

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE

**Mémoire de Fin d'Etudes  
de MASTER ACADEMIQUE**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **réseaux mobilité et systèmes  
embarqués**

*Présenté par*  
**ABDELI Lynda**

Thème

**Systeme de contrôle d'accès avec la  
Raspberry Pi**

Mémoire soutenu publiquement le 29/09/2018 devant le juré composé de :

Encadreur : M HEMDANI Chabane

Président : M SADOU Samir

Examineur : M YACINE Younes

2017/ 2018

# remerciement

*Au terme de ce modeste travail je remercie Dieu le tout puissant de nous accordé d'avoir accompli ce travail, qui présente le fruit de plusieurs années de sacrifices.*

*J' exprime mon profond et sincère remerciement à mon promoteur M<sup>me</sup> Hemdani, pour avoir crue en mes capacités pour mener à bien mon travail, qu'il a toujours dirigé avec une très grande rigueur scientifique. Sa disponibilité, ses conseils, ses encouragements et sont bonne humeur tout le long de mon travail.*

*Je tien également à remercier toute personne ayant contribué de près ou de loin à la réalisation de ce projet.*

*Mon vif remerciement ira aussi aux membres de jury qui me fera l'honneur de le juger et de l'enrichir par leurs propositions.*

# *Dédicaces*

*Je dédie ce mémoire à :*

*A mes chers parents*

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.*

*A Ma grande mère pour son amour et gentillesse*

*A la mémoire de mes grands-mères*

*A ma sœur Sabrina et son mari Mohamed*

*A mes frères Hamid et Mazigh*

*A mes amis qui m'encouragent*

*A tous ceux qui m'aiment*

**LYNDA**

## Liste des figures

Figure I.1 : Les critères d'efficacité d'un système embarqué.....	4
Figure II.2: Différentes catégories de systèmes embarqués .....	5
Figure III.3: Organisation d'un système embarqué.....	6
Figure II.1:Modèle A .....	16
Figure IV.2: Modèle B.....	17
Figure II.3:Modèle A+.....	18
Figure II.4:Modèle B+ .....	18
Figure II.5:Modèle 2B .....	18
Figure II.6:Modèle 3B .....	20
Figure VI.7:Modèle 0 .....	21
Figure II.8: Modèle zero w .....	21
Figure II.9: Modèle zéro WH .....	22
Figure II.10: Logo du système d'exploitation Raspbian .....	23
Figure II.10: Logo du système d'exploitation ArchLinux ARM.....	23
Figure II.11: Logo du système d'exploitation RISC OS .....	24
Figure II.11: Logo du openelec .....	24
Figure II.12: Raspberry pi comme un pc du bureau .....	25
Figure II.13: le bureau de Raspbian.....	26
Figure III.1 : Schéma de fonctionnement global du système .....	38
Figure III.2 : Vue simplifiée de l'architecture générale du système .....	39
Figure III.3: GPIO Raspberry Pi model 3B .....	40
Figure III.4: camera USB .....	41
Figure III.5: moteur pas à pas avec son driver UNL2003 .....	42
Figure III.6: Le graphe de flux de données pour le système embarqué dans le système de contrôle d'accès. ....	43
Figure III.7: Le graphe de Raffinement successives pour le système embarqué dans le système de contrôle d'accès .....	45
Figure III.8:Diagramme de cas d'utilisation relatif à l'utilisateur .....	48
Figure III.9 : Diagramme de cas d'utilisation relatif à l'administrateur.....	49
Figure III.10 : Diagramme de séquence « Demande d'accès de l'utilisateur ».....	50
Figure III.11: Diagramme de séquence « authentification de l'administra	
Figure III.12 : table des utilisateurs .....	51
Figure IV.1 : Architecture de connexion client-serveur avec PHP .....	54

FigureIV.2 : Installation de Raspbian avec Win32 Disk Imager.....	57
FigureIV.3 :configuration de l'adresse IP statique .....	58
Figure IV.4: configuration des paramètres wifi.....	58
Figure IV.5 : Logiciel Putty.....	59
Figure IV.6 : Invite de commande via SSH.....	60
Figure IV.7: LogicielTightVNC VIEWER .....	60
Figure IV.8 : Interface graphique sur logiciel TightVNC VIEWER.....	61
Tableau généré par la commande phpinfo sur une raspberry .....	63
Figure IV.10: activation du Daemon .....	65
Figure IV.11: désactivation la capture de la vidéo. ....	65
Figure IV.12: configuration de la caméra.....	66
Figure IV.13: Schéma de brochage entre le driver UNL2003 et la Raspberry Pi .....	67
Figure IV.14 : les branchements des composants de notre système.....	67
Figure IV.15 : Page d'authentification d'utilisateur. ....	68
Figure IV.16 : page d'authentification du l'administrateur .....	69
Figure IV.17 : partie ajout et suppression des utilisateurs.....	70
Figure IV.18 : partie vidéosurveillance .....	71
Figure IV.19 : partie photos.....	71
Figure IV.20: l'interface de l'administrateur via un smartphone .....	72

## Liste des tableaux

Tableau.1 : Les scenarios des tâches de l'administrateur de notre application .....	47
Tableau.2 : Les scenarios des tâches de l'administrateur de notre application .....	48

PDF Create! 5 Trial  
www.nuance.com

# Sommaire

## Introduction générale

### Chapitre I

I.2 Système embarqué .....	
I.1 Introduction .....	1
I.3 Historique .....	
I.4 Caractéristiques spécifiques d'un système embarqué.....	2
I.5 Domaines d'application d'un système embarqué .....	4
I.6 Catégories des systèmes embarqués.....	5
I.7 Architecture matérielle d'un système embarqué.....	6
I.7.1 Processeurs embarqués .....	8
I.7.2 Mémoires .....	
I.7.2.1 La mémoire vive RAM .....	9
I.7.2.2 La mémoire morte ROM .....	9
I.7.3 Entrées-sorties .....	10
I.8 Architecture logicielle d'un système embarqué.....	10
I.8.1 Systèmes d'exploitation embarqué .....	10
I.8.2 Programmation .....	12
I.9 Conclusion .....	

### Chapitre II

II.2 Introduction .....	15
II.2 Carte Raspberry .....	15
II.3 L'histoire de Raspberry Pi.....	15
II.4 Les modèles de Raspberry Pi.....	16
II.4.1 Raspberry Pi 1 .....	16
II.4.1.1 Modèle A.....	16

II.4.1.2	Modèle B .....	17
II.4.1.3	Différence du modèle A+ avec le A.....	17
II.4.1.4	Différence modèle du B+ avec le B .....	18
II.4.2	Raspberry Pi 2 .....	18
II.4.3	Raspberry Pi 3 .....	19
II.4.3.1	Raspberry Pi 3 modèle 3B .....	19
II.4.3.2	Raspberry Pi 3 modèle 3B+.....	20
II.4.4	Raspberry pi zero.....	20
II.4.4.1	Raspberry Pi Modèle Zéro.....	20
II.4.4.2	Raspberry Pi Zero W .....	21
II.4.4.3	Raspberry Pi Zero WH .....	21
II.5	Systèmes d'exploitation pour la Raspberry pi.....	22
II.5.1	Debian (Raspbian) : .....	22
II.5.2	Arch Linux ARM : .....	23
II.5.3	Risc OS : .....	23
II.5.4	OpenELEC et Raspbmc :.....	24
II.6	Les différentes utilisations du Raspberry Pi.....	25
II.6.1	Un ordinateur, tout simplement .....	25
II.6.2	Les systèmes embarqués .....	26
II.6.3	La domotique .....	26
II.6.4	L'utilisation MultiMedia .....	26
II.6.5	Les serveurs.....	26
II.6.5.1	Le serveur web : .....	27
II.6.5.2	Le serveur de stockage :.....	27
II.6.5.3	Le serveur de service : .....	27
II.7	Système de contrôle d'accès .....	28
II.8	Objectifs de contrôle d'accès .....	28

II.9 Composants de base des systèmes de contrôle d'accès .....	29
II.9.1 Face à l'utilisateur : .....	29
II.9.1.1 support .....	29
II.9.1.2 lectures .....	29
II.9.2 Face à l'administrateur : .....	30
II.9.3 Infrastructure : .....	30
II.10 Les différentes fonctionnalités de contrôle d'accès.....	30
II.10.1 Le contrôle d'accès pour optimiser la sécurité.....	31
II.10.2 Le contrôle d'accès pour gérer le flux des visiteurs.....	31
II.10.3 Le contrôle d'accès pour gérer les accès.....	31
II.11 Modes de communication .....	32
II.11.1 Communication avec des cartes à puce .....	32
II.11.2 RFID .....	32
II.11.3 Near field communication .....	32
II.11.4 RCID .....	32
II.11.5 communication avec Smartphone.....	33
II.11.6 communication basé sur L'IOT.....	33
II.12 Vidéosurveillance dans les systèmes contrôle d'accès.....	34
II.13 Gestion de porte.....	34
II.13.1 Types de portes .....	34
II.13.2 Équipements de contrôle de portes.....	35
II.14 Conclusion .....	35
Chapitre III	
III.1 Introduction .....	36
III.2 Analyse .....	36
III.2.1 Objectif du système.....	36
III.2.2 Fonctionnalité du système .....	36

III.2.3	Spécification des besoins fonctionnels et non fonctionnels .....	37
III.2.3.1	Besoins fonctionnels .....	37
III.2.3.2	Besoins non fonctionnels .....	37
III.2.4	Spécification des besoins matériels.....	38
III.2.4.1	Composant requis .....	
III.2.4.1.1	Module de traitement .....	
III.2.4.1.2	Module d'acquisition .....	41
III.2.4.1.3	Module de Réaction .....	41
III.2.4.1.4	Module d'affichage .....	42
III.3	Conception .....	43
III.3.1	Le graphe de flux de données.....	43
III.3.2	Le graphe de Raffinement successif.....	43
III.4	Analyse et conception de notre application web.....	46
III.4 .1	Langage UML .....	46
III.4.2	Vue fonctionnel .....	46
III.4.2 .1	Les acteurs de notre système .....	46
III.4.2 .2	Spécification des scénarios .....	47
III.4.2.3	Spécification des cas d'utilisation .....	48
III.4.3	Vue dynamique .....	49
III.4.3.1	Les diagrammes de séquence .....	49
III.5	La base de données : .....	
III.6	Conclusion.....	
Chapitre IV		
IV.1	Introduction.....	53
IV.2	Outils de développement utilisés.....	53
IV.2.1	Outils logiciels .....	53
IV.2.1.1	Le serveur web apache2.....	54
IV.2.1.2	MYSQL.....	54
IV.2.1.3	PHPMyAdmin .....	54
IV.2.1.4	Motion .....	54

IV.2.2 Langages utilisés .....	55
IV.2.2.1 HTML5 (Hypertext Markup Language 5).....	55
IV.2.2.2 JavaScript .....	55
IV.2.2.3 JQuery.....	55
IV.2.2.4 Ajax(Asynchronous JavaScript and XML) .....	55
IV.2.2.5 Ccss3 .....	56
IV.2.2.6 Python.....	56
IV.2.3 Outils matériel : .....	56
IV.3 Installation et configuration .....	56
IV.3.1 Préparation de Raspberry Pi .....	56
IV3.2 Installer le serveur web sur la Raspberry Pi.....	62
IV.3.3 préparation de la webacam (caméra USB) .....	63
IV3.3.1 capture de photo .....	64
IV3.3.2 vidéosurveillance.....	64
IV.4 Réalisation matérielle .....	66
IV.5 Présentation de l'application .....	68
IV.5.1 interface utilisateur .....	68
IV.5.2 Interface administrateur .....	69
IV.6 Conclusion .....	72
Conclusion générale .....	73
Perspectives .....	73

# *Introduction générale*

Le progrès de la technologie a permis une avancée sans précédente dans la maîtrise des processus ces progrès sont bien-sûre suivis d'un nombre indéfini de défis techniques et scientifiques. Parmi ces défis, on trouve le risque lié à la sécurité du personnel, matériel, environnement ...etc. Donc, le perfectionnement de ces systèmes reste une priorité pour assurer : le bon fonctionnement, la sécurité, des processus dans un environnement à risque. La sécurité représente une importance primordiale pour toutes les entreprises, que ce soit pour un système de surveillance, un système de contrôle d'accès, ce dernier devient de plus en plus populaire dans beaucoup d'entreprises, toutes catégories confondues. La capacité de limiter l'accès à des personnes pré-autorisées pour des salles d'entraînement, ou à circuler dans les différents départements de l'entreprise est certainement très attrayante. Toutefois l'électronique moderne et la technologie informatique ont apporté au système de contrôle d'accès un tout nouveau champ d'application tel que le contrôle d'accès mobile.

Les Smartphones sont partout. Tout le monde en a un et personne ne s'en sépare. La gamme des fonctions et applications s'étend à une vitesse exponentielle, et l'une des utilisations qui connaît la croissance la plus rapide concerne l'accès mobile aux bâtiments, parkings, réseaux et autres ressources. Elle tire parti du monde de plus en plus mobile dans lequel nous vivons, en permettant aux Smartphones, tablettes et autres appareils portables, de faire office d'identifiants et ça pour plusieurs raisons : premièrement Les technologies de présentation ou basées sur un geste sont bien plus faciles qu'une bonne vieille carte d'identification ainsi que les cartes d'identification ont une valeur moins importante aux yeux des utilisateurs et sont souvent perdues, tandis que les appareils mobiles sont précieusement conservés. En fin les cartes d'identification nécessitent l'accès à un lecteur visible, facile à vandaliser, toutefois l'accès mobile avec une plage de lecture plus longue permet de placer les lecteurs à l'intérieur des bâtiments et derrière des portes verrouillées, bien hors de portée.

C'est dans ce cadre que se situe notre projet de fin d'études intitulé « système contrôle d'accès avec la Raspberry PI » ce projet a pour objectif de développer une application embarquée sur la Raspberry Pi en utilisant les technologies web afin d'assurer le contrôle d'accès à des zones particulières.

Pour ce faire nous avons divisé notre travail en quatre chapitres, dans le premier chapitre on a présenté des généralités sur les systèmes embarqués telles que l'architecture matérielle, les systèmes et les langages utilisés.

Le deuxième chapitre est divisé en deux parties, dont la première partie concerne une présentation sur la Raspberry Pi, l'historique, ses modèles, ses systèmes d'exploitation et ses différentes utilisations, dans la deuxième partie nous avons illustré quelques principes de contrôles d'accès

Le troisième chapitre est consacré à l'analyse et conception de notre système.

La réalisation de notre système sera présentée dans le quatrième chapitre, dans lequel nous présenterons l'environnement de développement et les divers composants implémentés dans l'architecture de notre système.

Nous finirons ce rapport par une conclusion générale.

# **Chapitre I**

## **Généralités sur les systèmes embarqués**

## I.1 Introduction

Les technologies des systèmes embarqués, logiciel embarqué et microélectronique, ont la capacité de transformer tous les objets du monde physique du plus petit au plus grand, du plus simple au plus complexe en objets numériques, intelligents, autonomes et communicants. L'émergence des systèmes embarqués, amplifie de façon considérable cette révolution. De fait, le déploiement généralisé des systèmes embarqués modifie profondément notre environnement, et porteur de très nombreuses innovations de produits et d'usages et impacte l'ensemble des activités industrielles et de services.

## I.2 Système embarqué

Un système embarqué est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans les mémoires. Un système embarqué est autonome et ne possède pas des entrées/sorties standards tels qu'un clavier ou écran d'ordinateur.

Les systèmes embarqués sont désormais utilisés dans des applications diverses tel que le transport (avionique, espace, automobile, ferroviaire) dans les appareils électriques et électroniques (appareils photos, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables, sécurités), dans la distribution d'énergie, dans l'automatisation,...etc. L'attrait principal des systèmes embarqués vient du fait qu'ils permettent d'implémenter à faible cout des fonctions complexes dont la réalisation était unimaginable il ya quelques années seulement. Aujourd'hui, la tendance générale est d'utiliser les systèmes embarqués pour incorporer des fonctionnalités complexes, précédemment considérées comme exotiques dans les produits de tous les jours. [1]

## I.3 Historique

L'historique des systèmes embarqués est intrinsèquement lié à celui du développement des processeurs dont on peut retenir les dates suivantes :

**[1967]** : <<Apollo Guidance computer>> l'un des premiers systèmes embarqué environ un millier de circuits intégrés identiques (portes NAND).

**[1970]** : Missile Minuteman, Il était le système de guidage informatisé Autonetics D-17, qui utilise des circuits logiques à base de transistor et disque dur comme la mémoire principale.

[1971] : le premier microprocesseur, était le premier circuit intégré incorporant tous les éléments d'un ordinateur dans un seul boîtier: unité de calcul, mémoire, contrôle des entrées /sorties.

[1972] : l'Intel 8008 fut le premier microprocesseur 8 bits à voir le jour. Deux fois plus puissant que le 4004, le 8008 était composé de 3300 transistors et fabriqué lui aussi en 10 000 nm, et sa vitesse atteignait 0,5 MHz (500 kHz), voire 0,8 MHz un peu plus tard.

[1974] : L'Intel 8080 est l'un des premiers microprocesseurs conçus et fabriqués par Intel. Présenté en avril 1974, ce processeur 8 bits qui fonctionnait à 2 MHz, est généralement considéré comme le premier microprocesseur véritablement utilisable dans une unité centrale.

[1978] : Le Zilog Z80 est un microprocesseur 8 bits conçu et fabriqué par Zilog. Une des particularités de ce processeur est le couplage de certains registres (2 fois 8 bits) et le bus d'adresses 16 bits, permettant un traitement de l'information nettement plus rapide qu'avec un processeur 8 bits classique.

[1979] : le Motorola 68000 est un microprocesseur CISC 16/32 bits développé par Motorola. C'est le premier de la famille de microprocesseurs souvent appelée m68k ou 680x0, qui comprend notamment les microprocesseurs Motorola 68010, Motorola 68020, Motorola 68030, Motorola 68040 et Motorola 68060. Le nom du 68000 vient à la fois de la continuité avec la famille de microprocesseurs Motorola 6800 et du nombre de transistors qu'il contient, un peu plus de 68 000.[2]

De 80 à aujourd'hui : avènement d'une industrie nouvelle, utilisant la puissance, la miniaturisation et la robustesse des puces électroniques pour rendre intelligents, communicants et sûrs tous les objets de notre quotidien. [3]

## **I.4 Caractéristiques spécifiques d'un système embarqué**

Les principales caractéristiques d'un système embarqué sont les suivantes :

- **Fonctionnement en temps réel :**
  - Les applications embarquées doivent répondre rapidement aux événements internes ou externes.
  - Nécessaire dans les applications de système de contrôles.

- Le résultat peut être néfaste si le système ne réagit pas à l'immédiat à un évènement du système

- **Environnement :**

Un système embarqué est soumis à de nombreuses contraintes d'environnement, Il doit s'adapter et fonctionner avec eux.

- Température, vibrations, chocs, variations d'alimentation, corrosion, eau, feu, radiation.
- **Coût :** Il est évident qu'un système aussi performant et moins couteux par rapport à un autre système sera considéré comme une solution plus attractive. Plusieurs aspects peuvent influencer le coût d'un système sur le marché : la surface du silicium, le rendement de fabrication, le packaging, l'effort de conception, le coût de la programmation, le temps de mise sur le marché, etc...
- **Capacité mémoire adaptée:** La mémoire est une ressource limitée dans un grand nombre de systèmes embarqués (de quelques Kilo-octets dans une carte à puce à quelques Méga-octets dans un téléphone portable), et par conséquent une bonne utilisation de la ressource mémoire est cruciale pour la gestion de ces systèmes.
- **Consommation énergétique :** Celle-ci est définie par la puissance consommée par le matériel au cours de l'exécution. Elle est liée à la surface occupée par l'application sur le circuit. Une consommation excessive augmente le prix de revient du système embarqué car il faut alors des batteries de forte capacité.
- **Fonctionnement critique pour la sécurité des personnes :**
  - Le système doit toujours fonctionner correctement.
  - Sûreté à faible coût avec une redondance minimale.
  - Sûreté de fonctionnement du logiciel.
  - ses communications sont confidentielles et authentifiées. [4]

La figure I.1 reprend ces différents aspects et propose une ou plusieurs mesures permettant de les évaluer. Ainsi, on peut mesurer la taille d'un code directement en octets, plutôt pour les petits systèmes embarqués de très bas niveau, ou en lignes de code pour ceux de haut niveau. Pour les performances de l'exécution du programme, on peut mesurer le nombre de millions d'instructions exécutées par seconde (MIPS) pour évaluer la vitesse de calcul « pure » ou le nombre d'opérations

de lecture/écriture par seconde dans la mémoire, si le système embarqué doit manipuler beaucoup d'entrées/sorties.

Critère	Mesure
Consommation énergétique	Watt
Taille du code	Octets Lignes de code
Exécution du programme	MIPS Nombre de read/write par seconde
Poids et taille	Kilogramme, centimètre
Cout de fabrication	€, \$

**Figure I.1 : Les critères d'efficacité d'un système embarqué.**

Il faut bien entendu optimiser tous ces critères. De plus, il est important de directement penser à ces aspects lors de la conception du système embarqué car il n'est pas forcément évident d'en modifier un existant après coup pour mieux les satisfaire. En effet, étant donné les contraintes du système embarqué, notamment en termes de puissance de calcul, il ne sera pas forcément possible de tous les optimiser. Un choix devra donc être posé selon l'application visée. [2]

## I.5 Domaines d'application d'un système embarqué

Les systèmes embarqués sont introduits dans divers domaines, à savoir :

- **Le domaine grand public :**  
Smart phone, console de jeux, appareil photos, lecteur audio,.....
- **Les moyens de transport :**  
Ordinateur de bord, GPS, système de navigation, automobiles, avions, trains, bateau.
- **Les équipements médicaux:**  
Imagerie (rayon X, ultra-sons, IRM) endoscopie, caméra, monitoring, perfusion, lasers, chirurgie, stimulateur cardiaque.
- **Les équipements de télécommunication :**  
Station mobile, routeur, Gateway, satellite
- **Les équipements industriels :**  
Productions automatisées, systèmes de commande d'énergie, équipements de stockage
- **Les équipements de bureautiques :**  
Répondeurs, copieurs, imprimante,....

