

PROJETS ARDUINO – PEIP2

Année scolaire 2018-2019

Le robot attrapeur autonome

Etudiants : Sébastien IAFRATE
Sébastien RAFANO

Encadrant : Pascal MASSON

REMERCIEMENTS

Avant tout développement sur ce projet, il paraît nécessaire d'introduire ce rapport de projet par des remerciements à notre encadrant, Pascal Masson, Professeur d'Electronique, qui nous a formé, accompagné et conseillé tout au long de ce projet Arduino. Aussi, nous remercions Nassim Abderrahmane, doctorant, pour avoir consacré un peu de son temps à notre robot.

SOMMAIRE

Introduction.....	4
Chapitre I : L'élaboration du Projet	
I.1. Définition du projet/cahier des charges	5
I.2. Les contraintes initiales	6
I.3. Planning initial.....	7
I.4. Planning final.....	8
Chapitre II: Déroulement du Projet	
II.1. Matériel utilisé.....	9
II.2. Répartition des tâches.....	12
II.3. Algorithme.....	13
II.4. Avancement du projet et difficultés rencontrées	14
Conclusion.....	15
Bibliographie.....	16

Introduction

Le projet Arduino est pour nous, l'opportunité de percevoir et de simuler la gestion d'un projet en entreprise. C'est la première fois que nous avons l'occasion de réaliser une expérience professionnelle en groupe : respecter un cahier des charges, concevoir un robot, palier à des problèmes mais aussi rédiger des rapports.

Une autre nouveauté pour nous, celle de réaliser un projet basé sur la manipulation, l'expérimentation de composants électroniques et de lier cela à de l'informatique.

Ce rapport présentera trois parties, retracant l'élaboration du robot. Il présentera d'abord le choix du projet, le cahier des charges, les contraintes qui sont naturellement imposées et celles que nous nous sommes données et également les objectifs souhaités. Deuxièmement sera présenté les étapes du projet ainsi que les difficultés rencontrées. Finalement nous conclurons sur l'état final du projet et les perspectives qui pourraient être envisagées dans le but d'améliorer le robot.

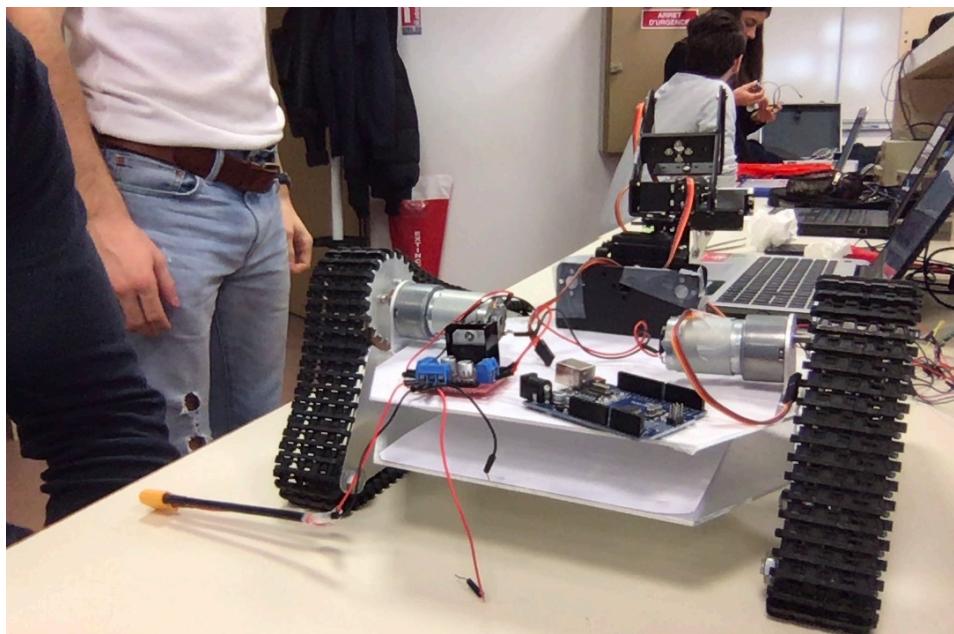
Chapitre I : L'élaboration du Projet

I.1. Définition du Projet/cahier des charges

Le robot attrapeur autonome a pour caractéristique fondamentale d'interagir avec son milieu. C'est à dire qu'une fois le robot en marche, celui-ci est capable de détecter des objets de différentes couleurs et, grâce à un dispositif Bluetooth, de se diriger vers l'objet de la couleur souhaitée. Puis, parmi les couleurs définies au préalable, le robot récupère l'objet (qui tout au long de notre projet sera un cube rouge ou vert) à l'aide d'un bras mécanique qui, lui aussi, est autonome.

