

# PROJETS ARDUINO – Peip2

*Année scolaire 2018-2019*

***"Roulette de casino Automatique"***

**Etudiants : Yidong LI  
Charly DUCROCQ**

**Encadrants : Pascal MASSON**

## **REMERCIEMENTS**

Merci à Mr. Pascal Masson, notre professeur en électronique pour son investissement sur, non seulement notre projet, mais aussi tout ceux de la promo.

Merci à Mr. Marc Forner, responsable du fablab pour son travail et son aide lors de nos sessions au fablab.

Et merci à tous nos camarades de groupe pour leurs aides et conseils aides précieux.

# SOMMAIRE

Introduction .....	4
Le cahier des charges .....	5
Partie Roulette .....	6
I.1. Etat actuel et idée de fonctionnement .....	6
I.2. Développement de la partie et problèmes rencontrés .....	7
I.3. Analyse du développement .....	12
Partie Traitement de la mise .....	14
II.1. Etat actuel et idée de fonctionnement .....	14
II.2. Développement de la partie et problèmes rencontrés .....	15
II.3. Analyse du développement .....	17
Conclusion .....	19
Bibliographie .....	20

# Introduction

Dans le cadre de notre formation en deuxième année Peip, dans l'enseignement en électronique Arduino, il nous est demandé d'imaginer, développer et concevoir un projet utilisant Arduino. Nous avons alors choisi comme sujet la roulette de casino. L'idée étant d'automatiser le processus de jeu afin qu'un croupier ne soit plus nécessaire.

Ce rapport commencera tout d'abord par la définition des objectifs initiaux, le cahier des charges. Par la suite, il se découpera en deux parties : la partie « roulette », produite par Charly Ducrocq et la partie « traitement de la mise », produite par Yidong Li. Chacune d'entre elles détaillant le développement et l'analyse du travail de leur production.

Enfin, il se terminera par une conclusion résumant l'état final de ce projet.

# Le cahier des charges

Notre objectif était donc d'automatiser une roulette de casino. Pour ce faire il fallait donc imaginer et réaliser des systèmes permettant :

1. La détection de la mise
2. Le triage de la mise en vue de sa redistribution
3. Le lancement de la roue et de la bille.
4. La détection du résultat du lancer
5. L'évacuation de la bille
6. La distribution des gains éventuels

Nous avons alors choisi de se séparer la tâche. Yidong s'occuperait de tout ce qui est traitement de la mise (les parties 1, 2 et 6 de la liste ci-dessus) pendant que Charly lui s'occuperait de tout ce que traite de la roue et du lancer (parties 3, 4 et 5 de la liste ci-dessus).

# PARTIE ROULETTE

## I.1. Etat actuel et idée de fonctionnement

Tout d'abord voici une photo du résultat de production de la roulette :

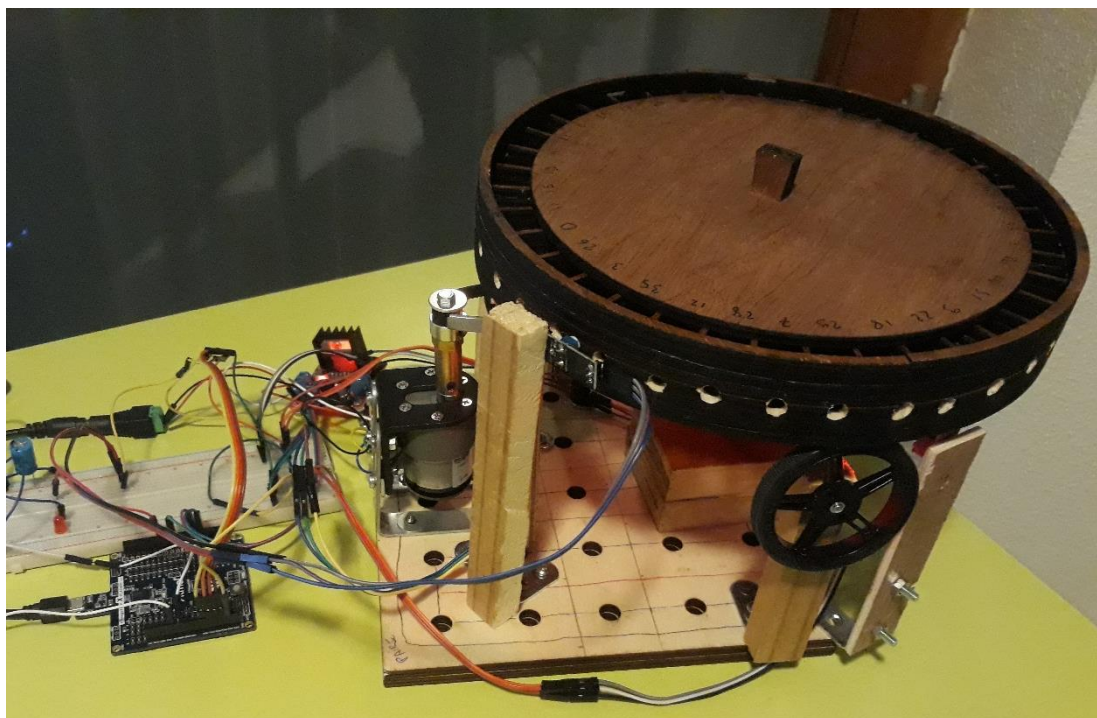


Figure 1 : Etat final de la roue

Ici, trois types de matériel électronique sont utilisés : un moteur courant continue 12V, quatre capteurs d'aimant et un capteur de proximité.

