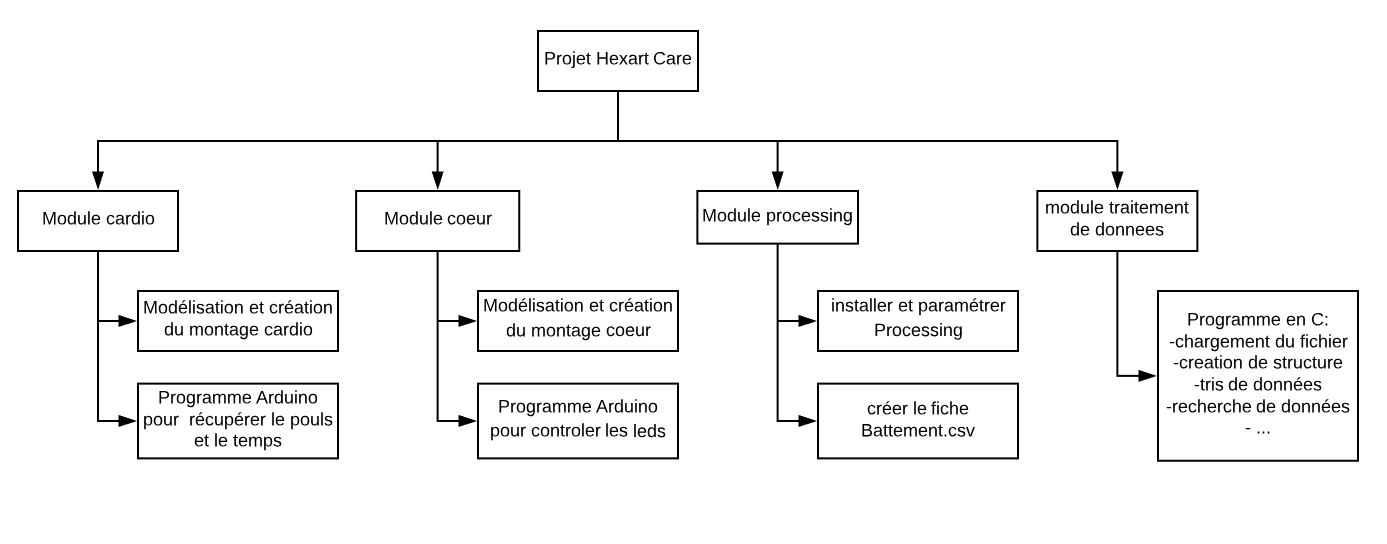
**Première partie : AVOIR UNE VUE D’ENSEMBLE DU PROJET**

### Objectif

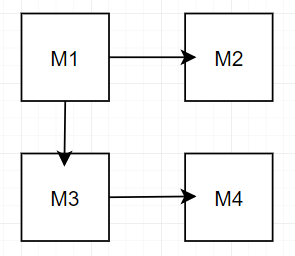
Réaliser un montage permettant de récupérer le pouls d’une personne à l’aide d’un programmer permettant d’afficher et de trier les résultats.

***Dessinez l’architecture du projet – comment avez-vous organisé le projet ?***

### organisation du projet

Ce projet se découpe en 4 modules, ci-dessous se trouve chaque module avec sa composition.

Ces modules ne sont pas indépendants et vont se connecter entre eux :



Voici l’organisation des modules entre eux

M1 : module cardio

M2 : module cœur de leds

M3 : module processing et acquisition de donnees

M4 : module lecture et traitement de donnees

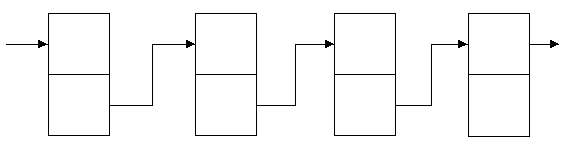
**Deuxième partie : ANALYSEZ LES STRUCTURES DE DONNEES DU PROJET**

1. ***Représentation graphique de toutes les structures nécessaires, organisation des fichiers de code Arduino et du code C et dépendances entre les fichiers.***

### structure

Pour assembler le pouls et le temps correspondant, on créer une structure de type *donnees* composée de deux entiers et d’un pointeur sur la donnée suivante.