➤ Les équipements de bâtiment :

Ascenseurs, système de surveillance, contrôle d'accès, systèmes d'éclairage...[3]

## I.6 Catégories des systèmes embarqués

Les systèmes embarqués peuvent être réparties en

- Calcul généraliste : similaire application bureau mais embarqué (assistant personnel, téléphone portable, etc.), consoles de jeux vidéos, set-top box
- Contrôle de systèmes : moteur, voiture, avion, processus chimique, nucléaire, navigation, etc.
- Traitement de signaux : compression vidéo, radar, flux de données, etc.
- Réseaux et communications : transmission données, commutation, routage, téléphone , internet ,etc. [4]



Figure II.2: Différentes catégories de systèmes embarqués.

## I.7 Architecture matérielle d'un système embarqué

Les systèmes embarqués utilisent généralement des microprocesseurs à basse consommation d'énergie ou des microcontrôleurs, dont la partie logicielle est en partie ou entièrement programmée dans le matériel, l'architecture matérielle liste les composants le constituant.

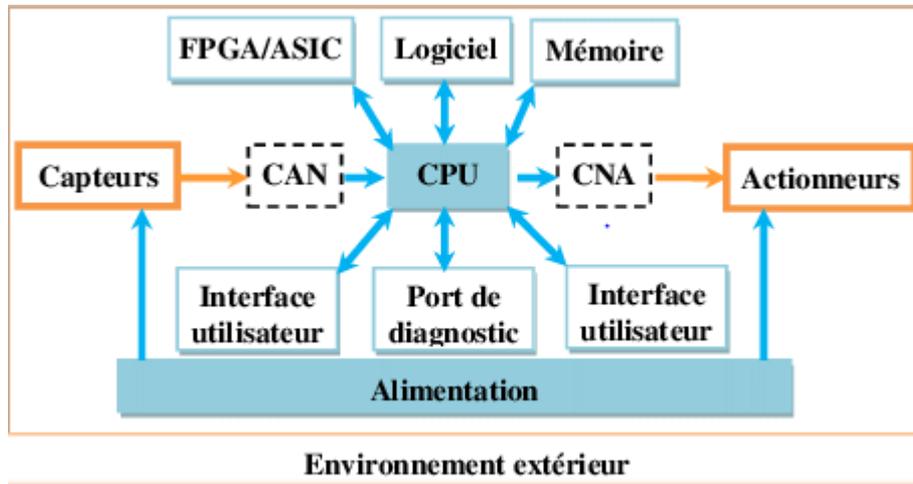


Figure III.3: Organisation d'un système embarqué.

Dans la figure (I.3) au-dessus :

Nous retrouvons en entrée des capteurs généralement analogiques couplés à des convertisseurs A/N<sup>1</sup>. On retrouve en sortie des actionneurs généralement analogiques couplés à des convertisseurs N/A. Au milieu, on trouve le calculateur mettant en œuvre un processeur embarqué et ses périphériques d'E/S. Il est à noter qu'il est complété parfois d'un circuit FPGA<sup>2</sup> jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur. L'architecture de base est la plupart du temps composée d'une unité centrale de traitement (CPU), d'un système d'exploitation qui réside parfois uniquement en un logiciel spécifique (ex: routeur). De même l'interface IHM<sup>3</sup> n'est pas souvent existante, mais est souvent utile pour reconfigurer le système ou vérifier son comportement. On retrouve en fait un système d'asservissement entre les entrées et les sorties, Il est à noter que l'expression la plus simple de cette figure est de considérer comme capteurs des interrupteurs et comme actionneurs des LED. Sur ce schéma théorique se greffe un paramètre important : le rôle de l'environnement extérieur. Un système embarqué doit faire face à des environnements plus hostiles. Il doit faire face à un ensemble de paramètres agressifs :

<sup>1</sup> A/N : Un convertisseur analogique-numérique est un montage électronique dont la fonction est de traduire une grandeur analogique en une valeur numérique

<sup>2</sup> FPGA : est un circuit intégré logique qui peut être reprogrammé après sa fabrication

<sup>3</sup> IHM : Interface Homme Machine Tout ce qui permet une communication homme/machine...

- Variations de la température. ·
- Vibrations, chocs. ·
- Variations des alimentations. ·
- Corrosion.
- Eau, feu, radiation

L'environnement dans lequel opère le système embarqué n'est pas contrôlé ou contrôlable. Cela suppose donc de prendre en compte ce paramètre lors de sa conception. On doit par exemple prendre en compte les évolutions des caractéristiques électriques des composants en fonction de la température, des radiations.... Les systèmes embarqués sont aujourd'hui fortement communicants. Cela est possible grâce aux puissances de calcul offertes par les processeurs pour l'embarqué (32 bits en particulier) et grâce aussi à l'explosion de l'usage de la connectivité Internet. La connectivité internet permet fondamentalement de contrôler à distance un système embarqué par Internet. Ce n'est en fait que l'aboutissement du contrôle à distance d'un système électronique par des liaisons de tout type : liaisons RS.232<sup>4</sup>, RS.485<sup>5</sup>, bus de terrain... Cela permet l'emploi des technologies modernes d'internet pour ce contrôle à distance par l'utilisateur : il suffit d'embarquer un serveur internet dans son équipement électronique pour pouvoir le contrôler ensuite à distance, de n'importe où, à l'aide d'un simple navigateur. Il n'y a plus d'interface humaine spécifique à concevoir pour cela, ce rôle étant rempli par le navigateur Internet. Cela est une réalité : les chauffagistes proposent maintenant des chaudières pouvant être pilotées par l'internet, Il faut aussi noter la montée en puissance des communications sans fil dans l'embarqué au détriment des communications filaires pour limiter le câblage et faciliter la mise en place du système embarqué. Le wifi et toutes les normes de réseaux sans fil IEEE 802.15<sup>6</sup> comme Zigbee<sup>7</sup> dans l'embarqué et surtout en domotique (réseaux de capteurs sans fil par exemple). Mais ne nous méprenons pas sur ces facilités et commodités, cela a bien sûr un revers : la sécurité du système embarqué, puisque connecté à Internet. [5]

Le fonctionnement d'un système se résume ainsi:

- Il reçoit des informations de l'environnement extérieur qu'il converti en signaux numériques.

---

<sup>4</sup> RS.232 : est une norme standardisant une voie de communication de type série.

<sup>5</sup> RS.485 : est une norme qui définit les caractéristiques électriques de la couche physique d'une interface numérique sérielle.

<sup>6</sup> IEEE 802.15 : est un protocole de communication défini par l'IEEE. Il est destiné aux réseaux sans fil

<sup>7</sup> Zigbee : est un protocole de haut niveau permettant la communication d'équipements personnels ou domestiques équipés de petits émetteurs radios à faible consommation

- L'unité de traitement composée du CPU, de la mémoire, du logiciel, de l'ASIC<sup>88</sup> et éventuellement d'un système externe traite l'information.
- Le traitement génère éventuellement une sortie qui est envoyée vers l'extérieure, les systèmes auxiliaire, les ports de monitoring ou l'IHM.

### I.7.1 Processeurs embarqués

Un processeur (CPU) est un composant présent dans de nombreux dispositifs électroniques qui exécute les instructions machine des programmes informatiques, c'est lui qui gère et contrôle la totalité des procédures et des enchaînements, exécute les instructions des programmes. Sans lui, un fonctionnement opérationnel et interactif des autres composants n'est pas possible, il est relié directement ou indirectement à tous les autres composants de la carte mère. [6]

Le processeur traite les programmes en chargeant les instructions depuis la mémoire, en les décodant et en les exécutants les unes après les autres. Les instructions d'un processeur peuvent être grossièrement classées en trois catégories : les instructions de calcul arithmétique et logique, les instructions d'accès mémoire (lecture ou écriture d'une information en mémoire) et les instructions de saut (conditionnel ou inconditionnel) qui permettent d'effectuer les tests et les boucles indispensables à tout algorithme.

Dans la conception d'un système embarqué, le choix du microprocesseur est un point sensible. En effet, il faut trouver un compromis entre la puissance de calcul nécessaire à l'application et les contraintes comme le coût de fabrication ou la consommation d'énergie. Les applications les plus légères se contentent d'un microcontrôleur, c'est à dire d'une puce qui intègre dans un même boîtier le processeur, la mémoire et les interfaces d'entrées-sorties. Cette solution permet, pour un prix modeste, l'utilisation de l'informatique embarquée dans des équipements où les contraintes d'encombrement sont importantes, mais reste très limitée en puissance de calcul. A l'inverse, des systèmes plus complexes comme ceux utilisés dans les télécommunications utilisent typiquement plusieurs processeurs dont certains sont dédiés au traitement du signal. L'architecture obtenue est alors beaucoup plus complexe, mais permet de rendre des services plus avancés. [7]

---

<sup>8</sup> l'ASIC : est un circuit intégré (micro-électronique) spécialisé.

## I.7.2 Mémoires

De même que les ordinateurs, les systèmes embarqués utilisent plusieurs types de mémoire : mémoire de travail, mémoire de masse, et mémoire morte.

### I.7.2. 1 La mémoire vive RAM

La RAM (Random Access Memory) est indispensable en tant que mémoire de travail, c'est pourquoi elle est présente dans toutes les formes de systèmes embarqués. Certaines applications comme le traitement de flux multimédia (encodage ou décodage audio/vidéo) en exigent même une quantité substantielle. Cependant son coût, son encombrement important, sa volatilité, synonyme de consommation d'énergie, en limitent fortement l'usage sur les plates-formes les plus contraintes.

### I.7.2.2 La mémoire morte ROM

La ROM (Read-Only Memory) au contraire, est persistante, et est économique tant en prix de revient qu'en encombrement. Elle est donc couramment employée dans les systèmes embarqués, typiquement pour y stocker le logiciel dans les cas où celui-ci n'est pas destiné à changer pendant toute la vie du système embarqué. Les systèmes qui nécessitent des données modifiables et persistantes doivent donc avoir recours à une forme de mémoire de masse : un disque dur, ou une mémoire non-volatile comme l'EEPROM (Electrically-Erasable Programmable Read-Only Memory) ou la mémoire Flash. Ces mémoires ne sont pourtant en général pas utilisées comme mémoire de travail, car leurs performances sont insuffisantes, surtout en écriture. De plus, ces mémoires ont typiquement une durée de vie assez limitée, de l'ordre de quelques milliers d'écritures.

La décision de stocker des données dans l'une ou l'autre mémoire relève donc encore une fois d'un compromis entre les performances (temps d'accès, persistance) et le coût (coût de fabrication, encombrement, consommation énergétique). Cependant, d'un genre de système embarqué à un autre, l'espace disponible peut varier considérablement : si sur un téléphone portable, on peut raisonnablement s'attendre à trouver plusieurs méga-octets, voire dizaines de méga-octets de mémoire vive, à l'inverse, une carte à puce ne disposera que de quelques kilo-octets de RAM. Lorsque l'architecture le permet, l'utilisation de mémoire virtuelle [CV65] permet de simplifier ce problème. Le principe est de considérer toutes les mémoires comme un espace uniforme du point de vue du processeur, et de confier à un coprocesseur spécifique appelé unité de gestion mémoire (MMU, ou Memory Management Unit) la tâche de traduire les adresses

virtuelles en adresses réelles. La mémoire virtuelle est en général utilisée conjointement avec un mécanisme de pagination, pour permettre au système de déplacer physiquement certaines pages d'une zone de mémoire à une autre. L'idée de ce va-et-vient (swapping) est de déplacer dynamiquement les pages les plus utilisées vers la mémoire la plus rapide, et de décharger les pages les moins utilisées vers une mémoire plus abondante mais plus lente. La présence de l'unité de gestion mémoire permet de faire cette manipulation d'une façon transparente pour le programme, puisque les adresses virtuelles des pages déplacées ne sont pas modifiées. Dans la plupart des systèmes embarqués, cependant, l'architecture ne comporte pas d'unité de gestion mémoire, et l'utilisation des différents types de mémoire doit être programmée spécifiquement. [8]

### **I.7.3 Entrées-sorties**

Un système embarqué n'est pas obligatoirement en contact direct avec son utilisateur. De nombreux systèmes embarqués travaillent en autonomie complète dans leur environnement, on parle alors parfois de systèmes enfouis. La présence d'une interface de communication est pourtant nécessaire, pour que le système puisse remplir sa fonction. Suivant les situations, il doit être capable de communiquer avec des dispositifs externes, comme des senseurs, boutons et circuits transducteurs, à travers des ports d'entrée. Ces différents dispositifs sont identifiés par des ports possédant une adresse. Dans l'autre sens, la communication d'information vers le monde extérieur se fait via des ports de sortie. Des exemples de sorties : les LEDs, écrans LCD, etc.

Il peut s'agir d'une liaison réseau, d'une connexion directe à l'équipement contrôlé, ou d'une interface spécifique, comme le connecteur des cartes à puce. Par exemple, un contrôleur de portail automatique disposera comme interface d'entrée d'un récepteur radio associé avec la télécommande, et comme interface de sortie d'une connexion directe avec les pré-actionneurs du moteur.[9]

## **I.8 Architecture logicielle d'un système embarqué**

La plupart des matériels embarqués nécessitent une gestion logicielles. Le logiciel qui interagit directement avec ce matériel.

### **I.8.1 Systèmes d'exploitation embarqué**

Un système d'exploitation est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications (traitement de texte, jeu vidéo, ...). Ainsi lorsqu'un programme désire accéder à une ressource matérielle, il ne lui est pas nécessaire d'envoyer des informations

spécifiques au périphérique, il lui suffit d'envoyer les informations au système d'exploitation, qui se charge de les transmettre au périphérique concerné via son pilote. En l'absence de pilotes il faudrait que chaque programme reconnaisse et prenne en compte la communication avec chaque type de périphérique.[10]

Un système d'exploitation embarqué est un système d'exploitation pouvant être installé sur un système embarqué. Ce système d'exploitation est conçu avec des spécificités à gérer afin de répondre à des besoins spécifiques au type de système embarqué.[11]

Quelques types de systèmes d'exploitation embarqués :

- **Windows CE:** est une variation de Windows pour les systèmes embarqués et autres systèmes minimalistes, utilisée notamment dans les PC de poche . Il utilise un noyau distinct des autres Windows plutôt qu'une version allégée et supporte les architectures processeur Intel x86 et similaires, MIPS (jusqu'à CE 3.0), ARM .
- **Firefox OS :**est un système d'exploitation mobile libre et destiné aux objets connectés proposé et développé par la Mozilla Corporation.
- **Windows phone :** c'est un système d'exploitation pour smart phones que Microsoft a publié en octobre 2010. Il est venu remplacer Windows Mobile en introduisant une interface utilisateur totalement redessinée et pensée pour les terminaux à écran tactile. Celle-ci se compose de tuiles dynamiques dont l'affichage évolue en temps réel selon l'activité de l'application concernée.
- **Palm OS :** développé initialement par U.S Robotics Corporation Il est fourni avec une suite d'applications de base pour gestionnaire d'informations personnelles. Plus tard les versions de cet OS ont été améliorées pour fonctionner sur des Smartphones.
- **Android :** est le système d'exploitation mobile de Google open-source qui équipe la majorité des Smartphones et tablettes du marché , Android était une startup dirigée par Andy Rubin et qui a été racheté par Google en 2005. Aujourd'hui, Android est le système d'exploitation mobile numéro un dans le monde.
- **eCos :** est un système d'exploitation temps réel. Il permet de réaliser des applications temps réel. Il est implémenté en C/C++ .[12]

## I.8.2 Programmation

Un système embarqué doté d'un processeur a besoin de code pour faire quelque chose. Ce dernier est placé dans une mémoire accessible par le processeur afin de pouvoir être exécuté, instruction par instruction. Typiquement, le software d'un système embarqué sera présenté sous la forme d'une image ROM, définitive et non modifiable.

Étant donné un système embarqué, on connaît la taille de la ROM qu'il possède et donc les adresses de tous les emplacements mémoire disponibles. Pour chacun de ces emplacements, il faut en définir le contenu binaire représentant des instructions à exécuter. Ces instructions sont écrites en respectant le jeu d'instructions du processeur utilisé, ce qui peut évidemment prendre beaucoup de temps à comprendre avant de commencer à programmer.

Pour éviter de mémoriser les codes machine de toutes les instructions d'un processeur donné, on peut programmer à l'aide d'un langage d'assemblage, représentation textuelle de plus haut niveau du code machine. Un programme écrit dans un langage d'assemblage est traduit en code machine par l'assembleur. Programmer à un tel niveau reste une tâche fortement consommatrice de temps, mais peut être plus efficace si l'on s'aide de bibliothèques de code préexistant.

Enfin, pour gagner du temps de développement, il est possible d'utiliser un *langage de haut niveau* qui offre des constructions et abstractions plus pratiques et proches de la façon de penser du développeur et du problème qu'il veut résoudre. Quelques exemples d'un tel langage, adapté aux systèmes embarqués :

- **le langage C :** Un programme écrit dans un langage de haut niveau est compilé en code machine par le compilateur, permet de développer des programmes concis et rapides.
- **Le langage c++ :** C'est une évolution orienté objet du langage C , Il constitue la base de plusieurs langages de programmation, le principal choix de programmation pour la plupart des applications de bureau les plus utilisées. un choix incontournable pour les pilotes de périphériques, les moteurs de jeux, les outils de traitement audio / image, les logiciels intégrés, etc.
- **Java :** Java est également utilisé dans de nombreux systèmes embarqués mais les programmes Java nécessitent une machine virtuelle Java (JVM) qui consomme beaucoup de ressources. C'est pourquoi ce langage n'est pas utilisé pour les appareils embarqués qui ne disposent que de ressources limitées.

- **Langage Ada** : langage de plus haut niveau que le C, mais conçu pour ce type de développement (permet de s'abstraire de l'OS)
- **Python** : Bien que n'étant pas traditionnellement associé aux systèmes embarqués, Python commence à être pris plus au sérieux dans les applications de systèmes embarqués, Python n'est peut-être pas aussi utile pour la programmation embarquée que C ou C ++, mais avec de nombreuses bibliothèques disponibles, il est facile d'implémenter des fonctionnalités qui le rendent tout aussi utile. Il est excellent pour automatiser les tests, collecter et analyser les données

On pourra également employer d'autres langages de programmation ou langages de scripts célèbres dans le monde Unix, perl ...

## **I.9 Conclusion**

On a vu dans ce chapitre un aperçu du monde des systèmes embarqués en générale, l'architecture matérielle et logicielle, ses catégories, ses domaines d'utilisation, les contraintes qu'il faut prendre en considération avant de concevoir un système embarqué.

Dans le chapitre suivant nous allons présenter la carte Raspberry Pi et le système de contrôle d'accès.

# **Chapitre II**

## **Carte Raspberry pi et systèmes de contrôle d'accès**

## II.2 Introduction

Le terme "contrôle d'accès" décrit toute technique utilisée pour contrôler le passage dans ou hors d'une zone. La serrure standard utilisant une clé en laiton peut être considérée comme une forme simple de "système de contrôle d'accès».

Les systèmes de contrôle d'accès sont le plus souvent utilisés pour contrôler l'entrée dans les extérieures des bâtiments. Ces systèmes peuvent également être utilisés pour contrôler l'accès à certaines zones situées à l'intérieur des bâtiments.

Comme notre travail consiste à concevoir un système de contrôle d'accès basé sur la carte Raspberry, ce chapitre a pour but de présenter la carte raspberry pi comme un module de notre projet, ses différents modèles, ses composants, ses fonctionnalités ainsi on décrira le système contrôle d'accès.

## II.2 Carte raspberry

Raspberry est une carte mère d'un nano-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser la Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et d'autres logiciels compatibles. [14]

## II.3 L'histoire de Raspberry Pi

Le Pi est un projet en évolution, envisagé pour la première fois en 2006 par Eben Upton et d'autres personnes travaillant au Laboratoire d'informatique de l'Université de Cambridge. Jack Lang, Rob Mullins et Alan Mycroft étaient également impliqués dans le projet. Ils craignaient qu'il y ait une disparité dans le niveau de compétence de nombreux étudiants en informatique potentiels. Les gens qui venaient au programme d'informatique à cette époque avaient généralement très peu d'expérience, ce qui était différent de tous les amateurs qui étaient venus à l'école par le passé. Les gens n'avaient tout simplement pas les compétences de programmation qu'ils avaient auparavant au niveau collégial.

Au fur et à mesure que le prix des composants diminuait, l'équipe a découvert qu'il serait possible de créer de petits ordinateurs qui seraient un moyen simple et rentable pour un plus grand

nombre d'enfants d'intégrer la programmation informatique avant d'aller à l'école. C'est aussi un moyen moins coûteux pour ceux qui se trouvent dans les pays en développement d'apprendre.

