

Projet Arduino – PEIP2

Année scolaire 2021/2022

Uberdrinks

Etudiant : Gaultier Julien

Encadrant : BENOUMAKTA, MASSON

Sommaire

Contenue

Sommaire	2
Introduction.....	3
Objectifs	4
Algorithme du projet	6
.....	6
Planning initiale/planning suivit	7
.....	7
Problèmes rencontrés	7
Points à améliorer en vue d'expériences futurs.....	9
Bibliographie.....	9
Conclusion	9

Introduction

Initialement, l'idée était de faire un projet capable de simplifier la vie au quotidien et, après maintes réflexions, Uberdrinks est apparu. Il fallait trouver une solution pour résoudre un problème qui nous est tous déjà arrivé. Vous l'avez deviné, il s'agit du dilemme qui oppose le fait de bouger alors que nous n'en avons pas envie et le fait de se priver lorsque nous avons faim/soif.

Le principe de Uberdrinks est simple, il consiste en l'apport de divers snacks/boissons, que nous avons placés au préalable sur le robot, à un endroit particulier choisi par l'utilisateur. Donc, grâce à Uberdrinks, nous pouvons avoir de quoi nous sustenter en 1 click sur notre smartphone.

La problématique derrière mon projet était de savoir si les robots sont une solution possible pour nous simplifier la vie au quotidien.

Pour répondre à cette problématique, quoi de mieux qu'un robot permettant de nous apporter divers objets, sans avoir besoin de fournir un quelconque effort. C'est pourquoi, Uberdrinks était le choix idéal.

Création du robot

La conception du robot s'est déroulée en 3 étapes :

- La première consistait à obtenir un châssis permettant d'accueillir tous les modules. Le châssis présent sur les voitures suiveuses de ligne (cf. Figure 2) s'est avéré être le plus optimal dans cette optique, car il alliait espace disponible et de nombreux trous. En tout, 2 châssis ont été nécessaires, 1 pour maintenir tous les modules du robot (suiveur de ligne, Bluetooth, détection d'obstacle) et 1 autre pour séparer la partie câblage de la partie stockage de nourriture (pas abouti car manque de temps).
- La seconde consistait à installer tous les modules que le robot allait utiliser pour pouvoir accomplir sa tâche. Cette étape a nécessité un peu de travail manuel car il a été nécessaire de modifier le châssis en ajoutant des trous permettant de visser tous les composants, notamment la carte Arduino. De plus, cette étape a dû être réalisée 2 fois, car la carte initialement installée ne permettait le bon fonctionnement du robot (cf. Problèmes rencontrés).
- La troisième consistait à brancher tous les modules entre eux. Cette étape n'a pas réellement posé de problème. Cependant, l'espace disponible étant limité, il a été nécessaire d'occuper tout l'espace disponible, cela rend donc toute modification du robot en lui-même très compliquée.

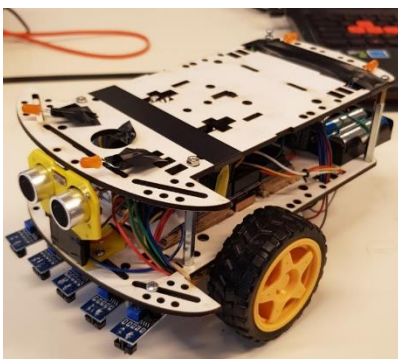
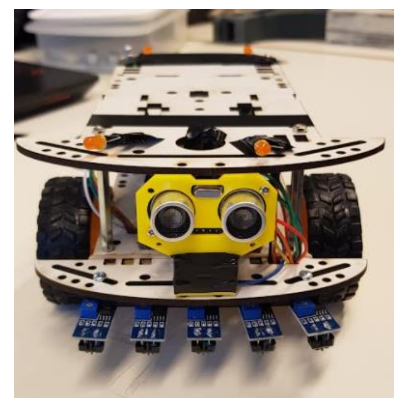


