

وزارة التعليم العالي والبحث العلمي

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE ABBES LAGHROUR-KHENCHELA

FACULTE DES SCIENCES ET TECHNOLOGIE

Département de Génie Indust



جامعة عباس لغرور خنشلة

كلية العلوم و التكنولوجيا

قسم : الهندسة الصناعية

## Mémoire de Master

No. Réf. : ...../...../2023

**Domaine** : Sciences Et Technologie

**Filière** : Automatique

**Spécialité** : Automatique et Informatique Industriel

### Thème

# Télé-opération contrôle d'un robot mobile basée sur un ESP32-CAM Streaming Serveur

Réalisé par :

- **Kheddouma Haroune**
- **Hassad Iheb**
- 

Dirigé par :

- **MALLEM Ali**

*Soutenu le 18/06/2023 devant la commission d'examen composée de :*

Dr.	DJAMAI Djemoui	Président
Dr.	MALLEM Ali	Rapporteur
Dr.	GUESSAM Ali	Examineur

Année universitaire 2022/2023

## Résumé

Le travail présenté dans ce mémoire consiste à élaborer un contrôle à distance d'un robot mobile ainsi mettre ce dernier suivre une couleur cible mobile. Le but est que le robot puisse se déplacer selon des directions choisies par l'opérateur (mode télé opération) ou suivre une couleur cible mobile (mode autonome). Entre autre choses nous proposons un système basé sur la carte ESP32-CAM et à l'aide d'un serveur Web qui affiche un flux vidéo de ce que le robot "voit". La mise en œuvre OpenCV est destinée à l'intégration des bibliothèques nécessaires pour effectuer le traitement sur les images et vidéos récupérées à partir de la carte ESP32-CAM. Cette dernière sera programmée à l'aide de l'IDE Arduino.

**Mots clés** – Robot mobile, ESP32-CAM, Web serveur, Streaming vidéo, OpenCV.

## ملخص

يتمثل العمل المقدم في هذا الملخص في تطوير جهاز تحكم عن بعد لروبوت متنقل بحيث يتبع لون الهدف المتحرك. الهدف هو أن الروبوت يمكنه التحرك في الاتجاهات التي يختارها المشغل (وضع التشغيل عن بعد) أو اتباع لون الهدف المتحرك (الوضع المستقل). من بين أشياء أخرى نقتراح نظامًا يعتمد على بطاقة ESP32-CAM واستخدام Webserver يعرض دفق فيديو لما «يراه» الروبوت. يهدف تنفيذ OpenCV إلى دمج المكتبات اللازمة لإجراء المعالجة على الصور ومقاطع الفيديو المسترجعة من بطاقة ESP32-CAM. سيتم برمجة هذا الأخير باستخدام Arduino IDE.

**الكلمات الدالة** - روبوت متنقل، ESP32-CAM، خادم ويب، فيديو متدفق، OpenCV.

## Summary

The work presented in this brief consists in developing a remote control of a mobile robot so that it follows a moving target color. The goal is that the robot can move in directions chosen by the operator (remote operation mode) or follow a moving target color (autonomous mode). Among other things we propose a system based on the ESP32-CAM card and using a Webserver that displays a video stream of what the robot "sees". The OpenCV implementation is intended to integrate the libraries needed to perform processing on the images and videos retrieved from the ESP32-CAM card. The latter will be programmed using the Arduino IDE.

**Tags** – Mobile robot, ESP32-CAM, Web server, Streaming video, OpenCV.

# *Dédicace*

**Ce travail est dédié :**

**A mes parents, pour tous leurs sacrifices, leur tendresse , leur soutien et leurs prière tout au long de mes études.**

**A mes chères sœurs pour leurs encouragement permanent et leur soutien morale .**

**A mes amis Radhouane ,Raouf ,Haithem ,Tarek ,Abba ,Anis , Issam , pour leur encouragement .**

**A Iheb , cher ami avant d'être binôme.**

**Merci d'être toujours la pour moi.**

**HAROUNE**

# *Dédicace*

**Je dédie ce modeste travail :**

**A mes parents pour leur patience , amour et soutien.**

**A mon frère Tadj Eddine et mes sœurs ( Amani et Baraa) qui n'ont pas cessé de m'encourager tout au long de mon parcours.**

**A Khaoula pour ses encouragements , sa confiance et surtout pour son amour.**

**A tous mes amis particulièrement ( Ramzi , Youcef , Imad et Oussama ) ma source de joie et bonheur.**

**A Haroune , cher ami avant d'être binôme.**

**IHEB**

## REMERCIEMENT

Tout d'abord nous tenons à remercier dieu qui nous a permis d'avoir la santé et la volonté de pouvoir faire ce travail, et de nous avoir permis de rencontrer des gens qui nous ont aidé dans notre parcours.

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voudrions témoigner toute notre reconnaissance. Nous voudrions tout d'abord adresser tout notre gratitude à notre encadreur de ce mémoire, MALLEM Ali, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion.

A la fin ,nous voudrions exprimer notre gratitude à tous les enseignants qui nous ont enseigné tout au long de la période universitaire.

# Sommaire

<b>INTRODUCTION GENERALE</b>	1
<b>Chapitre I (GENERALITES SUR LES ROBOTS MOBILES)</b>	
I.1. Introduction	3
I.2. Présentation du robot mobile	4
I.2.1. Classification des robots mobiles	4
I.2.1.1. Classification selon le degré d'autonomie	4
A. Robot télécommandé	5
B. Robot semi-autonome	5
C. Robot autonome :	5
I.2.1.2. Classification selon le domaine d'application	5
A. Les robots industriels et de service	5
B. Les robots militaires	5
C. Les robots de laboratoire	6
I.2.1.3. Classification selon le type de locomotion	6
A. Les robots mobiles à roues :	6
• Robots unicycle	6
• Robots tricycle	7
• Robots voiture	7
• Robots mobiles omnidirectionnels	8
B. Les robots mobiles à chenilles	9
C. Les robots mobiles marcheurs	10
D. Les robots mobiles rampants	11
I.3. Conclusion	12
<b>Chapitre II (GENERALITE SUR LE MATERIEL UTILISE)</b>	
II.1. ESP32 :	13
II.1.1. Utilité de l'ESP32	14
II.1.2. Connectivité	14
II.2. ESP32 CAM	15

II.2.1. Caractéristiques techniques de l'ESP32-CAM	16
II.3. La carte Arduino	17
II.3.1. Arduino MEGA	18
a. Caractéristiques techniques de L'Arduino Méga 2560	18
II.4. Plateforme mobile robotique	19
II.5. Afficheur Arduino	20
II.5.1.I2C	20
II.5.2.SPI	22
II.5.3.UART	23
II.6.Convertisseur de niveau logique (shifter)	24
II.7. Câble de prototypage	25
II.8. Module L298N	25
II.9..Batterie externe	26
II.10. Moteur CC	27
II.11. Conclusion	28

## **Chapitre III (CONTROLE D'UN ROBOT MOBILE BASEE SUR UN ESP32-CAM STREAMING SERVEUR)**

III.1. Introduction	29
III.2. Arduino IDE	29
III.2.1. Fenêtre générale de l'application IDE	29
III.2.2. L'exploitation de l'Arduino IDE	31
III.3. Navigateur Web (Browser)	31
III.4. Conception du contrôleur d'un robot mobile doté par ESP32-CAM	32
III.4.1. Fonctionnement de code	35
III.4.2. Testes et interprétations	38
III.5. Conclusion	41

## **Chapitre IV(CONTROLE ET SUIVI DES COULEURS PAR UN ROBOT MOBILE )**

IV.1. Introduction	42
IV.2. Généralités sur l'image	42

IV.2.1.Traitement d'image	42
IV.2.2.Seuillage d'image	43
IV.2.2.1. Seuillage globale	43
IV.2.2.2. Méthodes de seuillage	44
A. Méthode d'Otsu	44
B. Méthode Niblack	44
C. .Méthode Bernsen	45
IV.2.3.Détection des Contours	45
A. Différents types de contours	46
B. La fonction d'intensité	46
C. Le gradient d'une image	47
D. Le laplacien d'une image	47
IV.3. Langage HTML	48
IV.4. Langage Javascript	48
IV.5.La librairie OpenCV	49
IV.5.1.Fonctionnement d'OpenCV	50
IV.5.2.Traitement d'image par OenCV	50
• Le Tresholding (seuillage)	51
• Les filtres de détection de contours d'un objet sur une image	51
• Les filtres de lissage	51
• Les filtres morphologiques	52
IV.6. Design de contrôle et suivi des couleurs pour un robot mobile équipé d'ESP32-CAM basé sur OpenCV.js	52
IV.6.1. Aperçu général sur le projet	52
IV.6.2. Fonctionnement du projet	54
IV.6.2.1. Partie matérielle (Hardware)	54
IV.6.2.2. Partie logicielle (software)	55
IV.6.2.3. Les algorithmes de poursuite par le robot mobile	62
IV.7. Conclusion	65
<b>CONCLUSION GENERALE</b>	66
<b>BIBLIOGRAPHIE</b>	68

# Liste des Figures

## Chapitre I

FigureI.1: Robot mobile de type unicycle	6
Figure I.2 : Robots de type tricycle	7
Figure I.3:Robot mobile de type voiture	8
Figure I.4: Google Self-Driving Car	8
Figure I.5 : Robot de type omnidirectionnel.	9
Figure I.6 : Les robots mobiles à chenilles	9
Figure I.7: Le robot mobile marcheur	10
Figure I.8 : Le robot réalisé par Boston Dynamics Big Dog	11
Figure I.9: Le robot Octopodes T8 de Robugtix	11
FigureI.10: Robot rampant développer par l'université Carnegie Mellon	12

## Chapitre II

Figure II.1 : Carte ESP32	13
Figure II.2: Broches de l'ESP32	15
Figure II.3 :Carte ESP 32-CAM	16
Figure II.4 :Les composantes de la carte ESP32-CAM	17
Figure II.5 : La carte Arduino Méga 2560	18
Figure II.6 : La carte Arduino Méga 2560 composantes	19
Figure II.7 : Plateforme mobile robotique	20
Figure II.8:Protocole I2C	21
Figure II.9:Oled I2C	21
Figure II.10: Protocole SPI	22
Figure II.11: Oled SPI	23
Figure II.12 : Protocole UART	23
Figure II.13 :Oled UART	24

Figure II.14: Convertisseur de niveau (shifterlevel)	25
Figure II.15 : câble de prototypage	25
Figure II.16 : Carte L298N	26
Figure II.17 :Batterie 9V	26
Figure II.18: Batterie externe (Power bank)	27
Figure II.19: Moteur à courant continu	28

### **Chapitre III**

Figure III.1 :Arduino IDE	30
Figure III.2 :Exemple de Navigateur Web	32
Figure III.3 :Antenne externe	32
Figure III.4 :Plateforme mobile (Smart Kit Robot)	33
Figure III.5 :Pilote moteur L298N	33
Figure III.6 :Banque d'alimentation	34
Figure III.7 :Schéma de montage	34
Figure III.8 :Adresse IP récupérée dans moniteur série	37
Figure III.9 :La page web pour le contrôle du robot	37
Figure III.10 : Présentation du robot mobile	38
Figure III.11 : Contrôle par smart phone	38
Figure III.12 : Le robot move vers l'avant avec éloignement de la cible	39
Figure III.13 : Le robot move vers l'avant avec rapprochement de la cible	39
Figure III.14 : Le robot recule vers l'arrière	40
Figure III.15 : Le robot tourne vers la gauche	40
Figure III.16 : Le robot tourne vers la droite	41

### **Chapitre IV**

Figure IV.1 : Seuillage d'image	43
Figure IV.2 : Différents types de contours	46
Figure IV.3 : La fonction d'intensité	47

Figure IV.4 : Logo HTML	48
Figure IV.5 : Logo Javascript	49
Figure IV.6 : OpenCV	50
Figure IV.7 :Les algorithmes de seuillage par OpenCV	51
Figure IV.8 :Les filtres de détection de contours par OpenCV	51
Figure IV.9 :Les filtres de lissage par OpenCV	52
Figure IV.10 :Diagramme de Browser streaming serveur	53
Figure IV.11 :La partie hardware du projet	55
Figure IV.12 :Les deux fichiers du programme	55
Figure IV.13 :Adresse IP récupérée dans moniteur série	57
Figure IV.14 : Interface sur le web	58
Figure IV.15 :Initialisation de l'OpenCV.JS	58
Figure IV.16 :Piste de couleur RVB (niveau de grille)	59
Figure IV.17 :Sonde de couleur (probe couleur)	60
Figure IV.18 :les valeurs des couleurs (niveau de grille) pour la couleur ciblée	60
Figure IV.19 :contours de la cible	61
Figure IV.20 :suivi de la couleur	62

# INTRODUCTION GENERALE

Un robot mobile est un type de robot qui est capable de se déplacer de manière autonome ou contrôlée dans un environnement. Contrairement aux robots stationnaires qui restent fixes à un endroit, les robots mobiles sont conçus pour se déplacer et interagir avec leur environnement. Ils sont équipés de divers capteurs, tels que des capteurs de distance, des caméras, des gyroscopes, des accéléromètres, etc., qui leur permettent de percevoir leur environnement et de prendre des décisions en fonction des informations recueillies. Les robots mobiles peuvent être utilisés dans de nombreux domaines tels que l'industrie, la logistique, la surveillance, l'exploration spatiale, l'agriculture, la médecine, etc. Ils peuvent être contrôlés manuellement par un opérateur humain ou être dotés d'une certaine autonomie pour effectuer des tâches spécifiques de manière indépendante. [1]

Le contrôle des robots est une discipline de la robotique qui vise à déterminer comment un robot interagit avec son environnement et effectue des actions spécifiques. Le contrôle des robots englobe plusieurs aspects, tels que la planification des mouvements, la perception de l'environnement, la prise de décision et l'exécution des actions.

La navigation des robots mobiles fait référence à la capacité d'un robot à se déplacer de manière autonome dans son environnement en évitant les obstacles et en atteignant des destinations spécifiques. La navigation est un domaine de recherche actif dans la robotique, avec de nombreuses techniques et algorithmes développés pour améliorer la précision, l'efficacité et la sécurité des déplacements des robots dans des environnements variés.

Dans le domaine de la robotique mobile, la computer vision, ou vision par ordinateur, occupe une place essentielle en permettant aux robots de percevoir et d'analyser visuellement leur environnement. Grâce à l'utilisation de caméras et d'autres capteurs visuels, la computer vision offre aux robots mobiles la capacité de traiter les informations visuelles afin de prendre des décisions éclairées et d'interagir de manière autonome avec leur environnement.. De plus, la computer vision est essentielle dans les tâches d'interaction avec les objets. Les robots mobiles peuvent utiliser la vision par ordinateur pour détecter et reconnaître des objets

spécifiques, les saisir ou les manipuler de manière autonome. Cela ouvre des possibilités dans des domaines tels que la logistique, l'industrie, l'agriculture et l'assistance à la personne. [2]

Dans notre projet on s'intéresse à la télé-opération d'un robot mobile basant sur la carte ESP32-CAM qui nous a fourni une streaming vidéo à travers WI-FI . Le traitement d'image à l'aide des bibliothèque de l'OpenCV nous a facilité l'obtention des coordonnées de l'objet mobile afin qu'on puisse contrôler notre robot. Dans ce contexte, ce mémoire est organisé en quatre chapitres comme suit:

**Le premier chapitre :** Sera dédié aux généralités sur les robots mobiles est expliquer la classification des robots mobiles.

**Le deuxième chapitre :** Sera consacré sur généralité sur le matériel utilisés, on expliquera en détail le ESP32-CAM , Afficheur Arduino , est la carte L298N .

**Le troisième chapitre :** Sera porté sur l'exploitation de l'ARDUINO IDE ,puis on va voir le navigateur web qui permet d'extraire des ressources et des informations du Web en les téléchargeant, et on va faire la conception du contrôleur d'un robot mobile doté par ESP32-CAM . A la fin nous donnerons des test et des interprétation sur le projet .

**Le quatrième chapitre :** Nous touchons sur le traitement d'image et détection des contours et puis on a sacrifié notre travail pour faire un design de contrôle et suivi des couleurs pour un robot mobile équipé d'ESP32-CAM basé sur OpenCV.js .

Enfinement nous achevons notre projet par une conclusion générale.

# CHAPITRE I

## GENERALITES SUR LES ROBOTS MOBILES

### **I.1 Introduction**

Les robots mobiles sont des machines autonomes capables de se déplacer dans leur environnement pour accomplir diverses tâches. Ils sont équipés de systèmes de propulsion, de capteurs et d'algorithmes qui leur permettent de naviguer, de détecter des obstacles, de prendre des décisions et d'interagir avec leur environnement.

Les robots mobiles sont utilisés dans de nombreux domaines tels que l'industrie, la logistique, la santé, l'exploration spatiale, l'agriculture et la recherche. Leur utilisation permet d'automatiser des processus, d'améliorer l'efficacité, de réduire les coûts et d'accroître la sécurité dans de nombreux secteurs.

Il existe différents types de robots mobiles, chacun adapté à des tâches spécifiques. Les robots terrestres sont conçus pour se déplacer sur le sol, les robots aériens volent dans les airs, les robots aquatiques se déplacent dans l'eau, et il existe même des robots capables de naviguer dans des environnements extrêmes tels que les espaces confinés, les milieux sous-marins profonds ou les zones contaminées.

Les robots mobiles utilisent une combinaison de techniques pour se déplacer et interagir avec leur environnement. Certains robots utilisent des capteurs tels que des caméras, des lidars (systèmes de détection laser) ou des radars pour percevoir leur environnement. Ils analysent ensuite ces données pour cartographier l'environnement, détecter les obstacles et planifier des trajectoires. D'autres robots utilisent des méthodes de localisation, comme le GPS, les balises ou la vision par ordinateur, pour connaître leur position exacte dans l'environnement.

La robotique mobile est un domaine de recherche et de développement en constante évolution. Les avancées dans les domaines de l'intelligence artificielle, de la perception sensorielle, de la planification de trajectoire et de la miniaturisation des composants électroniques permettent aux robots mobiles d'accomplir des tâches de plus en plus complexes et de s'adapter à des environnements variés.

Cependant, il existe également des défis à relever dans le domaine des robots mobiles, tels que la sécurité, la fiabilité, l'autonomie énergétique et la prise en compte des aspects éthiques. La collaboration entre les chercheurs, les ingénieurs et les experts de différents domaines est essentielle pour faire progresser cette technologie et exploiter tout son potentiel

## **I.2 Présentation du robot mobile**

Un robot mobile est un système mécanique, électronique et informatique qui interagit physiquement avec son environnement afin d'accomplir un objectif donné. Cette machine est polyvalente et capable de s'ajuster à des variations de ses conditions de fonctionnement. Elle possède des capacités de perception, de prise de décision et d'action, lui permettant d'effectuer diverses tâches de manière adaptative, même en présence de situations inattendues.

Le robot mobile est équipé de moyens de déplacement dans son espace de travail. Selon son niveau d'autonomie ou d'intelligence, il peut être pourvu de capacités de perception et de raisonnement. Certains sont capables, avec un minimum d'intervention humaine, de cartographier leur environnement de travail et de planifier un itinéraire dans un environnement inconnu. Les robots mobiles les plus avancés sont principalement conçus pour être utilisés dans des environnements variables ou incertains, souvent en présence d'obstacles, nécessitant ainsi une grande adaptabilité à la tâche. [3]

### **I.2.1. Classification des robots mobiles**

La classification des robots repose sur plusieurs critères, dont les trois principaux sont le système de locomotion, le degré d'autonomie et le domaine d'application. Il est à noter que le terme "robots mobiles", bien qu'il englobe l'ensemble des robots à base mobile (par opposition aux robots manipulateurs), est généralement utilisé pour désigner spécifiquement les robots mobiles terrestres. Ce sont ces robots qui sont pris en compte dans ce document. [4]

#### **I.2.1.1 Classification selon le degré d'autonomie :**

Les robots autonomes sont des systèmes qui possèdent des capacités de prise de décision, ainsi que des moyens de collecte et de traitement de données, leur permettant d'accomplir un certain nombre de tâches avec une supervision humaine réduite, voire même en l'absence de celle-ci, dans un environnement inconnu. En fonction de ce critère, les robots mobiles peuvent être classés dans les catégories suivantes:

### ***A. Robot télécommandé :***

Les robots télécommandés sont des robots contrôlés par un opérateur, qu'il s'agisse d'une machine ou d'un être humain. L'opérateur donne des instructions précises pour chaque tâche élémentaire à effectuer, telles que avancer, reculer, tourner à droite, etc.

### ***B. Robot semi-autonome :***

Ce type de robot est capable d'accomplir un certain nombre de tâches de manière entièrement autonome, mais il peut être interrompu pour recevoir des commandes de contrôle de la part d'un opérateur.

### ***C. Robot autonome :***

L'autonomie d'un robot se définit par sa capacité à ajuster son comportement en fonction de son environnement. Cette autonomie implique que le système, dépourvu de personnel à bord, soit capable de capturer, percevoir, analyser, communiquer, planifier, prendre des décisions et agir dans le but d'atteindre les objectifs qui lui ont été assignés par un opérateur humain, à l'aide d'une interface homme-machine dédiée.