Le fait d’avoir créé une structure permet d’utiliser une liste chainée, et non un tableau. Ce principe permet d’avoir un nombre plus flexible de valeurs contrairement à un tableau qui lui est déjà défini.



nextDonnees

nextDonnees

nextDonnees

nextDonnees

Temps

Pouls

Temps

Pouls

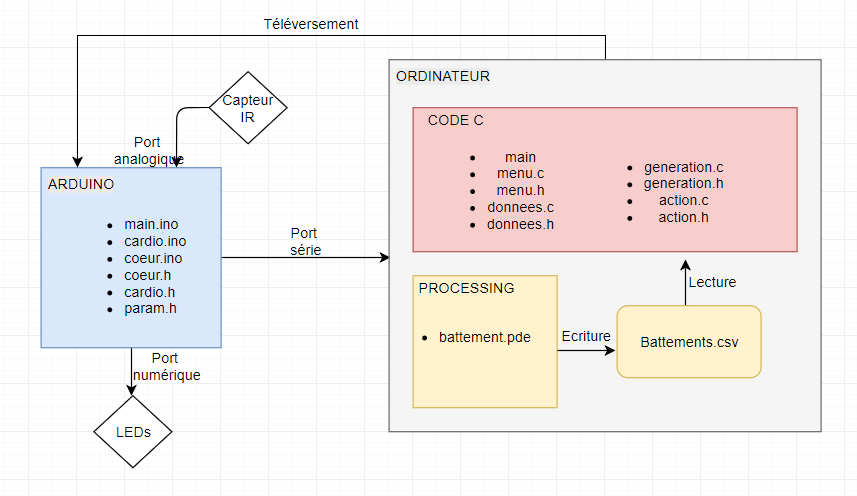
Temps

Pouls

Temps

Pouls

### dependances entre les fichiers

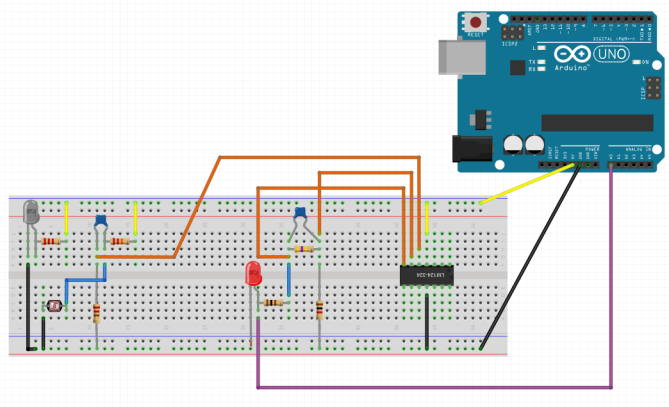


**Troisième partie : MODULARISATION & WORKFLOW DE FONCTIONS & SCHEMAS ELECTONIQUES**

1. ***Schémas électroniques avec les composants sur Fritzing (vue platine et vue schématique des modules cardio et cœur de LEDs ). Comme cette partie comporte une évaluation séparée du projet, vous pouvez faire un document à part entière.***