Figure 1 Uberdrinks



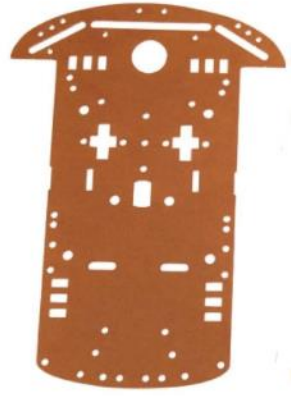


Figure 2 châssis

Objectifs

L'objectif à terme de ces séances était de faire un robot pouvant arriver à un endroit de notre choix. Le robot fonctionne sur la même base que les robots suiveurs de lignes utilisés en cours, à cela s'ajoute un module Bluetooth permettant de choisir le point d'arrivée du robot sur notre téléphone et un module d'évitement d'obstacle.

Explication de l'algorithme

Nous venons de voir qu'elles étaient les fonctionnalités que le robot devait posséder, à l'issue des 8 séances disponibles. Nous allons à présent nous intéresser à l'algorithme du projet que vous trouverez ci-dessous. L'algorithme se décompose en 3 parties : Bluetooth, suiveur de ligne et changement de direction.

Bluetooth

La partie Bluetooth marche en 3 temps. Dans un premier temps, l'utilisateur envoie au robot l'ordre de tourner à la n-ième intersection via la télécommande programmée sur l'application « Bluetooth Electronics » (cf. Figure 4). L'ordre est envoyé sous forme d'un « char ». Dans un second temps, le message que reçoit le robot est traduit en donnée exploitable sous la forme d'un « int ». Dans un troisième et dernier temps, une fois que le robot sait où il doit aller, ce dernier commence enfin son trajet jusqu'à son objectif.

Suiveur de ligne

La partie suiveuse de ligne consiste simplement à réajuster la trajectoire du robot en fonction de la valeur des capteurs. Si le capteur droit détecte la ligne, le robot va augmenter la puissance de son moteur gauche et diminuer la puissance du droit. De même, si le capteur gauche détecte la ligne. Cette fois, le robot va augmenter la puissance de son moteur droit et diminuer la puissance du moteur gauche. Enfin, si aucun des 2 capteurs ne détecte la ligne, le robot laisse la puissance de son moteur droit égale à celle du moteur gauche. (cf. Figure 1).

Changement de direction

La partie changement de direction fonctionne en 3 étapes. La première consiste à permettre au robot de détecter une intersection (cf. image) en fonction de la valeur du capteur central et extérieur. La seconde étape consiste à vérifier s'il s'agit de la bonne intersection. Pour se faire, un compteur est utilisé. Dès qu'une intersection est rencontrée, le robot va comparer la valeur « int » de son compteur à la valeur de l'ordre émis via Bluetooth. Si les 2 valeurs sont identiques, le robot entame son changement de trajectoire. Il va donc s'arrêter, mettre ses clignotants placés du côté de sa nouvelle trajectoire et tourner pour pouvoir poursuivre sa course. Sinon, on ajoute 1 au compteur.



Figure 4 télécommande

Comme il est possible de le voir, la télécommande actuellement utilisée est composée de 5 boutons correspondants au numéro de l'intersection à laquelle le robot devra tourner. La commande est envoyée sous la forme d'un « String » (« a » lorsqu'on appuie sur le bouton 1, « b » sur le bouton 2, ...). Cette valeur est traduite en une valeur de type « int » (1 pour « a », 2 pour « b ») par le robot.

