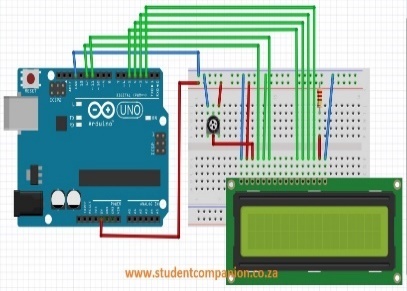
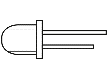
**Note :** Ceci est simplement « un modèle » à compléter selon vos soins. Des adaptations sont autorisées à condition de les justifier. **Ecoutez les conseils de votre parrain.**

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

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

C:\Users\depao\Pictures\EXIA\github ...png





***ARDUINO***



TRAITEMENT

BPM

Arduino

HeXart

Care

***PROGRAMMATION***

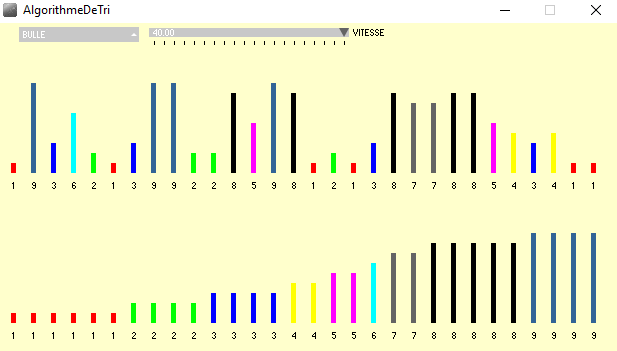


Code C

Arduino



MENU



Information

Relevés

Moyenne

TRI

**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.***

MODULE 2 (LED ARDUINO)

MODULE 4

Main ()

MAIN

Menu 1

Caterpillar

1 on 3 LED

1 on 2 LED

Every LED

1 on 1 LED

Time ascending

Pulse descending

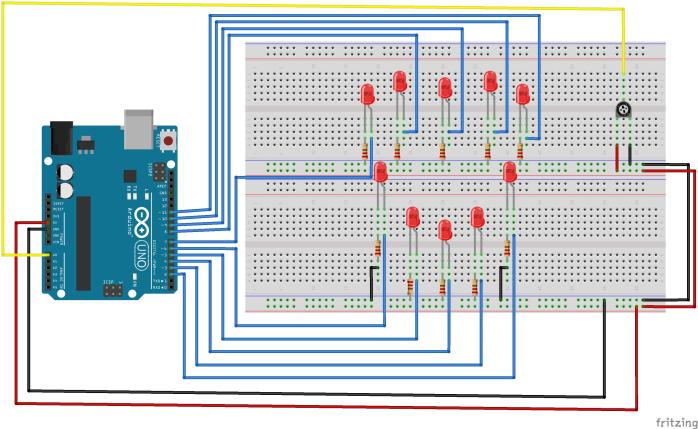
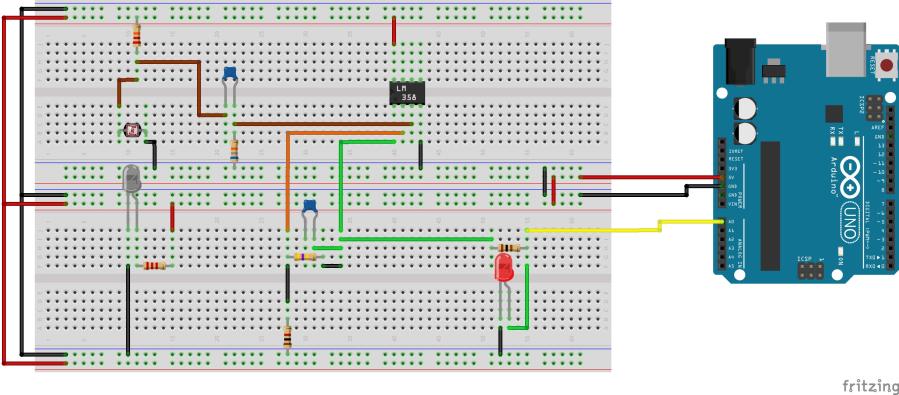
Time descending