Les capteurs d'aimant auront pour but de situer la roue à l'aide d'un aimant incruster sous la roue (voir *Figure 8 : Aimant sous la roulette*, page 11) afin de pallier au manque de précision du moteur courant continue.

Le capteur de proximité (on peut le voir en plein centre de la figure 1) lui servira à repérer la bille. En effet, pour lui le noir du côté de la roue et le vide correspondent à la même chose, absorbant son signal, alors que le blanc de la bille lui renvoie son signal lui permettant ainsi de la détecter.

Concernant le lancement de la bille et son éjection, ces deux objectifs n'ont malheureusement pas pu être réalisés par manque de temps. Notons cependant que la roue a été conçue afin de permettre l'éjection de la bille si on la poussait par le trou (voir *Figure 2 : Modélisation de la roulette*, page 7).

Pour en venir à son état de fonctionnement, la roulette détecte bien la bille et l'aimant, et bien que le programme prélève correctement les données nécessaires à la détermination de la position de la bille, la roulette ne peut pas renvoyer ladite position dans son état actuel. Plus de précisions concernant son dysfonctionnement dans la partie I.2. *Développement de la partie et problèmes rencontrés*.

## I.2. Développement de la partie et problèmes rencontrés

Le développement de la roulette s'est finalement vu dissocié en trois parties :

- Test des capteur et peaufinage de l'idée de conception (Séance 1 et 2)
- Construction du prototype (Séance 3 à 7)
- Développement du programme et essai réel (Séance 8)

On remarque déjà la conception du prototype s'est révélée être la partie la plus imposante du projet mais nous reviendrons sur ce point dans la partie I.3.

Concernant les deux premières séances, il n'y a pas grand-chose à dire de plus que dans son titre. On pourrait même dire qu'elles ont plus servi à discuter avec mon binôme et Mr. Masson du fonctionnement de la roulette que de faire de réels tests. C'est à la fin de la 2<sup>e</sup> séance qu'apparaît une évidence : sans prototype je ne pouvais rien faire. C'est ainsi que débute la construction de la roulette.

Tout d'abord il fallait imaginer et représenter la roulette. Pour ce faire, j'ai utilisé un logiciel de modélisation nommé Autodesk. Voici le résultat :

