**Dédicace**

**J**e dédie ce travail …

**A mes très chers parents,**

*Je vous dois ce que je suis aujourd’hui grâce à votre amour, à votre patience et vos innombrables sacrifices. Que ce modeste travail, soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait d’incroyable pour moi.*

*Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler*.

**A ma chère sœur,**

En témoignage de l’attachement de l’amour et de l’affectation que je porte pour toi. Je te dédié ce travail avec tous mes vœux de bonheur, de santé et de réussite.

***A Mon cher frère,***

Je te souhaite de fond mon cœur une merveilleuse vie et la réussite.

Que dieu te protège et te donne l’excès de bonheur.

**A mes chers amis,**

Pour les bons moments passés ensemble, le respect partagé et l’amitié sincère

Vous êtes pour moi des sœurs et frères et des amis sur qui peux compter.

Je vous souhaite une vie plein de santé et de bonheur.

**À mes professeurs,**

**À toute ma famille,**

**À tous ceux que j’aime et qui m’aiment.**

Que Dieu vous bénisse, trouvez ici l’expression de mon profond respect, mon amour, et ma reconnaissance.

…*Soulaima*

**Dédicace**

**J**e dédie ce travail …

###### **À ma mère,**

*Qui m’a transmis la vie, l’amour, le courage, toutes mes joies, mon amour et ma plus vive reconnaissance n’égalera jamais l’ampleur de vos sacrifices*

###### **À mon père,**

*Qui a souhaité vivre pour longtemps juste pour nous voir qu’est-ce que nous allons devenir*

*Pour l’amour dont vous me couvrez. Pour tout ce que vous avez fait et que vous ferez pour ma réussite et mon bonheur*

###### **À mes frères,**

*Vous avez toujours été pour moi une aide très précieuse, Que je les aime tant. Je vous souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.*

Que Dieu m’aide à être adepte de ce que vous m’avez appris.

**A mes chers amis,**

En témoignage de l’amitié qui nous unit et des bons moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie plein de santé et de bonheur.

**À mes professeurs,**

**À toute ma famille,**

**À tous ceux que j’aime et qui m’aiment.**

*Aucune dédicace ne saurait exprimer la profondeur de mon amour et l’immense attachement que je vous voue.*

…*Amina*

**Remerciement**

Le plus dur n’est pas de rédiger le rapport mais de remercier toutes les personnes qu’ont contribué de près ou de loin, à l’aboutissement de ce travail et nous ont soutenu pour réaliser ce projet.

Nous tenons tout d’abord à remercier **TECHNO-SOFT**  de nous avoir proposé ce stage.

Nos remerciements s’adressent à notre encadrant de stage Mr. Mohsen Habbachi de nous avoir acceptée pour effectuer ce stage de projet de fin d’études, pour son temps précieux, ses précieux conseils et ses discussions fructueuses tout au long du déroulement de ce stage.

Nous sommes aussi redevable envers notre enseignant, Mr. Haddada Karim d’avoir accepté de nous encadrer et de nous soutenir tout au long de stage. Nous tenons à lui exprimer toute nos admirations et nos reconnaissances.

Nous profitons de cette occasion pour remercier les membres du jury pour l’honneur qu’ils nous font de bien vouloir juger ce travail. Veuillez trouver dans ce rapport les qualités de clarté et de motivation qu’ils attendent.

Nous souhaitons remercier toutes les personnes qui nous ont accordé leurs conseils en or, et qui ont contribué à la rédaction de ce rapport.

Finalement, nous adresse mes profondes gratitudes à nos familles qui ont été toujours à nos côtés, pour ses soutiens et ses encouragements.

Table des matières

Introduction générale1

**Chapitre I : Cadre générale du projet**2

1. Introduction3
2. Présentation de la société3
3. Présentation générale3
4. Orientations stratégiques3
5. Produits et services4

4. But5

1. Problématique6
2. Solution6
3. La robotique et la sécurité7
   1. Définition de la robotique7
   2. Champs d'action7
   3. Les avantages7
   4. Les limites8
4. Conclusion8

Chapitre II : [Etude technologique du système](#_bookmark13) 9

1. [Introduction](#_bookmark14)10
2. [Description globale du système](#_bookmark15)10
3. Choix de l’unité de commande10
4. Raspberry Pi 3 (modèle B) 11
5. Spécifications techniques du microcontrôleur Raspberry Pi 312
6. Les besoins élémentaires13
7. Système d’exploitation14
8. Etude et choix du matériel15
9. Capteur ultrason (HC-SR04) 15
10. Capteur de mouvement (PIR) 16
11. Moteurs (DC) 18
12. Camera USB20
13. Environnement logiciel et langage de programmation21
14. Python21
15. Fritzing21
16. PuTTy22
17. Conclusion22

**Chapitre III : Test et configuration du système robotique**23

1. Introduction24
2. Architecture générale du robot de surveillance24
3. Diagramme de cas d’utilisation24
4. Schéma descriptif de système robotique25
5. Système de collection de données et de contrôle26
6. Configuration de la carte Raspberry 27

### Installation de système d'exploitation27

1. Connexion avec la carte du Raspberry Pi aux périphériques 28
2. Choix de l’adresse IP du Raspberry Pi28
3. Contrôle à distance via SSH29
4. Test de la plateforme robotique32
5. Capteur d’obstacle32
6. Capteur de mouvement32
7. Moteur électrique32
8. Caméra34
9. Conclusion 34

**Chapitre IV : Implémentation et réalisation d’une application web**35

1. Introduction36
2. Conception de l’interface de commande 36
3. Objectif36
4. Principe de fonctionnement de l'application36
5. Développement de l’application web 37
6. Configuration de serveur web 37
7. Langages et outils utilisées39
8. Description de l'interface de l'application45
9. Page d’authentification45
10. Page de surveillance46
11. Conclusion49

**Conclusion générale et perspectives**50

**Bibliographie**51

**Annexe**52

**Liste des Figures**

[**Figure 1 :** **Logo du TECHNO-SOFT SOLUTION** 6](file:///H:\Liste%20des%20Figures.docx#_Toc452062942)

[**Figure 2 :** Description de la carte Raspberry 12](file:///H:\Liste%20des%20Figures.docx#_Toc452062943)

[**Figure 3 :** Carte mémoire SD 13](file:///H:\Liste%20des%20Figures.docx#_Toc452062944)

[**Figure 4 :** Câble HDMI 13](file:///H:\Liste%20des%20Figures.docx#_Toc452062945)

[**Figure 5 :** Câble RJ45 14](file:///H:\Liste%20des%20Figures.docx#_Toc452062946)

[**Figure 6 :** Alimentation micro USB externe 14](file:///H:\Liste%20des%20Figures.docx#_Toc452062947)

[**Figure 7 :** Logo du système d’exploitation Raspbian 15](file:///H:\Liste%20des%20Figures.docx#_Toc452062948)

[**Figure 8 :** Capteur ultrason 15](file:///H:\Liste%20des%20Figures.docx#_Toc452062949)

[**Figure 9 :** Fonctionnement du capteur ultrason 16](file:///H:\Liste%20des%20Figures.docx#_Toc452062950)

[**Figure 10 :** Branchement du capteur ultrason 16](file:///H:\Liste%20des%20Figures.docx#_Toc452062951)

[**Figure11 :** Capteur de mouvement 17](file:///H:\Liste%20des%20Figures.docx#_Toc452062952)

[**Figure 12 :** Fonctionnement du capteur de mouvement 17](file:///H:\Liste%20des%20Figures.docx#_Toc452062953)

[**Figure 13 :** Branchement du capteur de mouvement 18](file:///H:\Liste%20des%20Figures.docx#_Toc452062954)

[**Figure 14 :** Moteur DC 18](file:///H:\Liste%20des%20Figures.docx#_Toc452062955)

[**Figure 15 :** Pont H 19](file:///H:\Liste%20des%20Figures.docx#_Toc452062956)

[**Figure 16 :** Circuit intégré L293D 19](file:///H:\Liste%20des%20Figures.docx#_Toc452062957)

[**Figure 17 :** Branchement des moteurs DC 20](file:///H:\Liste%20des%20Figures.docx#_Toc452062958)

[**Figure 18 :** Camera USB 20](file:///H:\Liste%20des%20Figures.docx#_Toc452062959)

[**Figure 19 :** Logo du Python 21](file:///H:\Liste%20des%20Figures.docx#_Toc452062960)

[**Figure 20 :** Logo du fritzing 22](file:///H:\Liste%20des%20Figures.docx#_Toc452062961)

[**Figure 21 :** Logo du PuTTy 22](file:///H:\Liste%20des%20Figures.docx#_Toc452062962)

[**Figure 22:** Diagramme de cas d’utilisation du robot 24](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

[**Figure 23 :** Fonctionnalité du robot 25](file:///H:\Liste%20des%20Figures.docx#_Toc452062963)

[**Figure 24 :** Système de collection des données et de contrôle 26](file:///H:\Liste%20des%20Figures.docx#_Toc452062964)

[**Figure 25 :** Interface de configuration au premier démarrage 29](file:///H:\Liste%20des%20Figures.docx#_Toc452062966)

[**Figure 26 :** Interface de configuration au deuxième démarrage 30](file:///H:\Liste%20des%20Figures.docx#_Toc452062967)

[**Figure 27 :** Interface de configuration au troisième démarrage 30](file:///H:\Liste%20des%20Figures.docx#_Toc452062968)

[**Figure 28 :** Interface de configuration au quaterième démarrage 31](file:///H:\Liste%20des%20Figures.docx#_Toc452062969)

[**Figure 29 :** Logiciel PuTTy 31](file:///H:\Liste%20des%20Figures.docx#_Toc452062970)

[**Figure 30 :** Resultat de test du capteur ultrason 32](file:///H:\Liste%20des%20Figures.docx#_Toc452062972)

[**Figure 31 :** Resultat de test du capteur de mouvement 33](file:///H:\Liste%20des%20Figures.docx#_Toc452062973)

[**Figure 32 :** Resultat de test du moteur électrique 33](file:///H:\Liste%20des%20Figures.docx#_Toc452062974)

[**Figure 33 :** Resultat de test de la camera par Motion 34](file:///H:\Liste%20des%20Figures.docx#_Toc452062975)

[**Figure 34 :** Les differentes actions à bénéficier de l’application web 36](file:///H:\Liste%20des%20Figures.docx#_Toc452062976)

[**Figure 35 :** Architecture client/serveur 37](file:///H:\Liste%20des%20Figures.docx#_Toc452062977)

[**Figure 36 :** Logo du HTML 40](file:///H:\Liste%20des%20Figures.docx#_Toc452062979)

[**Figure 37 :** Logo du CSS 40](file:///H:\Liste%20des%20Figures.docx#_Toc452062980)

[**Figure 38 :** Logo du JavaScript 41](file:///H:\Liste%20des%20Figures.docx#_Toc452062981)

[**Figure 39 :** Logo du jQuerry 41](file:///H:\Liste%20des%20Figures.docx#_Toc452062982)

[**Figure 40 :** Logo d’AJAX 42](file:///H:\Liste%20des%20Figures.docx#_Toc452062983)