Pulse ascending

Average Pulse Highest Pulse

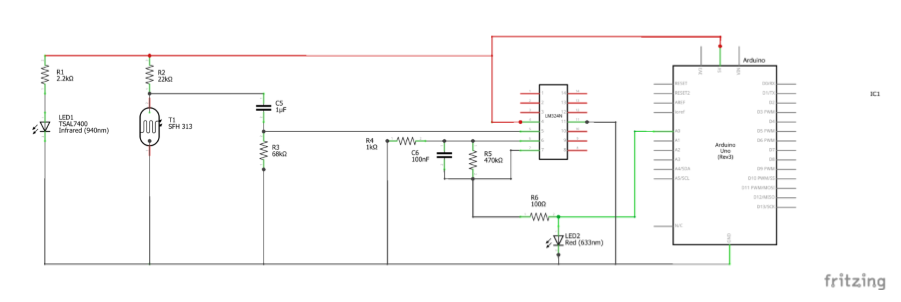
Patient record Data Lines Lowest Pulse

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

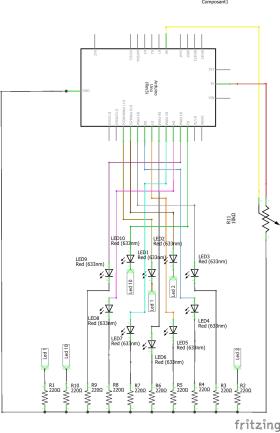
1. ***Schémas électroniques avec les composants sur Fritzing (vue platine et vue schématique des modules cardio et cœur de LEDs). Comme cette partie comporte une évaluation***

Module cœur de LED

Module cardio

******

Module cardio / vue platine

******

Module cœur de LED / vue platine

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 2

Main : appel la fonction menue

Menu : composé d’une structure conditionnelle qui renvoie soit vers la fonction LEDmenu ou soit vers HeartBeat

LEDmenu : on choisit le mode de clignotement des LED et ensuite on appel la fonction File qui permet d’écrire afin d’envoyer à la carte Arduino le code nécessaire au lancement de ce mode de fonctionnement

HeartBeat : affiche la valeur du capteur (récepteur IF) et affiche la tension d’entrée et calcule le pouls.

MODULE 4

Main : appel de la fonction Menu 1

Menu 1 : contient

-Menu Sort

-Calcule le Pouls de façon croissante

-Calcule le laps de temps de façon croissant

-Calcule le Pouls de façon décroissante

-Calcule le laps de temps de façon décroissant

-Menu Search

-Affiche les données du patient (pouls et temps)

-Affiche la tension moyenne

-Affiche le nombre de ligne dans les données

-Affiche le plus haut Battement Par Minute

-Affiche le plus bas BPM

-Menu LED (la fréquence d’allumage des LEDs se fait à chaque pulsation)

-Affiche 1 LED par 1 LED

-Affiche 1 LED sur 2

-Affiche 1 LED sur 3

-Affiche toutes les LEDs en même temps

-Affiche un allumage chenille (2 LEDs qui se déplacent)

Lecture et utilisation du .CSV : Avec le fgets on récupère les données du csv au format xx ;yy ;zz on les séparent grâce à la fonction strtok.

Tri des données : Algorithme de Tri à Bulle ; on tri de façon croissante ou décroissante selon la valeur UserChoice du SwitchCase. On affiche grâce à un For le contenu du tableau où sont stockés les valeurs.

Donner le nombre de ligne, le MIN/MAX : On lance le programme de tri et on compte combien de case contient le tableau, on affiche la valeur du nombre de ligne. Pour le min et le max on prend la première et la dernière valeur du tableau qui contient les données du .CSV.

Menu LED : On renvoie un fichier param.h qui va être lue par la carte Arduino, elle contient entre-autre un #DEFINE qui va incrémenter la variable afin de lancer le bon programme d’allumage des LEDs.

