

# HeXart Care

## Projet 1 : Cardiofréquencemètre

Vicart Clément | Billau Claire | Boivin Théo | Leroux Hugo

# Projet 1 : Cardiofréquencemètre

## TABLE DES MATIERES

TABLE DES MATIERES.....	1
TABLE DES FIGURES.....	1
INTRODUCTION .....	2
<i>Equipe :</i> .....	2
<i>Projet :</i> .....	3
ORGANISATION DU PROJET .....	4
<i>Gestion du projet</i> .....	4
<i>Réunions</i> .....	5
<i>Planning prévu</i> .....	6
<i>Planning réalisé</i> .....	6
CHOIX TECHNIQUES.....	7
<i>Logiciels utilisés</i> .....	7
<i>Maquette des montages des modules</i> .....	10

## TABLE DES FIGURES

Figure 1, 2, 3, 4, 5 :	Page 4
Figure 6 :	Page 5
Figure 7 :	Page 9
Figure 8, 9 :	Page 10
Figure 10, 11:	Page 11

# Projet 1 : Cardiofréquencemètre

## INTRODUCTION

Ce compte rendu a pour but de présenter chaque module réalisé par tous les membres de l'équipe. Nous avons pour chaque module (1, 2, 3 et 4) des images, schémas et fichiers sources des montages et logiciels réalisés et publiés sur GitHub.

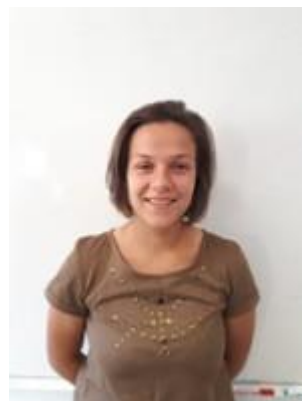
Des extraits de ces fichiers seront également présents en annexe.

### Equipe :

Ce projet a été réalisé par Clément Vicart (chef de projet), Théo Boivin, Claire Billau et Hugo Leroux.



*Clément Vicart*



*Claire Billau*



*Hugo Leroux*



*Théo Boivin*

# Projet 1 : Cardiofréquencemètre

## Projet :

Le projet, nommé HeXart Care, comporte 4 modules permettant finalement de livrer un cardiofréquencemètre grâce à une carte Arduino, des LEDS en forme de cœur et une LED infrarouge et un capteur.

- ➔ Le premier module s'appelant « cardio », s'appuie sur un schéma de base d'un montage composé d'une carte Arduino, de résistances, de condensateurs, d'une LED, d'un multiplicateur, d'une LED infrarouge, d'une pince pour doigt et enfin d'un récepteur infrarouge afin de mesurer les variations dues au rythme cardiaque d'une personne dont le doigt se trouve entre la LED infrarouge et du récepteur.  
Il est séparé en 2 parties dont le premier est de réaliser le montage et le deuxième de programmer l'Arduino afin qu'il calcule le pouls de la personne et envoie les informations sur la sortie série au format CSV.
- ➔ Le deuxième module s'appelle « cœur de LEDS », dont le but est de créer un montage de LEDS en forme de cœur, et de les faire clignoter selon le rythme cardiaque de la personne. Ce module est séparé en 3 parties : la première est le montage du circuit, la deuxième est la programmation de l'Arduino pour allumer les LEDS selon une configuration prédéfinie, et enfin la troisième est la programmation d'un logiciel permettant de demander à l'utilisateur quelle configuration d'allumage des LEDS il souhaite utiliser.
- ➔ Le troisième module permet, grâce à un logiciel nommé Processing, de récupérer les valeurs de pouls au format CSV envoyées par l'Arduino sur la sortie série et de les enregistrer dans un fichier « .csv » afin de les traiter par la suite.
- ➔ Enfin, le dernier module est destiné au traitement du fichier « .csv ». Il consiste à créer un programme afin de charger les données du fichier CSV et de les traiter afin de les afficher à l'utilisateur. Ce traitement peut être de les trier afin de les afficher dans l'ordre croissant ou décroissant en fonction du temps, etc...

Ces différents modules, assemblés ensemble permettent au final de créer un cardiofréquencemètre qui : permet à l'utilisateur de choisir la méthode d'allumage des LEDS, fait clignoter ces LEDS en fonction de son rythme cardiaque, de récupérer et stocker les données du rythme cardiaque et enfin de traiter les informations et de les afficher à l'utilisateur.



# Projet 1 : Cardiofréquencemètre

Ces « maps » visuelles nous ont permis d'organiser les tâches par importance en fonction du temps. Elles nous ont également permis de voir toutes les tâches à effectuer pour chaque module et ainsi d'organiser nos fichiers, nos schémas et ne rien oublier.

Afin de savoir chaque étape réalisée par tout le monde, un journal de bord a été tenu à jour afin de savoir, pour chaque jour et horaire, les étapes commencées et terminées ainsi que les réunions.

Chacun à donc commencé de son côté : pour les montages, ils ont premièrement été dessinés avec Fritzing. Pour le code, des schémas de fonctionnement ont été réalisés afin de savoir exactement comment organiser le code et les fichiers.