[**Figure 41 :** Logo du PHP 43](file:///H:\Liste%20des%20Figures.docx#_Toc452062984)

[**Figure 42 :** Logo du MySQL 43](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

[**Figure 43 :** Logo du Visual Studio Code 44](file:///H:\Liste%20des%20Figures.docx#_Toc452062985)

[**Figure 44 :** Logo d’Apache Cordova 44](file:///H:\Liste%20des%20Figures.docx#_Toc452062986)

[**Figure 45 :** Logo d’Android Studio 45](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

[**Figure 46 :** Interface d’authentification. 46](file:///H:\Liste%20des%20Figures.docx#_Toc452062987)

[**Figure 47 :** Interface page d’accueil 47](file:///H:\Liste%20des%20Figures.docx#_Toc452062988)

[**Figure 48 :** Interface menu principal 47](file:///H:\Liste%20des%20Figures.docx#_Toc452062989)

[**Figure 49 :** Interface ON/OFF 48](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

[**Figure 50 :** Interface Recording 48](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

[**Figure 51 :** Interface Alert 49](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

Liste des tableaux

[**Tableau 1 :** Etude comparative entre Arduino Uno et Raspberry Pi 3 11](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

[**Tableau 2 :** Description de la carte Raspberry Pi 12](file:///H:\Liste%20des%20Figures.docx#_Toc452062990)

Liste des abréviations

**HDCVI** **:** High Definition Composite Video Interface

**IP :** Internet Protocol

**GPS :** Global Positioning System

**GPRS :** General Packet Radio Service

**USB :** Universal Serial Bus

**SD :** Secure Digital

**RJ** **:** Registered Jack

**RAM** **:**   Random Access Memory

**FPGA :** Field Programmable Gate Array

**ARM :** Advanced RISC Machines

**SoC  :** System on Chip

**SDRAM  :**  Synchronous Dynamic Random Access Memory

**GPIO :** General Purpose Input/Output

**LPDDR :** Low Power Double Data Rate

**HDMI :** High Definition Multimedia Interface

**RCA** **:** [Radio Corporation of America](https://fr.wikipedia.org/wiki/Radio_Corporation_of_America).

**CSI :** Camera Serial Interface

**FTP :** File Transfer Protocol

**PIR :** Passive InfraRed sensor

**DC** **:** Direct Current

**PHP :** Hypertext Preprocessor

**MySQL :**  My Structured Query Language

**SSH :**  Secure SHell

**DHCP :** Dynamic Host Configuration Protocol

**HTTP :**  Hypertext Transfer Protocol

**HTML** **:**  Hyper Text Markup Language)

**CSS** **:** Cascading Style Sheets

**AJAX :** Asynchronous JavaScript And XML

**XML** **:** Extensible Markup Language

**SQL :** Structured Query Language

**API :** Application Programming Interface

Introduction générale

Aujourd'hui la robotique prend une place importante dans notre vie quotidienne. Elle est au service de l'homme dans de nombreux domaines y compris dans celui de la surveillance. En effet, la mobilité des robots et leurs capteurs les rendent particulièrement aptes à jouer les agents de sécurité. Par ailleurs, le besoin croissant de la surveillance automatisée d'environnements intérieurs, comme des aéroports, des entrepôts, des maisons et des usines de production, stimule le développement de système intelligent basé sur des capteurs mobiles. Différemment de dispositif de surveillance non-mobile traditionnel.

Dans ce contexte, vient la réalisation de ce projet, qui vise à concevoir et développer un système de surveillance basé sur une carte Raspberry pour assurer les différents contrôles. Ainsi, le propriétaire pourra commander son robot à distance via une application web. Ce travail essaie de fournir beaucoup plus de soutien dans notre vie quotidienne tout en répondant aux attentes des professionnels de la sécurité : réduire les coûts et les risques liés aux aléas de la sécurité.

Notre rapport est scindé en quatre parties :

* Le premier chapitre représente le cadre général du projet.
* Dans le deuxième chapitre, on étudiera les différents outils matériels et logiciels dont on a besoin lors de ce projet.
* La réalisation de notre plateforme robotique sera détaillée dans le troisième chapitre.
* Pour assurer un contrôle distant de notre robot, nous développerons une application mobile qu’on décrira dans le quatrième chapitre.

***Chapitre I : Cadre générale du projet***

1. **Introduction**

L’objectif de ce chapitre introductif est de mettre ce projet dans son contexte général.

On commence tout d’abord par la présentation de l’organisme d’accueil de notre stage. Ensuite, nous citons la problématique qui a abouti à notre projet et la solution proposée. Enfin, nous présentons une idée générale sur la robotique et la sécurité.

1. **Présentation de la société**
2. **Présentation générale**

TECHNO-SOFT est une entreprise tunisienne, spécialisée dans l'installation, la distribution et l’intégration des solutions de sécurité électronique (vidéo surveillance, de l'alarme anti-intrusion ou incendie, gestion des ressource humaine, contrôle d’accès, vidéophonie, géo localisation, les détecteurs des métaux, et les solutions de la domotique).

Elle maîtrise  l’ensemble des  technologies de l’électronique, de l’informatique et des télécommunications et elle permet de protéger, sécurisé, automatiser et améliorer les tâches au sein d’une maison, d’une résidence, d’un bâtiment, d’un espace commercial, industriel ou  public.

Fondée depuis 2011 par une équipe professionnelle des ingénieurs et des techniciens 100% Tunisiens, développé à travers des années, des technologies de référence ainsi que des équipements et des services de qualité.

TECHNO-SOFT arrivée à conquérir le marché tunisien en matière de sécurité en proposant des systèmes de sécurité destinés aussi bien aux particuliers qu'aux professionnels. Aujourd'hui, la société dispose d'une gamme aussi riche que variée de systèmes de sécurité ainsi que d'une équipe de professionnels des ingénieurs et des techniciens qualifiés prêts à répondre aux exigences de sa clientèle sur la totalité du territoire tunisien.

1. **Orientations stratégiques**

A son lancement, faisant du mieux servir le client sa priorité fondamentale, TECHNO-SOFT a aspiré de positionner la société sur trois piliers essentiels :

* la sécurité à toute épreuve
* la variété de gamme
* la satisfaction client

La société attache une grande importance à être force d’innovations dans ses offres de sécurité depuis notre expérience dans le domaine. C'est un virage stratégique qui s’opère puisque nous souhaitons apporter à nos clients une solution de sécurité à la fois fiable, variée, satisfaisante mais aussi adaptée.

1. **Produits et services**

Aujourd'hui la société dispose d'une gamme aussi riche que variée de la technologie audiovisuelle en élargissant sa gamme de produits et sa gamme de clientèle tout en se développant à l’échelle mondiale.

La société est spécialisée dans les infrastructures services. Elle bénéficie d'une expérience exigée au produit de sécurité, qui nous permet de savoir la meilleure qualité trouvée dans le marché et propose les services suivants :

* Camera de surveillance « IP, Analogique, HDCVI, etc. »

Un système de surveillance  a pour but de contribuer à la sécurité des biens et des  personnes, cette contribution peut se localiser dans une maison, une résidence, un  bâtiment, un espace commercial, industriel ou  public.

* Système d'alarme « Filaire et sans Fil »

Installer une alarme chez soi est certainement la solution la plus efficace pour limiter les risques d’être cambriolé. Mais  il est important d'investir dans un système d'alarme de qualité, adapté à votre local et à l'utilisation que vous souhaitez en avoir  TECHNO-SOFT, spécialiste de l'alarme sans fil et filaire nous propose une large gamme de systèmes d'alarme pour vivre en toute sécurité.

* Gestion de temps et de présence « Pointeuses, logiciel de pointage, etc. »

Etant conscient de l’importance du pointage au sein d’une entreprise nous offrons des pointeuses qui permettent de palier à tous les problèmes pouvant affecter ce système. Gérer le pointage c'est gérer les réactions, comportements et demandes des salariés et/ou de leurs représentants ainsi que des agents de contrôle face au système de pointage pour cela on  vous propose aussi un logiciel de gestion de temps de présence complet et facile à utiliser pour traduire les pointages en données de présence (retard, départ anticipé, heures sup, etc.).

* Contrôle d'accès

Le contrôle d'accès électronique fournit le moyen le plus efficace pour sécuriser vos locaux et vos employés en sécurisant les accès le plus sensibles. C’est la fonction permettant de limiter l'accès à un lieu, et en facilitant l'appréhension des auteurs d'actes de malveillance puisque toutes les entrées et sorties sont enregistrées et facilement retracées par le logiciel fourni.

* Contrôle des rondes

Parce que la gestion de la sécurité est devenue primordiale dans toutes les entreprises, TECHNO-SOFT vous propose une solution informatisée simple d’utilisation et efficace, l’objectif est de simplifier l’organisation et le contrôle les rondes des gardiens. Le principe consiste à mémoriser, en temps et en heure, le passage d'un agent de sécurité en des points précis, de façon à avoir une traçabilité des rondes effectuées. L'agent est muni à cet effet d'un contrôleur de ronde avec lequel il doit "badger" chaque point de contrôle. Ces informations peuvent alors être exploitées à partir du logiciel fourni en quelques clics.

* Solution de Géo localisation en temps réel  « GPS »

La solution de gestion de flotte par GPS/GPRS a été développée pour répondre de manière simple et économique à l’ensemble de vos besoins. Elle permet un suivi et une maîtrise totale de vos véhicules, afin d'optimiser leur utilisation et leur productivité.

* Détection d'incendie

La détection d’incendie regroupe l'ensemble des techniques et procédés qui concourent à la détection précoce d'un feu.

1. **But**

Le but de la société TECHNO-SOFT est :

* Garantir une qualité et fournir un service adapté aux besoins des clients.
* Etablir et maintenir un rapport de confiance avec les clients.
* Elaborer des solutions sur mesure et évolutives.

La figure 1 présente une image sur le logo de la société TECHNO-SOFT.



**Figure 1 :** Logo du TECHNO-SOFT SOLUTION

1. **Problématique**

La surveillance vidéo et le contrôle des bâtiments et des entreprises présentent un axe de recherche en développement continu vue au nombre croissant de menaces tel que le vol, l’intrusion, la violation de notre intimité, etc.

Dans ce projet, l’axe principal est la mise en œuvre d’une solution qui permet de surveiller et de contrôler des lieux précis en utilisant les nouvelles technologies dans la réalisation. Afin de remédier à ces problèmes, on propose dans le cadre de notre travail de concevoir et développer une solution intelligente qui répond aux exigences des utilisateurs.

1. **Solution**

Après avoir spécifié les différents besoins et pour fournir plus de sécurité, nous avons décidé de réaliser un robot mobile de surveillance piloté à distance par connexion Wifi. En effet, le propriétaire sera capable de garder un œil sur son bâtiment même s’il est loin et vérifier que tout se passe bien. Grâce à une application web, il aura la possibilité de contrôler son robot sans l’avoir dans son champ de vision.

Les principales fonctions seront les suivantes :

* Télésurveillance.
* Pilotage à distance du robot via une connexion Wifi.
* Visualisation du flux vidéo de la webcam en temps réel
* Détection d’intrus.
* Envoi d’alerte en cas d’intrusion.
* Mode autonome grâce à une fonction de détection d’obstacles et une autre à base de circuit préenregistré.
* Enregistrement des vidéos dans le mode autonome.

