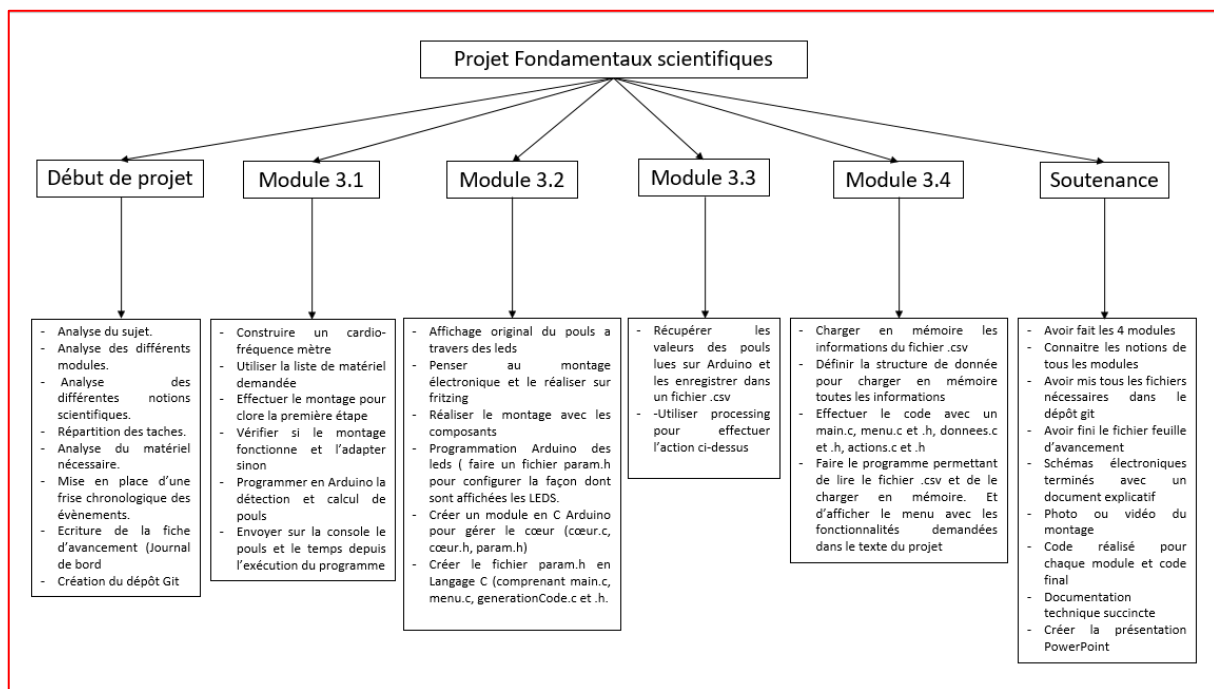


Note : Ceci est simplement « un modèle » à compléter selon vos soins. Des adaptations sont autorisées à condition de les justifier. **Ecoutez les conseils de votre parrain.**

Première partie : AVOIR UNE VUE D'ENSEMBLE DU PROJET

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



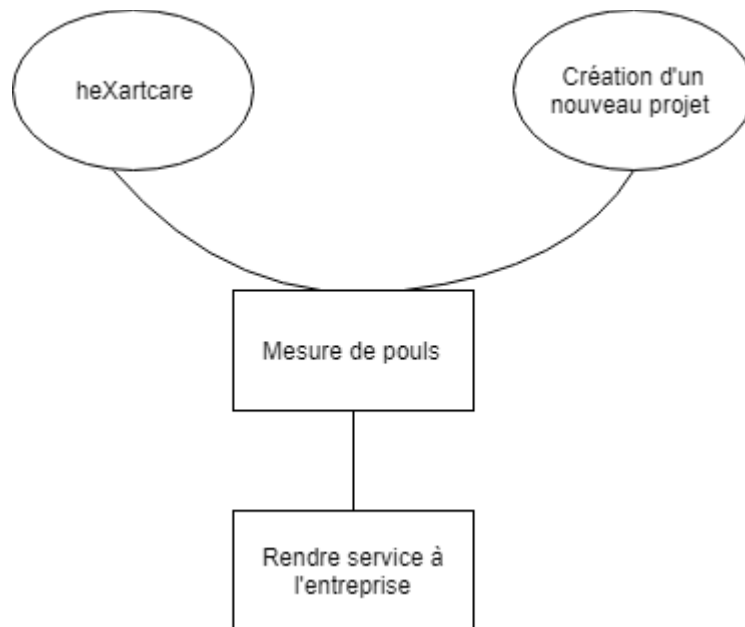
Cette architecture comporte toutes les tâches à effectuer pour chaque étape ou module du projet. Toutes les étapes du module sont à réaliser mais ne sont pas chronologiquement triées.