### module cardio

***Schéma du montage cardio***



***Vue platine du montage cardio***

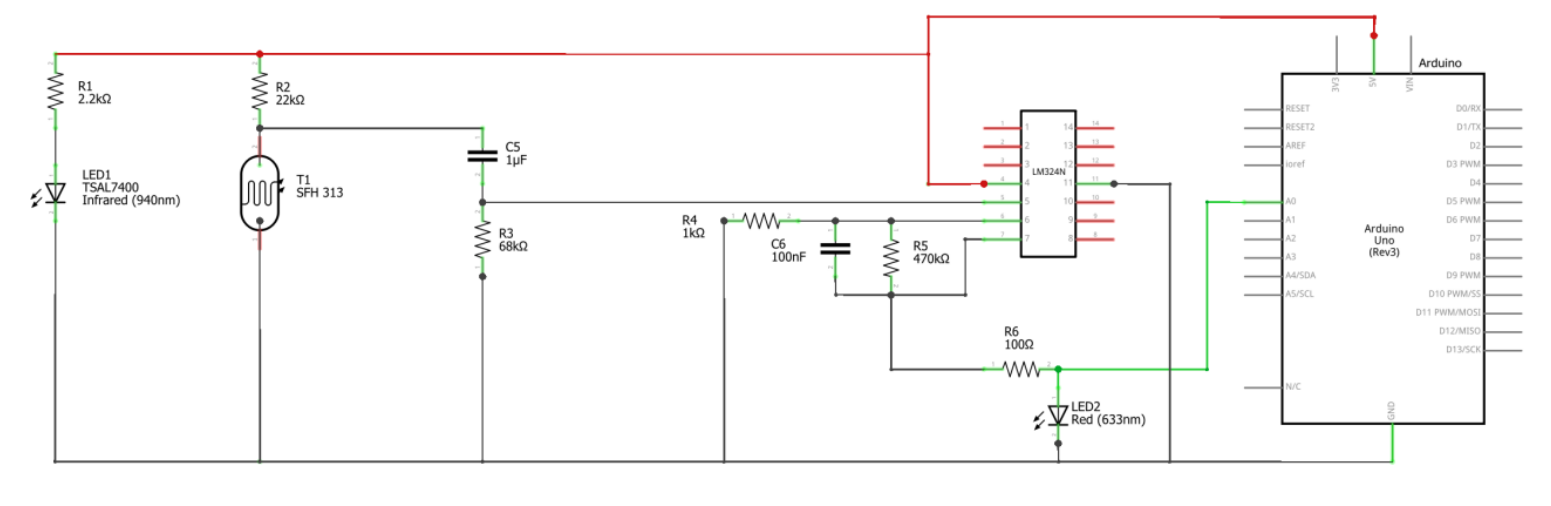
Liste des composants nécessaire au montage :

|  |  |
| --- | --- |
| Arduino UNO  LED infrarouge 940nm  LED rouge 622nm | Condensateur : 1uF et 100nF  Phototransistor SFH 313  Amplificateur opérationnel LM324  Résistance : 100, 1k, 2,2k, 22k, 68k, 470K |

Le schéma ci-dessus du cardiofréquencemètre permet de mesurer la fréquence cardiaque d’un individu.

La LED infrarouge TSAL7400 sert d’émetteur et le phototransistor de récepteur. Le phototransistor a la capacité de transformer un rayonnement optique (qui est en Infrarouge, dans notre cas) en signal électrique. Ensuite l’amplificateur opérationnel LM324 amplifie la différence de potentiel du circuit, il augmente donc la tension électrique du circuit (en Volt). Enfin la LED rouge du circuit sert de témoins, elle nous permet de visualiser le fait que le récepteur capte des variations dans le rayonnement IR.

Quand le récepteur perçoit une variations son signal de sortie va s’altérer ce qui va aller modifier la tension électrique amplifiée par l’amplificateur opérationnel. Cette tension passera par des condensateurs et des résistances ce qui allumera la LED.

