

## 1. LES ROBOTS.

### a) Un peu d'histoire.

L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "robota" signifie travail forcé. Un robot, c'est un appareil automatique conçu pour effectuer une tâche normalement exécutée par un être humain. Au cours de l'histoire on peut distinguer **3 générations de robot** :

#### - *La première génération*

Cette génération de robots correspond aux "Automates". Ces systèmes sont programmés à l'avance et permettent d'effectuer des actions répétitives. Un automate, c'est une machine animée par un mécanisme intérieur, qui imite l'apparence et les mouvements d'un être vivant. Cela peut être une machine, un mécanisme automatique, un robot industriel, un distributeur automatique.



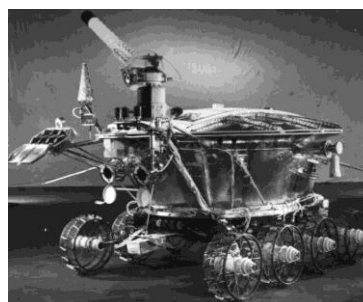
(1772), Pierre Jacquet-Droz invente des automates en formes de pantins de bois , dessinateur, musicien ou encore écrivain.

#### - *La deuxième génération*

Ces robots sont dotés de capteurs.

Ils peuvent réagir à l'environnement qui les entoure.

On trouve des capteurs de température, photoélectronique, à ultrasons pour par exemple éviter les obstacles et/ou suivre une trajectoire. Ces capteurs vont permettre au robot une relative adaptation à son environnement. Ces robots sont donc bien plus autonomes que les automates.



Le vaisseau spatial soviétique Luna 17 a posé le premier robot piloté à distance sur la Lune : Lunokhod 1.

#### Unimate 1950 :

Premier robot manipulateur industriel.

Il a été utilisé pour travailler sur les lignes d'assemblage de General Motors à partir de 1961



### - La troisième génération

Dotés d'intelligence artificielle ces robots peuvent prendre des décisions grâce à *des modèles mathématiques complexes tels que les réseaux de neurones*. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. Bien sûr il faudra attendre encore longtemps avant que le plus "intelligent" des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'Homme.



(2009) Le robot HRP-4C : est le premier robot humanoïde. Il a l'apparence de jeune fille japonaise. Ce robot est doté de 30 articulations lui permettant d'effectuer des mouvements souples et d'avoir une démarche fluide.

## b) Quels sont les domaines d'utilisation des robots ?

### - Les robots dans l'industrie

Les robots sont particulièrement présents dans le domaine industriel. Ils effectuent des tâches bien précises et de différentes formes.

La plupart des robots des robots est utilisée pour effectuer des tâches simples et répétitives.

On choisit d'utiliser la robotique lorsque:

- la tâche à effectuer est assez simple pour être robotisée
- les critères de qualité sur la tâche sont importants
- la tâche est pénible (peinture, charge lourde,...)

Les robots sont d'une précision et d'une rapidité incomparables, et contribuent ainsi à une optimisation considérable dans le domaine qu'est l'industrie : ils peuvent garantir la qualité et la productivité des pièces fabriquées.

Les robots sont très utilisés dans les chaînes de montage de l'industrie automobile, où ils y remplacent les ouvriers dans les tâches pénibles et dangereuses (peinture, soudage, emboutissage, etc.).

Robot de peinture



Robot soudeur



- ***la robotique médicale***

Des robots sont utilisés dans diverses branches du domaine médical, si bien dans les laboratoires d'analyses médicales, que dans la chirurgie ou encore entant qu'outil de rééducation.

Il existe des robots chargés d'assister des personnes handicapées ainsi que les personnes âgées qui accomplissent des tâches simples du quotidien. La robotique d'assistance a une grande importance, elle permet aux personnes qui habituellement ne peuvent pas vivre seuls de les rendre autonomes. Par exemple les fauteuils roulants sont de plus en plus perfectionnés. Sur cette image le fauteuil est muni d'un bras articulé permettant à la personne handicapée d'atteindre des objets qui lui sont inaccessibles.



Les robots sont également présents dans la médecine et dans la chirurgie, ils sont probablement les infirmiers du futur. Par exemple le système chirurgical Da Vinci est un robot chirurgical qui permet d'opérer à distance, soit dans la même pièce avec une machine comme intermédiaire, soit d'un endroit très éloigné, ce qui peut être très utile souvent. Celui-ci est principalement utilisé aux États-Unis et en Europe.



- ***Les robots exploreurs***

Ce type de robots est utilisé par l'homme pour explorer des zones dangereuses, des environnements difficiles, inaccessibles pour les humains comme par exemple les zones polaires, l'espace...L'exploration de la centrale nucléaire de Tchernobyl, par exemple, a été faite par un robot. Ces robots sont assimilés à des pantins, ils sont dirigés par l'homme.



*Spirit : robot envoyé sur Mars*

- **Les robots dans la vie quotidienne : les robots domestiques et ludiques :**

De nos jours les robots ont leur place dans la vie quotidienne : robots ménagers, domestiques, de services ...

La robotique ludique et la robotique domestique n'ont pas le même objectif à l'origine. Les robots domestiques ont pour fonction d'assister les hommes par exemple avec les robots-aspirateurs ou encore les tondeuses automatiques. La robotique ludique est un secteur en plein développement. Les robots ludiques sont comme des jouets. Ils permettent aussi de s'apercevoir des progrès techniques et technologiques ainsi que des recherches effectuées par l'homme.



*Créés par Sony, « Aibo » (1999) est un chien artificiel et « Qrio » un petit humanoïde( 2006).*

- **Les Humanoïdes**

Un humanoïde est un robot qui ressemble à l'homme, en effet ils prennent une forme humaine pour être plus proches de l'homme. A l'origine le robot humanoïde n'était qu'un mythe qu'on trouvait dans les films de science fiction. Aujourd'hui, ils existent réellement. Le Japon est le leader mondial de recherche et de mise au point de ce type de robot. Ces robots sont en constante évolution.

Ils peuvent être considérés comme des robots ludiques mais ils sont utiles également par exemple dans le domaine médical, pour les soins ou encore au suivi des patients.

Ces robots sont de plus perfectionnés : ils intègrent de plus en plus de nombreux capteurs ainsi que les caméras... Des recherches sont effectuées en permanence.

Peut être un jour seront-ils omniprésents dans la vie de l'homme.



*Uroa ou HRP-4 C*

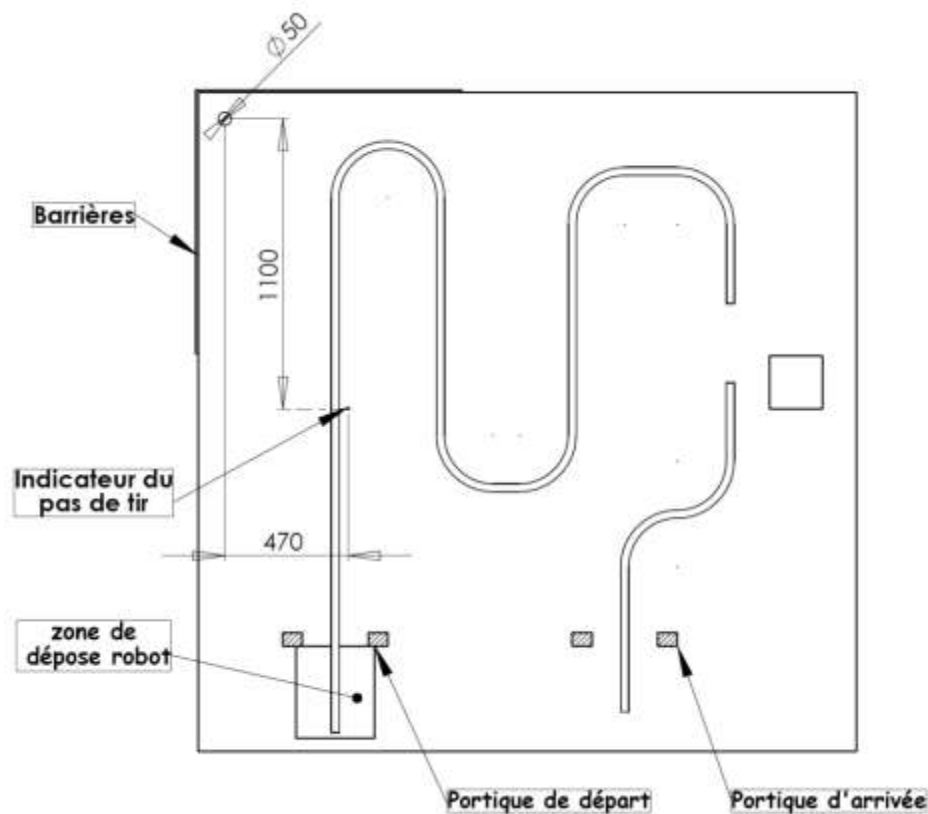
## 1.2 Notre problématique.

Nous allons mettre en œuvre un robot qui devra répondre aux critères du cahier des charges « CHALLENGE ROBOTIQUE 2011 » qui est le concours auquel nous allons participer.

### PRINCIPALE FONCTION

Chaque robot devra pouvoir tenir dans un carton format A4, il devra avoir comme dimensions maximales : 300mm de longueur, 200mm de largeur et 200mm de hauteur.

Comme son nom nous l'indique, le robot suiveur de ligne doit tout d'abord : suivre une ligne . Dans le cas du concours, la ligne sera noire sur fond blanc, peinte ou constituée de ruban adhésif de 30mm de largeur et de couleur mate ou bien brillante. Le robot devra passer sous un portique au départ et sous un second à son arrivée. Il faut aussi savoir qu'il manque un tronçon de bande noire ce qui complexifie le problème.



### SECONDE FONCTION DU ROBOT :

En plus de suivre la ligne il faut que ce robot soit capable de tirer une balle dans un panier ( un robot « Basketteur » ) ou d'envoyer une balle d'YGOLF vers une cible ( robot « golfeur »).

#### **Cas du robot « basketteur » :**

- Doit tirer une balle d'YGOLF de dimensions et de masses précises, dans un panier.
- Une seule balle est autorisée.
- Le panier est situé à une distance précise du poste de tir. (voir la piste)



- Il faut mettre en œuvre un système, comme une catapulte qui nous permettra de projeter la balle.

#### **Cas du robot « Golfeur » :**

- La balle est embarquée sur le robot depuis le départ
- Une seule balle par robot
- La balle d'YGOLF a des caractéristiques précises.
- Les rebonds sont pénalisés, le tir doit être de type « putting »
- Ici aussi il faut mettre en œuvre un dispositif qui permettra de pousser la balle.

→ Nous avons choisi de concevoir un robot suiveur de ligne de type Basketteur.

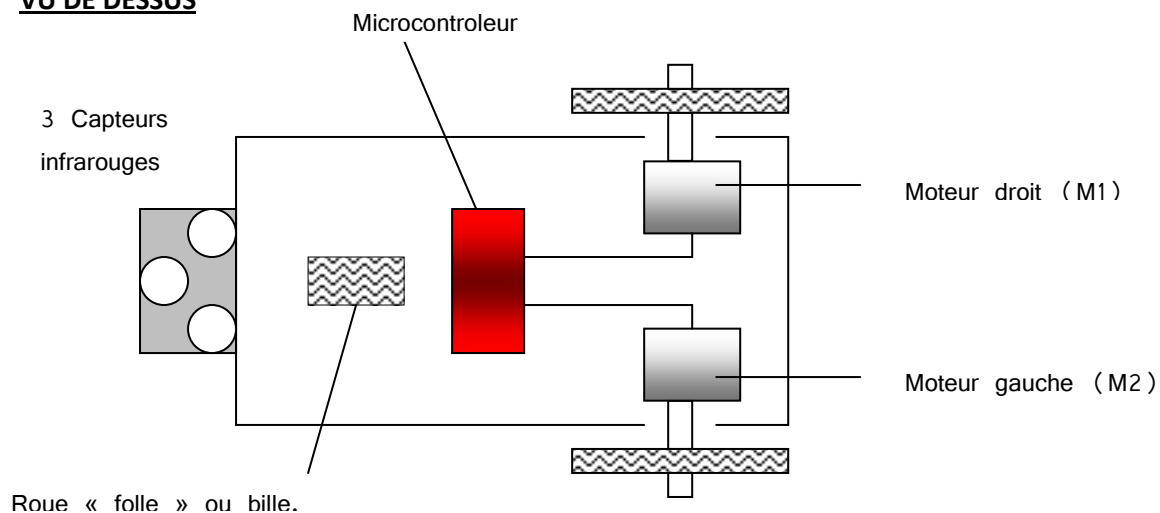
### **1.3 Mise en œuvre de notre robot théorique**

#### a) Fonctionnement simplifié du robot, pour suivre la ligne.

Nous allons tenter d'expliquer le fonctionnement de notre futur robot à l'aide de :

- seulement 3 capteurs infrarouges qui permettront de détecter la position du robot par rapport à la ligne
- 3 roues pour permettre au robot de se déplacer : une devant et deux derrière (pour diriger)
- 2 moteurs pour « contrôler » les 2 roues arrière
- une carte électronique avec un microcontrôleur

#### **VU DE DESSUS**



Pour que notre robot puisse être autonome, il est nécessaire d'établir un programme de fonctionnement à l'aide d'un microcontrôleur. Ce dernier aura donc pour fonction de faire fonctionner les moteurs en fonction du besoin. Mais avant cela, il faut acquérir l'information.

