



Rapport du Projet

Projet Pluridisciplinaires 2CS
Voiture robot télécommandée à base
web

10-06-2024

Équipe :

- Mohamed Miloudi
- Brahim Halloucha Djebbour
- Mohamed Amine Benchaa
- Mohamed Chakib Dhiya Eddine Ouzane

Encadrants :

- Mr Rahmoun Abdellatif
- Mr Bensenane hamdane

Table des matières

1	Introduction	3
2	Objectifs du Projet	4
3	Définitions et Abréviations	4
4	Références	5
5	Description Générale	6
5.1	Perspective du Produit	6
5.2	Comité de la Maîtrise d’Ouvrage	6
5.3	Comité de la Maîtrise d’Œuvre	6
5.4	Fonctionnalités du Produit	7
6	Composants du Projet	8
6.1	Système Embarqué	8
6.2	Interface Utilisateur	9
7	Diagrammes et Schémas	11
7.1	Diagramme d’Architecture du Systeme	11
7.2	les Diagrammes de sequence	12
7.2.1	Diagramme de sequence de capture d’une video et détection d’objets	12
7.2.2	Diagramme de sequence de controle de mouvement de la voiture	13
7.2.3	Diagramme de sequence de controle de rotation de la camera	14
8	Planification du Projet	15
8.1	Phase 1 : Conception et Planification	15
8.2	Phase 2 : Développement du Système Embarqué	15
8.3	Phase 3 : Développement de l’Interface Web	15
8.4	Phase 4 : Integration de l’Algorithme YOLO8	15
8.5	Phase 5 : Ajout de Fonctionnalités secondaires	15
8.6	Phase 6 : Tests	16
9	Conclusion	16

1 Introduction

Ce cahier des charges décrit en détail un projet de groupe visant à concevoir une voiture télécommandée en ligne (**Remote Controlled Car**), avec des fonctionnalités de collecte, de traitement et de visualisation en temps réel des données du véhicule. Le projet comprend le développement de composants matériels et logiciels interconnectés, afin de fournir une solution complète pour surveiller et contrôler à distance la voiture.

Le système embarqué repose sur une carte ESP32 qui sert de cerveau au véhicule, gérant les commandes de mouvement, les rotations du servo moteur, et le contrôle de la lampe de poche (flash light). Le serveur Web intégré à l'ESP32 permet à l'utilisateur de se connecter via un navigateur et d'accéder à une interface intuitive comprenant des boutons pour avancer, reculer, tourner à gauche, tourner à droite, contrôler la caméra attachée au servo moteur, et activer ou désactiver la lampe de poche. Cette interface utilisateur, montrée à l'utilisateur dans le navigateur pour contrôler la voiture robot, rend le pilotage du véhicule simple et efficace.

Afin d'assurer une navigation sécurisée, le véhicule est équipé d'un capteur ultrasonique (modèle HC-SR04) qui mesure la distance des obstacles à l'avant, envoyant ces informations à la carte ESP32 pour éviter les collisions. De plus, pour enrichir l'expérience de l'utilisateur, le système intègre une technologie avancée de détection d'objets alimentée par l'intelligence artificielle. Cette fonctionnalité repose sur l'algorithme YOLO8, où les images capturées par la caméra du véhicule sont transmises à une machine distante ou à un serveur Google Cloud Platform (GCP). L'algorithme analyse les images en temps réel et envoie les résultats au navigateur avec l'image capturée par la caméra de l'ESP32, permettant à l'utilisateur de visualiser les objets détectés directement depuis son navigateur.

Cependant, la fonctionnalité de détection d'objets consomme une grande quantité de ressources, notamment la bande passante de la communication avec le navigateur Web et la capacité de traitement du processeur de l'ESP32. Cela peut entraîner un ralentissement du flux vidéo, une diminution de la qualité des images, et une surchauffe de la carte ESP32. Pour pallier ces problèmes, un bouton permettant de désactiver la détection d'objets a été ajouté à l'interface, offrant ainsi une expérience vidéo plus fluide lors des démonstrations.