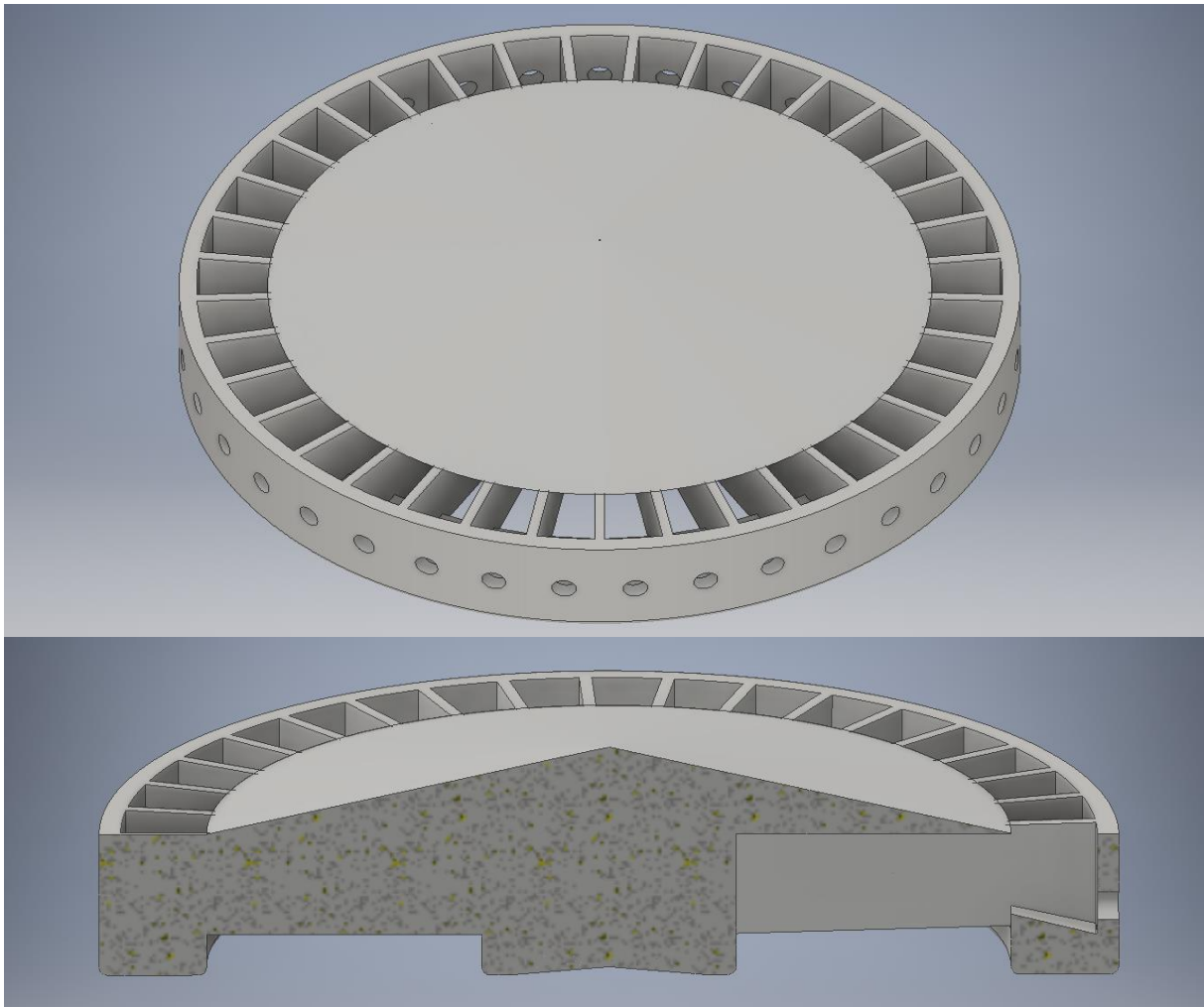


Figure 2 : Modélisation de la roulette

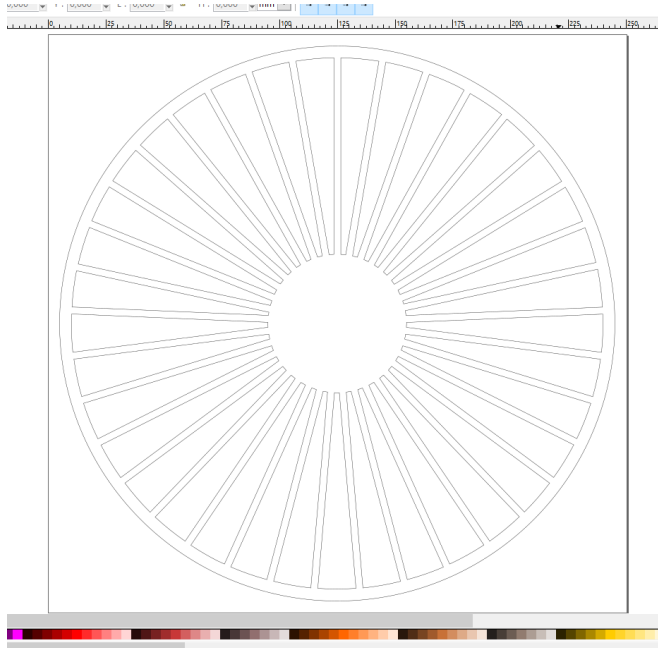
On peut voir ici que lorsque la bille tombe dans l'un des emplacements, elle est maintenue au bord de la roulette grâce à une pente/cale. De plus, afin de pouvoir évacuer la bille la roulette

a été pensé afin d'éjecter la bille par le centre en la poussant par le trou à l'extrémité de la roulette.

Ensuite, après m'être renseigné auprès de Mr. Masson concernant les différents moyens de conception, il m'a conseillé de découper plusieurs planches dans la forme souhaitée et de les coller les unes sur les autres afin d'obtenir couches par couches le prototype.

Au début, j'ai cru à tort qu'il fallait les découper à la main à la scie automatique et me suis lancé dans le dessin d'un patron pour ce faire. Cependant, grâce à Clément, un camarade de groupe, j'ai compris qu'il serait plus efficace d'utiliser la découpeuse laser du fablab. Il m'a alors conseillé un logiciel nommé Inkscape, utile pour le fonctionnement de celle-ci.

Voici un exemple de visuel Inkscape :



Ces fichiers sont ensuite lus par la découpeuse laser afin de découper comme demandé les planche de bois (voir Figure 4 ci-dessous).

Figure 3 : Exemple de rendu Inkscape



Figure 4 : Découpeuse laser



Après deux après-midis au fablab, la roulette est prête pour des essais.

A noter qu'un seul trou a été fait cette fois-ci. De plus, des cales ont été placées à l'aide des restes de la découpe afin de remplacer la pente imaginée dans la modélisation. Le reste des trous et cales seront percés et tracés plus tard.



Figure 5 : Roulette fini

Lors de la séance suivante, il était déjà bon de constater que le capteur de proximité fonctionnait correctement avec la roulette. Le trou de 6mm de diamètre était suffisant pour son utilisation et les irrégularités de perçage ne semblaient pas gêner la perception du capteur.