Pour la compréhension du projet. Ce que l'on veut obtenir est un cardio-fréquence mètre fonctionnel permettant de trouver le pouls d'une personne et de l'afficher via des LED en utilisant une carte Arduino.

On veut également effectuer un traitement des données avec l'utilisation de processing pour générer un fichier .csv qui sera utilisé dans le programme Arduino pour afficher par exemple le pouls maximal mesuré.



On peut aussi effectuer un diagramme bête à cornes pour avoir une vue d'ensemble de ce que l'on veut faire avec notre projet.



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.

On sait déjà que pour ce projet, tous les modules sont indépendants, on peut donc les effectuer sans avoir besoin d'une production d'un autre membre du groupe.

Pour le code Arduino, les fichiers sont organisés comme ceci : Il y a deux grands fichiers distincts, un fichier qui va s'occuper de la mesure de pouls et un autre qui va s'occuper d'allumer les LED en fonction de ce pouls. Dans ces fichiers-là, on créera des bibliothèques ce qui nous permettra de pouvoir optimiser au maximum le code. Il y aura donc les fonctions main.c, cardio.c et cardio.h pour toute la partie cardio et comme on veut juste faire allumer des LED, il n'y a pas besoin d'utiliser de bibliothèques dans le second fichier.

Cependant il va falloir utiliser différentes bibliothèques en Langage C pour pouvoir configurer l'affichage des LED comme on le souhaite. Il y aura donc le main.c, le menu.c et menu.h pour l'affichage du menu. Il y aura aussi le generationCode.c et generationCode.h pour afficher les différentes fonctions nécessaires pour créer le fichier param.h, qui contient nos paramètres d'allumage des LED.



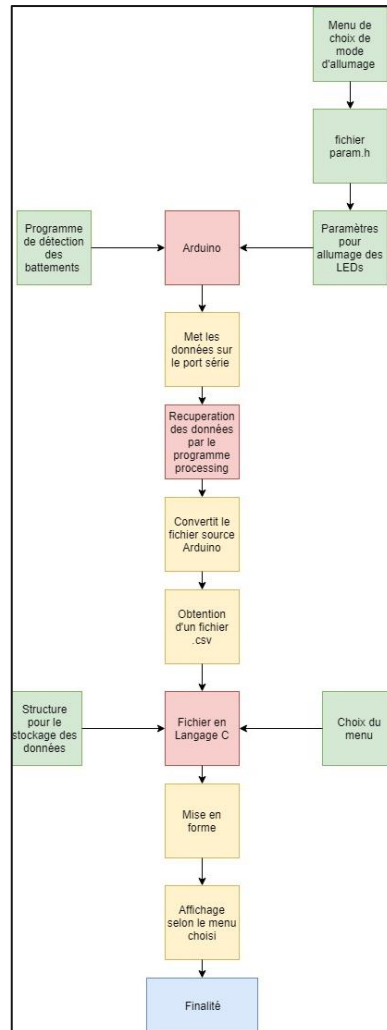
FEUILLE D'AVANCEMENT DU PROJET

Il faudra utiliser différents types de structures pour le quatrième module du projet, c'est-à-dire tout ce qui est en rapport avec le chargement en mémoire les informations du fichier .csv généré sous processing.

Le choix de la structure de données utilisée sera effectué de manière à avoir la plus efficace possible. On entend par là, avoir celle ayant la vitesse d'accès et de traitement la plus élevée.

Dépendances :

Voici la chronologie d'exécution du programme, si un élément vient à manquer, alors le programme ne fonctionnera pas.



Structures :

Structures du module 3.4 :

```
typedef struct ligne{
```

```
    int frequence;
```

```
    int temps;
```

```
} Ligne;
```

// Le structure de données qui va stocker les informations du csv est une file de données

```
typedef struct file{
```

```
    Ligne donnee;
```

```
    struct file *suivant;
```

```
} File;
```