1. **La robotique et la sécurité**
2. **Définition de la robotique**

La robotique peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles intellectuelles.

1. **Champs d'action**

Nous allons vous présenter les champs d'action où intervient la robotique dans la surveillance de zones et de périmètres délimités.

* La robotique intervient dans la sécurité des états et du public

Dans le déminage, la robotique militaire contrôle les frontières et surveille les sites sensibles (centrales nucléaires, aéroports, gares et chemins de fer).

* La robotique intervient aussi dans le gardiennage domestique

La robotique intervient également dans le gardiennage professionnel dans les zones industrielles et les surveillances d'entrepôts (sécurité d'incendie et détection d'intrusion).

* La robotique intervient également dans la sécurité environnementale

Elle prévient aux catastrophes naturelles et veille contre la pollution, en utilisant les drones qui analyse les surfaces terrestres et les composants atmosphériques. Elle utilise les robots marins et sous-marins qui détectent la pollution et étudie la faune.

1. **Les avantages**

Les robots sont pour nous synonymes d'aide, d'assistance, d'efficacité, de progrès, d'augmentation des connaissances et de diminution des situations a risque sur l'homme.

1. **Les limites**

La robotique présente un problème éthique et juridique : remplacement des humains par des robots. En outre, les robots ne sont pas toujours autonomes et fiables (la sécurité exige une grande fiabilité). Ils sont aussi une source d'accident.

1. **Conclusion**

Ce chapitre nous a permis de mettre le projet dans son cadre général en déterminant la problématique et en proposant par la suite une solution envisagée pour faire face à la situation courante. Dans le chapitre suivant, nous entamons l’étude technologique du système.

***Chapitre II :***

***Etude technologique du système***

1. **Introduction**

Dans ce chapitre nous allons étudier les outils nécessaires dans ce projet, leurs caractéristiques, ainsi que leur contribution dans l’organisation globale du système. Cette étude contient les spécifications techniques, le fonctionnement et le branchement des composants du système développé. Ensuite, nous présentons l’environnement logiciel et les langages de programmation utilisés.

1. **Description globale du système**

Rovio est un robot de surveillance commandé à distance par une application web.

Celui-ci peut être contrôlé par l’utilisateur à distance à travers les flèches directionnelles de l’application web dans un lieu donné. Il peut être également piloté grâce à une caméra embarquée au bord qui transmet des images en temps réel au poste de commande. Ainsi, l’utilisateur peut prendre les décisions nécessaires en se basant sur les images reçues. De plus, le robot est équipé d’un capteur qui permet d’alerter l’utilisateur d’un mouvement suspect produit lorsque le robot est au moment où l’utilisateur n’est pas devant son centre de contrôle. Nous avons travaillé sur un mode autonome. Ce mode procure au robot une intelligence qui lui permet de détecter l’obstacle et l’éviter. Dans ce mode, le robot circule tout seul en enregistrant le flux vidéo dans un répertoire.

1. **Choix de l’unité de commande**

L’unité de commande est la carte électronique qui commande et contrôle le fonctionnement du système. Dans ce sens, le Raspberry Pi et l’Arduino sont deux plateformes qui peuvent être utilisé dans notre projet.

Pour pouvoir s’adapter au meilleur choix, nous avons comparé leurs caractéristiques dans le tableau 1.

|  |  |  |
| --- | --- | --- |
|  | Arduino Uno | Raspberry Pi 3 |
| Prix | env. 23 € | env. 35 € |
| Taille | 7.6 x 1.9 x 6.4 cm | 8.5 x 5.6 x 1.7 cm |
| Mémoire vive | 0.002 MB | 1 Go |
| Vitesse d’horloge | 16 MHz | 1,2 GHz |
| Réseau embarqué | Aucun | 10/100 Ethernet filaire RJ45 |
| Capacité multitâche | Non | oui |
| Tension d’entrée | 7 à 12 V | 5 V |
| Mémoire flash | 32 KB | Carte SD (2 à 16 Go) |
| USB | 1 port | 4x port USB 2.0 |
| Système d’exploitation | Non | Distributions Lunix |

**Tableau 1 :** Etude comparative entre Arduino Uno et Raspberry Pi 3

Pour assurer le bon fonctionnement de notre robot, nous avons préféré d’utiliser le Raspberry

Pi 3 puisque qu’il est 75 fois plus rapide qu’un Arduino et possède 67.000 fois plus de RAM.

D’autre part, la carte Raspberry peut effectuer plusieurs tâches en même temps et peut se connecter à Internet en Wifi ou en filaire. De plus, elle est assez puissante, ce qui lui permet d’être utilisée pour des usages bureautiques ou de navigation web.

En bref, la carte Raspberry Pi 3 est la plus performante pour la réalisation de notre projet.

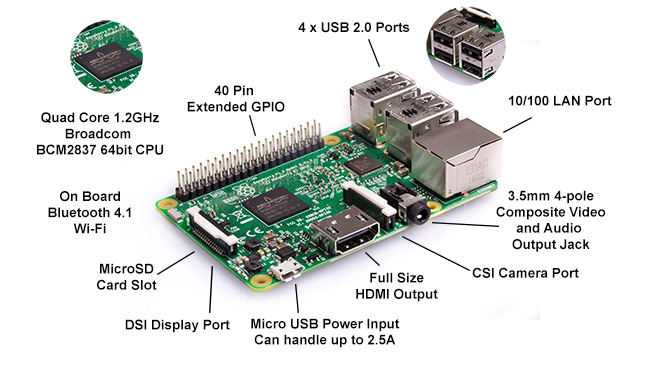
1. **Raspberry Pi 3 (modèle B)**

Le Raspberry Pi est un nano ordinateur qui s’exécute sous le système d’exploitation. Son cœur est un FPGA (Broadcom 2835) intégrant un processeur ARM11 cadencé à 700MHz et de nombreux périphériques.

Il a été créé par l'anglais David Braben dans le cadre de sa [fondation Raspberry Pi](https://fr.wikipedia.org/wiki/Fondation_Raspberry_Pi).

Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique. Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles.

La figure 2 illustre une vue de face de la carte Raspberry Pi 3.



**Figure 2 :** Description de la carte Raspberry

1. **Spécifications techniques du microcontrôleur Raspberry Pi 3**

Le modèle Rapsberry Pi 3 type B est composé d’éléments fiables dont les spécifications sont représentées dans le tableau 2 :

|  |  |
| --- | --- |
| Cadencement | 1,2 GHz |
| Puce (SoC) | Broadcom BCM2837 |
| Processeur | ARM Cortex-A53 64 bits quatre cœurs |
| Processeur graphique | Broadcom VideoCore IV double coeur (OpenGL ES 2.0, H.264 Full HD à 30 ips) |
| Mémoire (SDRAM) | 1GB LPDDR2 |
| Nombre de ports USB 2.0 | 4 |
| Port extension | GPIO 40 pins |
| Sorties vidéo | HDMI et RCA, plus 1 connecteur de caméra CSI |
| Sorties audio | Stéréo Jack 3,5mm ou HDMI |
| Sauvegarde des données | Carte MicroSD |
| Connexion réseau | * 10/100 Ethernet, WiFi 802.11n et Bluetooth 4.1 (BLE - Low Energy) |
| Périphériques | 17 × GPIO |
| Alimentation | 5v 2.5A via micro-USB |
| Dimensions | 85,60 mm × 53,98 mm × 17 mm |

**Tableau 2 :** Caractéristiques de la carte Raspberry Pi 3

1. **Les besoins élémentaires**

Pour faire fonctionner notre Raspberry Pi, nous avons besoin les accessoires suivants :

* Une carte mémoire SD

Le Raspberry Pi ne dispose pas de disque dur. En effet, celui-ci est remplacé par une carte MicroSD qui contient le système d’exploitation et les données. Concernant sa capacité, elle devra être au moins de 16 Go.

La figure 3 illustre une vue de faces de la carte mémoire SD.



**Figure 3 :** Carte mémoire SD

* Un câble HDMI

Le câble HDMI est utilisé pour piloter à distance un Raspberry Pi via SSH, l’installation d’une distribution Linux, sa configuration et son utilisation sur un écran.

La figure 4 illustre une vue de face du câble HDMI.



**Figure 4 :** Câble HDMI

* Un câble Ethernet RJ45

Le modèle B du Raspberry Pi est muni d’un port RJ45 afin qu’il puisse être relié sur un réseau. Il faut donc prévoir un câble de ce type pour pouvoir utiliser ce port.

La figure 5 illustre une vue de face sur le câble RJ45.



**Figure 5 :** Câble RJ45

* Alimentation micro USB externe

La Raspberry Pi 3 a besoin d’une nouvelle alimentation, plus puissante, fournissant une tension de 5 V et une intensité d’au moins 700 mA.

La figure 6 illustre une image d’alimentation micro USB externe.

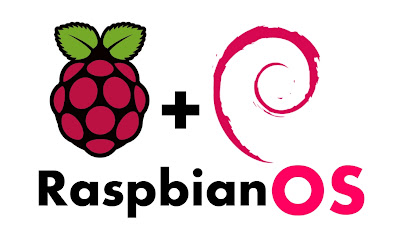


**Figure 6 :** Alimentation micro USB externe

1. **Système d’exploitation**

La distribution de système d’exploitation choisie c’est «Raspbian». C’est un système d’exploitation Linux dérivé de Debian mais allégé et adapté pour l’architecture du mini-ordinateur. Cette distribution est actuellement la plus populaire et la plus utilisée chez les possesseurs de Raspberry Pi. En effet elle est très polyvalente et parfaitement capable de faire office de serveur web ou FTP, de naviguer sur internet, de développer en PHP, MySQL, Python etc.

La figure 7 présente une image sur le logo Raspbian.



**Figure 7 :** Logo du système d’exploitation Raspbian

1. **Etude et choix du matériel**

Dans cette partie on va s’intéresser à la présentation des matériels utilisés pour la réalisation de notre projet en décrivant leur fonctionnement et leur branchement.

1. **Capteur ultrason (HC-SR04)**

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres, bien que des matériaux comme les vêtements puissent être difficiles à détecter.

La figure 8 illustre une image du capteur ultrason.

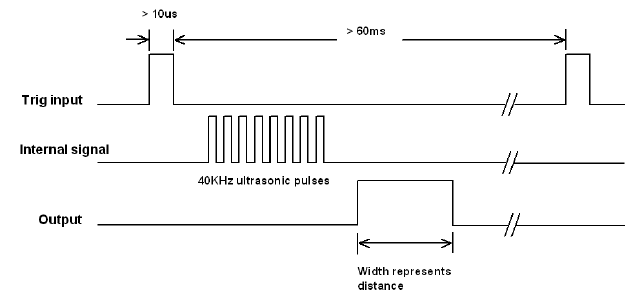


**Figure 8 :** Capteur ultrason

1. **Fonctionnement**

Pour déclencher une mesure, il faut présenter une impulsion "high" (5 V) d'au moins 10 µs sur l'entrée "Trig". Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée.

