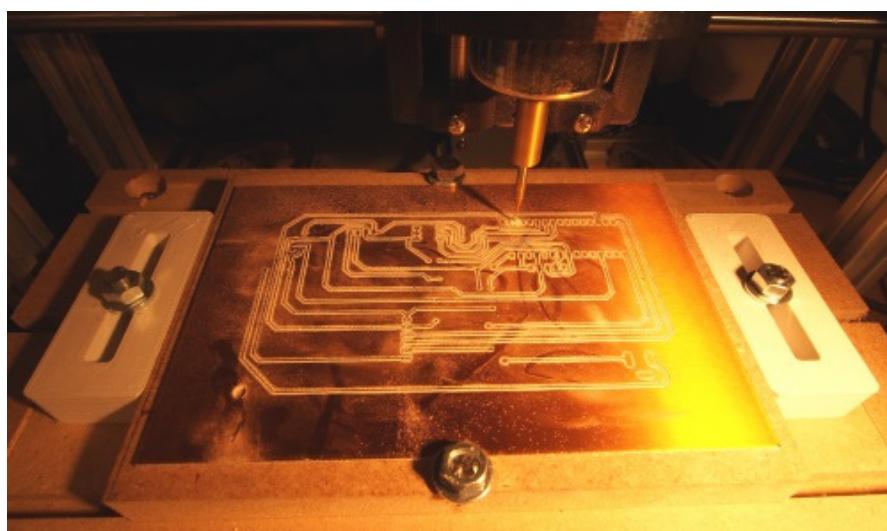
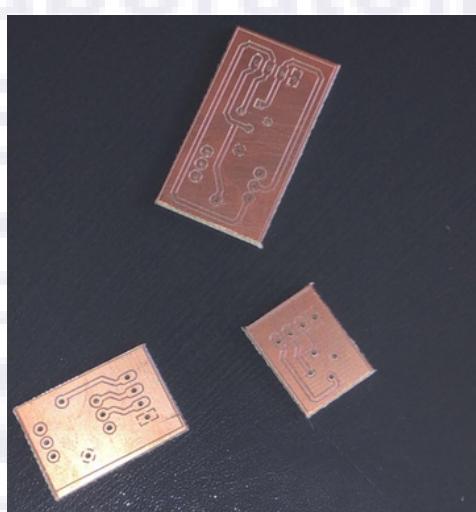


Image montrant un PCB gravé à l'anglaise

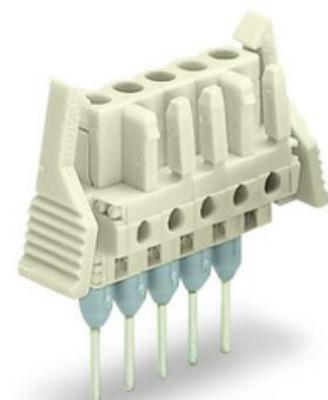
(cf : https://www.google.com/url?sa=i&url=https%3A%2F%2Flabo.sitagg.com%2F2017%2F09%2F10%2Fcreation-de-pcb-avec-une-cnc-gravure-a-langlaise%2F&psig=AOvVaw0WawyZiKdEhQqUXkT_a1WP&ust=1669632572249000&source=images&cd=vfe&ved=0CBEQjhxqFwoTCLD1pqOYzvsCFQAAAAAdAAAABAD)

J'ai donc obtenu les PCB ci-dessous, un accueillant les deux capteurs et deux accueillant un seul capteur.



PCB réalisés

Il a donc fallu **souder des ports connectiques femelles** pour chaque patte de composant semblables aux ports blancs à droite. Enfin, une fois toutes les soudures finalisées, on doit s'assurer qu'il n'y a **pas de problème de réalisation**. J'ai donc, grâce à un multimètre fait des **tests de continuité** pour chaque piste. Comme il n'y avait aucun faux contact, j'ai pu mettre mes capteurs et mon câble dans les connecteurs et tout faire fonctionner avec mon WIO Terminal.