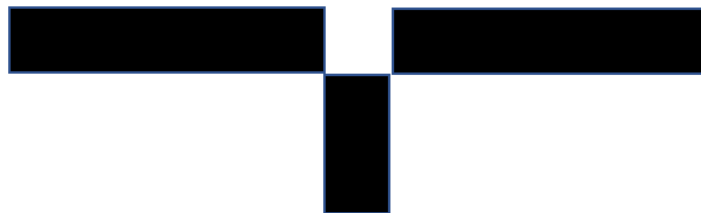
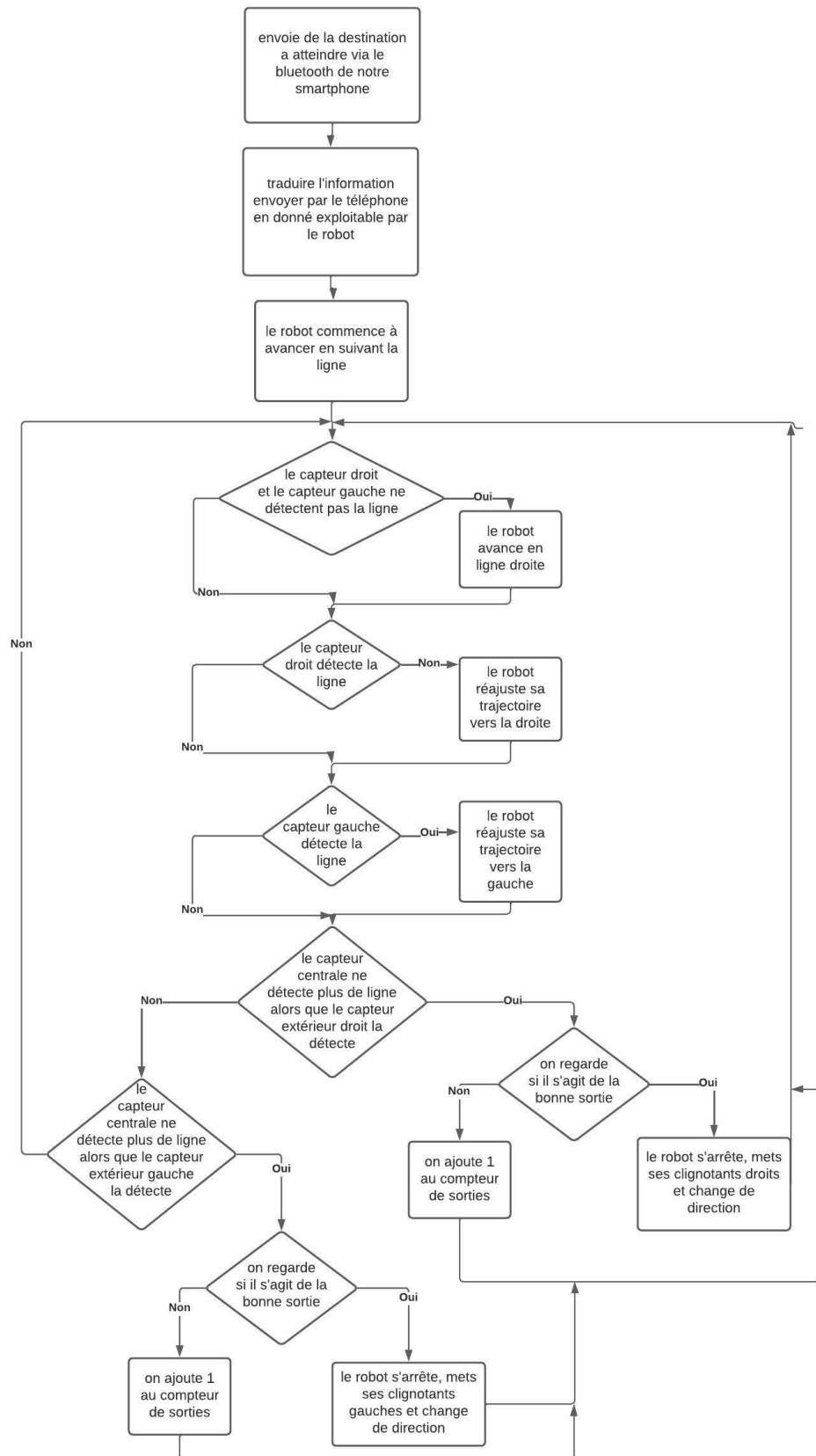


