

République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de 20 aout 1955 Skikda

Faculté des sciences

Département informatique



جامعة 20 أوت 1955 سكيكدة

كلية العلوم

قسم الاعلام الآلي

MEMOIRE

En vue de l'obtention du diplôme :

Master 2

Option : Génie Logiciel Avancée et Applications - GLAA

Réalisation d'un Système de Détection d'Obstructions dans les Canalisations d'Eaux Usées à l'Aide d'une Voiture- Robot Équipée d'un Microcontrôleur ESP32-Cam

Présenté par :

Rassim BENGUEDOUAD

Soutenu publiquement le : 02/07/2024

Devant la Commission du jury composée de :

JURY

Encadreur :	SELLAMI Samir	- MCA – ENSET-Skikda
Co-Encadreur :	BENOUDINA Lazhar	- MCB – Université 20 Août 1955 Skikda
Présidente :	LALAOUA Chahra	- MAA – Université 20 Août 1955 Skikda
Examinatrice :	MALLEM Zina	- MAA – Université 20 Août 1955 Skikda

Remerciements

À tous ceux, qui ont contribué de près ou de loin pour leurs informations et conseils judicieux, de m'avoir accordé leurs précieux temps à l'élaboration de ce mémoire.

Je tiens à exprimer ma profonde gratitude à mon encadreur, Monsieur Sellami Samir, MCA à l'École Normale Supérieure d'Enseignement Technologique (ENSET) de Skikda, pour ses conseils précieux, son soutien indéfectible et son orientation éclairée tout au long de ce travail.

Je tiens également à remercier chaleureusement le président de jury, Monsieur Lazhar Benoudina (MCA – Université 20 Août 1955 Skikda), pour l'honneur qu'il m'a fait en acceptant de présider le jury.

Tous mes remerciements accompagnés de ma gratitude vont aux membres du jury en qualité d'examineurs, Mme LALAOUA Chahra (MCA – Université 20 Août 1955 Skikda) et Mme MALLEM Zina (MCA – Université 20 Août 1955 Skikda), pour la célérité avec laquelle ils ont examiné, corrigé et critiqué ce travail.

Je tiens remercier les membres de l'équipe Wise Tech et exprimer ma gratitude envers Monsieur Mansour Lamine pour le soutien et l'aide dans l'aspect pratique de ce projet.

À tous ceux qui ont contribué de près ou de loin pour leurs informations et conseils judicieux, et de m'avoir accordé leur précieux temps pour l'élaboration de ce mémoire.

Dédicaces

Ce mémoire représente l'aboutissement d'un parcours rendu possible grâce au soutien indéfectible et à l'amour inconditionnel de nombreuses personnes.

Je dédie ce travail à toutes celles et ceux qui ont contribué de près ou de loin à la réalisation de ce projet.

À Dieu, Allah, pour m'avoir guidé et donné la force nécessaire pour surmonter chaque obstacle.

À ma mère, pilier de mon existence, dont l'amour et la patience sont des sources d'inspiration constantes.

À mon père, pour son soutien inconditionnel et ses encouragements infatigables.

À mes chers amis disparus, Wassim et Lokman, dont le souvenir reste gravé dans mon cœur.

À mes frères et sœurs, pour leur amour, leurs encouragements et leur aide précieuse tout au long de ce parcours.

À tous mes amis, pour leur gentillesse et leur soutien inestimable.

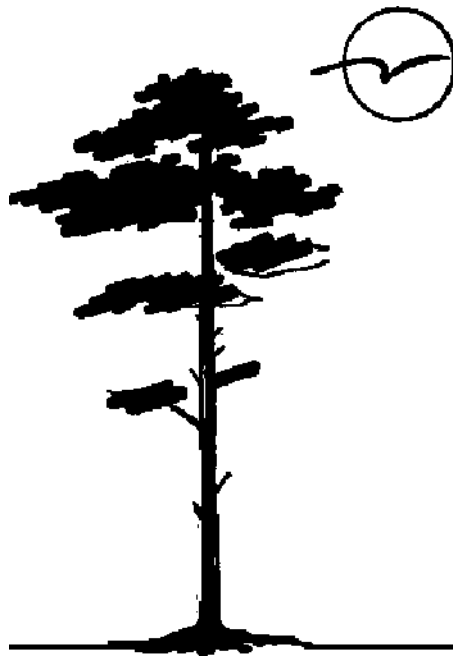
À toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

À VOUS.

Rassim

La connaissance est un arbre de vie qui grandit à la lumière des autres.

Le Paternel.



Résumé

La ville de Skikda, confrontée à des problèmes critiques d'assainissement liés à l'obstruction fréquente de ses canalisations d'eaux usées, a besoin de solutions innovantes pour prévenir les débordements, protéger l'environnement et assurer la santé publique. Cette étude propose le développement d'un système avancé de détection des obstructions, basé sur l'utilisation d'une voiture-robot équipée d'un microcontrôleur ESP32-Cam.

L'objectif principal de ce travail est de concevoir et de mettre en œuvre un système capable d'identifier les zones à risque de blocage avant qu'elles ne causent des débordements. La voiture-robot, équipée du microcontrôleur ESP32-Cam, sera capable de naviguer à l'intérieur des canalisations, collectant des données visuelles en temps réel pour faciliter la détection des signes d'obstructions.

Une application web de contrôle à distance sera également développée, permettant une surveillance et un contrôle efficaces du système depuis un appareil mobile ou un PC. En plus de la détection d'obstructions, le système proposé intégrera des capteurs de détection d'obstacles pour améliorer la sécurité de la navigation de la voiture-robot.

Ce projet vise à fournir une solution technologique pour résoudre les problèmes d'assainissement de Skikda, contribuant à réduire les débordements d'eaux usées et à améliorer la qualité de vie des citoyens. En optimisant l'efficacité des opérations de maintenance des réseaux d'assainissement, ce projet offrira également des économies substantielles aux autorités locales et à l'Office National d'Assainissement.

Cette étude présente une approche novatrice et pratique pour aider à relever les défis complexes de la gestion des réseaux d'assainissement dans un contexte urbain. Les résultats auront un impact significatif sur la gestion des infrastructures urbaines et la protection de l'environnement, offrant une solution durable et efficace pour les problèmes d'assainissement à Skikda et au-delà.

Mots-clés : Assainissement, Détection d'obstructions, Microcontrôleur ESP32-Cam, Voiture-robot, Application Web, Capture d'image, Vidéo en temps réel

Abstract

The city of Skikda, facing critical sanitation issues due to frequent blockages in its sewage pipes, requires innovative solutions to prevent overflows, protect the environment, and ensure public health. This study proposes the development of an advanced obstruction detection system, based on the use of a robotic car equipped with an ESP32-Cam microcontroller.

The primary objective of this research is to design and implement a system capable of identifying high-risk blockage areas before they cause overflows. The robotic car, equipped with the ESP32-Cam microcontroller, will be able to navigate inside the pipes, collecting real-time visual data to help detect signs of blockages.

A remote-control web application will also be developed, enabling effective monitoring and control of the system from a mobile device or PC. In addition to obstruction detection, the proposed system will integrate obstacle detection sensors to enhance the robotic car's navigation safety.

This project aims to provide a technological solution to Skikda's sanitation problems, contributing to reduced sewage overflows and improved quality of life for citizens. By optimizing the efficiency of sewer network maintenance operations, this project will also offer substantial savings to local authorities and the National Sanitation Office.

This study presents an innovative and practical approach to help addressing the complex challenges of sewer network management in an urban context. The results will significantly impact urban infrastructure management and environmental protection, providing a sustainable and effective solution to sanitation problems in Skikda and beyond.

Keywords: Sanitation, Obstruction Detection, ESP32-Cam Microcontroller, Robotic Car, Web Application, Image capturing, Video Streaming

ملخص

تواجه مدينة سكيكدة مشاكل حادة في الصرف الصحي بسبب الانسدادات المتكررة في أنابيب الصرف الصحي، مما يتطلب حلولاً مبتكرة لمنع الفيضانات وحماية البيئة وضمان الصحة العامة. تقترح هذه الدراسة تطوير نظام متقدم للكشف عن الانسدادات يعتمد على استخدام سيارة روبوتية مجهزة بوحدة تحكم ESP32-Cam.

الهدف الرئيسي من هذا البحث هو تصميم وتنفيذ نظام قادر على تحديد المناطق المعرضة لخطر الانسداد قبل أن تسبب الفيضانات. ستكون السيارة الروبوتية المجهزة بوحدة التحكم ESP32-Cam قادرة على التنقل داخل الأنابيب، وجمع البيانات المرئية في الوقت الحقيقي لتسهيل الكشف عن علامات الانسدادات.

سيتم أيضاً تطوير تطبيق ويب للتحكم عن بعد لتمكين المراقبة والتحكم الفعال في النظام بواسطة هاتف ذكي أو حاسوب شخصي. بالإضافة إلى الكشف عن الانسدادات، سيتكامل النظام المقترح مع مستشعرات كشف العوائق لتعزيز أمان التنقل للسيارة الروبوتية.

يهدف هذا المشروع إلى تقديم حل تكنولوجي لمشاكل الصرف الصحي في سكيكدة، مما يساهم في تقليل فيضانات مياه الصرف الصحي وتحسين جودة حياة المواطنين. من خلال تحسين كفاءة عمليات صيانة شبكات الصرف الصحي، سيساهم هذا المشروع كذلك في توفير تكاليف كبيرة للسلطات المحلية والديوان الوطني للصرف الصحي.

تقدم هذه الدراسة نهجاً مبتكراً وعملياً لتسهيل التعامل مع التحديات المعقدة لإدارة شبكات الصرف الصحي في سياق حضري. ستكون النتائج ذات تأثير كبير على إدارة البنية التحتية الحضرية وحماية البيئة، مما يوفر حلاً مستداماً وفعالاً لمشاكل الصرف الصحي في سكيكدة وما وراءها.

2.1.1 ملخص البحث: الصرف الصحي، الكشف عن الانسدادات، دارة التحكم ESP32-Cam، سيارة روبوتية، تطبيق ويب، التقاط الصور، فيديو في الوقت الحقيقي

Table des matières

<i>Remerciements</i>	<i>i</i>
<i>Dédicaces</i>	<i>ii</i>
<i>Table des matières</i>	<i>vii</i>
<i>Liste des figures</i>	<i>ix</i>
<i>Liste des tableaux</i>	<i>x</i>
Introduction générale	1
1. Contexte et justification du projet	1
2. Objectifs de l'étude	1
3. Organisation du mémoire	2
CHAPITRE 1. État de l'Art	4
1.1 Introduction	4
1.2 Détection des obstructions dans les canalisations	4
1.2.1 Techniques traditionnelles	4
1.2.2 Techniques de pointe	5
1.3 Comparaison des techniques de détection des obstructions dans les canalisations	6
1.4 Microcontrôleur ESP32-CAM	7
1.5 Applications des microcontrôleurs et des caméras ESP32-Cam dans les projets de robotique	9
1.5.1 Robotique	9
1.5.2 Véhicules autonomes	9
1.5.3 Drones	9
1.5.4 Sécurité domestique	10
1.5.5 Automatisation industrielle	10
1.6 Avantages de l'ESP32-CAM dans un système de détection d'obstructions	11
1.6.1 Inspection efficace des canalisations	11
1.6.2 Détection précoce des obstructions	11
1.6.3 Réduction des coûts de maintenance	11
1.6.4 Amélioration de la sécurité publique	12
1.7 Conclusion	12
CHAPITRE 2. Conception du Système	14
2.1 Introduction	14
2.2 Architecture générale d'un système de détection d'obstacles	14
2.2.1 Châssis de la voiture-robot	14
2.2.2 Système de propulsion	16
2.2.3 Système d'éclairage	16
2.2.4 Système de communication	16
2.2.5 Système de détection d'obstructions	16
2.2.6 Schéma du système	17
2.2.7 Fonctionnement du système	17
2.3 Conception de l'application Web de contrôle	18
2.3.1 Diagramme de Cas d'Utilisation (Use Case Diagram)	18

2.3.2	Diagramme d'Activités (Activity Diagram)	20
2.3.3	Diagramme d'États (State Diagram)	21
2.3.4	Diagramme de Séquence (Sequence Diagram)	23
2.3.5	Diagramme de Classe (Class Diagram)	25
2.3.6	Diagramme Modèle de Données (Data Model)	25
2.3.7	Diagramme de Composants (Component Diagram)	28
2.3.8	Diagramme de Déploiement (Deployment Diagram)	28
2.4	Conclusion	29
CHAPITRE 3. Réalisation et Test		31
3.1	Introduction	31
3.2	Mise en œuvre de la Voiture-Robot	31
3.2.1	Dimensions et Forme	32
3.2.2	Matériaux Utilisés	32
3.2.3	Installation des Composants	33
3.2.4	Caméra ESP32-CAM	36
3.3	Programmation du Microcontrôleur ESP32-CAM	37
3.3.1	Contrôle de la Voiture-Robot	37
3.3.2	Configuration de la Caméra ESP32-CAM	38
3.3.3	Codage du Microcontrôleur	38
3.3.4	Configuration de l'Arduino IDE	40
3.4	Développement de l'Application Web	50
3.4.1	Frameworks Web Utilisés	50
3.4.2	Code source de l'Application Web	51
3.5	Tests et Validation	55
3.5.1	Objectifs des Tests	55
3.5.2	Matériel Utilisé	55
3.5.3	Déroulement des Tests	56
3.5.4	Tests des Composants Logiciels	56
3.5.5	Tests des Composants Matériels	57
3.5.6	Tests en Environnements Simulés	59
3.5.7	Tests en Conditions Réelles	62
3.5.8	Analyse des Résultats	63
3.6	Conclusion	63
Conclusion et Perspectives		65
1.	Contributions	65
2.	Limites et Difficultés Rencontrées	65
3.	Perspectives et Travaux Futurs	66
Liste des abréviations		68
Références		<i>Erreur ! Signet non défini.</i>

Liste des figures

Figure 1.1 - Microcontrôleur ESP32-CAM	7
Figure 2.1 - Architecture générale du système	15
Figure 2.2 - Schéma conceptuel du Châssis avec 4 roues et 4 DC moteurs	15
Figure 2.3 - Schéma conceptuel montre la disposition des différents composants.....	17
Figure 2.4 - Diagramme de Cas d'Utilisation (Use Cases)	19
Figure 2.5 - Diagramme d'Activités (Activity Diagram)	21
Figure 2.6 - Diagramme de d'états (State Diagram).....	22
Figure 2.7 - Diagramme de Séquence (Sequence Diagram).....	24
Figure 2.8 - Diagramme de Classe (Class Diagram).....	26
Figure 2.9 - Diagramme Modèle de Données (Data Model)	27
Figure 2.10 - Diagramme de Composants (Component Diagram)	28
Figure 2.11 - Diagramme de Déploiement (Deployment Diagram).....	29
Figure 3.1 - Différentes vues de la voiture-robot	31
Figure 3.2 - Schéma électronique de la voiture et du ESP-32-Cam	33
Figure 3.3 - Support microcontrôleur ESP32-CAM	34
Figure 3.4 - Jeu des Batteries et leur Support.....	34
Figure 3.5 - Contrôleur Moteur	35
Figure 3.6 - Fixation des Moteurs.....	35
Figure 3.7 - LED ESP32-Cam	36
Figure 3.8 - Cache de protection	36
Figure 3.9 - Camera EPS32-Cam	37
Figure 3.10 - Programmation du microcontrôleur.....	38
Figure 3.11 - Arduino IDE – Menu : Fichier -> Préférences.....	41
Figure 3.12 - Arduino IDE - Gestionnaire de cartes supplémentaires	41
Figure 3.13 - Arduino IDE – Gestionnaire de cartes capture 1	42
Figure 3.14 - Arduino IDE - Gestionnaires de cartes capture 2	42
Figure 3.15 - Arduino IDE - Installation de la carte EPS32	43
Figure 3.16 - Arduino IDE - L'installation Espressif Systems EPS32 la version 2.0.9.....	43
Figure 3.17 - Arduino IDE - Selection AI Thinker ESP32-CAM.....	44
Figure 3.18 - Liste des ports USB avant le branchement de l'ESP32-CAM.....	45
Figure 3.19 - Liste des ports USB après le branchement de l'ESP32-CAM	45
Figure 3.20 - Raccourci de l'Arduino IDE pour lancer le téléversement du programme	46
Figure 3.21 - Arduino IDE – Lancement de téléversement du programme compilé.....	46
Figure 3.22 - Téléversement du programme dans la carte de développement ESP32-CAM.....	47
Figure 3.23 - Configuration initiale du projet dans l'IDE Arduino.....	48
Figure 3.24 - Définition des constantes et configuration WiFi	48
Figure 3.25 - Fonction rotateMotor pour le contrôle des moteurs	49
Figure 3.26 - Contrôle des directions du robot.....	49
Figure 3.27 - Structure HTML de l'application Web de Contrôle.....	51
Figure 3.28 - Initialisation et gestion des WebSockets	52
Figure 3.29 - Capturer et faire pivoter des photos.....	53
Figure 3.30 - Style CSS pour l'application Web	54
Figure 3.31 – Interface Utilisateur de l'application Web	55
Figure 3.32 - Création de l'environnement de test simulé.....	59
Figure 3.33 - Voiture-robot en situation dans l'environnement simulé	60
Figure 3.37 - Interface de contrôle de l'application web	61
Figure 3.35 - Test de navigation et détection d'obstacles.....	62

Liste des tableaux

Tableau 1.1 - Comparaison des techniques de détection des obstructions.....	6
Tableau 3.1 - Récapitulatif des cas de tests logiciels et matériels	58

Introduction générale

INTRODUCTION GENERALE

1. Contexte et justification du projet

La ville de Skikda, récemment confrontée à des problèmes majeurs d'assainissement [1], fait face à une sérieuse crise liée à l'obstruction fréquente des canalisations d'eaux usées. Ces obstructions entraînent des débordements désastreux, menaçant l'environnement local et la santé publique. Nous estimons que les autorités locales et l'Office National d'Assainissement¹ ont un besoin urgent d'améliorer les méthodes de détection et de prévention des obstructions dans le réseau de canalisations.

Dans ce contexte, ce projet revêt une importance cruciale. En proposant un système innovant de détection d'obstructions basé sur une voiture-robot équipée d'un microcontrôleur ESP32-Cam [2], il vise à répondre efficacement aux défis actuels de gestion des réseaux d'assainissement. En fournissant une solution technologique avancée, ce projet a le potentiel de réduire considérablement le temps nécessaire à la détection des obstructions, ce qui permettra des interventions rapides et ciblées pour prévenir les débordements.

La ville de Skikda et ses habitants bénéficieront directement des résultats de ce travail, en réduisant les nuisances causées par les débordements d'eaux usées et en améliorant la qualité de vie des citoyens. De plus, en optimisant l'efficacité des opérations de maintenance des réseaux d'assainissement, ce projet contribuera à réduire les coûts associés aux interventions d'urgence, offrant ainsi des économies significatives aux autorités locales et à l'Office National d'Assainissement.

Notre projet s'inscrit dans un contexte réel et urgent, visant à résoudre les problèmes concrets d'assainissement auxquels est confrontée la ville de Skikda. En proposant une solution innovante et pratique, il offre des perspectives prometteuses pour améliorer la gestion des réseaux d'assainissement et garantir un environnement sain et durable pour les générations futures.

2. Objectifs de l'étude

Nous décrivons dans les pointes suivantes les contributions de notre étude :

1. **Développer un système de détection des obstructions** : L'objectif principal de cette étude est de concevoir et de mettre en œuvre un système innovant capable de détecter les obstructions dans les canalisations d'eaux usées. Ce système permettra d'identifier les zones à risque de blocage avant qu'elles ne provoquent des débordements, facilitant ainsi les mesures préventives et réduisant les conséquences néfastes sur l'environnement et la santé publique.
2. **Concevoir une voiture-robot équipée d'un microcontrôleur ESP32-Cam** : Ce projet vise à développer un prototype robotique capable de naviguer à l'intérieur des canalisations d'eaux usées. La voiture-robot sera équipée d'un microcontrôleur ESP32-Cam, offrant ainsi une solution compacte et polyvalente pour la collecte de données visuelles et faciliter la détection d'obstructions.

¹ <https://ona-dz.org>

3. **Intégrer des capteurs de détection d'obstacles et une caméra haute résolution :** Pour permettre à la voiture-robot de détecter efficacement les obstructions, des capteurs de détection d'obstacles seront intégrés à sa conception. De plus, une caméra haute résolution intégrée au microcontrôleur ESP32-Cam sera utilisée pour collecter des données visuelles en temps réel, permettant une analyse précise des conditions des canalisations.
4. **Développer une application web de contrôle à distance :** En complément de la voiture-robot, une application web sera développée pour permettre le contrôle à distance du système. Cette application, accessible depuis un appareil mobile ou un ordinateur personnel, offrira une interface conviviale pour surveiller et contrôler les opérations de la voiture-robot, facilitant ainsi la gestion et la supervision du système.
5. **Évaluer les performances du système :** Une fois le système développé, des tests expérimentaux seront réalisés pour évaluer ses performances en conditions simulées. Les résultats obtenus seront analysés pour vérifier la précision de la détection d'obstructions et la fiabilité du système dans des environnements réels.

3. Organisation du mémoire

Ce mémoire est organisé selon une structure, comprenant trois chapitres qui guident le lecteur à travers toutes les étapes de l'étude et du développement du système de détection d'obstructions dans les canalisations d'eaux usées.

Après une *introduction générale* décrivant le contexte et la justification du projet, ainsi que les objectifs de la recherche, le *Chapitre 1- État de l'art* offre une revue de la littérature approfondie sur la détection des obstructions dans les canalisations d'eaux usées, ainsi que sur les technologies existantes pour la surveillance et l'inspection des réseaux d'assainissement, avec un accent particulier sur les applications des microcontrôleurs et des caméras ESP32-Cam dans les projets de robotique.

Le *Chapitre 2 – Conception du système* est consacré à la conception de notre système de détection d'obstructions et de l'application web de contrôle. Il présente l'architecture générale du système, la conception et la fabrication du prototype de la voiture-robot, ainsi que l'intégration des capteurs de détection d'obstacles et de la caméra ESP32-Cam. En utilisant divers diagrammes (cas d'utilisation, états, séquence, classes, données, composants et déploiement), nous avons visualisé les interactions entre les composants, les flux de données et les processus de traitement.

La réalisation de cette étude est détaillée dans le *Chapitre 3 – Réalisation et Test*. Ce chapitre décrit les matériaux et outils utilisés, ainsi que l'environnement de développement de l'application web de contrôle. Les résultats des tests effectués dans un environnement simulé, l'analyse des données collectées par la voiture-robot et l'évaluation des performances de détection d'obstructions y sont présentés et discutés. En outre, une interprétation des résultats par rapport aux objectifs du projet est fournie, accompagnée d'une discussion sur les limitations du système et les perspectives d'amélioration.

Enfin, la *conclusion générale* du mémoire récapitule les principales conclusions de l'étude, met en avant les contributions à la recherche et aux applications pratiques, et propose des recommandations pour de futures recherches dans ce domaine. La bibliographie et les annexes sont également incluses pour fournir des références complètes et des détails supplémentaires si nécessaire.

État de l'art

CHAPITRE 1. ÉTAT DE L'ART

1.1 Introduction

Dans ce chapitre, nous présentons une revue de la littérature sur les techniques de détection des obstructions dans les canalisations d'eaux usées, en mettant l'accent sur les méthodes traditionnelles et les technologies de pointe. Nous incluons également une comparaison des techniques, ainsi qu'une discussion sur leurs avantages et inconvénients. Nous explorerons également le microcontrôleur ESP32-Cam et ses applications dans les projets de robotique, en soulignant ces avantages et leur pertinence dans le domaine de l'assainissement.

1.2 Détection des obstructions dans les canalisations

1.2.1 Techniques traditionnelles

Cette section examine les méthodes traditionnelles utilisées pour détecter les obstructions dans les canalisations d'eaux usées. Les techniques abordées incluent les inspections visuelles, l'utilisation des ultrasons, les radiographies, et les méthodes de pression. Chaque technique sera détaillée avec ses avantages et inconvénients, ainsi que son application dans le secteur de l'assainissement.

- **Inspections visuelles**

Les inspections visuelles sont parmi les méthodes les plus anciennes et les plus couramment utilisées pour détecter les obstructions dans les canalisations. Elles consistent à insérer une caméra d'inspection à l'intérieur des tuyaux, souvent accompagnée de dispositifs d'éclairage, pour permettre une visualisation directe de l'état intérieur des canalisations. Cette méthode est particulièrement utile pour diagnostiquer des problèmes visibles tels que les fuites, les déformations, et les matériaux étrangers qui pourraient bloquer l'écoulement des eaux [3].

Les avantages des inspections visuelles incluent la simplicité et l'efficacité pour repérer des obstructions visibles. De plus, cette technique permet une évaluation en temps réel et peut être utilisée pour planifier des interventions immédiates. Cependant, elle présente aussi des inconvénients. Les inspections visuelles sont limitées aux problèmes visibles à l'œil nu, et leur efficacité peut être réduite dans des environnements complexes ou obscurs. De plus, le coût en main-d'œuvre et en équipement peut être élevé, surtout pour des inspections répétées ou sur de longues distances [4].

- **Ultrasons**

La méthode d'inspection par ultrasons utilise des ondes sonores à haute fréquence pour détecter les obstructions et les défauts dans les canalisations. Un capteur à ultrasons, placé à l'extrémité d'un câble flexible, est inséré dans le réseau de tuyaux. Les ondes sonores émises par le capteur se propagent à travers les parois des tuyaux et sont réfléchies par les obstructions ou les défauts internes. Les ondes réfléchies sont ensuite analysées pour déterminer la nature et la localisation des anomalies [5].

Les ultrasons offrent plusieurs avantages, notamment la capacité de détecter des obstructions et des défauts internes sans nécessiter de démontage des tuyaux. Cette technique est non invasive et peut fournir des informations détaillées sur l'état interne des canalisations.

Cependant, elle nécessite un équipement spécialisé et peut être affectée par les conditions du tuyau, telles que les dépôts ou la rugosité des parois internes, ce qui peut altérer la précision des mesures [6].

- **Radiographies**

Les radiographies utilisent des rayons X pour visualiser l'intérieur des canalisations et détecter les obstructions et les défauts structurels. Une source de rayons X est placée d'un côté de la canalisation, tandis qu'un détecteur est positionné de l'autre côté. Les rayons X traversent la canalisation et sont capturés par le détecteur, créant une image radiographique qui révèle les obstructions et les défauts internes [7].

Cette méthode offre une vue claire et détaillée des obstructions internes, ce qui est particulièrement utile pour les canalisations enterrées ou difficilement accessibles. Les radiographies peuvent détecter des problèmes structurels tels que les fissures, les affaissements, ou les matériaux étrangers bloquant l'écoulement. Toutefois, l'utilisation de rayons X nécessite des précautions strictes en raison des risques associés aux radiations ionisantes. De plus, le coût et la complexité de l'équipement peuvent limiter l'application de cette méthode à des situations spécifiques où une visualisation détaillée est cruciale [8].

- **Méthodes de pression**

Les méthodes de pression consistent à mesurer les variations de pression à l'intérieur des canalisations pour détecter les fuites et les obstructions. Des capteurs de pression sont placés à différentes positions le long du réseau de tuyaux. En surveillant les variations de pression, les techniciens peuvent localiser les zones où des obstructions ou des fuites affectent le flux d'eau. Cette technique est efficace pour repérer rapidement les zones problématiques et peut être utilisée en continu pour une surveillance en temps réel du réseau de canalisations. Les variations de pression peuvent indiquer non seulement la présence d'obstructions mais aussi leur gravité, permettant des interventions ciblées. Cependant, cette méthode peut être influencée par d'autres facteurs de variation de pression, tels que les changements de débit ou les conditions de fonctionnement du réseau, ce qui peut compliquer l'interprétation des données [9].

1.2.2 Techniques de pointe

Cette section explore les technologies modernes utilisées pour améliorer la détection des obstructions dans les canalisations. Nous nous concentrerons sur les capteurs à infrarouges, les capteurs de pression différentielle, et les capteurs de débit [10].

- **Capteurs à infrarouges**

Les capteurs à infrarouges sont utilisés pour détecter les variations de température le long des canalisations, ce qui permet d'identifier les zones anormales pouvant indiquer une obstruction. Ces capteurs mesurent la température des parois des tuyaux et repèrent les écarts thermiques causés par des obstructions ou des accumulations de matériaux. Cette méthode est particulièrement utile pour détecter des défauts thermiques invisibles à l'œil nu et peut être utilisée en complément d'autres techniques pour une analyse plus complète [11].

