Océane Duquenne – Florian Frischmann – Thibaut Lemmel – Thibault Hanna

HeXart Care / NOVEMBRE 2017

DOCUMENTATION TECHNIQUE

Table des matières

[Architecture du projet 1](#_Toc498631884)

[Module 1 : Capteur cardiaque 1](#_Toc498631885)

[montage electronique 1](#_Toc498631886)

[Schéma du montage : 2](#_Toc498631887)

[APERCU 2](#_Toc498631888)

[Code 2](#_Toc498631889)

[Module 2 2](#_Toc498631890)

[Montage 2](#_Toc498631891)

[cODE 2](#_Toc498631892)

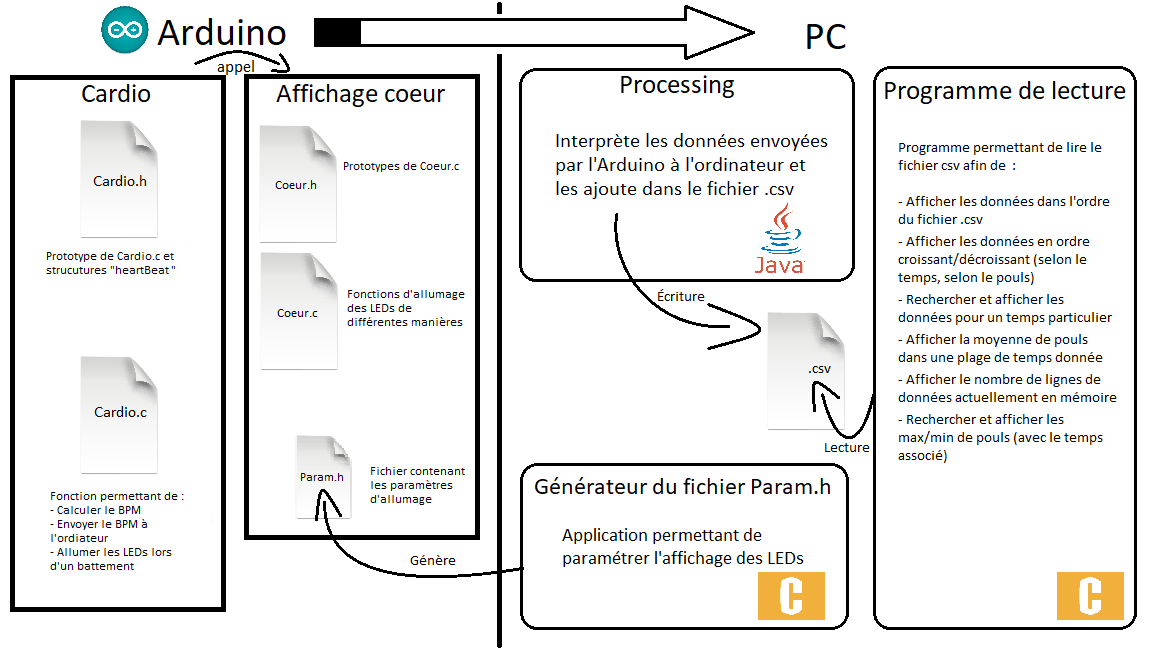
[Module 3 2](#_Toc498631893)

[Fonctionnement 3](#_Toc498631894)

[Module 4 3](#_Toc498631895)

[Module assemblé 3](#_Toc498631896)

# Architecture du projet

******

# Module 1 : Capteur cardiaque

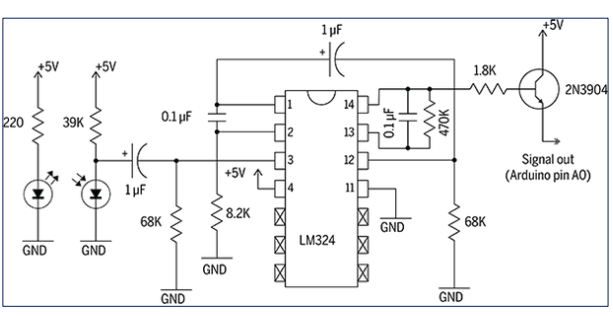
Afin de réaliser notre capteur cardiaque nous devons réaliser un montage et un code qui mis en semble forme un logiciel capable de donner le nombre de battement par minute d’un individu.