Figure 3 Intersection

Algorithme du projet



Planning initial/planning réellement suivi

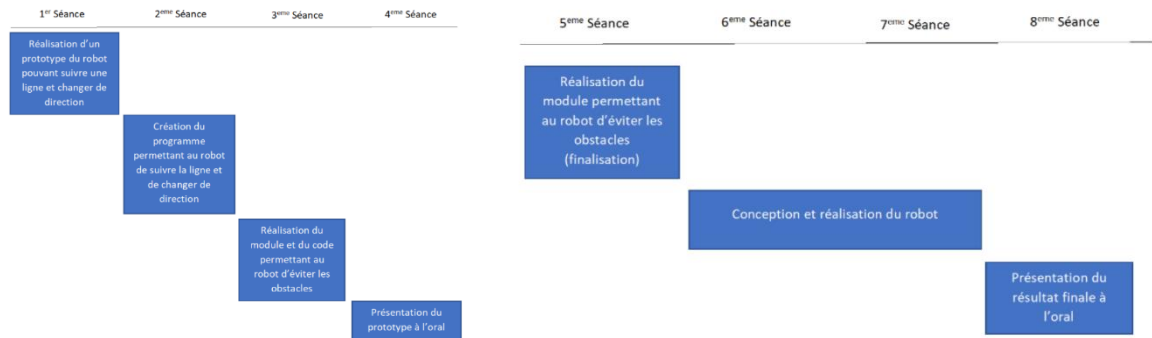


Figure 5 planning initiale

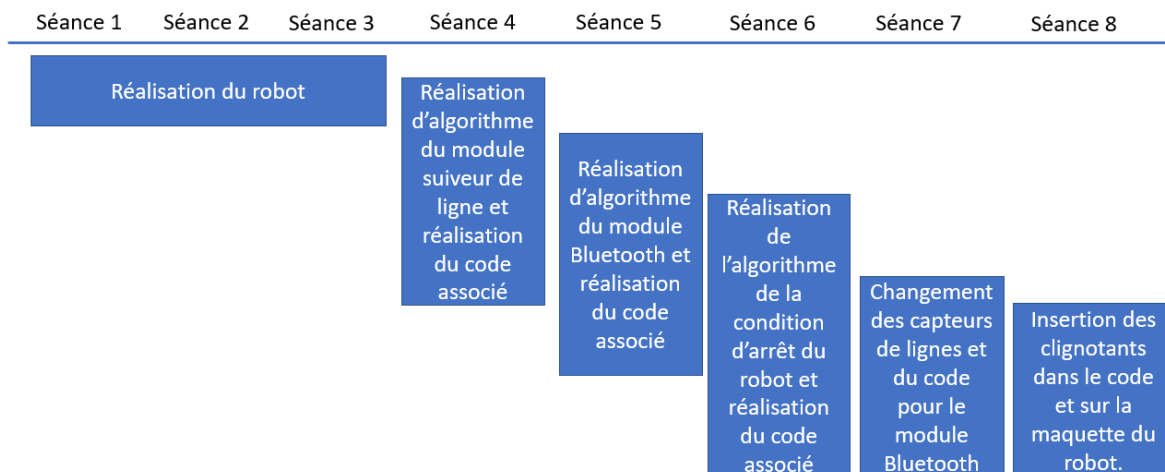


Figure 6 Planning suivi

Comme il est possible de le voir, le planning n'a pas du tout été respecté. La première raison à cela est qu'il était impossible de réaliser un robot possédant un mini-réfrigérateur. Un robot incorporant tous les capteurs a donc été réalisé. De plus, sa réalisation a été plus longue que prévu. La seconde raison est que les capteurs ne fonctionnaient pas de la façon voulue.