- **Capteurs de pression différentielle**

Les capteurs de pression différentielle mesurent les différences de pression entre deux points dans une canalisation. En analysant ces différences, les techniciens peuvent détecter des obstructions et des fuites. Cette méthode est efficace pour fournir une analyse en temps réel et localiser précisément les problèmes. Elle est particulièrement utile dans des systèmes de

canalisations complexes où les variations de pression peuvent révéler des anomalies non détectables par des méthodes visuelles [12].

- **Capteurs de débit**

Les capteurs de débit mesurent la vitesse de l'écoulement de l'eau dans les canalisations. Une diminution du débit peut indiquer la présence d'une obstruction. Ces capteurs permettent d'identifier les zones où le débit est réduit, offrant des indications précises sur l'emplacement des obstructions potentielles. Cette technologie est souvent utilisée en combinaison avec d'autres techniques pour une surveillance complète et continue du réseau de canalisations [13]. Malgré les avantages de ces techniques basées sur des capteurs modernes, elles restent affectées et sensibles aux variations de l'environnement. De plus, leur coût d'installation et de maintenance reste très élevé, limitant leur adoption généralisée.

1.3 Comparaison des techniques de détection des obstructions dans les canalisations

Le tableau 1.1 fournit une synthèse comparative des différentes techniques utilisées pour la détection des obstructions dans les canalisations, en mettant en évidence leurs avantages et inconvénients respectifs.

Les techniques de détection des obstructions dans les canalisations présentent des avantages notables, mais aussi des limitations significatives. Les inspections visuelles, bien que rapides et précises pour les défauts visibles, nécessitent une main-d'œuvre spécialisée et un équipement coûteux, et sont inefficaces pour les anomalies internes non visibles.

Tableau 1.1 - Comparaison des techniques de détection des obstructions

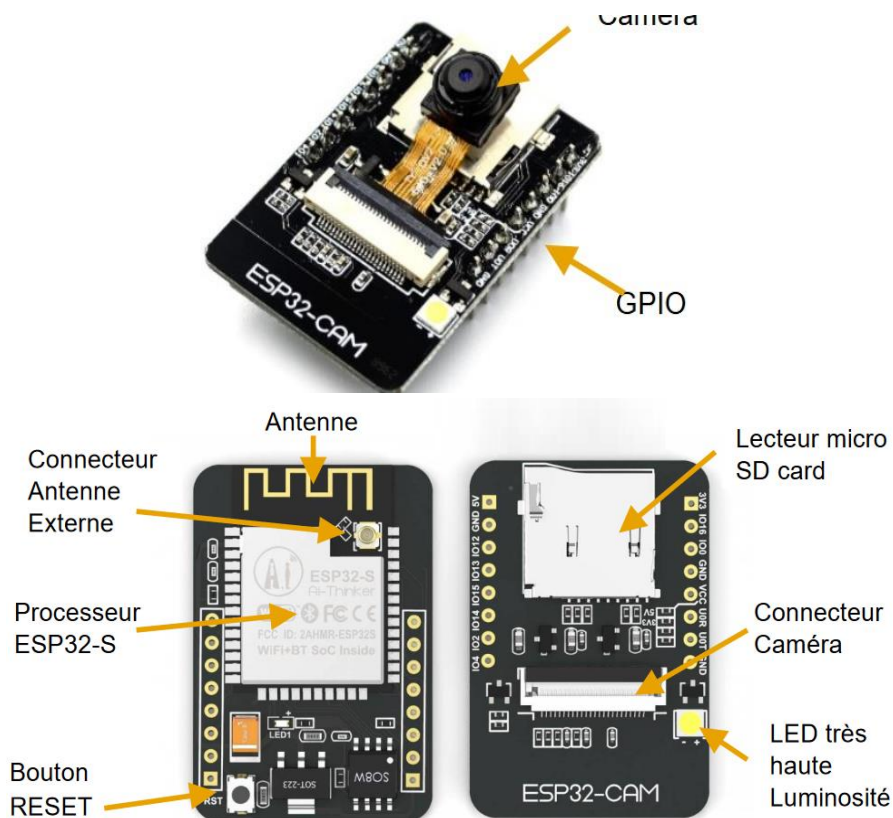
Technique	Avantages	Inconvénients
Techniques traditionnelles		
Inspections visuelles	- Détection rapide et précise des défauts visibles	- Limité aux problèmes visibles - Coût élevé en main-d'œuvre et équipements
Ultrasons	- Détection précise des obstructions et défauts internes - Non invasif	- Nécessite un équipement spécialisé - Affecté par les conditions du tuyau
Radiographies	- Vue détaillée des obstructions internes - Détection des problèmes structurels	- Risques liés aux radiations - Coût et complexité des équipements
Techniques de pointe		
Méthodes de pression	- Surveillance en temps réel - Identification précise des anomalies	- Influencée par les variations de pression externes - Calibration nécessaire
Capteurs à infrarouges	- Mesures non intrusives - Détection des anomalies thermiques invisibles	- Sensibles aux variations environnementales - Coût élevé des capteurs
Capteurs de pression différentielle	- Mesures précises et en temps réel - Détection rapide des fuites et obstructions	- Affecté par les changements de température et de pression - Calibration complexe - Coût élevé des capteurs

Les ultrasons, malgré leur capacité à détecter des défauts internes de manière non invasive, dépendent fortement des conditions des tuyaux et nécessitent des équipements coûteux. Les radiographies fournissent des images détaillées des obstructions internes, mais présentent des risques liés aux radiations et des coûts élevés d'équipement. Les méthodes de pression offrent une surveillance en temps réel, mais sont sensibles aux variations externes de pression et nécessitent une calibration précise. Les capteurs à infrarouges et les capteurs de pression différentielle sont affectés par les changements de température et de pression, nécessite une calibration complexe et leur coût reste très élevé.

Face à ces limitations, il est essentiel de considérer des solutions plus simples et moins coûteuses. L'utilisation d'un microcontrôleur ESP32-CAM pour réaliser une voiture-robot représente une alternative innovante. Cette approche utilise un microcontrôleur économique et facilement programmable, équipé d'une caméra haut résolution, pour faciliter la détection des obstructions dans les canalisations d'eaux usées. La voiture-robot peut naviguer de manière contrôlée à travers les canalisations, capturer des images en temps réel et transmettre les données à une station de contrôle pour analyse.

1.4 Microcontrôleur ESP32-CAM

Le microcontrôleur ESP32-CAM (voir Figure 1.1) est une solution intégrée et économique qui combine un microcontrôleur ESP32 avec une caméra OV2640. Cette configuration en fait un choix idéal pour les applications nécessitant une capture d'image et une transmission de données sans fil. L'ESP32-CAM est particulièrement apprécié dans les projets de robotique et d'IoT (Internet des objets) en raison de sa polyvalence, de sa faible consommation d'énergie et de ses capacités de communication avancées. Ce module est équipé d'un processeur double cœur, de connectivités Wi-Fi et Bluetooth, et peut être programmé facilement via l'environnement de développement Arduino ou Platform IO [14].



Le module ESP32-CAM intègre une caméra OV2640 qui permet de capturer des images de haute qualité avec une résolution allant jusqu'à 1600x1200 pixels. La caméra offre une excellente performance en termes de sensibilité à la lumière, ce qui est crucial pour les applications de détection dans des environnements faiblement éclairés, tels que les canalisations d'eaux usées. De plus, l'ESP32-CAM dispose d'une interface microSD pour le stockage des images et des vidéos, permettant une analyse ultérieure des données capturées. Cette capacité de stockage local est essentielle pour les applications où une connexion stable n'est pas toujours disponible [15].

L'ESP32-CAM est à la capacité de transmettre des images et des vidéos en temps réel via Wi-Fi. Cette fonctionnalité de streaming vidéo est extrêmement utile pour les applications de surveillance à distance, permettant aux opérateurs de visualiser les conditions internes des canalisations en temps réel. De plus, grâce à ses capacités de traitement embarqué, l'ESP32-CAM peut être utilisé pour exécuter des algorithmes par ordinateur directement sur le module, réduisant ainsi la nécessité de transférer des volumes importants de données vers un serveur externe pour traitement. Cette combinaison de capture d'image, de traitement et de transmission de données fait de l'ESP32-CAM une solution puissante et versatile pour les systèmes de détection d'obstructions et d'autres applications robotiques [16].

Caractéristiques [17] :

- Module ESP32-S intégré, prend en charge WiFi + Bluetooth,
- Appareil photo OV2640 avec flash Emplacement pour carte TF intégré,
- Prend en charge jusqu'à la carte TF 4G pour le stockage de données,
- Prend en charge la surveillance vidéo WiFi et le téléchargement d'images WiFi,
- Prend en charge plusieurs modes de veille, courant de sommeil profond aussi faible que 6 mA, L'interface de contrôle est accessible via un connecteur à broches, facile à intégrer et à intégrer dans les produits utilisateur.

Utilisation [18] :

La combinaison ESP32-CAM est utilisée dans diverses applications IoT, notamment :

1. **Téléchargement d'images d'appareils domestiques intelligents** : Utilisé pour capturer et envoyer des images de l'intérieur des appareils domestiques intelligents pour la surveillance et le contrôle.
2. **Surveillance sans fil** : Permet de surveiller à distance divers environnements, comme les maisons ou les bureaux, grâce à la capacité de streaming vidéo en temps réel.
3. **Agriculture intelligente** : Utilisé pour surveiller les cultures et les conditions environnementales, aidant ainsi les agriculteurs à prendre des décisions éclairées pour la gestion des ressources agricoles.
4. **Identification sans fil QR** : Capable de lire et de traiter les codes QR (Quick Response) pour diverses applications, y compris l'identification et le suivi des produits.
5. **Reconnaissance faciale** : Intégré dans les systèmes de sécurité pour la reconnaissance faciale, permettant un accès sécurisé et une surveillance améliorée.

1.5 Applications des microcontrôleurs et des caméras ESP32-Cam dans les projets de robotique

Les microcontrôleurs et les caméras ESP32-Cam offrent de nombreuses possibilités d'application dans les projets de robotique. Dans ce qui suit, quelques exemples d'applications courantes et des références pour approfondir la compréhension de ces applications [19].

1.5.1 Robotique

Dans le domaine de la robotique, les systèmes de détection d'obstacles basés sur l'ESP32-CAM jouent un rôle crucial en permettant aux robots de naviguer de manière autonome et sécurisée dans leur environnement [20]. Grâce à la caméra haute résolution et aux capacités de traitement embarquées de l'ESP32, les robots peuvent capturer et analyser des images en temps réel pour identifier et localiser les obstacles présents sur leur chemin. Ces informations sont ensuite utilisées pour ajuster la trajectoire du robot, évitant ainsi les collisions et permettant une navigation fluide et efficace. Les algorithmes de traitement d'image permettent de reconnaître différents types d'obstacles, tels que les murs, les meubles, ou même les personnes, et de réagir de manière appropriée. Cette capacité de détection et d'évitement d'obstacles est essentielle pour les robots de service, les robots industriels, et les robots de recherche, qui doivent opérer dans des environnements complexes et dynamiques [21].

1.5.2 Véhicules autonomes

Les systèmes de détection d'obstacles basés sur l'ESP32-CAM jouent un rôle vital dans le développement et l'opération des véhicules contrôlés et autonomes. Ces véhicules, qui incluent des voitures, des camions et des bus, dépendent de capteurs sophistiqués pour naviguer en toute sécurité dans des environnements complexes et dynamiques. L'ESP32-CAM, avec sa caméra haute résolution et ses capacités de traitement d'image en temps réel, permet aux véhicules autonomes de percevoir leur environnement de manière détaillée. En capturant des images et des vidéos en temps réel, le système peut identifier des obstacles tels que d'autres véhicules, piétons, cyclistes, et objets sur la route. Les algorithmes de traitement d'image embarqués analysent ces données pour détecter, classer, et évaluer les obstacles, permettant au véhicule de prendre des décisions éclairées sur l'évitement des collisions et la modification de la trajectoire [22].

1.5.3 Drones

Les systèmes de détection d'obstacles basés sur l'ESP32-CAM sont également essentiels dans le domaine des drones, permettant une navigation autonome et sécurisée. En intégrant une caméra haute résolution et des algorithmes de traitement d'image embarqués, les drones peuvent capturer des images en temps réel de leur environnement et analyser ces données pour identifier et éviter les obstacles tels que les bâtiments, les arbres, et les lignes électriques. Cette capacité de détection et d'évitement est cruciale pour les drones opérant dans des environnements urbains ou naturels complexes, où la présence d'obstacles peut représenter un danger significatif [23].

1.5.4 Sécurité domestique

Les systèmes de détection d'obstacles basés sur l'ESP32-CAM offrent également des applications puissantes et innovantes dans le domaine de la sécurité domestique. En intégrant une caméra haute résolution, l'ESP32-CAM peut surveiller en temps réel l'intérieur et l'extérieur des habitations, capturant des images et des vidéos de haute qualité. Ces systèmes peuvent être utilisés pour détecter des intrusions, surveiller les mouvements et identifier des situations anormales. Grâce aux capacités de traitement d'image embarquées, le système peut analyser les flux vidéo pour reconnaître les visages, détecter les mouvements et différencier entre des événements normaux et potentiellement dangereux, comme une porte laissée ouverte ou un étranger dans le jardin. Les notifications en temps réel peuvent être envoyées aux propriétaires via des applications mobiles connectées, leur permettant de réagir immédiatement à toute situation suspecte. La connectivité sans fil de l'ESP32 permet également une intégration facile avec d'autres dispositifs de sécurité domestique, comme les alarmes, les serrures électroniques et les systèmes d'éclairage. Cette synergie entre les différentes technologies de sécurité améliore la protection globale de la maison. En somme, l'ESP32-CAM enrichit les systèmes de sécurité domestique en fournissant une surveillance intelligente, réactive et automatisée, assurant ainsi une meilleure protection et une tranquillité d'esprit pour les résidents [24].

1.5.5 Automatisation industrielle

Les systèmes de détection d'obstacles basés sur l'ESP32-CAM jouent un rôle essentiel dans l'automatisation industrielle, où la précision, la sécurité et l'efficacité sont primordiales. Dans les environnements industriels, ces systèmes peuvent être utilisés pour surveiller les lignes de production, détecter les anomalies et assurer le bon fonctionnement des équipements. La caméra haute résolution de l'ESP32-CAM capture des images détaillées des machines et des processus en cours, permettant une analyse en temps réel. Les algorithmes de traitement d'image embarqués peuvent identifier des obstacles ou des défaillances potentielles, comme des pièces mal alignées, des obstructions sur les convoyeurs, ou des anomalies dans le mouvement des machines. Cette capacité de détection permet d'automatiser le contrôle qualité et de réagir rapidement aux problèmes, réduisant ainsi les temps d'arrêt et les coûts de maintenance. De plus, la connectivité sans fil de l'ESP32 facilite la communication entre les différents systèmes de l'usine, permettant une intégration fluide avec les systèmes de gestion de la production et les réseaux de l'Internet des objets (IoT) industriels. En combinant la surveillance visuelle continue avec une analyse intelligente, l'ESP32-CAM améliore l'efficacité opérationnelle, la sécurité des travailleurs, et la qualité des produits finis. En somme, l'ESP32-CAM apporte une couche supplémentaire de précision et d'intelligence à l'automatisation industrielle, contribuant à créer des environnements de production plus sûrs, plus efficaces et plus résiliants [25].

Ces exemples montrent la diversité des applications possibles pour les microcontrôleurs et les caméras ESP32-Cam dans les projets de robotique. Notre projet s'inscrit parfaitement dans cette lignée d'innovations. En utilisant un microcontrôleur ESP32-CAM, nous visons à développer une solution économique et efficace pour la détection d'obstructions dans les canalisations. La voiture-robot pourra naviguer de manière contrôlée à travers les tuyaux, capturer des images en temps réel via une application Web et transmettre les données pour une analyse précise, offrant ainsi une alternative pratique et simple aux méthodes traditionnelles de surveillance et d'inspection des canalisations d'eaux usées.

1.6 Avantages de l'ESP32-CAM dans un système de détection d'obstructions

1.6.1 Inspection efficace des canalisations

Le système de détection d'obstructions ESP32-CAM offre une méthode d'inspection extrêmement efficace des canalisations d'eaux usées. Grâce à l'utilisation d'une caméra haute résolution et d'algorithmes de traitement d'image avancés, le système est capable de capturer et d'analyser des images détaillées de l'intérieur des canalisations en temps réel. Cette efficacité permet de couvrir de grandes distances en peu de temps, réduisant ainsi le besoin de longues interruptions de service pour des inspections manuelles. De plus, la voiture-robot équipée du microcontrôleur ESP32-CAM peut naviguer de manière contrôlée à travers les canalisations, atteignant des zones difficiles d'accès où les méthodes traditionnelles d'inspection sont limitées. En fournissant des données précises et fiables sur l'état des canalisations, le système améliore considérablement la capacité à identifier et à localiser les problèmes potentiels, facilitant ainsi la planification des interventions de maintenance [26].

1.6.2 Détection précoce des obstructions

L'un des principaux avantages du système de détection d'obstructions en utilisant l'ESP32-CAM est sa capacité à détecter les obstructions à un stade précoce. En capturant des images haute résolution et en les analysant en temps réel, le système peut identifier les signes précurseurs de blocages, tels que l'accumulation de débris, les dépôts de graisse et les infiltrations de racines d'arbres. Cette détection précoce permet aux gestionnaires des infrastructures de canalisations d'intervenir avant que les obstructions ne deviennent critiques, évitant ainsi des débordements et des dommages coûteux. La capacité à repérer rapidement les anomalies structurelles contribue également à prolonger la durée de vie des canalisations en permettant des réparations préventives. En fournissant des alertes rapides et précises, le système ESP32-CAM permet de maintenir les canalisations en bon état de fonctionnement et de minimiser les risques de perturbations majeures [26].

1.6.3 Réduction des coûts de maintenance

L'intégration de la technologie ESP32-CAM dans les systèmes de détection d'obstructions permet de réaliser des économies significatives sur les coûts de maintenance. Les inspections traditionnelles des canalisations nécessitent souvent des équipes spécialisées et des équipements lourds, entraînant des dépenses élevées en main-d'œuvre et en logistique. En revanche, la voiture-robot équipée du microcontrôleur ESP32-CAM peut effectuer des inspections de manière contrôlée et continue, réduisant ainsi la dépendance à ressources humaines et matérielles coûteuses. De plus, la capacité à détecter les obstructions à un stade précoce permet de planifier des interventions de maintenance ciblées et efficaces, réduisant les coûts associés aux réparations d'urgence et aux interruptions de service imprévues. En optimisant les processus de maintenance et en minimisant les besoins en interventions manuelles, le système ESP32-CAM contribue à une gestion plus économique des infrastructures de canalisations [27].

1.6.4 Amélioration de la sécurité publique

L'amélioration de la sécurité publique est un avantage crucial du système de détection d'obstructions ESP32-CAM. Les obstructions dans les canalisations d'eaux usées peuvent entraîner des débordements, des inondations et des contaminations environnementales, posant des risques importants pour la santé publique. En assurant une surveillance constante et en fournissant des alertes rapides en cas de détection d'obstructions, le système ESP32-CAM permet d'intervenir rapidement pour prévenir ces incidents. La réduction des risques de débordements et de pollutions contribue à protéger les communautés locales et l'environnement. De plus, en maintenant les canalisations en bon état de fonctionnement, le système aide à garantir un approvisionnement fiable en services d'assainissement, essentiel pour la santé et le bien-être des populations. En somme, le système ESP32-CAM joue un rôle vital dans la prévention des dangers liés aux infrastructures de canalisations, renforçant ainsi la sécurité et la résilience des communautés [28].

1.7 Conclusion

Ce chapitre a examiné diverses techniques de détection des obstructions dans les canalisations d'eaux usées, des méthodes traditionnelles aux technologies modernes, ainsi que les applications innovantes et les avantages des microcontrôleurs et caméras ESP32-CAM. Chaque technique présente des avantages et des limitations en termes de coût, de complexité et d'efficacité.

Notre projet utilise l'ESP32-CAM pour créer une solution économique et efficace de détection des obstructions, combinant innovation et praticité. Le prochain chapitre, "Conception du Système", détaillera la conception de notre voiture-robot ainsi que l'application web de contrôle.

Conception du Système

CHAPITRE 2. CONCEPTION DU SYSTEME

2.1 Introduction

Dans ce chapitre, nous allons explorer en détail les différents aspects de la conception de notre système. Nous commencerons par présenter l'architecture générale, en décrivant les composants principaux et leurs rôles. Ensuite, nous examinerons la conception de la voiture-robot, y compris le châssis, le système de propulsion, les systèmes d'éclairage et de communication. Par la suite, la conception de l'application Web qui permet de contrôler et de surveiller la voiture-robot sera détaillée et discutée. Nous inclurons différents diagrammes UML pour illustrer la structure du système et les interactions entre les composants. Cette analyse exhaustive nous permettra de comprendre comment chaque élément contribue à l'efficacité globale du système et de garantir une mise en œuvre harmonieuse et fonctionnelle.

2.2 Architecture générale d'un système de détection d'obstacles

L'architecture générale du système de détection d'obstructions dans les canalisations (voir Figure 2.1) repose sur l'intégration de plusieurs composants clés dans une voiture-robot équipée d'un microcontrôleur ESP32-CAM. Ce système est conçu pour naviguer de manière contrôlée à travers les canalisations, capturer des images en temps réel et analyser les données pour identifier les obstructions. La conception modulaire de ce système permet une maintenance facile et des mises à jour futures sans nécessiter de modifications majeures. Le système comporte deux grandes parties : (1) la voiture-robot (Châssis) et (2) l'application Web de contrôle. La conception de l'application Web de contrôle sera abordé en détail plus tard dans ce chapitre (voir Section 2.3). La voiture-robot se compose principalement du châssis, du système de propulsion, du système d'éclairage, du système de communication et du système de détection d'obstructions. Chaque composant joue un rôle crucial dans le bon fonctionnement du système global, assurant ainsi une détection précise et efficace des obstructions dans les canalisations. L'intégration harmonieuse de ces composants permet au système de fonctionner de manière cohérente et fiable, même dans des environnements difficiles et contraints.

2.2.1 Châssis de la voiture-robot

Le châssis de la voiture-robot (voir Figure 2.2) est un élément essentiel de l'architecture du système, conçu pour être robuste, étanche et suffisamment compact pour naviguer dans des canalisations de diverses tailles. Le choix des matériaux, tels que le plastique ou l'aluminium, assure la durabilité et la résistance nécessaires pour supporter les conditions rigoureuses des canalisations, y compris la présence d'eau, de débris et de produits chimiques corrosifs. Le design du châssis doit également prendre en compte la facilité d'accès pour l'entretien et la réparation, permettant un démontage et un remontage rapides des composants internes. En outre, le châssis doit être conçu pour protéger les composants électroniques sensibles des chocs et des vibrations, garantissant ainsi un fonctionnement stable du système. Les dimensions du châssis sont optimisées pour permettre une mobilité agile à travers les sections étroites et sinueuses des canalisations, tout en offrant suffisamment d'espace pour intégrer tous les composants nécessaires du système.

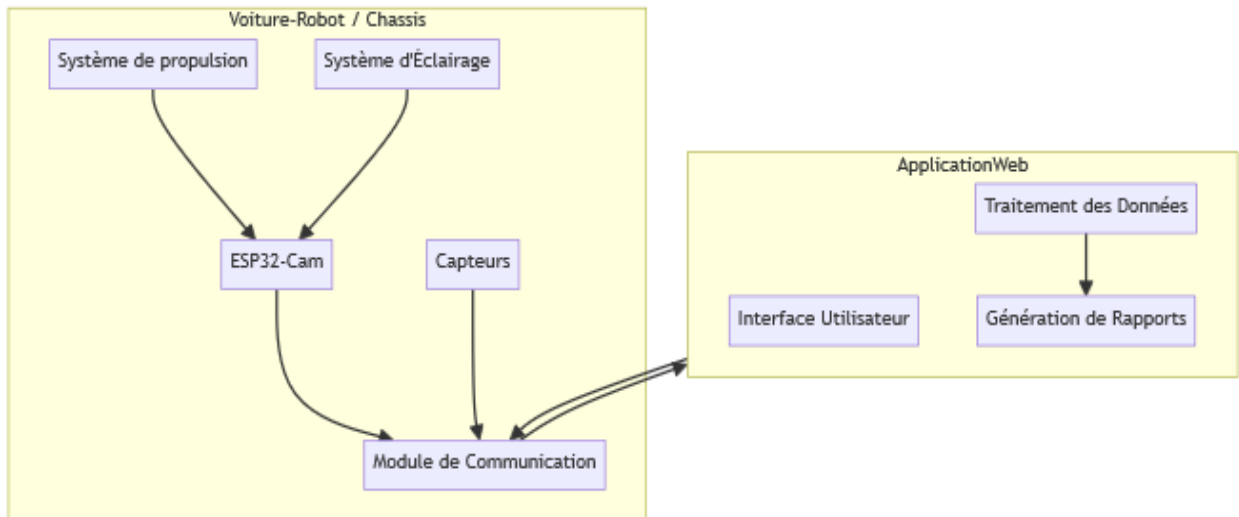


Figure 2.1 - Architecture générale du système

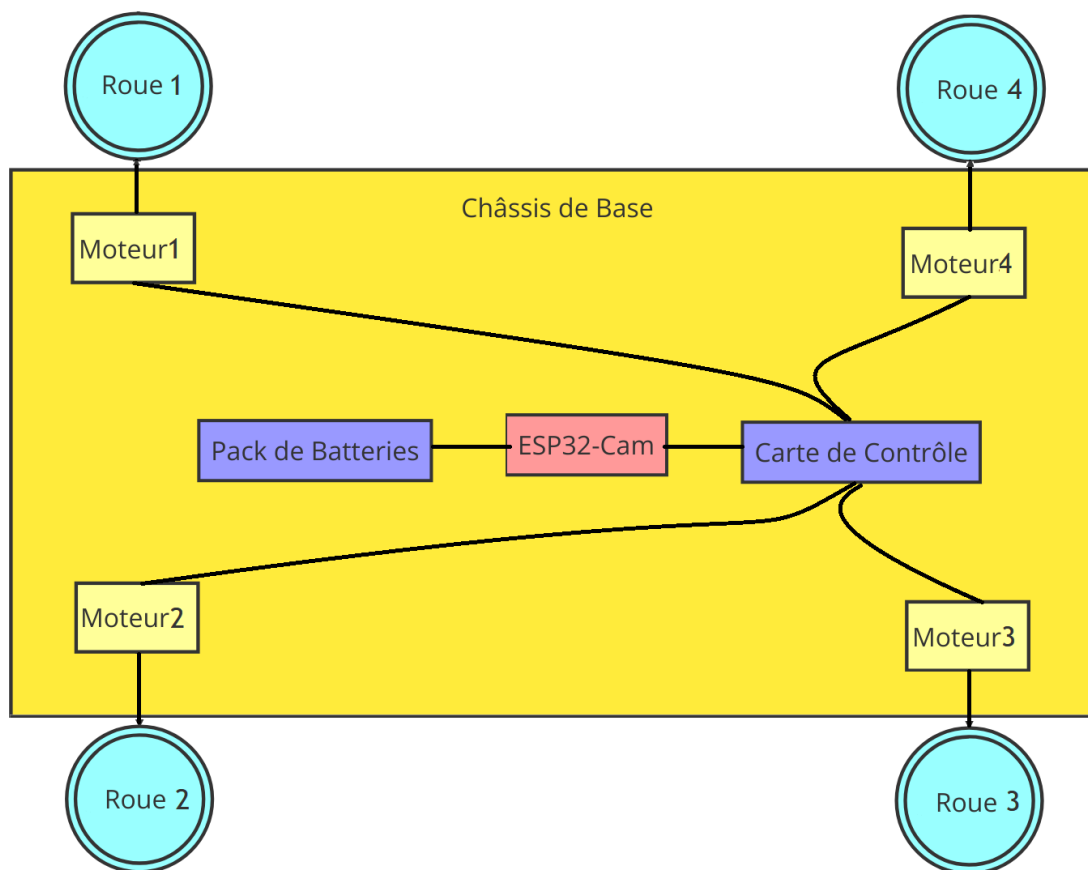


Figure 2.2 - Schéma conceptuel du Châssis avec 4 roues et 4 moteurs Direct Current (DC)

2.2.2 Système de propulsion

Le système de propulsion de la voiture-robot comprend des roues ou des chenilles adaptées aux conditions variées des canalisations. Ce système est conçu pour fournir une mobilité optimale, permettant à la voiture-robot de se déplacer avec agilité et précision. Les moteurs électriques utilisés pour la propulsion doivent être suffisamment puissants pour surmonter les obstacles et les inclinaisons, tout en étant économes en énergie pour maximiser l'autonomie de la batterie. Le contrôle de la propulsion est assuré par des algorithmes de navigation embarqués qui utilisent des capteurs pour ajuster la vitesse et la direction en temps réel, en réponse aux conditions de l'environnement. Ce niveau de contrôle précis est crucial pour éviter les collisions et garantir une navigation fluide à travers les canalisations. De plus, le système de propulsion doit être capable de fonctionner de manière fiable dans des conditions humides et boueuses, ce qui nécessite une conception résistante à l'eau et à la poussière.

