



République Tunisienne  
Ministère de l'Enseignement Supérieur,  
de la Recherche Scientifique

\*\*\*\*\*

Direction Générale des Etudes  
Technologiques



Institut Supérieur des  
Etudes Technologiques  
de Mahdia

الجمهورية  
التونسية

وزَرَّةُ التَّعْلِيمِ الْعُالَمِيِّ  
وَالْبَحْثِ الْعُالَمِيِّ

\*\*\*\*\*

الإدارة العامة للدراسات التكنولوجية

\*\*\*\*\*

المعهد العالي للدراسات التكنولوجية بالمهديّة

Code du projet :  
N°46-2021-2022

# PROJET

## DE FIN D'ETUDES

PRÉSENTE POUR OBTENIR LE TITRE :

DIPLÔME NATIONAL DE LICENCE

En Génie Electrique

Parcours :

**Thème du projet  
« Réalisation d'un robot de sécurité  
autonome ».**

Réalisé par : **MOHAMED AMINE BEN AMMAR**

SOUTENU LE **24-6-2022** DEVANT LE JURY D'EXAMEN :

**M./Mme WASSIM SINENE** Président  
**M./Mme OMAR ZARAI** Rapporteur  
**M./Mme ASMA CHAOUCH /FEHMI CHATMEN** Encadrant-  
ISET  
**M./Mme AFEF AYADI** Encadrant-Entreprise

A.U. : 2021-2022

## Dédicaces

*Je dédie ce modeste travail et ma profonde gratitude :*

*A mon très cher père , qui est toujours à mes côtés pour me soutenir et m'encourager.*

*Que ce travail traduit ma gratitude et mon affection.*

*A ma maman , qui m'a soutenu et encouragé durant ces années d'études.*

*Qu'elle trouve ici le témoignage de ma profonde reconnaissance.*

*Aux âmes de mes deux grands-pères , qui m'ont appris leurs expériences de vie.*

*A toute ma famille qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.*

*A tous mes amis qui m'ont toujours encouragées et supportées.*

*A tous ceux que j'aime.*

*Merci !*

# Remerciements

Au terme de ces quatre mois de stage, nous voulons adresser tous nos remerciements aux personnes qui ont participé de près ou de loin à la réalisation de ce projet.

Nous tenons à remercier dans un premier temps, nous remercions tout d'abord nos chers professeurs et encadrants **Mme.Asma Chaouch et M.Mohamed Fehmi Chatmen** pour leurs suivi et pour leurs énorme soutien, qu'ils n'ont cessé de nous prodiguer tout au long de la période du projet.

Nous exprimions également nos profondes gratitude et nous tenions à remercier **Mme.Afef Ayedi** de nous avoir accordé à l'opportunité de passer notre stage de fin d'études au sein de l'entreprise **ASTRE engineering** et pour sa aide formatif.

Nous remercions aussi les sociétés : **AMIRAL DECOR, STE NAJIA DE DISTRIBUTION ET DE SERVICE, BEST BUY, TIM Tôlerie, SFAX LAZER**, pour nous sponsoriser et rendre ce projet réalisable.

Nos vifs remerciements s'adressent également aux membres du jury **M.Wassim Sinene** et **M.Omar Zarai** qui ont accepté d'évaluer notre travail.

Enfin nous espérons que ce travail sera à la hauteur et pourra répondre aux attentes et exigences auxquels il a été destiné.



# Sommaire

<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>CHAPITRE I CADRE DE PROJET .....</b>	<b>2</b>
I. INTRODUCTION.....	3
II. PRÉSENTATION DE L'ENTREPRISE .....	3
1. <i>La fiche d'identité de la société .....</i>	3
2. <i>Organigramme .....</i>	4
3. <i>Clientèles et fournisseurs de l'entreprise .....</i>	4
a. Clientèles.....	4
b. Partenariat .....	5
III. ETUDE BIBLIOGRAPHIE.....	5
1. <i>Définition .....</i>	5
2. <i>Domaine ciblée .....</i>	5
3. <i>Les robots de surveillance existants .....</i>	6
a. Le robot PGuard .....	6
b. Le robot Jack .....	6
c. Le robot Picard .....	7
d. Le robot E-vigilante .....	8
IV. CAHIER DE CHARGE.....	8
1. <i>Problématique .....</i>	8
2. <i>Objective.....</i>	9
3. <i>Travail demandé.....</i>	9
V. CONCLUSION.....	9
<b>CHAPITRE II .....</b>	<b>10</b>
<b>ETUDE THEORIQUE.....</b>	<b>10</b>
<b>CHAPITRE 2 : ETUDE THEORIQUE.....</b>	<b>11</b>
I. INTRODUCTION.....	11
II. PRÉSENTATION DE LA SOLUTION .....	11
1. <i>Principe du fonction du système.....</i>	11
a. La surveillance intelligente .....	11
b. La navigation du robot .....	11
c. Contrôle et supervision à distance .....	11

2.	<i>Structure du système :</i>	12
III.	PRESENTATION DES DIFFERENTS BLOCS ET CHOIX DU MATERIELS ADEQUATS.....	14
1.	<i>Le bloc de traitement et de contrôle :</i>	14
1.1.	<i>Les nano ordinateurs .....</i>	14
a.	Généralité sur les nano ordinateur .....	14
b.	Choix du nano ordinateur .....	16
c.	Présentation du Raspberry Pi 4 (4 Go RAM).....	18
1.2.	<i>Les microcontrôleurs.....</i>	20
a.	Généralité sur les microcontrôleurs.....	20
b.	Choix du microcontrôleur .....	21
c.	Présentation de la carte Arduino UNO.....	23
2.	<i>Le bloc « distance des obstacles » .....</i>	24
a.	Choix du capteur de détection des obstacles.....	24
b.	Présentation du capteur ultrason choisi .....	25
3.	<i>Le bloc « vidéo en direct » .....</i>	25
4.	<i>Le bloc « intensité du lumière ».....</i>	26
5.	<i>Le bloc « support mobile ».....</i>	27
a.	Généralité sur les servomoteurs : .....	27
b.	Le servomoteur SG90 .....	27
6.	<i>Le bloc « refroidisseur d'unité ».....</i>	28
7.	<i>Le bloc « éclairage du milieu » .....</i>	29
8.	<i>Le bloc « signal d'alarme » .....</i>	29
9.	<i>Le bloc « commande des moteurs » .....</i>	29
9.1.	<i>Les driver moteur .....</i>	30
a.	Généralité sur les drivers moteur.....	30
b.	Le driver moteur L298N .....	30
9.2.	<i>Les moteurs à courant continue.....</i>	32
a.	Généralité sur le moteur à courant continu.....	32
b.	Choix des moteurs.....	32
c.	Le moteur jga25-370 .....	34
10.	<i>Le bloc « alimentation » .....</i>	35
a.	Généralité sur les batteries .....	35
b.	Choix du batterie 12 V.....	35
c.	Batterie Li-po 3S 3000 mAh.....	35
d.	Choix du batterie 5 V.....	36
e.	Batterie Li-ion 10000 mAh .....	36
11.	<i>Le bloc « contrôle/ visualisation à distance » .....</i>	37
IV.	OUTIL DE DEVELOPPEMENT .....	37
1.	<i>Fritzing.....</i>	37
2.	<i>Visual Studio .....</i>	38

3. <i>Arduino IDE</i> .....	38
4. <i>Solidworks</i> .....	39
V.    CONCLUSION.....	39
<b>CHAPITRE III ETUDE PRATIQUE.....</b>	<b>40</b>
I.    INTRODUCTION.....	41
II.   CABLAGE DU SCHEMA ELECTRIQUE.....	41
1. <i>Présentation du schéma complet</i> .....	41
2. <i>Raspberry Pi 4</i> .....	42
3. <i>Arduino UNO</i> .....	44
4. <i>Système d'alimentation</i> .....	44
5. <i>Système de détection d'obstacle</i> .....	45
6. <i>Système de mouvement</i> .....	46
7. <i>Système du camera</i> .....	46
8. <i>Système d'alarme</i> .....	47
9. <i>Système d'éclairage</i> .....	47
III.  INSTALLATION ET CONFIGURATION DU SYSTEME.....	48
1. <i>Installation du logiciel utilisée dans le pc</i> .....	48
a.    PuTTY.....	48
b.    VNC Viewer .....	48
c.    Raspberry Pi Imager .....	49
2. <i>Installation et configuration du système d'exploitation</i> .....	50
a.    Installation du système d'exploitation .....	50
b.    Configuration du système d'exploitation .....	53
3. <i>Installation des logiciel et bibliothèques utilisé dans le robot</i> .....	55
a.    Installation du Python 3 et les paquets nécessaires.....	55
b.    Installation du Open CV et ses bibliothèques.....	57
IV.   CONCLUSION.....	61
<b>CHAPITRE 4 CONCEPTION ET REALISATION .....</b>	<b>62</b>
I.    INTRODUCTION.....	63
II.   CONCEPTION LOGICIELLE.....	63
1. <i>Description du fonctionnement du système</i> .....	63
a.    Démarrage et configuration .....	65
b.    Détection des personnes.....	65
c.    Déclenchement du système d'alarme .....	69
d.    Système d'éclairage d'intelligent .....	69
e.    Visualisation et contrôle par l'application mobile .....	70
f.    Contrôle du servomoteur.....	71
g.    Contrôle du moteur.....	72

h. Evitement d'obstacle.....	73
2. <i>Description du fonctionnement d'application web</i> .....	74
III. CONCEPTION MECANIQUE.....	75
IV. LES ETAPES DE REALISATION DU ROBOT .....	76
1. <i>Etage 0</i> .....	76
2. <i>Etage 1</i> .....	77
3. <i>Etage 2</i> .....	78
4. <i>Etage 3</i> .....	78
V. CONCLUSION.....	80
<b>CONCLUSION GENERALE .....</b>	<b>81</b>
<b>NETOGRAPHIE .....</b>	<b>82</b>
<b>ANNEXES.....</b>	<b>83</b>

# Liste des tableaux

<i>Tableau 1 : Informations principaux sur ASTRE engineering .....</i>	3
<i>Tableau 2 : services réalisés par l'entreprise.....</i>	4
<i>Tableau 3 : partenaires de l'entreprise .....</i>	5
<i>Tableau 4 : les entrées et les sorties à connecter avec le nano ordinateur.....</i>	17
<i>Tableau 5 : comparaison entre Raspberry Pi 4 et NVIDIA Jetson.....</i>	17
<i>Tableau 6 : les entrées et les sorties à connecter avec le microprocesseur .....</i>	22
<i>Tableau 7 : comparaison entre différents microcontrôleurs.....</i>	22
<i>Tableau 8 : comparaison entre capteur ultrason et Lidar.....</i>	24
<i>Tableau 9 : consommation du courant .....</i>	36
<i>Tableau 10 : légende du montage électrique du système.....</i>	41
<i>Tableau 11: parameters de detectMultiScale .....</i>	68
<i>Tableau 12 : table de vérité de la fonction du port en H.....</i>	72

# Liste des figures

<i>Figure 1 : organigramme de la société.....</i>	4
<i>Figure 2 : la robot PGuard.....</i>	6
<i>Figure 3 : le robot Jack .....</i>	7
<i>Figure 4 : la robot Picard.....</i>	7
<i>Figure 5 : le robot E-vigilante .....</i>	8
<i>Figure 6 : Schéma synoptique des blocs du système .....</i>	12
<i>Figure 7 : schéma synoptique élémentaire du système .....</i>	13
<i>Figure 8 : photo du nano ordinateur Raspberry Pi 4 .....</i>	18
<i>Figure 9 : brochage du Raspberry Pi 4 .....</i>	19
<i>Figure 10 : architecture interne d'un microcontrôleur.....</i>	21
<i>Figure 11 : photo du carte Arduino UNO.....</i>	23
<i>Figure 12 : brochage de l'Arduino UNO .....</i>	24
<i>Figure 13 : photo du capteur ultrason utilisé .....</i>	25
<i>Figure 14 : photo du RPi camera V2.....</i>	26
<i>Figure 15 : photo du capteur LDR.....</i>	26
<i>Figure 16 : les constituants élémentaires d'un servo moteur .....</i>	27
<i>Figure 17 : photo du servomoteur SG 90.....</i>	28
<i>Figure 18 : radiateur du Raspberry Pi avec les ventilateurs .....</i>	28
<i>Figure 19 : photo du LED utilisé.....</i>	29
<i>Figure 20 : photo de l'avertisseur sonore à utiliser .....</i>	29
<i>Figure 21 : circuit d'un port en H.....</i>	30
<i>Figure 22 : L289N et ses composants.....</i>	31
<i>Figure 23 : force du poids sur un plan incliné.....</i>	32
<i>Figure 24 : force de mouvement et couple.....</i>	33
<i>Figure 25 : force d'accélération.....</i>	33
<i>Figure 26 : photo du moteur jga25-370 .....</i>	34
<i>Figure 27 : photo du batterie Li-po.....</i>	35
<i>Figure 28 : photo du batterie Li_ion .....</i>	37
<i>Figure 29 : interface fritzing .....</i>	38
<i>Figure 30 : interface Microsoft Visual Studio .....</i>	38
<i>Figure 31 : interface Arduino IDE .....</i>	38
<i>Figure 32 : interface solidworks .....</i>	39

Figure 33 : montage électrique complet du système .....	42
Figure 34 : alimentation du Raspberry Pi 4 .....	43
Figure 35 : connexion du camera avec la carte .....	43
Figure 36 : schéma des E /S avec Raspberry Pi .....	43
Figure 37 : port USB du carte Arduino UNO .....	44
Figure 38 : schéma de câblage du carte Arduino UNO .....	44
Figure 39 : schéma du câblage du batterie 12V.....	45
Figure 40 : schéma de câblage du US-100 .....	45
Figure 41 : schéma de câblage du SRF05 .....	45
Figure 42 : schéma du système du mouvement .....	46
Figure 43 : installation du caméra dans la carte .....	46
Figure 44 : schéma du câblage des servomoteurs .....	47
Figure 45 : schéma du câblage de l'avertisseur sonore .....	47
Figure 46 : schéma de câblage du système d'éclairage .....	47
Figure 47 : fenêtre d'accueil du PuTTY .....	48
Figure 48 : fenêtre d'accueil du VNC Viewer .....	49
Figure 48 : la fenêtre d'accueil du Raspberry Pi Imager .....	49
Figure 49 : menu d'options avancées.....	50
Figure 50 : liste des systèmes d'exploitation.....	51
Figure 51 : choisir la media de destination .....	51
Figure 53 : paramètre du Raspberry Pi OS .....	51
Figure 53 : confirmation de l'installation .....	52
Figure 54 : l'installation du système.....	52
Figure 55 : connexion au robot avec PuTTY .....	52
Figure 56 : ligna de commande du Raspberry Pi.....	53
Figure 57 : menu de configuration du Raspberry Pi OS.....	53
Figure 58 : le menu « Interface Options » .....	54
Figure 59 : confirmation d'activation de l'interface de camera .....	54
Figure 60 : confirmation d'activation du serveur VNC .....	54
Figure 61 : activation port série .....	55
Figure 62 : organigramme du programme de système .....	64
Figure 63 : les types des caractéristiques haar .....	66
Figure 64 : Illustration du fonctionnement d'une image intégrale.....	66
Figure 65 : fonctionnement du l'algorithme Adaboost .....	67
Figure 66 : organigramme de classificateurs en cascade .....	67
Figure 67 : Courbe de l'éclairement et variation de la résistance (LDR).....	70
Figure 68 : le fonctionnement du servomoteur par rapport au RC .....	71
Figure 69 : les deux états de base du port en H .....	73

<i>Figure 70 : fonctionnement du capteur ultrason .....</i>	73
<i>Figure 71 : UI de l'application web.....</i>	75
<i>Figure 71 : conception mécanique du robot.....</i>	76
<i>Figure 72 : étage 0 du robot.....</i>	77
<i>Figure 73 : étage 1 du robot.....</i>	77
<i>Figure 74 : installation des LED .....</i>	78
<i>Figure 75 : installation et câblage d'étage 2.....</i>	78
<i>Figure 76 : installation d'étage 3 .....</i>	79
<i>Figure 78 : câblage d'étage 3.....</i>	79
<i>Figure 79 : la réalisation finale du système.....</i>	80

# Introduction Générale

La Robotique fait partie des sciences des objets et des systèmes artificiels. Elle peut être vue comme la science de la perception et du mouvement et de leur intégration en une machine physique, mécanique et informatique.

L'une des nombreuses utilisations du robot est le robot de sécurité autonome ASR (Autonomous Security Robots), c'est un garde automatisés utilisés pour surveiller et effectuer des tâches de sécurité, combinant des technologies d'auto-conduite, la robotique et l'intelligence artificielle. Principalement, l'utilisation de ce type de robot représenté dans : la présence physique comme moyen de dissuasion, reconnaissance thermique et détection d'anomalies pour les personnes et les colis, détection d'accès non autorisé des personnes spécifiques...

Dans cette perspective, nous avons proposé dans le cadre de notre projet de fin d'études, la conception et la réalisation du robot S-Livid : robot de sécurité intelligente avec navigation autonome.

Ce présent rapport est articulé autour de quatre chapitres :

Le premier chapitre, sera consacré à la présentation de l'entreprise d'accueil, l'étude bibliographie des robots de sécurité autonome et notre cahier de charge de notre projet.

Dans le deuxième chapitre, on va donner notre solution pour ce projet, ainsi le choix des différents équipements et logicielles nécessaires pour sa réalisation.

Le chapitre trois sera dédié à la présentation des différents schémas électroniques ainsi que l'installation et la configuration de tous les systèmes, logiciel et bibliothèques afin qu'on puisse programmer le robot.

Dans le dernier chapitre, on va expliquer la conception logicielle et parler brièvement de la conception mécanique. On conclue ce chapitre par la présentation des étapes de réalisation du robot.

## Chapitre I

# *Cadre de projet*

# Chapitre 1 : cadre du projet

## I. Introduction

Dans ce chapitre on va présenter la société Astre incubateur de ce projet de fin d'études. Ensuite, on va présenter une étude bibliographique sur les robots autonome de sécurité, ce qui permet l'identification de ce qu'était déjà fait par les autres dans ce domaine, et par suite déterminer d'où on va partir pour la réalisation de ce projet, et préciser les améliorations qui pourraient être apporter. C'est ça qui va conduire à spécifier la valeur ajoutée sur laquelle on va travailler et à expliciter la problématique qui constitue l'objet de la recherche et l'étude d'une solution, dans les prochains chapitres, afin de l'exposer à la lumière par une réalisation satisfaisante. On va détailler l'objectif et le cahier de charge du projet.

## II. Présentation de l'entreprise

### 1. La fiche d'identité de la société

Cette fiche a pour but de présenter rapidement les informations principales sur la société ASTRE :

Tableau 1 : Informations principaux sur ASTRE engineering

Nom de l'entreprise	ASTRE engineering
Date de création	2003
Secteur d'activité	Pétrole et Gaz, Biologie, Pharmaceutique, Traitement des eaux, Industrie chimique...
Activité	Ingénierie de détail et de base, DCS design, Installation des systèmes de control-commandes, Étude électrique, Installation électrique et instrumentation, EPC : Engineering, Construction, Etude et installation des systèmes de sécurité, Ingénierie mécanique...
Adresse	Roue Tunis Km 3.5 ville Chahed -3031- Sfax – Tunisia
Téléphone	(+216) 74 258 133
Adresse mail	eng-nizardammak@astre-engineering.com

## 2. Organigramme

L'arborisant de la société est le suivant :

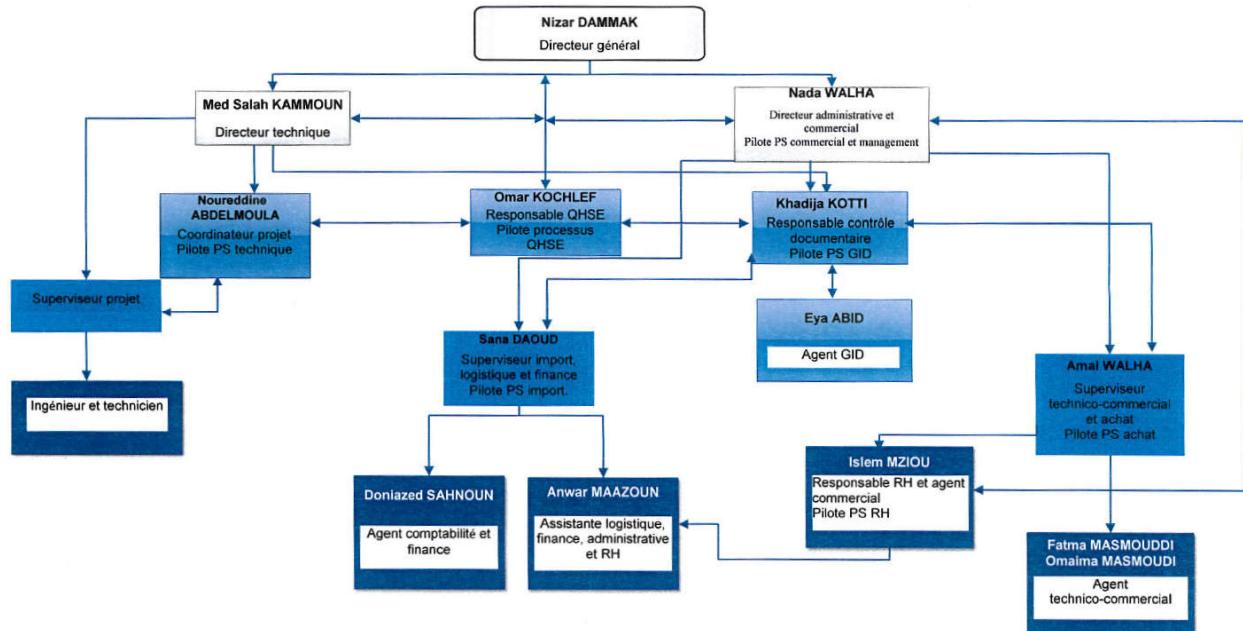


Figure 1 : organigramme de la société

## 3. Clientèles et fournisseurs de l'entreprise

### a. Clientèles

ASTRE est une société qui doit servir plusieurs clients qui sont présentés dans le tableau suivant :

Tableau 2 : services réalisés par l'entreprise

Nom du client	Service
METKA	Exploitation et maintenance des services électriques HT sur le site d'Adam
ENI Tunisia	Maintenance Instrumentation & Electrique sur tous les sites ENI.
TPS	Prestation de services d'ingénierie
VIVO-ENERGY	Conception et étude pour l'installation d'une ligne de remplissage de petits emballages de STL
STIR	Fourniture et prestation des services requis pour l'installation d'Allen Bradley matériel pour le laboratoire CFP
CERTE	Conception, fabrication et maintenance d'unité membrane de traitement d'eau
MAZARINE	Services du personnel de l'entrepreneur en gestion de projet

## b. Partenariat

Depuis sa création, ASTRE a développé un réseau de partenariat avec des fournisseurs et constructeurs internationaux afin de faciliter la réalisation de ses projets et satisfaire ses clients. Aujourd’hui, ASTRE a élaboré un partenariat avec :

Tableau 3 : partenaires de l’entreprise

Nom du partenaire	Secteur d’activité
Rockwell Automation	Fourniture et l’intégration des systèmes DCS et les équipements électriques basse et moyen tension
AI Group	Conception, fabrication et entretien installations clefs en main, véhicules, skids, matériels, émulseurs et poudres de protection contre les feux industriels.
Proscor	Fourniture des solutions techniques de haute qualité dan différent domaines.
EMERSON	Fournit à ses clients le gain du temps dans le déploiement des solutions techniques de haute qualité dans différents domaines.
ADF Services	Pour la fourniture des composants ADF
Magnetrol	Pour la fourniture des indicateurs de niveau, de température, de pression.

## III. Etude bibliographie

### 1. Définition

On appelle « robot de sécurité autonome », tous système mobile et autonome ou téléopéré permet d’assurer la sécurisation de site par la réalisation des rondes de surveillance. L’utilisation de ce type de robot de patrouille est la meilleure solution pour protéger les zones où il y a beaucoup de bâtiments, clôtures, ou d’autres obstacles qui ne permettent pas d’effectuer la vidéosurveillance à partir de positions lointaines.