Ils ne sont ni en développement ni en production et les premières unités ont été lancées sur le marché il y a peu de temps en 2012. Jusqu'à présent, la réception sur le Raspberry Pi a été assez bonne. Ils sont maintenant au point où ils essaient de développer environ 30 000 unités par mois. Les développeurs commencent enfin à réaliser leur rêve de fournir des ordinateurs abordables, utilisables et programmables pour tous ceux qui le souhaitent. [15]

## II.4 Les modèles de Raspberry Pi

Aujourd'hui, la Raspberry Pi en est à sa troisième génération. Nous allons voir en quelques lignes les différences entre chaque.

### II.4.1 Raspberry Pi 1

Cette première génération est arrivé en 2012 avec deux versions, les versions A et B puis l'année suivante les versions A+ et B+ , Les différences entre ces modèles sont la taille du GPIO, le nombre de ports USB et la présence ou non d'Ethernet, le support de carte microSD et la taille de la RAM. [16]

#### II.4.1.1 Modèle A

##### Modèle A

Processeur	ARM1176JZF-S (ARMv6) 700 MHz Broadcom 2835
RAM	256 Mo
Sorties Vidéo	Composite et HDMI
Sortie Audio	Sortie Jack 3.5
USB	1 Port USB 2.0
Ethernet	N / A



Figure II.1:Modèle A

### II.4.1.2 Modèle B

#### Modèle B

Processeur	ARM1176JZF-S (ARMv6) 700 MHz Broadcom 2835
RAM	512 Mo
Sorties Vidéo	Composite et HDMI
Sortie Audio	Sortie Jack 3.5
USB	Port USB 2.0
Ethernet	1 Port Fast Ethernet (10/100 Mbit/s)



Figure I.2: Modèle B

### II.4.1.3 Différence du modèle A+ avec le A

La Raspberry Pi A+ résulte de la version améliorée de la Raspberry Pi A :

- Plus petit : 65 mm de long (contre 86 mm)
- Lecteur de carte micro SD à la place du lecteur de carte SD
- GPIO 40 broches au lieu de 26 broches .
- Nouveau chipset audio
- Consommation électrique moindre



Figure II.3:Modèle A+

#### II.4.1.4 Différence modèle du B+ avec le B

le modèle B+ est une mise à niveau matérielle du B :

- GPIO 40 broches au lieu de 26 broches .
- 4 ports USB 2.0 au lieu de 2
- Réduction de consommation de 3,5 W à 3 W
- Nouveau chipset audio



Figure II.4:Modèle B+

### II.4.2 Raspberry Pi 2

Pour la version 2, il n'y a qu'un seul modèle, le B. Ce modèle est sortie en Février 2015, il est équipé du processeur Broadcom BCM2836 avec quatre cœurs ARMv7 à 900 MHz et 1 Go de RAM. [16]



Figure II.5:Modèle 2B

## II.4.3 Raspberry Pi 3

La raspberry de la troisième génération présente une puissance de traitement encore plus grande ainsi qu'une connectivité sans fils embarqué et le Bluetooth .

### II.4.3.1 Raspberry Pi 3 modèle 3B

Elle repose sur un processeur à quadruple cœur ARM Cortex-A53, le BCM2837 de Broadcom, cadencé à 1,2 GHz. Cela signifie qu'il est 50 à 60% plus rapide que le Raspberry Pi 2 B. La communication par Wi-Fi 802.11n et du Bluetooth 4.1 est maintenant intégrée au RPi 3 ; ce nouveau modèle est toujours rétro compatible avec les modèles précédents. [17]

Les principales caractéristiques de cette carte sont :

- Processeur BCM2837 de Broadcom, cadencé à 1,2 GHz
- Quadruple cœur ARM Cortex-A53 à 64 bits
- Wi-Fi 802.11 b/g/n intégré
- Bluetooth 4.1 (Classic & Low Energy) intégré
- Coprocesseur multimédia à double cœur Videocore IV®
- Mémoire LPDDR2 de 1 Go
- Supporte les distributions GNU/Linux ARM les plus récentes et Windows 10 IoT
- Connecteur micro-USB pour alimentation 2,5 A
- 1 port Ethernet 10/100
- 1 connecteur audio/vidéo HDMI
- 1 connecteur audio/vidéo RCA
- 4 port USB 2.0
- 40 broches d'E/S à usage général
- Antenne sous forme de puce
- Connecteur DSI pour écran
- Lecteur de carte micro-SD

- Dimensions : 85 x 56 x 17 mm

### II.4.3.2 Raspberry Pi 3 modèle 3B+

Le Raspberry Pi 3 Model B + est le dernier produit de la gamme Raspberry Pi 3, avec un processeur quad core 64 bits cadencé à 1,4 GHz, un WiFi bibande 2,4 GHz et 5 GHz, du Bluetooth 4.2 / BLE, un Ethernet plus rapide.

Le Raspberry Pi 3 Model B + conserve la même empreinte mécanique que le Raspberry Pi 2 Model B et le Raspberry Pi 3 Model B. [16]



Figure II.6:Modèle 3B

### II.4.4 Raspberry pi zero

Cette version petit format diffère de la version classique des cartes Raspberry Pi (Pi2, Pi3, Pi3B+, etc...) en adoptant un port mini-HDMI nécessitant un cordon adapté ainsi que deux ports micro-USB.

#### II.4.4.1 Raspberry Pi Modèle Zéro

Cette version est sortie en Novembre 2015 .La Raspberry Pi Zero possède les mêmes spécifications des modèles A/B mais avec un processeur cadencé à 1 GHz au lieu de 700 MHz auparavant. Comme il est plus petit il y a moins de connectique. [17]

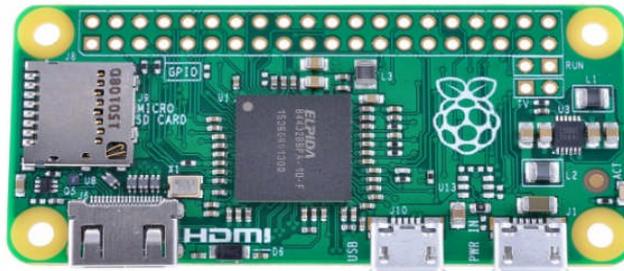


Figure III.7:Modèle 0

#### II.4.4.2 Raspberry Pi Zero W

Il s'agit, comme son nom l'indique, d'un Raspberry Pi Zero, lancé en novembre 2015, mais avec des fonctions sans-fil : du 802.11b/g/n et du Bluetooth 4.1.

Les caractéristiques sont strictement identiques à celles du Pi Zero, avec un processeur monocoeur Broadcom BCM 2835 à 1 GHz, 512 Mo de mémoire vive, un port mini HDMI et un port micro USB (un second port est présent pour l'alimentation). Un lecteur de carte micro SD est également de la partie, ainsi qu'un connecteur mini CSI pour brancher une caméra.[18]



Figure II.8: Modèle zero w

#### II.4.4.3 Raspberry Pi Zero WH

LE Raspberry Pi Zero WH est une version miniature de la carte Raspberry Pi classique. Ce modèle WH est basé sur un processeur ARM à 1 GHz accompagné de 512 Mo de RAM. Il intègre un port GPIO déjà soudé à la carte mère et dispose d'interfaces WiFi et Bluetooth. [19]



Figure II.9: Modèle zéro WH

## II.5 Systèmes d'exploitation pour la Raspberry pi

Les Raspberry Pi sont livrés par défaut vierges de tout système d'exploitation et de tout dispositif de stockage. C'est à l'utilisateur de choisir et d'installer le système d'exploitation qu'il veut (la plus grande majorité sont des systèmes Linux) et de choisir la carte de stockage de type micro-SD avec la capacité qu'il veut (16 Go minimum conseillés).

Il y a beaucoup de différents systèmes d'exploitation disponibles pour la Raspberry pi. Presque tous les systèmes d'exploitation qui fonctionnent sur un processeur ARM peuvent normalement être installé sur la Raspberry pi, nous présentons quelques-uns des plus populaires systèmes d'exploitation personnalisés pour le Raspberry Pi.

### II.5.1 Debian (Raspbian) :

Raspbian, est un système d'exploitation libre et gratuit basé sur Debian GNU/Linux optimisé pour fonctionner sur un Raspberry Pi . Il représente un ensemble de programmes et d'utilitaires s'assure que votre appareil fonctionne correctement. Raspbian n'est pas qu'un système d'exploitation, il vient également avec plus de 35 000 paquets dans un format adapté pour une installation rapide. Raspbian s'améliore chaque jour et est toujours en développement. Ce système est livré avec un environnement de bureau conçu pour ressembler aux systèmes d'exploitation que nous utilisons tous les jours. Utilisation de l'environnement de bureau est parfait pour les utilisateurs qui ne maîtrisent pas bien la console Linux. [20]

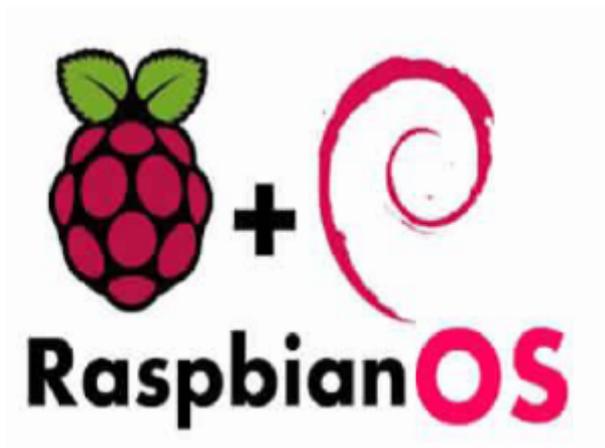


Figure II.10: Logo du système d'exploitation Raspbian

### II.5.2 Arch Linux ARM :

Arch Linux Arm est une distribution du Arch Linux spécialement conçue pour fonctionner sur des périphériques utilisant un processeur Arm tel que le Raspberry Pi qui utilise le processeur ARMv6, C'est un système d'exploitation d'usage général , particulièrement léger et simple.

Arch Linux n'est pas destiné aux utilisateurs qui n'ont pas utilisé de système d'exploitation Linux auparavant, car l'installation initiale installe simplement les applications minimales pour faire fonctionner le système et n'a pas de bureau, juste le shell .Ce dernier est un environnement puissant dans lequel vous tapez des commandes pour contrôler l'ordinateur, aucune souris requise. [21]



Figure II.10: Logo du système d'exploitation ArchLinux ARM

### II.5.3 Risc OS :

RISC OS est un système d'exploitation britannique conçu à l'origine par Acorn Computers Ltd à Cambridge, en Angleterre, et a été publié en 1987. Il a été spécialement conçu pour fonctionner le sur chipset ARM. C'est rapide, compact et efficace. RISC OS n'est pas une version de Linux, ni n'est en aucune façon liée à Windows.

ARM .RISC OS Pi est livré avec un petit ensemble d'utilitaires et applications, il comprend un navigateur appelé NetSurf, un simple éditeur de texte, une calculatrice scientifique, et il a aussi deux logiciels / gestionnaires de paquets. Bien que ce n'est pas un système d'exploitation moderne (comparé à Linux, Windows et OSX). [22]



Figure II.11: Logo du système d'exploitation RISC OS

#### II.5.4 OpenELEC et Raspbmc :

OpenELEC est une distribution dédiée à la transformation d'une machine en media center. Elle se base pour ça sur du noyau Linux, et l'excellent XBMC <sup>1</sup>(maintenant KODI).

Le principe est d'optimiser au mieux les performances du matériel sur lequel Openlec est installé, OpenELEC et Raspbm Vc sont deux systèmes qui permettent d'avoir un vrai média center sur la Raspberry pi. On les utilise pour la lecture de musique et de vidéo ou encore pour regarder des photos via Airplay. [23]



Figure II.11: Logo du openelec.

<sup>1</sup> XBMC signifiait à l'origine Xbox Media Center est un lecteur multimédia libre.

## II.6 Les différentes utilisations du Raspberry Pi

Il est bien évidemment impossible de dresser une liste exhaustive des projets réalisables avec la raspberry. Cependant, il est possible de distinguer un certain nombre d'usages globaux de celle-ci.

### II.6.1 Un ordinateur, tout simplement

La Raspberry Pi est un ordinateur Petit, bon marché, et souvent suffisant pour bien des utilisations. C'est sa raison d'exister. Il fournit une alternative aux personnes n'ayant pas les moyens d'acheter un ordinateur standard. Il permet aussi d'encourager l'apprentissage de la programmation.

Avec le Pi lui-même, la carte microSD et l'alimentation, on aura besoin d'un câble HDMI<sup>2</sup> et d'un affichage approprié. Comme avec un ordinateur traditionnel, on aura également besoin d'un clavier et d'une souris USB.

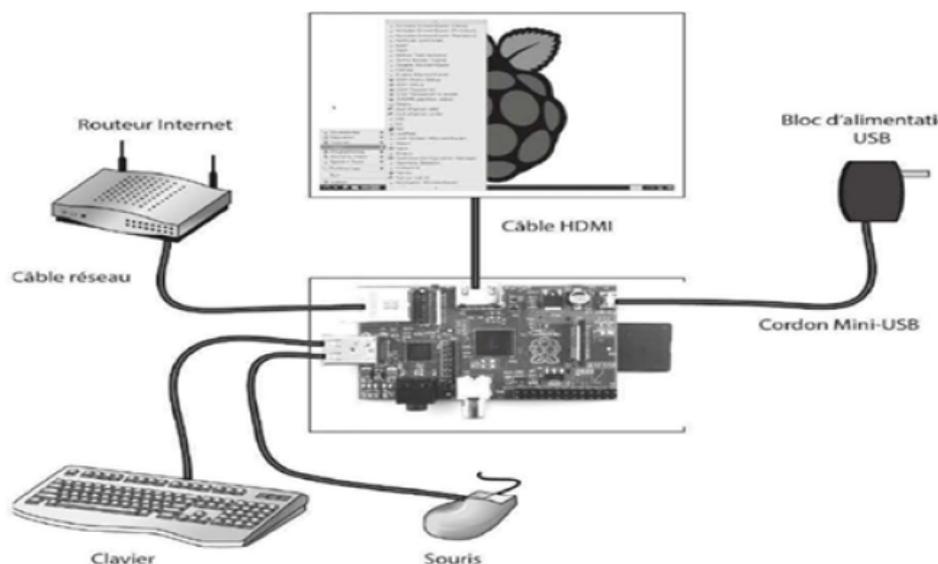


Figure II.12: Raspberry pi comme un pc du bureau ..

---

<sup>2</sup> HDMI : High Definition Multimedia Interface, est une norme et interface audio/vidéo totalement numérique pour transmettre des flux chiffrés constitués de données vidéo non compressées et des données audio pouvant être compressées.

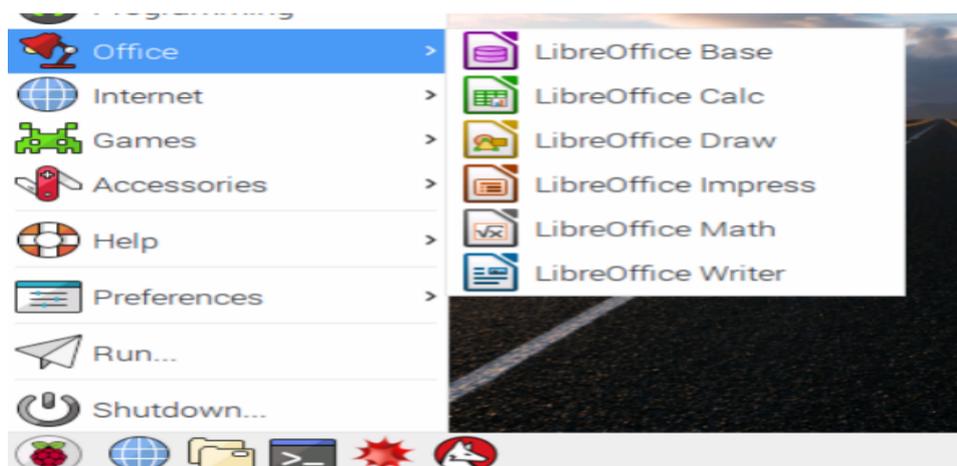


Figure II.13: le bureau de Raspbian

## II.6.2 Les systèmes embarqués

La raspberry étant particulièrement petite, et possédant des entrées GPIO, elle est souvent utilisée dans le cadre de projets liés à l'électronique, en tant que contrôleur central d'un système informatique, etc. De nombreux projets émergent, comme le contrôle d'une voiture télécommandée avec caméra, des drones, ou même un amusant projet sur le pilotage de drone.

## II.6.3 La domotique

Le problème avec la domotique c'est qu'il n'existe pas beaucoup d'interfaces complètes, et elles sont assez chères, mais le Raspberry Pi est présent pour résoudre ce problème. En effet, comme il est de taille réduite, il peut facilement s'intégrer dans une pièce. De plus, avec ses nombreuses entrées, notamment GPIO, il peut accueillir de nombreux modules pour communiquer avec des équipements domotiques.

## II.6.4 L'utilisation MultiMedia

Il s'agit là d'un des usages les plus développés du Raspberry Pi. Sous l'idée de MultiMedia pointe en fait surtout celle de média-center.

## II.6.5 Les serveurs

Dans le cadre d'un serveur, le Raspberry sera branché et fonctionne en permanence sans la moindre interruption. Il restera probablement proche du box internet, et sera contrôlé à distance via SSH, le moins souvent possible, le fait d'avoir besoin de prendre le contrôle d'un serveur étant en général dû au problème de celui-ci.

### II.6.5.1 Le serveur web :

Il s'agit d'un serveur visant à héberger un ou plusieurs sites web, qui devront être accessibles depuis l'extérieur.

### II.6.5.2 Le serveur de stockage :

Il s'agit d'un serveur qui permet de stocker vos fichiers pour y avoir accès depuis n'importe où. Il s'agira en général d'un serveur dit « FTP », c'est-à-dire une machine exécutant un logiciel permettant de traiter le protocole « File Transfert Protocole », qui est un protocole dédié au transfert de fichiers.

### II.6.5.3 Le serveur de service :

Il s'agit d'un serveur un peu à part, qui vise à héberger un service, en général pour pouvoir y accéder depuis n'importe où. On entend par « service » un logiciel, mais qui ne sera pas un client, c'est-à-dire un logiciel tournant au niveau de la machine de l'utilisateur, et permettant de faire des appels vers une entité centrale. Il va justement s'agir de cette entité centrale, responsable des opérations, des calculs, de la coordination des différents clients, etc. [24]

Dans ce qui se suit on présente deux exemples détaillés des différents mini-projets utilisant la raspberry pi.

#### ➤ station mobile pour la météo et mesure de la qualité de l'air « AirPi »

La pollution de l'air peut rapidement mener à divers problèmes de santé. De plus, il est souvent très difficile d'obtenir une valeur fiable de la qualité de l'air dans son propre environnement. Une solution est d'utiliser « l'AirPi » : le pack de Raspberry Pi et divers capteurs permettent de mesurer des valeurs telles que la température, l'humidité, la pression atmosphérique, le niveau d'UV, le monoxyde de carbone ou la teneur en dioxyde d'azote. Outre les informations sur la qualité de l'air, l'application Raspberry Pi fournit également des informations sur la météo. Grâce à la capacité Internet du nano-ordinateur, les valeurs mesurées peuvent être accessibles automatiquement via une interface Web.

#### ➤ Un système de surveillance

La vidéosurveillance est importante en ce qui concerne la sécurité de nos jours. Espaces commerciaux, écoles et les hôpitaux, les entrepôts et autres environnements intérieurs et extérieurs difficiles nécessitent des caméras haut de gamme. Les technologies actuelles nécessitent des RFID qui sont coûteuses et, par conséquent, le domaine de la sécurité devient coûteux.

L'utilisation de l'ordinateur à carte unique et à faible coût Raspberry Pi qui suit l'algorithme de détection de visage permet en quelques minutes de mettre en place un système de vidéo-surveillance avec de la détection de mouvements, ainsi que l'affichage d'une carte des différents moments d'activité et une sauvegarde de la surveillance sur le Raspberry Pi.

C'est un système qui capture réel images de temps et les affiche sur le navigateur en utilisant TCP / IP et qui permet aussi la diffusion vidéo en direct avec détection du visage humain.

## II.7 Système de contrôle d'accès

Un système de contrôle d'accès détermine qui est autorisé à entrer ou sortir, où ils sont autorisés à entrer ou à sortir, et quand ils sont autorisés à entrer ou à sortir. Historiquement, cela a été partiellement réalisé au moyen de clés et de serrures. Quand une porte est verrouillée, uniquement celui qui possède une clé peut entrer par la porte. Les serrures mécaniques à clés ne permettent pas des restrictions sur les moments et les dates d'accès. Elles ne fournissent aucun autre moyen de contrôle en ce qui concerne la copie des clés ou sur les personnes qui les ont utilisées. Quand une clé mécanique est perdue ou le détenteur de la clef n'est plus autorisé à utiliser la zone protégée, les verrous doivent être retapés. Le contrôle d'accès électronique utilise des systèmes plus intelligents pour éviter ce type de défauts en offrant un large éventail de pouvoirs qui peut être utilisé pour remplacer des touches mécaniques. Le contrôle électronique accorde l'accès en se basant sur les informations d'identification présentées. Lorsque l'accès est accordé, la porte est déverrouillée pendant une durée prédéterminée et la transaction est comptabilisée. Lorsque l'accès est refusé, la porte reste verrouillée et la tentative d'accès est enregistrée. Le système peut également surveiller la porte et déclenche une alarme si la porte est forcée ou maintenue ouverte trop longtemps après avoir été déverrouillée. [25]

## II.8 Objectifs de contrôle d'accès

Un système de contrôle d'accès n'accorde l'accès régulier qu'aux personnes autorisées. Il régit qui a accès, quand, où et éventuellement, avec qui.