2.2.3 Système d'éclairage

Le système d'éclairage de la voiture-robot utilise des LED haute puissance pour illuminer l'intérieur des canalisations, permettant à la caméra de capturer des images claires et détaillées. Un éclairage adéquat est crucial pour le bon fonctionnement du système de détection, surtout dans les environnements sombres des canalisations où la visibilité est limitée. Les LED doivent être disposées de manière à minimiser les ombres et les reflets, garantissant ainsi une illumination uniforme. La gestion thermique des LED est également un aspect important, car une surchauffe pourrait réduire leur efficacité et leur durée de vie. Le système d'éclairage doit être contrôlable à distance, permettant d'ajuster l'intensité lumineuse en fonction des conditions spécifiques de chaque inspection. Un éclairage bien conçu améliore non seulement la qualité des images capturées, mais aussi l'efficacité globale du processus de détection des obstructions.

2.2.4 Système de communication

Le système de communication de la voiture-robot utilise des protocoles sans fil, tels que le Wi-Fi ou le Bluetooth, pour la transmission des données en temps réel entre la voiture-robot et la station de contrôle. Ce système permet une communication bidirectionnelle efficace, facilitant le contrôle à distance de la voiture-robot et la réception des données collectées. La fiabilité et la sécurité de la transmission des données sont des aspects cruciaux, notamment pour éviter les interférences et les pertes de signal. Le système doit également être capable de fonctionner dans des environnements contraints, où les signaux peuvent être atténués par les parois des canalisations. En intégrant des modules de communication avancés, le système peut assurer une connexion stable et continue, même sur de longues distances. Cette capacité à transmettre des données en temps réel permet aux opérateurs de surveiller l'état des canalisations en direct et de prendre des décisions rapides en cas de détection d'obstructions ou d'autres anomalies.

2.2.5 Système de détection d'obstructions

Le système de détection d'obstructions repose sur plusieurs composants, dont des capteurs infrarouges, des capteurs ultrasons et une caméra haute résolution. Ces capteurs travaillent en synergie pour fournir une analyse complète de l'intérieur des canalisations. Les capteurs infrarouges et ultrasons détectent les anomalies structurelles et les obstructions avec une grande précision, tandis que la caméra haute résolution capture des images détaillées qui sont analysées

en temps réel par des algorithmes de traitement d'image. Ce système doit être capable de distinguer entre différents types d'obstructions, comme les dépôts de graisse, les racines d'arbres, et les débris solides, permettant ainsi une intervention ciblée. Les données collectées par ces capteurs sont intégrées et traitées pour générer une vue complète et précise de l'état des canalisations, facilitant la planification des interventions de maintenance.

2.2.6 Schéma du système

La figure 2.3 du système montre les connexions et interactions entre les différents composants, incluant le châssis, les capteurs, la caméra, les moteurs, et le module de communication. Ce schéma est essentiel pour la compréhension globale de l'architecture et pour guider la mise en œuvre et les tests du système.

En résumé, l'architecture générale du système de détection d'obstructions dans les canalisations à l'aide d'une voiture-robot équipée d'un microcontrôleur ESP32-CAM intègre des technologies avancées de capture et de traitement d'image, de communication sans fil, et de navigation autonome. Ce système est conçu pour offrir une solution efficace, économique et pratique pour l'inspection et la maintenance des infrastructures de canalisations d'eaux usées.

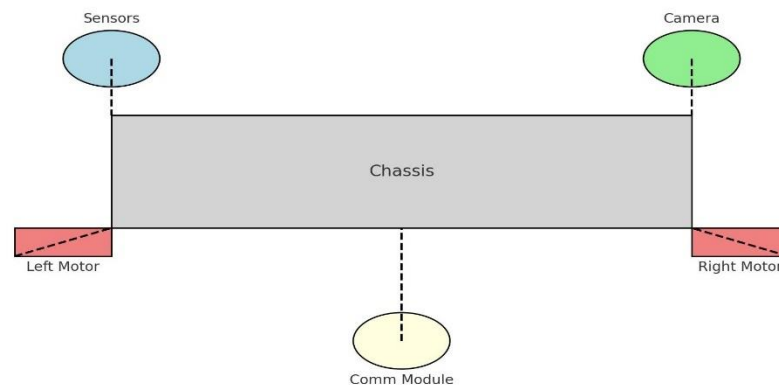


Figure 2.3 - Schéma conceptuel montre la disposition des différents composants

2.2.7 Fonctionnement du système

Le fonctionnement du système de détection d'obstructions dans les canalisations repose sur l'intégration et l'interaction de plusieurs composants clés, chacun jouant un rôle spécifique dans le processus global. Voici une explication détaillée de chaque composant et de son rôle dans le système :

- **Navigation contrôlée**

La voiture-robot se déplace à travers les canalisations en utilisant un système de propulsion composé de roues ou de chenilles. Ce système est guidé par des algorithmes de navigation embarqués qui utilisent des capteurs pour ajuster la vitesse et la direction en temps réel. Ces capteurs, tels que les gyroscopes et les accéléromètres, fournissent des données cruciales sur l'orientation et le mouvement de la voiture-robot, permettant une navigation précise même dans des environnements complexes.

- **Capture d'images en temps réel**

La caméra ESP32-CAM joue un rôle central en capturant des images haute résolution de l'intérieur des canalisations. Montée sur la voiture-robot, cette caméra est configurée pour

fonctionner efficacement dans des conditions de faible luminosité grâce à l'intégration de LED haute puissance. Les images capturées sont essentielles pour l'analyse des obstructions et la vérification de l'intégrité structurelle des canalisations.

- **Éclairage optimal**

Le système d'éclairage, composé de LED haute puissance, assure une illumination suffisante à l'intérieur des canalisations, permettant à la caméra de capturer des images claires et détaillées. Les LED sont positionnées stratégiquement pour minimiser les ombres et les reflets, améliorant ainsi la qualité des images et facilitant l'analyse des obstructions. La gestion thermique des LED est également prise en compte pour prévenir la surchauffe et prolonger leur durée de vie.

- **Transmission des données**

Le système de communication sans fil, utilisant des protocoles comme le Wi-Fi ou le Bluetooth, permet la transmission en temps réel des images et des vidéos capturées par la caméra ESP32-CAM à une station de contrôle. Ce système assure une communication bidirectionnelle, permettant à l'opérateur de surveiller et de contrôler la voiture-robot à distance. La stabilité et la sécurité de la transmission des données sont cruciales pour éviter les interférences et garantir une surveillance continue.

- **Analyse des données**

Les images et vidéos transmises sont analysées en temps réel par des algorithmes de traitement d'image embarqués. Ces algorithmes utilisent des techniques de vision par ordinateur pour détecter les obstructions, identifier leur nature (débris solides, dépôts de graisse, racines d'arbres, etc.) et déterminer leur emplacement précis dans les canalisations. L'analyse des données permet de générer des rapports détaillés sur l'état des canalisations, facilitant la planification des interventions de maintenance.

- **Rapport et intervention**

Les résultats de l'analyse des données sont rapportés à l'opérateur via l'interface utilisateur de la station de contrôle. L'opérateur peut alors visualiser les images et les vidéos des obstructions détectées, évaluer la gravité des problèmes et planifier des interventions ciblées pour éliminer les obstructions identifiées. Cette étape finale assure une gestion proactive et efficace des canalisations d'eaux usées, réduisant les risques de débordement et de dommages environnementaux.

2.3 Conception de l'application Web de contrôle

2.3.1 Diagramme de Cas d'Utilisation (Use Case Diagram)

Nous avons utilisé le langage de modélisation UML (Unified Modeling Language)². Le diagramme de cas d'utilisation représente les interactions entre les utilisateurs (acteurs) et le système. Pour notre système de détection d'obstructions dans les canalisations d'eaux usées, les principaux acteurs peuvent être l'opérateur système et l'administrateur (voir Figure 2.4). Dans ce qui suit les acteurs principaux et les cas d'utilisations dans notre système :

Acteurs :

- **Opérateur Système :** Personne qui utilise le système pour contrôler la voiture-robot.
- **Administrateur :** Personne qui gère les paramètres et les utilisateurs du système.

² <https://www.uml.org>

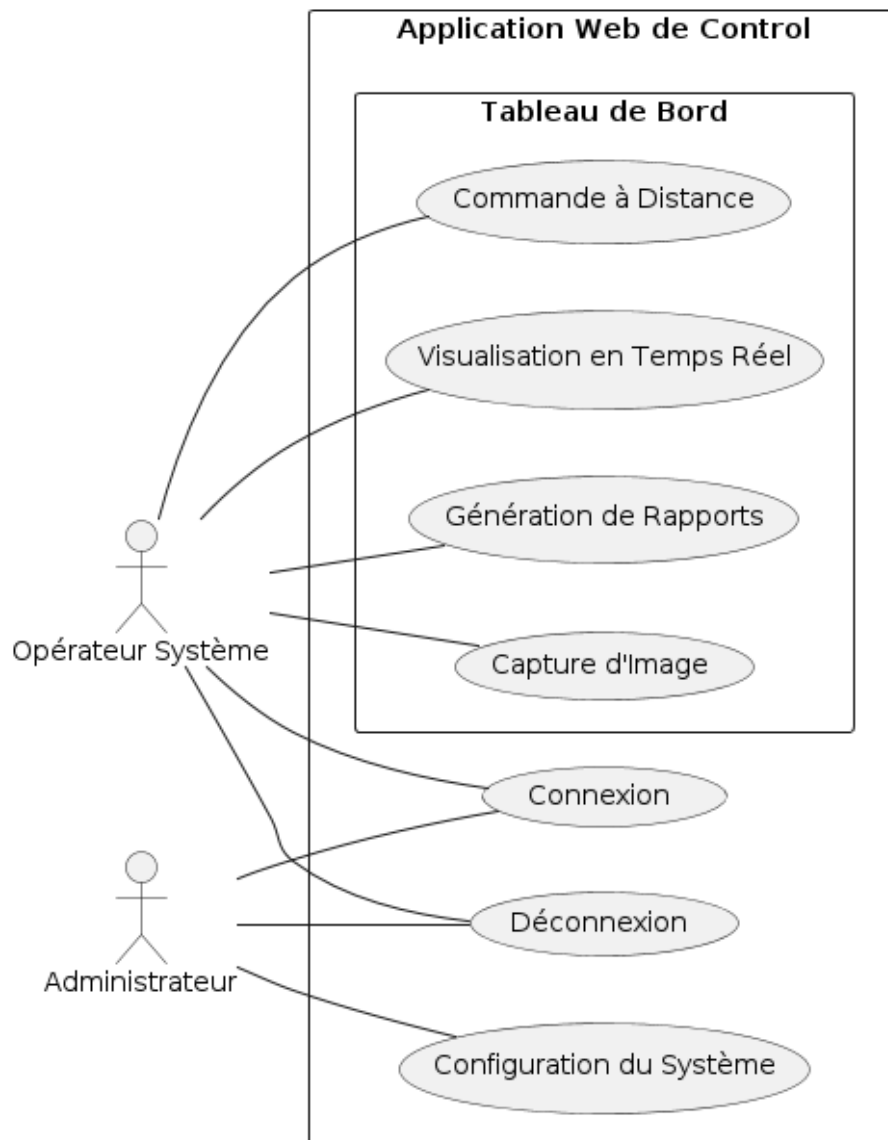


Figure 2.4 - Diagramme de Cas d'Utilisation (Use Cases)

Description des Cas d'Utilisation :

– **Connexion :**

- **Acteur :** Opérateur Système, Administrateur
- **Description :** L'utilisateur entre ses identifiants pour accéder au système.

– **Visualisation en Temps Réel :**

- **Acteur :** Opérateur Système
- **Description :** Permet à l'utilisateur de voir le flux vidéo en direct de la caméra ESP32-Cam pour surveiller les canalisations.

– **Commande à Distance :**

- **Acteur :** Opérateur Système
- **Description :** L'utilisateur peut envoyer des commandes pour déplacer la voiture-robot et contrôler ses actions (capture d'image, réinitialisation, etc.).

- **Capture d'Image :**
 - **Acteur :** Opérateur Système
 - **Description :** L'utilisateur peut capturer des images à partir du flux vidéo pour une analyse plus approfondie.

- **Configuration du Système :**
 - **Acteur :** Administrateur
 - **Description :** L'administrateur peut configurer les paramètres de la voiture-robot et du système (réglages des capteurs, paramètres réseau, etc.).

- **Génération de Rapports :**
 - **Acteur :** Opérateur Système, Administrateur
 - **Description :** Permet la création de rapports basés sur les données collectées par le système pour l'analyse et la documentation.

- **Déconnexion :**
 - **Acteur :** Opérateur Système, Administrateur
 - **Description :** L'utilisateur peut se déconnecter du système de manière sécurisée.

2.3.2 Diagramme d'Activités (Activity Diagram)

Un diagramme d'activités représente les flux de travail ou les processus dans un système. Pour notre système de détection d'obstructions dans les canalisations d'eaux usées, nous avons créé un diagramme d'activités décrivant le processus global depuis la connexion jusqu'à la détection des obstructions et la génération de rapports (voir Figure 2.5).

Description des Activités :

1. **Début :** Initialisation du processus.
2. **Connexion :** L'utilisateur entre ses identifiants pour accéder au système.
3. **Accès au Tableau de Bord :** L'utilisateur accède au tableau de bord après la connexion réussie.
4. **Visualisation en Temps Réel :** L'utilisateur visualise le flux vidéo de la caméra ESP32-Cam en temps réel.
5. **Commande de la Voiture-Robot :** L'utilisateur envoie des commandes pour déplacer la voiture-robot.
6. **Capture d'Image :** En cas de détection d'obstruction, l'utilisateur capture une image.
7. **Génération de Rapports :** L'utilisateur génère un rapport basé sur les données collectées.
8. **Déconnexion :** L'utilisateur se déconnecte du système.
9. **Fin :** Terminaison du processus.

Ce diagramme d'activités offre une vision claire des étapes séquentielles du processus utilisateur dans notre système. Chaque activité représente une étape cruciale pour la détection et la gestion des obstructions dans les canalisations d'eaux usées.

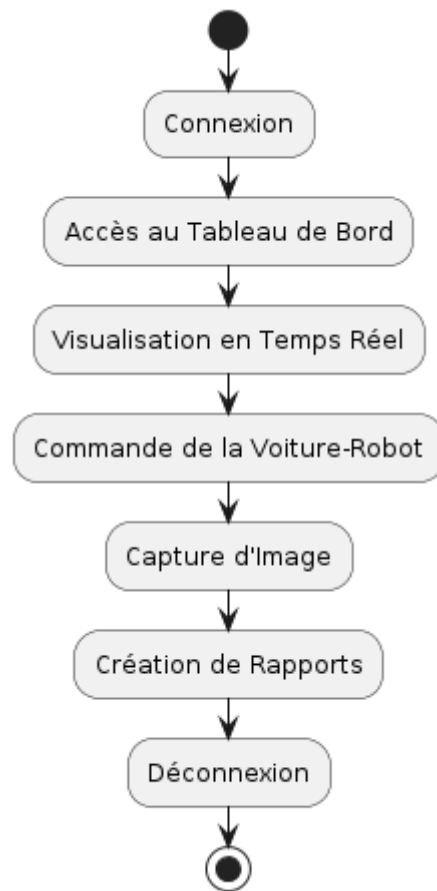


Figure 2.5 - Diagramme d'Activités (Activity Diagram)

2.3.3 Diagramme d'États (State Diagram)

Le diagramme d'états illustre les différents états par lesquels passe la voiture-robot équipée du microcontrôleur ESP32-CAM dans le système de détection d'obstructions des canalisations (voir Figure 2.6). Ce diagramme est essentiel pour comprendre les transitions entre les états et les conditions déclenchant ces transitions. Il aide à visualiser le comportement dynamique du système en réponse à divers événements.

États et Transitions :

- **Initialisation** : L'état initial où le système démarre et effectue les configurations nécessaires.
 - Transition vers **Attente** lorsque l'initialisation est réussie (Init OK).
- **Attente** : L'état où le système est prêt et en attente de commandes.
 - Transition vers **Mouvement** lors de la réception d'une commande de déplacement (Commande Reçue).
 - Transition depuis **Mouvement** lorsque le (Mouvement Terminé).
 - Transition depuis **Détection** si aucune obstruction n'est détectée (Obstruction Non Détectée).
 - Transition depuis **Analyse** lorsque l'analyse est complétée (Analyse Complétée).
 - Transition depuis **Rapport** après la génération du rapport (Rapport Généré).
- **Mouvement** : L'état où la voiture-robot se déplace.
 - Transition vers **Détection** si une obstruction est détectée (Obstruction Détectée).
 - Transition vers **Attente** lorsque le déplacement est terminé (Mouvement Terminé).

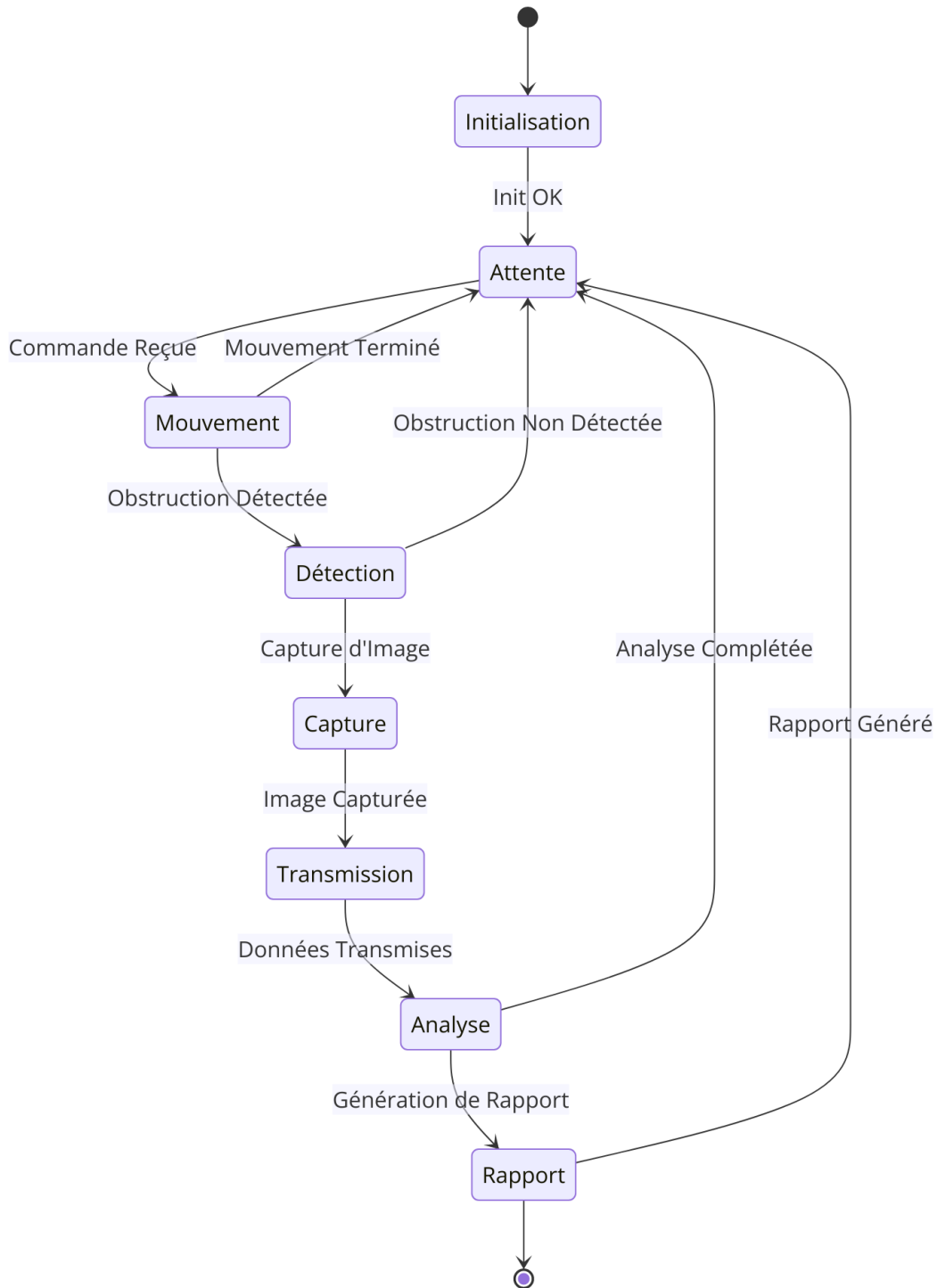


Figure 2.6 - Diagramme de d'états (State Diagram)

- **Détection** : L'état où le système détecte des obstructions dans les canalisations.
 - Transition vers **Capture** si une obstruction est détectée (Capture d'Image).
 - Transition vers **Attente** si aucune (Obstruction Non Détectée).
- **Capture** : L'état où le système capture des images de l'obstruction détectée.
 - Transition vers **Transmission** une fois l'image capturée (Image Capturée).
- **Transmission** : L'état où les données (images capturées) sont transmises à la station de contrôle.
 - Transition vers **Analyse** après la transmission des données (Données Transmises).

- **Analyse** : L'état où les données transmises sont analysées pour générer des rapports.
 - Transition vers **Rapport** après l'analyse des données (Génération de Rapport).
 - Transition vers **Attente** lorsque l'analyse est complétée (Analyse Complétée).
- **Rapport** : L'état où un rapport est généré basé sur les données analysées.
 - Transition vers **Attente** après la génération du rapport (Rapport Généré).

Ce diagramme fournit une vue claire et structurée des états opérationnels du système et des transitions associées, facilitant ainsi la compréhension et la gestion des différentes phases du fonctionnement de la voiture-robot.

2.3.4 Diagramme de Séquence (Sequence Diagram)

Un diagramme de séquence montre comment les objets interagissent dans un certain ordre au cours d'un processus spécifique. Pour notre système de détection d'obstructions dans les canalisations d'eaux usées, nous avons créé un diagramme de séquence (voir Figure 2.7) décrivant les interactions entre l'utilisateur, l'application web, la voiture-robot, le microcontrôleur ESP32-Cam, et la base de données (BDD) pendant le processus de surveillance et de commande. Ce diagramme fournit une vue détaillée des interactions dynamiques entre les différents composants de votre système, ce qui est essentiel pour comprendre le flux des opérations entre les acteurs et les objets.

Séquences Principales :

1. **Connexion** :
 - L'utilisateur se connecte à l'application web.
 - L'application web vérifie les identifiants auprès de la base de données.
 - Une fois authentifié, l'utilisateur accède au tableau de bord.
2. **Visualisation du Flux Vidéo** :
 - L'utilisateur demande à visualiser le flux vidéo en temps réel.
 - L'application web envoie une demande de streaming à la caméra ESP32-Cam.
 - La caméra ESP32-Cam transmet le flux vidéo en temps réel à l'application web.
 - L'application web affiche le flux vidéo à l'utilisateur.
3. **Commande de Mouvement** :
 - L'utilisateur envoie une commande de mouvement via l'application web.
 - L'application web transmet cette commande à la voiture-robot.
 - La voiture-robot déplace la caméra et confirme le déplacement.
4. **Capture d'Image** :
 - L'utilisateur demande la capture d'une image.
 - L'application web envoie une demande de capture à la caméra ESP32-Cam.
 - La caméra ESP32-Cam capture l'image et la transmet à l'application web.
 - L'application web enregistre l'image dans la base de données et confirme l'enregistrement à l'utilisateur.
5. **Génération de Rapport** :
 - L'utilisateur demande la génération d'un rapport.
 - L'application web extrait les données nécessaires de la base de données.
 - L'application web génère le rapport et le présente à l'utilisateur.
6. **Déconnexion** :
 - L'utilisateur se déconnecte de l'application web.
 - L'application web confirme la déconnexion.

Ce diagramme montre de manière détaillée comment les différents composants du système interagissent pour accomplir des tâches spécifiques, offrant ainsi une vue claire des processus internes et des flux de données essentiels au bon fonctionnement du système.

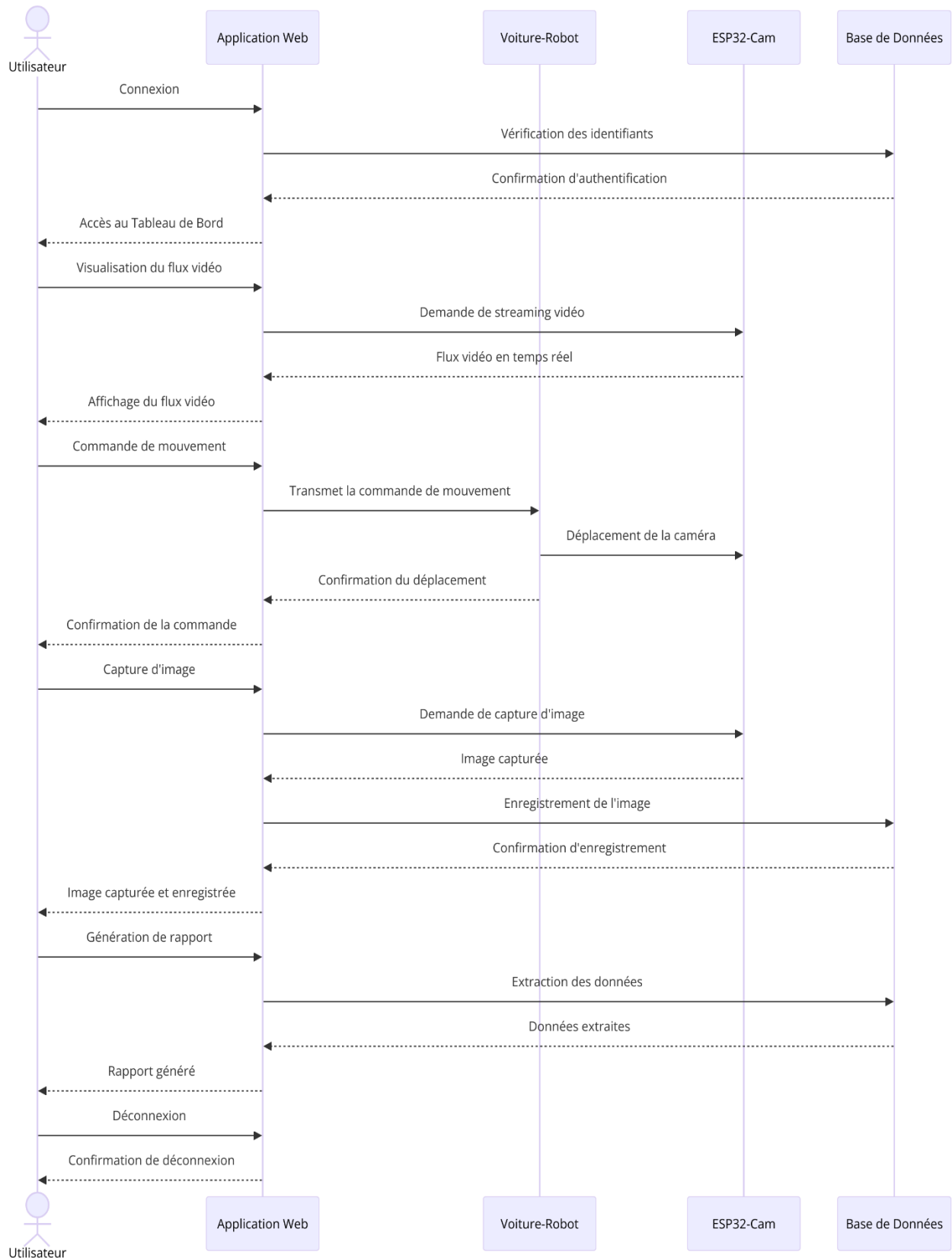


Figure 2.7 - Diagramme de Séquence (Sequence Diagram)

2.3.5 Diagramme de Classe (Class Diagram)

Un diagramme de classe montre les structures de données statiques du système, notamment les classes, leurs attributs, méthodes, et les relations entre elles. Pour notre système de détection d'obstructions dans les canalisations d'eaux usées, nous avons créé un diagramme de classe (voir Figure 2.8) représentant les principales entités et leurs relations. Ce diagramme de classe donne une vue d'ensemble de la structure statique de votre système, en montrant les classes principales, leurs attributs, méthodes et les relations entre elles.