Maintenant, il fallait construire un support pour la roulette auquel allaient être fixés capteurs et moteur.

Au départ, il était prévu de fixer la roulette directement sur le moteur, cependant celle-ci était trop lourde pour celui-ci afin qu'il puisse tourner à une vitesse contrôlable. Il était donc nécessaire d'utiliser un mécanisme intermédiaire afin de palier à ce problème. Après avoir récupéré une tige de métal de 8mm de diamètre, un roulement à bille et une courroie, je retourné donc au fablab afin de découper la barre, découper un cercle denté en bois (afin que la roue adhère à la courroie) et fixer le tout à la roulette.

On obtient alors le résultat visible sur la *Figure 6*.

Cependant, le roulement à bille s'est montré moins stable que prévue. Pour pallier à ce problème, il fallait donc placer des roues juste sous la roulette afin de la maintenir droite.

Il ne manquait plus alors que de fixer les capteurs. Après plusieurs séances de bricolage, on arrive alors au résultat visible sur la *Figure 7*.

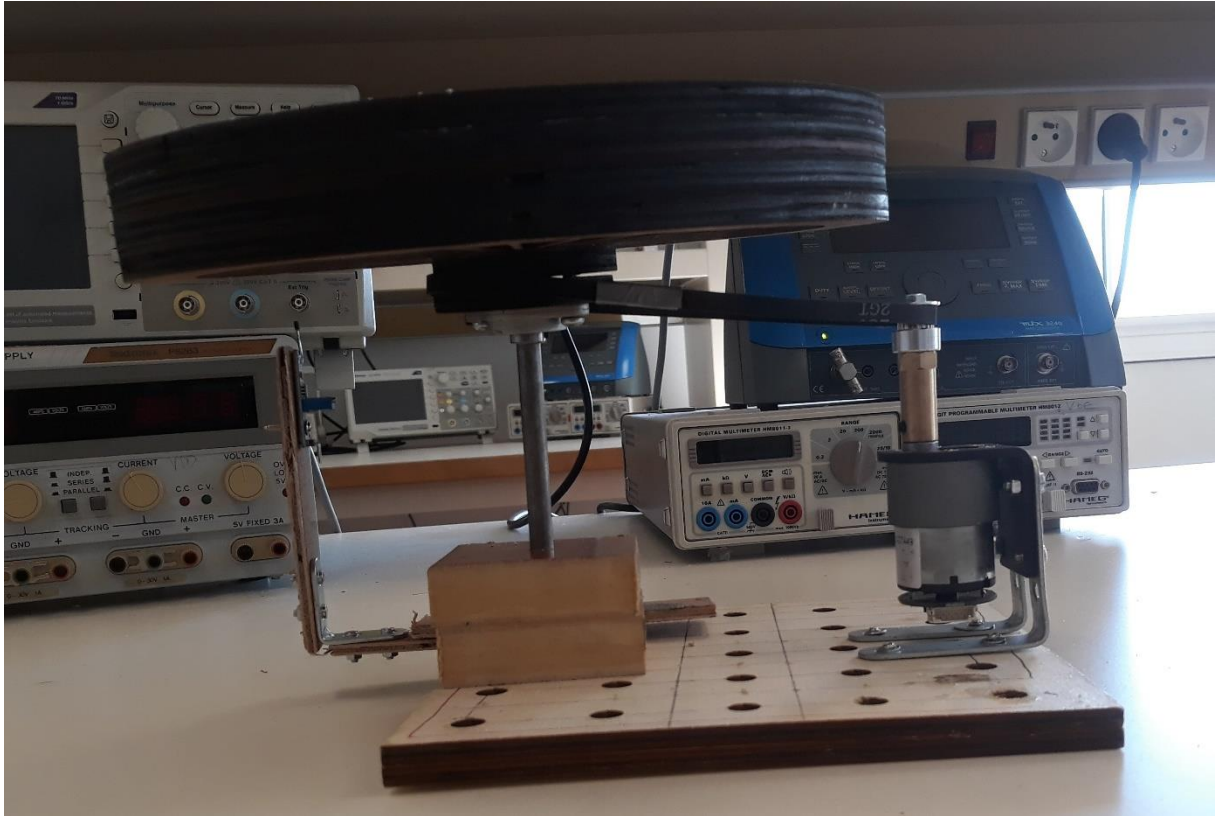


Figure 6 : Rendu "courroie"

