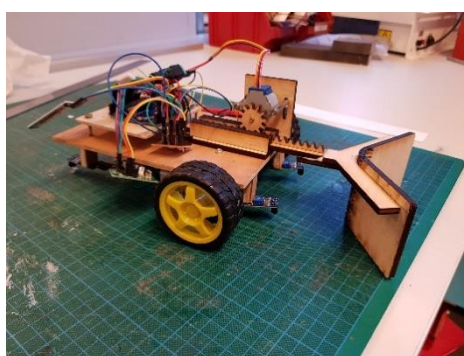


## PROJET ARDUINO - PEIP2

*Année scolaire 2018-2019*

### ***Robot Nettoyeur de Table Arduino"***



### Etudiants :

**Olivo Arnaud**

**Mokaddem Ahmed Yassine**

### Encadrants :

**Masson Pascal**

**Abdelrahmen Nassim**

## **REMERCIEMENTS**

Nous remercions Monsieur Forner pour son aide précieuse au Fab Lab.

Nous remercions Monsieur Masson pour sa disponibilité et pour ses conseils avisés.

Nous remercions Monsieur Abderrahmane pour sa disponibilité et ses observations.

# **SOMMAIRE**

## I. Introduction du projet

## II. Partie Déplacement :

1. Fabrication des pièces
2. Montage de la partie déplacement
3. Gestion des déplacements
4. Montage des capteurs
5. Gestion des capteurs

## III. Partie nettoyage :

1. Fabrication des pièces
2. Montage de la partie nettoyage
3. Gestion de la partie nettoyage

## IV. Planning

## V. Conclusion

## VI. Bibliographie

## VII. Annexes

# **I- INTRODUCTION**

Le RNTA est un Robot Nettoyeur de Table Arduino avec deux roues motrices et une roue « libre » qui est contrôlé par Bluetooth avec un smartphone. Le système qui nettoie la table va pousser les miettes en dehors de celle-ci tout en restant sur la table grâce à ces capteurs. Nous verrons d'abord la partie déplacement avec la construction, le montage et la gestion de celle-ci puis la partie nettoyage avec le même cheminement.

## **II- CAHIER DES CHARGES**

**Objectifs** : Robot nettoyeur de table et de miettes.

**Caractéristiques** :

Robot avec deux roues motrices et une roue « libre » qui est contrôlé par Bluetooth avec un smartphone. Le système qui va nettoyer la table sera contrôlé par un moteur... qui va pousser les miettes en dehors de la table. Le robot devra rester sur la table et ne pas tomber.

Le robot fera ~15cm de long et ~10cm de large.

**Découpage du projet** :

**Partie déplacement** :

- Montage
  - Moteurs
  - Bluetooth
  - Détecteurs
  - Socle
  
- Algorithme
  - Gestion des déplacements
  - Télécommande
  - Gestion Bluetooth

**Gestion miettes** :

- Fabrication de la balayette
- Montage moteur pas à pas et balayette
- Algorithme

### **III- PARTIE DEPLACEMENT**

Dans cette partie nous avons surtout travaillé le montage du robot pour qu'il puisse se déplacer correctement ainsi que la partie algorithmique qui va avec.

#### **III.1. Fabrication des pièces**

Nous avons choisi de faire un robot qui fait 18 cm de long et 12 cm de largeur.

Matériel utilisé durant cette partie :

- 2 moteurs CC du type ceux de TP
- L298 – Quadruple demi pont en H
- Deux Roues
- Une roue libre
- Un support
- Arduino
- 3 Keyes IR-01 (détection du sol)
- Module Bluetooth HC- 05
- Pille de 9V pour l'alimentation.

#### **III.2. Montage de la partie déplacement**

Nous avons d'abord commencé par découper la plaque de bois avec la scie mise à disposition en prenant une marge sur le bois pour pouvoir gérer les erreurs de découpage. Cette plaque fait office de support pour tous les composants.

