

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد- تلمسان

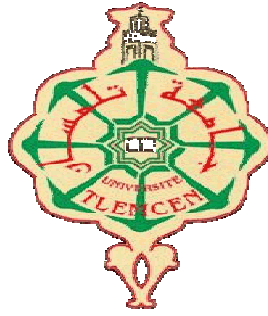
Université Aboubakr Belkaïd-Tlemcen

كلية التكنولوجيا

Faculté de Technologie

Département de Génie Electrique et Electronique (GEE)

Filière : Electronique



MASTER INSTRUMENTATION

PROJET DE FIN D'ETUDES

Présenté par : TALEB Hafsa & SOLTANI Sid Ahmed

Intitulé du Sujet

**Conception d'un système de surveillance intelligent à l'aide  
d'un module A6 GSM/GPRS**

Soutenu en 2019, devant le jury composé de :

M <sup>r</sup> NEMMICHE Ahmed	MAA	Univ. Tlemcen	Président
M <sup>r</sup> MOULAI KHATIR Ahmed Nassim	MCB	Univ. Tlemcen	Encadreur
M <sup>r</sup> MASSOUM Nouredine	MCB	Univ. Tlemcen	Examineur1

Année Universitaire 2018-2019

# Remerciements

*On remercie ALLAH, le tout puissant, qui nous a donné la force, la volonté et surtout le courage pour accomplir ce modeste mémoire.*

*Au terme de ce travail, on tient à remercier particulièrement Dr Moulai Khatir Nassim, pour son encadrement, sa disponibilité, ses critiques et ses remarques pertinentes. Il nous a conseillé et guidé du début du projet à sa fin.*

*Aux membres du jury qui nous honorent de leur présence.*

*On voudrait aussi adresser nos sincères remerciements à tous les enseignants de département GEE Pour leurs enseignements et les cours intéressants qu'ils nous ont prodigué pendant notre cursus à l'Université Abou Bekr Belkaid.*

*Et finalement on tient à remercier chaleureusement et respectivement toute personne qui a participé de près ou de loin à la réalisation de ce projet.*

## ***Dédicaces***

*Tout d'abord, je rends grâce à Dieu le tout puissant de m'avoir donné la foi et la force de mener à terme ce modeste travail que je dédie particulièrement :*

*A mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, mon père que Dieu lui prête longue vie, santé et joie.*

*A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie, mon bonheur, ma mère que j'adore.*

*A ma sœur jumelle Amina qui m'a conseillée, aidée et encouragée durant toute la période de préparation de ce mémoire.*

*A mes frères Zakarya, Walid et Mohieddine qui m'ont été d'un grand soutien aussi bien moral que matériel.*

*Aux familles Taleb et Chenika, du plus grand au plus petit pour leurs encouragements.*

*A mon encadreur Dr Moulai Khatir Nassim pour la qualité de ses orientations, de ses conseils, pour sa patience et sa disponibilité durant toute la période de ce travail.*

*A mes amies et sœurs notamment Malika, Imène, Yasmine, Ikram et Fatima pour leur soutien et leur loyauté envers ma personne.*

*A tous qui m'ont inculqué le savoir du primaire jusqu'à l'université.*

***Taleb Hafsa***

## *Dédicaces*

*À mes chers parents*

*pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études*

*À mes chères frère et sœurs*

*pour leurs encouragements permanents, et leur soutien moral*

*À toute ma famille*

*pour leur soutien tout au long de mon parcours universitaire*

*À tout La famille SOLTANI*

*DALIA*

*BAGHDAD*

*NEDJADI*

*et DJEBBAR*

*À tous mes amis*

*qui m'avez toujours soutenu et encouragé*

*À tous ceux que j'aime et ceux qui m'aime*

*Soltani Sid Ahmed*

## Liste des abréviations

GSM : Global System for Mobile communication)

GPRS : General Packet Radio Service

RPi : Raspberry pi

PNG : Portable Network Graphics

JPEG : acronyme de Joint Photographique Experts Group

USB : Universal Serial Bus

HD : Haute définition

SMTP : Simple Mail Transfer Protocol

PiR : Passive infrared sensor

GND : la masse

Os : Operating System

Gpio : General Purpose Input/Output

IP : Internet Protocol

TCP : Transmission Control Protocol

CCTV : Closed-circuit television

FPGA : Field-Programmable GateArray

SMS : Short Message Service

PC : Personal Computer

LAN : Local Area Network

WLAN : Wireless Local Area Network

LCD : liquide cristal display

Spi : Serial Peripheral Interface

RAM : Mémoire vive (Random Access Memory)

TE : Terminal Equipment

TA : Terminal Adaptator

ME : Mobile Equipment.

# Listes de figures

## **Chapitre I:**

Figure I.1.Système de Vidéosurveillance analogique avec magnétoscopes traditionnels

Figure I.2.Système de vidéosurveillance analogique avec enregistreur numérique

Figure I.3.Système analogique avec enregistreur numérique réseau

Figure I.4.Système de vidéosurveillance sur IP avec serveur vidéo

Figure I.5.Système de vidéosurveillance sur IP avec caméras réseau

## **Chapitre II:**

Figure II.1.Schéma fonctionnel du système de sécurité maison basé sur Arduino [12]

Figure II.2.Système de sécurité à base de Raspberry Pi

FigureII.3.Le Raspberry Pi

Figure II.4.Modèle zéro

FigureII.5.Modèle zéro W

Figure II.6.Modèle zéro WH

Figure II.7.Modèle A

Figure II.8.Modèle A+

Figure II.9.Modèle B1

Figure II.10.Modèle B1+

Figure II.11.Modèle B2

Figure II.12.Modèle B3

Figure II.13.Modèle A3+

Figure II.14.Modèle B3+

Figure II.15.Les composants standard d'un Raspberry P i B3

Figure II.16.Alimentation 2.5 A -- 5V

Figure II.17.Connexion des Les accessoires avec le Raspberry

FigureII.18.Interface du SDFormatter

Figure II.19.Interface du logiciel Etche

Figure II.20.Liste des distributions de NOOBS

Figure II.21.Interface de Putty Configuration

Figure II.22.Slogan de langage Python

Figure II.23.Le port GPIO.BCM

Figure II.24.Carte Arduino UNO

Figure II.25.Brochage interne d'une carte Arduino UNO

Figure II.26.Brochage externe d'Arduino UNO

Figure II.27.le microcontrôleur d'Arduino UNO

Figure II.28.les pins du microcontrôleur d'un Arduino UNO

Figure II.29.Slogan du logiciel Arduino

Figure II.30.Module GSM A6

Figure II.31.Fonctionnement d'une commande AT

Figure II.32.Capteur PIR

Figure II.33.Fonctionnement d'un capteur PIR

Figure II.34.L'afficheur I2C LCD

Figure II.35.Sac à dos LCD I2C

Figure II.36.Adresses I2C

### **Chapitre III:**

Figure III.1.Camera USB (Webcam)

