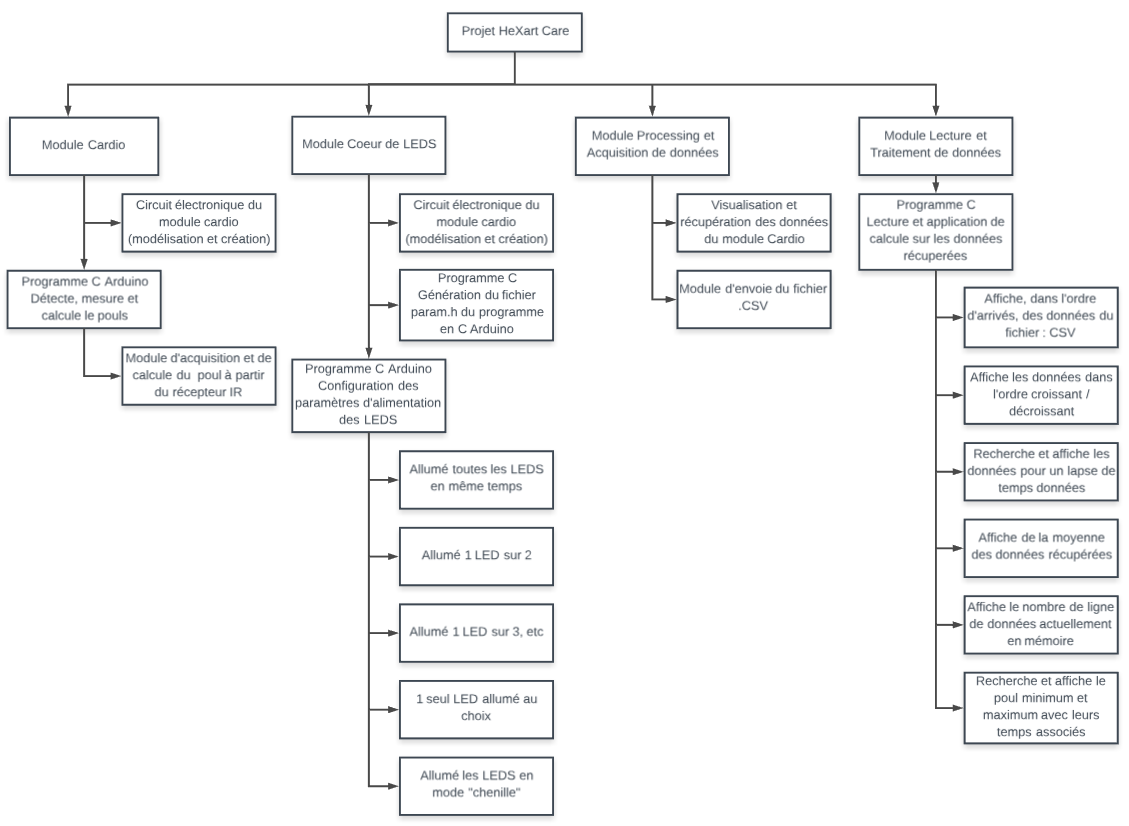
**Première partie : AVOIR UNE VUE D’ENSEMBLE DU PROJET**

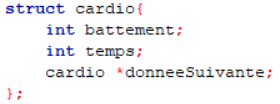
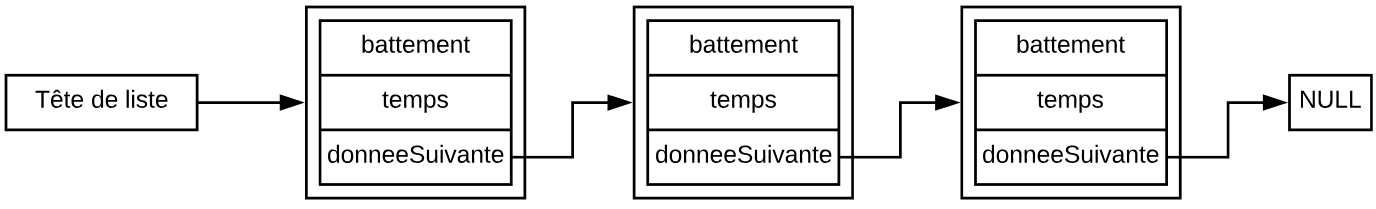
1. ***Dessinez l’architecture du projet – comment avez-vous compris le projet ?***