En résumé, ce projet combine des composants matériels et logiciels pour créer une voiture télécommandée intelligente, capable de fournir des données en temps réel et d'intégrer des fonctionnalités avancées d'IA pour la détection d'objets, tout en garantissant une expérience utilisateur optimale.

2 Objectifs du Projet

Les objectifs principaux de ce projet sont multiples. Tout d'abord, nous visons à fournir une interface conviviale et intuitive pour rendre le pilotage du véhicule simple et efficace. De plus, notre objectif est d'assurer une disponibilité totale du système, permettant aux utilisateurs de piloter la voiture à tout moment et depuis n'importe où. et ce Robot Car peut être utilisée dans de nombreux cas d'utilisation, qu'il s'agisse de loisirs, de surveillance à distance ou de toute autre application.

En résumé, notre objectif est de créer un système de pilotage à distance d'un **Robot car** qui fonctionne **anytime anywhere**.

3 Définitions et Abréviations

- **YOLO** : "You Only Look Once", une technique de détection d'objets en temps réel.
- **YOLO8** : Une version améliorée de l'algorithme YOLO pour la détection et la reconnaissance d'objets.
- **Servo moteur** : Un moteur électrique qui permet un contrôle précis de la position angulaire.
- **ESP32** : Une carte microcontrôleur à faible coût avec Wi-Fi et Bluetooth intégrés, utilisée pour les projets d'IoT.
- **ESP32 CAM** : Une version de l'ESP32 avec une caméra intégrée, utilisée pour les applications de vision par ordinateur.
- **Moteur DC** : Un moteur à courant continu, souvent utilisé dans les applications où un contrôle de vitesse variable est nécessaire.
- **Arduino Motor Shield (L298N Motor Driver)** : Un module complémentaire pour Arduino permettant de contrôler des moteurs à courant continu et des moteurs pas à pas.
- **GCP** - *Google Cloud Platform*, une suite de services cloud proposés par Google, comprenant le stockage de données, le calcul, l'analyse de données, l'apprentissage automatique et bien plus encore.

4 Références

- l'article: **ESP32 with DC Motor and L298N Motor Driver – Control Speed and Direction** sur randomnerdtutorials.com
- l'article: **ESP32 Web Server** sur randomnerdtutorials.com

5 Description Générale

5.1 Perspective du Produit

Le produit final sera une voiture robot dotée d'une interface conviviale et intuitive pour un pilotage simple et efficace, avec une disponibilité totale permettant de la contrôler à tout moment et depuis n'importe où. Cette voiture robot pourra être utilisée dans de nombreux cas, tels que les loisirs, la surveillance à distance ou d'autres applications. En résumé, notre objectif est de créer un système de pilotage à distance fonctionnant anytime anywhere.

5.2 Comité de la Maîtrise d'Ouvrage

Nom et Prénom	Fonction	Email
Mohamed Miloudi	chef de groupe et Développeur	mo.miloudi@esi-sba.dz
Brahim Halloucha Djebbour	Développeur	b.hallouchadjebbour@esi-sba.dz
Mohamed Amine Benchaa	Développeur	ma.benchaa@esi-sba.dz
Mohamed Chakib Ouzane	Développeur	mc.ouzane@esi-sba.dz

Table 1: Liste des noms des membres d'équipe avec leurs fonctions et emails

5.3 Comité de la Maîtrise d'Œuvre

Nom et Prénom	Email
Djamel Amar Bensaber	d.amarbensaber@esi-sba.dz
Abdellatif Rahmoun	a.rahmoun@esi-sba.dz
hamdane bensenane	h.bensenane@esi-sba.dz

Table 2: Liste des noms et emails des membres de la Comité de la Maîtrise d'Ouvrage

