

## Rapport de Projet: PinkVision

Partie électronique

Réalisé par Pierre-Louis D'AGOSTINO  
& Nicolas SOURNAC  
& Ugo PROIETTI  
& Florian WAUTERS



Faculté  
des Sciences

En collaboration avec CPUMons  
Bachelier en Sciences Informatiques  
Année 2021-2022

## **Abstract**

Ce rapport de projet contient l'ensemble des explications et indications concernant la partie électronique du projet PinkVision.

# Contents

1. Introduction . . . . .	1
2. Démontage . . . . .	1
3. Le moteur . . . . .	1
4. La PWM : Qu'est-ce que c'est ? . . . . .	2
5. Montage . . . . .	3
6. Implémentation Pytho . . . . .	4
7. Le servo moteur . . . . .	4

# 1. Introduction

En ce qui concerne la partie électronique et physique du projet, nous avons décidé d'y accorder une forte importance pour pouvoir ensuite travailler dans les meilleures conditions possibles. Nous avons d'abord identifié plusieurs étapes à réaliser :

1. Démonter la voiture si possible.
2. Répertorier les différents composants (moteur, servo moteur, etc).
3. Créer un support pouvant contenir les différents circuits imprimés.
4. Réfléchir à un montage de câbles.

# 2. Démontage

Lors de notre première rencontre, nous avons démonté la voiture, partie qui ne fut pas laborieuse étant donné la bonne qualité du modèle. Chaque élément de la voiture a pu être retiré facilement, ce qui nous a permis d'accéder directement au servo moteur et au moteur CC.

# 3. Le moteur

Pour pouvoir utiliser le moteur CC (à courant continu), nous avons été contraint d'utiliser un **pont en H** qui est un composant électronique permettant de contrôler le sens de rotation de celui-ci via un signal PWM envoyé depuis la Raspberry. Ce composant permet également d'éviter que tout courant induit par le moteur n'endommage la Raspberry si celle-ci était directement connectée au moteur CC.

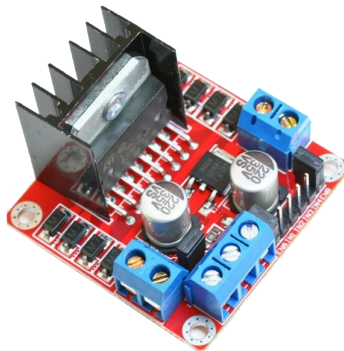


Figure 1: Pont en H (ref: L298N)

## 4. La PWM : Qu'est-ce que c'est ?

La PWM (pulse with modulation) permet d'envoyer des impulsions de tension (High ou Low) au moteur CC en passant par le pont en H.

Deux valeurs sont à prendre en compte lors de l'utilisation du PWM : **la fréquence** et le **rapport cyclique**.

Il est important de bien déterminer la fréquence du signal PWM, cette valeur est fixe et dépend du moteur utilisé. En revanche, le rapport cyclique varie entre 0 et 100% et représente le pourcentage de temps où le signal envoyé sera en "High". Celui-ci sert alors à contrôler la vitesse de rotation du moteur. Plus cette valeur est élevée, plus le moteur tourne vite. Ci-dessous se trouvent deux représentations graphiques de la PWM :

