

16/11/2018

Projet : HeXart Care



ALLART MATHIS
GROUPE 6

Table des matières

I.	Introduction.....	3
1.	Présentation Projet	3
2.	Présentation Groupe	3
3.	Organisation Projet	4
II.	Gestion du projet.....	5
1.	Différentes tâches	5
2.	Réunion	5
3.	Planning prévu.....	6
4.	Planning réel.....	6
III.	Choix Techniques.....	7
1.	Logiciels utilisés	7
2.	Maquettes (Fritzing et Physique)	8
3.	Phase de test	10
4.	Réalisation	11
IV.	Conclusion	18
1.	Problème(s) rencontré(s)	18
2.	Perspective d'évolution.....	18
3.	Bilan	18

Table des Figures

Figure 1 : Logo HeXart Care	3
Figure 2 : Molin Thomas, Responsable matériel	3
Figure 3 : Allart Mathis, Chef de projet	3
Figure 4 : Oliver Leo, Membre	3
Figure 5 : Boudringhin Jonathan, Membre	3
Figure 6 : Tableau lors de la mise en place du projet.....	4
Figure 7 : Représentation graphique des tâches à réaliser.....	5
Figure 8 : Illustration d'une réunion.....	5
Figure 9 : Diagramme de Gantt du planning prévu	6
Figure 10 : Diagramme de Gantt du planning reel	6
Figure 11 : Tableau des logiciels utilisés.....	7
Figure 12 : platine d'essai fritzing du module 3.1	8
Figure 13 : Schéma Fritzing du module 3.1	8
Figure 14 : Schéma électrique physique du module 3.1	9
Figure 15 : Platine d'essai Fritzing du module 3.2	9
Figure 16 : Schéma Fritzing du module 3.2	9
Figure 17 : Schéma électrique physique du module 3.2	10
Figure 18 : Tableaux de tests 1/2	10
Figure 19 : Tableaux de tests 2/2	10
Figure 20 : Schémas électroniques module 1 1/2	11
Figure 21 : Schémas électroniques module 1 2/2	11
Figure 22 : Schéma Fritzing du Module 2	12
Figure 23 : Platine d'essai Fritzing du Module 2.....	12
Figure 24 : Schéma électronique physique module 2	13
Figure 25 : Schéma électronique module 2 en fonctionnement.....	13
Figure 26 : Menu affiché sur la console	14
Figure 27 : Exemple de fonction pour écrire dans le param.h	14
Figure 28 : ligne modifiée dans le processing	15
Figure 29 : Fichier csv obtenu.....	15
Figure 30 : Menu dans la console module 4.....	16
Figure 31 : affichage des valeurs triées	16
Figure 32 : Affichage du résultat d'une recherche	16
Figure 33 : Choix affiché dans la console	16
Figure 34 : Programme données.c	17
Figure 35 : ALLART Mathis.....	19
Figure 36 : OLIVIER Léo.....	19
Figure 37 : BOUDRINGHIN Jonathan	19
Figure 38 : MOLIN Thomas	19

I. Introduction

1. Présentation Projet

Le contexte du projet est d'aider une startup, nommée HeXart Care, à réaliser un **lecteur portable de fréquence cardiaque**. Pour cela quatre modules sont à réaliser :

- Le module Cardio
- Le module Cœur de Leds
- Le module Processing et acquisition de données
- Le module Lecture et Traitement de Données



FIGURE 1 : LOGO HEXART CARE

2. Présentation Groupe

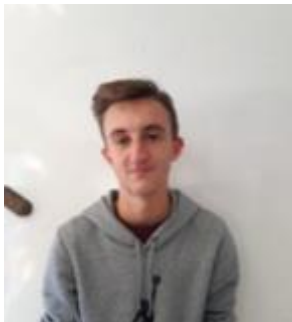


FIGURE 3 : ALLART MATHIS, CHEF DE PROJET



FIGURE 2 : MOLIN THOMAS,
RESPONSABLE MATÉRIEL,
ELECTRONICIEN



FIGURE 5 : BOUDRINGHIN
JONATHAN, INGENIEUR
PROGRAMMATION
ARDUINO ET C

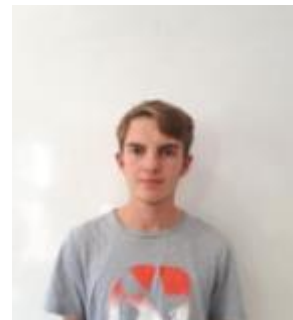


FIGURE 4 : OLIVER LEO,
INGENIEUR
PROGRAMMATION C

3. Organisation Projet

Pour réaliser ce projet, on a décider de donner un module à chaque personne :

Module Cardio	Module Cœur de Leds	Module Processing	Module Lecture et Traitement de données
MOLIN Thomas	BOUDRINGHIN Jonathan	ALLART Mathis	OLIVIER Leo