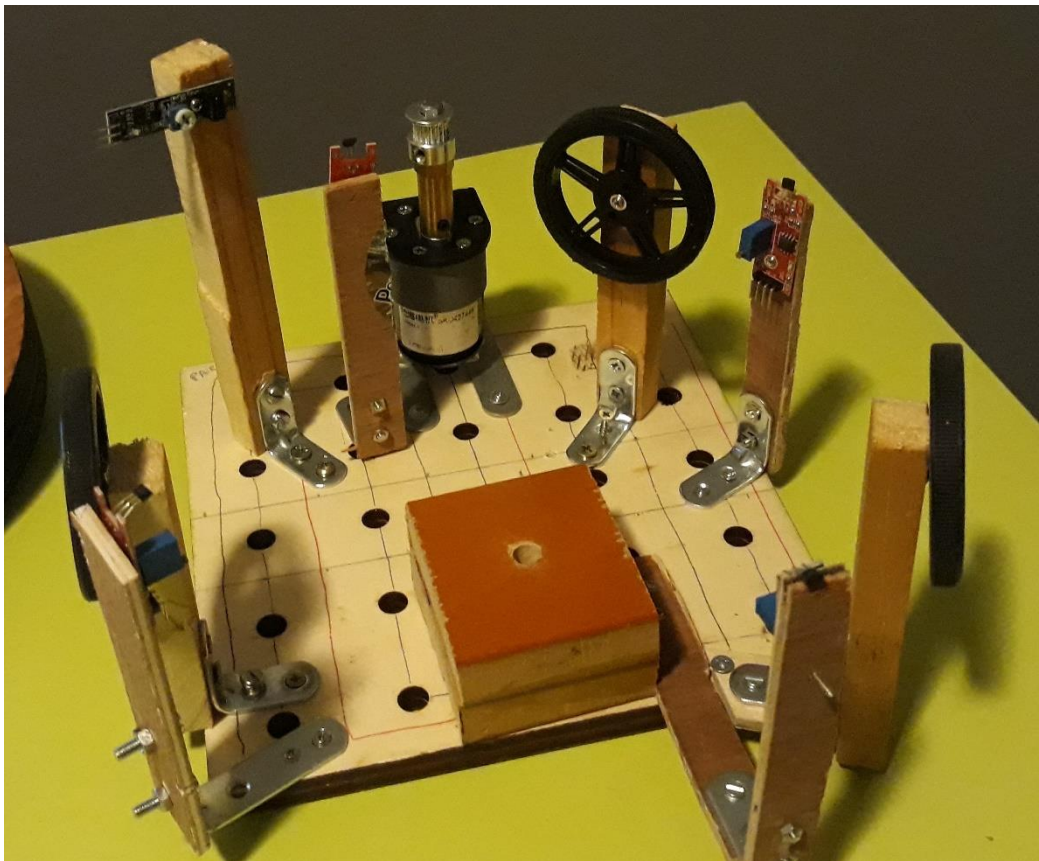


Figure 7 : Rendu "support"

De plus, entre temps, un aimant a été incrusté sous la roue :



Figure 8 : Aimant sous la roulette

Notons que, entre-temps, il m'est venu à l'idée de souffler la bille plutôt que de la pousser pour son éjection. Cela aurait grandement faciliter la tâche car on n'aurait plus eu à se soucier de comment faire passer une tige ou autre dans des trous de 6 mm dont les positions sont peu fiables car percé à la main et du fait d'arrêter la roulette avec le bout trou pile au bon endroit pour l'éjection. Cependant, les pompe à air qu'on a pu trouver n'était pas assez puissant pour la mise en utilisation de ce système. On a alors dû revenir à notre idée de « pousseur » de départ même si on n'a pas eu le temps de le concevoir.

Après avoir percer les trous restants et placer les cales restantes, je pu enfin m'attaquer à la programmation et au test de celle-ci.

Le programme est consultable ici :

<https://github.com/CharlyDucrocq/RouletteAuto/blob/master/Documents/Programmes/Roulette.ino>

Globalement, le programme peut être résumé cette forme :

- Lancement de la roue
- Diminution de la vitesse pour la lecture
- A chaque fois que l'aimant est détecté, le programme enregistre le chrono et le n° de capteur
- Lorsque la bille est détectée, le chrono est enregistré et comparer à celui du capteur d'aimant le plus récent.
- Les données sont traitées et afficher

Deux choses sont à savoir sur le développement de ce programme.

Premièrement, une fonction est ici nécessaire afin d'arrêter le programme en cours afin d'enregistrer les capteurs lorsque qu'il détecte quelque chose. Cette fonction se nomme `attachInterrupt()`.

Deuxièmement, les différences de chrono relevé sont trop variables et des fois, ce sont des valeurs aberrantes qui sorte. Pour pallier à ce problème, je refais la mesure 10 fois. De ces valeurs, je retire les valeurs aberrantes puis je fais la moyenne des valeurs restante.

