

# RAPPORT DE TP 'e-Health'

Hervé VALETTE, Shunyang ZHANG

Octobre 2021 - EPF

## TP1 : Plateforme e-Health pour s'initier en IoT

CAPTEURS & MESURES.....	2
A.    Saturomètre (Oxymètre de pouls) .....	2
B.    Capteur de position .....	2
C.    Electrocardiogramme .....	3
D.    Capteur de flux d'air .....	3
E.    Capteur de température.....	4
F.    Capteur de réponse cutanée galvanique (GSR).....	4
G.    Mesure avec tous les capteurs en simultanée.....	5
AFFICHAGE ET STOCKAGE DONNÉES MESURÉES DANS UN FICHER CSV .....	5
AFFICHAGE ET STOCKAGE DONNÉES MESURÉES DANS UNE BASE DE DONNÉES .....	6

### MISE EN CONTEXTE ET OBJECTIFS

Pour ce TP, un module d'extension e-Health telle que celui présenté avec la figure 1 gauche est utilisé. Les différents capteurs vont être utilisés et leurs données mesurées à l'aide de la shield e-Health présentée à la figure 2 droite.

Les objectifs sont donc de configurer et programmer des capteurs sous Arduino et d'assurer l'acheminement des données des capteurs connectés utilisés.

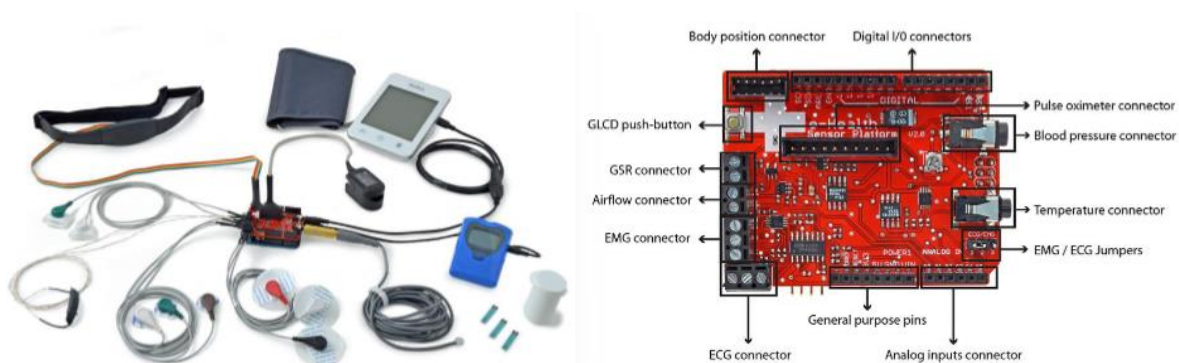


Figure 1 : Plateforme e-Health (gauche) et Description des entrées de la shield e-Health (droite)

## CAPTEURS & MESURES

### A. Saturomètre (Oxymètre de pouls)

Ce premier capteur mesure la saturation en oxygène (%SPo2) dans le sang et la pulsation cardiaque (PRbpm) de la personne. La première mesure se fait à l'aide d'une lumière rouge et une lumière infrarouge en utilisant le principe que plus on trouve d'oxygène dans le sang, plus ce dernier sera perméable à la lumière rouge.

La figure ci-dessous montre comment la mesure a été réalisée et avec quel code ainsi que les résultats obtenus. On peut voir que le capteur ne fonctionnait pas correctement notamment avec la saturation en oxygène égale à zéro. Les branchements étant correcte et ceci étant vrai pour plusieurs autres groupes, nous concluons que le matériel était ce jour là simplement défectueux.