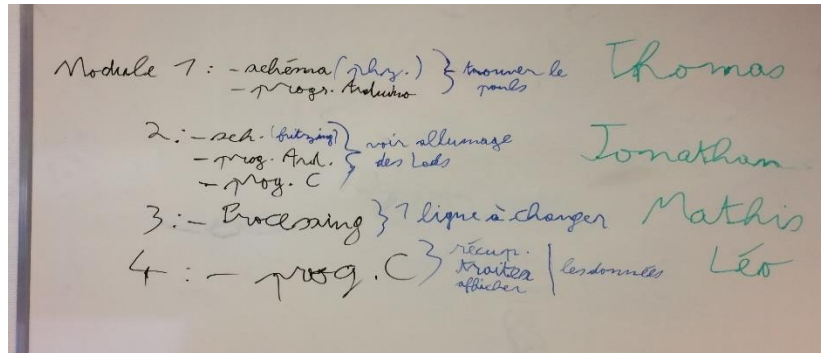


FIGURE 6 : TABLEAU LORS DE LA MISE EN PLACE DU PROJET

II. Gestion du projet

1. Différentes tâches

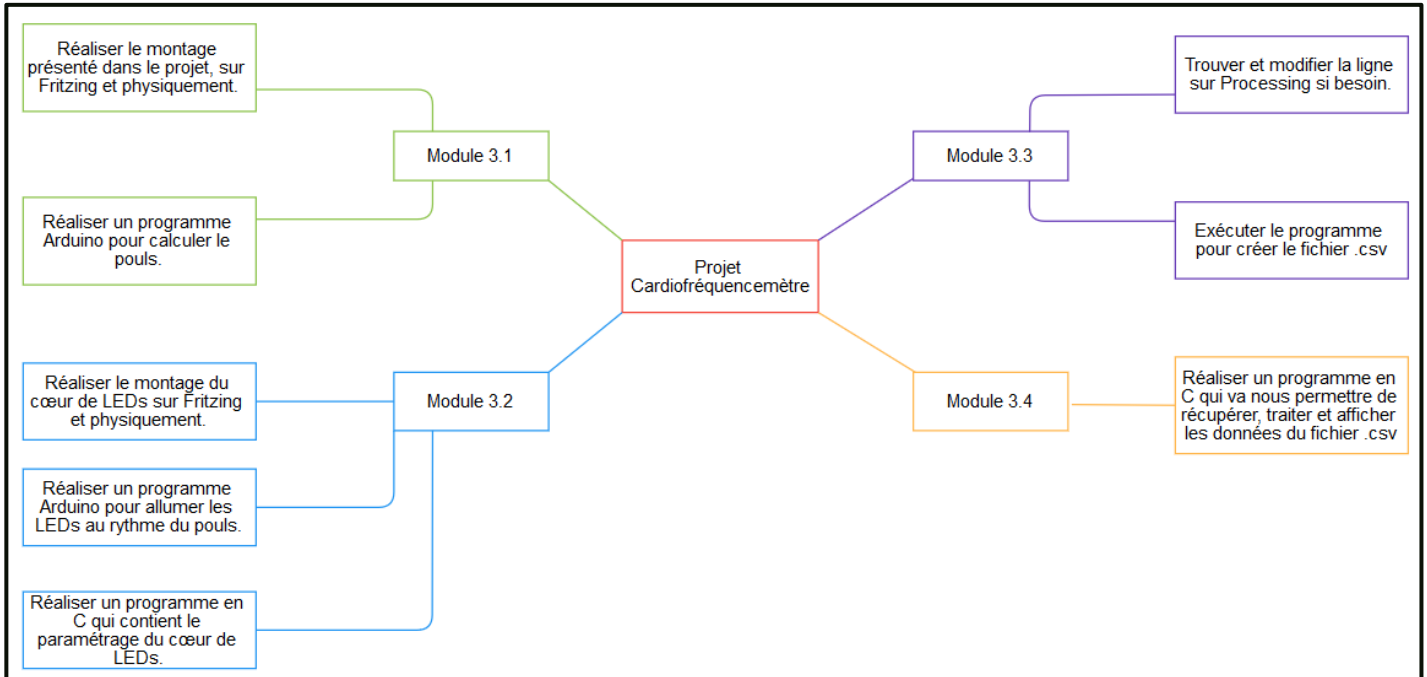


FIGURE 7 : REPRESENTATION GRAPHIQUE DES TACHES A REALISER

2. Réunion

On a organisé une réunion chaque matin pour vérifier que personne n'était en retard dans le groupe et savoir si une personne était en difficulté et donc aider cette personne par la suite.



FIGURE 8 : ILLUSTRATION D'UNE REUNION

Groupe 6

3. Planning prévu

Au début du projet, nous avons définis des tâches à réaliser, puis à quel moment faire ces tâches.

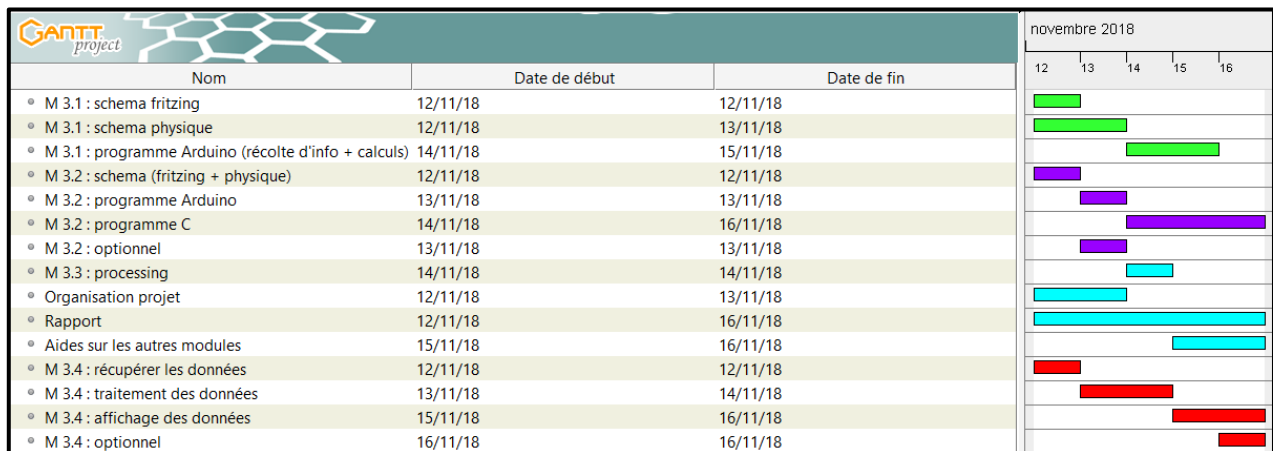


FIGURE 9 : DIAGRAMME DE GANTT DU PLANNING PREVU

4. Planning réel

On peut voir que le planning réel n'est pas le même à cause des problèmes rencontrés et des tâches découvertes durant le projet.



FIGURE 10 : DIAGRAMME DE GANTT DU PLANNING REEL

III. Choix Techniques

1. Logiciels utilisés







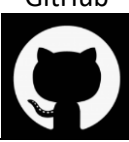



Logiciels	Descriptif
Word 	Réalisation du rapport, de la feuille d'avancement.
Excel 	Réalisation de la fiche de suivi, du tableau de tests.
Arduino 	Réalisation de programmes Arduino.
Code Blocks 	Réalisation de programmes en C.
Processing 	Réalisation du module pour la création du fichier .csv
Gantt Project 	Réalisation du diagramme de Gantt
GitHub 	Travail en collaboration
Cacoo 	Réalisation de schémas pour le rapport
Trello 	Organisation des tâches
Fritzing 	Réalisation de schémas électronique

FIGURE 11 : TABLEAU DES LOGICIELS UTILISÉS

2. Maquettes (Fritzing et Physique)

Pour les modules 1 et 2, les schémas Fritzing ont été réalisés suivis de la réalisation physique.

