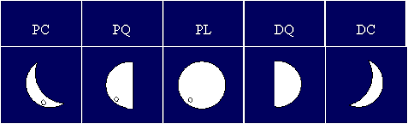
**Compte rendu**

**Séance n°1 14/12/2018**

Aujourd’hui j’ai passé beaucoup de temps à lire la documentation, notamment les explications sur les LED car le document était en anglais. J’ai testé les programmes Arduino envoyé, notamment celui appelé matrix. *Voir vidéo 1*

Il y avait beaucoup de nouvelles bibliothèques à intégrer.

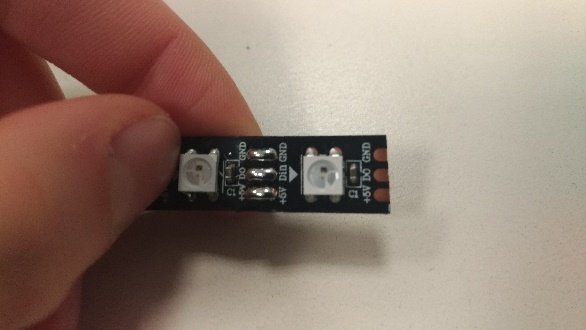
Après j’ai regardé comment coder la partie qui affiche les cadrans lunaires en fonction de la date. Pour exercer un cycle complet : c’est-à-dire pour passer de pleine lune à pleine lune cela met 30 jours.

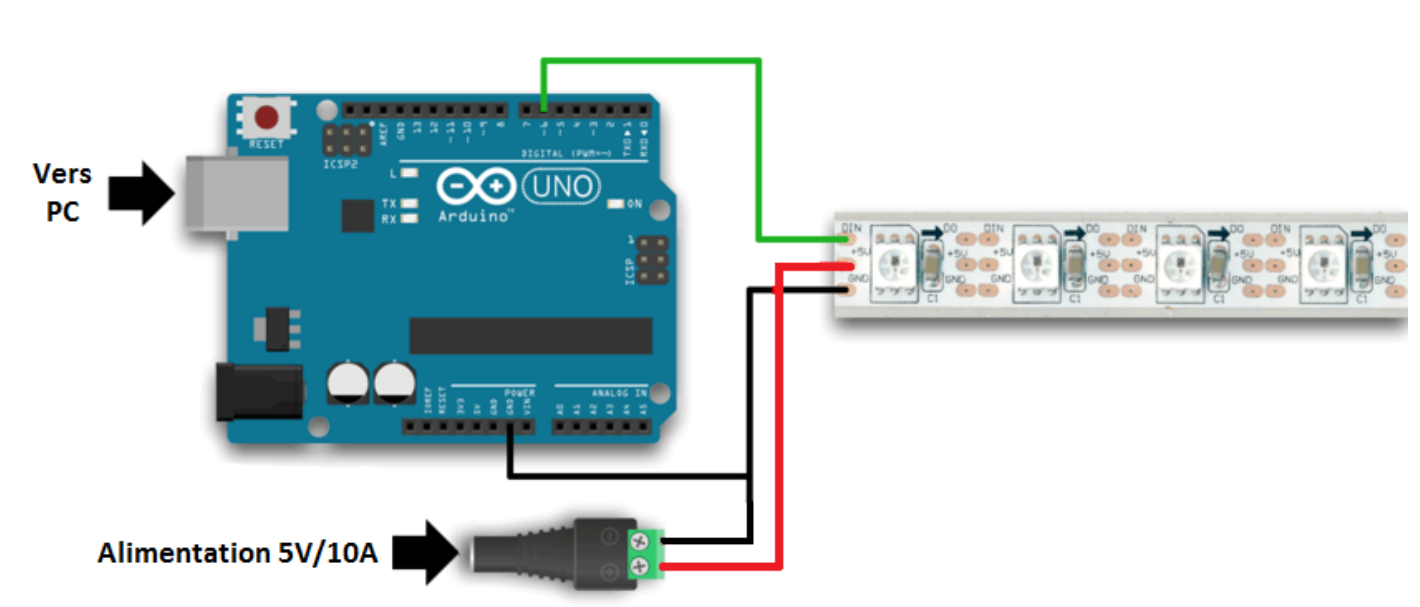


J’ai trouvé que le fichier appelé DemoReel100 serait bien pour réveiller les gens avec ses lumières vives. *Voir vidéo 2*.