### 2. Domaine ciblé

Les robots de sécurité peuvent être utilisés en mode télécommandé ou en mode autonome. Cela signifie qu'une seule personne peut surveiller plusieurs robots de sécurité et prendre le contrôle si quelque chose d'important est détecté. Ils peuvent être utilisés dans les espaces extérieurs, les grandes installations et les espaces extérieurs tels que les parkings, les lieux de divertissement, les campus d’entreprise et d’enseignement et les installations hospitalières multi-bâtiments.

Mais, malgré tout ça, l'utilisation de ces robots est limitée à car les robots de sécurité coûtent environ 8 000 \$/mois. Ces types de robots sont principalement contractés via une solution de location

mensuelle ou dans une solution « Robot as a Service – RaaS ». Bien sûr, le coût final dépend des spécifications du véhicule et des besoins de l'entreprise. [3]

### 3. Les robots de surveillance existants

On distingue principalement deux types de robot du surveillance : les robot de petite dimension designer aux maison et les entrepôts, et les robot de dimensions plus au moins important pour l'usage industrielle et les data centres. On va citer quelque exemple sur les robots du surveillance.

#### a. Le robot PGuard

Le P-Guard est un robot développé en Tunisie de sûreté / sécurité mobile, multi-services et évolutif, autonome ou téléopéré permettant de déporter rapidement et de manière sécurisée, la vue, l'ouïe et la voix.

Grâce à son système de localisation GPS à précision centimétrique, le P-Guard peut patrouiller dans un environnement dense en toute sécurité. Il possède également une présence dissuasive permanente grâce à ses 4 caméras pour une vision immersive à 360°.

Son intelligence artificielle permet au drone terrestre de reconnaître plus de 500 classes d'objets. Grâce à son Lidar 3D, le robot détecte et évite des obstacles statiques ou dynamiques et peut facilement cohabiter avec les différents flux présents sur le site.<sup>[10]</sup>



Figure 2 : la robot PGuard

#### b. Le robot Jack

TBC-France a conçu, JACK, le premier robot autonome de surveillance made in France. Un robot de sécurité équipé d'une intelligence artificielle embarquée, d'une navigation ultra-précise et d'une technologie évoluée.

Neuf mois ont été nécessaires pour développer de A à Z, le premier robot autonome qui intègre des technologies de pointe et une intelligence artificielle. Sa mission est de patrouiller et de surveiller

les grands espaces intérieurs et extérieurs difficiles à couvrir par des caméras fixes. Ses huit caméras infrarouges et sa caméra zoom panoramique PTZ (Pan-Tilt-Zoom) lui permettent de transmettre en temps réel les images de surveillance, de jour comme de nuit et de détecter les éventuelles anomalies.<sup>[11]</sup>



Figure 3 : le robot Jack

### c. Le robot Picard

Le robot Picard est un robot Canadien permet d'effectuer une vidéosurveillance à distance à l'aide d'une caméra PTZ intelligente sur de moyennes distances, soit à partir d'une position stationnaire ou en mouvement, et à l'aide de six caméras panoramiques à 360 degrés. Lors du choix d'une position favorable pour la vidéosurveillance stationnaire, la caméra PTZ est capable de détecter les personnes à une distance de jusqu'à 100 mètres dans n'importe quelle direction. Tout en se déplaçant entre les positions pour effectuer la vidéosurveillance, le système d'analyse vidéo à bord offre la possibilité de détecter les personnes à une distance de jusqu'à 20 mètres.<sup>[12]</sup>



Figure 4 : la robot Picard

#### d. Le robot E-vigilante

E-vigilante est un robot français dédié à la surveillance interne des entrepôts et des sites industriels. Ce robot réduit les coûts et les risques associés aux risques de sécurité tout en optimisant la surveillance dans les sites surveillés.

Mobile, autonome et polyvalent, e-vigilante effectue des rondes automatisées et prévient immédiatement la personne en charge de la surveillance du site lors de la détection d'un incident. Cette dernière peut alors prendre la main à distance et en temps réel sur le robot, afin d'effectuer une levée de doute grâce à la caméra, au micro et aux haut-parleurs intégrés.<sup>[13]</sup>

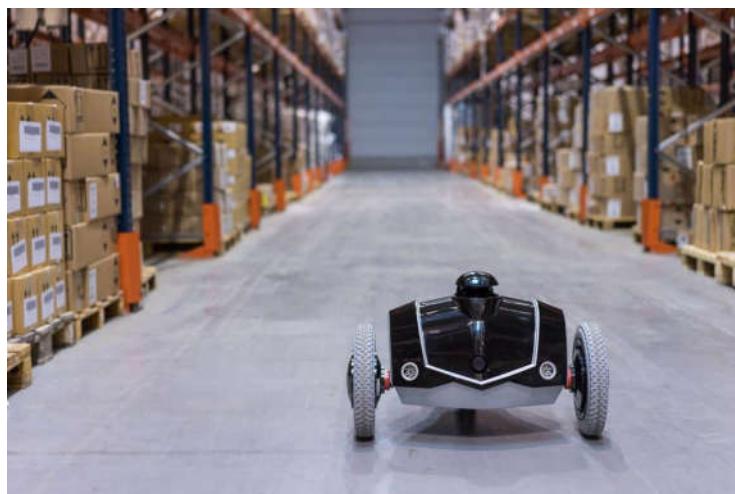


Figure 5 : le robot E-vigilante

## IV. Cahier de charge

Après avoir tous ces types de robots de surveillance, on a trouvé que le coup de ces robots est trop cher, ainsi que l'intelligence de ces robots est plus au moins limitée. Dans ce sens, on a pensé de réaliser un robot avec un apprentissage non supervisé qui peut même décider l'action ou la réaction en se basant sur la situation sans intervention humaine, et au même temps avec un prix moins cher. On a commencé la première étape dans le stage de perfectionnement, dans lequel on a l'occasion de prendre des principales connaissances sur la navigation autonome. Ce projet de fin d'étude est notre deuxième étape, et on continuera jusqu'au arrivé à notre but final.

### 1. Problématique

Pour que le robot de surveillance autonome soit efficient et remplace totalement ou quasiment les tâches réalisées par un surveillant humain, et le fait gagner d'effort, du temps et de l'argent, il doit naviguer et aux mêmes temps chercher à des étrangers dans son milieu. Tout d'abord la navigation doit être efficace, il doit naviguer en détection de tous les obstacles dans son schéma. Ensuite, on doit

avoir un système de détection des personnes efficace et rapide qui fonctionne en temps réel, c'est important d'avoir un système d'alarme pour alerter l'utilisateur dans le milieu de robot. Puis on doit avoir un système de contrôle et de visualisation du robot à l'aide d'un périphérique tel qu'un Smartphone ou un ordinateur connecté à un réseau internet. Cela permet d'observer et commander à tout moment le robot en temps réel. Enfin, on souhaite réaliser le robot et montrer notre réalisation. C'est ça l'objet de notre projet, conception et réalisation du robot de surveillance autonome. On va présenter l'étude théorique et pratique, la conception et la réalisation de notre solution dans les prochains chapitres.

## 2. Objective

Notre objectif est d'étudier, faire le concept et réaliser un système de surveillance intelligent autonome contrôlé par internet par une application web.

## 3. Travail demandé

- Présentation de notre solution
- Choix des matériels
- Câblage des composants
- Installation des logiciels adéquats
- Conception logiciel et mécanique du robot
- Réalisation du robot

## V. Conclusion

Dans ce chapitre, on a présenté en premier lieu la société ASTRE engineering. En deuxième lieu, on a consacré à la recherche et l'étude bibliographique dans laquelle on a parlé aux personnes ciblées et d'exposer les différents robots de surveillance existants en précisant leurs qualité et défaut. En dernier lieu, on a présenté la problématique dans un cahier de charge bien détaillé qu'on va essayer de résoudre aux chapitres suivants en commençant par l'étude théorique et pratique, la conception et la présentation de notre solution.

# Chapitre 2 : Etude théorique

## I. Introduction

Dans ce chapitre nous a présenté la description de la solution adoptée pour répondre au besoin expliciter dans le cahier de charge. Simultanément, on va essayer d'effectuer un choix judicieux du matériel et logiciels à utiliser pour la réalisation pratique en tenant compte de certains paramètres tels que la disponibles aux marchés, les caractéristiques techniques, et le coût, etc.

## II. Présentation de la solution

### 1. Principe du fonction du système

Notre robot de surveillance autonome est un système fonctionne selon un principe de fonctionnement efficace pour assurer le but de surveiller le milieu en évitant les obstacles. Ce système est composé de trois partie principale : la surveillance intelligente, la navigation du robot et le contrôle et supervision à distance.

#### a. La surveillance intelligente

Le système du navigation fonctionne en deux paramètres : en mode manuel ou autonome. Le mode manuel sert à contrôler la navigation par l'utilisateur à travers le système de contrôle et supervision. Contrairement, le mode automne navigue le robot sans interaction d'utilisateur, le robot détecte les obstacles et choisie le chemin de déplacement en évitant ces obstacles.

#### b. La navigation du robot

Le contrôle et supervision à distance sont faits via une interface de l'application quand l'utilisateur est connecté. Même en absence d'utilisateur, la détection et la navigation autonome continuent à fonctionner normalement.

#### c. Contrôle et supervision à distance

Le contrôle et supervision à distance sont faits via un interface de l'application quand l'utilisateur est connecté. Même en absence d'utilisateur, la ditection et la navigation autonome continuent à fonctionner normallement.

## 2. Structure du système :

On présente notre solution dans ce qui suit, comme c'est indiqué sur le schéma synoptique de la figure 6 ci-dessous. On a déterminé les éléments ou blocs nécessaires pour le bon fonctionnement du système de surveillance autonome.

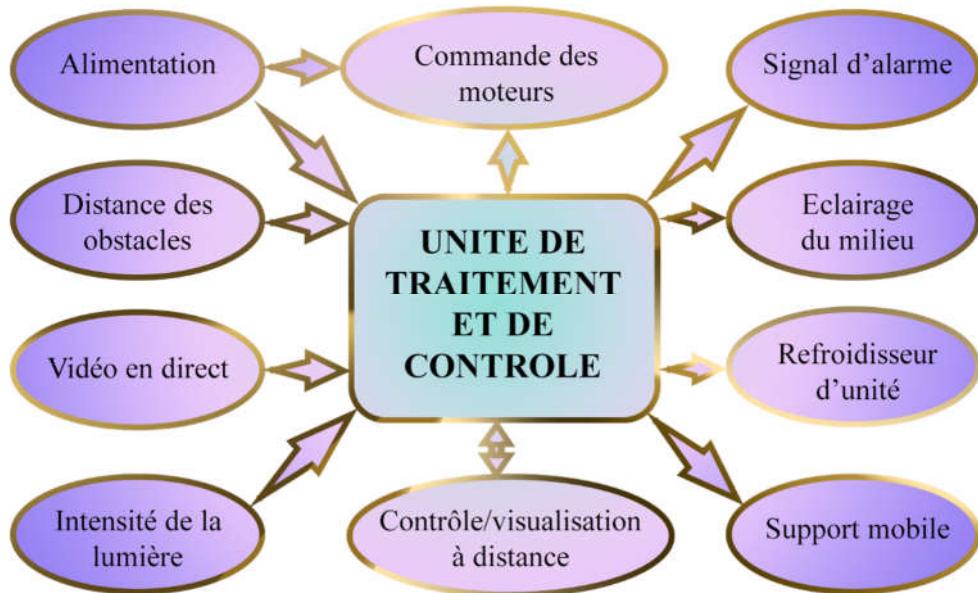


Figure 6 : Schéma synoptique des blocs du système

On distingue les blocs suivants :

- Un bloc « traitement et de contrôle » : Reçoit des informations ou/et des instructions d'autre blocs, les traite et fournit des commandes aux autres blocs. C'est lui qui gère et qui maintient le bon fonctionnement de tout le système.
- Un bloc « alimentation » : il alimente le bloc « commande des moteurs » par une tension continue de 12V, et une tension 5V continue pour alimenter les autres blocs.
- Un bloc « distance des obstacles » : c'est un capteur qui fournit une information au bloc de traitement et de contrôle représenté par la distance des objets et obstacle devant le robot.
- Un bloc « vidéo en direct » : c'est une caméra qui fournit une information au bloc de traitement et de contrôle représenté par un vidéo en temps réel.
- Un bloc « intensité de la lumière » : c'est un capteur qui fournit une information au bloc de traitement et de contrôle représenté par la valeur de l'intensité de la lumière dans le milieu du robot.
- Un bloc « contrôle/visualisation à distance » : Il constitue une interface homme-machine dont l'accès à la partie commande pourrait être à distance pour contrôler le système et le configurer

facilement à tout moment et de n'importe quel endroit du monde. Afin que l'accès à cette interface soit facile, on préfère réaliser une application web mobile à consulter via un ordinateur ou smartphone connecté au réseau Wifi ou Internet.

- Un bloc « support mobile » : c'est un support pour la caméra composé de deux servo moteurs commandé par le bloc de traitement et de contrôle.
- Un bloc « refroidisseur d'unité » : c'est utiliser pour refroidir le système de traitement et de contrôle, composée de deux ventilateurs électriques.
- Un bloc « éclairage du milieu » : il active deux leds pour éclairer la zone de visualisation de la caméra.
- Un bloc « signal d'alarme » : en cas d'une détection d'une personne, il prend une photo, l'envoyer par mail et fournit une alarme auditive avec un avertisseur sonore pour signaler l'utilisateur.
- Un bloc « commande des moteurs » : Grace à ce bloc on met les moteurs électriques des roues du robot en marche ou en arrêt. La mise en marche de ces moteurs permet la navigation du robot dans son milieu.

En cherchant un élément qui répond parfaitement au besoin on présente ci-dessous à la figure 7 le schéma synoptique élémentaire de solution du système en spécifiant un peu plus les détails de chaque bloc et les matériels à utiliser :

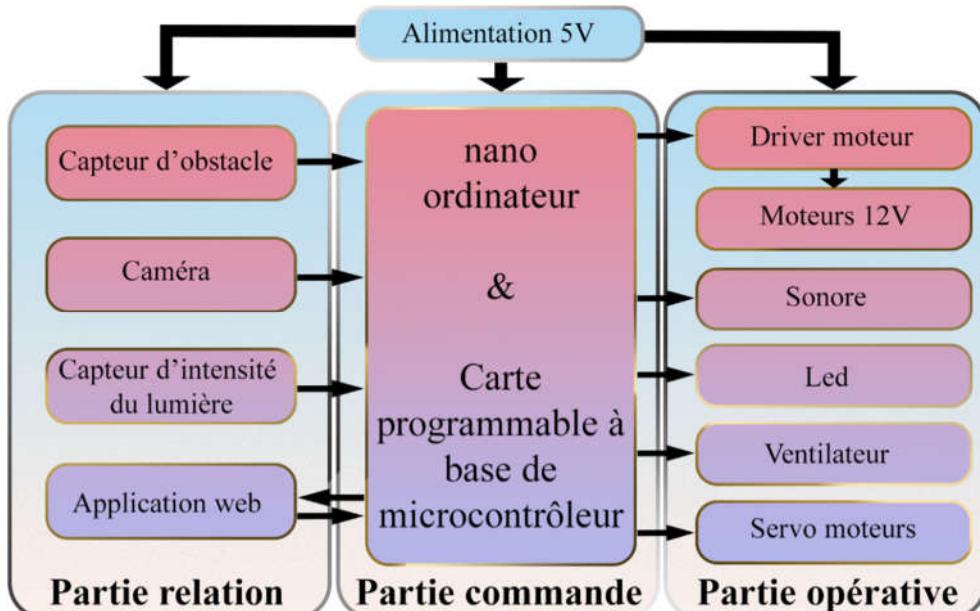


Figure 7 : schéma synoptique élémentaire du système

Ce schéma est composé de 4 parties :

- Partie relation : contient les capteurs, la caméra et l'application web.

- Partie commande : contient un nano ordinateur et un microcontrôleur pour la commande du robot.
- Partie opérative : composé de 4 moteurs et leurs driver, un sonore, deux LED, deux ventilateurs et deux servo moteur.
- Partie alimentation :
  - Une alimentation 12V pour alimenter les moteur et le driver
  - Une alimentation 5V pour les autres composants

### **III. Présentation des différents blocs et choix du matériaux adéquats**

On va maintenant présenter chaque bloc du solution d'une manière détaillée pour mieux expliquer le contenu du projet. Et on va aussi choisir le matériel adéquat pour notre solution en expliquant notre choix.

#### **1. Le bloc de traitement et de contrôle :**

Ce bloc est à base de nano ordinateur et un microcontrôleur programmable. Ainsi, dans ce qui suit, on va présenter en général ces deux composants comme des composant électronique très répondu dans le monde de l'électronique vu à l'importance et l'utilité remarquable.

En particulier, on va parler des nano ordinateurs sur lesquels sont implantés un microprocesseur et processeur graphique cabalent de traiter les images et les vidéos rapidement et un module WIFI. Pour choisir un tel nano ordinateur répond à notre cahier de charge d'avoir un vidéo en direct sur internet avec une détection précis des personnes vues par la caméra.

Ainsi, on va parler des cartes électroniques sur lesquelles ils y ont au moins un port analogique et compatible avec la communication série. Puisqu'on va choisir ensuite une telle carte pour répondre aux exigences citées précédemment tel que le capteur d'intensité du lumière et les capteurs d'obstacles.

##### **1.1. Les nano ordinateurs**

###### **a. Généralité sur les nano ordinateur**

Un nano ordinateur désigne un ordinateur dont la taille est très inférieure à celle d'un micro-ordinateur. Certains modèles peuvent ainsi tenir dans une main et ne pas dépasser la taille d'une carte de crédit, tout en offrant des caractéristiques techniques proches des micro-ordinateurs standard. Les nano ordinateurs sont basés sur des microprocesseurs. Cette architecture permet d'avoir un

microprocesseur de plus petite taille que ceux utilisés dans les ordinateurs personnels classiques, et offre également de meilleures possibilités d'intégration des composants. Avec le développement des architectures ARM, très utilisées en matière de développement web, dans les smartphones et les tablettes, on a regroupé sur une même puce le microprocesseur et d'autres composants, un processeur graphique (GPU), une RAM, une carte Ethernet et une carte réseau (Wifi, Bluetooth). Le « cerveau » d'un nano ordinateur ne dépasse donc pas les quelques centimètres carrés, le reste de la carte électronique support étant essentiellement utilisé pour placer les composants physiques, les connecteur GPIO, le connecteur de camera, le connecteur d'affichage, les connecteurs HDMI, les ports USB.

- Le système sur puce (SoC)

Le premier composant et sans doute le plus important, le système sur puce (SoC). Le terme « système sur puce » (system-on-chip en anglais) : une puce en silicium, appelée circuit intégré, qui contient l'essentiel du système de nano ordinateur. Il comprend l'unité centrale de traitement (CPU), généralement considérée comme le « cerveau » d'un ordinateur, et l'unité de traitement graphique (GPU), qui gère l'aspect visuel.

- Mémoire

Un cerveau sans mémoire ne servant pas à grand-chose, juste à côté du SoC se trouve une autre puce. Il s'agit de la mémoire vive (RAM). Lorsque on travaille avec un nano ordinateur, c'est la mémoire vive qui contient tout ce que vous faites ; ce n'est que lorsque on enregistre un travail qu'il s'inscrit sur la carte microSD. Ensemble, ces composants forment la mémoire volatile et non volatile : la RAM volatile se vide de son contenu chaque fois que le nano ordinateur s'éteint, tandis que la carte microSD non volatile sauvegarde son contenu.

- Module réseau

C'est le composant qui permet au nano ordinateur de communiquer sans fil avec d'autres appareils. En soi, ce module remplit un double rôle : celui de WiFi pour la connexion aux réseaux informatiques, et celui de Bluetooth radio pour se connecter à des périphériques tels que des souris et pour envoyer ou recevoir des données de dispositifs intelligents situés à proximité, tels que des capteurs ou des smartphones.

- Les connecteurs physiques

**GPIO** : Sur le bord de la carte, on trouve des broches métalliques, réparties. Il s'agit du connecteur GPIO (General-Purpose Input/Output), une fonctionnalité qui permettait au nano

ordinateur de communiquer avec du matériel supplémentaire, qu'il s'agisse de LED et de boutons jusqu'aux capteurs de température, aux joysticks ou aux moniteurs de fréquence cardiaque.

**USB (Universal Serial Bus)** : Ces ports nous permettent de connecter n'importe quel périphérique compatible USB au nano ordinateur, des claviers et souris aux appareils photo numériques et clés USB. D'un point de vue technique, il existe deux types de ports USB : ceux qui comportent des parties noires sont des ports USB 2.0, basés sur la deuxième version de la norme Universal Serial Bus ; ceux qui comportent des parties bleues sont des ports USB 3.0, plus rapides, basés sur la troisième version, plus récente.

**Port Ethernet** : on peut utiliser ce port pour connecter le nano ordinateur à un réseau informatique câblé à l'aide d'un câble muni d'un connecteur RJ45 à son extrémité.

**HDMI** : les ports d'interface multimédia haute définition transmettent à la fois des signaux audio et vidéo, tandis que « haute définition » est un indicateur d'excellente qualité. Ils servent à connecter au nano ordinateur à un dispositif d'affichage : un moniteur d'ordinateur, un écran TV ou un projecteur.

**Connecteur de camera** : également connu sous le nom d'Interface série de la caméra (CSI). Cela vous permet d'utiliser une Caméra Module spécialement conçu à cet effet.

**Connecteur d'affichage** : également connu sous le nom d'interface série d'affichage (DSI) conçu pour être utilisé avec un écran tactile pour le nano ordinateur.

### b. Choix du nano ordinateur

Un nano ordinateur constitue un bon choix pour l'unité de traitement et contrôle principale dans notre cas. Il peut être considéré comme le noyau de la partie commande de notre système. Notre choix sera basé sur des critères bien spécifiés, qui sont liés directement à notre application envisagée, en tenant compte évidemment du facteur coût. Voici quelques critères de base :

- Il faut dans un premier temps déterminer le nombre d'entrées/sorties à installer dans le nano ordinateur. Pratiquement, on compte : une entrée CSI pour caméra, une entré du microcontrôleur avec USB, 2 entrées digitales et 9 sorties digitales. Le tableau 4 donne une idée complète ces les entrées/sorties :

Tableau 4 : les entrées et les sorties à connecter avec le nano ordinateur

Périphérique lier au nano ordinateur	Entrée/sortie (E/S)	Nombres
Driver moteur	S	6
Avertisseur sonore	S	1
Servo moteur	S	2
Camera	E	1
Capteur à distance	E	2
Microcontrôleur	E	1

- La rapidité d'exécution est un élément important. Une vitesse d'exécution rapide du programme est exigée pour qu'il contrôle le fonctionnement du système en temps réel.
- Il faut avoir un processeur graphique pour les traitements des vidéos et des images.
- La taille de la mémoire RAM interne pour mémoriser des données est également importante pour notre application, il faut avoir au minimum 3 Go de RAM .
- La longueur du programme de l'application détermine la taille de la mémoire non volatile. Pratiquement nous avons besoin d'une grande taille pour stocker le système d'exploitation, le code, les bibliothèques et les outils de développements.

On distingue 2 types de nano ordinateur compatibles aux besoins :

Tableau 5 : comparaison entre Raspberry Pi 4 et NVIDIA Jetson

	Raspberry Pi 4	NVIDIA Jetson
<b>Photo</b>		
<b>CPU</b>	Quad-core ARM Cortex-A72 64-bit @ 1.5 Ghz	Quad-core ARM Cortex-A57 64-bit @ 1.42 Ghz
<b>GPU</b>	Broadcom VideoCore VI (32bit)	NVIDIA Maxwell w/ 128 CUDA cores @ 921 Mhz
<b>Mémoire</b>	1 Go, 2 Go, 4 Go, et 8 Go LPDDR4	2 Go ou 4 GO LPDDR4
<b>E/S</b>	<ul style="list-style-type: none"> <li>- 2 ports micro-HDMI</li> <li>- Port USB type C pour</li> </ul>	<ul style="list-style-type: none"> <li>- HDMI 2.0</li> <li>- 4 ports USB 3.0</li> </ul>

	alimentation - 2 ports USB 3.0 - 2 ports USB 2.0 - 3 mm audio-vidéo jack - Gigabit Ethernet - CSI (camera serial interface) - DSI (display serial interface) - Connecteur 40 broches - SPI, I2C, UART	- Port USB 2.0 micro B - Support MicroSD - Gigabit Ethernet - 2 MIPI CSI-2 DPHY lanes - Connecteur à 40 broches (GPIO, I2C, I2S, SPI, UART) - Connecteur à 12 broches (signaux d'alimentation et associés, UART) - Connecteur de ventilateur à 4 broches
<b>Système d'exploitation</b>	Raspberry Pi OS, Ubuntu, RISC OS, Retro Pi ...	Linux4Tegra basée sur Ubuntu 18.04
<b>Prix</b>	600 dt: 8Go RAM 350 dt: 4Go RAM 230 dt: 2Go RAM	300 dt: 2Go RAM 675 dt: 4Go RAM

En tenant compte de tous les critères mentionnés précédemment, du coût qui doit être raisonnable, et en consultant un catalogue des nano ordinateurs, on tombe sur un excellent choix qui répond parfaitement à notre besoin. C'est Raspberry Pi 4 modèle 4 Go de RAM, puisqu'il est moins cher que la carte NVIDIA Jetson qu'il est surdimensionné.