Basées sur des mesures électroniques, organisationnelles et architecturales, les systèmes modernes de contrôle d'accès protègent des bâtiments, les équipements et empêchent le vol de propriété intellectuelle.

Le défi particulier du contrôle d'accès tient au fait qu'il faut limiter aussi peu que possible la liberté de mouvement des personnes autorisées. Le contrôle d'accès possède également un rôle

préventif important par l'effet de dissuasion .Il n'empêche pas les attaques par les personnes autorisées mais permet néanmoins leur identification en cas d'incident.

## II.9 Composants de base des systèmes de contrôle d'accès

Les systèmes de contrôle d'accès varient considérablement en termes de types de niveaux de complexité.

### II.9.1 Face à l'utilisateur :

La partie la plus connue des systèmes de contrôle d'accès est les cartes, les badges d'identification et, plus récemment, les applications pour Smartphones qui ont un système d'authentification présentée à la porte pour la déverrouiller. Celles-ci sont également appelées informations d'identification car elles contiennent les données de l'utilisateur qui indiquent au lecteur de vous autoriser à être sur le lieu, ou en d'autres termes, que vous êtes un participant autorisé.

#### II.9.1.1 support

Le lecteur transmet à la centrale les caractéristiques d'identification relatives à un utilisateur, l'identification d'un utilisateur nécessite la possession d'une carte ou d'un tag, la connaissance d'un code, des caractéristiques personnelles (biométrie), ou une combinaison de plusieurs de ces éléments, parmi les supports les plus fréquemment utilisés:

**Carte :** est une carte en plastique spéciale équipé d'un circuit intégré (puce) contenant une logique matérielle , une mémoire ou un microprocesseur .Les cartes à puce sont contrôlées par des lecteurs de cartes spéciaux.

**Clé :** Dans les solutions mécatronique ou il y a utilisation parallèle de lecteurs sans contact et de cylindres mécaniques.

**Porte-clés / bracelet :** dans cette variante, un composant RFID est intégré dans un porte-clés ou un bracelet. La distance de lecture est nettement inférieure à celle de la carte à puce.

#### II.9.1.2 Lecteurs

Le lecteur de carte est monté sur le mur à côté de la porte. Il lit les données sur le justificatif d'identité et envoie une requête au serveur pour déverrouiller la porte, les lecteurs se déclinent en une foule des modèles ou de conception avec plusieurs critères :

- Avec /sans clavier
- Technologie de lecteur (mifare, Hitag....)
- Distance de lecture
- Protection contre le vandalisme
- Classe de protection (utilisation intérieure /exterieure)
- Avec /sans écran
- Intégration dans la serrure, dans la gâche

### II.9.2 Face à l'administrateur :

Le coté administrateur est le tableau de bord de gestion, ou portail, ou l'administrateur de bureau, le responsable de sécurité informatique définit les paramètres des personnes autorisées à accéder aux locaux et dans quelles circonstances. Cela implique un tableau de bord de gestion, et un moyen de fournir un accès, tel qu'un périphérique de programmation.

Dans les systèmes plus avancés, l'aspect des opérations manuelles peut être automatisé. Par exemple, le provisioning (création et suppression d'accès) peut être effectué automatiquement en connectant le tableau de bord d'accès à l'annuaire des employés d'entreprise. Lorsqu'une nouvelle embauche apparaît dans le système, un nouvel accès est automatiquement positionné via une API<sup>3</sup> ou un service d'intégration de base de données.

### II.9.3 Infrastructure :

Matériel de verrouillage électrique de porte, panneaux de contrôle d'accès et serveurs de contrôle d'accès.

## II.10 Les différentes fonctionnalités de contrôle d'accès

Pour L'autorisation d'accès Il y a trois façons de prouver son identité face à un système informatique :

- de montrer ce que l'on sait (un mot de passe ou code) sur Smartphone.
- de montrer ce que l'on possède (un objet, telle une carte à puce, un badge).
- de prouver qui l'on est grâce à une caractéristique physique propre (biométrie).
- 

---

<sup>3</sup> API : une interface de programmation applicative est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

### II.10.1 Le contrôle d'accès pour optimiser la sécurité

- **Les codes d'agression** : ce code permet d'ouvrir la porte sécurisée tout en déclenchant une alarme silencieuse. Une solution efficace en cas d'agression d'un salarié qui coopère pour ne pas courir de danger tout en signalant dans le même temps l'agression dont il fait l'objet.
- **Pilotage des sas** : permettant l'accès à une seule personne à la fois .
- **Gestion des alarmes** : en cas d'intrusion.
- Identification des tentatives d'accès suspectes.
- **Paramétrage des zones d'accès** : définir et reconnaître les personnes et/ou les horaires autorisés.

### II.10.2 Le contrôle d'accès pour gérer le flux des visiteurs

- **Anti-passback** : cette fonction empêche qu'un même badge ouvre deux fois de suite une même porte dans un délai de temps défini au préalable. Utile lorsque votre établissement filtre le flux de ses visiteurs avec des tourniquets ou des barrières de parking par exemple : la tentation d'ouvrir à plusieurs visiteurs avec le même badge est, de ce fait, annihilée.
- Système de vérification d'identité.
- **Fonction portier** : cette option vous permet à vous, votre portier ou votre accueil d'ouvrir aisément des portes depuis votre station de travail.

### II.10.3 Le contrôle d'accès pour gérer les accès

- **Profils et groupes de profils** : cette fonction précise le temps où l'accès est permis sur une porte. Ainsi, vous pouvez définir quels profils accèdent à quels locaux dans quels délais. Pour faciliter la tâche, il est possible de rassembler les profils en groupes de profils. Par exemple, vous pouvez décider qu'un groupe de personnes peut pénétrer dans tels ou tels lieux même quand l'alarme d'alerte retentit pour des raisons de sécurité.
- **Logiciels de décomptage** : recense le nombre de personnes présentes dans chaque pièce sécurisée de l'établissement.
- **Visibilité du droit d'accès** : les logiciels permettent de savoir quelles personnes ont le droit de se trouver dans quels lieux. Ils regroupent les informations sur le droit d'accès de chaque individu. Ils peuvent aussi émettre des rapports sur les accès. [25]

## II.11 Modes de communication

Les systèmes de contrôle d'accès actuel utilisent pratiquement plus des cartes à puce et des technologies sans contact comme les Smartphone. Les autres technologies de carte (bandes magnétique, code- barre, infrarouge, etc..) se rencontrent encore parfois sur des systèmes plus anciens. [26]

### II.11.1 Communication avec des cartes à puce

Les cartes équipées d'une puce à contact sont généralement utilisées pour l'accès au PC car elles permettent différentes possibilités de codage et de sécurité.

Les puces sont équipées de contacts dont l'emplacement a été défini dans une norme ISO<sup>4</sup> c'est pourquoi il faut des lecteurs de cartes onéreux capable de mettre en rapport les capteurs de contact dans le lecteur avec les surfaces de contact de la carte.

### II.11.2 RFID

Le RFID (radio frequency identification) identification à l'aide d'ondes électromagnétiques et désigne une technologie de systèmes d'émetteurs –récepteurs pour l'identification automatique et sans contact et la localisation des objets par des ondes radio cette technologie facilite considérablement la saisie de données. Un système RFID se compose toujours d'un transpondeur intégré dans un objet par exemple badge contenant un code déterminé ainsi que d'un lecteur pour la lecture de ce code.

### II.11.3 Near field communication

Le Near field communication est une norme de transmission internationale pour l'échange sans contact de données par technologie radio sur des distances de quelques centimètres.

### II.11.4 RCID

La RCID ( Resistive capacitive identification ) est une technologie de transmission d'informations à l'aide de champs électriques (quasi) statique totalement inoffensifs pour le corps. Contrairement à la technologie RFID , la RCID n'utilise pas de champs électromagnétiques.

---

<sup>4</sup> ISO : nternational Organization for Standardization ,Document établi par consensus et approuvé par un organisme reconnu, qui fournit, pour des usages communs et répétés, des règles, des lignes directrices ou des caractéristiques, pour des activités ou leurs résultats garantissant un niveau d'ordre optimal dans un contexte donné.

Dans les champs électrostatiques, la transmission d'informations se fait par des électrodes. Le corps de l'utilisateur devient porteur du signal émetteur. Dès que celui-ci touche l'électrode réceptrice, la moitié du circuit électrique est fermée. L'autre moitié se ferme via l'environnement. Si on porte un émetteur sur soi par exemple dans la poche du pantalon et que on touche un récepteur par exemple poignée de porte ou tapis de sol, le circuit se ferme. [26]

### II.11.5 communication avec Smartphone

Le contrôle d'accès mobile fonctionne selon le même principe que le courrier électronique accessible par mobile, une fois l'application téléchargée, les utilisateurs peuvent accéder à leur courrier et y répondre, à condition de saisir les informations d'identification correctes. Il en va de même pour le contrôle d'accès sur Smartphone. Une fois autorisés et les utilisateurs ont téléchargé l'application de contrôle d'accès sur leur Smartphone, ils peuvent faire la même chose, à savoir, connectez-vous avec leur compte d'utilisateur et dès que leurs clés autorisées apparaissent, sélectionnez la porte à ouvrir.

### II.11.6 communication basé sur L'IOT<sup>5</sup>

L'Internet des objets est un système de contrôle d'accès connu sous le nom de système de verrouillage intelligent. Dans ce système, chaque verrou, contrôleur d'accès au verrou, lecteur de carte et autres périphériques associés se voient attribuer une adresse IP distincte, utilisée pour la communication entre les périphériques, ses derniers sont normalement connectés via des réseaux sans fil à leurs logiciels de gestion consolidés ou à leurs applications mobiles. Ces applications peuvent être configurées pour le fonctionnement automatique de différents verrous et contrôleurs. Les alertes et notifications de sécurité peuvent également être configurées pour recevoir des applications en temps réel.

Chaque appareil / machine est configuré pour ses conditions de fonctionnement, ses critères, sa sensibilité et son autorité dans le logiciel de contrôle de gestion de base, qui est utilisé comme contrôleur de l'ensemble du système. Un contrôle en double de ce système est également utilisé comme application mobile sur les appareils mobiles.

---

<sup>5</sup> **IOT** : L'Internet des Objets est un réseau d'objets physiques dédiés qui intègrent des technologies pour détecter ou interagir avec leurs états internes ou leurs environnements externes. L'IoT constitue un écosystème d'objets, de communications, d'applications et d'analyses des donnée

## II.12 Vidéosurveillance dans les systèmes contrôle d'accès

Il peut être très utile d'intégrer le contrôle d'accès dans un système de vidéosurveillance .Cela facilite la constatation a posteriori d'un abus éventuel de laissez passer à l'aide d'enregistrement d'accès autorisé et non autorisés .Dès qu'une personne s'approche d'un accès ou d'une porte ou s'identifie sur un lecteur , les programmes correspondants permettent d'activer immédiatement la caméra via le détecteur de mouvement .Pour le personnel de sécurité ,il est possible d'effectuer un contrôle de sécurité , il est possible d'effectuer un contrôle supplémentaire soit en direct sur l' écran soit ultérieurement sur la base des images enregistrées.

## II.13 Gestion de porte

Les systèmes de contrôle d'accès doivent être associés à la commande de porte, il faut toute fois des contrôleurs et composants de porte spéciaux .Les portes et lecteurs sont commandé surveillé par un contrôleur de porte.

La performance d'un contrôleur de porte dépend toutefois du niveau de la porte elle même et des composants de sécurité selon la fréquence et la situation de la porte , il peut y avoir plusieurs centaine de passage par jour ,ce qui représente une sollicitation énorme pour des pièces mécaniques et électromécaniques , il est donc essentiel de s'occuper de la planification suffisamment tôt ,en évitant les composants qui tombent en panne , il est tout important de bien réfléchir aux fonctions de protection de portes par exemple contre l'effraction, l'incendie , les chocs, mais aussi aux issues de secours .[26]

### II.13.1 Types de portes

Il existe différentes sortes de portes et d'accès :

- **Portes tambour** : deux ou quatre battants de porte tournants dans un espace circulaire pour permettre les entrées et sorties simultanées.
- **Portes à deux battants** : séparation des espaces intérieurs et extérieurs, verrouillage par une serrure.
- **Portes coulissantes** : un vantail ou plusieurs guidé en haut ou en bas s'ouvrant latéralement.
- **Tourniquets tripodes** : équipement fixe muni de trois barres pour un accès individuel
- **Sas** : cabinet avec deux portes en vis- a vis comme mesure d'individualisation.

- **Sas avec capteurs** : accès surveillé par des capteurs équipé de portes coulissantes ou battants.

### II.13.2 Équipements de contrôle de portes

- **Serrures intelligentes** : Les serrures électromécaniques intègrent un lecteur d'accès qui simplifie le câblage et réduit les coûts d'installation. Elles sont reliées et pilotées par le logiciel de contrôle d'accès.
- **Gâches électriques** : Les gâches électriques sont un composant des serrures électromécaniques. L'accès se fait par rupture ou par émission de courant.
- **Serrure autoverrouillante** : le verrou se referme automatiquement après chaque ouverture de la porte. [26]

## II.14 Conclusion

Dans ce chapitre nous avons présenté en premier lieu la carte Raspberry pi et ses différents modèles, ses systèmes d'exploitation, les différentes utilisations de cette carte Raspberry pi .De même nous avons donné une vue globale sur les systèmes de contrôle d'accès, commençant par ses composants de base, ses différentes fonctionnalités ainsi les modes de communication actuel. et à la fin nous avons abordé la vidéosurveillance et la gestion de porte.

Dans le chapitre suivant nous allons aborder l'analyse et la conception de notre projet.

# **Chapitre III**

## **Analyse et conception**

## III.1 Introduction

La conception de toute solution logicielle doit être traitée avec précision et détail, précédée d'une analyse profonde et bien réfléchie, car elle est le reflet du futur système avant même sa concrétisation. Dans le but d'avoir une meilleure analyse et de rendre la conception du logiciel plus complète nous allons étudier tous les outils et le matériel qui nous serviront à la réalisation de notre application. Puis nous allons passer à la phase conception.

## III.2 Analyse

Au cours de cette phase, nous recensons les besoins fonctionnels et les contraintes de mettre le système. Les besoins fonctionnels décrivent le comportement et les fonctions du système requis, ils doivent être clairement décrits afin que ce système soit capable de satisfaire les besoins et les demandes des parties prenantes, tandis que les contraintes sont des limitations, dans lesquels le système doit fonctionner.

### III.2.1 Objectif du système

La réalisation d'un système d'accès aux zones sécurisé tel que des établissements ou de locaux à l'intérieur d'une entreprise, afin d'offrir davantage de sécurité et de permettre un meilleur suivi d'accès et garantir une meilleure traçabilité à base des systèmes embarqués.

### III.2.2 Fonctionnalité du système

Dans notre projet nous allons développer une application embarquée utilisant les technologies web qui permettent de gérer le système de contrôle d'accès par un Smartphone ou par un ordinateur, Cette application peut être divisée en deux parties :

#### ➤ Première partie pour l'utilisateur

En cas de demande d'accès, l'utilisateur utilise son Smartphone qui est connecté au même réseau local de l'entreprise, une application web sera chargée cette dernière lui permettra de s'authentifier par un identifiant unique et un mot de passe, les informations d'identification sont transmises au module central. Ce dernier compare ces informations aux données dont il dispose et au même temps une photo sera capturée et enregistrée, puis le résultat de la comparaison détermine si la demande d'accès est accordée, une fois l'utilisateur est vérifié il peut ouvrir la porte depuis l'application.

#### ➤ Deuxième partie pour l'administrateur

Dans la même application, une partie est limitée à l'administrateur qui gère l'application, il peut ajouter ou supprimer des utilisateurs de la base de données selon les droits d'accès et consulter les photos prises pour les utilisateurs qui ont essayé d'accéder. En plus l'application offre la possibilité d'assurer la sécurité de la zone par une caméra de surveillance.

### III.2.3 Spécification des besoins fonctionnels et non fonctionnels

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les exploitants et d'identifier les cas d'utilisation.

Le système à réaliser doit satisfaire et répondre aux exigences de l'utilisateur. A ce titre les besoins fonctionnels et les besoins non fonctionnels seront présentés.

#### III.2.3.1 Besoins fonctionnels

Lors de la demande d'accès à une zone, l'utilisateur doit se connecter au même réseau local de système installé pour qu'il puisse accéder à l'application web sur son Smartphone ensuite il va s'authentifier avec son identifiant unique et un mot de passe.

Le système doit alors remplir les fonctionnalités suivantes :

- Attente d'un événement (charger l'application).
- Extraction du contenu (identifiant).
- Envoi de l'identifiant au serveur ainsi son mot de passe.
- Vérification des droits d'accès.
- Demande de réessayer en cas d'erreur.
- L'envoi de la réponse :
  - Autoriser en cas de validation
  - Refus.
- Prendre une photo.
- Vidéosurveillance

#### III.2.3.2 Besoins non fonctionnels

Il s'agit des besoins qui caractérisent le système, ce sont des besoins en matière de performance, de type matériel ou de type conception. Le système doit répondre aux critères suivants :

- ✓ La rapidité de traitement.
- ✓ La performance.
- ✓ La facilité d'utilisation.

On peut résumer Le fonctionnement global du système dans le schéma de la figure III.1

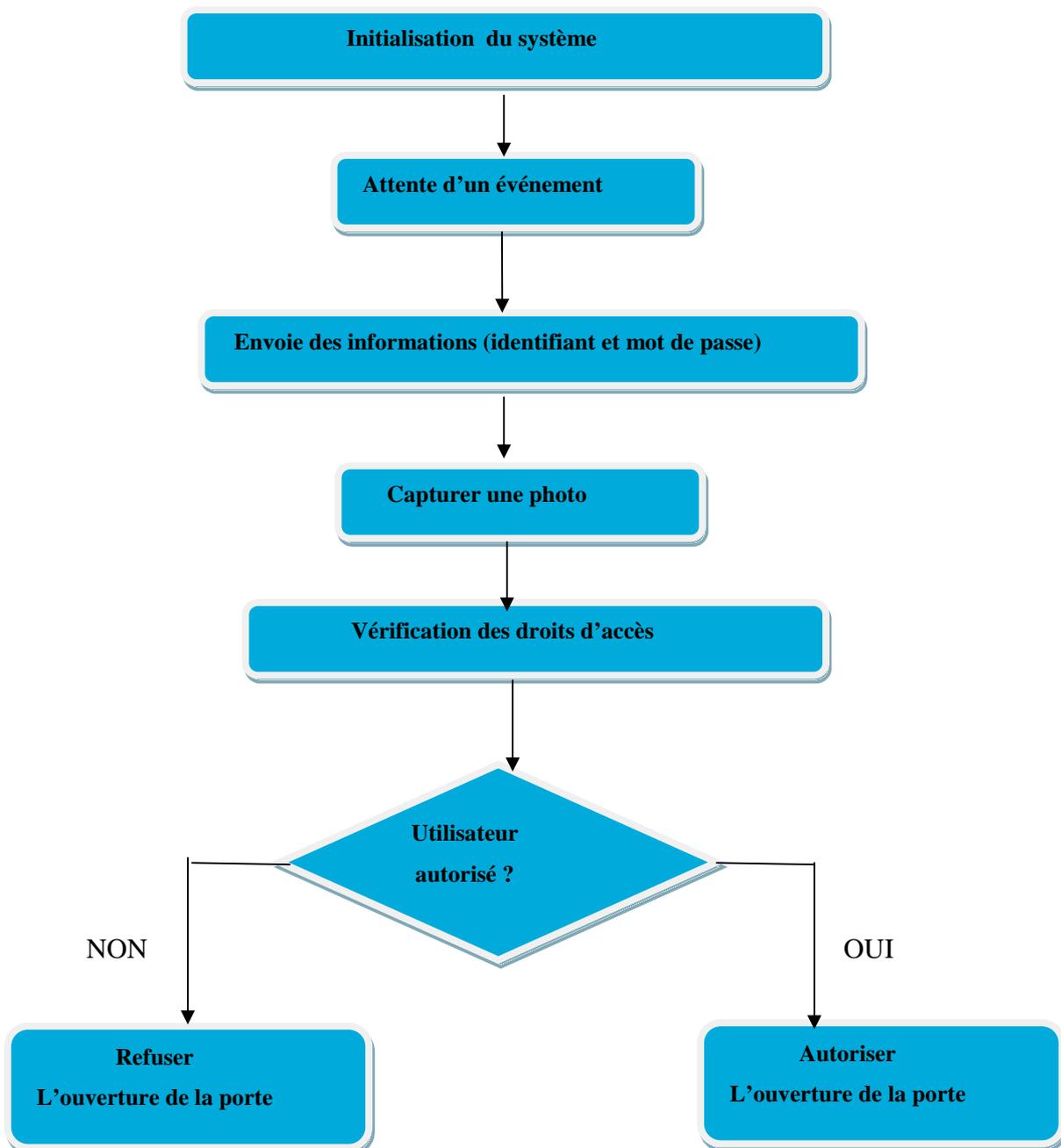


Figure III.1 : Schéma de fonctionnement global du système

### III.2.4 Spécification des besoins matériels

Afin de réaliser le système décrit précédemment, on doit utiliser un ensemble de composants matériels en interaction, chacun réalise une tâche précise. Le système doit

disposer d'un module central qui est chargé de commander et de recevoir les instructions des autres modules, c'est lui qui gère et maintient le fonctionnement du système.