Nous avons donc au début du projet récupérer une base de robot construit par d'autres élèves lors de leur projet l'année précédente. Nous avons dû évaluer complètement le robot et définir toutes les améliorations à apporter pour qu'il soit mécaniquement correct afin de pouvoir le rendre autonome et efficace.

Ce robot s'apparente à un petit char, il dispose de 2 chenilles entraînées par 2 moteurs qui le font avancer, reculer ou tourner. Ces chenilles le rendent en effet très stable, le poids du robot est plus réparti sur le sol, et lui permettent d'avoir une très bonne adhérence au sol.



Base récupérée au début du projet

L'enjeu de ce projet Arduino est de réaliser un objet connecté, ayant une fonction utile pour un particulier mais aussi de se rapprocher le plus d'un travail en entreprise en le rendant commercialisable.

Ce robot autonome pourrait alors être capable d'analyser une pièce et de ranger des objets de tailles adaptées à un endroit précis.

Pourquoi le rendre autonome ?

Rendre le projet Arduino contrôlable et dirigeable par Bluetooth serait trop accessible au vu des connaissances acquises en électronique. Le rendre autonome pourrait avoir une réelle application pour un particulier, il n'aurait pas besoin de consacré du temps à ranger sa maison, son robot le ferait à sa place.

I.2. Contraintes initiales

La contrainte principale de ce projet est de rendre le robot autonome, car le robot doit être d'une extrême précision pour qu'il réalise ce que nous souhaitons.

Une autre contrainte importante est le temps, il faut respecter le temps imposé : 8 séances de 3 heures réparties sur trois mois.

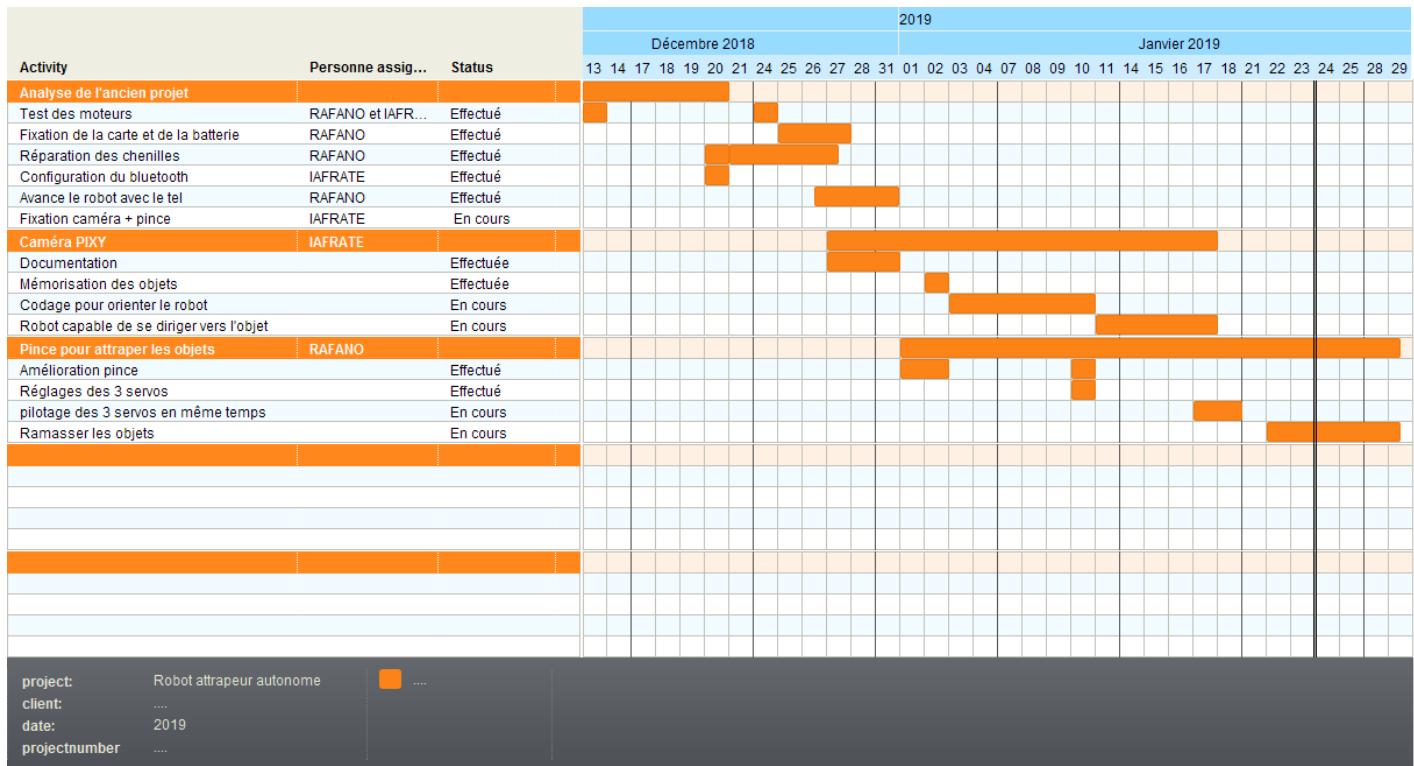
De plus, le fait que notre projet soit un robot sur lequel est fixé tous les composants que nous utilisons, nous ne pouvons pas travailler en même temps sur le robot, entre deux séances, seulement un de nous deux peut avancer sur le robot, néanmoins, nous avons un peu contourner ce problème car lorsqu'un de nous deux avait le robot chez lui, l'autre effectuait des recherches pour ne pas perdre trop de temps.