A. Module 1

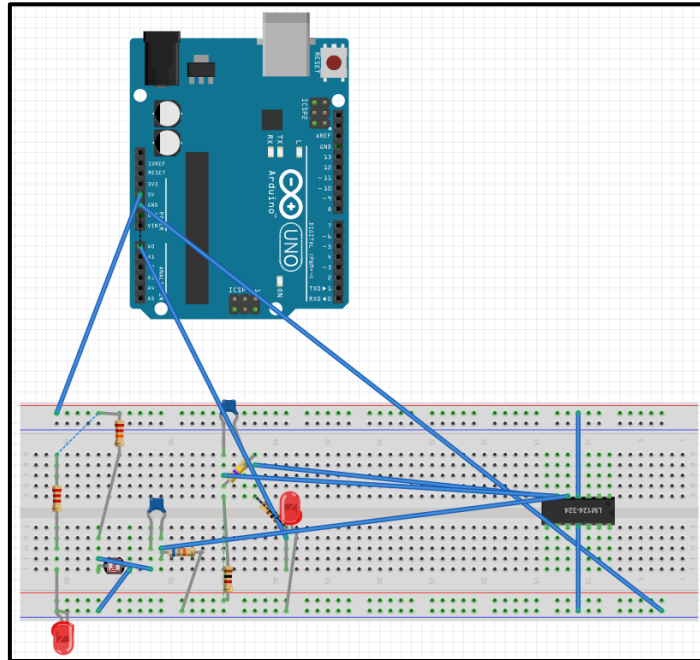


FIGURE 12 : PLATINE D'ESSAI FRITZING DU MODULE 3.1

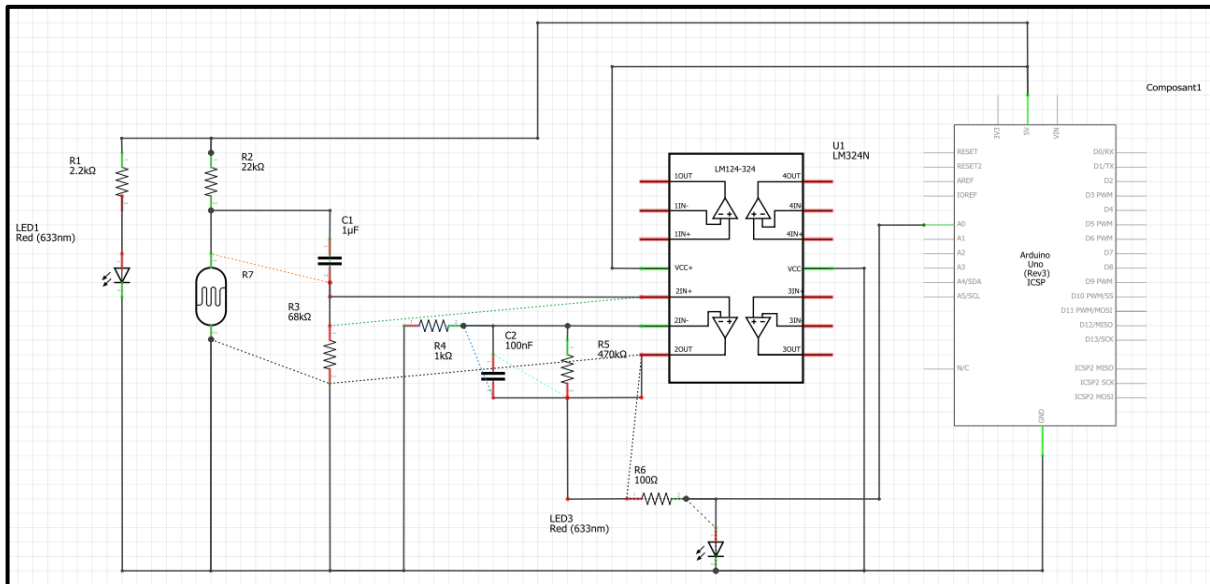


FIGURE 13 : SCHEMA FRITZING DU MODULE 3.1

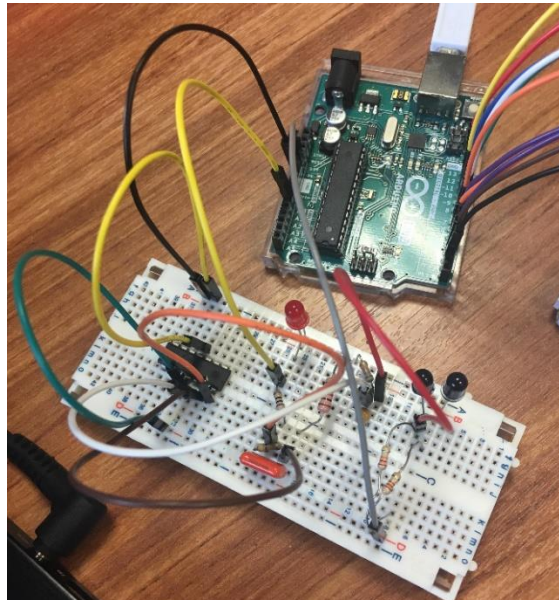


FIGURE 14 : SCHEMA ELECTRIQUE PHYSIQUE DU MODULE 3.1

B. Module 2

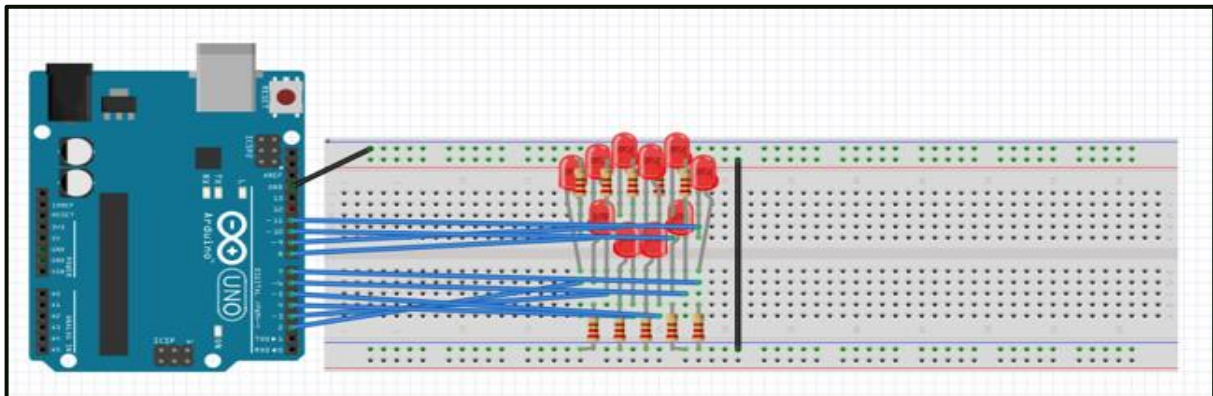


FIGURE 15 : PLATINE D'ESSAI FRITZING DU MODULE 3.2

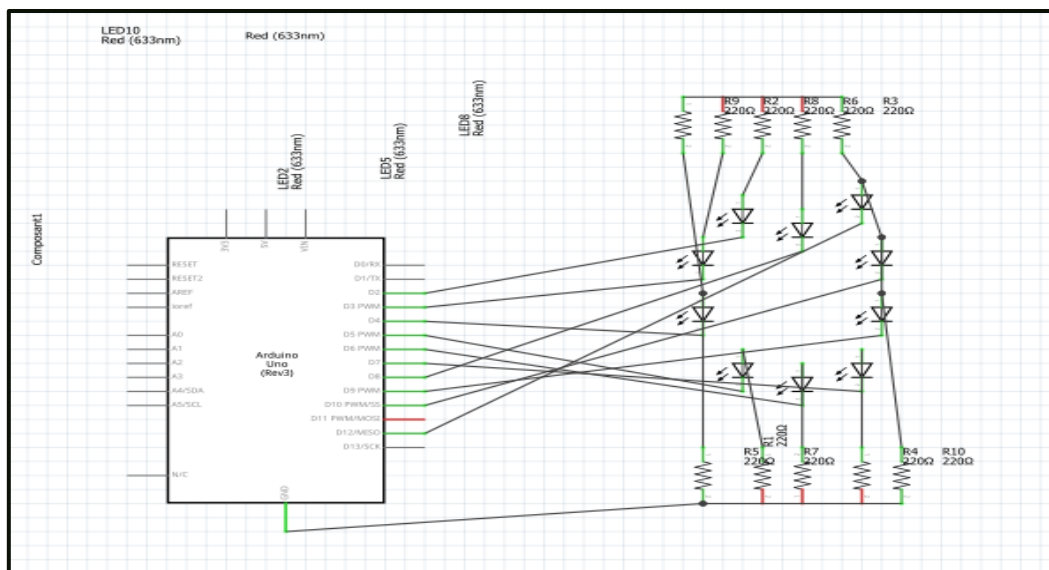
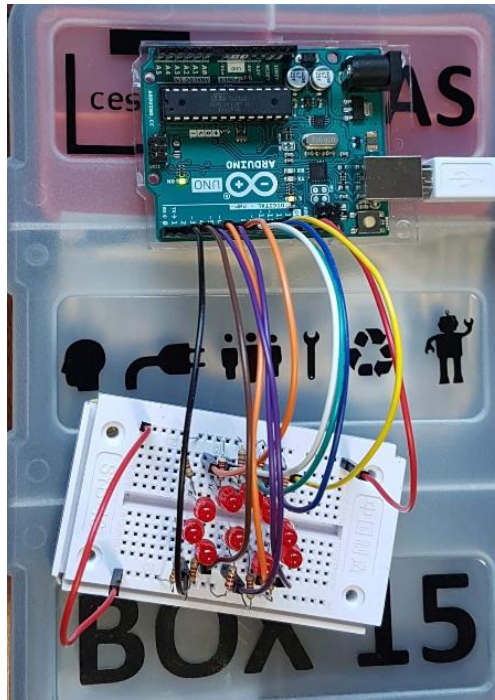


FIGURE 16 : SCHEMA FRITZING DU MODULE



**FIGURE 17 : SCHEMA ELECTRIQUE PHYSIQUE
DU MODULE 3.2**

3. Phase de test

Au début du projet, j'ai recherché des tests qui pourrait être réaliser durant le projet, je les ai classés dans un Excel.

Module 3.1		Module 3.2	
Tests	Validé ?	Tests	Validé ?
Mesure de plusieurs poulx	Oui	Allumer une led	Oui
Test du récepteur	Oui	Réaliser les fonctions	Oui
Envoyer des valeurs arbitraires	Oui	Test de l'allumage avec une valeur arbitraire du poulx	Oui

FIGURE 18 : TABLEAUX DE TESTS 1/2

Module 3.3		Module 3.4	
Tests	Validé ?	Tests	Validé ?
Test de création du fichier csv avec des valeurs arbitraires	Oui	Affichage de valeurs d'une structure	Oui
Création du fichier csv sur plusieurs colonnes	Oui	Test du tri fusion avec des valeurs arbitraires	Oui
		Affichage de valeurs d'un .csv avec des valeurs arbitraires	Oui
		Test de recherche séquentielle de valeurs dans un tableau	Oui
		Test de recherche dichotomique de valeurs dans un tableau	Oui

FIGURE 19 : TABLEAUX DE TESTS 2/2

4. Réalisation

A. Module 1

Pour réaliser le montage du cardiofréquencemètre nous nous sommes basés sur le montage proposé et avons utilisé ce matériel :

- 6 résistances (2.2k Ω , 22k Ω , 68k Ω , 1k Ω , 470k Ω et 100 Ω).
- 2 condensateurs (1 μ F et 100nF).
- 1 phototransistor sensible à l'infrarouge.
- 1 LED infrarouge.
- 1 LED.
- 1 Amplificateur LM324N.
- 1 Arduino Uno.