- **Utilisateur** : Représente un utilisateur du système avec ses informations personnelles et les actions qu'il peut effectuer, comme se connecter, se déconnecter, visualiser le flux vidéo, envoyer des commandes, capturer des images, et générer des rapports.
- **Application Web** : Représente l'application web qui gère la connexion des utilisateurs, la transmission des commandes, l'affichage du flux vidéo et la compilation des rapports.
- **VoitureRobot** : Représente la voiture-robot équipée pour détecter les obstructions. Elle peut recevoir des commandes de mouvement et détecter des obstructions.
- **ESP32-Cam** : Représente le microcontrôleur ESP32-Cam qui fournit le flux vidéo en temps réel et capture les images.
- **Rapport** : Représente les rapports générés à partir des données collectées par la voiture-robot et le ESP32-Cam.
- **Commande** : Représente les commandes envoyées par l'utilisateur pour contrôler la voiture-robot.

2.3.6 Diagramme Modèle de Données (Data Model)

Le modèle de données (voir Figure 2.9) représente la structure des données pour le système de détection d'obstructions dans les canalisations. Il inclut les entités principales, leurs attributs et les relations entre elles.

Entités et Attributs :

1. **Utilisateur**
 - utilisateurID (PK)
 - nom
 - prenom
 - email
 - motDePasse
2. **Connexion**
 - connexionID (PK)
 - dateHeure
 - utilisateurID (FK)
3. **Vidéo**
 - videoID (PK)
 - cheminFichier
 - dateCapture
 - utilisateurID (FK)

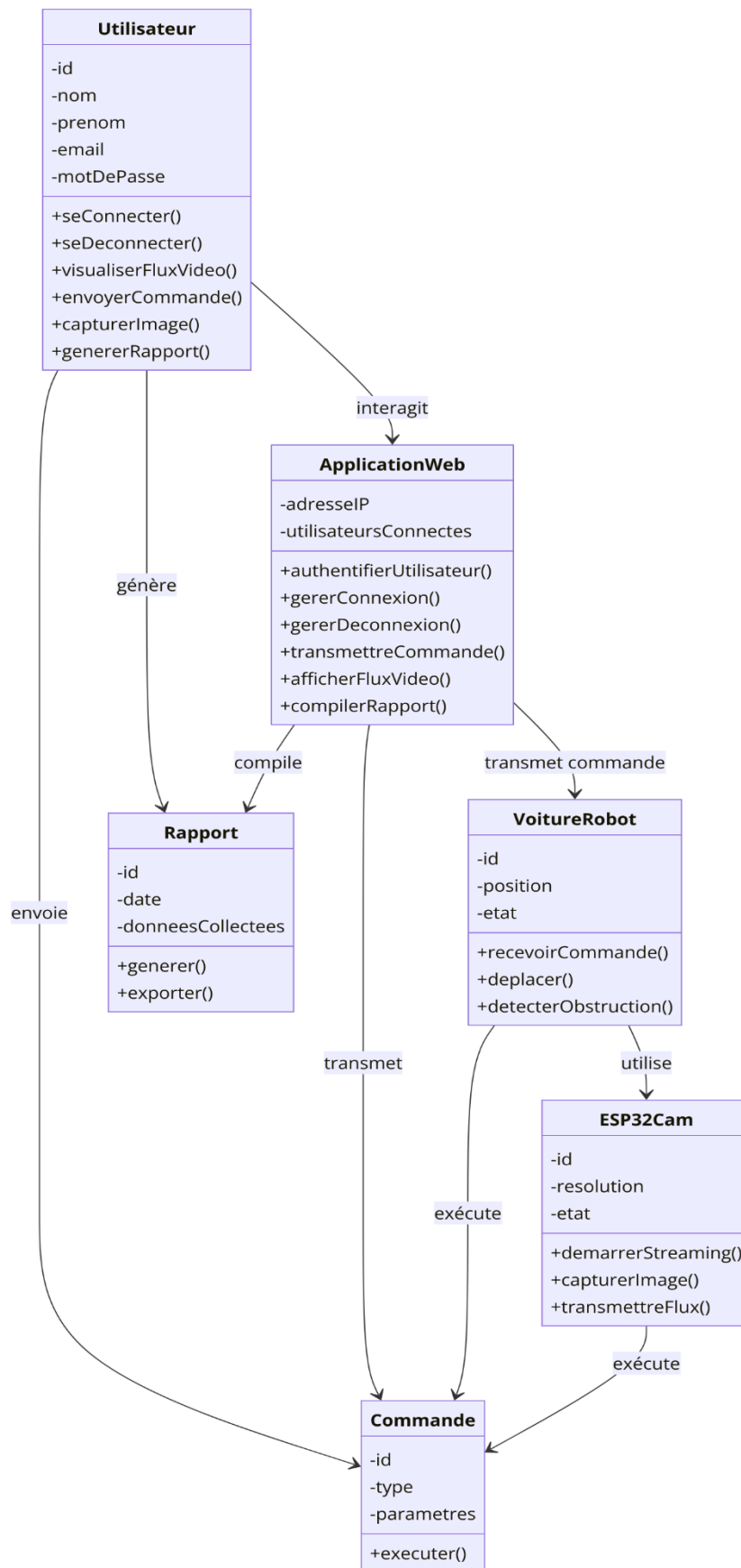


Figure 2.8 - Diagramme de Classe (Class Diagram)

4. Rapport

- rapportID (PK)
- titre
- contenu
- dateGeneration
- utilisateurID (FK)

5. Commande

- commandeID (PK)
- type
- parametres
- dateHeure
- utilisateurID (FK)

Relations :

- **Utilisateur** effectue des **Connexions**
- **Utilisateur** capture des **Vidéos**
- **Utilisateur** génère des **Rapports**
- **Utilisateur** envoie des **Commandes**
- **Connexion** déclenche des **Commandes**
- **Vidéo** est incluse dans des **Rapports**

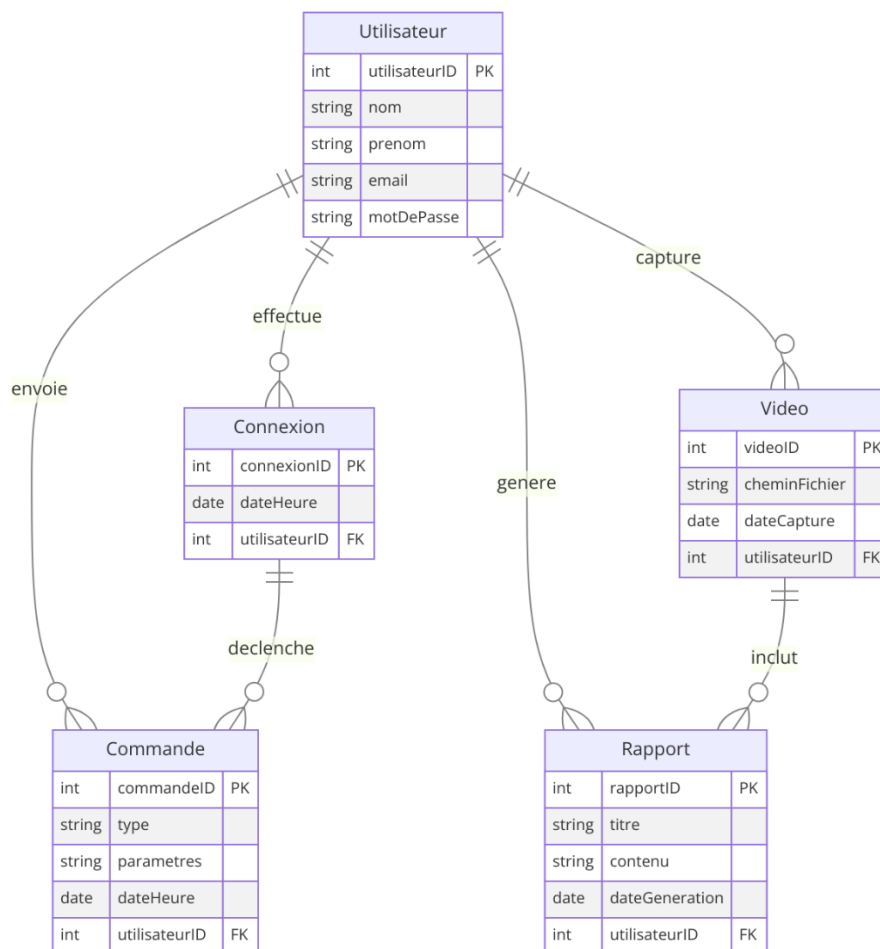


Figure 2.9 - Diagramme Modèle de Données (Data Model)

2.3.7 Diagramme de Composants (Component Diagram)

Le diagramme de composants illustre les différents modules logiciels de notre système de détection d'obstructions dans les canalisations d'eaux usées (voir Figure 2.10). Ce diagramme permet de visualiser la structure du système, les interactions entre les composants logiciels et les dépendances existantes.

Composants Principaux

- **Application Web :**
 - **Interface Utilisateur (UI) :** Fournit l'interface graphique pour l'utilisateur.
 - **Service d'Authentification (Auth) :** Gère l'authentification des utilisateurs.
 - **Service de Streaming Vidéo (Video) :** Gère la diffusion en continu du flux vidéo de l'ESP32-Cam.
 - **Service de Génération de Rapports (Report) :** Permet la génération de rapports basés sur les données collectées.
 - **Service de Commandes à Distance (Command) :** Transmet les commandes à la voiture-robot.
 - **Base de Données (BDD) :** Stocke les données utilisateurs, les vidéos capturées et les rapports générés.
- **Voiture-Robot :**
 - **Interface de Capteurs (Sensor) :** Gère les données des capteurs de la voiture-robot.
 - **Contrôle des Moteurs (MotorCtrl) :** Gère les mouvements de la voiture-robot.
 - **ESP32-Cam (ESP32Cam) :** Capture et transmet le flux vidéo.
 - **Module de Communication (Communication) :** Assure la communication entre la voiture-robot et l'application web.

Les interactions principales incluent la transmission de commandes de l'application web à la voiture-robot, la capture et le stockage des vidéos par l'ESP32-Cam, et la génération de rapports basés sur les données collectées par les capteurs.

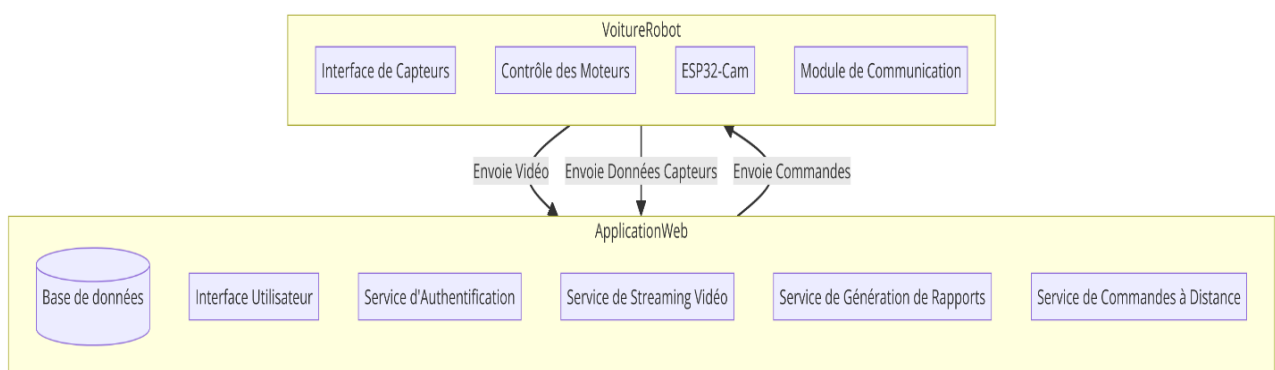


Figure 2.10 - Diagramme de Composants (Component Diagram)

2.3.8 Diagramme de Déploiement (Deployment Diagram)

Le diagramme de déploiement montre la configuration physique du matériel et des logiciels dans notre système de détection d'obstructions (voir Figure 2.11). Ce diagramme aide à comprendre comment le système est déployé dans un environnement réel, en illustrant les interactions entre les composants matériels et logiciels.

Composants Principaux :

- **Serveur :**
 - **Application Web (AppWeb) :** Héberge l'application web de contrôle.
 - **Base de Données (BDD) :** Stocke toutes les données relatives au système.
- **Voiture-Robot :**
 - **ESP32-Cam (ESP32Cam) :** Capture et transmet le flux vidéo en temps réel.
 - **Contrôleur Moteur (Motor) :** Gère les mouvements de la voiture-robot.
 - **Capteurs (Sensors) :** Collecte des données sur les conditions internes des canalisations.
- **Réseau :**
 - **WiFi :** Assure la communication sans fil entre la voiture-robot et le serveur.
 - **4G/5G :** Offre une option de communication mobile pour des connexions à distance.

Les liaisons principales incluent la communication WiFi entre l'application web et la voiture-robot pour la transmission des commandes et des données, ainsi que la communication réseau pour l'accès à distance via 4G/5G.

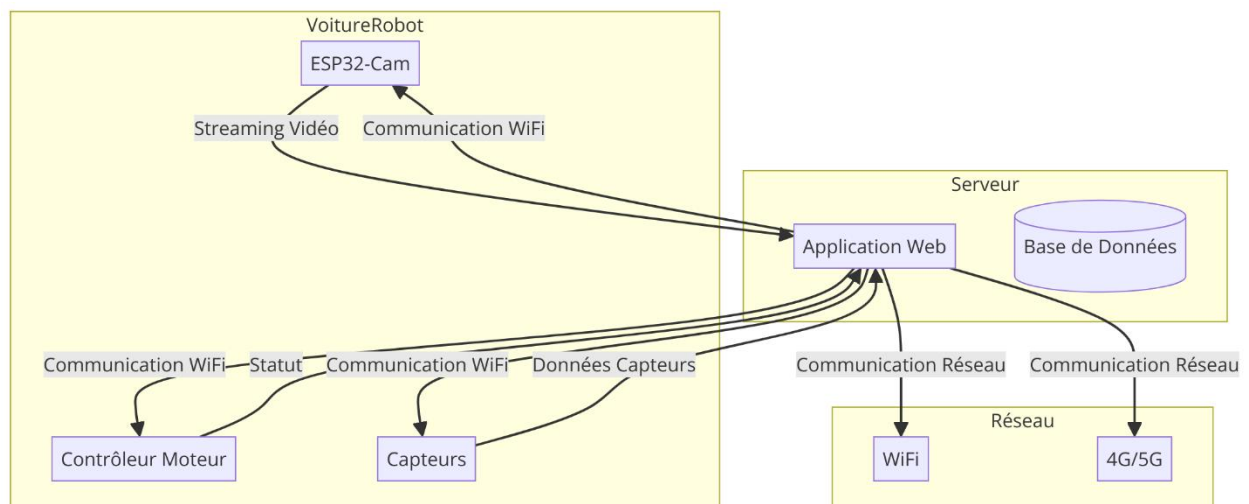


Figure 2.11 - Diagramme de Déploiement (Deployment Diagram)

2.4 Conclusion

Dans ce chapitre, nous avons détaillé la conception du système de détection d'obstructions dans les canalisations d'eaux usées et de l'application web de contrôle. En utilisant divers diagrammes (cas d'utilisation, états, séquence, classes, données, composants et déploiement), nous avons visualisé les interactions entre les composants, les flux de données et les processus de traitement. L'application web permet la visualisation en temps réel, le contrôle à distance de la voiture-robot équipée de l'ESP32-CAM, la capture d'images et la génération de rapports détaillés.

La conception repose sur une architecture modulaire où chaque composant interagit de manière cohérente avec les autres. Cette structure garantit une base solide pour le développement, les tests et la maintenance, assurant un fonctionnement efficace et fiable dans des conditions réelles. Le chapitre suivant décrira la réalisation et les tests du système, en détaillant les étapes de mise en œuvre, les configurations matérielles et logicielles, ainsi que les résultats des tests effectués pour valider les performances et la fiabilité du système.

Réalisation et Test

CHAPITRE 3. REALISATION ET TEST

3.1 Introduction

L'objectif de ce chapitre est de détailler les différentes étapes de la réalisation et de la mise en œuvre du système de détection d'obstructions dans les canalisations d'eaux usées. En intégrant des composants mécaniques robustes, des capteurs avancés et une application web de contrôle, ce système promet d'offrir une solution innovante et efficace qui facilite la gestion des infrastructures d'assainissement en permettant de faire face aux défis complexes de manière proactive et efficiente.

Ce chapitre se concentrera sur l'assemblage de la voiture-robot, l'intégration des capteurs nécessaires, le développement du logiciel pour contrôler le système, et les tests effectués pour valider son fonctionnement. Chaque section couvrira des aspects spécifiques du projet, allant des choix des matériaux et des outils utilisés, à la programmation du microcontrôleur et à la mise en place des tests expérimentaux pour évaluer l'efficacité du système dans des conditions simulées.

3.2 Mise en œuvre de la Voiture-Robot

La mise en œuvre de la voiture-robot constitue une étape cruciale dans la réalisation du système de détection d'obstructions dans les canalisations d'eaux usées. Cette section détaille les différentes considérations prises en compte pour concevoir un prototype efficace et fiable capable de naviguer dans des environnements complexes et contraints.



Figure 3.1 - Différentes vues de la voiture-robot

3.2.1 Dimensions et Forme

Dimensions

1. **Longueur et Largeur** : La voiture-robot présentée sur la Figure 3.1 montre notre voiture-robot sous différents angles avec un châssis relativement compact, adapté aux espaces restreints des canalisations. La longueur de la voiture doit être suffisante pour accueillir la batterie, les moteurs, et le système de contrôle sans compromettre sa maniabilité. Une largeur optimisée permet de s'assurer que la voiture peut naviguer sans se coincer dans les canalisations étroites.
2. **Hauteur** : La hauteur doit être minimisée pour permettre à la voiture-robot de passer sous les obstacles potentiels ou dans des sections de canalisation plus basses. Toutefois, elle doit être suffisante pour protéger les composants électroniques contre les éclaboussures d'eau.

Forme

- **Châssis Rectangulaire** : Le châssis rectangulaire utilisé est un choix judicieux pour maximiser l'espace disponible pour les composants tout en assurant une distribution équilibrée du poids.
- **Roues et Roulettes** : Les roues de la voiture-robot sont larges et équipées de bandes de roulement pour améliorer l'adhérence sur les surfaces humides et inégales des canalisations. Leur positionnement symétrique assure une stabilité optimale pendant les déplacements. La hauteur des roues est ajustée pour offrir un dégagement suffisant entre le châssis et le sol de la canalisation, évitant ainsi les blocages par les petits débris.
- **Support de la Caméra** : La caméra ESP32-CAM à l'avant de la voiture, surélevée légèrement pour une vue dégagée. Cette position stratégique permet à la caméra de capturer des images claires de l'intérieur des canalisations. Un support incliné pourrait être envisagé pour ajuster l'angle de vue de la caméra en fonction des besoins spécifiques de l'inspection.

3.2.2 Matériaux Utilisés

Le choix des matériaux est essentiel pour assurer la durabilité et la résistance de la voiture-robot. Les matériaux sélectionnés doivent résister à l'eau, aux produits chimiques présents dans les eaux usées, et aux impacts mécaniques.

- **Châssis en Plastique** : La Figure 3.1 montrent que le châssis est fabriqué en plastique, un matériau transparent, résistant et léger. Le plastique est idéal pour les prototypes car il est facile à couper et à assembler tout en offrant une bonne rigidité structurelle.
- **Roues en Caoutchouc** : Les roues en caoutchouc, ce qui fournit une meilleure adhérence et une certaine absorption des chocs, important pour les déplacements dans des canalisations potentiellement irrégulières.
- **Batteries** : Les batteries sont placées sur le châssis à l'arrière de manière équilibrée pour éviter les basculements. Leur position centrale par rapport à l'axe des roues arrière aide à maintenir le centre de gravité bas, augmentant ainsi la stabilité de la voiture-robot.

3.2.3 Installation des Composants

Une fois les dimensions et les matériaux définis, l'installation des composants essentiels peut commencer. Cela inclut les moteurs, les roues, les systèmes de direction, et autres éléments mécaniques et électroniques. L'installation doit être réalisée de manière à garantir à la fois la fonctionnalité et la robustesse du système.

- **Moteurs** : Des moteurs à courant continu ou des moteurs pas à pas sont choisis en fonction des besoins de précision et de puissance. Ils doivent être assez puissants pour propulser la voiture-robot à travers les canalisations, même en présence d'obstacles ou de pentes.
- **Roues et Systèmes de Direction** : Les roues en caoutchouc assure une bonne adhérence. Pour les canalisations particulièrement difficiles, des chenilles peuvent être utilisées à la place des roues. Les systèmes de direction doivent permettre des manœuvres précises pour naviguer dans des environnements étroits et sinueux.
- **Étanchéité** : Tous les composants électroniques doivent être protégés contre l'humidité. Des boîtiers étanches sont utilisés pour isoler les circuits électroniques et les moteurs de l'eau et des débris.

Disposition des Composants Électroniques

1. **Organisation des Câbles** : Les câbles reliant les différents composants sont disposés de manière à minimiser l'encombrement et à réduire le risque de déconnexions accidentelles (voir Figure 3.2). Ils sont fixés avec des attaches pour maintenir l'ordre et éviter qu'ils ne se déplacent pendant le fonctionnement de la voiture-robot. Cette organisation est essentielle pour assurer la fiabilité des connexions électriques, surtout dans un environnement de travail potentiellement mouvementé.

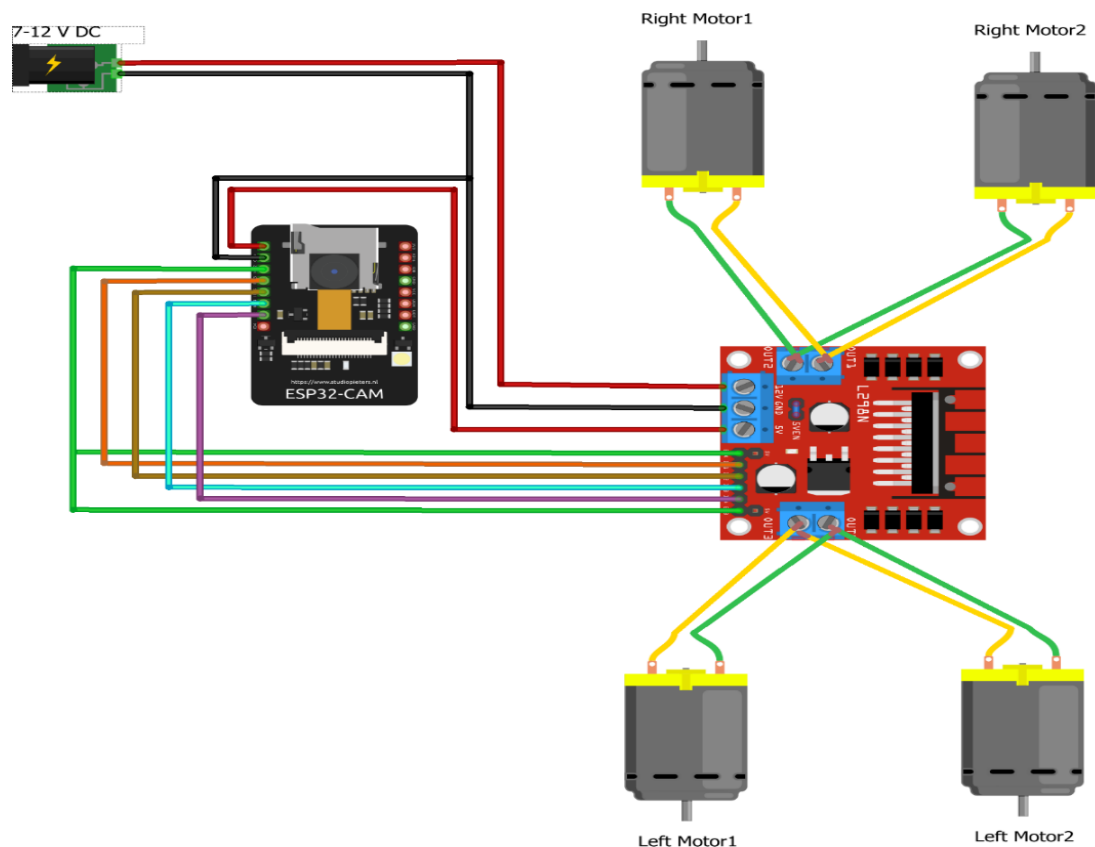


Figure 3.2 - Schéma électronique de la voiture et du ESP-32-Cam

Câblage :

Il est nécessaire de câbler 5 broches de l'ESP32-CAM : les 2 broches d'alimentation, les 2 broches de Universal Asynchronous Receiver-Transmitter (UART) n°0 et enfin la broche GPIO0 qui permet de faire démarrer l'ESP32-CAM en mode "transfert de programme".

Pour l'alimentation en 5V :

- **Fil rouge**, la broche 5V du convertisseur Universal Serial Bus (USB) /TTL (Transistor-Transistor Logic) est à relier au +5V de l'ESP32-CAM.
- **Fil noir**, la broche GND (Ground) du convertisseur USB/TTL est à brancher à la masse de l'ESP32-CAM (GND).
- Pour la communication TTL :
- **Fil vert**, la broche Tx/D (Transmission Data) du convertisseur USB/TTL est à relier à l'entrée U0R de l'ESP32-CAM, elle permet de transmettre les données vers la cible.
- **Fil bleu**, la broche Rx/D du convertisseur USB/TTL connectée à la sortie U0T (Transmission Data of UART number 0) de l'ESP32-CAM reçoit les données transmises par la cible.
- Pour le démarrage en mode "transfert de programme":
- **Fil jaune**, la broche IO0 (Input/Output 0) de l'ESP32-CAM est à mettre à la masse (GND).

2. **Montage du Microcontrôleur ESP32-CAM** : Le microcontrôleur ESP32-CAM est monté à l'avant de la voiture-robot sur un support dédié (voir Figure 3.3). Ce positionnement permet à la caméra de capturer des images claires de l'intérieur des canalisations sans obstruction. Le support incliné optimise l'angle de vue, garantissant que les images capturées couvrent une zone suffisante pour détecter les obstructions. De plus, le support protège le microcontrôleur contre les vibrations et les chocs.

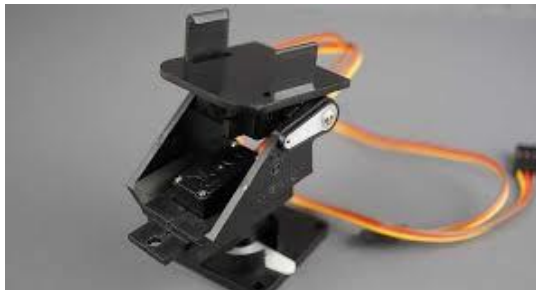


Figure 3.3 - Support microcontrôleur ESP32-CAM

3. **Batteries** : Les batteries, placées à l'arrière du châssis, sont montées dans un support qui les maintient fermement en place (voir Figure 3.4). Leur position centrale par rapport à l'axe des roues arrière contribue à maintenir un centre de gravité bas, augmentant ainsi la stabilité de la voiture-robot. Cette disposition est cruciale pour éviter les basculements, surtout lorsque la voiture-robot se déplace sur des surfaces inégales ou rencontre des obstacles.



Figure 3.4 - Jeu des Batteries et leur Support

Modules de Contrôle des Moteurs

1. **Contrôleurs de Moteurs** : Le module de contrôle des moteurs (voir Figure 3.5) est installé près des moteurs pour réduire la longueur des câbles nécessaires, ce qui diminue la résistance et améliore l'efficacité électrique. Ce module est monté sur un support isolé pour protéger les circuits contre les vibrations et les chocs.



Figure 3.5 - Contrôleur Moteur

Caractéristiques du driver :

- **Pilote** : L298N de type double canal en pont en H ;
 - **Tension logique** : 5VCC ;
 - **Tension d'entraînement** : 5v à 35v ;
 - **Courant d'entraînement** : 2A ;
 - **Puissance** : 25W ;
 - **Dimensions** : 43*43*27mm ;
2. **Fixation des Moteurs** : Les moteurs sont fixés solidement au châssis avec des supports en plastique robuste (voir Figure 3.6). Cette fixation assure que les moteurs restent en place même lors de manœuvres brusques ou lorsque la voiture-robot rencontre des obstacles. Les supports sont également conçus pour absorber les vibrations, réduisant ainsi l'usure des composants mécaniques et électroniques.



