

MASANTE Anna
CHARBONNIER Sarah

PROJET D'ARDUINO

L'horloge lunaire



SOMMAIRE

INTRODUCTION

Présentation et problématique

Cahier des charges

Planning théorique

Planning réel

DÉVELOPPEMENT DU PROJET

Matériel

Algorithme

Schéma du montage

Difficultés et solutions

Electronique

Informatique

Assemblage

CONCLUSION

Objectifs atteints ?

Améliorations possibles ?

Remerciements

INTRODUCTION

Avec ce projet, nous souhaitons réaliser un objet utile, esthétique et original. Au cours de notre recherche d'idée, un projet semblable était proposé, mais celui-ci n'utilisait pas de plaque tournante, la lune avait été découpée en différents quartiers qui s'allumaient selon la date. Nous avons donc complexifié un peu le projet, en décidant que ce serait une planche qui tournerait à l'intérieur d'une lune et grâce à cette lune, nous pourrions connaître le cadran lunaire du jour mais aussi l'heure. Pour réaliser toutes ces actions nous devons être capable de remettre à la position initiale cette plaque.

Nous voulions tout d'abord créer un objet comportant trois modes :

- Le mode Date qui lui fait tourner la plaque de façon à afficher le cadran du jour,
- Le mode Heure qui fait tourner la plaque jusqu'à l'heure souhaitée,
- Le mode Disco qui affiche des lumières de toutes les couleurs qui pourrait être utilisé par-dessus les deux autres modes.

Finalement nous avons fait quatre différents modes :

- Le mode Obstacle, qui permet de choisir la position de la plaque pour remettre la lune à l'heure,
- Le mode Date,
- Le mode Heure,
- Le mode Disco.

Le mode Obstacle n'était pas prévu mais quand nous avons compris qu'il était difficile de faire revenir la plaque toute seule à sa position initiale, nous avons décidé de configurer ce mode afin d'établir la position initiale nous-même.

Notre principale problématique était de commander chacun de ces modes à l'aide d'un module Bluetooth.

Nous devons réaliser ce projet en huit séances.

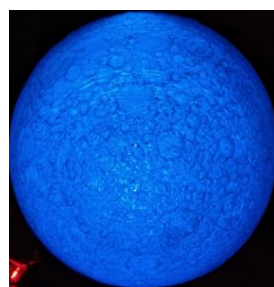
Voici des photos des quatre modes :



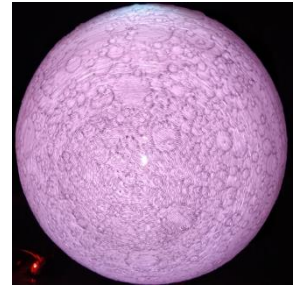
Mode "P"
obstacle



Mode "D"
date



Mode "H"
heure



Mode "R"
disco

Cahier des charges

Au début de notre projet nous avons mis en place un planning afin de savoir toutes les actions que nous devons mener.

Voici notre planning théorique :

Séance	1	2	3	4	5	6	7	8
ANNA	Code lumière + Bluetooth	Code lumière	Lune 3D	Choix par Bluetooth	Code lumière	Code cadran	Choix par Bluetooth	Montage + vérifications
SARAH	Code Moteur	Code Moteur	Plaque + tige	Plaque + tige	Position Plaque	Code heure	Vitesse Moteur	Montage + vérifications

Voici celui que nous avons suivi :


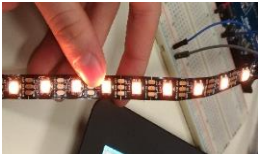
Séance	1	2	3	4	5	6	7	8
ANNA	Code lumière	Lumière + Bluetooth	Lune 3D	Montage + Code cadran	Code cadran	Code cadran	Code complet	Code complet
SARAH	Code Moteur	Code Moteur	Plaque tournante	Montage + Plaque/tige	Code horloge	Montage + Code heure	Code détecteur	Détecteur + Montage