De surcroît, il est nécessaire d'intégrer une fonctionnalité Bluetooth au robot. Le robot serait en communication avec un smartphone par l'intermédiaire d'une application.



Enfin, le robot serait capable de distinguer différents objets en fonction de leurs couleurs et ceci en temps réel.

I.3. Planning initial

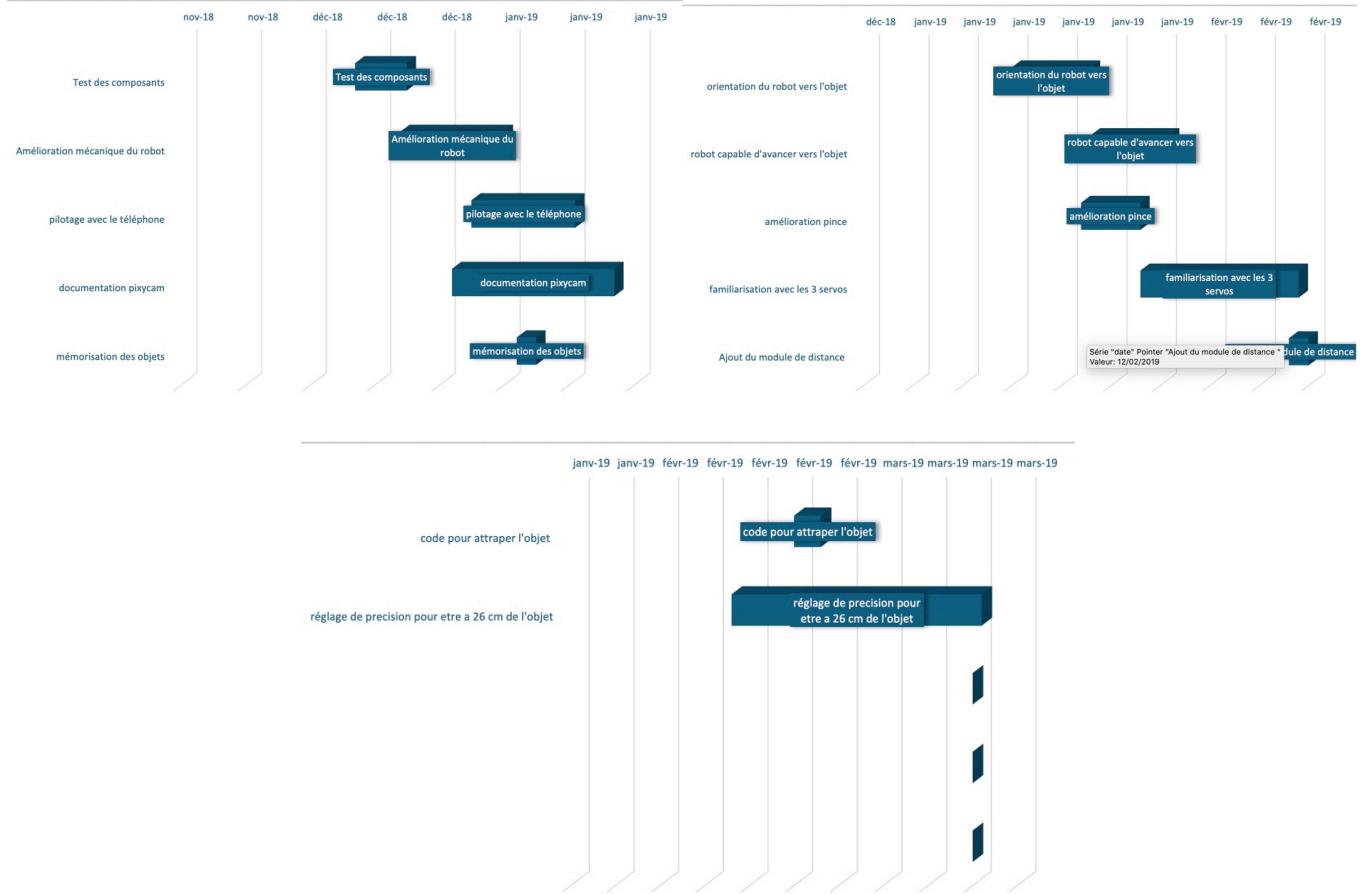


La récupération d'un ancien projet nécessite un état des lieux, tant au niveau bibliographique que mécanique.

C'est pourquoi 2 semaines complètes devaient être dédiés à l'analyse de l'ancien projet. Une fois la caméra PIXY réceptionnée, élément clef pour l'autonomie du robot, 2 séances de travaux pratiques ajoutées à du travail à la maison devraient suffire pour permettre au robot de se diriger librement vers les objets.

Pendant ce temps, le collaborateur du projet consacrerait 4 séances à l'amélioration du bras mécanique et des servomoteurs qui le composent.

I.4. Planning final



Si l'on compare les deux plannings, on peut remarquer que certaines parties du projet ont fait prendre un retard considérable.

Premièrement, l'analyse de l'ancien projet et le réglage des problèmes associés ont rajoutés 2 semaines de travail en plus.

