

HeXart Care

Un cœur qui bat
au rythme
de vos rêves!



CONTENU

1.	Introduction	3
2.	Quelques notions scientifiques sur le sujet	4
3.	Description du projet.....	5
3.1.	MODULE CARDIO	5
3.2.	MODULE cœur DE LEDS	8
3.3.	MODULE PROCESSING ET ACQUISITION DE DONNEES	10
3.4.	MODULE LECTURE ET TRAITEMENT DE DONNEES.....	11
4.	Modalités d'évaluation	12
5.	Livrables techniques attendus	12
6.	Quelques conseils	16
7.	Organisation du projet	17



1. Introduction

San Diego, 01/11/2017.

HeXart Care est une startup très prometteuse spécialisée dans l'électronique et l'informatique. Son dernier projet innovant est un lecteur portatif grand public de la fréquence cardiaque.

L'entreprise a développé un savoir-faire depuis une dizaine d'années et elle s'impose petit à petit comme un acteur important dans la recherche et l'implémentation de solutions innovantes dans le monde de la santé.



Malheureusement cette bonne perspective a été coupée net le 31/10/2017 après avoir été victime d'un sabotage industriel. L'ingénieur principal de l'entreprise a disparu mystérieusement, emportant avec lui tous les prototypes ! Tous les salariés ont découvert leur poste formaté et toutes les sauvegardes ont été détruites. L'entreprise se trouve en grande difficulté car un prototype du projet devait être présenté le 20 novembre auprès des investisseurs.

HeXart Care a fait appel par le passé à des stagiaires Exia en A4 lors de sa mission à l'étranger. Les étudiants ingénieurs de l'EXIA sont très appréciés. Le dirigeant a fait appel à nous pour l'aider. Tous nos étudiants de A4 étant en stage actuellement, nous lui avons proposé de présenter le projet à nos étudiants de première année qui travaillent sur les fondamentaux scientifiques et informatiques en ce début d'année.

En accord avec l'entreprise, les meilleures équipes seront sélectionnées pour la présentation aux investisseurs lors de la soutenance du 20/11.



PROJET FONDAMENTAUX SCIENTIFIQUES

Julio Santilario jsantilario@cesi.fr

A1 2017/2018



2. Quelques notions scientifiques sur le sujet

La fréquence cardiaque est le nombre de battements cardiaques (ou pulsations) par unité de temps, généralement la minute. C'est une notion quantitative qui peut aussi se définir en nombre de cycles par seconde, par l'inverse de la période.

Par abus de langage, on confond souvent la fréquence cardiaque avec le rythme cardiaque ou le pouls. Le rythme cardiaque est une notion qualitative qui désigne la manière avec laquelle s'effectue une révolution cardiaque, la manière dont les cycles se succèdent. Le rythme cardiaque est le mécanisme à l'origine de la contraction des ventricules. Le pouls désigne quant à lui la perception au toucher de l'artère battante, permettant, certes d'évaluer les battements cardiaques, mais d'apporter également d'autres renseignements. La manière la plus simple d'évaluer le rythme cardiaque est de prendre le pouls

Selon les espèces animales, la fréquence cardiaque est très inégale. Ainsi chez l'homme, elle est d'environ 70 battements par minute ; chez la baleine, le plus gros mammifère actuel, la fréquence cardiaque est inférieure à 20 battements par minute ; chez les chiens, elle est comprise entre 70 et 90 battements par minute, chez les chats, elle est comprise entre 110 et 130 battements par minute, et chez la souris de 500 à 600 battements par minute.

Plusieurs conditions peuvent faire accélérer ou ralentir la fréquence cardiaque. Sa mesure est un outil diagnostique très important pour les médecins et les vétérinaires.

Chez les humains, la fréquence cardiaque au repos varie selon l'âge :

- Nouveau-né : 140 +/- 50
- 1–2 ans : 110 +/- 40
- 3–5 ans : 105 +/- 35
- 6–12 ans : 95 +/- 30
- Adolescent ou adulte : 70 +/- 10
- Personne âgée : 65 +/- 5

Chez l'adulte en bonne santé, au repos, le pouls se situe entre 50 (sportif pratiquant l'endurance) et 80 pulsations par minute. Pendant un effort, la fréquence cardiaque maximale théorique est de 220 moins l'âge (exemple : 180 à 40 ans).

La fréquence cardiaque de repos n'est pas constante sur 24 heures à cause des différents cycles biologiques. Elle est au maximum vers midi. La digestion, la chaleur et le froid entraînent une augmentation de la fréquence cardiaque.

Les facteurs de stress influent sur la fréquence cardiaque via trois mécanismes : les mécanismes nerveux, chimiques et physiques.