## **I.2.1.2 Classification selon le domaine d'application :**

L'un des plus grands avantages des robots mobiles réside dans leur potentiel d'application illimité. C'est pourquoi nous allons présenter ici quelques domaines d'application[5] :

### ***A. Les robots industriels et de service :***

Il y a des robots mobiles conçus spécifiquement pour des applications industrielles, qui se concentrent principalement sur le transport et la distribution dans des environnements tels que les usines, les mines, les hôpitaux et les ateliers. Ils sont également utilisés pour des tâches de nettoyage, d'entretien, de maintenance, de surveillance et de manutention. Par ailleurs, les robots de service sont conçus pour aider les personnes handicapées dans leurs déplacements, guider les personnes aveugles et piloter des voitures autonomes.

### ***B. Les robots militaires :***

Il existe de nombreuses utilisations de la robotique mobile dans le domaine militaire. Ce domaine présente l'avantage de nécessiter des spécifications précises, telles que la vitesse des véhicules, leur capacité à surmonter les obstacles (la robustesse de ces robots est essentielle) et leur réactivité rapide.

### C. Les robots de laboratoire :

Les laboratoires travaillant dans le domaine de la robotique utilisent des robots de laboratoire pour valider des travaux théoriques sur la perception ou la planification de mouvement. Tous les robots utilisés dans ces laboratoires proviennent initialement de recherches menées dans ce domaine .

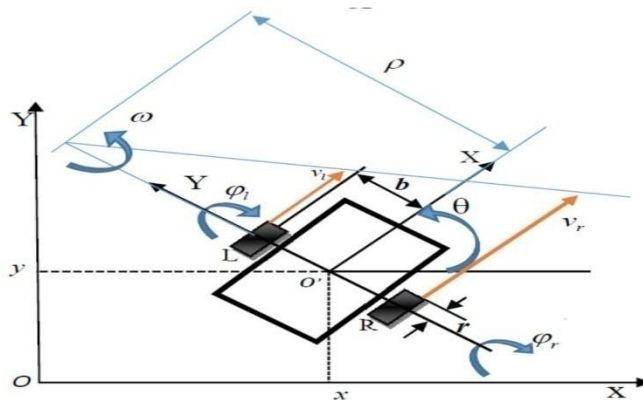
#### I.2.1.3 Classification selon le type de locomotion :

##### A. Les robots mobiles à roues :

Les robots à mobilité sur roues sont les structures mécaniques les plus couramment utilisées. Ce type de robot permet des déplacements faciles mais nécessite une surface relativement plane. On peut distinguer plusieurs catégories de robots à roues, principalement déterminées par la position et le nombre de roues utilisées. Nous mentionnerons ci-dessous les quatre catégories principales de robots à roues. [6].

- **Robots unicycle :**

Un robot de type unicycle peut être activé par une seule roue ou deux roues indépendantes. Il utilise des capteurs d'attitude (gyroscope) et de déplacement (accéléromètre) pour maintenir sa stabilité. Son centre de rotation se situe sur l'axe reliant les deux roues motrices. Le schéma des robots de type unicycle est donné en Figure 1.

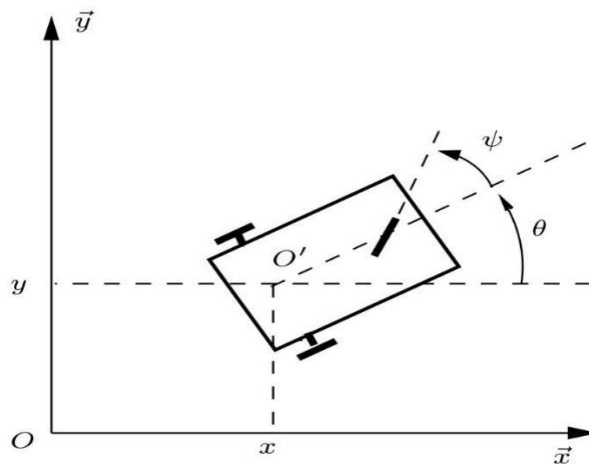


**FigureI.1:** Robot mobile de type unicycle

Un robot unicycle est un robot qui se déplace dans un plan en deux dimensions avec une certaine vitesse vers l'avant, mais sans mouvement latéral instantané, car les robots unicycle sont des systèmes non-holonomes. Il est impossible de le déplacer dans une direction perpendiculaire aux roues de locomotion.

- **Robots tricycle :**

Un robot de type tricycle se compose de deux roues fixes alignées sur un même axe et d'une roue centrale orientable positionnée le long de l'axe longitudinal (voir Figure I.2). Le mouvement du robot est déterminé par la vitesse des deux roues fixes et par l'orientation de la roue orientable.

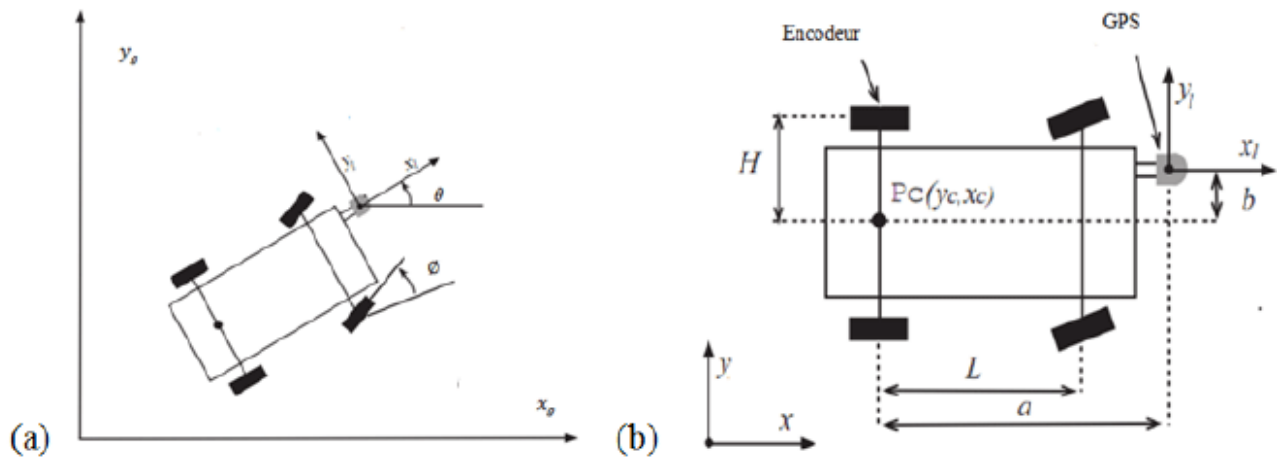


**Figure I.2 :** Robots de type tricycle

Ce robot est considéré comme non-holonyme. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes.

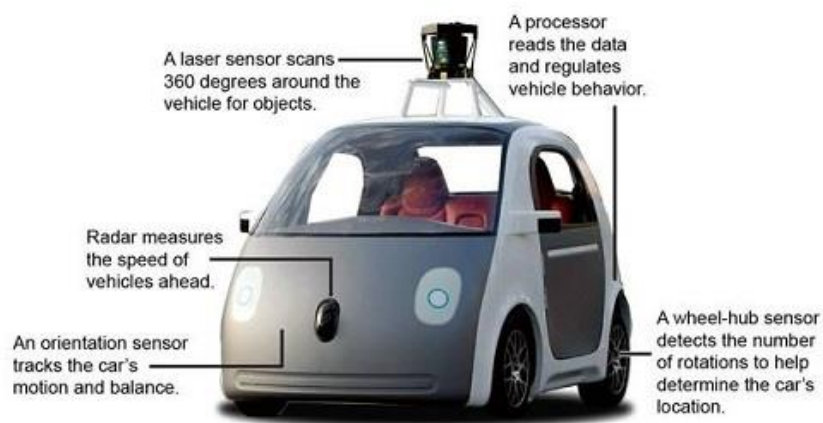
- **Robots voiture :**

Le robot de type voiture ressemble au tricycle en ce sens qu'il est composé de deux roues fixes alignées sur un même axe, ainsi que de deux roues orientables placées sur un même axe, comme illustré dans la Figure I.3. Cependant, le robot de type voiture est plus stable car il possède un point d'appui supplémentaire. Toutes les autres caractéristiques du robot de type voiture sont identiques à celles du robot tricycle. On peut d'ailleurs ramener le robot tricycle au robot de type voiture en remplaçant les deux roues avant par une seule roue située au centre de l'axe, tout en maintenant le centre de rotation inchangé.



**Figure I.3:** Robot mobile de type voiture

Lorsque l'on parle de robot, cela fait référence à une voiture autonome, c'est-à-dire sans conducteur ni télé-pilotage. L'un des projets les plus prometteurs dans ce domaine est actuellement le projet "Google Self-Driving Car Project" (voir la Figure I.4). Ces voitures autonomes sont équipées de nombreux capteurs qui leur permettent d'avoir une vision à 360 degrés, couvrant une distance équivalente à deux terrains de football dans toutes les directions. Elles sont ainsi capables de détecter différents types d'obstacles tels que des cyclistes, d'autres véhicules, des oiseaux, voire même des sachets en plastique. Ces informations contribuent à assurer une navigation sûre, sans intervention humaine.

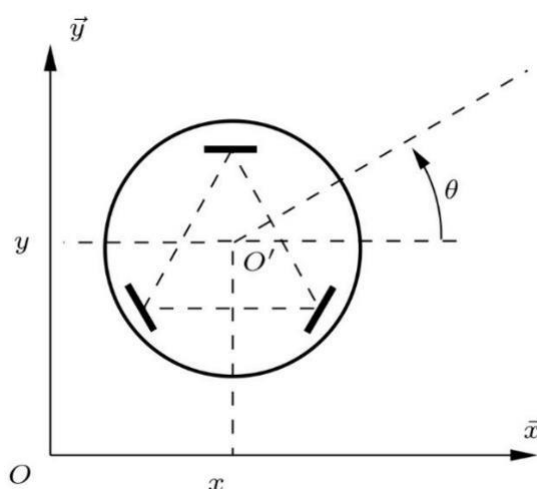


**Figure I.4:** Google Self-Driving Car

- **Robots mobiles omnidirectionnels :**

Si un robot mobile peut être contrôlé de manière indépendante en termes de vitesses de translation sur les axes  $x$  et  $y$ , ainsi que de vitesse de rotation autour de l'axe  $z$ , on le qualifie d'omnidirectionnel. Cela signifie qu'il est capable de se déplacer librement dans toutes les directions. Généralement, un robot omnidirectionnel est composé de trois roues orientables décentrées disposées en forme de triangle équilatéral, voire (Figure I.5).

L'avantage considérable du robot omnidirectionnel réside dans sa holonomie, c'est-à-dire sa capacité à se déplacer dans toutes les directions. Cependant, cette caractéristique s'accompagne d'une complexité mécanique bien plus importante.



**Figure I.5 :** Robot de type omnidirectionnel.

**B. Les robots mobiles à chenilles :**

Les robots mobiles à chenilles offrent l'avantage d'une excellente adhérence au sol et de la capacité à surmonter les obstacles. Leur utilisation est principalement axée sur les terrains accidentés ou de mauvaise qualité en termes d'adhérence, tels que la présence de boue ou d'herbe. Ils sont généralement utilisés à des fins militaires, notamment comme robots de surveillance ou de déminage voire la figure suivante (figure I.6) [7]:



**Figure I.6 :** Les robots mobiles à chenilles

***C. Les robots mobiles marcheurs :***

Les robots mobiles marcheurs sont conçus pour effectuer une variété de tâches dans des endroits difficiles, dangereux ou inaccessibles pour les humains. Leur structure avec de nombreux degrés de liberté les rapproche des robots manipulateurs. La locomotion est contrôlée en termes de coordonnées articulaires. Les méthodes de contrôle des articulations définissent le concept d'allure, qui garantit un déplacement stable de l'ensemble. Les différentes techniques étudiées se rapprochent de la marche humaine (robots de type humanoïde, voir Figure 7). Ce sont des robots bipèdes capables de se balancer de manière complètement autonome [8].



**Figure I.7:** Le robot mobile marcheur

De plus, il existe des robots mobiles qui imitent les mouvements des animaux, notamment ceux à quatre pattes tels que les chevaux (voir Figure I.8). Dans cette catégorie, il est intéressant de mentionner qu'en 2015, le robot guépard développé par Boston Dynamics a marqué l'histoire en réalisant une série de sauts. C'était la première fois qu'un robot à quatre jambes parvenait à courir et à sauter des obstacles de manière autonome.



**Figure I.8 :** Le robot réalisé par Boston Dynamics Big Dog

Il convient également de mentionner l'inspiration tirée de la locomotion des insectes, qu'ils aient quatre pattes, six pattes, voire plus, comme les araignées (voir Figure I.9). L'adaptation au support constitue un défi particulier pour les marcheurs. Il s'agit de choisir le meilleur point de contact en équilibrant à la fois l'avancement et la stabilité, en utilisant des capteurs de proximité, de contact ou de vision.

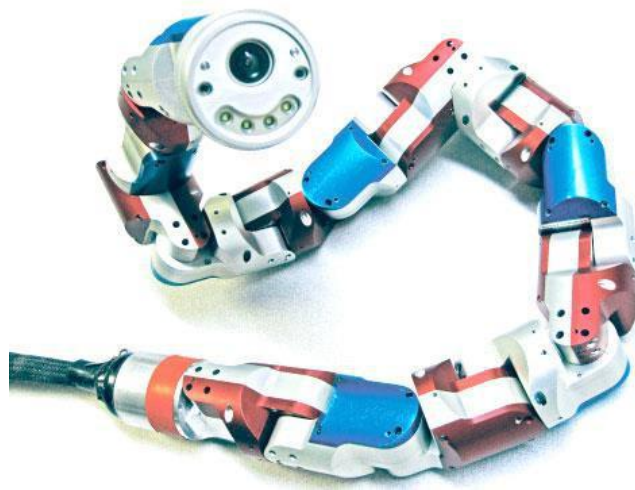


**Figure I.9:** Le robot Octopodes T8 de Robugtix

#### ***D. Les robots mobiles rampants :***

La reptation est une méthode de déplacement adaptée aux environnements en forme de tunnel, qui donne lieu à la formation de structures filiformes (voir Figure I.10). Ce système est constitué d'un ensemble de modules dotés de multiples degrés de mobilité. Les techniques utilisées dans ce contexte sont inspirées des méthodes de locomotion des serpents [9]:

- Le type scolopendre correspond à une structure articulée non extensible qui peut se déplacer selon deux axes orthogonaux.
- Le type lombric est caractérisé par trois articulations, comprenant deux rotations orthogonales et une translation dans la direction du mouvement principal.
- Le type péristaltique implique un déplacement relatif d'un module par rapport à ses voisins



**FigureI.10:** Robot rampant développé par l'université Carnegie Mellon

Le robot de la Figure I.10 a été développé afin d'explorer les tuyaux de la centrale nucléaire Zwentendorf en Autriche qui a été construite en 1970.

### **I.3 Conclusion**

Dans ce chapitre, on a abordé un aperçu général sur les robots mobiles, où on a présenté les différents types des robots ainsi leurs classes. Dans la littérature on trouve que les robots mobiles sont classifiés selon le degré d'autonomie, le domaine d'application. Pour le domaine d'application les robots sont classés comme suit :

- Les robots mobiles à roues
- les robots mobiles à chenilles
- les robots mobiles marcheurs
- Les robots mobiles rampants.

L'état de l'art qu'on a fait dans ce chapitre nous a permis de choisir le robot mobile qui respecte les contraintes demandés dans notre projet qui est le robot tricycle.

# CHAPITRE II

## GENERALITE SUR LE MATERIEL UTILISE

Dans ce chapitre nous allons présenter les composants électroniques et le matériel qui sont utilisées dans notre projet, où on va présenter chaque composant selon leur fiche technique et leur intérêt dans ce projet

Les différents matériels et composants électronique qui font intervenir dans notre projet sont donnés ci-dessous :

- La carte ESP32-CAM
- Plateforme d'un robot mobile
- Moteurs CC
- Shield L298N
- Arduino
- Batteries externes
- Afficheur OLED

### II.1. Carte ESP32

L'ESP32 est une petite carte électronique, appelée microcontrôleur, facile à prendre en main grâce à ses ressemblances avec la carte Arduino qui est bien plus répandue. L'ESP32 peut avoir différentes tailles mais la version la plus courante est la suivante:



**Figure II.1 :** Carte ESP32

ESP32 est une série de microcontrôleurs de type système sur une puce (SoC) d'EspressifSystems, basé sur l'architecture Xtensa LX6 de Tensilica (en), intégrant la gestion du Wi-Fi et du Bluetooth (jusqu'à LE 5.0 et 5.1) en mode double, et un DSP. C'est une évolution d'ESP8266. Le principal outil de développement est ESP-IDF, logiciel libre développé par Espressif, écrit en C et utilisant le système temps réel FreeRTOS. Il intègre un nombre important de bibliothèques et on retrouve dans son écosystème des bibliothèques tierce libres pour différents types de périphériques liés à l'embarqué et au temps réel [10] .

### **II.1.1 Utilité de l'ESP32**

Comme mentionné précédemment, l'ESP32 est une carte électronique permettant de réaliser des projets «IoT» (Internet of Things) assez facilement. Elle possède en effet une connectivité assez complète, que nous détaillerons dans la partie suivante. L'ESP32 est assez simple à prendre en main car elle est cousine de la carte Arduino. Il est donc possible de l'utiliser comme la carte microcontrôleur italienne, en utilisant le même langage de programmation, les mêmes modules complémentaires et surtout le même logiciel de compilation (Arduino IDE). Pas de changement fondamental d'interface pour l'utilisateur, ce qui la rend d'autant plus simple à utiliser.

Par ailleurs, on peut relever que sa taille miniature est un avantage considérable : elle mesure en effet moins de 3 cm par 5 cm, malgré les technologies qu'elle embarque. L'ESP32 est sans doute la carte microcontrôleur qui possède le meilleur compromis taille / connectivité / entrées sorties. Cela en fait un outil redoutable pour la miniaturisation des projets connectés.

Elle est aussi tout à fait adaptée à la réalisation de prototypes et non uniquement à la réalisation de projets «finis et définitifs». Elle est une référence dans l'univers Maker pour ces nombreuses raisons, il est donc très facile de trouver à l'aide de l'internet pour utiliser cette carte [11].

### **II.1.2Connectivité :**

L'objet «IoT» est très intéressant, cependant il nécessite un accès à Internet afin de pouvoir assurer une communication permanente avec l'utilisateur. L'ESP32 est en ce sens une solution intéressante car elle contient au sein de son architecture un module WiFi intégré. Dans cette mesure, et donc sans avoir besoin de monopoliser des broches en ajoutant un module WiFi externe, l'utilisateur va être capable de connecter son objet IoT à Internet grâce à trois lignes de codes (pour les réseaux domestiques classiques).

Un second atout majeur de l'ESP32 est sa puce Bluetooth intégrée, qui peut aussi être utilisée en mode «BLE» (Bluetooth LowEnergy) consommant moins d'énergie. La consommation d'énergie est en effet un critère important dans la conception d'objets IoT. Ils requièrent une connexion quasi permanente à Internet et doivent donc rester allumés ou en état de veille à chaque instant afin de pouvoir recevoir les instructions de l'utilisateur. Pour des raisons écologiques et économiques il est ainsi fondamental de réaliser un objet qui consomme le moins d'énergie que possible en état de veille, d'où l'utilité de la technologie BLE notamment.

En terme des entrées et sorties, l'ESP32 est semblable aux cartes Arduino et aux cartes Raspberry. On relève en effet l'utilisation des ports GPIO (General Purpose Input/Output) qui sont les broches permettant de servir soit d'entrée, soit de sortie selon le programme téléversé[12].