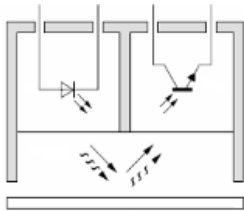
Pour cela nous utiliserons 3 capteurs infrarouges. Il faudra que ces capteurs soient à la fois émetteurs et récepteurs des rayons infrarouges.

Ce type de capteur doit être capable de détecter :

- une couleur foncé
- une couleur claire

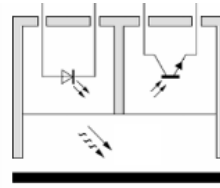
Ainsi chaque capteur peut détecter s'il est au dessus d'une ligne ou non.

### ABSENCE DE LIGNE



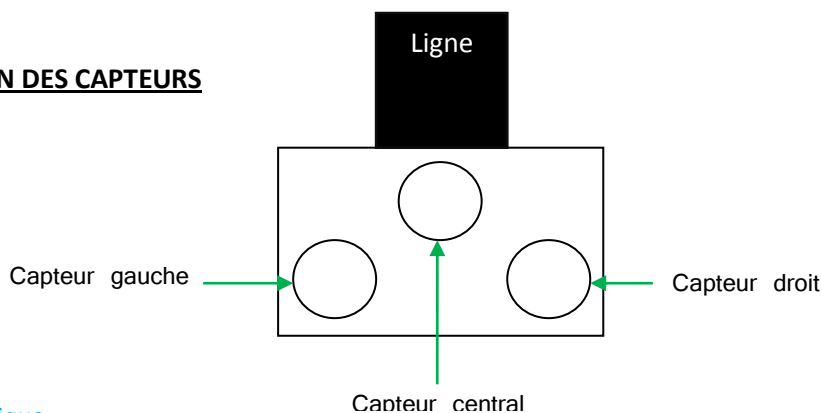
Les rayons émis par le capteur sont réfléchis par la ligne blanche et atteignent la partie réceptrice du capteur.

### PRESENCE DE LIGNE



Les rayons émis par le capteur sont absorbés par la ligne noire, ainsi ils n'atteindront pas la partie réceptrice du capteur.

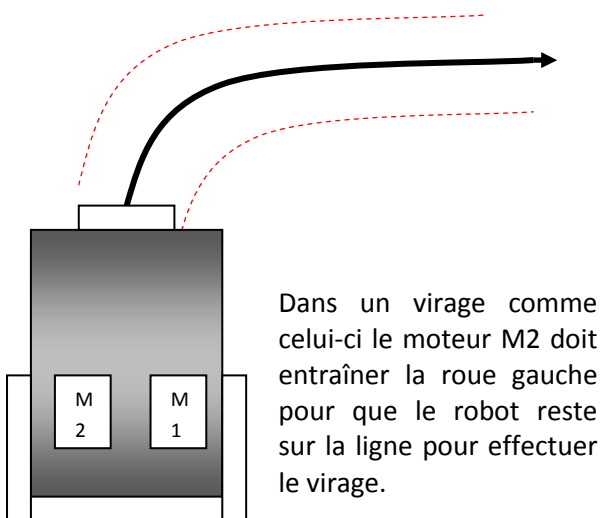
### DISPOSITION DES CAPTEURS



### Etude mécanique

Pour que notre robot puisse avancer, il faut le munir de 2 roues qui seront chacune reliées à un moteur. Pour pouvoir stabiliser le robot et le guider, nous le munirons d'une « roue folle », on peut choisir ce type de roue ou bien mettre une bille.

Ce sont les deux roues fixes, installées à l'arrière gauche et l'arrière droit du robot qui guideront le système le long de la ligne.



Dans un virage comme celui-ci le moteur M2 doit entraîner la roue gauche pour que le robot reste sur la ligne pour effectuer le virage.

### Algorithme général :

Début

Selon Position Robot

Centre : Moteur gauche & moteur droit  $\leftarrow 1$

Gauche : Moteur gauche  $\leftarrow 1$

Droite : Moteur droit  $\leftarrow 1$

Fin Selon

Fin

- Si les deux moteurs tournent à la même vitesse, le robot avance en ligne droite.
- Si l'un des deux moteurs ralentit, le robot va pivoter du côté du moteur le plus lent. Le changement de direction se fait donc en adaptant la vitesse de rotation des moteurs à la courbe de la ligne.
- 

Si l'un des deux capteurs des extrémités chevauche la ligne noire, la modification de son signal entraîne une diminution de la vitesse du moteur du côté du capteur, donc un pivotement du robot et son recentrage par la suite.

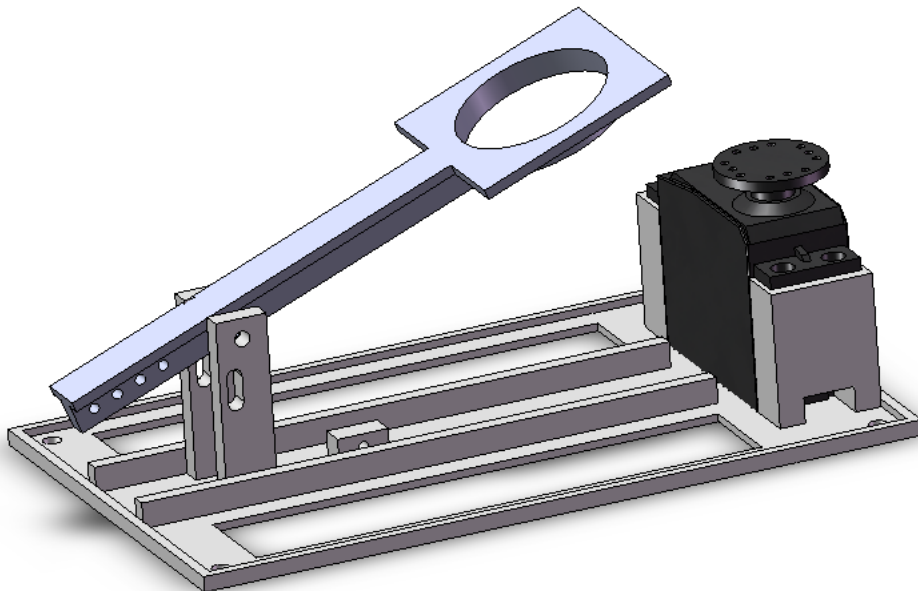


Les moteurs que nous utiliserons seront des moteurs à courant continu (MCC). C'est le moteur électrique classique. Il existe un lien proportionnel entre la vitesse de rotation et la tension à ses bornes, et entre courant et couple. Cela nous permettra de pouvoir ralentir un moteur par rapport à l'autre et ainsi effectuer un virage.

### b) Etude du système de projection de la balle

Nous allons mettre au point un lanceur qui aura la forme d'une catapulte. Nous l'étudierons plus en détail dans une partie ultérieure du dossier.

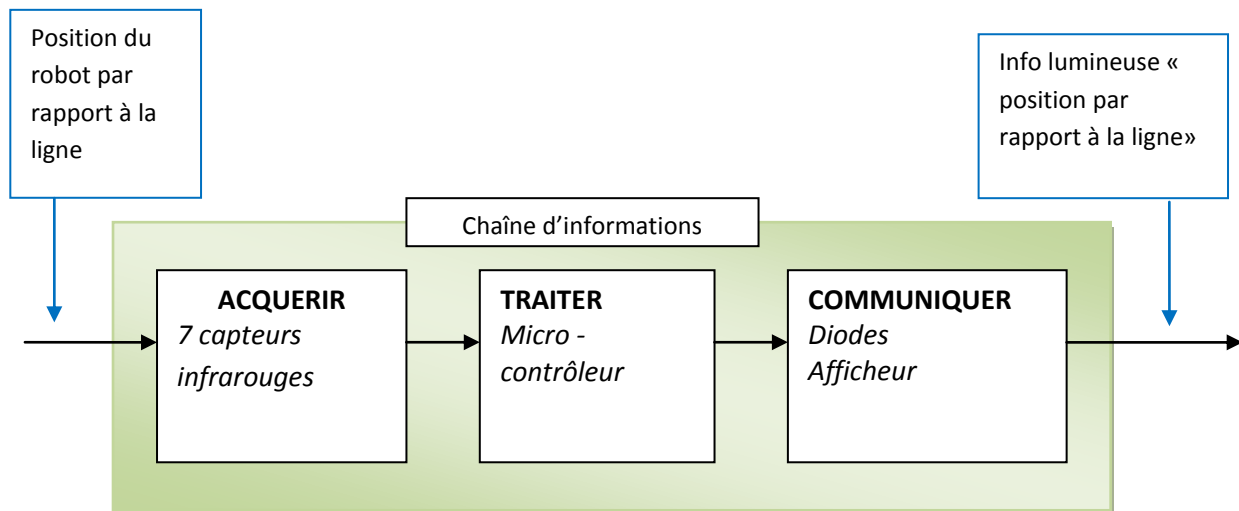
Voici une représentation de notre futur lanceur avec le logiciel Solidworks.





#### **1.4 Schéma Fonctionnel du robot suiveur de ligne**

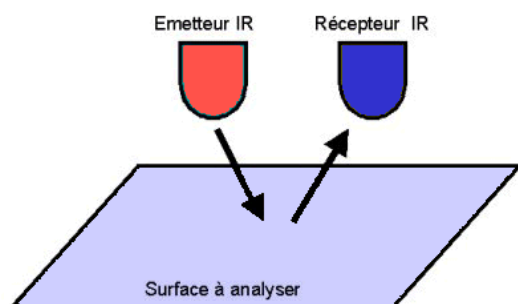
## 2. PARTIE GENIE ELECTRIQUE DU SYSTEME



### 2.1 Analyse de la fonction «ACQUERIR »

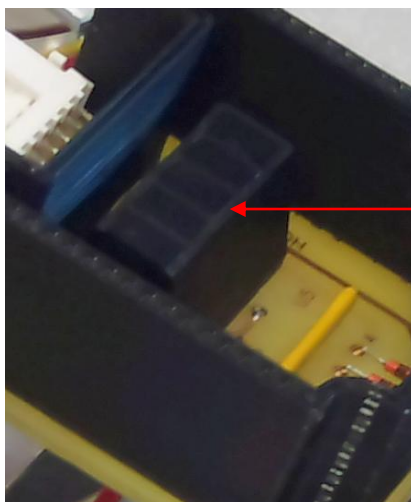
Notre robot détecte sa position par rapport à la ligne noire tracée sur la piste grâce à cinq capteurs d'infrarouges.

Chaque capteur est composé d'un émetteur et d'un récepteur.



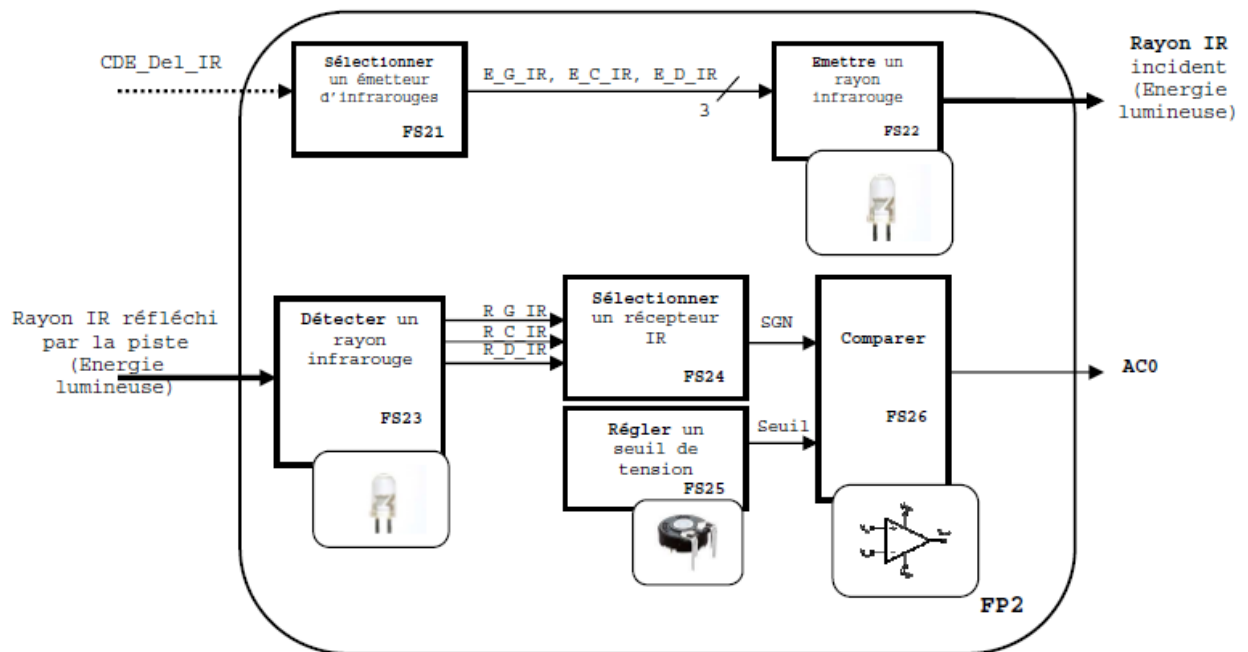
Un rayon infrarouge émis est réfléchi par la piste blanche alors qu'il est absorbé par la ligne noire.

Nous avons choisi d'utiliser un capteur intégré pour notre robot car celui-ci est facile à réaliser et que la piste ne sera pas bosselée, car si cela aurait été le cas, la distance stricte définie par le constructeur nous aurait handicapée.