Ports connectiques femelles

(cf images ports : https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.wago.com%2Ffr%2Fconnecteurs%2Fconnecteur-femelle-pour-bornes-sur-rail%2Fp%2F722-150_005-000_039-000&psig=AOvVaw254LMnPX28qr6RECDh1tmQ&ust=1669633954958000&source=images&cd=vfe&ved=0CBEQjhxqFwoTCNjCtqqdzvsCFQAAAAAdAAAABAD)

Laboratoire
ESYCOM

IV - d.2) Réalisation du bracelet connecté et son fonctionnement

A ce stade, on a les différents éléments techniques mais il manque un moyen de faire **tenir** les **WIO** et sa **batterie** sur un **bras** et un moyen de faire tenir le **PCB** sur le **doigt**.

Concernant le WIO Terminal et sa batterie, une approche très simple pour juste un prototype est un **braceard** que les coureurs utilisent pour maintenir leur téléphone portable durant leur activité.

On remarque que ce brassard permet en plus à un câble de passer par des ouvertures prévues à cet effectif par le bas.

Du fait que ça soit un brassard, il s'**adapte** également à la **morphologie** de chacun et est **réutilisable** à volonté.

(cf :https://www.decathlon.fr/p/bracelet-grand-telephone-running/_R-p-162251?mc=8368943&c=NOIR)



Brassard de marque Decathlon



Doigtier en PDMS fabriqué en salle blanche

Pour fixer le PCB sur le doigt, il était prévu dans un premier temps de l'intégrer à un doigtier en PDMS fabriqué en salle blanche. Cependant, comme j'ai mis des connecteurs pour les pattes de mes capteurs, l'épaisseur était trop importante et rendait cela impossible. Comme le stage touchait à sa fin, je me suis servi d'un sparadrap (qu'on enroule donc autour du doigt et du circuit imprimé). Cela permet, encore une fois de s'adapter à la morphologie de chacun. Cependant, ce système n'est pas réutilisable, à chaque usage il faut un nouveau sparadrap.

Maintenant que les différents éléments sont recueillis, nous pouvons passer à la mise en place et au fonctionnement de ce dernier. Commençons donc par un **inventaire** de ce qui est **nécessaire** pour le **bon fonctionnement** du dispositif.

Il faut :

- Batterie externe (+câble USB type-c),
- Carte micro-SD,
- WIO Terminal (avec code téléversé)
- PCB avec mlx90614 et max30102

- Câble nappe
- Brassard
- Sparadrap

Nous commençons par brancher une extrémité du **câble nappe** au port **1, 3, 5 et 9** du WIO Terminal. On branche l'autre **extrémité** au **PCB** en faisant bien attention aux correspondances ! (SDA avec SDA...) Si les branchements sont mal faits et que nous branchons le WIO à une alimentation trop longtemps, cela va endommager les capteurs ! Cela peut durer quelques secondes, les capteurs vont "seulement" chauffer sans conséquences, mais il ne faut pas que ça excède quelques instants. Il faut donc veiller à ce que ces derniers ne soient pas brûlants dès le démarrage. Puis, on insère la **carte SD** dans le port situé sur le **côté droit du WIO Terminal**. Ensuite, on pose le **PCB** sur l'**extrémité d'un doigt** et on **l'enroule** avec du **sparadrap**. Il ne faut **pas** qu'il y ait **trop de pression** sur le doigt pour que les capteurs mesurent correctement les données.

On branche la **batterie** au **WIO Terminal** (à son **port USB type-c situé en bas**).

On met le **WIO Terminal** dans le **brassard** qu'on attachera autour biceps.

Lorsque le WIO Terminal est allumé il suffit d'appuyer sur le bouton en haut à gauche (comme indiqué sur l'écran d'attente) pour démarrer l'enregistrement.

Pour terminer le processus, il suffit d'appuyer sur le bouton en haut à droite.

Son fonctionnement est plutôt élémentaire. D'un point de vue plus général que vu dans les parties précédentes, le dispositif est alimenté par une batterie externe. cette dernière délivre l'énergie nécessaire permettant au WIO Terminal d'afficher sur son écran et d'enregistrer les données récoltées, par nos capteurs, (qui sont eux aussi alimentés par cette batterie) sur la carte SD. Ces données sont transmises électriquement par le biais du PCB puis du câble nappe.