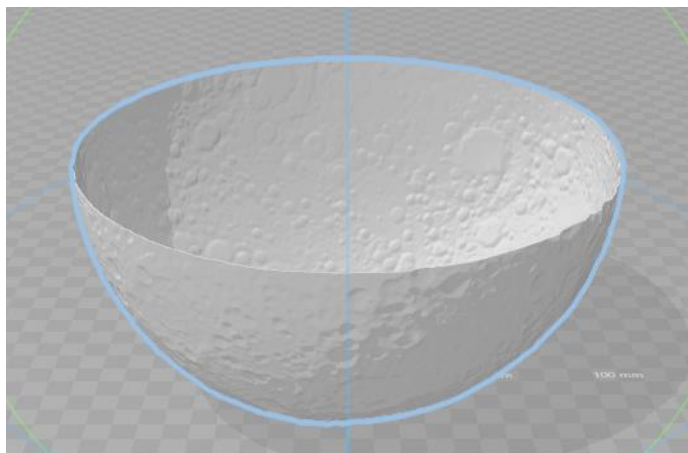
Ci-dessous les diagrammes de Gantt correspondant à ces programmes :



Nous pouvons voir que le programme initial n'a pas totalement été respecté, en effet, nous avons sous-estimé la partie consacrée au codage du cadran lunaire. Contrairement à ce qui était initialement prévu, nous nous sommes toutes les deux occupées de la partie montage et du code de l'horloge (correspond au mode D).

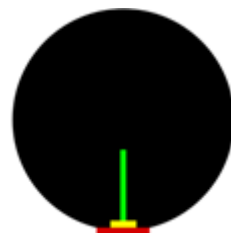
Développement de chacune des séances :

	Anna	Sarah
Séance 1	<p>Lors de la première séance, 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 des diodes, il y avait beaucoup de nouvelles bibliothèques à intégrer. Le montage du bandeau de LED a été appris et compris.</p> <p>Je me suis intéressée à la durée d'un cycle lunaire et j'ai aussi recherché l'application Bluetooth que nous allons utiliser pour réaliser ce projet.</p>	<p>Il a tout d'abord fallu établir le programme des séances que l'on devait théoriquement suivre. Ensuite je devais étudier le fonctionnement du moteur qui allait servir à faire tourner la plaque dans le moteur. Ce moteur est un moteur pas à pas.</p> 
Séance 2	<p>Durant cette séance je me suis occupée de la partie code, j'ai essayé de comprendre comment grâce au Bluetooth je pouvais envoyer des instructions à la carte Arduino. Cela permettrait de choisir la couleur des diodes et de comprendre toutes les commandes liées au Bluetooth, elles seront utiles dans la réalisation du projet.</p> 	<p>L'objectif étant de faire tourner le moteur pour placer la plaque à un endroit voulu. J'ai cherché un code sur internet cependant celui-ci présentait certains problèmes comme celui où le moteur ne tournait pas mais vibrait. Il était donc préférable de récupérer le code sur la page du cours. Celui-ci permet de faire tourner le moteur de 2048 pas c'est à dire un tour complet, il suffisait juste de le modifier pour qu'il tourne du nombre de pas voulu.</p>
Séance 3	<p>Nous avons recherché le fichier de la Lune, il a fallu trouver un logiciel capable de créer de deux demi-lunes à partir d'une lune entière. Lors de cette étape nous avons rencontré quelques problèmes (voir <i>Difficultés rencontrées</i>). Je me suis occupée de découper la lune, du choix de l'épaisseur et de sa taille. Quelques jours plus tard nous avons récupéré les deux demi-lunes.</p>	<p>La plaque, supposée tourner dans la lune, a été imprimée au fablab à l'aide du logiciel Inkscape. Il fallait prendre les dimensions de la lune qu'Anna allait imprimer et réduire le diamètre de 2-3mm afin que la plaque tourne sans poser de problème avec la forme non lisse de la lune.</p>



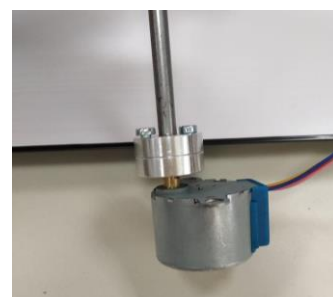
Séance 4 J'ai aussi réfléchi aux différentes options pour régler l'heure grâce au Bluetooth, à une fonction, une librairie ou un module.
J'ai commencé à coder la fonction durée qui calcule le nombre de jours entre deux dates.