Les capteurs

## Etude structurelle de « Emettre un faisceau lumineux infrarouge »

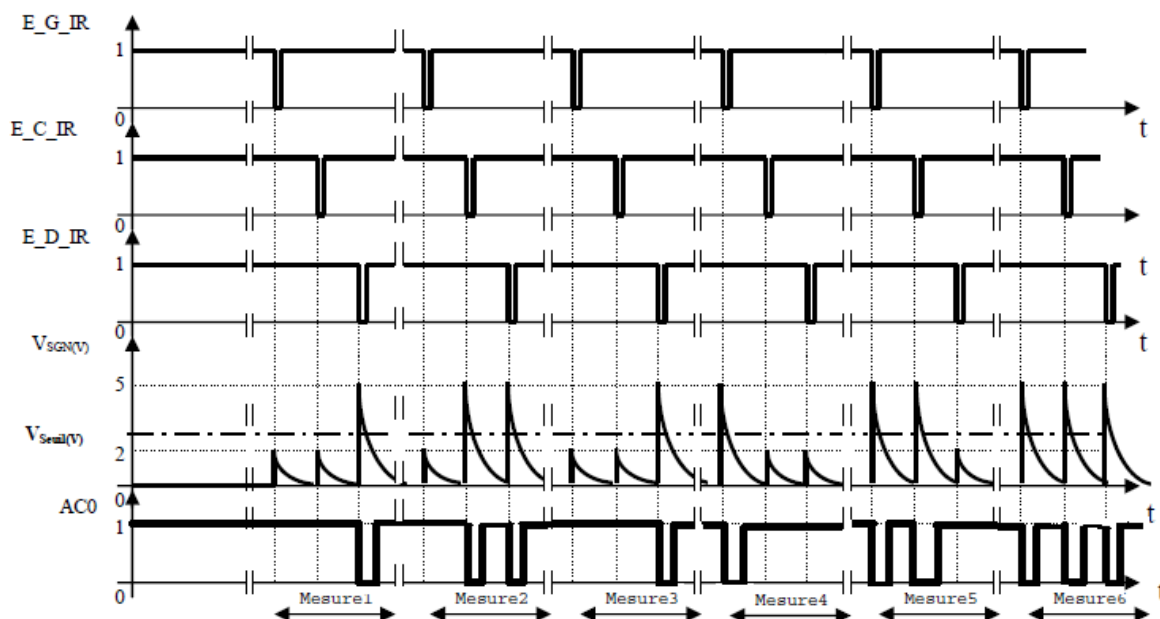


Les signaux de commande des LED infrarouges sont identifiés par  $E_G\_IR$ ,  $E_D\_IR$  et  $E_C\_IR$ . On utilise le multiplexage, ce qui veut dire que l'on fait passer plusieurs informations à travers un seul support de transmission. En commandant les LED chacune leur tour, on s'assure que le phototransistor IR qui lui est associé délivre une information représentative de la position du capteur par rapport à la ligne.

Les signaux issus des récepteurs IR sont transmis, l'un après l'autre car ils sont multiplexés, sur une même ligne. Ceux-ci sont représentés par  $V_{sgn}$  sur le chronogramme ci-après.



Dans le but de savoir si l'on se trouve sur la piste blanche ou sur la ligne noire, on utilise un seuil et un comparateur. Nous prendrons  $V_{seuil} = 2V$ . Cette valeur a été déterminée après des mesures. Nous avons obtenus le gabarit en annexe 1. Cette tension servira de limite entre le noir et le blanc.



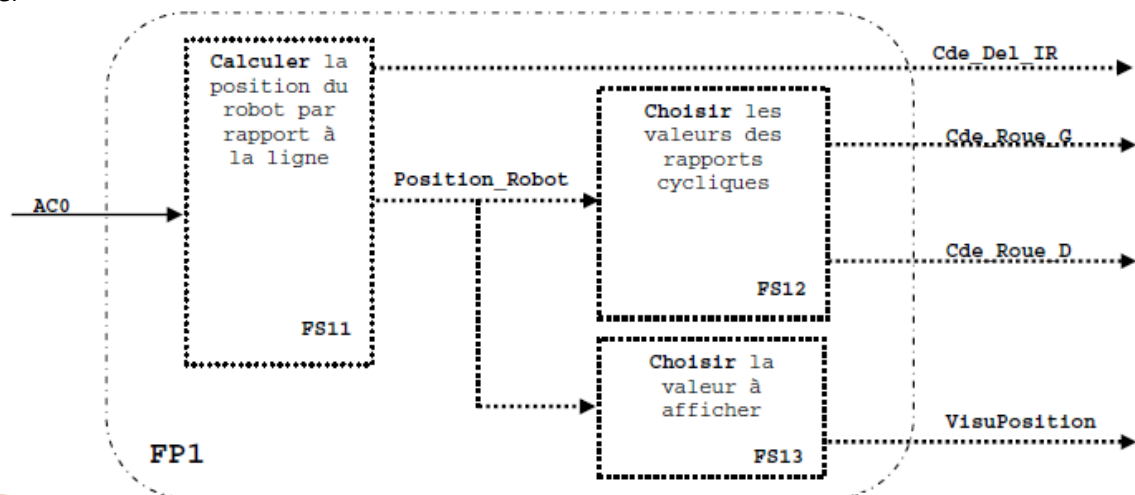
Dans ce cas ci-dessus, les différentes positions du robot correspondantes aux mesures sont :

Mesure	Position
1	Gauche
2	Gauche +
3	Gauche
4	Droite
5	Droite +
6	Hors Ligne

Le gabarit obtenu précédemment correspond à une piste particulière. En effet, nous avons ré effectué des mesures avec un éclairage plus ou moins fort ainsi qu'une ligne faite de différentes façons : une ligne peinte, de scotch ou imprimée. (Voir annexe 1 bis)

## 2.2 Etude de la fonction « TRAITER »

La fonction « traiter » s'organise autour de trois fonctions secondaires : « calculer la position du robot par rapport à la ligne », « choisir les valeurs des rapports cycliques » et « choisir la valeur à afficher »



### a) « calculer la position du robot par rapport à la ligne »

Le tableau en annexe 2 permet de montrer l'évolution des variables traitées par l'algorithme dans cette même annexe. La première colonne donne les résultats obtenus lors du premier parcours de l'algorithme. A la fin du traitement, c'est-à-dire que la boucle a été parcourue trois fois, la variable Position\_Robot contient un chiffre représentatif de la position du robot par rapport à la ligne.

Notre robot est équipé de cinq capteurs d'infrarouges. Ces capteurs sont connectés à un port d'entrées sorties de microcontrôleur. Sur la carte SSI le PORT A est réservé à la commande des capteurs d'infrarouges. L'algorithme nous permet de calculer la position du robot par rapport à la ligne à partir de l'information délivrée par un capteur comprenant cinq cellules.

### b) « choisir les valeurs des rapports cycliques »

Pour faire varier la vitesse de rotation d'un moteur à courant continu, il suffit de modifier la valeur moyenne de la d.d.p présente à ses bornes en le commandant par un signal dit PWM ou MLI.

Les chronogrammes ci-dessous sont extraits des datasheets ATMEL.

Nous avons trouvé que  $U_{mot} = OC1x$  car le transistor est supposé parfait.

Après de petits calculs, nous avons trouvé que  $OCR1x = (1-\alpha) * OCR1C$

Cette relation permet de déterminer les valeurs à envoyer pour obtenir les rapports cycliques souhaités. Ces coefficients nous permettront de corriger la trajectoire du robot en fonction de sa position. Toutes ces valeurs sont répertoriées dans le tableau suivant :

$\alpha$ (%)	0	5	10	15	20	25	30	35	40	45	50	55	60
OCR1x	255	243	230	217	204	192	179	166	153	141	128	115	102

65	70	75	80	85	90	95	100
90	77	64	51	39	26	13	0

Ces corrections sont représentées sur le graphe des liaisons et dans le programme en annexe 3.

### c) « Choisir la valeur à afficher »

Pour choisir la valeur à afficher, il nous faut affecter différents codes à la variable VisuPosition.

L'algorithme ainsi que le programme en langage C en annexe 4 correspondent à la commande des LED en fonction de la position du Robot.

## 2.3 Etude de la fonction « COMMUNIQUER »

La fonction « communiquer » est matérialisée par cinq LED rouges placées sur la carte. L'état d'une LED permet de vérifier que les capteurs détectent la ligne.

Une LED éclairée indique que le capteur correspondant voit la ligne. Elle est éteinte dans le cas contraire.

Les émetteurs d'infrarouge et les LED rouges sont placés sur le même port d'entrées/ sorties. Pour déterminer la valeur de la variable VisuPosition, il faut prendre en compte le fait que les émetteurs sont commandés avec un état logique 0.

Les différentes positions du robot nous amènent au tableau suivant :

<b>Hors_ligne</b>	<b>0b11111111</b>	-   -   -   -   -
<b>Visu_Gauche_3+</b>	<b>0b11111110</b>	*   -   -   -   -
<b>Visu_Gauche_2+</b>	<b>0b11111100</b>	*   *   -   -   -
<b>Visu_Gauche_+</b>	<b>0b11111000</b>	*   *   *   -   -
<b>Visu_Gauche</b>	<b>0b11110000</b>	*   *   *   *   -
<b>Visu_Centre</b>	<b>0b11100000</b>	*   *   *   *   *
<b>Visu_Droite</b>	<b>0b11100001</b>	-   *   *   *   *
<b>Visu_Droite_+</b>	<b>0b11100011</b>	-   -   *   *   *
<b>Visu_Droite_2+</b>	<b>0b11100111</b>	-   -   -   *   *
<b>Visu_Droite_3+</b>	<b>0b11101111</b>	-   -   -   -   *
<b>Hors_ligne</b>	<b>0b11111111</b>	-   -   -   -   -



## 2.4 Un servomoteur pour piloter la catapulte.

Un servomoteur est une interface entre le monde électronique (programmes) et le monde mécanique (catapulte). En effet, c'est lui qui transmet nos « ordres » afin de propulser la balle dans le panier.

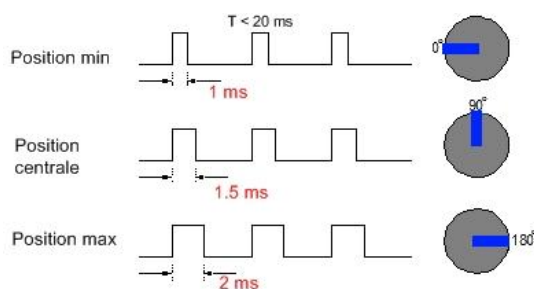
Les servomoteurs sont constitués :

- D'un moteur à courant continu
- D'un réducteur à la sortie du moteur dans le but d'avoir un couple
- Un potentiomètre à résistance variable en fonction de la position angulaire de celui-ci
- D'un programme pour contrôler sa position angulaire



Il existe trois manières de piloter un servomoteur.

En effet, nous aurions pu utiliser le PWM ou un TIMER.



Seulement, pour des raisons de simplicité et de temps, nous avons favorisé un programme avec des fonctions `delay_ms()`.

Notre programme en annexe 5 permet à notre catapulte d'être libérée au bon moment grâce à un capteur sur notre robot.

### **3. PARTIE GENIE MECANIQUE DU SYSTEME**

#### **3.1 Encombrement**

##### **a) Contraintes et cahier des charges**

Pour commencer notre projet, nous avons reçu le cahier des charges donné par les organisateurs du concours. Celui-ci comporte les dimensions maximales prise par le robot, soit :

- Longueur maximale 300 mm
- Largeur maximale 200 mm
- Hauteur maximale 200 mm

Il indique aussi que les robots doivent utiliser une source d'énergie autonome et doivent posséder un interrupteur marche/arrêt.

Nous avons pris connaissance de l'annexe du cahier des charges, qui contiennent les différentes contraintes afin de réaliser le châssis.

##### **- Des contraintes dimensionnelles :**

La carte SSI doit être placée de façon à ce que l'afficheur soit lisible dans le sens du déplacement. Sur cette même carte, le réglage du seuil noir/blanc, les points test, les connecteurs de programmation ainsi que les connecteurs d'alimentation doivent tous être accessibles.

La carte alimentation doit se trouver au plus près des accumulateurs. Le connecteur de programmation et le cavalier « sécurité » doivent être accessibles.

