Océane Duquenne – Florian Frischmann – Thibaut Lemmel – Thibault Hanna

HeXart Care / NOVEMBRE 2017

DOCUMENTATION TECHNIQUE

Table des matières

[Architecture du projet 2](#_Toc498673566)

[Module 1 : Capteur cardiaque 2](#_Toc498673567)

[montage electronique 2](#_Toc498673568)

[Schéma du montage : 3](#_Toc498673569)

[Apercu 3](#_Toc498673570)

[Code 3](#_Toc498673571)

[Module 2 : Système d’affichage 4](#_Toc498673572)

[Montage 4](#_Toc498673573)

[Apercu 4](#_Toc498673574)

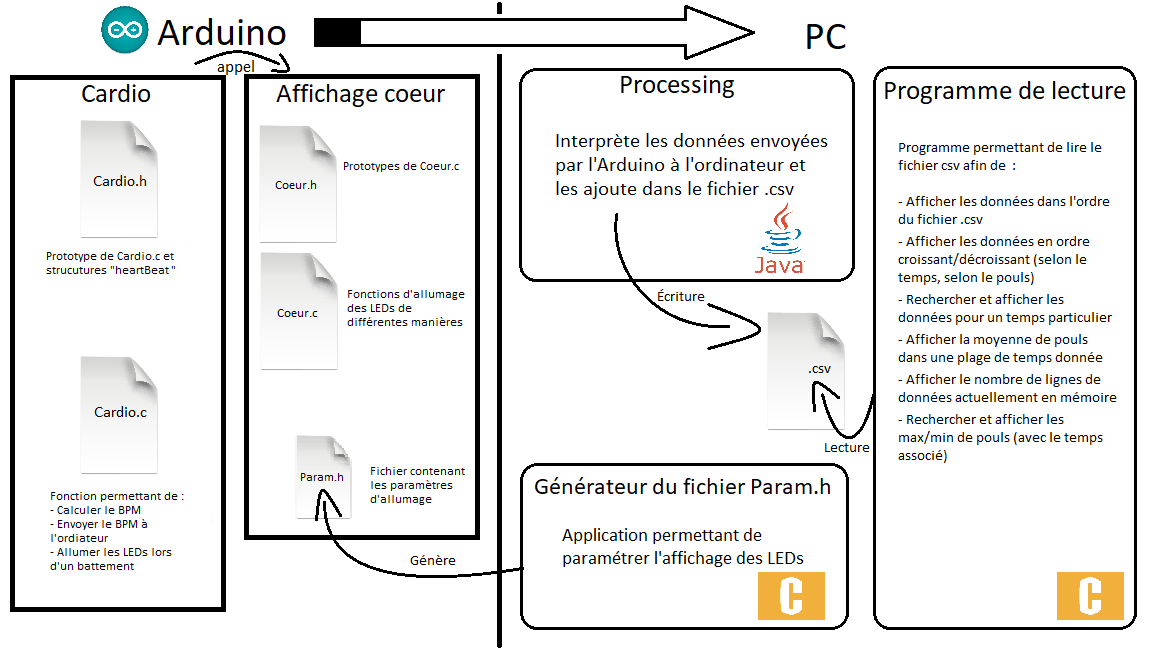
[Code 4](#_Toc498673575)

[Module 3 : Traitement de l’information 4](#_Toc498673576)

[Fonctionnement 4](#_Toc498673577)

[Module 4 : Interface utilisateur 5](#_Toc498673578)

# Architecture du projet

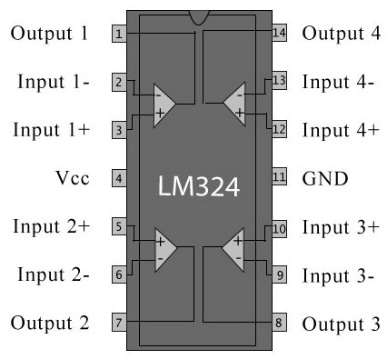
******

# Module 1 : Capteur cardiaque

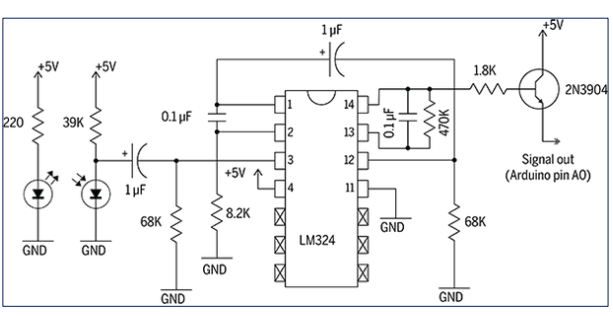
Afin de réaliser notre capteur cardiaque nous devions réaliser un montage et un code qui mis ensemble forme notre prototype capable de donner le nombre de battement par minutes d’un individu.