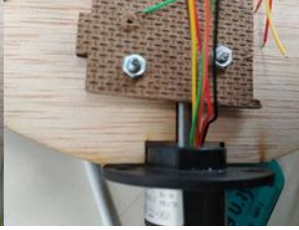
Lors de son impression, la demi-lune a été imprimée d'1 cm plus petit que prévu, c'est à dire 19 cm au lieu de 20 cm de diamètre car l'imprimante ne permettait pas plus grand. Il a donc été nécessaire de retailler une nouvelle plaque, qui rentre dans cette nouvelle lune. Nous en avons profité pour faire un trou à l'emplacement de la tige dans la plaque tournante. Nous avons finalement taillé la pour qu'elle ait une longueur adéquate.



Séance 5 Les deux lunes n'ont pas exactement la même taille, il faudra retailler la planche. On a reçu un module Tiny RTC qui permet de récupérer l'heure et, à terme, de réaliser le mode H. J'ai pu commencer le codage des différents modes.





Nous avons, pendant cette séance pu fixer la tige au moteur.
Il a fallu commencer le code de l'horloge, c'est à dire à quelle position doit se trouver la plaque selon l'heure de la journée.



Séance 6	<p>J'ai finalisé la fonction durée qui fonctionne, j'ai travaillé à mettre tous les éléments de code ensemble. Nous nous sommes demandé comment fixer le capteur infrarouge qui permettra de détecter l'obstacle, ce qui nous sera utile dans le mode P. On a fixé la période lunaire comme étant égale à 29, et le 6 mars comme le début d'un cycle où la lune est noire (nouvelle lune).</p> <pre>void loop() { int i=duree(22,4,2018,2,7,2021); Serial.println(i); 1167</pre>	<p>Pour le montage de la lune au moteur il ne manquait plus que de fixer la plaque tournante à la tige, elle-même reliée au moteur. Il fallait régler certains problèmes notamment celui de la gêne occasionnée par les fils des LED lorsque la plaque est en mouvement.</p> 
Séances 7-8	<p>Durant ces séances, j'ai réglé toutes les erreurs, toutes les parties du code s'exécutent sans erreur. On peut changer de mode quand on le souhaite. Durant ces deux séances on a beaucoup réalisé de tests car nous rencontrions, là-encore, différents problèmes.</p>	<p>Pendant la séance 7, il fallait initialiser la position 'initiale' de la plaque tournante, c'est-à-dire, qu'à chaque changement de mode, elle se replace toute seule à cette position. Cependant, lors des tests avec le détecteur infrarouge (Keyes-01), nous avons trouvé préférable de placer l'obstacle devant le détecteur afin que nous puissions nous-même choisir cette position. Nous avons donc rajouté le mode 'P' afin que l'utilisateur replace la lune.</p>

DEVELOPPEMENT DU PROJET

Matériel

 <p>Moteur pas à pas fourni avec un circuit UNL2003</p>	 <p>Module Tiny RTC</p>
 <p>HC-06 Module Bluetooth</p>	 <p>Détecteur Infrarouge Keyes-01</p>