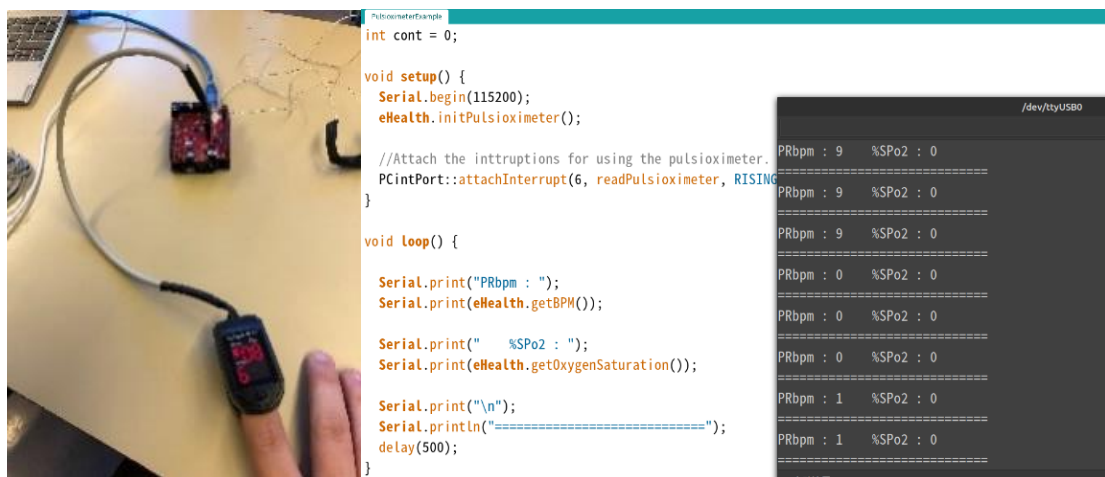


Figure 2 : Branchements, mesure et affichage résultats saturomètre

### B. Capteur de position

La mesure de la position du corps se fait à l'aide d'une centrale inertielle composée de trois accéléromètres qui mesurent les accélérations linéaires selon trois axes orthogonaux et d'un gyromètre mesurant la rotation et la vitesse angulaire.

Le capteur est ainsi capable de détecter cinq positions : « Supine position », « Left lateral decubitus », « Righth lateral decubitus », « Prone position » et « Stand or sit position ».

La figure ci-dessous montre comment la mesure a été réalisée et avec quel code ainsi que les résultats obtenus. On peut voir que le capteur fonctionne correctement mais on que lorsqu'un changement de position se fait au moment de la mesure, « non-defined position » est affiché.

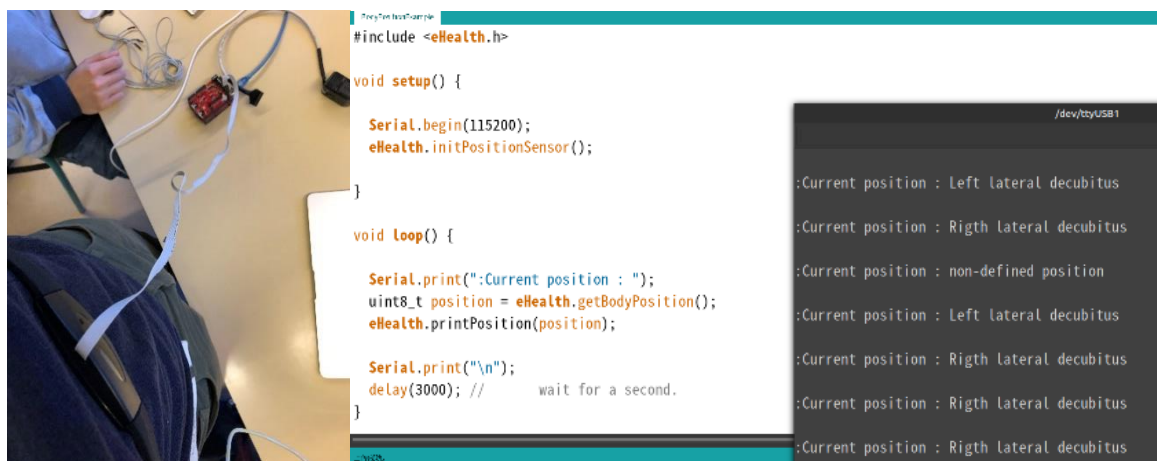


Figure 3 : Branchements, mesure et affichage résultats capteur de position

### C. Electrocardiogramme

Ce capteur mesure l'activité électrique du cœur, ce dernier produisant une impulsion régulière en se contractant afin de faire circuler le sang dans le cœur. Cette mesure se fait à l'aide de trois électrodes, une positive, une négative et une dernière neutre.

La figure ci-dessous montre comment les électrodes sont placées, le matériel utilisé et les résultats obtenus. Les résultats obtenus ont cette fois-ci été tracé afin de mieux comprendre qu'ils représentent en fait les événements cardiaques QRS qui permettent d'étudier la dépolarisation ventriculaire du cœur.