Pour plus d'information : https://fr.wikipedia.org/wiki/Fr%C3%A9quence_cardiaque



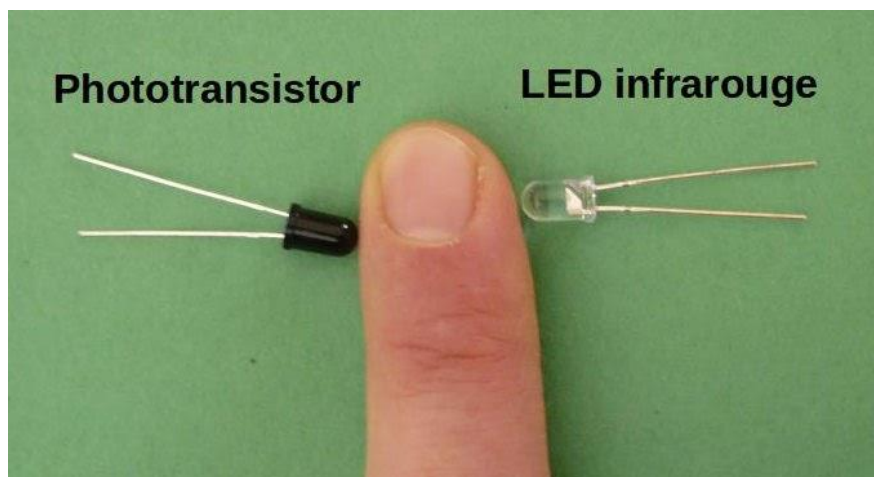
3. Description du projet

Très peu des ressources ont été récupérées après le sabotage. Sachant que l'ingénieur disparu était presque le seul à connaître l'intégralité du projet, personne n'est capable de vérifier si les documents retrouvés peuvent être exploités.

Voici la présentation du projet par le PDG de **HeXart Care**, M. KARTER Smith, de différents modules du projet sur lesquels l'entreprise était en train de travailler.

3.1. MODULE CARDIO

L'objectif de ce module est de construire un cardio-fréquencemètre se basant sur la photopléthysmographie, une méthode qui consiste à détecter les battements du cœur en mesurant le volume de sang dans les tissus au moyen d'une source de lumière et d'un détecteur.



La liste de matériel disponible est la suivante :

- LED Infrarouge (TSAL7400) - Tige la plus courte à GND
- Phototransistor sensible à l'infrarouge (TOPS-050) - Tige la plus longue à GND.
- Amplificateur opérationnel (LM324).
- Transistor (2N3904).
- Arduino Uno.
- Quelques condensateurs
- Jeux de résistances.

Malheureusement les seuls schémas, retrouvés après le sabotage, sont assez anciens. Voici quelques extraits de sites concurrents qui peuvent vous être utiles pour le montage et la compréhension de cette première partie. Ils sont votre seule ressource actuellement. :



<http://makezine.com/projects/ir-pulse-sensor/>

<http://electroniqueamateur.blogspot.fr/2015/01/mesurez-votre-rythme-cardiaque-avec.html>

<http://www.ni.com/example/13557/en/#h33>

<http://www.robotshop.com/letsmakerobots/simple-and-sensitive-finger-type-pulse-detector>

Voici quelques schémas du montage du départ :

