

## Gestion de Projet 2024-2025

# RAPPORT TECHNIQUE COMPOSANTS SÉLECTIONNÉS ET ALTERNATIVES

Année Académique 2024-2025

Groupe N°10

*CATAKLI Samet*

*BODJONA Kevin*

*LALLEMENT Corentin*

*Lubanguku Esdras*

*MOTTE Yann*

---

## Table des matières

Table des matières .....	2
1. Raspberry Pi 3 B.....	3
2. HC-SR04 (x3) .....	3
3. Capteur IR (Line Follower) .....	4
4. Servomoteurs () .....	4
5. Moteurs DC (x2) .....	5
6. Driver L298N.....	5
7. Driver PCA9685.....	5
8. INA219 .....	6
9. TCS3472 (Capteur RGB) .....	6
10. PiJuice .....	6
11. XL1509 .....	7
12. Tableau récapitulatif .....	7
Conclusion .....	9

---

## 1. Raspberry Pi 3 B

### Pourquoi celui-là :

- Suffisant pour gérer plusieurs capteurs et moteurs en simultané.
- Intègre nativement le WiFi, le Bluetooth et propose un port GPIO complet.
- Supporte Python et des bibliothèques spécialisées (GPIO Zero, pigpio, etc.).
- Large compatibilité avec divers HAT (PiJuice, PCA9685...).

### Pourquoi pas un autre :

- **Arduino** : Capacités de calcul et de connectivité limitées pour des tâches avancées ou de multiples capteurs.
- **Raspberry Pi 4** : Plus onéreux, génère davantage de chaleur et peut se révéler surdimensionné pour un simple pilotage.
- **ESP32** : Très performant pour l'embarqué, mais moins intuitif à configurer pour un déploiement rapide, et plus exigeant en gestion bas-niveau.

---

## 2. HC-SR04 (x3)

### Pourquoi celui-là :

- Excellent rapport qualité-prix.
- Facile à interfacer.
- Précision suffisante pour la plupart des applications robotiques.
- Permet une détection frontale, latérale ou arrière (anti-collision) en installant plusieurs modules.

### Pourquoi pas un autre :

- **VL53L0X/L1X (ToF laser)** : Plus précis, mais plus onéreux, portée plus réduite et sensible à la lumière ambiante.
- **Capteur infrarouge analogique** : Fiabilité moindre, fortement influencé par la nature de la surface ciblée.
- **Radar Doppler** : Solution coûteuse et complexe, souvent disproportionnée pour de la simple détection d'obstacle.

---

### 3. Capteur IR (Line Follower)

Pourquoi celui-là :

- Installation simple (souvent trois fils).
- Réactivité suffisante pour différencier clairement le noir du blanc.
- Conçu spécifiquement pour le suivi de ligne.

Pourquoi pas un autre :

- **Caméra OpenCV** : Complexité de traitement trop importante sur un Raspberry Pi 3, peut entraîner des ralentissements.
- **Capteurs à effet Hall** : Adaptés à la détection d'aimants ou de rotation, non pertinents pour le suivi visuel d'une ligne.
- **Capteurs capacitifs** : Non adaptés au suivi d'une ligne peinte ou dessinée, car ils ne détectent pas la réflexion lumineuse.

---

### 4. Servomoteurs ()

Pourquoi ceux-là :

- Faciles à contrôler par un simple signal PWM.
- Recommandés pour des mouvements angulaires précis (pour un bras robotique, par exemple).
- Compatibles avec le PCA9685 (permet une gestion indépendante du PWM pour plusieurs servos).

Pourquoi pas un autre :

- **Moteur pas-à-pas** : Plus complexe à piloter (drivers spécifiques), plus coûteux et plus lourd.
- **Moteurs DC + encodeurs** : Exigeant en termes de programmation pour des angles précis, souvent excessif si l'on ne vise que 0 à 180°.
- **Servomoteurs à retour d'état** : Plus onéreux, utiles seulement pour des applications très exigeantes en matière de feedback.

---

## 5. Moteurs DC (x2)

Pourquoi ceux-là :

- Solutions simples et puissantes pour la propulsion d'un robot.
- Compatibles avec les contrôleurs standards tels que le L298N.
- Peu coûteux et rapidement remplaçables en cas de panne ou d'évolution du projet.

Pourquoi pas un autre :

- **Moteurs brushless** : Nécessitent un contrôleur ESC, plus chers et moins faciles à mettre en œuvre dans des montages basiques.
- **Moteurs pas-à-pas** : Consomment davantage, conception plus lourde, pertinents pour des positions ultra-précises.
- **Servos à rotation continue** : Généralement moins puissants et offrant un contrôle de vitesse moins précis.

---

## 6. Driver L298N

Pourquoi celui-là :

- Pilote classique, gère deux moteurs DC.
- Facile à trouver et à alimenter, documentation abondante.
- Accepte une alimentation externe adaptée aux moteurs.

Pourquoi pas un autre :

- **L293D** : Moins performant, chauffe plus.
- **TB6612FNG** : Meilleure efficacité mais parfois moins disponible que le L298N.
- **Contrôleur brushless** : Inadapté à des moteurs DC.