Figure III.2.Capture par fswebcam

Figure III.3.Image capturée et sauvegardée

Figure III.4.Câblage du Capteur PIR avec Raspberry Pi

Figure III.5.Initialisation et configuration du système

Figure III.6.Câblage d'afficheur I2C avec Raspberry Pi

Figure III.7.Connexion du module A6 GSM avec Arduino

Figure III.8.Organigramme du système

Figure III.9.Circuit globale du système

Figure III.10.Boîtier du système de sécurité

Figure III.11.Boîtier du système de sécurité

Figure III.12.Réception de l'appel

Figure III.13.Réception de l'email



# Table des matières

<b>Introduction général.....</b>	<b>5</b>
<b>Chapitre I : Généralités sur la vidéosurveillance.....</b>	<b>7</b>
I.1.Pourquoi la Vidéosurveillance ?.....	8
I.2.Présentation.....	8
I.3.Apparition des systèmes de vidéosurveillance.....	8
I.4.Evolution des systèmes de vidéosurveillance.....	9
I.4.1. Système de Vidéosurveillance analogique avec magnétoscopes traditionnels.....	9
I.4.2. Système de vidéosurveillance analogique avec enregistreur numérique .....	10
I.4.3. Système de vidéosurveillance analogique avec enregistreur numérique réseau.....	10
I.4.4. Système de vidéosurveillance sur IP avec serveur vidéo.....	11
I.4.5. Système de vidéosurveillance sur IP avec une caméra réseau.....	11
I.4.6. Réseau sans fil.....	12
I.5.Problématique.....	12
I.6.Objectif du projet.....	13
I.7.Justification.....	14
<b>Chapitre II : Les systèmes de sécurité intelligents.....</b>	<b>15</b>
II.1. Historiques des systèmes de sécurité à base d'un Raspberry Pi.....	16
II.2. Evolution de la sécurité.....	17
II.3. Définition du système de sécurité.....	18
II.4. Technologies de sécurité actuelle.....	18
II.4.1 Système de sécurité maison basé sur Arduino.....	18
II.4.1.1.Mode de fonctionnemt .....	19
II.4.2. Les systèmes CCTV (Closed-Circuit Television).....	19
II.4.3. les Système IP de surveillance à distance.....	20
II.4.4. Système de surveillance à base de Raspberry Pi.....	20
II.4.4.1. Pourquoi des systèmes de sécurité maisons à base de Raspberry Pi?.....	21
II.5. Les modules utilisés.....	22
II.5.1. Le Raspberry Pi.....	22
II.5.1.1. Les modèles Raspberry Pi.....	22

Le Modèle Zéro.....	22
Le Modèle A.....	23
Le Modèle B.....	23
II.5.1.2. Le Raspberry Pi type B3.....	25
A. Les composants du RPi B+.....	25
B. connexion des composants d'un Raspberry Pi.....	27
Les accessoires indispensables (ou très utiles) pour un Raspberry Pi.....	27
C. Les différents systèmes d'exploitation pour Raspberry Pi .....	28
D. Premier démarrage, installer Raspbian.....	29
E. Éteindre le Raspberry Pi.....	35
F. Programmation du RPi B+.....	35
II.5.2. La Carte Arduino.....	37
II.5.2.1. Les différents carte Arduino .....	37
II.5.2.2. La carte Arduino Uno.....	37
II.5.2.2.1. programmation de la carte Arduino Uno.....	41
II.5.3. Module GSM A6.....	41
II.5.3.1. Généralités sur les commandes AT.....	42
II.5.3.2. Schéma de fonctionnement.....	43
II.5.3.3. Commandes de base.....	43
II.5.3.4. Commandes de gestion d'un appel.....	44
II.5.3.5. Les options de la commande D.....	44
II.5.4. Capteur de mouvement PIR.....	44
II.5.4.1. Principe de fonctionnement du capteur PIR.....	45
II.5.5. L'afficheur I2C LCD.....	45
II.5.5.1. Caractéristiques d'un I2C LCD.....	46
II.5.5.2. Pourquoi choisir I2C dans notre projet.....	46
II.5.5.3. Sac à dos LCD I2C.....	47
II.6. Conclusion.....	48
<b>Chapitre III. Conception et Analyse .....</b>	<b>49</b>
Introduction.....	50
III.1. Matériel de conception (Installation et configuration des modules du système) .....	50

III.2. Mettre à jour le système de la Raspberry Pi.....	50
III.3. Activer la camera USB (webcam) .....	50
III.3.1. Serveur webcam avec fswebcam.....	51
III.4. Configuration du capteur de mouvement PIR.....	53
III.5. Connexion du LCD I2C avec Raspberry Pi.....	54
III.5.1. Activer l'I2C sur le Pi.....	54
III.5.2. Connexion I2C Raspberry Pi.....	56
III.6. Connexion du Module A6 GSM avec Arduino.....	58
III.7. La génération et l'envoi du courrier électronique.....	60
III.8 Conception du système.....	62
III.8.1. Organigramme du système de sécurité basé sur Raspberry Pi.....	62
III.8.2. Développer un code source complet.....	64
III. 9. Conclusion.....	67
<b>Conclusion générale.....</b>	<b>69</b>
<b>Références Bibliographique.....</b>	<b>70</b>

### **Introduction générale :**

De nos jours, Le besoin des systèmes de sécurité intelligents pour la protection des lieux et des personnes est en augmentation rapide. En effet, l'évolution des technologies analogiques vers le numérique pour les installations de vidéosurveillance révolutionne le marché mondial des applications vidéo pour la surveillance des logements, lieux de travail, établissements, villes, routes... etc. qui sont devenus aujourd'hui des espaces de vie intelligents,

La vidéosurveillance est l'une des solutions de sécurité les plus anciennes et répandues, vue quelle présente divers avantages comme la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers dans un monde où l'homme s'absente régulièrement de son foyer ou de son lieux de travail pendant des heures ou même des jours, par un effet dissuasif sur les voleurs et les vandales en fournissant des preuves de criminalité et de responsabilité

L'apparition des caméras IP a amorcé le passage de la technologie analogique par télévision en circuit fermé CCTV (Closed-Circuit TeleVision) vers les caméras numériques qui fonctionnent par le biais d'un réseau informatique à l'aide d'un câbles ou par un réseau wifi, et qui permettent l'installation des réseaux de vidéosurveillance avec un grand nombre de caméras, ces réseaux sont indispensables dans les lieux publics où des centaines de caméras de surveillance peuvent être déployées

La majorité de ces systèmes comptent beaucoup sur la présence humaine pour surveiller les scènes et déceler les comportements ou les événements suspects, mais malheureusement, la capacité de la surveillance humaine présente certaines limitations, en raison des heures continues de surveillance. Pour cela les systèmes à caméras numériques IP sont conçus intelligemment dans le but de répondre à tous les besoins futurs sans s'appuyer sur l'opérateur humain

La disponibilité des microcontrôleurs (PIC, ARM, AVR, ...) et les systèmes embarqués à faible coût comme Arduino, Raspberry PI, FPGA, BeagleBone, avec leurs propres plateformes Open Source, est une cause importante des incroyables innovations et inventions électroniques qui ont changé nos vies, en fournissant un large espace de développement avec des possibilités d'implémentation très vastes et une diversité exceptionnelle.

Dans cette thèse de Master, on va proposer une architecture d'un système intelligent sous la forme d'une box domotique à faible coût, qui présente comme éléments intelligents : une

## Introduction Générale

---

carte Raspberry Pi 3, une cartes Arduino UNO, une carte mini GSM /GPRS basée sur le module GPRS A6, un capteur de mouvement PIR et une simple webcam pour réaliser un système de vidéosurveillance assurant la détection du mouvement ainsi que l'envoi d'alerte en même temps par SMS ou appel téléphonique et par un email avec la photo de l'intrus en pièce jointe.

Ce mémoire est formé de trois chapitres, le premier est une présentation d'un état de l'art sur la technologie actuelle des systèmes de vidéosurveillance, et comme solution on propose et on décrit notre système à faible coût dans un deuxième chapitre ainsi que les différentes composantes utilisées dans ce projet. Le troisième chapitre est dédié à la présentation des étapes de réalisation du système, ses codes sources, ses modes de fonctionnements, ainsi que la mise en place du boîtier réalisé.

# **Chapitre I:**

# **Généralités sur la**

# **Vidéosurveillance**

## **Chapitre I : Généralités sur la Vidéosurveillance**

### **I.1 Pourquoi la Vidéosurveillance ?**

La vidéosurveillance (terme officiel : vidéo protection) est un système composé de caméras et de transmission d'images, disposé dans un espace public ou privé afin de pouvoir le surveiller à distance (télésurveillance). Les images transmises par ce système, peuvent être traitées, visionnées puis archivées ou détruites.