Figure 3.6 - Fixation des Moteurs

Systèmes d'Éclairage

- **Installation des LED** : Les systèmes d'éclairage LED sont installés autour de la caméra ESP32-CAM pour assurer une illumination uniforme de l'intérieur des canalisations. La LED est fixée sur le EPS32Cam permettant de diriger la lumière précisément là où elle est nécessaire, améliorant ainsi la qualité des images capturées (voir Figure 3.7).



Figure 3.7 - LED ESP32-Cam

- **Protection des Composants** : Tous les composants électroniques sont protégés par un boîtier résistant à l'eau et à la poussière (voir Figure 3.8). Cette protection est essentielle pour garantir la longévité des composants dans un environnement de travail potentiellement humide et sale.



Figure 3.8 - Cache de protection

3.2.4 Caméra ESP32-CAM

La caméra ESP32-CAM est l'un des composants centraux du système de détection (voir Figure 3.9). Elle permet de capturer des images en temps réel de l'intérieur des canalisations, fournissant des données visuelles cruciales pour identifier les obstructions

- **Fixation de la Caméra** : La caméra est montée à l'avant de la voiture-robot, orientée de manière à capturer une vue claire des parois internes des canalisations. La fixation doit être robuste pour résister aux vibrations et aux chocs pendant le déplacement de la voiture-robot.
- **Étanchéité** : Pour protéger la caméra de l'eau et des débris, un boîtier étanche est utilisé. Ce boîtier est équipé d'une fenêtre transparente devant l'objectif de la caméra (voir Figure 3.3).
- **Connectivité** : La caméra ESP32-CAM est connectée à l'unité de contrôle (ESP32) via des

interfaces de communication appropriées, pour permettre la transmission en temps réel des images capturées.

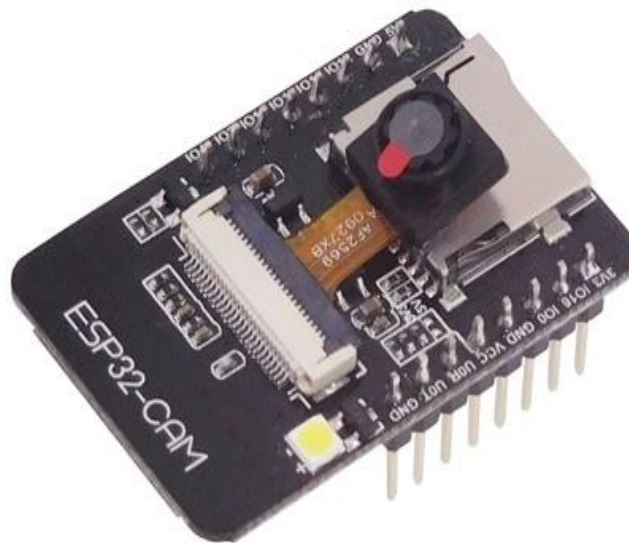


Figure 3.9 - Camera EPS32-Cam

3.3 Programmation du Microcontrôleur ESP32-CAM

Le développement du microcontrôleur ESP32-CAM est une étape cruciale dans la réalisation du système de détection d'obstructions dans les canalisations d'eaux usées. Cette section aborde en détail les aspects techniques liés à la configuration de la caméra, à la programmation du microcontrôleur, et les configurations nécessaires de l'environnement Arduino IDE.

3.3.1 Contrôle de la Voiture-Robot

La première étape du développement logiciel consiste à programmer l'ESP32 pour contrôler les moteurs, la caméra et les capteurs [19].

- **Programmation des Moteurs** : Le logiciel doit permettre un contrôle précis des moteurs pour assurer une navigation fluide. Cela inclut la gestion de la vitesse, la direction, et la synchronisation des roues ou des chenilles.
- **API et Bibliothèques** : Utilisation des bibliothèques de contrôle moteur adaptées à l'ESP32, telles que Servo ou DCMotor.
- **Contrôle de la Caméra ESP32-CAM** : La caméra doit capturer des images en temps réel et les envoyer à l'unité de contrôle pour traitement.
- **Configuration de la Caméra** : Paramètres de résolution, taux de capture, et compression des images.
- **Transmission des Images** : Utilisation de protocoles de communication comme Wi-Fi pour envoyer les images capturées à une station de contrôle.
- **Lecture des Données des Capteurs** : Programmation pour lire et interpréter les données des capteurs.

3.3.2 Configuration de la Caméra ESP32-CAM

La caméra ESP32-CAM (voir Figure 3.9) est un composant central du système, capable de capturer des images haute résolution et de les transmettre pour analyse en temps réel.

Paramètres et Réglages Nécessaires

- **Résolution et Fréquence d'Image** : La résolution doit être configurée pour équilibrer la qualité des images et les contraintes de bande passante. Des résolutions courantes incluent 640x480 (VGA) et 1280x720 (HD). La fréquence d'image doit être suffisante pour capturer des mouvements rapides dans les canalisations, typiquement entre 15 et 30 fps.
- **Compression d'Image** : Utilisation de formats de compression comme JPEG pour réduire la taille des fichiers sans sacrifier trop de qualité, facilitant ainsi la transmission rapide des images.
- **Balance des Blancs et Exposition** : Ajustements automatiques ou manuels de la balance des blancs et de l'exposition pour garantir des images claires dans des conditions de lumière variable.
- **Protection Contre l'Eau et les Débris** : Le boîtier doit être étanche pour protéger la caméra des conditions humides et des débris dans les canalisations. Les matériaux utilisés peuvent inclure des polymères robustes ou des alliages métalliques résistants à la corrosion.
- **Transparence Optique** : Le boîtier doit comporter une fenêtre transparente devant l'objectif de la caméra pour ne pas altérer la qualité des images. Cette fenêtre doit être résistante aux rayures et à la déformation.

3.3.3 Codage du Microcontrôleur

Le microcontrôleur ESP32-Cam qui intègre la caméra, doit être programmé pour gérer la capture des images, le traitement initial des données, et la communication avec les autres composants du système. Pour programmer l'ESP32-CAM, un module convertisseur USB vers TTL est utilisé (voir Figure 3.10). Le TTL (Transistor-Transistor Logic) est une norme de communication utilisée pour connecter des périphériques électroniques. Ce convertisseur permet de relier le microcontrôleur à un ordinateur via un port USB pour télécharger le code et effectuer le débogage. En utilisant le convertisseur USB vers TTL, nous pouvons transférer les programmes de l'environnement de développement intégré (IDE) à l'ESP32-CAM, facilitant ainsi le développement et les tests des applications embarquées.

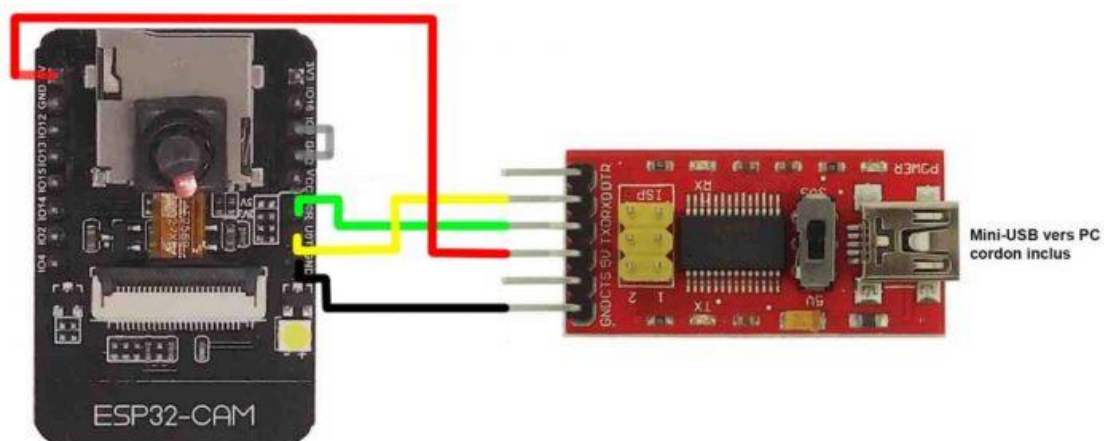


Figure 3.10 - Programmation du microcontrôleur

- **Langages et environnements de développement utilisés**

La programmation du microcontrôleur ESP32 pour notre système de détection d'obstructions s'appuie principalement sur les langages C et C++, en raison de leur efficacité et de leur performance sur les plateformes embarquées. L'environnement de développement intégré (IDE) le plus couramment utilisé est l'Arduino IDE, qui offre une interface conviviale et une vaste bibliothèque de ressources spécifiques à l'ESP32. En plus de l'IDE Arduino, l'IDE Platform IO est également utilisé pour sa flexibilité et ses capacités avancées de gestion de projets. Ces environnements permettent de gérer facilement les dépendances, de compiler le code et de flasher le micro-logiciel sur l'ESP32, tout en fournissant des outils de débogage pour identifier et résoudre les problèmes de manière efficace.

- **Fonctions principales et flux de contrôle**

Les fonctions principales du micro-logiciel de l'ESP32-CAM incluent la capture d'images, le traitement d'images, la gestion de l'éclairage, et la communication des données capturées à la station de contrôle. Le flux de contrôle commence par l'initialisation des composants, où le microcontrôleur configure la caméra, les LED d'éclairage, et les modules de communication. Ensuite, le micro-logiciel entre dans une boucle principale où il capture régulièrement des images à partir de la caméra ESP32-CAM. Ces images sont immédiatement traitées pour détecter les obstructions. Les résultats du traitement d'image sont ensuite envoyés via le module Wi-Fi à une station de contrôle. En parallèle, le micro-logiciel surveille les commandes entrantes de l'opérateur, telles que les ajustements des paramètres de la caméra ou les instructions de navigation pour la voiture-robot. Cette architecture modulaire assure une performance optimale du système tout en permettant une interaction fluide avec l'utilisateur.

Firmware de Base

- **Capture d'Image** : Le firmware doit inclure des routines pour initier et contrôler la capture d'images. Cela inclut la configuration des paramètres de la caméra et le déclenchement de la capture en réponse à des commandes ou à des événements détectés par les capteurs.
- **Traitement Préliminaire des Images** : Implémentation d'algorithmes de traitement d'image de base pour améliorer la qualité des images, comme le filtrage de bruit, l'amélioration de contraste, et la détection de contours.
- **Stockage Temporaire** : Gestion de la mémoire pour stocker temporairement les images capturées avant leur transmission, en utilisant la mémoire flash intégrée ou des modules de stockage externes.
- **Communication**
 - **Protocole Wi-Fi** : Le microcontrôleur ESP32 est équipé de capacités Wi-Fi, permettant la transmission sans fil des images et des données de capteurs vers une station de contrôle. Le firmware doit gérer la connexion au réseau, la sécurisation des communications, et la gestion des interruptions de connexion.
- **Interface avec les Capteurs**
 - **Lecture des Capteurs** : Le firmware doit inclure des routines pour lire les données des capteurs à ultrasons, en utilisant des protocoles comme Inter-Integrated Circuit (I2C) ou Serial Peripheral Interface (SPI). Ces données doivent être synchronisées avec les captures d'image pour permettre une analyse cohérente.

- **Intégration des Données** : Les données des capteurs doivent être intégrées aux images capturées, soit en ajoutant des métadonnées aux fichiers d'image, soit en envoyant les données dans un format de paquet combiné.

Mise à Jour et Maintenance du Micro-logiciel

- **Méthodes de mise à jour**

La mise à jour du micro-logiciel de l'ESP32-CAM est réalisée principalement par la méthode Over-The-Air (OTA), qui permet de flasher de nouvelles versions du FireWire à distance, sans nécessiter un accès physique au microcontrôleur. Cette méthode est particulièrement utile pour les systèmes déployés dans des environnements difficiles d'accès comme les canalisations d'eaux usées. Pour mettre en place les mises à jour OTA, le micro-logiciel initial inclut un programme de démarrage (boot loader) qui vérifie la disponibilité des mises à jour via un serveur distant chaque fois que le système démarre. Si une nouvelle version du FireWire est détectée, le boot loader télécharge et installe automatiquement la mise à jour. Ce processus assure que le système reste à jour avec les dernières améliorations et corrections de bugs, tout en minimisant les interruptions de service [29].

- **Maintenance et amélioration continue**

La maintenance du micro-logiciel implique une surveillance continue des performances du système et la mise en œuvre de mises à jour régulières pour améliorer la stabilité et l'efficacité. Les logs de performance et les diagnostics intégrés sont utilisés pour détecter et analyser les anomalies ou les dégradations de performance. Ces données permettent aux développeurs d'identifier les problèmes potentiels et de proposer des solutions avant qu'ils n'affectent le fonctionnement du système. En outre, des cycles réguliers de révision et de mise à jour du FireWire garantissent que les nouvelles fonctionnalités et les améliorations de sécurité sont intégrées de manière proactive. La maintenance continue assure non seulement la fiabilité du système à long terme, mais permet également d'adapter le micro-logiciel aux évolutions technologiques et aux besoins changeants des utilisateurs.

3.3.4 Configuration de l'Arduino IDE

Nous avons utilisé l'Arduino IDE ³ comme éditeur du code source, compilateur et éditeur de lien. Pour plus de détail sur l'installation et son premier lancement,

Installation du gestionnaire de carte ESP32 pour Arduino

Les cartes du fabricant “**Espressif Systems**” ne sont pas gérées nativement par l'environnement de développement Arduino. Nous allons donc procéder à son installation. Pour cela, dans le menu “**Fichier**”, choisir “**Préférences**” (voir Figure 3.11).

³ <https://www.arduino.cc/en/software>

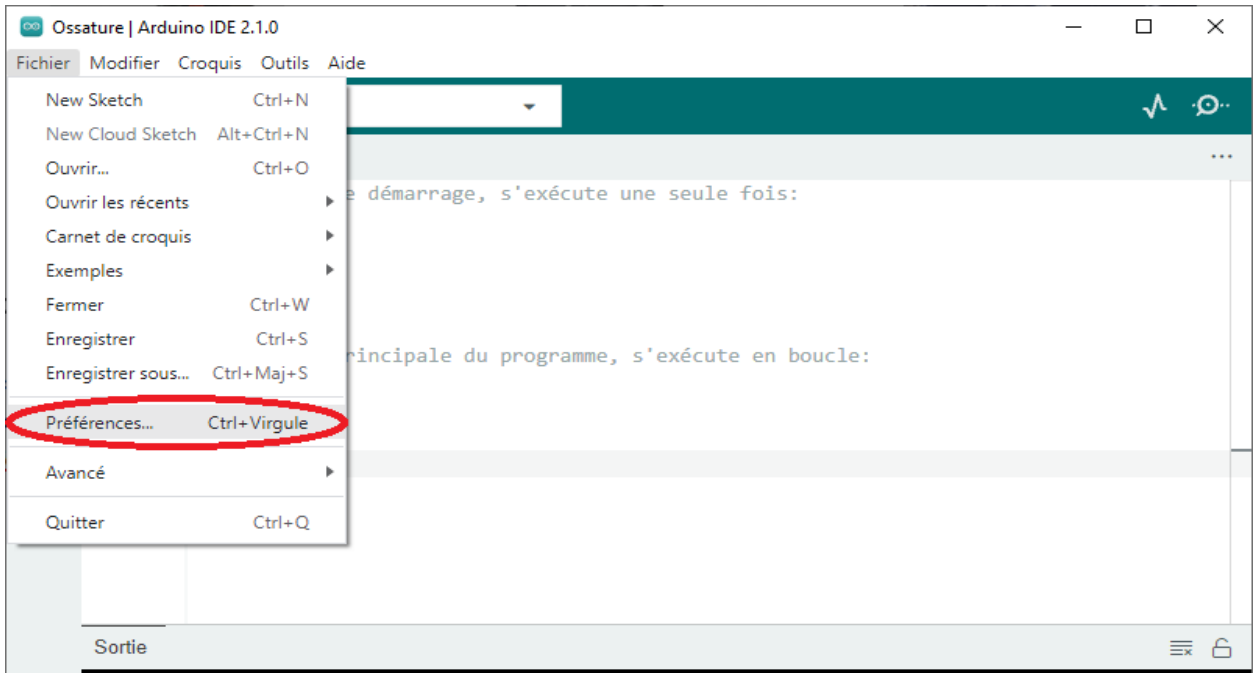


Figure 3.11 - Arduino IDE – Menu : Fichier -> Préférences

Puis dans la zone de saisie **“URL de gestionnaire de cartes supplémentaires”** (voir Figure 3.12). Renseigné l’URL des cartes de la famille ESP32 dont l’ESP32-CAM fait partie :

https://espressif.github.io/arduino-esp32/package_esp32_index.json

Remarque : Si vous avez déjà d’autres URL de cartes supplémentaires renseignées, il vous suffit de séparer les URL à l’aide d’une virgule.

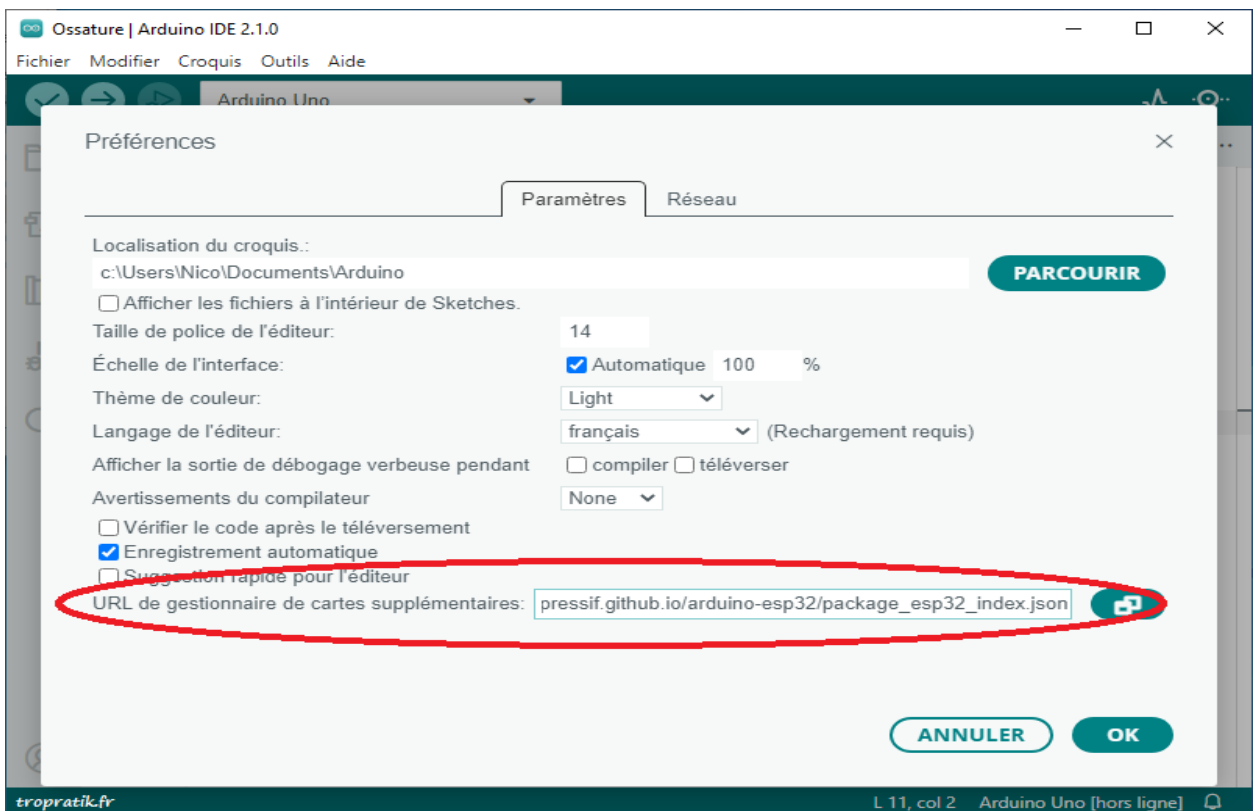


Figure 3.12 - Arduino IDE - Gestionnaire de cartes supplémentaires

Maintenant que cette URL est renseignée, son installation proprement dite se réalise dans le gestionnaire de carte. Vous pouvez y accéder de 2 façons. La façon la plus simple d'accéder au gestionnaire de carte est de cliquer sur la deuxième icône en forme de carte Arduino UNO présente sur le menu latéral à gauche de la fenêtre (voir Figure 3.13).

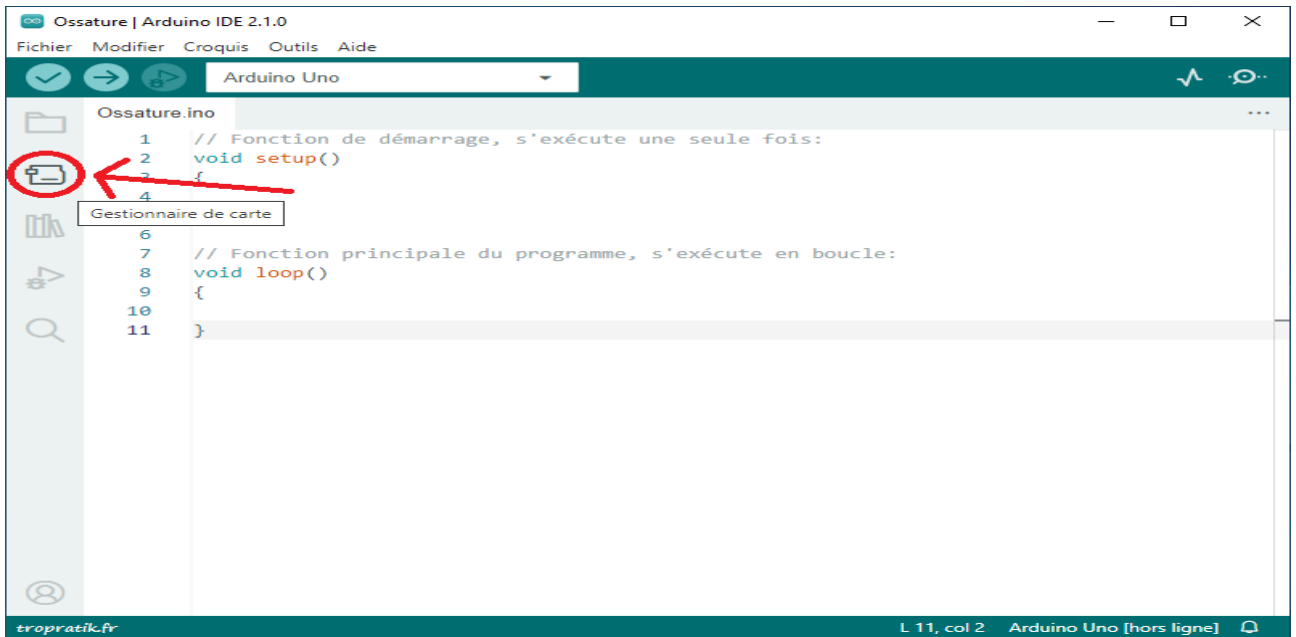


Figure 3.13 - Arduino IDE – Gestionnaire de cartes capture 1

La deuxième façon d’y accéder est le menu **“Outils”**, choisissez **“Carte”** puis cliquer sur **“Gestionnaire de carte”** (Figure 3.14).

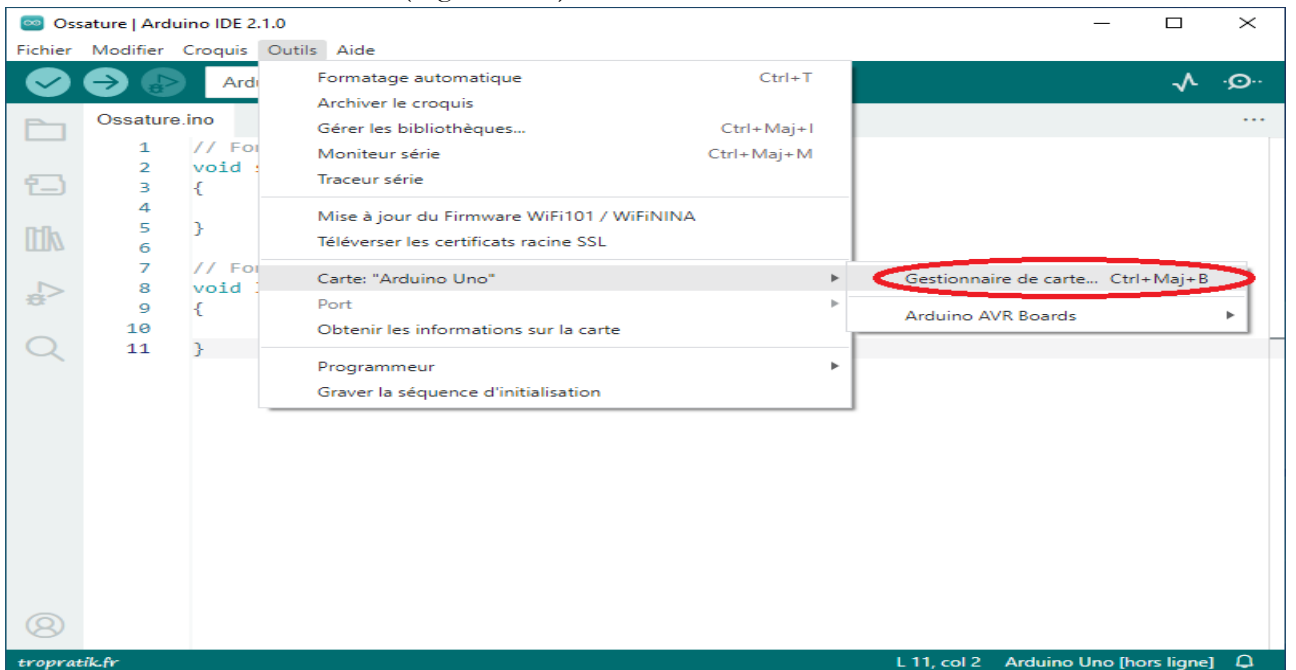


Figure 3.14 - Arduino IDE - Gestionnaires de cartes capture 2

La fenêtre des gestionnaires de cartes s’ouvre alors. En saisissant **“esp32”** dans la zone de recherche, vous ne verrez plus que la plateforme **“esp32 by Espressif Systems”** s’affiche.

Ensuite, vous pouvez réaliser sa mise en place en cliquant sur le bouton **“Installer”** correspondant (voir Figure 3.15).

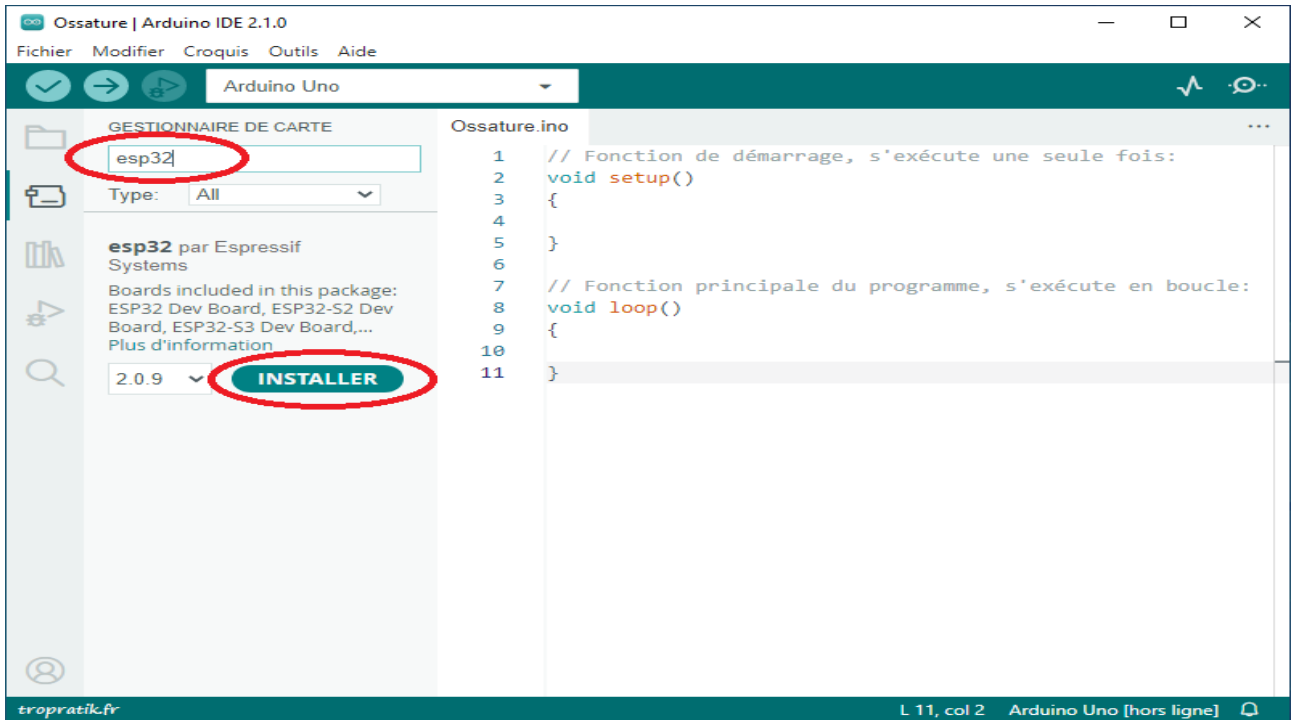


Figure 3.15 - Arduino IDE - Installation de la carte EPS32

À la fin de l’opération, le message “Plateforme installée avec succès esp32:2.x.x” sur la zone de sortie confirme que tout s’est bien déroulé. Le statut “2.x.x INSTALLED” apparaît également sous la plateforme “esp32 by Espressif Systems” dans le gestionnaire de cartes (voir Figure 3.16).