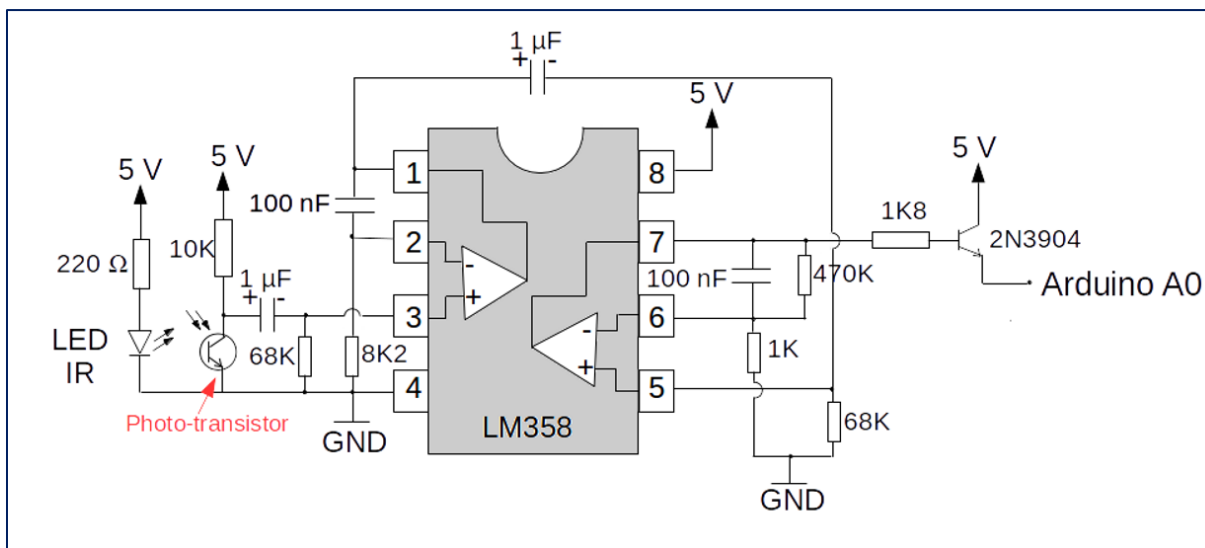


Schéma électronique du circuit "Cardio-fréquencemètre" – Version 1

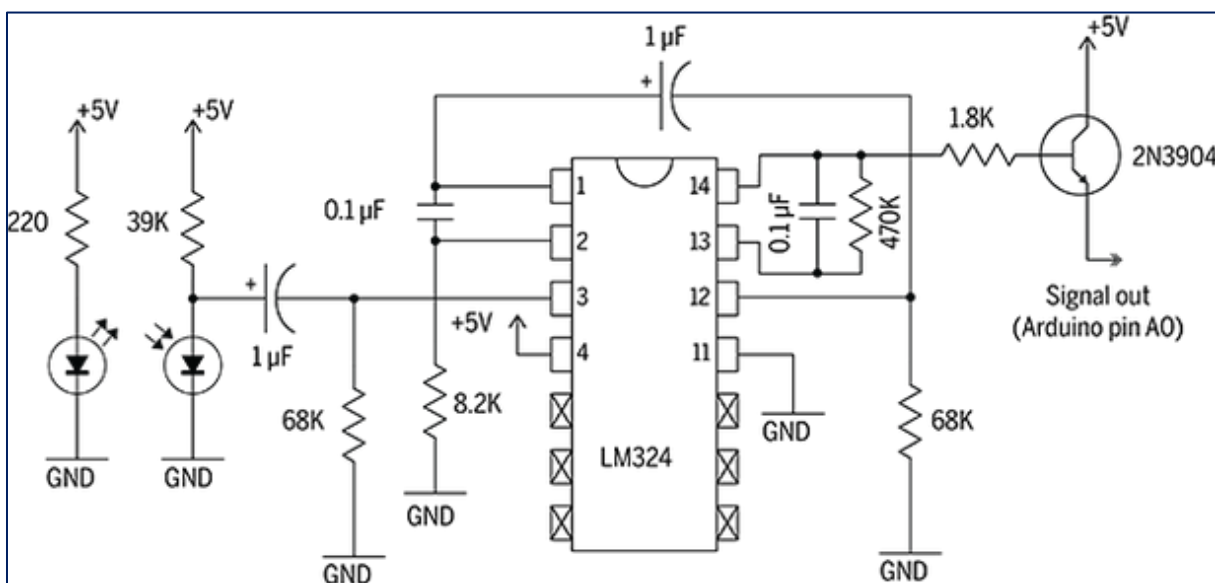
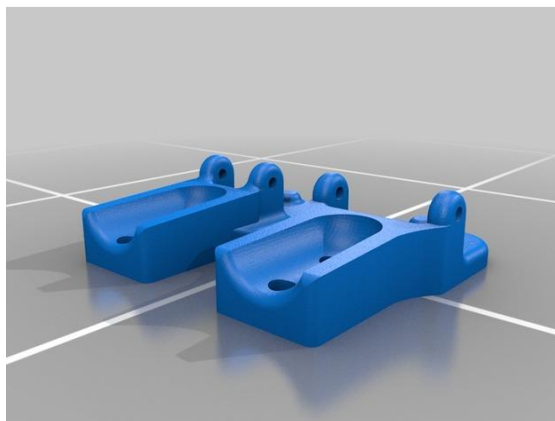


Schéma électronique du circuit "Cardio-fréquencemètre" - Version 2



Vous ne disposez pas de circuit LM358 mais le schéma avec les ampli-ops devrait vous aider à trouver une solution équivalente avec le module LM324, si vous partez sur le 1^{er} schéma.

Un ensemble des pinces pour le doigt déjà imprimées avec nos imprimantes 3D seront mis à votre disposition pour ce projet.



La principale difficulté de ce module est que nous ne savons pas s'il fonctionne : le prototype sur lequel travaillait l'ingénieur a disparu également ainsi que le rapport des tests. Les pinces comportent 2 trous en haut et 2 trous en bas. Nous ne savons pas l'endroit exacte où il faudra mettre l'émetteur et le récepteur.

N'ayant aucune certitude, le challenge principal est de faire le montage. Lors de la présentation aux investisseurs, il est important de montrer le montage même s'il ne fonctionne pas complètement.