L'installation d'un system de vidéosurveillance passe par une analyse très précise afin de répondre aux exigences et aux besoins de l'utilisateur.

### **I.2 Présentation**

En raison du besoin de renforcer les niveaux de sécurité dans différents domaines d'activité; un des moyens les plus utilisé est la vidéosurveillance qui consiste à recevoir un flux vidéo a partir d'un certaine nombre de caméras de surveillance sur un PC localement ou à distance.

Les caméras font leur apparition au début des années 2000, remarquablement dans de nombreuses villes européennes, notamment en Angleterre où Londres est la ville la plus surveillée du monde avec une armada de caméras de vidéosurveillance installée depuis le début des années 1990

### **I.3. Apparition des systèmes de vidéosurveillance**

La vidéosurveillance a fait son apparition d'abord au Royaume Uni, suite aux attaques de l'IRA (Armée républicaine irlandaise en anglais Irish Republican Army) dans les années 1970 et 1980, ces premières expériences ont conduit à des programmes de grande ampleur au début des années 1990. Ce succès à poussé le gouvernement à faire une campagne auprès de la population, et a décidé de lancer une série d'installations de caméras partout dans les villes. Actuellement, les caméras au Royaume-Uni couvrent une grande partie des centres villes, et de nombreuses gares et parkings, le chiffre est approximativement à 400000 caméras à Londres et 4 millions au Royaume-Uni au total.

Ailleurs, la France a aussi installé des systèmes de vidéosurveillance. Le nombre était estimé en 1998 à un million de cameras dont 150000 dans divers lieux du le domaine public tels que les routes, les aéroports, les gares, les transports publics. Ces installations vidéo commencent aussi à fleurir dans plusieurs villes.

Un sondage dans différents pays a présenté 71% d'avis favorables à l'installation d'un tel système dans les parkings. Cependant il existe aussi des associations qui militent contre toute forme de surveillance.

## I.4. Évolution des systèmes de vidéosurveillance

Les systèmes de vidéosurveillance existent depuis environ 25 ans. Les premières caméras utilisées en vidéosurveillance avaient des images de basse qualité et noir et blanc, et qui ne peuvent pas être zoomées, ni changées en angle de vue, contrairement aux caméras modernes qui sont en couleur et plus performantes, permettent ainsi une mise au point très nette, et des dispositifs d'enregistrement et d'analyse plus précis et plus efficaces.

Aujourd'hui, les caméras réseau sont utilisées et les serveurs informatiques pour l'enregistrement vidéo dans un système entièrement numérique qui offre la possibilité de détecter et de suivre des mouvements en pilotant ces caméras avec des ordinateurs,

### I.4.1. Système de Vidéosurveillance analogique avec magnétoscopes traditionnels

Ce système ancien de vidéosurveillance est encore actuellement le plus utilisé dans les entreprises, commerces et habitations de particuliers.

Il est équipé d'une ou plusieurs caméras analogiques qui capturent des images et les envoient à un enregistreur à durée limitée (type magnéscope) par le biais d'un signal analogique, d'un moniteur (ou téléviseur) et d'un câblage (le transfert d'images se fait via un câble dit coaxial).

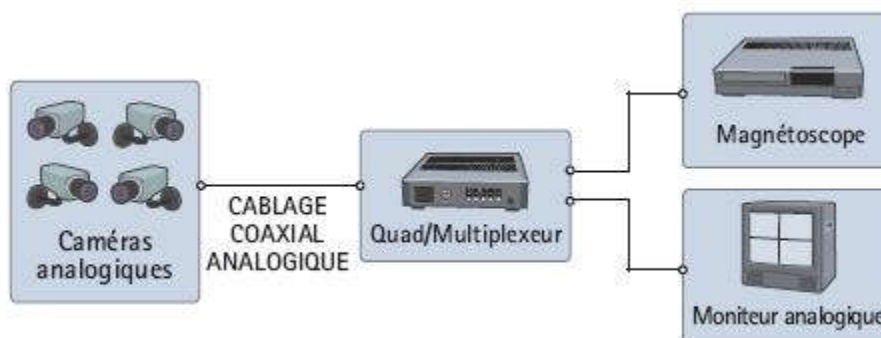


Figure I.1. Système de Vidéosurveillance analogique avec magnétoscopes traditionnels



Ces systèmes analogiques présentent généralement, un multiplexeur connecté entre la caméra et le moniteur, qui permet de visualiser et d'enregistrer le contenu de plusieurs caméras sur un même moniteur et magnétoscope.

#### I.4.2. Système de vidéosurveillance analogique avec enregistreur numérique

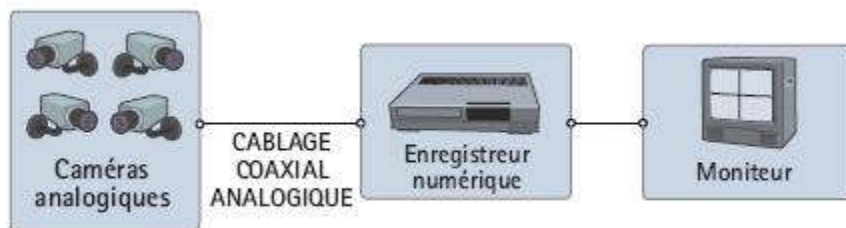


Figure I.2. Système de vidéosurveillance analogique avec enregistreur numérique

Avec un enregistreur vidéo numérique ou « digital », l'enregistrement ne se fait plus sur bandes magnétiques mais sur des disques durs où les séquences vidéo sont numérisées et compressées de telle façon à emmagasiner quotidiennement un volume important d'images.

#### I.4.3. Système de vidéosurveillance analogique avec enregistreur numérique réseau

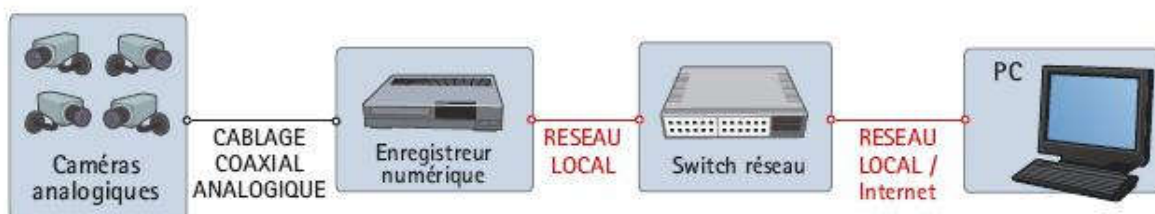


Figure I.3. Système analogique avec enregistreur numérique réseau

C'est un système de vidéosurveillance analogique passant configuré par un enregistreur numérique en réseau par l'intermédiaire d'une connexion à travers un port Ethernet. La vidéo étant numérisée et compressée sur l'enregistreur numérique, les images sont visualisées sur PC à distance par un réseau informatique à l'aide d'un client Windows spécifique, ou simplement un navigateur web standard. Les enregistreurs numériques en réseau permettent donc la visualisation d'images vidéos à distance sur PC et le contrôle du système (de prise de vue/de la caméra) à distance.