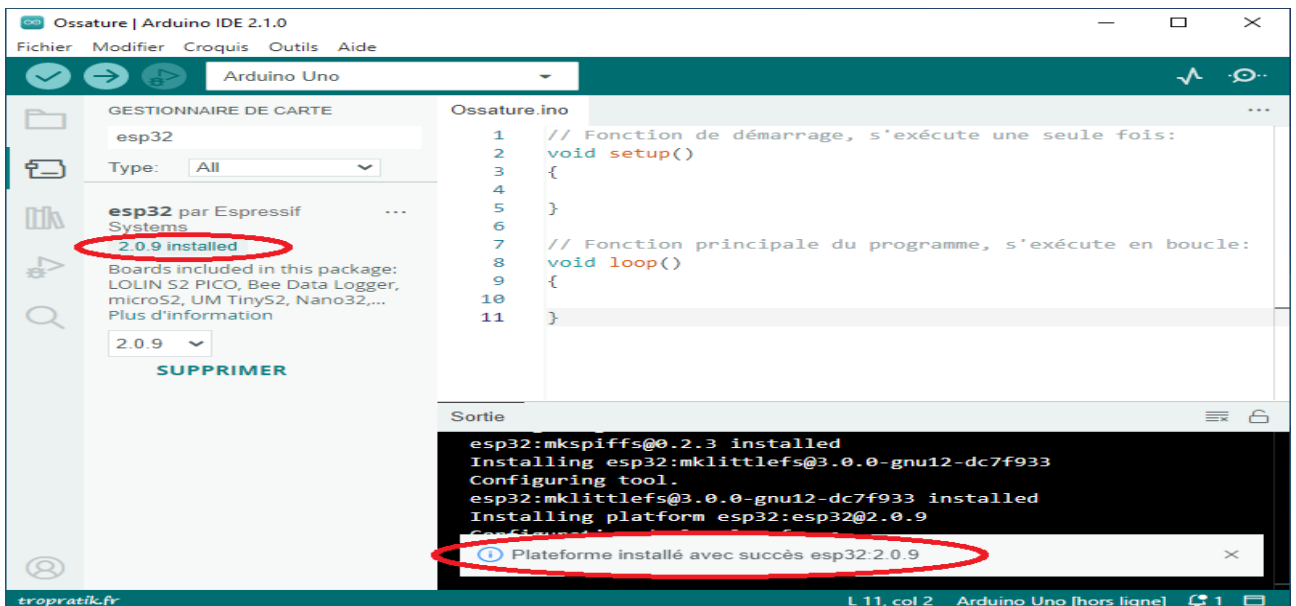


Figure 3.16 - Arduino IDE - L’installation Espressif Systems EPS32 la version 2.0.9

Sélection du module ESP32-CAM

Maintenant, nous pouvons indiquer à l’Arduino IDE que le code logiciel est destiné à un microcontrôleur de type ESP32-CAM. Dans le menu “Outils”, choisissez l’élément commençant par “Carte”, “esp32” et enfin “AI Thinker ESP32-CAM” (voir Figure 3.17)

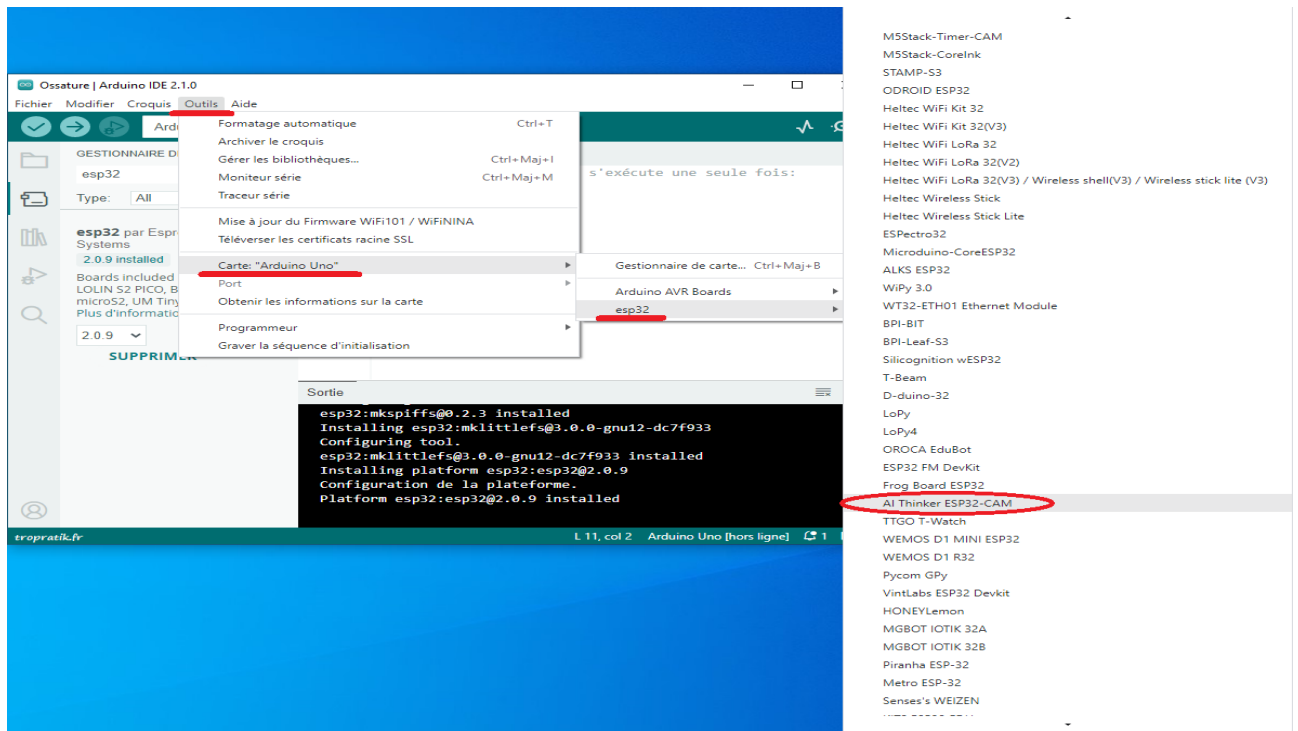


Figure 3.17 - Arduino IDE - Selection AI Thinker ESP32-CAM

Compilation du code logiciel

La compilation ne pose pas de difficulté particulière un fois le type de carte **“AI Thinker ESP32-CAM”** sélectionné.

Transfert du programme dans la carte ESP32-CAM

La carte ESP32 et donc le module ES32-CAM ont 2 différents modes de fonctionnement :

- Le mode normal d’exécution de programme
- Le mode de transfert de programme

Le basculement vers l’un ou l’autre de ces 2 modes s’effectue au démarrage :

- Si son entrée *GPIO* est reliée à la masse, l’ESP32-CAM bascule en mode transfert de programme par liaison série.
- Dans les autre cas (*GPIO* déconnectée ou alimentée en +3,3V), l’ESP32-CAM bascule en mode d’exécution du programme stocké sur sa mémoire flash.

Pour le transfert du programme que nous avons compilé, la première chose à faire est de lister les ports USB visibles par l’Arduino IDE avant même le branchement de votre convertisseur USB/TTL à notre PC.

Dans le menu **“Outils”**, cliquez sur **“Port”** puis notez les ports listés voir (Figure 3.18)

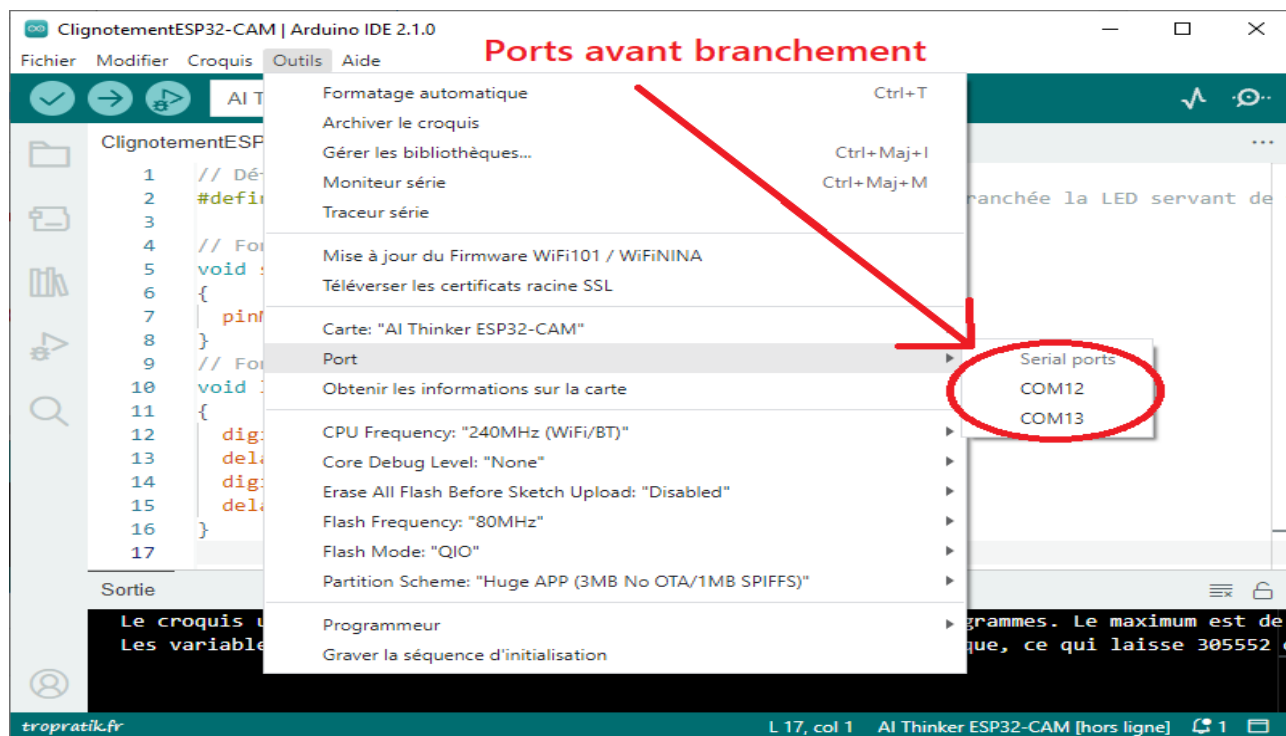


Figure 3.18 - Liste des ports USB avant le branchement de l'ESP32-CAM

Le schéma de câblage fourni au début de cet section (voir Figure 3.10) permet de basculer l'ESP32-CAM en mode de transfert de programme. Pour transférer notre programme Arduino dans l'ESP32-CAM, nous allons donc réaliser ce câblage puis brancher le convertisseur USB/TTL à notre PC :

Maintenant si nous allons dans le menu "Outils" puis cliquez sur "Port" nous voyons un nouveau port référencé. C'est celui qui correspond à notre carte de développement ESP32-CAM. Après nous sélectionnons le port pour indiquer à l'Arduino IDE d'utiliser le port COM reliée à l'USB pour le transfert du programme (voir Figure 3.19).

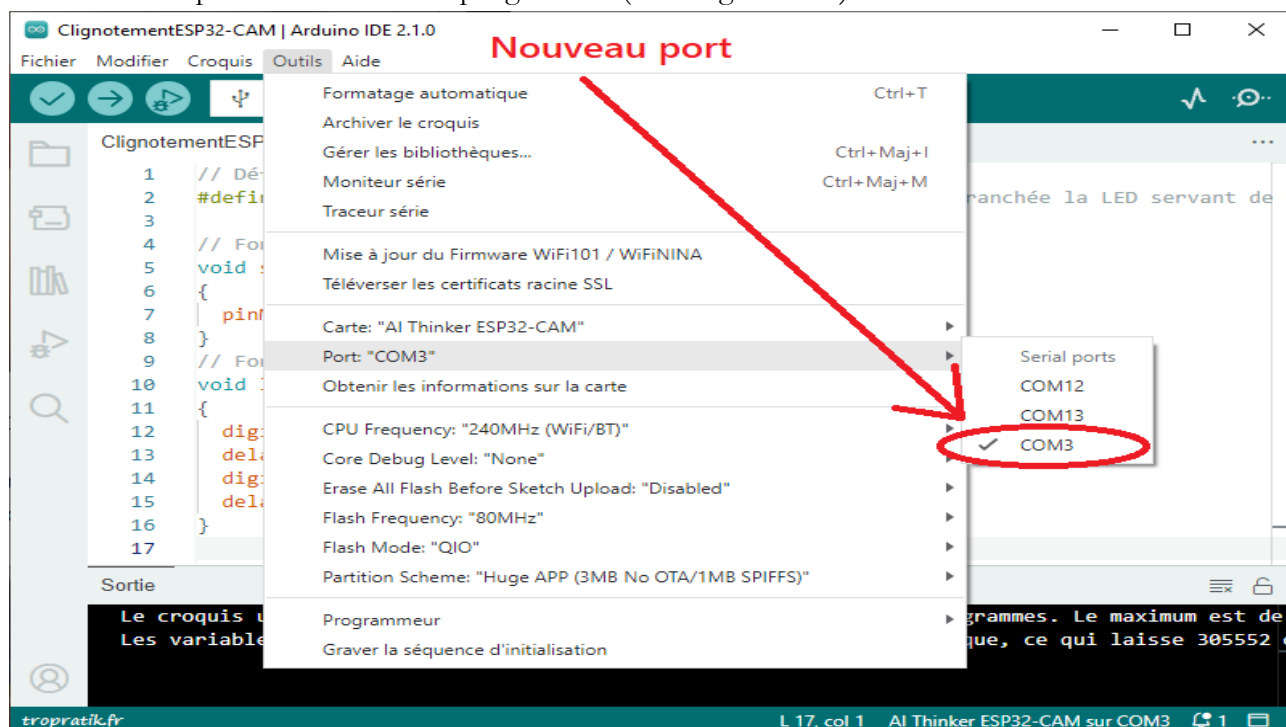


Figure 3.19 - Liste des ports USB après le branchement de l'ESP32-CAM

Afin que l'état de l'entrée *GPIO* soit bien pris en compte par l'ESP32-CAM, nous débranchons puis rebranchons l'alimentation de l'ESP32-CAM (broche 5V, fil rouge - voir Figure 3.2) pour le redémarrer. Nous pouvons maintenant lancer le transfert de ce programme dans la carte ESP32-CAM (voir Figure 3.20) en utilisant au choix :

- L'icône représentant une flèche horizontale :

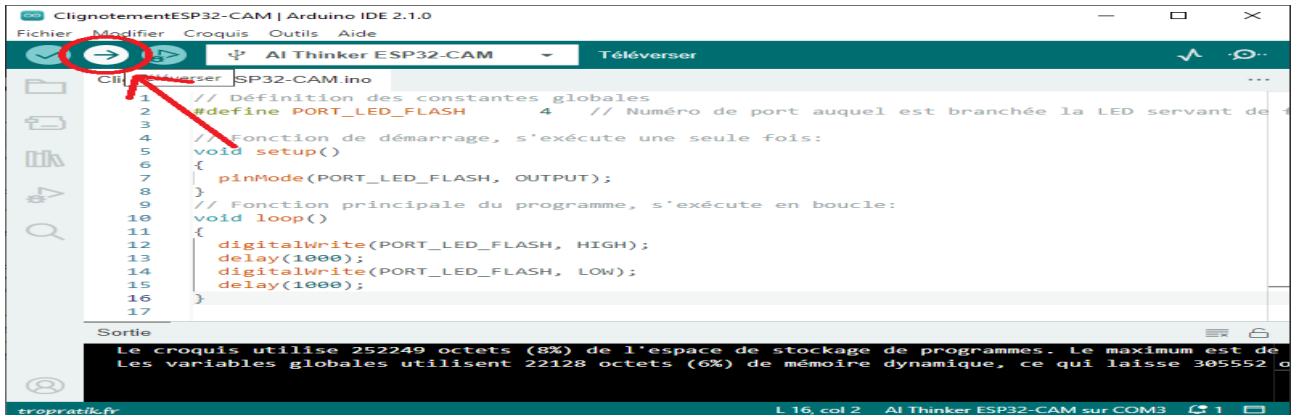


Figure 3.20 - Raccourci de l'Arduino IDE pour lancer le téléversement du programme

Le Raccourci de l'Arduino IDE pour lancer le téléversement du programme compilé sur la carte de développement est illustré dans la Figure 3.20.

Nous pouvons aussi Téléverser le code dans : Le menu “Croquis”, en cliquant sur “Téléverser” (voir Figure 3.21) :

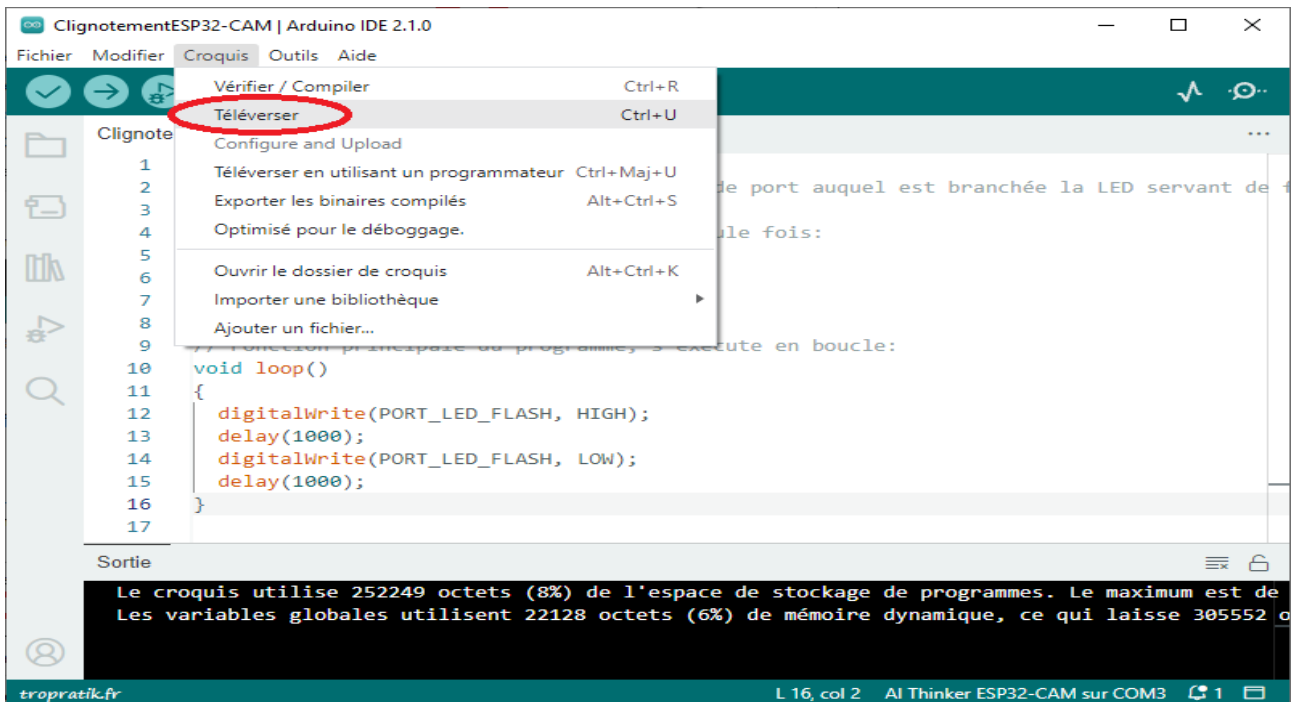


Figure 3.21 - Arduino IDE – Lancement de téléversement du programme compilé

Comme lors de l'étape de compilation, l'Arduino IDE affiche le message **“Compilation du croquis”** et une barre de progression. Une fois ce transfert terminé, si tout s'est bien déroulé, il va afficher le message **“Téléversement fait.”** (voir Figure 3.22) :

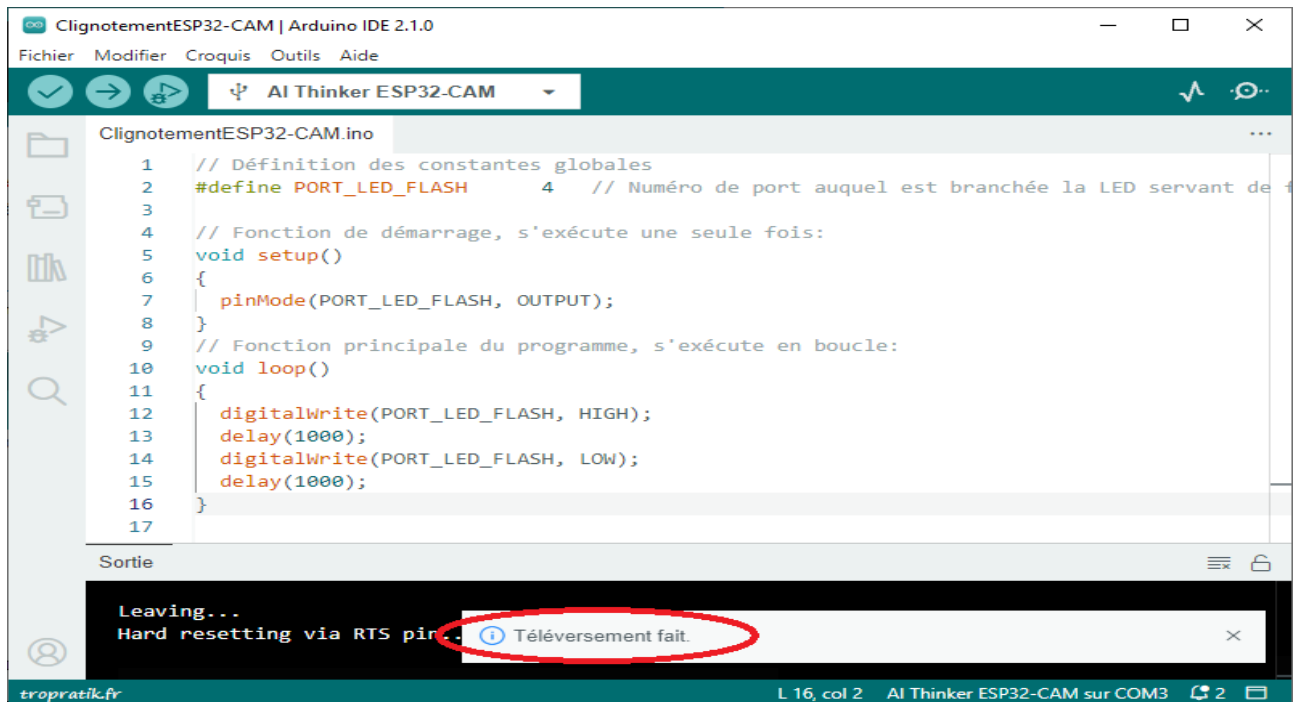


Figure 3.22 - Téléversement du programme dans la carte de développement ESP32-CAM

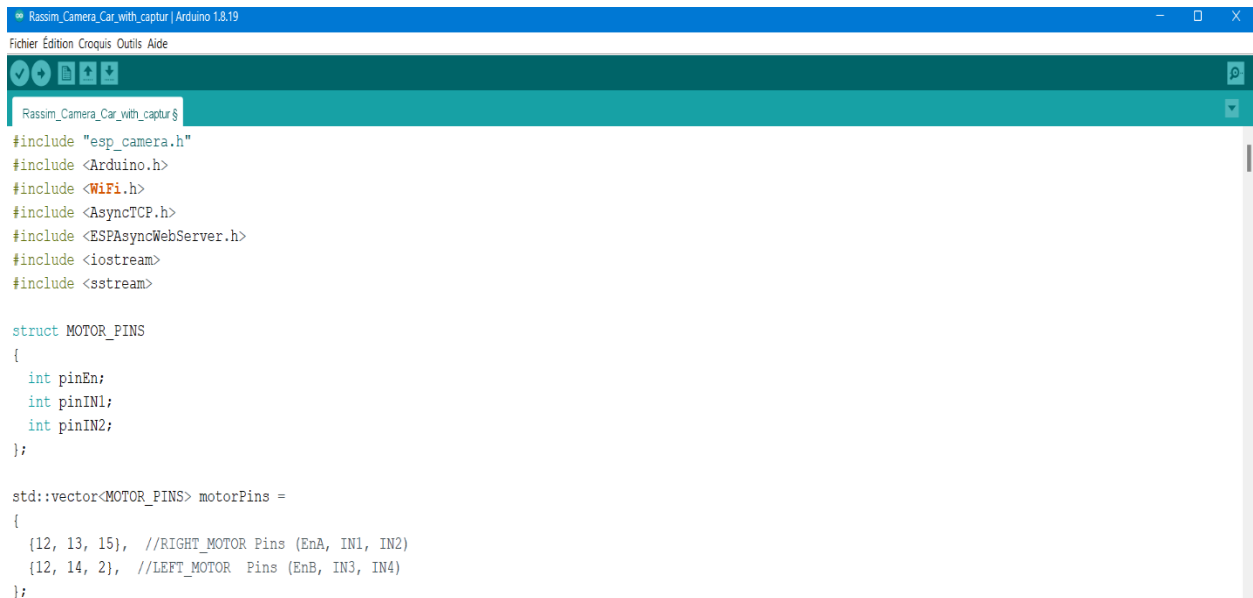
Il nous reste plus qu'à sortir l'ESP32-CAM du mode de transfert de programme. Pour cela :

1. Nous débranchons l'entrée *GPI0* de la masse (fil jaune du - voir Figure 3.10)
2. Nous débranchons puis rebranchons l'alimentation de l'ESP32-CAM (fil rouge du - voir Figure 3.10) pour le redémarrer.

3.3.6 Projet Voiture-Robot dans l'IDE Arduino

La Figure 3.23, nous voyons la configuration initiale du projet dans l'IDE Arduino. Les différentes bibliothèques nécessaires sont incluses en haut du fichier. Parmi elles, **esp_camera.h**, **Arduino.h**, **WiFi.h**, **AsyncTCP.h**, **ESPAsyncWebServer.h**, **iostream**, et **sstream**. Ces bibliothèques permettent de configurer et de gérer la caméra ESP32, la connexion WiFi, et le serveur web asynchrone, ainsi que d'utiliser des fonctionnalités de base d'Arduino et des flux d'entrée/sortie en C++.