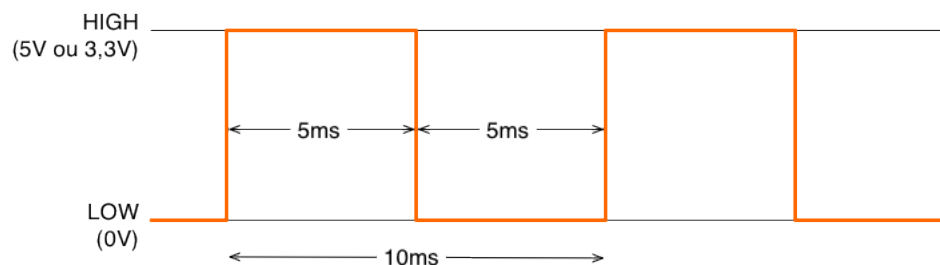


Figure 2: PWM à 50%

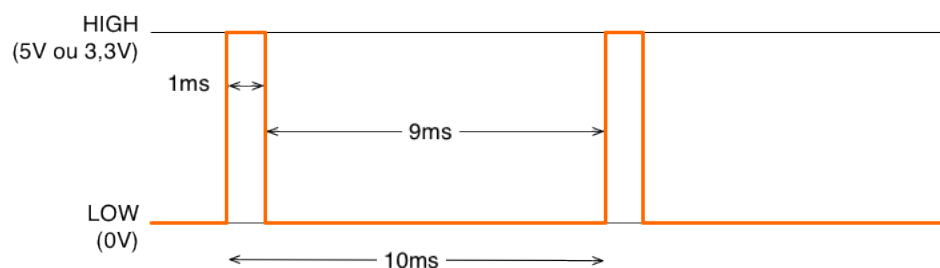


Figure 3: PWM à 10%

## 5. Montage

En ce qui concerne le montage effectué sur la RaspBerry, deux pin GPIO sont utilisés pour commander le sens de rotation du moteur et un pin GPIO réservé à la PWM est utilisé afin d'envoyer le signal de tension PWM.

Le pont en H est branché à la batterie (pour être alimenté), au moteur (pour lui indiquer le sens de rotation et transmettre le signal PWM), la RaspBerry (pour recevoir le signal PWM).

Le servo moteur est quant à lui directement branché à la RaspBerry. Le point suivant énonce l'utilisation que nous faisons du servo moteur.