#### I.4.4. Système de vidéosurveillance sur IP avec serveur vidéo

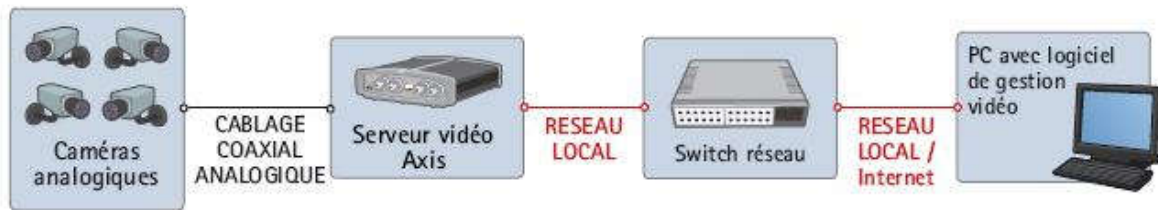


Figure I.4. Système de vidéosurveillance sur IP avec serveur vidéo

Les systèmes de vidéosurveillance actuels fonctionnent sur réseaux IP avec une association à un serveur vidéo qui assure la migration du système de vidéosurveillance analogique vers une solution de vidéosurveillance sur IP. Ainsi, la caméra analogique est branchée sur le serveur vidéo pour assurer la numérisation et la compression des séquences vidéo. De son côté, le serveur vidéo est connecté sur le réseau informatique pour le transport des images vidéos vers un PC ou un serveur via un commutateur réseau. Les images vidéo sont alors enregistrées sur le disque dur du PC. Ce type de système permet l'enregistrement hors site, et de faire évoluer le système, une caméra à la fois.

#### I.4.5. Système de vidéosurveillance sur IP avec caméras réseau

C'est un système entièrement numérique flexible et évolué, où les images vidéo sont transmises sur réseau IP à l'aide de caméras réseau. Une caméra réseau présente l'association d'une caméra numérique et un ordinateur, et permet la numérisation et la compression vidéo. Les images vidéo sont acheminées par réseau IP par l'intermédiaire des commutateurs réseau, pour être restituées et enregistrées sur un PC/serveur standard à l'aide d'outils de gestion vidéo.

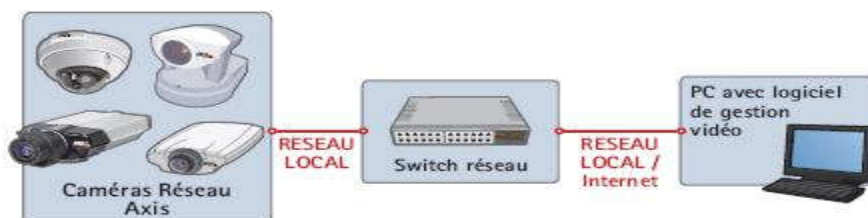


Figure I.5. Système de vidéosurveillance sur IP avec caméras réseau

L'une des évolutions les plus probables de la vidéosurveillance est le rapprochement des enregistrements avec des données biométriques qui permettrait d'analyser les démarches

inhabituels des individus. De même, des recherches récentes misent sur la prévisibilité du comportement humain dans les espaces publics: un voleur ne se comporterait pas de la même façon qu'un usager. L'ordinateur peut identifier ce genre de mouvements et donner l'alerte.

#### **I.4.6. Réseau sans fil**

Actuellement dans la plupart des bâtiments, la solution sans fil peut s'avérer la plus intéressante pour l'utilisateur, tant financièrement que sur le plan fonctionnel.

Pour le problème à certains bâtiments classés par exemple, où l'installation d'un câblage endommagerait inévitablement l'intérieur, ou à certains sites (commerces par exemple) pour lesquels la caméra doit être régulièrement déplacée et où l'on ne souhaite pas devoir tirer chaque fois de nouveaux câbles, la solution est l'utilisation courante de la technologie sans fil concerne les bâtiments ou les sites que l'on souhaite relier sans pour autant devoir entreprendre de lourds et coûteux travaux au sol.

La technologie sans fil s'applique à la fois aux systèmes de vidéo sur IP et aux systèmes analogiques et se divise en deux catégories principales :

- LAN sans fil (Wireless LAN, ou WLAN) : définit un réseau local, c'est-à-dire sur de courtes distances et en principe à l'intérieur.
- Ponts sans fil : pour relier des bâtiments ou des sites par une liaison rapide, point à point longue distance et à grande vitesse. Les technologies micro-ondes et laser sont couramment utilisées.

### **1.5 Problématique**

Notre objectif dans cette thèse est de développer un système de vidéosurveillance intelligent à faible coût à l'aide de la technologie innovante du microordinateur Raspberry pi 3 [1].

C'est un système qui doit pouvoir détecter n'importe quel mouvement (intrus), activez une caméra (simple webcam) pour prendre des photos de la vidéo lors de la détection du mouvement, puis envoyez une alerte par SMS au propriétaire de l'installation et en même temps un courrier électronique avec la photo de l'intrus en pièce jointe.

Le coût d'installation de ce système de sécurité dépend des composants utilisés :

- Un Raspberry Pi 3 de 6000 DA [1]
- Un Arduino UNO de 3000 DA
- Une webcam de 500 DA

- Un Capteur de mouvement PIR de 200 DA
- Un module GSM/GPRS A6 de 1000 DA
- Un écran LCD I2C 2 X 16 de 250 DA

Donc notre système va nous coûter environ 10000 DA et qui est beaucoup moins chère que les systèmes de vidéosurveillance intelligents actuels

Un autre aspect de ce projet consiste à présenter une idée de la surveillance et d'enregistrement de la photo d'un intrus à l'aide d'une caméra. Tout objet passant dans le champ de vision de la caméra sera détecté puis enregistré par l'envoi de sa photo par email.

### **1.6 Objectifs du projet:**

L'objectif principal de ce projet est de concevoir et développer un système de sécurité intelligent comprenant des fonctionnalités tel que la détection de mouvement, l'envoi par courrier électronique ou SMS aux propriétaires d'installations. Les objectifs spécifiques sont:

- Etudier et décrire comment le Raspberry Pi peut être interfacé avec un capteur de mouvement PIR
- Etudier comment un Raspberry Pi peut être programmé pour pouvoir envoyer un email à un mailhub prescrit.
- Etudier et décrire comment un Arduino peut être interfacé avec un module GSM A6 pour envoyer des SMS et effectuer des appels.
- Concevoir et mettre en œuvre un système de détection et de suivi de mouvement en temps réel

Ce projet est axé sur la mise au point d'un système permettant de détecter les mouvements et de réagir rapidement en capturant une image et en la relayant sur un périphérique administrateur via Internet.

Le système nécessite donc :

- Un module Raspberry Pi
- Un capteur de mouvement PIR
- Une carte Arduino UNO
- Un module GSM A6
- Un LCD I2C pour afficher l'état de détection

Une Connexion Internet.

Cependant, ce projet ne tentera pas de concevoir le dispositif de détection de mouvement, la caméra ou le raspberry Pi. Il utilisera donc ces systèmes avec un script de programme approprié pour réaliser un système de surveillance en temps réel comme souhaité.

### **1.7. Justification**

Le système de sécurité à concevoir dans le cadre de ce projet peut être largement utilisé pour surveiller les biens mobiliers et immobiliers. Le propriétaire doit pouvoir surveiller sa propriété où qu'elle se trouve dans le monde. Il ne remplacera pas l'utilisation des systèmes de vidéosurveillance, mais réduira le coût de mise en œuvre d'un système de sécurité de base. Cela permettra donc aux petits propriétaires de maison de sécuriser leur installation à un coût moins cher.

**Chapitre II :**  
**Les systèmes de**  
**sécurité intelligents**

## CHAPITRE II. Les systèmes de sécurité intelligents

### II.1. Historiques des systèmes de sécurité à base d'un Raspberry Pi

De nos jours, les chercheurs et les développeurs ont mis au point un large éventail de systèmes de surveillance utilisés pour la surveillance à distance, les alertes et le contrôle des tâches via des outils informatiques faciles à mettre en œuvre. Certains ont jusqu'à présent été réalisés alors que d'autres restent encore une proposition.

Les chercheurs Padmashree A. Shake et Sumedha S. Borde ont présenté un système de surveillance domicile intégré utilisant un système d'alerte basé sur la détection de petits mouvements. Ils ont travaillé sur la mise en place d'un système à faible coût, et d'une consommation d'énergie réduite en utilisant un ensemble de différents capteurs.