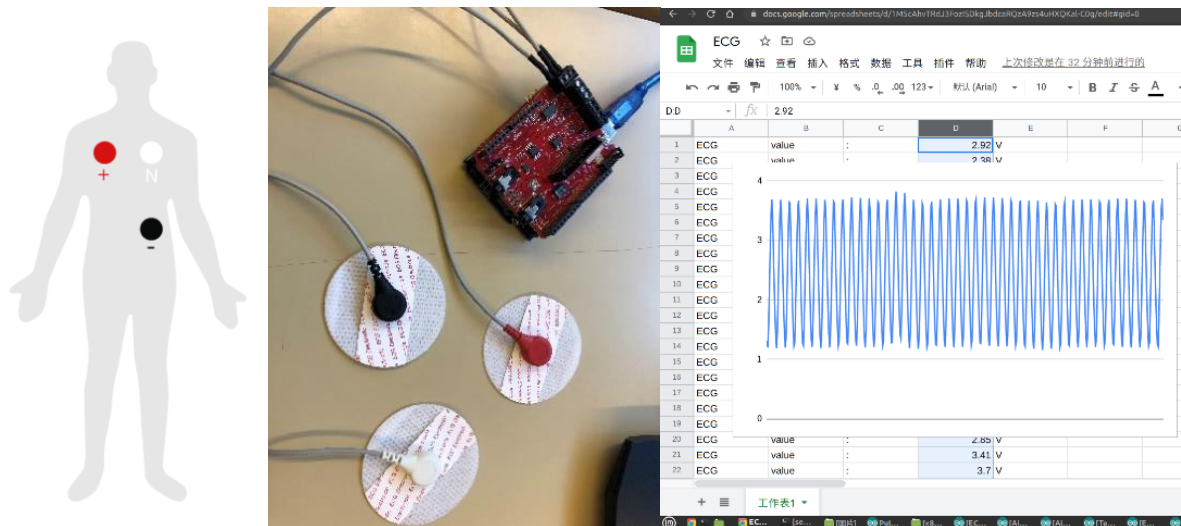


Figure 4 : Branchements, mesure et affichage résultats électrocardiogramme

### D. Capteur de flux d'air

Ce capteur permet de mesurer le flux d'air sortant de chaque narine à l'aide de deux pics placés à leur sortie. La figure ci-dessous montre comment la mesure a été réalisée et les résultats obtenus. On peut voir que ces derniers sont représentés visuellement avec une quantité de points représentant l'intensité du souffle avec ici l'exemple d'un cycle respiratoire complet (inspiration + expiration).

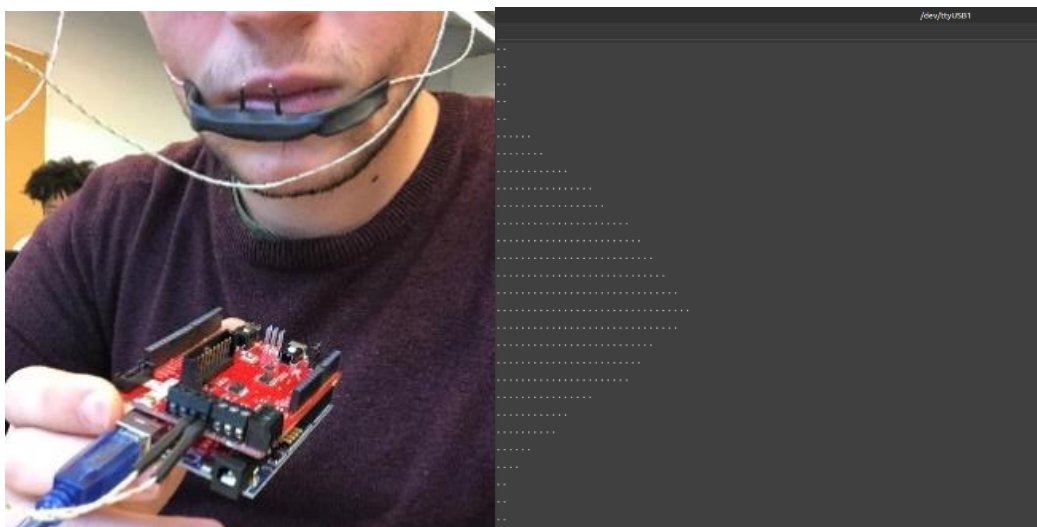


Figure 5 : Branchements, mesure et affichage résultats capteur de flux d'air

## E. Capteur de température

Le capteur de température mesure en degré Celsius la température du corps à l'aide d'un thermomètre placé à l'extrémité d'un doigt. La figure ci-dessous montre comment la mesure a été réalisée et avec quel code ainsi que les résultats obtenus. On peut voir que capteur fonctionne bien avec une température du corps normale d'un peu plus de 37°C.

