

# PROJET 1

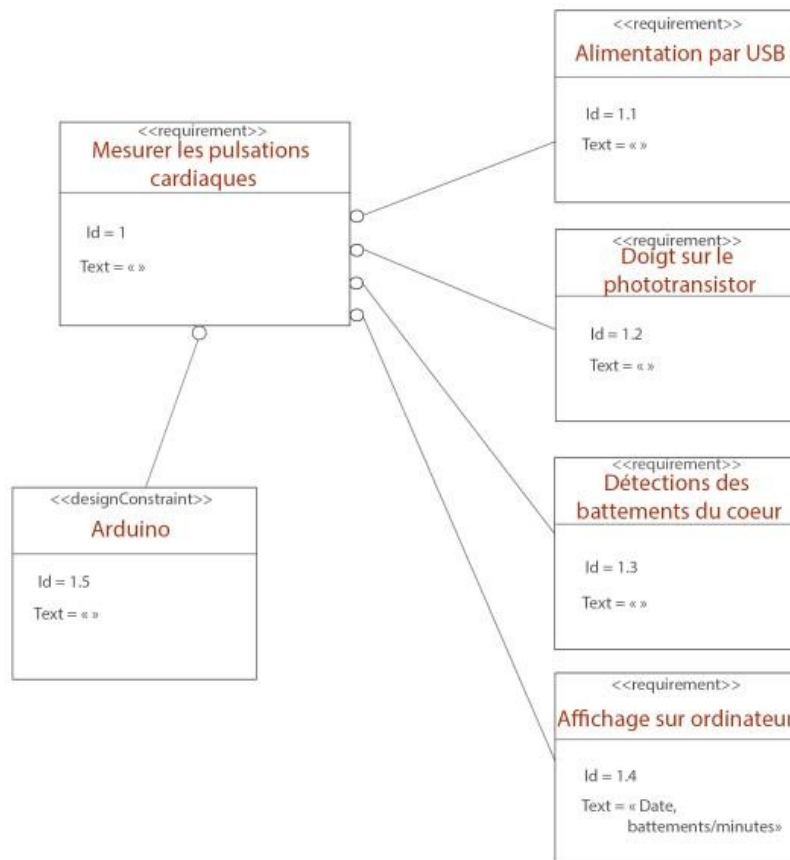
HEXART CARE

Wilfried – Boris – Kanwe – Serges

## TABLE DES MATIÈRES

Diagramme d'exigence.....	2
Assignation des tâches.....	2
Planning previsionnel.....	3
Analyse des écarts.....	4
Réalisation des modules.....	4
Module 1 :.....	5
Module 2 :.....	7
Module 3 :.....	8
Module 4 :.....	8
Bilans.....	9
Groupe.....	9
Conclusion.....	10

## DIAGRAMME D'EXIGENCE



## ASSIGNATION DES TACHES

Afin d'être le plus efficace possible, chaque personne du groupe a été assignée sur des tâches où elle se sentait à l'aise. Par exemple les personnes ayant un niveau correct en C ont plus manipulé les parties où elles étaient sujettes à faire du code.

Voici les principales tâches sur lesquelles les membres de notre équipe ont travaillé :

Taches	Membres de l'équipe assigné
Montage du module cardio	Kanwe Fruissala
Montage du module cœur	Kanwe, Serges, Wilfried, Boris
Code permettant de calculer le rythme cardiaque	Boris et Wilfried
Système d'affichage de LED	Serges
Application C afin de paramétrer le système d'affichage	Boris
Module 3 Processing permettant la transition de donnée sur PC	Serges et Wilfried
Module 4 de lecture de donnée	Wilfried, Bori, Kanwe, Serges