Nous pouvons voir par exemple, pour le module 2 :

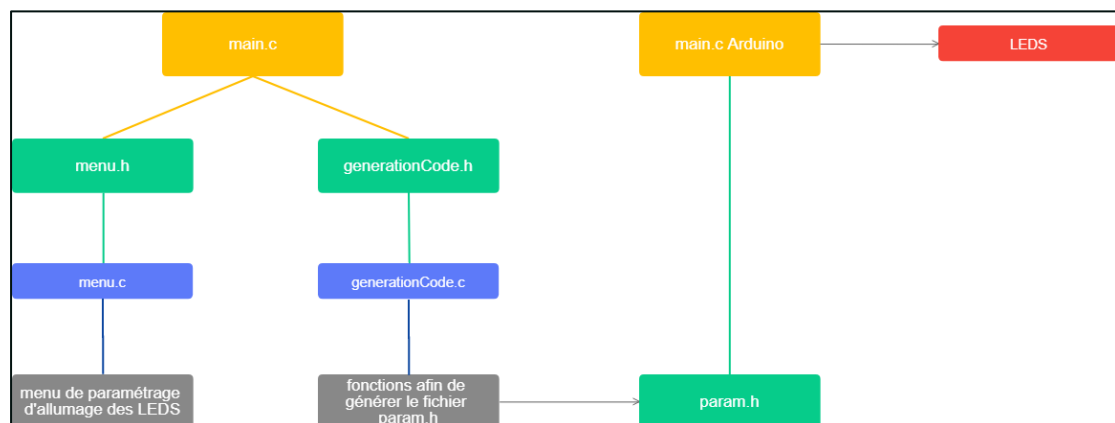


Figure 6

## Réunions



Avec notre équipe, nous avons également planifiés des réunions au titre d'une par jour. Ainsi, nous pouvions communiquer et de savoir chaque jour l'avancement de chaque personne dans leur module et/ou leurs tâches, de savoir pour chaque personne ce qui leur restait à faire et

# Projet 1 : Cardiofréquencemètre

ainsi optimiser la charge de travail, en répartissant les tâches les plus complexes sur plusieurs personnes.

De ce fait, pour une personne bloquée nous avons pu répartir la/les tâche(s) avec une autre personne, et une personne qui avançait beaucoup pouvait ainsi aider quelqu'un d'autre sur un module ou une tâche.

Ces réunions entre membres de l'équipe sont très importantes également pour la gestion de projet, afin d'optimiser au maximum la vitesse d'avancement des modules.

Nous avons également réalisé 2 réunions entre chefs de projets, afin de parler sur l'avancement de chaque groupe, comparer notre propre avancement sur les autres et nous échanger des méthodes de travail, des solutions et des méthodes de gestion.

## Planning prévu

Nous avons, dès le début du projet, organisé un planning, jour par jour, de l'avancement prévu de chaque tâche de chaque module afin d'organiser notre charge de travail sur la semaine prévue pour le projet.

Ce planning prévu est le suivant, pour chaque personne :



Planning prévu.xlsx

Certaines tâches ayant pris plus ou moins de temps dû à certains blocages ou problèmes, nous avons donc répondu à un autre planning que celui-ci afin d'organiser au mieux la gestion de ce projet.

## Planning réalisé

Finalement, le planning suivant a été réalisé pour chaque personne :



Planning  
réalisé.xlsx

# Projet 1 : Cardiofréquencemètre

## CHOIX TECHNIQUES

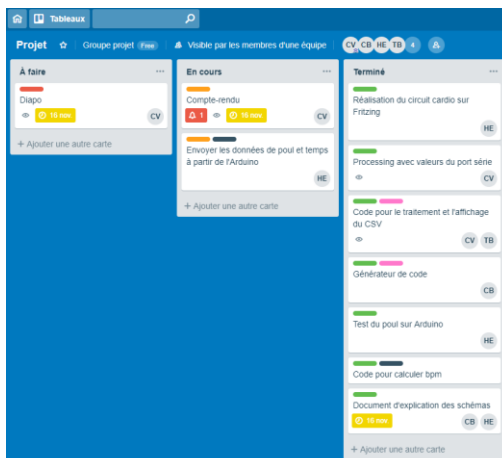
### Logiciels utilisés



Nous avons utilisé différents logiciels afin de simplifier certaines tâches comme : la communication, le partage de fichiers, le partage de code, la création de maps visuelles, de schémas...



Tout d'abord, afin de simplifier la communication et de savoir à tout instant les tâches à réaliser, celles en cours de réalisation et celles terminées, nous avons utilisé Trello. Trello est un logiciel de gestion des tâches permettant notamment :



- ➔ De créer des listes afin de trier les différentes tâches,
- ➔ De créer des « étiquettes » afin de donner des codes couleurs à certaines tâches,
- ➔ D'assigner des membres à certaines tâches,
- ➔ De définir une date limite pour laquelle une date doit être réalisée.

Pour notre projet, Nous avons créé 3 listes pour 3 types de tâches : A faire, En cours, Terminée.