Avec ces traitements, je me suis retrouvé avec des valeurs très prometteuse. Voir tableau ci-dessous :

*Tableau 1 : Différences de chrono prélevées*

N°	Essai 1	Essai 2	Essai 3	Essai 4	Essai 5	Différence max en %
0	321	313	311			3,164556962
32	266	264				0,754716981
15	231	225	231			2,631578947
19	190	179	185			5,962059621
4	144	142				1,398601399
21	103	103	106			2,870813397
2	63	65	63			3,125
25	24	24	23			4,255319149
17	10	8	10			22,22222222
34	482	505	481	485	507	5,263157895
6	440	429	482	458	431	11,63556531
27	383	402	377	375	391	6,94980695
13	322	353	320	312	321	12,33082707
36	268	255	270	254		6,106870229
11	223	212	200	202		10,87470449

Malheureusement, lors de la prise de ces mesures, la courroie s'est mise à sauter. Je pense que le problème vient de l'érosion du trou dans lequel et planté la barre de métal. A force de retirer et replacer la roulette encore et encore, le trou c'est dilater, entraînant un très léger basculement de la roue vers le moteur.

J'ai par la suite essayé de combler ça en resserrant un peu la courroie cependant, cela fait que la roulette est un peu plus penchée qu'avant. De certes très peu, mais cela suffit à décaler les trous de quelque mm vers le bas au niveau du capteur de proximité.

A ce moment-là, il ne restait plus que 2 jours avant la présentation. Je n'avais plus le temps de corriger quoi que ce soit.

### I.3. Analyse du developpement

Tout d'abord, voici la différence entre ce qui était prévue et ce qui s'est passé (Voir *Tableau 2*).

Ce qui saute tout de suite aux yeux, c'est la différence entre les deux temps de production du prototype. En effet, la plus grosse erreur aura été la sous-estimation de ce temps. La production du prototype aurait dû être prioritaire car ici, sans lui, rien ne pouvait avancer.

Ensuite, beaucoup de travail n'as pas été fait.

- Ejection de la bille
- Lancement de la bille
- Communication (Bluetooth ou autre) entres les parties des deux binômes

Tableau 2 : Planning initial/final Roulette

		10/12/2018	17/12/2018	09/01/2019	14/01/2019	06/02/2019	13/02/2019	27/02/2019	06/03/2019
<b>Planning initial</b>									
Roulette	Maquette								
	capteurs								
	Prototype								
	Programmation								
	"pousseur"								
	lancement bille								
<b>Planning final</b>									
Roulette	Maquette								
	capteurs								
	Prototype								
	Programmation								
	"pousseur"								
	lancement bille								

Certes cela s'explique par le manque de temps cependant, avec du recul on peut se dire que bien que le manque de prototype entravait la production de l'« éjecteur » de bille, cela n'entravait en rien la production des deux autres systèmes. Ces deux parties aurait dû être fait en parallèles.



# PARTIE TRAITEMENT DE LA MISE

## II.1. Etat actuel et idée de fonctionnement

Voici une photo du dispositif dans l'état actuel :

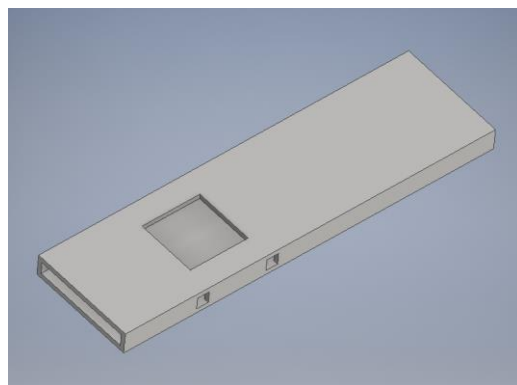


*Figure 1. Etat final du dispositif de la partie traitement de la mise.*

Ici, deux types de composant électrique sont utilisés : 4 servomoteurs et un capteur de couleur.

Les servomoteurs sont utilisés pour le système de blocage, le système de triage et le système de redistribution. Pour redistribuer les mises, il nous faut des servomoteurs de puissance un peu plus élevée. Les autres sont des servomoteurs de type de 9 grammes.

Le capteur de couleur fonctionne comme suit : lorsque le capteur de couleur est activé, il disperse l'onde lumineuse vers l'extérieur et induit le nombre d'ondes lumineuses reflétées par l'objet, et calcule la valeur RVB de l'objet en fonction du nombre d'ondes lumineuses retournées, afin que nous puissions déduire la couleur de l'objet.



*Figure 2. Passage pour les mises*

Dans la photo, en dessus il y a un tuyau noir, sous forme de parallélépipède rectangulaire (Figure 2). Le capteur de couleur est monté sur le dessus, près de la position du trou carré. Quand une mise est insérée dans le tuyau et se déplace à la position en face du capteur de couleur, le capteur peut distinguer la couleur de cette mise.

Une fois cette mise distinguée, le capteur de couleur va envoyer une valeur correspondante au système de triage (Figure 3), c'est-à-dire au servomoteur. Ce servomoteur va utiliser cette valeur pour tourner un angle spécifique de sorte que la mise tombe dans l'un des 5 récipients (Figure 4).



*Figure 4. Les cinq récipients*

La partie du système de redistribution présente un problème au niveau physique. Malheureusement à cause de mon erreur de construction, les deux premières parties prennent tellement d'espace que ma sortie pour le système de redistribution ne peut pas être utilisée correctement, ce qui fait que le système ne fait seulement que la moitié de l'objectif.