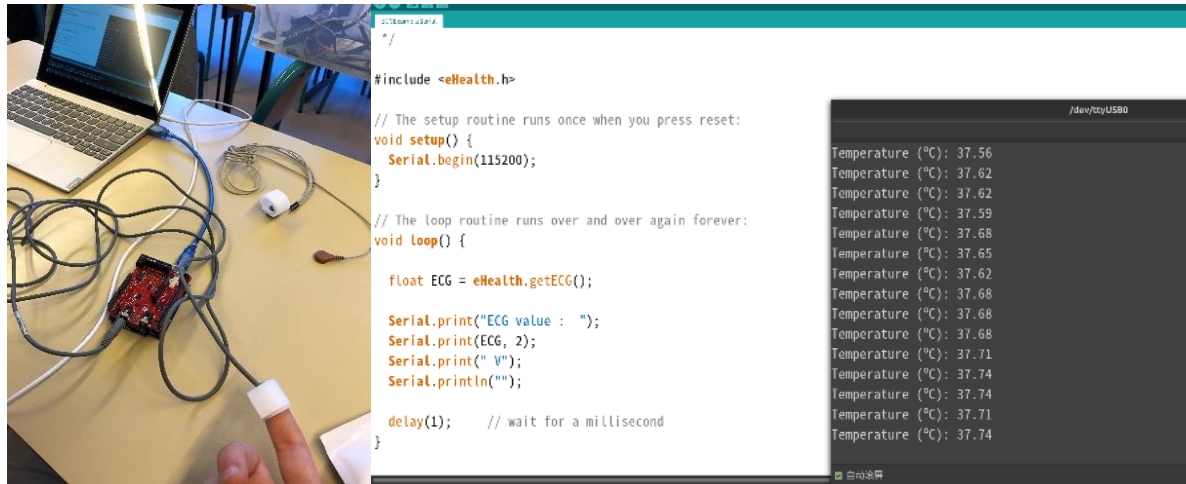


Figure 6 : Branchements, mesure et affichage résultats capteur de température

## F. Capteur de réponse cutanée galvanique (GSR)

Le capteur GSR (Galvanic Skin Response) permet de mesurer la conductivité électrique de la peau qui va varier en fonction de la quantité de sueur à sa surface. Cette mesure se fait avec deux électrodes fixées à l'extrémité de deux doigts d'une même main. Des émotions fortes (stress, choc...) pouvant entraîner des stimuli nerveux entraînant la sécrétions de sueur, cela peut servir par exemple à réaliser une analyse de la qualité du sommeil. Trois variables sont alors mesurées : La conductance, la résistance, la conductance en Volt.

La figure ci-dessous montre comment la mesure a été réalisée et avec quel code ainsi que les résultats obtenus. On peut voir que les valeurs varient de 0.88 pour la résistance et de 132 303.25 pour la résistance en une seconde. Cette variation peut être expliqué par la sensibilité et la qualité du capteur, la présence de sueur au départ de la mesure ou bien par une fixation trop souple du système au niveau des doigts.



Figure 7 : Branchements, mesure et affichage résultats capteur de réponse cutanée galvanique

## G. Mesure avec tous les capteurs en simultanée

Enfin, une mesure avec tous les capteurs branchés en même temps a été réalisé afin de réaliser une mesure complète. La figure ci-dessous montre comment la mesure a été réalisée et avec les codes précédents mis bout à bout ainsi que les résultats obtenus sur le moniteur série de l'IDE d'Arduino.