Analyse des filtres présents dans le schéma :

Filtre passe bas

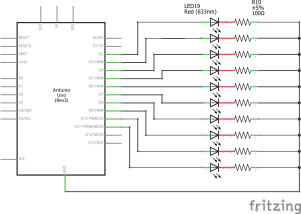
Filtre passe haut

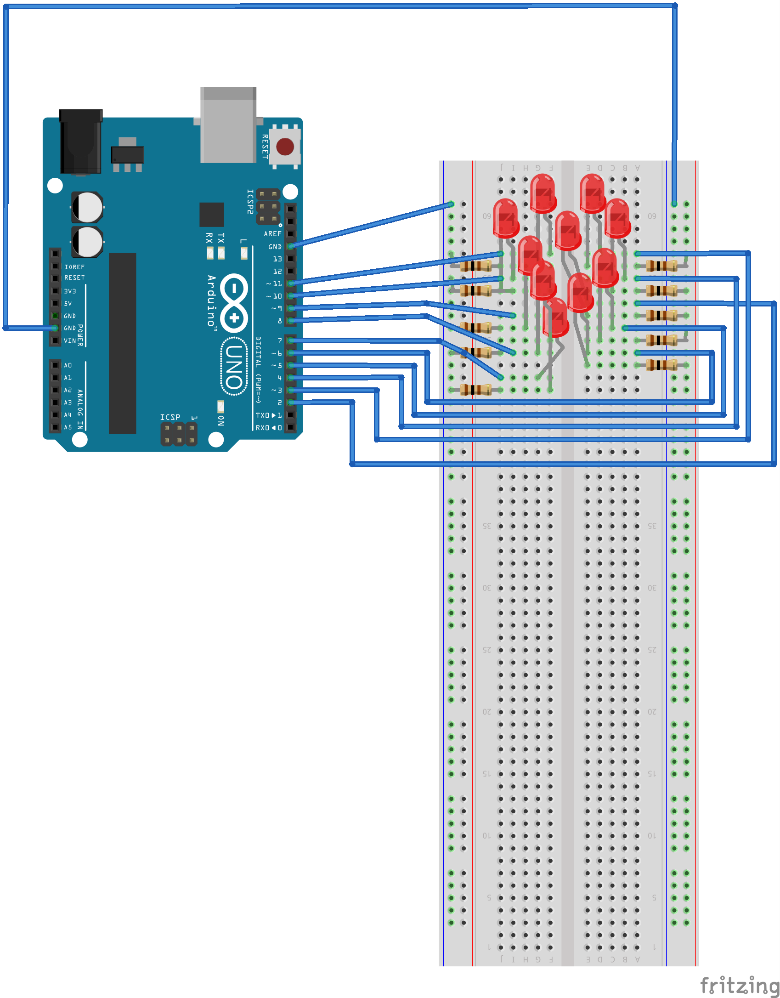
Les filtres permettent d’atténuer la branche qui se situe entre la résistance et le condensateur que si ils se trouvent en série. Selon la position de la résistance et du condensateur par rapport au sens de courant, on obtient un filtre passe-bas ou filtre passe-haut, ils laisseront donc passer une différente plage de fréquence.

On observe aussi un filtre RC en parallèle à la sortie de l’amplificateur, mais il n’influe pas sur la tension.

### module cœur de leds

Le montage fourni dans le projet permet seulement d’allumer toutes les LEDs mais pas de les contrôler. Pour cela nous avons réalisé et représenté un circuit qui lui contrôle les LEDs individuellement.



******

Voici le montage du cœur de LEDs ci-contre avec 10 LEDs et 10 résistances.

Nous avons placé 10 LEDs en formant un cœur, chaque cathode est reliée à un port digital par un fil électrique. Nous utilisons les ports de 2 à 11 . Ensuite les anodes sont reliées à une résistance de 100 Ohm.

L’utilisation des ports digitaux permet de contrôler l’ouverture et l’état des ports et donc par la suite allumer ou éteindre chaque LED. Le contrôle de ses ports permet de créer les fonctions pour les différents modes de clignotement.

1. ***Description algorithmique chaque partie du projet (module cardio, module cœur de LEDs (inclus la génération automatique du paramétrage à partir d’un programme en C), module Preprocessing/acquisition des données, module lecture et traitement de données en C)***

### module cardio

Le but de cette fonction est de lire via le port analogique les valeurs que retourne le récepteur. Avant que le montage cardio soit fonctionel, nous utilisons une première version de cardio.c qui permettait au pouls de prendre une valeur aléatoire mais cohérente.