Ce système aide à surveiller les activités domestiques en temps réel de n'importe où et à partir d'un microcontrôleur qui est aujourd'hui considéré comme une solution open source [2].

D. Jeevanand a travaillé à la conception d'un système de capture vidéo sur réseau en utilisant le Raspberry Pi. Le système proposé fonctionne sur la capture vidéo et la distribution avec des systèmes en réseau pour alerter la personne de l'administration via l'alarme SMS suite à la demande du client. C'est un système conçu pour fonctionner dans des situations en temps réel et basé sur un Raspberry Pi en temps réel et avec un moniteur vidéo et un système d'alerte [3].

Sneha Singhd et son équipe ont décrit le système de surveillance vidéo par caméra IP utilisant la technologie Raspberry Pi. Les chercheurs visaient à développer un système qui capture des images en temps réel et les affiche dans un navigateur en utilisant le TCP / IP.

L'algorithme de détection de visage est implémenté sur Raspberry Pi, qui permet le streaming vidéo en direct ainsi que la détection des visages [4].

Mahima F. Chauhan et Gharge Anuradha ont proposé de concevoir et de développer un système de vidéosurveillance en temps réel basé sur le serveur Web intégré dans le Raspberry PI B+ , c'est un système à faible coût, bonne ouverture et qui fournit de meilleures solutions de sécurité dans les banques, l'industrie, l'environnement et les arts militaires [5].

En 2014 Jadhav G. J a utilisé de divers capteurs, module sans fil, unité de microcontrôleur et un module d'empreintes digitales pour formuler et mettre en œuvre un système de surveillance efficace. Lui et son équipe ont adopté un noyau ARM en tant que processeur de base du système. Le capteur PIR est utilisé pour détecter le mouvement dans la zone de vision, tandis que le capteur vibrant est utilisé pour détecter tout événement de vibration.

La technique de détection d'intrusion est proposée en utilisant le capteur PIR qui détecte le mouvement et déclenche un système d'alerte et d'envoi de messages courts via un module GSM vers un numéro de téléphone spécifié [6].

En 2014, Sanjana Prasad et ses collègues ont travaillé au développement d'un système de surveillance mobile intelligente basé sur le Raspberry Pi et le détecteur de mouvement PIR. L'objectif de leur recherche est de développer un système de sécurité pour maisons avec smart phones basé sur un module de capture d'informations associé à un module de transmission basé sur la technologie 3G fusionnée avec des applications Web.

Le Raspberry Pi contrôlera les événements du capteur PIR et exploite les caméras vidéo pour le streaming vidéo et les tâches d'enregistrement. Ce système a la capacité de compter le nombre d'objets dans une scène [7].

Uday Kumar a travaillé à la mise en place d'un système de surveillance à distance sans fil à faible coût en utilisant le Raspberry Pi. Les caméras CCTV sans fil sont largement utilisées dans les systèmes de surveillance à bas prix. Il a mis en place un système de surveillance à faible coût utilisant une caméra et un Raspberry Pi et les images acquises doivent être transférées à un drop box à l'aide d'un dongle internet 3G [8].

### **II.2. Evolution de la sécurité**

Avec l'invention de l'électricité, l'art de la protection de la maison a été grandement amélioré. En 1853, le premier brevet sur les alarmes électromagnétiques permettait aux entreprises et aux riches résidents de se procurer des objets de valeur. Des contacts magnétiques ont été installés sur les fenêtres et les portes qui, une fois déclenchées, envoyés un signal via le câblage électromagnétique et déclencher une alarme. Ces révolutionnaires systèmes de sécurité ont été efficaces pour dissuader les cambriolages [9].

Les systèmes de surveillance ont beaucoup progressé. Après le système d'alarme, la caméra vidéo analogique avec magnétophone à cassettes a évolué. Il y avait une mauvaise image et pas d'accès à distance, pour surmonter les inconvénients de ce système, les enregistreurs vidéo numériques ont évolué, ils ont donné des images de bonne qualité et permettent la transmission de signaux vidéo via des réseaux de données et permettait ainsi une surveillance à distance [10].

Les Enregistreurs vidéo réseau ont ensuite émergé. Ils ont les avantages des DVR, ils donnent plus d'options de stockage et de connexion réseau.



La version la plus récente est celle utilisée par Cisco Video Surveillance Platform. Ils offrent un accès et un contrôle à distance sécurisés de n'importe où, un stockage redondant à sécurité intégrée, une intégration facile avec d'autres systèmes [10].

### II.3. Définition du système de sécurité

La sécurité signifie littéralement une manière ou méthode par laquelle quelque chose est sécurisé par un système de composants et dispositifs d'interfonctionnement.

D'autre part, les systèmes de sécurité sont des réseaux de dispositifs électroniques intégrés fonctionnant avec un panneau de commande central pour la protection contre cambrioleurs et autres intrus potentiels [11].

Les systèmes de sécurité fonctionnent sur le concept simple de sécuriser les points d'entrée dans une maison avec des capteurs qui communiquent avec un panneau de commande ou centre de commande installé dans un endroit pratique.

Les capteurs sont généralement placés dans les entrées ainsi que des fenêtres facilement accessibles

Un système de sécurité pour maison typique a les caractéristiques suivantes

Composants: un panneau de commande, principal contrôleur d'un système de sécurité, des portes et capteurs de fenêtre, capteurs de mouvement, caméras de sécurité filaires ou sans fil, sirène ou alarme à décibels élevés [10].

### II.4. Technologies de sécurité actuelles

#### II.4.1 Système de sécurité maison basé sur Arduino

Ce projet de système de sécurité porte sur la conception et le développement d'un système de contrôle du vol pour la maison, qui est utilisé pour prévenir et contrôler toute tentative de vol. Le système développé utilise un système embarqué comprenant un microcontrôleur matériel ouvert (Arduino) et un modem basé sur la technologie du système mondial de communication mobile (GSM) [12].

Le système conçu et développé peut être installé à la maison. Une interface de détection d'intrusion est également connectée au système de sécurité basé sur un microcontrôleur. Le système donc intègre un capteur infrarouge passif (PIR) pour la détection de mouvement. En cas de tentative d'intrusion, un message d'avertissement est transmis par le système (sous

forme de SMS au téléphone portable du propriétaire, ou vers n'importe quel numéro de téléphone mobile préconfiguré pour un traitement ultérieur.

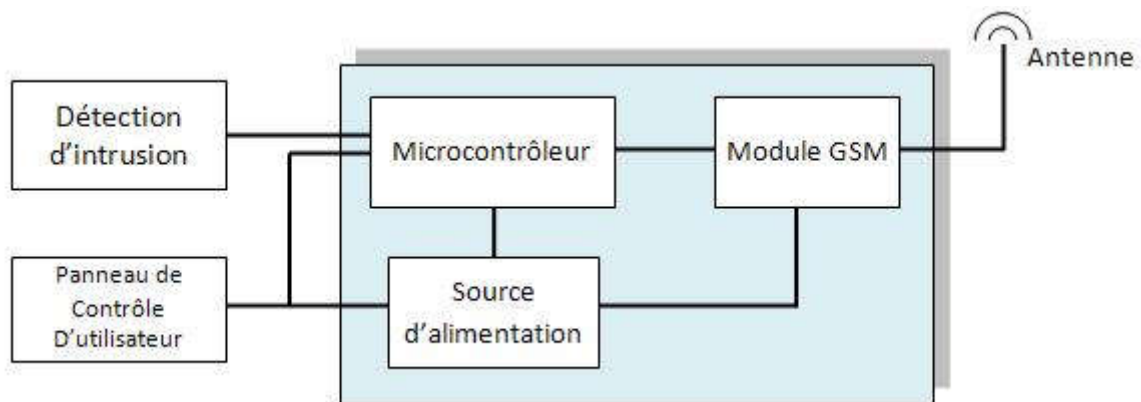


Figure II.1: Schéma fonctionnel du système de sécurité maison basé sur Arduino [12]

Le système de sécurité comprend un microcontrôleur Arduino Uno, un système standard basé sur Modem GSM / GPRS SIM900A et capteur PIR. L'ensemble du système peut être alimenté par n'importe quel bloc d'alimentation ou batterie de 12VDC / 2A

#### II.4.1.1. Mode de fonctionnement :

Lorsque la puissance d'entrée est appliquée au système, le système passe en mode veille. Toutefois, lorsque les terminaux du connecteur joignant PIR au microcontrôleur Arduino sont court-circuités, le message d'avertissement préprogrammé est automatiquement transmis au numéro de mobile concerné. Ce système ne transmet toutefois pas l'image de l'intrus. Il ne transmet qu'un message de notification.