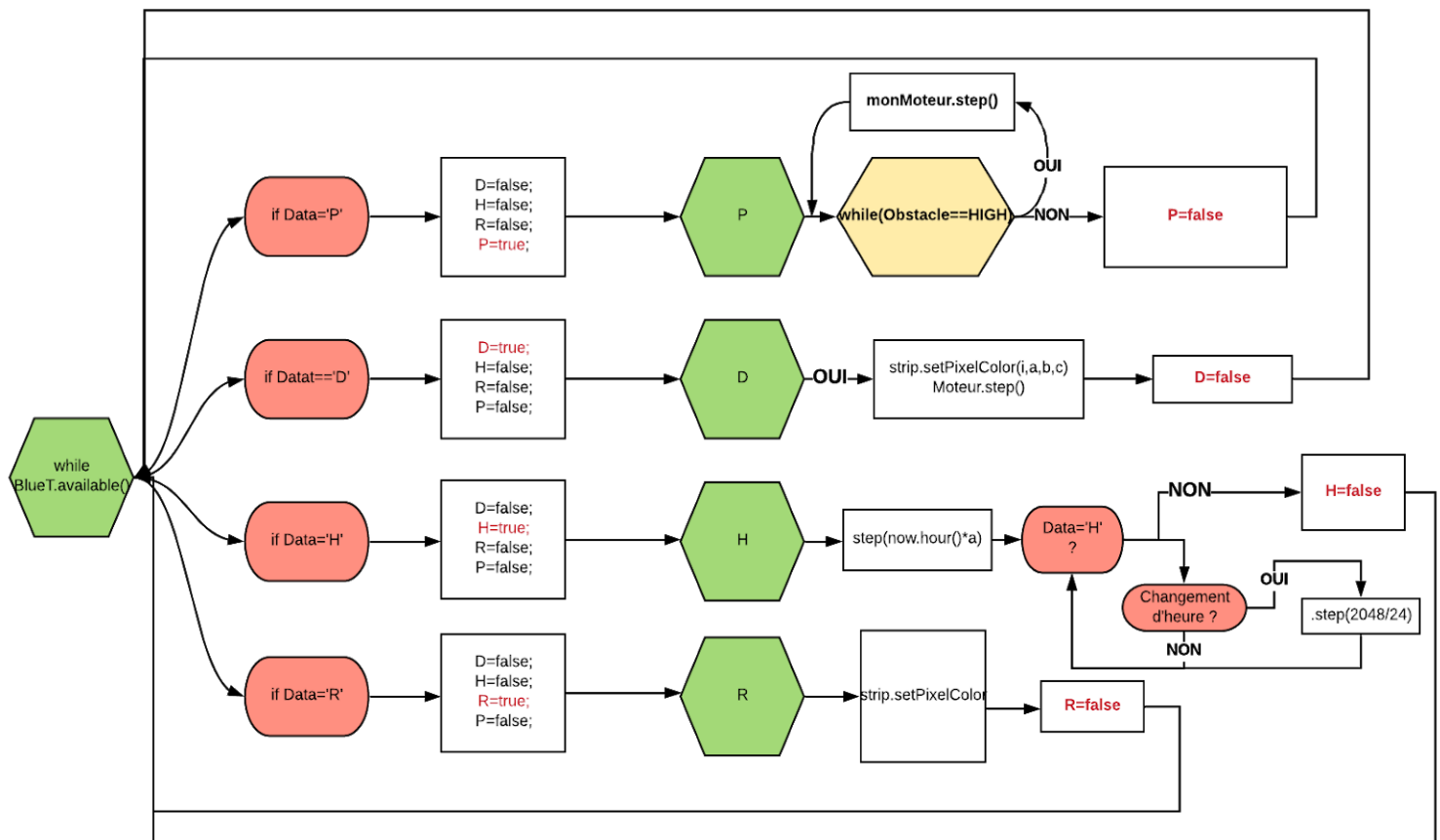
Le moteur pas à pas permet de faire tourner la plaque et la lune ensemble afin de positionner la lune selon la position demandée. Ce moteur tourne de 2048 pas pour faire un tour.

Le module Tiny RTC d'horloge en temps réel permet de récupérer la date et l'heure en temps réel. On récupère la date du jour pour déplacer la lune selon le quartier allumé du jour et l'heure afin de l'afficher à l'aide de la lune.

Le module Bluetooth permet de communiquer avec la carte Arduino depuis le téléphone pour choisir le mode voulu.

Le détecteur infrarouge permettait initialement de définir la position initiale de la lune mais il permet maintenant, en mode 'P', de positionner la lune à un endroit souhaité. La lune va tourner sur elle-même tant que le détecteur ne détectera pas d'obstacle.