# Projet 1 : Cardiofréquencemètre



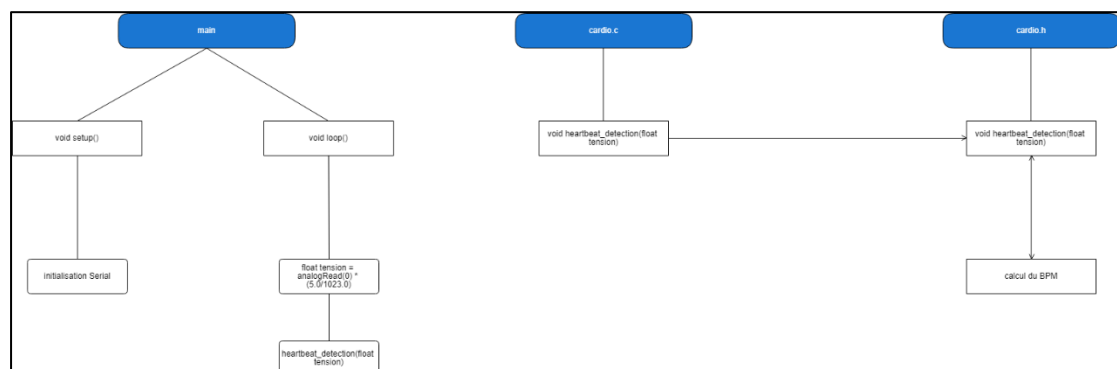
Ensuite, pour la création des « maps » visuelles, nous avons choisi Mindmeister. Ce logiciel en ligne nous permet de créer des bulles, contenant des mots clés afin de structurer nos idées et notre organisation en créant des liens entre chaque bulle, créant ainsi une hiérarchie. Nous nous sommes servis de ce logiciel afin de structurer les différents modules et ainsi avoir les principales lignes directrices de notre projet (voir les figures 1 à 5).



Cacoo est également un logiciel en ligne permettant de créer une structure visuelle (un organigramme). Ce logiciel nous a permis de structurer notre code en y insérant les noms de fichiers, de fonctions et/ou leur utilité avant de commencer à programmer afin d'optimiser la vitesse développement et de communication entre les programmeurs par la suite.

Ainsi, chaque programme possédant son organigramme associé est plus facile à développer car on sait à l'avance de quelles ressources nous auront besoin.

Voici un exemple de réalisation d'un organigramme avec Cacoo :



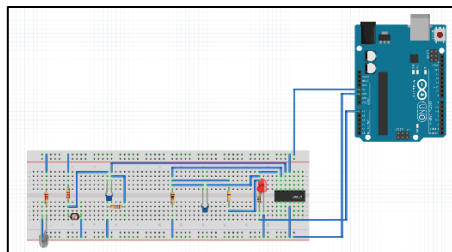
# Projet 1 : Cardiofréquencemètre



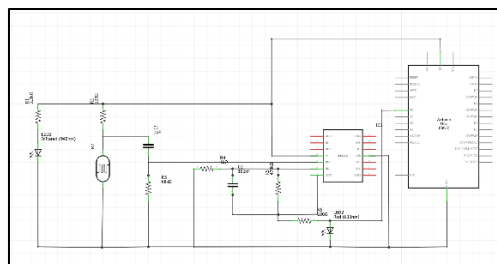
Ensuite, afin de modéliser nos montages électroniques et de les schématiser, nous avons utilisé le logiciel Fritzing. Ce logiciel permet de créer des schémas électroniques, puis de les exploiter afin de créer des vues « platine », c'est-à-dire ces mêmes circuits mais avec des textures pour chaque composant, afin de mieux visualiser le circuit électronique.

Ce logiciel a été utilisé pour créer les schémas et les vues platines des montages de cardiofréquencemètre et du cœur de LEDs.

Nous avons donc par exemple :



*Figure 6 (vue platine)*



*Figure 7 (vue schématique)*



Enfin, afin de pouvoir partager notre code et nos fichiers rapidement, nous avons utilisé le site GitHub avec le logiciel Git.

# Projet 1 : Cardiofréquencemètre

Git est un logiciel permettant de télécharger du code sur des « repositories » en ligne et à la vue de tous, et modifiable par toutes les personnes autorisées.

Ce logiciel intègre également des fonctionnalités de gestion de version, ce qui permet à chaque membre de l'équipe de consulter toutes les modifications apportées à un document ou à un code source.

Grâce à ce logiciel, nous pouvons rapidement nous échanger notre code afin de corriger certaines erreurs mais également de sauvegarder tout notre projet.