### c. Présentation du Raspberry Pi 4 (4 Go RAM)

Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM de la taille d'une carte de crédit, développé en Bretagne dans le cadre de la fondation Raspberry Pi. La première carte était développée en 2012 sous le nom « Raspberry Pi Model B », mais on va utiliser le modèle développé en 2018. En parallèle la fondation a été développé un système d'exploitation nommé Raspbian et basée sur Debian Linux (on peut utiliser autre système d'exploitation tel que Ubuntu et Retro Pi). La figure 8 ci-dessous monte une photo de Raspberry Pi 4.



Figure 8 : photo du nano ordinateur Raspberry Pi 4

- Caractéristique du Raspberry Pi 4 <sup>[annexe 1]</sup>

Le module ESP8266-NodeMCU possède les caractéristiques et périphériques suivantes :

- Quad-core CPU ARM Cortex-A72 64-bit @ 1.5 Ghz
- GPU Broadcom VideoCore VI (32bit)
- 4 Go RAM LPDDR4
- H.265 et H.264 encodeur vidéo avec 2 ports micro HDMI
- 802.11 b/g/n/ WIFI et 5.0 Bluetooth
- CSI et DSI
- 40 broches configurables
- Interface SPI, I2C
- UART sur les broches dédiées
- 4 ports USB
- Brochage du Raspberry Pi 4

Ce nano ordinateur possède 40 broches dont chacune a une ou plusieurs fonctions. Les fonctions principales sont montrées à la figure 9 ci-dessous. On trouve également des broches d'alimentations pour utiliser la tension 3.3V ou 5V afin d'alimenter des éléments externes, et des broches de la masse GND.



Figure 9 : brochage du Raspberry Pi 4

- Système d'exploitation du Raspberry Pi 4

Même qu'il y'a plusieurs choix des systèmes d'exploitation on a favorisé la version Raspberry Pi OS basé sur Debian 10 (Buster) aux autres systèmes d'exploitation comme Ubuntu et Retro Pi puisque les bibliothèques de traitement d'image et de vidéo sont plus stable. Ainsi, on va programmer avec le langage Python puisque c'est un langage complet et puissant dans spécifiquement dans notre domaine. Il est orienté objet mais n'impose pas ce type de programmation. Sa syntaxe reste très simple et le code peut être très lisible.

## 1.2. Les microcontrôleurs

### a. Généralité sur les microcontrôleurs

Un microcontrôleur est un ordinateur monté dans un circuit intégré. Les avancées technologiques en matière d'intégration, ont permis d'implanter sur une puce de silicium de quelques millimètres carrés la totalité des composants qui forment la structure de base d'un ordinateur. Comme tout ordinateur, on peut décomposer la structure interne d'un microcontrôleur en trois parties : Les mémoires, le processeur, les périphériques, et de bus de liaison. Les microcontrôleurs sont de taille tellement réduite qu'ils peuvent être sans difficulté implantés sur l'application même qu'ils sont censés piloter.

- Les mémoires

Le microcontrôleur est basé sur une architecture de processeur de type Harvard, c'est à dire qu'il y a une séparation des bus d'instructions et de données ainsi que de l'espace d'adressage. Il contient trois types de mémoire qui sont la suivante : La mémoire FLASH sur laquelle est stocké le programme à exécuter. La mémoire SRAM ou mémoire vive est la mémoire où le programme stocke les variables lorsqu'ils s'exécutent. La mémoire EEPROM sur laquelle le programmeur peut stocker des données qui ne seront pas effacées après la mise hors tension.

- Le processeur

Le processeur de microcontrôleur est une unité de traitement de l'information se compose de trois entités : L'ALU (Unité Arithmétiques et Logiques), son registre d'état (STATUS), et le registre de travail (Working Register) et les registres d'usage général (File Registers).

- Les périphériques

**Timer** : Le module Timer est un compteur généralement de 8 bits, accessible en lecture et en écriture, dont la retenue alimente un drapeau d'interruption, l'horloge de ce compteur peut être

interne ou externe par l'intermédiaire d'un pré diviseur (3 bits) Il divise la fréquence du signal qu'on lui fournit par une constante qu'il est nécessaire de programmer.

**Comparateurs analogiques** : Le module est composé d'un ou de deux comparateurs analogiques. La tension de référence peut être interne ou externe et par rapport à laquelle on compare les signaux analogiques externes.

**Convertisseur analogique numérique** : Il transforme une tension électrique en un nombre exprimée sur N bits. Ce convertisseur est composé de multiplexeur analogique 8 entrées permet de sélectionner l'entrée analogique à convertir, un échantillonneur mémorise la tension à convertir pendant la conversion et un convertisseur analogique numérique de 10 bits.

**Chien de garde** : Le chien de garde (WDT) est un périphérique de sécurité qui provoque un Reset du microcontrôleur en cas de « plantage » du programme. Les bus : Comme nous l'avions vu plus haut, les éléments d'un microcontrôleur, les mémoires et les périphériques sont reliés entre eux par des bus.

- Architecture interne

La figure 10 ci-dessous montre l'architecture interne d'un microcontrôleur en général :

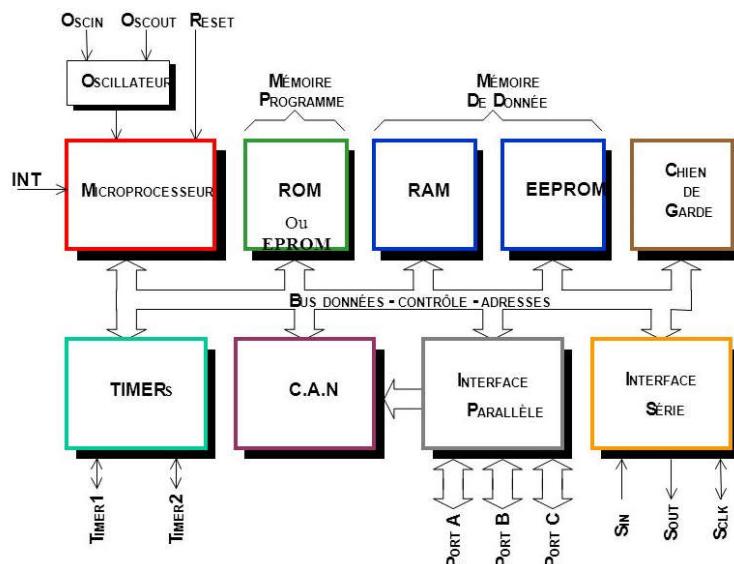


Figure 10 : architecture interne d'un microcontrôleur

### b. Choix du microcontrôleur

Un microcontrôleur est pour assister la fonction de nano ordinateur dans notre cas. Notre choix sera basé sur des critères bien spécifiés, qui sont liés directement à notre application envisagée, en tenant compte évidemment du facteur coût. Voici quelques critères de base :

- Il faut dans un premier temps déterminer le nombre d'entrées/sorties pour cette carte. Pratiquement, on compte : une entrée analogique, 4 entrées digitales, une sortie digitale et une sortie USB pour communication série. Le tableau 6 donne une idée complète sur les entrées/sorties qu'on a besoin :

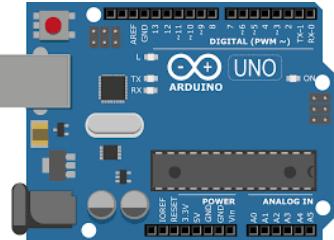
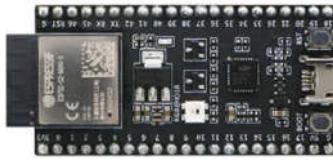
Tableau 6 : les entrées et les sorties à connecter avec le microprocesseur

Périphérique lier au microprocesseur	Entrée/sortie (E/S)	Nombre
Capteur d'intensité du lumière	E	1
Capteurs à distance	E	4
LED	S	1
Nano ordinateur	S	1

- La rapidité d'exécution est un élément important. Une vitesse d'exécution rapide du programme est exigée pour qu'il envoie les données au nano ordinateur en temps réel.

On distingue 3 types de microcontrôleur compatible avec la besoin présenté dans le tableau 7 ci-dessous :

Tableau 7 : comparaison entre différents microcontrôleurs

	Arduino UNO	ESP32-S2-DevKitM	PIC16F877
<b>Photo</b>			
<b>microcontrôleur</b>	ATmega328P	ESP32-S2FH4	PIC16F877
<b>Mémoire</b>	<ul style="list-style-type: none"> <li>- 32 Ko mémoire Flash</li> <li>- 2 Ko mémoire RAM</li> </ul>	<ul style="list-style-type: none"> <li>- 320 Ko mémoire Flash</li> <li>- 16 Ko mémoire RAM</li> </ul>	<ul style="list-style-type: none"> <li>- 14 Ko mémoire Flash</li> <li>- 368 o RAM</li> </ul>
<b>E/S</b>	<ul style="list-style-type: none"> <li>- 28 broches totales</li> <li>- 14 broches E/S numériques</li> <li>- 6 broches E/S analogique</li> <li>- 1 ports USB</li> </ul>	<ul style="list-style-type: none"> <li>- 43 broches GPIO totales</li> <li>- 4 broches SPI</li> <li>- 2 broches UART</li> <li>- 2 broches I2C</li> <li>- 1 port USB</li> </ul>	<ul style="list-style-type: none"> <li>- 40 broches totales</li> <li>- 33 broches E/S</li> <li>- SPI et I2C</li> <li>- Port série SSP</li> </ul>
<b>programmation</b>	C , C++, assembleur	LUA, C, C++	BASIC, C
<b>Prix</b>	35 dt	45 dt	35 dt

En tenant compte de tous les critères mentionnés précédemment, du coût qui doit être raisonnable, et en consultant un catalogue des microcontrôleurs et modules à base de microcontrôleurs, on déduit que la carte ESP32 est trop cher et surdimensionné, et l'utilisation du PIC16F877 n'est pas pratique puisque il nécessite une fabrication de carte électronique. Donc, on tombe sur un excellent choix qui répond parfaitement à notre besoin. C'est la carte Arduino UNO.

### c. Présentation de la carte Arduino UNO

La carte Arduino UNO est une carte à microcontrôleur. Pour pouvoir l'utiliser, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB, l'alimentation étant fournie par le port USB. La figure 11 ci-dessous montre une photo du carte Arduino UNO.



Figure 11 : photo du carte Arduino UNO

- Caractéristiques du carte Arduino UNO [annexe 2]

La carte Arduino UNO possède les caractéristiques et périphériques suivantes :

- Microcontrôleur ATmega328P 8-bit @ 16 Mhz
- 32 Ko mémoire programmable Flash
- 2 Ko mémoire RAM
- 1 Ko mémoire EEPROM
- 5 V tension de fonctionnement
- 7-12 V tension d'alimentation recommandé
- 14 broches E/S numériques
- 6 broches E/S analogiques
- Prix = 35 dt
- Brochage de l'Arduino UNO

L'Arduino UNO possède 28 broches dont chacune a une ou plusieurs fonctions. Les fonctions principales sont montrées à la figure 12 ci-dessous. On trouve également des broches d'alimentations

pour utiliser la tension 3.3V ou 5V afin d'alimenter des éléments externes et des broches de la masse GND. Et en plus de ça, un connecteur USB avec son circuit intégré de communication, à utiliser que ce soit pour le téléversement du programme ou bien pour l'alimentation avec la tension 5V via un port USB externe.

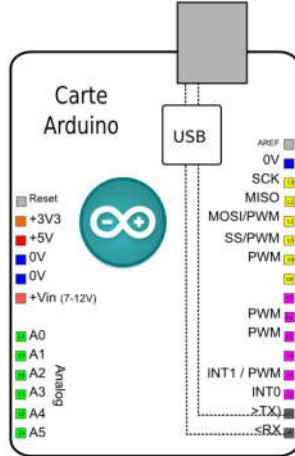


Figure 12 : brochage de l'Arduino UNO

- Programmation de l'Arduino UNO

L'Arduino peut être programmé de plusieurs façon tel que la langage C, assembleur. Mais on va utiliser la langage C++ avec l'IDE Arduino est plus pratique, c'est pour cela nous avons l'utilisée.

## 2. Le bloc « distance des obstacles »

### a. Choix du capteur de détection des obstacles

Pour que le robot évite les obstacles et au même temps décide le chemin clair il faut avoir un ou des capteurs pour mesurer la distance des obstacles possibles et l'éviter. On distingue 2 types de capteur compatible à notre besoin présenté dans le tableau 8 :

Tableau 8 : comparaison entre capteur ultrason et Lidar

	Capteur ultrason ( SRF05 )	Capteur Lidar (TF-Mini)
<b>Photo</b>		
<b>Plage de mesure</b>	2 cm – 450 cm	10 cm – 12 m
<b>Précision</b>	± 2 mm	± 5 cm de 0,1 à 6 m ± 1 % de 6 à 12 m
<b>Champs de vision</b>	15°	3,6°
<b>Prix</b>	9 dt	204 dt

On a décidé d'utiliser les capteurs ultrason comme celui montré à la figure 13 ci-dessous. Le capteur ultrason est relativement peu cher comparativement à d'autres types des capteurs. Avec sa précision, le côté d'émission du capteur est large, ce qui est avantageux pour la détection d'un objet rapproché.

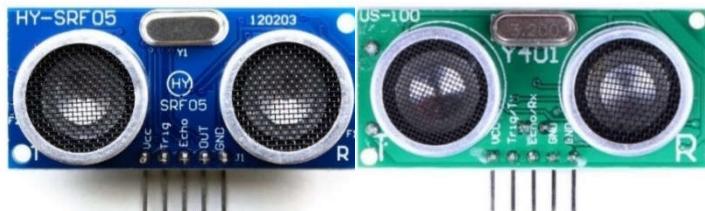


Figure 13 : photo du capteur ultrason utilisé

#### b. Présentation du capteur ultrason choisi [annexe 3] [annexe 4]

En constatant le catalogue des capteur ultrason, on a choisi le capteur HY-SRF05 pour l'installer à la carte Arduino UNO, il est caractérisé par :

- Tension de fonctionnement : 5 VDC
- Courant : < 2 mA
- Angle du capteur : < 15°
- Distance de détection : 2cm - 450 cm
- Précision : ~2 mm
- Prix : 9.2 dt

Puisque on a utilisé 2 capteurs SRF05 avec la carte Arduino, on a choisi le capteur US-100 pour bracher avec le nano ordinateur. Ce dernier est le même que SRF05 mais la tension de fonctionnement du US-100 peut être 5VDC ou 3 VDC, ce qui est compatible avec les sorties GPIO du Raspberry Pi 4.

### 3. Le bloc « vidéo en direct »

C'est la diffusion en temps réel du vidéo capté par la caméra du robot, la caméra doit avoir une bonne résolution et un taux élevé d'image par seconde (frame per second or FPS) pour que le vidéo soit clair et que le système de détection fonctionne parfaitement. Afin de constater le catalogue des camera existant compatible avec la carte Raspberry, on va utiliser RPi camera V2 pour notre projet. La figure 14 ci-dessous montre la photo du camera V2.



Figure 14 : photo du RPi camera V2

La caméra V2 est un module miniature avec une définition de 8 Mégapixels et 1080p en vidéo. La caméra se branche sur le connecteur CSI existant sur la carte Raspberry Pi. Elle est caractérisée par : [annexe 5]

- Capteur d'image : Sony IMX 219 PQ CMOS
- Résolution : 8 mégapixels
- Résolution des photos : 3280 x 2464
- Taux de transfert d'image maximum : 1080p (30 FPS) et 720p (60 FPS)
- Connection avec Raspberry Pi : 15 broches, câble plat connecté au port CSI
- Les fonctions de contrôle des images : contrôle d'exposition automatique, filtre de bande automatique, étalonnage automatique du niveau de noir, balance des blancs automatique, détection automatique de la luminance 50/60 Hz

#### 4. Le bloc « intensité du lumière »

L'importance de ce bloc est de calculer l'intensité du lumière, c'est pour activer le bloc « éclairage du milieu » si le milieu du robot devient sombre. Pour cela on va utiliser un capteur photorésistance LDR (Light Dependent Resistor) comme indique la figure 15 ci-dessous.

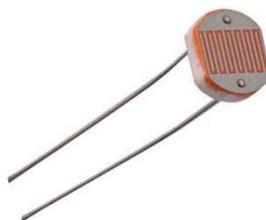


Figure 15 : photo du capteur LDR

Une photorésistance est un dispositif semi-conducteur qui diminue sa résistance électrique lorsqu'il est exposé à la lumière. La partie principale d'une photorésistance est un élément semi-conducteur (par exemple, du sulfure de plomb ou du sulfure de cadmium) positionné de telle sorte que la lumière le frappe.

## 5. Le bloc « support mobile »

### a. Généralité sur les servomoteurs :

Ce bloc est composé de deux servomoteurs pour la rotation horizontale et verticale du camera. Le servomoteur est un système relativement complexe qui intègre dans un boîtier de petite taille des constituants électroniques et électromécaniques. Sa fonctionnalité est de tenir une position angulaire donnée fournie au système en entrée sous forme d'un signal électrique.

Commençons par présenter les constituants élémentaires. La figure 16 ci-dessous montre comment ces constituants sont placés dans le boîtier.

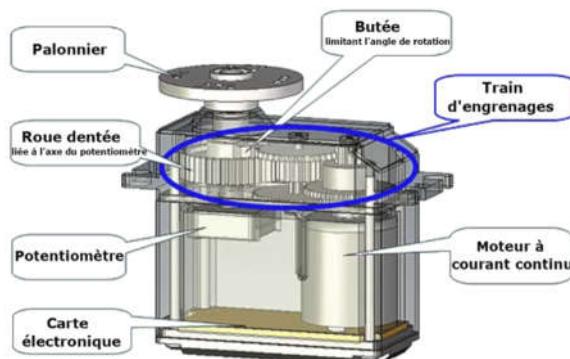


Figure 16 : les constituants élémentaires d'un servo moteur

Les principaux constituants sont :

- Un moteur électrique à courant continu (MCC)
- Un train d'engrenages
- Un potentiomètre relié au train d'engrenages par l'intermédiaire d'une roue dentée, pour fournir la mesure de l'angle de déviation de l'axe
- Une carte électronique pour gérer l'asservissement
- Dans le servomoteur standard, il y a une butée mécanique qui limite l'excursion en angle de -90° à +90° par rapport à la position dite neutre, soit une excursion de 180°
- Enfin, le palonnier permet la liaison avec la partie mobile, les roues du robot pour les deux servomoteurs à rotation continue et le capteur de distance, pour le servomoteur standard qui entraîne la tourelle du capteur de distance

### b. Le servomoteur SG90

On a choisi ce servomoteur puisqu'il est léger et moins cher que les autres servos, la figure 17 ci-dessous montre la photo du servomoteur SG90.



Figure 17 : photo du servomoteur SG 90

Le SG90 est caractérisé par : [annexe 6]

- Tension de fonctionnement : 4.8 VDC
- Vitesse de fonctionnement : 0.1s/ 60°
- Couple de décrochage : 1.8 kgf.cm
- Prix : 9 dt

## 6. Le bloc « refroidisseur d'unité »

Ce bloc indispensable pour découvrir les hautes performances des modèles Raspberry Pi4. On a utilisé un boîtier en alliage d'aluminium avec boîtier en métal pour dissipateur de chaleur à double ventilateur de refroidissement est spécialement conçu pour Raspberry Pi4.

Ce dissipateur thermique permet au Raspberry Pi de fonctionner avec une très grande fiabilité et avec une efficacité supérieure pour les applications qui nécessitent la puissance de calcul maximale du Pi4 qu'on va l'utiliser. La figure 18 ci-dessous montre la photo du radiateur avec ces 2 ventilateurs.



Figure 18 : radiateur du Raspberry Pi avec les ventilateurs

Le radiateur est caractérisé par les caractéristiques suivantes :

- Plage de la tension : 5 VDC
- Courant nominale : 0.12 A
- Vitesse : 4200 tr/min
- Niveau du bruit : 23 dBA
- Prix : 45 dt

## 7. Le bloc « éclairage du milieu »

Contrôler par la carte Arduino et la capteur LDR, lorsque l'intensité de la lumière diminue dans le milieu ce bloc sert à alimenter 2 LED en tête du robot pour éclairer le milieu. La figure 19 ci-dessus présente la photo du LED utilisé.



Figure 19 : photo du LED utilisé

## 8. Le bloc « signal d'alarme »

C'est nécessaire pour avertir l'opérateur afin que le robot détecte une personne. Ce bloc prend une photo du personne détecté et l'envoyer par e-mail. Au même, il déclenche un avertisseur sonore « Buzzer », comme celui montré à la figure 20 ci-dessous, qui génère un son de même fréquence que celle de la tension appliquée entre ses bornes.



Figure 20 : photo de l'avertisseur sonore à utiliser

Le buzzer utiliser possède les caractéristiques suivantes : [annexe 7]

- Nom du produit : LPB1475B
- Tension nominale : 12 VDC
- Tension de fonctionnement : 3~16 VDC
- Consommation du courant :  $\leq 12 \text{ mA}$
- Sortie sonore à 10 cm :  $\geq 85 \text{ dB}$
- Prix : 1.5 dt

## 9. Le bloc « commande des moteurs »

Pour achever le mouvement de robot on doit avoir des moteurs est un driver moteur pour commander et alimenter les moteurs.

## 9.1. Les driver moteur

### a. Généralité sur les drivers moteur

Une commande de moteur ou un driver de moteur : C'est un ensemble qui entre le moteur et l'utilisateur qui permet de convertir des signaux électriques pour que le moteur puisse les interpréter. Un Driver moteur peut se diviser en 2 fonctions :

- L'alimentation du moteur avec ses contraintes de tensions, courants et puissances à dissiper ainsi que la protection contre les surtensions et les surintensités. C'est l'électronique de puissance.
- Le séquenceur qui gère la chronologie des impulsions pour contrôler le sens de rotation désiré et la vitesse du moteur

Le driver moteur hacheur est structuré en port H, il servant à contrôler la polarité aux bornes d'un dipôle. Il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom comme indique la figure 21. Les commutateurs peuvent être des relais, des transistors, ou autres éléments de commutation en fonction de l'application visée.

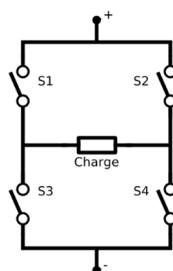


Figure 21 : circuit d'un port en H

En tenant compte de notre utilisation et du coût qui doit être raisonnable, et en consultant un catalogue moteur driver, on a choisi le driver moteur L298N.

### b. Le driver moteur L298N

Le L298N est un circuit très populaire et bon marché, il peut délivrer jusqu'à 2A en pointe et 20W en continu et possède son propre circuit d'alimentation logique. Ce module intègre une entrée de tension. La plage de tensions bascule entre les 3 V et les 35 V, selon la configuration du cavalier régulateur de tension. Avec le cavalier régulateur on peut contrôler des moteurs de 5 à 12 V et on pourra utiliser la broche de 5 V du module pour ajouter d'autres Shields. Sans ce cavalier régulateur, on peut contrôler des moteurs de 12 à 35 V, et dans ce cas la troisième broche agira comme entrée de 5 V pour alimenter la partie logique du driver.

Les sorties des moteurs 1 et 2 nous fournirons l'énergie nécessaire pour démarrer les moteurs. Assurez-vous que la polarité des moteurs est la même sur les deux entrées. Dans le cas contraire, vous devrez peut-être les échanger lorsque vous réglerez les deux moteurs vers l'avant et vers l'arrière.