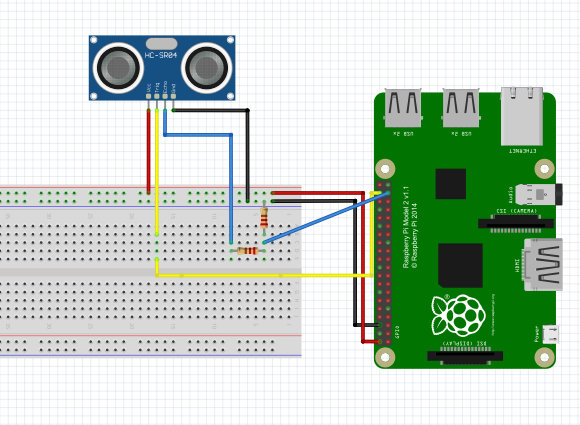
La figure 9 présente le principe de fonctionnement du capteur ultrason.



**Figure 9 :** Fonctionnement du capteur ultrason

1. **Branchement**

Le branchement du capteur ultrason se fait comme le montre la figure 10.

****

**Figure 10 :** Branchement du capteur ultrason

1. **Capteur de mouvement (PIR)**

Le capteur de mouvement passif à infrarouge PIR est un dispositif qui détecte les objets en mouvement, en particulier les personnes dans son champ de vision en se basant sur l’infrarouge.

La figure 11 illustre une image du capteur de mouvement.



**Figure 11 :** Capteur de mouvement

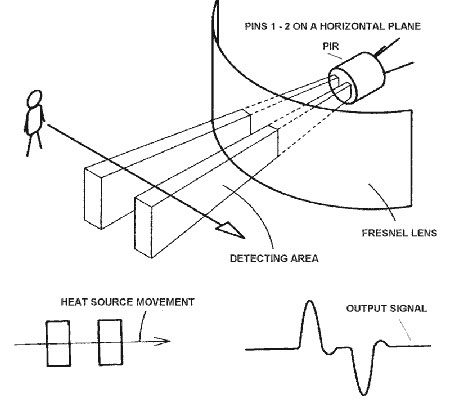
1. **Fonctionnement**

Le capteur PIR est un capteur Numérique :

Si un mouvement est détecté, le signal passe au niveau HAUT (3,3V) et un délai est déclenché.

Si aucun mouvement n’est pas détecté, le signal est mis au niveau BAS (0).

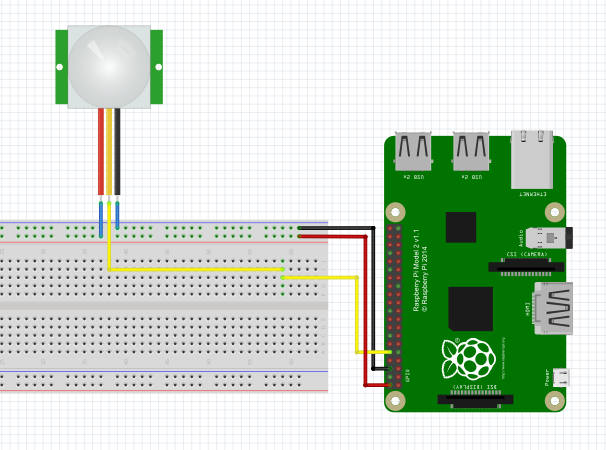
La figure 12 présente le principe de fonctionnement du capteur de mouvement.



**Figure 12 :** Fonctionnement du capteur de mouvement

1. **Branchement**

Le branchement du capteur de mouvement se fait comme le montre la figure 13 :



**Figure 13 :** Branchement du capteur de mouvement

1. **Moteurs (DC)**

Notre système nécessite 3 roues :

* Deux roues motrices (2 moteurs DC) pour le mouvement.
* Une roue fixe pour l'équilibre du robot.

Pour avoir un mouvement de rotation, on applique une tension continue aux bornes du moteur en fonction de la différence de potentiel. Le moteur va donc tourner dans un sens ou dans l’autre.

La figure 14 illustre une image du moteur DC.



**Figure 14 :** Moteur DC

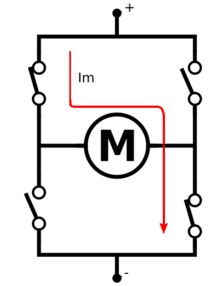
1. **Fonctionnement**

Pour faire avancer ou reculer notre robot, il faudra inverser le sens du courant, mais nous ne pourrons pas démonter le robot et inverser le branchement bornes pendant qu’il roule !

Le pont en H permet justement de contrôler un moteur sans changer le circuit.

A l’aide des interrupteurs, on peut faire passer le courant dans le moteur d’un sens ou dans l’autre en basculant deux interrupteurs.

La figure 15 présente le schéma du pont H.



**Figure 15 :** Pont H

* **Le circuit intégré L293D**

Nous avons vu que le pont H permet de contrôler un moteur avec des interrupteurs. Mais nous n’allons pas courir derrière notre robot pour le contrôler en appuyant sur des interrupteurs !

Le circuit intégré L293D contient deux ponts H, et permet de contrôler jusqu’à deux moteurs bidirectionnels indépendamment sans se préoccuper de basculer les interrupteurs.

Ce circuit intégré permet également d’alimenter les moteurs par une autre source d’alimentation que la carte Raspberry Pi.

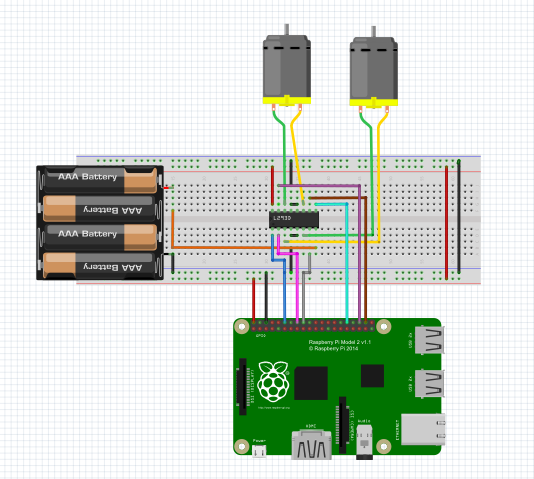
La figure 16 illustre une image du circuit intégré L293D.



**Figure 16 :** Circuit intégré L293D

1. **Branchement**

Le branchement de deux moteur DC se fait comme le montre la figure 17.



**Figure 17** **:** Branchement des moteurs DC

1. **Camera USB**

Dans notre projet on va utiliser une webcam pour sécuriser nos locaux et surveiller le mouvement, ainsi que d’enregistrer des vidéos à l’aide d’une application web.

En effet, la webcam est une caméra vidéo qui alimente ou diffuse son image en temps réel vers un réseau informatique (application).

Lorsque le flux vidéo est capturé, il peut être sauvegardé, visualisé ou envoyé à d'autres réseaux via des systèmes tels que l'internet et envoyé par e-mail en pièce jointe.

Pour utiliser cette caméra, il suffit de relier la webcam au port USB du Raspberry et activer son fonctionnement dans la partie de commande d’interface de matérielle du Raspberry.

La figure 18 illustre une image de la camera USB.



**Figure 18 :** Camera USB

1. **Environnement logiciel et langage de programmation**

Dans cette partie, nous présentons les logiciels et langages de programmation utilisés dans ce projet.

1. **Python**

Python est un langage de programmation inventé par Guido van Rossum. Sa première version est sortie en 1991. Il est l’un des langages de programmation les plus intéressants du moment.

Python est un langage facile à apprendre et leur code est plus lisible, il est donc plus facile à maintenir. Il permet d’écrire des scripts très simples et grâce à ses nombreuses bibliothèques, on peut travailler sur des projets plus ambitieux.

Python est un langage de programmation officielle qui a été pris en charge comme langue d’apprentissage éducatif sur le Raspberry Pi. Il est d’ailleurs installé par défaut dans Raspbian. Cependant, n’importe quel langage de programmation qui compilera sur un processeur ARMv6 est utilisable comme un moyen de développement sur le Raspberry Pi.

La figure 19 présente une image sur le logo du python.



Figure 19 : Logo du Python

1. **Fritzing**

Le logiciel conçu par cette association, également nommé Fritzing, est un [logiciel](https://fr.wikipedia.org/wiki/Logiciel) d'édition de circuit imprimé. Il est adapté aux débutants ou confirmés en électronique pour faire rapidement des circuits simple, Le logiciel comporte trois vues principales :

* La « Platine d'essai », où l'on voit les composants tels qu'ils sont dans la réalité et ou l'on construit le montage.
* La « Vue schématique », représentant le schéma fonctionnel du circuit.
* Le « Circuit imprimé », représentant la vue du circuit imprimé.

La figure 20 présente une image sur le logo du Fritzing.



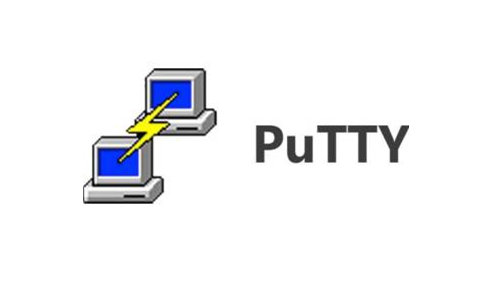
**Figure 20 :** Log du Fritzing

1. **PuTTy**

**PuTTY**est Un émulateur de terminal permettant de se connecter à distance à des serveurs en utilisant les protocoles SSH, Telnet ou Rlogin. L'ensemble des sessions peuvent être automatiquement enregistrées dans un rapport qui pourra être consulté ultérieurement.

À l'origine disponible uniquement pour [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows), il est à présent porté sur diverses plates-formes [Unix](https://fr.wikipedia.org/wiki/Unix) (et non-officiellement sur d'autres plates-formes). PuTTY est écrit et maintenu principalement par [Simon Tatham](https://fr.wikipedia.org/wiki/Simon_Tatham).

La figure 21 présente une image sur le logo du PuTTy.



**Figure 21 :** Logo du PuTTy

1. **Conclusion**

Tout au long de ce chapitre, nous avons étudié les outils du système à réaliser. En effet, nous avons identifié les différentes fonctionnalités des principaux composants employés ainsi que les logiciels et langages de programmation utilisés dans notre projet.

Cette étude adoptée facilite par la suite la validation de notre plateforme robotique, qui sera par la suite le contenu de notre troisième chapitre.

***Chapitre III :***

***Test et configuration du système robotique***

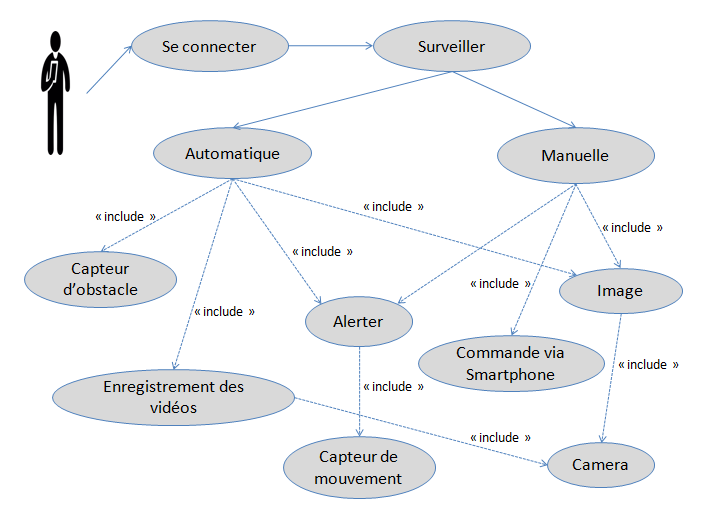
1. **Introduction**

Dans ce chapitre nous intéressons à donner une description sur la validation de notre plateforme robotique.