Pour autoriser l'accès à un endroit spécifique, le système doit disposer d'un module d'accès, qu'est une application web d'authentification qui permet de récupérer l'identifiant de l'utilisateur et son mot de passe afin de les vérifier. Cette application est hébergée dans un serveur web embarqué sur le nano ordinateur RaspberryPi, et elle déclenche un moteur relié à une porte puis l'ouvrir si un accès est accordé. Comme illustré dans la figure III.2:

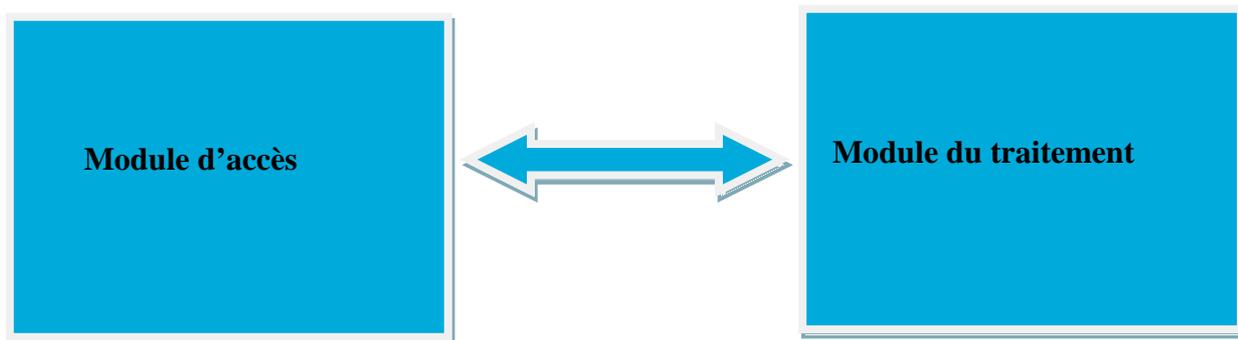


Figure III.2 : Vue simplifiée de l'architecture générale du système

### III.2.4.1 Composant requis

Avant d'entrer dans la conception du système, il faut faire le point sur les composants que nous allons utiliser.

D'après la structure du système, il exigera quatre types de modules :

#### III.2.4.1.1 Module de traitement

Le module de traitement est composé d'une **carte Raspberry pi3**

Ce modèle de nano ordinateur est plus rapide et plus puissant que ses prédécesseurs. Il a amélioré la puissance de gestion pour prendre en charge des périphériques USB externes plus puissants. Raspberry Pi 3 est maintenant livré avec un module sans fil intégré et une connectivité Bluetooth. [28] Parmi les composants de la Raspberry Pi on va s'intéresser au GPIO et la carte SD.

#### ➤ Les GPIO

GPIO est l'acronyme anglais de "*General Purpose Input Output*" ou plus simplement Entrée/sortie pour usage général. Ces entrées sorties permettent d'étendre les fonctionnalités du Raspberry pi en lui donnant la possibilité d'agir sur les leds ou des afficheurs LCD par exemple, lire l'état d'un interrupteur, d'un capteur, etc.

Le connecteur GPIO dispose actuellement de différents types de connexions on distingue :

- Des broches pour une interface I2C qui permet de se connecter sur du matériel en utilisant uniquement 2 broches/pins de contrôle.
- Une interface SPI pour les périphériques SPI, à un concept similaire à I2C mais avec un standard différent.
- Les broches Rx et Tx pour la communication avec les périphériques séries.
- PWM (acronyme de "Pulse Width Modulation" pour modulation de largeur d'impulsion) permettant le contrôle de puissance
- PPM (acronyme de "Pulse Position Modulation" pour modulation positionnelle d'impulsion) permettant de contrôler des servos moteurs.

La figure III.3 illustre les broches GPIO du Raspberry Pi3

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 2  
29/02/2016

www.element14.com/RaspberryPi

Figure III.3: GPIO Raspberry Pi model 3B

➤ **La carte SD**

La carte Raspberry Pi n'a pas de disque dur pour le stockage de masse, mais un lecteur de carte mémoire SD (Secure Digital) comme celles que l'on peut trouver équipant les appareils photo numériques. Avant de démarrer la Raspberry pi, il faut préparer une carte SD et y installer la totalité du système d'exploitation, qui fera fonctionner la Raspberry. La taille de la carte SD (8GO min) est un des critères puisqu'elle limitera les fonctionnalités de la

Raspberry, sa vitesse d'accès est également un critère important (classe 10) car la rapidité de la carte jouera sur celle du système final.

#### III.2.4.1.2 Module d'acquisition

Ce module est composé **d'une caméra USB**

Dans notre projet on a implémenté une caméra de surveillance afin d'assurer la sécurité de notre zone. Pour ce faire, on aura besoin d'une webcam qui sera branché à l'un des quatre ports USB de la Raspberry.

##### ➤ **Caractéristiques :**

- Capteur de résolution 640×480.
- Capteur d'image: 1/4 Haute Résolution CMOS VGA Capteur.
- Support ingénieux et idéal pour les écrans d'ordinateurs portables et les surfaces planes.
- Prête à l'emploi, aucun pilote à installer.

La figure III.4 illustre une caméra USB



**Figure III.4: camera USB**

#### III.2.4.1.3 Module de Réaction

Ce module sera **un moteur pas à pas**

Le moteur pas-à-pas se situe entre le moteur à courant continu ordinaire et le servomoteur. Le moteur pas-à-pas a l'avantage de pouvoir être placé dans une position déterminée et de la garder, ou de tourner dans un sens ou dans l'autre à volonté d'un « pas » (step) à la fois. Mais il peut aussi tourner de façon « continue ».

Pour notre projet, nous avons utilisé un moteur pas à pas unipolaire à 5 fils, il est constitué de bobines disposant d'un point milieu qui permettent d'alimenter un axe magnétique soit dans un sens soit dans un autre en alimentant seulement une demi-bobine.

Le moteur se branche aux pins de la Raspberry via un contrôleur driver. [29]

➤ **Caractéristiques :**

- Un petit moteur économique permet de motoriser plusieurs montages, il fonctionne en 5v.
- Le moteur comporte 64 pas par tour.
- Des diodes de visualisation permettent de contrôler le bon fonctionnement.
- Le driver ULN2003

Le moteur se branche sur driver ULN2003 avec son fil (fourni), muni d'un connecteur à 5 broches (avec détrompeur de sens de branchement)

- Rouge = commun
- Orange=Pin1
- Jaune = Pin2
- Pourpre =Pin3
- Bleu=Pin4
- Le driver UNL2003 est relié à une alimentation de 5v et le GND (la masse)

La figure III.3.5 illustre un moteur pas à pas avec son driver



**Figure III.5: moteur pas à pas avec son driver UNL2003**

#### III.2.4.1.4 Module d'affichage

L'aspect de l'affichage sera présenté sous forme d'une application web, son développement nécessite les composants suivants :

- Serveur web (apache, PHP, MySQL)
- Smartphone ou un ordinateur

### III.3 Conception

Après avoir présenté le matériel nécessaire dans de ce chapitre, nous allons, suivre une méthode de conception qui aura pour rôle la formalisation des étapes préliminaires du développement de notre système. La phase de conception permet de décrire de manière plus ou moins précise le fonctionnement futur de notre système et de faciliter sa réalisation.

Un système est un ensemble de composants qui sont combinés ensemble pour exécuter une tâche compliquée d'une manière coordonnée. Le composant est utilisé ici dans un sens très large, y compris les logiciels, le matériel informatique, l'alimentation.

#### III.3.1 Le graphe de flux de données

Le graphe de flux de données est un schéma fonctionnel du système montrant le flux d'informations. Les flèches pointent de la source vers la destination. Les rectangles représentent les composants matériels et les ovales sont des modules logiciels.

Nous utilisons des données graphiques de flux dans la conception de haut niveau, car ils décrivent le fonctionnement global du système, tout en cachant les détails de la façon dont il fonctionne.

La figure représente le graphe de flux de données pour le système de contrôle d'accès :

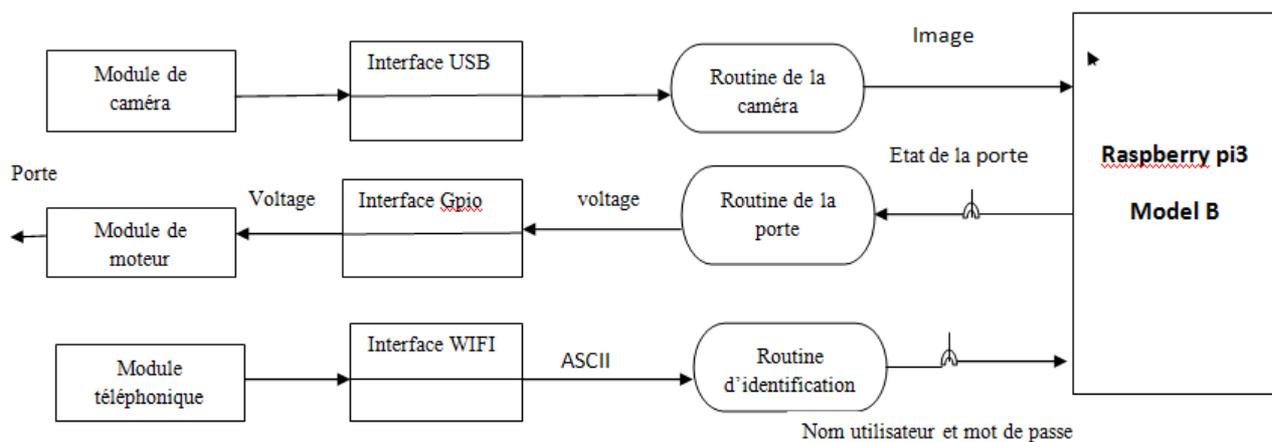


Figure III.6: Le graphe de flux de données pour le système embarqué dans le système de contrôle d'accès.

#### III.3.2 Le graphe de Raffinement successif

Raffinement successif, affinement progressif, et la décomposition systématique sont trois termes équivalents à une technique pour convertir un énoncé de problème dans un algorithme logiciel.

Nous commençons par une tâche puis on la décompose en un ensemble de sous-tâches plus simples. Ensuite, les sous-tâches sont décomposées en sous-tâches encore plus simples. Nous faisons des progrès aussi longtemps que chaque sous-tâche est plus simple que la tâche elle-même.

Finalement, une sous-tâche est si simple qu'elle peut être convertie en un code de logiciel. On peut décomposer une tâche de trois manières différentes, comme le montre la figure III.7 :

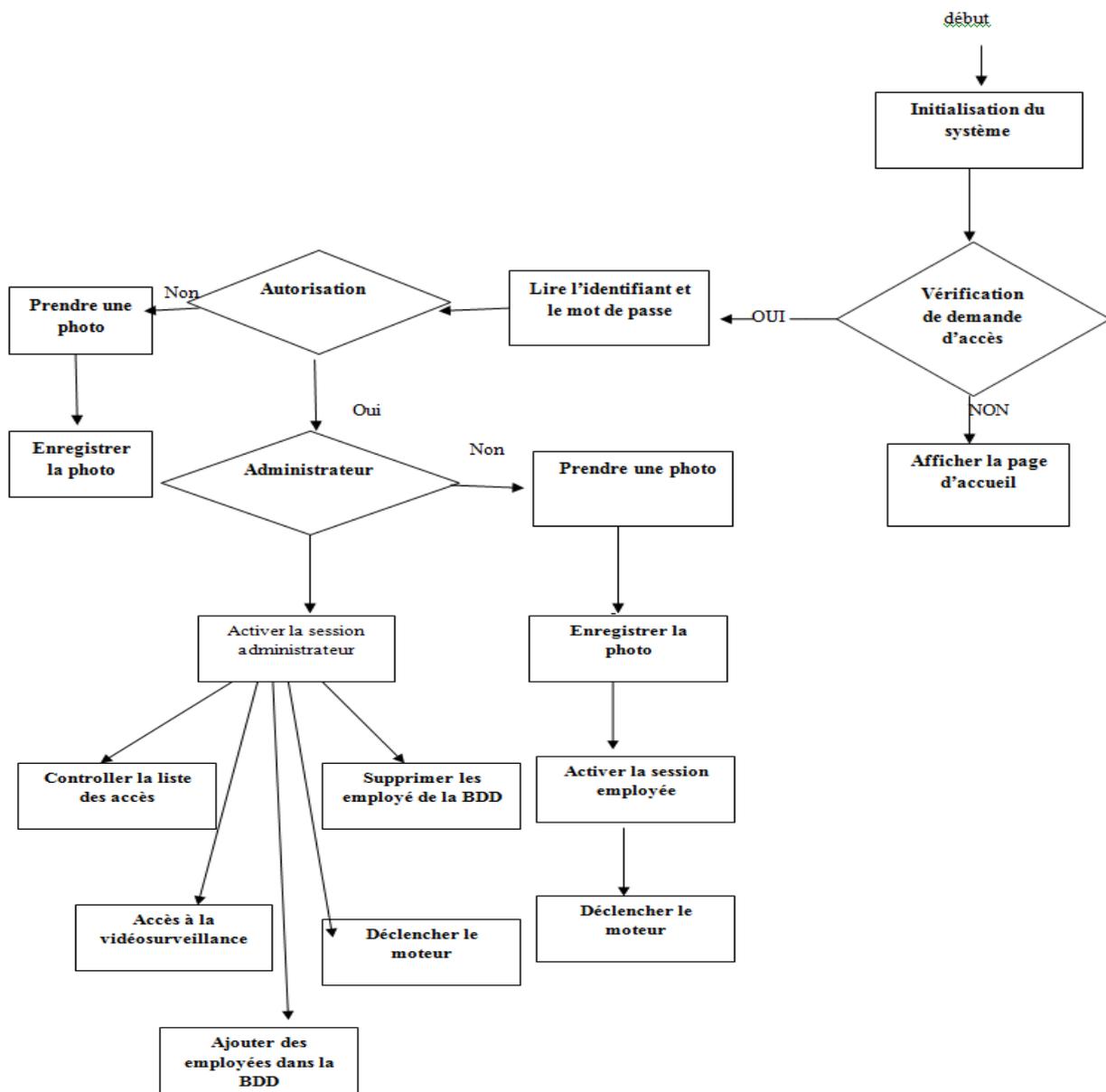


Figure III.7: Le graphe de Raffinement successives pour le système embarqué dans le système de contrôle d'accès

## III.4 Analyse et conception de notre application web

La phase de conception permet de décrire de manière plus ou moins précise le fonctionnement futur de notre système et de faciliter sa réalisation.

### III.4.1 Langage UML

L'UML est un langage de modélisation qui permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation. L'UML est sous l'entière responsabilité de l'OMG (Object Management Group). Il a été conçu pour servir de support à une analyse basée sur les concepts objet. Il se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, à spécifier et documenter des systèmes, à concevoir des solutions et communiquer des points de vue.

L'UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. Nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression des besoins jusqu'au codage. C'est un langage formel possédant les caractéristiques suivantes :

- Un langage sans ambiguïtés.
- Un langage universel pouvant servir de support pour tout langage orienté objet.
- Un moyen de définir la structure d'un programme.
- Une représentation visuelle permettant la communication entre les acteurs d'un même projet.
- Une notation graphique simple, compréhensible même par des non informaticiens

UML s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la présentation de concepts particuliers d'un système logiciel. Nous allons représenter ceux qui sont utilisés dans notre projet et qui sont :

- Les Diagrammes de cas d'utilisation.
- Les Diagrammes de séquence.

### III.4.2 Vue fonctionnel

#### III.4.2.1 Les acteurs de notre système

Notre projet possède deux acteurs l'utilisateur et l'administrateur.

**Administrateur :** c'est l'administrateur du système, il gère l'application de système de contrôle d'accès.

**L'utilisateur** : cet acteur a le droit de se servir de notre plateforme de système de contrôle d'accès localement en utilisant son téléphone.

### III.4.2 .2 Spécification des scénarios

#### ➤ Administrateur

Acteur	Tâche	Scénarios
<b>Administrateur</b>	T0 : Se connecter à la plateforme Administrateur	S0 : Saisir l'@IP de Raspberry pi et l'URL de site dans le navigateur
	T1 : S'authentifier	S1 : Accéder au formulaire d'authentification S2 : Saisir les coordonnées S3 : Valider l'authentification
	T2 : Ajouter des utilisateurs T3 : Supprimer des utilisateurs	S4 : Remplir le formulaire d'ajout
	T4 : Consulter des photos prise pour les utilisateurs	S5 : Cliquer sur le lien de la photo
	T5 : Se déconnecter	S : Cliquer sur le lien de « déconnexion »

**Tableau.1 : Les scenarios des tâches de l'administrateur de notre application.**

➤ **Utilisateur**

Acteur	Tâche	Scénarios
<b>Utilisateur</b>	T0 : Se connecter à la plateforme	S0 : Saisir l'@IP DE Raspberry pi et l'URL de site dans le navigateur
	T1 : s'authentifier	S1 : Valider l'authentification

**Tableau.2 : Les scenarios des tâches de l'administrateur de notre application.**

### III.4.2.3 Spécification des cas d'utilisation

Le digramme de cas d'utilisation décrit le comportement fonctionnel d'un système et permet de définir les exigences de ce dernier.

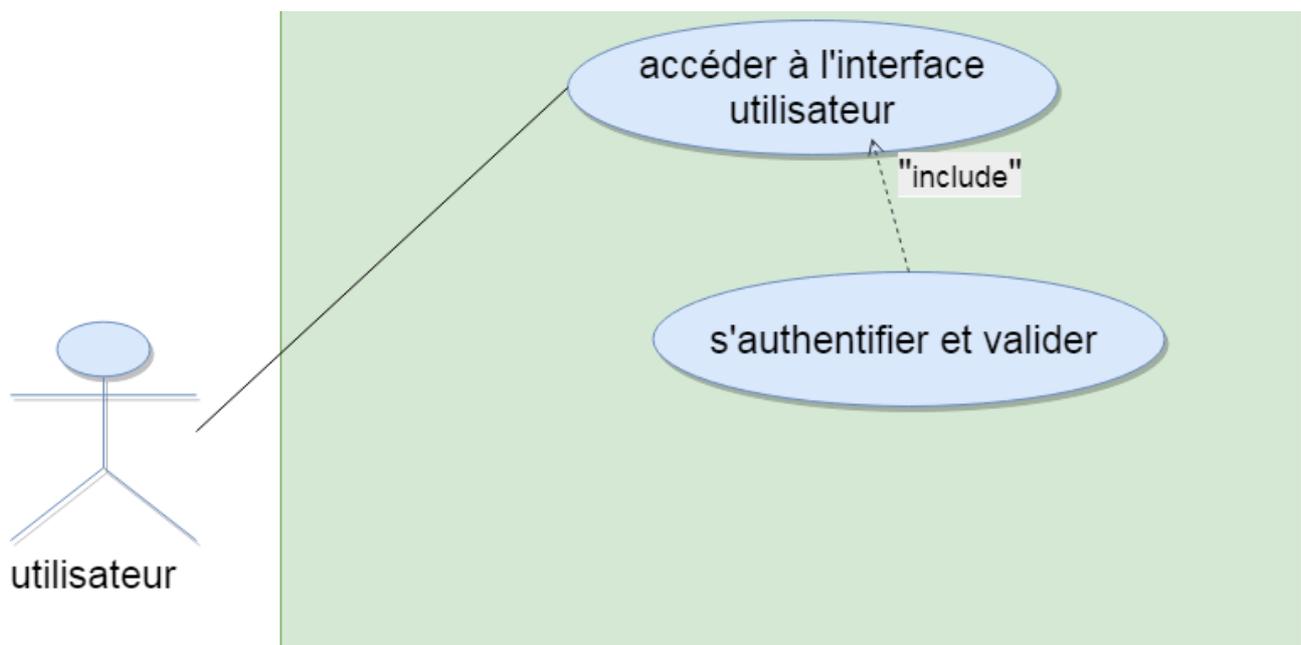
Un digramme de cas d'utilisation permet également d'identifier les interactions entre les utilisateurs (acteurs) et le système.

**Un cas d'utilisation :** représente une interaction entre l'acteur et le système.

**Association de cas d'utilisation :** est une relation unidirectionnelle qui décrit le lien entre un cas d'utilisation et un acteur.

➤ **Utilisateur**

La figure III.8 présente « le diagramme de cas d'utilisation relatif à l'utilisateur »



### Figure III.8: Diagramme de cas d'utilisation relatif à l'utilisateur

L'utilisateur doit s'authentifier afin d'avoir accès pour ouvrir la porte localement.

