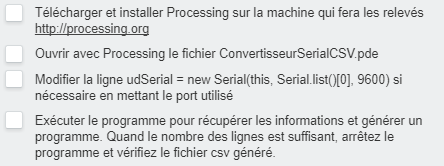
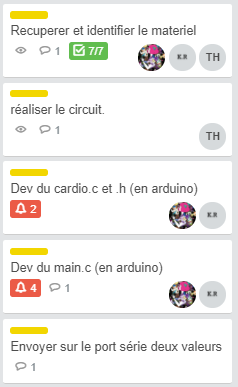
**Première partie : AVOIR UNE VUE D’ENSEMBLE DU PROJET**

1. ***OBJECTIFS***

Finir le travail

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



1. ***MACRO PLANNING (ETAPES)***

Voir les tableaux en tout en bas

**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.***

Code arduino :

sketch\_nov14a :

int i ;

If choix =

-Choix 1

-choix 2

-choix 3

-choix 4

Chenille ;

Chenille\_1sur2 ;

Chenille\_1sur3 ;

Chenille\_tout ;

Arduino processing  :

long temps = millis() ;

int nombre = random(70,90) ;

Serial.println(nombre);

Serial.println(temps)

Battement\_coeur :

import processing.serial.\*;

PrintWriter output;

Serial udSerial;

udSerial = new Serial(this, Serial.list()[0], 9600);

output = createWriter ("Battements.csv");

if (udSerial.available() > 0)

String SenVal = udSerial.readString();

output.print(SenVal);

if (SenVal != null)

System.out.print(SenVal);

output.flush();

output.close();

exit();

Cœur.cpp :

void Chenille(int i)

while(1)

pinMode(i,OUTPUT);

digitalWrite(i,HIGH);

delay(100);

digitalWrite(i,LOW);

for (i = 2; i <12; i++)

void Chenille\_1sur2(int i)

pinMode(i,OUTPUT);

digitalWrite(i,HIGH);

i++;

while(1)

for (i = 2; i <12; i++)

delay(500);

digitalWrite(i,LOW);

i++;

for (i = 2; i <12; i++)

delay(500);

void Chenille\_1sur3(int i)

while(1)

delay(500);

pinMode(i,OUTPUT);

digitalWrite(i,HIGH);

i++;

i++;

for (i = 2; i <12; i++)

digitalWrite(i,LOW);

i++;

i++;

for (i = 2; i <12; i++)

void Chenille\_tout(int i)

while(1)

pinMode(i,OUTPUT);

digitalWrite(i,HIGH);

for (i = 2; i <12; i++)

delay (500);

digitalWrite(i,LOW);

for (i = 2; i <12; i++)

delay(500);

void Chenille\_Ay(int i)

pinMode(i,OUTPUT);

digitalWrite(i,HIGH);

delay(100);

while(1)

pinMode(i,OUTPUT);

digitalWrite(i,LOW);

delay(100);

for (i = 2; i <12; i++)

for (i = 2; i <12; i++)

Cœur.h :

void Chenille(int i);

void Chenille\_1sur2(int i);

void Chenille\_1sur3(int i);

void Chenille\_tout(int i);

void Chenille\_Ay(int i);

**(Dossier Generationcode)**

Main.c :

int main()

int ch;

int choice = menu(&ch);

ecriture\_code (choice);

Menu.c :

int menu (int \*ch)

printf("Choississez le schema que le coeur doit reproduire \n 1.Chenille 2.Clignote 1 sur 2 3.Allume 1 sur 3 4.Tout Clignote 5.Chenille Aymeric");

scanf("%d",&ch);

return ch;

Menu.h :

int menu (int \*ch);

GenerationCode.c :

void ecriture\_code (ch)

FILE \*f = NULL;

f = fopen("../param.h", "w");

switch(ch)

case 1 :

fprintf(f,"#define choix 1"); //J'écris dans param.h une Macro qui servira a sélectionner un mode d'allumage dans mon code Arduino

break;

case 2 :

fprintf(f,"#define choix 2");

break;

case 3 :

fprintf(f,"#define choix 3");

break;

case 4 :

fprintf(f,"#define choix 4");

break;

case 5 :

fprintf(f,"#define choix 5");

break;

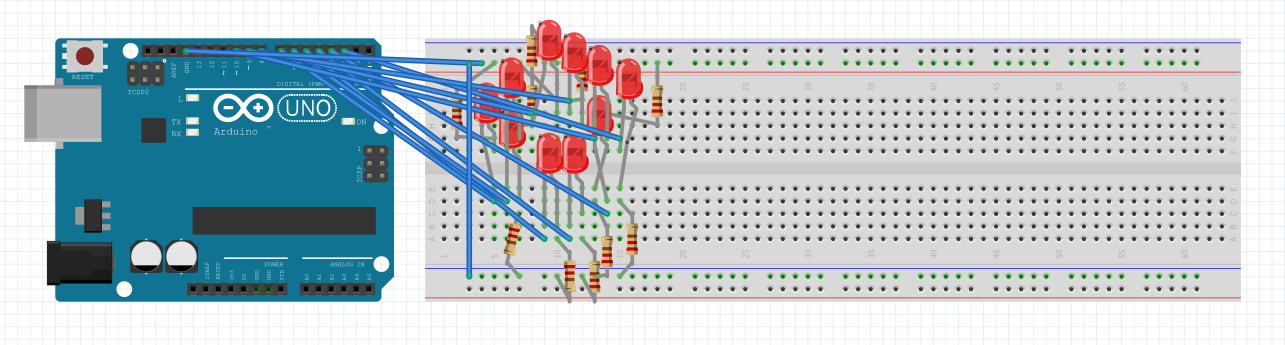
fclose(f);