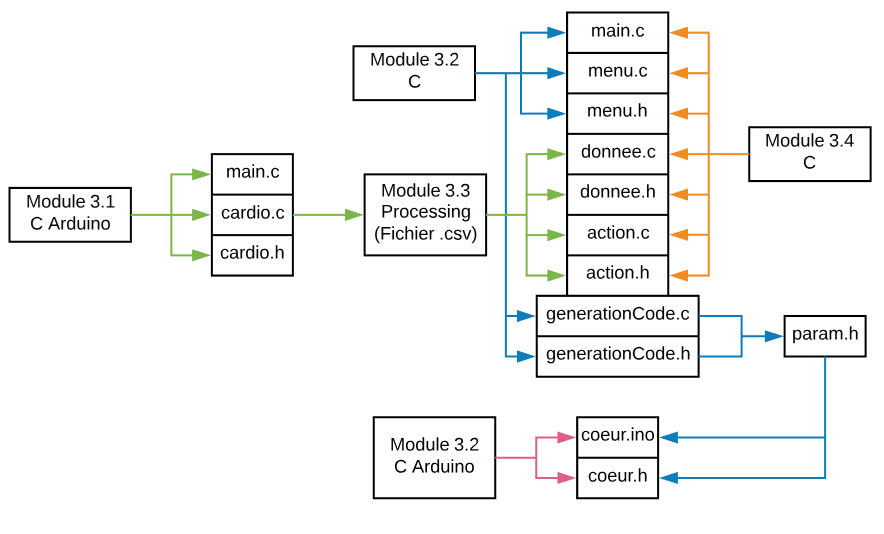
Le but est de créer une machine portative permettant de compter les pulsations cardiaques par minute. Chaque pulsation cardiaque allumera un cœur en LEDS en fonction des envies de l’utilisateur (clignotement de toute les LEDS, 1 LED sur 2 allumées, …). Chaque pulsation va alors être stockées dans un fichier avec son temps associés (temps écoulé en millisecondes depuis le lancement du programme). Nous aurons pour finir un dernier programme en C permettant d’afficher, de trier et de rechercher n’importe quelle donnée dans le fichier de donnée créer par les autres modules.



**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.***

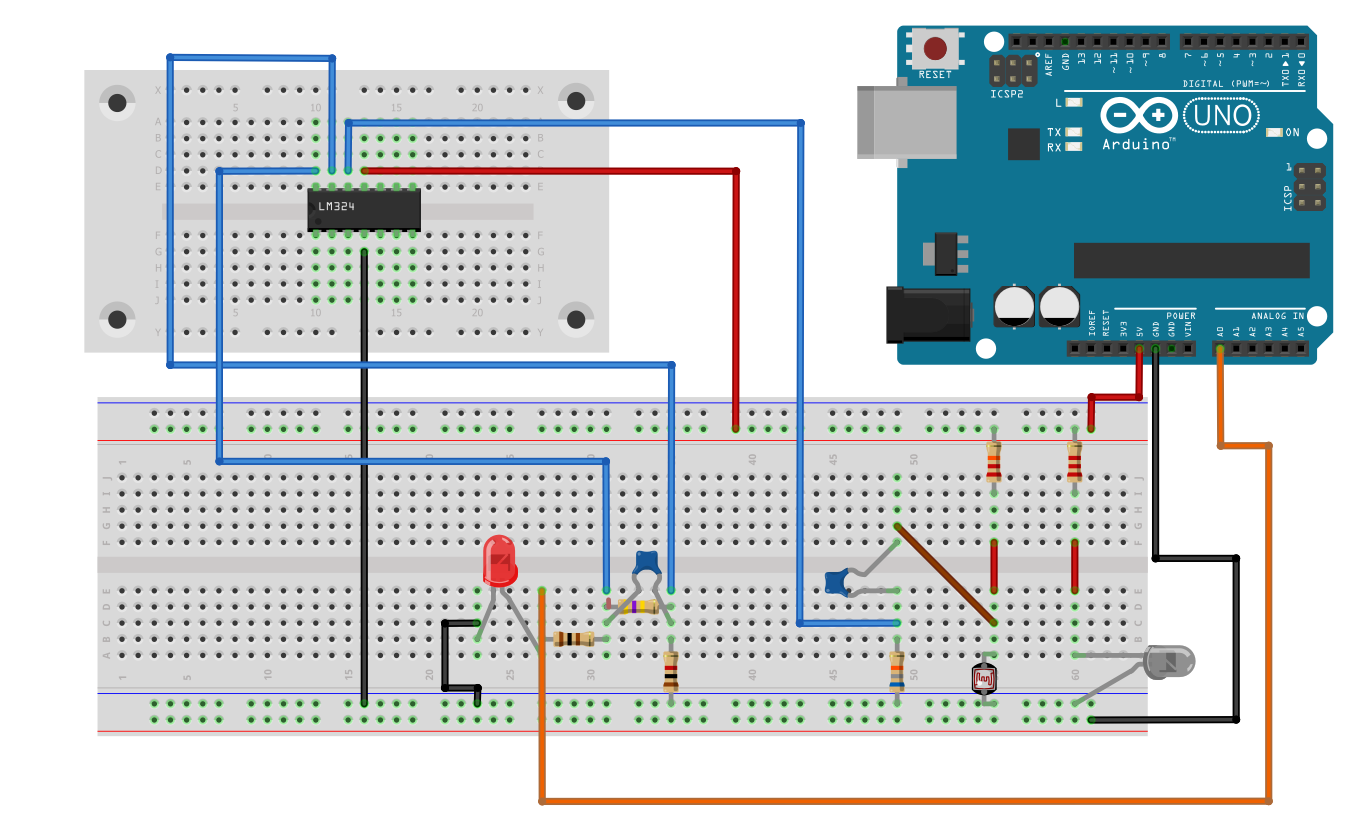
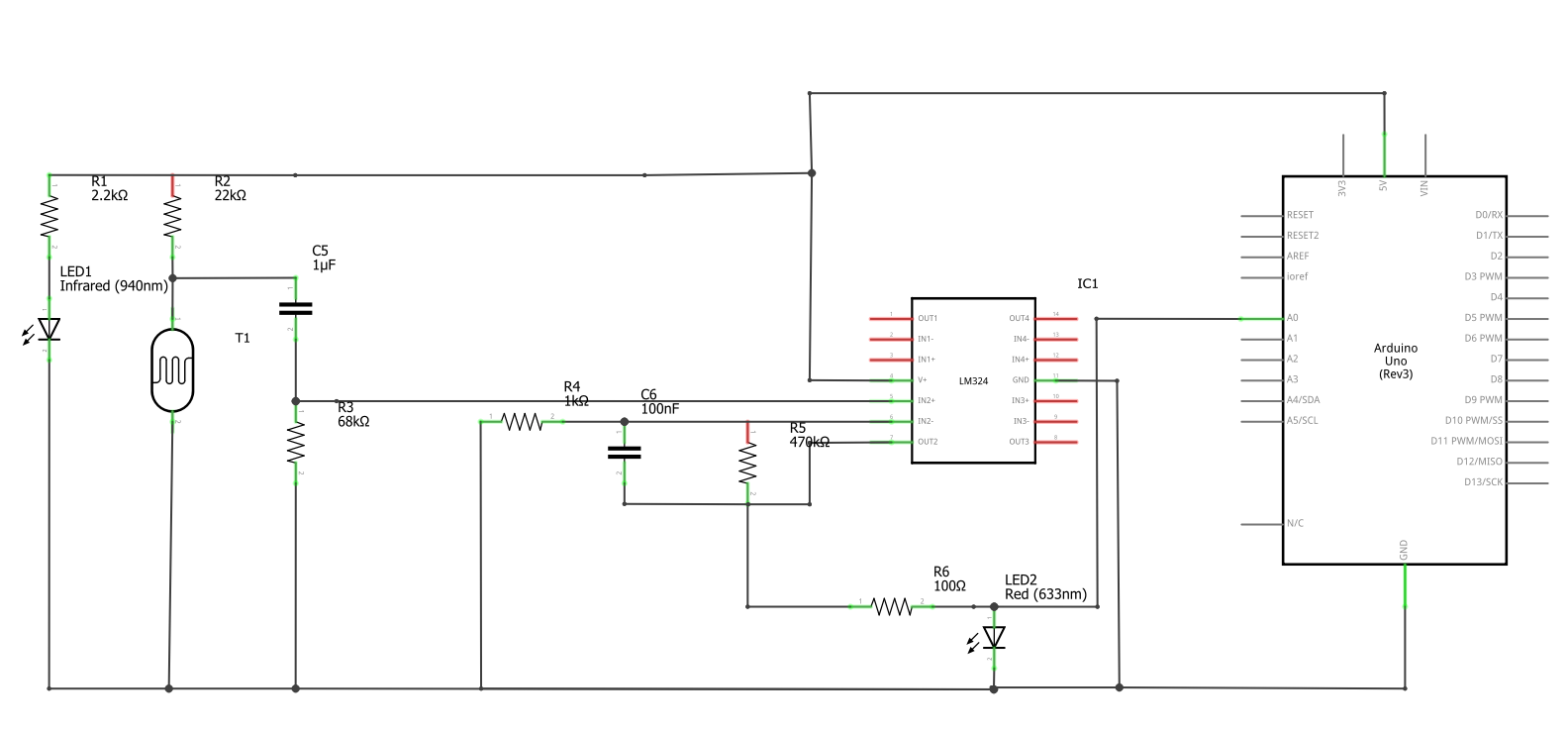
**Structure Cardio :**   
La structure « cardio » possède 3 types de données :   
 - Un entier « battement » indiquant le pouls d’une personne récupéré depuis le module cardio.   
 - Un entier « temps » indiquant le temps en millisecondes depuis le lancement du programme Arduino.   
 - Un pointeur « donneeSuivante » permettant de parcourir une liste chaînée du type « struct cardio ».   
  