Aussi, l'ajout d'un module de distance était imprévu, la caméra PIXY semblait être capable d'évaluer la distance séparant celle-ci de l'objet. Pour autant cela était trop imprécis et a donc rajouté un travail supplémentaire, bien plus que ce qui était initialement prévu.

Ces 2 éléments sont les causes du retard rencontré à la fin du projet, il n'a pas été facile de maîtriser complètement ces 2 composants, nous avons dû faire face à de nombreux problèmes tout au long de notre projet, parfois dur à régler.

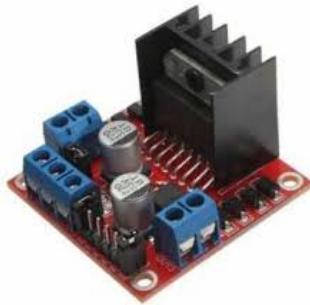
Chapitre II : Déroulement du Projet

II.1. Matériel utilisé

Lors de la réalisation du projet Arduino, l'utilisation de composants étudiés durant les 3 mois de cours précédent ont été nécessaire. Voici une liste de ces composants :

Cartes :

- 1 carte L298 : alimente les moteurs à courant continu (0 à 12V), alimentée par une batterie « high power ».



- 1 carte arduino uno : alimente les servomoteurs, la Pixycam, le module de distance et le module HC-06 (5V), alimentée par une batterie portative.

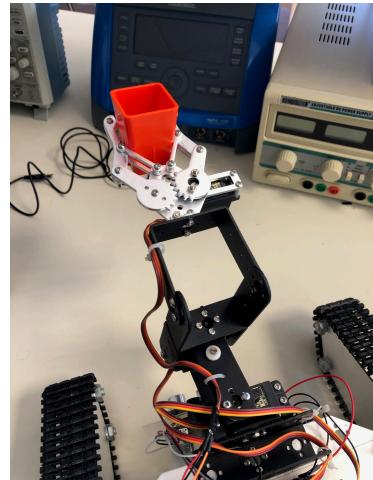


Moteurs :

- 2 moteurs à courant continu : font tourner les chenilles pour déplacer le robot.