Maintenant il faut que j’ai un simple programme qui laissent les lumières allumées de couleur blanches, mais nous souhaiterions que ces lumières soient télécommandées par le Bluetooth. Je pense qu’on pourrait réaliser un réveil qui pourrait être soit avec des lumières vives ou alors comme un soleil qui se lève.

<http://www.fablabredon.org/wordpress/2017/12/17/lumiere-sur-larduino-avec-de-la-couleur-et-des-led/>





J’ai remarqué que ce n’était pas exactement les mêmes diodes, le GND et le reste sont inversés, de plus je n’ai pas besoin de brancher ma bande de LED sur une alimentation autre que l’Arduino car je n’ai que 10 LED.

Je suis en train de rechercher comment contrôler la couleur de toutes mes LED grâce au Bluetooth. Une des difficultés est de trouver la bonne application car je n’ai pas l’impression que l’application actuelle « Bluetooth Electronics » soit la bonne.

<http://www.e-learning-avenue.com/Systemes/Telechargements/Arduino%20Uno/Bluetooth_rgb.pdf>

**Séance n°2 18/12/2018**

J’ai réussi à ouvrir le fichier envoyé par M. Masson, grâce à ce programme je peux choisir manuellement la couleur des LED. Je peux aussi choisir la luminosité. Je reste sur l’idée de changer la couleur grâce au Bluetooth.

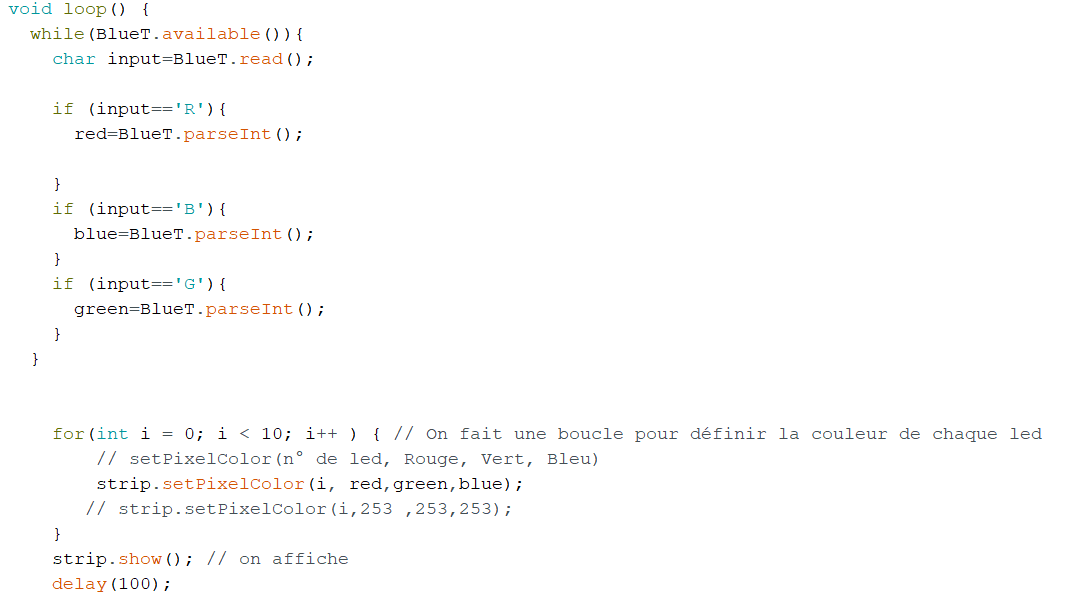
Chaque couleur et chaque degré de luminosité est associé à un degré nombre. Quand je fixe 223 pour chaque couleur j’ai une lumière blanche.

Maintenant je cherche un moyen de régler la luminosité, car le code d’origine ne me permet pas de le faire. J’ai essayé de faire

Mais cela n’a pas fonctionné, un seule diode s’allume et la luminosité était très forte.