Cardio.c (version1)

Long freq = (pouls/1000)\*60

Long temps = millis()

Float pouls = random (1000 ; 1333)

Retourne la fréquence et le temps

Cardio.c (version 2)

Float pouls

Long temps = millis()

Pouls = analogRead(0)

Récupère les valeurs du pouls grâce au port analogique et les affiches sur la console

Une fois le montage fini, nous pouvons récupérer des valeurs de pouls grâce à un branchement sur le port analogique A0. Pour cela on

### module cœur de leds

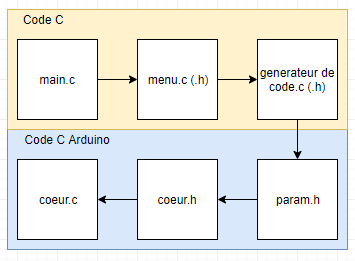
### 

Le module cœur est composé de trois fichiers permettant de gérer les différents modes de clignotements.

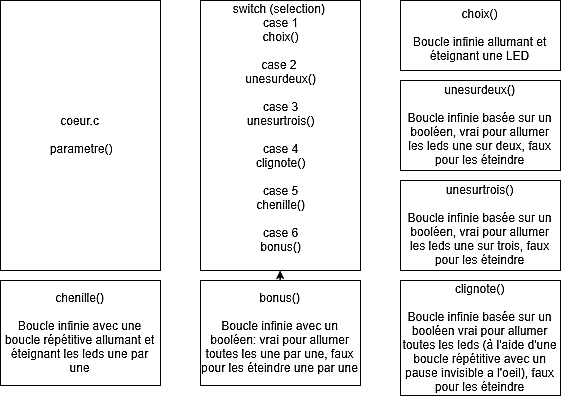
Arduino : cœur.c, cœur.h , param.h

C : menu.c , generationCode.c, generationCode.h

Le cœur.c comporte seulement les fonctions qui sont déclarées dans cœur.h et non un void setup/loop car ensuite tous les fichiers Arduino seront assemblés et donc auront un main.c commun.



Cependant ce module doit lier le menu pour le choix de clignotement des leds au code cœur.c. Comme on ne peut pas passer directement de C à Arduino, on utilise param.h qui est utilisé tel un fichier texte. Pour écrire dans ce fichier on utilise le fichier générationCode.c , il va donc charger le fichier param.h et y écrire le numéro correspondant au choix de l’utilisateur.



Une fois la valeur récupérée, elle se stocke dans sélection, pour ensuite être utiliser dans la fonction parametre() qui va après lancer le mode de clignotement sélectionné.

### module Processing

Processing est un logiciel permettant de récupérer toutes les valeurs des pouls lues sur la sortie série de Arduino pour ensuite les enregistrer dans un fichier .csv

Récupération des données par Processing

Console Arduino :

Affichage du pouls et du temps correspondant grâce à cardio.c

Insertion des données dans un fichier .csv

void setup

void draw

Ecrit les valeurs dans le document

Affiche les valeurs dans la console

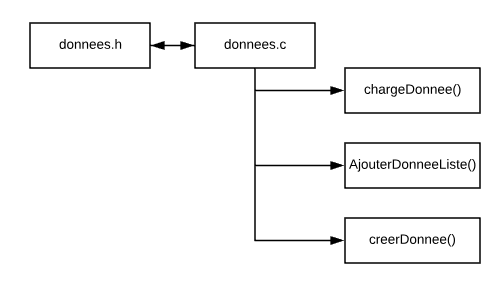
Si valeur entrante

Créer le document « battements.csv »

Ouvre le port que l’on utilise

### module lecture et traitement de donnees

Le but de donnees.c est de lire le fichier .csv pour en récupérer les informations pour ensuite les mettre en structure et créer la liste chainée, pour après les traités.