La deuxième partie de ce module est la programmation Arduino qui détecte, mesure et calcule le pouls à partir des informations envoyées par le récepteur IR. Le code devra être organisé en deux fichiers :

- main.c (qui contient le loop et le setup)
- cardio.c (et .h) qui contient les fonctions propres à la récolte des informations et aux calculs pour avoir le pouls.

Ce module doit envoyer sur le port série deux valeurs (au format csv avec un « ; ») :

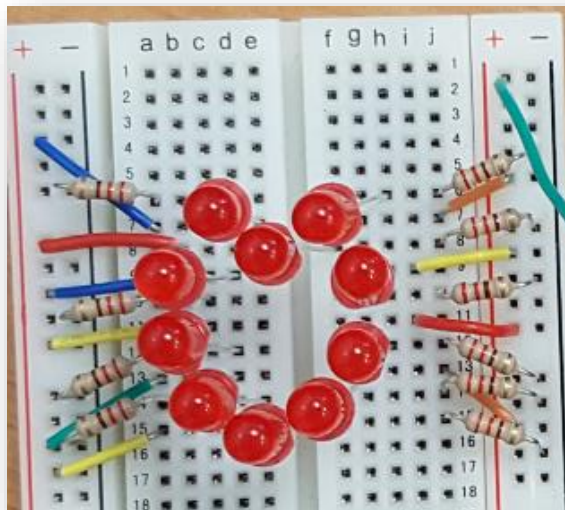
- le nombre de millisecondes depuis le démarrage du programme Arduino
- le pouls calculé à partir des informations récoltées par le récepteur IR.

En attendant le montage et la mise au point du module Cardio, **vous devez envoyer via le port série cette information** à intervalles réguliers (avec le même format décrit ci-dessus) générée de façon aléatoire (avec des valeurs cohérentes pour un pouls).



3.2. MODULE CŒUR DE LEDS

L'objectif est de réaliser un affichage original du pouls à travers un cœur de LEDS rouges. Nous souhaitons qu'un ensemble de LEDs (formant un cœur) soit allumé à chaque battement de cœur détecté. Voici un exemple de cœur.



Chaque LED doit être contrôlée de façon unitaire par l'Arduino (ce qui n'est pas forcément ce que vous voyez dans l'image ci-dessus).

La liste de matériel dont vous disposez est la suivante :

- Des transistors BC 547B K6 E (45V, 100 mA TO-92).
- 10 LEDs rouges classiques,
- Un jeu de résistances

La première partie de ce module est donc de penser au montage électronique, de le modéliser sur Fritzing (ou logiciel équivalent) et ensuite de le réaliser avec les composants donnés. Si d'autres petits composants s'avèrent nécessaires (résistances, transistors, ...), n'hésitez pas à contacter le responsable de votre centre.

La deuxième partie est la programmation Arduino des LEDs. Vous aurez un fichier *param.h* qui vous permettra de configurer la façon dont seront allumées les LEDs :

- Toutes les LEDs allumées en même temps au rythme des battements de cœur détectés par le module Cardio.
- 1 LED sur 2 allumée
- 1 LED sur 3 allumée, etc ...
- 1 seule LED allumée au choix (voir ci-après pour plus d'information)



- Allumer le cœur en mode « chenille » (une seule à la fois et selon le rythme des battements de cœur celle allumée s'éteint et on allume la suivante et ainsi de suite ...). Il est également possible d'allumer plus d'une LED à la fois pour faire la transition.
- Toute autre idée peut être proposée (*optionnel*)

Créez un module en C Arduino pour gérer le cœur (cœur.c, cœur.h et param.h)

La troisième partie est un programme en Langage C (*pas C Arduino*). Le but est de créer le fichier param.h de la deuxième partie à travers un générateur de code en C. Le programme devra être organisé en 3 fichiers :

- main.c : contient la fonction main et lance le menu en console.
- menu.c (et .h) : contient toutes les fonctions nécessaires pour créer les menus de paramétrage du cœur de LEDs
- generationCode.c (et .h): contient toutes les fonctions nécessaires pour créer le fichier param.h contenant le paramétrage pour le cœur de LED. Ce fichier est indispensable pour la compilation de ce module. Aucune contrainte sur la façon de générer le fichier param.h ni sur le contenu de ce fichier (sachant néanmoins, qu'il doit contenir le paramétrage du cœur de LEDs)



3.3.MODULE PROCESSING ET ACQUISITION DE DONNEES

Nous voulons récupérer toutes les valeurs des poulx lues sur la sortie série de l'Arduino et les enregistrer dans un fichier csv.

La visualisation et la récupération des données se basent sur l'environnement Processing 2.

