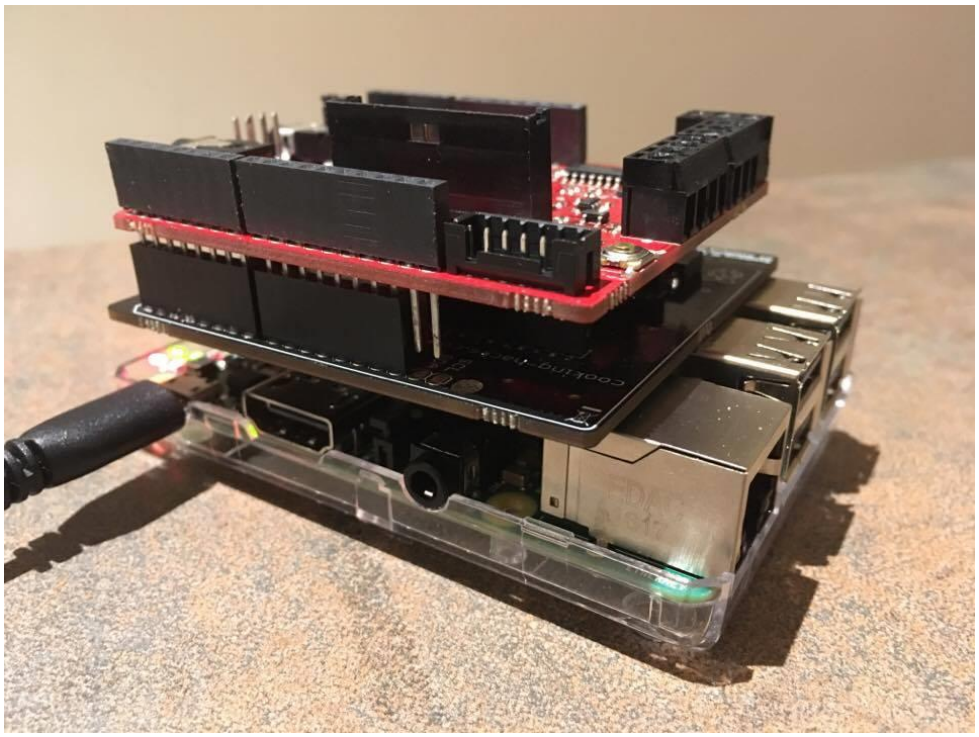
PROJET FAS – ANALYSEUR DE STRESS



IG3

Alex AUFAUVRE – Toinon GEORGET

L'objectif du projet est de récupérer les données d'une personne en situation de stress via des capteurs santé (sudation et température) et de les analyser pour déterminer les effets de la situation sur l'augmentation du stress chez cette personne.

INTRODUCTION

Dans le cadre de ce projet FAS, nous allons nous intéresser au domaine médical, et nous avons choisi d'utiliser pour cela les capteurs du kit santé e-Health de Cooking Hacks.

Problèmes rencontrés

Nous avons malheureusement rencontré de nombreux problèmes concernant le fonctionnement de ces capteurs, des drivers, des librairies et des exemples fournis sur le site officiel. Nous avons testé tous les capteurs, en passant par les combinaisons Raspberry/Shield Intermédiaire/Shield Santé ou Arduino/Shield Santé. Or les données récupérées sont toutes aberrantes et leur exploitation est impossible.
Nous avons donc passé énormément de temps à essayer de faire fonctionner le matériel, et nous avons été contraints de changer de projet de cours de semestre.

Projet initial – Kit Assistance Secourisme

Nous étions initialement partis sur l'idée d'un Kit Assistance Secourisme, l'objectif étant de récupérer des données sur une personne en situation de malaise et les analyser pour donner une conduite à tenir. On utilise des capteurs santé suivants : Pouls/Saturation en O2, Tensiomètre, Dextromètre (mesure du taux de glucose dans le sang), « Air flow sensor » mesure de la ventilation, ECG (électrocardiogramme). Les données sont collectées sur le Raspberry et celui-ci les analyse pour déterminer la cause éventuelle du malaise et les actions à entreprendre (appeler les secours, mettre en position assise...). L'analyse est complétée par une série de questions posées à l'utilisateur concernant l'état de la victime. Pour la communication entre l'utilisateur et la machine, on utilise un écran tactile sur lequel sont affichées les questions et la marche à suivre.

L'idée était de faciliter la prise en charge de la victime par une personne lambda, non formée aux premiers secours, d'éviter les mauvaises manipulations et d'améliorer la rapidité d'intervention des secours en effectuant les premières analyses avant leur arrivée.
Mais nous avons dû revoir nos ambitions à la baisse en cours du projet car les capteurs utilisés ne fonctionnaient pas.