#### ➤ Administrateur

La figure III .9 présente« diagramme de cas d'utilisation relatif à l'administrateur »

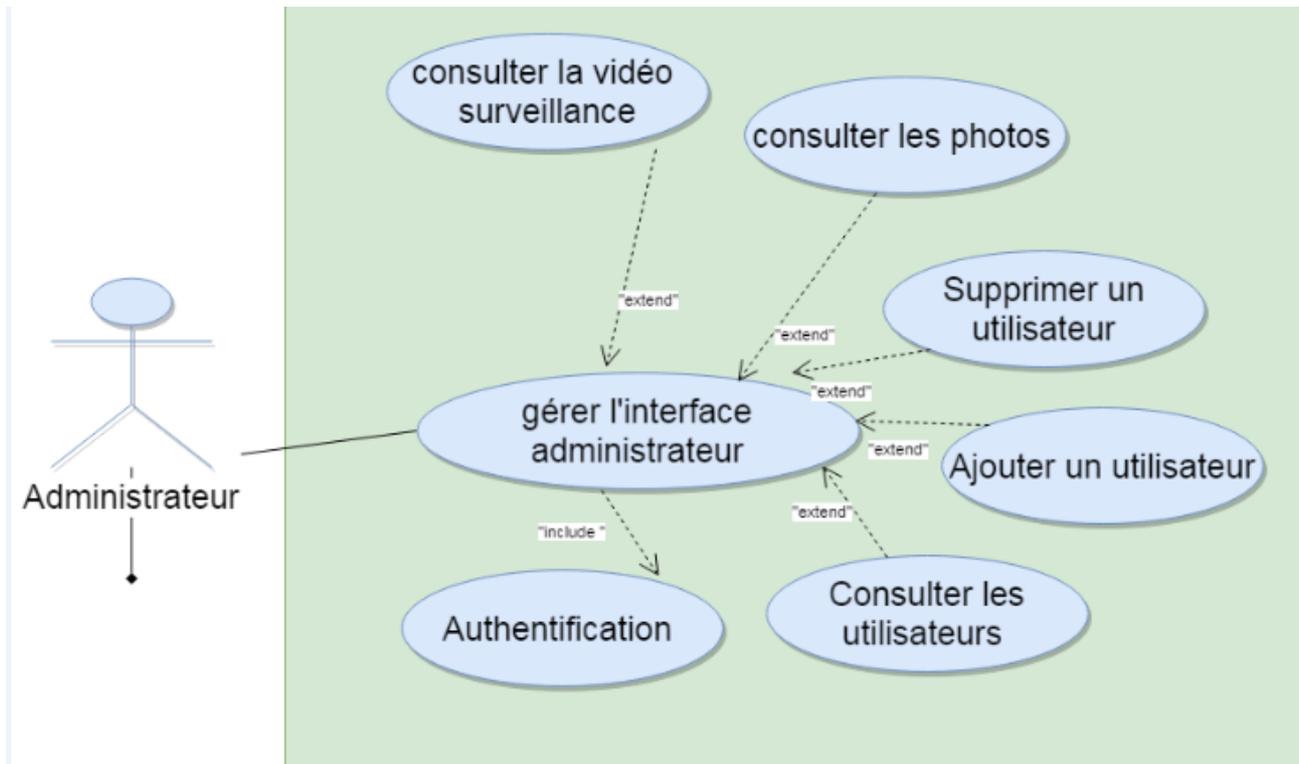


Figure III.9 : Diagramme de cas d'utilisation relatif à l'administrateur

L'administrateur s'authentifie afin de pouvoir gérer l'application en ce qui concerne l'ajout, la suppression et la consultation des utilisateurs.

## III.4.3 Vue dynamique

### III.4.3.1 Les diagrammes de séquence

Les diagrammes de séquence décrivent le déroulement de chaque cas d'utilisation, en montrant la façon dont les diverses entités mises en œuvre dans les cas d'utilisation interagissent et collaborent dans le temps afin de réaliser les fonctionnalités attendues.

Le but est de déterminer :

- Les diverses entités, appelées objets, mises en jeu dans la réalisation d'une fonctionnalité
- Les interactions entre ces différentes entités

- Le déroulement dans le temps de ces entités

Typiquement, chaque cas d'utilisation déterminé dans le diagramme de cas d'utilisation fait l'objet d'une étude temporelle des interactions en utilisant un diagramme de séquence.

Les diagrammes de séquence sont donc une suite logique du diagramme de cas d'utilisation.

➤ **Diagramme de séquence de « demande d'accès de l'utilisateur » :**

Quand un utilisateur se connecte au serveur, il peut accéder à l'interface de la plateforme via son téléphone portable ; puis saisit son login et son mot de passe, ces informations seront vérifiées, si tout est bon, la porte s'ouvrira. La figure III.10 présente le diagramme de séquence « authentification de l'utilisateurs ».

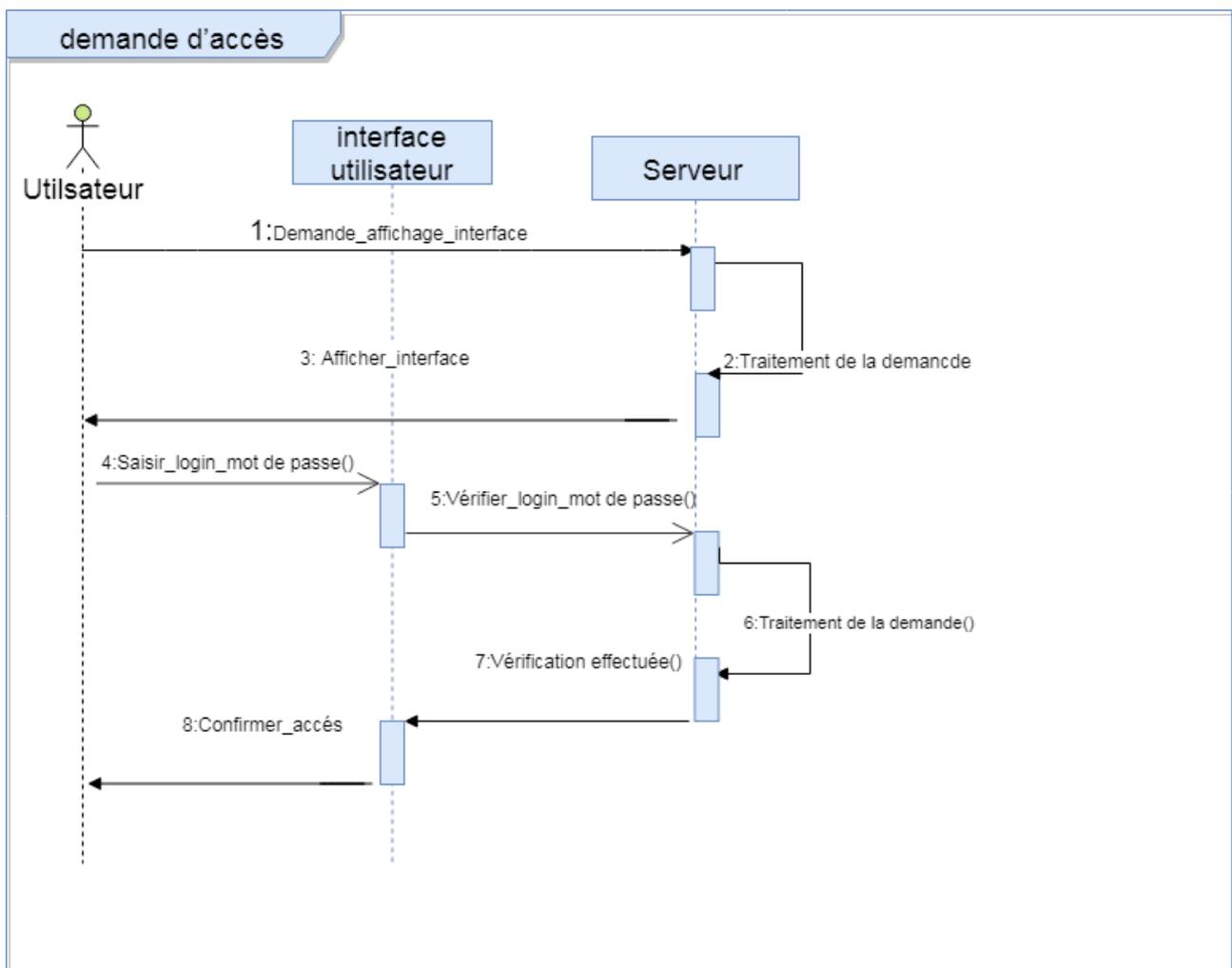


Figure III.10 : Diagramme de séquence « Demande d'accès de l'utilisateur »

➤ **Diagramme de séquence « authentification de l'administrateur »**

Quand l'administrateur se connecte au serveur il peut accéder à son interface via un téléphone ou un ordinateur, puis il saisit son mot de passe cette information sera vérifiées, si

tout est bon, l'interface de l'administrateur sera affichée, la figure III.11 présente le diagramme de séquence « authentification de l'administrateur ».

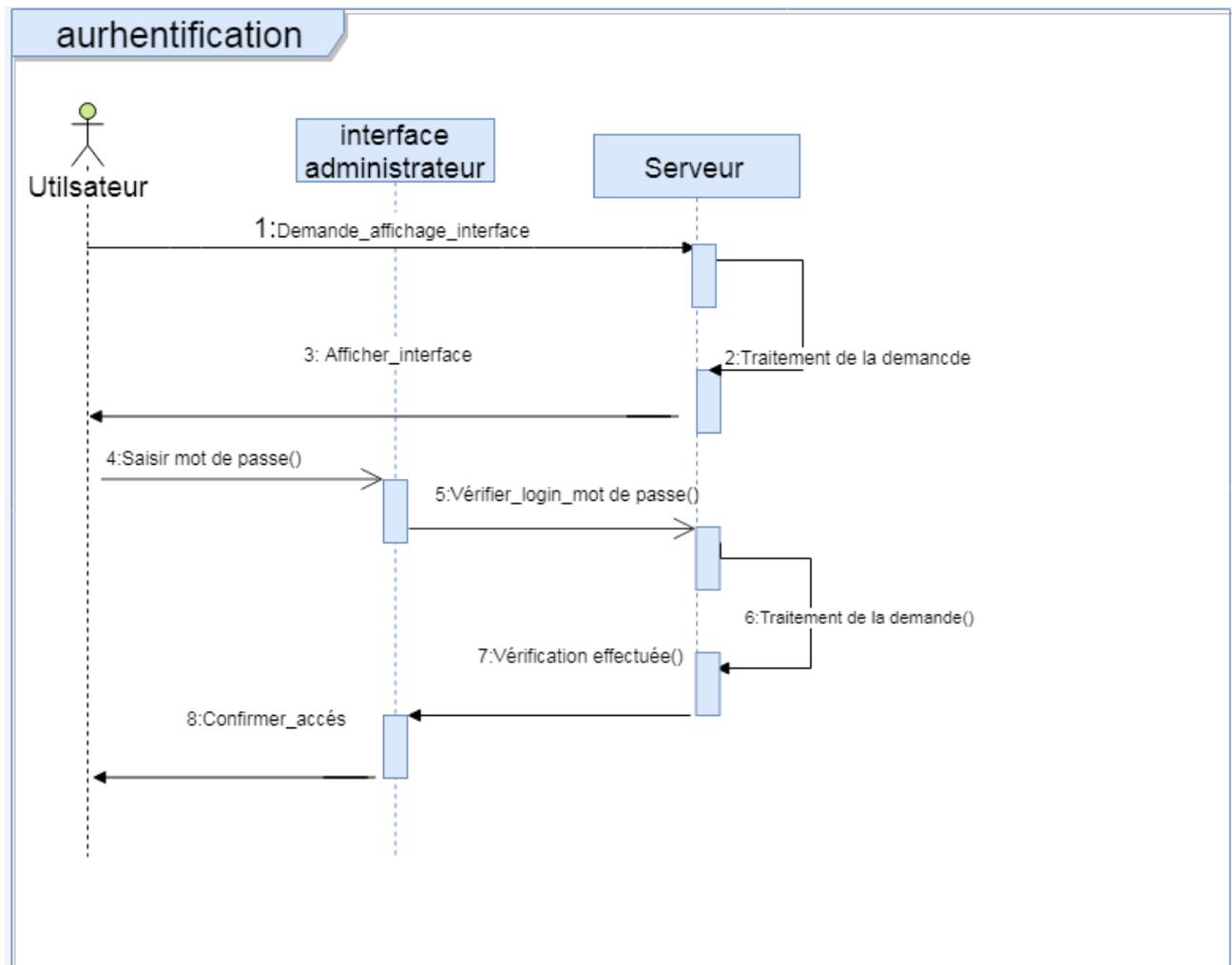


Figure III.11: Diagramme de séquence « authentification de l'administrateur »

### III.5 La base de données :

Après avoir modélisé notre application web avec les différents diagrammes offerts par le langage de modélisation UML une mise en œuvre d'une base de données sur un serveur web est nécessaire pour que l'administrateur puisse enregistrer des utilisateurs, faire une consultation, une saisie ou une mise à jour et vérification à chaque accès.

Utilisateurs
Id_username
password

Figure III.12 : table des utilisateurs

## III.6 Conclusion

Tout au long de ce chapitre, nous avons exposé les différents besoins auxquels doit répondre l'application ainsi que l'étude des différents concepts et moyens utilisés pour la réalisation de cette application.

Dans le chapitre suivant, nous allons aborder la réalisation de notre système en pratique.

# **Chapitre IV**

## **Réalisation et Test**

## IV.1 Introduction

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

La réalisation est la phase la plus importante après celle de la conception. Le choix des outils de développement détermine énormément le coût en temps de programmation, ainsi que la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi en des composants logiciels formant notre système. Dans un premier temps, nous allons procéder à la spécification de l'environnement matériel et logiciel utilisé dans notre projet.

Ensuite nous intéresserons à décrire les différentes étapes de réalisation de l'application « système de contrôle d'accès ».

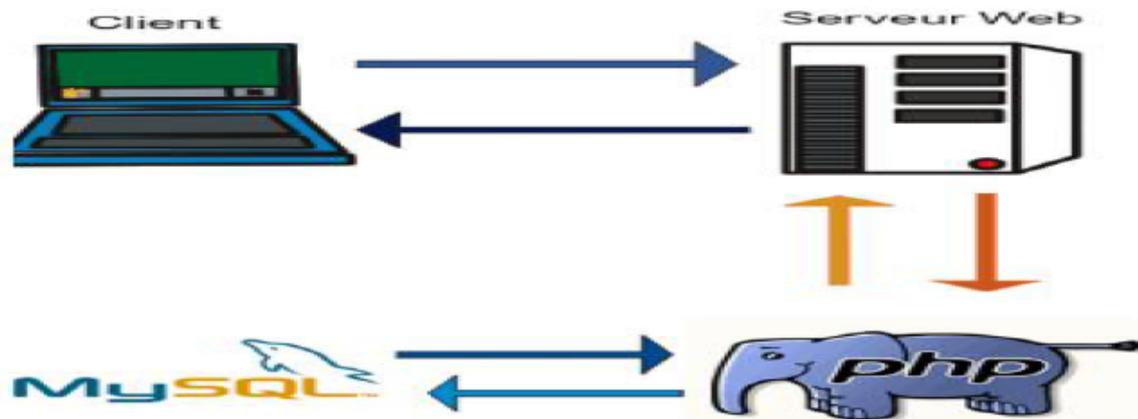
## IV.2 Outils de développement utilisés :

Pour développer notre système, nous avons eu recours à divers éléments notamment les outils logiciels, les langages de programmation et les outils matériels qui nous ont permis la mise en œuvre de notre système.

### IV.2.1 Outils logiciels

Pour pouvoir accéder à notre Raspberry à distance via WiFi et contrôler les GPIOs, on a besoin de rendre notre carte un serveur Web capable d'héberger un ou plusieurs sites Internet et assurer donc la communication avec un client et répondre à ses demandes grâce au protocole réseau http. Le plus souvent, un serveur Web utilise d'autres logiciels qui fonctionnent en collaboration avec le serveur HTTP (Apache, IIS), comme le serveur de base de données (MySQL, Oracle), et pour communiquer avec un serveur web, on a besoin d'un langage de programmation serveur comme PHP. Dans notre projet, on va opter pour installer la combinaison (Apache, PHP, MySQL)

La figure IV.1 présente l'architecture de connexion client-serveur avec PHP.



**Figure IV.1 : Architecture de connexion client-serveur avec PHP**

### IV.2.1.1 Le serveur web apache2

Apache 2 est un logiciel qui permet de mettre en disposition sur le réseau, d'un site web (page html, PHP), plus précisément apache permet à la machine d'analyser les requêtes d'un utilisateur (sous forme http), et de retourner le fichier correspondant à la requête (ou une erreur si le fichier n'est pas trouvé, ou la requête est mal formulée).

### IV.2.1.2 MYSQL

Le système de gestion des bases de données MySQL est disponible depuis 1996, mais son 2<sup>d</sup> développement remonte à 1979, ainsi il est le plus populaire des serveurs de bases de données. MySQL est un système de gestion de bases de données relationnelles (SGBDR) robuste et rapide. Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent servir simultanément d'une même base de données pour y accéder rapidement.

### IV.2.1.3 PHPMysqlAdmin

PHPMysqlAdmin est une application développée en PHP, et qui vise à fournir une interface simplifiée pour MySQL. Elle nous permet, par exemple, de voir rapidement et de façon lisible le contenu de notre base de données, ou de la manipuler sans avoir besoin de faire vous-mêmes nos requêtes MySQL.

### IV.2.1.4 Motion

Motion est un logiciel très simple d'utilisation, disponible pour Raspbian et parfaitement adapté à la vidéo-surveillance avec le Raspberry Pi. La fonction première de Motion est de faire de la vidéo-surveillance avec diffusion en ligne, en temps réel, via une ou plusieurs caméras ! Motion permet également de faire de la « motion capture », c'est-à-dire, que Motion est capable de comparer la dernière image prise avec la nouvelle pour faire de la détection de mouvements.

## **IV.2.2 Langages utilisés**

### **IV.2.2.1 HTML5 (Hypertext Markup Language 5)**

C'est la dernière version majeure d'HTML (format de donnée conçu pour représenter les pages web). Il permet notamment d'implanter de l'hypertexte dans le contenu des pages repose sur un langage de balisage. Html permet aussi de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont les images les formulaires de saisie et des éléments tels que les applets.

### **IV.2.2.2 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 du JavaScript pour contrôler les données saisies dans mes formulaires HTML, par exemple pour vérifier si le mot de passe Saisit par l'utilisateur est bien correct.

### **IV.2.2.3 JQuery**

JQuery est une bibliothèque JavaScript qui facilite le développement d'applications web et s'appuie sur l'interaction entre JavaScript et HTML, afin de simplifier des commandes communes de JavaScript et d'ajouter des fonctionnalités supplémentaires aux sites Web. Au cours des dernières années JQuery est devenue la bibliothèque JavaScript le plus populaire utilisée dans le développement web.

### **IV.2.2.4 Ajax (Asynchronous JavaScript and XML)**

Ajax « XML et JavaScript asynchrones », 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.

### IV.2.2.5 Css3

Le terme **CSS** est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelées les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

### IV.2.2.6 Python

Python est un langage de programmation, multi paradigmes et multi plat formes. Il favorise la programmation impérative structuré, fonctionnelle et orienté 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'exception. Python est un langage puissant, à la fois facile à apprendre et riche en possibilités, Il dispose de nombreuses fonctionnalités intégrées au langage. Ainsi, il existe ce qu'on appelle des bibliothèques qui aident le développeur à travailler sur des projets particuliers.

Le langage python est placé sous une licence libre et fonctionne sur la plupart des plateformes. Python est un langage de programmation interprété, c'est à dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter.

le python est un langage recommandé pour les nouvelles technologies comme par exemple le Raspberry, beaglebone, arduino, et dans notre projet nous allons utiliser du python pour commander le moteur pas à pas .

## IV.2.3 Outils matériel :

**Raspberry pi3** : elle reçoit des informations et elle envoie des commandes aux autres modules .

**Moteur pas à pas** : il reçoit des commandes de la Raspberry pour ouvrir la porte.

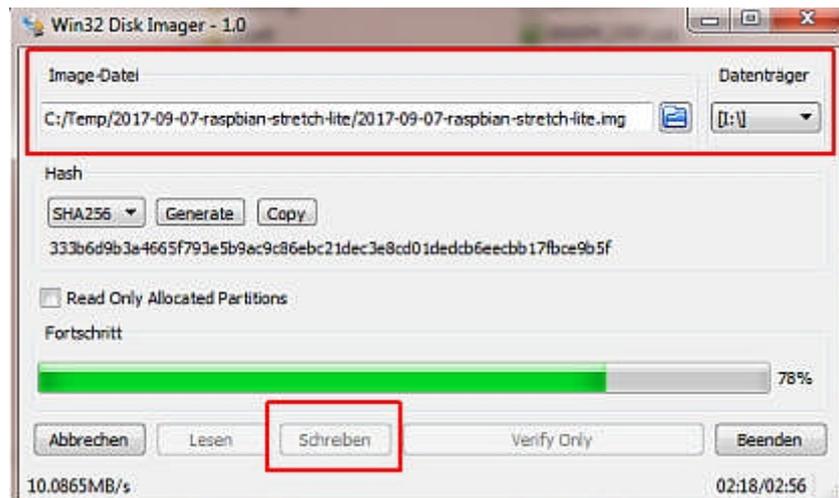
**Webcam** : reçoit des commandes pour prendre des photos et caméra de surveillance.

## IV.3 Installation et configuration

### IV.3.1 Préparation de Raspberry Pi

#### ➤ Préparation de la carte SD sous windows

D'abord, nous avons téléchargé Raspbian à partir du site officiel : **[raspberrypi.org/downloads](http://raspberrypi.org/downloads)**. Puis, nous l'avons installé sur la carte mémoire par le biais d'un logiciel d'écriture d'image : « **Win32 Disk Imager** ». La figure IV.2 présente la phase d'installation de Raspbian avec Win32 Disk Imager.



FigureIV.2 : Installation de Raspbian avec Win32 Disk Imager

### ➤ Branchement des accessoires et alimentation

Nous avons suivi ces étapes pour démarrer notre Raspberry Pi pour la première fois

- ✓ Insertion la carte SD dans son support.
- ✓ Branchement un clavier USB et une souris au Raspberry Pi.
- ✓ Connexion de la sortie HDMI à votre téléviseur ou moniteur.
- ✓ Branchement l'alimentation du Pi.