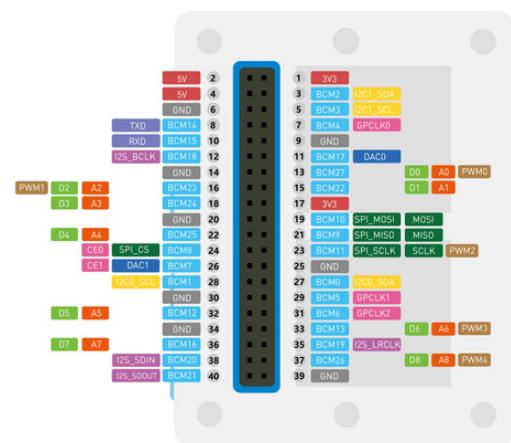
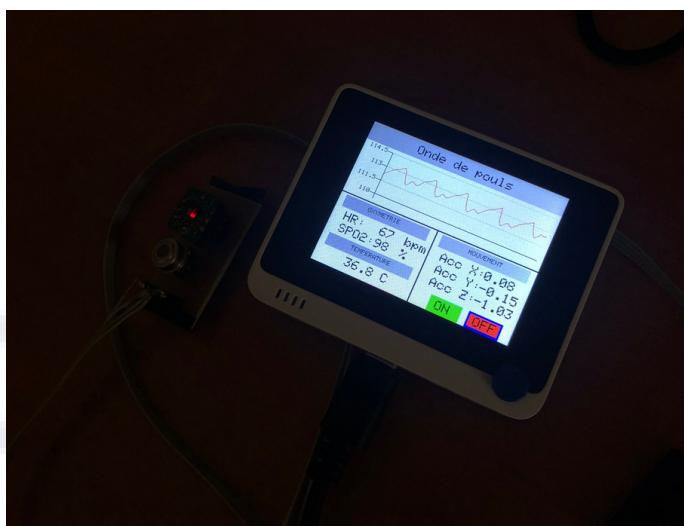


Schéma des pins du WIO Terminal (vue de dos)

On peut voir sur la gauche le **dispositif final**, avec les valeurs affichées à l'écran du WIO Terminal et son PCB / son câble nappe. Le programme est sur "fin" c'est pour cela qu'il y a ces valeurs mais aucun doigt.

IV - e) Comparaison

Maintenant que le dispositif est en état de marche, il est nécessaire de s'**assurer** du **bon fonctionnement** de ce dernier. On ne peut affirmer que ce bracelet est juste dans son utilisation sans le tester dans de vraies conditions.

Pour ce faire, j'ai **installé** le **dispositif** sur **moi** et sur **quelques camarades** (pour prouver qu'il donne des valeurs en fonction de la personne et qu'elles ne sont pas fixes).

Il y a eu des **mesures au repos** mais également **en mouvement** des données suivantes : BPM, SPO2, Température, mouvement.

Ces données sont enregistrées sur la carte SD et on a dans le fichier .csv le temps à laquelle la valeur a été ajouté.



Montre EMPATICA E4

Je disposais d'une montre médicale déjà commercialisée donc des tests avaient étaient faits et approuvés pour celle-ci. la marque dispose d'un site (image dessus) :

<https://www.empatica.com/en-eu/research/e4/>

Pour récolter les informations et utiliser cette montre, on doit installer leur application (ANDROID, IOS...) et se créer un compte.

Une fois l'application installée, il suffit de suivre le guide d'utilisation pour lancer l'enregistrement.

Ce dispositif me donnait que certaines valeurs intéressantes (j'avais le BPM, mouvement et la température).