Voici les étapes à suivre:

- Télécharger et installer Processing sur la machine qui fera les relevés <http://processing.org>
- Ouvrir avec Processing le fichier *ConvertisseurSerialCSV.pde*
- Modifier la ligne `udSerial = new Serial(this, Serial.list()[0], 9600)` si nécessaire en mettant le port utilisé.
- Exécuter le programme pour récupérer les informations et générer un programme. Quand le nombre des lignes est suffisant, arrêtez le programme et vérifiez le fichier csv généré.

Voici un exemple du fichier «.pde » que vous trouverez dans les ressources. Prenez le fichier tel quel et générez le fichier CSV (en ressources sur Moodle).

```
//From Arduino to Processing to Txt or cvs etc.
//import
import processing.serial.*;
//declare
PrintWriter output;
Serial udSerial;

void setup() {
    udSerial = new Serial(this, Serial.list()[0], 9600);
    output = createWriter ("Battements.csv");
}

void draw() {
    if (udSerial.available() > 0) {
        String SenVal = udSerial.readString();
        if (SenVal != null) {
            output.println(SenVal);
        }
    }
}

void keyPressed() {
    output.flush();
    output.close();
    exit();
}
```



3.4. MODULE LECTURE ET TRAITEMENT DE DONNEES

Ce module sera intégralement réalisé en Langage C (*pas C Arduino*). La mission de ce module est de charger en mémoire les informations du fichier csv générées par le module précédent et de traiter les données pour afficher certaines informations à partir de données brutes.

Premièrement, pensez à définir la structure de données pour charger en mémoire toutes les informations du fichier .csv :

- Pensez à une *struct* pour stocker chaque ligne d'information.
- Pensez à une structure de données permettant de manipuler les données facilement (file, arbre, ...).
- Pensez à comment vous-allez la représenter en interne (tableau statique, liste chaînée, ...)

Le code devra être organisé de la manière suivante :

- main. c
- menu.c (et menu.h) : contient les fonctions pour afficher les menus
- donnees.c (et donnees.h): contient toutes les fonctions nécessaires pour la manipulation de la structure de données et la lecture du fichier
- actions.c (et actions.h) : contient toutes les fonctions nécessaires pour le traitement de données (voir ci-après pour plus d'informations sur les actions possibles)

En résumé, le programme démarre, lit le fichier csv et le charge en mémoire. Il affiche ensuite un menu avec au moins les fonctionnalités suivantes :

- Afficher les données dans l'ordre du fichier .csv
- Afficher les données en ordre croissant/décroissant (selon le temps, selon le pouls)
- Rechercher et afficher les données pour un temps particulier
- Afficher la moyenne de pouls dans une plage de temps donnée
- Afficher le nombre de lignes de données actuellement en mémoire
- Rechercher et afficher les max/min de pouls (avec le temps associé)
- *Autres (facultatif)...*
- Quitter l'application

Optionnel : S'il vous reste du temps, vous pouvez faire rajouter, par le module Processing, une date et une heure (à la place du nombre de millisecondes) dans le fichier csv. Ceci peut se faire en modifiant le fichier « .pde ». Par conséquent, dans les options d'affichage du module 3.4, vous pouvez faire des tris et recherches sur ces données au lieu de le faire sur le temps.



4. Modalités d'évaluation

Le projet comporte deux notes dans deux UEs différentes :

- Note du projet Fondamentaux scientifiques (UE1.1)
- Note sur les schémas électroniques du projet Fondamentaux scientifiques (UE1.2)

La note finale du projet sera calculée de la manière suivante :

- à 40% sur la réalisation des 4 modules obligatoires (note de groupe).
- à 40% sur la question individuelle posée par les jurys lors de la soutenance
- à 10% par l'évaluation par les pairs
- à 10% par le jury en fonction de la prestation orale lors de la soutenance

Les grilles avec les critères d'évaluations seront disponibles sur Moodle.

IMPORTANT : le projet est composé d'une partie OBLIGATOIRE. Ce qui est demandé doit d'être totalement réalisé et présenté le jour de la soutenance. Si cette partie n'est pas réalisée, la note sera C pour chaque personne du groupe (voire D si les jurys constatent un manque important de travail).

Les fonctionnalités optionnelles peuvent vous aider à compenser certains manques dans vos réalisations. **MAIS** attention à ne pas vous disperser pour autant dans trop de choses à la fois. Elles seront prises en compte seulement si la partie obligatoire est réalisée.