Nous allons présenter dans cette phase en premier temps, l’architecture générale du robot de surveillance. Ensuite nous allons citer la configuration de la carte de commande. Enfin, nous allons tester le fonctionnement des différents composants utilisés.

1. **Architecture générale du robot de surveillance**
2. **Diagramme de cas d’utilisation**

Le diagramme de cas d’utilisation est un diagramme utilisé pour donner une vision globale du comportement fonctionnel de notre système. Il décrit l'interaction entre les acteurs qui sont les utilisateurs et les cas d’utilisations. Ce diagramme est expliqué dans la figure 22.

****

**Figure 22 :** Diagramme de cas d’utilisation du robot

1. **Schéma descriptif du système robotique**

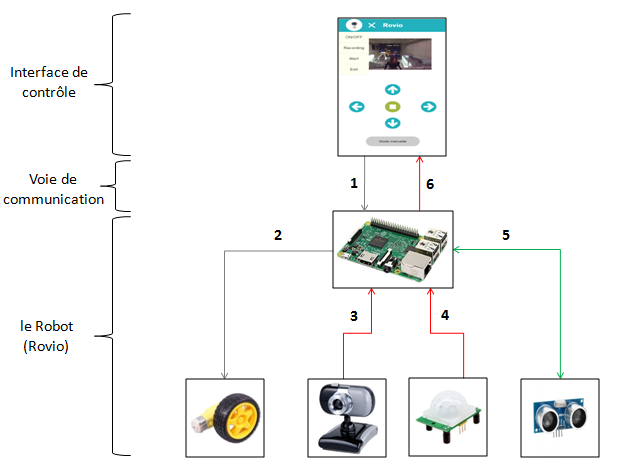
Le schéma descriptif du système robotique présente une décomposition hiérarchisée des fonctions du système, allant des fonctions de service et passant par les fonctions techniques jusqu'à l'énoncé des solutions technologiques. Cette décomposition est détaillée dans la figure 23.

ff.PNG

**Figure 23 :** Fonctionnalités du robot

1. **Système de collection de données et de contrôle**

Le système de surveillance de la salle serveurs peut être représenté par la figure 24.



**Figure 24 :** Système de collection de données et de contrôle

* Message n°1 : Ces messages sont émis à partir de l’application vers le robot. Ils se décomposent en trois catégories :
  + Message de la mise en marche du robot (on/off).
  + Message de commande du robot (avancer, tourner, reculer).
  + Message d’activation de la mode autonome.
* Message n°2 : Ces messages sont les ordres directs envoyés aux moteurs après traitement de messages n°1 au niveau de l’unité de traitement embarquée.
* Message n°3 : Ces messages représentent le flux vidéo transmis vers l’unité de commande.
* Message n°4 : Ces messages représentent l’état du capteur de mouvement transmis vers la carte Raspberry et détection de mouvement.
* Message n°5 : Ces messages se décomposent en trois catégories :
  + Ordre d’activation le mode autonome.
  + Transmettre des valeurs du capteur ultrason vers la carte Raspberry et détection d’obstacle.
* Message n°6 : Ces messages représentent le flux vidéo transmis en temps réel, l’enregistrement vidéo et les alerte du capteur de mouvement vers le centre de contrôle.

1. **Configuration de la carte Raspberry**

La carte Raspberry est très délicate et compliquée dans son utilisation. C’est pour cette raison que nous avons consacré cette partie pour décrire comment la prendre en main.

1. **Installation de système d'exploitation**

Avant que nous commencions à utiliser notre Raspberry Pi, nous devons installer le système d'exploitation Raspbian.

Tout d’abord, on télécharge la dernière image Raspbian sur notre ordinateur à partir du site officiel de Raspberry : <https://www.raspberrypi.org/downloads/raspbian/>.

On utilise la version "LITE", car nous installons le Raspberry Pi comme un serveur sans tête (nous allons nous connecter via SSH, sans avoir de clavier / écran connecté).

Ensuite, on insère la carte mémoire MicroSD dans notre ordinateur et on installe Etcher à partir de: <https://etcher.io/>.

Par ailleurs, Etcher est un programme permettant de faire clignoter des images sur des cartes mémoire.

Enfin, on lance Etcher en suivant ces étapes :

* On clique sur le bouton "Sélectionner une image" et on trouve le fichier zip Raspbian qu’on a téléchargé.
* On clique sur le bouton "Select drive" et on spécifie la carte mémoire comme emplacement cible.
* On clique sur le "Flash!" bouton pour écrire l'image sur la carte mémoire.

Une fois qu’Etcher a fini d'écrire l'image sur la carte mémoire, on la retire de notre ordinateur.

1. **Connexion avec la carte du Raspberry Pi aux périphériques**

Pour que le Raspberry Pi soit prêt à démarrer, on doit :

* Insérer la carte mémoire MicroSD dans le Raspberry Pi.
* Connecter le clavier USB.
* Connecter le câble HDMI.
* Connecter l'alimentation micro-USB.

Lorsque le Raspberry Pi a fini de démarrer, on se connecte en utilisant le nom d'utilisateur: pi et le mot de passe: raspberry.

1. **Choix de l’adresse IP du Raspberry Pi**

Il peut être intéressant de mettre la carte Raspberry pi en IP statique pour pouvoir la manipuler plus facilement.

* **Connaissance de l’adresse IP**

Le plus simple est d'utiliser la commande "ifconfig" dans un terminal du Raspberry Pi. Cette commande fournit les informations concernant le réseau (gateway), y compris aussi l'adresse IP.

* **Fixation de l’adresse IP**

Par défaut la Raspberry Pi utilise DHCP pour configurer ses interfaces réseau. Le DHCP pour eth0 est activé dans le fichier « /etc/network/dhcpcd.conf ». On se rendre donc dans le terminal et taper :

* *Sudo nano /etc/network/dhcpcd.conf*

Une fois le fichier ouvert, on doit passer vers la fin du fichier et ajouter ceci :

# configuration ip static pour eth0

interface eth0

static ip\_address=192.168.1.13/24

static routers=192.168.1.1

static domain\_name\_servers=192.168.1.1

Pour la connexion wifi :

# configuration ip static pour wlan0

interface wlan0

static ip\_address=192.168.1.9/24

static routers=192.168.1.1

static domain\_name\_servers=192.168.1.1

1. **Contrôle à distance via SSH**

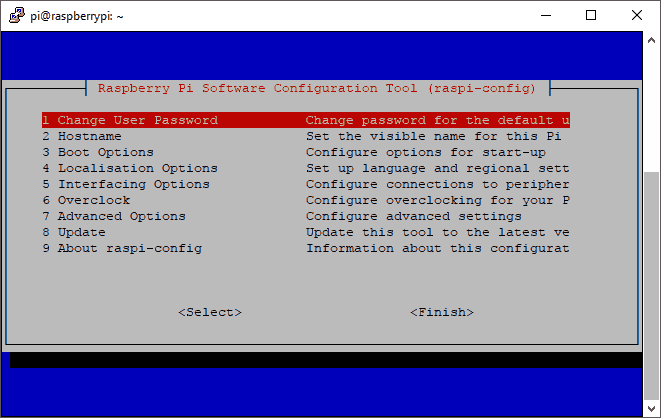
SSH désigne à la fois un logiciel et un protocole de communication informatiques. Ce protocole possède par ailleurs la particularité d’être entièrement chiffré. Cela signifie donc que toutes les commandes via SSH sont totalement secrètes.

SSH est aussi une fonctionnalité de Linux qui permet d’ouvrir une session de terminal sur la carte Raspberry Pi à partir d’une ligne de commande d'ordinateur hôte. Pour utiliser SSH, nous devrions l’activer.

Si le serveur SSH n’avait pas activé sur la carte Raspberry lorsqu’elle avait démarré pour la première fois, il suffit de passer au terminal et entrer la commande suivante pour démarrer la configuration du Raspberry :

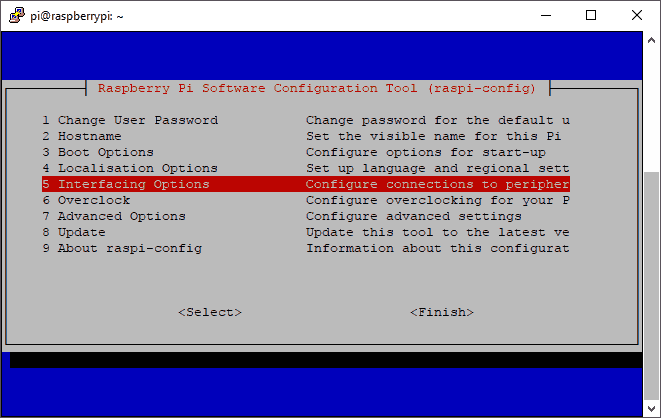
* *Sudo raspi-config*

La figure 25 présente le résultat d’affichage de cette commande.



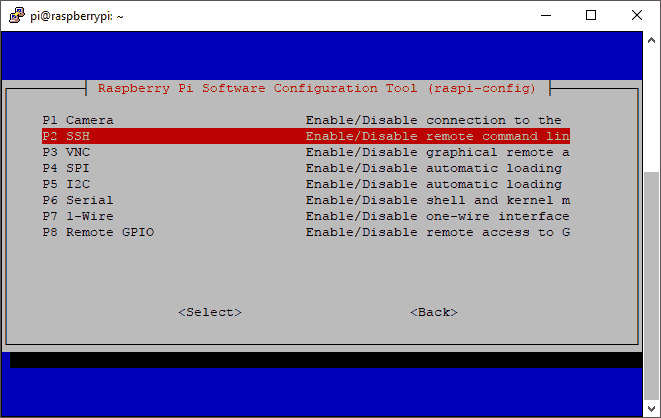
**Figure 25 :** Interface de configuration au premier démarrage

D’abord, on sélectionne l'option 5 Interfacing Options comme le montre la figure 26.

****

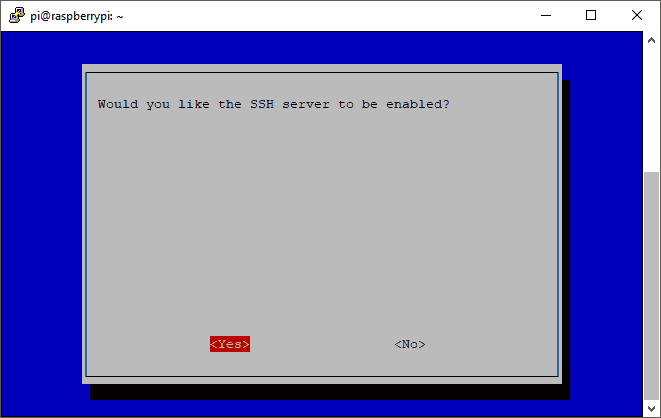
**Figure 26 :** Interface de configuration au deuxième démarrage

Ensuite, on sélectionne l'option P2 SSH, pour activer SSH comme l’indique la figure 27.