## montage electronique

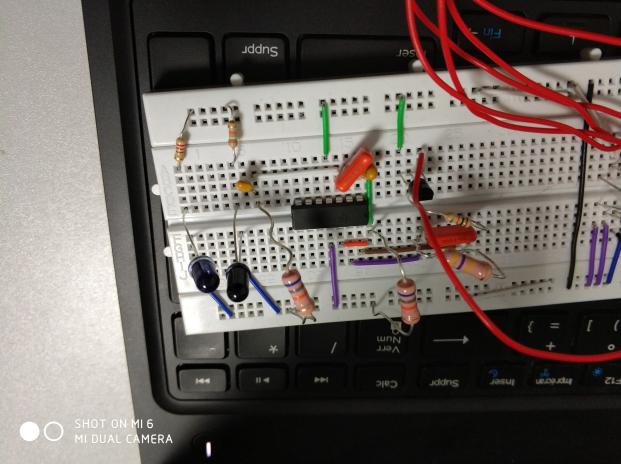
Le montage électronique est composé d’un[e] :

* LED IR (1)
* Phototransistor (1)
* LM324
* Arduino
* Condensateur
  + 1 µf (2)
  + 0.1 µF (2)
* Resistance ohm
  + 220
  + 39K
  + 68K (2)
  + 8.2K
  + 1.8K
  + 470K
* Transistor NPN

### Schéma du montage :



## Apercu



## Code

Le circuit est fait de telle manière à obtenir un signal plutôt constant cependant quand une pulsation arrive dans notre doigt le phototransistor ne vas plus capter autant de lumière durant un cours instant et on pourra donc obtenir un pic au niveau de la tension en sortie A0. C’est grâce à ce pic que nous pouvons faire un traitement coté logiciel de telle manière à calculer le nombre de battements des utilisateurs. On pourra donc considérer qu’à chaque fois où notre valeur passe au-dessus d’un certain seuil, il s’agit d’un battement.

En suivant ce principe il nous suffit d’enregistrer tous les pics au-dessus de ce seuil sur un temps donné puis effectuer un produit en croix, si on fait une acquisition sur 10 sec il suffira de faire le nombre de battements obtenue sur 10 sec \* 60 / 10 pour obtenir notre nombre de battements par minute.

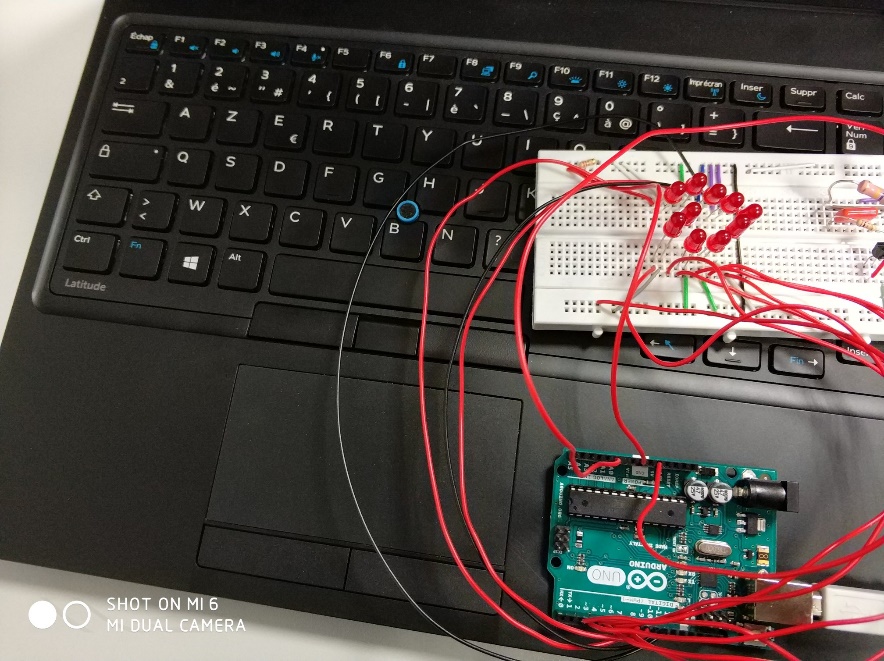
Sur un circuit où le signal n’est pas très bien capté, il peut être nécessaire d’attendre que le pic redescende avant de poursuivre l’exécution du programme pour éviter de compter plusieurs fois le même battement.