Si tout se passe correctement, nous verrons une liste de messages défiler à l'écran. Ces messages de log montrent tous les processus du système d'exploitation qui se lancent au démarrage de notre Pi. Nous verrons l'interface réseau s'initialiser et les différents périphériques seront détectés.

### ➤ Attribuer une adresse IP pour la Raspberry Pi

L'adresse IP est configurée dans le fichier `/etc/network/interfaces` et nous aurons besoin de modifier ce fichier pour la configuration du dynamique à statique. On tapera commande suivante :

```
sudo nano /etc/network/interfaces
```

Puis on aura le fichier en console et on peut l'éditer :

D'abord pour auto wlan0 elle existe déjà donc il suffit de lui enlever le #

On met aussi un # au début pour la phrase **allow-hotplug wlan0** pour la mettre en commentaire donc elle devienne sans effet, on ajoute la configuration suivante:

```
auto wlan0
```

```
iface wlan0 inet static
```

```
address 192.168.43.10 (l'adresse IP de la Rasperry)
```

```
netmask 255.255.255.0 (le masque sous réseau)
```

```
gateway 192.168.43.1 (l'adresse de la passerelle)
```

La figure IV.3 illustre le fichier de configuration

```
iface eth0 inet manual

#allow-hotplug wlan0
auto wlan0
iface wlan0 inet static
    address 192.168.43.10
    netmask 255.255.255.0
    gateway 192.168.43.1
    dns-search google.com
    dns-nameservers 8.8.8.8 8.8.4.4
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

FigureIV.3 :configuration de l'adresse IP statique .

### ➤ Configuration wifi

Nous allons configurer notre Raspberry pour qu'il se connecte à un réseau wifi pour quel sera accessible depuis l'extérieur avec des logiciels tel que : puTTY ,VNC

Nous modifions le fichier de configuration en faisant :

**sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf**

ensuite nous ajoutons les paramètres de notre réseau , son nom (ssid) et la mot de passe (psk)

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="non ne touche pas"
    psk="lynda1234"
    key_mgmt=WPA-PSK
}
```

Figure IV.4: configuration des paramètres wifi

## ➤ Accès à distance au Raspberry Pi

Alors à cette étape nous avons donné à notre carte une adresse IP fixe qui va nous servir à la manipulation de la carte à distance via le protocole SSH et VNC. Cette adresse sera également utilisée pour :

### ✚ Connexion à distance via SSH

Secure Shell est un protocole de réseau crypté pour initier des sessions shell textuelles sur des machines distantes de manière sécurisée. Cela permet à un utilisateur d'exécuter des commandes sur l'invite de commande d'une machine sans qu'il soit physiquement présent à proximité de la machine. SSH a été créé en 1995 pour le principal but de permettre la prise de contrôle à distance d'une machine à travers une interface en lignes de commande.

Pour faire le contrôle à distance sur le Raspberry Pi on doit installer le SSH du côté serveur sur notre Raspberry pi et d'un autre client SSH sur notre ordinateur. Le serveur SSH est déjà installé par défaut et activé par défaut sur la Raspberry Pi, ce qui reste seulement d'installer un client SSH sur notre ordinateur ; pour cela on va installer le logiciel Putty sous Windows qui est un client SSH. La figure IV.5 présente la fenêtre de démarrage de Putty.

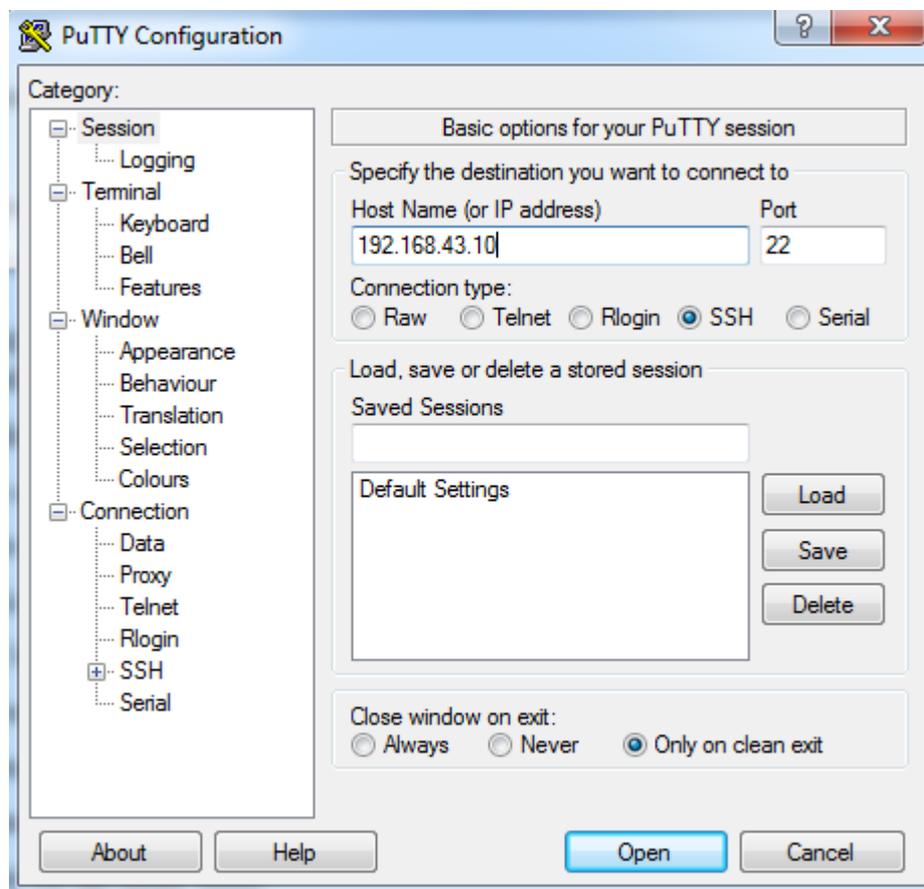
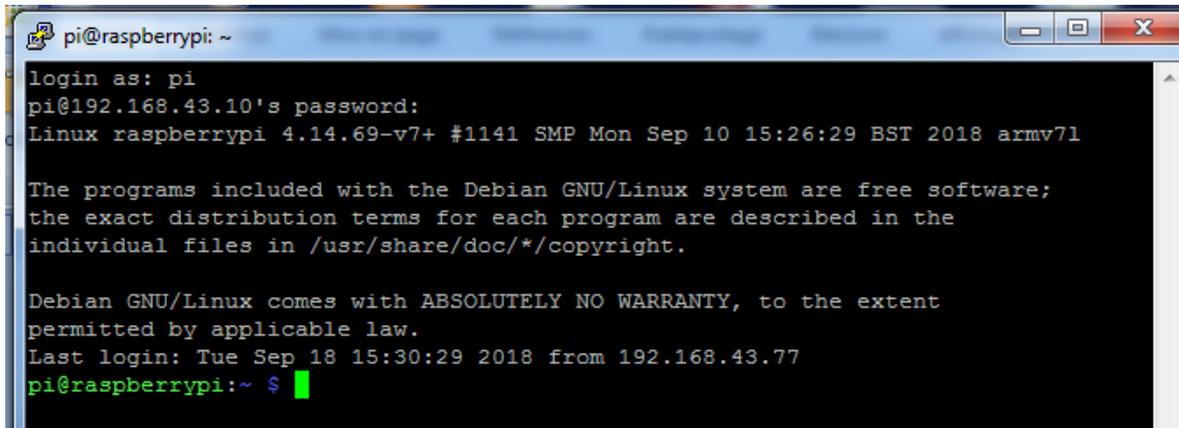


Figure IV.5 : Logiciel Putty.

Après son installation on entre l'adresse IP de la carte puis on clique sur « Open », et après l'authentification (en entrant le nom d'utilisateur pi et notre mot de passe), on se connectera directement sur l'invité des commandes de notre système d'exploitation Raspbian comme l'indique la figure IV.6 :



```
pi@raspberrypi: ~
login as: pi
pi@192.168.43.10's password:
Linux raspberrypi 4.14.69-v7+ #1141 SMP Mon Sep 10 15:26:29 BST 2018 armv7l

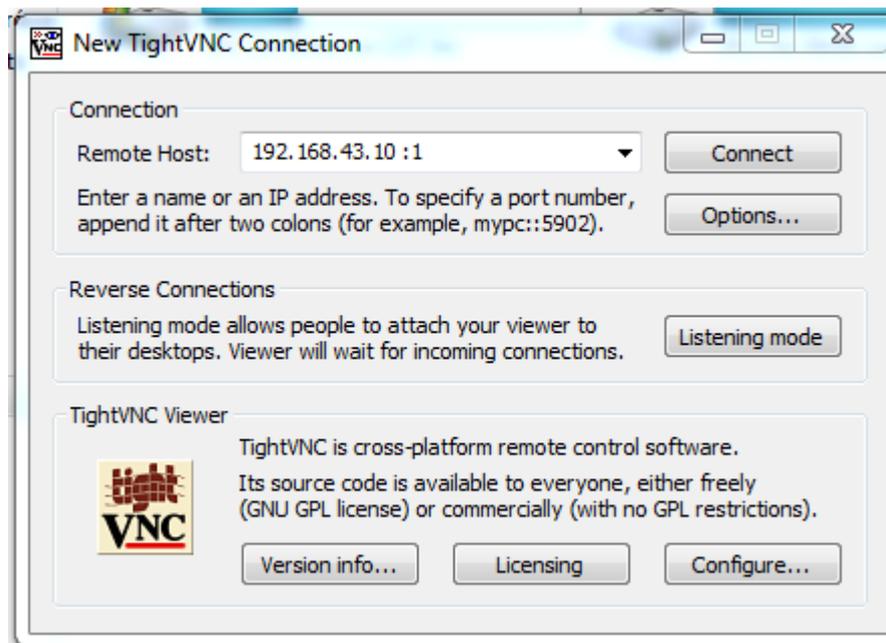
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Sep 18 15:30:29 2018 from 192.168.43.77
pi@raspberrypi:~ $
```

**Figure IV.6 : Invite de commande via SSH**

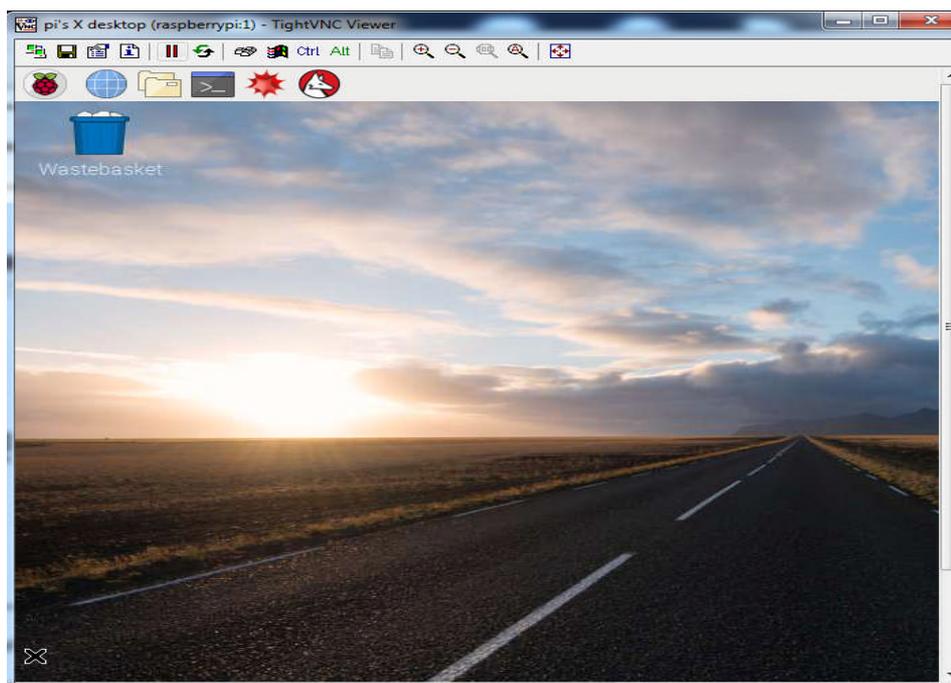
#### **+ Connexion à distance via VNC**

Virtual Network Computing est un système de visualisation et de contrôle 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 qui est un protocole simple pour l'accès à distance aux interfaces graphiques des utilisateurs. Comme pour le SSH, le contrôle à distance sur le Raspberry Pi nécessite une installation du VNC du côté d'un serveur sur notre Raspberry Pi et d'un autre d'un client VNC sur notre ordinateur, la distribution de Dexter Industries « Raspbian For Robots » contient déjà un serveur VNC, il ne reste seulement que d'installer un client VNC sur notre ordinateur, le logiciel qui va jouer le rôle d'un client VNC est TightVNC VIEWER. La connexion est très simple en entrant l'adresse IP suivie de : 1 indiquant le port de l'interface graphique, puis en cliquant sur « Connect » comme montre la figure IV.7:



**Figure IV.7: Logiciel TightVNC VIEWER**

Après l'authentification (en entrant notre mot de passe) le logiciel TightVNC VIEWER affiche l'interface graphique du système d'exploitation (voir figure IV.8), donc à ce moment on a tous les droits et toute la liberté d'utiliser l'interface graphique de notre système d'exploitation sur notre ordinateur en bénéficiant du clavier, souris et écrans de notre propre ordinateur, comme si le Raspberry est branchée à un écran par le câble HDMI et utilise un clavier et souris via les ports USB.



**Figure IV.8 : Interface graphique sur logiciel TightVNC VIEWER**

## IV3.2 Installer le serveur web sur la Raspberry Pi

Pour se faire on doit installer les modules suivants : Apache2, php7, MySQL

### ➤ Installation du serveur Apache :

Avant d'installer le serveur on met à jour le Raspbian avec les commandes suivantes :

**apt-get update**

**apt-get upgrade**

Une fois Raspbian mis à jour, nous allons installer le serveur Apache, avec la commande suivante :

**sudo apt install apache2**

Au passage, nous allons en profiter pour donner des droits au dossier d'apache qui vous permettra de facilement administrer les sites. Pour cela, on lance les commandes suivantes :

**sudo chown -R pi:www-data /var/www/html/**

**sudo chmod -R 770 /var/www/html/**

Une fois l'installation terminée, nous pouvons tester qu'Apache fonctionne correctement en Nous rendant l'adresse de la Raspberry. Pour cela, il faut tenter d'accéder à la Raspberry depuis le port 80 (ce port n'étant pas encore ouvert depuis l'extérieur, il faudra le faire depuis la Raspberry elle-même :

**wget -O verif\_apache.html <http://127.0.0.1>**

### ➤ Installation de PHP

On Install PHP avec la commande suivante :

**sudo apt install php php-mbstring**

Pour savoir si PHP fonctionne correctement, ce n'est pas très compliqué, et la méthode est relativement proche de celle employée pour Apache.

Nous allons en premier lieu supprimer le fichier « **index.html** » dans le répertoire « /var/www/html ».

**sudo rm /var/www/html/index.html**

Puis on a créé un fichier « **index.php** » dans ce répertoire, avec cette ligne de commande

**echo "<?php phpinfo(); ?>" > /var/www/html/index.php**

Et on saisie l'adresse : [192.168.43.10](http://192.168.43.10) dans notre navigateur web il faut avoir le résultat illustré dans la figure V.9

PHP Version 7.0.30-0+deb9u1	
System	Linux raspberrypi 4.14.69-v7+ #1141 SMP Mon Sep 10 15:26:29 BST 2018 armv7l
Build Date	Jun 14 2018 13:50:25
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/15-xml.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-dom.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mbstring.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-simplexml.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini, /etc/php/7.0/apache2/conf.d/20-wddx.ini, /etc/php/7.0/apache2/conf.d/20-xmlreader.ini, /etc/php/7.0/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.0/apache2/conf.d/20-xsl.ini
PHP API	20151012
PHP Extension	20151012

Figure IV.9 : Tableau généré par la commande `phpinfo` sur une raspberry.

### ➤ Installation de MYSQL

On install MYSQL avec la ligne de commande suivante :

```
sudo apt install mysql-server php-mysql
```

Pour vérifier le fonctionnement de MySQL, cette fois nous utiliserons uniquement la ligne de commande. Pour cela, nous allons simplement nous connecter via la commande :

```
sudo mysql --user=root
```

### ➤ Installation de PHPMyAdmin

On Install MYSQL avec la ligne de commande suivante :

```
sudo apt install phpmyadmin
```

Pour vérifier le bon fonctionnement de PHPMyAdmin, nous allons simplement tenter d'y accéder, en utilisant l'adresse de notre Raspberry suivi de `/phpmyadmin`.

## IV.3.3 préparation de la webacam (caméra USB)

Nous avons utilisé la caméra pour deux fonctions :

### IV3.3.1 capture de photo

➤ **Installer le paquet fswebcam pour capturer des images**

On tape la ligne de commande suivante :

```
sudo apt-get install fswebcam
```

on créer un dossier pour enregistrer des photos capturer

```
mkdir webcam
```

on ajoute un script qui permet de récupérer les photo selon la date et l'heure

### IV3.3.2 vidéosurveillance

En on va lancer une mise à jour pour s'assurer qu'on est entrain d'installer la dernière version de motion, on tape la commande suivante :

```
sudo apt-get update -y && sudo apt-get upgrade -y
```

On lance après l'installation de motion

```
sudo apt-get install motion -y
```

On passé ensuite à la modification du fichier de configuration :

```
sudo cp /etc/motion/motion.conf /etc/motion/motion.conf.back
```

Le fichier de configuration contient beaucoup de paramètres, ici on va d'intéresser seulement aux plus importants

Pour commencer à éditer le fichier :

- 1- On commence par mettre un « on » sur « daemon » (voir la figure IV.10)

```

pi@Raspberry: ~
GNU nano 2.7.4      Fichier : /etc/motion/motion.conf      Modifié
# Rename this distribution example file to motion.conf
#
# This config file was generated by motion 4.0
#####
# Daemon
#####
# Start in daemon (background) mode and release terminal (default: off)
daemon on
# File to store the picture ID - also called pid file - (default: not defined)

```

Figure IV.10: activation du Daemon.

- 2- On désactive la capture de la vidéo `ffmpeg_output_movies` : « on » pour la capture « off » pour la désactiver (voir la figure IV.11)

```

# Type of output images
# Valid values: jpeg, ppm (default: jpeg)
picture_type jpeg
#####
# FFMPEG related options
# Film (movies) file output, and deinterlacing of the video input
# The options movie_filename and timelapse_filename are also used
# by the ffmpeg feature
#####
# Use ffmpeg to encode movies in realtime (default: off)
ffmpeg_output_movies on
# Use ffmpeg to make movies with only the pixels moving
# object (ghost images) (default: off)
ffmpeg_output_debug_movies off

```

Figure IV.11: désactivation la capture de la vidéo.

- 3- Désactiver la capture de photo et utiliser juste la vidéosurveillance : **output\_picture off**
- 4- On autorise la lecture du flux depuis internet pour permettre l'accès en ligne à la vidéosurveillance : **stream\_localhost off** : `stream_localhost off` (voir la figure IV.12).

```
stream_maxrate 1
# Restrict stream connections to localhost only (default: on)
stream_localhost off

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual stream rate by desired number of frames
# Actual stream rate is the smallest of the numbers framerate and stream_maxrate
stream_limit 0
```

**Figure IV.12: configuration de la caméra.**

- 5- On démarre motion (vidéo streaming) au démarrage de la carte : on va utiliser une tâche cron de l'utilisateur root .

**Sudo crontab -e**

Ensuite à la fin du fichier, on ajoute la ligne suivante :

```
@reboot motion > /dev/null 2> &1
```

- 6- Par défaut, Motion diffuse la vidéo sur le port 8081 : Pour accéder à la surveillance vidéo de notre Raspberry Pi en local host, on tape l'adresse **http : //192.168.43.10:8081** .

Dans notre application, on a transformé la Raspberry pi qui gère la caméra avec motion en serveur web (page administrateur) contient le flux vidéo.

## IV.4 Réalisation matérielle :

Nous présentons dans cette partie les schémas de câblages de nos modules .