#### II.4.2. Les systèmes CCTV (Closed-Circuit Television)

La vidéosurveillance consiste à utiliser des caméras vidéo pour transmettre un signal à un endroit spécifique, sur un nombre limité de moniteurs. Il diffère de la télévision de radiodiffusion en ce sens que le signal n'est pas transmis ouvertement, bien qu'il puisse utiliser un point à point (P2P), un point à plusieurs points ou un maillage sans fil [13]. Aux États-Unis, le premier système commercial de télévision en circuit fermé est apparu en 1949, appelé Vericon [14].

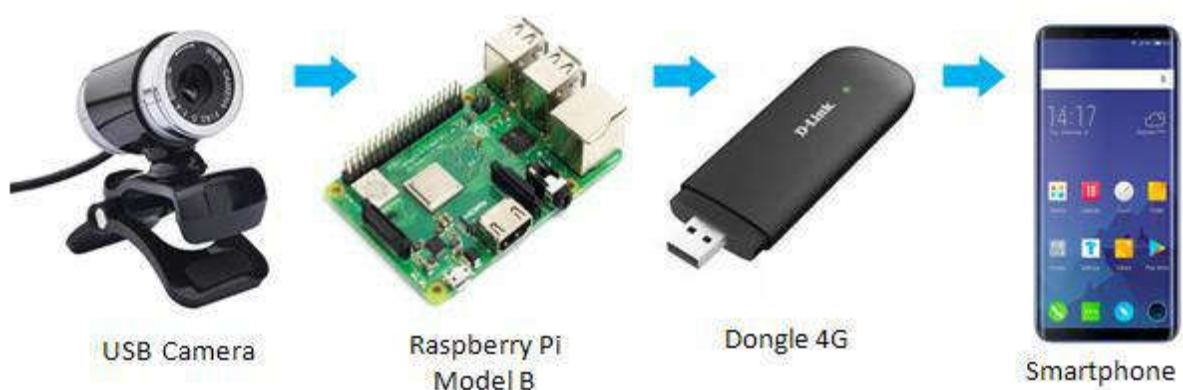
### II.4.3. les Système IP de surveillance à distance

La surveillance IP est une version numérisée et mise en réseau de la télévision en circuit fermé (CCTV). Dans un système de surveillance IP, une caméra IP enregistre une séquence vidéo et le contenu qui en résulte est distribué sur un réseau IP [15]. C'est un système qui offre des avantages supplémentaires:

- Capacité améliorée de visualisation et de contrôle à distance. Toute personne sur le réseau peut potentiellement voir la vidéo de n'importe quelle caméra connectée au réseau.
- Le stockage IP permet de stocker des données dans n'importe quel emplacement géographique.
- Plus grande facilité de distribution. Une image d'un suspect de crime, par exemple, peut être immédiatement distribuée.
- La possibilité de se connecter à la messagerie électronique et à d'autres systèmes de communication de manière à pouvoir envoyé automatiquement.

### II.4.4. Système de surveillance à base de Raspberry Pi

Un Raspberry Pi peut être utilisé pour mettre en œuvre un système de sécurité avec détection de mouvement, traitement d'image et mécanisme d'alerte. L'alerte doit contenir une capture de photo ou de vidéo transmis sur Internet. Cela permettra donc aux utilisateurs de surveiller les maisons de partout dans le monde.



*Figure II.2 Système de sécurité à base de Raspberry Pi*

Comme le montre la figure ci-dessus, l'architecture du système est composée de cinq modules:

- Raspberry Pi Modèle B
- Camera USB
- Capteur PIR
- WiFi Dongle
- Moniteur

Le rôle de chaque module sera discuté en détail dans la partie suivante de ce chapitre.

### II.4.4.1. Pourquoi des systèmes de sécurité maisons à base de Raspberry Pi?

Plusieurs critères ont été utilisés pour sélectionner un système de sécurité requis pour protéger nos biens, Le principal de tous ces facteurs a été le coût de la mise en œuvre d'un tel système Le Raspberry Pi est également un appareil très polyvalent dont les fonctionnalités ne sont pas limitées.

Le Raspberry Pi est également un appareil très polyvalent dont les fonctionnalités ne sont pas limitées. Il peut être prolongé d'être simplement un dispositif de sécurité vers un dispositif de contrôle de la température, d'éclairage automatique et même un serveur proxy. Les raisons suivantes expliquent la nécessité de disposer d'un système de sécurité à base de Raspberry Pi :

- Un système de caméra IP a la capacité de diffuser des messages d'alarme sur Internet ainsi que le système de sécurité basé sur Raspberry Pi. Cependant, le coût d'une caméra IP la rend difficilement abordable pour les petits propriétaires (insérez le coût plus la citation). Ils peuvent donc être déployés dans de grandes installations industrielles, des forces de défense, des services de police, etc.
- Le système de sécurité basé sur le microcontrôleur Arduino peut être relativement moins coûteux à mettre en œuvre que le système basé sur Raspberry Pi, mais sa capacité de mémoire le rend plus inefficace, en particulier lors de la tentative d'interface avec d'autres modules, par ex. caméra, moniteurs, capteurs de mouvement, souris et clavier. Raspberry Pi a une mémoire extensible de stockage sur carte SD et peut être élargi pour répondre aux besoins d'un individu. De plus, le microcontrôleur Arduino nécessite un modem GSM pour lui permettre de transférer des informations à travers le l'Internet. Le Raspberry Pi possède un port pour le connecter à Internet.
- Un système de surveillance CCTV est coûteux à acheter et à installer par rapport au système en question. Un système DVR est nécessaire pour le connecter aux réseaux de données via TNP / IP et un DVR coûte très cher. Par conséquent, un tel système peut ne pas un tel système peut ne pas être offert par les propriétaires de maison à faible revenu.

Le système de sécurité proposé dans ce projet est un système à base d'un Raspberry pi B3 en parallèle avec d'autres modules intelligents qui vont être décrit dans la section suivante

### II.5. Les modules utilisés

#### II.5.1. Le Raspberry Pi

Le Raspberry Pi est un micro-ordinateur basé sur Linux et sur l'architecture ARM [16]. Il a été construit principalement pour aider au développement de jeux open source [17], il n'est pas plus grand qu'une carte de crédit. Il a été conçu par David Braden, sa première mise en vente c'est fait le 29 février 2012 pour la somme de 25 euros.



*Figure II.3 Le Raspberry Pi*

En Février 2015 la fondation annonce avoir vendu 5 Millions d'exemplaires. Par la suite différents modèles Raspberry Pi sont apparues et ils sont présentés comme suit :

#### II.5. 1.1. Les modèles Raspberry Pi

**Le Modèle Zéro:** Il y a 3 types Zéro, Zéro W et Zéro WH

##### Modèle zéro :

Le Raspberry Pi Zéro est deux fois plus petit qu'un modèle A+, et deux fois plus utilitaire et plus élégant.