**Liste Chaînée :**  
  
Nous avons alors utilisé une liste chaînée du type « struct cardio » pour y stocker une infinité de structure ayant le même fonctionnement de stockage. On peut alors voire que la liste chaînée est composée d’une tête de liste, d’un nombre indéfini d’élément et d’une fin de liste.  
Cette liste chaînée est fabriqué à partir des données du fichier .csv.

**Organisation des fichiers (Arduino et C) et dépendances entre les fichiers :**  
****

Nous possédons 4 modules distinct de base, cependant nous pouvons entrevoir une certaine interconnectivité entre les fichiers lors de la mise en place du projet. Tout d’abord, étant donné que les modules 3.2 et 3.4 possèdent des fichiers de même nom (main.c, menu.c et menu.h), nous avons décidé de les reliés et d’en faire un grand et seul fichier permettant de gérer les options des deux modules simultanément. Cet énorme module permet tout d’abord de modifier le mode d’affichage des LEDS grâce à la création d’un fichier « param.h » fessant le lien entre les modules en C et les modules en C Arduino. Pour finir l’énorme module permet aussi le tri, la recherche et la lecture de données du fichier .csv obtenu depuis le module cardio.

**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.***

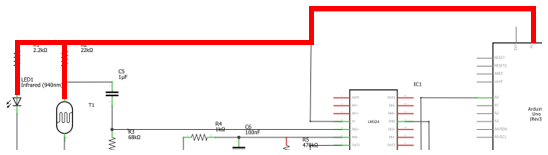
**Module Cardio :  
Schéma :**  
  


**Fiche technique :  
1 / Utilisation du circuit :**On cherche à reproduire un électrocardiogramme, qui permet de capter les pulsations  
cardiaques de l’utilisateur.