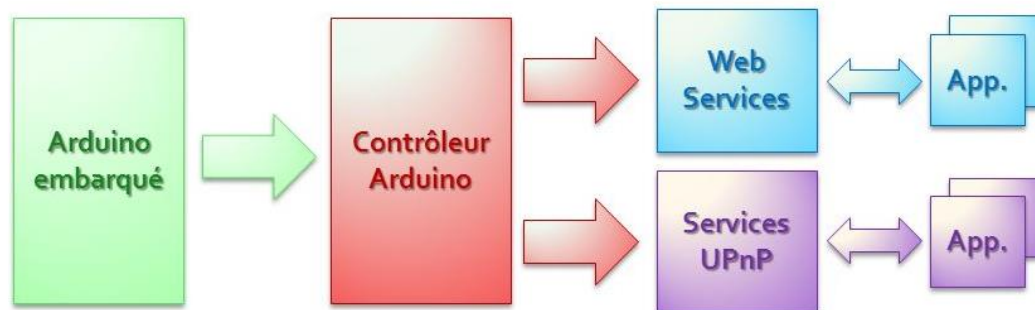
5. Livrables techniques attendus

Les livrables à mettre à disposition des tuteurs dans **votre dépôt Git** sont :

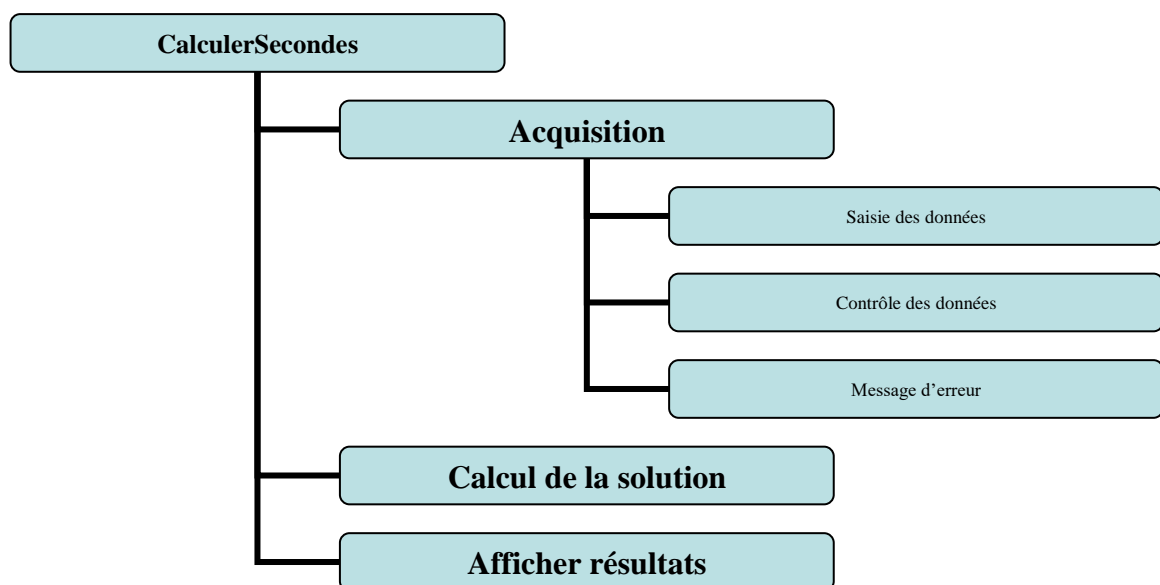
- Le fichier « Feuille d'avancement groupe »
- Les schémas électroniques réalisés (modules cardio et cœur de LEDs) avec un petit document (1 ou 2 pages max.) d'explications techniques.
- Une photo ou une petite vidéo de votre montage
- Le code réalisé (**C Arduino et C**) organisé par modules (M1, M2, M3 et M4). Le code doit être commenté.
- Une documentation technique **succincte** sur l'architecture de vos exécutables : structures utilisées, modularisation des fichiers, organisation de fonctions. Un schéma vaut mieux qu'un grand commentaire de 30 pages! Voir les exemples ci-dessous.



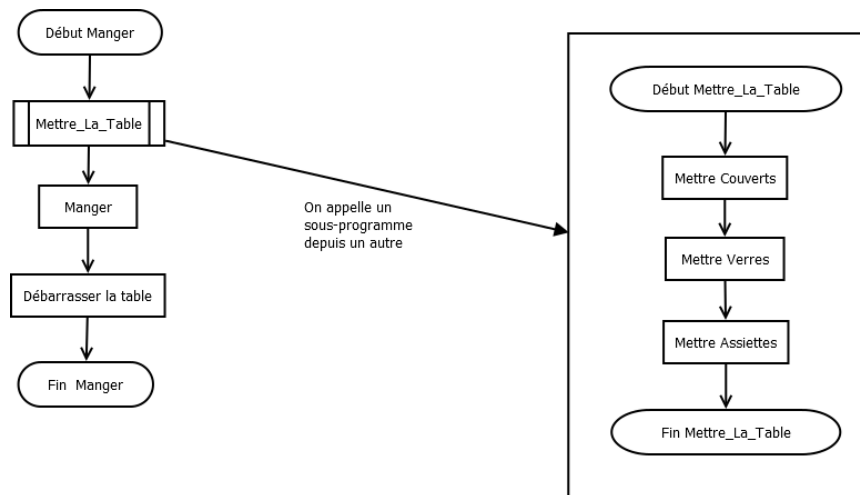
Exemple pour l'architecture logiciel (modularisation) :