---

## 7. Driver PCA9685

Pourquoi celui-là :

- Permet de générer jusqu'à 16 canaux de PWM de façon indépendante sur une liaison I2C.
- Décharge le Raspberry Pi des calculs de PWM, limitant ainsi les risques de conflits système.
- Suffisamment de canaux pour les trois servomoteurs, voire d'autres extensions futures.

Pourquoi pas un autre :

- **PWM logiciel via GPIO** : Instable sur Raspberry Pi à cause de l'ordonnancement multitâche.

- **Contrôle direct via pigpio** : Possible, mais limite le nombre de canaux PWM disponibles.
- **HAT Servo Adafruit** : Cher, même s'il repose sur le même composant.

## 8. INA219

**Pourquoi celui-là :**

- Mesure de courant et de tension par le bus I2C, pratique pour la télémétrie et la sécurité.
- Large diffusion et documentation, adapté au monitoring de la consommation de moteurs ou d'une batterie.

**Pourquoi pas un autre :**

- **ACS712** : Sortie analogique nécessitant un convertisseur ADC supplémentaire sur un Raspberry Pi.
- **INA260** : Plus précis, mais plus onéreux.
- **Capteurs Hall maison** : Moins précis et demande un travail de calibration plus élaboré.

## 9. TCS3472 (Capteur RGB)

**Pourquoi celui-là :**

- Capteur de couleur complet (R, G, B, Clear), adapté à la détection de signaux ou d'objets colorés.
- Interface I2C, simplifiant la communication avec le Raspberry Pi.
- Souvent employé en robotique pour des tâches de reconnaissance de couleurs ou de tri.

**Pourquoi pas un autre :**

- **TCS3200** : Produit une fréquence proportionnelle à l'intensité lumineuse, plus compliqué à interpréter.
- **Caméra et traitement couleur** : Sur Raspberry Pi 3, requiert une charge de calcul importante.
- **Photodiodes RGB distinctes** : Implémentation plus complexe, nécessite un circuit spécifique pour chaque canal.

## 10. PiJuice

**Pourquoi celui-là :**

- Carte d'alimentation (HAT) intégrant une batterie, assurant la fonction d'UPS.

- Gestion automatisée de la charge et possibilité de contrôles avancés (ex. extinction propre, monitoring).
- Format compact, s'adapte parfaitement sur le Raspberry Pi.

**Pourquoi pas un autre :**

- **PowerBank + câble USB** : Ne propose pas de télémétrie ni de contrôle logiciel.
- **UPS HAT génériques** : Qualité et documentation très variables.
- **Batterie LiPo + régulateur** : Implique un câblage et une programmation plus complexes pour obtenir un vrai UPS.

## 11. XL1509

**Pourquoi celui-là :**

- Régulateur step-down (buck) capable de convertir par exemple 12 V vers 5 V ou 3,3 V.
- Format relativement compact, prend en charge des courants pouvant aller jusqu'à 2 A.

**Pourquoi pas un autre :**

- **LM2596** : Module plus volumineux et légèrement moins efficace.
- **Régulateur linéaire (7805)** : Dissipation thermique importante, gaspillage d'énergie.
- **Alimentation USB standard** : Ne garantit pas systématiquement une stabilité suffisante pour alimenter des moteurs en parallèle.

## 12. Tableau récapitulatif

Composant	Pourquoi ce choix ?	Pourquoi pas un autre ?
Raspberry Pi 3 B	Polyvalent, facile, GPIO complet	ESP32/Arduino limités ou plus complexes

Composant	Pourquoi ce choix ?	Pourquoi pas un autre ?
HC-SR04 (x3)	Bon rapport qualité-prix, fiable, simple	ToF plus onéreux, IR moins fiable
Capteur IR (Line Follower)	Rapide, simple, efficace	Caméra trop lourde, autres capteurs inadaptés
Servomoteurs (x3)	Mouvement angulaire précis	Stepper trop complexe, servos feedback coûteux
Moteurs DC (x2)	Simple, parfaits pour la propulsion	Brushless/Stepper plus chers ou complexes
L298N	Contrôleur classique, facile à trouver	TB6612FNG plus efficace, moins courant
PCA9685	16 canaux PWM indépendants	PWM logiciel instable sur Raspberry Pi
INA219	Mesure tension/courant via I2C	Alternatives analogiques moins pratiques



Composant	Pourquoi ce choix ?	Pourquoi pas un autre ?
TCS3472	Détection RGB complète, I2C	Caméra trop lourde à gérer
PiJuice	Alimentation UPS dédiée	PowerBank : aucun contrôle logiciel
XL1509	Régulateur efficace, compact	7805 gaspille l'énergie et chauffe

## Conclusion

Chaque composant retenu présente une combinaison optimale de fiabilité, de simplicité d'intégration et de cohérence technique pour un projet utilisant un Raspberry Pi 3 B. Les solutions écartées l'ont été en raison de coûts, de complexités de mise en œuvre ou de spécifications inadaptées aux besoins visés.