Le code définit également une structure **MOTOR_PINS** pour stocker les broches des moteurs, ainsi qu'un vecteur **motorPins** pour les moteurs droit et gauche. Les broches sont assignées respectivement pour les moteurs droit et gauche afin de contrôler leurs mouvements. Des constantes pour les directions de mouvement (**UP**, **DOWN**, **LEFT**, **RIGHT**) et pour les lumières sont également définies, facilitant ainsi le contrôle de la voiture-robot.



```

Rassim_Camera_Car_with_captur §
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <iostream>
#include <sstream>

struct MOTOR_PINS
{
  int pinEn;
  int pinIN1;
  int pinIN2;
};

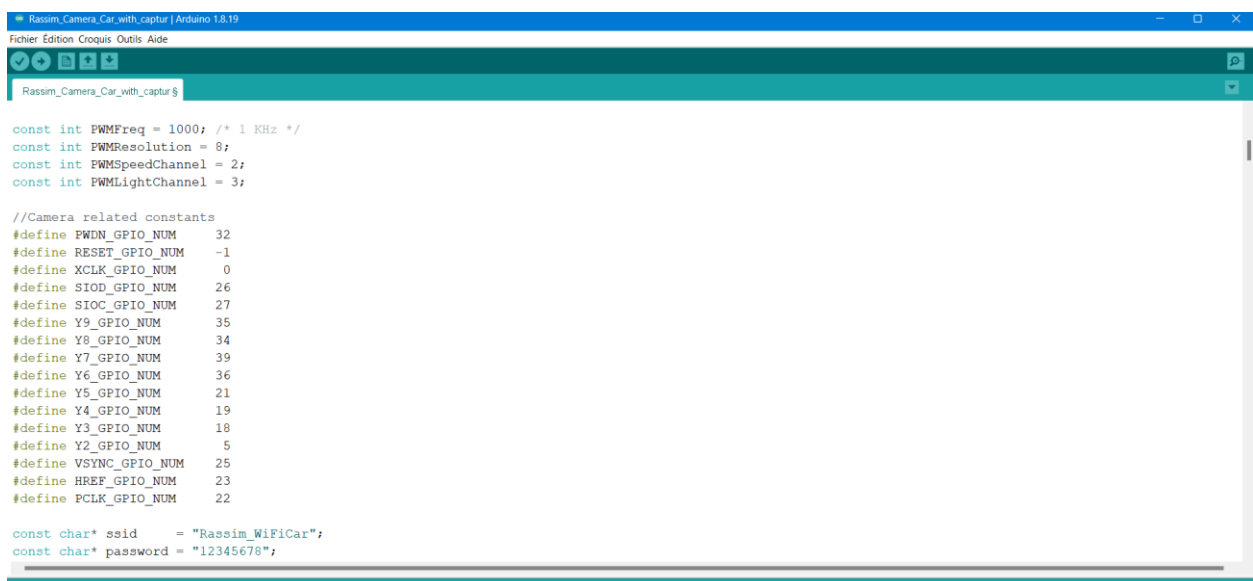
std::vector<MOTOR_PINS> motorPins =
{
  {12, 13, 15}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
  {12, 14, 2}, //LEFT_MOTOR Pins (EnB, IN3, IN4)
};

```

Figure 3.23 - Configuration initiale du projet dans l'IDE Arduino

La Figure 3.24, montre la définition des constantes utilisées pour la caméra et la configuration WiFi dans l'IDE Arduino. Les constantes pour la fréquence Pulse Width Modulation (**PWM**), la résolution, et les canaux **PWM** sont définies en haut. Ces constantes sont cruciales pour contrôler la vitesse des moteurs et l'intensité des lumières du robot.

Les constantes liées à la caméra, telles que **PWDN_GPIO_NUM**, **RESET_GPIO_NUM**, **XCLK_GPIO_NUM**, et d'autres, sont définies pour spécifier les broches **GPIO** utilisées par la caméra ESP32. Ces définitions permettent de configurer correctement la caméra pour capturer des images de haute qualité. Les identifiants **SSID** et mot de passe **WiFi** sont également définis, ce qui permet au robot de se connecter à un réseau sans fil et d'être contrôlé à distance.



```

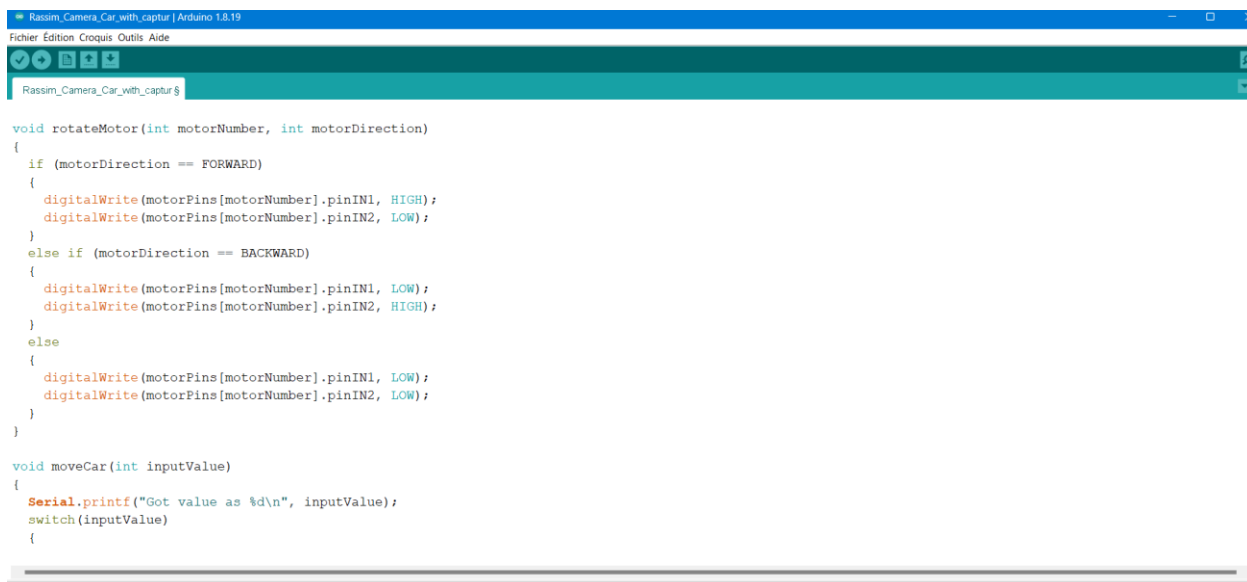
Rassim_Camera_Car_with_captur §
const int PWMFreq = 1000; /* 1 KHz */
const int PWMResolution = 8;
const int PWMspeedChannel = 2;
const int PWMLightChannel = 3;

//Camera related constants
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27
#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

const char* ssid = "Rassim_WiFiCar";
const char* password = "12345678";

```

Figure 3.24 - Définition des constantes et configuration WiFi



```

void rotateMotor(int motorNumber, int motorDirection)
{
  if (motorDirection == FORWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1, HIGH);
    digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
  else if (motorDirection == BACKWARD)
  {
    digitalWrite(motorPins[motorNumber].pinIN1, LOW);
    digitalWrite(motorPins[motorNumber].pinIN2, HIGH);
  }
  else
  {
    digitalWrite(motorPins[motorNumber].pinIN1, LOW);
    digitalWrite(motorPins[motorNumber].pinIN2, LOW);
  }
}


void moveCar(int inputValue)
{
  Serial.printf("Got value as %d\n", inputValue);
  switch(inputValue)
  {

```

Figure 3.25 - Fonction rotateMotor pour le contrôle des moteurs

Dans la Figure 3.25, nous voyons la fonction **rotateMotor** utilisée pour contrôler les moteurs du robot. Cette fonction prend en entrée le numéro du moteur et la direction du mouvement (**avant**, **arrière**, **arrêt**). Selon la direction spécifiée, les broches **IN1** et **IN2** du moteur sont configurées pour contrôler le sens de rotation. Si la direction est avant, **IN1** est mis à **HIGH** et **IN2** à **LOW**, et vice versa pour le mouvement arrière. Cette configuration permet un contrôle précis du mouvement des moteurs.

La fonction **moveCar** est également visible, prenant une valeur d'entrée pour déterminer la direction du mouvement du robot. Cette fonction utilise une déclaration switch pour appeler **rotateMotor** avec les paramètres appropriés en fonction de la direction spécifiée (**avant**, **arrière**, **gauche**, **droite**, **arrêt**). Cela permet une gestion simple et efficace des mouvements de la voiture-robot.



```

case DOWN:
  rotateMotor(RIGHT_MOTOR, BACKWARD);
  rotateMotor(LEFT_MOTOR, BACKWARD);
  break;

case LEFT:
  rotateMotor(RIGHT_MOTOR, FORWARD);
  rotateMotor(LEFT_MOTOR, BACKWARD);
  break;

case RIGHT:
  rotateMotor(RIGHT_MOTOR, BACKWARD);
  rotateMotor(LEFT_MOTOR, FORWARD);
  break;

case STOP:
  rotateMotor(RIGHT_MOTOR, STOP);
  rotateMotor(LEFT_MOTOR, STOP);
  break;

default:
  rotateMotor(RIGHT_MOTOR, STOP);
  rotateMotor(LEFT_MOTOR, STOP);
  break;
}

```

Figure 3.26 - Contrôle des directions du robot

La Figure 3.26 montre le code qui contrôle les directions de la voiture-robot en utilisant des cases de la structure switch dans la fonction **moveCar**. Chaque case (**DOWN, LEFT, RIGHT, STOP**) appelle la fonction **rotateMotor** pour les moteurs droit et gauche avec les directions appropriées. Par exemple, pour tourner à gauche, le moteur droit avance tandis que le moteur gauche recule. De même, pour tourner à droite, le moteur droit recule et le moteur gauche avance.

Cette structure de contrôle permet une navigation précise et fluide du robot dans différentes directions, en fonction des commandes reçues. La fonction **moveCar** permet ainsi d'implémenter facilement les commandes de mouvement du robot à partir des entrées reçues via l'interface Web ou les signaux de contrôle à distance.

Le code source du projet complet est disponible sur le dépôt GitHub suivant ⁴:

https://github.com/SamRepository/EPS32Cam_RobotCar

3.4 Développement de l'Application Web

Le développement de l'application Web est une étape importante pour assurer le bon fonctionnement et le contrôle de tous les composants de la voiture-robot. Cette section détaille la programmation nécessaire pour l'interface Web en utilisant HTML, Javascript et CSS.

3.4.1 Frameworks Web Utilisés

Pour le développement de l'application web de contrôle, plusieurs frameworks web ont été utilisés pour assurer une interface utilisateur réactive, intuitive et esthétique. Les frameworks utilisés sont les suivants :

HTML (HyperText Markup Language)

HTML est le langage de balisage standard pour la création de pages web. Il est utilisé pour structurer le contenu et les éléments de l'application web. Avec HTML, nous définissons les entêtes, paragraphes, liens, images, et autres éléments nécessaires à l'affichage de la page.

Les points clés de l'utilisation de HTML dans notre application web incluent :

- La structuration de la page web pour une meilleure lisibilité et organisation.
- La définition des éléments interactifs comme les formulaires de connexion et les boutons de commande.
- L'intégration avec JavaScript et CSS pour une fonctionnalité et un design amélioré.

JavaScript

JavaScript est un langage de programmation permettant de créer des interactions dynamiques et des fonctionnalités avancées sur les pages web. Il est essentiel pour la mise en œuvre des fonctionnalités interactives de l'application de contrôle.

Les points clés de l'utilisation de JavaScript dans notre application web incluent :

- La gestion des événements utilisateur comme les clics et les soumissions de formulaires.
- La mise à jour dynamique du contenu de la page sans rechargement complet grâce à l'API Fetch et AJAX.
- La communication en temps réel avec la voiture-robot via WebSockets.
- La manipulation du DOM (Document Object Model) pour des modifications en direct des éléments de la page.

⁴ https://github.com/SamRepository/EPS32Cam_RobotCar

CSS (Cascading Style Sheets)

CSS est utilisé pour décrire la présentation des documents HTML. Il permet de contrôler la mise en page, les couleurs, les polices, et d'autres aspects visuels de l'application web, assurant ainsi une expérience utilisateur cohérente et attrayante.

Les points clés de l'utilisation de CSS dans notre application web incluent :

- La définition de la mise en page et des styles pour une interface utilisateur uniforme et esthétique.
- L'utilisation de Flexbox et Grid pour une mise en page réactive qui s'adapte à différentes tailles d'écran.
- L'application de transitions et animations pour améliorer l'interactivité et l'engagement utilisateur.
- La personnalisation des thèmes et des styles pour une meilleure identification visuelle de l'application.

En intégrant ces frameworks web, nous avons pu développer une application de contrôle de la voiture-robot qui est non seulement fonctionnelle mais aussi agréable à utiliser, tout en garantissant une performance et une réactivité optimales.

3.4.2 Code source de l'Application Web

La Figure 3.26 montre la structure HTML de l'application Web, utilisée pour contrôler la voiture-robot équipée d'un ESP32-CAM. Le code inclut un style CSS intégré pour personnaliser l'apparence de l'application, et le corps de la page est centré avec une couleur de fond blanche. Une table est utilisée pour organiser les éléments, avec une image pour afficher la vidéo de la caméra (**cameraImage**) et des boutons pour contrôler le mouvement de la voiture. Les boutons envoient des commandes via la fonction **sendButtonInput**.

```
</head>
<body class="noselect" align="center" style="background-color:white">

  <!--h2 style="color: teal;text-align:center;">Wi-Fi Camera &#128663; Control</h2-->

  <table id="mainTable" style="width:400px;margin:auto;table-layout:fixed" CELSPACING=10>
  <tr>
  <img id="cameraImage" src="" style="width:400px;height:250px"></td>
  </tr>
  <tr>
  <td></td>
  <td class="button" onTouchstart='sendButtonInput("MoveCar","1")' onTouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8679;</span></td>
  <td></td>
  </tr>
  <tr>
  <td class="button" onTouchstart='sendButtonInput("MoveCar","3")' onTouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8678;</span></td>
  <td class="button"></td>
  <td class="button" onTouchstart='sendButtonInput("MoveCar","4")' onTouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8680;</span></td>
  </tr>
  <tr>
  <td></td>
  <td class="button" onTouchstart='sendButtonInput("MoveCar","2")' onTouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8681;</span></td>
  <td></td>
  </tr>
  </table>
```

Figure 3.27 - Structure HTML de l'application Web de Contrôle

```

<script>
var websocketCameraUrl = "ws:///" + window.location.hostname + "/Camera";
var websocketCarInputUrl = "ws:///" + window.location.hostname + "/CarInput";
var websocketCamera;
var websocketCarInput;

function initCameraWebSocket ()
{
  websocketCamera = new WebSocket(websocketCameraUrl);
  websocketCamera.binaryType = 'blob';
  websocketCamera.onopen = function(event) {};
  websocketCamera.onclose = function(event) {setTimeout(initCameraWebSocket, 2000)};
  websocketCamera.onmessage = function(event)
  {
    var imageId = document.getElementById("cameraImage");
    imageId.src = URL.createObjectURL(event.data);
  };
}

function initCarInputWebSocket ()
{
  websocketCarInput = new WebSocket(websocketCarInputUrl);
  websocketCarInput.onopen = function(event)
  {
    var speedButton = document.getElementById("Speed");
    speedButtonInput("Speed", speedButton.value);
  };
  websocketCarInput.onclose = function(event) {setTimeout(initCarInputWebSocket, 2000)};
  websocketCarInput.onmessage = function(event) {};
}

function initWebSocket ()
{
  initCameraWebSocket ();
  initCarInputWebSocket ();
}

function sendButtonInput(key, value)
{
  var data = key + "," + value;
  websocketCarInput.send(data);
}

window.onload = initWebSocket;
document.getElementById("mainTable").addEventListener("touchend", function(event) {
  event.preventDefault ()
});

var deg = 0;
function capturePhoto () {
  var xhr = new XMLHttpRequest();
  xhr.open('GET', "/capture", true);

```

Figure 3.28 - Initialisation et gestion des WebSockets

La Figure 3.28 montre la configuration des **WebSockets** dans le fichier JavaScript de l'application Web. Deux URLs de **WebSocket** sont définies pour la caméra et les commandes de la voiture. Les fonctions **initCameraWebSocket** et **initCarInputWebSocket** initialisent les connexions **WebSocket**, permettant la transmission des images de la caméra et des commandes de contrôle de la voiture en temps réel. Les événements **onopen**, **onclose** et **onmessage** sont définis pour gérer la connexion **WebSocket**. Les fonctions JavaScript pour initialiser et gérer les **WebSockets**. La fonction **initWebSocket** appelle les fonctions spécifiques de la caméra et des commandes de la voiture pour établir les connexions. La fonction **sendButtonInput** envoie les commandes de mouvement au serveur **WebSocket**. De plus, un événement **onload** est attaché à la fenêtre pour initialiser les **WebSockets** au chargement de la page, et un écouteur d'événements est ajouté à la table principale pour prévenir les comportements par défaut.

```

function capturePhoto() {
  var xhr = new XMLHttpRequest();
  xhr.open('GET', "/capture", true);
  xhr.send();
}
function rotatePhoto() {
  var img = document.getElementById("photo");
  deg += 90;
  if(isOdd(deg/90)){ document.getElementById("container").className = "vert"; }
  else{ document.getElementById("container").className = "hori"; }
  img.style.transform = "rotate(" + deg + "deg)";
}
function isOdd(n) { return Math.abs(n % 2) == 1; }

</script>
</body>

```

Figure 3.29 - Capturer et faire pivoter des photos

La Figure 3.29 présente les fonctions JavaScript pour capturer des photos et faire pivoter les images. La fonction **capturePhoto** utilise un objet **XMLHttpRequest** pour envoyer une requête **GET** au serveur pour capturer une photo. La fonction **rotatePhoto** permet de faire pivoter l'image capturée de 90 degrés à chaque appel, en alternant entre des classes CSS pour ajuster l'orientation de l'image. La fonction **isOdd** vérifie si le nombre de rotations est impair pour déterminer l'orientation de l'image.

```

<style>
body { text-align:center; }
.vert { margin-bottom: 20%; }
.hori{ margin-bottom: 10%; }

.arrows {
  font-size:40px;
  color:red;
}
td.button {
  background-color:black;
  border-radius:25%;
  box-shadow: 5px 5px #888888;
}
td.button:active {
  transform: translate(5px,5px);
  box-shadow: none;
}

.noselect {
  -webkit-touch-callout: none; /* iOS Safari */
  -webkit-user-select: none; /* Safari */
  -khtml-user-select: none; /* Konqueror HTML */
  -moz-user-select: none; /* Firefox */

  -ms-user-select: none; /* Internet Explorer/Edge */
  user-select: none; /* Non-prefixed version, currently
  supported by Chrome and Opera */
}

.slidecontainer {
  width: 100%;
}

.slider {
  -webkit-appearance: none;
  width: 100%;
  height: 15px;
  border-radius: 5px;
  background: #d3d3d3;
  outline: none;
  opacity: 0.7;
  -webkit-transition: .2s;
  transition: opacity .2s;
}

.slider:hover {
  opacity: 1;
}

.slider::-webkit-slider-thumb {

```

```

    -webkit-appearance: none;
    appearance: none;
    width: 25px;
    height: 25px;
    border-radius: 50%;
    background: red;
    cursor: pointer;
}
.slider::-moz-range-thumb {
    width: 25px;
    height: 25px;
    border-radius: 50%;
    background: red;
    cursor: pointer;
}
</style>

```

Figure 3.30 - Style CSS pour l'application Web

La Figure 3.30 montre les styles CSS utilisés pour l'application Web. Les styles définissent l'apparence du corps de la page, les classes `vert` et `hori` pour la rotation des images, et les styles pour les flèches et les boutons. Les boutons ont un arrière-plan noir, des coins arrondis, et une ombre portée. Lorsqu'ils sont activés, ils se déplacent légèrement pour simuler un effet de clic. La classe `noselect` empêche la sélection de texte sur les éléments, améliorant l'expérience utilisateur. Les styles CSS pour le conteneur de curseur utilisé pour ajuster des paramètres tels que la vitesse de la voiture. Le conteneur de curseur (`.slidecontainer`) et le curseur (`.slider`) sont stylisés pour correspondre à l'apparence générale de l'application. Le curseur a une apparence personnalisée avec des coins arrondis et un fond gris. Les styles pour l'état au survol (`hover`) et les pouces de curseur sont également définis, permettant une interaction utilisateur fluide et visuellement cohérente. Les éléments `::-webkit-slider-thumb` et `::-moz-range-thumb` sont personnalisés pour avoir une taille, une forme et une couleur spécifiques. Les pouces du curseur ont une apparence ronde et rouge, avec un curseur en forme de pointeur pour indiquer leur interactivité. Ces styles assurent que le curseur est non seulement fonctionnel mais aussi esthétique, aligné avec le thème global de l'application Web. La Figure 3.31 montre l'interface Web (GUI) de contrôle de voiture-robot.

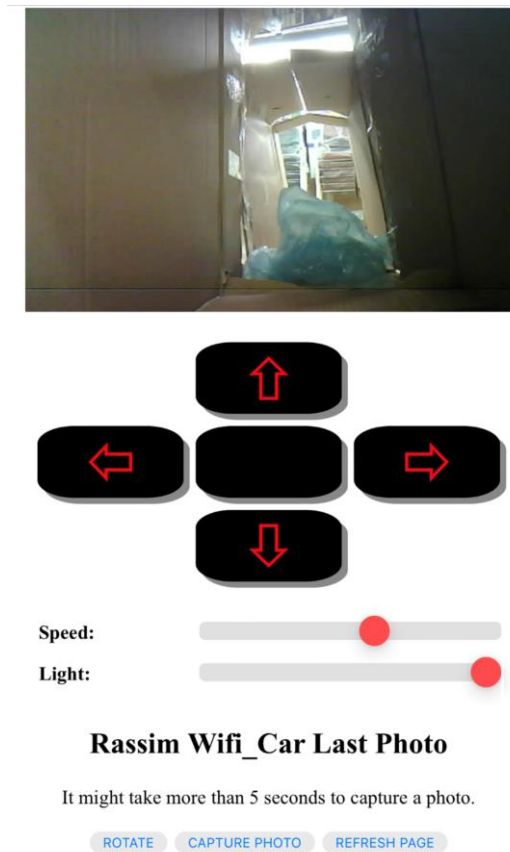


Figure 3.31 – Interface Utilisateur de l'application Web

Le code source complet de l'application Web de contrôle est disponible sur le dépôt GitHub suivant : https://github.com/SamRepository/EPS32Cam_RobotCar

3.5 Tests et Validation

Les tests et la validation sont des étapes nécessaires pour assurer que le système de détection d'obstructions fonctionne correctement et répond aux exigences spécifiées. Cette section décrit les processus de tests en environnements simulés et en conditions réelles, ainsi que les méthodes de validation des performances du système.

3.5.1 Objectifs des Tests

Les principaux objectifs des tests expérimentaux sont :

- **Évaluer les Performances du Système** : Mesurer l'efficacité du système dans des conditions réelles pour détecter les obstructions.
- **Identifier les Problèmes ou Limitations** : Détecter les éventuelles défaillances ou limitations du système afin de les corriger.
- **Recueillir des Données pour l'Amélioration** : Collecter des données qui permettront d'améliorer les algorithmes de détection et les composants matériels.

3.5.2 Matériel Utilisé

Le matériel nécessaire pour les tests comprend :

- **Système de Détection d'Obstructions** : La voiture-robot équipée de la caméra ESP32-CAM et des capteurs à ultrasons.
- **Ordinateur Portable ou Smartphone** : Pour contrôler la voiture-robot et enregistrer les données en temps réel.
- **Application Web de Contrôle et de Visualisation** : Pour surveiller les opérations et visualiser les images capturées.
- **Instruments de Mesure** : Mètre ruban, caméra externe pour des enregistrements de référence, et autres instruments de mesure pertinents.

3.5.3 Déroulement des Tests

Les tests doivent suivre une procédure bien définie pour garantir des résultats cohérents et fiables.

Préparation du Site de Test

- **Sélection du Site** : Choisir ou créer des canalisations représentatives des conditions réelles.
- **Installation des Marques de Référence** : Placer des marques ou des objets de référence à des positions connues pour faciliter l'évaluation des résultats.

Exécution des Tests

- **Introduction de la Voiture-Robot** : Insérer la voiture-robot dans la canalisation.
- **Lancement du Système de Détection** : Activer la caméra et les capteurs pour commencer la détection d'obstructions.
- **Collecte des Données** : Enregistrer les images et les mesures des capteurs en temps réel.

3.5.4 Tests des Composants Logiciels

Pour assurer le bon fonctionnement de la voiture-robot et de son système de contrôle, plusieurs scénarios de test ont été définis et exécutés. Chaque cas de test inclut les conditions initiales, les étapes de test, les résultats attendus et les critères d'acceptation pour valider la fonctionnalité testée.

Connexion de l'Utilisateur

Le premier cas de test consiste à vérifier la connexion de l'utilisateur à l'application web. L'utilisateur entre ses identifiants et clique sur "**Connexion**". Le résultat attendu est que l'utilisateur soit authentifié et accède au tableau de bord sans erreur. Ce test assure que le processus d'authentification fonctionne correctement et que les utilisateurs peuvent accéder aux fonctionnalités de l'application.

Visualisation du Flux Vidéo

Ensuite, nous testons la visualisation du flux vidéo en temps réel de l'ESP32-Cam. L'utilisateur connecté attend la "Visualisation du Flux Vidéo". Le résultat attendu est que le flux vidéo soit affiché clairement et sans interruption. Ce test vérifie la capacité du système à transmettre et afficher un flux vidéo en direct, essentiel pour la surveillance des canalisations.

Commande de Mouvement

Le troisième cas de test porte sur les commandes de mouvement de la voiture-robot. L'utilisateur envoie des commandes pour déplacer la voiture-robot (avant, arrière, gauche, droite). Le résultat attendu est que la voiture-robot exécute ces commandes correctement. Ce

test est crucial pour s'assurer que la voiture-robot peut naviguer efficacement dans les canalisations.

Capture d'Image

Ce test vérifie la fonctionnalité de capture d'image. L'utilisateur connecté clique sur "**Capter une Image**". Le résultat attendu est que l'ESP32-Cam capture une image claire et l'enregistre correctement. Ce test garantit que la caméra peut capturer des images de haute qualité, nécessaires pour l'analyse des obstructions.

Génération de Rapport

Le test suivant concerne la génération de rapports. L'utilisateur clique sur "**Générer un Rapport**". Le système doit alors générer un rapport basé sur les données collectées et le rendre accessible. Ce test vérifie que le système peut compiler et présenter les données de manière utile pour les utilisateurs.

Déconnexion de l'Utilisateur

Ce cas de test assure que l'utilisateur peut se déconnecter de l'application web. L'utilisateur clique sur "**Déconnexion**", et le résultat attendu est qu'il soit redirigé vers la page de connexion. Ce test vérifie que le processus de déconnexion fonctionne correctement, garantissant la sécurité des sessions utilisateur.

3.5.5 Tests des Composants Matériels

Test des LEDs

L'utilisateur envoie des commandes pour allumer et éteindre les LEDs. Le résultat attendu est que les LEDs répondent correctement aux commandes, ce qui est crucial pour l'inspection visuelle des canalisations.

Test des Capteurs

Les lectures des capteurs sont vérifiées via l'interface utilisateur. Les capteurs doivent fournir des données précises correspondant aux conditions réelles, assurant ainsi la détection fiable des obstructions.

Test de la Batterie

L'état de la batterie est vérifié via l'interface utilisateur. Ce test assure que les informations sur la batterie sont précises et permettent de surveiller l'autonomie de la voiture-robot.

Test des Moteurs

Chaque moteur est testé en envoyant des commandes spécifiques. Les moteurs doivent répondre correctement, assurant la capacité de la voiture-robot à se déplacer de manière fiable.

Le Tableau 3.1 résume les tests logiciels et matériels effectués pour valider les différentes fonctionnalités de la voiture-robot, garantissant ainsi que tous les composants fonctionnent correctement et que le système répond aux exigences spécifiées.

Tableau 3.1 - Récapitulatif des cas de tests logiciels et matériels

Cas de Test	Condition Initiale	Étapes de Test	Résultat Attendu	Critères d'Acceptation
Tests des Composants Logiciels				
Connexion de l'Utilisateur	Application web accessible	1. Entrer identifiants 2. Cliquer "Connexion"	Accès au tableau de bord	Tableau de bord sans erreur
Visualisation du Flux Vidéo	Utilisateur connecté	1. Attendre "Visualisation du Flux Vidéo"	Flux vidéo affiché	Flux vidéo clair et sans interruption
Commande de Mouvement	Utilisateur connecté	1. Envoyer commande de mouvement	Voiture-robot exécute la commande	Mouvement correct de la voiture-robot
Capture d'Image	Utilisateur connecté	1. Cliquer "Capturer une Image"	Image capturée et enregistrée	Image claire et disponible
Génération de Rapport	Utilisateur connecté	1. Cliquer "Générer un Rapport"	Rapport généré basé sur les données	Rapport complet et accessible
Déconnexion de l'Utilisateur	Utilisateur connecté	1. Cliquer "Déconnexion"	Déconnexion de l'application	Page de connexion affichée
Tests des Composants Matériels				
Test des LEDs	Voiture-robot allumée	1. Envoyer commande allumer LED 2. Envoyer commande éteindre LEDs	LEDs répondent aux commandes	LEDs allumées et éteintes correctement
Test des Capteurs	Voiture-robot allumée	1. Vérifier lectures des capteurs	Données des capteurs précises	Données correspondant aux conditions réelles
Test de la Batterie	Voiture-robot allumée	1. Vérifier état de la batterie	État de la batterie affiché correctement	Informations précises sur la batterie
Test des Moteurs	Voiture-robot allumée	1. Envoyer commandes pour chaque moteur	Moteurs répondent correctement	Moteurs fonctionnent sans défaut

3.5.6 Tests en Environnements Simulés

En plus des tests élémentaires, des tests préliminaires sont effectués dans des environnements simulés qui reproduisent les des canalisations réelles. Ces tests visent à identifier et corriger les problèmes avant les essais en conditions réelles. Les scénarios incluent des variations de luminosité, la présence de débris, et différentes configurations de canalisation (droites, courbes, intersections).