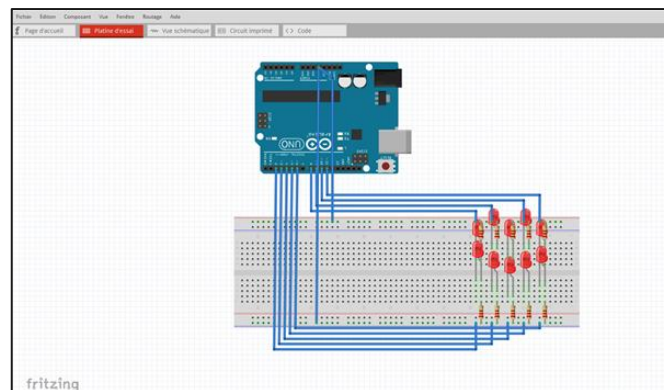


Troisième partie : MODULARISATION & WORKFLOW DE FONCTIONS & SCHEMAS ELECTRONIQUES

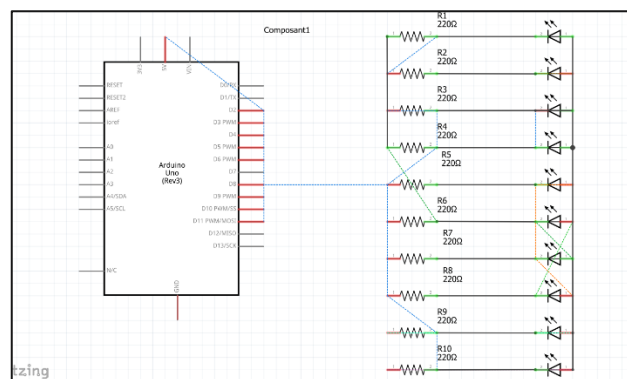
1. *Schéma électroniques avec les composants sur Fritzing (vue platine et vue schématique des module cardio et cœur de LEDs). Comme cette partie comporte une évaluation séparée du projet, vous pouvez faire un document à part entière.*

Voici les schémas de fritzing pour le module 3.2 réalisés par REGALL Nicolas :

- Tout d'abord, le montage complet du cœur :



Et ensuite la vue platine de ce même montage (comprenant peut-être des erreurs).



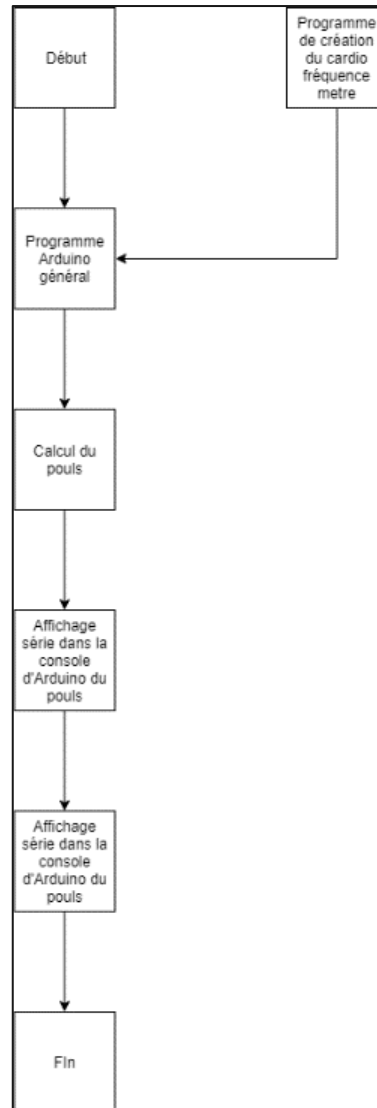
Pour le module 3.1, il est possible de retrouver tous les schémas dans le document word à ce sujet .



2. *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)*

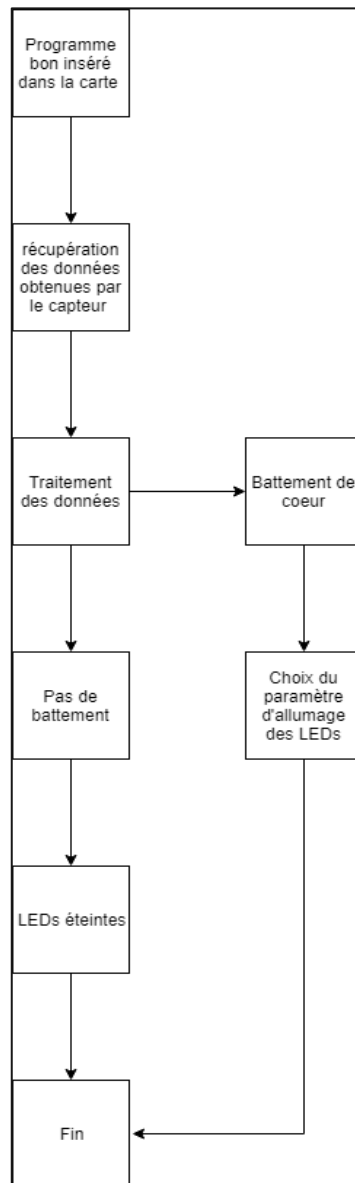
Module cardio :

