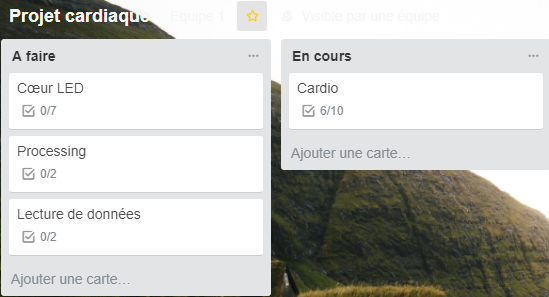
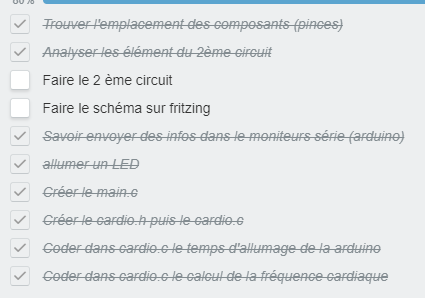
**Première partie : AVOIR UNE VUE D’ENSEMBLE DU PROJET**

1. ***OBJECTIFS***

Créer un circuit permettant de capter le poul et coder un programme permettant d’aficher et trier les resultats

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





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

Commencer chaque module dans l’ordre et se répartir les tâches entre développement et conception de circuit

**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.***

***Organisation des fichiers du code Arduino :***

Main.c

Cardio.h

Param.h

Cœur.h

Cardio.c

Cœur.c

**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.***

* Circuit du Cœur :

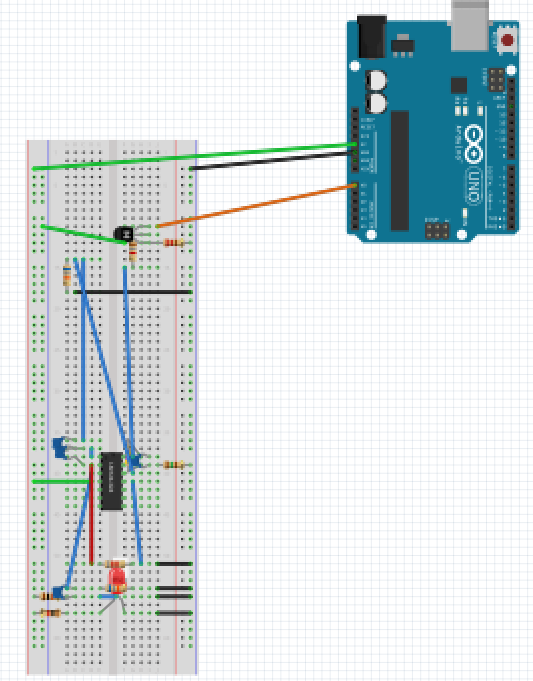
Une image contenant texte

Description générée avec un niveau de confiance élevé

Voici le montage du cœur de LEDs ci-contre avec 9 LEDs et 9 résistances.

On a donc placé les LEDs pour que celle-ci fasse la forme d’un cœur puis nous les avons d’abord reliées avec des ponts qu’on a branchés sur **-**. Ces ponts sont reliés sur l’anode de la LED, c’est-à-dire la patte la plus petite de la LED alors que la plus grande, la cathode est reliée sur la même ligne avec une résistance de 220 Ohms qui est la plus adéquate pour le circuit. Ainsi on relie la résistance à un pin du côté numérique hormis le pin 0 et 1 donc au 2. La résistance est donc reliée à l’Arduino par un fil. Ensuite les ponts sont tous reliées au – et un fil relie les – des deux côtés et du côté gauche un fil est relie le – du project board à un pin du côté numérique et qui permet d’avoir du courant dans les deux côtés en – de la project board

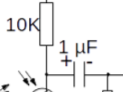
. Circuit général :



Voici le circuit général du module 1 qui est censé représenté un cardiofréquence mètre qui permet de mesurer la fréquence du rythme cardiaque d’un individu. Son principe est simple, il met en présence une LED IR (infrarouge), et un phototransistor associé à des résistances pour éviter la surtension du circuit, un amplificateur qui sert à augmenter la tension d’entrée de façon a ce qu’elle puisse passer le transistor de fin de circuit et pour enfin rejoindre l’Arduino. Le transistor utilisé est un transistor « tout ou rien » qui permet de faire passer le courant à partir d’un certain seuil limite. Si ce seuil est dépassé le signal sera envoyé directement sur un port analogique de l’Arduino. On utilise aussi un condensateur qui associé à une résistance formera un filtre (soit passe bas soit passe haut).