Explication montage

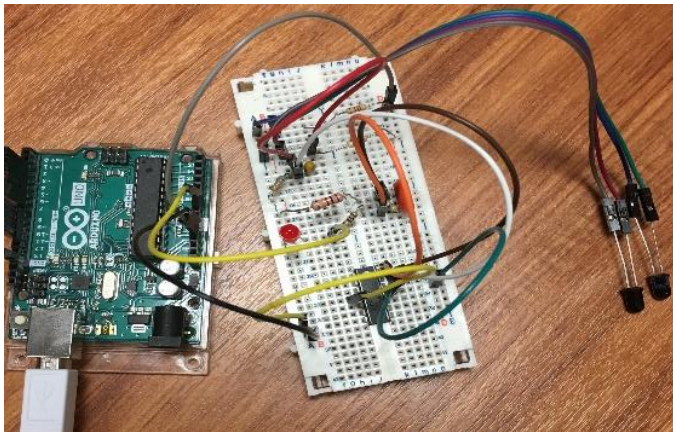


FIGURE 20 : SCHEMAS ELECTRONIQUES MODULE 1 1/2

Ce montage est construit avec deux filtres passe-bas (RC). Ces filtres permettent de laisser passer les basses fréquences et d'atténuer les hautes fréquences. Le montage n'est pas le même que le montage proposé. Nous avons changé la résistance relire à la LED infrarouge et avons mis une résistance de 220 Ω , donc moins puissante, afin que plus de courant puisse passer dans la LED infrarouge. La LED s'allume lorsqu'un doigt est sur le phototransistor, cela indique que le phototransistor reçoit des données.

Programme du pouls :

Afin de pouvoir mesurer le pouls nous avons créé un programme capable de détecter, mesurer et calculer notre pouls grâce au montage précédent. Son fonctionnement est simple, le programme enregistre des valeurs et les compares, ces valeurs sont considérées comme une pulsation si elle passe un certain seuil. Ensuite le programme entre dans une boucle de condition pour savoir si la valeur enregistrée est une pulsation ou un parasite. Lorsque le programme détecte que la valeur est une pulsation alors il enregistre le temps où cette pulsation a était enregistré, puis le programme effectue le calcul du pouls.

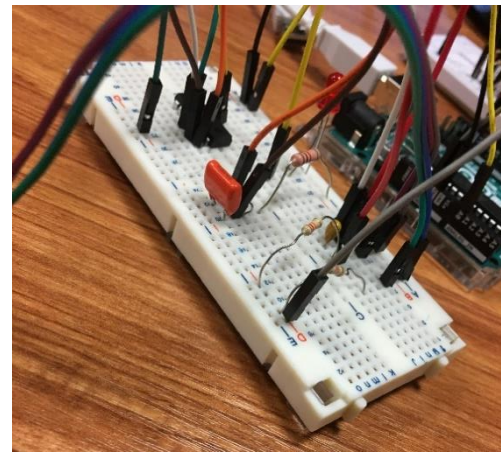


FIGURE 21 : SCHEMAS ELECTRONIQUES MODULE 1 2/2

Groupe 6

Programme qui envoie deux valeurs en csv :

Ce programme est utilisé pour générer aléatoirement des valeurs de pouls avec le temps en milliseconde depuis le démarrage du programme. Les valeurs du pouls peuvent aller de 60 à 80, car ceux sont les fréquences cardiaques au repos pour un adolescent ou un adulte.

B. Module 2

SCHEMA FRITZING

Pour commencer ce module nous avons utilisé Fritzing pour concevoir et tester les schémas électriques. Pour réaliser ce schéma nous utilisons 10 leds, 10 résistances et une carte Arduino.