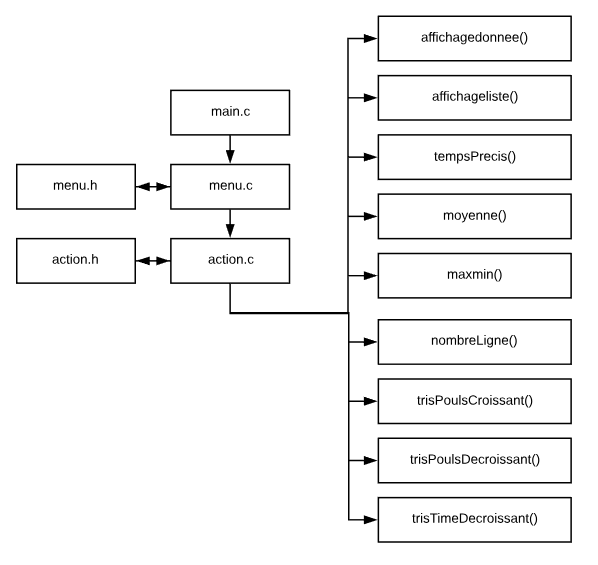


chargeDonnee() : permet de récupérer les informations du fichier .csv

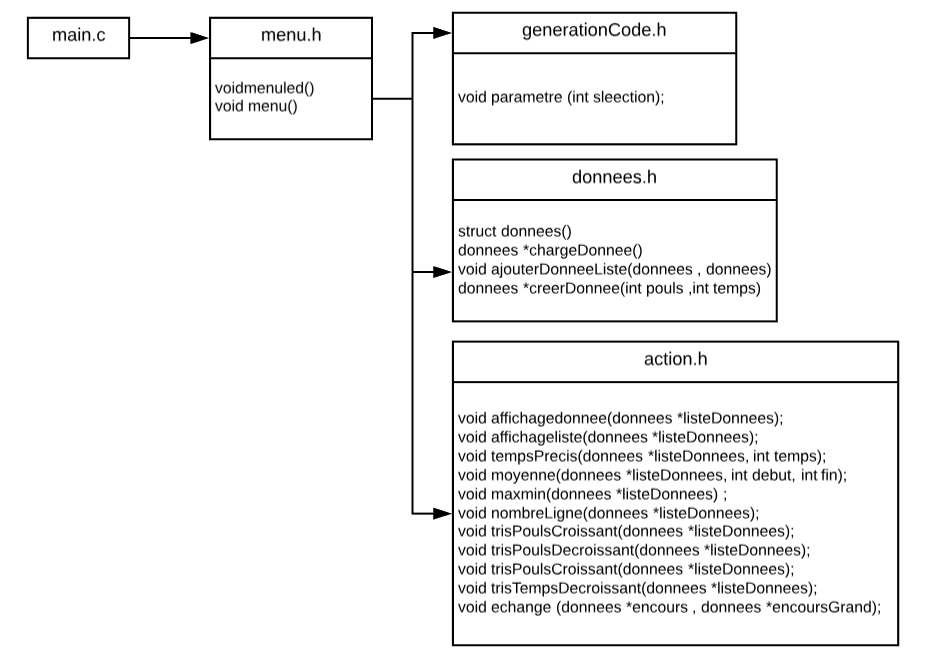
creerDonnee() : appelée dans chargeDonnee, elle permet de créer une structure à l’aide des informations récupérées

AjouterDonneeListe() : après avoir créer notre nouvelle structure, cette fonction permet de l’insérer dans la liste chainée