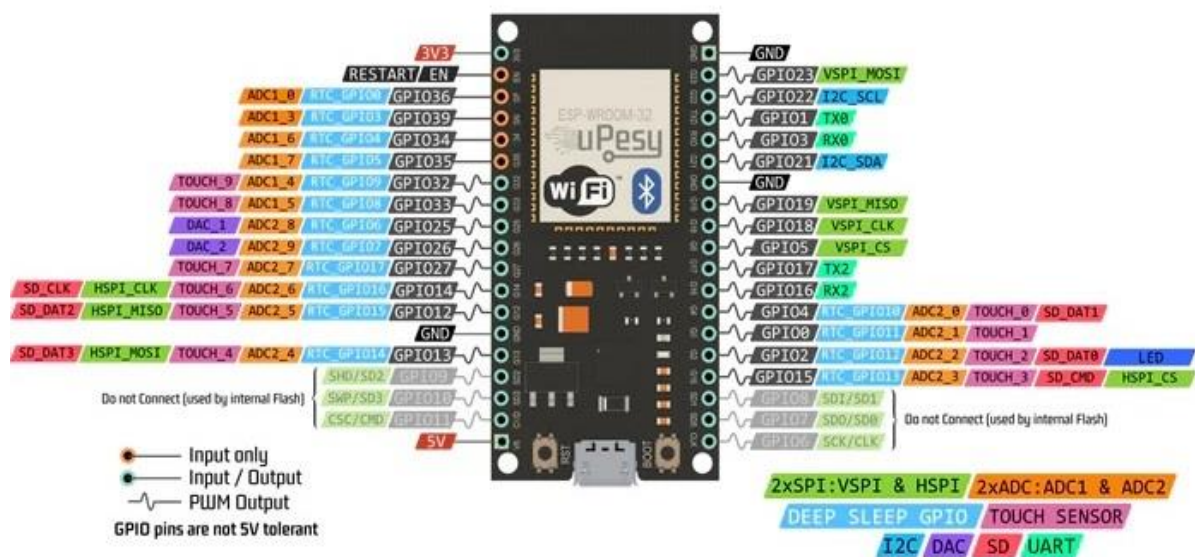


Figure II.2: Broches de l'ESP32

## II.2. ESP32-CAM :

ESP32-CAM est une carte de développement à faible coût basée sur l'ESP32 avec une caméra compacte. C'est une solution idéale pour les projets d'application, de prototypage de systèmes de domotique ou de IoT (Internet of Things). La carte intègre le Wi-Fi, le Bluetooth traditionnel et le BLE (Bluetooth LowEnergy), avec deux processeurs LX6 32 bits hautes performances. Il adopte une architecture pipeline à 7 phases, un capteur sur puce, un capteur Hall, un capteur de température, etc., et les plages de modulation de fréquence

principales de 80 MHz à 240 MHz, Entièrement compatible avec les normes Wi-Fi 802.11b/g/n/e/i et Bluetooth 4.2, il peut être utilisé en mode maître pour construire un contrôleur de réseau autonome, ou comme esclave relié à d'autres MCU hôtes pour augmenter les performances. Il convient aux appareils électroménagers intelligents, au contrôle sans fil industriel, à la surveillance sans fil, à l'identification QR sans fil, à la reconnaissance faciale, aux signaux GPS sans fil et à d'autres applications IoT. C'est une solution idéale pour les applications IoT [13].



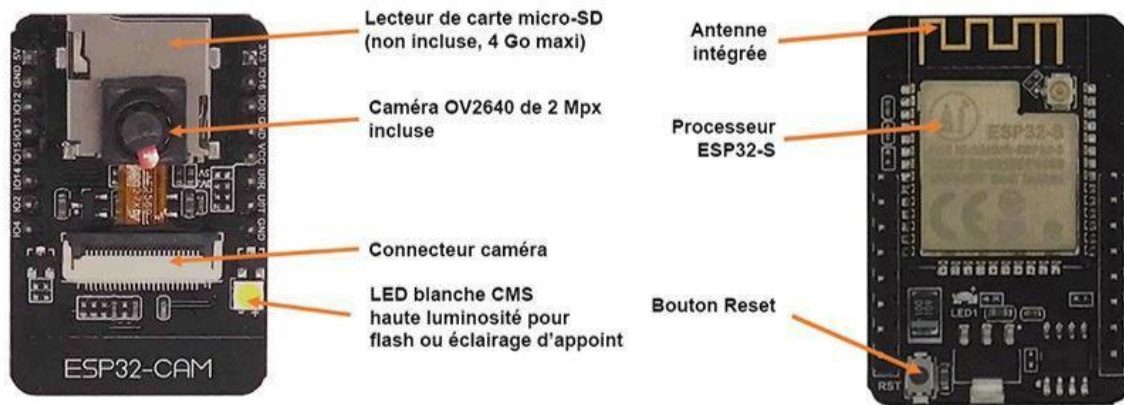
**Figure II.3 :** Carte ESP 32-CAM

### **II.2.1 Caractéristiques techniques de l'ESP32-CAM :**

Le module ESP32-CAM dispose quelques caractéristiques techniques très intéressantes disponibles sur la fiche technique du fabricant. Les caractéristiques les plus importantes sont [14] :

- Connectivité : Wi-Fi 802.11b / g / n + Bluetooth 4.2 avec BLE. Prend en charge le téléchargement d'images via Wi-Fi.
- Ports d'E/S : UART, SPI, I2C et PWM. Il a 9 broches GPIO.
- Fréquence d'horloge : jusqu'à 160Mhz.
- Puissance de calcul du microcontrôleur : jusqu'à 600 DMIPS.
- Mémoire : 520 Ko de SRAM + 4 Mo de PSRAM + fente pour carte SD
- Appareil photo : Prend en charge les caméras OV2640, ces types de caméras ont :
  - Un capteur de 2 MP
  - Résolution de l'image UXGA 1622 × 1200 px

- Format de sortie YUV422, YUV420, RGB565, RGB555 avec compression de données 8 bits.
- Transfert de l'une image entre 15 et 60 FPS.



**Figure II.4 :**Les composantes de la carte ESP32-CAM

### II.3 La carte Arduino

Une carte Arduino est une carte électronique programmable basée sur un microcontrôleur. Elle est utilisée pour créer des projets électroniques interactifs et des prototypes. L'Arduino est un système open-source, ce qui signifie que son matériel et son logiciel sont disponibles librement, ce qui permet aux utilisateurs de modifier et de partager leurs conceptions.

La carte Arduino est dotée de broches d'entrée/sortie numériques et analogiques qui permettent de connecter différents composants électroniques tels que des capteurs, des actionneurs, des afficheurs, etc. Ces broches peuvent être contrôlées en utilisant un langage de programmation spécifique à Arduino, dérivé du langage C/C++.

Les cartes Arduino sont disponibles dans différentes variantes, chacune ayant ses propres spécifications et fonctionnalités. La carte la plus couramment utilisée est l'ArduinoUno, qui est idéale pour les débutants en électronique et en programmation. Il existe également d'autres modèles comme l'ArduinoMega, l'Arduino Nano, l'Arduino Due, etc., qui offrent plus de broches d'E/S, une plus grande puissance de traitement ou des fonctionnalités spécifiques [15].

### II.3.1 Arduino MEGA :

L'Arduino Méga 2560 est un autre produit populaire parmi les cartes Arduino. Parfaite pour les applications Arduino un peu plus grandes, elle est constituée de tous les éléments nécessaires pour permettre la construction d'objets évolués [16].



Figure II.5 : La carte Arduino Méga 2560

#### a. Caractéristiques techniques de L'Arduino Méga 2560 :

- Alimentation:
  - via port USB ou
  - 7 à 12 V sur connecteur alim
- Microprocesseur: ATmega2560
- Mémoire flash: 256 kB
- Mémoire SRAM: 8 kB
- Mémoire EEPROM: 4 kB
- Interfaces:
  - 54 broches d'E/S dont 14 PWM
  - 16 entrées analogiques 10 bits
- Intensité par E/S: 40 mA
- Cadencement: 16 MHz

- 3 ports série
- Bus I2C et SPI
- Gestion des interruptions
- Fiche USB B
- Version:Rev 3
- Dimensions: 107 x 53 x 15 mm [17]

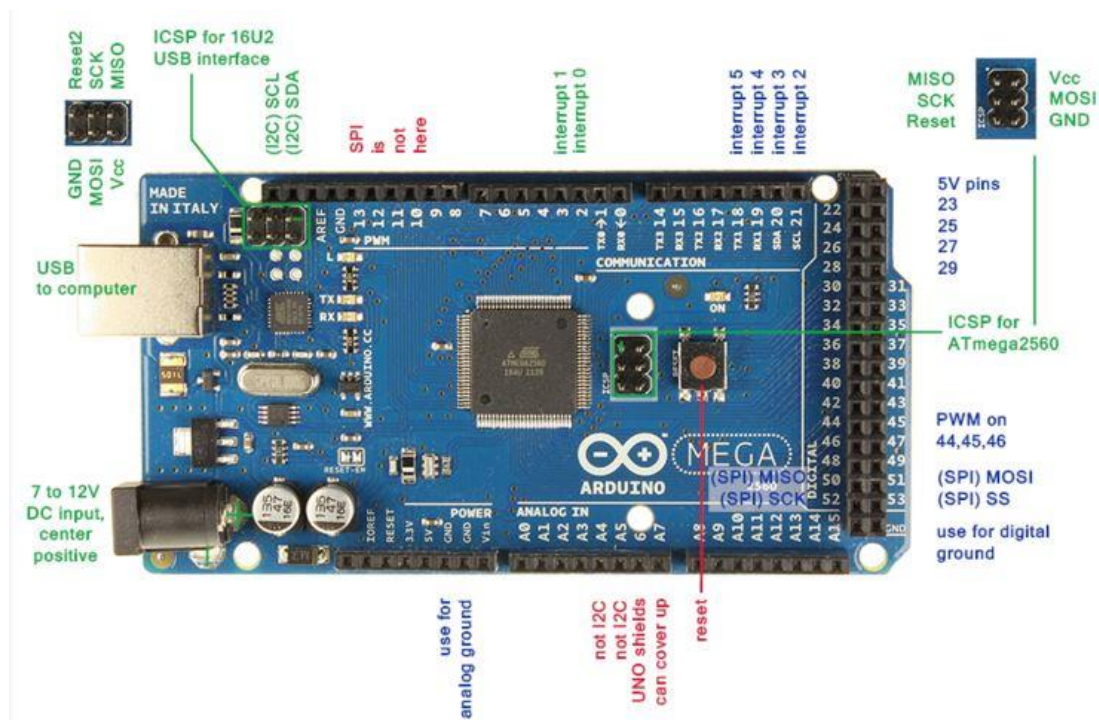
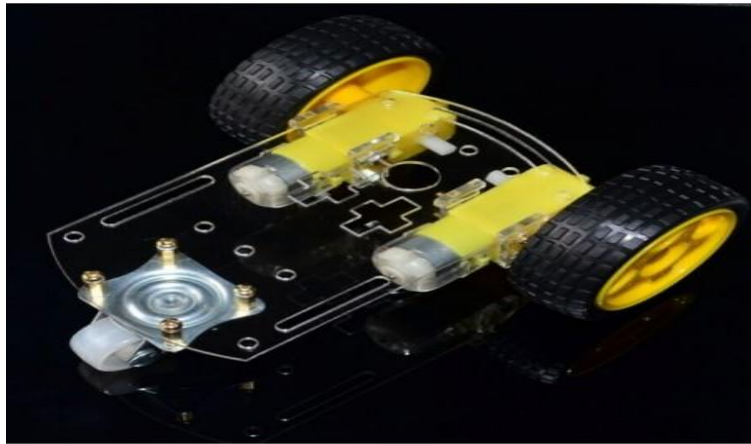


Figure II.6 : La carte Arduino Méga 2560 composantes

## II.4 Plateforme mobile robotique

Une plateforme mobile robotique est un système robotique qui intègre une base mobile pour se déplacer dans l'environnement. Ces plates-formes sont généralement utilisées comme base pour construire différents types de robots mobiles tels que des robots d'exploration, des robots de service, des robots de livraison, des robots de surveillance, etc [18].



**Figure II.7** : Plateforme mobile robotique

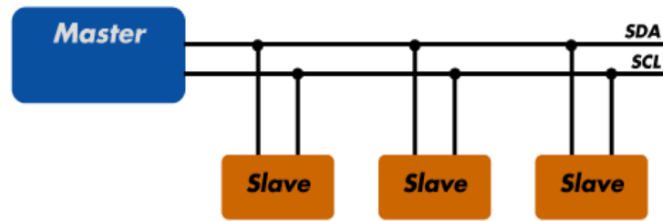
## **II.5 Afficheur OLED**

I2C, SPI et UART sont couramment utilisés comme moyens de communication entre les appareils au sein d'un système embarqué en raison de leur simplicité et de leur facilité d'utilisation. Bien qu'ils atteignent tous un objectif similaire d'envoi de données, ils présentent chacun de nombreuses différences ainsi que des avantages et des inconvénients à prendre en compte lors du choix de celui à utiliser.

### **II.5.1 OLED I2C :**

I2C (Inter-Integrated Circuit) est un protocole de communication série largement utilisé qui permet à plusieurs appareils électroniques de communiquer entre eux à l'aide d'une interface à deux fils. Il a été développé par Philips Semiconductors (aujourd'hui NXP Semiconductors) au début des années 1980 et est depuis devenu un protocole standard dans l'industrie électronique.

I2C utilise deux lignes bidirectionnelles : une ligne de données série (SDA) et une ligne d'horloge série (SCL). Les appareils sur le bus I2C sont connectés dans une configuration maître-esclave, où l'appareil maître initie et contrôle la communication avec les appareils esclaves [19].



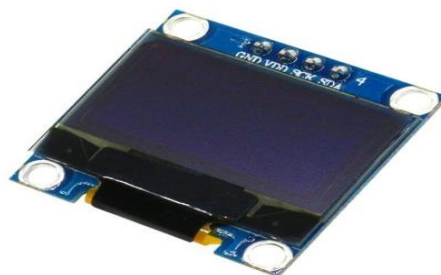
**Figure II.8:**Protocole I2C

Le protocole I2C prend en charge plusieurs maîtres et plusieurs esclaves sur le même bus, ce qui le rend adapté à la communication entre divers CI (circuits intégrés) tels que des capteurs, des EEPROM, des horloges en temps réel, des contrôleurs d'affichage et de nombreux autres types d'appareils.

Chaque appareil sur le bus I2C a une adresse unique, permettant au maître d'identifier et de communiquer avec des appareils esclaves spécifiques. La communication est généralement initiée par le périphérique maître, qui envoie une condition de démarrage suivie de l'adresse esclave 7 bits et d'un bit de lecture/écriture pour indiquer la direction du transfert de données. Le maître envoie ou reçoit ensuite des données vers/depuis le dispositif esclave de manière séquentielle.

I2C prend en charge différents modes de transfert de données, y compris le mode standard (100 kbit/s), le mode rapide (400 kbit/s), le mode rapide plus (jusqu'à 1 Mbit/s) et le mode haut débit (jusqu'à 3,4 Mbit/s). Ces modes définissent la vitesse maximale à laquelle les données peuvent être transférées sur le bus.

Dans l'ensemble, I2C est un protocole de communication polyvalent et largement adopté qui permet un échange de données efficace et fiable entre les différents circuits intégrés d'un système.



**Figure II.9:**Oled I2C

## II.5.2 OLED SPI :

Le protocole SPI (Serial Peripheral Interface) est couramment utilisé pour communiquer avec les écrans OLED. Il permet au microcontrôleur d'envoyer des données et des commandes à l'écran et d'en recevoir un retour. Les écrans OLED utilisent généralement une interface SPI pour transmettre des données en série, ce qui implique un périphérique maître (par exemple, un microcontrôleur) et un périphérique esclave (l'écran OLED).

La communication SPI pour les écrans OLED implique généralement les connexions suivantes :

- Horloge série (SCK) : cette ligne transporte des impulsions d'horloge générées par l'appareil maître pour synchroniser la transmission de données entre le maître et l'écran OLED.
- Master Out Slave In (MOSI): Cette ligne transporte les données du maître vers l'écran OLED. Le maître envoie des commandes et des données de pixel à l'écran à l'aide de cette ligne.
- Master In Slave Out (MISO): Cette ligne permet à l'écran OLED de renvoyer des données à l'appareil maître, bien qu'elle ne soit pas couramment utilisée dans la communication OLED SPI.
- Chip Select (CS) : Cette ligne est utilisée pour sélectionner ou activer l'affichage OLED. L'appareil maître active la ligne CS pour communiquer avec l'écran et la désactive lorsqu'elle n'est pas utilisée.

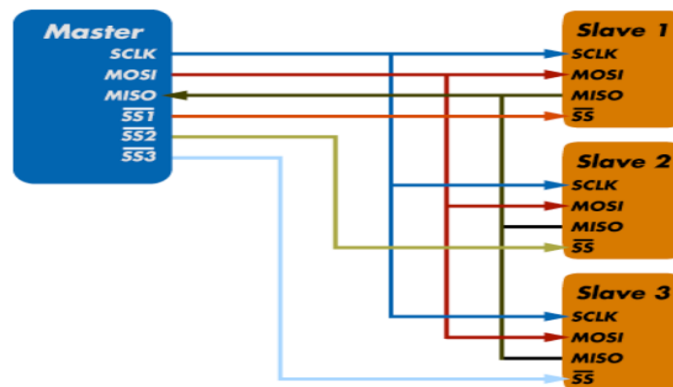


Figure II.10: Protocole SPI

L'OLED SPI fait référence au protocole de communication utilisé pour interfacier les écrans OLED à l'aide de la norme Serial Peripheral Interface (SPI). Ce protocole permet à un microcontrôleur ou à d'autres appareils d'envoyer des commandes et des données à l'écran OLED pour rendre des graphiques ou afficher des informations [20].



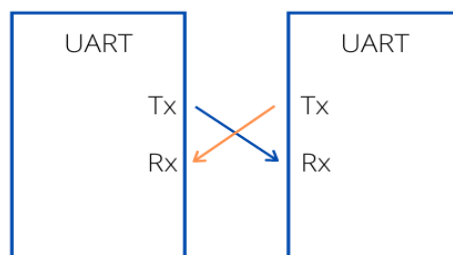
**Figure II.11:** Oled SPI

### II.5.3 OLEDUART :

UART, ou Universal Asynchronous Receiver/Transmitter, est un circuit physique dans un microcontrôleur ou un circuit intégré (CI) unique qui est utilisé pour mettre en œuvre une communication série entre des appareils dans un système embarqué. Essentiellement, le but principal d'un UART est de transmettre et de recevoir des données série.

Dans la communication UART, deux UART communiquent directement entre eux ; l'UART sur l'appareil émetteur, ou l'UART émetteur, reçoit des données parallèles de la CPU (microprocesseur ou microcontrôleur) et les convertit en données série. Ces données série sont transmises à l'UART sur l'appareil récepteur ou à l'UART récepteur. L'UART récepteur reconvertit les données série reçues en données parallèles et les envoie au CPU. Pour que l'UART convertisse les données série-parallèle et parallèle-série, des registres à décalage sur l'UART émetteur et récepteur sont utilisés.

Dans la communication UART, seuls deux fils sont nécessaires pour la communication : les données circulent de la broche Tx de l'UART émetteur (émetteur Tx) à la broche Rx de l'UART récepteur (récepteur Rx).



**Figure II.12 :** Protocole UART

Les données UART sont envoyées sur le bus sous la forme d'un paquet. Un paquet se compose d'un bit de départ, d'une trame de données, d'un bit de parité et de bits d'arrêt. Le bit de parité est utilisé comme mécanisme de contrôle d'erreur pour garantir l'intégrité des données.

L'UART est considéré comme "universel" car les paramètres, y compris la vitesse de transfert et la vitesse des données, sont configurables par le développeur. UART prend en charge la transmission de données bidirectionnelle, y compris les opérations semi-duplex et duplex intégral. Il est également asynchrone, ce qui signifie qu'il n'utilise pas de signal d'horloge pour synchroniser les bits de sortie de l'UART émetteur avec les bits d'échantillonnage de l'UART récepteur. Sans horloge, l'UART de réception et de transmission doit être sur le même débit en bauds ou débit binaire. Cela permet au système de savoir où et quand les bits ont été cadencés [21].



**Figure II.13 :Oled UART**

## **II.6 convertisseur de niveau logique (shifter)**

Un convertisseur de niveau logique est un dispositif qui permet de convertir un signal d'un niveau logique à un autre. Il est utilisé lorsque vous avez besoin de faire communiquer des composants électroniques qui fonctionnent avec des niveaux logiques différents, par exemple entre des circuits logiques TTL (Transistor-Transistor Logic) et des circuits logiques CMOS (Complementary Metal-Oxide-Semiconductor).

Il existe différents types de convertisseurs de niveau logique, mais l'un des plus couramment utilisés est le convertisseur bidirectionnel de niveau logique. Ce type de convertisseur permet de convertir un signal bidirectionnel entre deux niveaux logiques différents. Il utilise généralement des transistors pour réaliser la conversion [22].



**Figure II.14:** Convertisseur de niveau (shifterlevel)

## II.7. Câble de prototypage

Les câbles sont utilisés pour brancher les composants électroniques entre eux (Arduino, breadboards, ...) pour faire du prototypage.



**Figure II.15 :** câble de prototypage

## II.8. Module L298N

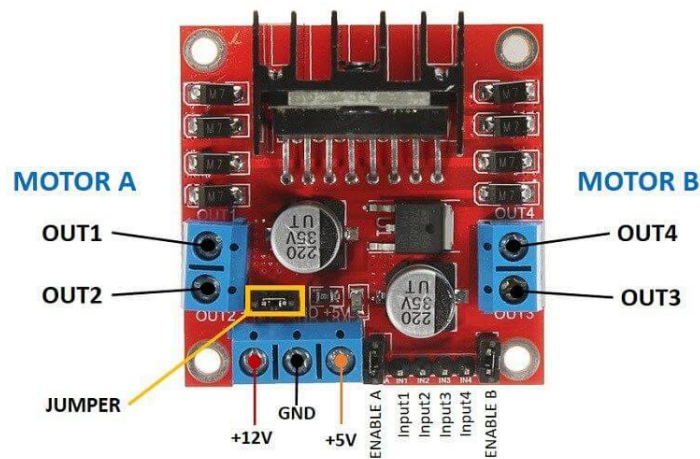
La carte L298N est un module de commande de moteur utilisé pour contrôler la vitesse et la direction des moteurs à courant continu (DC) et des moteurs pas à pas. Il est couramment utilisé dans les projets de robotique, les imprimantes 3D, les véhicules télécommandés et d'autres applications nécessitant un contrôle précis des moteurs.