Une fois les données des deux parts récoltées j'ai, grâce à Excel, calculé l'erreur relative (petit rappel ci-dessous) :

4. ERREUR RELATIVE, INCERTITUDE RELATIVE

L'erreur relative est le quotient de l'erreur absolue à la valeur exacte.

$$\varepsilon_r = \frac{\delta X}{X_e} = \frac{X - X_e}{X_e}$$

Comme il s'agit d'un nombre sans dimension (pas d'unité), on l'exprime généralement en pourcentage (%):

$$\varepsilon_r \% = \frac{\delta X}{X_e} \cdot 100 = \frac{X - X_e}{X_e} \cdot 100$$

Egalement, si la valeur exacte de la grandeur est inaccessible, on prendra la limite supérieure de l'erreur relative qui n'est autre que l'incertitude relative : $\frac{\Delta X}{X_e}$

On peut l'exprimer en % : $\frac{\Delta X}{X_e} \cdot 100$

Ainsi, j'obtiens, **5% d'erreur relative pour le BPM** pour une **valeur moyenne entre 75 et 80** et **1% pour le mouvement**. Pour la température, on ne peut pas vraiment comparer puisque si l'on est critique, la valeur dépend de la partie du corps, Cependant le dispositif que j'ai mis en place tourne entre **36.5 et 37.5° Celcius**, ce qui est très pertinent. De plus, on doit se rappeler que nos **capteurs** ont des **marges d'erreur**, notamment celui de température qui est à +0.2°.

On doit donc prendre ces comparaisons avec des pincettes et voir si cela est cohérent ou non.

Concernant la SPO2, j'avais un capteur avec une pince à placer sur le doigt de la marque BioSignalPlux. Cela ne m'a apporté qu'une confirmation que ma **SPO2** était cohérente puisque j'avais **1% d'erreur**, avec un taux entre **96 et 99%** avec une moyenne à **98%**. J'ai réussi à la faire chuter à 95% en serrant mon bras avec un brassard permettant de calculer l'onde de pouls.

Globalement, on peut dire que pour une première approche et surtout une approche rapide, cela est encourageant. On va voir qu'avec un peu plus de travail et de temps, on pourra répondre aux limites du projet mais également l'améliorer en y ajoutant des fonctionnalités.

Laboratoire
ESYCOM



IV - f) Limites et ouverture

Durant ce projet, j'ai beaucoup appris bien que mon savoir demeure encore vague, mais j'ai pu me rendre compte de **certaines limites**. La première, c'est **moi**. Je ne suis qu'encore élève et en plus tout juste sorti du cycle préparatoire. Je n'ai donc pas toute la théorie et la pratique nécessaire pour un développement optimal. Confier ce projet à quelqu'un de plus expérimenté avec cette base serait idéal.

Ensuite, le choix du **langage de code** est selon moi important. J'ai eu du mal en Arduino à faire fonctionner l'ensemble en temps réel. Je pense que c'est un langage très bien pour s'amuser, pour voir rapidement ce qu'il est possible ou non de faire, ou encore pour des projets plus simples.

Une excellente alternative est le C, permettant un codage dynamique, en temps réel... Il a énormément d'avantages sur tous les points. On pourra même optimisé le code donc réduire les erreurs de calcul (qui sont basés sur le temps) et la dépense énergétique.

Enfin, **les capteurs** sont tous deux infrarouges. Cela est idéal pour une très faible consommation d'énergie mais les mesures sont moins bonnes et sujettes à des erreurs externes. Comme avec le max30102 (SPO2, BPM) qui va être influencer par la couleur de peau, si la personne consomme du tabac, la lumière environnante... En dehors de cela, il existe des capteurs IR étant bien plus précis car ils sont prévus à usage médical. En plus, ils sont plus petits (certes plus dur à souder) mais cela permettrait de créer un boîtier similaire à la montre EMPATICA.

Pour les **pistes de développement supplémentaires**, J'ai vu qu'il existait des **tissus** qui **récupèrent l'électricité du corps humain pour vêtements connectés**. Cela pourrait être un appoint électrique pour que le dispositif tienne plus longtemps sans être rechargeé.

La **fabrication composants électroniques** maison pour une optimisation spatiale est aussi à envisager.

L'**ajout d'un ressort en spiral** (même système que les ceintures de sécurité) pour que le câble nappe se replie.