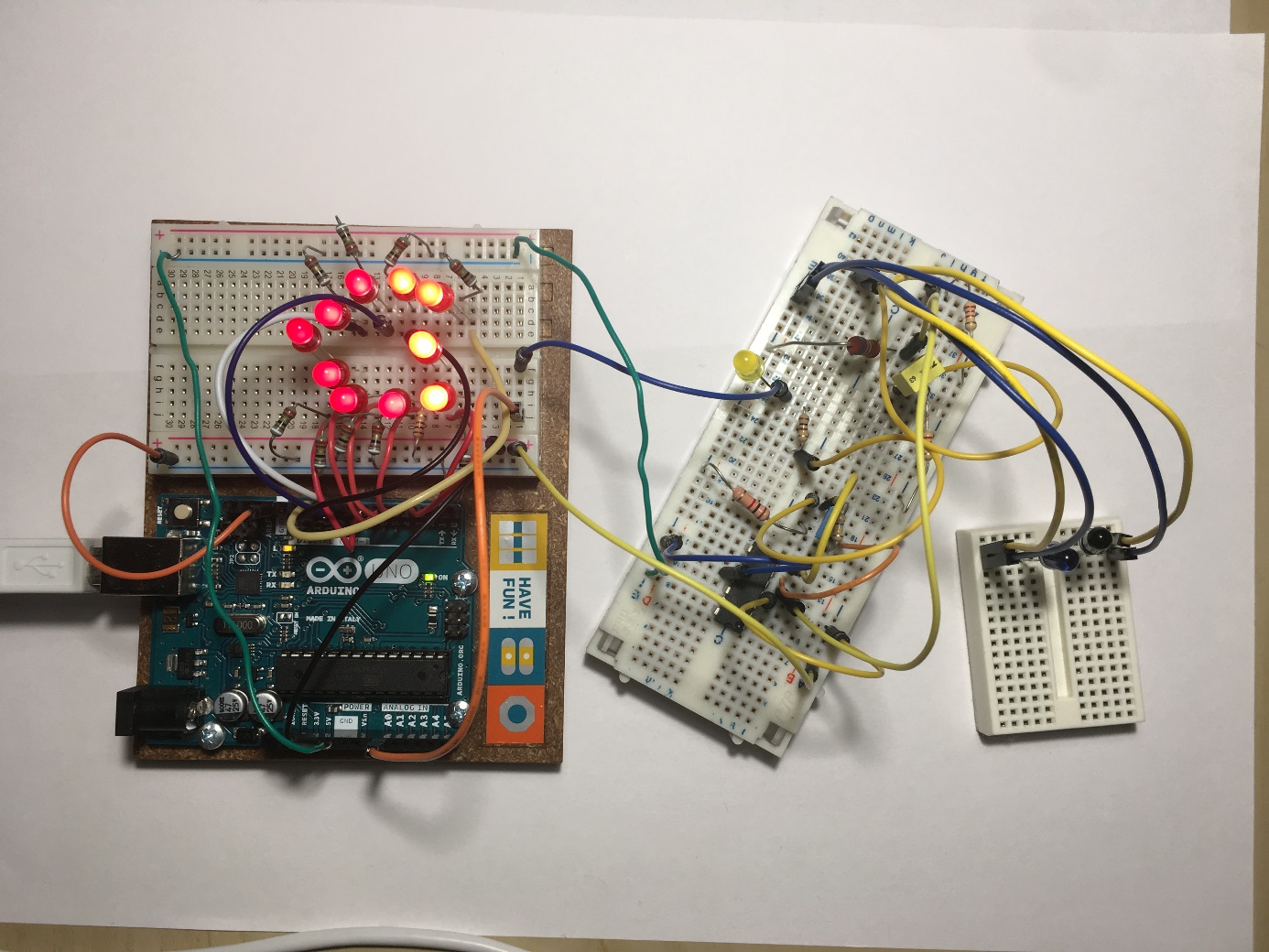
Le but de action.c est de traiter les données des structures, on peut faire une moyenne de pouls sur une plage de temps donné, trouver le pouls maximum et minimum, trier le pouls de manière croissante/décroissante …



1. ***Prototypes de l'ensemble des fonctions du projet (faites-le le plus « graphiquement » possible)***



### Montage final fonctionnel



Une fois tous les modules assemblés, on peut observer à gauche le cœur en led, au centre le circuit du module cardio et à droite le récepteur et émetteur.