Les pins IN 1 e IN 2 nous servent à contrôler le sens de rotation du moteur 1, et les pins IN3 et IN4, celui du moteur 2. Ces pins fonctionnent :

- IN 1 -> HIGH; IN 2 -> LOW: le moteur tourne dans un sens.
- IN 1 -> LOW; IN 2 -> HIGH: le moteur tourne dans le sens inverse.

Et c'est pareil pour les pins IN 3 et IN 4 du moteur 2.

Pour contrôler la vitesse de rotation des moteurs on doit retirer les cavaliers des pins ENA et ENB. On les connecte à deux sorties PWM de la plaque Arduino de façon qu'on puisse envoyer une valeur entre 0 et 255 pour gérer cette vitesse des moteurs. Avec les cavaliers installés, les moteurs tourneront toujours à la même vitesse. la figure 22 présente le driver L298N et ses composants.

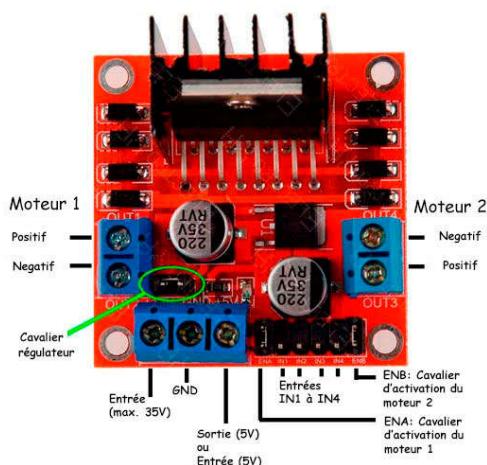


Figure 22 : L298N et ses composants

Les caractéristiques du L298N sont : [annexe 8]

- Tension logique : 5V
- Tension d'entrainement : 5V-35V
- Courant logique : 0mA – 36mA
- Courant d'entrainement : 2A MAX
- Puissance maximale : 25 W
- Prix : 15 dt

## 9.2. Les moteurs à courant continu

### a. Généralité sur le moteur à courant continu

Le moteur à courant continu, appelé aussi moteur DC, fait partie de la classe des moteurs électriques et sert essentiellement à transformer de l'énergie électrique en énergie mécanique. La plupart des formes de moteurs DC reposent, dans ce contexte, sur les forces magnétiques et disposent de mécanismes internes de types électronique ou électromécanique. Les moteurs à courant continu conventionnels se caractérisent d'abord par le collecteur qui change périodiquement le sens du flux électrique à l'intérieur du moteur. Une des plus puissantes versions du moteur DC classique que l'on retrouve dans de nombreuses applications est le moteur DC sans balais ; celui-ci fonctionne sans frottement et offre donc notamment une durée de vie plus longue.

### b. Choix des moteurs

Pour calculer le couple, la puissance et le courant requis par un robot mobile à roues, plusieurs principes doivent être utilisés : notion de vecteurs ; Équilibre des forces 2D ; Du pouvoir ; Courant et tension. Pour rouler sur une surface horizontale, les moteurs d'un robot à roues doivent produire suffisamment de couple pour surmonter les imperfections de la surface ou des roues, ainsi que les frottements dans le moteur lui-même. Donc théoriquement, le robot ne nécessite pas beaucoup de couple pour se déplacer purement horizontalement. Pour qu'un robot puisse gravir une pente à vitesse constante (sans accélération ni décélération), il doit produire suffisamment de couple pour "contrecarrer" l'effet de la gravité, qui autrement le ferait rouler sur la pente. Sur une surface inclinée (à un angle  $\theta$ ) cependant, une seule composante de son poids ( $mg_x$  parallèle à la surface) fait descendre le robot. L'autre composante,  $mg_y$ , est équilibrée par la force normale que la surface exerce sur les roues.

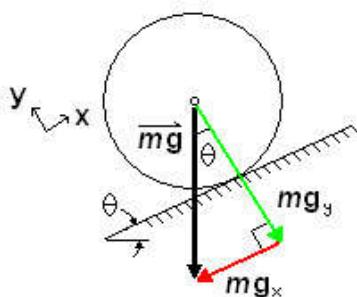


Figure 23 : force du poids sur un plan incliné

$$mg_x = m \cdot g \cdot \sin(\theta)$$

$$mg_y = m \cdot g \cdot \cos(\theta)$$

Pour que le robot ne glisse pas sur la pente, il doit y avoir un frottement entre la roue et la surface. C'est le frottement ( $f$ ) qui « produit » le couple.

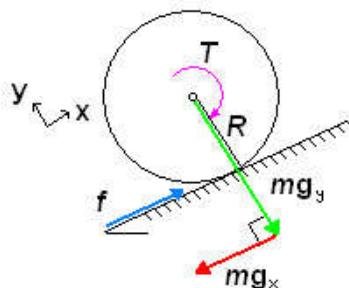


Figure 24 : force de mouvement et couple

Le couple ( $T$ ) nécessaire est de :  $T = f \cdot R$

Pour sélectionner le bon moteur, nous devons considérer le "pire scénario", où le robot n'est pas seulement sur une pente, mais accélère.

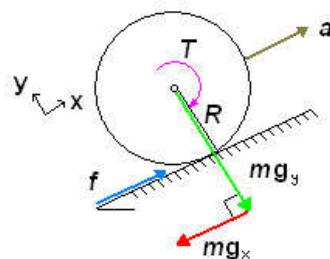


Figure 25 : force d'accélération

On note maintenant que toutes les forces ( $F$ ) sont le long des axes  $x$  et  $y$ . On équilibre les forces dans la direction  $x$  :  $\sum F_x = M \cdot a = f - m \cdot g_x$

En insérant l'équation pour le couple et l'équation pour  $mg_x$ , nous obtenons :

$$m \cdot a = \frac{T}{R} - m \cdot g \cdot \sin(\theta)$$

$$T = R \cdot m \cdot (a + g \cdot \sin(\theta))$$

Cette valeur de couple représente le couple total nécessaire pour accélérer le robot sur une pente. Cependant, cette valeur doit être divisée par le nombre total ( $N$ ) de roues motrices pour obtenir le couple nécessaire à chaque moteur d'entraînement. On note :

- Masse du robot :  $m = 3 \text{ kg}$
- Accélération minimale :  $a = 0.05 \text{ m.s}^{-2}$
- Angle maximale de décallage :  $\theta = 10^\circ$
- Rayon du roue :  $R = 0.065 \text{ m}$
- Nombre de moteur : 4

$$\begin{aligned}
 T &= \frac{(a + g \cdot \sin(\theta)) \cdot m \cdot R}{N} \\
 &= \frac{0.05 + 9.8 \cdot \sin(10) \cdot 3 \cdot 0.065}{4} \\
 T &= 0.085 \text{ Nm}
 \end{aligned}$$

En tenant compte de tous les critères mentionnés précédemment, du coût qui doit être raisonnable, et en consultant un catalogue des moteur DC, on tombe sur un excellent choix qui répond parfaitement à notre besoin. C'est le moteur JGA25-370.

### c. Le moteur jga25-370

Le moteur jga25-370 est un moteur motoréducteur, il converti la vitesse élevée et le faible couple d'origine du moteur en un état de faible vitesse et de couple élevé. L'arbre de transmission est doté d'un moteur puissant, ce qui vous permet de brûler un faible bruit et une plus longue durée de vie. Puissance élevée, faible résistance à haute température, forte capacité de pénétration.

Ce moteur de fil est fait d'un matériau abs de haute qualité, qui peut assurer une résistance et une durabilité élevées. Il est aussi économique et efficace, avec une faible consommation d'énergie pour obtenir un excellent moteur électrique. La figure 26 montre le moteur jga25-370.



Figure 26 : photo du moteur jga25-370

Ce moteur est caractérisé par : [annexe 9]

- Tension nominale : 12 VDC
- Rapport du réduction : 9 :6
- Couple en charge : 1.2 Kg.cm = 0.11 N.m
- Vitesse en charge : 500 tr/min
- Courant en charge : 0.45 A
- Couple de décrochage : 0.17 N.m
- Courant de décrochage : 1.3 A

## 10. Le bloc « alimentation »

### a. Généralité sur les batteries

Une batterie d'accumulateurs, ou plus communément une batterie, est un ensemble d'accumulateurs électriques reliés entre eux de façon à créer un générateur électrique de tension et de capacité désirée. Ces accumulateurs sont parfois appelés éléments de la batterie ou cellules.

On appelle aussi batterie les accumulateurs rechargeables destinés aux appareils électriques. La batterie d'accumulateurs permet de stocker l'énergie électrique sous forme chimique et de la restituer sous forme de courant continu, de manière contrôlée.

### b. Choix du batterie 12 V

On a câblé les 4 moteur du robot avec cette batterie. En cours de la navigation du robot, les moteurs fonctionnent avec 70% de vitesse maximale, et au même temps consomme 70% du courant

$$I_M = \frac{I \times 70}{100} = 1.4 A$$

La consommation maximale des moteurs électrique est 1.4 A pour chaque heure.

En tenant compte de tous les critères mentionnés précédemment, du coût qui doit être raisonnable, et en consultant un catalogue des batteries, on tombe sur un excellent choix qui répond parfaitement à notre besoin. C'est la batterie Li-po 3s 3000 mAh.

### c. Batterie Li-po 3S 3000 mAh

Les batteries au lithium polymère sont utilisées dans de nombreuses industries de loisirs RC. Au cours des dernières années, les batteries Li-po sont devenues le choix de batterie le plus populaire pour tous ceux qui recherchent une durée de fonctionnement plus longue et une puissance plus élevée. Les batteries Li-po pèsent moins et peuvent être fabriquées dans presque n'importe quelle taille ou forme. Les batteries Li-po ont des capacités plus élevées, contiennent plus de puissance et ont un taux de décharge plus élevé, ce qui signifie qu'elles ont plus de punch. Pour une utilisation et un stockage sûr des batteries Li-po, vous devez suivre les règles et traiter les batteries avec le respect qu'elles méritent. La figure 27 présente une photo du batterie Li-po.



Figure 27 : photo du batterie Li-po

Cette batterie est caractérisée par :

- Capacité minimale : 3000 mAh
- Tension maximale : 11.1 V
- Nombre des cellules : 3
- Décharge constante : 25c
- Prix : 135 dt

#### d. Choix du batterie 5 V

Exceptant les moteurs électriques tous les autre composant du robot fonction à la tension de 5 V, la consommation du courant de tous les composant est recapsuler dans le tableau 9 ci-dessous.

Tableau 9 : consommation du courant

Composants	Nombre du pièces	Courant maximale(I)
Raspberry Pi 4	1	1.45
Arduino Uno	1	0.05
Capteur ultrason	3	0.15
Caméra V2	1	0.28
Capteur LDR	1	0.002
Servomoteur	2	0.2
Ventilateur	2	0.24
LED	2	0.12
Buzzer	2	0.006
<b>Totale</b>	15	2.5

La consommation maximale de ces composants est 2.5 A pour chaque heure.

En tenant compte de tous les critères mentionnés précédemment, du coût qui doit être raisonnable, et en consultant un catalogue des batteries, on tombe sur un excellent choix qui répond parfaitement à notre besoin. C'est la batterie Li-ion 5V 10000 mAh.

#### e. Batterie Li-ion 10000 mAh

Une batterie lithium-ion, ou accumulateur lithium-ion, est un type d'accumulateur lithium. Ses principaux avantages sont une énergie massique élevée ainsi que l'absence d'effet mémoire. Enfin, l'auto-décharge est relativement faible par rapport à d'autres accumulateurs. Cependant, le coût reste

important et a longtemps cantonné le lithium aux systèmes de petite taille. la figure 28 présente une photo du batterie Li-ion utilisé.



Figure 28 : photo du batterie Li\_ion

## 11. Le bloc « contrôle/ visualisation à distance »

Ce bloc est intéressant car on peut contrôler le fonctionnement du système à distance, que ce soit dans l'endroit où est installé avec une connexion directe à l'unité de commande via une connexion Wifi ou bien de n'importe quel endroit du monde à l'aide d'une connexion Internet. L'accès à ce bloc pise fait de n'importe quel périphérique, tels qu'un ordinateur ou Smartphone, etc., connecté au même réseau Wifi que lui à l'aide d'une adresse IPv4 générée automatiquement par le réseau Wifi ou bien imposée à la configuration. La communication suit le protocole HTTP 0.0.0.0 :5000. Pour avoir l'accès à distance via une connexion Internet, on doit le connecter sur un réseau Wifi généré par un routeur connecté à l'Internet puis faire une configuration simple de l'adresse IP et du port sur le routeur appelée routage IP. Notre besoin se résume en l'aptitude de se connecter à distance pour contrôler ou configurer la partie commande qui gère tout le système. Donc, on doit créer une application web mobile avec les langages HTML, CSS et JavaScript. Ainsi, quand un utilisateur saisi l'adresse IP de la partie commande dans un navigateur web sur son périphérique. Comme ça il peut voir contrôler et visualiser le système en temps réel.

## IV. Outil de développement

### 1. Fritzing

Fritzing est un logiciel open source qui rend l'électronique accessible à tous en tant que matériau créatif. Ce logiciel a un outil logiciel, un site Web communautaire et des services dans l'esprit de Processing et Arduino, favorisant un écosystème créatif qui permet aux utilisateurs de documenter leurs prototypes, de les partager avec d'autres, d'enseigner l'électronique dans une salle de classe, et de concevoir et fabriquer des PCB professionnels.



Figure 29 : interface fritzing

## 2. Visual Studio

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

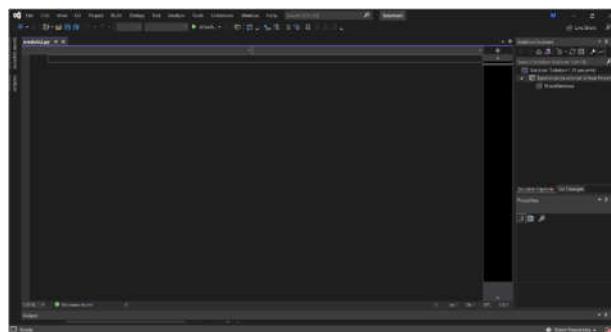


Figure 30 : interface Microsoft Visual Studio

## 3. Arduino IDE

Le logiciel open source Arduino (IDE) facilite l'écriture de code et son téléchargement sur la carte. Ce logiciel peut être utilisé avec n'importe quelle carte Arduino.

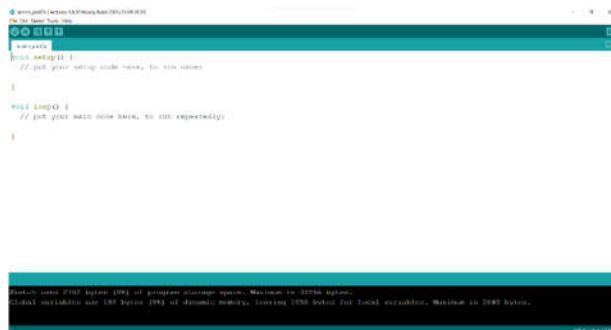


Figure 31 : interface Arduino IDE

#### 4. Solidworks

Solidworks est un logiciel de conception assistée par ordinateur, appartenant à la société Dassault Systèmes. Il utilise le principe de conception paramétrique et génère trois types de fichiers qui sont liés : la pièce, l'assemblage, et la mise en plan. Ainsi toute modification sur un de ces trois fichiers sera répercutée sur les deux autres.

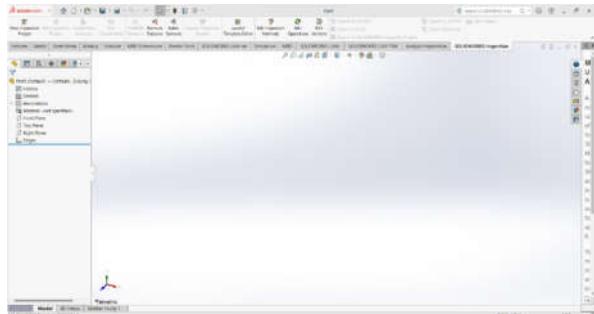


Figure 32 : interface solidworks

#### V. Conclusion

Au cours de ce chapitre, on a présenté la solution pour le système du surveillance autonome, sa structure et son principe de fonctionnement. Puis on a fait l'étude de façon détaillée en choisissant le matériel « hardware » adéquat et les outils de programmation « Software » en expliquant notre choix. L'étude pratique de cette solution sera l'objectif du chapitre suivant.

Chapitre III

*Etude pratique*

# Chapitre 3 : Etude Pratique

## I. Introduction

Dans ce chapitre, on va présenter les schémas électriques associés à notre solution avec ses différents composants. Ensuite, on va décrire l'installation et configuration du système et l'environnement de travail.

## II. Câblage du schéma électrique

Dans cette partie, on va réaliser les schémas électriques du système, pour cela on a besoin d'un logiciel de Conception Assistée par ordinateur (CAO) électrique. Ainsi on va choisir d'utiliser le logiciel Fritzing pour créer les schémas. Puis on a effectué un test réel pour assurer que la solution fonctionne correctement.

### 1. Présentation du schéma complet

La figure 33 ci-dessous montre le montage complet du système. Le schéma n'est pas très clair car il est de grande taille, mais c'est juste utile ici pour avoir une idée en grosso modo sur les différents blocs sur lesquels on va parler en détails dans ce qui suit.

On voit que notre schéma électrique est bien organisé grâce à une schématisation modulaire en se basant au schéma synoptique élémentaire du système. Dans ce qui suit, on va présenter en détails les schémas de câblage des différents composants en ajoutant des explications relatives aux caractéristiques de chacun d'eux.

Tableau 10 : légende du montage électrique du système

Numéro	Nom	Numéro	Nom
1	Batterie 5V 10000 mAh	8	Arduino UNO
2	Batterie 12V 3000mAh	9	Driver moteur L289N
3	Capteur US-100	10	Moteur électrique
4	Camera V2	11	Buzzer
5	Capteur LDR	12	Servomoteurs SG90
6	Capteurs SRF05	13	LED
7	Raspberry Pi 4		

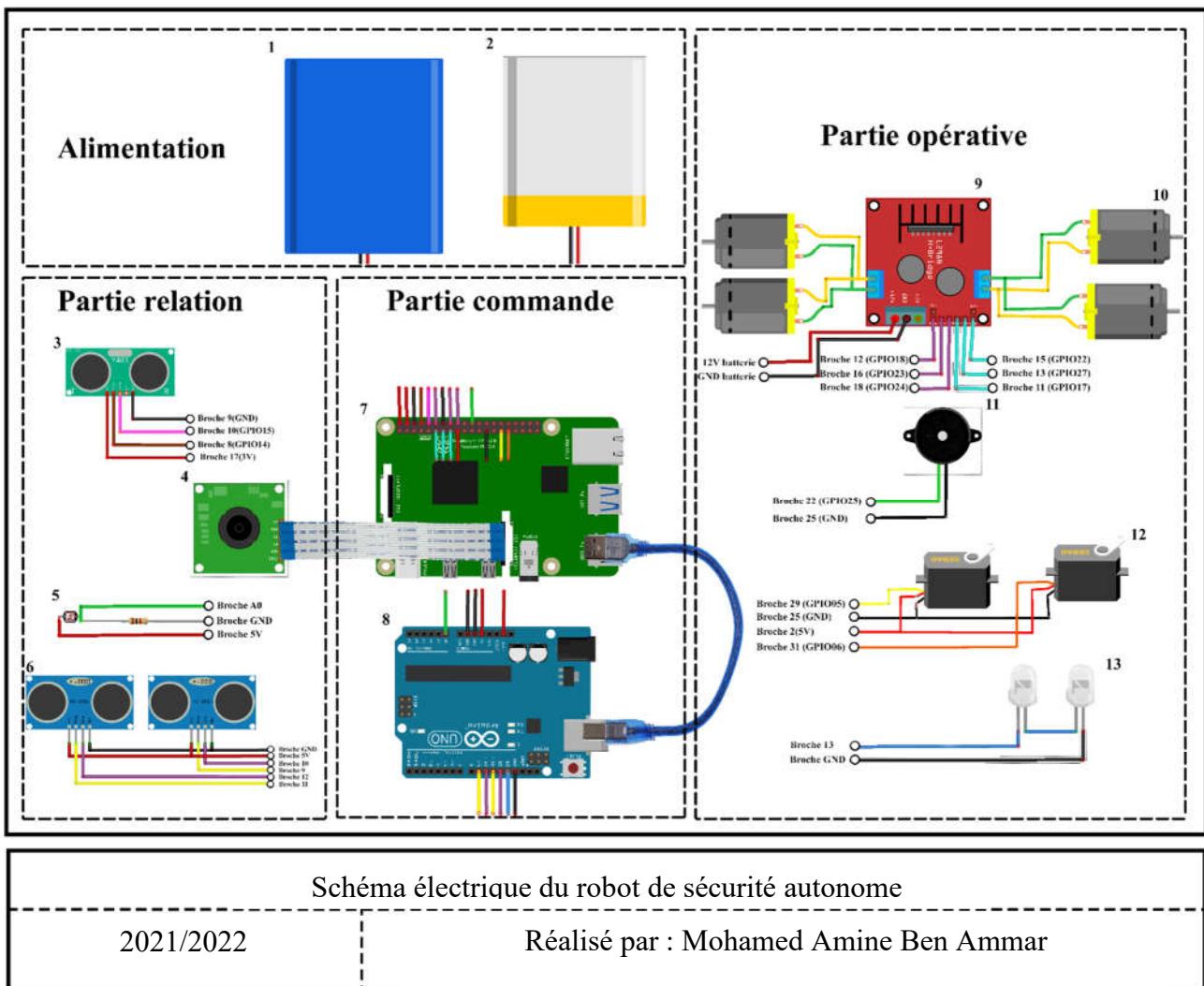


Figure 33 : montage électrique complet du système

## 2. Raspberry Pi 4

Raspberry Pi 4 est la partie principale de la partie commande. Son alimentation électrique est assurée avec une tension continue de valeur 5V et un courant de 2A ou 3A par un câble USB type C connecté au port nommé « power » montré dans la figure 34. Le nano ordinateur possède un régulateur de tension qui stabilise la tension d'alimentation 5V à 3V. Le capteur de détection US-100 doit être alimenté à une tension 3V. Ainsi, les deux ventilateurs et les deux servomoteurs doivent être alimentés par une tension de 5V, pour cela on va utiliser la broche 17(3V) et la broche 9(GND) pour l'alimentation le capteur US-100, les broches 4(5V) et 6(GND) à connecté avec les ventilateurs et les broches 4(5V) et 25(GND) pour les servomoteurs.



Figure 34 : alimentation du Raspberry Pi 4

On a 4 entrées : les broches 8(GPIO 14) et 10 (GPIO 15) pour le capteur ultrason, le connecteur caméra pour la caméra indiquée par la figure 35, et le port USB 3.0 pour connecter la carte Arduino UNO. Toutes les entrées sont rappelées au +Vcc par des résistances internes programmables (Internal Pull-up).



Figure 35 : connexion du camera avec la carte

On a aussi 9 sorties : les broches 11(GPIO 17), 12(GPIO 18), 13(GPIO 27), 15(GPIO 22), 16(GPIO 23) et 18(GPIO 24) sont connecté au driver moteur, la broche 22(GPIO 25) pour l'avertisseur sonore, et les broches 29(GPIO05) et 31(GPIO06) pour contrôler les servomoteurs. Les connections entrées/sorties avec les broches du Raspberry Pi 4 sont montré dans la figure 36.

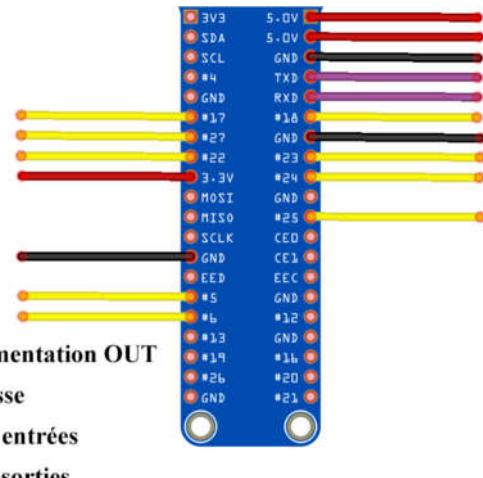


Figure 36 : schéma des E /S avec Raspberry Pi

### 3. Arduino UNO

La carte Arduino UNO est la partie secondaire de la partie commande. Son alimentation électrique est assurée avec une tension continue de valeur 5V avec un câble montré dans la figure 37. Les capteurs de détection SRF05 et le capteur LDR doivent être alimenté à une tension 5V.