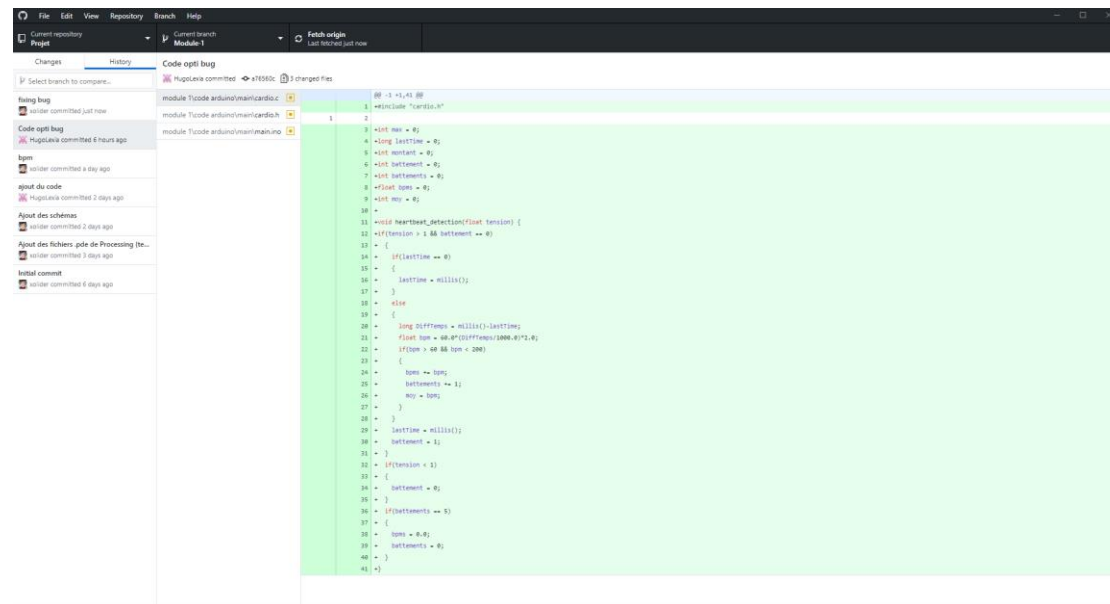


Figure 7 (capture d'écran du logiciel graphique GitHub)

## Maquette des montages des modules

Module 1 :

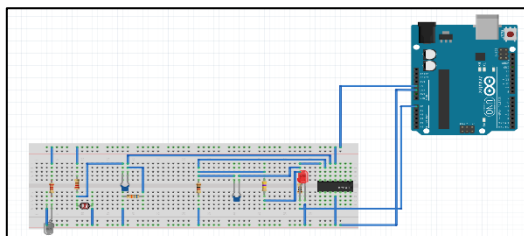


Figure 8

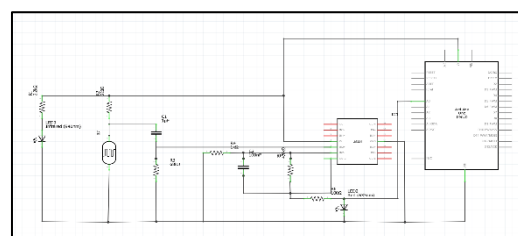


Figure 9

Afin de réaliser ce module, les tâches ont été réparties en différentes parties :

# Projet 1 : Cardiofréquencemètre

D'abord, le schéma électrique, réalisé d'abord sur Fritzing afin d'avoir une vue schématique du circuit à réaliser. Pour réaliser ce montage, nous avons eu besoin d'une carte Arduino, une LED émettrice infrarouge, un récepteur infrarouge, un multiplexeur LM324N, 6 résistances (100, 220, 1k, 2.2k, 68k et 470k  $\Omega$ ), 2 condensateurs (1 $\mu$ F et 100nF) et une LED rouge.

Seul problème, le composant multiplexeur n'existait pas sur Fritzing. Il a donc été importé depuis Internet.

Ensuite, une fois le schéma terminé sur Fritzing, la vue platine a été générée avec les composants requis, mais pas placés. Après placement des composants, nous obtenons donc une vue correspondant au circuit schématique.

Une fois ces schémas terminés, nous avons fait la réalisation du montage sur breadboard. Les LEDs émettrice et réceptrice servent à capter le pouls grâce au volume de sang présent dans la veine du doigt. Celles-ci sont reliées chacune à une résistance afin d'éviter leur dégradation. Une fois celui-ci capté, le signal passe dans un filtre passe-bande composé de condensateurs et plusieurs résistances, puis amplifié par le multiplexeur.

Le montage correspondant à ce module est le suivant :

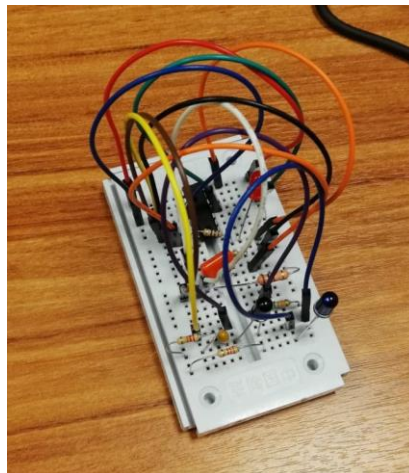


Figure 10

Module 2 :

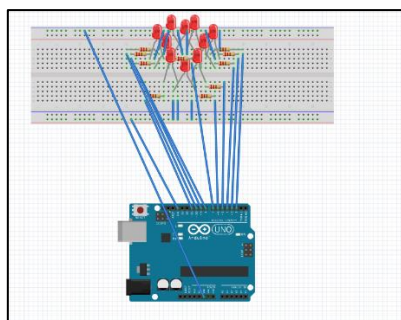


Figure 10 (vue platine)