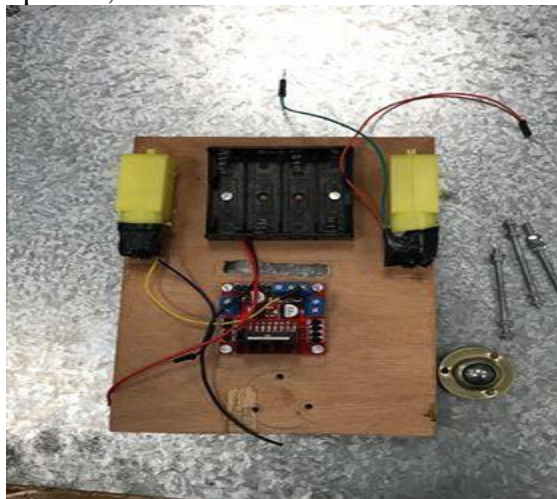


Planche principale

Nous avons soudé les câbles au connecteur des moteurs et nous avons testé les moteurs en faisant les branchements avec carte Arduino et le module L298. Nous avons utilisé les mêmes

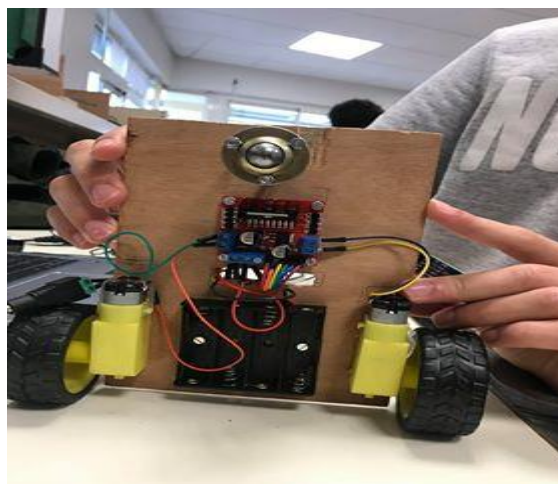
branchements que ceux réalisés durant les TD. Nous avons eu des problèmes de téléversement et un des moteurs ne fonctionnait pas car les soudures n'étaient pas bien faites sur l'un des moteurs. Nous avons refait les soudures sur ce moteur et finalement on a réussi à faire marcher les deux moteurs.

Ensuite nous avons commencé à percer le support en bois pour pouvoir fixer les composants, nous avons fait un trou au milieu avec une perceuse et la lime pour faire passer les fils, ensuite nous avons voulu coller les moteurs mais nous avons rencontré un problème. En effet la colle ne tenait pas car nous ne disposions pas de colle pour bois et plastique. Nous avons également fixé la boîte à piles et la puce LS à l'aide de boulons. Enfin on a procédé à la fixation de la troisième roue mais nous avons rencontré quelques difficultés car il fallait que cette roue soit au même niveau que les 2 roues avant donc nous avons décidé de créer nos propres entretoises pour pouvoir le réaliser. Nous avons coupé une tige filetée pour pouvoir fixer la roue à l'aide d'écrous. Nous pensions que 4 piles de 1,5 volts suffiraient à alimenter le robot, c'est pour cela que nous avons d'abord fixé un support de piles de ce type. Mais au final 6 V n'est pas suffisant pour alimenter le robot donc nous avons décidé de prendre une pile de 9 volts et nous avons par la suite enlevé le support de 4 piles 1,5 volts.



Première assemblage

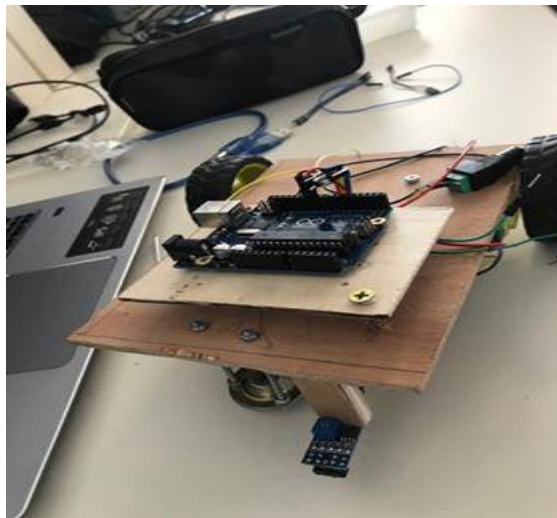
Montage de la roue arrière du robot avec nos entretoises faites maison. Notre robot commence à prendre forme.