Si un délai supplémentaire était disponible, le module évitement d'obstacle pourrait devenir fonctionnel. De plus, un compartiment permettant de stocker des aliments aurait pu être incorporé au robot. Enfin, il aurait été possible de créer un code permettant au robot de retourner à sa base de rechargement.

Problèmes rencontrés

- Carte Arduino en elle-même : dans un premier temps, une carte Arduino Uno était installée à bord du robot. Or, il y a eu un souci au niveau de la place qu'offrait cette carte. Il manquait de nombreux ports sur cette dernière. Pour résoudre ce problème, un changement de carte

a été obligatoire, et la carte Uno a été remplacée par une carte méga, offrant ainsi suffisamment de place pour pouvoir accueillir tous les modules.

- Module suiveur de ligne : les capteurs initialement installés sur le prototype ne détectaient pas correctement les lignes. Pour résoudre ce problème, un remplacement de capteurs a été nécessaire.
- Module Bluetooth : la carte Arduino méga possède un tout autre fonctionnement au niveau du Bluetooth par rapport aux autres cartes. Pour résoudre ce problème, le code a donc dû être modifié pour pouvoir fonctionner.
- Module évitement d'obstacle : ce module n'a pas pu être fini à temps à cause du retard pris lors de la réalisation des autres parties du projet. L'algorithme a pu être fini à temps et l'ébauche du code aussi. Cependant, il n'a pas été possible de l'implémenter dans le code final. Vous trouverez l'algorithme de ce module ci-dessous, suivi d'une brève explication de son fonctionnement.