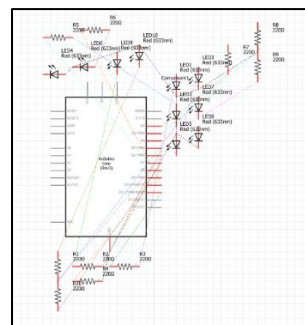
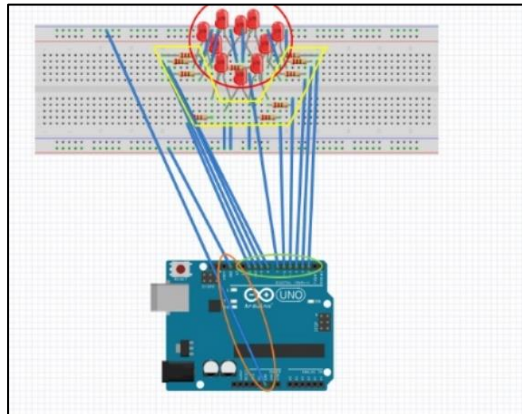


Figure 11 (vue schématique)

# Projet 1 : Cardiofréquencemètre

Pour ce montage, nous avons eu besoin de : 10 LEDS rouges, 10 résistances de 220  $\Omega$ , une vingtaine de câbles, une breadboard et une carte Arduino UNO.

Avant de commencer le montage, nous avons utilisé Fritzing afin de modéliser et schématiser le montage électronique.



Entouré en rouge, nous pouvons voir les LEDS qui doivent représenter un cœur. Ces LEDS sont contrôlées unitairement, donc ne sont pas montées en série mais à l'unité sur la carte Arduino. De plus, il a fallu les brancher de façon que le courant passe dans le bon sens de la LED.

Entouré en jaune, On peut trouver les résistances. Celles-ci ont pour rôle de s'opposer au courant afin de baisser sa charge dans le circuit pour ne pas faire griller les LEDS. Chaque

résistance sont branchées en série avec une LED.

Nous avons ensuite les câbles permettant de relier chaque LED à l'Arduino. Chaque LED est branchée sur une sortie digitale de l'Arduino, permettant ensuite de les programmer à l'unité.

Enfin, un câble relie également toutes les LEDS à la masse (GND) de la carte Arduino

Nous avons donc finalement un montage permettant d'allumer une série de 10 LEDS indépendamment les unes des autres :

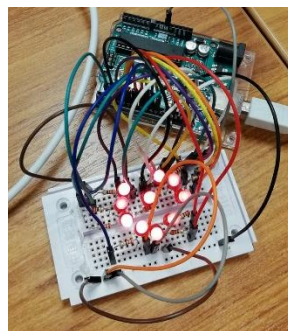


Figure 13

## Phase de tests

# Projet 1 : Cardiofréquencemètre

C'est durant la phase de tests que nous avons rencontrés le plus de problèmes. Cette étape est cruciale dans le projet, elle permet de mettre en pratique le fonctionnement de tous les modules, indépendamment des uns des autres en premier lieux, puis tous réunis.

Cette phase à durer 2 jours, nécessaires au bon fonctionnement de chaque module.

Tout d'abord, nous avons commencé par le cœur de LEDS : Nous avons essayé plusieurs codes afin de tester leur bon fonctionnement unitaire, c'est-à-dire leur fonctionnement indépendant des autres LEDS. Passé ce test, nous avons testé plusieurs méthodes d'allumage des LEDS comme indiqué dans le cahier des charges du projet.

Ensuite, nous avons réalisé les tests du module 4, à savoir afficher et organiser les valeurs d'un fichier CSV avec le temps en première colonne et le pouls en seconde. Ce module nous a posé beaucoup d'erreurs dus à la compilation : erreurs de déclarations, de types de données, d'affichage, etc...

Ensuite, nous avons testé le module 1. C'est ce module qui nous a posé le plus de problèmes : nous devons réussir à récupérer les pics de fréquences sur l'entrée analogique de la carte Arduino, puis trouver un calcul afin de les convertir en battements par minutes par rapport au temps. La majorité des problèmes ont résidés dans la récupération des valeurs sur l'entrée analogique : Comment placer la LED et le récepteur afin de recevoir un pouls, comment transformer ces valeurs en battements par minutes et surtout comment l'implémenter en C Arduino.

Enfin, nous avons testé la récupération des battements par minutes grâce au Logiciel Processing et les enregistrer dans un fichier CSV. Le plus gros problème de ce test a été de récupérer les valeurs en intégrité : parfois, seulement une partie de ce qu'envoyait la carte Arduino sur la sortie Série était récupérée par le logiciel Processing.

Finalement, nous avons réussi à régler la majorité des problèmes liés à ces modules grâce à des solutions fournies par Internet ou par des personnes tierces.

Nous avons également rencontré beaucoup de problèmes lors de la fusion des modules, en particulier le module 1 et 2. En effet, fusionner les codes Arduino n'a pas été facile, en raison de la compilation pour cette carte. Par exemple, nous avons dus trouver un moyen pour afficher des valeurs sur la sortie série à partir du fichier principal de l'Arduino, car elle ne nous permet pas de le faire dans un autre fichier que celui-ci. Pour résoudre ce problème, nous avons par exemple utilisé une fonction « callback », qui est transmise en paramètre d'appel d'une autre fonction.