La carte L298N utilise le circuit intégré L298N, qui est un double pont en H. Cela signifie qu'il peut contrôler deux moteurs indépendamment, en inversant la direction du courant pour changer la direction de rotation du moteur. Chaque moteur peut être contrôlé avec une tension pouvant aller jusqu'à 35 volts et un courant pouvant aller jusqu'à 2 ampères en continu (avec des pointes allant jusqu'à 3 ampères).

La carte L298N comporte généralement des borniers pour connecter facilement les moteurs, ainsi que des broches de commande pour contrôler la vitesse et la direction des

moteurs. Les broches de commande peuvent être activées à l'aide de signaux logiques (0 ou 5 volts) provenant d'une carte Arduino, d'un Raspberry Pi ou d'un autre microcontrôleur.

Il est important de noter que la carte L298N nécessite une alimentation externe pour alimenter les moteurs. Elle dispose d'un bornier séparé pour connecter une source d'alimentation appropriée, qui doit être compatible avec les spécifications des moteurs utilisés[23].



**Figure II.16 :** Carte L298N

## II.9. Batterie externe

Une batterie externe est un dispositif de stockage de l'électricité transportable pour un usage personnel. Les batteries externes sont généralement des batteries lithium-ion, appréciées pour leur grande densité énergétique (autour de 200 Wh/kg).

Les batteries externes disposent d'une capacité plus ou moins grande en fonction de l'usage attendu (outils, engins de déplacement). Les batteries externes sont fournies avec un chargeur spécifique s'il s'agit d'un branchement de recharge sur l'électricité domestique [24].



**Figure II.17 :**Batterie 9V



**Figure II.18:** Batterie externe (Power bank)

## II.10. Moteur CC

Un moteur à courant continu (CC) est un type de moteur électrique qui convertit l'énergie électrique en mouvement mécanique. Il est largement utilisé dans de nombreuses applications, des petits appareils électroniques aux gros équipements industriels.

Le fonctionnement d'un moteur à courant continu repose sur le principe de l'interaction entre un champ magnétique et un courant électrique. Le moteur est composé d'un rotor (partie mobile) et d'un stator (partie fixe). Le rotor comprend des bobines de fil conducteur, appelées enroulements, tandis que le stator comprend des aimants permanents ou des bobines d'excitation.

Lorsqu'un courant électrique est appliqué aux enroulements du rotor, un champ magnétique est créé autour des enroulements. Ce champ magnétique interagit avec le champ magnétique du stator, ce qui génère une force qui fait tourner le rotor. La direction de rotation dépend du sens du courant électrique dans les enroulements.

Pour inverser la direction de rotation d'un moteur à courant continu, il suffit de modifier le sens du courant dans les enroulements. Cela peut être réalisé en inversant la polarité de la source d'alimentation ou en utilisant un dispositif appelé commutateur de direction.

Les moteurs à courant continu présentent plusieurs avantages, tels qu'un couple élevé de démarrage, un contrôle précis de la vitesse et de la direction, une réversibilité facile et une large plage de vitesses. Ils sont souvent utilisés dans les applications nécessitant un contrôle de vitesse précis, comme les machines-outils, les robots, les ventilateurs et les véhicules électriques[38].



**Figure II.19:** Moteur à courant continu

## **II.11. Conclusion**

Dans ce chapitre on a abordé les différents composants et matériels qui font intervenir dans notre projet, cela nous a permis de concevoir notre système qui est base sur des diverses cartes (ESP32-CAM, Arduino Méga). La conception est une étape indispensable dans le but de réaliser un projet et lui assurer un bon fonctionnement.

# CHAPITRE III

## CONTROLE D'UN ROBOT MOBILE BASE SUR ESP32-CAM STREAMING SERVEUR

### III.1. Introduction

Dans ce chapitre nous allons voir une présentation de la télé opération contrôle d'un robot mobile qui est doté par un ESP32-CAM à travers WIFI. Nous pourrons contrôler le robot à l'aide d'un serveur Web qui affiche un flux vidéo de ce que le robot "voit", ainsi nous pouvons contrôler notre robot à distance même s'il est hors de notre vue. L'ESP32-CAM sera programmé à l'aide de l'IDE Arduino.

Dans ce chapitre, on s'est intéressé par les composants suivants :

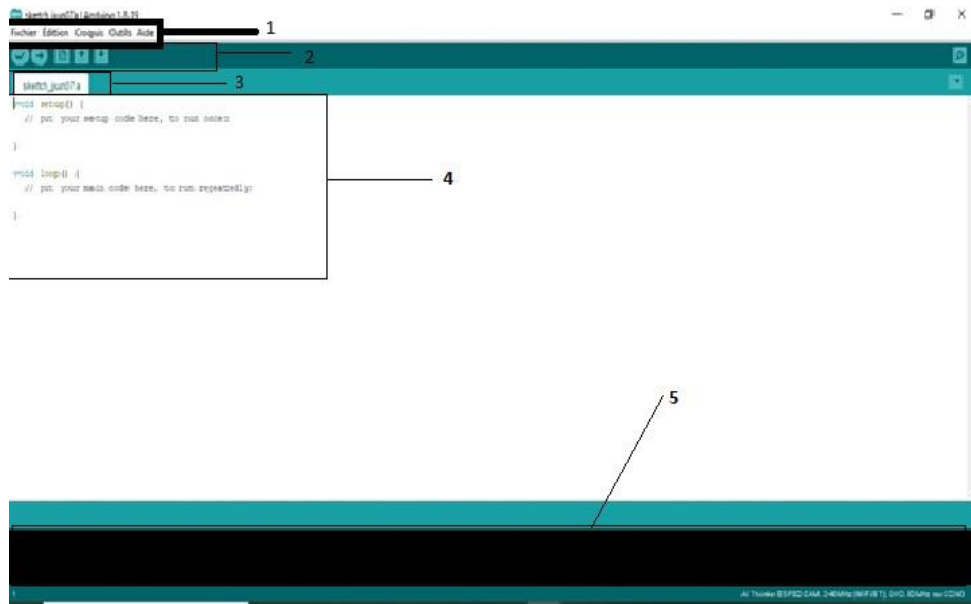
- ESP32-CAM
- Une carte L298N
- Plateforme mobile
- Batterie externe
- Circuit de prototypage

### III.2. Arduino IDE

L'IDE Arduino (environnement de développement intégré) est une application logicielle utilisée pour écrire, compiler et télécharger du code sur les microcontrôleurs Arduino. Il fournit une interface facile à utiliser pour les utilisateurs débutants et avancés pour développer et tester leurs projets Arduino. L'environnement de programmation Arduino appelé IDE est distribué sur le site d'Arduino (compatible avec Windows, Linux et mac). L'interface de l'application Arduino vous permet de créer et éditer un programme (appelé sketch) qui sera compilé puis téléversé sur la carte Arduino. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme téléversé sur la carte[25].

#### III.2.1. Fenêtre générale de l'application IDE





La figure ci-dessous présente la fenêtre de l'Arduino IDE :



**Figure III.1 :**Arduino IDE

1. Un menu : permet créer de nouveaux sketches, de les Sauvegarder et gérer les paramètres de communication avec la carte Arduino.
2. Une barre d'actions.
3. Un ou plusieurs onglets correspondant aux sketches.
4. Une fenêtre de programmation.
5. Une console qui affiche les informations et erreurs de compilation et de téléversement du programme.

**Barre d'actions :**

	<p>Bouton « Verify » (Vérifier) ; il permet de compiler votre programme et de vérifier si des erreurs s'y trouvent. Cette procédure prend un certain temps d'exécution et lorsque est terminée, elle affiche un message de type « Binary sketch size : ... » indiquant la taille du sketch téléversé.</p>
	<p>Pour transmettre le sketch compilé avec succès sur la carte Arduino dans le microcontrôleur.</p>
	<p>Bouton « New » (Nouveau) ; ce bouton permet de créer un nouveau sketch.</p>
	<p>Bouton « Open » (Ouvrir) ; il fait apparaître un menu qui permet d'ouvrir un sketch qui figure dans votre dossier de travail ou des exemples de sketches intégrés au logiciel.</p>



Bouton « Save » (Sauvegarder) ; il permet de sauvegarder votre sketch

### III.2.2. L'exploitation de l'Arduino IDE

Afin qu'on puisse utiliser l'Arduino IDE on suit les étapes suivantes :

- Lancer Arduino IDE : Ouvrez le logiciel Arduino IDE
- Connectez Arduino : Connectez la carte Arduino avec l'ordinateur à l'aide d'un câble USB.
- Sélectionnez la carte : dans l'IDE Arduino, accédez au menu "Outils" et sélectionnez la carte appropriée .
- Select Port : Dans le menu "Tools", choisissez le port série auquel de carte Arduino est connectée. Sous Windows, ce sera généralement quelque chose comme "COM1" ou "COM2", tandis que sur macOS, il apparaîtra comme "/dev/cu.usbmodem..." ou "/dev/tty.usbmodem...".
- Écrire du code : dans l'IDE Arduino, on à commencer à écrire le code dans la zone de l'éditeur. Le code se compose de deux fonctions principales : setup() et loop(). La fonction setup() est exécutée une fois au début, tandis que la fonction loop() s'exécute à plusieurs reprises.et écrire le code à l'intérieur de ces fonctions.
- Vérifier/Compiler : Cliquez sur l'icône de coche (ou allez dans "Sketch" > "Vérifier/Compiler") pour compiler le code. L'IDE vérifiera les éventuelles erreurs de syntaxe dans le code. S'il y a des erreurs, elles seront affichées dans la zone "Sortie" en bas de l'IDE.
- Upload : Une fois le code compilé avec succès, cliquez sur l'icône de la flèche droite (ou allez dans "Sketch" > "Upload") pour uploader le code sur la carte Arduino. L'IDE compilera à nouveau le code, puis le téléchargera sur la carte. Nous verrons la progression du téléchargement dans la zone "Sortie".
- Surveiller la sortie série (facultatif) : si le code utilise l'objet série pour imprimer la sortie, Nous pouvons ouvrir le moniteur série dans l'IDE Arduino en cliquant sur l'icône de la loupe (ou en allant dans "Outils" > "Moniteur série"). Il nous permet de visualiser la sortie de la carte Arduino.

### III.3. Navigateur Web (Browser)

Le navigateur Web est un logiciel. Son but est de visualiser le contenu d'une page internet. D'une manière générale, un browser permet d'extraire des ressources et des

informations du Web en les téléchargeant (sites internet, images, vidéos...) afin de les afficher sur l'écran du périphérique d'un utilisateur (un smartphone ou un ordinateur, par exemple). Le premier navigateur Web était appelé WorldWideWeb : il a été créé en 1990 par Sir Tim Berners-Lee.

Les meilleurs navigateurs web sont Google Chrome, Mozilla Firefox, Microsoft Edge (ex Internet Explorer), Safari, Brave, Opera, Tor Browser, Vivaldi ou encore Sidekick.

L'interface utilisateur est généralement personnalisable, pour vous permettre de simplifier la navigation grâce à des interfaces intuitives. Les navigateurs web permettent de naviguer sur Internet sur desktop (Windows, macOS ou Linux) mais aussi sur mobile (iOS, Android) grâce aux versions mobiles [26].



**Figure III.2 :**Exemple de Navigateur Web

#### **III.4. Conception du contrôleur d'un robot mobile doté par ESP32-CAM**

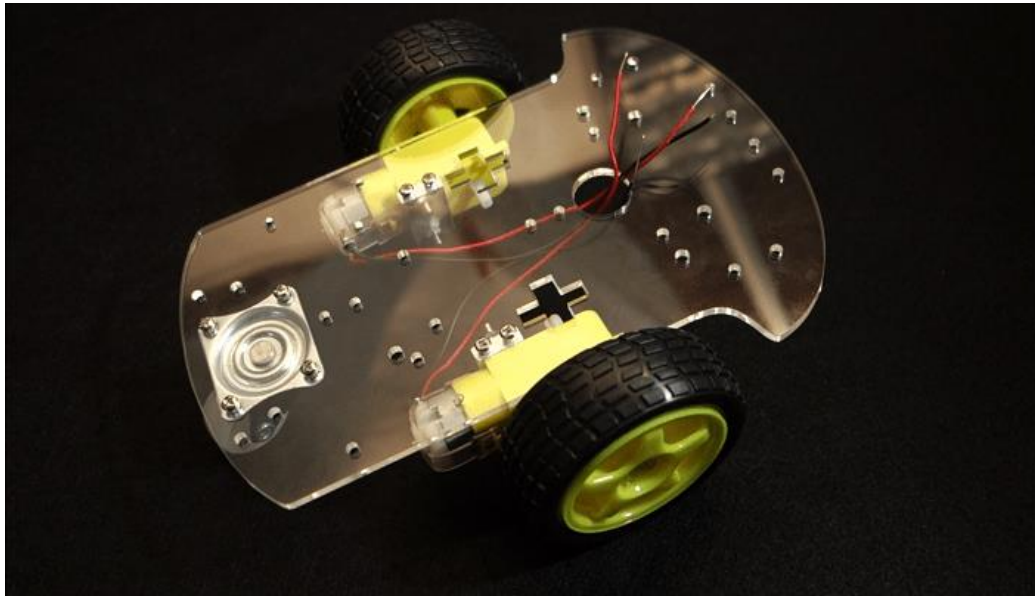
Le robot sera contrôlé via Wi-Fi à l'aide d'une carte ESP32-CAM. Nous créerons une interface Web pour contrôler le robot, accessible depuis n'importe quel appareil (Laptop, Smartphone) de notre réseau local.

La page Web affiche également un streaming vidéo de ce que le robot "voit". Pour des bons résultats avec le streaming vidéo, nous utilisons un ESP32-CAM avec antenne externe.



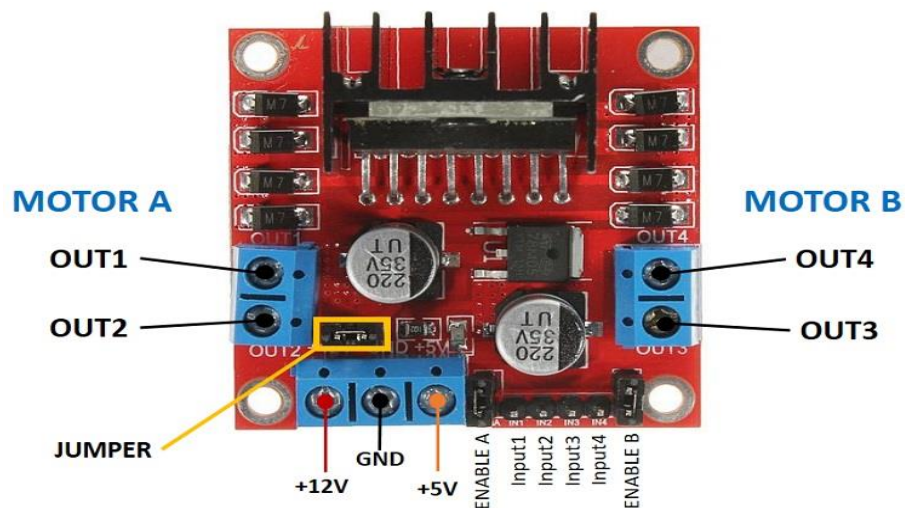
**Figure III.3 :**Antenne externe

Pour réaliser notre projet, nous allons utiliser une plateforme mobile (Smart Kit) où elle est facile à assembler. Cette plateforme constituée de deux moteurs à courant continu attachés par deux roues.



**Figure III.4 :**Plateforme mobile (Smart Kit Robot)

Il existe de nombreuses façons de contrôler les moteurs à courant continu du robot, dans notre projet nous utiliserons le pilote de moteur L298N qui permet de contrôler facilement la vitesse et la direction de 2 moteurs à courant continu.



**Figure III.5 :**Pilote moteur L298N

Les moteurs consomment beaucoup de courant, donc on doit utiliser une alimentation externe pour les moteurs. Cela signifie qu'on a besoin de deux sources d'alimentation différentes. L'une pour alimenter les moteurs à courant continu, et l'autre pour alimenter

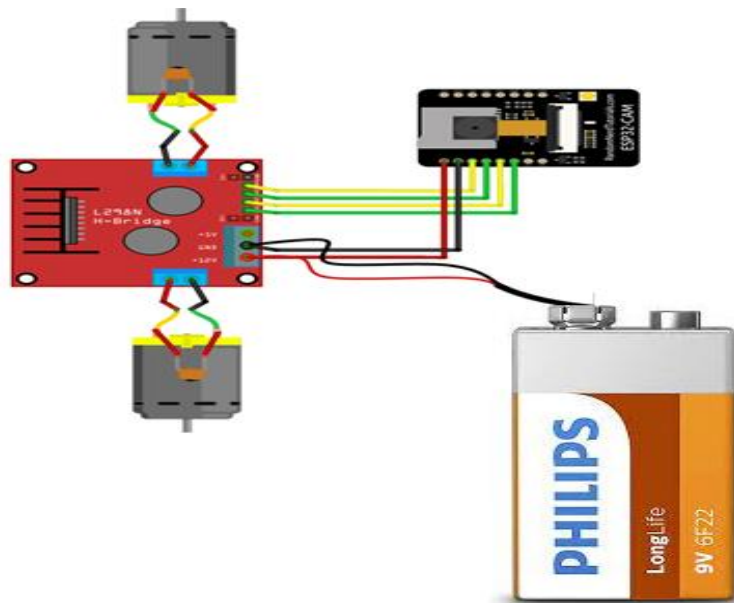
l'ESP32.

Pour alimenter le robot (moteurs) en utilisant une source d'alimentation simple qui est une batterie 9V (voir chapitre II). Nous avons utilisé une banque d'alimentation/chargeur portable pour alimenter ESP32-CAM et cela a bien fonctionné.



**Figure III.6 :**Banque d'alimentation

On commence notre travail par connecter l'ESP32-CAM avec le pilote du moteur comme indiqué dans le schéma de principe(Figure III.7). Où nous avons utilisé la pile pour alimenter les moteurs et la banque d'alimentation pour alimenter ESP32-CAM.



**Figure III.7 :**Schéma de montage

Le tableau suivant montre les connexions entre l'ESP32-CAM et le pilote de moteur L298N.