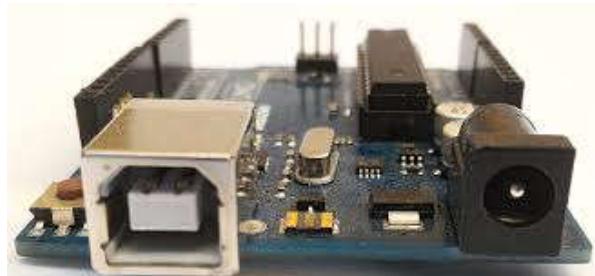


Figure 37 : port USB du carte Arduino UNO

On a 4 entrées : les broches A0 pour le capteur LDR, les broches 9, 10, 11 et 12 pour les capteurs ultrason SRF05. Toutes les entrées sont rappelées au +Vcc par des résistances internes programmables (Internal Pull-up).

On a aussi 9 sorties : le broches 13 pour le fonctionner le LED, le port USB pour connecter la carte Arduino UNO. Les connections entrées/sorties avec les broches d'Arduino UNO sont montré dans la figure 38.

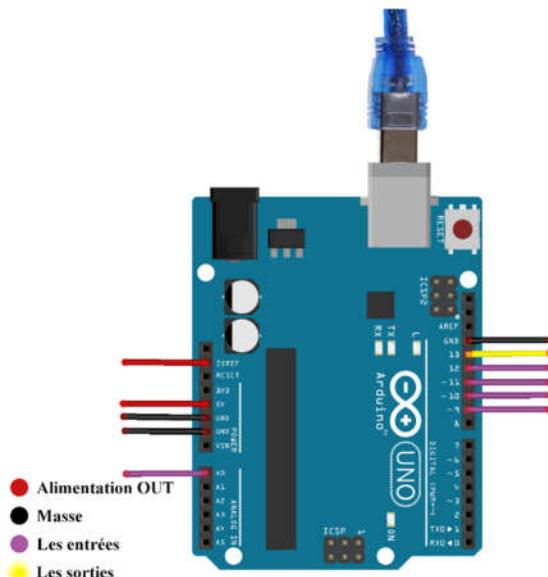


Figure 38 : schéma de câblage du carte Arduino UNO

### 4. Système d'alimentation

Pour l'alimentation on a utilisé 2 batteries : une de 5V connecté avec la carte Raspberry avec un câble USB type C-type, l'autre batterie de 12 V est connectée avec le driver moteur pour alimenter

les 4 moteurs comme indique le schéma du figure 39 (on a connecté le GND de la batterie avec une broche GND de ma carte Raspberry Pi pour avoir un GND commun).

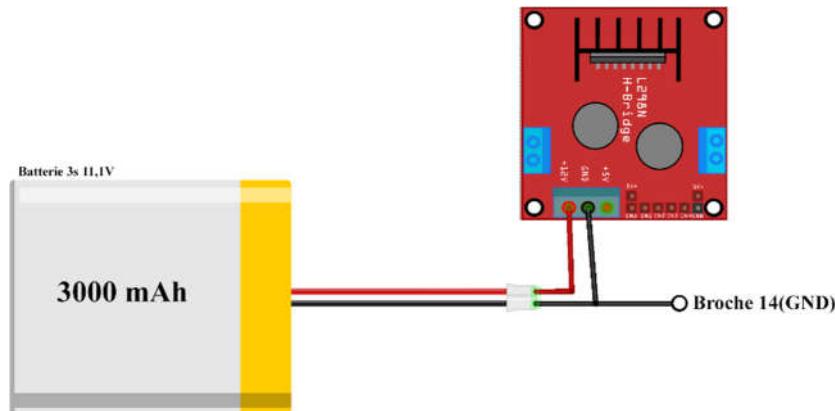


Figure 39 : schéma du câblage de la batterie 12V

## 5. Système de détection d'obstacle

Les capteur ultrason sont pour détecter les obstacles devant le robot. Le capteur US-100 est le capteur principal connecté avec la carte Raspberry Pi avec les broches 8 et 10 nommé respectivement TXD0 et RXD0 (serial port) comme indique le schéma dans la figure 40.

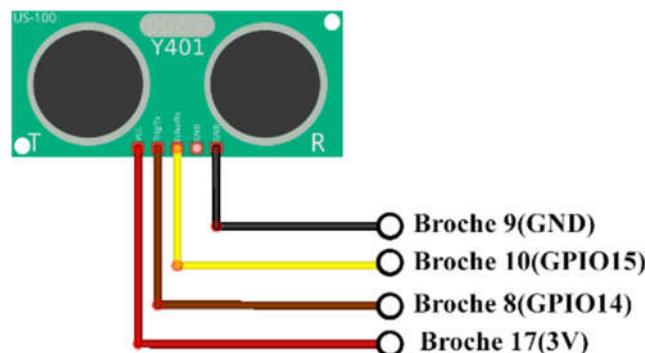


Figure 40 : schéma de câblage du US-100

Les 2 autres capteurs SRF05 sont connecté avec la carte Arduino UNO comme c'est indiqué à la figure 41.

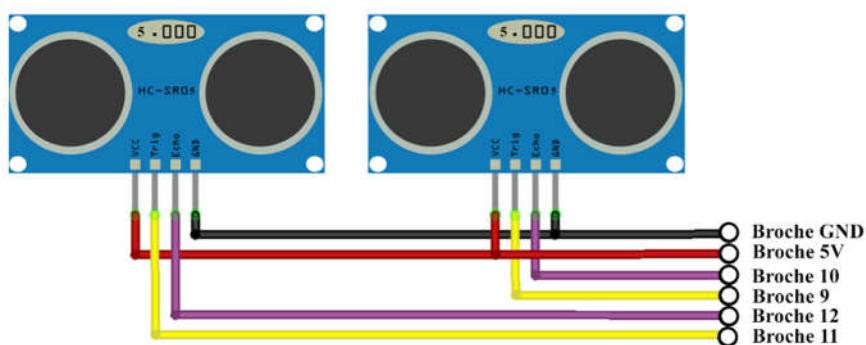


Figure 41 : schéma de câblage du SRF05

## 6. Système de mouvement

Les moteurs permettent le mouvement du robot, ils sont Contrôlés par la carte Raspberry Pi à travers le driver moteur, et alimenté par 12 V fourni la batterie Li-po. Chaque moteur est contrôlé par 3 broches du Raspberry Pi in1, in2 et EN. L'un des moteurs est connecté par les broches 12(GPIO18), 16(GPIO23) et 18(GPIO24), l'autre moteur est braché avec les broches 11(GPIO17), 13(GPIO27) et 15(GPIO22). Le schéma est présenté dans la figure 42 suivante.

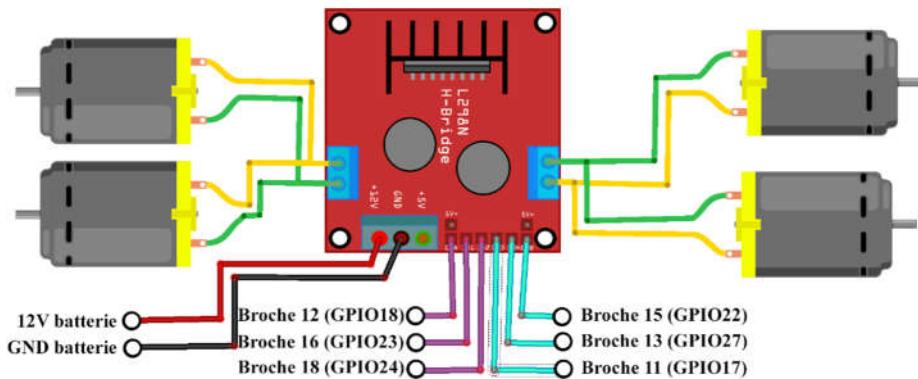


Figure 42 : schéma du système du mouvement

## 7. Système du camera

Ce système est composé d'une caméra et 2 servomoteurs constitué le support du camera. En effet, l'installation du caméra est assez facile, if soufie d'installé le câble plat du caméra dans son support dans la carte Raspberry Pi comme indique la figure 43.



Figure 43 : installation du caméra dans la carte

Ainsi, pour câbler les servomoteurs on doit les alimenter par 5V et connecter les broches de signal des servos au broches 29(GPIO05) et 31(GPIO06) du Raspberry Pi, le schéma du câblage est présenté dans la figure 44.

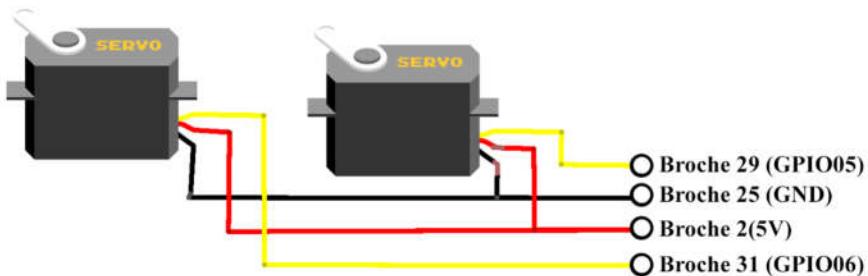


Figure 44 : schéma du câblage des servomoteurs

## 8. Système d'alarme

L'avertisseur sonore qu'on va utiliser peut fonctionner sous tension de 3V. Pour cela on n'a pas besoin d'utiliser un transistor. Connecté avec la broche 22(GPIO 25), la figure 45 ci-dessous montre le schéma de câblage de l'avertisseur sonore.

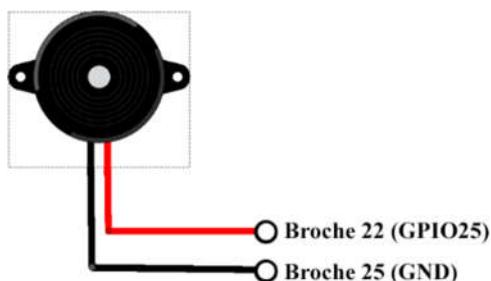


Figure 45 : schéma du câblage de l'avertisseur sonore

## 9. Système d'éclairage

Le système d'éclairage permet d'alimenter 2 LED quand le milieu du robot devient sombre et la vision depuis la caméra devient impossible pour éclairer la zone de vision du caméra, ce système est connecté avec la carte Arduino UNO. Le capteur LDR est alimenté par 5V et câblé avec un résistor de valeur de 1 KΩ, la valeur donnée par le capteur lit par la broche analogique A0. A la fois les LED sont connectées avec la broche numérique 13, comme indique la figure 46 ci-dessous.

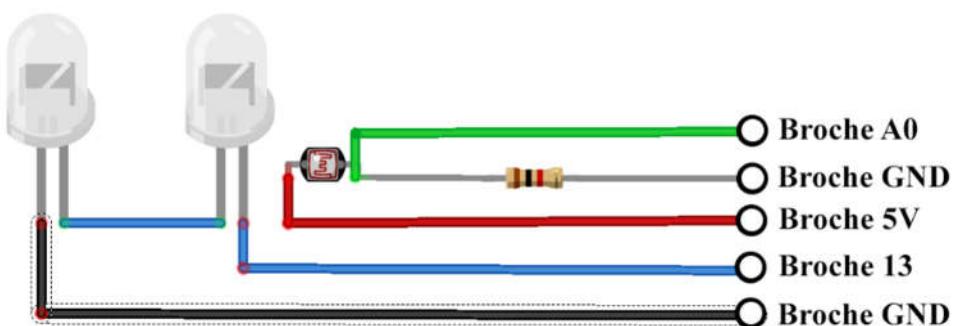


Figure 46 : schéma de câblage du système d'éclairage

### III. Installation et configuration du système

Pour on peut programmer le robot, il faut installer les logiciels et les bibliothèques correspondants. Dans ce qui suit on va expliquer les étapes d'installation de tous les systèmes qu'on va utiliser.

#### 1. Installation du logiciel utilisée dans le pc

##### a. PuTTY

PuTTY est un émulateur de terminal pour Windows permettant la connexion à une machine distante par protocole ssh. Avec ce logiciel, on peut travailler, depuis l'ordinateur personnel, sur le robot, en mode ligne de commandes. La figure 47 présente la fenêtre d'accueil du logiciel.

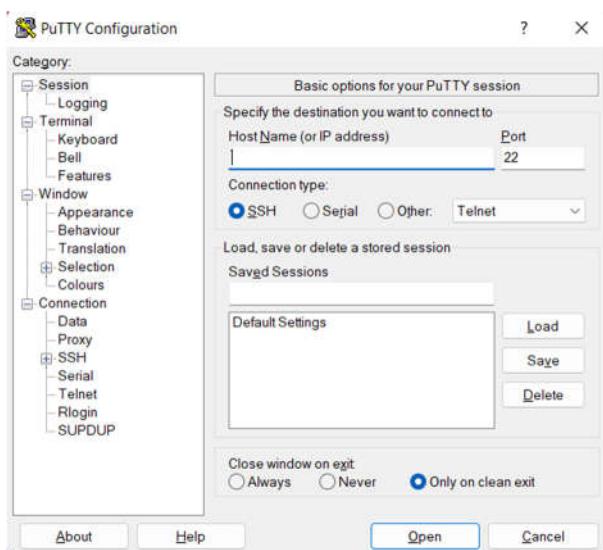


Figure 47 : fenêtre d'accueil du PuTTY

##### b. VNC Viewer

VNC (Virtual Network Computing, littéralement « informatique virtuelle en réseau ») est un système de visualisation et de contrôle de l'environnement de bureau d'un ordinateur distant. Il permet au logiciel client VNC de transmettre les informations de saisie du clavier et de la souris à l'ordinateur distant, possédant un logiciel serveur VNC à travers un réseau informatique. Il utilise le protocole RFB pour les communications.

VNC est indépendant du système d'exploitation : un client VNC installé sur n'importe quel système d'exploitation peut se connecter à un serveur VNC installé sur un système d'exploitation différent ou identique. Il existe des clients et des serveurs VNC pour la plupart des systèmes d'exploitation. Plusieurs clients peuvent se connecter en même temps à un unique serveur VNC.

Parmi les utilisations de ce protocole, on peut citer le support technique à distance, l'administration et la maintenance de systèmes ou logiciels ne permettant que des contrôles graphiques et demandant l'utilisation de la souris ou bien encore la visualisation distante d'applications diverses et variées, dans un but éducatif par exemple. La figure 54 présente la fenêtre d'accueil du logiciel.

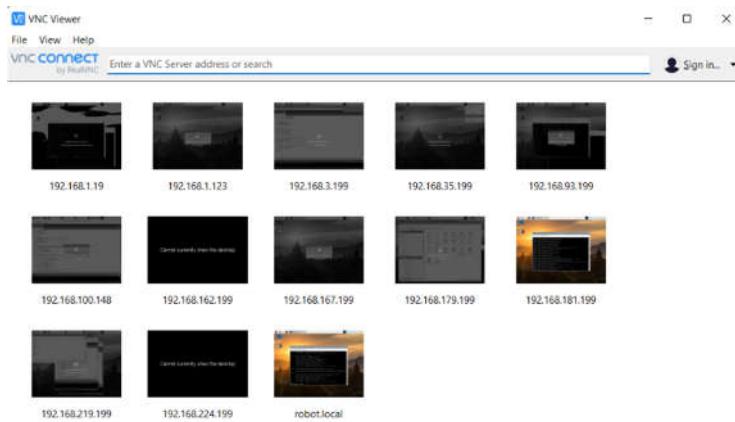


Figure 48 : fenêtre d'accueil du VNC Viewer

### c. Raspberry Pi Imager

Raspberry Pi Imager est une application permettant d'installer un système d'exploitation compatible sur un ordinateur Raspberry Pi. L'outil comprend une liste d'OS téléchargeables directement depuis le site de la Raspberry Foundation, mais il est possible d'installer une image personnalisée. La figure 48 présente la fenêtre d'accueil du logiciel.

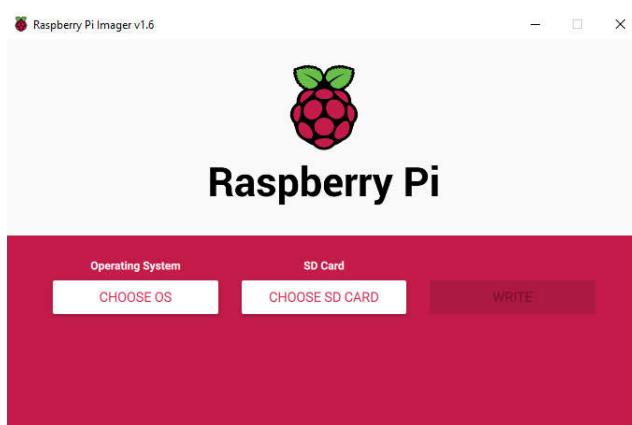


Figure 48 : la fenêtre d'accueil du Raspberry Pi Imager

Autre que ce logiciel est officiel de la société de Raspberry Pi, il est distingué des autres logiciels d'un menu d'option de configuration et installation, cela permet la configuration Wi-Fi manuelle, activation de SSH, définition du nom d'hôte et de mot de passe, définition de la langue d'utilisation ou encore l'activation de la télémétrie, comme indique la figure 49.

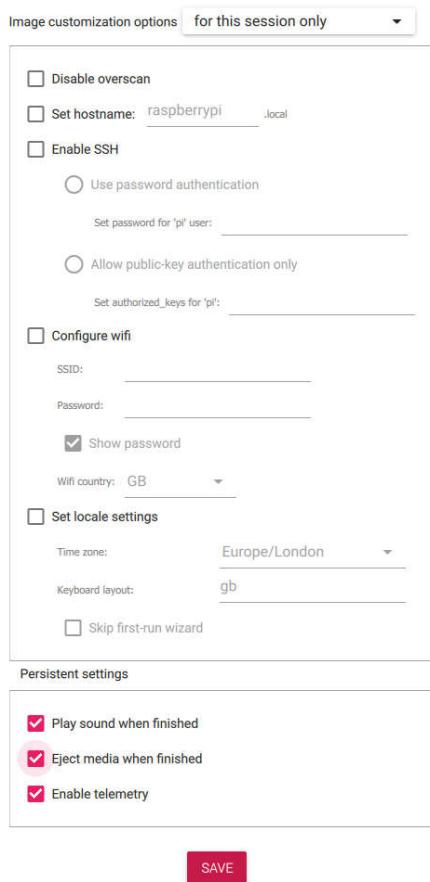


Figure 49 : menu d'options avancées

## 2. Installation et configuration du système d'exploitation

Pour ce système on va utiliser le système d'exploitation Raspberry Pi OS.

### a. Installation du système d'exploitation

Raspberry Pi OS (anciennement nommé Raspbian) est un système d'exploitation libre et gratuit basé sur Debian optimisé pour fonctionner sur les différents Raspberry Pi. Étant donné les ressources limitées des nano-ordinateurs pour lesquels Raspberry Pi OS est principalement destiné, il intègre des logiciels réputés pour être légers et peu gourmands en ressources. Pour installer le Raspberry Pi OS, on suit la méthode ci-dessous :

- Une fois que on a téléchargé le logiciel Raspberry Pi Imager sur le site officiel et la carte SD connectée au PC, on peut alors le lancer pour démarrer.
- On commence par choisir le système à installer. Plusieurs systèmes sont disponibles dans la liste. Pour notre cas on va installer Raspberry Pi OS basée sur Debian 10 (Buster), comme indiqué la figure 50.

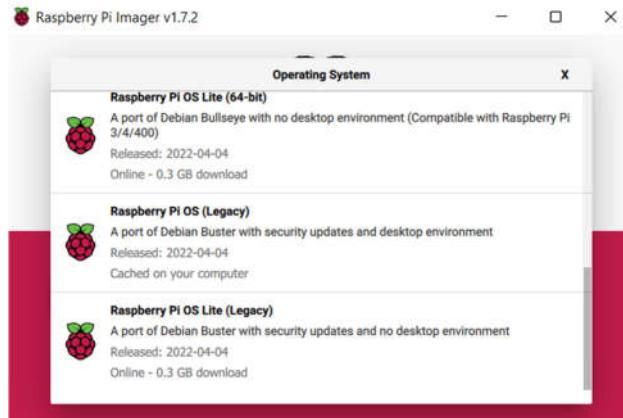


Figure 50 : liste des systèmes d’exploitation

- Choisissez ensuite le média de destination (figure 51).

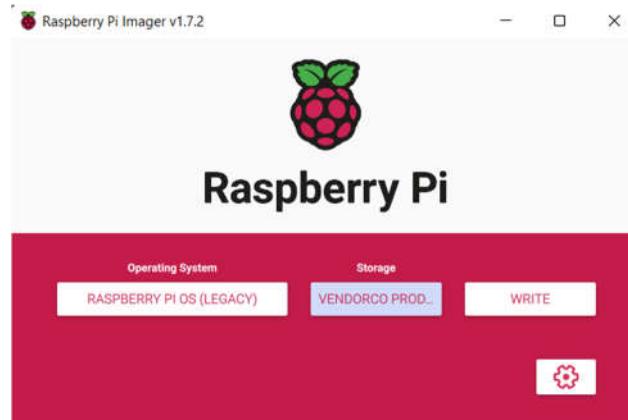


Figure 51 : choisir la media de destination

- On clique sur le bouton ‘Advanced option’ pour configurer nom et le mot de passe de la carte, activer le ssh et configurer la connexion internet (figure 52).



Figure 52 : paramètre du Raspberry Pi OS

- Un message vous demandera de confirmer l'action, choisissez YES (figure 53).

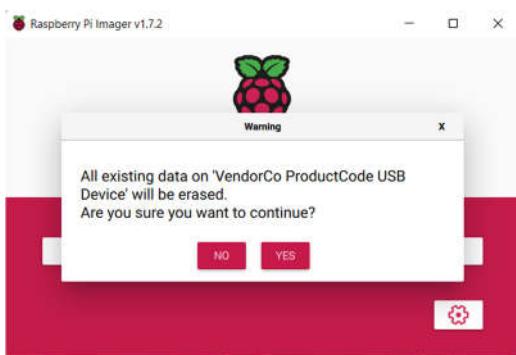


Figure 53 : confirmation de l'installation

- Patiencez le temps de l'écriture de l'image sur le média. Cela prend en général quelques minutes (figure 54).



Figure 54 : l'installation du système

Lorsque l'installation est finie, on peut insérer la carte SD dans le Raspberry Pi et lancer la carte. On va commander la carte avec PuTTY, donc si on connecte le Raspberry Pi et l'ordinateur avec un câble Ethernet, on écrit dans la case de nom d'hôte : « robot.local ». Ou on attend que la carte définie le réseau configuré dans les options avancées d'installation et on écrit dans la case de nom d'hôte l'adresse IP du Raspberry Pi (par exemple : 192.168.1.123). Après la connexion on doit écrire le nom d'utilisateur et le mot de passe configuré dans l'étape d'installation, comme indique la figure 55.



Figure 55 : connexion au robot avec PuTTY

- Une fois la connexion est terminée, on peut accès au ligne de commande du Raspberry Pi comme indiqué dans la figure 56.

```

robot@robot: ~
login as: robot
robot@192.168.100.148's password:
Linux robot 5.15.33-v7+ #1543 SMP Wed Apr 13 15:24:21 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 1 11:19:11 2022 from 192.168.100.74
robot@robot: ~
  
```

Figure 56 : ligna de commande du Raspberry Pi

### b. Configuration du système d'exploitation [1]

Comme tous les systèmes d'exploitation, Raspberry Pi OS doit configurer pour le régler avec notre besoin, mais après on doit vérifier que le système est mis à jour, on lance la commande :

\$ sudo apt update

Cette fonction permet de chercher les mise à jours, si le système trouve des mise à jour, on lance la commande :

\$ sudo apt upgrade

Puis, on va effectuer la configuration du système on écrit :

\$ sudo raspi-config

Cette opération permet de configurer et choisir de plusieurs paramètres comme indiqué la figure 57.