##### **La dimension des câbles :**

- Nappe capteur ligne 18-19 cm
- Câble  $\mu C$  -> Câble moteur 25-26 cm
- Câble moteur 18-19 cm

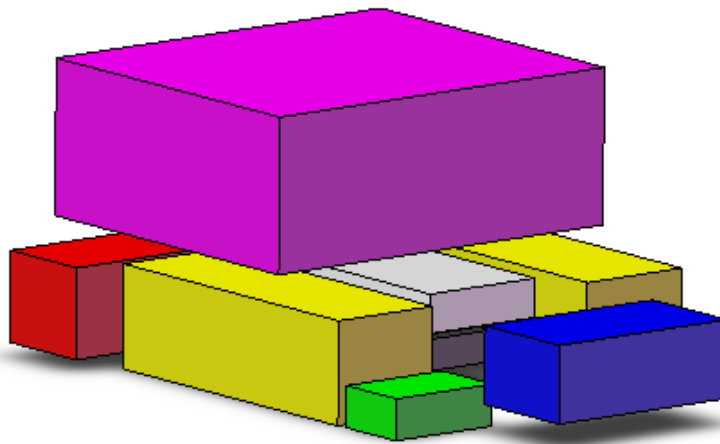
##### **- Les erreurs des années précédentes à ne pas reproduire :**

Les erreurs à ne pas reproduire sont les suivantes :

- les roues touches le châssis
- les accès aux différents réglages ne sont pas accessibles
- les différents connecteurs tels que le connecteur de programmation ne sont pas accessibles
- Accumulateur trop loin de la carte alimentation
- Porte fusible trop loin de l'interrupteur
- La charge est répartie que sur un côté
- Aucun passage de câbles n'est prévu

### b) 1<sup>er</sup> encombrement

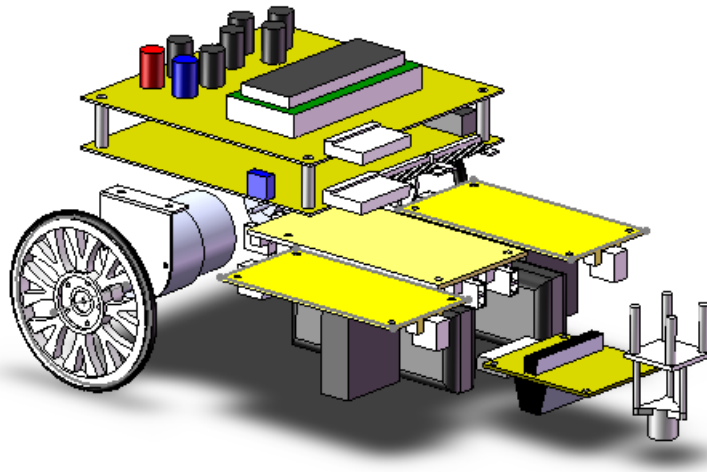
Dans un premier temps, nous devons imaginer la forme globale de notre robot. Une fois celle-ci délimitée, on fait ce qu'on appelle un encombrement. Celui-ci consiste à créer des blocs de la tailles des éléments a placé pour ensuite les disposer dans l'espace à l'aide de SolidWorks. Les disposer ainsi sert à savoir à l'avance si nos pièces placé tiennent dans l'espace contraint. Il sert de plus à voir où placé les objets pour répondre aux contraintes, à savoir les contraintes de poids, de proximité (entre les accumulateurs et la carte alimentation par exemple) ...



Dans le cas de notre encombrement (ci-dessus), on a placé la plupart du poids en bas du robot. Laissant place au dessus à la **carte afficheur** et une large place au lanceur (non représenté car les dimensions n'était pas encore connus). Nous avons donc en bas, les éléments suivant : **2 moteurs**, **2 cartes moteurs**, **carte capteur**, carte alimentation, **Carte cible**, **2 accumulateurs**.

### c) 2<sup>ème</sup> encombrement

Le second encombrement est beaucoup plus précis que le précédent. En effet, au lieu d'avoir des cubes aux dimensions des pièces, on a directement les pièces. Il permet donc de visualiser le robot sur SolidWorks plus clairement



La disposition des éléments est la même que lors du 1<sup>er</sup> encombrement mais avec des pièces plus détaillées et avec les roues ainsi que la bille à l'avant en plus.

### **3.2 Pièces fabriquées**

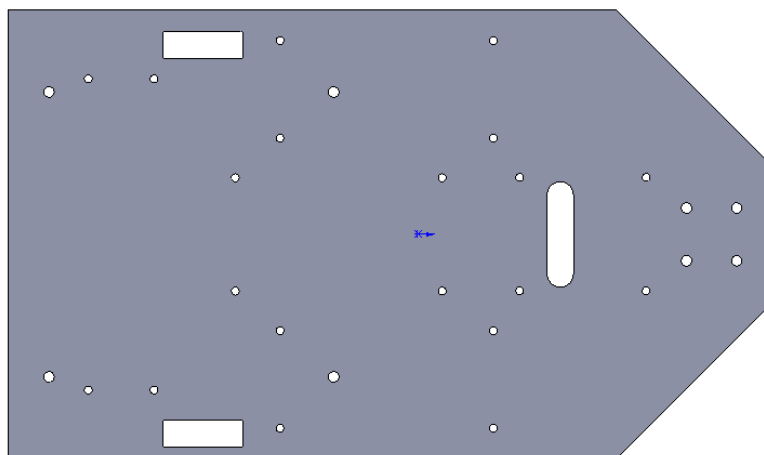
#### a) Le châssis

La première pièce que nous avons fabriquée est le châssis. Pour cela nous disposons d'une plaque d'aluminium de la taille d'une feuille A4 d'une épaisseur de 1 mm.

Pour notre châssis, nous nous sommes basés sur le 2<sup>ème</sup> encombrement. On a pu délimiter une taille et une forme de notre plaque. L'encombrement est particulièrement intéressant au moment de percer la plaque puisqu'il nous donne le placement précis des différents perçages à prévoir sur la plaque en fonction des vis que nous allons utiliser.

Concevoir le châssis précisément sur SolidWorks est primordial.

Nous avons aussi prévu des passages de fils à plusieurs endroits grâce à l'encombrement.



*Nous avons opté pour une forme simple rectangulaire, « cassée » à l'avant*

### b) Les roues

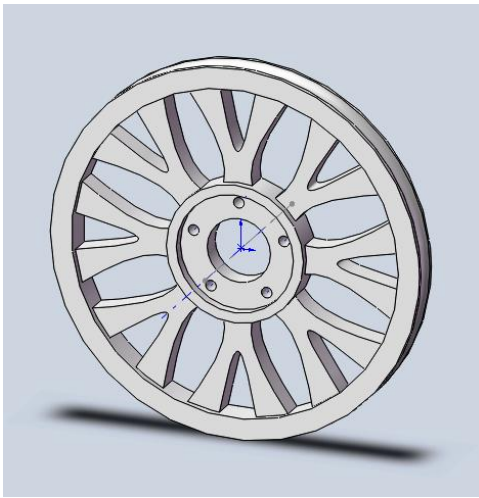
Cette année, grâce à l'imprimante à 3 dimensions nous avons eu la possibilité de créer notre propre roue. Pour plus d'informations concernant l'imprimante 3D, se référer à l'annexe 8 à la fin du dossier.

Afin d'optimiser la vitesse du robot au maximum, les moteurs étant donnés, il y avait une vitesse angulaire maximale pour les roues. Pour que le robot puisse aller plus vite avec la même vitesse angulaire, il fallait augmenter le diamètre de ses roues

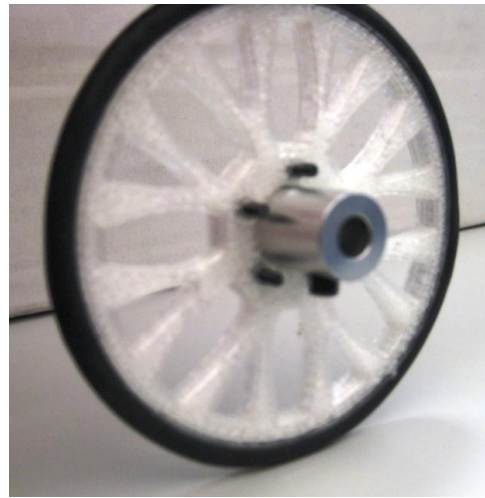
Les roues seront donc plus grandes que celles des années précédentes, et pour une même vitesse du robot elles demanderont moins de puissance aux moteurs.

Toute fois nos roues doivent être bien centrées par rapport à notre plaque pour ne pas avoir de problème de trajectoire.

De plus nous devons faire attention à ce que nos roues ne touchent pas le châssis.



*Réalisation de la roue sur SolidWorks*



*Voici l'assemblage avec le moyeu et le joint.*

### c) Le cache capteur

Nous avons dû isoler notre carte capteur de la lumière. Les capteurs étant à infrarouge, doivent être protégés de toutes les perturbations possibles de la lumière pour optimiser la détection de la ligne noire, et ainsi avoir la trajectoire la plus précise possible dans toutes les environnements lumineux possibles.

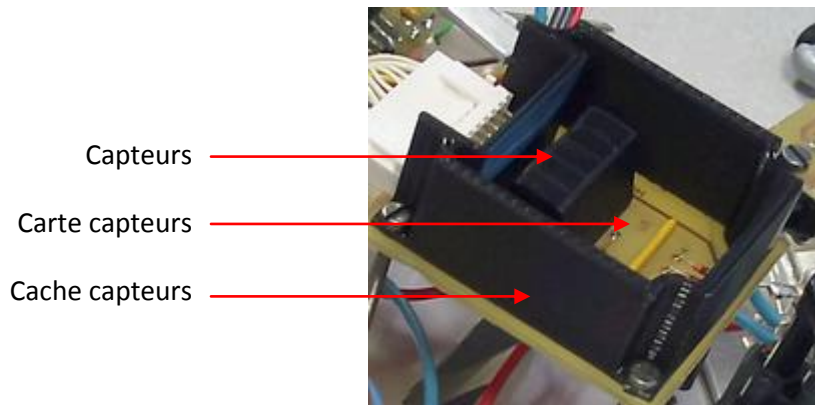
Nous avons donc élaboré une pièce pour les protéger de la lumière.

Notre pièce devait respecter les formes de la plaque du capteur (dimension, encombrement due aux composants...)

De plus, les capteurs doivent se situer à plus ou moins 5 millimètres du sol pour un bon fonctionnement, donc nous étions contraints de ne pas dépasser une certaine distance par rapport à au sol pour que le cache capteurs ne touche pas le sol.

Nous avons alors élaboré une pièce à partir de SolidWorks, pour pouvoir faire une simulation de notre solution sur la carte capteur, grâce à la fonction assemblage.

Puis grâce à une imprimante 3D nous avons pu créer notre propre pièce. Néanmoins nous devons faire attention à ce que notre pièce soit réalisable par l'imprimante.



*Assemblage de la Carte et du cache capteur*

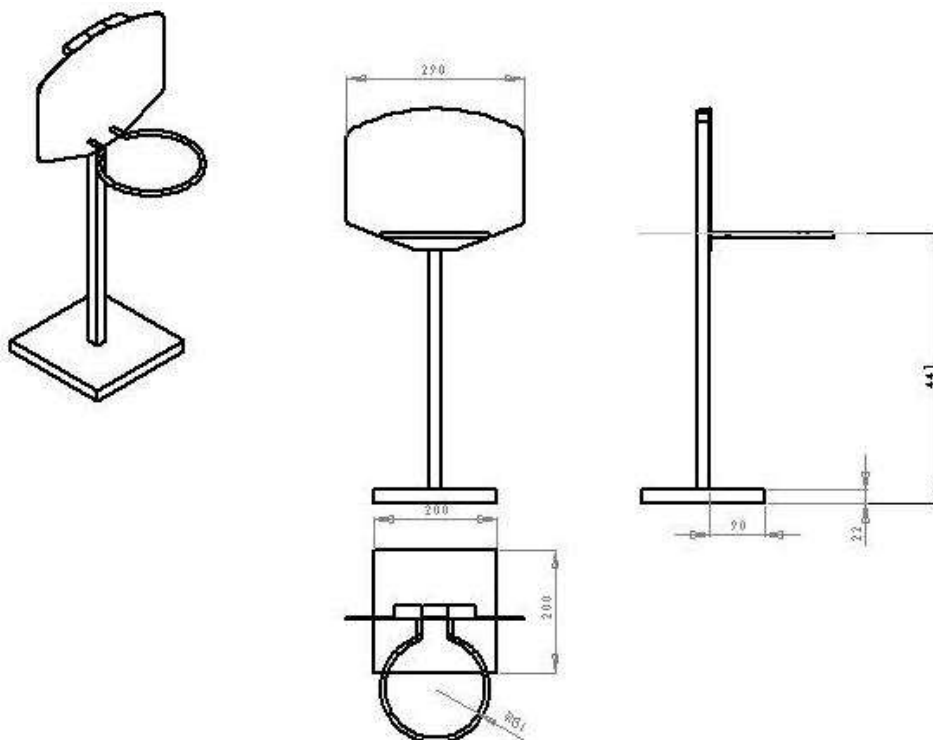
#### d) Le lanceur

Lors du concours, nous participerons à l'épreuve du lanceur.

Le but du lanceur est d'envoyer une balle d'YGOLF dans un panier de basket se situant à 1650 mm du robot lors du lancé. Une seule balle est autorisée par robot.