Algorithme pour le module cardio



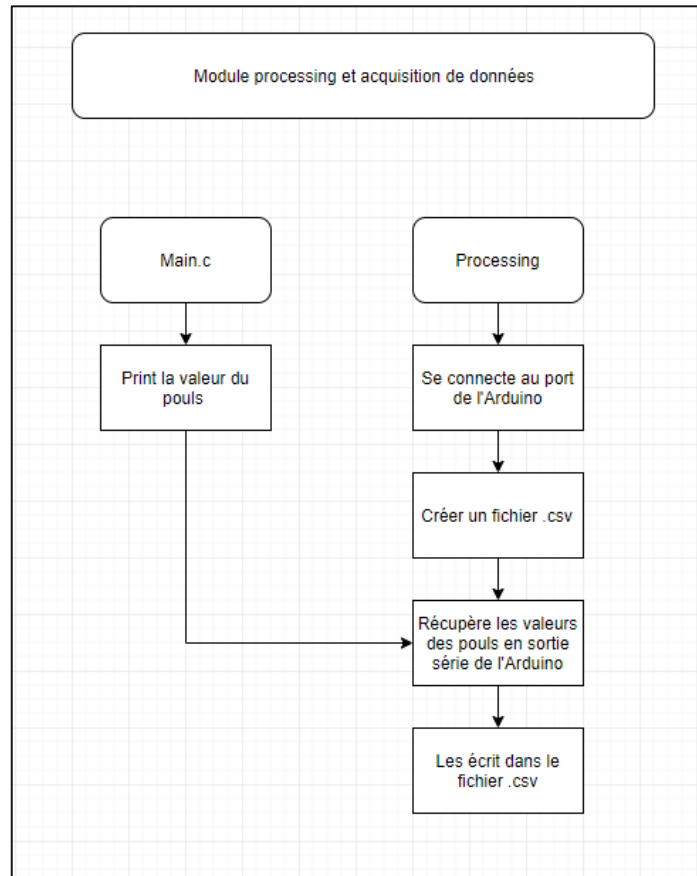
Module cœur de LEDs :

Algorithme pour le module Cœur de LEDs



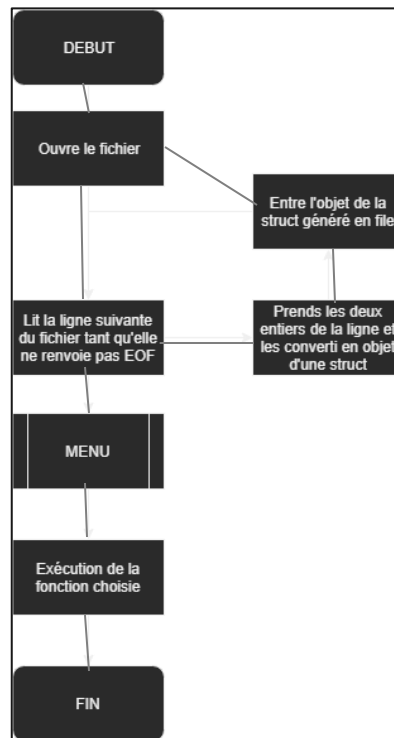
Module PreProcessing/acquisition des données :

Algorithme pour le module PreProcessing



Module lecture et traitement des données en C :

Algorithme pour le module lecture et traitement des données en C



La fonction MENU impose un choix à l'utilisateur, qui devra donc choisir quel traitement l'ordinateur va appliquer à la file de données.

Les choix du menu se répartissent en différentes fonctions, qui sont:

0: Quitter (stoppe le programme)

1: Afficher les données dans l'ordre enregistré

2: Afficher les données en ordre croissant/décroissant selon le pouls/le temps

2.1: Pouls

2.1.1: Croissant

2.1.2: Décroissant

2.2: Temps

2.2.1: Croissant

2.2.2: Décroissant

3: Rechercher les données pour un temps particulier

4: Afficher la moyenne de pouls dans une plage de temps donnée



5: Afficher la taille de la file

6: Rechercher le Maximum/Minimum de pous

6.1: Maximum

6.2: Minimum



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

Module 3.1 :

Void setup () → main.ino

Void loop () → main.ino

Void affichagePouls() → cardio.h / affichagePouls() → cardio.ino (le pouls va être affiché dans la console avec le temps)

Void calculPouls() → cardio.ino / calculPouls() → cardio.h

Module 3.2 :

Il y a uniquement la partie main.ino pour la partie arduino.