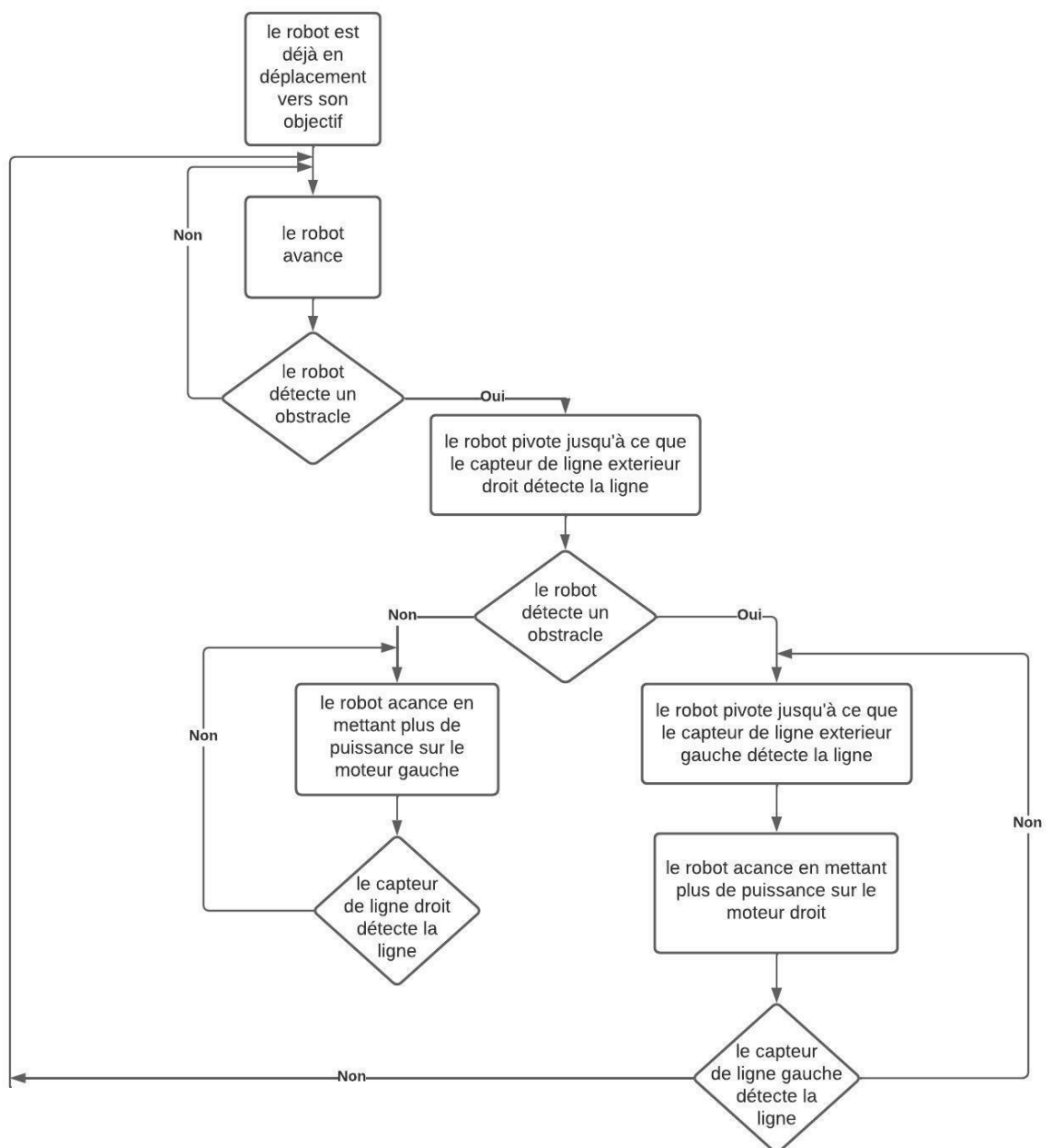


Figure 7 algorithme évitement d'obstacle

Le code se décompose en 2 parties. La première consiste à regarder si un obstacle est oui ou non présent sur la trajectoire du robot en utilisant capteur à ultrason présent à l'avant du robot (cf. Figure 1). La seconde consiste à faire pivoter le robot d'un côté et à regarder si la voie est libre de ce côté. Si oui, le robot va donc avancer en effectuant une trajectoire courbée autour de l'obstacle. Sinon, le robot va chercher de l'autre côté un chemin disponible.

Points à améliorer en vue d'expériences futures

Le plus gros problème rencontré durant cette expérience a été le manque de prise d'initiative. Il a fallu attendre 4 séances pour penser à regarder si les problèmes ne venaient pas du code ou des branchements, mais plutôt des capteurs installés. Cela a donc coûté énormément de temps et s'est traduit par un module inachevé.

Bibliographie

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

[Bluetooth Controlled Car with Arduino Mega - Arduino Project Hub](#)

Conclusion

Rendu final

Malheureusement, le projet n'a pas pu être achevé. Pour pouvoir être complètement achevé, il aurait fallu 1 à 2 séances de plus. Cependant, les modules terminés et installés sur le robot ont fonctionné de la façon dont je l'avais imaginé. De plus, ce projet m'a permis d'améliorer mon travail en autonomie et m'a permis de réellement expérimenter le fait d'avoir une date de rendu fixe.

Finalement, j'ai réussi à répondre à la problématique, en créant un robot permettant d'effectuer des tâches pénibles, comme le transport d'objet, à la place de l'homme. C'est pourquoi les robots permettent de faciliter la vie au quotidien.

Perspective d'avenir

Une fois le projet initial terminé, il aurait été possible de créer une base de recharge permettant à l'utilisateur de choisir l'objet que doit apporter le robot. De plus, il aurait été possible d'implémenter un code permettant au robot de retourner au niveau de cette base de recharge.

Mais encore, ce type de robot pourrait être utilisé dans divers domaines, comme dans le domaine médical (le robot pourrait apporter aux patients des médicaments), dans la livraison (sous

la forme d'un facteur automatisé), ... Ce type de robot pourrait donc être utilisé dans de nombreux domaines, permettant ainsi de remplacer l'humain sur des tâches pénibles.