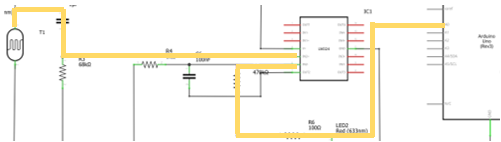
Une photorésistance détectera les variations de l’infrarouge produite par une LED à chaque pulsation. L’utilisateur pourra voir son pouls grâce à une deuxième LED, et grâce à un programme sur le moniteur série, sa fréquence cardiaque  
  
**2 / Matériel :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Résistances : | LEDS : | Capteur : | Condensateur : | Microcontrôleur |
| R1 : 2.2kΩ | LED 1 : Infrarouge (940nm) | Photorésistance | C5 :  1µF | LM324 |
| R2 : 22kΩ | LED 2 : Rouge (633nm) |  | C6 :  100nF | Arduino UNO |
| R3 : 68kΩ |  |  |  |  |
| R4 : 1kΩ |  |  |  |  |
| R5 : 470kΩ |  |  |  |  |
| R6 : 100Ω |  |  |  |  |

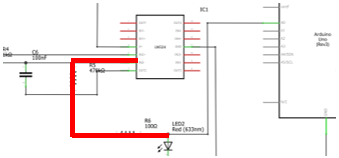
**3 / Fonctionnement :**La carte Arduino alimente la LED infrarouge et la photorésistance, qui renvoie un courant à chaque fluctuation de la lumière infrarouge. La tension passe par deux filtres, pour supprimer les interférences, et ainsi avoir un signal le plus propre possible. La tension va ensuite dans le LM324, pour ensuite être envoyé vers la LED rouge et la carte ensuite. Grâce à cela la LED s’allume et s’éteint en fonction du pouls de l’utilisateur, et les informations peuvent être retranscrites sur le moniteur série de la carte Arduino grâce à un programme simple.