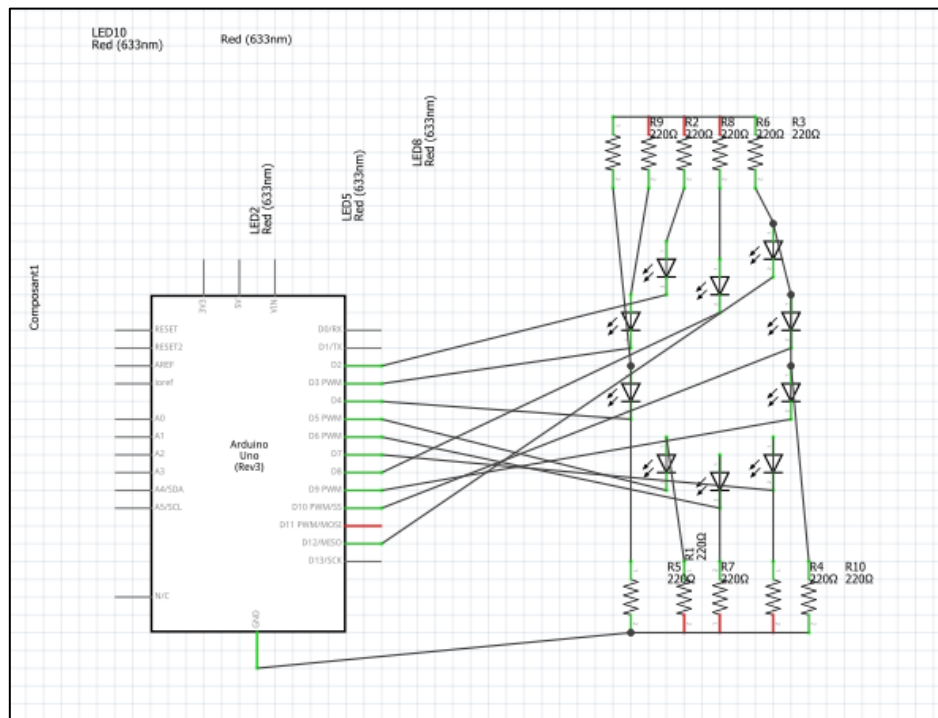


FIGURE 22 : SCHEMA FRITZING DU MODULE 2

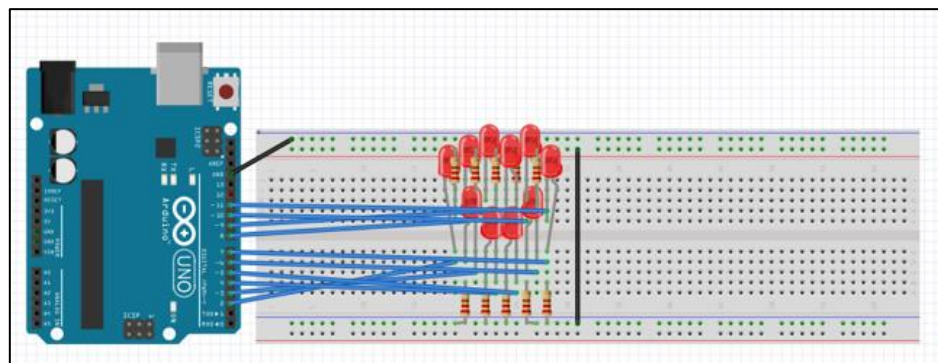


FIGURE 23 : PLATINE D'ESSAI FRITZING DU MODULE 2

SCHEMA ELECTRONIQUE

Dans un deuxième temps après avoir tester les composants nous avons réalisé le schéma électrique, après avoir recommencé plusieurs fois, le schéma à était enfin réalisé. Dans cette partie les leds permettent de faire le cœur et simuler au rythme du pouls. Ensuite nous nous servons des résistances pour éviter une hausse de courant dans les leds et pour donc éviter de griller celle-ci.

Pour finir nous utilisons l'Arduino et des pins pour faire clignoter les leds relier par des câbles, du pins 2 au pins 11. Mais aussi un pin pour le ground et un autre câble est brancher aux deux extrémités pour pouvoir installer le ground aux 10 résistances.

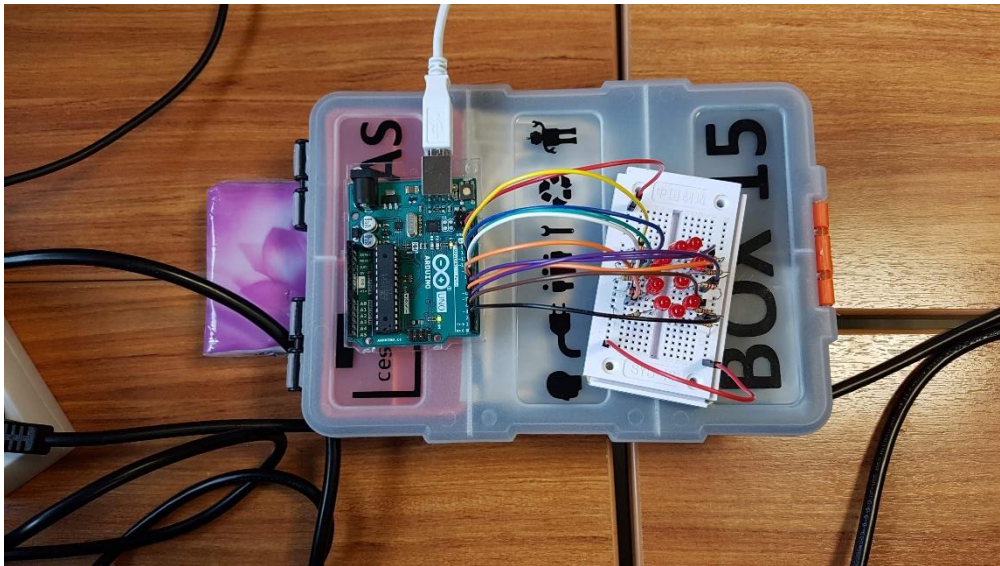


FIGURE 24 : SCHEMA ELECTRONIQUE PHYSIQUE MODULE 2

ARDUINO

Après avoir fait tout ce qui était la partie électronique, je suis passé au code Arduino, le code Arduino qui m'a permis de faire des programmes dans 3 fichiers différents, le cœur.c qui permet de mettre les pins utilisés donc le setup, ensuite une fonction loop qui permet de mettre le menu avec les différents appels de fonctions dans l'Arduino.

Ensuite le cœur.h qui contient tous les prototypes de fonctions.

Pour finir le param.h qui permet d'avoir la fonction avec les paramètres via le code c.

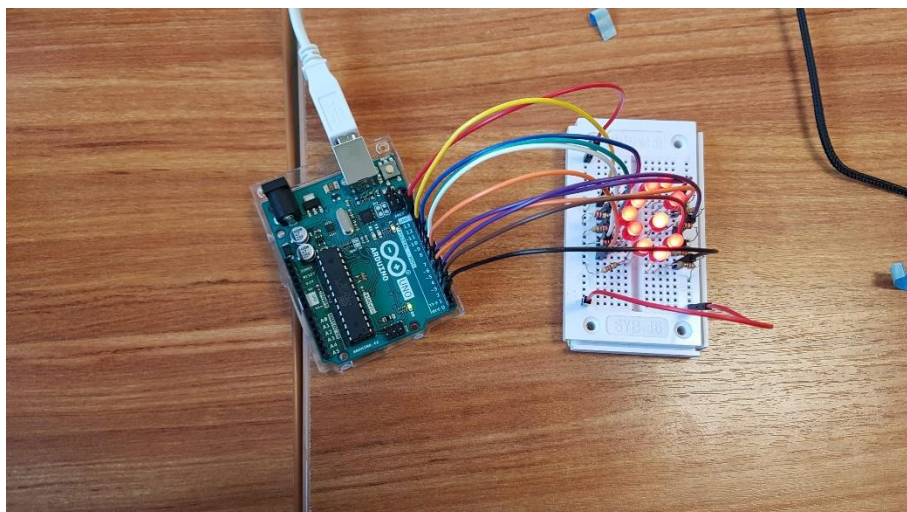


FIGURE 25 : SCHEMA ELECTRONIQUE MODULE 2 EN FONCTIONNEMENT

Groupe 6

Code en C

Ensuite dans le code C il nous faut simplement un main qui appelle le menu, dans le menu se trouve les choix de l'utilisateur de l'allumage des leds, avec un menu.h qui permet de décrire les prototypes des fonctions.