**Figure 27 :** Interface de configuration au troisième démarrage

Enfin, on confirme avec YES comme l’illustre la figure 28.

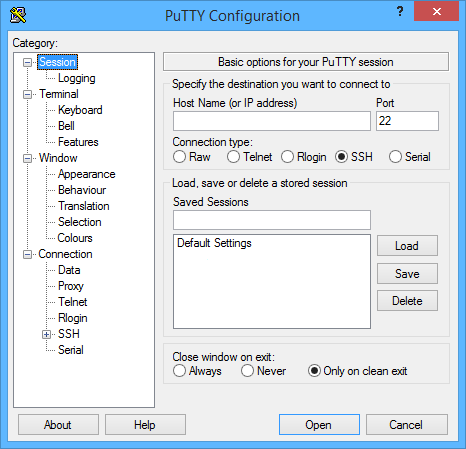


**Figure 28 :** Interface de configuration au quatrième démarrage

SSH est actuellement activé, et on devrait être dans le menu principal à nouveau.

Après l’activation de SSH, il reste seulement d’installer un client SSH sur notre ordinateur. Pour cela on va installer le logiciel PuTTy sous Windows : http://www.putty.org/.

Suite à l’installation, on ouvre PuTTy, on tape l'adresse IP de notre Raspberry Pi, puis on clique sur Open comme l’indique la figure 29.



**Figure 29 :** Logiciel PuTTy

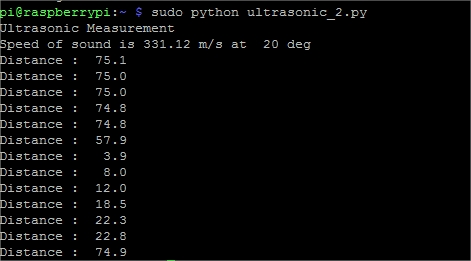
Ensuite, on se connecte en utilisant le nom d'utilisateur Pi et le mot de passe que vous avez spécifié.

Enfin, on se trouve directement sur l’invite des commandes de notre système d’exploitation Raspbian. Maintenant nous sommes en mesure d'exécuter notre Raspberry Pi en "mode sans tête".

1. **Test de la plateforme robotique**
2. **Capteur d’obstacle**

Le mode autonome permet à Rovio de s’autoguider sans avoir besoin de l’intervention d’un utilisateur pour l’orienter vers une direction précise et d’éviter d’entrer en collision avec un corps étranger. Pour se faire un module de détection d’obstacle a été mis en place.

Après l’exécution du code python, on constate le résultat dans la figure 30.



**Figure 30 :** Résultat de test du capteur ultrason

1. **Capteur de mouvement**

Face à la nécessité de l’ajout d’un module assurant la fonctionnalité de surveillance d’une manière plus efficace et plus performante, nous avons intégré à Rovio un module de détection du mouvement. En revanche, ceci permettra à notre robot de ne pas se limiter à la tache de la transmission en flux vidéo de ce qu’il voit en face, mais en plus de détecter et d’envoyer une alerte à l’utilisateur de l’application de tout corps dans son champ de vision que ce soit un être humain, un objet quelconque en déplacement ou un objet initialement en état stable subissant un mouvement.

La programmation du code python nous permet à chaque détection d’écrire « motion Detected » et si on n’a pas de détection d’écrire « pas de détection ». Le test du programme donne les résultats prévus comme l’indique la figure 31.



**Figure 31 :** Résultat de test du capteur de mouvement

1. **Moteur électrique**

Le mouvement du robot de surveillance est essentiellement le premier objectif de notre projet. C’est pour cela qu’on doit utiliser deux moteurs DC pour actionner et déplacer notre système robotique.

Après le codage du mouvement, on exécute notre programme.Le résultat du test est représenté par la figure 32.

## Capture.PNG

**Figure 32 :** Résultat de test du moteur électrique

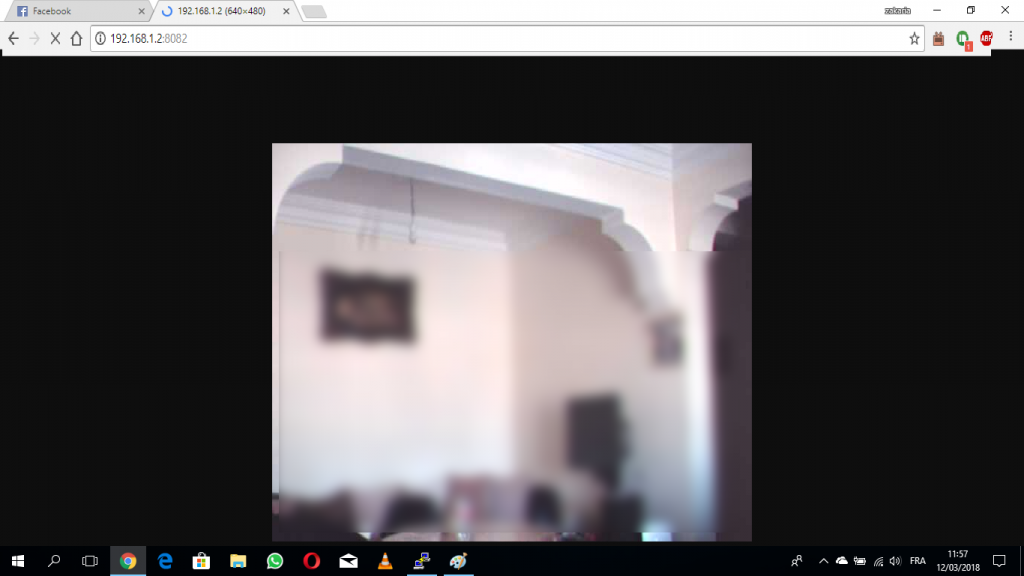
1. **Caméra**

La transmission du flux vidéo en temps réel est une tache primordiale. Elle consiste à faire visualiser la caméra du Rovio à distance afin d’assurer la commande du robot. De plus, elle permet d’enregistrer des vidéos dans le mode autonome. Pour cela, nous avons eu recours au « Motion » qui est un logiciel pour la surveillance vidéo et à accès en ligne, disponible pour le Raspberry Pi. Après l’installation et la configuration de ce logiciel on peut accéder à la vidéo diffusée sur le port du Raspberry Pi. Pour cela, on démarre Motion par la commande :

* *sudo service motion start*

Ensuite, on ouvre le navigateur et on tape l’adresse IP du Rapsberry Pi et le port 8081 (par défaut). Puis, on indique le login et mot de passe.

La figure 33 présente le résultat du test.



**Figure 33 :** Résultat de test de la caméra par Motion

1. **Conclusion**

Dans ce chapitre, nous avons détaillé les différentes fonctionnalités du système ainsi que les configurations et les tests des composants nécessaires à la réalisation de notre plateforme robotique.

Le dernier chapitre sera entièrement consacré pour développer l’application web qui permettra de gérer et consulter les différents équipements de notre robot.

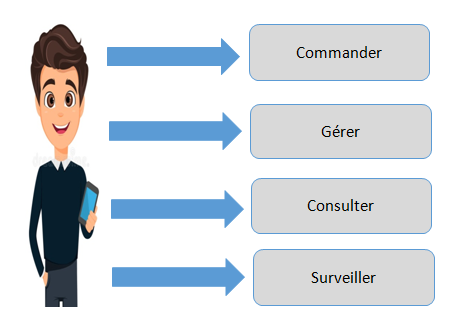
***Chapitre IV : Implémentation et réalisation d’une application web***

1. **Introduction**

L’exploitation des différentes fonctionnalités du système nécessite une interface visuelle qui facilite la gestion du robot et la surveillance des lieux. Pour cette raison, nous avons choisi de réaliser une application web. Ce dernier chapitre sera consacré pour la présentation de notre plateforme, tout en essayant de mettre l'accent sur les différents détails : le principe de fonctionnement, la configuration du serveur web, les langages de développement et la description des interfaces.

1. **Conception de l’interface de commande**
2. **Objectif**

Notre objectif est de créer une application web pour gérer notre robot à distance. Les différentes fonctionnalités offertes sont représentés dans la figure 34.



**Figure 34 :** Les différentes actions à bénéficier de l’application web

1. **Principe de fonctionnement de l'application**

Le Client qui est souvent l’utilisateur de robot envoi une requête à partir de son appareil (portable) vers le serveur apache de la carte Raspberry qui à son tour reçoit la demande, la décode puis affiche la page requise.

L'accès à l'application est modélisé dans la figure 35.



**Figure 35 :** Architecture client /serveur

1. **Développement de l’application web**
2. **Configuration de serveur web**

Un serveur web est un serveur informatique qu’on va utiliser pour publier ou héberger des sites web sur Internet. En effet, le serveur web n’est pas seulement une machine, mais désigne aussi le logiciel qui permet à la machine d’analyser les requêtes des utilisateurs (requêtes HTTP par exemple), et de retourner le fichier correspondant à la requête.

Dans ce sens, l’ordinateur Raspberry Pi est connecté en réseau par un connecteur Ethernet et/ou une liaison Wifi. Il peut devenir un serveur web permettant le contrôle distant de processus ou la transmission à distance des données à travers Internet.

1. **Installation de WebPy**

WebPy est un framework python qui permet d'écrire des pages web en utilisant python. La beauté de cela est, nous pouvons écrire Python en HTML ce qui rend les choses faciles. WebPy exécute un serveur dans Raspberry Pi et la page Web que nous avons créée en utilisant WebPy sera accessible à tous les appareils connectés au réseau. Après le démarrage du serveur, nous pouvons envoyer des commandes via la page HTML. Ces commandes seront prises en entrée et les actions correspondantes seront effectuées.

WebpPy accède au fichier index.html dans le dossier des modèles et au CSS, aux images, etc. dans le dossier statique.

Pour installer le WebPy dans notre Raspberry, on doit tapez le code suivant :

*~ $ wget http://webpy.org/static/web.py-0.37.tar.gz*

*~ $ tar xvzf web.py-0.37.tar.gz*

*~ $ cd web.py-0.37*

*~ $ sudo python setup.py installer*

1. **Installation d’Apache**

Le logiciel Apache est un [serveur HTTP](http://www.materiel-informatique.be/http.php) en Open Source utilisé principalement sur les [hébergements Internet](http://www.materiel-informatique.be/hebergement.php) en Linux, bien qu'il soit également utilisable en Windows, Unix ou OS X. C'est actuellement le plus utilisé sur le WEB. Différentes fonctionnalités sont implantées comme la possibilité d'utiliser un seul serveur Internet pour héberger [plusieurs sites](http://www.materiel-informatique.be/site.php), l'utilisation des langages interprétés Perl, [PHP](http://www.materiel-informatique.be/php.php) et Python, la création directe de [pages HTML](http://www.materiel-informatique.be/html.php), etc.