## **II.2. Développement de la partie et problèmes rencontrés**

Globalement, deux principaux problèmes ont été rencontrés.

1. Le problème de détection, la couleur ne peut pas être détectée lorsqu'une mise passe.
2. La construction du dispositif du système de triage.

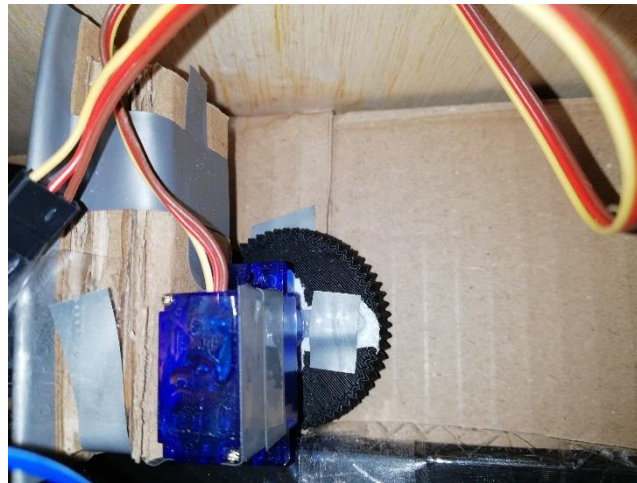
•

Pour construire le système du capteur de couleur, j'ai utilisé un tuyau en carton pour tester le fonctionnement. Et j'ai observé que la mise traverse le tuyau beaucoup plus vite que le capteur de couleur distingue la couleur, et même si j'accélère la distinction du capteur de la couleur dans le programme, il ne résoudra pas le problème, et le capteur ne pourra pas obtenir sa couleur quand il passe, en d'autres termes, le capteur n'a pas le temps d'évaluer la mise.

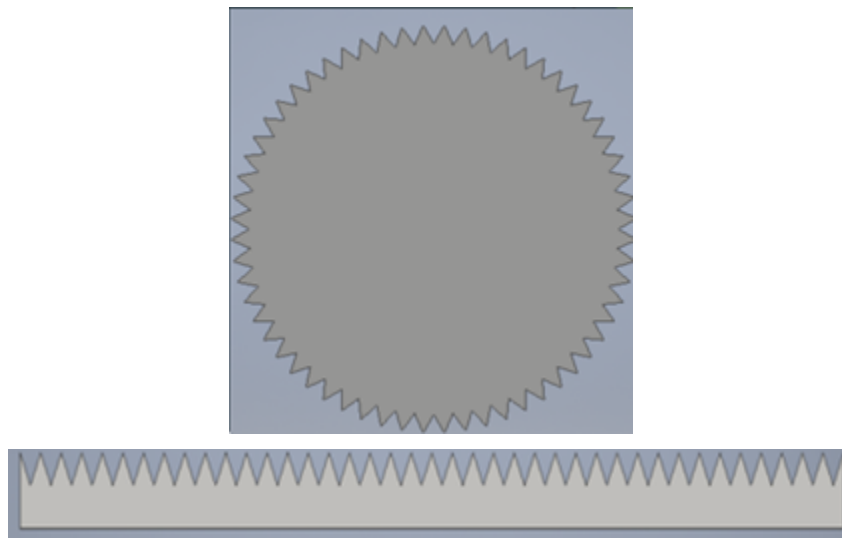


*Figure 5. Capteur de couleur TCS3200*

Pour résoudre ce problème, j'ai pensé à contrôler le passage des mises d'une manière similaire à un « interrupteur ». J'ai fait un engrenage basé sur un servomoteur, puis des trous pour traverser le tube horizontalement. Quand une mise arrive, l'engrenage pousse une tige à travers les trous pour bloquer la mise, et une fois que la couleur de la mise est détectée et distinguée, l'engrenage va reculer la tige pour permettre à la mise de passer.