- 3 servomoteurs : Ces 3 servomoteurs définissent les 3 mouvements possibles du bras mécanique, mouvement vertical, horizontal et l'ouverture ou fermeture de la pince.



- **Pixycam** : Sert à détecter un objet d'une couleur définie au préalable. L'ajout de 7 signatures au maximum est possible par cette caméra. Le robot pourra, au maximum, détecter 7 objets de couleurs différentes. Un logiciel adapté « PIXY MOM » permet d'observer ce que voit la caméra mais aussi de lui mettre en mémoire ces signatures.



- **HC-06** : Le module a pour fonction d'établir la connexion Bluetooth entre le robot et le téléphone portable



Modules de distances :

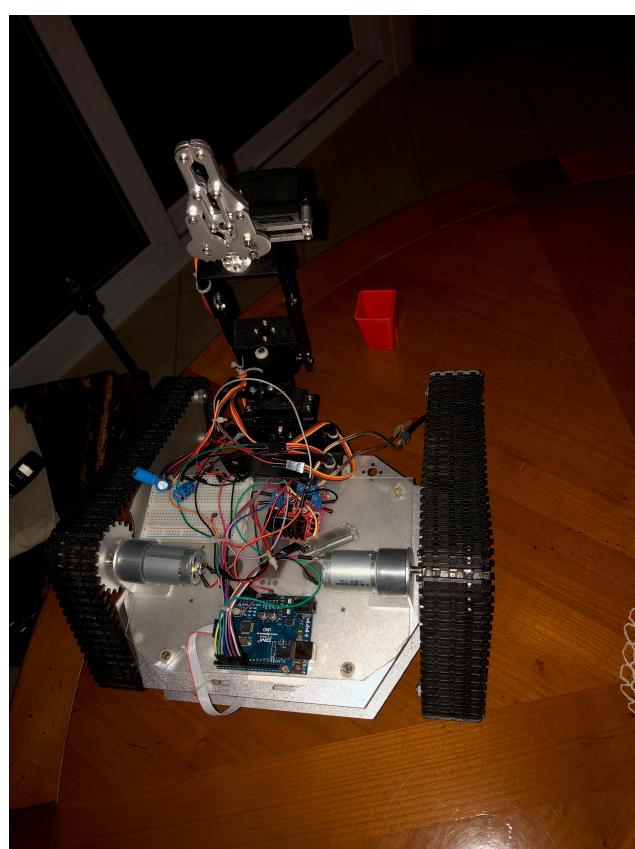
- HC-SR04 : module de distance par onde acoustique.



- SHARP GP2Y0A21YK0F : module de distance par infrarouge.



Vue d'ensemble du robot final :



II.2. Répartition des tâches

La répartition des tâches a été la plus équitable possible afin que chacun effectue le même volume de travail et participe le plus possible dans l'avancé du projet.

La partie « test » de l'ancien projet a été effectué dans les premières séances et a donc été faite par les 2 étudiants.

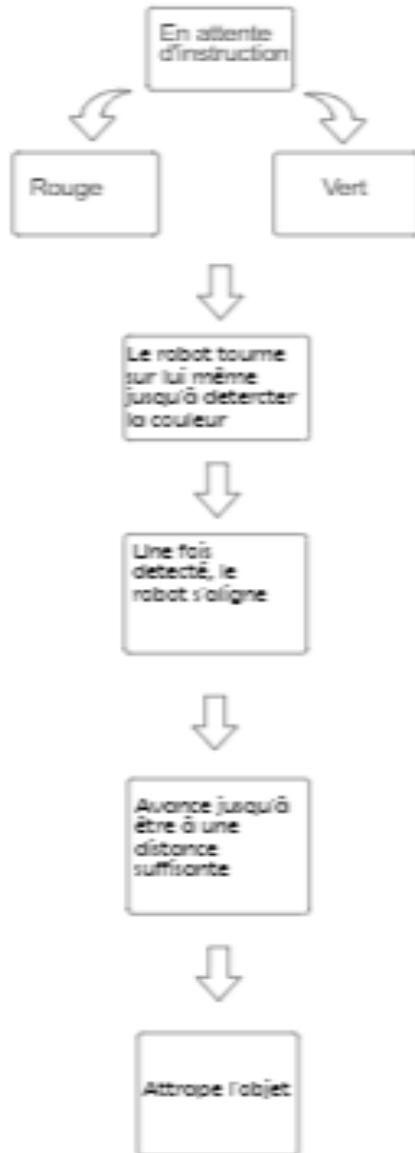
Les modifications afin de sécuriser le robot et de le rendre mécaniquement correct, c'est à dire la réparation d'un des moteurs, des chenilles, la fixation du plexiglas et des cartes et l'amélioration du bras mécanique pour le rentre optimal a été assigné à Sébastien RAFANO ainsi que les phases de réglages, de pilotage des 3 servomoteurs et de codage pour ramasser un objet.