## PLANNING PREVISIONNEL

NOM : Wilfried								
Tâches	09/11	10/11	11/11	12/11	13/11	14/11	15/11	16/11
Prise de connaissance du sujet	✓	✓	✓	✓				
Réflexion et schématisation du circuit (cardio)			✓					
Réaliser le circuit cardio (+ fixation de problème)				✓				
Mise en commun du travail				✓			✓	
Développement du programme d'affichage					✓	✓		
Mise en commun finale							✓	
Tests & finalisation							✓	
Préparation orale								✓
NOM : Kanwe								
Tâches	09/11	10/11	11/11	12/11	13/11	14/11	15/11	16/11
Prise de connaissance du sujet	✓	✓	✓	✓				
Réflexion et schématisation du circuit (cardio)		(✓)						
Réaliser le circuit cardio (+ fixation de problème)				✓	✓			
Mise en commun du travail				✓			✓	
Développement du programme d'affichage					✓	✓		
Mise en commun finale							✓	
Tests & finalisation							✓	✓
NOM : Boris								
Tâches	09/11	10/11	11/11	12/11	13/11	14/11	15/11	16/11
Prise de connaissance du sujet	✓	✓	✓	✓				
Réflexion et schématisation du circuit (cardio)		(✓)		✓				
Réaliser le circuit cardio (+ fixation de problème)				✓	✓			
Mise en commun du travail				✓			✓	
Développement du programme d'affichage					✓	✓		
Mise en commun finale							✓	
Tests & finalisation							✓	✓
NOM : Serges								
Tâches	09/11	10/11	11/11	12/11	13/11	14/11	15/11	16/11
Prise de connaissance du sujet	✓	✓	✓	✓				
Réflexion et schématisation du circuit (cardio)		(✓)		✓				
Réaliser le circuit cardio (+ fixation de problème)				✓	✓			
Mise en commun du travail				✓			✓	
Développement du programme d'affichage						✓	✓	
Mise en commun finale							✓	
Tests & finalisation							✓	✓

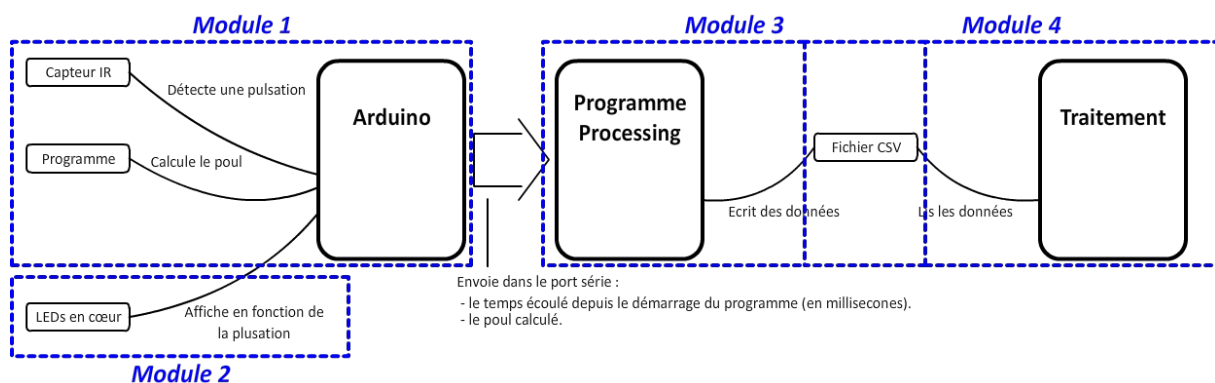
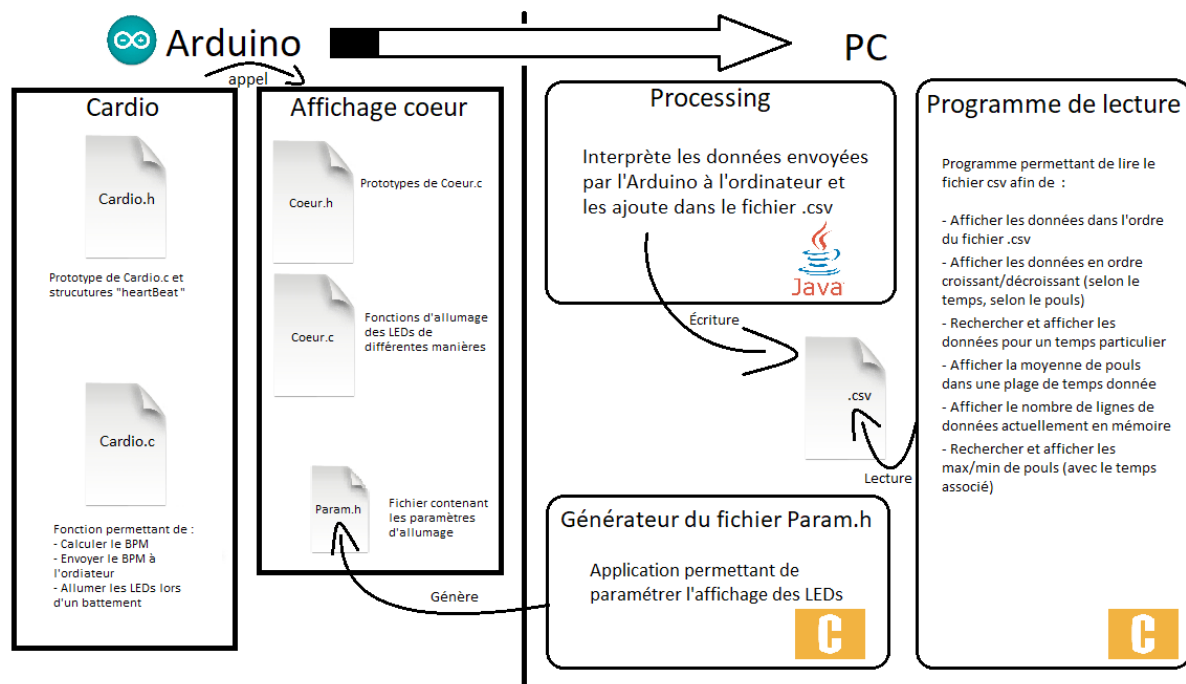
## ANALYSE DES ÉCARTS