# Projet 1 : Cardiofréquencemètre

## Réalisation

### Module 1 :

La réalisation de ce module a été effectuée en gardant les circuits électroniques présentés dans la partie maquette, et le code a été revus plusieurs fois dans la phase de tests pour correspondre au mieux aux attentes.

Le code se trouve en annexe de ce document et peut être retrouvé sur GitHub

### Module 2 :

La réalisation de ce module a également été faite en gardant le même schéma présenté dans la partie Maquette. Le code a également été remanié plusieurs fois à cause de bugs et d'erreurs.

Ce code se trouve également en annexe et sur GitHub

### Module 3 :

Le code du module 3 à légèrement été modifié afin de palier à certains bugs. Le code se trouve en annexe et sur GitHub

### Module 4 :

Ce module a nécessité de grosses performances en développement, manipulant des structures de données, des fichiers...

Une partie du code de ce module se trouve en annexe et sur GitHub

# Projet 1 : Cardiofréquencemètre

## CONCLUSION

### Perspectives d'évolution

Les perspectives d'évolution visent les fonctionnalités que le projet pourrait avoir en plus afin de l'améliorer à destination de l'utilisateur final. Comme amélioration, nous pouvons par exemple ajouter :

- Un buzzer qui effectue un « bip » à chaque battement du cœur,
- Une boîte afin de cacher toute la partie électronique et protéger les composants,
- Un bouton afin de lancer l'enregistrement des battements et pour arrêter cet enregistrement,
- ...

### Bilan

Bilan personnel :

Claire :

« J'ai trouvé ce projet intéressant car il reprenait l'ensemble des points étudiés depuis le début de l'année. J'ai pu retravailler certaines notions où j'avais eu du mal.

Réaliser le projet est sympathique car il nous permet de travailler également en autonomie. Dès qu'il y avait un problème ou une erreur, le but était de trouver la solution la plus efficace dans un temps limité.

Il y avait une bonne ambiance dans le groupe, avec aucunes tensions. Nous cherchions toujours à nous entraider sur les différents modules. Tout le monde s'est bien impliqué dans le projet, chacun à son niveau et avec ses compétences.

C'était un bon premier projet, qui donne envie d'en réaliser d'autres. »

Théo :

« J'ai beaucoup aimé ce projet pour plusieurs raisons

Tout d'abord sur le plan technique,

J'ai trouvé ce projet intéressant dans un premier temps parce qu'il reprend tout ce que nous avons vu durant les prosits et les CER.

Mais aussi car ayant à réaliser le module 4, il a été consisté exclusivement de langage C, ce que je préfère par apport au montage électronique, de plus cela m'a permis de renforcer mes connaissances en C et donc de m'améliorer

Il m'a également plus sur le plan humain dans le sens ou le fait de travailler en groupes est différents de ce dont on a l'habitude



# Projet 1 : Cardiofréquencemètre

Notre groupe a été très dynamique ce qui a été un gros avantage pour la réalisation de ce projet, tout le monde a pu s'entraider et c'était donc bien plus sympa que de travailler seul dans son coin. »

Hugo :

« Ce premier projet aura été très intéressant sur plusieurs points. Premièrement, la façon de travailler change de celle adoptée en pro et le travail d'équipe ainsi que la cohésion de groupe sont très mis en avant. Deuxièmement, la mise en application des connaissances permettait de se rendre vraiment compte de ce que l'on est capable de réaliser. Malgré mes difficultés, le module 1 a été fini et les autres membres du groupe étaient présents en cas de besoin. L'ambiance de travail était vraiment très agréable et sérieuse. Le contenu du projet était très intéressant car il permet de se rendre compte de la portée et de la diversité des applications possibles de nos connaissances. »

Clément :

« Ce premier projet m'a permis d'en apprendre davantage sur l'électronique et tout ce que l'on peut en faire, tous les produits que nous pouvons créer grâce à ça. Il m'a également permis de découvrir en profondeur la programmation Arduino et toutes ses applications possibles.

J'ai découvert également grâce à ce projet les applications possibles de tous les composants que nous avons utilisés, liés à leur programmation. »