La documentation, la familiarisation et le paramétrage de la Pixycam ainsi que la mémorisation des objets, la capacité du robot de s'orienter face à l'objet et d'avancer vers ce dernier et enfin la configuration du système Bluetooth avec le code pour choisir l'objet que l'on veut ramasser ont été réalisé par Sébastien IAFRATE.

Ensuite, les deux étudiants ont travaillé ensemble en cours et tour à tour chez eux sur la capacité du robot à s'arrêter à une distance bien précise de l'objet pour que le bras mécanique puisse l'attraper. Le module choisi au départ était le module HC-SR04 (description ci-dessus) mais, suite aux événements et aux problèmes rencontrés, ce module a été remplacé par le SHARP GP2Y0A21YK0F, un module plus efficace et plus précis.

Ce travail de répartition des tâches a été très bénéfique pour un plausible travail en groupe futur. En effet, ce projet requiert un nombre important de qualités et ne peut être qu'une bonne expérience pour notre vie à venir.

II.3. Algorithme



Un seul programme : Divisé en blocs suivant la chronologie du scénario :

- . Phase de détection
- . Phase d'alignement
- . Phase de rapprochement
- . Phase de préhension

Pendant la phase de détection, le robot tourne sur sa droite jusqu'à ce que la caméra PIXY détecte l'objet. Une fois détecté, on rentre dans la phase d'alignement : la camera PIXY va définir un axe sur lequel se trouve l'objet et le robot va faire en sorte de se retrouver face à l'objet avec de simples boucles if. Ensuite intervient la phase de rapprochement : le robot va avancer droit vers l'objet, grâce au module de distance qui renvoie constamment la distance qui sépare le robot de l'objet. Le robot va s'arrêter dès que la distance renvoyée est inférieure à 26 cm, c'est à dire la longueur du bras articulé. Pour finir, on rentre dans la phase de préhension : le bras mécanique va descendre sur l'objet et la pince va se refermer pour pouvoir saisir ce dernier et le décoller du sol (démonstrations disponibles sur le GitHub).

II.4. Avancement du projet et difficultés rencontrés

Comme précisé auparavant et bien illustré a l'aide des deux plannings, un nombre important d'imprévus ont perturbé l'avancée du projet et le respect du cahier des charges.

Dans un premier temps, le robot était incapable de s'arrêter à chaque fois a la bonne distance de l'objet, cela marchait quelques fois mais pas assez souvent pour être satisfaisant.

Un autre problème qui était selon nous très difficile à régler est que comme nous apprenions à la caméra la couleur des objets et pas vraiment leur forme, tous les éléments de la même couleur que notre cube qui étaient dans la pièce étaient détectable par la caméra, ainsi il était possible que le robot se dirige vers un extincteur de la pièce par exemple. De plus, en fonction de l'éclairage de la pièce, la caméra pouvait également ne pas reconnaître les cubes et donc faire échouer le robot.

Une dernière difficulté rencontrée qui a été illustrée lors de notre présentation est au niveau de la pince du bras mécanique et plus précisément au niveau du servomoteur qui dirige l'ouverture ou la fermeture de la pince. Initialement, nous demandions à la pince de se fermer grâce à la fonction « servowrite(angle) » sur Arduino, mais lorsque l'objet se situe dans la pince et que cette dernière se ferme, la pince se fermara sur le cube et le servo n'arrivera donc pas à faire l'angle que l'on souhaite. Lorsque cela se produit, tous les servomoteurs se « déconnecte » et ne sont plus fonctionnels. Pour résoudre ce problème, nous avions donc fait de nombreux test pour déterminer la valeur de l'angle qu'il faut soumettre au servomoteur pour que la pince se ferme pile sur le cube. Malgré cela, du fait que notre cube n'a exactement la même largeur sur toute sa hauteur, si le bras essaye de lever le cube par son extrémité, la pince ne pourra pas se fermer comme elle le souhaite et le même phénomène se produit.