5.4 Fonctionnalités du Produit

- Il y a un serveur Web ESP32 dans la carte, avec une interface qui comprend ces boutons :
 - avancer, reculer, tourner à gauche, tourner à droite: afin que l'utilisateur puisse contrôler le mouvement du robot car.
 - deux boutons pour tourner le servo moteur (rotation à gauche, rotation à droite) auquel la caméra (avec la carte) est attachée, permettant une vue à 180 degrés sans avoir à tourner ou déplacer la voiture
 - un bouton de torch (flash light) afin que l'utilisateur puisse voir dans l'obscurité.
 - L'utilisateur peut contrôler ces fonctions à distance depuis l'interface de son navigateur.
- Détection d'obstacles avec un capteur ultrasonique.
- Détection d'objets avec YOLO8, où le serveur envoie un flux d'images à une machine distante (dans notre cas, un de nos ordinateurs ou à un serveur Google Cloud Platform (GCP)). La machine distante exécute l'algorithme sur l'image et envoie les résultats (les objets détectés) au navigateur avec l'image capturé par la camera de l'ESP32, que l'utilisateur peut voir depuis son navigateur.
- La voiture robot est contrôlée à distance, permettant à l'utilisateur de la contrôler à tout moment et en tout lieu (**anytime anywhere**).

6 Composants du Projet

6.1 Système Embarqué

- **ESP32 :**
 - reçoit les commandes de mouvement de l'utilisateur et les envoie au Motor Shield (L298N Motor)
 - reçoit les commandes de rotation du servo moteur de l'utilisateur et les envoie au servo moteur
 - reçoit les signaux de la lampe de poche de l'utilisateur et contrôle la lampe de poche sur la carte ESP32
 - envoie le flux d'images (vidéo) capturées par la caméra à la machine distante (notre machine ou le serveur GCP "Google Cloud Platform") pour la détection d'objets
- **Arduino Motor Shield (L298N Motor) :**
 - reçoit les commandes de la carte ESP32 et contrôle les DC motors (chaque deux moteurs sont indépendants)
- **Moteurs DC :** les deux moteurs droits se tourne ensemble, et les deux moteurs gauches se tourne ensemble afin de réaliser différents mouvements :
 - avancer : moteurs droits en avant et moteurs gauches en avant
 - reculer : moteurs droits en arrière et moteurs gauches en arrière
 - tourner à gauche : moteurs droits en avant et moteurs gauches en arrière
 - tourner à droite : moteurs droits en arrière et moteurs gauches en avant
- **Servo Moteur :** reçoit les commandes de l'ESP32 et tourne à droite/-gauche pour une vue d'un demi-cercle (vue de 180 degrés)
- **Capteur Ultrasonique :** du modèle HC-SR04 qui mesure la distance disponible devant la Voiture Robot et envoie cette distance à la carte ESP32. Si la mesure est inférieure à un certain seuil, un obstacle est détecté

- **Machines Distantes** : Une machine personnelle et un serveur GCP (Google Cloud Platform)
 - reçoivent le flux d'images (vidéo) capturées par la caméra depuis le serveur ESP32
 - exécutent l'algorithme YOLO8 pour la détection d'objets sur ces images
 - envoient le flux d'images (vidéo) capturées par la caméra ainsi que les objets détectés au navigateur de l'utilisateur

6.2 Interface Utilisateur

- L'interface utilisateur comprend plusieurs boutons permettant de contrôler différentes fonctions de la voiture robot.
- Les boutons de direction :
 - Flèche vers le haut : pour avancer
 - Flèche vers le bas : pour reculer
 - Flèche gauche : pour tourner à gauche
 - Flèche droite : pour tourner à droite
 - Carré au centre : pour arrêter le mouvement
- Les boutons du servo moteur :
 - **Servo Left** : pour tourner le servo moteur vers la gauche
 - **Servo Right** : pour tourner le servo moteur vers la droite
- Le bouton de la lampe de poche :
 - **Flash** : pour allumer ou éteindre la lampe de poche afin que l'utilisateur puisse voir dans l'obscurité

- Le bouton de désactivation détection d'objets :
 - **Disable obj detection** : pour désactiver la détection d'objets. Nous avons ajouté ce bouton car le flux de données (le flux d'images et les objets détectés) consommait trop de nos ressources (la bande passante de la communication avec le navigateur Web et l'unité de calcul 'processeur' de l'ESP32), ce qui entraînait un ralentissement de la vidéo (seulement deux images par seconde "2fps"), une diminution de la qualité de la vidéo et une surchauffe de la carte ESP32. Nous avons donc ajouté ce bouton pour une expérience vidéo plus fluide lors de la démonstration.