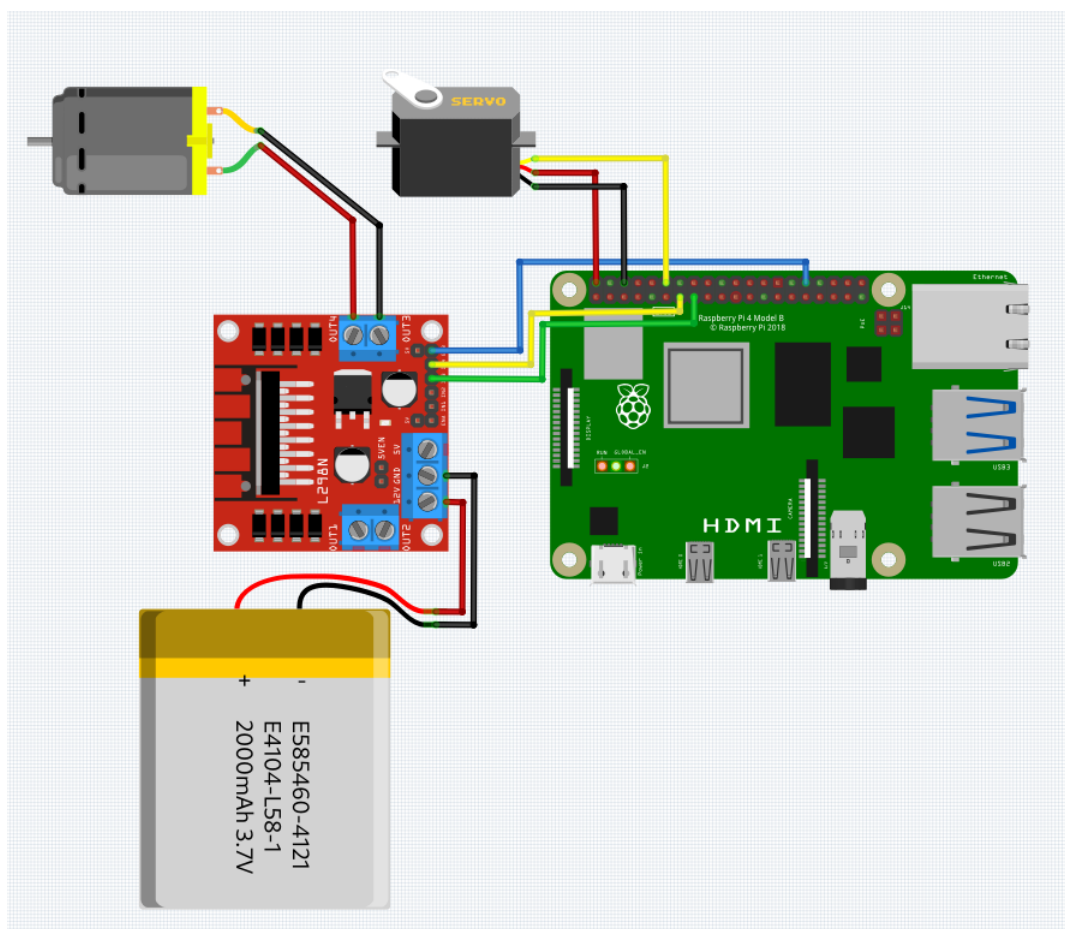


Figure 4: Schéma du montage/câblage

## 6. Implémentation Python

Au niveau de l'implémentation bas niveau, le contrôle des pin GPIO ainsi que du PWM peut se faire via des scripts en Python. La PWM fonctionne comme un objet possédant deux attributs importants : la fréquence et le rapport cyclique. Les méthodes `ChangeFrequency()` et `ChangeDutyCycle()` permettent de modifier la fréquence ainsi que le rapport cyclique. Notre implémentation utilise une adaptation du design pattern "State pattern" dont le diagramme de classe est visible ci-dessous. Nous avons fait le choix d'utiliser ce design pattern en raison de sa bonne cohérence avec le problème et de sa modularité au niveau de la gestion et de l'ajout de nouveaux états. L'implémentation de cette couche bas niveau sera sans doute amenée à être améliorée par la suite afin d'affiner le contrôle de la voiture. Il est à noter que la communication avec la voiture depuis un appareil externe (Ordinateur, téléphone) s'effectue via un réseau wifi émit depuis la raspberry.

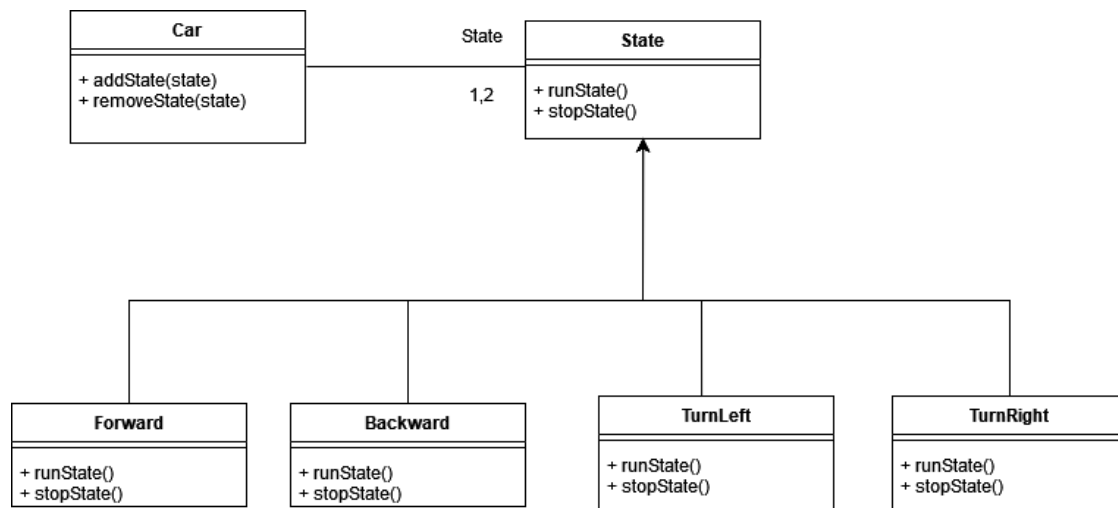


Figure 5: Diagramme de classe

## 7. Le servo moteur

Le servo moteur est directement piloté via les pin GPIO de la RaspBerry, et encore une fois nous allons utiliser la PWM. La fréquence standard pour utiliser un servo moteur est de 50 Hz.