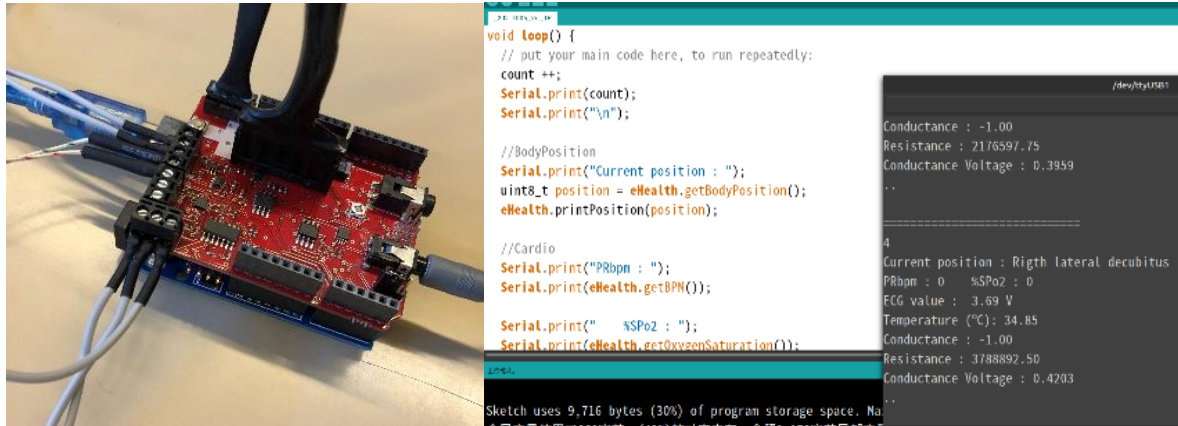


Figure 8 : Branchements et affichage résultats de tous les capteurs en simultanée

## AFFICHAGE ET STOCKAGE DONNÉES MESURÉES DANS UN FICHIER CSV

L'objectif est maintenant d'ajouter les données mesurées dans un fichier CSV directement après qu'elles sont récupérées sur Arduino grâce à le code Python de la figure 9.

```
import serial
import time
import csv

ArduinoSerial = serial.Serial('COM4', 115200)
time.sleep(2)
fieldnames = ['SPO2', 'Position', 'ECG', 'Airflow', 'Temperature', 'Resistance', 'Conductance']

f = open('valeurs.csv', 'w')
f.write(fieldnames)

while 1:
    data = str(ArduinoSerial.readline().decode('UTF-8'))
    ArduinoSerial.flush();
    f.write(data)
f.close()
```

Figure 9 : Code Python utilisé pour implémenter fichier CSV

Ce code permet la création (ou la mise à jour si le fichier existe déjà) d'un fichier CSV nommé « valeurs.csv » avec les valeurs récupérées tout au long de la mesure avec les différents capteurs avec une organisation des données telle que celle présentée dans le tableau suivant :

SPO2	Position	ECG	Airflow	Temperature	Resistance	Conductance
0	Left lateral decubitus	2.85	..	37.56	651592.43	1.53
0	Right lateral decubitus	3.41	..	37.62	577966.25	1.53
0	Left lateral decubitus	3.69	.....	37.59	519289.18	2.41
0	Left lateral decubitus	3.7	.....	37.68	494202.78	1.62



## AFFICHAGE ET STOCKAGE DONNÉES MESURÉES DANS UNE BASE DE DONNÉES

Cette dernière étape consiste toujours à afficher et stocker les données récoltées sur Arduino mais pour les implémenter cette fois-ci sur une base de données mise en ligne. Différentes librairies sont utilisées dont « **mysql.connector** » pour la connexion à la base de données se fait à l'aide la librairie et « **DateTime** » pour dater les mesures ce qui servira comme ID aux mesures pour les différencier.

La connexion se fait dans une première boucle « try/except ». Il faut alors utiliser la fonction **connect()** avec comme arguments les valeurs de '**host**', '**user**', '**password**' et '**database**' données qui sont respectivement '**195.144.11.150**', '**zdj62853**', '**MIN2022!!**', '**zdj62853**'.

Ensuite, à l'intérieur du « try », on utilise une boucle « while » dans laquelle on récupère les données et on sépare dans un tableau array nommé **Data** suite avec la commande **split()**.

Ensuite on ajoute une seconde boucle « try/except » qui servira à rajouter des données tant qu'il y en a à rajouter et en enlevant les erreurs potentielles à l'aide du « except ». L'ajout quant à lui se fait en trois temps.

1. D'abord à l'aide de la ligne de commande (requête sql) «**INSERT INTO () VALUES ()**» avec respectivement comme arguments :
  - les intitulés de la mesure comme par exemple '**Date\_Mesure**' ou bien '**Temperature**' et en dernier l'identifiant patient qui correspond ici à l'identifiant donné à notre groupe, le numéro 4.
  - le type de données recherché comme par exemple '**%s**'
2. Ensuite la mise en place des valeurs dans l'ordre souhaité avec une ligne de code telle que « **val = (MTime, Data[0], Data[1], Data[2], 4)** »
3. On ajoute les données dans une ligne de la BDD à l'aide la la ligne de code « **cursor.execute(sql, val)** »

Le code se termine enfin par une dernier partie « finally » en dehors des autres boucles. C'est dans cette dernier partie que l'on ferme la connexion une fois que toutes les données ont pu être enregistrer