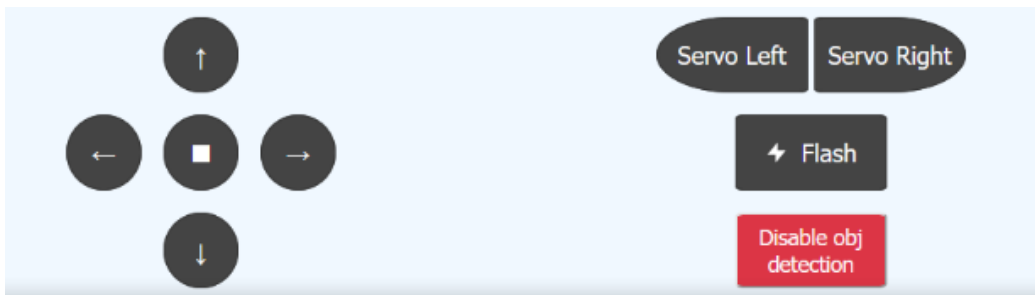


Figure 1: l'interface montrée à l'utilisateur dans le navigateur pour contrôler la voiture robot

7 Diagrammes et Schémas

7.1 Diagramme d'Architecture du Systeme

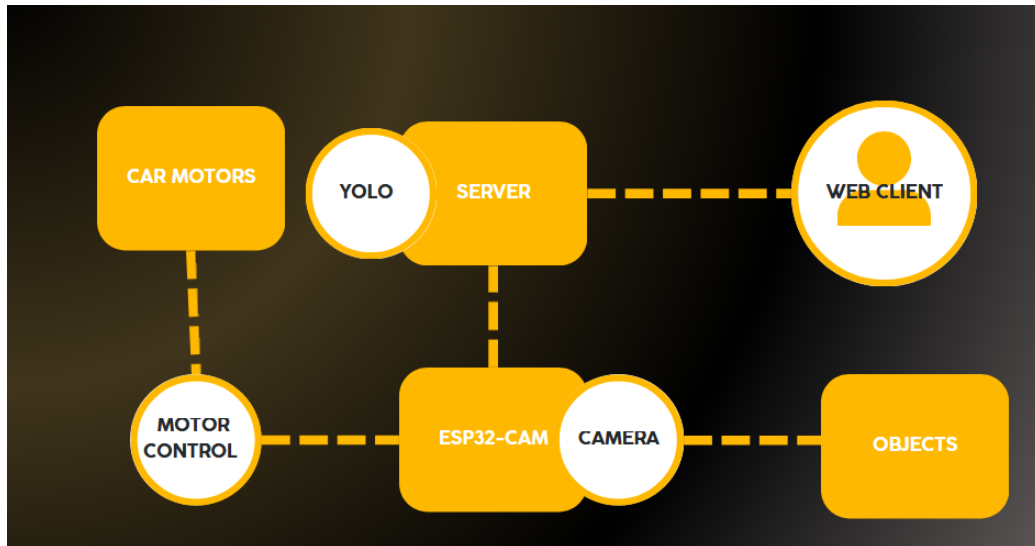


Figure 2: Diagramme d'Architecture du Systeme

7.2 les Diagrammes de sequence

7.2.1 Diagramme de sequence de capture d'une video et détection d'objets

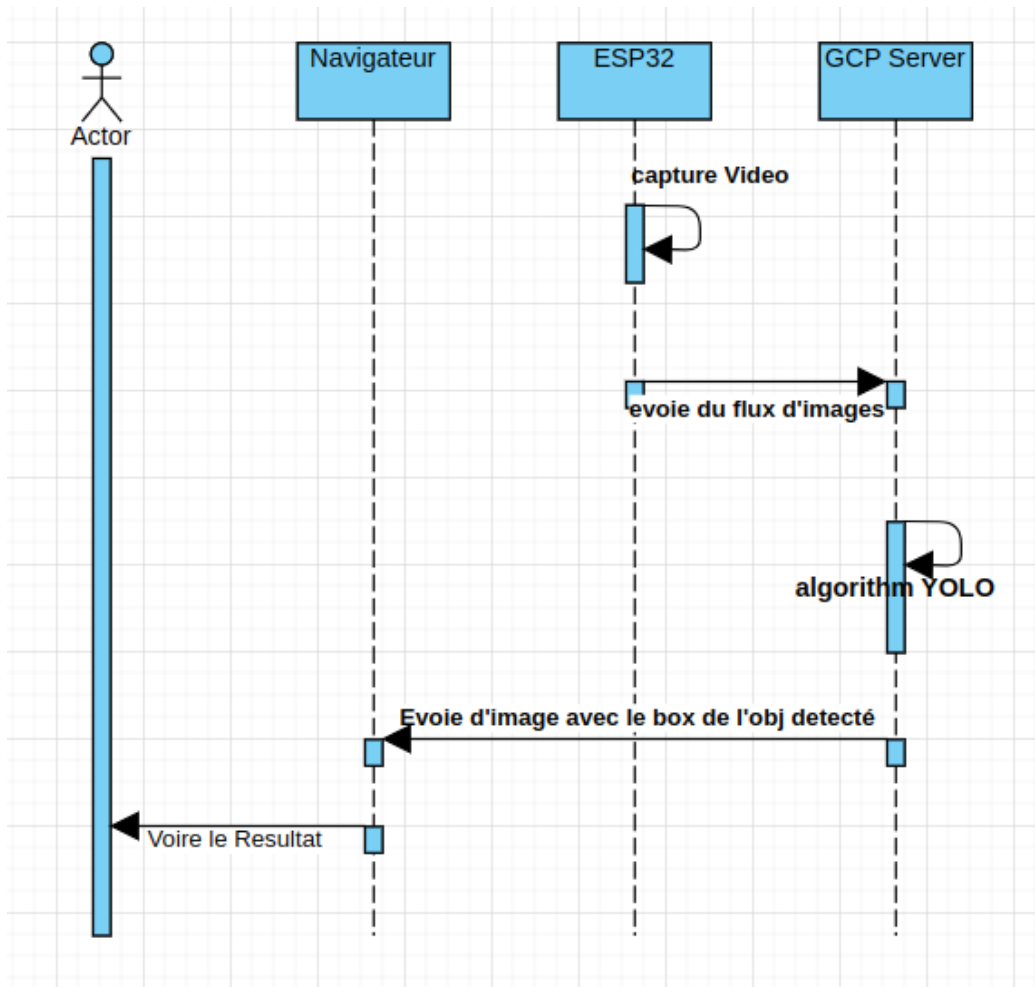


Figure 3: Diagramme de sequence de capture d'une video et détection d'objets

7.2.2 Diagramme de sequence de controle de mouvement de la voiture

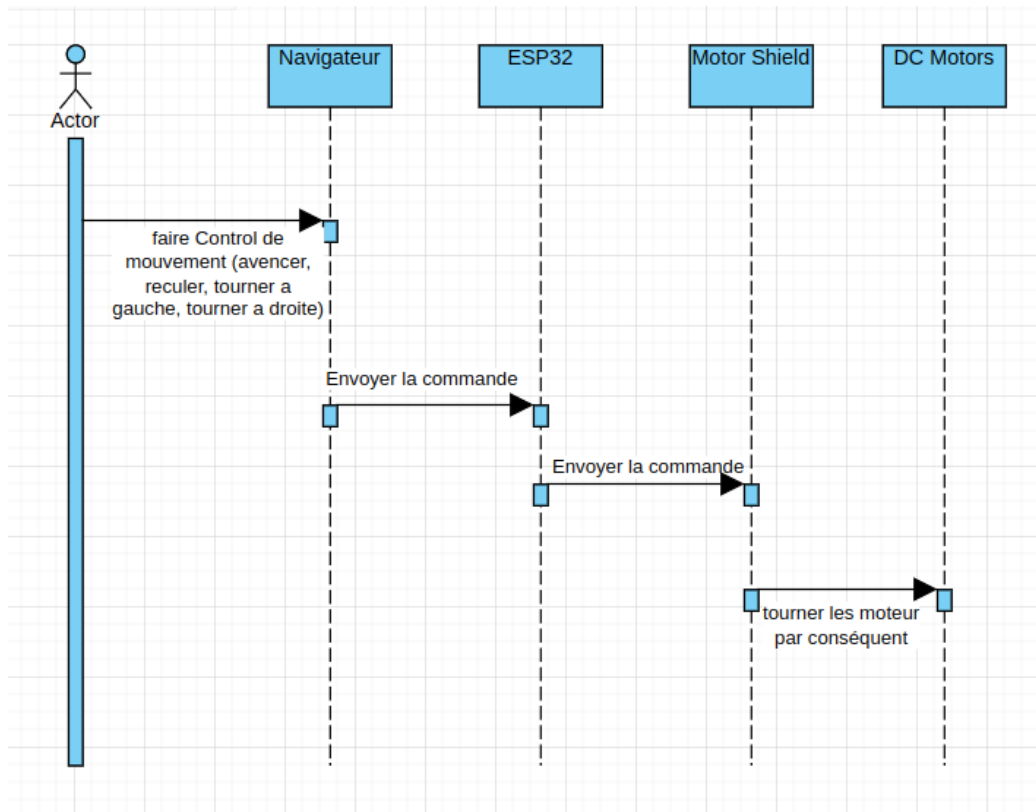


Figure 4: Diagramme de sequence de controle de mouvement de la voiture

7.2.3 Diagramme de sequence de controle de rotation de la camera

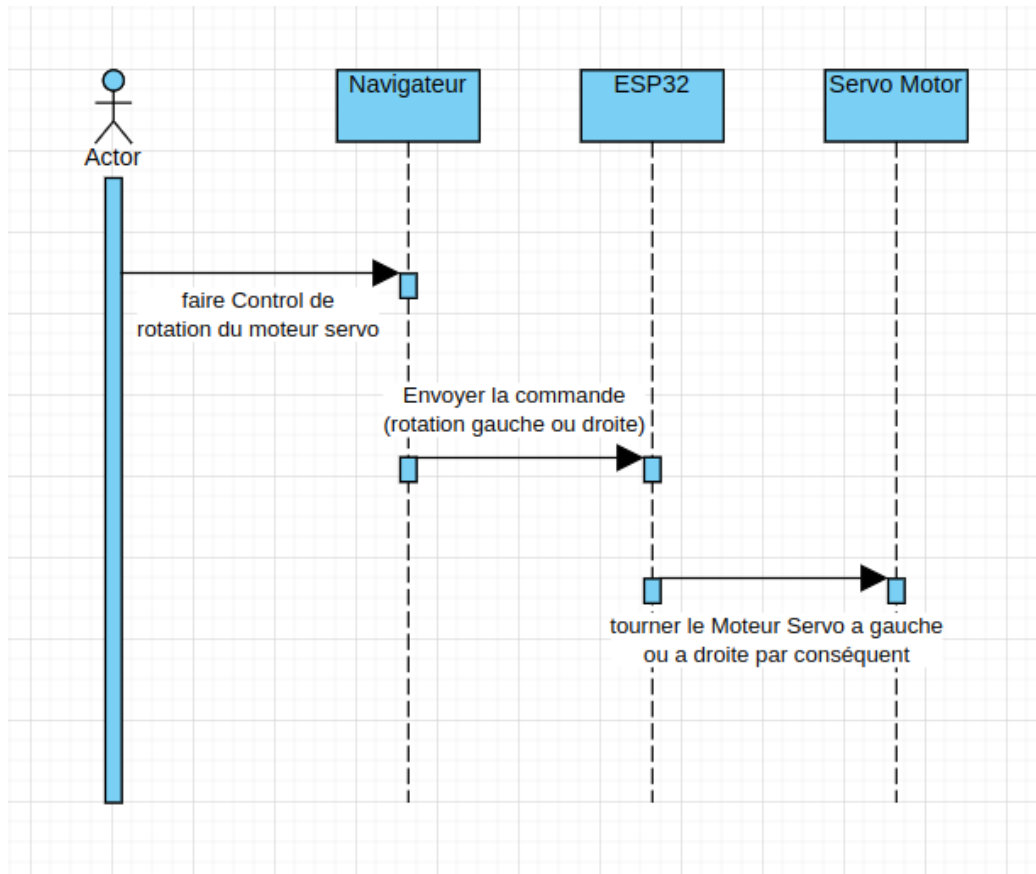


Figure 5: Diagramme de sequence de controle de rotation de la camera

8 Planification du Projet