Projet final – Kit Analyseur de Stress

Le projet abordé par la suite est un analyseur de stress. Pour ce projet nous avons gardés deux capteurs du kit santé, notre Raspberry et ses shields. Nous voulions au départ utiliser les capteurs de rythme cardiaque, de sudation et de tension. Deux d'entre eux ne fonctionnant pas nous avons utilisé le Galvanic Skin Response (sudation) et le Body Temperature (température corporelle). En effet, ce sont les deux seuls capteurs pour lesquels nous sommes arrivés à récupérer des données crédibles (bien que le GSR donne des résultats variables).

L'objectif est de mettre une personne dans une situation de stress et de récupérer les données via ces deux capteurs afin de déterminer les effets de la situation sur l'augmentation du stress chez cette personne. Le stress fait l'objet de nombreuses études sérieuses afin d'améliorer le bien-être des individus (en entreprise par exemple), de détecter des problèmes de santé, etc…

L'expérience peut consister à jouer à un jeu vidéo (violent, stressant, …), à poser des questions sur les problèmes de la personne (au sujet du travail, des relations sociales, …) ou alors à s'en servir à la manière d'un détecteur de mensonge par exemple.

L'expérience se déroule de la manière suivante :

- Un calibrage de 30 sec/1min qui prend la moyenne des 10 plus grandes valeurs obtenues en température et la moyenne des 10 plus grandes valeurs obtenues en conductance (GSR) pour une personne dans son état initial.

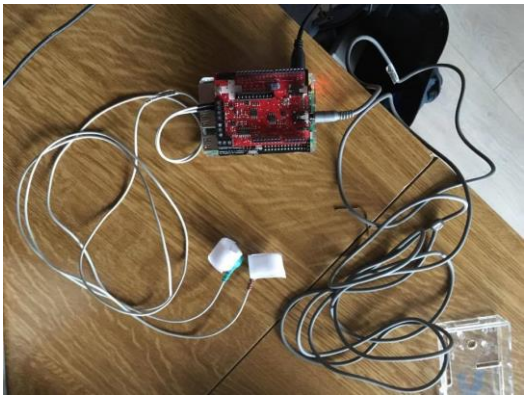
- Une expérience de 3 min qui prend les 30 plus grandes valeurs de conductance et de température et en fait la moyenne pour une personne dans son état de stress.

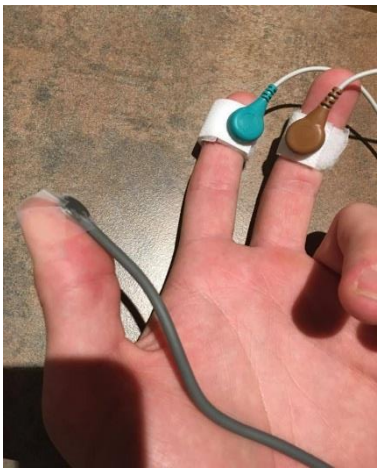
- Un résultat qui fait le rapport de chacune des moyennes et donne l'augmentation en pourcentage.

MODE D'EMPLOI

1) Connecter le shield Arduino sur le shield Intermédiaire puis sur le Raspberry comme sur l'image de présentation du rapport.

2) Brancher le capteur de température sur la prise type « jack » et le capteur de sudation sur les deux « dominos » les plus au centre du shield Arduino. Leur ordre n'est pas important.





3) Disposer les capteurs sur la personne comme sur la photo ci-contre.

4) Compiler le fichier projet FAS dans le dossier du projet en ligne de commande sur le Raspberry de la manière suivante :

```
pi@raspberrypi:~ $ g++ -lpthread -lrt projet_FAS.cpp arduPi.cpp eHealth.cpp -o projet_FAS
```

5) Exécuter le fichier en ligne de commande :

```
pi@raspberrypi:~ $ sudo ./projet_FAS
```

6) Attendre le temps de calibrage (30 sec ou 1 min, défini au début du code) puis effectuer l'expérience (3min, peut être modifié au début du code). Les résultats s'affichent sur le terminal.