Exemple pour la pile d'appels de fonctions :



Exemple d'algorithme:



Exemples de représentation de structures en C :

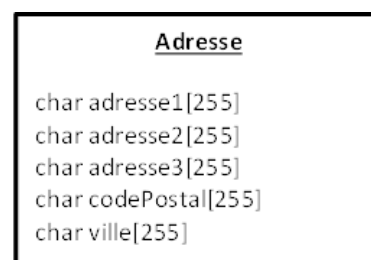
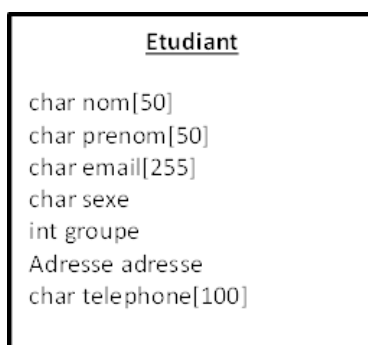
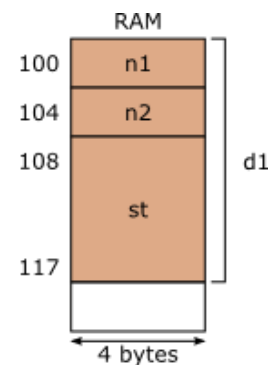
```

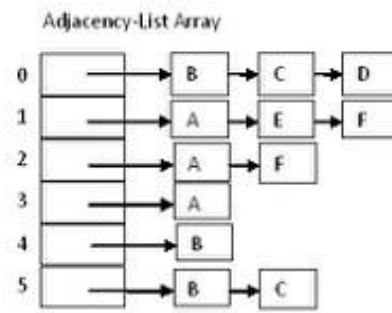
struct Etudiant {
    char nom[50];
    char prenom[50];
    char email[255];
    char sexe;
    int groupe;
    Adresse adresse;
    char telephone[100];
}

struct Adresse {
    char adresse1[255];
    char adresse2[255];
    char adresse3[255];
    char codePostal[255];
    char ville[255];
}
    
```

```

struct contact
{
    int n1;
    float n2;
    char st[10];
} d1;
    
```





Un autre exemple de structure dynamique

Représentation de la répartition en bibliothèques :

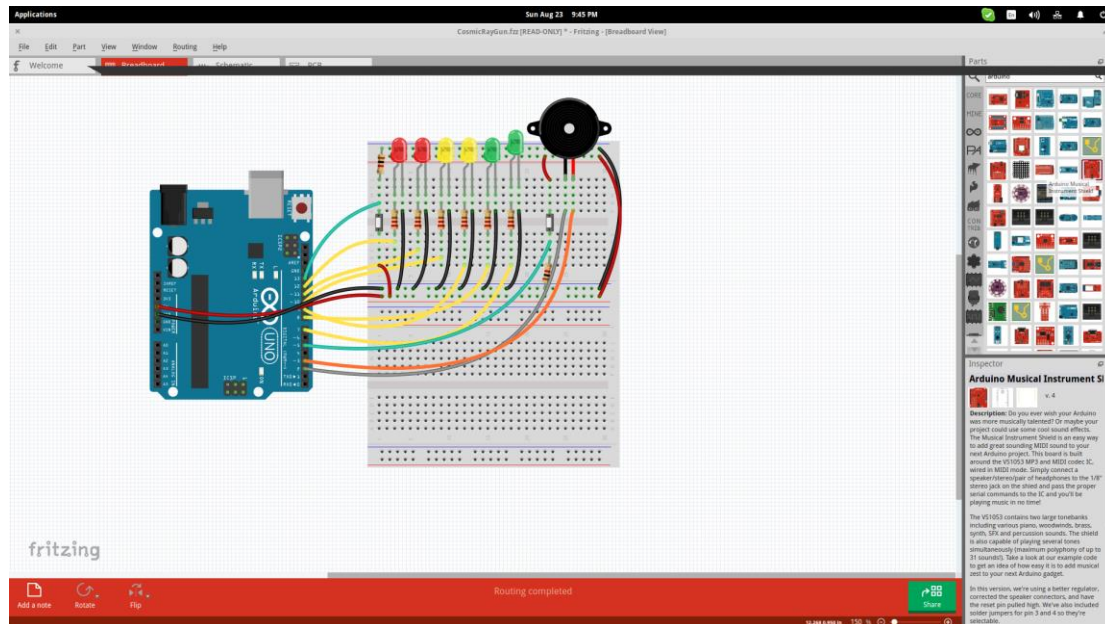
etudiant.h

```
typedef struct Etudiant Etudiant
void afficherEtudiant(Etudiant etudiant)
void saisirEtudiant(Etudiant *etudiant)
```

adresse.h

```
typedef struct Adresse Adresse
void afficherEtudiant(Adresse adresse)
void saisirEtudiant(Adresse *adresse)
```

Schémas Fritzing (vue Platine ou vue schématique)



6. Quelques conseils

Une bonne organisation de l'équipe est nécessaire dès le départ. Passez du temps à chercher une bonne organisation vous fera gagner du temps après.