Il y a juste l'appel de la bibliothèque param.h

main.c : contient la fonction main et lance le menu en console.

Menu.c (et .h) : contient toutes les fonctions nécessaires pour créer les menus de paramétrage du cœur de LEDs

generationCode.c (et .h): contient toutes les fonctions nécessaires pour créer le fichier

param.h contenant le paramétrage pour le cœur de LED. Ce fichier est indispensable pour la compilation de ce module. Aucune contrainte sur la façon de générer le fichier

param.h ni sur le contenu de ce fichier (sachant néanmoins, qu'il doit contenir le paramétrage du cœur de LEDs)

Module 3.3 :

Void setup() → partie s'exécutant une seule fois, ou il y a la création du fichier.

Void draw() → fonction où on définit tout le reste qu'on veut dans notre programme, ou bien appeler des sous fonctions.

Void keyPressed() → Fonction de détection d'appui sur une touche.

Module 3.4 :

DANS MENU.H

// Cette fonction affiche le menu

int afficherMenu(int choixMenu);

// Cette fonction affiche le sous menu du choix n°2

int afficherMenu2(int choixMenu2) ;

// Cette fonction affiche le sous menu du choix n°3

PROJET A1 – FONDAMENTAUX SCIENTIFIQUES

Julio Santilario jsantilario@cesi.fr

A1 2018/2019



```
int afficherMenu6(int choixMenu6);  
  
DANS DONNEES.H  
  
// Chaque objet de cette structure est une ligne du fichier csv  
  
// La fonction qui va lire le csv  
  
int lecture();  
  
// La fonction qui ajoute un élément à la file de données  
  
void file_ajout(File **p_file, Ligne donnee);  
  
// La fonction qui retire un élément de la file  
  
Ligne file_retrait(File **p_file);  
  
// La fonction qui vide la file  
  
void file_clear(File **p_file);
```



Quatrième partie : REPARTISSEZ-VOUS LES TACHES

Nom : POMMERY Tristan	Rôle principal : Chef de Projet Module 3.1
------------------------------	---

Tâches	09/1 1	10/1 1	11/1 1	12/1 1	13/1 1	14/1 1	15/1 1	16/1 1	17/1 1	18/1 1	19/1 1
Faire les recherches											
Effectuer le prototype du montage											
Programmer en Arduino la détection et le calcul du pouls											
Effectuer le montage terminal											
Préparer la soutenance											



FEUILLE D'AVANCEMENT DU PROJET

Nom : REGALL Nicolas	Rôle principal : Gestionnaire Module 3.2
-----------------------------	---

Tâches	09/1 1	10/1 1	11/1 1	12/1 1	13/1 1	14/1 1	15/1 1	16/1 1	17/1 1	18/1 1	19/1 1
Faire des recherches											
Faire le montage sur Fritzing											
Faire le code Arduino et le tester											
Mettre en commun avec le module 3.1											
Faire le param.h											
Aider d'autres modules											
Soutenance											



PROJET – FONDAMENTAUX SCIENTIFIQUES

FEUILLE D'AVANCEMENT DU PROJET

Nom : SCHUBNEL Thomas

Rôle principal : Gérant GitHub | Module 3.3

Tâches	09/11	10/11	11/11	12/11	13/11	14/11	15/11	16/11	17/11	18/11	19/11
Faire des recherches											
Test du programme avec des valeurs créer											
Test du programme avec les valeurs du pouls											
Mettre en relation avec le module 3.4											
Aide sur d'autre modules (Surtout module 3.2)											
Préparation de la soutenance											



PROJET – FONDAMENTAUX SCIENTIFIQUES

FEUILLE D'AVANCEMENT DU PROJET

Nom : CARRE Etienne	Rôle principal : Aide rédacteur Module 3.4
----------------------------	---

Tâches	09/1 1	10/1 1	11/1 1	12/1 1	13/1 1	14/1 1	15/1 1	16/1 1	17/1 1	18/1 1	19/1 1
Analyser le module 3.4											
Implémenter la bibliothèque menus.h											
Implémenter la bibliothèque donnees.h											
Implémenter la bibliothèque actions.h											
Mettre en relation le code avec le csv fourni par le module 33											
Préparer la soutenance											