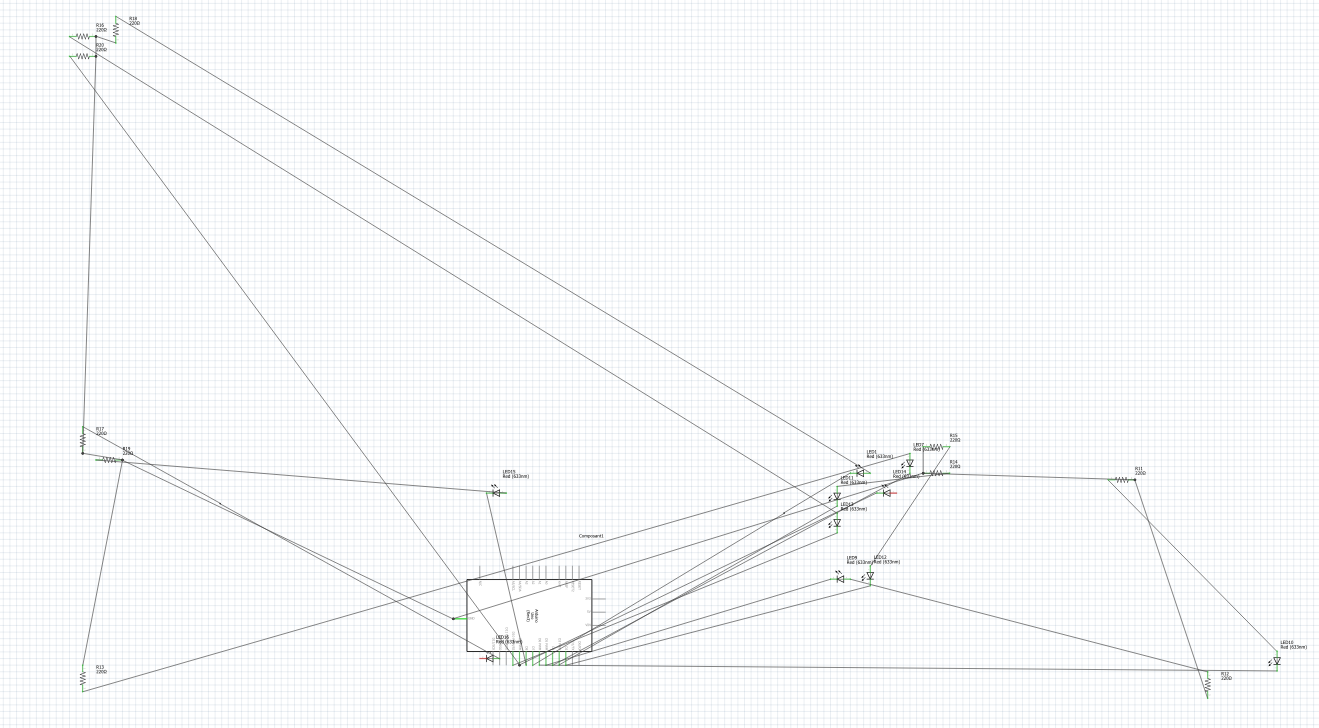
GenerationCode.h :

void ecriture\_code (int ch);

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

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.***





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)***

**Quatrième partie : REPARTITION DES TÂCHES**

|  |  |
| --- | --- |
| **Nom : HOYO** | **Rôle principal : Elec, Dev** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** | **20/11** |
| Trello |  |  |  | Fait |  |  |  |  |  |  |  |
| Récupérer et identifier le matériel |  |  |  | Fait |  |  |  |  |  |  |  |
| Réaliser le circuit |  |  |  | En cours | En cours | En cours | Fait |  |  |  |  |
| Développement du programme pour calculer la fréquence |  |  |  |  | En cours | Fait |  |  |  |  |  |
| Récupérer toutes les valeurs des pouls |  |  |  |  |  | Fait |  |  |  |  |  |
| Mettre les données pouls et temps dans un tableau |  |  |  |  |  |  | En cours |  |  |  |  |
| Envoyer sur le port série deux valeurs |  |  |  |  |  |  | fait |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : Rigal Kévin** | **Rôle principal : dev** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** | **20/11** |
| Récupérer et identifier le matériel |  |  |  | fait |  |  |  |  |  |  |  |
| Dev main.c (arduino) |  |  |  | En cours |  |  |  |  |  |  |  |
| Trello |  |  |  | fait |  |  |  |  |  |  |  |
| fritzing |  |  |  |  | En cours | fait |  |  |  |  |  |
| Récupérer et identifier le matériel (3.2)  Réaliser le montage physiquement  processing  Faire le module 3.4 (menu, main, données, actions)  Feuille d’avancement |  |  |  |  | Fait  fait  En cours | fait | En cours  fait |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : Laporte** | **Rôle principal : Elec,Dev** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** | **20/11** |
| Trello |  |  |  | Fait |  |  |  |  |  |  |  |
| Récupérer et identifier le matériel |  |  |  | Fait |  |  |  |  |  |  |  |
| Montage système Cœur LEDs |  |  |  |  | Fait |  |  |  |  |  |  |
| Code Allumage LED |  |  |  |  | Fait |  |  |  |  |  |  |
| Code pour gérer les mode LED en C (param.h)  Code Module 4 |  |  |  |  |  | Fait | Fait |  |  |  |  |

**Clôture du projet**

1. ***Validation, tests fonctionnels***
   1. Non fonctionnels
   2. Fonctionnels
   3. Fonctionnels
   4. ?
2. ***Retour d’expérience (REX) du projet***

chaud

1. ***Bilan***

***chaud***

1. ***Planning réel***