*Alimentation en +5V de la LED infrarouge, et de la photorésistance*

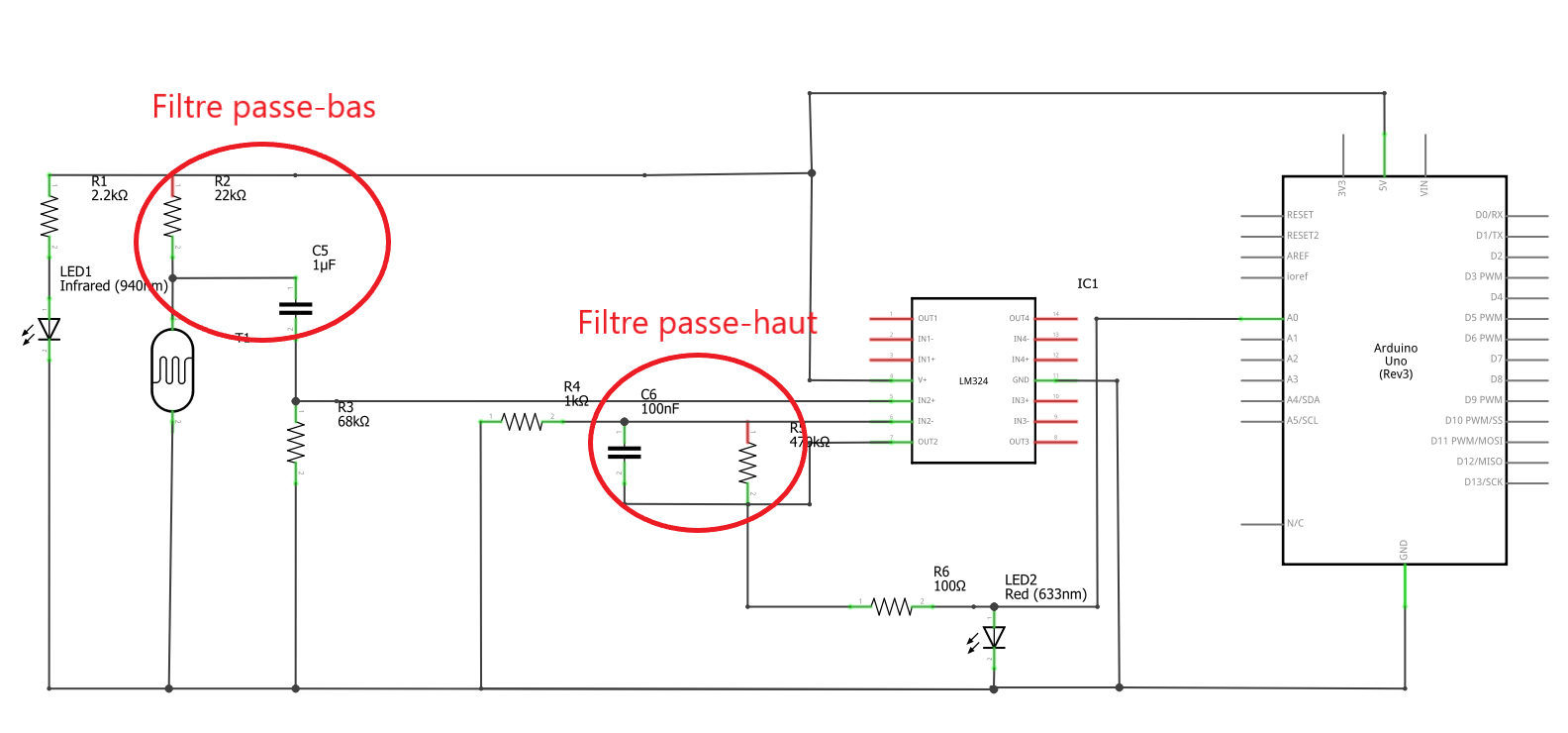


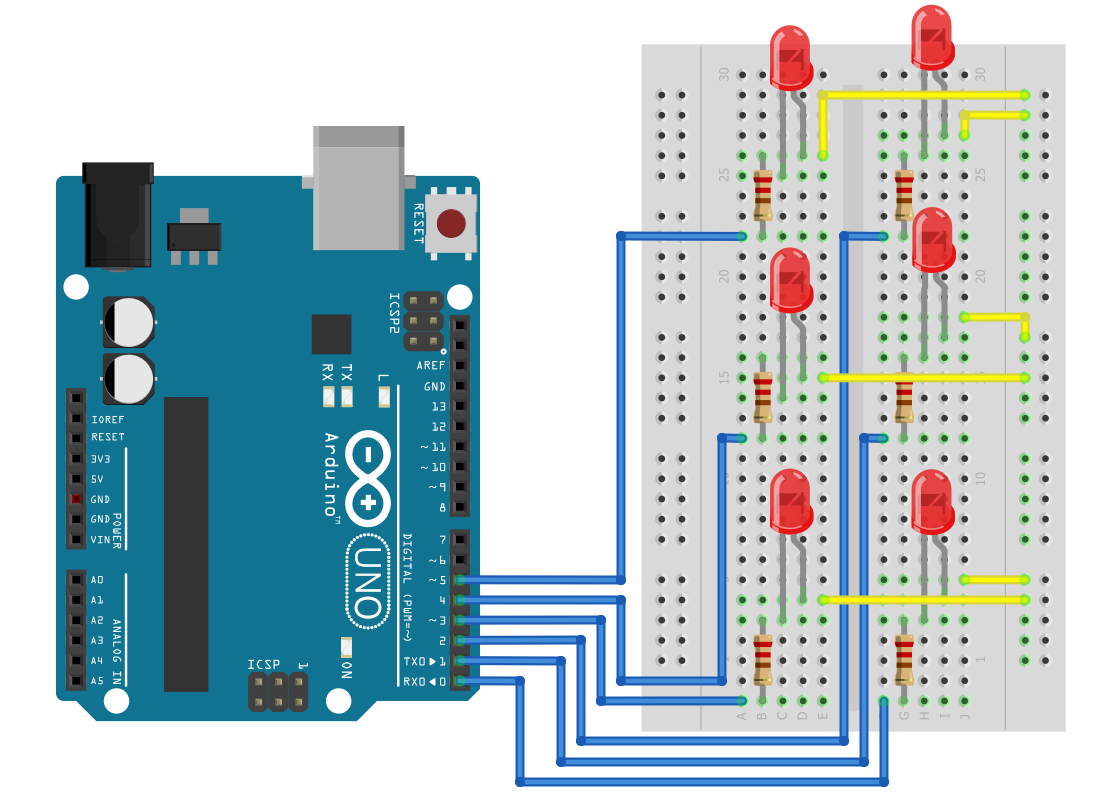
*Envoi des informations du capteur vers la LED rouge te la carte Arduino*