Le diamètre de la balle est de 42mm

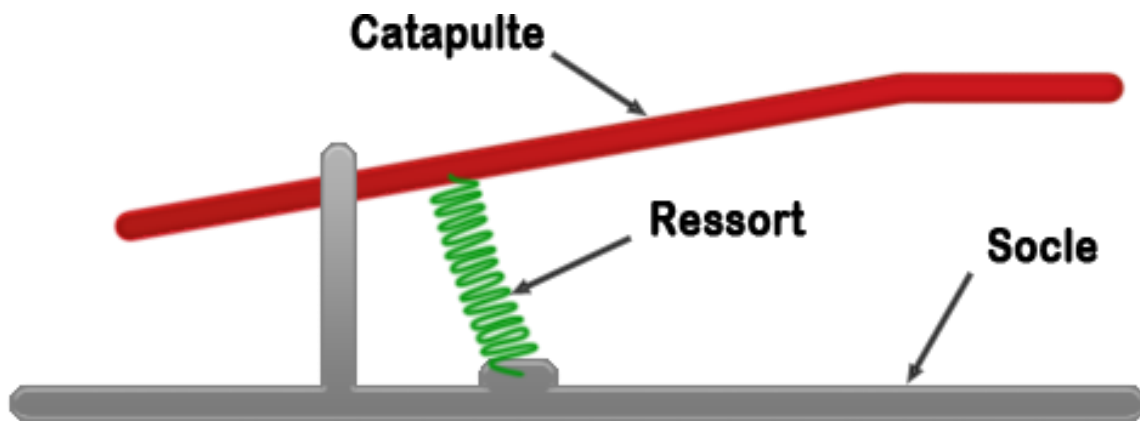
Les dimensions du panier son donné ci-dessous :



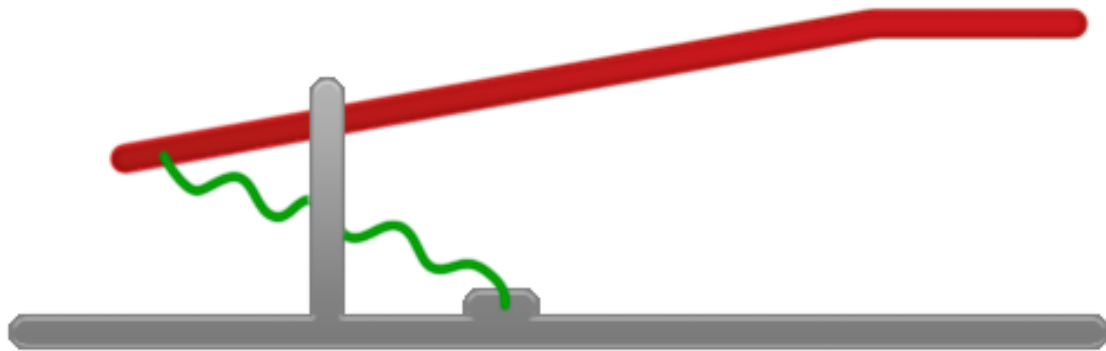
Nous avons décider de fabriquer notre lanceur sous la forme d'une catapulte à l'aide d'un ressort. Nous avons le choix entre 3 dispositions différentes du ressort :

- La première : le ressort est placé en dessous de la catapulte et la pousse

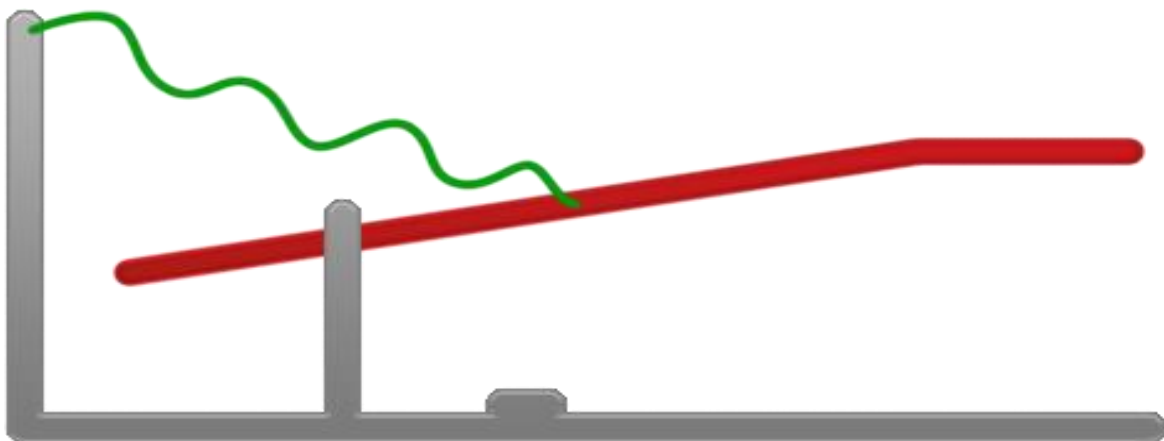




- La deuxième : le ressort est toujours placé en dessous mais tire la catapulte



- La troisième : le ressort est cette fois ci placé au dessus de la catapulte et la tire



Pour le lanceur de notre robot, nous avons retenu la seconde solution que nous avons trouvé plus pratique à concevoir.

#### - *Le réglage :*

Pour le réglage d'un lanceur, plusieurs solutions s'offre à nous.

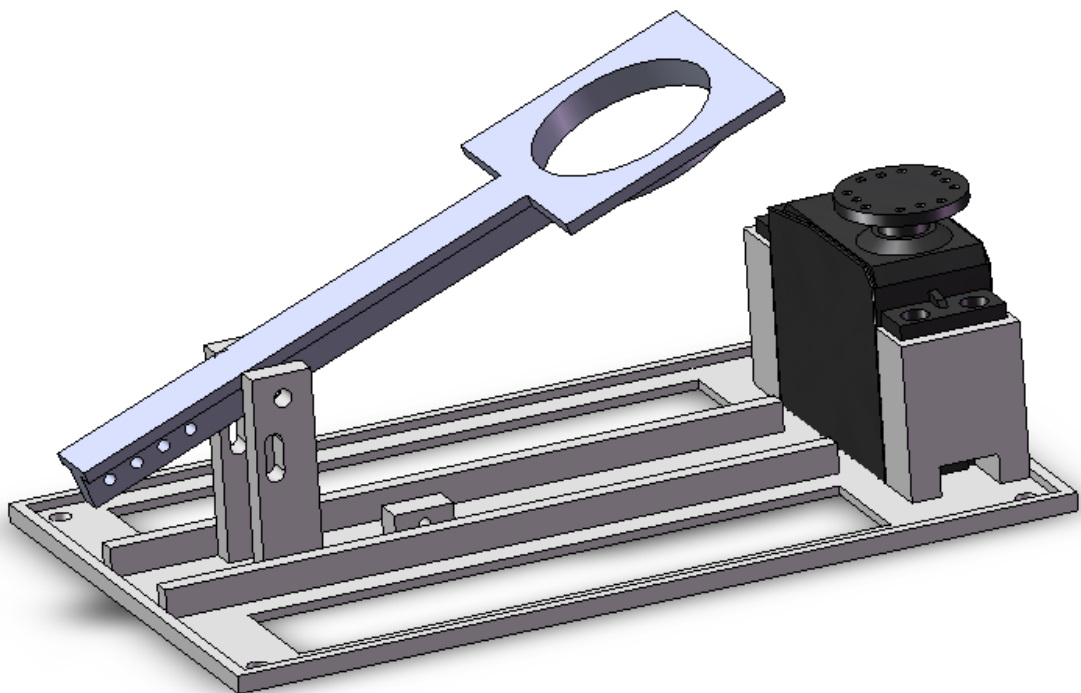
On peut tout d'abord agir sur les ressorts en prenant des pièces de longueurs et de raideurs différentes ou alors modifier les points d'attaches du ressort sur la catapulte (il sera plus ou moins tendu lorsque le lanceur est en position initiale).

On peut ensuite modifier la taille de la catapulte avec des tiges plus ou moins longues pour intervenir sur la portée du lanceur.

On peut enfin, en dernier réglage, positionner une butée à des endroits différents ce qui va agir sur l'angle de la position finale de la catapulte.

Ainsi, lors de la conception SolidWorks, nous avons voulu garder un maximum de réglage possible après la fabrication.

- **Notre lanceur :**

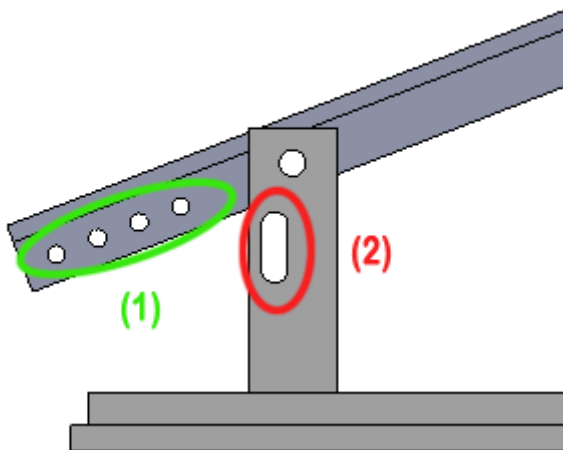


*Modèle sous SolidWorks*

Il est donc avec le ressort qui est placé en dessous qui tire la catapulte et le moteur placé derrière cette dernière.

- *Les réglages disponibles après la fabrication :*

- On peut changer le type de ressort
- possibilité de régler le point d'accrochage du ressort a la catapulte grâce à 4 percages disponible (1)
- possibilité de régler la position de la butée qui est dans notre cas une vis qui bloquera la catapulte (2)



Le principal problème rencontré est celui de la solidité. Pour ne pas que notre lanceur se plie ou se casse, il a fallu prévoir des renforcements en ajoutant des légères bordures pour solidifier le tout.

Notre pièce a ensuite été réalisée par l'imprimante 3D :

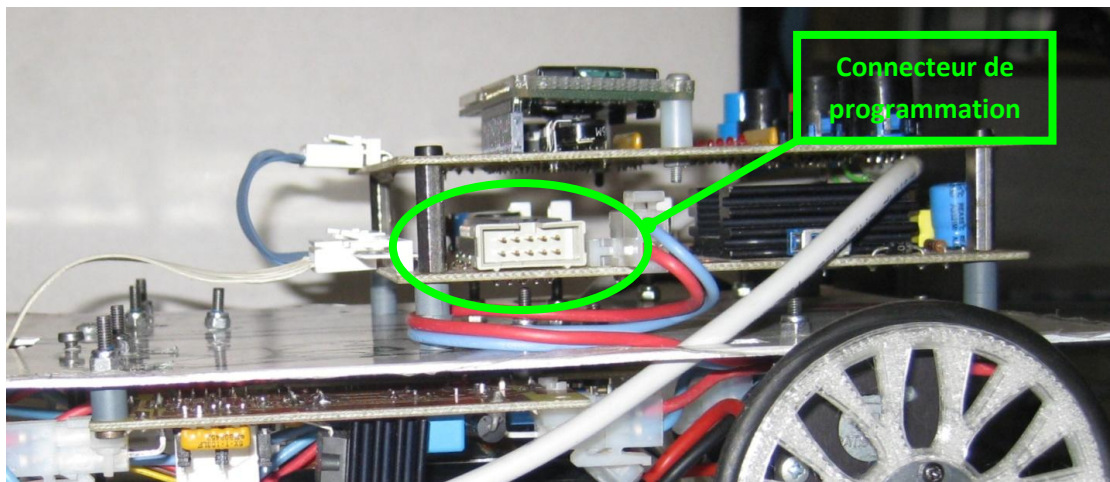


### 3.3 Assemblage

Lors de l'assemblage nous avons plusieurs contraintes à respecter.

En effet nos composants devaient dans un premier temps tous tenir sur la plaque, mais nous avons une certaine disposition à respecter.

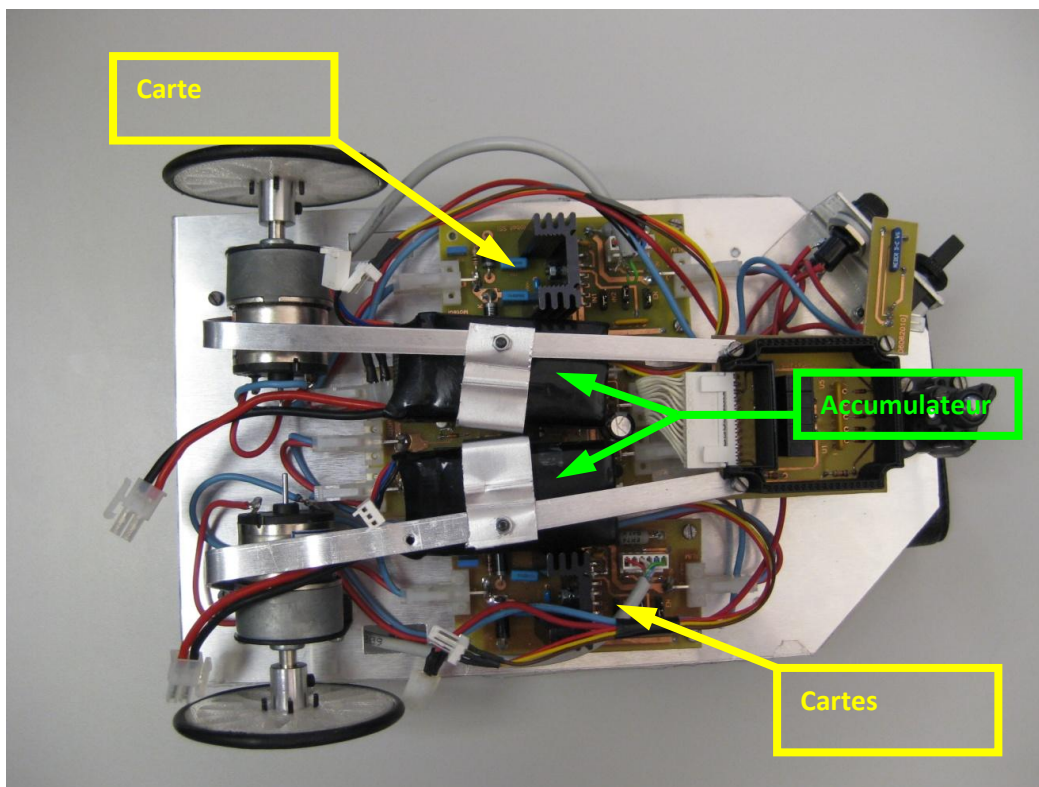
Il fallait que le connecteur de programmation soit accessible pour pouvoir inclure le programme dans le microcontrôleur.