### ➤ Branchement du moteur pas à pas

On branche les GPIOs de la Raspberry aux pins du driver UNL2003 :

La pin1 du driver est branché au pin 15 du Raspberry pi

La pin2 du driver est branché au pin 29 du Raspberry pi

La pin3 du driver est branché au pin 30 du Raspberry pi

La pin4 du driver est branché au pin 31 du Raspberry pi

L'alimentation (5v) est branchée au pin 4 de la Raspberry pi

La masse (GND) au pin 9 de la Raspberry pi

Le câblage est illustré dans la figure IV.13

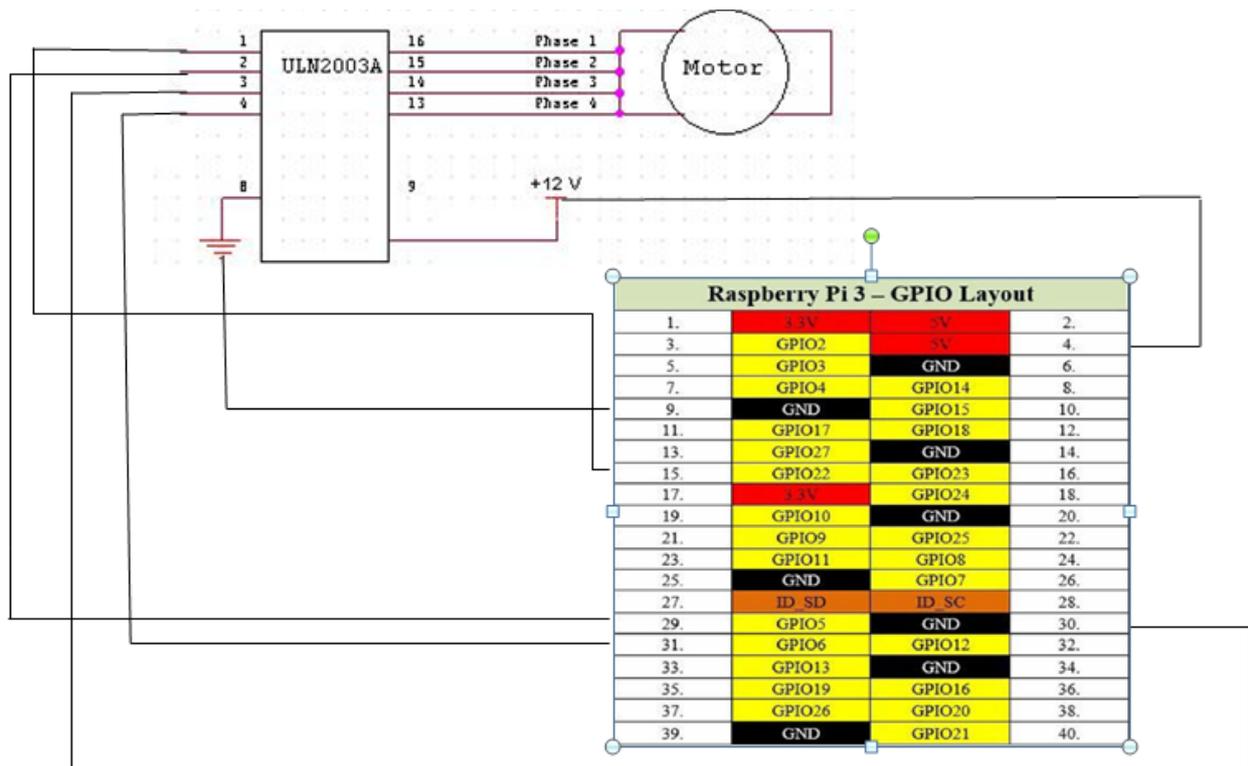


Figure IV.13: Schéma de brochage entre le driver UNL2003 et la Raspberry Pi

- Réalisation d'une maquette pour faire une démonstration de notre projet, la figure IV.14 présente cette maquette avec le matériel utilisé.



Figure IV.14 : les branchements des composants de notre système

## IV.5 Présentation de l'application

### IV.5.1 interface utilisateur

#### ➤ Accéder au serveur :

Pour accéder à notre à notre application, il faut se connecter au même réseau local puis entrer l'adresse IP de la Raspberry Pi dans l'URL suivi du nom de fichier .Dans notre cas:

192.166.43.10/smartphone

Une page d'authentification s'affiche comme suit :

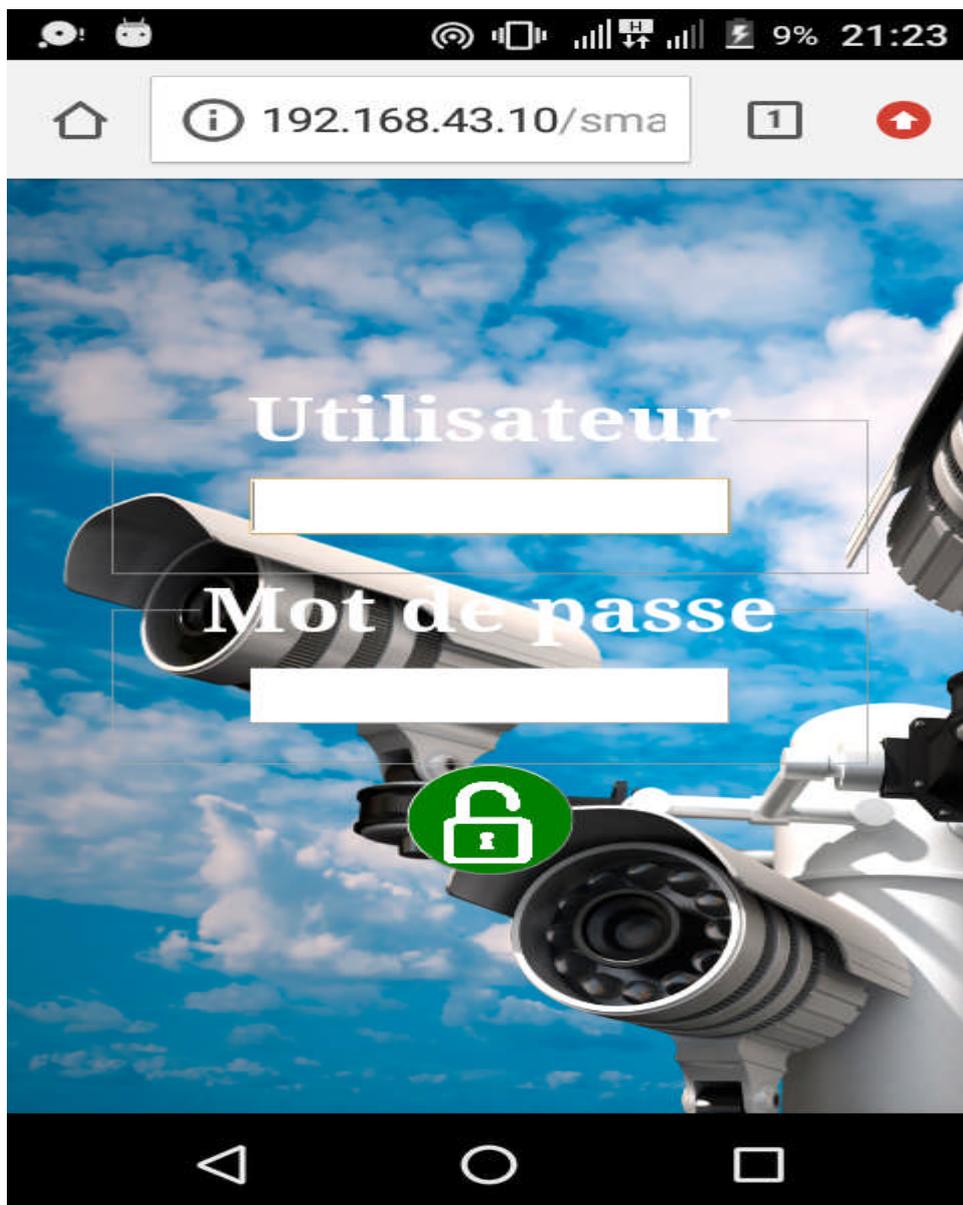


Figure IV.15 : Page d'authentification d'utilisateur.

## IV.5.2 Interface administrateur

### ➤ Accéder au serveur

Pour que l'administrateur peut accéder à son espace il faut entrer l'adresse IP dans L'url suivie du nom du fichier .192.168.43.10/smartphone/admin.

The image shows a login interface for an administrator. At the top, there is a large, blue, 3D-style stamp that says 'ADMINISTRATION'. Below the stamp, the text 'Mot de passe :' is centered. Underneath this text is a rectangular input field for the password. At the bottom center of the form is a blue circular button with the word 'ACCEDE' written in white capital letters.

Figure IV.16 : page d'authentification du l'administrateur.

### ➤ Page historique

Après que l'administrateur saisie son mot de passe, la page historique qui apparait, cette page contient trois parties une pour consulter ajouter et supprimer les utilisateurs, la deuxième pour consulter les images capté dans quelle jours et quelle heure lors de l'accès des utilisateurs et la troisième présente une vidéo surveillance en streaming envoyé par la caméra, à partir d'elle l'utilisateur peut surveiller la zone de son bureau. Les figure IV.17 ,IV18 ,IV19 illustrent ces différentes parties



Figure IV.17 : partie ajout et suppression des utilisateurs

# Video Surveillance

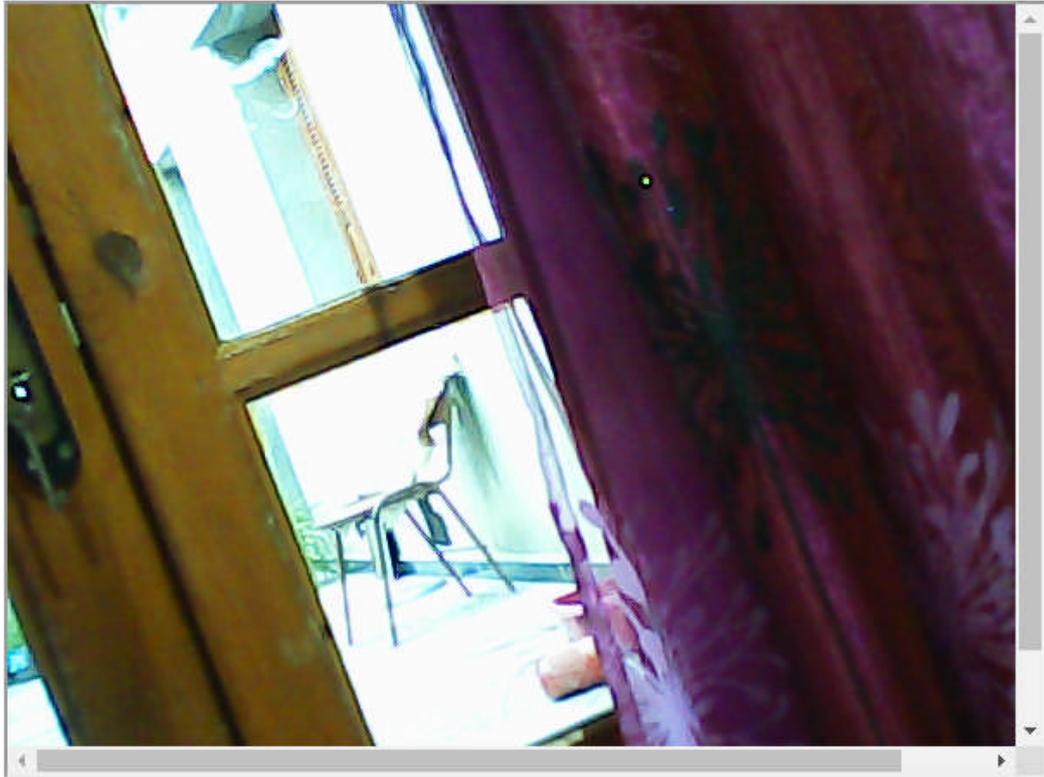


Figure IV.18 : partie vidéosurveillance

## Photos

 <a href="#">2018-09-17_2105.jpg</a>	2018-09-19 10:10 317K
 <a href="#">2018-09-17_2106.jpg</a>	2018-09-19 10:10 317K
 <a href="#">2018-09-18_0148.jpg</a>	2018-09-19 10:10 300K
 <a href="#">2018-09-18_1534.jpg</a>	2018-09-19 10:10 295K
 <a href="#">2018-09-19_0952.jpg</a>	2018-09-19 10:10 286K
 <a href="#">2018-09-19_0953.jpg</a>	2018-09-19 10:10 287K
 <a href="#">2018-09-20_2116.jpg</a>	2018-09-20 21:16 317K

Figure IV.19 : partie photos

Ici dans la figure IV.19 l'administrateur peut consulter les photos générées par la caméra selon la date et l'heure en cliquant sur la photo pour la visionner.

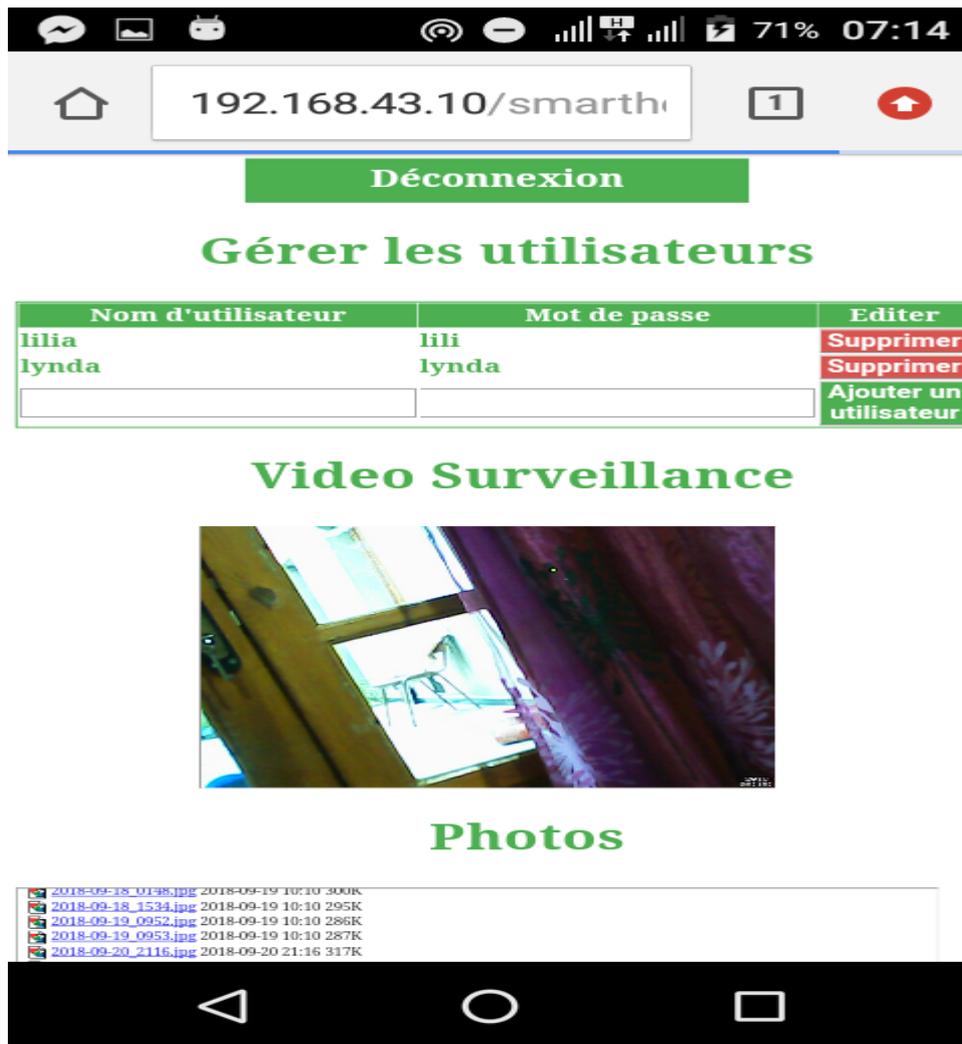


Figure IV.20: l'interface de l'administrateur via un smartphone

## IV.6 Conclusion

Au cours de ce de dernier chapitre, nous avons décrit les étapes de réalisation et de test de notre système. En premier lieu, nous avons présenté les outils de développement utilisé ainsi que les configurations matérielles et les brochages nécessaires à sa réalisation.

Ensuite, dans la deuxième partie, nous avons présenté quelques tests de notre application ainsi que les résultats obtenus illustrés par quelques figures.

Comme nous l'avons mentionné, notre système offre à l'administrateur de l'application les fonctionnalités nécessaires pour gérer les accès à la zone qu'il veut sécuriser .En ayant les résultats attendus, on peut conclure que le système mise en place répond bien au besoin de contrôle d'accès.

## Conclusion général

L'objectif du présent du travail étant de concevoir et de développer un système permettant de contrôler l'accès à des zones particulière ainsi que de les surveiller .L'utilisation d'un nano ordinateur comme composant de base pour concevoir notre système rend l'échange de l'information rapide avec les deux autres composants (caméra et moteur), ainsi que elle est utilisée comme un serveur web rend le système mobile et facile.

Afin de mieux cerner et comprendre les caractéristiques du matériel utilisé, nous avons présenté dans le premier chapitre les généralités sur les systèmes embarqués, leur architecture, leurs types. La carte Raspberry et ses modèles ainsi que le contrôle d'accès ont été introduits au deuxième chapitre.

Pour développer notre système, nous avons suivi un cycle d'analyse, conception, implémentation, tests et déploiement. Les deux premières phases ont fait objet du troisième chapitre dans lequel les deux parties du système (logicielle et matérielle) ont été traitées .le quatrième chapitre a donc porté sur les deniers phases l'implémentation et les tests.

Au terme de ce travail élaboré dans le cadre de notre projet de fin d'études , nous considérons ce projet nous été bien bénéfique vu qu'il nous a permis de consolider nos connaissance vers le développement d'un système embarqué qui sera utile dans le domaine de contrôle d'accès .en effet ,l'apport de notre projet se résume surtout dans la découverte et la familiarisation avec les techniques de développement qui nous ont permis d'améliorer nos compétences et nos acquis en ce qui concerne linux embarqué et la programmation

## Perspectives

- Pour la suite de notre travail, nous avons comme perspectives : l'alimentation du matériel utilisés notamment le remplacement de moteur pas à pas par un autre qui est plus puissant (le moteur pas à pas de 5v est utilisé juste pour la démonstration).
- Utilisation d'une alarme en cas de tentative d'intrusion.
- Automatiser la fermeture de la porte

## ***Bibliographie***

- [17]: Raspberry PI Robotics Projects , Work through a mix of amazing robotic projects using the Raspberry Pi Zero or the Raspberry Pi 3 ,RICHARD GRIMMET , livre ,2016
- [20]: RASPBERRY PI3 , *ENCHANTED GUIDE FOR STARTERS*,livre,2017 THOMAS CHARLESTON
- [28]: Getting Started with Raspberry pi3, Agus Kurniawan,2016
- [29]: Patrice Moteurs pas Aa pas et PC ,Patrice Oguic ,2004
- [1]:<https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.html>.
- [2] : [https://www.memoireonline.com/05/12/5830/m\\_Les-systemes-embarques4.html](https://www.memoireonline.com/05/12/5830/m_Les-systemes-embarques4.html).
- [3] :[https://www.fafiec.fr/images/contenu/menuhaut/observatoire/etudes/2013/systemes-embarqu%C3%A9s/SE-Ecosysteme\\_des\\_Systemes\\_Embarques\\_-\\_20140606.pdf](https://www.fafiec.fr/images/contenu/menuhaut/observatoire/etudes/2013/systemes-embarqu%C3%A9s/SE-Ecosysteme_des_Systemes_Embarques_-_20140606.pdf)
- [4] : <https://www.esen.tn/portail/medias/documents/enseignement/1488739733495.pdf>
- [5] : [https://www.memoireonline.com/05/12/5830/m\\_Les-systemes-embarques53.html](https://www.memoireonline.com/05/12/5830/m_Les-systemes-embarques53.html)
- [6] : <https://fr.wikipedia.org/wiki/Processeur>
- [7] : <https://www.universalis.fr/encyclopedie/ordinateurs/2-principes-de-fonctionnement-d-un-processeur/>
- [8] : Synthèse de gestionnaires mémoire pour applications Java temps-réel embarquées Guillaume Salagnac <https://tel.archives-ouvertes.fr/tel-00288426/document>
- [9] : <http://perso.citi.insa-lyon.fr/gsalagnac/publications/salagnac-PhD.pdf>
- [10] : Introduction à la notion de système d'exploitation , [www.Introduction à la notion de système d'exploitation.pdf](http://www.Introduction%20%C3%A0%20la%20notion%20de%20syst%C3%A8me%20d%27exploitation.pdf)
- [11] : [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_d%27exploitation\\_embarqu%C3%A9](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation_embarqu%C3%A9)
- [12] : Systèmes d'exploitation embarqués - ELECINF344 - [rfc1149.net](http://rfc1149.net)

[13] : Système embarqué : Aspect software - UKO ,

<https://www.ukonline.be/cours/embeddedsystems/programming/chapitre1-3>

[14] : <https://fesc.asso.fr/Raspberry-Pi-un-mini-mini-PC>

[15] : <http://raspberrypiorder.com/the-history-of-raspberry-pi/>

[16] : <https://www.supinfo.com/articles/single/5654-differents-raspberry->

[18] : [https://fr.wikipedia.org/wiki/Raspberry\\_Pi](https://fr.wikipedia.org/wiki/Raspberry_Pi)

[19] : <https://www.gotronic.fr/art-carte-raspberry-pi-zero-wh-27670.htm>

[21] : <http://www.raspberrypi.com/connect.com/operating-system-s/item/114-software-os-arch-linux-arm>

[22] : Comparative Analysis of Different Operating Systems for a Raspberry Pi .pdf ,  
National Conference on Innovative Trends in Science and Engineering

[23]: <https://the-raspberry.com/os-raspberry-pi>

[24]: [http://fr.wikipedia.org/wiki/contrôle\\_d'accès](http://fr.wikipedia.org/wiki/contrôle_d'accès), Articles : contrôle d'accès, Auteurs :  
[http://fr.wikipedia.org/w/index.php?title=Contrôle\\_d'accès &action=history\\_control d acces](http://fr.wikipedia.org/w/index.php?title=Contrôle_d'accès&action=history_control_d_acces)

[25] : BOUHARAOUA Abderrahim, BOUKLI HACENE Mohammed Imad ,  
Automatisation d'une maison intelligente ,mémoire,Université Aboubakr Belkaïd – Tlemcen

[26]: [https://w1.siemens.ch/.../ch/fr/.../08\\_ext\\_online\\_Kataschuere\\_Contrôle\\_d'accès\\_fr.pdf](https://w1.siemens.ch/.../ch/fr/.../08_ext_online_Kataschuere_Contrôle_d'accès_fr.pdf)

[27] : <https://www.companeo.com/securite-electronique/guide/controle-acces-fonctionnalites#0>