8.1 Phase 1 : Conception et Planification

- **Identification des Besoins** : Analyse des exigences fonctionnelles et techniques.
- **Spécification des Exigences Techniques** : Rédaction des spécifications techniques détaillées.
- **Élaboration des Diagrammes** : Création des schémas et diagrammes nécessaires pour la compréhension globale du projet.

8.2 Phase 2 : Développement du Système Embarqué

- Installation et configuration des composants matériels.
- Câblage avec des fils de connexion (jumper wires).

8.3 Phase 3 : Développement de l'Interface Web

- Développement de l'interface web pour permettre le contrôle de la voiture robot.

8.4 Phase 4 : Integration de l'Algorithme YOLO8

- Développement de l'algorithme YOLO8 et déploiement sur un serveur GCP.
- Mise en place de la communication entre le serveur GCP et l'ESP32.

8.5 Phase 5 : Ajout de Fonctionnalités secondaires

- Ajout de la fonctionnalité de la rotation de la caméra pour une vue de 180° (avec le servo-moteur).
- Intégration du détecteur d'obstacles à ultrasons (avec HC-SR04).

8.6 Phase 6 : Tests

Bien que nous ayons effectué des tests en parallèle avec les travaux et la construction de la voiture, nous avons dû ajouter des tests finaux pour nous assurer que tout fonctionne parfaitement.

- **Tests Unitaires et d'Intégration** : Vérification du bon fonctionnement de chaque composant.
- **Tests d'Utilisateurs et Feedback** : Collecte de feedback des utilisateurs et ajustements finaux.

9 Conclusion

Ce projet a été une expérience enrichissante et un excellent travail à réaliser. Il a été particulièrement gratifiant d'utiliser différentes technologies qui nous étaient nouvelles, comme celles de l'intelligence artificielle, notamment YOLO. Ce projet a également été une preuve concrète de ce que nous avons étudié dans nos cours sur les systèmes embarqués avec Mr. Rahmoun. Nous avons constaté que la gestion des ressources est primordiale dans de tels systèmes.

En effet, nous avons rapidement remarqué une pénurie de ressources énergétiques : un simple ensemble de piles ne suffisait pas à alimenter tout le système. Nous avons trouvé une solution à ce problème en utilisant une batterie externe (power bank). De plus, nous avons été confrontés aux limites de la bande passante et des capacités de calcul du processeur de la carte ESP32 lors de l'ajout de la détection d'objets. Ces défis nous ont permis de mieux comprendre et d'apprécier l'importance de l'optimisation des ressources dans les systèmes embarqués.

En résumé, ce projet nous a non seulement permis de mettre en pratique nos connaissances théoriques, mais il nous a également offert une expérience précieuse dans la résolution de problèmes complexes en utilisant des technologies avancées.