Avant d’installer le serveur Apache, on doit assurer que la machine est bien à jour. Pour ce faire nous devons posséder les droits d’administrateur, soit en étant connecté en root, soit via la commande sudo.

* *Sudo apt-get update et Sudo apt-get upgrade*

Une fois la Raspberry Pi est à jour, il faut installer le serveur Apache.

* *Sudo aptitude install apache2*

1. **Installation de PHP**

Après l’installation de serveur Apache, on a besoin d’installer un interpréteur de langage PHP, afin d’utiliser le langage, qui est principalement utilisé pour rendre notre application dynamique. C’est-à-dire que l’utilisateur envoie des informations au serveur qui lui renvoie les résultats modifiés en fonction de ces informations, contrairement à un site statique qui ne s’adapte pas aux informations fournies par un utilisateur.

L’installation de PHP se fait avec la ligne de commande :

* *Sudo aptitude install php*

1. **Installation de MySQL**

Pour stocker les informations d’une façon structurée, cohérente et organisée sur le serveur, on a besoin d'utiliser une base de données qu’on va l’exploiter via un SGBD. Dans notre cas, on utilise MySQL.

Par la suite, on doit installer mysql-server et php5-mysql (qui servira de lien entre php et mysql) en tapant les commandes suivantes :

* *sudo aptitude install mysql-server php5-mysql*

Lors de l’installation de mysql-server, il nous demande un mot de passe pour le compte administrateur MySQL, qui sera utilisé plus tard dans l’accès au serveur.

1. **Configuration phpMyAdmin pour fonctionner avec apache**

PhpMyAdmin est en fait une application développée en PHP. Elle vise à fournir une interface simplifiée pour MySQL.

L'objectif est de modifier la configuration d’apache pour pouvoir accéder à phpMyAdmin via ce serveur. Cela se fait en tapant la commande suivante :

* *Sudo nano /etc/apache2/apache2.conf*

Le fichier de configuration sera chargé avec Nano. A la fin de ce fichier, on ajoute la commande suivante :

* *Include /etc/phpmyadmin/apache.conf*

Finalement, On sauvegarde le fichier et on redémarre le service apache /etc/init.d/apache2 restart.

1. **Langages et outils utilisées**
2. **Langages**

* HTML

HTML est un langage de description et non pas un langage de programmation. Ce langage est basé sur le principe de balisage de texte qui permet d’inclure des informations variées (textes, images, sons, animations etc.) et de créer des documents hypertextes à fin d'établir des relations cohérentes entre ces informations affichables par un navigateur.

La figure 36 présente une image sur le logo du HTML.



**Figure 36 :** Logo du HTML

* CSS

Les feuilles de style en cascade CSS sont un langage qui permet de contrôler facilement la présentation visuelle des pages web et d’ajouter facilement du style (polices, couleurs, espacement, etc.) à ces pages.

Le but de CSS est séparer la structure d'un document HTML et sa présentation. En effet, avec HTML on peut définir à la fois la structure et la présentation de la page, mais cela pose quelques problèmes. Avec le couple HTML/CSS, on peut créer des pages web où la structure du document se trouve dans le fichier HTML tandis que la présentation se situe dans un fichier CSS externe, pour éviter les redondancesde classes identiques, tout en clarifiant le code. De plus, CSS ajoute des fonctionnalités nouvelles par rapport à HTML au point de vue du style.

La figure 37 présente une image sur le logo du CSS.



**Figure 37 :** Logo du CSS

* JavaScript

JavaScript est un langage de programmation objet et événementiel utilisé pour la gestion des interactions dans les navigateurs web. JavaScript est lié au langage de balisage HTML. On peut insérer du code JavaScript dans des pages HTML grâce à la balise <script> ou par l’appel des propriétés et des méthodes, qui existent dans un fichier script (.js) et leur associer des actions en fonction d'événements déclenchés par le visiteur.

Nous avons décidé d’utiliser JavaScript pour contrôler les données saisies dans les formulaires HTML, par exemple pour vérifier si le mot de passe saisit par l’utilisateur est bien correct.

La figure 38 présente une image sur le logo du JavaScript.



**Figure 38 :** Logo du JavaScript

* *JQuery*

JQuery est une bibliothèque JavaScript qui facilite le développement d’applications web et s’appuie sur l'interaction entre JavaScript et HTML. Elle simplifie des commandes communes de JavaScript ajoute des fonctionnalités supplémentaires aux sites Web.

La figure 39 présente une image sur le logo du jQuery.



**Figure 39 :** Logo du jQuery

* AJAX

AJAX est une architecture qui permet de construire des sites web dynamiques interactifs à base de différentes technologies ajoutées aux navigateurs web.

L'idée de d'AJAX est de faire communiquer une page Web avec un serveur Web en évitant les téléchargements de pages complètes et en réduisant le trafic entre le client et le serveur. Un moteur Ajax est en fait un objet ou une fonction JavaScript qui est appelée chaque fois qu'une information est sollicitée auprès du serveur, car ce modèle traditionnel fournit un lien vers une autre ressource (une autre page Web par exemple). Chaque lien fait appel au moteur Ajax, lequel planifie et exécute la requête. Cette dernière est établie de manière asynchrone, ce qui signifie que l'exécution du code n'attend pas de réponse avant de se poursuivre.

La figure 40 présente une image sur le logo de l’AJAX.



**Figure 40 :** Logo d’AJAX

* Langage PHP

PHP est un langage de programmation qui s’intègre dans nos pages HTML. Il permet de rendre automatiques des tâches répétitives, notamment grâce à la communication avec une base de données.

PHP peut être utilisé soit comme un langage de script répondant à des besoins simples et à court terme, soit comme un langage de programmation complexe permettant de mettre en place des applications d’entreprise.

La figure 41 présente une image sur le logo du PHP.

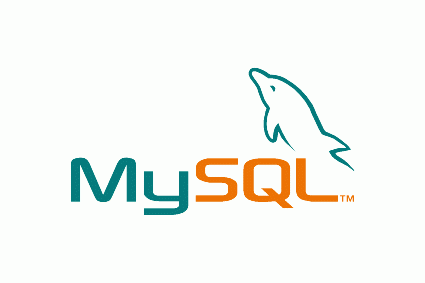
****

**Figure 41 :** Logo du PHP

* MySQL

MySQL est un système de base de données, souvent couplé au langage PHP. Ce système utilise le langage SQL pour interroger, alimenter ou mettre à jour les bases de données. Ainsi, dans le cas d’un site Web, le langage PHP va établir la connexion à la base de données, puis envoyer le code SQL nécessaire aux opérations sur la base de données. En retour, MySQL va renvoyer des données ou des informations, qui seront traitées par le code PHP.

La figure 42 présente une image sur le logo du MySQL.



**Figure 42 :** Logo du MySQL

1. **Outils**

* Visual Studio code

Visual Studio Code est un éditeur de code source récemment lancé par Microsoft avec la particularité d’être diffusé sur Github donc en open source. Il est disponible pour Windows mais aussi Linux et Mac OS X et vient concurrencer directement les éditeurs modernes tels que : Atom, Brackets et SublimeText.

La figure 43 présente une image sur le logo du Visual Studio Code.

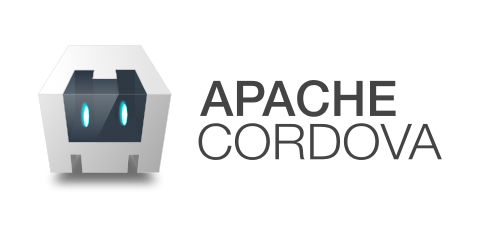


**Figure 43 :** Logo du Visual Studio Code

* Apache Cordova

Apache Cordova est un framework de développement mobile open-source. Il permet d'exploiter les technologies Web courantes telles que HTML5, CSS3 et JavaScript pour développer des applications multi-plateformes, évitant ainsi l'utilisation des langages natifs propres aux différentes plates-formes mobiles. Les applications s'exécutent dans des wrappers ciblés pour chaque plate-forme. Elles s'appuient sur des API conformes aux standards permettant l'accès aux ressources de chaque appareil, etc. Nous avons utilisé ce Framework pour le développement de notre application mobile multiplateforme.

La figure 44 présente une image sur le logo de l’Apache Cordova.

****

**Figure 44 :** Logo d’Apache Cordova

* Android Studio

Android Studio est créé par Google I/O le 15 mai 2013, destiné pour la création des applications mobile Android. Il est open source et disponible gratuitement, permettant de réaliser des projets sur différents types de support, tablette ou Smartphone. Principalement utilisé pour éditer des fichiers Java étant le langage d’une application Android native ainsi que des fichiers de mise en page XML avec la possibilité de visualiser le rendu et les manipuler en utilisant une interface graphique.

La figure 45 présente une image sur le logo d’Android Studio.



**Figure 45 :** Logo d’Android Studio

1. **Description de l'interface de l'application**

Dans ce qui suit, nous présentons les différentes interfaces de l’application que nous avons réussie à les mettre en place.

Ces interfaces englobent six possibilités de traitement fondamentales :

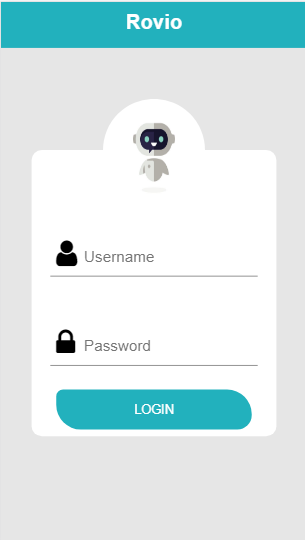
* La mise en marche du robot
* La télésurveillance
* La commande du robot
* L’envoi d’alerte en cas d’instruction
* Le mode autonome
* L’enregistrement des vidéos

1. **Page d’authentification**

La sécurité est un enjeu majeur pour notre application. C’est une contrainte à ne pas négliger. C’est pourquoi qu’on doit assurer l'authentification de l'utilisateur.

Lors de lancement de notre application, l’interface d’authentification s’affiche et invite l’utilisateur à se connecter en introduisant le nom d’utilisateur et le mot de passe à condition qu'ils aient déjà inscrit par l’administrateur. Ainsi, il sera capable d'accéder et de profiter de différents services de l'application. L'inscription est une tâche privée qui est gérée seulement par l'administrateur.

La page d’authentification dispose d’un logo de l’application, deux zones de texte « Username » et « Password » et un bouton de connexion « LOGIN » comme le montre la figure 46.



**Figure 46 :** Interface d’authentification

1. **Page de surveillance**

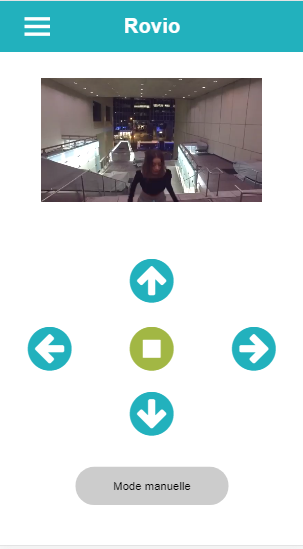
Après l’authentification, l’utilisateur sera redirigé par le système vers l’interface d'accueil. Cette interface permet au client de piloter son robot à travers les flèches directionnelles et également de visualiser en temps réel ce qui se déroule à travers la caméra.