Pensez à des choses simples et commencez par réaliser ce qui est demandé et pas ce qui n'est pas demandé ! **Concentrez-vous sur toutes les fonctionnalités obligatoires demandées** pour ne pas en oublier une importante.

N'attendez pas d'avoir les 4 modules développés pour faire des tests. Chaque module peut être testé de façon unitaire. Par exemple,

- Pour le module 3.3, n'attendez pas d'avoir le module cardio opérationnel, créez les données nécessaires à envoyer au port série dans le module 3.1
- Pour le module 3.4, n'attendez pas d'avoir le fichier produit par le module 3.3 : créez vous-même un fichier .csv.

Vous devez désormais être familiarisés avec les fonctions d'ouverture, lecture, écriture et fermeture de fichiers, notamment grâce au DM sur les automates. Ceci ne devrait plus être un obstacle à la réalisation de ce prototype. Pensez également à utiliser la fonction `sscanf` pour découper chaque ligne lue dans le fichier csv.

Même si les spécifications de ce projet vous permettent d'assigner des fonctionnalités distinctes à chaque personne du groupe, on vous recommande vivement d'assurer collectivement l'analyse du début de projet. Après l'analyse, vous pourrez vous dispatcher les tâches restantes.

Mettez-vous d'accord sur les structures à utiliser au début du projet car plusieurs personnes du groupe pourraient être amenées à les utiliser.

Vous devez montrer, lors de la soutenance :

- que les programmes répondent aux spécifications fonctionnelles et techniques.
- vos connaissances personnelles sur les objectifs pédagogiques du bloc.

Si vous implémentez des algorithmes de tri pour l'affichage des données, vous pouvez, dans un premier temps utiliser des algorithmes simples (bulle, insertion, sélection) et dans un deuxième temps, s'il vous reste du temps, implémenter un algorithme plus efficace (merge, quicksort). De même pour la recherche (faites d'abord une recherche séquentielle puis une recherche dichotomique). Mais, attention, pour que la recherche dichotomique soit efficace, la liste doit être préalablement triée.

Vous pouvez utiliser n'importe quelle structure de données présentée dans le module ou n'importe quelle autre que vous connaissez si vous êtes à l'aise avec la programmation en C. Dans tous les cas, réfléchissez bien à la plus adaptée à chaque problématique et sachez argumenter le pourquoi de votre choix.



ET SURTOUT... AMUSEZ-VOUS BIEN !!!

L'ELECTRONIQUE ET LA PROGRAMMATION DOIVENT ETRE UN PLAISIR !!!

7. Organisation du projet

Le projet se déroule en groupes de **3 ou 4 personnes** (en fonction de la taille des promos) du 10 (fin d'après-midi) au 20 novembre 2017.

- Le lundi 13 novembre, en fin de journée, chaque groupe doit rendre le document décrivant l'analyse du projet (cf. document "Feuille d'avancement"). Pour cette phase, vous pouvez être accompagnés par un **étudiant de cycle ingénieur (A3 ou A5)** qui sera votre référent pour le projet, **en fonction de leur disponibilité dans le centre.**
- Les livrables attendus sont à rendre le vendredi 17 novembre avant 13h30 en version électronique sur le dépôt Git ou par mail (voir l'organisation avec votre tuteur responsable du projet).
- L'après-midi du 17 novembre sera consacré à la préparation de la soutenance avec la réalisation d'un support (ppt, prezi, autre ...)
- Dans la journée du 20 novembre, chaque groupe devra présenter et défendre son travail au cours d'une soutenance orale avec démonstration complète de l'application entre 15-20 minutes puis séance de questions individuelles d'une même durée. Chaque membre du groupe devra prendre la parole de manière équitable. Le chef de projet présentera l'organisation de l'équipe et la distribution des tâches pendant la semaine.

Pour toute question sur le projet, n'hésitez pas de demander à votre tuteur.



Remerciements à Sébastien RAMON et Hugues DELALIN (Arras), Ludovic BARRANDON et Nicolas ANTINI (Toulouse), Baptiste DENAEYER (Lyon) pour leur contribution à ce projet.