Je viens de comprendre qu’il me suffit de baisser les paramètres ensembles pour que l’intensité baisse.

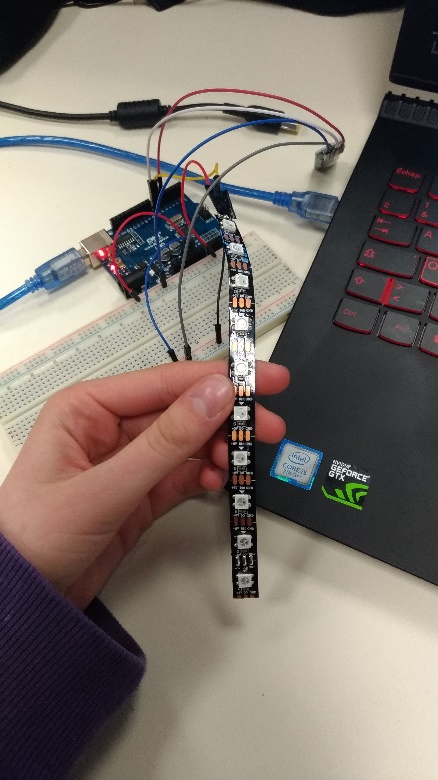
Je cherche toujours maintenant à changer la couleur de la LED.

Sur Bluetooth Electronics (l’application), il y a déjà une feuille qui est préréglée avec 3 jauges correspondant aux couleurs Rouge, Bleu et Vert. (*mettre photo de la capture d’écran*) J’ai donc écris le programme permettant de recevoir une valeur comprise entre 0 et 223 qui correspondra à l’indice de la couleur. Il y a des difficultés car je ne savais pas comment lire les chiffres qui suivaient l’envoi de la lettre pour indiquer que c’est du rouge. Par exemple pour la jauge rouge, l’application envoie un R puis une valeur. On lit le R, puis on réalise un parseInt() de ce que l’on a envoyé et on lit les chiffres suivants la lettre.

Au début cela ne fonctionnait pas car j’avais oublié d’écrire BlueT.begin(9600), qui permet de communiquer.

La semaine prochaine j’ai prévu de commencer à coder le réveil.





**Préparation séance n°3**

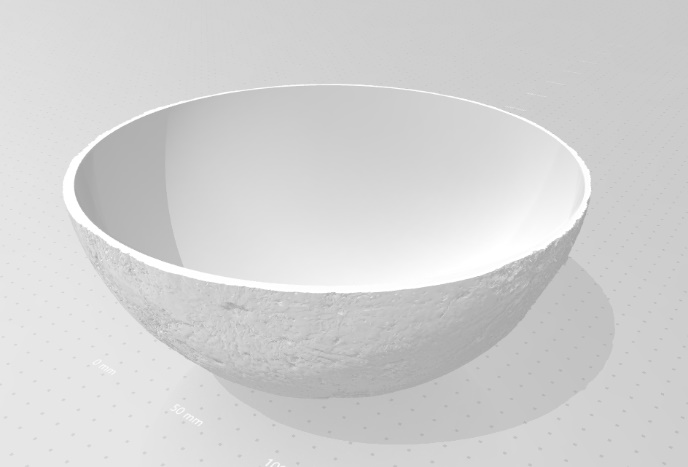
Avant la troisième séance, nous avons avec Sarah chercher le fichier nous permettant de réaliser la Lune. J’ai cherché un logiciel qui nous permettrait de créer 2 de demi-lunes. J’avais le choix entre plusieurs fichiers : celui que nous avons choisi est celui où l’intérieur de la Lune est lisse. Il faut connaitre les dimensions de cette Lune car nous devons aussi créer la plaque qui tournera à l’intérieur, cette plaque nous permettra de visualiser les différents quartiers de la Lune.

Figure 1 Demi-Lune avec trop de détails

Le moteur que nous a donné M Masson est compatible avec une certaine tige que nous n’avons pas, nous nous demandions si cette tige existait ou si nous allions devoir la fabriquer.

Le problème maintenant c’est qu’il faut trouver un moyen de gérer l’épaisseur de la demi-lune car cela doit laisser passer la lumière des diodes.