Algorithme



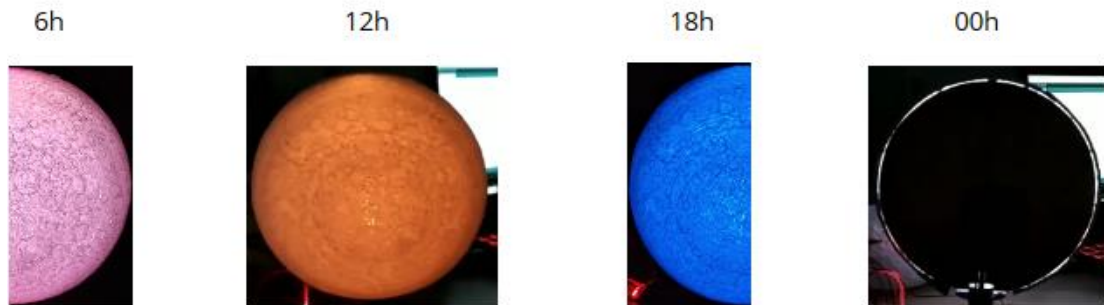
Le Bluetooth nous permet de choisir le mode que l'on souhaite. Si j'envoie 'P' par exemple, toutes les autres variables se mettent à **FALSE**, cette action nous permet de réaliser un seul mode. Quand nous avons réalisé l'action correspondant à ce mode, la variable se met à **FALSE**, si nous voulons continuer avec un mode différent, nous n'avons qu'à cliquer sur la lettre correspondante sur le téléphone.

Dans le cas du mode 'P' qui détecte les obstacles : tant que le capteur ne détecte pas d'obstacle, le moteur tourne, dans le cas contraire le moteur arrête de tourner et la variable se met à **FALSE**.

Pour le mode 'D' (date), les lumières s'allument (en blanc), et le moteur tourne jusqu'à la position calculée. Par exemple, nous avons fixé le 6 mars comme étant un jour de nouvelle lune (lune noire) quand nous entrons la date du jour, l'Arduino calcule le nombre de jours entre les deux dates et ainsi on peut retrouver le cadran lunaire du jour. On va calculer le nombre de jours écoulés entre une nouvelle lune (fixée par exemple au 6 mars 2019), et le jour même pour faire tourner de $(\text{nombre de jours}) * (1 \text{ tour} / \text{nombre de jours dans un cycle lunaire}) = \text{jours} * (2048/29)$. Quand la planche est bien positionnée, le moteur s'arrête et la variable D=**FALSE**.

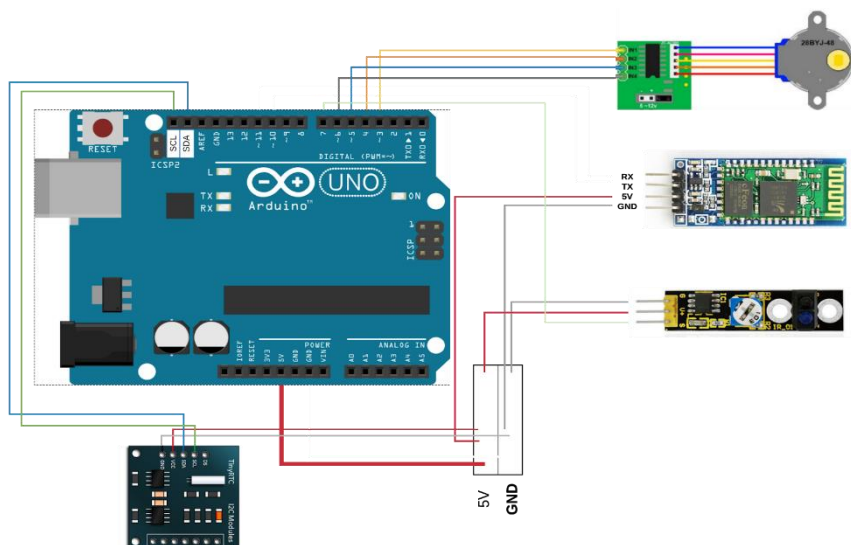
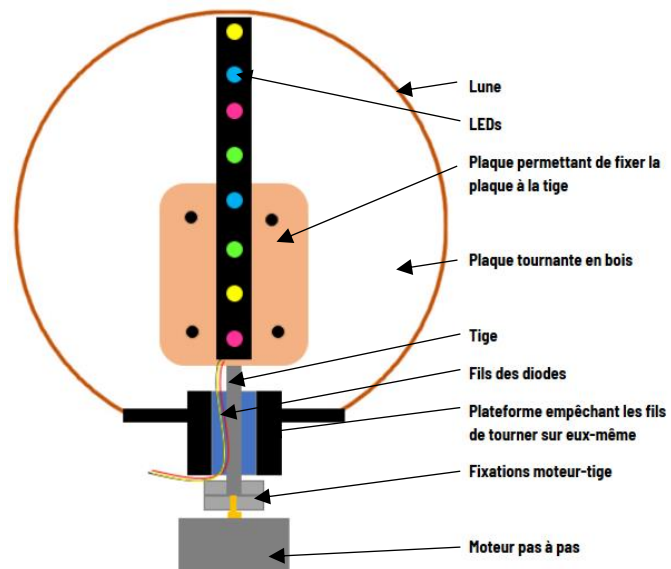
De la même façon pour le mode 'H' (heure), à minuit la lune face à nous sera noire, à 6h et à 18h, elle sera allumée de moitié, à midi, elle sera complètement allumée. La lune tourne de $(\text{nombre d'heures}) * (1 \text{ tour} / \text{nombre d'heure dans une journée}) = \text{heures} * (2048/24)$. Ce mode