MOYENS MATERIELS

Nous utilisons le matériel suivant :

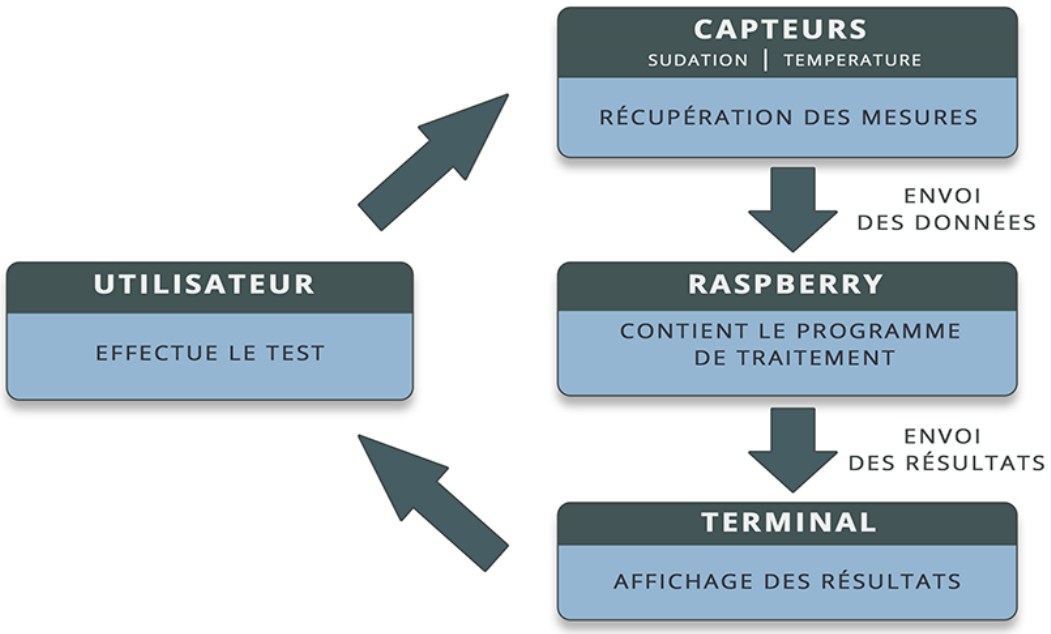
- Raspberry Pi 3
- Shield adaptateur
- Shield santé e-Health
- Capteur Galvanic Skin Response
- Capteur Body Temperature
- Ordinateur avec un terminal

Les deux capteurs santé utilisés sont représentatifs pour la détection du stress, ils vont permettre de pouvoir analyser les symptômes principaux. Nous avons choisi de travailler sur un Raspberry (d'où l'utilisation du shield intermédiaire) car c'est l'installation qui nous permet une meilleure récupération des résultats. Enfin nous avons besoin d'un ordinateur pour lancer le programme et afficher les résultats sur le terminal.

MOYENS HUMAINS

Tout au long du projet, nous avons travaillé ensemble pour le développement du projet, la recherche de documentation, les essais de capteurs, la partie matérielle et la partie logicielle. De plus, Toinon a fait l'acquisition d'un Raspberry Pi 3 personnel pour pouvoir avancer sur le projet à tout moment.

ARCHITECTURE DU PROJET



CODE

Notre projet se compose de 5 fichiers :

- ArduPi.cpp / ArduPi.h : librairie des drivers permettant de récupérer les données depuis le Shield Arduino : ces librairies proviennent directement du site CookingHacks, développeur des capteurs e-Santé et n'ont pas été modifiées
- eHealth.cpp / eHealth.h : librairie des drivers de transition Arduino vers Raspberry : ces libraires proviennent également du site CookingHacks. Nous n'utilisons que les fonctions liées au capteur GSR et température.
- Projet_FAS : fichier contenant notre programme principal ainsi que ses fonctions associées.
    - Main : programme principal qui exécute consécutivement le calibrage et l'expérience ainsi que l'affichage des résultats
    - moyTab : fonction qui calcule la moyenne du tableau de données passé en paramètre
    - insertVal : insert la valeur de manière ordonnée dans la tableau, tous deux passés en paramètre
    - initTab : fonction qui initialise un tableau de float à 0
    - setup : fonction qui initialise un paramètre du capteur GSR

PERSPECTIVES

Si nous disposions de plus de temps et de matériel fonctionnant intégralement, nous aurions pu continuer notre projet initial de Kit Assistance Secourisme. Un mois à se consacrer uniquement à ce projet aurait pu suffire, en s'appuyant sur l'aide de spécialistes du domaine médical, et une commercialisation aurait pu être envisagée.

A l'heure actuelle, concernant le Kit d'Analyse de Stress, notre projet est opérationnel (à condition que les données des capteurs ne soient pas erronées) et permet d'obtenir un résultat significatif de l'augmentation du stress chez un individu selon une situation particulière.
Une analyse plus précise pourrait être proposée en ajoutant les capteurs Pulsioximeter (puslation cardiaque/oxygénation du sang), Blood Pressure (tension artérielle) pour les mesures ou encore ECG (électrocardiogramme).
Pour ce qui est de l'affichage des résultats, une interface graphique via un écran tactile serait un bon moyen d'améliorer l'ergonomie. Cela permettrait une meilleure communication avec l'utilisateur, et un projet davantage « user-friendly ».