## montage electronique

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

* LED IR (1)
* Phototransistor (1)
* LM234
* 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 ou le transistor ne vas plus capter autant de lumière durant un cours instant et on pourra donc obtenir un pique au niveau de la tension en sortie A0. C’est grâce a ce pique que nous pouvons faire un traitement coté logiciel de telle manière à calculer le nombre de battement des utilisateurs. On pourra donc considérer qu’à chaque fois ou notre valeur passe haut dessus d’un certain seuil il s’agit d’un battement.

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

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

# Module 2 : système d’affichage

## Montage

L’idée ici et 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 manière l’affichage des LEDs nous sommes donc obligé de brancher chaque LED sur un pin diffèrent de l’Arduino afin de pouvoir les allumer des manières indépendantes.

## cODE

Il nous suffit donc 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.

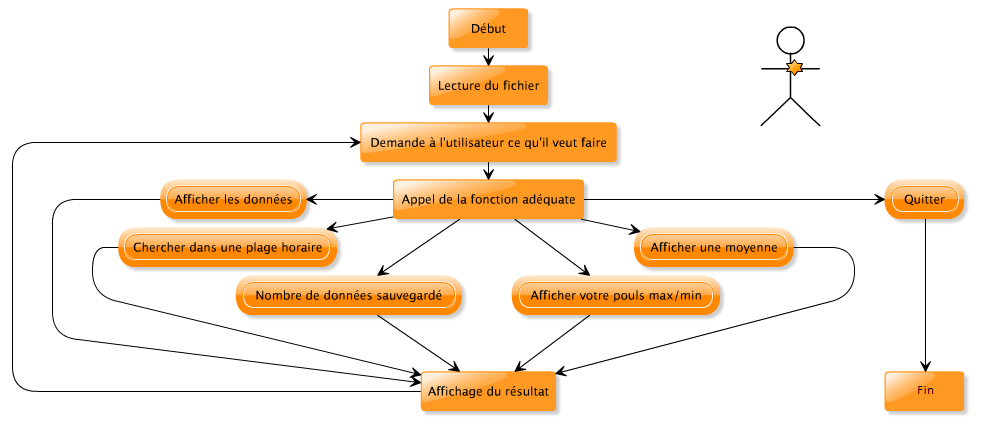
En ce qui concerne la génération du fichier param.h on fait un menu avec des choix pour l’utilisateurs en fonction de son choix on écrit dans le fichier param.h « define mode choix » et si nécessaire on peut faire un second define pour ajouter une information par exemple si on prend le cas du mode allumer 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.

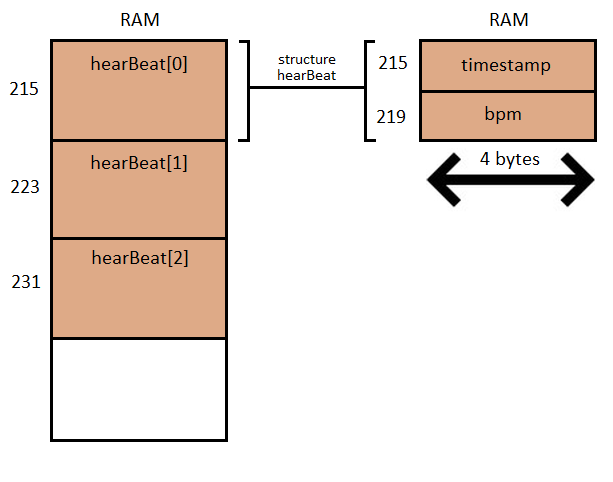
# Module 3

## Fonctionnement

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

# Module 4

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

Pour cela on enregistre le contenue du fichier dans un tableau de structure qui sera ensuite plus évident à manipule