**Séance n°3 11/01/19**

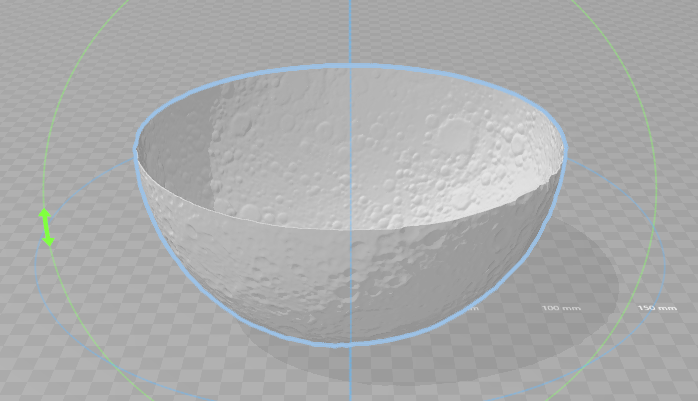
Durant cette séance j’ai réussi à modifier l’épaisseur de la demi-Lune mais cela faisait un rendu très bizarre qui n’était pas adapté. Je suis donc restée avec les deux demi lunes de départ.

Figure 2 Demi-Lune version 2 creuse

Je suis passée au FabLab pour imprimer les deux demi lunes car la version que j’avais envoyé à Sarah contenait des détails trop petits pour l’imprimante 3D. De plus un autre problème c’est posé une fois au FabLab, quand j’ai utilisé une autre version de la Lune, et que je l’ai séparé en deux demi-Lunes, les parties à imprimer étaient pleines au lieu d’être creuses.

J’ai donc téléchargé Meshmixer qui m’a permis de rendre la Lune creuse et de l’exporter au format STL (qui permet d’imprimer en 3D).

D’après le monsieur du FabLab, je devrai pouvoir récupérer la demi-sphère mardi. Pendant ce temps Sarah a imprimer la planche tournante.

Une image contenant table, intérieur, tasse, café

Description générée avec un niveau de confiance très élevé**Préparation de la séance n°4**

Une image contenant mur, intérieur, personne

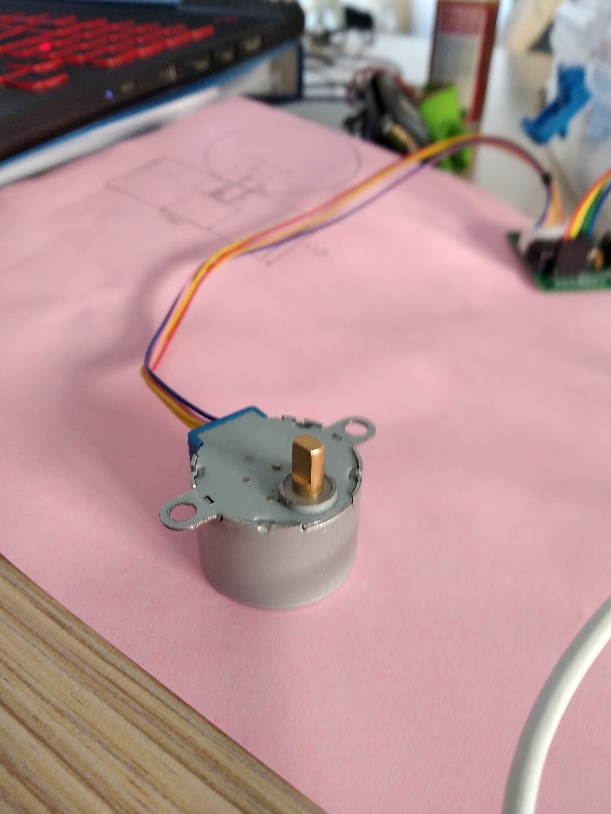
Description générée avec un niveau de confiance très élevéLa personne qui s’occupait du FabLab ne savait pas trop comment faire, la première « version » de demi-Lune que je lui ai donné était pleine de plus, l’imprimante ne pouvait pas l’imprimer car cette Lune avait un angle de courbure trop important et donc l’imprimante aller imprimer dans le vide. Le logiciel Meshmixer et un tutoriel trouvé sur Internet m’a permi de régler le problème de la Lune pleine. Pour imprimer la demi-lune, l’imprimante a créer des supports afin d’imprimer le haut de la demi-lune.