Il fût difficile de coder tout le scénario dans un seul unique programme. C'est pourquoi il fût nécessaire de reprendre tout le code à mi-parcours afin de le diviser en phases, comme décrit ci-dessus, avec l'utilisation de booléens.

La superficie du robot fût aussi un problème. Initialement, il a été pensé de placer un bac en plastique sur le robot afin d'y déposer les objets avec la pince.

Impossible, il a été décidé d'attraper puis de redéposer à côté l'objet choisi.

Même si très rapidement le planning n'était plus respecté, nous étions satisfaits car il y avait une réelle avancée significative qui nous rapprochait de plus en plus vers notre but.

Mais, comme on peut le voir dans le planning final, nous avons mis quasiment 1 mois pour réussir à faire avancer le robot jusqu'à 26 cm de l'objet (soit la distance du bras) pour le saisir.

Cela est dû au fait que nous exigeions une extrême précision, en effet, si le robot n'était pas parfaitement en face de l'objet ou si le robot s'arrêtait à 1 cm de plus ou de moins, le bras ne pouvait pas attraper l'objet.

Conclusion et perspectives d'avenir

La plupart des problèmes proviennent de l'imprécision lié au module de distance et à la caméra PIXY. Avec du matériel plus performant et donc bien plus couteux il serait bien plus facile d'attraper les objets à tous les coups.

Deuxièmement il semble nécessaire de continuer le code afin qu'une fois l'objet attrapé, le robot aille déposer l'objet dans un récipient de façon toujours autonome. Il serait aussi possible de paramétrier la caméra PIXY afin que le robot puisse attraper d'autres objets, pas que rouge ou vert.

Pour finir il faudrait également permettre au robot de réaliser plusieurs fois le scénario. C'est-à-dire qu'une fois l'objet placé dans le récipient, le robot devrait pouvoir aller chercher d'autres objets. Cela devrait être plutôt rapide à concevoir grâce à la présence de phases codées en booléens. Il faut garder à l'esprit que, même si des petits problèmes persistent, le robot est fonctionnel. Le choix de la couleur de l'objet à attraper fonctionne à tous les coups, l'alignement fonction aussi, il reste simplement à améliorer la précision du module distance pour permettre d'attraper les objets à tous les coups.

Si on nous donne encore 9 séances dans notre projet, le robot aurait fonctionné à tous les coups. Avec du recul et notre expérience, nous avons pensé à une alternative qui aurait pu simplifier le problème principal rencontré, en effet, il aurait suffi de dire au robot de s'arrêter quand la distance de l'objet et de 30-35 cm, de baisser le bras, d'ouvrir la pince et d'avancer droit vers l'objet sur une distance d'environ 10 cm, si le robot est bien aligné, l'objet rentre à l'intérieur de la pince et le robot pousse l'objet sur quelques centimètres puis l'attrape et le soulève du sol. Cette alternative demande donc une moins grande précision et nous pensons qu'en appliquant cette méthode, le robot serait capable d'effectuer ce que nous voulions au départ.

Bibliographie

Contrôle du module de distance par infrarouge :

<https://www.instructables.com/id/Get-started-with-distance-sensors-and-Arduino/>

Documentation et code pour les composants étudiés en cours :

<http://users.polytech.unice.fr/~pmasson/Enseignement-arduino.htm/>

Documentation sur la Pixy CAM :

<https://pixycam.com/pixy-cmucam5/>

<http://www.cmucam.org/projects/cmucam5>

<https://www.generationrobots.com/fr/401980-capteur-camera-pixy-cmucam5.html>

<https://www.lextronic.fr/shield-audio-video/30471-module-de-reconnaissance-video-pixy.html>

<https://www.deviceplus.com/how-tos/arduino-guide/an-intro-to-cmucam5-pixy-vision-camera-sensor/>

<https://www.instructables.com/id/Using-a-PixyCam-CMUcam5-to-Get-Distance-of-a-Known/>

Ancien projet : Clean Robot, réalisé par des élèves de Polytech :

<https://github.com/BenLil/Clean-Robot>