Il est clair que l'orientation du servo moteur est déterminée par le rapport cyclique.

Nous pouvons faire varier l'angle du moteur de 0 à 180 degrés, pour les besoins d'orientation de la voiture nous n'utilisons que 90 degrés d'envergure de rotation.

La position initiale des roues se situe donc lorsque l'angle du moteur est de 90 degrés.

La position droite se situe alors lorsque l'angle du moteur est de 45 degrés et la position gauche lorsque l'angle du moteur est de 135 degrés.

Si vous souhaitez trouver plus d'informations sur les servo moteurs liés à une RaspBerry, [voici un pdf](#) regroupant beaucoup d'informations que nous avons utilisées.



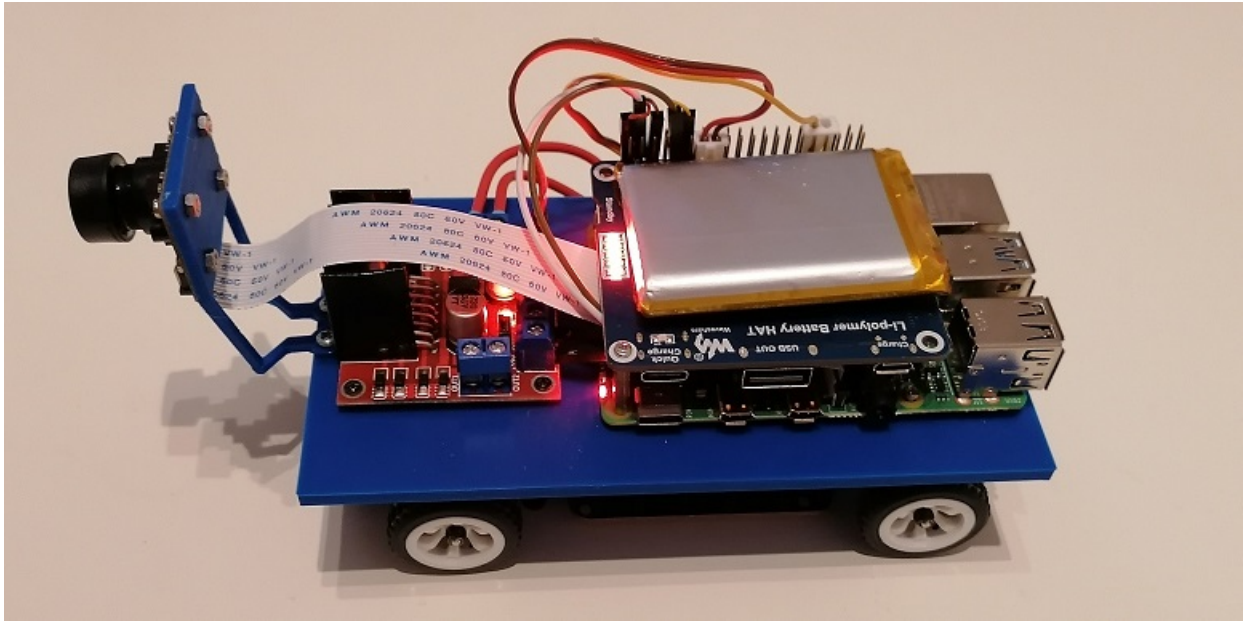


Figure 6: La PinkVision