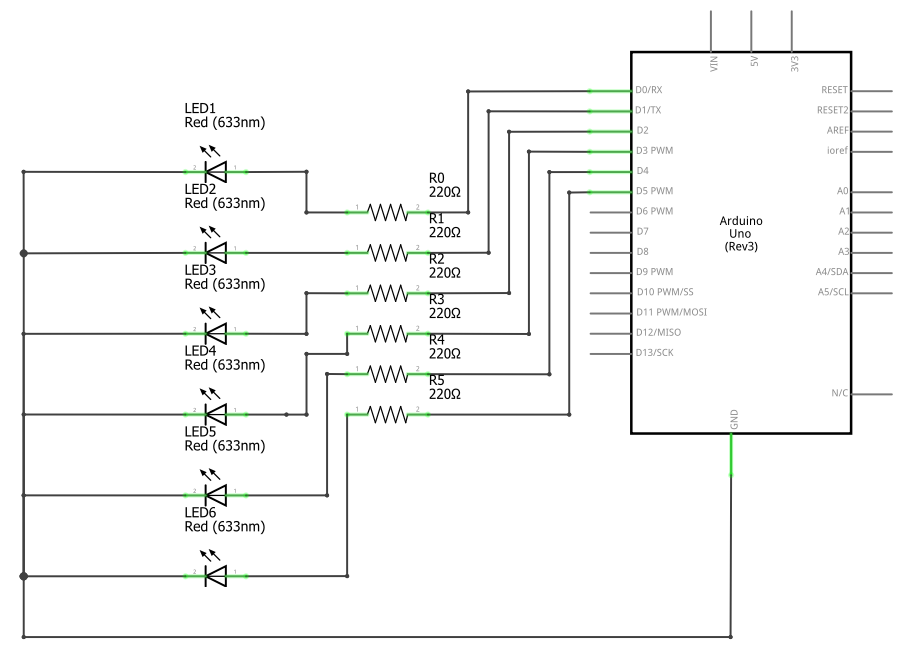


*Allumage de la LED en fonction de la photorésistance*

**4 / Filtres :**

**

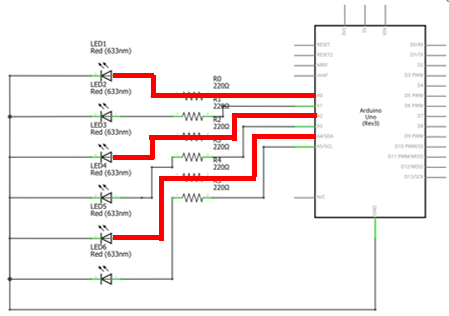
**Module Coeur :**  




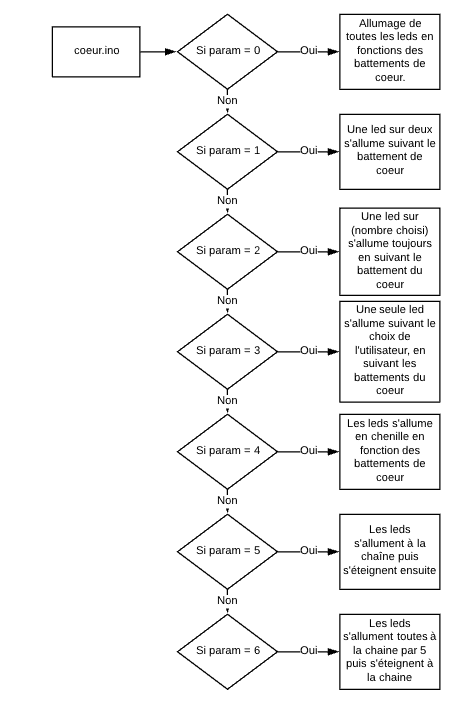
**Fiche technique :  
1 / Utilisation du circuit :**Le cœur sert à afficher les pulsations cardiaques de l’utilisateur d’une façon plus lisible, grâce à plusieurs LED (10 en tout) qui affichent les pulsations de plusieurs façons. En effet plusieurs modes d’allumage peuvent être choisi :

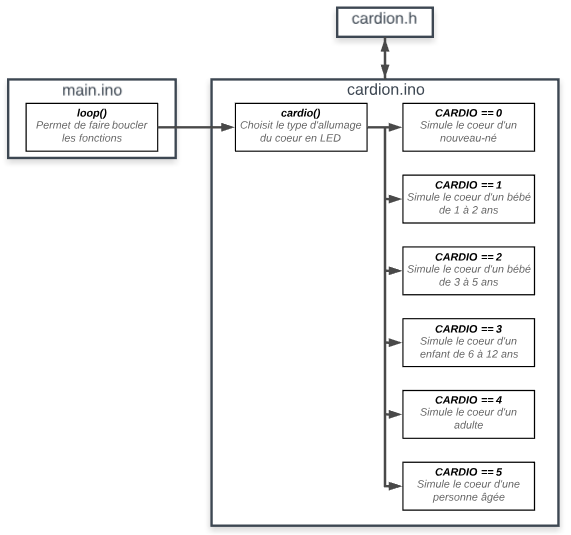
* Toutes les LED allumée en même temps en fonction des battements que le cardiofréquencemètre capte.
* Une LED sur deux allumée, une LED sur trois allumée…
* Une seule LED allumée (au choix)
* Un allumage en mode « chenille » en fonction des battements

**2 / Matériel :**Pour ce montage, on utilise dix résistances de 220Ω, dix leds rouges, et une carte Arduino pour alimenter les LEDS.  
 **3 / Fonctionnement :**On a branché les différentes LEDS sur un port analogique chacune, ainsi on peut contrôler leurs allumages. Ainsi en alimentant un seul port on peut allumer une LED, on bien alimenter un port sur deux et ainsi allumer une LED sur deux.