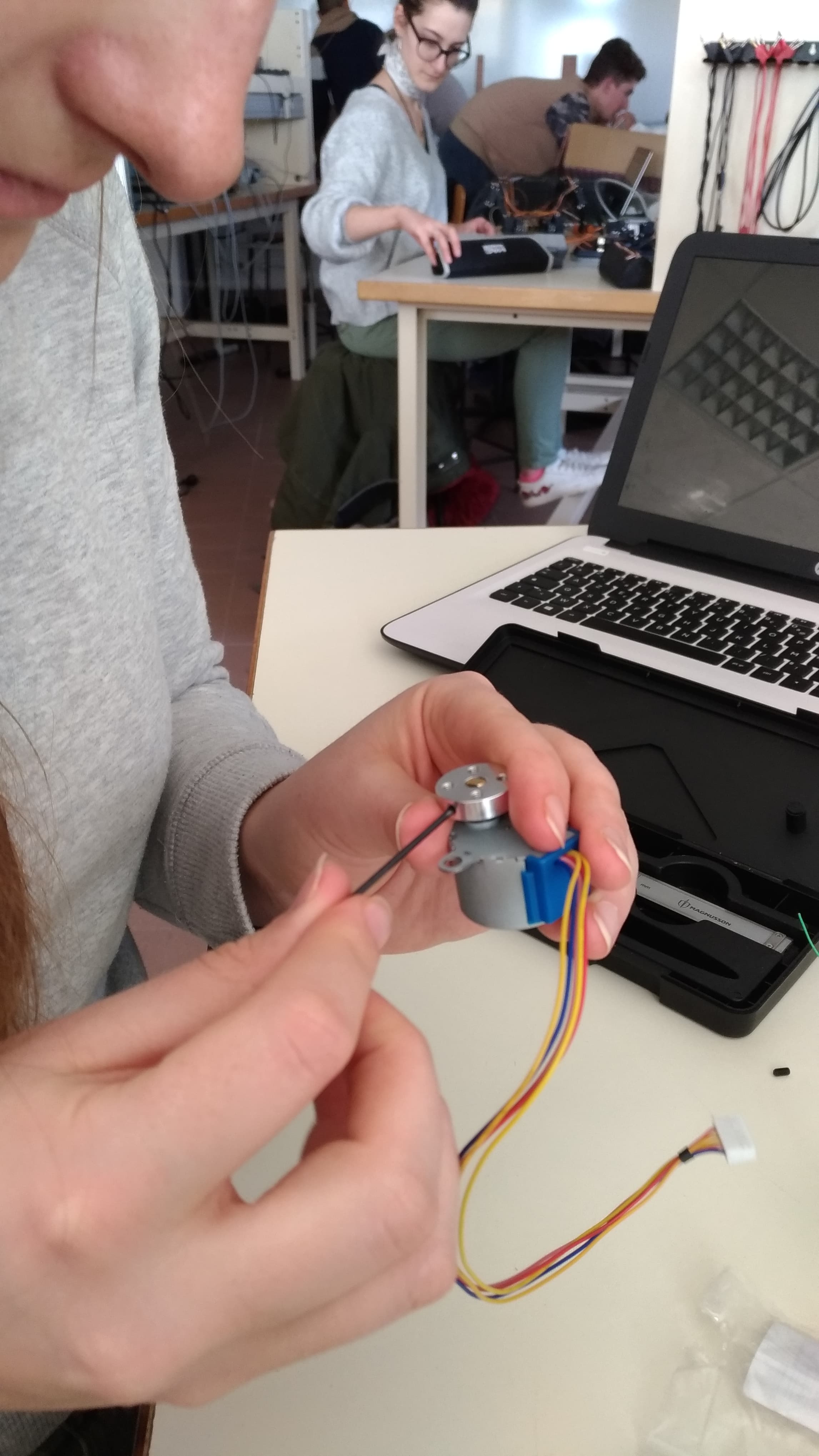
<https://formlabs.com/blog/how-to-hollow-out-3d-models/>

Après avoir vidé j’ai séparé la Lune en deux grâce à 3D Builder.