Dans un deuxième temps, nous utilisons un generationCode.c qui permet d'écrire dans le param.h et donc faire clignoter les leds. Ensuite, on a donc un generationCode.h qui permet d'y installer tous les prototypes de fonctions du code generationCode.c.

Ci-dessous le menu du module 3.2 représenter.

```

BIENVENUE DANS LE MENU

=====

|| 0 : Toutes les LEDs s'eteignent ||
|| 1 : Toutes les LEDs clignent ||
|| 2 : 1 LED sur 2 ||
|| 3 : 1 LED sur 3 ||
|| 4 : 1 LED au choix (reste) ||
|| 5 : Chenille ||
|| 6 : Option 1 : LED aleatoire ||
|| 7 : Option 2 : Ordre croissant ||
|| 8 : Option 3 : Toutes les LEDs sont allumees ||
|| 9 : Option 4 : 1 LED au choix (clignotent) ||

=====

Quel est votre choix?
-----

```

FIGURE 26 : MENU AFFICHE SUR LA CONSOLE

Le code que l'on utilise pour écrire dans le param.h, voici un exemple avec une led allumé au choix.

```

//led au choix reste allumé
void mode_de_leds4(int choix, int choix_led)
{
    FILE* fichier = NULL;

    fichier = fopen("coeur.c/param.h", "w");

    fprintf(fichier,
        "int mode = %d; \n"
        "int led = %i; \n\n"
        "void choixled() {\n"
        "digitalWrite (led, HIGH);\n"
        "digitalWrite (led, LOW);}\n", choix, choix_led);

    fclose(fichier);
}

```

FIGURE 27 : EXEMPLE DE FONCTION POUR ECRIRE DANS LE PARAM.H

C. Module 3

Ce module consiste à modifier si besoin un programme Processing donné. Ce programme permet de créer un fichier .csv avec des valeurs reçues de l'Arduino.

Modifications

Pour ma part, voici ce que j'ai modifier dans le programme :

```
// On écrit cette valeur dans le fichier csv
output.print(SenVal);
```

FIGURE 28 : LIGNE MODIFIÉE DANS LE PROCESSING

J'ai enlevé le 'ln' dans le print car j'ai remarqué qu'il **provoquait des cases vides** dans le fichier .csv.

J'ai ajouté des fonctions pour pouvoir afficher, avant les valeurs de poulx et de millisecondes, la **date** (JJ/MM/YYYY) et l'**heure** (H : min :sec). Pour cela j'ai utilisé une variable à laquelle j'attribuée **second()** pour les secondes, **minute()** pour les minutes... Ensuite il suffit de les écrire dans le fichier avec des '/' ou des ':' entre chaque élément.

Test

Pour tester j'ai créé un programme Arduino qui envoie des valeurs de poulx aléatoires, avec la fonction **random(60, 100)** pour envoyer des valeurs entre 60 et 100, et les valeurs du compteur dans le for, en envoyant le **i** à chaque tour de for.

Ensuite, j'ai téléversé ce programme dans l'Arduino et j'ai lancé le programme Processing, voici le fichier .csv obtenu :

15/11/2018	
10:56:51	
67	0
69	1
93	2
78	3
70	4
92	5
84	6
98	7
63	8
89	9

On a bien la date et l'heure.

FIGURE 29 : FICHIER CSV OBTENU

Sur la première colonne on a les valeurs aléatoires de poulx.

Dans la deuxième colonne on a les valeurs du compteur.

Groupe 6

D. Module 4

CODE en C :

Dans ce module nous devons lire un fichier .csv, utiliser et traiter les données du fichier .csv afin de les afficher. Toutes ces actions doivent être réalisées en langage C, nous avons donc utilisé l'IDE Code Blocks.

Pour réaliser ce module, nous avons utilisé trois fichiers sources et 3 fichiers headers qui sont : menu, actions et donnees.

Le fonctionnement du programme se fait de cette façon ; le main va appeler la fonction menu qui se trouve dans le fichier menu.c. Puis le programme va nous afficher ceci sur la console.

```
Choisit une fonctionnalites qui se trouve dans la liste suivante:
1 - Les donnees dans l ordre du fichier .csv
2 - Les donnees en ordre croissant
3 - Les donnees en ordre decroissant
4 - Chercher une donnees
5 - La moyenne de poulx dans une plage de temps donnee
6 - Le nombre de lignes de donnees actuellement en memoire
7 - La valeur maximale de poulx
8 - La valeur minimale de poulx
```

FIGURE 30 : MENU DANS LA CONSOLE MODULE 4

Puis on choisit une fonctionnalité dans la liste et on tape son numéro dans la console. Puis une fonction (qui se trouve dans actions.c) va être appelée et exécutée.

Pour les données qui sont triées on utilise un tri fusion.

```
50 53 54 55 55 56 56 57 58 59 59 60 64 66 67 67 71 71 71 74 77 77 78 79
79 80 80 81 82 84 84 85 86 86 87 88 88 89 90 92 93 94 95 95 96 97 98 98
```

FIGURE 31 : AFFICHAGE DES VALEURS TRIÉES

Pour rechercher une donnée on utilise une recherche séquentielle.

```
Tableau donne :
77 59 55 95 50 86 53 67 88 96 98 93 88 98 57 66 90 78 71 80 94 71 77 85 81 67 79 71 95 82 84 56 86 64 60 56 98 79
87 97 55 80 54 84 89 100 74 92 Element a rechercher : 55
La valeur 55 se trouve a la position 2.
La Valeur qui correspond en ms est 2
```

FIGURE 32 : AFFICHAGE DU RESULTAT D'UNE RECHERCHE

Puis on peut choisir si on veut continuer a utiliser le programme ou pas.

```
Continuer =1
Quitter =0
```

FIGURE 33 : CHOIX AFFICHÉ DANS LA CONSOLE

Groupe 6

Dans donnees.c on va créer une structure de donnée qui va récupérer les données du fichier.csv qui sont le temps et le poul. Cette structure va être utilisée presque à tous les fonctions du fichier actions.c .