**Quatrième partie : REPARTISSEZ-VOUS LES TACHES**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nom : KESSAS Laetitia** | | | **Rôle principal : Chef de projet** | | | | | | | | |
| **Tâches** | **11/11** | **12/11** | | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Montage cœur led |  | X | |  |  |  |  |  |  |  |
| Programme coeur |  |  | | X |  | X |  |  |  |  |
| Programme cardio |  |  | |  | X | X |  |  |  |  |
| Programme module 4 |  |  | |  | X | X |  |  |  |  |
| Schéma Fritzing |  | X | |  | X |  |  |  |  |  |
| Fiche d’avancement |  |  | | X | X | X | X |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nom : BOURDEAU Kévin** | | | | | **Rôle principal : Programmation** | | | | | | | | | | | |
| **Tâches** | **11/11** | **12/11** | **13/11** | | | **14/11** | | **15/11** | | **16/11** | | **17/11** | | **18/11** | | **19/11** |
| Processing |  |  | |  | | | X | | X | |  | |  | |  |  |
| Programme cardio |  |  | |  | | |  | | X | |  | |  | |  |  |
| Programme module 4 |  |  | |  | | |  | | X | |  | |  | |  |  |
| Fiche d’avancement |  |  | | X | | |  | |  | |  | |  | |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nom : CROS Thibault** | | | | **Rôle principal : Conception des circuits** | | | | | | |
| **Tâches** | **11/11** | **12/11** | **13/11** | | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Montage cardio |  | X | X | |  |  |  |  |  |  |
| Analyse montage, filtres |  |  | X | | X |  |  |  |  |  |
| Schéma Fritzing |  |  |  | | X |  |  |  |  |  |
| Programme cardio |  |  |  | | X |  |  |  |  |  |
| Fiche avancement |  |  |  | |  | X |  |  |  |  |

|  |  |
| --- | --- |
| **Répartition prévisionnelle des tâches** | **Bilan des tâches faites** |
| Kevin BOURDEAU |  |
| * Programmation du module cardio * Installation et paramétrage de Processing * Algorithme de tris | Installation et paramétrage de Processing  Création des structures grâce au .csv |
| Thibault CROS |  |
| Faire le circuit du module cardio  Modéliser le montage du module cardio  Analyse des composants et filtres du montage | Faire le circuit du module cardio  Modéliser le montage du module cardio  Analyse des composants et filtres du montage |
| Laetitia KESSAS |  |
| Réaliser le montage cœur  Modéliser le montage cœur  Programmation du module cœur  Algorithme de recherches | Réaliser le montage cœur  Modéliser le montage cœur  Programmation du module cardio  Programmation du module cœur  Algorithme de recherches  Algorithme de tris  Regroupement de tous les codes |

### Retour d’expérience (REX)

Pour ma part, Kévin Bourdeau, je me suis occupé avec Laetitia du développement du cardiofréquencemètre, puis du module 3 et enfin une partie du module 4. Grâce à un bon chef de projet, la répartition des tâches fut bonne.

Pour ma part, Thibault CROS, j’ai énormément appris de ce projet, notamment sur le travail en groupe et la coopération. J’ai travaillé principalement sur l’électronique du module cardio.

Pour ma part, Laetitia KESSAS, je me suis occupée principalement de tout le développement ( C et C Arduino) et d’un montage électronique. Le fait d’avoir touché à tous les modules m’a permis de bien comprendre les interconnections de ses derniers. La seule difficulté de ce projet a été la communication entre nous.

### Bilan

Nous avons réussi à mener le projet jusqu’à la fin. Le cardiofréquencemètre récupère bien le pouls, mais par manque de temps tout le traitement des données se fait à l’aide de valeurs aléatoires mais cohérente.