Une fois la Lune obtenu nous avons testé si l’épaisseur laissait bien passer la lumière. C’est une réussite.

Une image contenant intérieur, plafond, mur, cuisine

Description générée avec un niveau de confiance très élevé**Séance n°4 18/01/19**

M Masson nous a donné la tige de 5mm en fer qui nous servira pour faire tourner la planche. Nous avions besoin de l’emboiter sur le moteur mais les deux embouts n’étaient pas compatibles. Nous avons pris une chose en fer (voir image 1).

*Embouts pas compatibles*

*Image 1 : petit truc en fer*

Une image contenant personne

Description générée avec un niveau de confiance très élevéLa tige que nous a donné M Masson était trop grande, nous l’avons coupé à la scie. Elle mesure maintenant environ 15 cm, nous avons gardé un peu de marge pour la retailler plus tard.

Après nous nous avons fixé le tout ensemble pour pouvoir faire tourner la tige.

Nous nous sommes mises d’accord sur les nouvelles dimensions de la planches, puis de mon côté j’ai commencé à coder « l’horloge qui tourne ».

J’ai commencé à chercher une librairie qui me donnerait l’heure, j’ai téléchargé celle-ci :

<https://github.com/adafruit/RTClib/blob/master/examples/datecalc/datecalc.ino>

Il faut que j’ai l’heure car la position de la Lune dépendra de l’heure donc j’ai aussi pensé que je pouvais régler l’heure avec le Bluetooth, grâce à 3 jauges différentes, une allant de 0 à 24 et l’autre de 0 à 59. Après il faudra rajouter une fonction « minuterie » qui permettrait de faire tourner la tige en fonction du temps.

Il sera plus simple pour moi d’avoir une fonction ou une librairie ou un module qui soit une horloge, après je pourrais toujours régler l’heure avec le Bluetooth.

**Séance n°5 13/02/2019**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Aujourd'hui nous comptions faire tourner la Lune mais ce ne fut pas le cas.

Pour commencer il a fallu fixer la tige entre deux planches de bois pour qu’elle ne bouge plus. Avec Sarah nous sommes allées dans l’atelier, pendant que Sarah tenait les 3 planches, je réalisais 4 trous qui allaient permettre de fixer la tige afin qu’elle ne bouge plus.

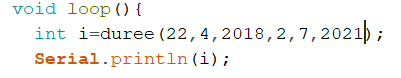
Après nous avions quelques problèmes avec le servomoteur car celui-ci se bloquait ou ne tournait pas correctement. Sarah a réglé le problème en fixant bien les fils qui gênaient la rotation.

Pendant ce temps j’ai continué la partie code, il fallait que je réalise une fonction “durée” qui donnait le nombre de jours entre deux dates et ainsi en faisant une simple division, la position de la plaque pour former les quartiers lunaires. Je n’ai pas réussi à trouver une librairie ou une fonction arduino qui m’aurait permit de réaliser cela, il a donc fallu que je la crée.

Il faut savoir que comme le logiciel Arduino ne permettait de lire clairement le code, je l’ai réalisé sur un autre éditeur de texte, j’ai prévu de l’écrire en java pour pouvoir le tester rapidement

**Séance n°6 27/02/2019**

La séance précédente j’étais en train de créer la fonction « durée » qui permet de calculer la durée entre deux dates. Aujourd’hui je viens de la terminer et j’ai vérifié que sur Internet, que la date donnée était vraie.



*Nb de jours renvoyé*

*Calcule la durée entre le 22/04/2018 et le 02/07/2021*

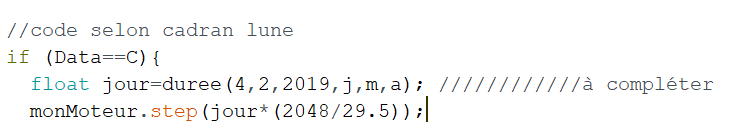


Les problèmes que j’ai eu à résoudre concernaient la manière d’écrire le code, parfois je n’avais pas mis 2 « = » :

Exemple : if (m1=m2) que j’ai corrigé par if(m1==m2).