Le principal écart concerne le montage électronique alors qu'il était initialement prévu de finir le montage dès le premier jour de travail, nous avons rencontré de nombreux problèmes qui nous ont handicapés durant toute la durée du projet.

Un autre problème a été la mise en commun puisqu'une partie du projet était dysfonctionnel nous avons dû trouver un autre moyen afin de simuler les battements du cœur.

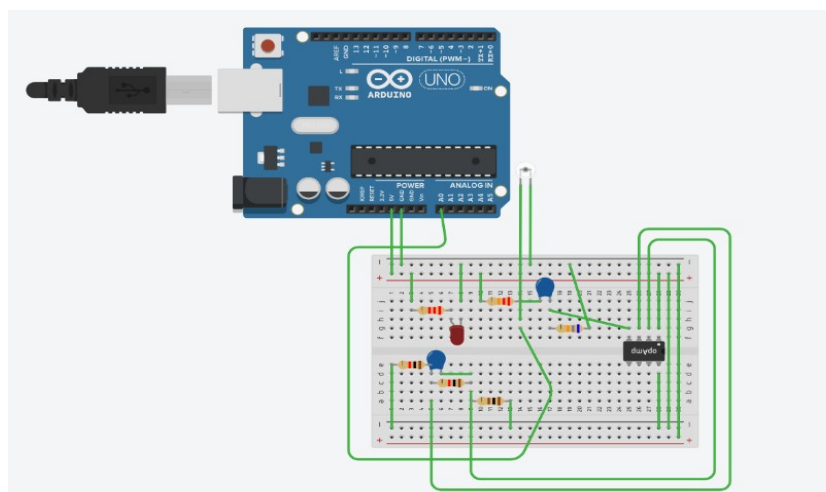
## RÉALISATION DES MODULES

### Schémas explicatifs de notre projet :



## MODULE 1 :

Nous avons d'abord commencé à réaliser le schéma qui était compliqué à réaliser le schéma électrique sur tinkercad.