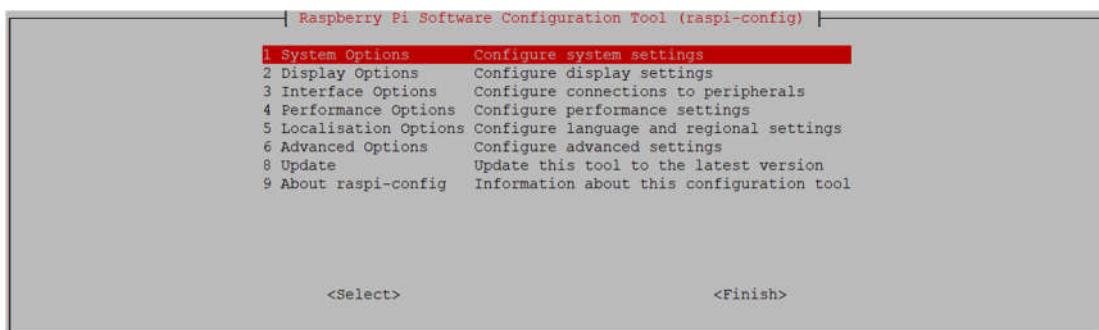


Figure 57 : menu de configuration du Raspberry Pi OS

Pour notre cas on va activer l'interface de camera, le serveur VNC et le port série ; tous ces 3 paramètres se trouvent sous le menu « Interface Options » comme indique la figure 58.

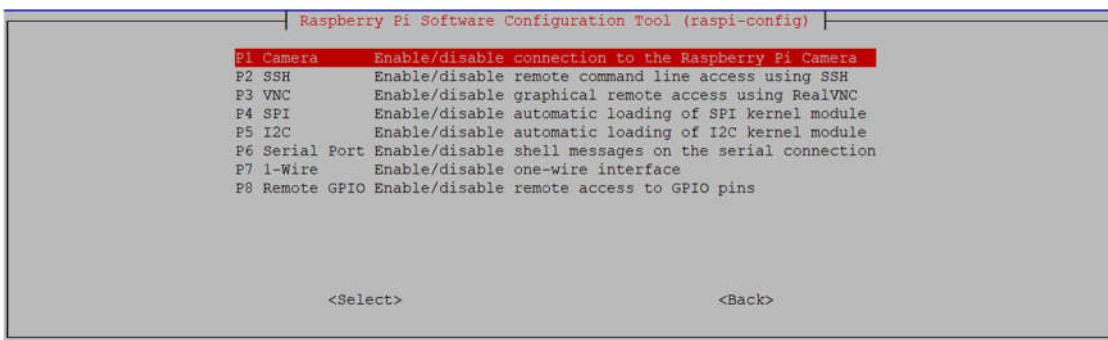


Figure 58 : le menu « Interface Options »

Pour activer l’interface de camera, on clique le bouton « Camera » puis on clique sur « Yes » (figure 59).



Figure 59 : confirmation d’activation de l’interface de camera

Pour activer le serveur VNC, on clique sur le bouton « VNC » puis on clique sur « Yes » (figure 60)



Figure 60 : confirmation d’activation du serveur VNC

Pour activer le port série, on clique sur le bouton « Serial Port », puis on clique sur « No » pour l’accessibilité du connections du Shell dans le port série et « Yes » pour activer le port série matériel (figure 61).

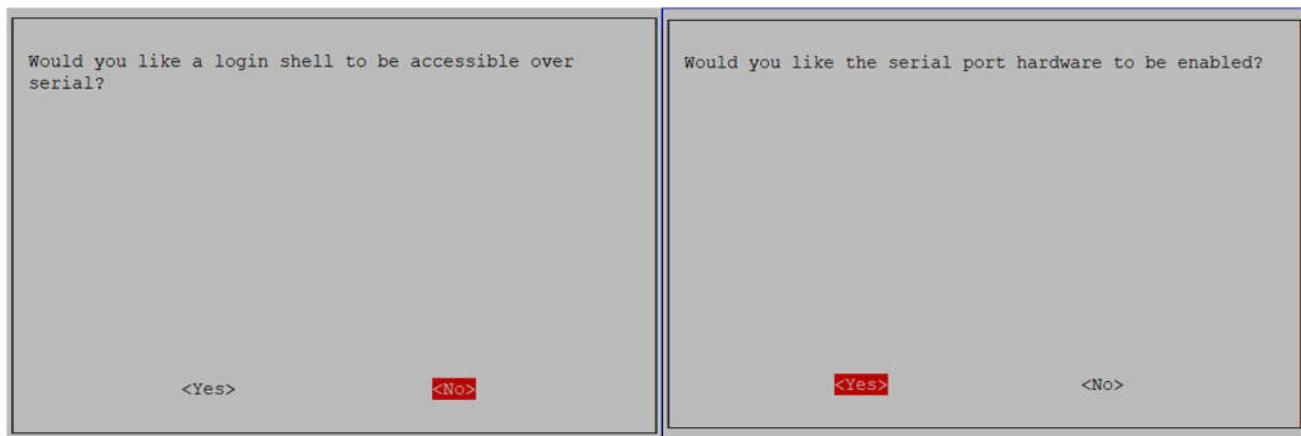


Figure 61 : activation port série

Enfin, on fait redémarrer la carte Raspberry Pi pour effectuer ces changements, on saisit la commande suivante :

```
$ sudo reboot
```

### 3. Installation des logiciel et bibliothèques utilisé dans le robot

#### a. Installation du Python 3 et les paquets nécessaires

Python est un langage de programmation interprété, multiparadigmes et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.

Python est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives. On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique.

La version par défaut dans Raspberry Pi OS est 3.7.3, mais nous avons installé la version 3.9.5 pour l'installer dans notre système, les étapes d'installation sont montrées ci-dessous :<sup>[4]</sup>

- On télécharge le dernier fichier Python avec :

```
$ wget https://www.python.org/ftp/python/3.9.5/Python-3.9.5.tgz
```

- On extrait les fichiers avec :

```
$ sudo tar -zxf Python-3.9.5.tgz
```

- On ouvre le dossier contenant les fichiers extraits :

```
$ cd Python-3.9.5
```

- On exécute la commande de configuration :

```
$ ./configure --enable-optimizations
```

- Une fois terminé, on exécute ces commandes pour installer :

```
$ sudo make -j 4
```

```
$ sudo make altinstall
```

- Pour que la nouvelle version soit la version d'exécution par défaut on exécute ces commandes :

```
$ cd /usr/bin
```

```
$ sudo rm python3
```

```
$ sudo ln -s /usr/local/bin/python3.9 python3
```

Pour installer les paquets nécessaires on va utiliser PIP (Package Installer for Python), on n'a pas besoins d'installer PIP puisque c'est inclus par défaut dans la distribution officielle du Python. Pour notre projet on doit installer les paquets :

- Numpy

NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes. Pour installer Numpy on exécute le code :

```
$ pip3 install numpy
```

- Imutils

C'est Une série de fonctions pratiques pour faciliter les fonctions de traitement d'image de base telles que la traduction, la rotation, le redimensionnement, la squelettisation et l'affichage des images Matplotlib avec Open CV et Python 2.7 ou Python 3. Pour installer imutils on exécute le code :

```
$ pip3 install imutils
```

- Flask

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités. Pour installer Flask on exécute le code :

```
$ pip3 install flask  
$ pip3 install flask-basicAuth==0.2.0
```

- Bibliothèque de capteur US-100 :

Pour fonctionner le capteur us-100 avec la carte Raspberry Pi 4, on doit installer la bibliothèque Adafruit\_Blinka qui fournit le support CircuitPython en Python. On exécute la commande :

```
$ pip3 install adafruit-circuitpython-us100
```

### b. Installation du Open CV et ses bibliothèques

Open CV (Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La bibliothèque Open CV 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.

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 (composantes connexes, GrabCut) ;
- Morphologie mathématique.

Ainsi, Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra)

- Détection de droites, de segment et de cercles par Transformée de Hough
- Détection de visages par la méthode de Viola et Jones
- Cascade de classifieurs boostés
- Détection de mouvement, historique du mouvement
- Poursuite d'objets par mean-shift ou Camshift
- Détection de points d'intérêts
- Estimation de flux optique (Méthode de Lucas–Kanade)
- Triangulation de Delaunay
- Diagramme de Voronoi
- Enveloppe convexe
- Ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés

Pour installer Open CV on doit suivre les étapes ci-dessous :<sup>[6]</sup>

- Pour commencer, on assure de mettre à jour le système en exécutant les commandes ci-dessous:

```
$ sudo apt update  
$ sudo apt upgrade
```

- Ensuite, on installe l'outil de développement CMake nécessaire à la construction d'OpenCV. On exécute la commande :

```
$ sudo apt install build-essential cmake pkg-config
```

- Pour installer des packages d'E/S d'image qui nous permettront de charger divers formats de fichiers image à partir du disque, on exécute la commande :

```
$ sudo apt install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev
```

- Ces packages d'E/S vidéo permettent de lire différents formats de fichiers vidéo et de travailler avec des flux vidéo on exécute la commande :

```
$ sudo apt install libavcodec-dev libavformat-dev libscale-dev  
$ sudo apt install lib4l-dev libxvidcore-dev libX264-dev
```

- Pour afficher des images sur l'écrans d'ordinateur et même développer des interfaces utilisateur graphiques (GUI) pour les projets, on a besoin d'un module appelé highgui. Cela demande à installer toutes les bibliothèques GTK nécessaires. Exécutez les commandes ci-dessous :

```
$ sudo apt install libfontconfig1-dev libcairo2-dev  
$ sudo apt install libgdk-pixbuf2.0-dev libpango1.0-dev  
$ sudo apt install libgtk2.0-dev libgtk-3-dev
```

- Pour effectuer des opérations matricielles, on installe des dépendances supplémentaires. On exécute la commande ci-dessous :

```
$ sudo apt install libatlas-base-dev gfortran
```

- Lorsque tous les logiciels tiers sont installés, Open CV lui-même peut être téléchargé. Il y a deux packages nécessaires ; la version de base et les apports supplémentaires. Après le téléchargement, vous pouvez décompresser les fichiers. On exécute les commandes :

```
$ cd ~  
$ wget -O opencv.zip https://github.com/opencv/opencv/archive/4.5.5.zip  
$ wget -O opencv_contrib.zip  
https://github.com/opencv/opencv\_contrib/archive/4.5.5.zip  
$ unzip opencv.zip  
$ unzip opencv_contrib.zip  
$ mv opencv-4.5.5 opencv  
$ mv opencv_contrib-4.5.5 opencv_contrib  
$ rm opencv.zip  
$ rm opencv_contrib.zip
```

- Avant de commencer la construction proprement dite de la bibliothèque, il reste une petite étape à franchir. On doit créer un répertoire où tous les fichiers de construction peuvent être situés. On exécute les commandes :

```
$ cd opencv  
$ mkdir build  
$ cd build
```

- Il est maintenant temps de passer à une étape importante. Ici, on dit à CMake quoi, où et comment créer Open CV sur la carte Raspberry. On exécute la commande :

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D PYTHON3_PACKAGES_PATH=/usr/lib/python3/dist-packages \
-D OPENCV_GENERATE_PKGCONFIG=ON \
-D BUILD_EXAMPLES=ON ..
```

- On a déjà configuré Open CV pour l'installation. On commence maintenant à compiler avec les quatre cœurs du Pi, en exécutent la commande ci-dessous :

```
$ make -j4
```

- Une fois le processus de compilation terminé sans erreur, on peut maintenant installer Open CV, en exécutent les commandes ci-dessous :

```
$ sudo make install
$ sudo ldconfig
```

- Il reste une chose à faire avant que l'installation d'Open CV soit terminée. Cela réinitialise l'espace d'échange à ses 100 Mo d'origine. La mémoire flash ne peut écrire qu'un nombre limité de cycles. En fin de compte, cela usera la carte SD. Il est donc sage de réduire au minimum l'échange de mémoire. De plus, cela ralentit également l'application. Les commandes finales suppriment les fichiers zip désormais inutiles et redémarrent le système afin que toutes les modifications soient implémentées.

```
$ sudo nano /etc/dphys-swapfile
Mettre CONF_SWAPSIZE=100
$ cd ~
$ rm opencv.zip
$ rm opencv_contrib.zip
$sudo reboot
```

## IV. Conclusion

Au cours de ce chapitre, on a présenté les schémas électriques du solution pour faire fonctionner le robot en commençant par le schéma compét jusqu'aux schémas élémentaire de chaque bloc du système. Ensuite, on a installé les logiciels nécessaires dans le pc pour la communication entre le pc et le robot. Enfin, on a montré les étapes d'installation du système d'exploitation, les logiciels et les bibliothèques utilisé dans le robot. La conception et la réalisation du projet sera l'objectif de du chapitre suivant.

## Chapitre 4

# *Conception Et réalisation*

# Chapitre 4 : Conception et réalisation

## I. Introduction

Dans ce chapitre on va décrire la conception logicielle et mécanique, en expliquant les détails de chaque partie du système. Finalement, on va présenter les étapes de création du robot.

## II. Conception logicielle

Pour obtenir un fonctionnement automatique, organisé et bien précis de notre système, on a besoin d'utiliser de programmer les carte Raspberry Pi et Arduino UNO et de décrire d'une manière concise et facile les opérations de manipulation de données et l'évolution du déroulement du programme en fonctions du temps et des tâches à exécuter. Plus précisément, c'est un code qui permet la détection des personnes, les capter et envoyer par mail ; et au même temps naviguer automatiquement dans le milieu.

### 1. Description du fonctionnement du système

Tout d'abord, on note que le nano ordinateur Raspberry Pi 4 exécute les instructions du programme en boucle fermée avec une grande vitesse en donnant à chaque instruction une durée de temps très courte ensuite il quitte pour exécuter les autres instructions puis revient vite de nouveau pour exécuter l'instruction qui a déjà quitté. Ainsi, pratiquement on contrôle très bien tous les blocs de notre système approximativement en temps réel.

Au départ, après le démarrage du carte et l'initialisation des variables, le système fait la configuration de nom d'hôte pour permet la connexion à distance. On a présenté brièvement et en grosso modo à l'organigramme montré à la figure 62. Et avant de commencer, on note les variables qu'on va les utiliser :

- Dt : variable interne de détection qui indique la présence d'une personne dans le champs du caméra
- Ldr : capteur d'intensité du lumière
- Cu : variable interne de la connexion qui indique que l'utilisateur est connecté
- S1 : variable du servomoteur à l'axe x
- S2 : variable du servo moteur à l'axe y
- An : variable de paramètre que change entre le mode manuel et automatique

- Us1 : capteur de détection d'obstacle en avant
- Us2 : capteur de détection d'obstacle à gauche
- Us3 : capteur de détection d'obstacle à droite
- Fwd : commande de navigation du robot en avant
- Bwd : commande de navigation du robot en arrière
- Rt : commande de rotation du robot à droite
- Lt : commande de rotation de robot à gauche

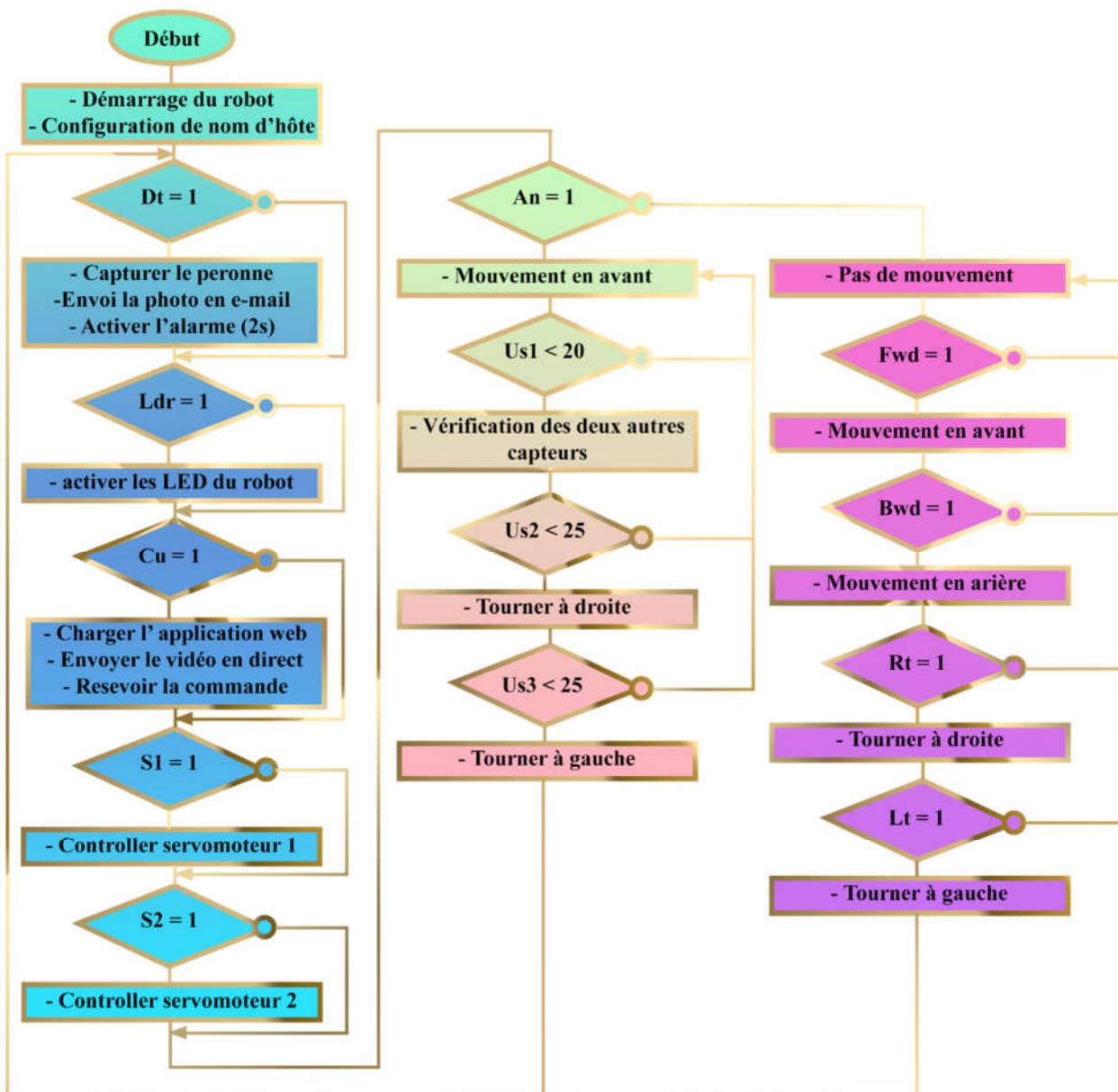


Figure 62 : organigramme du programme de système

Les fonctions qu'on a utilisées pour la programmation des carte Raspberry Pi et Arduino sont :

### a. Démarrage et configuration

Le démarrage du robot permet de fonctionner le programme principale du robot et de configurer l'accès à distance en rendra la page web accessible en accédant à [http://\[ip\]: 5000/](http://[ip]: 5000/) sur n'importe quel ordinateur ou smartphone du réseau. Pour activer cette fonction on tape dans le programme :

```
app.run(host='0.0.0.0', debug=False)
```

L'option --host de flask run, ou le paramètre host de app.run(), contrôle l'adresse écoutée par le serveur de développement. 0.0.0.0 est une valeur spéciale qu'on ne peut pas utiliser directement dans le navigateur, on doit naviguer jusqu'à l'adresse IP réelle du robot sur le réseau. On doit également ajuster le pare-feu pour autoriser l'accès externe au port qui est 5000 dans notre cas. [7]

### b. Détection des personnes

La détection des corps est une technologie informatique qui détermine l'emplacement et la taille des corps humains dans des images (numériques) arbitraires. Elle détecte les traits du corps et ignore tout le reste, comme les bâtiments et les arbres. On va achever la détection des personnes à l'aide de classificateurs en cascade basés sur des fonctionnalités de Haar. Haar Cascade est une approche basée sur Machine Learning où de nombreuses images positives et négatives sont utilisées pour former le classificateur.

- Images positives – Ces images contiennent les images que nous voulons que notre classificateur identifie.
- Images négatives – Images de tout le reste, qui ne contiennent pas l'objet que nous voulons détecter.

L'algorithme peut être expliqué en quatre étapes : [5]

- Calcul des caractéristiques de Haar

La première étape consiste à collecter les fonctionnalités de Haar. Une caractéristique de Haar est essentiellement des calculs qui sont effectués sur des régions rectangulaires adjacentes à un emplacement spécifique dans une fenêtre de détection. Le calcul consiste à additionner les intensités de pixels dans chaque région et à calculer les différences entre les sommes. Voici quelques exemples de fonctionnalités Haar dans la figure 63 ci-dessous.

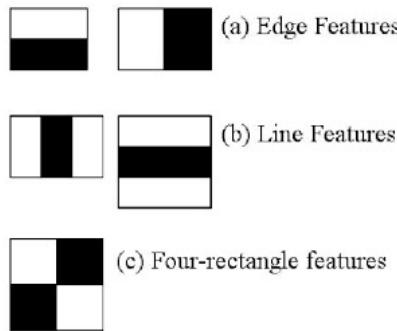


Figure 63 : les types des caractéristiques haar

Ces caractéristiques peuvent être difficiles à déterminer pour une grande image. C'est là que les images intégrales entrent en jeu car le nombre d'opérations est réduit en utilisant l'image intégrale.

- Création d'images intégrales

Sans trop entrer dans les mathématiques derrière cela, les images intégrales accélèrent essentiellement le calcul de ces caractéristiques de Haar. Au lieu de calculer à chaque pixel, il crée à la place des sous-rectangles et crée des références de tableau pour chacun de ces sous-rectangles. Ceux-ci sont ensuite utilisés pour calculer les caractéristiques de Haar comme indiqué dans la figure 64 ci-dessous.

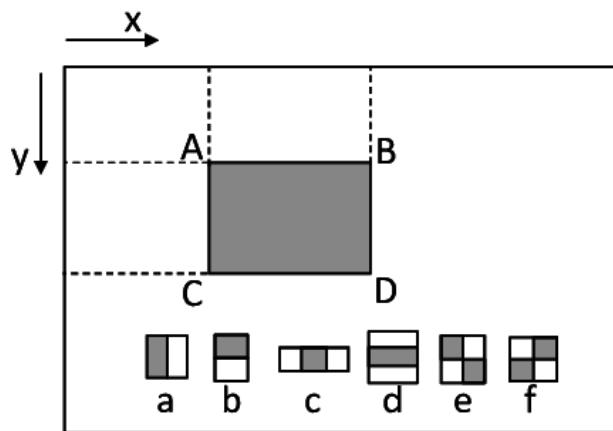


Figure 64 : Illustration du fonctionnement d'une image intégrale

Il est important de noter que presque toutes les fonctionnalités de Haar ne seront pas pertinentes lors de la détection d'objets, car les seules fonctionnalités importantes sont celles de l'objet. Cependant, pour déterminer les meilleures caractéristiques qui représentent un objet parmi les centaines de milliers de caractéristiques de Haar, on doit utiliser Adaboost.

- Utiliser Adaboost

Adaboost choisit essentiellement les meilleures fonctionnalités et forme les classificateurs à les utiliser. Il utilise une combinaison de « classificateurs faibles » pour créer un « classificateur fort »

quel l'algorithme peut utiliser pour détecter des objets.

Les apprenants faibles sont créés en déplaçant une fenêtre sur l'image d'entrée et en calculant les caractéristiques de Haar pour chaque sous-section de l'image. Cette différence est comparée à un seuil appris qui sépare les non-objets des objets. Parce qu'il s'agit de « classificateurs faibles », un grand nombre de caractéristiques de Haar est nécessaire pour la précision afin de former un classificateur puissant. Le fonctionnement de Adaboost est indiqué dans la figure 65 ci-dessous.

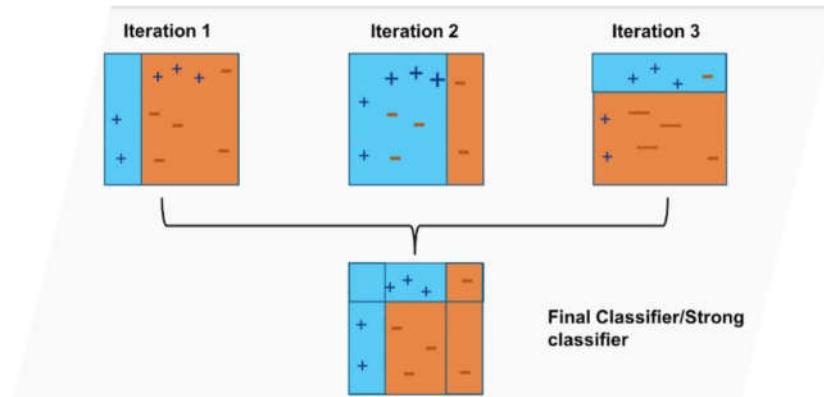


Figure 65 : fonctionnement du l'algorithme Adaboost

- Implémentation de classificateurs cascade

Le classificateur en cascade est composé d'une série d'étapes, où chaque étape est une collection d'apprenants faibles. Les apprenants faibles sont formés à l'aide du boosting, ce qui permet un classificateur très précis à partir de la prédiction moyenne de tous les apprenants faibles.

Sur la base de cette prédiction, le classificateur décide soit d'indiquer qu'un objet a été trouvé (positif), soit de passer à la région suivante (négatif) comme indique la figure 66 ci-dessous. Les étapes sont conçues pour rejeter les échantillons négatifs le plus rapidement possible, car la majorité des fenêtres ne contiennent rien d'intéressant.

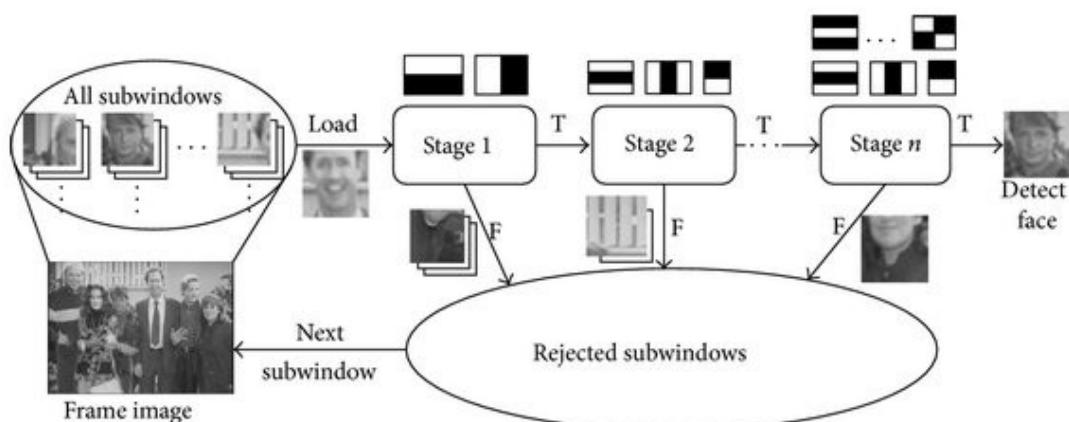


Figure 66 : organigramme de classificateurs en cascade

Pour implémenter le classificateurs haar dans le programme python, on doit suivit les étapes ci-dessous :

- La classe CascadeClassifier pour détecter des objets dans un flux vidéo. On doit extraire le fichier xml contenant le classificateur.