On peut également **ajouter différents capteurs** pour récolter plus de données qui étaient à voir dans le cahier des charges (mais non nécessaires)

Et enfin le **développement d'un algorithme intelligent de prédiction** à partir des données enregistrées pour savoir l'évolution du patient.

ESYCOM

V - Conclusion

J'aimerais encore une fois remercier Gaëlle LISSORGUES mais également les personnes qui m'ont aidé durant ces presque deux mois.

Ce stage a été ma première opportunité pour appliquer mes connaissances du supérieur et ainsi comprendre un peu mieux ce que fait un ingénieur. Il a été un atout supplémentaire au stage ouvrier de première année dans ma réflexion quant au projet personnel et professionnel.

J'ai pu, à l'inverse de mon travail à Super U évoluer beaucoup plus librement et on a écouté ce que j'avais à dire et on m'a aidé. Là où à Super U je n'étais qu'un employé parmi d'autres qui servait de main d'oeuvre. Cela me conforte dans l'idée que travailler en petit comité, comme dans des start-up me plairait plus de par l'environnement de travail que je considère comme étant plus agréable.

Par ailleurs, étant désireux de découvertes et curieux, je souhaite découvrir d'autres domaines, d'autres équipes de travail mais encore de nouveaux métiers pour élargir ma palette de connaissances et avoir un point de vue plus global du monde du travail. Je me doute bien qu'il existe une infinité de conditions de travail. Le monde du travail n'est qu'une appellation désignant un système extrêmement complexe que nul ne pourrait résumer tellement il est vaste et imprévisible.

C'est pour quoi, j'aimerais la prochaine fois faire un stage ou un projet dans un autre domaine tel que le réseau ou dans une autre entreprise et si possible ne pas le faire en recherche et développement.

Enfin, concernant ma spécialisation au sein de l'ESIEE PARIS, ce stage a été révélateur. J'étais assuré de m'orienter dans un domaine totalement informatique, mais au final, la filière système embarqué peut également être un choix. Il va donc falloir que je me penche plus sur la question pour savoir quelle filière choisir...

Hugo CARANGEOT, élève en E3E au sein de l'ESIEE PARIS.

SaO_2 can be assessed in extracted arterial blood by means of co-oximetry or by measurement of blood gases. SaO_2 can also be obtained noninvasively *in vivo* by pulse oximetry, which utilizes the different light absorption spectra for oxygenated and deoxygenated hemoglobin. In order to isolate the contribution of the arterial blood to the light absorption, photoplethysmography (PPG)—the measurement of light absorption changes due to cardiac-induced arterial blood volume changes—is used. Light transmission through tissue decreases during systole because of the increase in the arterial blood volume during systole and increases during diastole (Figure 1): the PPG signal reflects light absorption changes due to arterial blood volume changes. Due to the discrepancy between the oxygen saturation values obtained by measurements in extracted blood and by *in vivo* pulse oximetry, the latter are denoted by SpO_2 .

Several studies have shown a potential error of 3–4% between SpO_2 and direct SaO_2 measurements in extracted blood in critically ill adult patients [1,2]. An even larger error is expected in neonates and children in intensive care units [3–6].

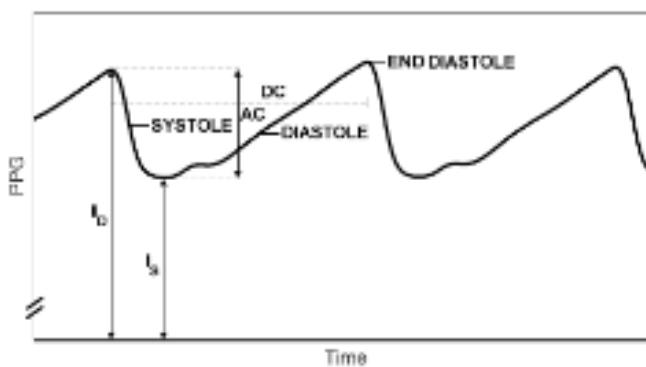


Figure 1. The photoplethysmography (PPG) pulses. The transmitted light through the tissue decreases during systole and increases during diastole. AC is the difference between the maximal (I_D) and minimal (I_S) light transmission through the tissue; DC is the mean light transmission through the pulse.