Ensuite nous avons donc coder les fonctions qui permettent de faire afficher le pouls.

```
main.ino  cardio.h  cardio
#include <Arduino.h>
void pouls () ;
```

main.ino	cardio.h	cardio	coeur.h	coeur	param.h
----------	----------	--------	---------	-------	---------

```

#include "cardio.h"
#include <Arduino.h>
void pouls() // Fonction calculant le pouls total
{
  alternativdelay();
  time = millis();
  if ((time > (2900 + temps)) && (time < (3200 + temps)))
  {
    Serial.print(poul * 20);
    Serial.print(",");
    Serial.println(time);
    poul = 0;
    temps = temps + 3000;
  }
  else
  {
    if (analogRead(A0) < 350)
    {
      battement = 1;
      poul += battement;
    }
    else
    {
      battement = 0;
      poul += battement;
    }
  }
}

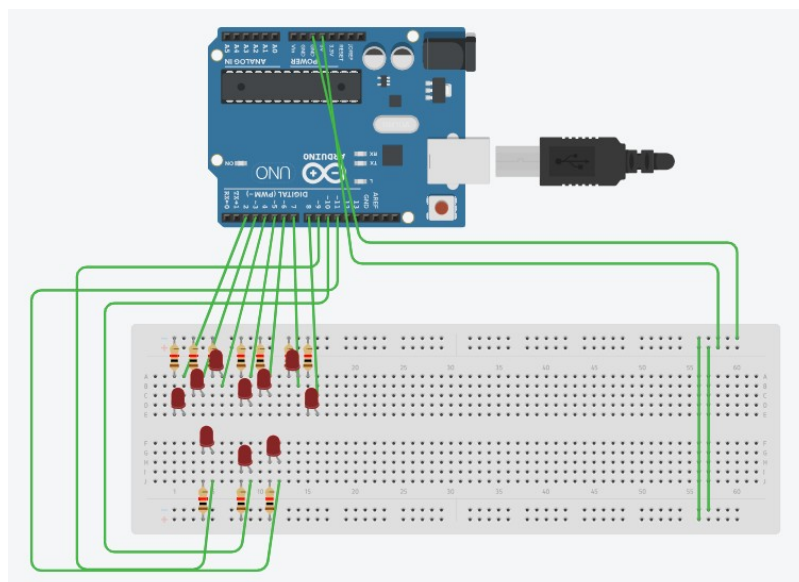
```

Voici la bibliothèque et le code qui permet d’afficher le pouls grâce au premier montage.

## MODULE 2 :

Nous avons réalisé un code qui permet de régler l’affichage des LEDs en fonctions de leur emplacement sur la carte Arduino. Nous avons donc fait un code pour chaque affichage par exemple clignoter, allumer une LED sur deux etc.

Dans cette partie nous avons aussi coder une partie en C qui est le début de notre code final.



### MODULE 3 :

Nous avons repris le code qui était donné en changeant le port donné. Ce code permet de réaliser un fichier CSV qui prélève les valeurs qui seront ensuite introduites dans notre code en C.

### MODULE 4 :

Ici nous avons fait la plus grosse partie du projet qui consiste à coder en C toutes les fonctions qu'on peut choisir comme mode d'affichage c'est-à-dire si on veut nos battements dans l'ordre croissant si on veut faire une moyenne de nos bous dans un intervalle de temps donné.

### BILANS

#### GROUPE

A la première lecture du sujet nous avons trouvé ce projet très intéressant mais assez complexe. La répartition des tâches a été automatique du fait des compétences de chacun. Dès le premier jour nous avons eu des difficultés sur le module 1 avec le montage du schéma incomplet donné dans le guide. Au niveau de la programmation, nous avons d'abord dû apprendre à programmer en C Arduino afin de réaliser les différents parties de celui-ci et revoir les principales notions qui nous ont servi durant le projet. Nous nous sommes beaucoup documentés afin de comprendre tous les composants. Nous avons dû apprendre à travailler en groupe en utilisant des logiciels de versioning.

### CONCLUSION

Nous avons réussi à obtenir un prototype quasi-fonctionnel, notre montage nous a tout de même permis de tester toutes les parties codées en simulant les battements de cœur par des mouvements au-dessus du phototransistor.