En effet, la vidéo s'affichera automatiquement dés le lancement de la page d’accueil grâce au système de surveillance en ligne.

En cliquant sur le bouton gris, le système se change de la mode manuelle à la mode automatique. Dans cette mode, le robot sera autonome à l’aide du capteur ultrason.

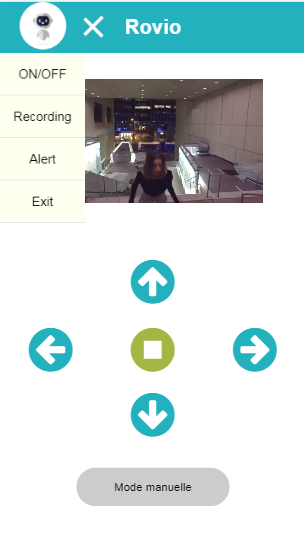
Donc on n’est pas obliger d’utiliser les flèches directionnelles pour faire avancer, reculer ou tourner le robot.

La figure 47 présente l’interface d’accueil.



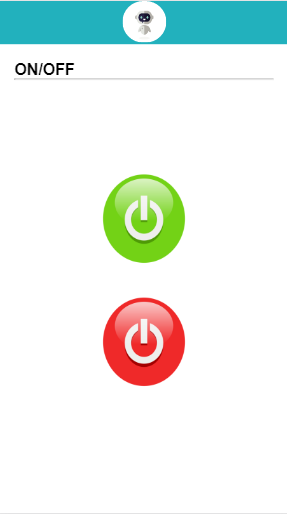
**Figure 47 :** Interface page d'accueil

Comme l’illustre la figure 48, notre espace est dotée d’un menu déroulant intuitif qui englobe les fonctionnalités offertes par notre application.



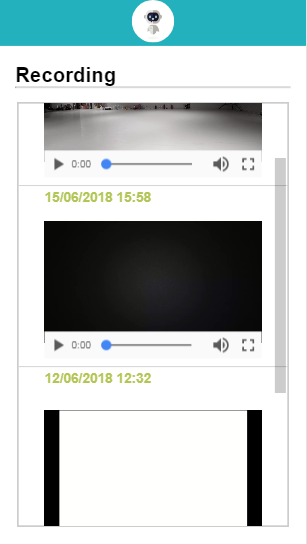
**Figure 48 :** Interface menu principal

Lorsque que l’utilisateur veut contrôler la mise en marche de son robot il suffit de cliquer sur l’item « ON/OFF » et d’appuyer soit sur le bouton vert pour l’activer, soit sur le bouton rouge pour le désactiver comme l’indique la figure 49.



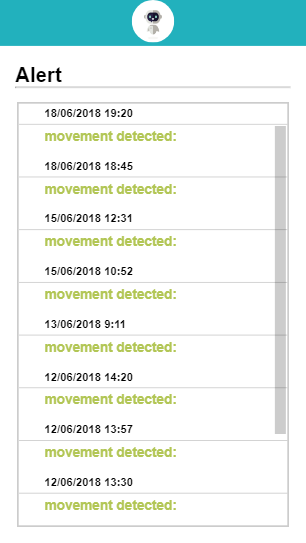
**Figure 49 :** Interface ON/OFF

Lorsque que l’utilisateur veut consulter les enregistrements vidéo il suffit de cliquer sur l’item « Recordnig » L’ensemble des vidéos s’affiche dans une liste qui contient la date d’enregistrement comme le modélise la figure 50.



**Figure 50 :** Interface Recording

Lorsque que l’utilisateur veut consulter les messages d’alerte il suffit de cliquer sur l’item « Alert ». Cette interface est réservée pour informer le client d’un cas de mouvement en lui envoyant un message détaillé (date et heure de détection). Ces notifications sont représentées dans la figure 51.



**Figure 51 :** Interface Alert

1. **Conclusion**

Ce dernier chapitre décrit le fruit de notre travail, il présente la conception, le développement et la description des différentes interfaces de notre application.

Conclusion générale et perspective

A travers notre expérience réalisée dans la société TECHNO-SOFT dans le cadre de stage de fin d’étude, nous avons eu l’occasion d’approfondir nos connaissances théoriques acquises pendant les différents stades de notre formation et d’acquérir une bonne compétence au niveau de la réalisation pratique et le développement du système embarqué.

Ce projet s’articule autour de la conception et la réalisation d’un robot mobile, commandé à distance via une application web.

Pour aboutir aux objectifs décrits dans le cahier des charges, nous avons répartie notre travail en quatre grandes parties. La première pour une présentation générale du projet. La deuxième pour une étude préalable et le choix des matériels et logiciels. La réalisation de plateforme robotique a fait l’objet de la troisième. Le quatrième chapitre a été consacré à détaillé le principe de fonctionnement de l’application web ainsi que le développement et la description de interfaces.

Au terme de ce travail effectué, nous considérons que ce projet nous a été bénéfique et enrichissant vu qu’il nous a permis de renforcer nos connaissances vers la conception d’un système robotique qui sera utile dans le domaine de l’électronique embarqué. En effet nous avons eu l’occasion de manipuler une nouvelle technologie pour la surveillance avec des différents composants .En plus nous nous sommes familiarisées avec des cartes électroniques des techniques que nous ne maitrisions pas auparavant tel que le Raspberry Pi 3. Nous avons pu à travers ce stage approfondi nos acquis et nos performances en programmation avec le langage python.

Le projet conçu est évolutif, en effet on peut lui rajouter d’autres améliorations pour réaliser un système plus performant qui répond aux nouvelles technologies courantes telles que :

* Implémenter un système de reconnaissance facial.
* Rendre le robot accessible depuis n’importe qu’elle emplacement à travers l’internet.
* Retour automatiquement pour charger.
* Détecter les incendies et les fuites de gaz.

Bibliographie

<http://www.technosoft.tn>

<https://raspberry-pi.developpez.com>

<https://www.raspberrypi.org>

<http://espace-raspberry-francais.fr>

https://www.w3schools.com

<https://etcher.io/>

http://www.putty.org

<http://electroniqueamateur.blogspot.com>

https://www.developpez.com

Annexe

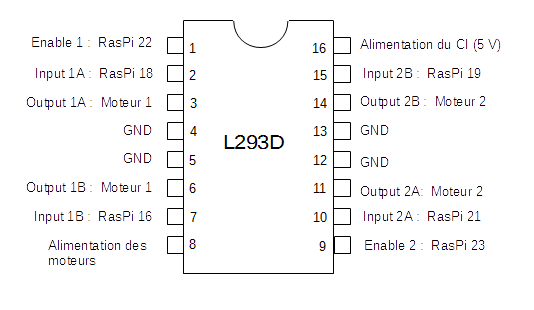
1. **Brochage**
2. **Carte Raspberry**

Le brochage du L293D est illustré comme le montre la figure suivante :



1. **Circuit intégré L293D**

Le brochage du L293D est illustré comme le montre la figure suivante :



1. **Motion**
2. **Installation de motion sur la Raspberry Pi.**

Dans un premier temps, nous **assurons que la Raspberry Pi est bien à jour à l’aide des deux commandes suivantes :**

* *sudo apt-get update*
* *sudo apt-get upgrade*

Une fois ces commandes terminées et la Raspberry Pi à jour, nous installons maintenant Motion.

Puisque nous utilisons une webcam standard avec notre Raspberry Pi, **nous allons chercher Motion en ligne, dans les dépôts officiels de Raspbian.** Il nous suffit alors d’utiliser la ligne de commande suivante :

* *sudo apt-get install motion*

Ceci fait,**nous lançons la commande suivante pour rendre la caméra du Raspberry Pi compatible avec motion :**

* *sudo modprobe bcm2835-v4l2*

Sitout c’est bien passé, il ne nous reste plus **qu’à rendre cette modification permanente :**

* *echo "bcm2835-v4l2" | sudo tee -a /etc/modules*

Motion est maintenant installée sur notre Raspberry Pi, nous allons donc pouvoir passer à l’étape suivante.

1. **Configurer Motion sur votre Raspberry Pi**

Nous allons maintenant tâcher de configurer Motion sur notre Raspberry Pi pour en faire une caméra de sécurité.

Dans un premier temps, nous assurons que Motion tourne en permanence. Pour cela, nous éditons le fichier « /etc/default/motion » et nous modifions la ligne

« start\_motion\_daemon » en « *start\_motion\_daemon=yes* »

Maintenant, nous modifions un certain nombre de points de la configuration de Motion, afin de régler au mieux notre système de vidéo-surveillance.

Pour cela, nous éditons le fichier « /etc/motion.conf », dans certains cas ce fichier se trouvera plutôt dans « /etc/motion/motion.conf ».

Nous allons modifier certaines lignes de ce fichier de façon à ce qu’elles correspondent aux lignes ci-dessous :

#On active le daemon

daemon on

#On utilise le fichier /tmp/motion.log pour stocker les logs

logfile /tmp/motion.log

#On règle la résolution (adapter la résolution à celle de votre camera. Ici, nous avons réglé pour la caméra de la raspberry pi)

width 1280

height 720

#On règle le nombre d'images prises par seconde

framerate 2

#On règle le nombre d'images enregistrées avant et après un mouvement (ici 2 avant 2 après)

pre\_capture 2

post\_capture 2

#On règle le code, pour une lecture sur de nombreux médias

ffmpeg\_video\_codec msmpeg4

#On autorise la lecture du flux depuis internet pour permettre l'accès en ligne à la vidéo-surveillance

stream\_localhost off

Par défaut Motion diffuse la vidéo sur le port 8080 de la Raspberry Pi. Si nous souhaitons modifier ce port, nous éditons le fichier de configuration vu précédemment et modifier la ligne « webcontrol\_port ».

Pour ce qui est de la résolution, nous privilégions une résolution compatible avec la caméra ou éventuellement une résolution inférieure. En effet demander du fullHD n’est pas très logique si notre caméra ne permet que 480p !

De plus nous devons penser à votre connexion, si nous avons une simple connexion ADSL la diffusion par internet avec un framerate élevé (24ips) ou une forte résolution sera possiblement un peu lente.

1. **Accéder à la surveillance vidéo de votre Raspberry Pi à distance**

Nous allons, très probablement, avoir envie de voir notre vidéo surveillance en direct sur votre Raspberry Pi depuis n’importe où. Nous avons d’ailleurs configuré Motion dans ce sens.

Une fois que notre Raspberry Pi est accessible à distance, nous n’avons plus qu’à y accéder par internet en tapant directement l’URL de notre Raspberry Pi dans notre navigateur (si possible plutôt Firefox, Chrome semble avoir quelques difficultés), suivi du numéro de port (ex : <http://votreurl.no-ip.biz:8080>). Nous pouvons également accéder au flux vidéo via VLC !

Voilà, notre Raspberry Pi est maintenant équipée pour faire un système de vidéosurveillance efficace !