# Module 2 : Système d’affichage

## Montage

L’idée ici est de réalisé un module permettant de faire clignoter les LEDs à chaque battement du cœur capté par le module 1, nous devons permettre à l’utilisateur de sélectionner parmi plusieurs manières l’affichage des LEDs. Nous sommes donc obligés de brancher chaque LED sur un pin diffèrent de l’Arduino afin de pouvoir les allumer des manières indépendantes.

## Apercu



## Code

Il nous suffit de créer une fonction permettant l’allumage des LEDs en fonction du mode qui est en define dans le fichier param.h, Et une fonction permettant d’éteindre toutes les LEDs afin d’allumer les LEDs tant que l’on a un pique au niveau de la sortie A0 de l’Arduino puis une fois avoir réobtenue une valeur normale rééteindre les LEDs.

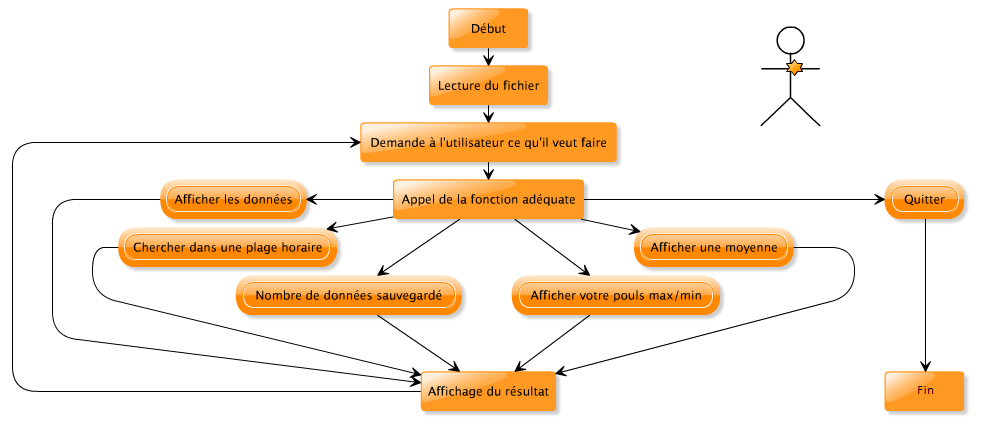
En ce qui concerne la génération du fichier param.h on fait un menu avec des choix pour l’utilisateurs en fonction de son choix on écrit dans le fichier param.h « define mode choix » et si nécessaire on peut faire un second define pour ajouter une information par exemple si on prend le cas du mode allumer une seul led au choix on doit dire qu’elle mode sa sera ainsi que quel LED allumé on pourra ajouter un « #define parameter NUMERODELALED » une fois avoir écrit dans notre fichier on peut ré-envoyer notre projet Arduino grâce à la commande system(‘’arduino –upload notrefichier.ino’’) cette commande ne fonctionne que si on ajoute dans le variable d’environnement Path le répertoire d’Arduino.

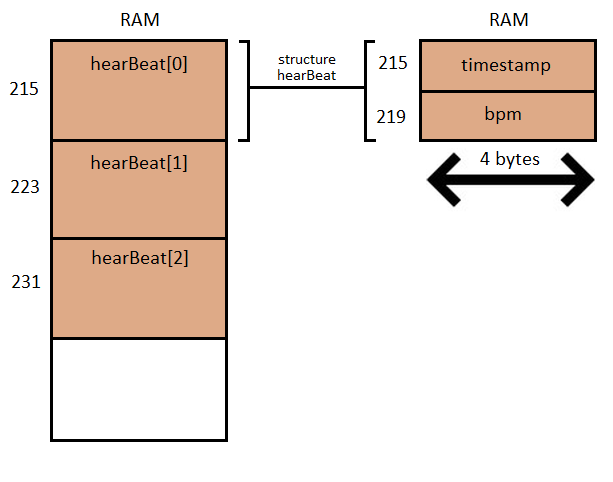
# Module 3 : Traitement de l’information

## Fonctionnement

Le module 3 a pour objectif de récupérer toutes les données envoyées de l’Arduino à l’ordinateur par le port Serial et de l’enregistrer dans un fichier. Nous avons effectué quelques modifications par exemple le fichier ne se supprime plus à chaque fois où on lance Processing, et au lieu d’enregistrer le temps d’exécution de la carte Arduino on enregistre le timestamp correspondant à la date où le pouls a été enregistré.

# Module 4 : Interface utilisateur

Le module 4 permet de lire le fichier enregistré par le module 3 et de le trier.

Pour cela on enregistre le contenu4 du fichier dans un tableau de structure qui sera ensuite plus évident à manipuler.