The theory of conventional pulse oximetry has been described in several publications [7–10], and some aspects of it, relevant to our study, will be presented here. A PPG-based parameter R—the ratio of ratios—is defined by

$$R = \frac{(AC/DC)_1}{(AC/DC)_2} \quad (1)$$

where AC and DC are the peak-to-peak amplitude and the mean value of the PPG pulse, respectively (Figure 1), and the subscripts 1 and 2 refer to the two wavelengths. Utilizing the Beer–Lambert equation, the measured parameter R and the physiological parameter SaO_2 are related through Equation (2) [7–9]:

$$\text{SaO}_2 = \frac{\epsilon_{D1} - R\epsilon_{D2}}{R(\epsilon_{O2} - \epsilon_{D2}) + (\epsilon_{D1} - \epsilon_{O1})} \quad (2)$$

where ϵ_O and ϵ_D are the extinction coefficients for HbO_2 and Hb, respectively. Equation (2) is valid if light attenuation in tissue is only due to absorption by hemoglobin and scattering effects can be neglected. In addition, the relationship between SaO_2 and R, given in Equation (2), is valid when $AC/DC \ll 1$. Otherwise, R in Equation (2) should be replaced by R' , where R' is defined as the ratio of $\ln(I_D/I_S)$ for the two wavelengths:

$$R' = [\ln(I_D/I_S)]_1 / [\ln(I_D/I_S)]_2 \quad (3)$$

The accuracy of Equation (2) is also limited because of the limited precision with which the values of the extinction coefficients of HbO_2 and Hb are known; different studies provided significantly different data [11]. Since light scattering also contributes to light attenuation by increasing the

Fiche d'évaluation de stage technicien par le suiveur en entreprise

Nom de l'étudiant : Hugo Carangeot
Nom de l'entreprise : ESIEE ESYCÖM
Dates de début et de fin du stage technicien : 14/09/2022 au 04/11/2022
Nom du responsable qui a rempli ce formulaire : Gaelle Lissorgues

Merci de préciser votre réponse par des commentaires quand vous le jugez utile.

L'étudiant a-t-il correctement effectué la mission qui lui a été confiée ?

Hugo s'est très investi sur le projet et a montré une très bonne implication tout au long du stage.

Il a fourni un travail de qualité.

Etudiant très sérieux.

L'étudiant a-t-il correctement interagi avec les autres membres du personnel de l'entreprise ?

oui, bonne communication. Points intermédiaires fréquents.

L'étudiant a-t-il fait preuve d'autonomie et d'initiative quand la situation l'exigeait ?

grande autonomie dans son travail et Hugo a réussi à proposer des solutions par lui-même.

L'étudiant a-t-il su mettre au service du thème qui lui a été confié les connaissances techniques qu'il avait ? Semble-t-il avoir acquis au cours du stage les connaissances techniques nécessaires ?

Hugo a beaucoup appris (Arduino en particulier) et a su chercher dans ses expériences comment résoudre certaines difficultés techniques.

Sur quels points, l'étudiant a-t-il encore à progresser ? Quels sont les points qui sont corrects en l'état ?

points positifs: compte-rendus réguliers et de bonne qualité technique, qualité de communication

pas de points négatifs à signaler

.../...

Globalement, diriez-vous que vous êtes satisfait du travail fourni par l'étudiant ?
travail extrêmement satisfaisant

Merci de compléter le tableau ci-dessous.

**I - APPRECIATIONS GLOBALES par le tuteur
"entreprise"**

Pas du tout acquis	Insuffisamment maîtrisé. Des compléments sont nécessaires	Juste maîtrisé	Assez bien maîtrisé	Bien maîtrisé	Très bien maîtrisé	Parfaitement maîtrisé
06	08	10	12	14	16	18

Travail réalisé

					X	
					X	

Méthodes de travail

Date et signature
07/11/2023