détecte aussi s'il y a un changement d'heure, dans ce cas, la plaque tourne de $2048/24$. Tant que l'on n'envoie pas de lettre différente de H par le Bluetooth, le programme continue, dans le cas contraire la variable H=FALSE. Dans ce mode heure, nous avons voulu faire un dégradé de couleur selon les heures de la journée comme montré ci-dessous :



Le dernier mode 'R' (mode Disco) permet lui d'afficher des lumières aléatoires tant que l'on n'envoie pas de valeurs différentes de R par le Bluetooth.

**Schéma du montage
Moteur/Lune**



**Schéma du montage
électronique**

DIFFICULTES RENCONTREES

Electronique

Pour la partie électronique, nous avons rencontré certains problèmes tels que les fils des LEDs qui se détachaient trop facilement et le moteur qui avait tendance à vibrer sans tourner. Nous avons alors changé le moteur et cela a fonctionné. Lorsque l'on déplaçait la lune, il y avait aussi de nombreux faux-contacts entre les fils notamment avec le module infrarouge qui ne captait plus correctement les obstacles à la bonne distance.

Informatique

Lors de la partie codage nous avons eu beaucoup de problèmes (que nous avons résolus). Le premier que nous avons rencontré est la création de la fonction durée, qui calcule le nombre jours entre deux dates. Nous avons mis beaucoup de temps à la réaliser car le code de cette fonction était assez long.

De plus pour réaliser une fonction la plus précise possible il fallait savoir combien il y avait de jours dans chaque mois, en prenant aussi en compte si l'année est bissextile ou non. Cette action était réalisée dans une autre fonction qui renvoie le nombre de jour selon le mois et l'année (bissextile ou non).

Un autre problème que nous avons rencontré était de pouvoir changer de mode quand on le souhaitait. Au début nous n'y arrivions pas, c'est-à-dire que l'on était bloqué sur le premier mode sélectionné. Pour cela, dans le cas du mode R et du mode H, nous avons rajouté une ligne de code qui redemandait la data (P, D, H ou R).

```
Data = BlueT.read();  
if (Data=='P' || Data=='R' || Data=='D'){  
    H=false;  
}
```

Nous n'arrivions pas non plus à faire que le moteur arrête de tourner lorsque le capteur infrarouge détectait un obstacle. Le moteur ne s'arrêtait pas car avant de s'arrêter il devait effectuer un tour complet puis avant de recommencer à tourner, il vérifiait s'il n'y avait pas d'obstacle. Si c'était le cas, il s'arrêtait, mais il ne détectait pas l'obstacle pendant son tour. Pour régler le problème nous avons l'avons fait tourner de 10 pas pour qu'il vérifie plus souvent :

```
while(isObstacle==HIGH){ //tant qu'il n'y a pas d'obstacle  
    digitalWrite(LED, LOW);  
    Serial.print("pas d'obstacle");  
    isObstacle = digitalRead(isObstaclePin);  
    monMoteur.step(10); //va à la position initiale ///////  
}
```

L'Assemblage

En ce qui concerne l'assemblage, nous avons rencontré plusieurs problèmes :