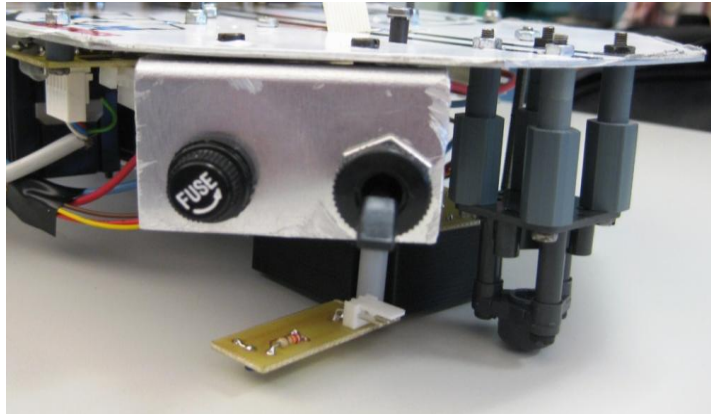
Nous l'avons donc laissé sur sa carte d'origine et nous l'avons rendu accessible.

Les accumulateurs doivent être placés le moins loin possibles de la carte d'alimentation car les câbles ne sont pas très longs.

Nous les avons placé juste en dessous des cartes, et grâce à un système nous évitons que ces derniers ne puissent tomber.

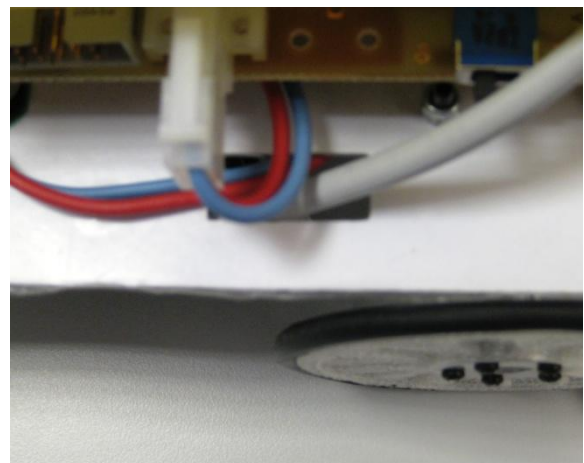
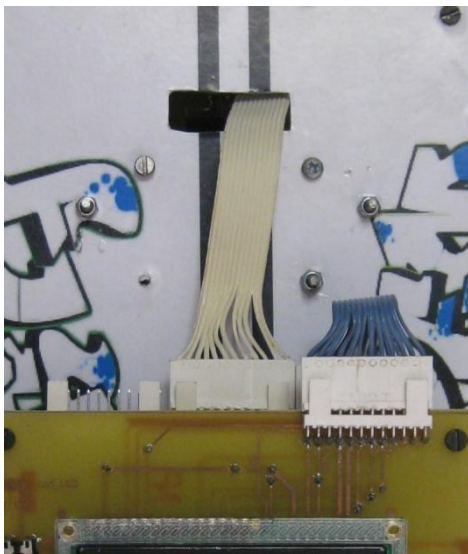


Il fallait que le porte fusible soit placé près de l'interrupteur.  
Pour une raison pratique, il était préférable de le placer vers l'avant du robot, pour que l'on puisse le déconnecter en fin de course.



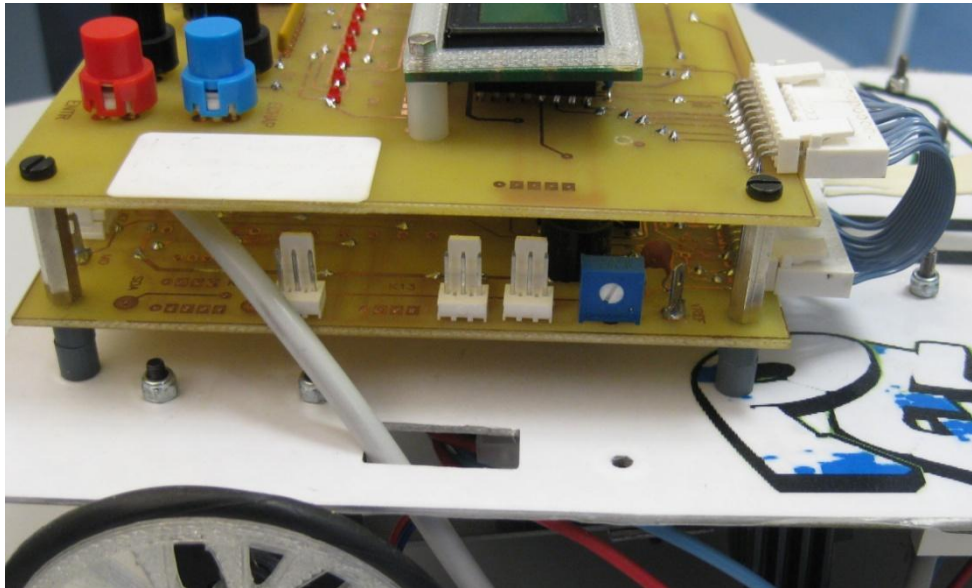
*Le porte fusible et l'interrupteur sont à l'avant du robot et près l'un de l'autre.*

Nous avons aussi prévu des passages pour les câbles pour éviter d'avoir des fils qui partent partout.

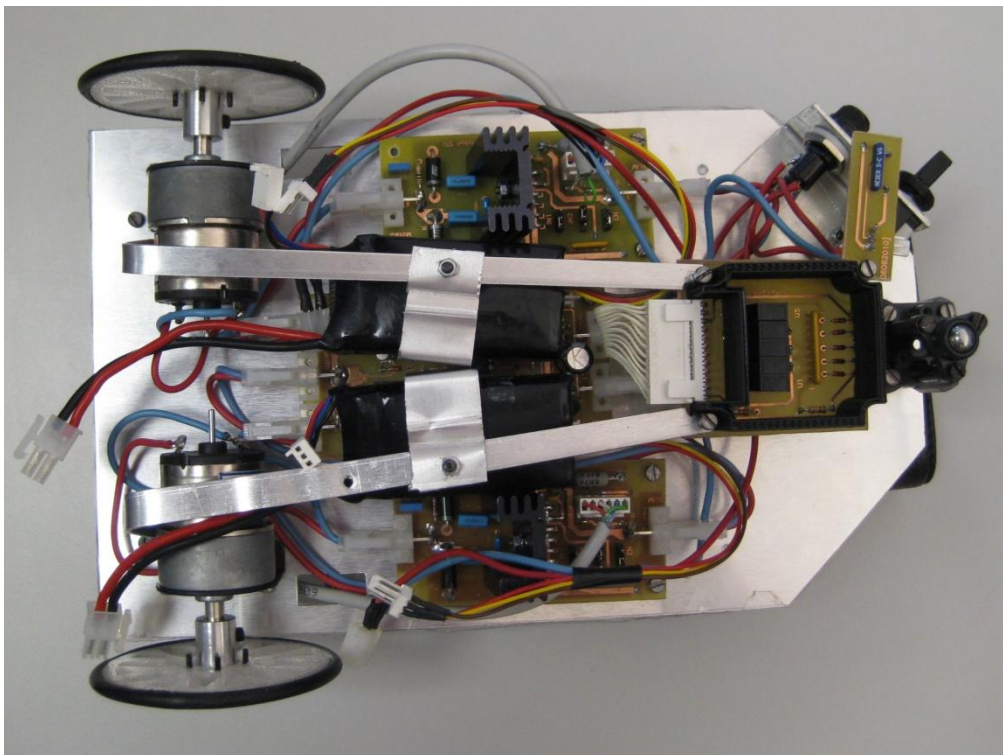


Il fallait que le groupe en Génie électrique ait accès au comparateur, pour pouvoir effectuer leurs réglages. Nous avons placé la carte afficheur au dessus du robot pour ne pas qu'ils soient gênés par d'autres éléments.





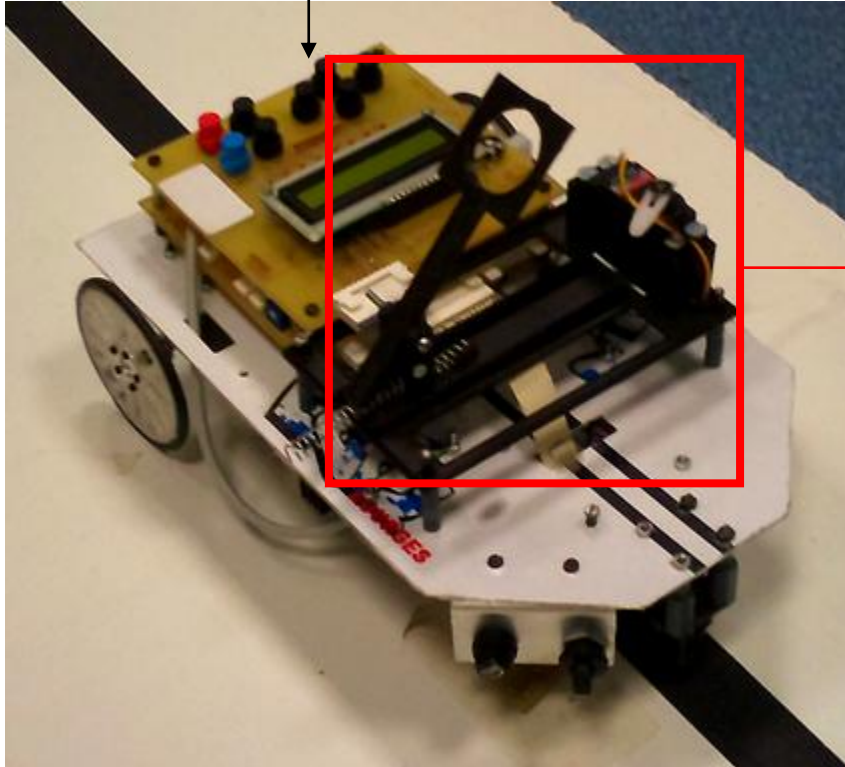
Ensuite, nous avons essayé de répartir le poids de façon équilibré pour pas que notre robot ait des problèmes d'équilibre lors du démarrage dans les virages : sur la piste.



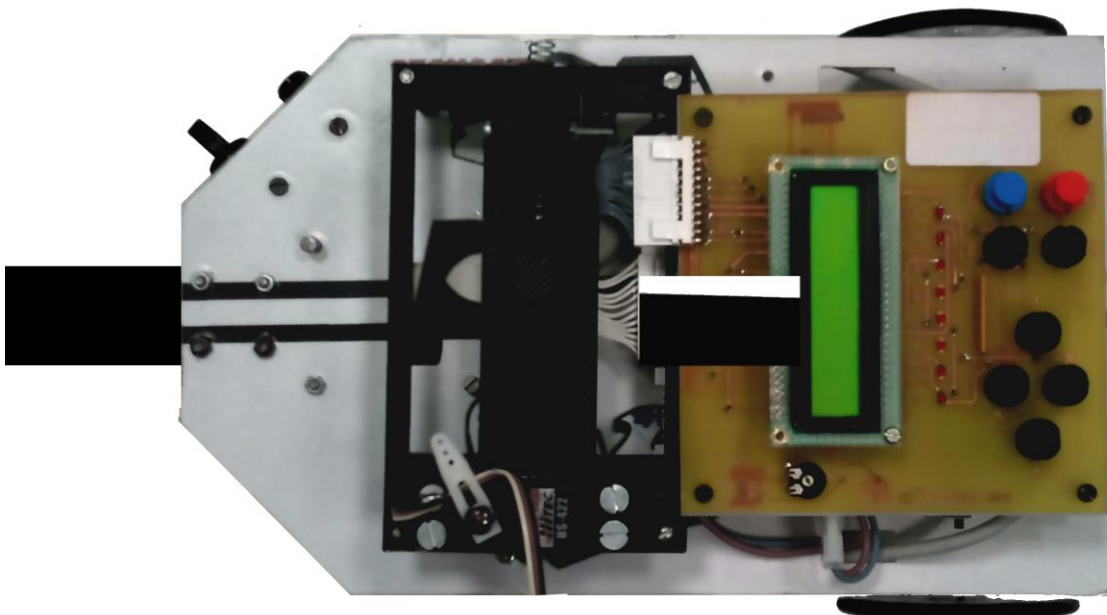


Voici notre robot :

Partie programmation  
avec l'afficheur , les  
boutons poussoirs.