Ceux sont des erreurs basiques, qui n’étaient pas très longues à résoudre. Après il y avait celles qui correspondaient à l’algorithme, à la méthode de calcul. Je mets sur le GitHub la fonction duree.

Puis nous avons réfléchis à comment nous allions utiliser le capteur infrarouge, car nous ne savions pas comment nous pouvions le fixer de manière à ce que le capteur détecte bien le « noir ». A la fin nous avons décidé de fixer sur la base du servomoteur (qui tourne en même que la planche), un papier avec une gommette noire et en dessous le capteur infrarouge.

On a repris le code maintenant on peut calculer la position que la planche doit atteindre en fonction de la date. 

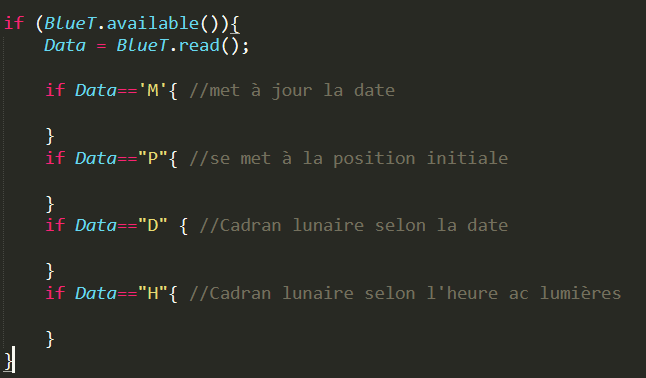
<https://fr.wikipedia.org/wiki/Mois_lunaire>

29,5 période lunaire entre deux nouvelles lunes

<https://www.calendrier-365.fr/lune/calendrier-lunaire.html>

On fixe le 6/03 (noir) comme étant la phase de base

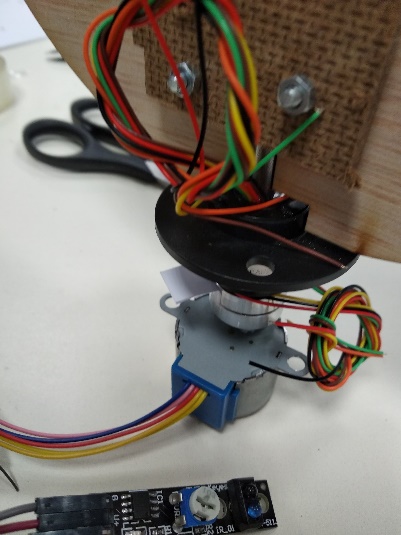
**Séance n°7 du 06/03**



Aujourd’hui j’ai passé la séance à mettre les morceaux de programmes déjà réalisé ensembles. J’ai fait plusieurs essais et j’ai dû demander de l’aide à mes camarades de classe. Lors des précédentes j’avais réalisé les fonctions qui permettaient de calculer le nombre de jours dans un mois (selon l’année, bissextile ou non) et la durée entre deux dates.

Ces fonctions me permettent aujourd’hui de réaliser la suite du programme. Notre horloge Lune a 4 fonctionnalités :

* Aller à la position initiale, quelle que soit l’angle de rotation, cela permettra de ne pas recalculer à chaque fois
* Permet de donner l’heure, par exemple si tout est noir, il est minuit sinon il est midi
* Permet de donner le cadran lunaire
* Permet de mettre un mode Disco où les lumières clignotent dans tous les sens

L’un des problèmes que j’avais était que je ne savais pas où stocker les variables car avec le loop, elles se réinitialisent tout le temps.

*Fixation du papier qui permettra de se rendre à la position initiale grâce aux degrés entre l’endroit où le papier est détecté et la position initiale*

**Séance 8 13/03**

Nous avons finalisé le montage, nous avons réussi à faire tourner la Lune correctement. Cependant le moteur devait être cassé car parfois il ne tournait plus du tout, au dernier moment nous avons donc dû prendre un nouveau moteur.

Au moment du passage à l’oral nous avons eu plusieurs problèmes : faux – contact dans les fils, le capteur infrarouge qui ne fonctionne pas et les lumières qui ne s’allument plus.