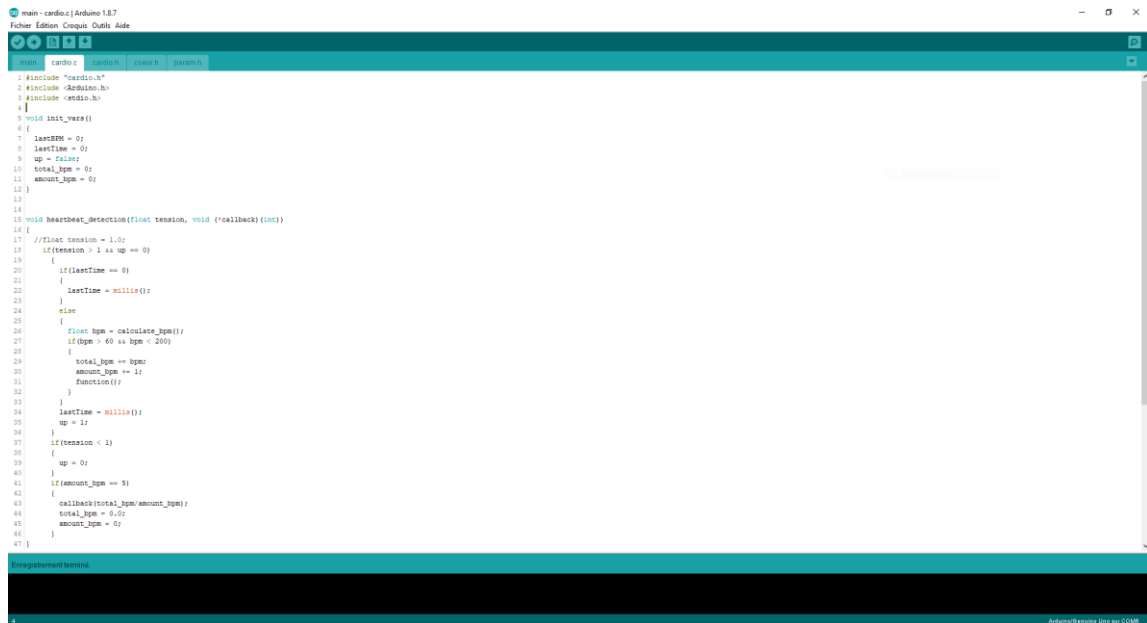
## Bilan de groupe :

La communication a été notre point clé dans ce projet. Nous avons beaucoup travaillé sur ce point afin d'avancer efficacement, et c'est ce qui nous a permis de réussir chaque tâche de chaque module du projet. L'ambiance générale du groupe a été bonne, et chacun a su aider quelqu'un d'autre en cas de besoin. Ce projet nous a beaucoup apporté personnellement et socialement, il nous a appris à travailler en groupe pour le même but.

## ANNEXE

Module 1 :

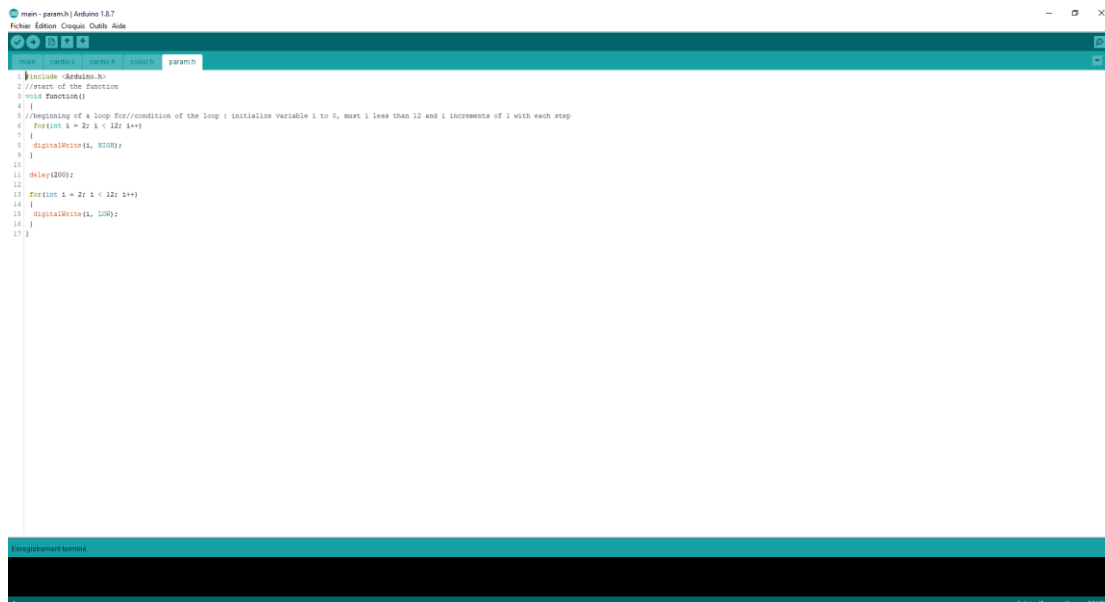
# Projet 1 : Cardiofréquencemètre



```
1 #include "cardio.h"
2 #include <Arduino.h>
3 #include <arduino.h>
4
5 void init_vars()
6 {
7     lastBPM = 0;
8     lastTime = 0;
9     up = false;
10    total_bpm = 0;
11    amount_bpm = 0;
12 }
13
14
15 void heartbeat_detection(float tension, void (*callback)(int))
16 {
17     //float tension = 1.0;
18     if(tension > 1 || up == 0)
19     {
20         if(lastTime == 0)
21         {
22             lastTime = millis();
23         }
24         else
25         {
26             float bpm = calculate_bpm();
27             if(bpm > 60 || bpm < 200)
28             {
29                 total_bpm += bpm;
30                 amount_bpm += 1;
31                 callback();
32             }
33             lastTime = millis();
34         }
35         up = !up;
36     }
37     if(tension < 1)
38     {
39         up = 0;
40     }
41     if(amount_bpm == 5)
42     {
43         callback(total_bpm/amount_bpm);
44         total_bpm = 0;
45         amount_bpm = 0;
46     }
47 }
```