Fin de l'assemblage de la partie déplacement

Nous avons fait le montage de la partie déplacement du robot en ajoutant un interrupteur pour pouvoir arrêter le robot quand il sera alimenté par les piles. En effet, le robot est alimenté au début par un adaptateur secteur de 12 Volts pour ne pas utiliser des piles uniquement pour des tests. Nous avons fait quelques tests en Bluetooth avec les programmes fait en TD comme celui pour initialiser le module Bluetooth et celui qui fait tourner les roues dans chaque direction en alternance malheureusement nous avons eu quelques problèmes avec le moteur droit qui ne fonctionnait pas à cause des soudures qui s'étaient défaites. Après avoir corrigé ce problème le robot fonctionnait bien.

Nous avons fixé la carte Arduino à une planche découpée qui a été également fixé à notre robot car les vices fixaient la carte L298 et dépassaient de la planche principale et cela risquait d'endommager la carte Arduino.



Support de la carte Arduino

### **III.3. Gestion des déplacements**

Tout d'abord nous avons fait l'algorithme sur papier pour avoir une vue d'ensemble puis nous l'avons fait sur le logiciel Publisher pour qu'il soit plus propre et clair. Nous avons ajouté à l'algorithme des « Yes / No » à chaque if et nous avons rendu les if plus complet et clair.

Pour le programme nous nous sommes inspirés de ce que nous avons déjà fait en td. On initialise tous les ports nécessaires de l'Arduino puis dans la fonction void loop, on regarde s'il y a un caractère dans le canal BlueT du Bluetooth, si un caractère est disponible il est stocké dans la variable Data puis le programme fait plusieurs tests avec la variable pour voir si elle correspond à un caractère prédéfini si c'est le cas le programme rentre dans la boucle "if" qui correspond et exécute les lignes par exemple si la touche avant de la télécommande est activée alors le smartphone envoie le caractère "F" pour forward. La carte Arduino lit le caractère "F" est fait la partie du programme dans le "if". Il met la direction des deux moteurs vers l'avant puis met la vitesse des moteurs à un PWM de 175. Il affiche également le caractère "F" sur le moniteur série pour qu'en cas de problème nous puissions voir si cette partie marche bien. Nous avons fait une boucle de ce type là pour chaque direction et pour régler la vitesse à un PWM de 175 et une qui met la vitesse à 0 pour arrêter le robot. Pour tourner, le robot arrête le moteur dans la direction dans laquelle on veut tourner et met en marche le moteur opposé.



Télécommande

### **III.4. Montage des capteurs**

Nous avons découpé une planche de bois pour fabriquer des supports pour les détecteurs IR qui détecteront que le robot est bien sur la table et vont permettre de faire un arrêt d'urgence pour prévenir une chute de la table.

Nous avons défini les emplacements des capteurs sur le robot et nous avons fixé les supports avec de la colle mais nous nous sommes rendus compte que les supports des capteurs à l'avant étaient trop longs, nous avons donc dû les détacher et les couper à la bonne taille. Ce fut une opération compliquée à cause de la taille des supports.



Pose des capteurs infrarouge

### **III.5. Gestion des capteurs**

Pour la gestion des capteurs, nous avons dû modifier notre programme en y ajoutant un "if" avant toutes les autres pour que les capteurs soient une priorité. Les capteurs infrarouges renvoient 0V s'il détecte la table et 5V s'il ne la détecte plus, le programme met la valeur reçue