Hauteur : 30 mm

Largeur : 65 mm

Poids : 9g



*Figure II.4–Modèle zéro*

##### Modèle zéro W :

Le Raspberry Pi Zéro W étend la famille Pi Zéro avec une connectivité Bluetooth et LAN sans fil supplémentaire.

CPU : 1 GHz

Mémoire vive : 512 Mb



*Figure II.5 Modèle zéro W*

Modèle zéro WH :

Même carte que Raspberry Pi Zero W mais avec un connecteur 40 pins GPIO déjà soudé.

SPI + I2C  
Camera + MicroSD



*Figure II.6 Modèle zéro WH*

**Le Modèle A :**

Il y a 2 types : A et A+

Modèle A :

Le Raspberry Pi A est équipé de 256 RAM et ne dispose pas de port Ethernet, et il est moins puissants.

2 Sorties vidéo : Composite et HDMI  
1 Sortie audio stéréo Jack 3,5 mm  
1 Port USB 2.0 ;

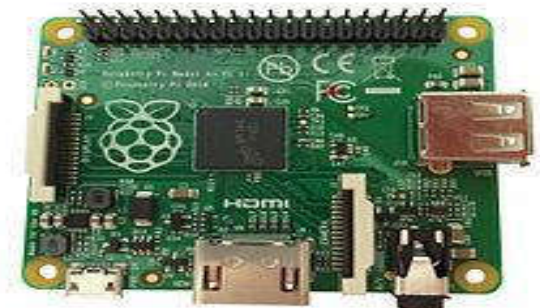


*Figure II.7 Modèle A*

Modèle A+ :

Plus petit que le Raspberry Pi A, Lecteur de carte microSD au lieu et place du lecteur SD et consommation électronique moindre.

Processeur 64 bits quad core à 1,4 GHz  
Wifi bi-bande 2,4 GHz et 5 GHz  
Bluetooth 4.2/BLE.



*Figure II.8 Modèle A+*

**Le Modèle B :**

Il y a 6 types :

Modèle B1 :

Différence par rapport aux modèles A, il dispose deux ports deux ports USB, d'un port Ethernet et de 512 RAM.

Deux ports USB 2.0 au lieu de un  
La RAM passe à 512 Mo Deux  
trous de fixation



*Figure II.9 Modèle B1*

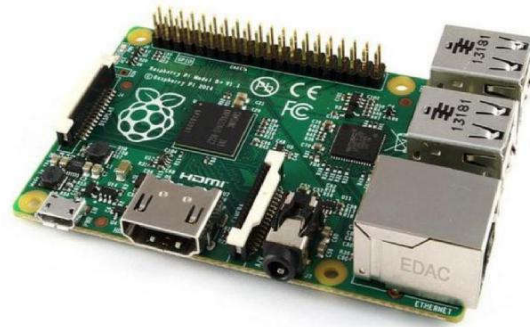
Modèle B1+ :

Le modèle B1+ est la révision finale du Raspberry Pi d'origine. Il a remplacé le modèle B1

GPIO 40 broches

4 ports USB 2.0

Réduction de consommation de 3,5 W à 3 W



*Figure II.10 Modèle B1+*

Modèle B2 :

Plus performant, plus rapide et plus sophistiqué que son prédécesseur, le Raspberry Pi 2 répond à une demande de plus en plus forte tout en gardant la même taille

RAM de 1 Go

Processeur Broadcom BCM2836

Quatre cœurs ARMv7 à 900 MHz



*Figure II.11 Modèle B2*

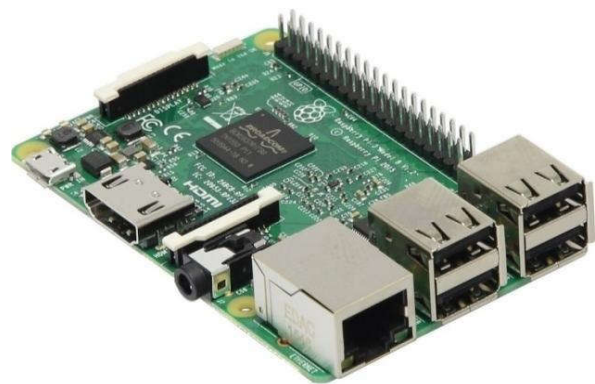
Modèle B3 :

Le Raspberry Pi 3 Model B est plus performante que Raspberry Pi 2. Il est à présent équipé nativement du Wifi b/g/n et du Bluetooth 4.1.

Processeur Broadcom BCM2837 64 bit

Quatre cœurs ARM Cortex-A53 à 1,2 GHz,

Puce Wifi 802.11n et Bluetooth 4.1 intégrée



*Figure II.12 Modèle B3*

Modèle A3+ :

Le Pi Model A3+ dispose d'un processeur plus puissant que le B+

Par contre, en termes de RAM, c'est moins bien que le B+, on retrouve ainsi moins de prises USB A 2.0.

Broadcom BCM2837B0

1,4 GHz Quatre cœurs ARM Cortex-A53



*Figure II.13 Modèle A3+*

Modèle B3+ :

Le RPi 3 B+ est une évolution du modèle B. Le processeur Broadcom BCM2837B0 gagne 200 Mhz par rapport à son prédécesseur.

Wifi Dual-band 802.11ac  
Version 4.2 du Bluetooth  
Broadcom BCM2837B0  
1,4 GHz Quatre cœurs ARM Cortex-A53

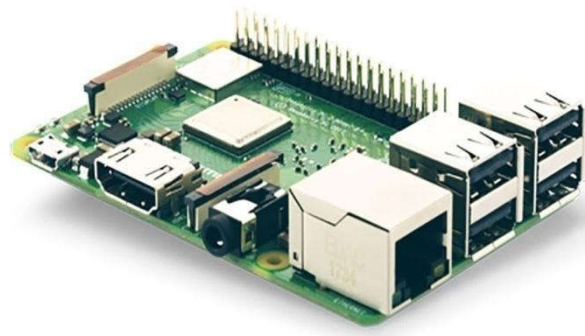


Figure II.14 Modèle B3+

Pour notre projet, nous avons choisi le Raspberry Pi B3 pour beaucoup de raisons notamment sa puissance et sa rapidité par rapport au autre Type de Raspberry