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

***MODULE 2***

int main(){

menu() ;

}

void menu() ;

void LEDmenu() ;

void file() ;

**MODULE 4**

void menu();

void SearchAndShow();

void Sort();

int Leave(int exit);

void file();

typedef struct

{

int poul;

char \* jours;

/\* char \* mois;

char \* heures;

char \* minutes;

char \* secondes; \*/

int millis;

}contact\_infos;

typedef enum

{

POUL,

/\* JOURS,

MOIS,

HEURES,

MINUTES,

SECONDES, \*/

MILLIS,

NB\_TYPE

}infos\_type;

char \* str\_dup (const char \* str);

contact\_infos \* get\_contact (const char \* nomcontact, int UserChoice, int Choice);

int launch(int UserChoice, int Choice);

int ChoiceMenu(int UserChoice,int exit);

int ChoiceSort(int UserChoice, int exit);

int ChoiceSearchAndShow(int Choice, int exit);

int Leave(int exit);

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

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **GROUPE** | | | |  | | | | | | |
| **Tâches** | **12/11** | **13/11** | **14/11** | | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Module 1 | X |  |  | |  |  |  |  |  |
| Module 2 | X | X |  | |  |  |  |  |  |
| Module 3 |  | X | X | | X |  |  |  |  |
| Module 4 |  |  | X | | X | X |  |  |  |
| Fritzing du module 1 | X |  |  | |  |  |  |  |  |
| Montage M1 | X |  | X | |  |  |  |  |  |
| Programmation  Arduino | X |  | X | |  |  |  |  |  |
| Processing | X | X | X | |  |  |  |  |  |
| Programmation M2 |  | X |  | |  |  |  |  |  |
| Calcul du poul |  |  | X | | X | X |  |  |  |
| Programmation M4 |  |  | X | | X | X |  |  |  |
| Assemblage montage et code |  |  |  | | X | X |  |  |  |
| **Création d’un CSV via Processing** |  |  | X | | X | X |  |  |  |
| Rassemblement des modules |  |  |  | |  | X |  |  |  |
| Powerpoint |  |  |  | |  | X | X | X |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nom : DALLAPORTA Timoté** | | | | **Electronique** | | | | | | |
| **Tâches** | **12/11** | **13/11** | **14/11** | | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Fritzing | X | X |  | |  |  |  |  |  |
| Montage Arduino | X |  | X | | X |  |  |  |  |
| Programmation  LED M2 | X | X | X | |  |  |  |  |  |
| Montage M1, récolte des données |  |  | X | |  |  |  |  |  |
| Programmation Arduino | X | X | X | | X |  |  |  |  |
| Création d’un CSV via Processing |  |  |  | | X |  |  |  |  |
| Rassemblement  Des modules |  |  |  | | X |  |  |  |  |
| Powerpoint |  |  |  | |  | X | X | X |  |
| **Nom : GIBHARDT Antoine** | **Rôle principal : Codeur** |  |  | |  |  |  |  |  |
| **Tâches** | **12/11** | **13/11** | **14/11** | | **15/11** | **16/11** |  |  |  |
| Montage Arduino | X |  |  | |  |  |  |  |  |
| Code Arduino | X |  |  | |  |  |  |  |  |
| Processing | X |  |  | |  |  |  |  |  |
| Fichier CSV  proccessing | X | X | X | |  |  |  |  |  |
| Programmation M2 |  | X |  | |  |  |  |  |  |
| Programmation M4 |  |  | X | |  |  |  |  |  |
| Menu M4 |  |  | X | |  |  |  |  |  |
| Algorithme de recherche |  |  | X | | X |  |  |  |  |
| Création d’un CSV via Processing |  |  |  | | X |  |  |  |  |
| Rassemblement  Des modules |  |  |  | | X |  |  |  |  |
| Powerpoint |  |  |  | |  | X | X | X |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nom :DE PAOLIS Lucas** | | | | **Rôle principal : Chef de projet / Codeur** | | | | | | |
| **Tâches** | **12/11** | **13/11** | **14/11** | | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** |
| Montage Arduino | X |  |  | |  |  |  |  |  |
| Fiche d’avancement projet | X | X | X | | X | X |  |  |  |
| Programmation LED | X |  |  | |  |  |  |  |  |
| Programmation M2 |  | X |  | |  |  |  |  |  |
| Programmation M1 |  |  | X | |  |  |  |  |  |
| Lecture/Extraction CSV |  |  | X | |  |  |  |  |  |
| Algorithme de tri |  |  | X | | X |  |  |  |  |
| Fonction M4 |  |  |  | | X | X |  |  |  |
| Rassemblement  Des modules |  |  |  | | X |  |  |  |  |
| Powerpoint |  |  |  | |  | X | X | X |  |