L298N Motor Driver	ESP32-CAM
IN1	GPIO 14
IN2	GPIO 15
IN3	GPIO 13
IN4	GPIO 12

**Tableau III.1** : Connexions des pins entre ESP32-CAM et L298N

### III.4.1. Fonctionnement de code

En premier temps on choisit le type de caméra qui est dans notre projet : AI THINKER, après on introduit les informations d'identification du réseau dans ces deux lignes de code, où notre réseau est OPPO A54 :

```
// Replace with your network credentials
const char* ssid = "OPPO A54";
const char* password = "1234567890";
```

En deuxième temps on définit les GPIO de l'ESP32 qui contrôleront les moteurs. Chaque moteur est contrôlé par deux broches :

```
#define MOTOR_1_PIN_1 14
#define MOTOR_1_PIN_2 15
#define MOTOR_2_PIN_1 13
#define MOTOR_2_PIN_2 12
```

Ensuite, on définit les commandes qui font commander le robot en différentes situations (avant, arrière, gauche, droite) :

1. Avant :

```
digitalWrite(MOTOR_1_PIN_1, 1);
digitalWrite(MOTOR_1_PIN_2, 0);
digitalWrite(MOTOR_2_PIN_1, 1);
digitalWrite(MOTOR_2_PIN_2, 0);
```

2. Droite :

```
digitalWrite(MOTOR_1_PIN_1, 1);
digitalWrite(MOTOR_1_PIN_2, 0);
```

```
digitalWrite(MOTOR_2_PIN_1, 0);
```

```
digitalWrite(MOTOR_2_PIN_2, 0);
```

3. Gauche :

```
digitalWrite(MOTOR_1_PIN_1, 0);
```

```
digitalWrite(MOTOR_1_PIN_2, 0);
```

```
digitalWrite(MOTOR_2_PIN_1, 1);
```

```
digitalWrite(MOTOR_2_PIN_2, 0);
```

4. Arrière :

```
digitalWrite(MOTOR_1_PIN_1, 0);
```

```
digitalWrite(MOTOR_1_PIN_2, 1);
```

```
digitalWrite(MOTOR_2_PIN_1, 0);
```

```
digitalWrite(MOTOR_2_PIN_2, 1);
```

5. Stop :

```
digitalWrite(MOTOR_1_PIN_1, 0);
```

```
digitalWrite(MOTOR_1_PIN_2, 0);
```

```
digitalWrite(MOTOR_2_PIN_1, 0);
```

```
digitalWrite(MOTOR_2_PIN_2, 0);
```

En dernier temps on introduit nos informations institutionnelles ( université, faculté, département) ainsi le titre du projet à travers des codes HTML :

```
<h1>Controle du robot mobile basant sur ESP32-CAM streaming serveur</h1>
```

```
<h2>Universite Abbes LaghrourKHenchela</h2>
```

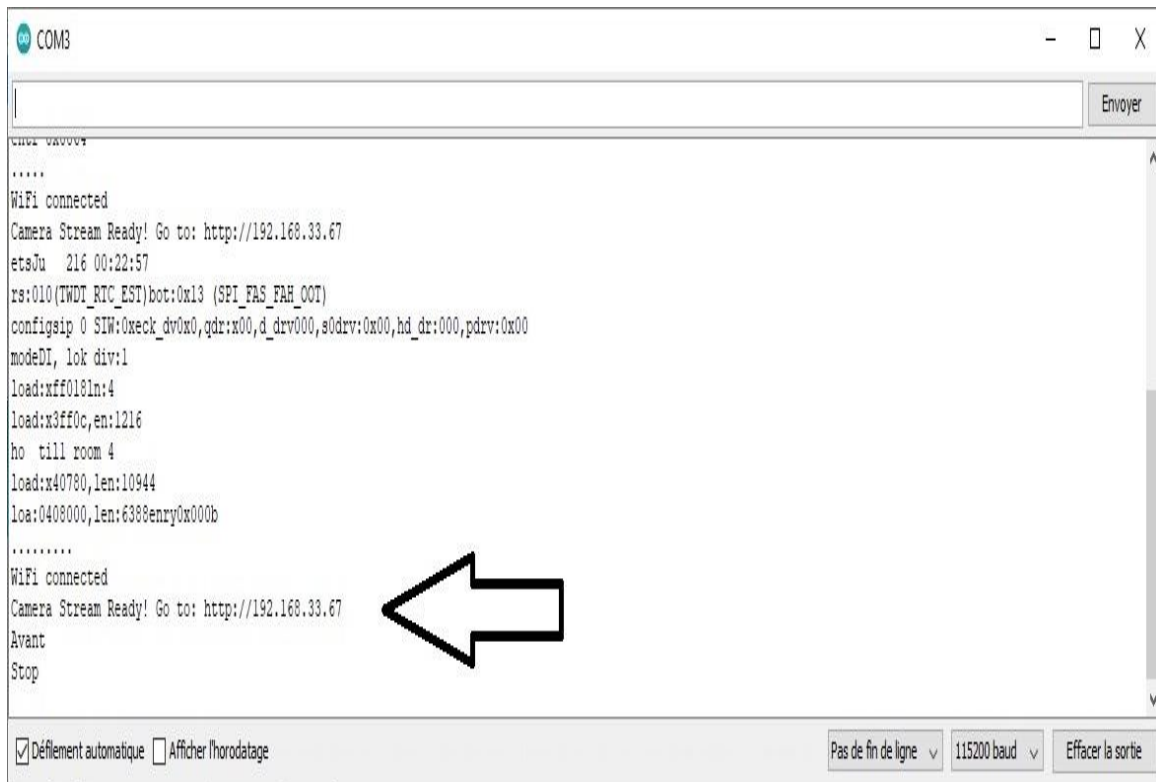
```
<h3>Faculte des sciences et technologie</h3>
```

```
<h4>Departement de génie industriel</h4>
```

```
<h5>M2 Automatique et informatique industrielle</h5>
```

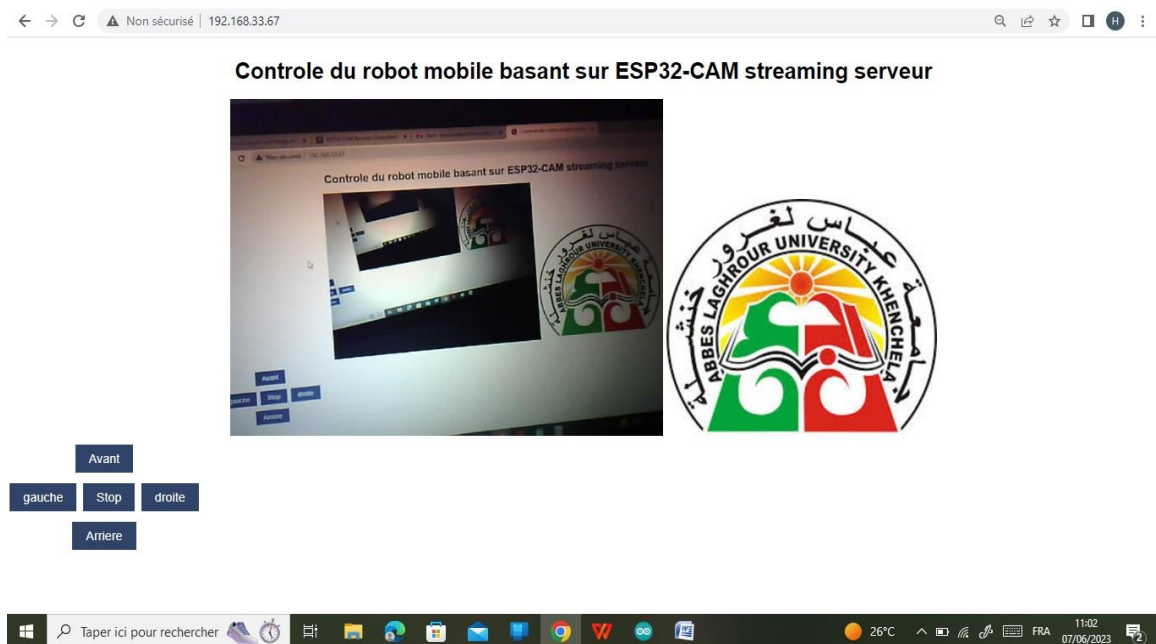
Finalement on télécharge le programme sur la carte ESP32 à travers le port COM3 tout on cliquant sur le Botton Téléverser. Après le téléchargement on ouvre le moniteur série pour récupérer l'adresse IP selon la figure (III.8).

L'adresse IP récupérée est : 192.168.33.67



**Figure III.8 :** Adresse IP récupérée dans moniteur série

On ouvre un navigateur et on saisit l'adresse IP récupérée. Une page Web similaire devrait s'ouvrir comme indiquée la figure suivante :



**Figure III.9 :** La page web pour le contrôle du robot

La page web ouverte à travers l'adresse IP récupérée présente la vidéo streaming

transmise par l'ESP32-CAM qui est Back-end et le Laptop définit comme Front-end. Les cinq onglets (Avant, Arrière, Droite, gauche, Stop) sont les commandes du robot.

### III.4.2 Testes et interprétations

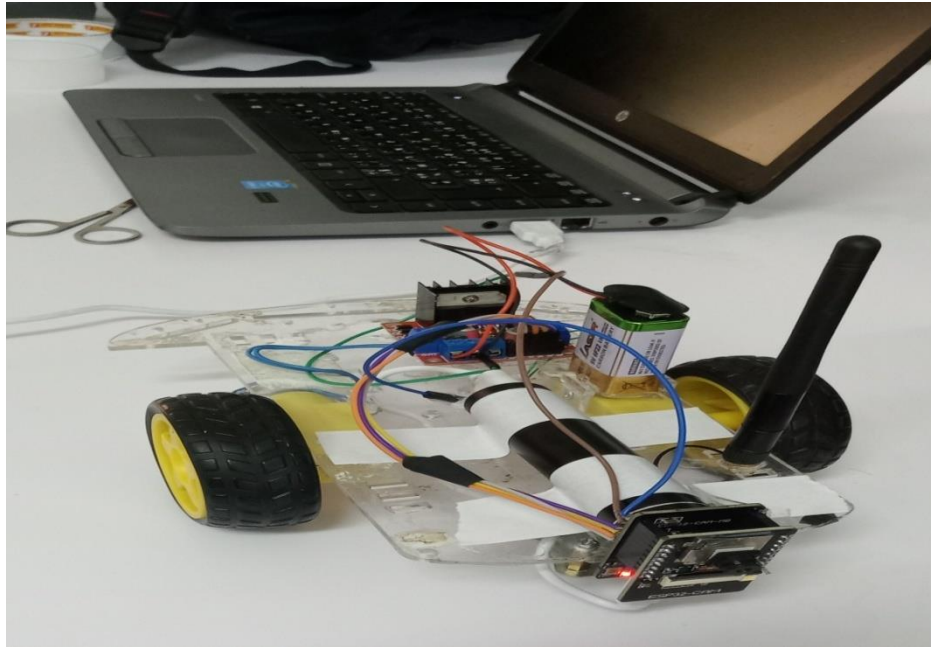


Figure III.10: Présentation du robot mobile

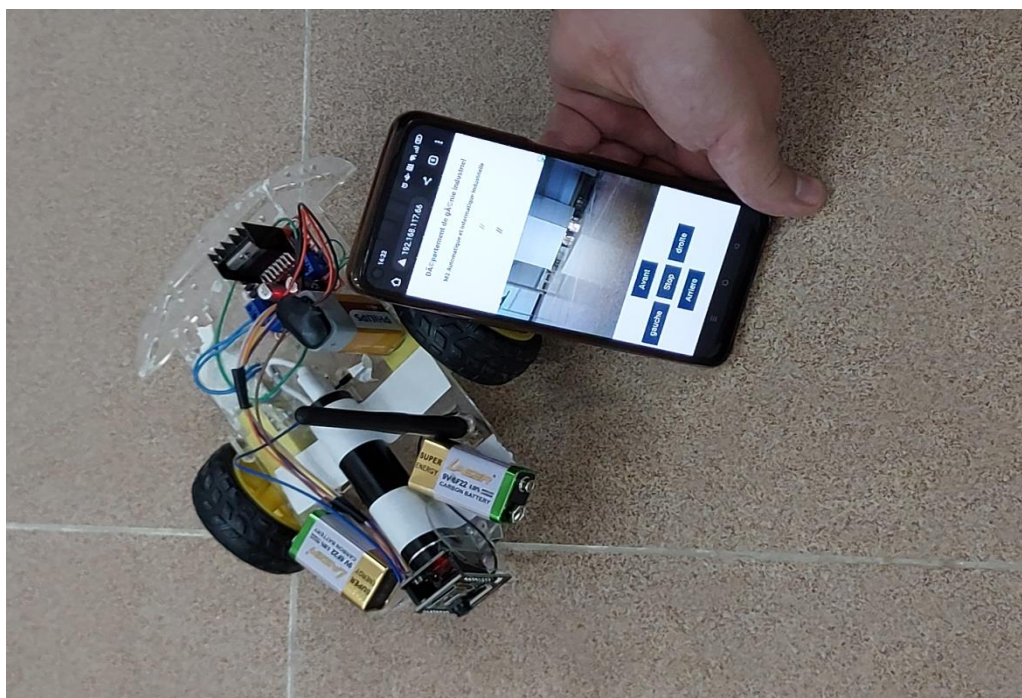
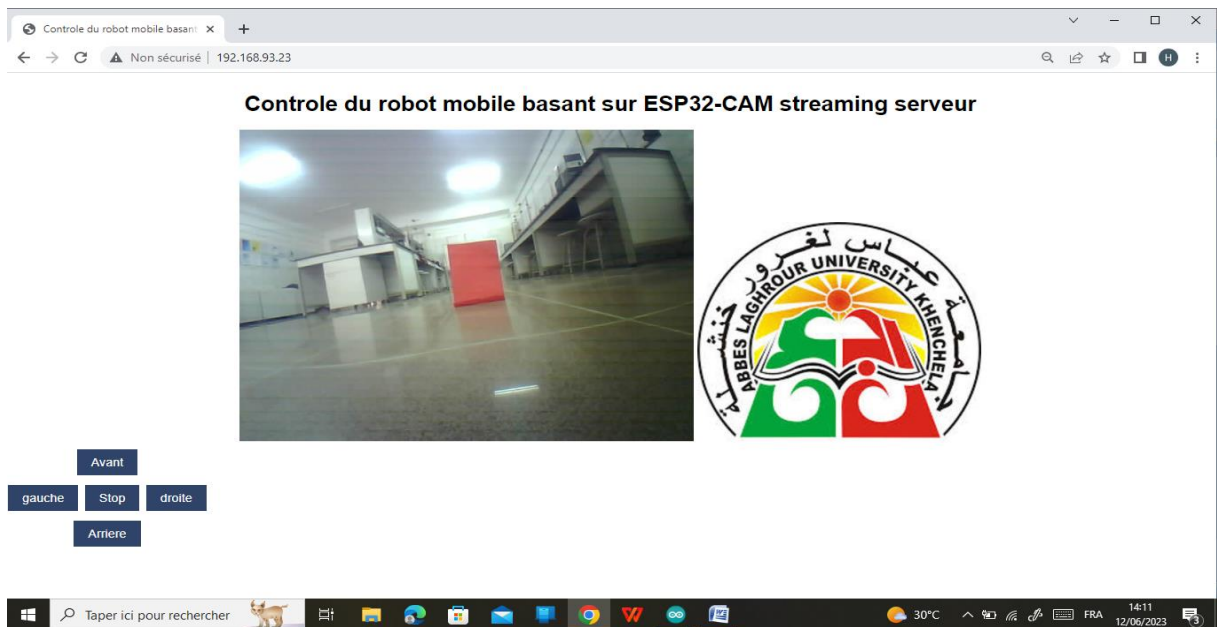


Figure III.11 : Contrôle par smart phone

Dans ce que suit on va présenter les différentes simulations et test appliqués sur notre

robot, où on a commandé le robot vers les différentes directions et situations (avant, arrière, droite, gauche). On remarque que le robot rejoint vers les directions choisies et garder l'émission de la vidéo par la technique streaming.

- **Test 1 :** on dirige le robot vers l'avant.



**Figure III.12 :** Le robot move vers l'avant avec éloignement de la cible



**Figure III.13 :** Le robot move vers l'avant avec rapprochement de la cible

- **Test 2 :** on dirige le robot vers l'arrière.



**Figure III.14 :** Le robot recule vers l'arrière

- **Test 3:** on oriente le robot vers la gauche.



**Figure III.15 :** Le robot tourne vers la gauche

- **Test 3:** on oriente le robot vers la droite.



**Figure III.16 :** Le robot tourne vers la droite

### III.4. Conclusion

Dans ce chapitre, nous avons construit un robot télécommandé et contrôlé à l'aide de l'ESP32-CAM et un serveur Web.

L'ESP32-CAM nous a permis à contrôler les moteurs à courant continu qui font mouvoir le robot ainsi à l'aide du pilote de moteur L298N qui facilite le contrôle de la vitesse et la direction.

Si on souhaitera contrôler notre robot en dehors de la portée de notre réseau local, on peut envisager de définir l'ESP32-CAM comme point d'accès. De cette façon, l'ESP32-CAM n'a pas besoin de se connecter à notre routeur, il crée son propre réseau wi-fi et les appareils wi-fi à proximité comme smartphone peuvent s'y connecter.

Le travail réalisé dans ce chapitre montre l'efficacité de contrôle à distance ainsi facilite au robot mobile de sa naviguer sans collision avec les objets.

# CHAPITRE IV

## CONTROLE ET SUIVI DES COULEURS PAR UN ROBOT MOBILE

### IV.1. Introduction

L'idée que nous allons construire dans ce projet est crée un serveur Web qui permet le suivi des couleurs d'un objet en mouvement par un robot mobile sur l'interface du serveur web, on joue avec plusieurs configurations pour bien sélectionner la couleur que l'on souhaite suivre. Ensuite, le navigateur (Browser) envoie les coordonnées  $x$  et  $y$  en temps réel du centre de masse de l'objet en mouvement à la carte ESP32-CAM, cette dernière prendra les commandes nécessaires pour le robot afin qu'il puisse suivre la couleur en mouvement. Le travail qui sera présenté dans ce chapitre basé sur les images récupérées par ESP32-CAM, donc on s'est intéressé par le traitement d'image et ses applications donc, dans ce que suit on va présenter un aperçu général sur le traitement d'image.

### IV.2. Généralités sur l'image

Une image est une représentation planaire d'un objet quelconque. Mathématiquement, c'est une fonction bidimensionnelle de la forme  $f(x, y)$ , où  $f(x_0, y_0)$  est la valeur de  $f$  aux coordonnées  $(x_0, y_0)$ . Donc, c'est un processus continu 2D résultant d'une mesure physique. L'amplitude de  $f$  est appelée intensité (ou niveau de gris) de l'image au point de coordonnées  $(x, y)$ [27].

Si une image est générée par un processus physique,  $f(x, y)$  correspond à l'énergie irradiée par ce processus. Exemples de processus physiques: Emission et réflexion d'ondes lumineuses, Rayonnement infrarouge, rayons X, Echo ultrason, etc...

#### IV.2.1. Traitement d'image

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années. On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

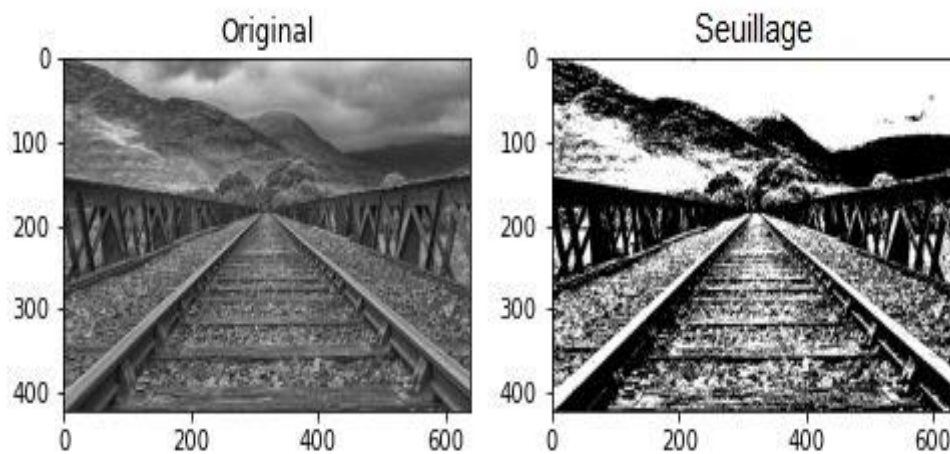
De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes[28].

Elle propose la plupart des opérations classiques en traitement bas niveau des images :

- lecture, écriture et affichage d'une image ;
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs ;
- lissage, filtrage ;
- seuillage d'image (méthode d'Otsu, seuillage adaptatif)
- segmentation
- morphologie mathématique.

#### IV.2.2. Seuillage d'image

Le seuillage d'image est une technique de binarisation d'image, consistant à remplacer les niveaux de gris d'une image par un ensemble de pixels prenant la valeur 255 (blanc) ou 0 (noir) selon que sa valeur initiale est inférieure ou supérieure à une valeur définie comme seuil[29].



**FigureIV.1 :**Seuillage d'image

##### IV.2.2.1. Seuillage globale :

Soit une image  $I$  de dimension  $N \times M$ ,  $I(i,j)$  étant la valeur d'intensité d'un pixel de la ligne  $i$  et de la colonne  $j$ , le seuillage global ou simple d'images consiste à fixer un seuil  $s$ , à partir duquel tout pixel ayant une intensité supérieure ou égale au seuil  $s$  est affecté de la valeur 255 et les autres ayant la valeur 0. L'image binaire est obtenue par la formule

suivante[30] :

$$g(i, j) = \begin{cases} 255 & \text{si } (i, j) \geq s \\ 0 & \text{sinon} \end{cases} \quad (\text{IV.1})$$

#### IV.2.2.2. Méthodes de seuillage :

Il existe plusieurs méthodes de seuillage d'images plus complexes et plus performantes que le seuillage simple, les plus utilisées :

- Méthode d'Otsu.
- Méthode de Niblack.
- Méthode de Bernsen.

#### D. Méthode d'Otsu :

La méthode d'Otsu essaye de trouver le seuil  $S$  qui sépare l'histogramme de façon optimale en deux segments (qui maximise la variance inter-segments ou bien qui minimise une mesure de variance intra-segment). Le calcul de la variance inter-classe ou intra-classes est basée sur l'histogramme normalisé  $H = [h_0 \dots h_{255}]$  de l'image d'où

$$\sum h_i = 1. \quad (\text{IV.2})$$

La variance inter-classe pour chaque seuil  $s$  est donnée par :

$$\sigma_{\text{inter}}^2 = q_1(s) \times q_2(s) \times [\mu_1(s) - \mu_2(s)]^2 \quad (\text{IV.3})$$

Tel que:

$$\mu_1(s) = \frac{1}{q_1(s)} \sum_{i=0}^{s-1} h(i) \times i \quad \text{et} \quad \mu_2(s) = \frac{1}{q_2(s)} \sum_{i=s}^{255} h(i) \times i \quad (\text{IV.4})$$

$$q_1(s) = \sum_{i=0}^{s-1} h(i) \quad \text{et} \quad q_2(s) = \sum_{i=s}^{255} h(i) \quad (\text{IV.5})$$

La limite de cette méthode est qu'elle n'est applicable que lorsque l'image est bimodale (l'histogramme comporte deux pics) [31].