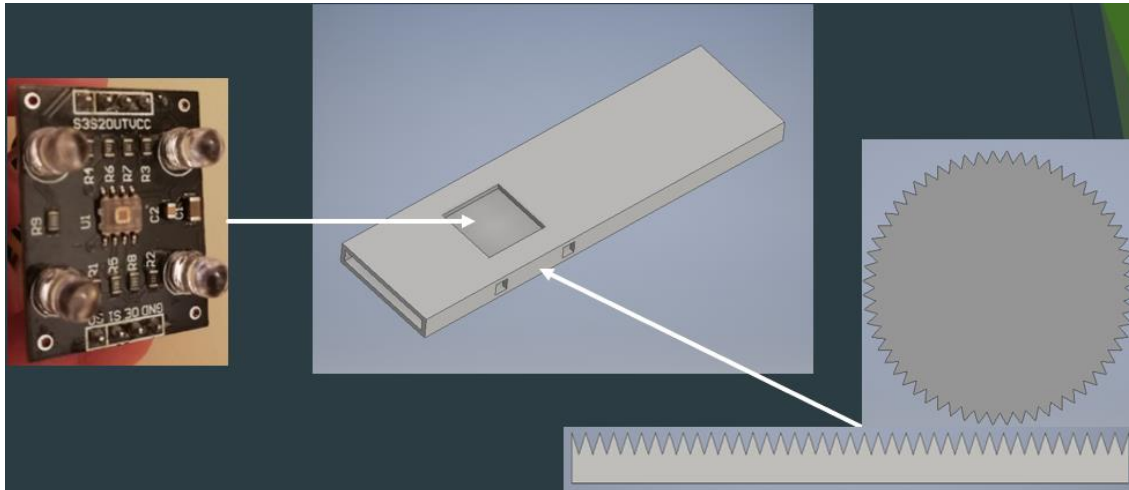


*Figure 6. Système de blocage*



*Figure 7. L'engrenage et la tige*





*Figure 8. Le diagramme d'assemblage*

• Pour le système de triage, j'ai pensé à utiliser 5 servomoteurs associés aux 5 récipients pour trier les mises (comme celui d'un monnayeur), mais je me suis aperçu que je n'aurais pas assez de port pour le reste de mon circuit parce qu'il me faut déjà 8 ports (6 pour le capteur de couleur et 2 pour les servomoteurs du système de redistribution).

Donc je n'ai choisi qu'un seul servomoteur pour trier les mises, composé d'un passage pour glisser les mises.

Avec un tel système, même s'il rend le procédure un peu compliqué et moins efficace, c'est encore réalisable pour seulement une carte Arduino.

### II.3. Analyse du développement

Planning initial		10/12/2019	17/12/2019	09/01/2020	14/01/2020	06/02/2020	13/02/2020	27/02/2020	06/03/2020
manipulation jeton	Capteur couleur								
	Triage								
	Redistribution								
	panneau de contrôle								
Planning finale									
manipulation jeton	Capteur couleur + stop								
	Triage								
	construction physique								
	Redistribution								
	panneau de contrôle								

*Tableau 1*

Dans le tableau, nous pouvons voir clairement ce qui était prévue et ce qui s'est fait à la fin.

Au début, lorsque j'ai exécuté le capteur de couleur, j'ai mis trop de temps pour classer la couleur en raison de couleur de mise.

Et puis lorsque j'ai essayé de construire le système de triage, il n'y avait presque pas de progrès pendant une séance parce que je ne savais que le servomoteur que j'ai obtenu était un servomoteur qui tourne tout le temps.

L'erreur la plus importante dans l'ensemble du plan était d'estimer le temps de la production physique. L'imprimante n'a pas bien imprimé le composant attendus trois fois pendant l'impression du tuyau, jusqu'à la quatrième fois pour obtenir le composant qui pourraient être utilisable. Concernant l'assemblage, parce que fait à la main, les composant, bien que fonctionnels, ne peuvent pas être précisément fixé à la position souhaitée. On se retrouve donc à devoir reconstruire encore et toujours.

Même en raison du manque de temps, je devrais construire le système séparément au niveau physique dès le début, et enfin le combiner ensemble.

**Ce qui ne fonctionne qu'en théorie n'est d'aucune aide à la pratique !**

# *Conclusion*

Pour conclure, il a été fait :

- Pour la roulette :
  - Le prototype de roulette et son support
  - Le programme permettant de mesurer la différence de temps entre le capteur d'aimant le plus proche et l'aimant au moment où la bille est détectée
  - Le squelette du programme final (il faut compléter les fonction lancerBoule() et ejecterBoul())
- Pour le traitement de la mise
  - Le prototype du système de détecter et trier de mise et son support.
  - Le programme permettant de distinguer la couleur de la mise et de le trier dans les récipients correspondants.
  - Une partie du système de redistribution de la mise.

Il reste à faire :

- Pour la roulette :
  - Le traitement du résultat fournis afin de déterminer l'emplacement de la bille.
  - Le lancement de la bille.
  - L'éjection de la bille.
- Pour le traitement de la mise.
  - La redistribution de la mise en fonction de résultat fourni par le jeu de roulette.
- Pour les deux :
  - La communication entre les deux parties. Nous pouvons utiliser les potentiomètres (ou Bluetooth) pour échanger les données entre deux cartes Arduino, donc réaliser la communication entre deux parties.

# ***Bibliographie***

Autodesk Inventor : <https://www.autodesk.fr/>

TCS3200: <https://www.gotronic.fr/pj2-tcs3200-fr-1446.pdf>

Timer : [https://www.pjrc.com/teensy/td\\_libs\\_TimerOne.html](https://www.pjrc.com/teensy/td_libs_TimerOne.html)

Générateur de boites : [http://carrefour-numerique.cite-sciences.fr/fablab/wiki/doku.php?id=projets:generateur\\_de\\_boites](http://carrefour-numerique.cite-sciences.fr/fablab/wiki/doku.php?id=projets:generateur_de_boites)

Info attachInterrupt : <https://www.locoduino.org/spip.php?article64>