```
object_classifier = cv2.CascadeClassifier("models/fullbody_recognition_model.xml")
```

- Pour chaque image de cadre, l'image sera convertie en image en gray scale que détectera l'image.
- cv2.CascadeClassifier.detectMultiScale détecte des objets de différentes tailles dans l'image d'entrée. Les objets détectés sont renvoyés sous la forme d'une liste de rectangles. Les paramètres de detectMultiScale sont montré dans le tableau 11 ci-dessous.

Tableau 11: parameters de detectMultiScale

Nom du fonction	Définition
image	Matrice de type CV_8U contenant une image où les objets sont détectés
objects	Vecteur de rectangles où chaque rectangle contient l'objet détecté, les rectangles peuvent être partiellement à l'extérieur l'image originale
scaleFactor	Paramètre spécifiant de combien la taille de l'image est réduite à chaque échelle d'image.
minNeighbors	Paramètre spécifiant le nombre de voisins que chaque rectangle candidat doit avoir pour le conserver.
flags	Paramètre ayant la même signification pour une ancienne cascade que dans la fonction cvHaarDetectObjects. Il n'est pas utilisé pour une nouvelle cascade.
minSize	Taille d'objet minimale possible. Les objets plus petits que cela sont ignorés
maxSize	Taille d'objet maximale possible. Les objets plus grands que cela sont ignorés. Si maxSize == minSize, le modèle est évalué sur une seule échelle.

- Tous les rectangles sont lus dans la boucle et créés, avec cv2.rectangle.

```
def fullbody_classification(img):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    bodies = fullbody_cascade.detectMultiScale(
        gray,
        scaleFactor = 1.03,
        minNeighbors = 3,
        minSize = (30, 30),
        flags=cv2.CASCADE_SCALE_IMAGE)
```

```
for (x, y, w, h) in bodies:
    cv2.rectangle(img, (x,y), (x+w, y+h), (255, 0 , 0), 3)
```

- cv2.VideoCapture(index) démarre la capture vidéo, où index est l'emplacement de la vidéo, pour la caméra vidéo, index=0.

```
cap = cv2.VideoCapture(0)
```

- Pendant que la vidéo est capturée, l'image est constamment extraite et traitée dans fullbody\_recognition, puis affichée à l'aide de cv2.imshow(tag, img).
- cv2.waitKey est une commande pour fermer la boucle while avec un événement externe.

```
while cap.isOpened():
    img = cap.read()
    fullbody_classification(img)
    cv2.imshow('img', img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
```

### c. Déclenchement du système d'alarme

Après que le corps humain a été détecté, doit envoyer la capture au utilisateur par e-mail. On a utilisé les bibliothèques smtplib et email.MIME (Multipurpose Internet Mail Extensions). [4]

- Pour envoyer un mail on doit connecter à un compte gmail définir le mail récepteur.

```
fromEmail = 'frommail@gmail.com'
fromEmailPassword = 'password'

toEmail = 'tomail@gmail.com'
```

- Après, on doit attacher la capture du corps détecté dans le mail.

```
msgImage = MIMEImage(image)
msgImage.add_header('Content-ID', '<image1>')
msgRoot.attach(msgImage)
```

- Enfin, en effectue l'envoi de mail.

```
smtp = smtplib.SMTP('smtp.gmail.com', 587)
smtp.starttls()
smtp.login(fromEmail, fromEmailPassword)
smtp.sendmail(fromEmail, toEmail, msgRoot.as_string())
smtp.quit()
```

### d. Système d'éclairage d'intelligent

Le phénomène de photoconductivité est l'augmentation de la conductivité électrique d'un semi-conducteur lorsqu'il est soumis à un rayonnement électromagnétique. Le principe de fonctionnement d'une photorésistance est basé sur l'apparition de porteurs de charge mobiles (électrons) suite à l'absorption par le semi-conducteur de l'énergie lumineuse ; par conséquent, sa résistance diminue,

c'est-à-dire qu'il y a une conductivité supplémentaire comme indique la courbe dans la figure 67 ci-dessous.

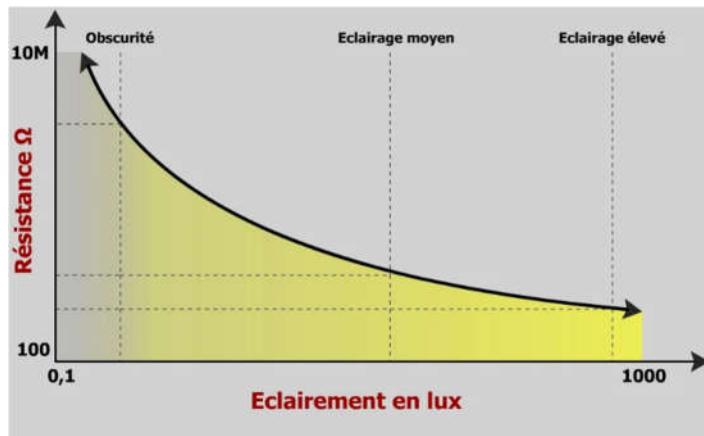


Figure 67 : Courbe de l'éclairement et variation de la résistance (LDR)

Pour programmer le LDR on doit déterminer la valeur d'éclairement ou on doit allumé les Leds ? l'exemple du programme est indiqué ci-dessous.

```
analogValue = analogRead(LIGHT_SENSOR_PIN);
if (analogValue < 20)
    digitalWrite(LED_PIN, HIGH);
else
    digitalWrite(LED_PIN, LOW);
```

#### e. Visualisation et contrôle par l'application mobile

Pour achever la visualisation et le contrôle du robot on doit avoir une communication entre le programme principale et l'application mobile pour envoyer et recevoir les données.

- Programme principale

Dans cette partie on doit envoyer le vidéo en directe et préciser les paramètres de contrôle que le robot applique en cas d'activation d'un certain bouton, on doit utiliser la fonction app.route du flask pour mapper les URL à une fonction spécifique qui gérera la logique de cette URL. Les infrastructures Web modernes utilisent des URL plus significatives pour aider les utilisateurs à se souvenir des URL et à simplifier la navigation.<sup>[7]</sup>

```
@app.route('/')
@app.route('/video_feed')
def video_feed():
    return Response(gen(video_camera),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
```

```
@app.route('/up_side')
def up_side():
    data1 = "FORWARD"
    motor1.forwards(60)
    motor2.forwards(60)
    return 'true'
```

- Programme de l'application web

Dans cette application on doit afficher le vidéo envoyé par la caméra et les boutons pour qu'on peut commander le robot en utilisant la langue HTML.

```
<img id="bg" "{{ url_for('video_feed') }}" />

```

Après, on doit lier le bouton du contrôle avec le URL précisé par Flask pour achever la communication avec le programme principal du robot par la langue Java Script.

```
$("#Forwardmotor").on("mousedown", function () {
    $.get('/upside');
});
```

#### f. Contrôle du servomoteur

Le pilotage du servomoteur se fait à l'aide d'un signal modulé en largeur d'impulsions (MLI ou PWM – Pulsed Width Modulation), c'est une méthode permettant de générer un signal rectangulaire de période T fixe de 20 ms et dont le rapport cyclique est le rapport du temps haut sur la période, noté,  $RC = \frac{th}{T}$  est variable, comme indique la figure 68 ci-dessous.

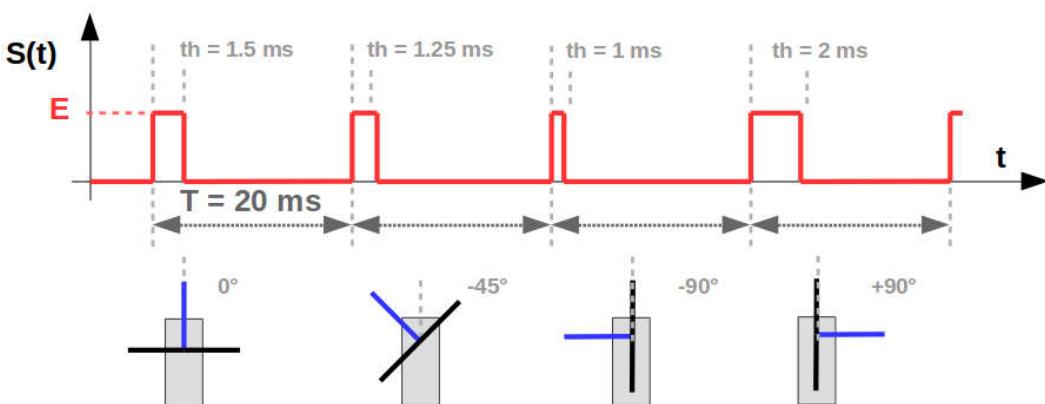


Figure 68 : le fonctionnement du servomoteur par rapport au RC

L'angle que peut atteindre le servomoteur est compris entre  $-90^\circ$  et  $+90^\circ$  par rapport à l'axe principal du servomoteur. La position angulaire de  $0^\circ$  correspond à un temps haut de 1,5 ms (par rapport à 20 ms de période). Lorsque la durée du temps haut du signal de commande diminue, jusqu'à 1 ms, l'angle est négatif (jusqu'à  $-90^\circ$ ). Lorsque la durée du temps haut du signal de commande augmente, jusqu'à 2 ms, l'angle est alors positif (jusqu'à  $90^\circ$ ).

Pour programmer le servo moteur on suit les étapes suivantes :

- Le servo accepte un rapport cyclique de 5 à 10 % comme signal.

```
MIN_DUTY = 5
MAX_DUTY = 10
```

- On convertie le nombres de degrés en rapport cyclique, par rapport à MAX\_DUTY et MIN\_DUTY.

```
def deg_to_duty(deg):
    return (deg - 0) * (MAX_DUTY - MIN_DUTY) / 180 + MIN_DUTY
```

- On définit le signal PWM sur 50Hz. On démarre ensuite le servo en envoyant un rapport cyclique de 0%, indiquant essentiellement que le servo ne prend aucune commande pour le moment.

```
pwm = GPIO.PWM(servo_signal_pin, 50)
pwm.start(0)
```

- Ensuite, on a tester le fonctionnement du servo moteur de 0° à 180° à l'aide d'une boucle for et la fonction range() (cette fonction de python est exclusive, on a donc définie sur 181). Enfin, on obtient la valeur du rapport cyclique en traduisant les degrés souhaités à l'aide de deg\_to\_duty(deg). Après avoir reçu la valeur duty\_cycle, on envoi le signal au servo en utilisant pwm.ChangeDutyCycle(duty\_cycle).

```
for deg in range(181):
    duty_cycle = deg_to_duty(deg)
    pwm.ChangeDutyCycle(duty_cycle)
```

### g. Contrôle du moteur

La disposition en pont en H est généralement utilisée pour inverser la polarité/la direction du moteur, mais peut également être utilisée pour « freiner » le moteur, lorsque le moteur s'arrête soudainement, lorsque les bornes du moteur sont court-circuitées, ou pour laisser le moteur « fonctionnement libre » jusqu'à l'arrêt, car le moteur est effectivement déconnecté du circuit. Le tableau 12 résume le fonctionnement, avec A-D correspondant au schéma du figure 69 ci-dessus.

Tableau 12 : table de vérité de la fonction du port en H

S1	S2	S3	S4	Résultat
1	0	0	1	Le moteur tourne à droite
0	1	1	0	Le moteur se déplace vers la gauche
0	0	0	0	Côtes de moteur
1	0	0	0	
0	1	0	0	
0	0	1	0	

0	0	0	1	
0	1	0	1	Freins moteur
1	0	1	0	
X	X	1	1	Court-circuit
1	1	X	X	

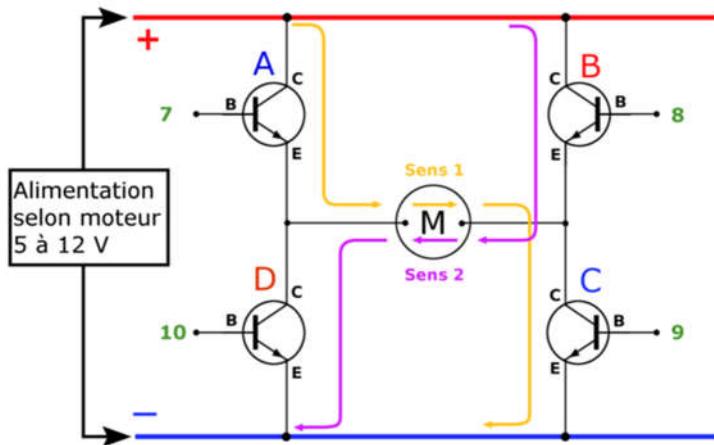


Figure 69 : les deux états de base du port en H

Pour programmer un moteur on doit indiquer les sorties du carte somme indiqué la tableau 12 avec high signifie 1 et low signifie 0. Aussi on défie le port enable en 100Hz pour commander la vitesse du moteur.

```

GPIO.output(in1,GPIO.HIGH)
GPIO.output(in2,GPIO.LOW)
pwm = GPIO.PWM(en, 100)
pwm.ChangeDutyCycle(50)
    
```

#### h. Evitement d'obstacle

Le principe de fonctionnement du capteur est entièrement basé sur la vitesse du son. Comme il est indiqué au figure 70, voilà comment se déroule une prise de mesure :

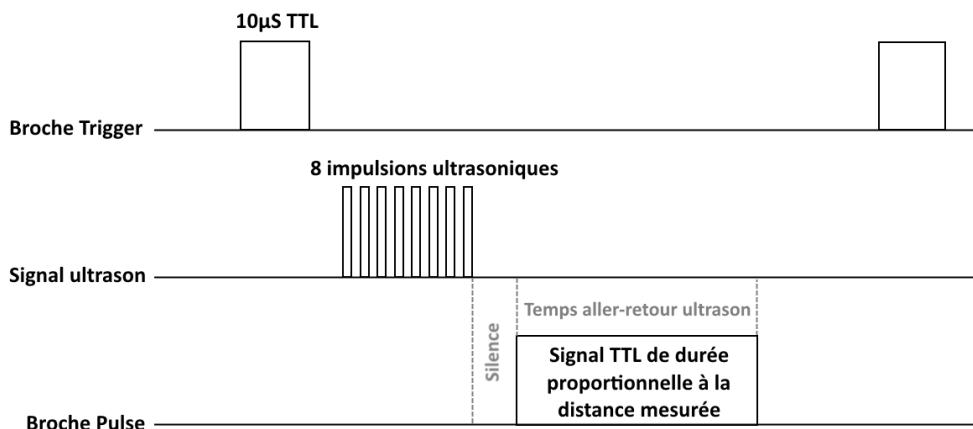


Figure 70 : fonctionnement du capteur ultrason

- On envoie une impulsion HIGH de 10µs sur la broche TRIGGER du capteur.
- Le capteur envoie alors une série de 8 impulsions ultrasoniques à 40KHz .
- Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retourne dans l'autre sens vers le capteur.
- Le capteur détecte l'écho et clôture la prise de mesure.

Le signal sur la broche ECHO du capteur reste à HIGH durant les deux dernières étapes, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance.<sup>[8]</sup>

Pour calculer la distance mesurée par le capteur, on utilise l'équation :

$$v = \frac{d \times 2}{t} \quad \text{donc} \quad d = \frac{v \times t}{2}$$

- t étant le temps vu au-dessus entre le début de l'émission et le début de la réception.
- v étant la vitesse du son qui est de 330 m.s<sup>-1</sup>.
- on devise par 2 puisque le son fait le parcourt du distance une fois lorsqu'il sortit de l'émission et l'autre fois en retour au réception.

On peut calculer la distance en cm par l'équation  $d = \frac{t/v}{2}$  avec :

- v = 29 µs.cm<sup>-1</sup>
- t est calculé en µs

Le programme du calcul du distance est :

```

digitalWrite(trigger, LOW);
delayMicroseconds(2);
digitalWrite(trigger, HIGH);
delayMicroseconds(10);
digitalWrite(trigger, LOW);

responseTime = pulseIn(echo, HIGH);