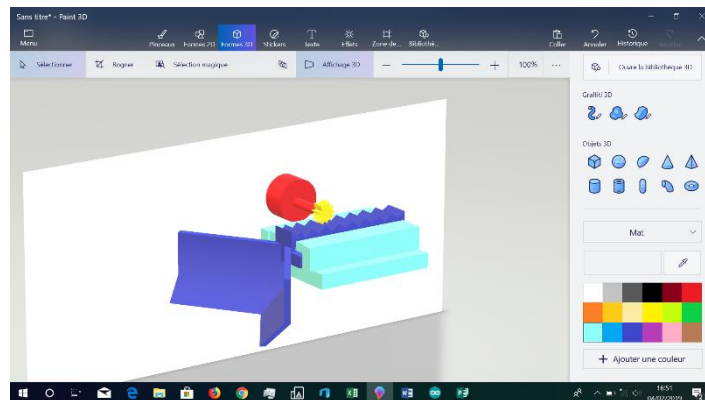


par l'Arduino par chaque capteur dans une variable, si au moins une de ces variables est égale au 1 logique soit 5V le programme rentre dans la boucle et arrête le robot. Le problème avec ce code est que le robot ne peut plus bouger, en effet le capteur reste au-dessus du vide et le robot reste immobile, nous avons donc rajouté une petite marche arrière faite automatiquement. Lorsqu'un des capteurs ne détecte plus la table, le robot fait donc une petite marche arrière d'une demie seconde et revient sur la table, le robot peut de nouveau être contrôlé. Le seul bémol de l'utilisation de capteur infrarouge est que si la table est d'une couleur trop sombre, les capteurs ne vont pas détecter la table et le robot ne sera pas opérationnel. Une solution serait de remplacer les capteurs infrarouges par un capteur à ultrason dirigé vers la table.

## **IV-Partie nettoyage**

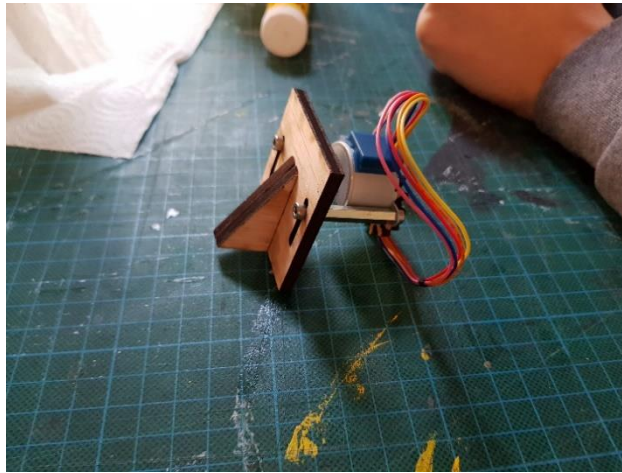
### **IV.1. Fabrication des pièces**

Dans cette partie, nous avons commencé avec une modélisation 3D disponible ci-dessous. Il y a 3 pièces différentes. Un Support, en bleu clair, qui est collé sur le robot et qui dirige la balayette. La 2ème pièce visible sur le modèle 3D en bleu foncé est la balayette qui est collée à un support et à une crémaillère, enfin la 3ème pièce en jaune représente le pignon qui est fixé au moteur pas à pas. Nous pensions d'abord faire ces pièces en plastique imprimer en 3D mais notre modélisation ne pouvait pas être utilisée pour imprimer en 3D en effet cette modélisation a été faite sur le logiciel Paint 3D et il n'est pas possible d'exporter le fichier dans un format utilisable pour une imprimante 3D. Après être passé au Fab Lab nous avons décidé de faire les pièces en bois découpé par la découpeuse laser. C'est bien plus rapide et il ne faut qu'un plan et non une modélisation en 3D pour les faire.



Modélisation 3D de la partie nettoyage

Nous avons d'abord découpé le pignon et la crémaillère puis les pièces pour faire le support qui sont juste des rectangles de différente taille. Enfin on a fait la balayette et son support. Ces deux dernières ont été les plus techniques. Le support car il s'agit d'une pièce unique donc faire le plan était compliqué notamment à cause de l'arrondie. Et pour la balayette, M FORNER du Fab Lab nous a conseillé d'utiliser un procédé qui permet de cintrer du bois. Ce procédé consiste à faire des traits à intervalle régulier sur le bois. Nous avons ensuite fait le support pour le moteur Pas à Pas.