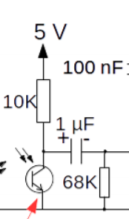
Il a été mis à notre disposition une illustration du circuit on peut donc en tirer :

* Filtre passe Bas :



La résistance est connectée à l’entrée du courant ce qui représente bien un passe bas

* Filtre passe Haut :

 la résistance de 68K est reliée au condensateur d’1 microF, ces derniers prennent leur sources au niveau de la sortie de la résistance 10K,donc ils ne reçoivent pas directement le courant de la source il s’agit donc bien d’un filtre passe Haut

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)***

Cœur.c

Tableau pour définir les pins de arduino

Cardio LED ***/***Cœur.c

Cœur led() ;

Boucle infinie avec incrémentation de chaque pins avec alternation entre LED allumé et LED éteintes (avec délais de 100 ms)

Switch avec nos fonctions de cœur LED

Case 1 :

Cœur\_led() ;

Case 2 :

Chenille() ;

Case 3 :

Un\_surdeux() ;

Case 4 :

un\_surtrois();

case 5 :

aleatoire\_led();

Chenille() ;

Boucle infinie avec incrémentation de chaque pins avec alternation entre LED allumé et LED éteintes (avec délais de 100 ms)

Un\_surdeux() ;

Boucle infinie avec incrémentation +2 de chaque pins avec alternation entre LED allumé et LED éteintes (avec délais de 100 ms)

Puis incrémentation de chaque nombre impair pour créer une alternance

tp=temps() ;

poul=FCM() ;

affichage de tp  « ; » poul

FCM() ;

Déclaration de variable temps

Affectation de time à la fonction millis()

Délais de 1seconde

Temps() ;

Déclaration de variable temps

Affectation de time à la fonction millis()

Délais de 1seconde

Aleatoire() ;

Un nombre aleatoire entre 2 et 12 (pins de leds)

Ensuite alternation entre allumé et eteint.

Un\_surtrois() ;

Boucle infinie avec incrémentation +3 de chaque pins avec alternation entre LED allumé et LED éteintes (avec délais de 100 ms)

Puis incrémentation de chaque nombre impair pour créer une alternance

Processing

Récupération des données par processing (code java)

Programme générant un poul aléatoire (60-80)

Insertion des données dans un fichier excel (csv)

Cœur.c

Affiche une console avec les différents choix

Main

On déclare choice qui va permettre d’affficher la variable de choix dans le arduino

On appel

la fonction choix

et

generation

Créer et écris un fichier param.h dans notre Arduino pour effectuer notre choix et lors du téléversement faire que le LED exécute ce dernier

Traitement

**Quatrième partie : REPARTITION DES TÂCHES**

|  |  |
| --- | --- |
| **Nom : Chambefort Hugo** | **Rôle principal : Chef de projet** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **10/11** | | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** | **20/11** |
| Développement | |  |  |  |  |  |  |  |  |  |  |  |
| Planners | |  |  |  |  |  |  |  |  |  |  |  |
| Github | |  |  |  |  |  |  |  |  |  |  |  |
| Organisation | |  |  |  |  |  |  |  |  |  |  |  |
| TEST | |  |  |  |  |  |  |  |  |  |  |  |
| Livrable | |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom :** Bourdeau Kevin | **Rôle principal :** Gestion des circuits |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** | **20/11** |
| Réalisation du cœur |  |  |  |  |  |  |  |  |  |  |  |
| Schéma fritzing |  |  |  |  |  |  |  |  |  |  |  |
| Developpement |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Nom : Abdoun Nadir** | **Rôle principal : Conception de circuit** |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tâches** | **10/11** | **11/11** | **12/11** | **13/11** | **14/11** | **15/11** | **16/11** | **17/11** | **18/11** | **19/11** | **20/11** |
| Réalisation du circuit général |  |  |  |  |  |  |  |  |  |  |  |
| Schéma fritzing |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

**Clôture du projet**

1. ***Validation, tests fonctionnels***

Mis à part que l’on n’arrive pas à récupérer les données de notre circuit (module 1) les autres parties sont fonctionnels (on peut récupérer les données avec des simulations)

1. ***Retour d’expérience (REX) du projet***

On a pu mêler électronique et développement pour une cause concrète même si notre projet n’est pas parfait on a pu avancer afin de résoudre nos problèmes en les approfondissant

1. ***Bilan***
2. ***Planning réel***