responseDistance = responseTime / 29 / 2;

```

## 2. Description du fonctionnement d'application web

L'application web à développer est simplement un code à ajouter au programme principal à développer en python. Ce code est à écrire en langages HTML, CSS et JS comme on a indiqué précédemment. Pour en faire, on va utiliser Visual studio pour le rédiger et l'enregistrer puis le tester sur un navigateur jusqu'on obtient un code convenable qui répond parfaitement à notre besoin à

déplacer dans la carte Raspberry Pi. On a décidé de créer une application d'une page web. Elle présente le vidéo direct du caméra, les boutons de contrôle des servomoteurs, commande des moteurs et l'interrupteur qui change entre la navigation autonome ou manuelle. Et le code de style développé par le langage CSS qui est partagé entre la plupart des éléments du page web sera ajouté dans un fichier séparé à appeler à l'entête de page web. Le code JavaScript est nécessaire ici pour communiquer entre l'application web et le programme python avec Flask, il envoie des requêtes en arrière-plan en employant des instructions Ajax puis reçoit le résultat et fait la modification nécessaire des éléments actifs tels que les boutons de commande.

L'application s'affiche sur tout l'écran et les éléments html sont placés horizontalement pour occuper tout l'espace disponible. La figure 71 ci-dessous montre des photos la page de l'application web ouverte sur le navigateur « Google Chrome » sur un ordinateur.



Figure 71 : UI de l'application web

### III. Conception mécanique

Pour la construction de notre robot, on veut construire un boîtier rigide, robuste, durable et facile à ouvrir et fermé pour le maintien des composants électrique. On a engagé un expert pour nous aider à réaliser la vision et le but à atteindre.

Le boîtier est constitué de 5 pièces, fabriqué de la tôle, « base plate », 2 « side plate », « top cover0 » et « top cover1 ». La conception est délayée dans la figure 71 ci-dessous. La pièce « base plate » est la plus basse pièce, elle est connectée avec les 2 « side plate » qui sont attaché à côté. Les 2 pieces « top cover0 » et « top cover1 » sont attaché aussi au 2 side plate, avec la pièce « top cover1 » est en haut.

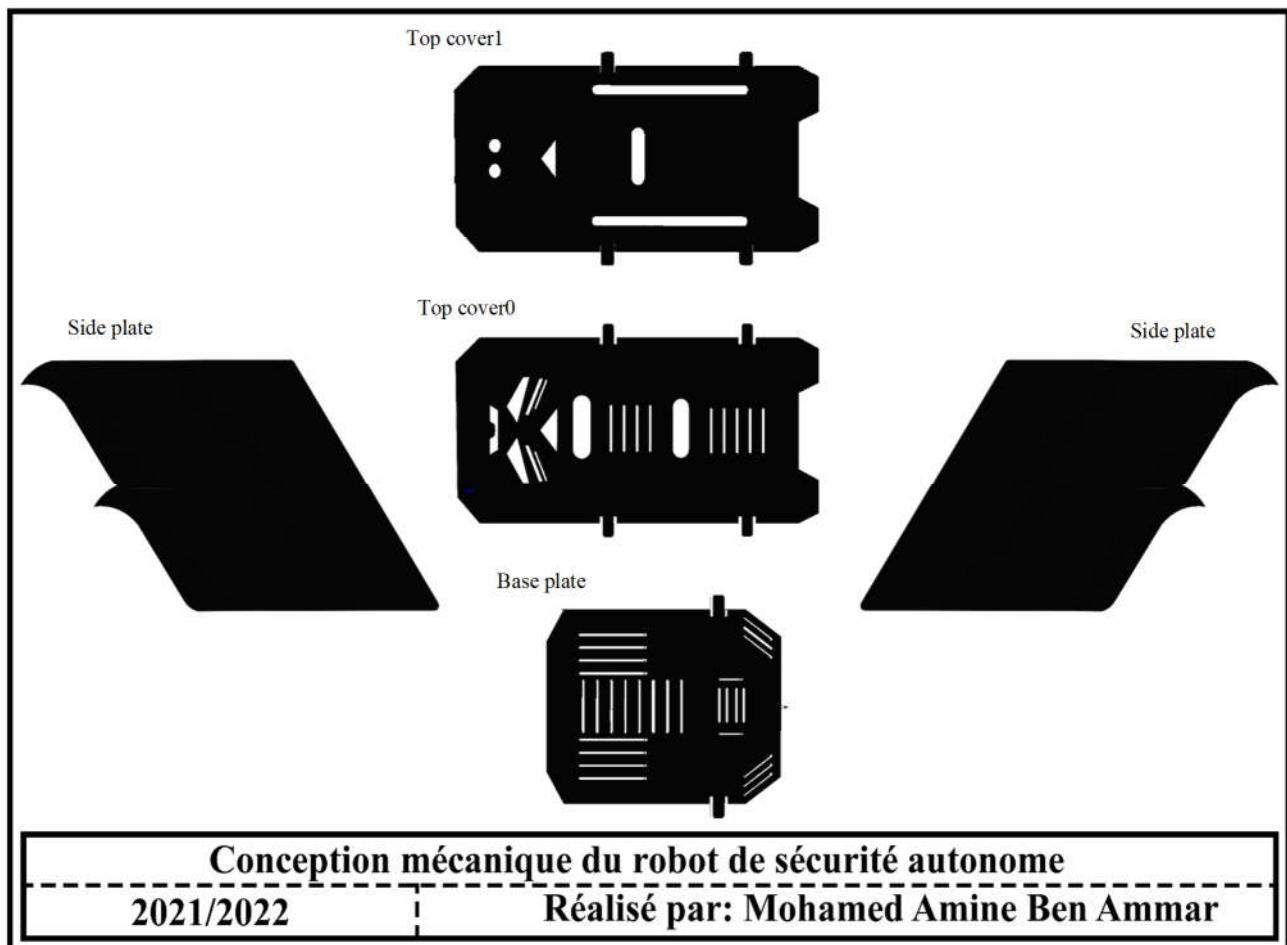


Figure 71 : conception mécanique du robot

## IV. Les étapes de réalisation du robot

Après la réalisation du boîtier du robot, et on se basent sur les schémas du câblage expliquer sur le chapitre 3. On doit rassembler les pièces du boîtier, installer et câbler les composants électroniques en se basant sur les schémas.

L'installation du robot était faite sur 4 étapes : chaque étape contient l'installation des composants bien déterminé sur l'une des pièces du robot.

### 1. Etage 0

Pour effectuer un mouvement stable du robot, on a installé les 4 moteur et ses roues dans l'étage 0 du robot qui est la partie bas de la plaque « base plate », chaque moteur est installé avec une roue dans un côté, comme indiqué la figure 72 ci-dessous. Les câbles des moteur vont passer à travers la plaque pour les préparer à l'installation de l'étage 1.

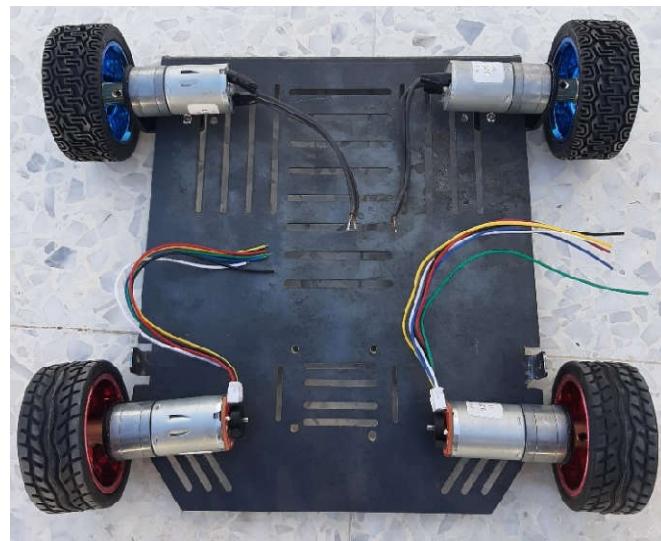


Figure 72 : étage 0 du robot

## 2. Etage 1

Dans cet étage on a câblé les 2 moteurs à droite avec l'entrée droite de la carte L298N, et les 2 autres moteurs avec l'autre entrée de la carte. On a installé aussi les 2 batteries de 5V et 12V comme indiqué la figure 73. On a installé un interrupteur entre la batterie 12V et le driver moteur pour couper le courant en cas de surcharge de la batterie. Aussi on a séparé les composants électriques au métal du boîtier par une couche isolante pour protéger l'utilisateur et les composants dans un cas de court-circuit. Les composants électroniques de cet étage sont connectés avec l'étage 2 par le câble d'alimentation de 5V pour la carte Raspberry Pi et les 6 broches sorties du driver moteur pour permettre le contrôle des moteurs.

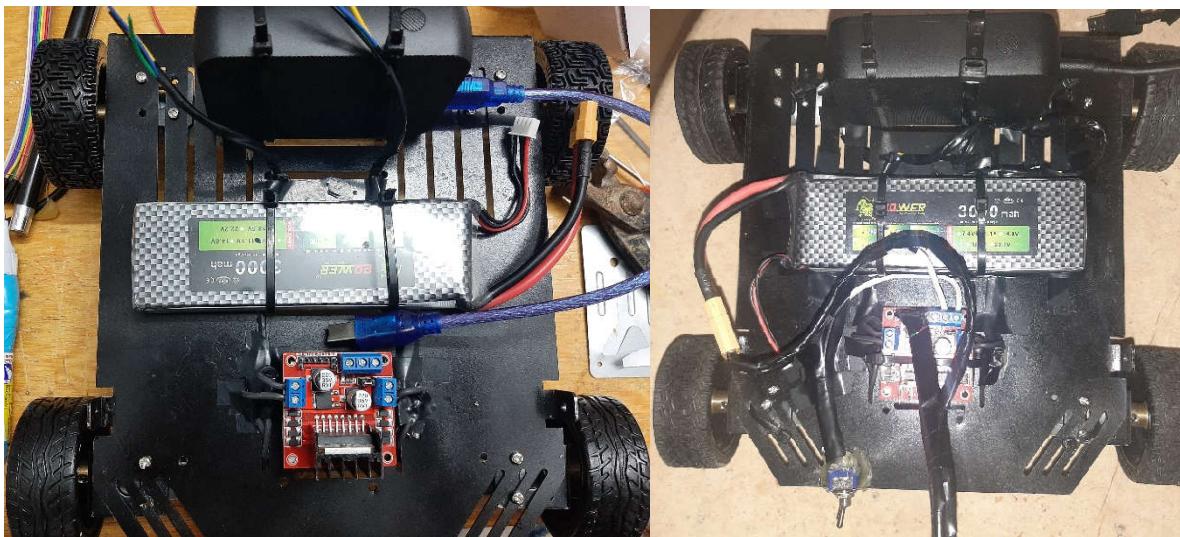


Figure 73 : étage 1 du robot

### 3. Etage 2

On a installé dans cet étage les 2 LED (figure 74), les 2 capteur ultrason SRF05 et on a les connecté avec la carte Arduino UNO. Au même temps on a installé les 2 cartes de contrôle Raspberry Pi 4 et Arduino UNO. Aussi, on a connecté les 2 cartes pour la connexion série et l'alimentation du carte Arduino. Enfin, on a terminé le câblage nécessaire des autres composants déjà installer comme indique la figure 75 ci-dessous.

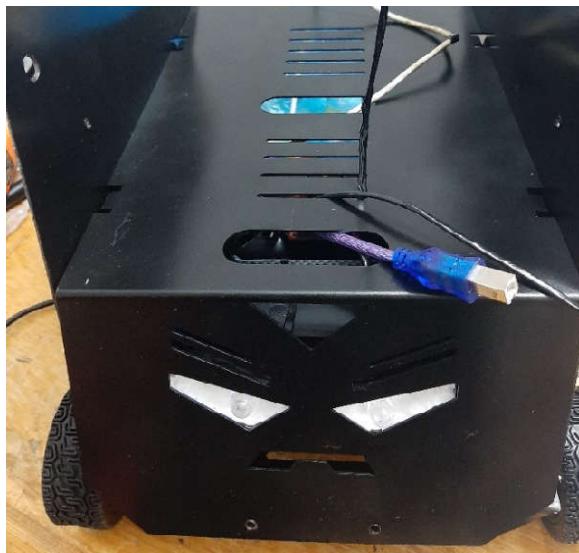


Figure 74 : installation des LED

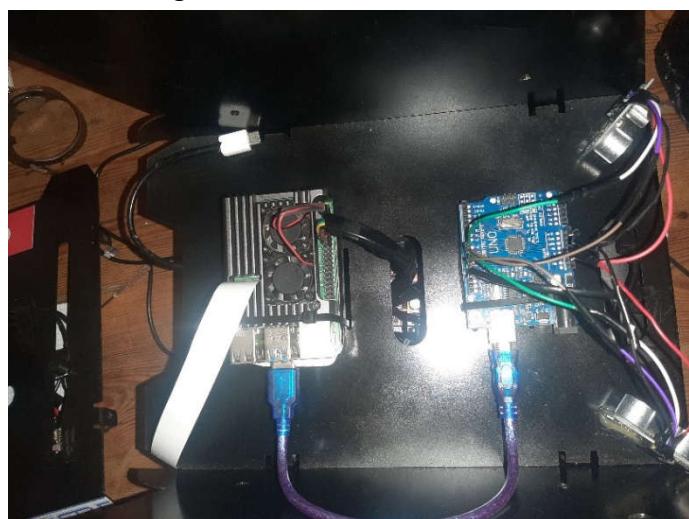


Figure 75 : installation et câblage d'étage 2

### 4. Etage 3

On a installé dans cet étage le capteur us100 en avant du robot. Au même temps on a installé la caméra et son support composé de 2 servomoteurs, le capteur LDR et l'avertisseur sonore comme indique la figure 77.



Figure 76 : installation d'étage 3

Afin de terminer l'installation, on a câbler le capteur LDR avec la carte Arduino, et les autres composants tel que la caméra et les servomoteurs avec la carte Raspberry. On a essayé à avoir un montage propre et facile à maintenir comme indique la figure 78.

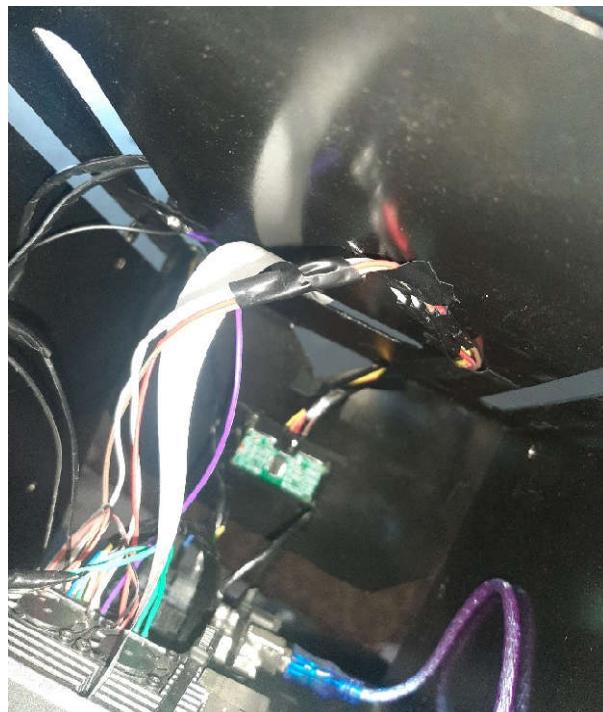


Figure 78 : câblage d'étage 3

Afin de câbler et installer tous les composants du robot, on a installer et tester les programmes développer dans le début de ce chapitre. On a avoir à la fin un robot de sécurité autonome entièrement fonctionnel.

Le produit final du projet est montré dans la figure 79.



Figure 79 : la réalisation finale du système

## V. Conclusion

Au cours de ce chapitre, on a présenté et expliqué le concept logiciel de la solution pour achever l'objectif d'avoir un robot de sécurité autonome en commençant par la présentation du fonctionnement principale jusqu'à l'explication de chaque partie du programme. Puis, on a présenté brièvement le concept mécanique. Et enfin, on a réalisé un prototype du robot complet, en décrivent les étapes d'installation réalisé.

# *Conclusion générale*

Au cours de l'étude et de la réalisation de ce projet de fin de parcours, on a exploité les connaissances acquises durant toute la formation à l'Institut Supérieur des Etudes Technologiques de Mahdia, et les connaissances issues des recherches des robot de sécurité autonome (ASR) qu'on a effectué tout seul pour apprendre des nouvelles choses et suivre l'évolution des technologies, qui ne cessent plus à se développer d'un jour à autre, pour examiner bien la problématique posée, qui consiste à réaliser un robot de sécurité autonome avec une détection intelligente des personnes, tout en ajoutant des améliorations aux technologies existantes dans ce domaine.

Ainsi, on a commencé dans un premier chapitre à présenter la société incubatrice de ce projet et de faire une étude bibliographique pour prendre une bonne idée sur ce qui existe déjà afin d'expliciter la problématique et de créer de la valeur ajoutée. Puis, dans un deuxième chapitre, on a présenté notre solution à cette problématique en faisant une étude complète et détaillée des différents blocs du système et le choix des matérielles utilisés. Aussi, dans un troisième chapitre, on a présenté les schémas électrique associés à notre solution, et on a décrit l'installation des logiciels et bibliothèque utilisé pour programmer le robot. Et enfin, dans un dernier chapitre, on a expliqué la conception logicielle en expliquant chaque partie du programme, et on a validé notre travail par la réalisation pratique d'un prototype. On peut dire qu'on a achevé notre travail avec succès. Notre solution permet la navigation autonome avec détection intelligente des personnes, en plus du contrôle et visualisation à distance du système en temps réel.

On souligne que ce projet était une bonne occasion pour approfondir et enrichir nos connaissances théoriques et pratiques. On a acquis sûrement une bonne expérience au cours de son étude et réalisation. Mais en fait ce n'est rien à ajouter de plus. Au contraire, on peut faire d'autres améliorations si on parle des perspectives. On peut penser par exemple à connecter le robot à un serveur contient les visages des personnes connue et il va détecte seulement les étrangers. On peut aussi ajouter un assistant vocal qui peut communiquer et comprendre les ordres de l'utilisateur ou même communiquer avec les étrangers pour montrer les zones interdites. Aussi on peut exploiter la communication à distance, qu'on peut communiquer avec un serveur pour enregistrer les images et les vidéos. Beaucoup d'idées, mais à ce stade, on n'a que valoriser ce travail et penser à le transformer en un produit à commercialiser et l'exploiter au moins dans notre marché Tunisien local. On est capable d'apporter le plus et de contribuer à l'évolution de notre pays et notre économie.

# *Netographie*

- [1] www.raspberrypi.com
- [2] picamera.readthedocs.io
- [3] www.agvnetwork.com
- [4] docs.python.org
- [5] docs.opencv.com
- [6] www.roboticsbackend.com
- [7] flask.palletprojects.com
- [8] lesrn.adafruit.com
- [9] qengineering.eu
- [10] www.enovarobotics.eu
- [11] www.securite-mag.com
- [12] smprobotics.com
- [13] www.u-become.com

# *Annexes*

# Annexe 1 : fiche technique de Raspberry Pi

## 2.1 Hardware

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 1, 2 and 4 Gigabyte LPDDR4 RAM options
- H.265 (HEVC) hardware decode (up to 4Kp60)
- H.264 hardware decode (up to 1080p60)
- VideoCore VI 3D Graphics
- Supports dual HDMI display output up to 4Kp60

## 2.2 Interfaces

- 802.11 b/g/n/ac Wireless LAN
- Bluetooth 5.0 with BLE
- 1x SD Card
- 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution
- 2x USB2 ports
- 2x USB3 ports
- 1x Gigabit Ethernet port (supports PoE with add-on PoE HAT)
- 1x Raspberry Pi camera port (2-lane MIPI CSI)
- 1x Raspberry Pi display port (2-lane MIPI DSI)
- 28x user GPIO supporting various interface options:
  - Up to 6x UART
  - Up to 6x I2C
  - Up to 5x SPI
  - 1x SDIO interface
  - 1x DPI (Parallel RGB Display)
  - 1x PCM
  - Up to 2x PWM channels
  - Up to 3x GPCLK outputs

## 5.1 GPIO Interface

The Pi4B makes 28 BCM2711 GPIOs available via a standard Raspberry Pi 40-pin header. This header is backwards compatible with all previous Raspberry Pi boards with a 40-way header.

### 5.1.1 GPIO Pin Assignments

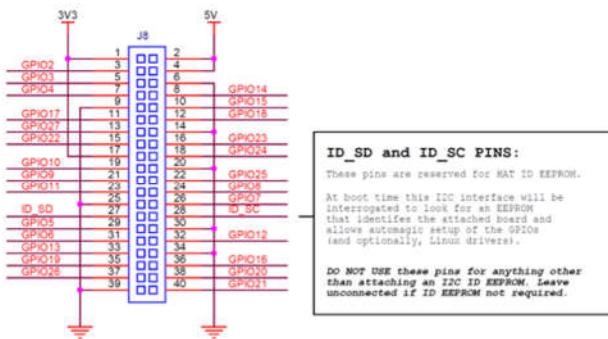


Figure 3: GPIO Connector Pinout

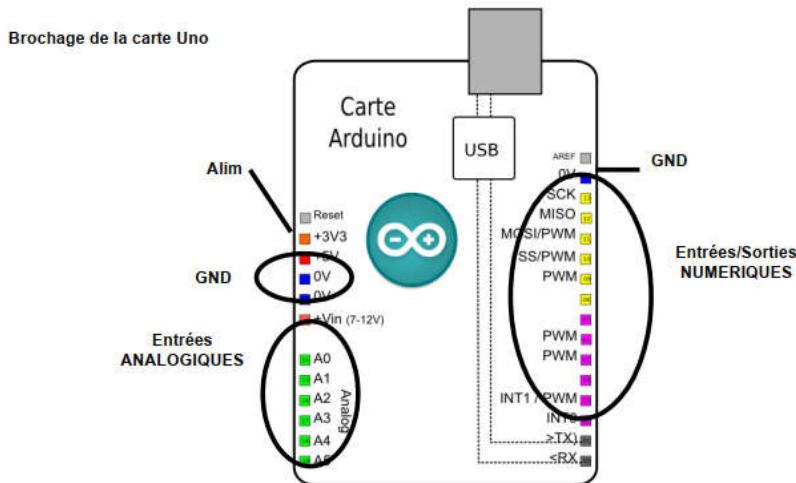
As well as being able to be used as straightforward software controlled input and output (with programmable pulls), GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI.

In addition to the standard peripheral options found on legacy Pis, extra I2C, UART and SPI peripherals have been added to the BCM2711 chip and are available as further mux options on the Pi4. This gives users much more flexibility when attaching add-on hardware as compared to older models.

## Annexe 2 : fiche technique d'Arduino UNO

### Synthèse des caractéristiques

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Consommation maxi admise sur port USB (5V)	500 mA avant déclenchement d'un fusible
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM pour commander les moteurs)
Broches d'entrées analogiques	6 (utilisables aussi en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA par sortie, mais ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Mémoire Programme Flash	32 Ko
Mémoire RAM (mémoire volatile)	2 Ko
Mémoire EEPROM (mémoire non volatile)	1 Ko
Vitesse d'horloge	16 MHz



## Annexe 3 : fiche technique du capteur SRF05

### Pins Details

5-Pin, One each for VCC, Trigger, Echo, Out and Ground.

**Trig** : Plusing a HIGH on this pin will send a Pulse

**Echo** : The pulse length is proportional to the distance from the obstacle  
( 100usec to 25ms, times out if 30ms )

### Specifications

Check the datasheet for details and graphs but these are the highlights:

- **Trigger Pin Format:** 10 uS digital pulse
- **Sound Frequency:** 40 kHz
- **Echo Pin Output:** 0-Vcc
- **Echo Pin Format:** output is DIGITAL and directly proportional with range. See our conversion formula above.
- **Measurement Range:** 2cm to ~4.5m
- **Measurement Resolution:** 0.3cm
- **Measurement Angle:** up to 15 deg
- **Measurement Rate:** 40 Hz
- **Supply Voltage:** 4.5V to 5.5V
- **Supply Current:** 10 to 40mA
- **Connector:** standard 5-pin male connector which can plug directly into breadboards.
- **Static current :** less than 2mA
- **Detection distance:** 2cm-450cm

## Annexe 4 : fiche technique du capteur US-100

### TECHNICAL DETAILS

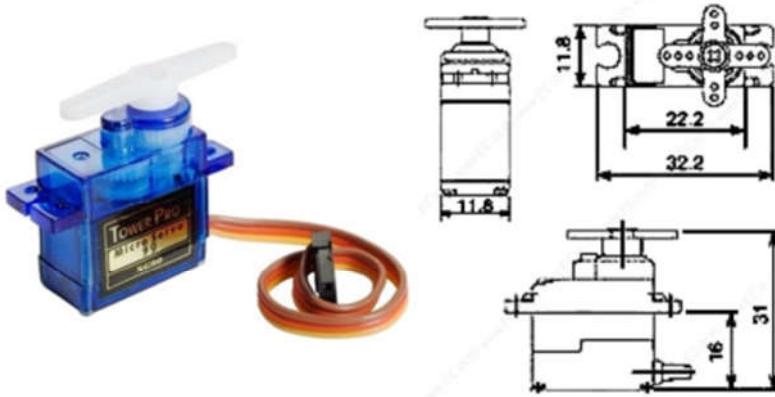
- Power & Logic Voltage: DC 2.4V~5.5V
- Current: 2mA
- Operating Temperature: -20~ + 70°
- Measuring Angle: Less than 15°
- Detection Distance: 2cm – 450 cm
- Accuracy: 0.3cm +1%
- Sensor dimensions (excluding header): 45 x 20mm
- Weight: 9g

## Annexe 5 : fiche technique du Rpi caméra V2

Product Name	Raspberry Pi Camera Module
<b>Product Description</b>	High Definition camera module compatible with all Raspberry Pi models. Provides high sensitivity, low crosstalk and low noise image capture in an ultra small and lightweight design. The camera module connects to the Raspberry Pi board via the CSI connector designed specifically for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the processor.
<b>RS Part Number</b>	<b>913-2064</b>
<b>Specifications</b>	
<b>Image Sensor</b>	Sony IMX 219 PQ CMOS image sensor in a fixed-focus module.
<b>Resolution</b>	8-megapixel
<b>Still picture resolution</b>	3280 x 2464
<b>Max image transfer rate</b>	1080p: 30fps (encode and decode) 720p: 60fps
<b>Connection to Raspberry Pi</b>	15-pin ribbon cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI-2).
<b>Image control functions</b>	Automatic exposure control Automatic white balance Automatic band filter Automatic 50/60 Hz luminance detection Automatic black level calibration
<b>Temp range</b>	Operating: -20° to 60° Stable image: -20° to 60°
<b>Lens size</b>	1/4"
<b>Dimensions</b>	23.86 x 25 x 9mm
<b>Weight</b>	3g

## Annexe 6 : fiche technique du servomoteur SG90

### SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

#### Specifications

- Weight: 9 g
  - Dimension: 22.2 x 11.8 x 31 mm approx.
  - Stall torque: 1.8 kgf·cm
  - Operating speed: 0.1 s/60 degree
  - Operating voltage: 4.8 V (~5V)
  - Dead band width: 10 µs
  - Temperature range: 0 °C – 55 °C
- Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is all the way to the left. ms pulse) is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

## Annexe 7 : fiche technique du buzzer

#### Specification:

Rated Voltage	(VDC)	12
Operating Voltage	(VDC)	3~16
*Rated Current	(mA)	≤12
*Sound Output at 10cm	(dB)	≥85
*Resonance Frequency	(KHz)	4.0±0.5
Tone		Continuous
Operating Temperature	(°C)	-40~+85
Storage Temperature	(°C)	-40~+90
Weight	(g)	0.96
Housing Material		PPO
RoHS		Yes

## Annexe 8 : fiche technique du driver moteur L298N

### Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low:  $-0.3V \leq V_{in} \leq 1.5V$ .
- High:  $2.3V \leq V_{in} \leq V_{ss}$ .
- Enable signal input voltage range :
  - Low:  $-0.3 \leq V_{in} \leq 1.5V$  (control signal is invalid).
  - High:  $2.3V \leq V_{in} \leq V_{ss}$  (control signal active).
- Maximum power consumption: 20W (when the temperature  $T = 75^{\circ}\text{C}$ ).
- Storage temperature:  $-25^{\circ}\text{C} \sim +130^{\circ}\text{C}$ .
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

## Annexe 9 : fiche technique du moteur jga25-370

齿轮电机型号 : JGA25-370-18250 功率4W 使用电压 DC12.0V 贵金属电刷特点噪音小 耗电量低											
减速比	4.4	9.28	21.3	34	45	75	100	165	217	362	478
减速器长度 L	15.0	17.0	19.0	21.0	21.0	23.0	23.0	25.0	25.0	27.0	27.0
空载转速 rpm	1930	915	400	250	190	110	85	50	39	23	18
空载电流 MA	$\leq 150$	$\leq 150$	$\leq 150$	$\leq 150$	$\leq 150$	$\leq 150$					
额定转矩 kg.cm	0.5	1.2	2.7	4.3	5.0	8.0	8.0	8.0	8.0	8.0	8.0
额定转速 rpm	105	500	230	140	100	65	55	40	32	20	17
额定电流 MA	$\leq 750$	$\leq 750$	$\leq 750$	$\leq 750$	$\leq 750$	$\leq 750$					
瞬间允许转矩 kg.cm	1.0	2.0	4.5	7.5	9.0	不允许超过9.0kg.cm					
霍尔反馈分辨率	48.4	102.1	234.3	374	495	825	1100	1815	2387	3982	5258

The drawing shows a side view of the motor with various dimensions labeled: total length L, width 12, height 25, bore diameter φ25, and a front view showing a stepped shaft with diameters φ7, φ4, and 3.5. To the right is a circular mounting hole pattern with a diameter of φ24.4, a central hole of φ14, and two tapped holes of φ3.0 labeled "1SDM3.0x2TAPPED HOLE 2 PLACES".

UNIT: MILLIMETRE(mm)