*cardio.c*

## Module 2 :

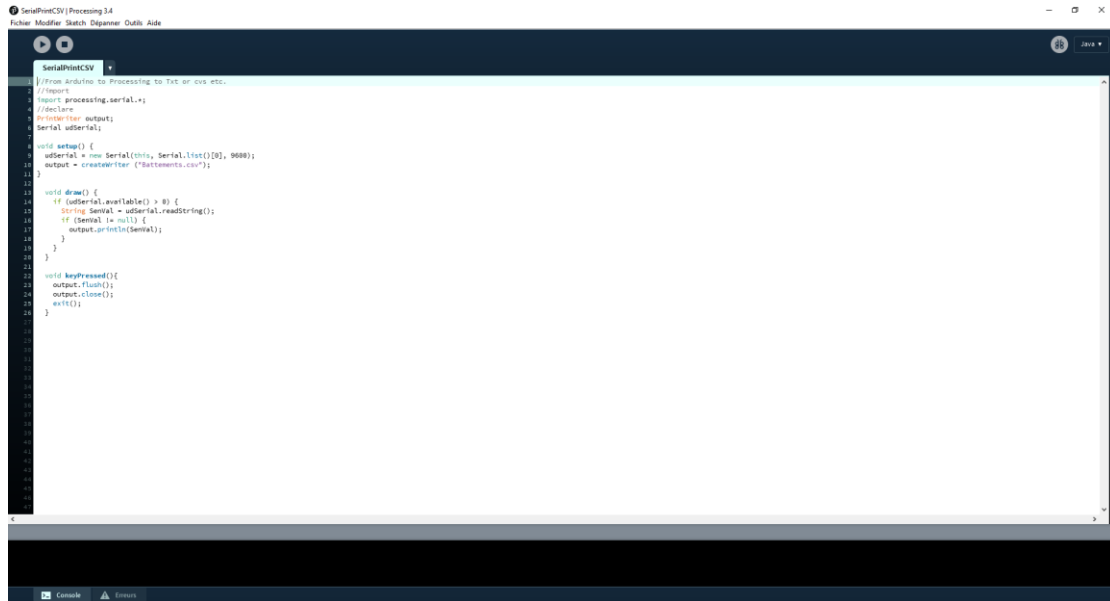


```
1 #include <Arduino.h>
2 //start of the function
3 void Function()
4 {
5     //beginning of a loop for/condition of the loop : initialize variable i to 2, must i less than 12 and i increments of 1 with each step
6     for(int i = 2; i < 12; i++)
7     {
8         digitalWrite(1, HIGH);
9     }
10
11    delay(200);
12
13    for(int i = 2; i < 12; i++)
14    {
15        digitalWrite(1, LOW);
16    }
17 }
```

*exemple d'un param.h*

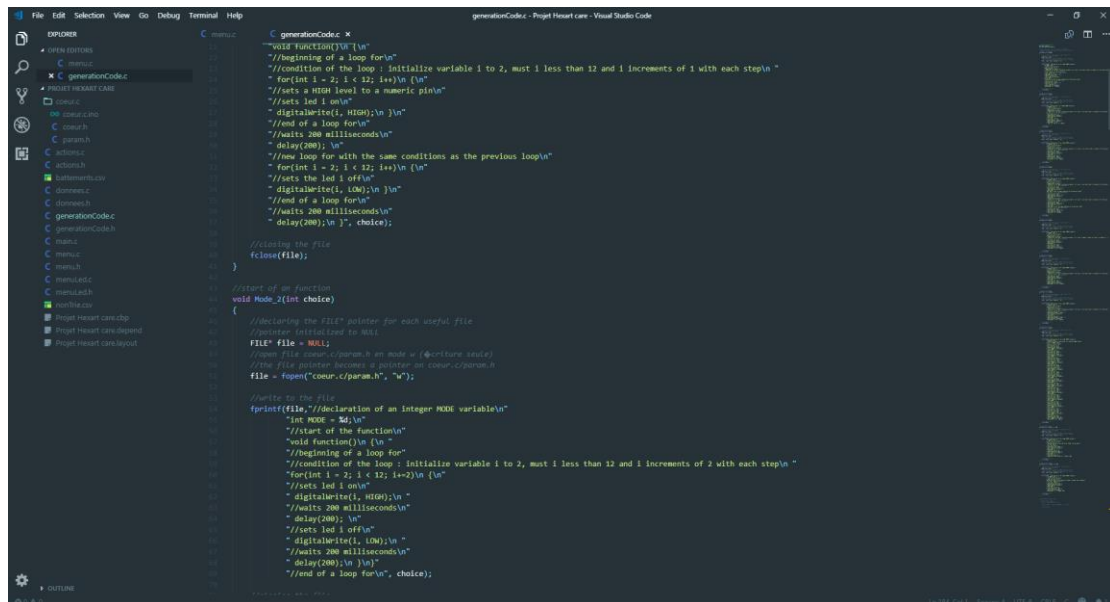
## Projet 1 : Cardiofréquencemètre

## Module 3 :



## code Processing

## Module 4 :



*exemple code generationCode.c 1*