Figure 3.32 - Création de l'environnement de test simulé

Dans la Figure 3.32, nous voyons la construction de l'environnement de test simulé. L'environnement est constitué de grandes boîtes en carton disposées de manière à simuler une canalisation. Cette configuration permet de tester les capacités du robot à naviguer dans un espace confiné similaire à celui des vraies canalisations d'eaux usées. Les cartons sont soigneusement positionnés pour créer un tunnel continu, mettant à l'épreuve les capteurs et la maniabilité du robot dans des conditions réalistes.

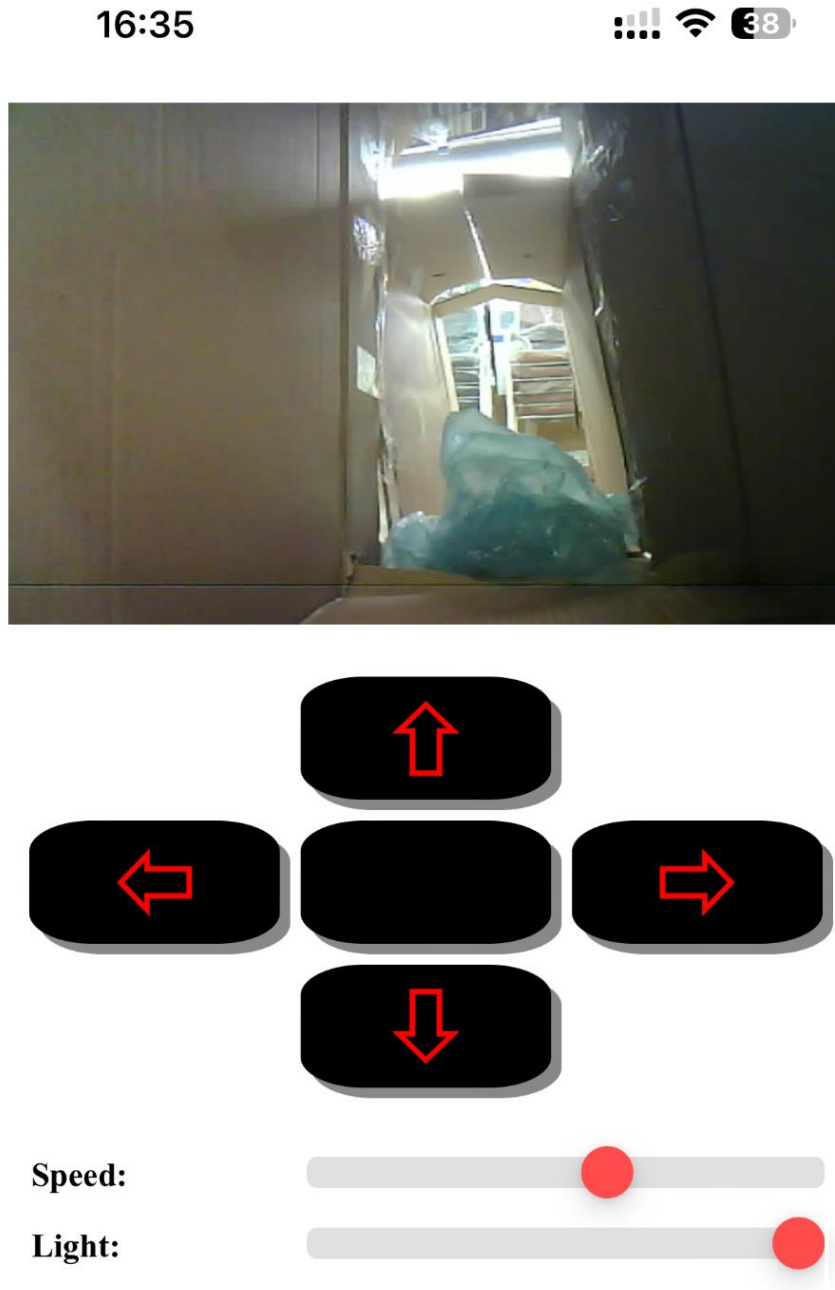
Dans la Figure 3.33 montre la voiture-robot en position dans l'environnement de test en carton. Le robot est orienté pour parcourir le tunnel créé par les cartons. Cette étape est permet de vérifier la capacité du robot à avancer, reculer et tourner correctement dans un espace confiné. L'utilisation des boîtes en carton permet de simuler les obstacles et les virages présents dans les vraies canalisations, testant ainsi l'efficacité des capteurs et du système de navigation du robot.



Figure 3.33 - Voiture-robot en situation dans l'environnement simulé

La Figure 3.34 montre l'interface de l'application web utilisée pour contrôler la voiture-robot. L'interface inclut un flux vidéo en direct de la caméra ESP32-CAM montée sur le robot, ainsi que des boutons directionnels pour contrôler les mouvements du robot (avant, arrière, gauche, droite). Des curseurs permettent de régler la vitesse et l'intensité de la lumière, et des boutons supplémentaires permettent de capturer des photos et de rafraîchir la page. Cette interface permet un contrôle intuitif et en temps réel du robot dans l'environnement simulé.

Dans la Figure 3.35, la voiture-robot est en train de naviguer dans le tunnel de carton. On peut voir la lumière LED allumée à l'avant du robot, éclairant le chemin et aidant à la détection des obstacles. Les morceaux de plastique à bulles simulant des débris ajoutent un niveau de complexité supplémentaire, permettant de tester la capacité du robot à identifier et éviter les obstacles. Ce test est essentiel pour valider les performances du robot dans des conditions proches de la réalité, assurant son efficacité dans la détection et l'évitement d'obstructions



It might take more than 5 seconds to capture a photo.

ROTATE

CAPTURE PHOTO

REFRESH PAGE

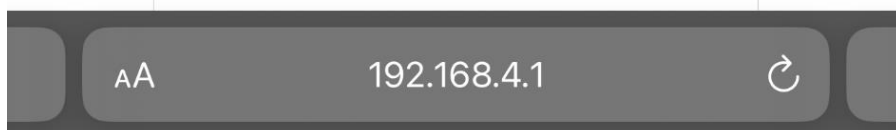


Figure 3.34 - Interface de contrôle de l'application web



Figure 3.35 - Test de navigation et détection d'obstacles

3.5.7 Tests en Conditions Réelles

Après validation en environnement simulé, le système doit être testé dans des canalisations réelles pour évaluer ses performances dans des conditions d'utilisation pratique.

- **Sélection des Sites de Test** : Choisir des canalisations représentatives des conditions d'utilisation réelles.
- **Accessibilité** : Sélectionner des canalisations accessibles pour faciliter l'introduction et la récupération de la voiture-robot.
- **Variabilité des Conditions** : Inclure des sites avec différentes configurations (droites, courbes, intersections) et des états de propreté variés (propre, avec débris, obstrué).
- **Préparation du Système** : Préparer le système pour les tests en conditions réelles.
- **Vérification Fonctionnelle** : S'assurer que tous les composants fonctionnent correctement et que les batteries sont chargées.
- **Exécution des Tests** : Lancer le système dans les canalisations et surveiller ses performances.
- **Navigation et Détection** : Observer la capacité de la voiture-robot à naviguer et à détecter les obstructions en temps réel.

- **Collecte des Données** : Enregistrer les images et les mesures des capteurs pour une analyse ultérieure.

Avant le déploiement sur le terrain, le firmware et les configurations de la caméra doivent être rigoureusement testés et débogués.

3.5.8 Analyse des Résultats

Données Collectées

Les données collectées durant les tests incluent les images capturées par la caméra ESP32-CAM, les lectures des capteurs (distances mesurées, niveaux de batterie, etc.), ainsi que les logs de commandes et réponses du système. Ces données sont analysées pour évaluer la performance et la fiabilité du système.

Méthodes d'Analyse

Les images sont analysées par un opérateur à l'aide de logiciels de traitement d'image pour détecter les obstructions et évaluer la clarté et la précision des captures. Les lectures des capteurs sont comparées aux conditions réelles pour vérifier leur exactitude. Les logs de commandes et réponses sont examinés pour s'assurer que le système répond de manière fiable et cohérente aux commandes de l'utilisateur.

Résultats des Tests

Les résultats montrent que la voiture-robot répond correctement aux commandes de mouvement, capture des images claires et précises, et transmet les données en temps réel sans interruption significative. Les tests des composants, tels que les LEDs, les capteurs, et les moteurs, confirment leur bon fonctionnement et leur intégration efficace dans le système global.

3.6 Conclusion

Ce chapitre a démontré la faisabilité et l'efficacité du système de détection d'obstructions dans les canalisations d'eaux usées. Nous avons détaillé les étapes de la réalisation, de l'assemblage à l'intégration des capteurs, ainsi que le développement du logiciel de contrôle. La réalisation de la voiture-robot a utilisé des matériaux robustes et résistants à l'eau, intégrant une caméra ESP32-CAM et des capteurs. Le prototype créé a montré une navigation fiable dans des conditions complexes. Le développement du logiciel sur le microcontrôleur ESP32-CAM et l'application web de contrôle a assuré une interaction fluide avec la voiture-robot. Les tests ont montré que le système répondait correctement aux commandes et fournissait des données précises. Les tests en environnements simulés ont permis d'identifier et de corriger des problèmes potentiels. Les scénarios de test ont validé la capacité du robot à naviguer et à détecter les obstructions efficacement. Les composants intégrés ont assuré un fonctionnement harmonieux. Des limites ont été identifiées, notamment la sensibilité de la caméra aux variations de lumière et aux conditions environnementales difficiles, ainsi que des problèmes de connectivité WiFi. Ces défis nécessitent des améliorations futures pour renforcer la robustesse du système. En conclusion, ce chapitre a illustré une démarche rigoureuse pour développer un système innovant de détection des obstructions. Les résultats sont prometteurs et ouvrent la voie à de nouvelles étapes et améliorations, confirmant la viabilité du système dans l'assainissement urbain.

Conclusion générale

CONCLUSION ET PERSPECTIVES

1. Contributions

Ce mémoire a exploré de manière exhaustive le développement d'un système de détection d'obstructions dans les canalisations d'eaux usées à l'aide d'une voiture-robot équipée d'un microcontrôleur ESP32-CAM. Les contributions significatives de ce travail peuvent être résumées en trois principaux axes abordés dans les chapitres précédents.

Chapitre 1 : Revue de la Littérature et Conception

Dans le premier chapitre, nous avons présenté une revue complète des techniques de détection d'obstructions dans les canalisations, incluant à la fois les méthodes traditionnelles et les technologies de pointe. Cette revue a permis de positionner notre travail dans le contexte de l'état de l'art et de justifier le choix des technologies utilisées. Nous avons également introduit les concepts clés liés à l'utilisation de microcontrôleur EPS32-Cam dans des applications robotiques et d'assainissement, en soulignant son pertinence pour améliorer la détection et la gestion des infrastructures d'eaux usées.

Chapitre 2 : Conception du Système

Le deuxième chapitre a détaillé la conception du système, en décrivant l'architecture générale et les composants principaux. Nous avons abordé la conception de la voiture-robot, incluant le châssis, le système de propulsion, et les systèmes d'éclairage et de communication. Une attention particulière a été portée à la conception de l'application web, qui permet le contrôle et la surveillance à distance de la voiture-robot. Les différents diagrammes UML ont été utilisés pour illustrer la structure du système et les interactions entre les composants, fournissant une base solide pour le développement et la mise en œuvre.

Chapitre 3 : Réalisation et Tests

Le troisième chapitre a décrit la mise en œuvre pratique du système, y compris l'assemblage de la voiture-robot, l'intégration des capteurs, et le développement du logiciel de contrôle. Nous avons également effectué des tests rigoureux dans des environnements simulés et réels pour valider la fonctionnalité et la fiabilité du système. Les résultats des tests ont confirmé la capacité du système à détecter efficacement les obstructions et à naviguer dans des conditions complexes, répondant ainsi aux objectifs du projet.

2. Limites et Difficultés Rencontrées

La phase de test en environnement réel a révélé plusieurs limites et difficultés.

Limites Techniques

1. **Sensibilité aux Conditions Environnementales** : La caméra ESP32-CAM, bien que performante, a montré des limites en termes de sensibilité à la lumière et aux conditions environnementales difficiles, comme les variations de luminosité et l'humidité élevée. Ces conditions ont parfois entraîné une qualité d'image réduite, rendant la détection des obstructions moins précise.

2. **Problèmes de Connectivité** : Les tests ont également mis en évidence des défis liés à la connectivité WiFi dans les environnements souterrains et confinés. La transmission des données en temps réel a parfois été interrompue, affectant la continuité de la surveillance et du contrôle à distance.

Difficultés Opérationnelles

1. **Maniabilité de la Voiture-Robot** : La navigation de la voiture-robot dans des canalisations étroites et sinueuses a présenté des défis. Les roues ont parfois eu du mal à adhérer aux surfaces glissantes, et la taille compacte de la voiture-robot a limité sa capacité à surmonter certains obstacles.
2. **Durabilité des Composants** : L'exposition prolongée à l'eau et aux produits chimiques présents dans les canalisations a mis à l'épreuve la durabilité des matériaux et des composants électroniques. Bien que des mesures de protection aient été prises, certaines pièces ont nécessité un entretien fréquent ou un remplacement.

Contraintes de Temps

La fonction de génération des rapports et les tests en environnement réel nécessitent plus de temps pour être réalisés de manière exhaustive. Malheureusement, la durée limitée du projet n'a pas permis de compléter cette fonction et mener des tests réels aussi longuement et aussi fréquemment que nécessaire pour obtenir une validation complète et robuste du système. Cette contrainte a limité notre capacité à explorer pleinement toutes les éventualités et à optimiser les performances du système en conditions réelles.

3. Perspectives et Travaux Futurs

Malgré les défis rencontrés, les résultats obtenus sont prometteurs et ouvrent la voie à plusieurs axes d'amélioration :

Améliorations Techniques

1. **Optimisation des Algorithmes de Détection** : Développer des algorithmes plus sophistiqués pour améliorer la précision de la détection des obstructions, en utilisant par exemple des techniques de traitement d'image avancées et d'apprentissage automatique avec les CNN (Convolutional Neural Networks) pour mieux analyser les images capturées.
2. **Amélioration de la Robustesse** : Renforcer la résistance des composants de la voiture-robot aux conditions environnementales difficiles, notamment en utilisant des matériaux plus robustes et en améliorant les protections contre l'humidité et la corrosion.

Extensions Fonctionnelles

1. **Intégration de Capteurs Supplémentaires** : Ajouter des capteurs supplémentaires, tels que des capteurs de gaz pour détecter les fuites de gaz nocifs, ou des capteurs de pression pour mieux évaluer l'état des canalisations.
2. **Amélioration de la Connectivité** : Explorer des solutions de connectivité alternatives, comme les réseaux maillés ou l'utilisation de technologies de communication à longue portée (LoRa) pour garantir une transmission de données fiable dans les environnements souterrains.

Déploiement à Grande Échelle

1. **Tests à Grande Échelle** : Effectuer des tests à plus grande échelle pour évaluer la performance du système dans divers environnements urbains et ruraux, et adapter le système aux différentes configurations de canalisations.
2. **Collaboration avec les Services Publics** : Travailler en collaboration avec les services publics et les gestionnaires d'infrastructures pour intégrer le système dans les programmes de maintenance et de gestion des infrastructures d'eaux usées, facilitant ainsi une adoption plus large et une optimisation des processus d'assainissement.

En conclusion, ce mémoire a non seulement démontré la faisabilité et l'efficacité d'un système de détection d'obstructions basé sur une voiture-robot équipée d'un ESP32-CAM, mais a également ouvert la voie à de nombreuses améliorations et extensions potentielles. Les contributions de ce travail fournissent une base solide pour les futures travaux et développements dans le domaine de l'inspection et de la maintenance des infrastructures d'eaux usées.

Liste des abréviations

- **4G/5G** : 4eme/5eme Generation

4G et 5G sont des générations de technologies de communication mobile.

- **API** : Application Programming Interface

Interface de programmation qui permet aux applications de communiquer entre elles.

- **BDD** : Base de Données

Désigne les bases de données utilisées pour stocker les informations.

- **CNN** : Convolutional Neural Networks

Les Convolutional Neural Network sont des réseaux de neurones spécialisés dans le traitement et la reconnaissance d'images.

- **CSS** : Cascading Style Sheets

Langage de style utilisé pour décrire la présentation des documents HTML.

- **C++** : C Plus Plus

C++ est un langage de programmation orienté objet, dérivé du langage C.

- **DC** : Direct Current

Courant continu utilisé dans les moteurs et les circuits électroniques.

- **ESP32-CAM** : Microcontrôleur ESP32 avec caméra intégrée

Microcontrôleur utilisé dans le projet pour capturer des images et les traiter.

- **FPS** : Frames Per Second

Les Frames Per Second (FPS) mesurent le nombre d'images affichées ou capturées par seconde dans une vidéo ou une animation.

- **GND** : Ground

GND est la référence de tension zéro dans un circuit électrique, servant de point de retour pour les courants électriques et stabilisant la tension.

- **GUI** : Graphical User Interface

Interface utilisateur graphique permettant une interaction utilisateur plus intuitive.

- **GPIO** : General Purpose Input/Output

Les GPIO sont des broches sur les microcontrôleurs utilisés pour lire ou contrôler des signaux numériques.

- **HTML** : Hypertext Markup Language

Langage de balisage utilisé pour créer des pages web.

- **HTTP** : Hypertext Transfer Protocol

Protocole de communication utilisé pour transmettre des pages web.

- **IDE** : Integrated Development Environment

Environnement de développement intégré utilisé pour coder et déboguer des applications.

- **IoT** : Internet of Things

Réseau d'objets physiques interconnectés capables de collecter et d'échanger des données.

- **IP** : Internet Protocol

Protocole Internet utilisé pour acheminer les paquets de données sur un réseau.

- **I2C** : Inter-Integrated Circuit

Bus de communication utilisé pour la connexion de périphériques électroniques.

- **JS** : JavaScript

Langage de programmation utilisé pour créer du contenu web dynamique.

- **JSON** : JavaScript Object Notation

Format de fichier utilisé pour l'échange de données structuré entre un serveur et une application web.

- **LoRa** : Long Range

LoRa est une technologie de modulation sans fil utilisée pour les communications longue portée.

- **LED** : Light Emitting Diode

Diodes électroluminescentes utilisées pour l'éclairage dans le projet.

- **MCA** : Maître de Conférences Agrégé

Maître de conférences agrégé, désigne un titre académique.

- **PC** : Personal Computer

Ordinateur personnel utilisé pour contrôler et surveiller le système.

- **PWM** : Pulse Width Modulation

Technique de modulation de largeur d'impulsion utilisée pour contrôler les moteurs et autres dispositifs.

- **QR** : Quick Response

Est un type de code-barres en deux dimensions (2D) qui peut être lu rapidement par des appareils numériques, comme les smartphones, pour accéder à des informations, des sites web ou des applications de manière instantanée.

- **SD** : Secure Digital

Cartes de stockage de données utilisées dans les appareils électroniques.

- **SSID** : Service Set Identifier

Le nom unique attribué à un réseau Wi-Fi, permettant aux utilisateurs de l'identifier et de s'y connecter.

- **SPI** : Serial Peripheral Interface

Interface de communication série utilisée pour connecter des microcontrôleurs et des périphériques.

- **TTL** : Transistor-Transistor Logic

TTL (Transistor-Transistor Logic) est une technologie de conception de circuits numériques basée sur l'utilisation de transistors bipolaires pour réaliser des portes logiques et d'autres fonctions logiques.

- **UART** : Universal Asynchronous Receiver-Transmitter

Récepteur-transmetteur asynchrone universel utilisé pour la communication série entre les périphériques.

- **UML** : Unified Modeling Language

Langage de modélisation unifié utilisé pour la conception de systèmes logiciels.

- **USB** : Universal Serial Bus

Bus série universel utilisé pour connecter des périphériques à un ordinateur.

- **WiFi** : Wireless Fidelity

Technologie de réseau sans fil permettant la connexion à Internet.

- **XML** : eXtensible Markup Language

Langage de balisage extensible utilisé pour définir des règles de mise en forme de documents.

Références

- [1] S. Arslan, « Assainissement des points noirs à Skikda : Un bilan mitigé », *El watan*, 30 août 2023. Consulté le: 5 mai 2024. [En ligne]. Disponible sur: <https://elwatan-dz.com/assainissement-des-points-noirs-a-skikda-un-bilan-mitige>
- [2] J. Trifunovic, « ESP32-CAM Beginner's Guide », Medium. [En ligne]. Disponible sur: <https://gerrix-dev.medium.com/esp32-cam-beginners-guide-578a23f80065>
- [3] AGM TEC, « Informations sur l'inspection visuelle des réseaux d'assainissement avec AGM TEC inspection visuelle des réseaux d'assainissement », *Caméra canalisation*. Consulté le: 16 juin 2024. [En ligne]. Disponible sur: <https://www.camera-canalisation.com/blog/informations-sur-linspection-visuelle-des-reseaux-dassainissement-avec-agm-tec>
- [4] AGM TEC, « Guide complet sur les caméras d'inspection de canalisation : Questions et réponses », AGM TEC. Consulté le: 5 mai 2024. [En ligne]. Disponible sur: <http://www.agm-tec.com/articles/349-guide-complet-sur-les-cameras-d-inspection-de-canalisation-questions-et-reponses/>
- [5] M. Beller et U. Schneider, « Pipeline inspection utilizing ultrasound technology: On the issue of resolution », *1st International Conference on Pipeline Integrity*, p. 139-148, janv. 2008.
- [6] E. Ziani, M. Bennouna, et R. Boissier, « Une technique ultrasonore pour la supervision des canaux d'irrigation ouverts », *rseau*, vol. 20, n° 3, p. 287-297, 2007, doi: 10.7202/016503ar.
- [7] Vidisco, « Using Digital Radiography (X-ray) For Pipe Inspection | Vidisco », Vidisco Portable Digital Radiography Systems. Consulté le: 4 juin 2024. [En ligne]. Disponible sur: <https://vidisco.com/digital-radiography-for-pipe-inspection/>
- [8] B. Hena, Z. Wei, C. I. Castanedo, et X. Maldague, « Deep Learning Neural Network Performance on NDT Digital X-ray Radiography Images: Analyzing the Impact of Image Quality Parameters—An Experimental Study », *Sensors*, vol. 23, n° 9, Art. n° 9, janv. 2023, doi: 10.3390/s23094324.
- [9] B. Cui et H. Wang, « Analysis and prediction of pipeline corrosion defects based on data analytics of in-line inspection », *Journal of Infrastructure Preservation and Resilience*, vol. 4, n° 1, p. 14, juin 2023, doi: 10.1186/s43065-023-00081-w.

- [10] H. N. Benfetta, B. Achour, B. S, H. N, et O. A, « TECHNIQUES FOR DETECTING THE PROGRESS OF THE LEAKS IN DAMS AND OTHER ARTIFICIAL RESERVOIRS », *Larbyss Journal*, ISSN 1112-3680, n°30, Juin 2017, vol. n°30, juin 2017.
- [11] C.-Y. Tsai, Y.-W. Lin, H.-M. Ku, et C.-Y. Lee, « Positioning System of Infrared Sensors Based on ZnO Thin Film », *Sensors*, vol. 23, n° 15, Art. n° 15, janv. 2023, doi: 10.3390/s23156818.
- [12] J. Li, G. Hu, Y. Zhou, C. Zou, W. Peng, et J. Alam SM, « Study on Temperature and Synthetic Compensation of Piezo-Resistive Differential Pressure Sensors by Coupled Simulated Annealing and Simplex Optimized Kernel Extreme Learning Machine », *Sensors*, vol. 17, n° 4, Art. n° 4, avr. 2017, doi: 10.3390/s17040894.
- [13] Q. Ma *et al.*, « Pipeline In-Line Inspection Method, Instrumentation and Data Management », *Sensors*, vol. 21, n° 11, Art. n° 11, janv. 2021, doi: 10.3390/s21113862.
- [14] L.Chastain, D.Souder, et K.Thomas, « ESP32Cam-Prise en Main », Toulouse, mai 2021. Consulté le: 9 juin 2024. [En ligne]. Disponible sur: <https://pedagogie.ac-toulouse.fr/sii/system/files/2021-12/Tuto-ESP32Cam-PriseenMain.pdf>
- [15] Yves Pelletier, « Électronique en amateur: ESP8266 / ESP32 », Électronique en amateur. Consulté le: 9 juin 2024. [En ligne]. Disponible sur: <https://electroniqueamateur.blogspot.com/p/launchpad-et-energia.html>
- [16] Sebastian, « ESP32 Camera: Hardware and GPIO Functions », DEV Community. Consulté le: 16 juin 2024. [En ligne]. Disponible sur: <https://dev.to/admantium/esp32-camera-hardware-and-gpio-functions-3i6e>
- [17] « ESP32-CAM. Broches et équipements », Domotique et Objets Connectés. Consulté le: 9 juin 2024. [En ligne]. Disponible sur: <https://projetsdiy.fr/esp32-cam-broches-et-equipements-esp-eye-ai-thinker-ttgo-t-camera-m5stack-timer-camera-domotique-et-objets-connectes-a-faire-soi-meme/>
- [18] Components101, « ESP32-CAM Camera Module », Components101. Consulté le: 9 juin 2024. [En ligne]. Disponible sur: <https://components101.com/modules/esp32-cam-camera-module>
- [19] V. O. Oner, *Developing IoT Projects with ESP32 - Second Edition: Unlock the full Potential of ESP32 in IoT development to create production-grade smart devices*, 2nd ed. edition. Birmingham Mumbai: Packt Publishing, 2023.

- [20] D. Hercog, T. Lerher, M. Truntič, et O. Težak, « Design and Implementation of ESP32-Based IoT Devices », *Sensors*, vol. 23, n° 15, Art. n° 15, janv. 2023, doi: 10.3390/s23156739.
- [21] D. N. Venu, « IOT Surveillance Robot Using ESP-32 Wi-Fi CAM & Arduino », *IJFANS*, vol. 11, 2012, [En ligne]. Disponible sur: <https://www.ijfans.org/uploads/paper/90c9affd781e762994e524485d59b585.pdf>
- [22] A. Anand, M. Cherukuri, I. Samanta, et K. Swain, *Unmanned Integrated Autonomous Vehicle: Swayambhu*. 2023. doi: 10.1109/ODICON54453.2022.10010332.
- [23] A. Kaur, A. Jadli, A. Sadhu, S. Goyal, A. Mehra, et Rahul, « Cloud Based Surveillance using ESP32 CAM », in *2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IOE)*, nov. 2021, p. 1-5. doi: 10.1109/ITSS-IOE53029.2021.9615334.
- [24] Z. S. KEBAILI Ibtissem, « Etude et réalisation d'un système de télésurveillance avec détection de présence, Basé sur une carte ESP32-CAM. », Thesis, faculté des sciences et de la technologie univ bba, 2022. Consulté le: 19 juin 2024. [En ligne]. Disponible sur: <https://dspace.univ-bba.dz:443/xmlui/handle/123456789/2513>
- [25] A. Prasetyo, J. Jamaaluddin, et I. Anshory, « PCB (Printed Circuit Board) Etching Machine Using ESP32-Camera Based Internet of Things », *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, n° 2, Art. n° 2, avr. 2023, doi: 10.12928/biste.v5i2.8132.
- [26] T. L. Nguyen *et al.*, « Autonomous control for miniaturized mobile robots in unknown pipe networks », *Front. Robot. AI*, vol. 9, nov. 2022, doi: 10.3389/frobt.2022.997415.
- [27] P. Debenest, M. Guarnieri, et S. Hirose, « PipeTron series - Robots for pipe inspection », in *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*, oct. 2014, p. 1-6. doi: 10.1109/CARPI.2014.7030052.
- [28] J. M. Aitken *et al.*, « Simultaneous Localization and Mapping for Inspection Robots in Water and Sewer Pipe Networks: A Review », *IEEE Access*, vol. 9, p. 140173-140198, 2021, doi: 10.1109/ACCESS.2021.3115981.
- [29] J. Beningo, « Exécuter des mises à jour OTA à l'aide du microcontrôleur ESP32 et de sa structure ESP-IDF », DigiKey. Consulté le: 19 juin 2024. [En ligne]. Disponible sur: <https://www.digikey.fr/fr/articles/how-to-perform-ota-updates-using-esp32-microcontroller-and-esp-idf>