#### E. Méthode Niblack :

L'algorithme de Niblack calcule un seuil local à chaque pixel en glissant une fenêtre rectangulaire sur toute l'image. Le seuil  $T$  est calculé en utilisant la moyenne  $m$  et l'écart-type  $\sigma$  de tous les pixels dans la fenêtre (voisinage du pixel en question). Ainsi le seuil  $T$  est donné par :

$$T = m + k \times \sigma \quad (\text{IV.6})$$

Tel que  $k$  est un paramètre utilisé pour déterminer le nombre de pixels de contours considérés comme des pixels de l'objet, et prend des valeurs négatives ( $k$  est fixé - 0.2 par les auteurs). Dans une fenêtre de taille  $25 \times 25$  donne des bons résultats.

Il existe dans la littérature plusieurs travaux d'évaluation des méthodes de binarisation dont plusieurs d'entre eux, considèrent la méthode de Niblack (avec une étape de post-traitement de Yanowitz and Bruckstein) Comme la meilleur de point de vue des résultats obtenus[32].

#### ***F. Méthode Bernsen :***

C'est une méthode locale adaptative dont le seuil est calculé pour chaque pixel de l'image. Ainsi pour chaque pixel de coordonnées  $(x, y)$ , le seuil est donné par:

$$T(x, y) = \frac{Z_{\text{bas}} + Z_{\text{haut}}}{2} \quad (\text{IV.7})$$

Tel que  $Z_{\text{bas}}$  et  $Z_{\text{haut}}$  sont le niveau de gris le plus bas et le plus haut respectivement, dans une fenêtre carré  $r \times r$  centré sur le pixel  $(x, y)$ .

Cependant si la mesure de contraste  $C(x, y) = (Z_{\text{haut}} - Z_{\text{bas}})$  est inférieure à un certain seuil  $l$ , alors le voisinage consiste en une seule classe: fond ou bien texte. Dans les tests réalisés montre que le choix de  $l=15$  et  $r=15$  donne les meilleurs résultats[33].

#### **IV.2.3.Détection des Contours**

La détection de contour est une étape préliminaire à de nombreuses applications de l'analyse d'images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- discontinuités de la fonction de réflectance (texture, ombre),
- discontinuités de profondeur (bords de l'objet),

Et sont caractérisés par des discontinuités de la fonction d'intensité dans les images. Le principe de la détection de contours repose donc sur l'étude des dérivées de la fonction d'intensité dans l'image : les extréma locaux du gradient de la fonction d'intensité et les passages par zéro du Laplacien. La difficulté réside dans la présence de bruit dans les images[34].

Les approches utilisées pour la détection des contours qui sont couramment utiles dans la littérature sont :

- **Approche gradient** : détermination des extréma locaux dans la direction du gradient.
- **Approche laplacien** : détermination des passages par zéro du laplacien.

Ces approches reposent sur le fait que les contours correspondent des discontinuités d'ordre 0 de la fonction d'intensité.

Le calcul de dérivée nécessite un pré-filtrage des images. Filtrage linéaire pour les bruits de moyenne nulle (par exemple bruit blanc Gaussien, filtre Gaussien). Filtrage non-linéaire pour les bruits impulsionnels (filtre médian par exemple).

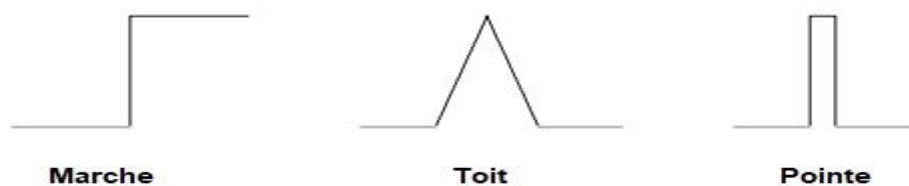
Les différentes approches existantes se classent ensuite suivant la manière d'estimer les dérivées de la fonction d'intensité :

- Différences finies.
- Filtrage optimal.
- Modélisation de la fonction d'intensité.

#### ***E. Différents types de contours***

Les différents contours trouvés dans la littérature sont :

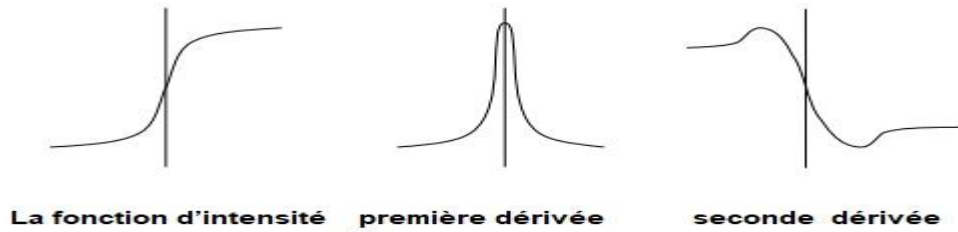
- marche.
- toit.
- pointe.



**Figure IV.2 :** Différents types de contours

#### ***F. La fonction d'intensité***

La fonction d'intensité au voisinage d'un contour en marche et ses dérivées première et seconde.



**Figure IV.3 :**La fonction d'intensité

**G. Le gradient d'une image :**

Le gradient d'une image est le vecteur  $\nabla I(x, y)$  défini par :

$$\nabla I(x, y) = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^t \quad (IV.7)$$

Il est donc caractérisé par un module  $m$  et une direction  $\phi$  dans l'image :

$$m = \sqrt{\left( \frac{\partial I(x, y)}{\partial x} \right)^2 + \left( \frac{\partial I(x, y)}{\partial y} \right)^2} \quad (IV.8)$$

$$\phi = \arctan\left( \frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x} \right) \quad (IV.9)$$

La direction du gradient maximise la dérivée directionnelle.

La dérivée de  $I(x, y)$  dans une direction donnée  $d$  s'écrit :

$$\nabla I(x, y) \cdot d. \quad (IV.10)$$

Le gradient d'une image filtrée :

$$\nabla I'(x, y) = \nabla (I(x, y) * h(x, y)) = \nabla I(x, y) * h(x, y) = I(x, y) * \nabla h(x, y). \quad (IV.11)$$

**H. Le laplacien d'une image :**

Le laplacien d'une image d'intensité  $I(x, y)$  est défini par :

$$\nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2}. \quad (IV.12)$$

Invariant aux rotations de l'image.

Le laplacien est souvent utilisé en amélioration d'images pour accentuer l'effet de contour :

$$I'(x, y) = I(x, y) - c \nabla^2 I(x, y). \quad (IV.13)$$

Sensibilité au bruit accrue par rapport au gradient.

Le laplacien d'une image filtrée :

$$\Delta I'(x, y) = \Delta I(x, y) * h(x, y) = I(x, y) * \Delta h(x, y). \quad (IV.14)$$

### IV.3. Langage HTML

L'HTML est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.



**Figure IV.4 :**LogoHTML

Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme le texte et de placer des éléments interactif, tel des liens, des images ou bien encore des animations. Ces éléments ne sont pas dans le code source d'une page codé en HTML mais "à coté" et la page en HTML ne fait que reprendre ces éléments. Pour visualiser une page en HTML il est nécessaire d'utiliser un navigateur web.

La plupart du temps d'autres langages informatiques sont associé à une page codé en HTML. Par exemple le CSS, qui permet de mettre en forme le contenu d'une page codé en HTML. Il y a également des langages informatiques qui dépendent entièrement du HTML, tel que le Javascript[35].

### IV.4. Langage Javascript

JavaScript est un langage de programmation principalement utilisé sur Internet, en complément de HTML et CSS. Il utilise des scripts pour créer du contenu dynamique. Il complète ainsi les deux autres langages de base du Web et peut stocker des valeurs, faire

des opérations ou encore exécuter du code selon certains évènements. C'est un langage orienté prototype, c'est-à-dire semblable à un langage orienté objet, mais sans classe.



**Figure IV.5 :**Logo Javascript

Bien qu'on le retrouve principalement côté client, il peut aussi être utilisé côté serveur. Lorsqu'il se charge côté client, c'est un langage particulièrement rapide et efficace qui ne nécessite pas d'interrogation vers le serveur. Il est apprécié pour sa simplicité, sa flexibilité et sa puissance. JavaScript, à ne pas confondre avec Java, fut créé en 1995 par la Netscape Communication Corporation. D'abord appelé LiveScript, il prit le nom de JavaScript lorsque l'entreprise s'associa avec SUN, à l'origine de Java. Il s'est parfaitement adapté aux changements du Web au fil des années. JavaScript est un langage de script populaire qui permet d'améliorer l'utilisation web, en rendant l'expérience plus dynamique et agréable. C'est d'ailleurs le seul langage de programmation natif du Web[36].

#### **IV.5.La librairie OpenCV**

OpenCV est une librairie open source, considérée comme l'outil standard pour la Computer Vision (vision par ordinateur) et le traitement d'images .

Au fil des dernières années, d'importants progrès ont été réalisés dans le domaine de l'IA (Intelligence artificielle). Toutefois, ces avancées concernent principalement le Deep Learning (l'apprentissage profond), la Computer Vision et le traitement d'image.

La Computer Vision ou vision par ordinateur est un processus permettant aux machines de comprendre le contenu des images et vidéos. Cette technologie est à la source d'innovations importantes comme les véhicules autonomes, la robotique, ou la réalité augmentée. Elle est aussi exploitée pour la lecture automatisée d'images, la détection de visages ou de formes et la reconnaissance de texte.

Le traitement d'images consiste à analyser et à manipuler les images numériques,

notamment pour améliorer leur qualité ou les modifier. Cette technique est notamment utilisée par les applications de retouche photo. L'objectif peut aussi être de les analyser pour en tirer des informations, tout comme on analyse des données statistiques ou textuelles.

Aujourd'hui, il existe un outil considéré comme standard pour le traitement d'image et la vision par ordinateur : OpenCV. Cette bibliothèque logicielle est utilisée par les plus grandes entreprises du monde entier, pour développer des applications de Computer Vision et de Deep Learning[37].



**Figure IV.6 :**OpenCV

#### **IV.5.1.Fonctionnement d'OpenCV**

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes en partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

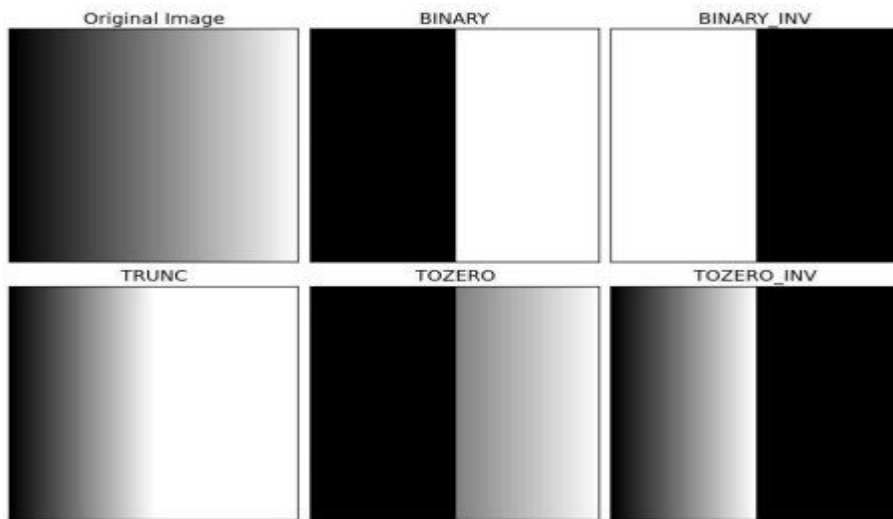
Pour comprendre le fonctionnement d'OpenCV, il faut d'abord définir ce qu'est la vision par ordinateur. Branche de l'IA, la vision par ordinateur a pour principal but de permettre à une machine d'analyser, traiter et comprendre des images prises par un système (une caméra par exemple)

OpenCV fournit un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API. Ce qui permet d'effectuer tout un tas de traitements sur des images (extraction de couleurs, détection de visages, de formes, application de filtres,...). Ces algorithmes se basent principalement sur des calculs mathématiques complexes, concernant surtout les traitements sur les matrices (car une image peut être considérée comme une matrice de pixels) [37].

#### **IV.5.2.Traitement d'image par OpenCV**

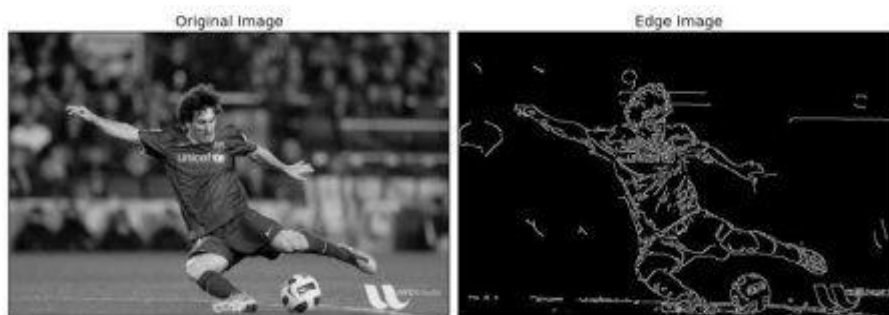
Les algorithmes d'OpenCV permettent d'appliquer divers traitements sur les images pour faciliter la détection d'éléments précis dans celles-ci. Voici quelques exemples :

- **Le Thresholding (seuillage)** : d'une image permet de définir une valeur de pixel (correspondant à une couleur) qui servira de seuil. Au-dessus ou en dessous de cette valeur (selon l'algorithme), tous les pixels se verront assigner une autre valeur. Il existe 4 algorithmes de Thresholding[29].



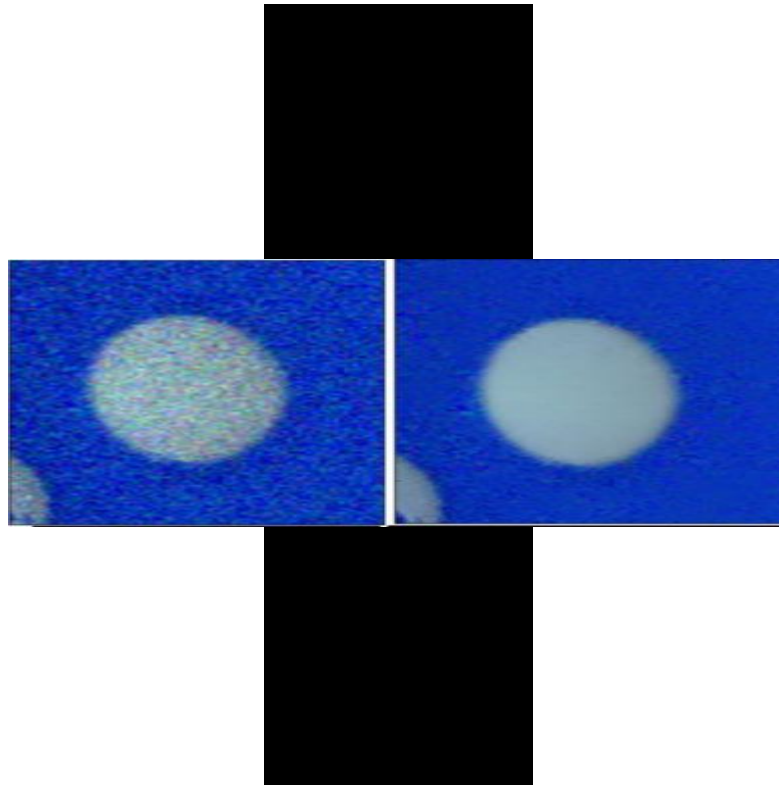
**Figure IV.7** :Les algorithmes de seuillage par OpenCV

- **Les filtres de détection de contours d'un objet sur une image (ex: Filtre de Canny)**



**Figure IV.8** :Les filtres de détection de contours par OpenCV

- **Les filtres de lissage** : permettant de réduire le bruit d'une image. Parmi eux, nous avons notamment les filtres par convolution, se basant sur les pixels voisins pour faire une moyenne.



**Figure IV.9 :**Les filtres de lissage par OpenCV

- *Les filtres morphologiques* : permettant, sur une image binaire (uniquement noire et blanche avec le fond en noir et l'objet sur lequel on effectue le traitement en blanc) d'appliquer une érosion ou dilatation, c'est-à-dire affiner l'objet pour l'érosion et l'élargir pour la dilatation.

## **IV.6. Design de contrôle et suivi des couleurs pour un robot mobile équipé d'ESP32-CAM basé sur OpenCV.js**

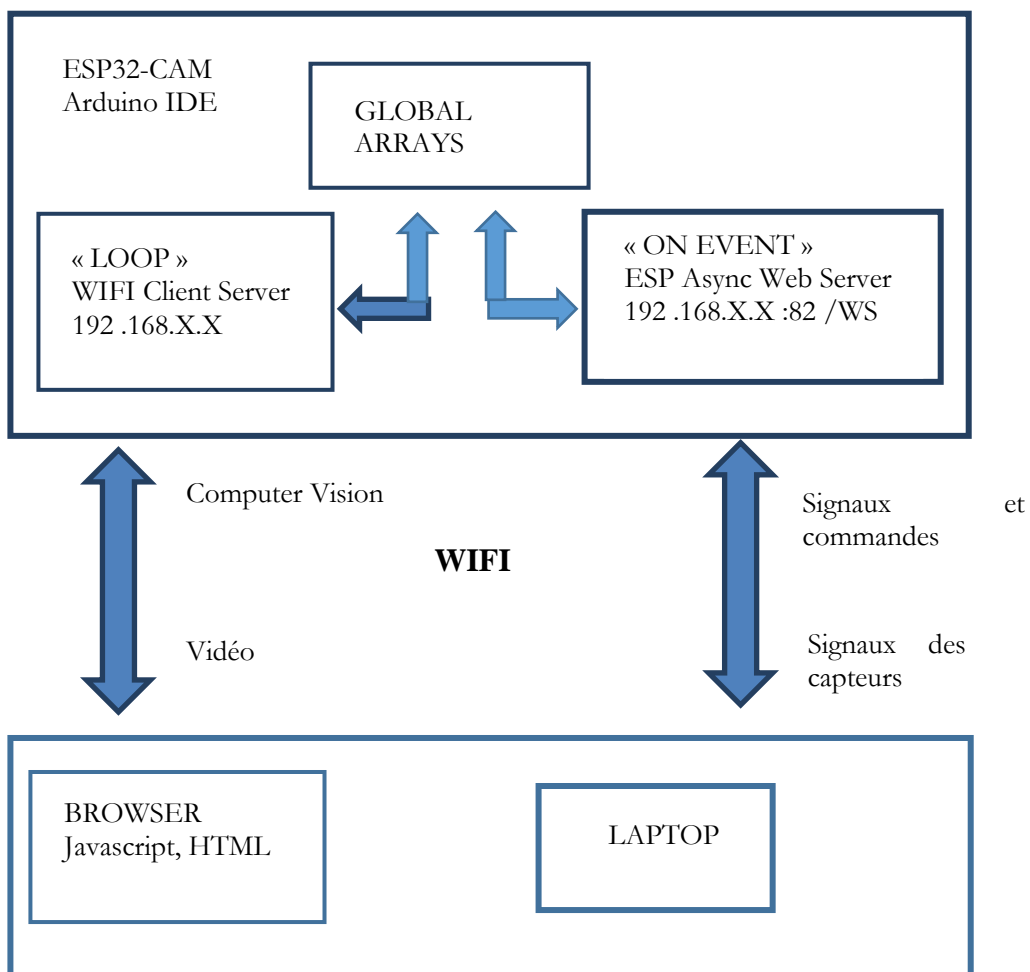
### **IV.6.1. Aperçu général sur le projet**

L'ESP32-CAM permet au client de visualiser des images fixes ou vidéo dans un navigateur à travers l'adresse IP fournie par l'ESP. Le travail qu'on va construire dans ce chapitre crée un serveur Web qui permet le suivi des couleurs d'un objet en mouvement. Sur l'interface du serveur web, on joue avec plusieurs configurations pour bien sélectionner la couleur qu'on souhaite suivre. Ensuite, le navigateur envoie les coordonnées  $x$  et  $y$  en temps réel du centre de masse de l'objet en mouvement à la carte ESP32, cette dernière peut faire mouvoir le robot de telle sorte qu'il suit l'objet en mouvement.

Dans ce travail l'ESP32 est l'extrémité (Front-End) du robot qui envoie/reçoit des données au processeur principal du robot. Ce mode, dans lequel on contrôle le robot, s'appelle le mode Téléportation. Le navigateur (browser) est le Back-end du robot.

Le mode Téléportation s'agit du mode autonome dans lequel le robot est contrôlé par des programmes. Dans ce mode, OpenCV (vision par ordinateur) est utilisé pour guider le robot. Dans notre travail on peut mesurer le contenu RGB de l'image récupérée par la caméra, et à l'aide des algorithmes de seuillage on définit les coordonnées du centre de gravité d'un objet en mouvement sur lesquels le robot peut suivre la couleur choisie auparavant.