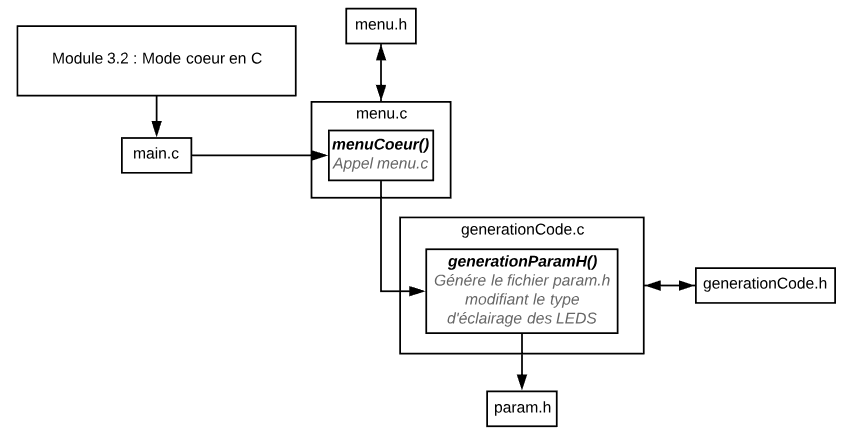
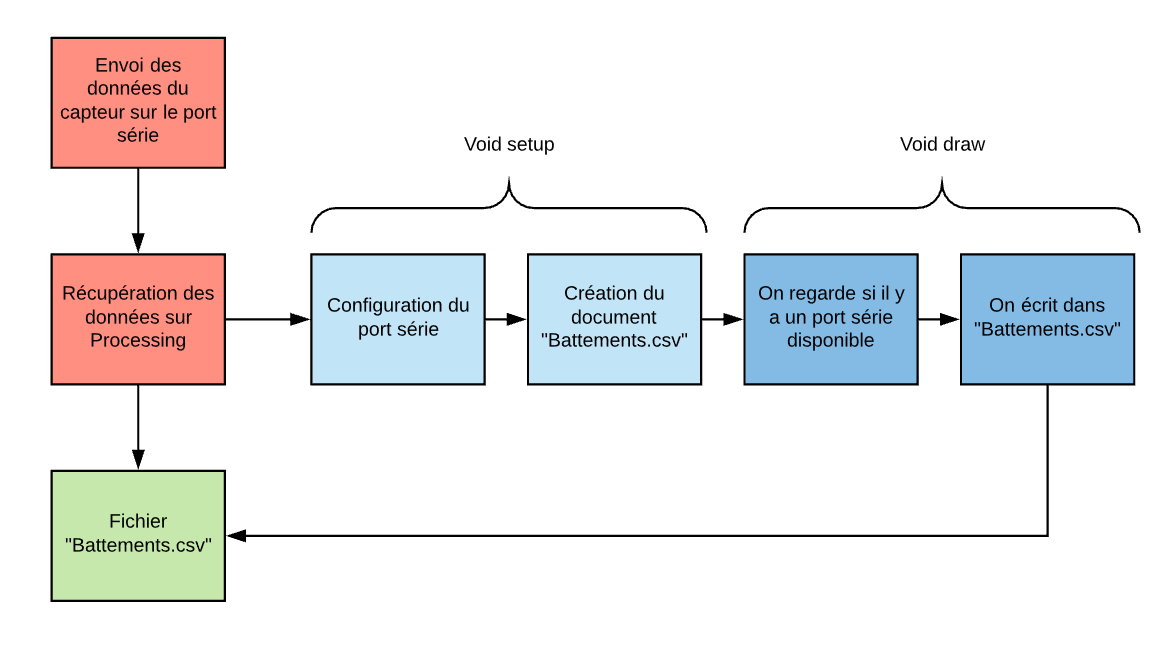
*Allumage d’une LED sur deux*

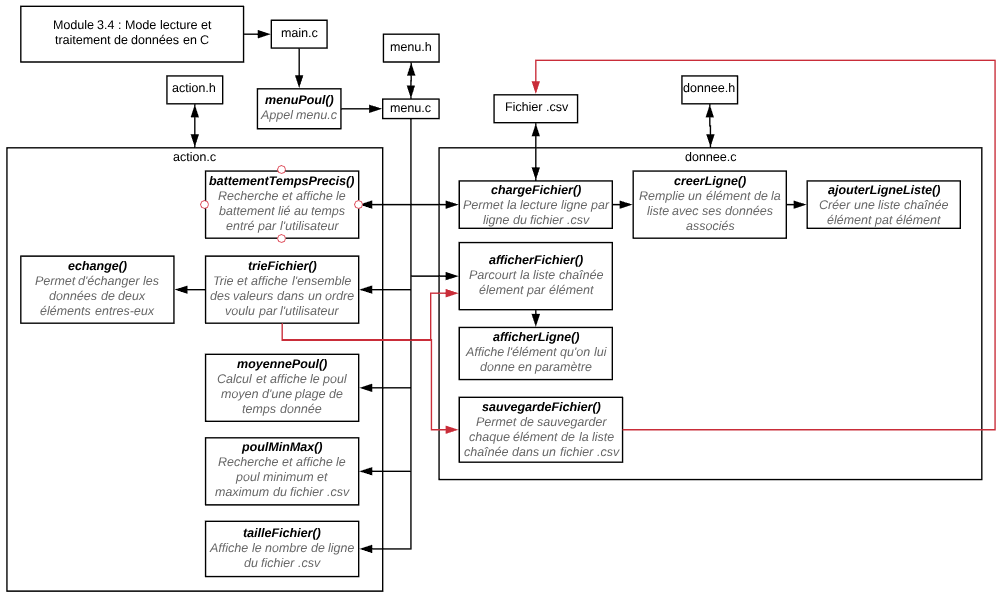
1. ***Description algorithmique chaque partie du projet (module cardio, module cœur de LEDs (inclus la génération automatique du paramétrage à partir d’un programme en C), module Preprocessing/acquisition des données, module lecture et traitement de données en C)***