- La partie qui permettait aux fils de ne pas s'entortiller sur eux-mêmes ne coulissait pas assez bien avec la plaque et donc cela exerçait une résistance. Nous avons donc attachée la partie non tournante à la partie du moteur qui ne tourne pas à l'aide d'un fil afin de simplifier le coulissage.
- La tige reliant le moteur à la plaque tournante était fixée à l'aide d'une petite plaque en bois ce qui faisait que la plaque ne tournait pas selon un seul axe. Nous avons donc changé cette petite plaque par une plaque en fer, il suffisait simplement de poncer la tige pour qu'elle rentre facilement dans le trou prévu de la plaque en fer.
- Le moteur n'était pas stable, en effet, la planche et la lune lui imposaient un poids trop grand pour qu'il reste correctement en place et donc quand le moteur tournait la lune tombait ; il fallait à chaque fois utiliser beaucoup de scotch pour fixer le moteur à la table pour que rien ne bouge.
- Il était difficile de fixer les deux demi-lunes entre elles à cause de leur composition (résine plastique), nous avons essayé de scotcher les demi-lunes mais cela n'a pas tenu. Lors de la dernière séance, nous avons récupéré 4 petites planches de plastique qui ont été utilisées pour coller les deux demi-lunes ensemble.
- Nous voulions que la plaque tourne dans la lune, mais nous n'arrivions pas à faire un socle adéquat pour tenir la lune, finalement on a fait tourner la lune avec la plaque.

CONCLUSION

Objectifs atteints ?

Nous avons réussi à commander une lune, imprimée à l'imprimante 3D, à l'aide du module Bluetooth et d'un téléphone.

Les deux demi-lunes sont tenues sur la plaque tournante, elle-même correctement fixée au moteur. Les LEDS que l'on a utilisées dans la lune tournent avec la plaque en bois sans poser de problème aux fils, en effet, les fils ne s'entortillent pas sur eux-mêmes grâce à la plateforme coulissante fixée au moteur.

Les LEDs et le moteur fonctionnent quand ils reçoivent les commandes du module Bluetooth.

Cependant lors de l'oral, à cause de nombreux faux-contacts, de dysfonctionnements du matériel, les lumières ne s'allumaient plus ou peu, le capteur ne détectait pas correctement les obstacles et le moteur ne tournait plus.

Améliorations possibles ?

Nous aurions aimé que la lune se replace à sa position initiale, toute seule, à chaque fois que l'on change de mode. Cependant, nous n'avons pas réussi à établir cette position initiale. Il a fallu trouver une alternative, et imposer un obstacle devant le détecteur infrarouge afin de positionner la lune comme souhaitée. Cela a simplifié le travail, car lorsque l'on se situe à différents endroits, on peut changer la position initiale alors que dans le cas contraire, la personne devait se déplacer.

Il aurait été préférable de fabriquer un socle correct afin de cacher les fils, de stabiliser le moteur et de permettre à la lune de ne pas tourner avec la plaque mais qu'elle soit fixe. Cela aurait aussi permis de fixer le circuit entier.

A la vue du nombre de faux-contacts rencontrés lors du déplacement de la lune, il faudrait remplacer les fils par des fils plus solides. Cela permettrait de faire fonctionner la lune en permanence, même si celle-ci est déplacée.

Remerciements

Nous souhaitons adresser nos remerciements à

- M. Masson pour ses cours et le matériel fourni
- Nassim Abderrahmane
- Marc Former au Fablab pour nous avoir aidé à imprimer la lune et tailler la plaque

BIBLIOGRAPHIE

Impression de la Lune

<https://www.instructables.com/id/Print-Your-Own-Moon/>

Branchement des lumières

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

Bibliothèques

<https://platformio.org/lib/show/83/RTClib>

https://github.com/adafruit/Adafruit_NeoPixel

<https://www.arduino.cc/en/reference/wire>

<https://www.arduino.cc/en/reference/stepper>

Calendrier lunaire

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

Moteur pas à pas

<http://users.polytech.unice.fr/~pmasson/Enseignement/Elements%20de%20robotique%20avec%20arduino%20-%20Moteurs%20-%20Projection%20-%20MASSON.pdf>

Détecteur infrarouge

<http://users.polytech.unice.fr/~pmasson/Enseignement/Elements%20de%20robotique%20avec%20arduino%20-%20Distance%20et%20detection%20d%20obstacles%20-%20Projection%20-%20MASSON.pdf>