La figure suivante présente le diagramme de navigation du robot basé sur streaming vidéo :



**Figure IV.10 :**Diagramme de Browser streaming serveur

La partie supérieure de la figure IV.10 est le matériel résidant sur le robot mobile ; Le pc portable (LAPTOP) est le contrôleur connecté avec l'ESP32 via WIFI.L'ESP32 héberge également un serveur ESP32.

La partie OpenCV.js de la page résidente dans le navigateur exécute un programme de vision par ordinateur dont les résultats sont renvoyés à l'ESP32. L'ESP32 envoie ensuite ces résultats aux actionneurs qui représentent les moteurs à courant continu du robot.

La page du navigateur (google chrome) utilise la bibliothèque OpenCV.js pour une bibliothèque limitée de fonctions OpenCV pour le traitement d'images de vision par ordinateur.

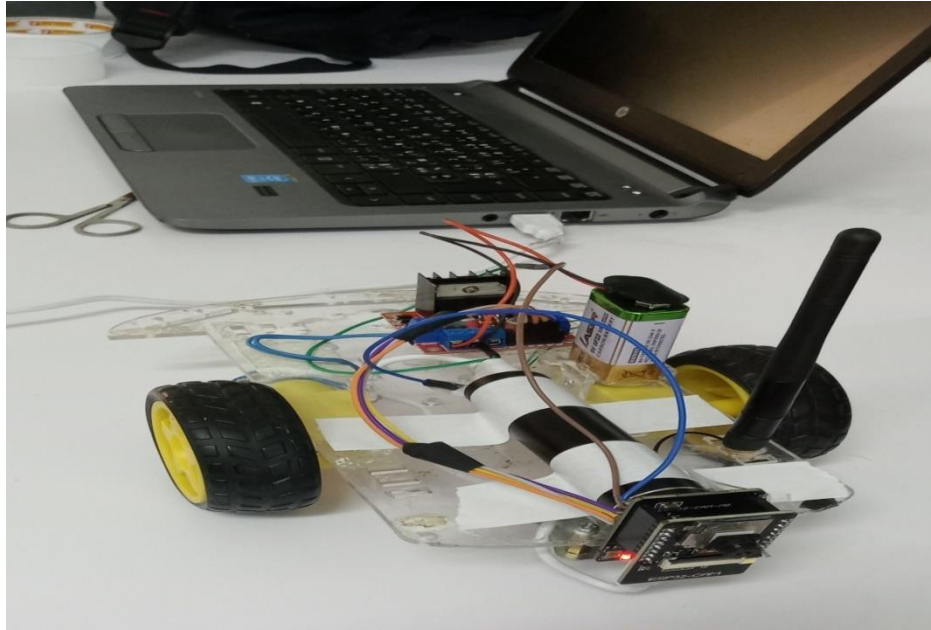
OpenCV.js est une liaison JavaScript pour un sous-ensemble sélectionné de fonctions OpenCV pour la plate-forme Web. OpenCV.js utilise Emscripten, qui est un compilateur LLVM vers JavaScript, pour compiler les fonctions OpenCV avec une bibliothèque d'API.

Les tests qui seront présentés dans ce travail sont basés sur la détection du centroïde de l'objet coloré par OpenCV classique et la transmission des coordonnées au robot mobile qui peut se diriger vers l'objet cible en fonction des coordonnées d'écran des centroïdes.

## **IV.6.2. Fonctionnement du projet**

### **IV.6.2.1. Partie matérielle (Hardware)**

La partie hardware dans ce travail est identique avec celle du chapitre III, dont on a opté une plateforme mobile avec deux moteurs à courant continu qui sont pilotés par un module L298N. Une carte ESP32-CAM pour contrôler et délivrer une vidéo streaming qui est reçue par un Laptop via WIFI. La figure ci-dessous présente le matériel utilisé dans cette partie du projet.



**Figure IV.11** :La partie hardware du projet

#### IV.6.2.2. Partie logicielle (software)

Dans la partie software on va présenter les programmes et les codes qui s'exécutent sur la carte ESP32 ainsi sur le navigateur installé sur le Laptop.

Le programme se compose de deux parties:

- ✓ Le programme serveur qui s'exécute sur l'ESP32-Cam.
- ✓ Le programme client qui s'exécute sur le navigateur.

Le programme est divisé en deux fichiers :

- le fichier :OCV\_ColorTrack\_P.ino contenant le programme serveur
- le fichier d'en-tête (Header) : index\_OCV\_ColorTrack.h contenant le programme client (HTML, CSS et JavaScript avec OpenCV.js).



**Figure IV.12** :Les deux fichiers du programme

En premier temps on introduit les informations d'identification du réseau dans ces deux lignes de code, où notre réseau est OPPO A54 :

```
// Replace with your network credentials
const char* ssid = "OPPO A54";
const char* password = "1234567890";
```

En deuxième temps on définit les broches pour le module ESP32-CAM AI-Thinker qui est compatible avec notre caméra :

```
//ANN:0
//      AI-Thinker
#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
```

Dans ce programme le serveur se trouvent des instructions d'une importance vitale, qui permettent le contrôle d'accès. Cela indique au navigateur d'autoriser l'image de la caméra et OpenCV.js, qui ont des origines différentes, à travailler ensemble dans le programme. Sans ces instructions, le navigateur Chrome génère des erreurs. Voir le code ANN1.

Finalement après la compilation du programme, on télécharge le programme sur la carte ESP32 à travers le port COM3 tout en cliquant sur le bouton Téléverser. Après le téléchargement on ouvre le moniteur série pour récupérer l'adresse IP comme indiqué sur la figure ci-dessous.

L'adresse IP récupérée est : 192.168.1.6



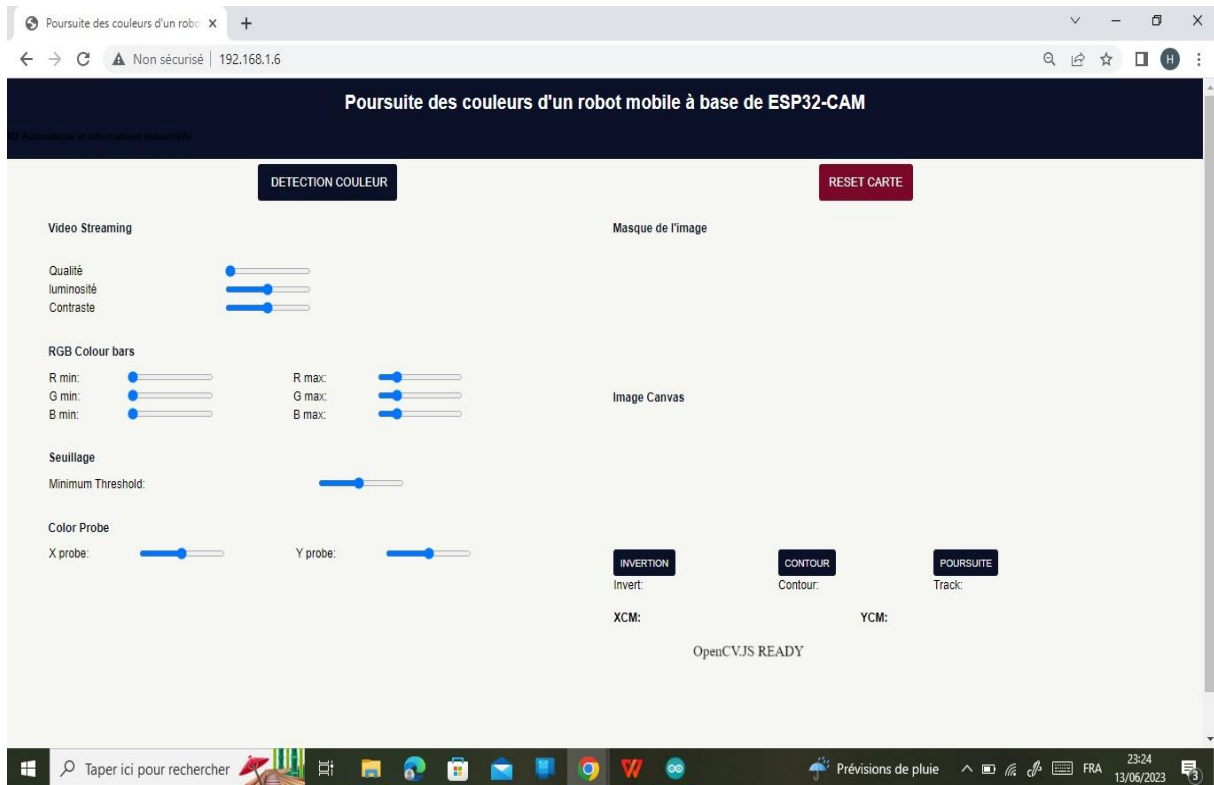
```
COM3
ESP IP Address: http://192.168.1.6
et Jun 8 2016 00:22:57

rt0x1 (POWERON_RESET,boot:0x3 (SPI_FAST_FLASH_BOOT)
configsip: 0, SWFP:0xe
clk_drv:0x0,q_drv:0x0,d_drv:0x00,cs0drv:0x00,hd_drv:0x00wp_drv:0x00
mode:DIO, lock iv:1
load:0x3fff0018,ln:4
load:0x3fff000c,len:1216
ho 0 tail 12room 4
load:0x40080000,ln:10944
load:0x40080400,len:6388
entr 0x400806b4
D@ESP IP Address: http://192.168.1.6
```

**Figure IV.13** :Adresse IP récupérée dans moniteur série

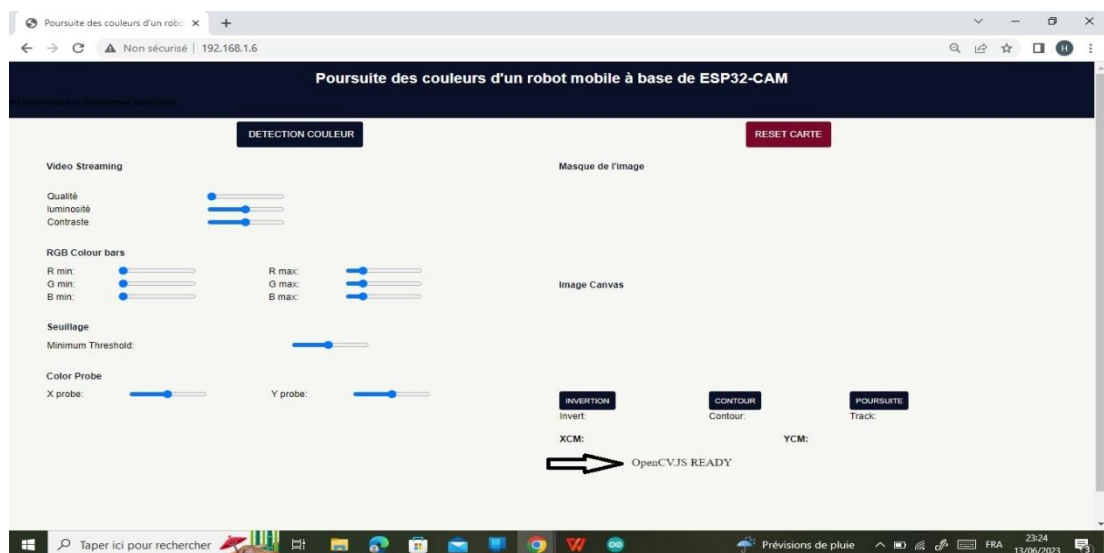
Après l'introduction de l'adresse IP récupérée sur le navigateur on aura l'interface présenté sur la figure (IV.14). Les onglets de la page web obtenue sont :

- **Détection des couleurs** : nous a permis de voir la vidéo streaming
- **Reset carte** : redémarre la carte ESP32
- **Inversion** : on inverse la binarisation d'une image binaire
- **Contour** : affiche les contours de l'image
- **Poursuite** : donne les coordonnées x, y du centre de gravité d'un objet mobile.



**Figure IV.14 :** Interface sur le web

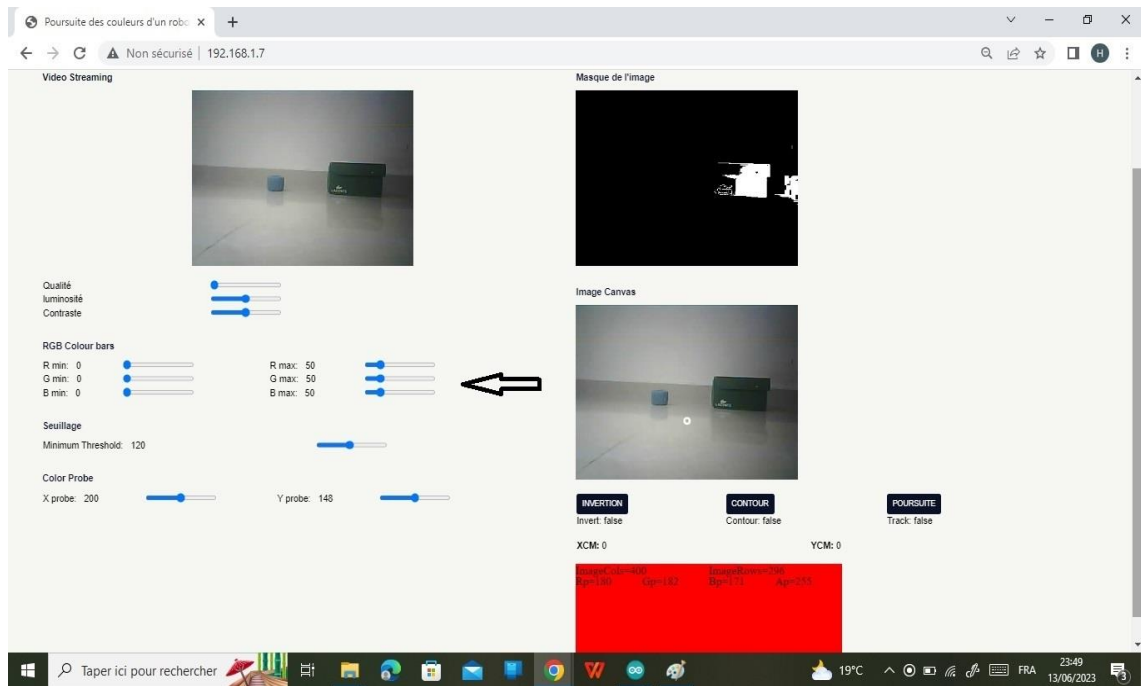
Avant lancer la détection des couleurs il faut initialement confirmer que le module qui signale que L'OpenCV.js a été initialisé comme indiqué sur la figure (IV.15) où il est mentionné par la flèche. Une fois l'initialisation terminée, nous pouvons cliquer sur le bouton Détection des couleurs.



**Figure IV.15 :** Initialisation de l'OpenCV.JS

La flèche mentionnée sur la figure (IV.16) présente le niveau de couleurs (RGB) de chaque couleur dans l'image. Ceux-ci sont utilisés pour fixer des limites à la gamme de couleurs autorisée dans l'image "traitée". Où ils sont réglables de telle sorte on obtient que la couleur ciblée en blanc et le reste en noir.

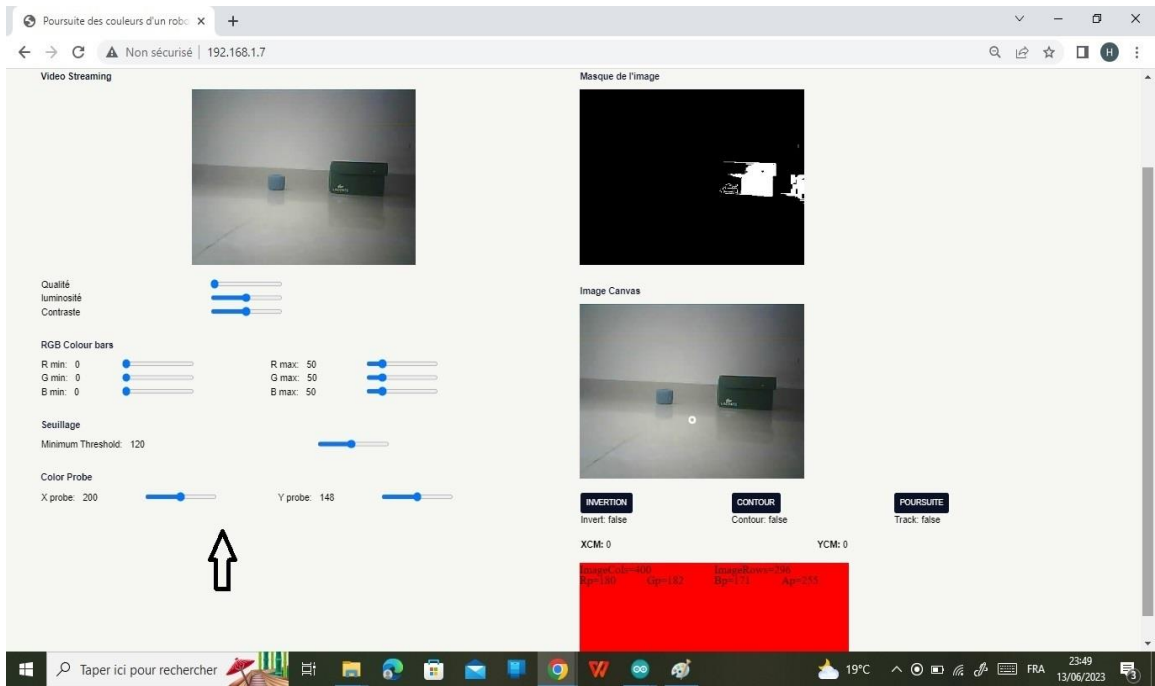
Les valeurs maximales et minimales de rouge, vert et bleu (RGB) sont appliquées à l'OpenCV, utilisant la fonction **inRange()**.



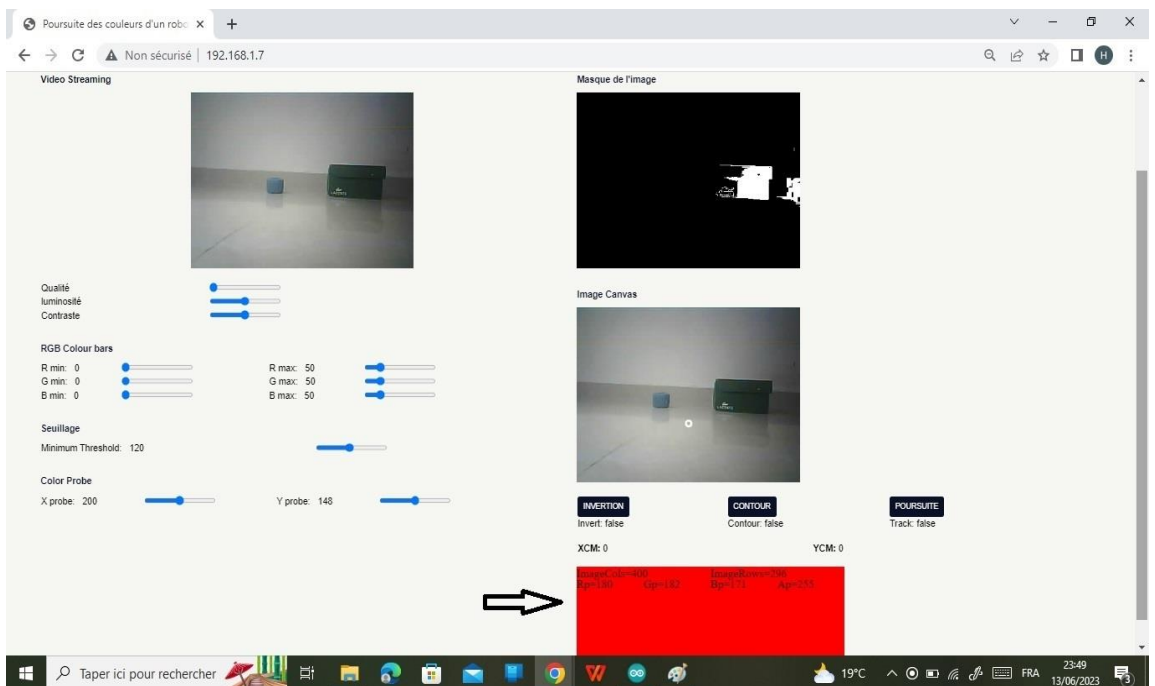
**Figure IV.16 :**Piste de couleur RVB (niveau de couleurs)

L'image est à 4 canaux ; RGBA où A est le niveau de transparence. Dans notre travail, A sera fixé à 100% d'opacité, soit 255. Le code est basé sur le fait que, outre le plan A, l'image possède 3 plans de couleur, RGB, chaque pixel de chaque plan ayant une valeur comprise entre 0 et 255. Les limites haute/basse sont appliquées aux plans de couleur correspondants pour chaque pixel.