**Module cardio (en Arduino) :**   


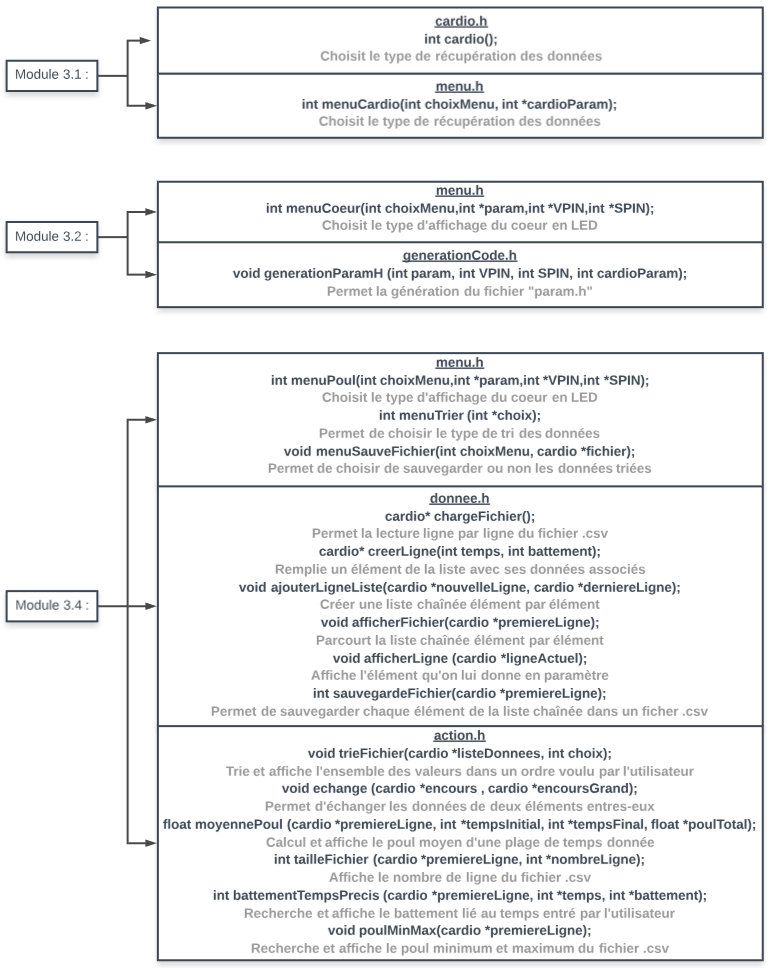
**Module cœur (en Arduino) :**  


**Module cœur (en C) :**

  
  
**Module Preprocessing / acquisition des données :**   


**Module lecture et traitement de données :**   


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



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

|  |  |
| --- | --- |
| **Nom : GASTON Clément** | **Rôle principal : Chef de projet** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **09/11** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Gestion de projet / Feuille Projet |  |  |  | X | X | X | X | X |  |  |  |
| Circuit électronique (Module 3.1) |  |  |  |  |  |  |  |  |  |  |  |
| Programme en C Arduino (Module 3.1) |  |  |  |  |  |  |  |  |  |  |  |
| Circuit électronique (Module 3.2) |  |  |  | X |  |  |  |  |  |  |  |
| Programme en C (Module 3.2) |  |  |  |  | X | X |  |  |  |  |  |
| Programme en C Arduino (Module 3.2) |  |  |  | X | X |  |  |  |  |  |  |
| Visualisation et récupération des données (Module 3.3) |  |  |  |  |  |  |  |  |  |  |  |
| Programme C (Module 3.4) |  |  |  |  | X | X | X |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : LORENDEAUX Anthony** | **Rôle principal :** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **09/11** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Gestion de projet / Feuille Projet |  |  |  | X |  |  |  | X |  |  |  |
| Circuit électronique (Module 3.1) |  |  |  | X | X |  |  |  |  |  |  |
| Programme en C Arduino (Module 3.1) |  |  |  |  | X | X | X |  |  |  |  |
| Circuit électronique (Module 3.2) |  |  |  | X |  |  |  |  |  |  |  |
| Programme en C (Module 3.2) |  |  |  |  |  |  |  |  |  |  |  |
| Programme en C Arduino (Module 3.2) |  |  |  |  |  |  |  |  |  |  |  |
| Visualisation et récupération des données (Module 3.3) |  |  |  |  |  | X | X |  |  |  |  |
| Programme C (Module 3.4) |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : ROSSI Florian** | **Rôle principal :** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **09/11** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Gestion de projet / Feuille Projet |  |  |  | X |  |  |  | X |  |  |  |
| Circuit électronique (Module 3.1) |  |  |  | X |  |  | X |  |  |  |  |
| Programme en C Arduino (Module 3.1) |  |  |  |  |  | X |  |  |  |  |  |
| Circuit électronique (Module 3.2) |  |  |  | X |  |  |  |  |  |  |  |
| Programme en C (Module 3.2) |  |  |  |  | X |  |  |  |  |  |  |
| Programme en C Arduino (Module 3.2) |  |  |  | X | X |  |  |  |  |  |  |
| Visualisation et récupération des données (Module 3.3) |  |  |  |  |  | X | X |  |  |  |  |
| Programme C (Module 3.4) |  |  |  |  | X | X |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : RIPOLL-DAUSA Marine** | **Rôle principal :** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **09/11** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Gestion de projet / Feuille Projet |  |  |  | X |  |  | X | X |  |  |  |
| Circuit électronique (Module 3.1) |  |  |  | X | X | X |  |  |  |  |  |
| Programme en C Arduino (Module 3.1) |  |  |  |  |  |  |  |  |  |  |  |
| Circuit électronique (Module 3.2) |  |  |  |  |  |  |  |  |  |  |  |
| Programme en C (Module 3.2) |  |  |  |  |  |  |  |  |  |  |  |
| Programme en C Arduino (Module 3.2) |  |  |  |  |  |  |  |  |  |  |  |
| Visualisation et récupération des données (Module 3.3) |  |  |  |  |  |  |  |  |  |  |  |
| Programme C (Module 3.4) |  |  |  |  |  |  |  |  |  |  |  |