Support du moteur pas à pas

## **IV.2. Montage de la partie nettoyage**

Pour le montage de la partie nettoyage nous avons collé du support entre elle puis nous avons collé le support à notre planche principale avec de la colle à bois. Nous avons également fixé le pignon au moteur pas à pas et la crémaillère à son support. Nous avons collé la balayette à son support et nous avons relié la balayette à la crémaillère. Pour cette étape nous avons dû créer une petite pièce car la surface de contact entre la balayette et la crémaillère était trop faible pour que la colle soit efficace. Nous avons également mis du savon sur le support de la crémaillère pour réduire la friction du mécanisme. Enfin nous avons collé le support du moteur pas à pas au robot.



Collage des pièces de la partie nettoyage

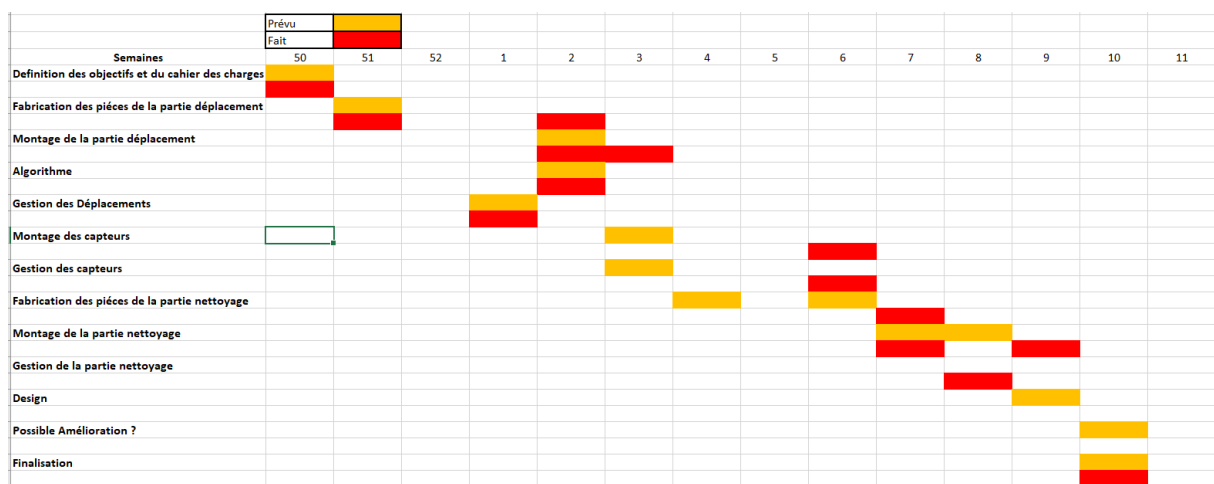
## **IV.3. Gestion de la partie nettoyage**

Pour la gestion du moteur pas à pas nous avons simplement suivi ce qui est indiqué dans le cours de M. Masson. C'est à dire inclure la librairie "Stepper.h" et nous avons dû réduire le nombre de tour du moteur pas à pas pour que la crémaillère ne rentre pas trop dans le robot ou

ne sorte du mécanisme pour cela nous avons pris le nombre de pas qui correspond à un tour du moteur pas à pas et nous l'avons divisé par 360 pour avoir un degré puis nous l'avons multiplié par 280 pour obtenir la bonne distance parcourue par la crémaillère.

## V- Planning

Nous avons été un petit peu optimiste pour le planning mais nous avons plus ou moins réussi à tenir nos délais. Nous n'avons pas eu à se précipiter sur la fin. Cependant nous n'avons pas eu le temps de faire un design plus propre et travaillé.



Le planning

## VI- Conclusion :

Pour conclure, malgré les difficultés que nous avons rencontré lors de notre projet le RNTA fonctionne et remplit la plupart des objectifs qu'on s'est fixés, il se déplace correctement, il est télécommandé par un smartphone via Bluetooth et possède un système de nettoyage qui fonctionne. Reste que plusieurs améliorations sont possibles sur ce robot. Nous pouvons notamment améliorer son design. Nous aurions aussi pu faire un robot plus petit pour qu'il soit plus maniable sur une table et peut-être plus autonome.