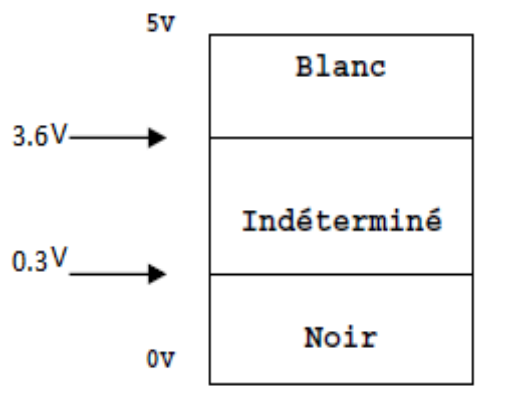


Notre lanceur



## 4. ANNEXES

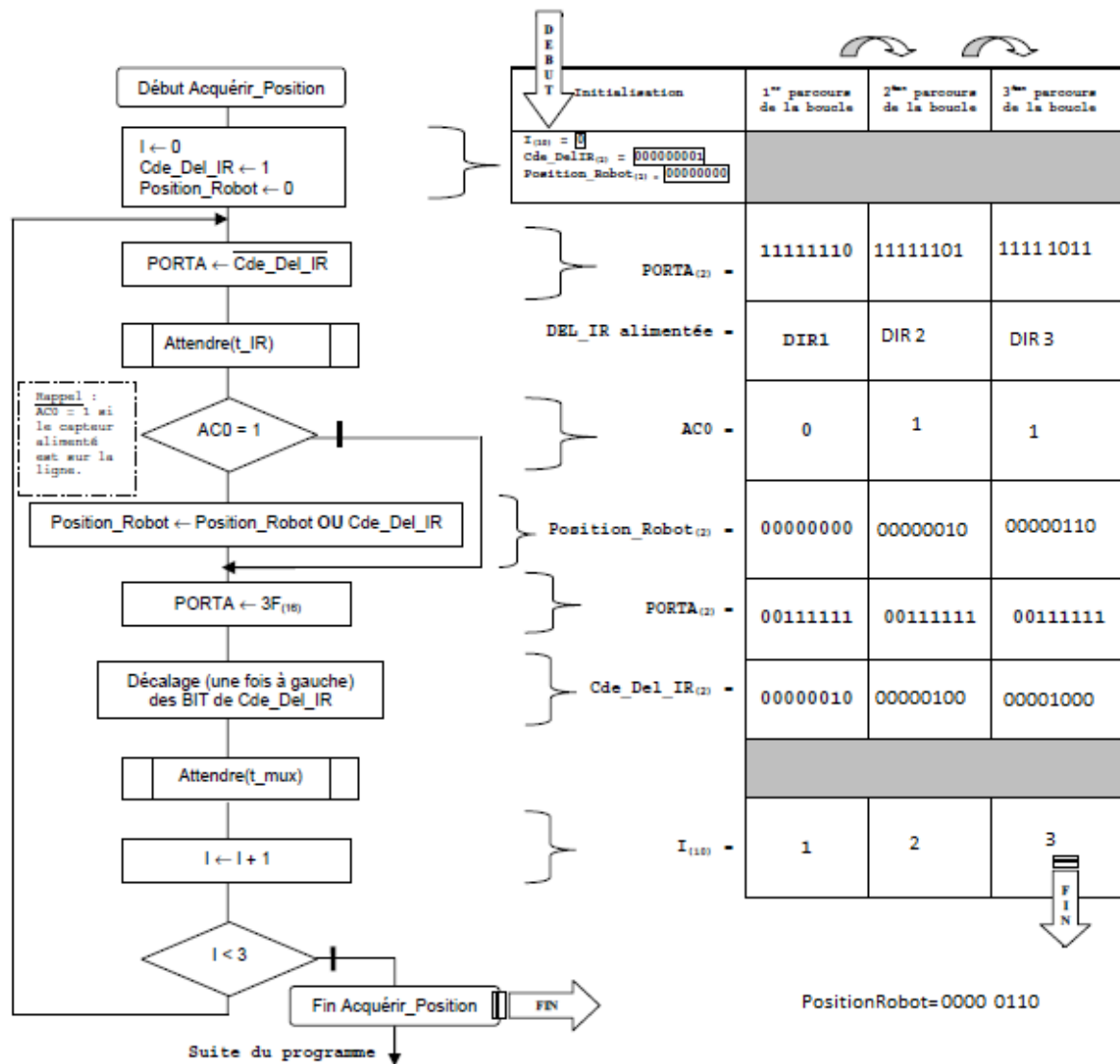
### Annexe 1 : Gabarit pour un éclairage ambiant et une piste peinte



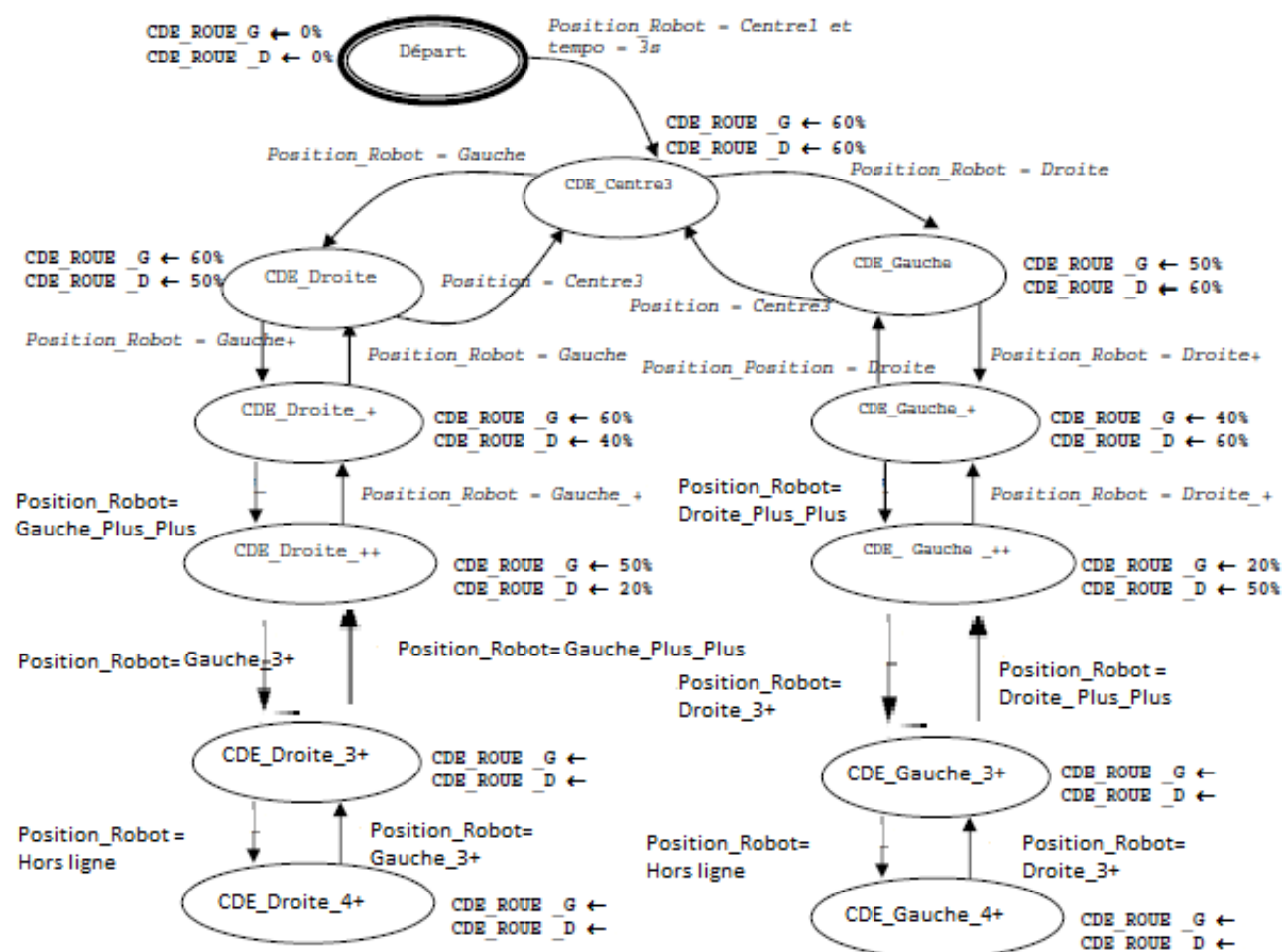
### Annexe 1bis : Gabarit en fonction de chaque cas

Nature du noir	Eclairage	
	« Normal »	« Fort »
Peinture mat	5V 3.6 V → Blanc 0.3 V → Indéterminé 0V → Noir	5V 4.9 V → Blanc 3.34 V → Indéterminé 0V → Noir
Scotch	5V 4.3 V → Blanc 0.5 V → Indéterminé 0V → Noir	5V 4.8 V → Blanc 2.54 V → Indéterminé 0V → Noir
Encre imprimante (Annexe 2)	5V 3.9 V → Blanc 0.7 V → Indéterminé 0V → Noir	5V 4.36 V → Blanc 2.1 V → Indéterminé 0.8 V → Noir

## Annexe 2 : algorithme et tableau de calculs de position du robot



### Annexe 3 : Graphe des liaisons



```

switch (EtatCdeMot)

{ // Début switch (EtatCdeMot)

    case DEPART: Arret_Moteurs();

        if ((Position_Robot == CENTRE_1) || (Position_Robot == CENTRE_2) || (Position_Robot ==
CENTRE_3))

            {

                delay_ms(3000);

                Active_Timer();

                EtatCdeMot = CDE_CENTRE1;

            }

            else EtatCdeMot = DEPART;

        break;

    case CDE_CENTRE1: EtatCdeMot = CDE_CENTRE3;

        break;

    case CDE_CENTRE2: EtatCdeMot = CDE_CENTRE3;

        break;

    case CDE_CENTRE3: CDE_ROUE_G = _60PC; CDE_ROUE_D = _60PC;

        if (Position_Robot == GAUCHE) EtatCdeMot = CDE_DROITE;

            else if (Position_Robot == DROITE) EtatCdeMot = CDE_GAUCHE;

                else EtatCdeMot = CDE_CENTRE3;

        break;

    case CDE_DROITE: CDE_ROUE_G = _60PC; CDE_ROUE_D = _50PC;

        if (Position_Robot == CENTRE_3) EtatCdeMot = CDE_CENTRE3;

            else if (Position_Robot == GAUCHE_PLUS) EtatCdeMot = CDE_DROITE_PLUS;

                else EtatCdeMot = CDE_DROITE;

        break;

```

```

case CDE_DROITE_PLUS:CDE_ROUE_G = _60PC ; CDE_ROUE_D = _40PC ;

    if (Position_Robot == HORS_LIGNE)EtatCdeMot = CDE_DROITE_2PLUS;

    else if (Position_Robot == GAUCHE) EtatCdeMot = CDE_DROITE;

        else EtatCdeMot = CDE_DROITE_PLUS;

break;

case CDE_DROITE_2PLUS:CDE_ROUE_G = _50PC ; CDE_ROUE_D = _20PC ;

    if (Position_Robot == GAUCHE_PLUS)EtatCdeMot = CDE_DROITE_PLUS;

        else EtatCdeMot = CDE_DROITE_2PLUS;

break;

case CDE_GAUCHE:CDE_ROUE_G = _50PC ; CDE_ROUE_D = _60PC ;

    if (Position_Robot == DROITE_PLUS) EtatCdeMot = CDE_GAUCHE_PLUS;

        else if (Position_Robot == CENTRE_3) EtatCdeMot = CDE_CENTRE3;

            else EtatCdeMot = CDE_GAUCHE;

break;

case CDE_GAUCHE_PLUS:CDE_ROUE_G = _40PC ; CDE_ROUE_D = _60PC ;

    if (Position_Robot == HORS_LIGNE) EtatCdeMot = CDE_GAUCHE_2PLUS;

        else if (Position_Robot == DROITE) EtatCdeMot = CDE_GAUCHE;

            else EtatCdeMot = CDE_GAUCHE_PLUS;

break;

case CDE_GAUCHE_2PLUS:CDE_ROUE_G = _20PC ; CDE_ROUE_D = _50PC ;

    if (Position_Robot == DROITE_PLUS) EtatCdeMot = CDE_GAUCHE_PLUS;

        else EtatCdeMot = CDE_GAUCHE_2PLUS;

break;

```



#### **Annexe 4 : affectation de la variable VisuPosition**

```
// Connexions des LED Visu et IR au port A
//-----
// Del_D reliée à PA3 (D1) E_D_IR reliée à PA0 (DIR1)
// Del_C reliée à PA4 (D2) E_C_IR reliée à PA1 (DIR2)
// Del_G reliée à PA5 (D3) E_G_IR reliée à PA2 (DIR3)
// Mots de commande des LED Visu (- => Del éteinte * => Del éclairée)
//-----
// Etat des Dels de visualisation
//-----
// Del_G | Del_C | Del_D VisuPosition(16) Remarque
//-----
#define Visu_Hors_ligne 0b00111111 // off | off | off 3F - - -
#define Visu_Droite_Plus 0b00011111 // on | off | off 1F * - -
#define Visu_Centre1 0b00101111 // off | on | off 2F - * -
#define Visu_Droite 0b00001111 // on | on | off 0F * * -
#define Visu_Gauche_Plus 0b00110111 // off | off | on 37 - - *
#define Visu_Centre2 0b00010111 // on | off | on 17 * - *
#define Visu_Gauche 0b00100111 // off | on | on 27 - * *
#define Visu_Centre3 0b00000111 // on | on | on 07 * * *
//-----
```

Début

Selon PositionRobot ;

```
Centre1 : VisuPosition ← visu_centre1 ;

Centre2 : VisuPosition ← visu_centre2 ;

Centre3 : VisuPosition ← visu_centre3 ;

Centre4 : VisuPosition ← visu_centre4 ;

Centre5 : VisuPosition ← visu_centre5 ;

Droite : VisuPosition ← visu_Droite ;

Droite+ : VisuPosition ← visu_Droite_Plus ;

Droite++ : VisuPosition ← visu_Droite_Plus_Plus ;

Droite3+ : VisuPosition ← visu_Droite_3Plus ;

Droite4+ : VisuPosition ← visu_Droite_4Plus ;

Gauche : VisuPosition ← visu_Gauche ;

Gauche+ : VisuPosition ← visu_Gauche_Plus ;

Gauche++ : VisuPosition ← visu_Gauche_Plus_Plus ;

Gauche3+ : VisuPosition ← visu_Gauche_3Plus ;

Gauche4+ : VisuPosition ← visu_Gauche_4Plus ;
```