Dans l'image canevas placée dans la partie inférieure droite de la page web, un cercle blanc est remarqué. Ce cercle est positionné par les coordonnées Xprobe et Yprobe qui sont mentionnées par la flèche dans la figure (IV.17). Le cercle blanc est placé sur la couleur ciblée afin qu'on puisse obtenir les valeurs des couleurs (RVB). Les valeurs des couleurs RVB sont données sur le rectangle rouge comme indiqué par flèche sur la figure (IV.18).



**Figure IV.17** :Sonde de couleur (probe couleur)

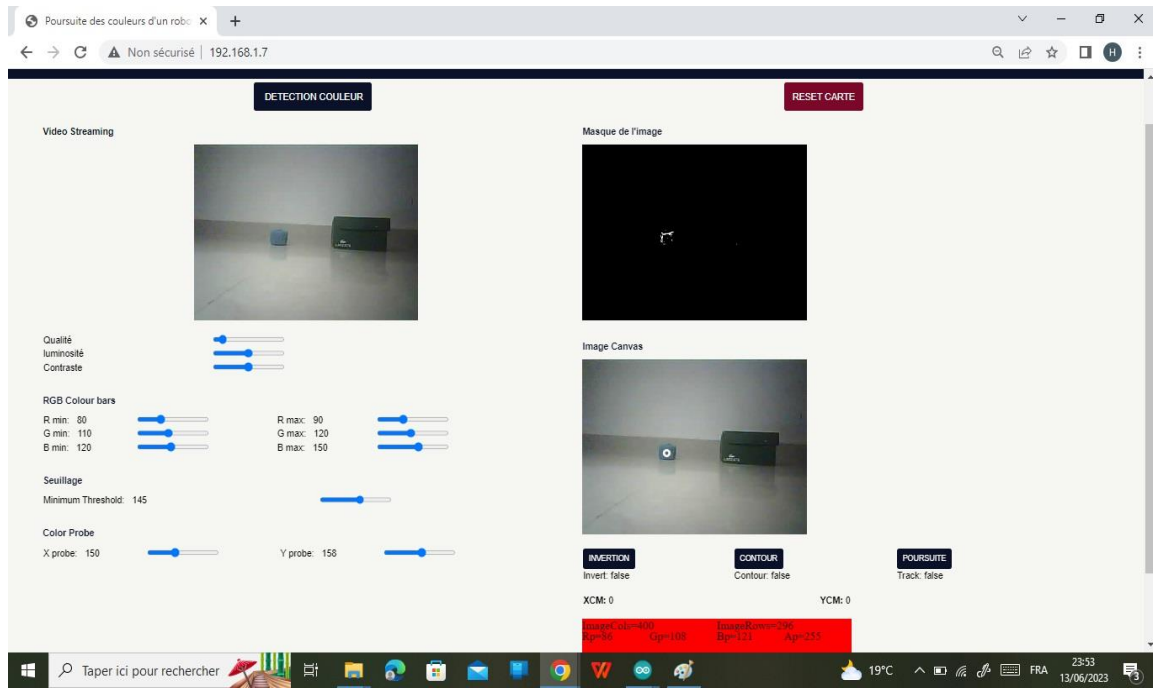


**Figure IV.18** :les valeurs des couleurs (niveau de couleurs) pour la couleur ciblée

Une fois les valeurs des couleurs RVB sont obtenues, on ajuste les valeurs de chaque couleur sur les bars RGB (entre 0 et 255). Lorsque les valeurs optimales pour une couleur cible souhaitée sont trouvées à l'aide de la sonde X, Y et définies par la barre des couleurs, la cible dans l'image binaire est blanche et le reste de l'image est noir, idéalement, comme indiqué dans l'image du haut dans la colonne de droite. Les résultats idéals ne peuvent

généralement être réalisés que lorsque les conditions d'éclairage peuvent être étroitement contrôlées. L'éclairage intérieur standard doit être acceptable.

Les contours sont trouvés à l'aide de la fonction *findContours*, où est un tableau de contours du ou des objets blancs trouvés dans l'image binaire.

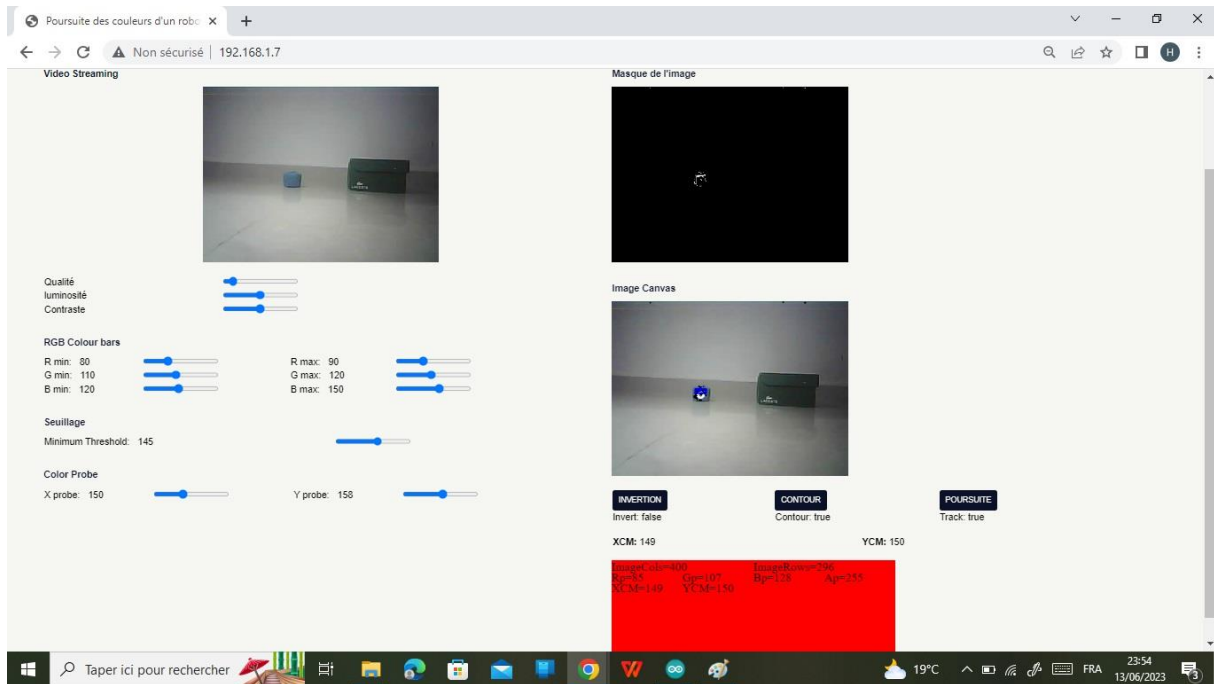


**Figure IV.19:**contours de la cible

Une fois les contours sont définis, on définit  $M_{00}$  qui est le moment zéro de la "zone" délimitée par un contour. Dans OpenCv, c'est en fait le nombre de pixels entourés par le contour.  $M_{10}$  et  $M_{01}$  sont le nombre pondéré de coordonnées  $x$  et  $y$  de pixels inclus.

L'origine du système de coordonnées  $x,y$  se trouve dans le coin supérieur gauche de l'image.  $X$  est positif horizontal vers la droite et  $Y$  est positif vertical vers le bas. Par conséquent,  $M_{10}/M_{00}$  et  $M_{01}/M_{00}$  sont les coordonnées  $x,y$  du centroïde d'un contour.

Finalement les coordonnées  $x, y$  du centre de la cible sont définies tout on cliquant sur le Boton **poursuite**(voir le rectangle rouge bas droite) figure IV.20, on peut contrôler le robot de telle sorte il suit l'objet mobile avec ces coordonnées qui sont variées en fonction de la mobilité de l'objet.



**Figure IV.20** :suivi de la couleur

### IV.6.2.3. Les algorithmes de poursuite par le robot mobile

Le robot mobile est commandé par la carte ESP32-CAM une fois il reçoit les valeurs  $x$ ,  $y$  du centre de gravité de l'objet mobile selon l'organigramme IV.1.

L'image prise dans nos tests est de dimensions  $400 \times 400$ , alors les coordonnées du centre de l'image sont définies par :  $x_c=200$ ,  $y_c=200$ . Afin que le robot puisse suivre la couleur ciblée, on a introduit quatre fonctions dans le programme qui sont :

- Void forward() : bouger vers l'avant
- Void stop() : arrêter
- Void right() : tourner à droite
- Void left() : : tourner à gauche

Les fonctions forward et stop sont basées sur des sorties numériques, mais les fonctions right et left sont basées sur des sorties MLI (PWM). Les sorties PWM pour une carte ESP32 se diffèrent par rapport une carte Arduino , où pour une carte Arduino l'utilité est simple.

Pour programmer un pin d'une carte ESP32 comme sortie PWM il faut suivre les étapes suivantes :

- Choisir un canal PWM (entre 0 et 15)

- Choisir la fréquence du PWM
- Choisir la résolution de la largeur d'impulsion entre 1 et 16 bits
- Choisir le PIN GPIO qui va générer le signal PWM
- Attribuer la valeur de la tension que l'on veut en sortie

Pour programmer les Sorties PWM sur Arduino IDE on suit la syntaxe suivante :

```

int pwmChannel = 0; //Choisit le canal 0
int frequence = 1000; //Fréquence PWM de 1 KHz comme exemple
int resolution = 8; // Résolution de 8 bits, 256 valeurs possibles
int pwmPin = 15;

void setup(){
  // Configuration du canal 0 avec la fréquence et la résolution
  choisie
  ledcSetup(pwmChannel, frequence, resolution);

  // Assigne le canal PWM au pin 15
  ledcAttachPin(pwmPin, pwmChannel);

  // Créer la tension en sortie choisi
  ledcWrite(pwmChannel, 64); //1V
}

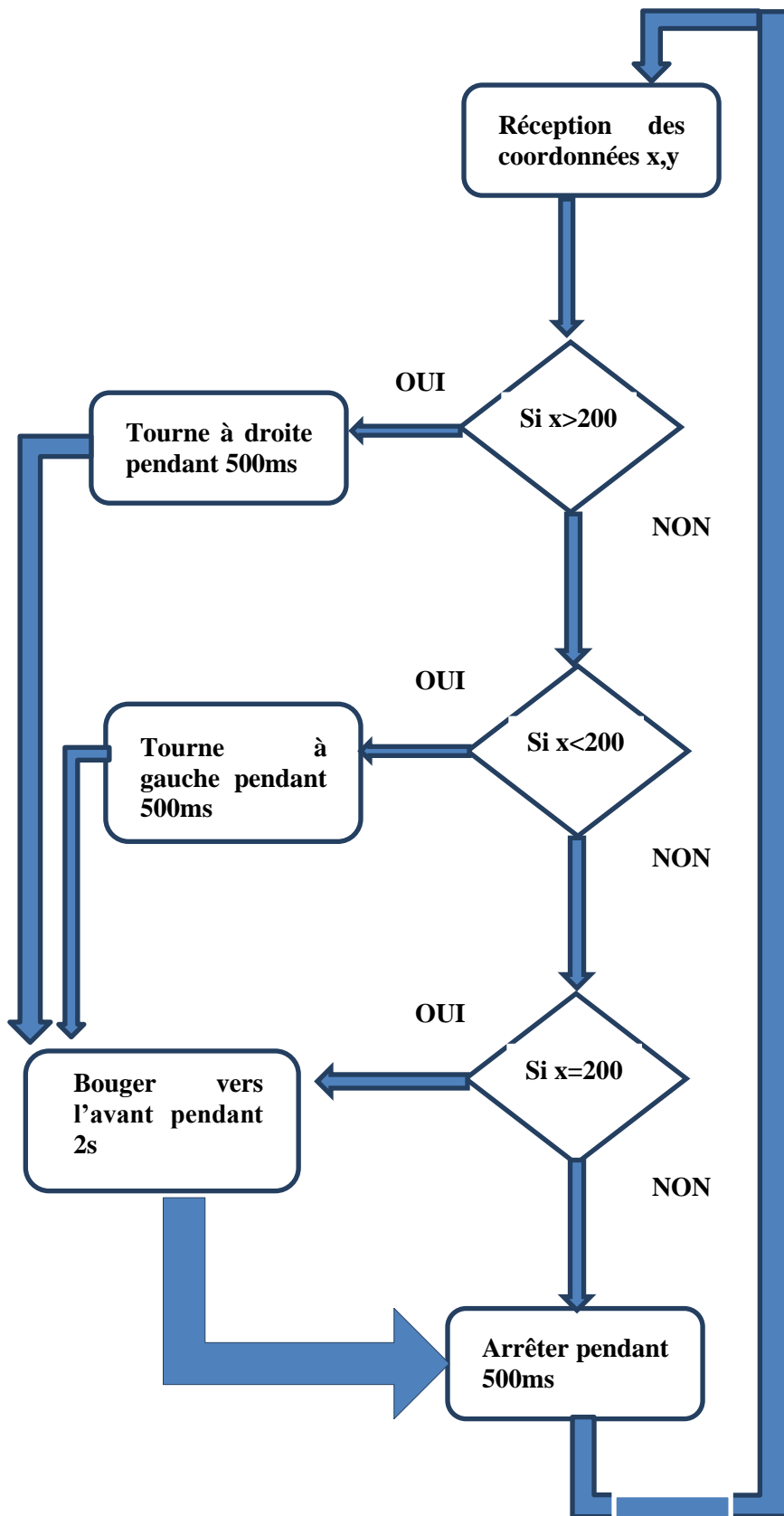
void loop(){
}

```

A partir de ce programme, on peut générer des commandes PWM sur les Pins 13, 14, 15, 12. Ces commandes nous a permis de contrôler les moteurs CC du robot sans perdre la cible.

Les tests faits montrent que notre robot suit la cible en mouvement sans la perdre par rapport aux tests qui sont faits en basant sur des commandes numériques (TOR).

Les résultats obtenus sont présentés avec des vidéos.



**Organigramme IV.1 :** Algorithme de contrôle et suivi des couleurs par le robot mobile

## **IV.7. Conclusion**

Dans ce chapitre on a présenté le suivi des couleurs par un robot mobile basant sur le serveur Web de caméra ESP32 et OpenCV. La caméra ESP32, avec sa petite taille, sa capacité Wi-Fi, sa haute technologie et son faible coût, promet d'être une nouvelle capacité de capture d'image frontale intéressante pour les applications de serveur Web OpenCV.

Dans ce chapitre on a prouvé l'efficacité des algorithmes de traitement d'image basés sur l'OpenCV en termes de détection des cibles mobiles. Les commandes PWM délivrées par la carte ESP32 permettent au robot mobile de suivre les objets mobiles et ne les perdent pas par le contrôle des vitesses du robot.

# CONCLUSION GENERALE

Dans notre projet, nous avons conçu et réalisé un robot mobile à base d'une carte ESP 32-CAM. Ce robot est contrôlé à distance via WIFI, où la carte ESP32 émet un streaming vidéo à partir d'un web serveur afin qu'on puisse visualiser et surveiller à distance et en temps réel notre robot.

La première partie de ce mémoire est consacrée à donner une vision générale sur la robotique et les différents types des robots mobiles pour qu'on puisse bien choisir le type de notre robot.

Dans la seconde partie, nous avons présenté le matériel et les différents composants électroniques nécessaires dans notre système, où on a donné un aperçu général sur la carte ESP32 ainsi la carte ESP32-CAM et aussi donné une description des moteurs et le driver L298N.

En troisième partie, nous avons construit un robot télécommandé et contrôlé à l'aide de l'ESP32-CAM et un serveur Web. L'ESP32-CAM nous a permis à contrôler les moteurs à courant continu qui font mouvoir le robot et le contrôler en différentes directions.

La dernière partie de ce mémoire est consacré à la conception de contrôle d'un robot mobile pour le suivi des couleurs basant sur le serveur Web de caméra ESP32 et OpenCV.

De manière générale, ce travail nous a fait découvrir le monde de la robotique en général, spécialement la robotique mobile et ses perspectives. Ainsi la télé opération et le contrôle à distance de la robotique à partir des vidéos récupérées de la carte ESP32-CAM. Dans ce projet, nous avons présenté un travail qui montre comment utiliser les serveurs de streaming ESP32/ESP32-CAM dans des systèmes robotiques utilisant OpenCV.js.

En réalisant ce projet, on a pris conscience de l'existence des imperfections, donc on a pensé à des perspectives d'améliorations et d'autres idées permettant de l'améliorer. On donne comme perspectives :

- ✓ Assurer une communication I2C avec une carte Arduino Méga afin d'assurer des commandes supplémentaires pour le robot et réserver la carte ESP32 pour d'autres applications.
- ✓ Introduire un afficheur OLED pour afficher l'adresse IP sans faire revenir au

moniteur série.

- ✓ Introduire des techniques de l'intelligence artificielle pour des applications sur la détection des objets mobiles et les suivre.
- ✓ Contrôler le robot à partir de n'importe quel endroit par appliquant la technologie internet des choses (IOT).

# Bibliographie

- [1] Siegwart, R., & Nourbakhsh, I. (2006). Introduction aux robots mobiles autonomes. Presses polytechniques et universitaires romandes.
- [2] "Vision par ordinateur et robotique mobile" par Roland Chapuis, Luc Jaulin et Daniel Pucci (2006).
- [3] Giovanni Adorni, Stefano Cagnoni, Stefan Anderle, Gerhard K. Kraetzschmar, Monica Mordonini, Michael Plagge, Marcus Ritter, Stefan Sablatnög and Andreas Zell, 'Vision-based localization for mobile robots'. Journal of Robotics and Autonomous Systems, n° 36, pp. 103-119, Elsevier Science Publishers, 2001.
- [4] Fraichard, T. et Asama, H. (2006). Robotique mobile : bases fondamentales, méthodes et perspectives. Éditions Eyrolles.
- [5] Siegwart, R., Nourbakhsh, I. R., et Scaramuzza, D. (2011). Introduction à la robotique. Éditions Dunod.
- [6] Martinet, P. et Alami, R. (2008). La robotique mobile autonome. Éditions Presses Polytechniques et Universitaires Romandes.
- [7] Gouttefarde, M., Krut, S., et Pierrot, F. (2017). Mécanique des robots mobiles à chenilles. Éditions Springer.
- [8] Hugues, J. (2004). La robotique mobile. Éditions Dunod.
- [9] Bonnet, M. et Hugues, J. (2015). Introduction à la robotique mobile. Éditions Ellipses.
- [10] "ESP32 : Programmation avancée" par Charles Bell.
- [11] "L'ESP32 en pratique : Domotique, objets connectés, robotique" par Olivier Hinfray.
- [12] "L'ESP32 pour les objets connectés : WiFi et Bluetooth Low Energy" par Pierre Ficheux.
- [13] "Programmer l'ESP32-CAM avec Arduino" par Olivier Hinfray.
- [14] "ESP32-CAM : Guide de mise en œuvre" par Lionel Yvan Fosso.
- [15] "Getting Started with Arduino" par Massimo.
- [16] "Arduino Cookbook" par Michael Margolis.
- [17] "Exploring Arduino: Tools and Techniques for Engineering Wizardry" par Jeremy Blum.
- [18] "Introduction to Autonomous Mobile Robots" par Roland Siegwart et Illah Nourbakhsh.
- [19] "The I2C Bus: From Theory to Practice" par Dominique Paret.

- [20] "SPI Master and Slave Operation: Understanding SPI Protocol and Implementing SPI Communication" par Ankit Patel.
- [21] "Serial Communications: A Comprehensive Guide" par Mark E. Hazelton.
- [22] "Comprendre l'électronique : 2e édition" par Michel Bréal.
- [23] "Arduino pour les Nuls" par John Nussey.
- [24] "Tout savoir sur les batteries externes" par Tom's Guide.
- [25] McRoberts, Michael. "Beginning Arduino." Apress, 2010.
- [26] Dubois, Thibaut. "Créez des sites web avec HTML5 et CSS3." Editions Eyrolles, 2019.
- [27] "Traitement d'images et de vidéos" par Philippe Robert et Olivier Lezoray (2011).
- [28] "Introduction au traitement d'images" par Pierre Soille.
- [29] "Traitement d'images : Algorithmes et applications" par Philippe Bolon.
- [30] "Traitement du signal et des images : Méthodes et applications" par Alain Dieterlen.
- [31] N. OTSU. "A Threshold Selection Method from Gray-Level Histograms". IEEE transactions on Systems, Man and Cybernetics, 1979, vol. 9 (1), 62-66.
- [32] W. Niblack. "An Introduction to Digital Image Processing", pages 115–116. Prentice Hall, 1986.
- [33] J. Bernsen, "Dynamic thresholding of grey-level images", in Proc. 8th Int'l conf. on Pattern Recognition, Paris, France, 1986, pp. 1251-1255.
- [34] "Analyse d'images et vision par ordinateur" par Jean-Michel Jolion.
- [35] "HTML5 : Maîtrisez l'avenir du Web" par Raphaël Goetter.
- [36] "Eloquent JavaScript : Une introduction moderne à la programmation" par Marijn Haverbeke.
- [37] "OpenCV 4 par la pratique : Découvrez le traitement d'images et la vision par ordinateur" par Alexandre Boucaud et Benjamin Fauqueur.
- [38] "Les moteurs à courant continu" par Michel Peltier.