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "donnees.h"
// création des listes chaînées
Valeurs* stocker (Valeurs* vlist)
{
    // initialisation
    FILE* fichier = NULL;
    int temps= 0 ;
    int i = 0;
    int poul = 0 ;

    // ouverture du fichier
    fichier = fopen("battements.csv", "r");
    // vérification qu'il existe bien le fichier
    if (fichier != NULL)
    {
        //stocker dans le tableau
        while(fscanf(fichier, "%i ; %i", &vlist[i].poul, &vlist[i].temps) != EOF)
        {
            // avancer le tableau
            i++;
        }
        // fermer le fichier
        fclose(fichier);
    }

    return vlist;
}
```

FIGURE 34 : PROGRAMME DONNEES.C

IV. Conclusion

1. Problème(s) rencontré(s)

Nous avons rencontré quelques problèmes durant ce projet :

La réalisation des schémas électriques, surtout celui du module 3.1.

Quelques difficultés avec les structures dans le programme C du module 3.4.

Problème lors du tri qui sépare les valeurs de pous et de temps, il est donc impossible de rechercher une valeur par rapport au temps après avoir trié.

Lors de la création du fichier csv, des cases vides apparaissaient entre chaque lignes, et parfois, des valeurs étaient coupées.

Problème de convention dans le programme en C du module 3.2, une ligne était trop longue et a été écrite sur plusieurs lignes.

Problème d'argumentation avec le float, affiche une valeur aléatoire lors d'un arrondi, par exemple pour 1/3 il affiche 0,336.

Problème avec l'acquisition de la valeur du pous avec le programme Arduino.

2. Perspective d'évolution

On pourrait trouver un moyen d'améliorer la précision de la mesure de la valeur du pous.

On pourrait afficher la valeur du pous sur un écran LCD.

On pourrait créer une boîte pour mettre l'Arduino et les schémas à l'intérieur (en laissant accessible le cœur de leds et la pince avec la led et le phototransistor IR)

3. Bilan

A. Groupe

Le projet s'est bien passé dans le groupe, tout le monde était concerné et travailleur, on a eu une bonne entente entre nous. C'est dommage que le chef de projet n'est pas pu beaucoup coder. Le module 3.1 a été compliqué pour nous. On a trouvé ce projet intéressant, il était agréable à réaliser. Il nous a permis de revoir tout ce qu'on avait vu auparavant et de découvrir une façon de s'organiser.

B. Personnelle

ALLART Mathis :

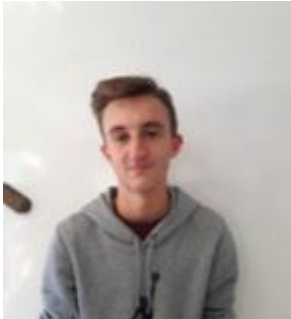


FIGURE 35 : ALLART MATHIS

J'ai trouvé très intéressant le contexte du sujet et la finalité. Par rapport à mon module, il était intéressant car j'ai découvert des fonctions (comme celle pour avoir l'heure) dans Processing. Mais la meilleure chose pour moi dans ce projet était le fait d'être chef de projet par rapport à toutes les responsabilités que j'ai eu, si tu rates ce que tu fais ou tu oublies quelque chose d'important, tout le groupe coule avec toi, mais dans le même temps on doit vérifier que tout le monde est dans les temps pour ne pas rater non plus. Par contre, vu que mon module n'était pas très long, j'ai moins programmé que tous les autres membres mais j'ai quand même fait quelques fonctions et j'ai aidé si une personne était en difficulté. C'est la seule chose que je reproche à ce rôle, le fait de moins programmer et de faire d'autres choses que je trouve moins intéressante à faire comme un Gantt ou un Trello par exemple. Ce projet m'a donc apporté, grâce au rôle de chef de projet, de l'assurance et il m'a aidé dans le domaine de l'organisation et de prévisions des tâches à faire.

OLIVIER Léo :

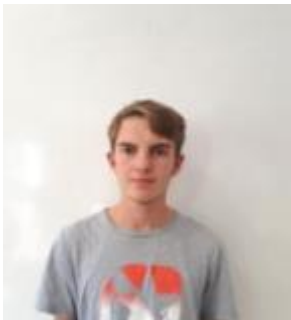


FIGURE 36 : OLIVIER LEO

J'ai trouvé le sujet très intéressant car il nous a fait utiliser toutes les connaissances apprises précédemment comme par exemple les algorithmes de tri et de recherche, l'électricité... Mon module m'a permis de consolider mes bases en langage C, et d'apprendre des choses comme l'exploitation d'une structure de données. J'ai trouvé que l'organisation pouvait être améliorée sur certains points comme par exemple se répartir les tâches en faisant les modules un par un, par exemple tout le monde fait le module 1 puis ensuite le module 2... Sinon, l'organisation était correcte et personnellement je trouve qu'elle nous a fait gagner beaucoup de temps. J'ai eu besoin d'aide pour ce module et j'ai aimé le fait que des promos supérieurs viennent nous aider lorsqu'on leur demande, cela m'a permis d'avancer plus rapidement.

Groupe 6

BOUDRINGHIN Jonathan :



FIGURE 37 : BOUDRINGHIN JONATHAN

C'était un bon projet, j'ai aimé réaliser le module 3.2. A la fin je me suis occupé avec Thomas du module 3.1 que j'ai trouvé moins intéressant que mon module. J'ai trouvé que l'organisation était bonne mais il y a des moments où les tâches n'étaient pas réalisées en temps et en heure. Le matériel était de meilleure qualité que l'année dernière ce qui nous a permis de réaliser au mieux le projet. Le projet m'a apporté plus de connaissances en C et en Arduino. J'ai trouvé que l'investissement du groupe était bon mais c'est dommage que Mathis n'ait pas pu coder plus.

MOLIN Thomas :



FIGURE 38 : MOLIN THOMAS

Le projet était intéressant. Mon module était compliqué, j'ai eu quelques difficultés pour réaliser le schéma électrique et le programme Arduino. Le projet m'a apporté des connaissances dans le langage Arduino et m'a permis de revoir ce que l'on avait vu en électronique dans l'année. J'ai trouvé le groupe très travailleur et l'organisation était bonne. Le travail que j'ai fourni ne me convient pas, je voulais mieux faire. J'ai trouvé bien le fait de s'entraider entre groupe, cela m'a permis de résoudre certains problèmes que j'avais.