Fin selon

FIN

## Annexe 5 : programme servomoteur

```
while(1)
{
    if (Entree==0)
    {
        Sortie=1;
        delay_us(1500);
        Sortie=0;
        delay_us(13500);
    }
    else
    {
        Sortie=1;
        delay_us(2000);
        Sortie=0;
        delay_us(13000);
    }
}
```

## Annexe 6 : Programme rapports cycliques

#define _0PC	255	// 0%
#define _5PC	243	// 5%
#define _10PC	230	// 10%
#define _15PC	217	// 15%
#define _20PC	204	// 20%
#define _25PC	192	// 25%
#define _30PC	179	// 30%
#define _35PC	166	// 35%
#define _40PC	153	// 40%
#define _45PC	141	// 45%
#define _50PC	128	// 50%
#define _55PC	115	// 55%
#define _60PC	102	// 60%
#define _65PC	90	// 65%
#define _70PC	77	// 70%
#define _75PC	64	// 75%
#define _80PC	51	// 80%
#define _85PC	39	// 85%
#define _90PC	26	// 90%
#define _95PC	13	// 95%
#define _100PC	0	// 100%

## **Annexe 7: programme PWM et gestion de la coupure**

```
case DEPART: Arret_Moteurs();
    if ((PositionRobot == CENTRE_1) || (PositionRobot == CENTRE_3) || (PositionRobot == CENTRE_5))
    {
        Active_Moteurs();
        EtatCdeMot = CDE_CENTRE;
    }
    else EtatCdeMot = DEPART;

break;

//-----
//-----
//-----
// "Suivi de la ligne"

case CDE_CENTRE : CDE_MOTG = PWM[0][0]; CDE_MOTD = PWM[0][1];
    if (PositionRobot == GAUCHE) EtatCdeMot = CDE_DROITE;
        else if (PositionRobot == DROITE) EtatCdeMot = CDE_GAUCHE;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_COUPURE;
        else EtatCdeMot = CDE_CENTRE;

break;

case CDE_DROITE: CDE_MOTG = PWM[1][0]; CDE_MOTD = PWM[1][1];
    if (PositionRobot == CENTRE_5 ) EtatCdeMot = CDE_CENTRE;
        else if (PositionRobot == GAUCHE_PLUS) EtatCdeMot = CDE_DROITE_PLUS;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_COUPURE;
        else EtatCdeMot = CDE_DROITE;

break;

case CDE_DROITE_PLUS: CDE_MOTG = PWM[2][0]; CDE_MOTD = PWM[2][1];
    if (PositionRobot == GAUCHE ) EtatCdeMot = CDE_DROITE;
        else if (PositionRobot == GAUCHE_2PLUS) EtatCdeMot = CDE_DROITE_2PLUS;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_COUPURE;
        else EtatCdeMot = CDE_DROITE_PLUS;

break;

case CDE_DROITE_2PLUS: CDE_MOTG = PWM[3][0]; CDE_MOTD = PWM[3][1];
    if (PositionRobot == GAUCHE_PLUS ) EtatCdeMot = CDE_DROITE_PLUS;
        else if (PositionRobot == GAUCHE_3PLUS) EtatCdeMot = CDE_DROITE_3PLUS;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_COUPURE;
        else EtatCdeMot = CDE_DROITE_2PLUS;

break;

case CDE_DROITE_3PLUS: CDE_MOTG = PWM[4][0]; CDE_MOTD = PWM[4][1];
    if (PositionRobot == GAUCHE_2PLUS ) EtatCdeMot = CDE_DROITE_2PLUS;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_HORS_LIGNE;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_COUPURE;
        else EtatCdeMot = CDE_DROITE_3PLUS;

break;

case CDE_HORS_LIGNE: CDE_MOTG = PWM[5][0]; CDE_MOTD = PWM[5][1];
    if (PositionRobot == GAUCHE_3PLUS ) EtatCdeMot = CDE_DROITE_3PLUS;
        else EtatCdeMot = CDE_DROITE_2PLUS;

break;

case CDE_GAUCHE: CDE_MOTG = PWM[6][0]; CDE_MOTD = PWM[6][1];
    if (PositionRobot == CENTRE_5 ) EtatCdeMot = CDE_CENTRE;
        else if (PositionRobot == GAUCHE_PLUS) EtatCdeMot = CDE_DROITE_PLUS;
        else if (PositionRobot == HORS_LIGNE) EtatCdeMot = CDE_COUPURE;
        else EtatCdeMot = CDE_DROITE;

break;

case CDE_COUPURE: CDE_MOTG = PWM[0][0]; CDE_MOTD = PWM[0][1];
    if (PositionRobot == CENTRE_5 ) EtatCdeMot = CDE_CENTRE;
        else if (PositionRobot == GAUCHE_PLUS) EtatCdeMot = CDE_DROITE_PLUS;
        else EtatCdeMot = CDE_DROITE;

break;
```

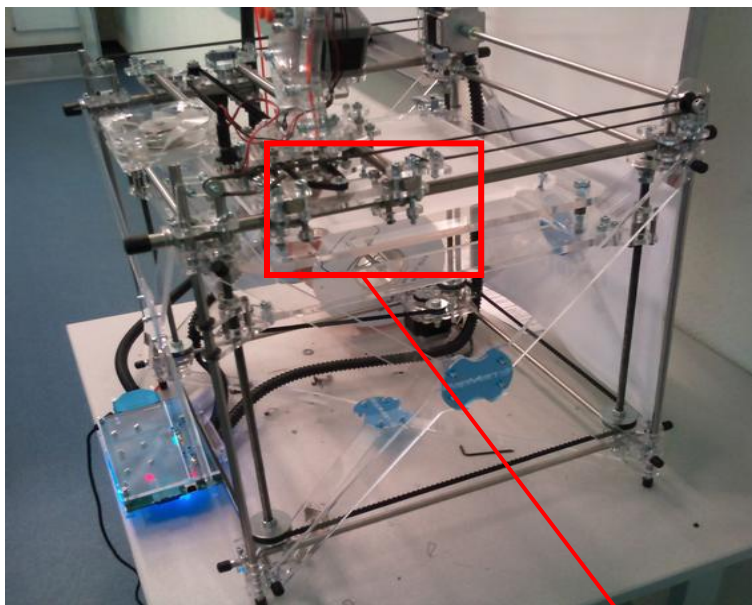
### Annexe 8 : l'imprimante 3 dimensions :

L'imprimante 3D à la même fonction qu'une imprimante classique sauf qu'elle reproduit des volumes.

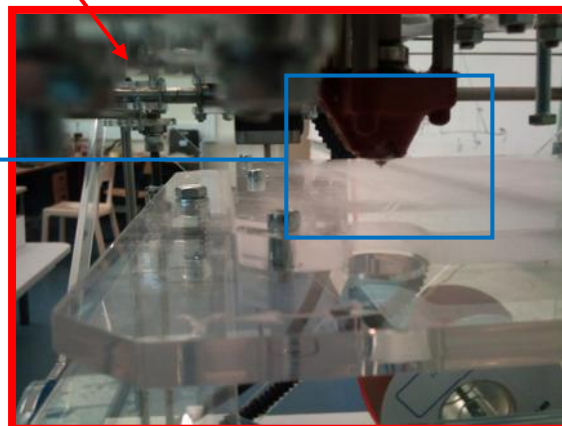
La création d'un volume se fait par l'entassement successif de couches de plastique à partir d'un plan. L'empilement de ces couches finit par former un volume.

Ce petit stylet dépose un liseré de plastique fondu. Au début on obtient une surface plane, mais au fur et à mesure que le stylet dépose des couches de plastique sur les précédentes on finit par avoir un volume comme par exemple notre catapulte.

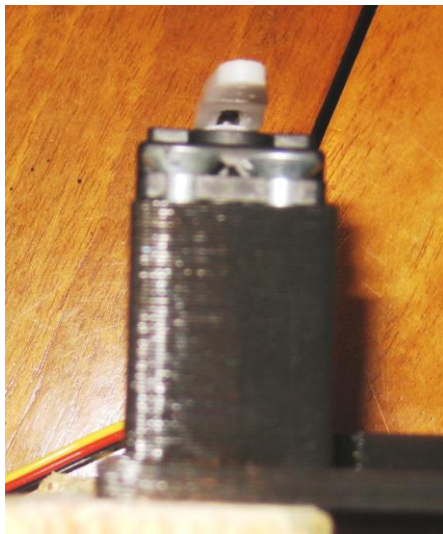
Toutefois nous ne pouvons pas élaborer la pièce de notre choix comme nous le voulons, nous devons toujours respecter la contrainte qui consiste un empilement de couches successives, nous ne pouvons pas créer de surface ne reposant sur aucun support.



Stylet



Sur cette photo on peut distinguer les couches successives laissées par le stylet.

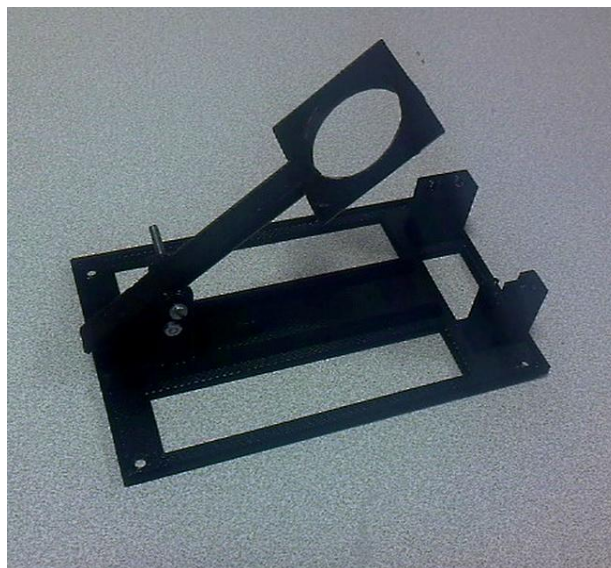


ON PEUT VOIR LE LISERET  
DE PLASTQUE

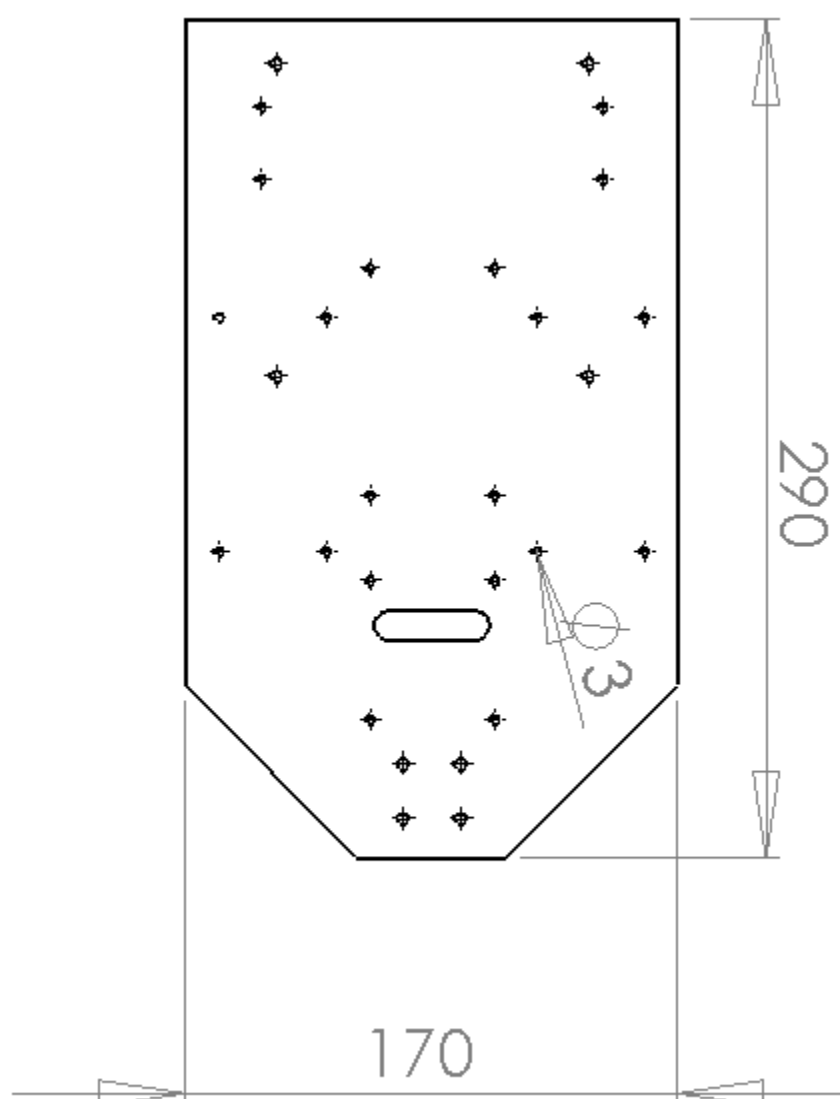


Le stylet dessine d'abord les contours de la pièce, puis remplit la surface à l'intérieur. Il reproduit cette action un certain nombre de fois, en fonction de la pièce à élaborer.

Voici notre lanceur fabriqué à l'aide cette imprimante :



**Annexe 9 :**



Annexe 10 :