## **VII- Bibliographie :**

Durant notre projet on a utilisé les sites suivants :

Le site traceparts pour obtenir les plans pour le pignon et la crémaillère à l'adresse suivante :

<https://www.traceparts.com/en/product/bosch-rexroth-assembly-technologies-pinion?CatalogPath=TRACEPARTS%3ATP01004004&Product=30-05112015-064333&SelectionPath=1%7C1%7C1%7C0%7C>

Et le site de M. Masson pour les cours bien détaillés :

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

## VIII- Annexes :

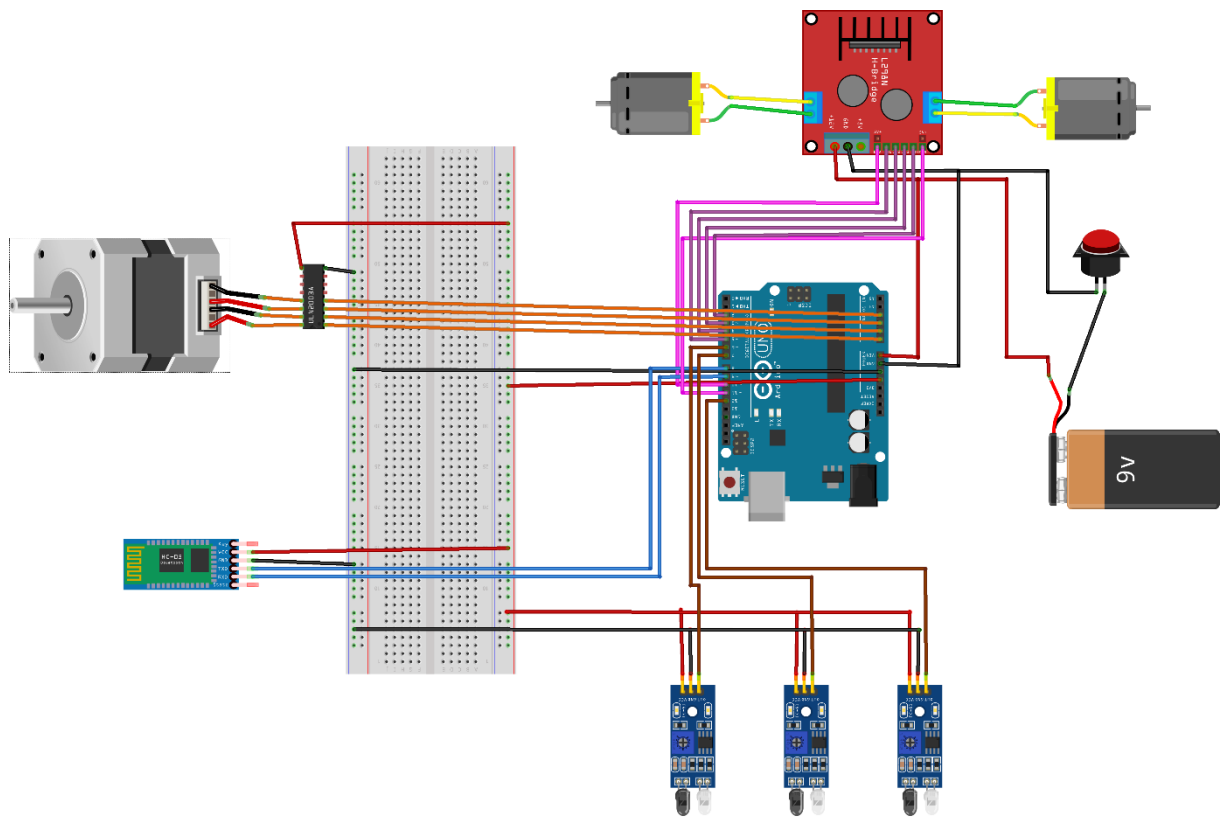
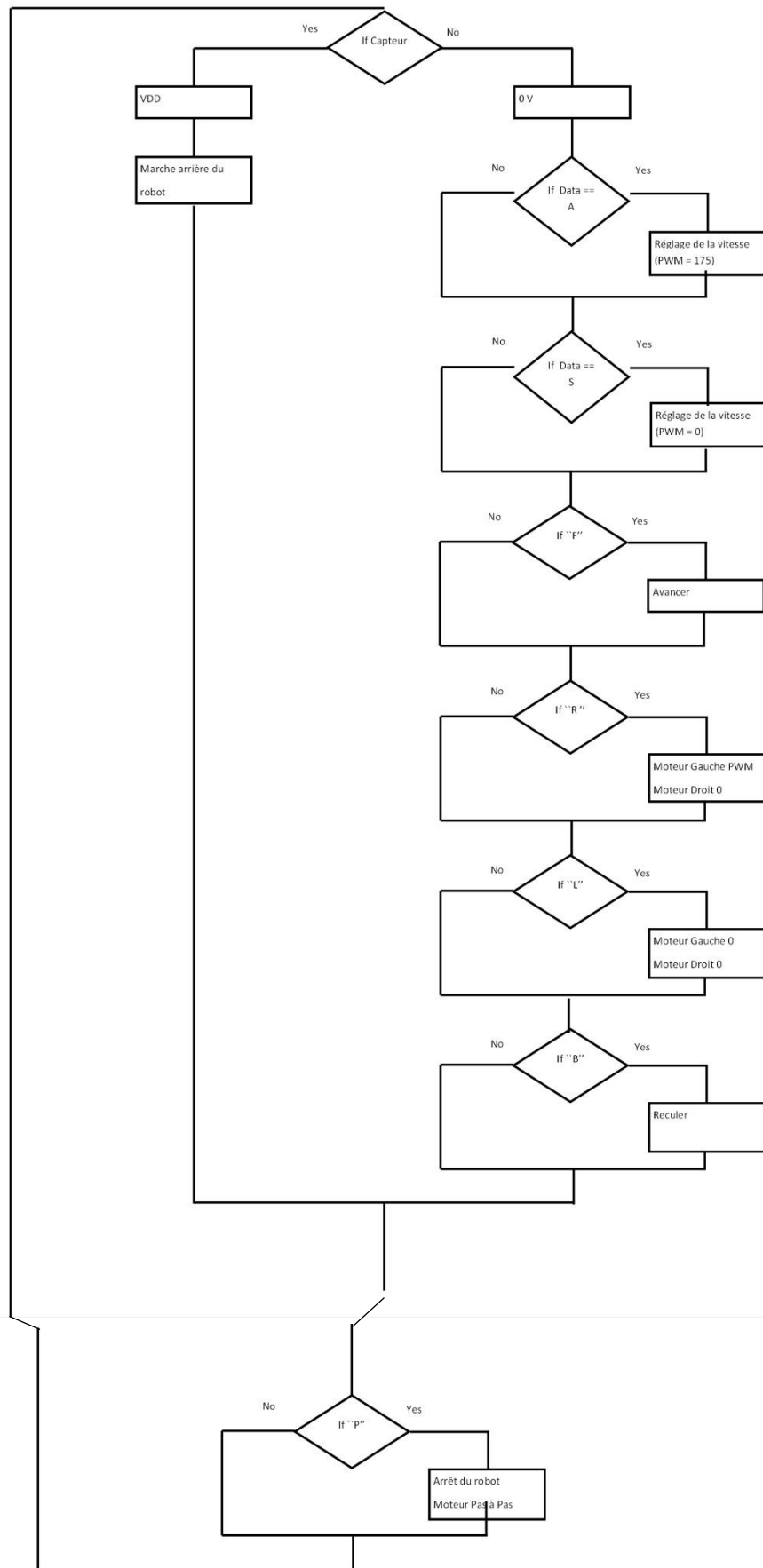


Schéma du montage



**Algorithme**