**II.5. 1.2. Le Raspberry Pi type B3**

**A. Les composants du RPi B+**

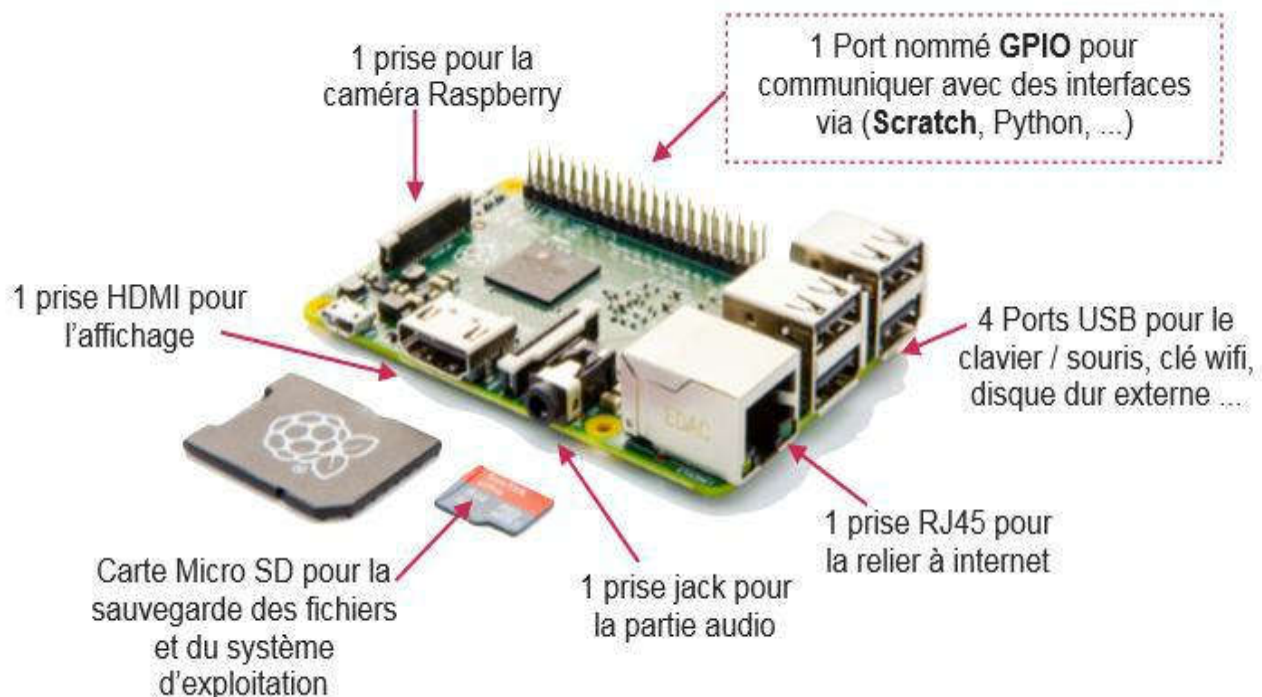


Figure II.15 Les composants standard d'un Raspberry P i B3

C'est une carte qui réunit des composants aussi essentiels que le processeur et les connecteurs d'extension ; ces différents composants sont :





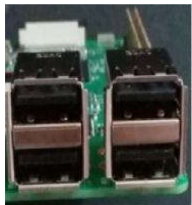
SOC (Système On a Chip): Cette puce représente un ordinateur intégré avec le processeur CPU, la mémoire RAM et GPU



Sortie HDMI : La sortie utilisée pour connecter le RPi à des écrans de télévision de haute qualité, sachant que cette sortie transmet à la fois la vidéo et l'audio au téléviseur



Prise audio: la sortie audio de 3,5 mm est identique à la taille standard pour la plupart des types d'enceintes



4 ports USB : Les ports USB tels que ceux de l'ordinateur traditionnel et utilisés pour connecter un périphérique USB tel que le clavier et la souris, différents périphériques de support tels qu'un disque flash USB ou des cartes réseau sans fil, et vous pouvez utiliser ces entrées pour connecter tout ce qui fonctionne avec l'ordinateur traditionnel.



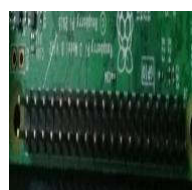
Port Ethernet(RJ45): passerelle utilisée pour connecter le RPi à l'ordinateur et aux réseaux Internet.



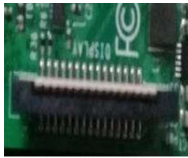
Micro USB: L'alimentation est utilisée dans l'alimentation au RPi en la connectant à n'importe quelle batterie 5V traditionnelle



Fente pour carte SD: le port d'entrée pour la carte mémoire avec le système d'exploitation et tous les fichiers de la framboise.



GPIO Pins: Le concurrent de contrôle électronique, qui vous permet de contrôler les divers composants électroniques et rend le relais capable de remplir les fonctions de microcontrôleur



Affichage DSI: Utilisé pour connecter des écrans tactiles tels que des Smartphones et des tablettes



Entrée caméra CSI: une caméra haute résolution spécialement conçue pour le RSP

## B. connexion des composants d'un Raspberry Pi

### Les accessoires indispensables (ou très utiles) pour un Raspberry Pi

#### Une carte SD :

Le Raspberry est livré sans disque dur. Pour le faire fonctionner, le système d'exploitation et tous les fichiers seront stockés sur une carte SD.

Donc cette carte SD sera le disque dur du Raspberry Pi, il est donc obligatoire d'en avoir une pour pouvoir installer le système d'exploitation et qui doit être de bonne qualité et très performante pour ne pas ralentir le Pi. L'idéal est d'avoir une carte au standard SDHC (classe 10) pouvant atteindre au maximum 80 Mo/s. Le système d'exploitation peut s'accommoder de 2Go, donc une carte de 16Go ou 32 Go est largement suffisante.

Pour brancher le Raspberry avec un écran, on a plusieurs possibilités :

- Un téléviseur ou un écran HDMI à l'aide d'un câble HDMI
- Un téléviseur ou un écran HDMI à l'aide d'une entrée vidéo composite (connecteur jaune)
- Un moniteur équipé uniquement d'un connecteur D-SUB ou DVI : Il faut faire l'acquisition d'un adaptateur HDMI->DSUB ou HDMI->DVI

Il est préférable d'installer le RPi dans un petit boîtier pour le protéger des aléas du quotidien.

#### Une alimentation

Généralement l'alimentation n'est pas fournie avec le Raspberry Pi et Il est parfois difficile de savoir quelle alimentation choisir. Pour choisir la bonne :

- Il faut avoir une alimentation avec un courant de sortie maximum à 2.5 A et une tension de 5V.
- Il faut commander une alimentation officielle, légèrement plus cher.



Figure II.16 Alimentation 2.5 A -- 5V.

### Un câble HDMI

Le câble HDMI nous servira simplement à avoir un affichage, ce qui sera très pratique pour installer l'OS et utiliser le Pi.

### Un clavier et une souris



Un clavier et une souris sera obligatoire pour installer le système d'exploitation et paramétrer le Raspberry Pi mais on peut les considérer comme des accessoires optionnels puisque la plupart possèdent un clavier et une souris chez eux.



*Figure II.17 Connexion des Les accessoires avec le Raspberry*

### C. Les différents systèmes d'exploitation pour Raspberry Pi

Après savoir quel Raspberry Pi choisir et les différents accessoires importants pour son fonctionnement et son utilisation, la chose la plus concrète est le choix du système d'exploitation.

Les distributions les plus utiles et les plus populaires sont

- Les distributions sous Linux
- Windows 10
- Les distributions pour Media Center
- Les distributions pour les jeux vidéo

Le système d'exploitation de référence pour Raspberry Pi et qui est basé sur Linux est le **Raspbian** qui présente une distribution polyvalente optimisée spécialement pour Raspberry Pi, et qui permettra la familiarisation très facilement avec le matériel.

Il existe plusieurs versions de Raspbian :

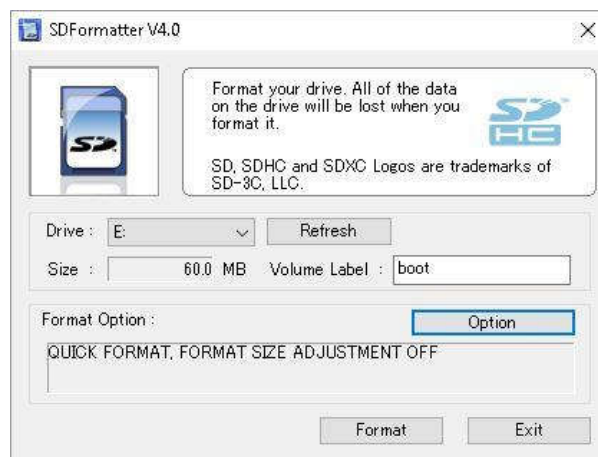
- Wheezy (basée sur Debian 7)
- Jessie (basée sur Debian 8)
- Stretch (basée sur Debian 9)

Dans ce travail on à utiliser la version Jessie

### D. Premier démarrage, installer Raspbian

#### Formater la carte SD

Même si la carte SD est neuve, le mieux est de la formater avant de copier les fichiers d'installation. Il est recommandé d'utiliser le logiciel de formatage SD Formatter disponible pour Windows et développé par la SD Association.

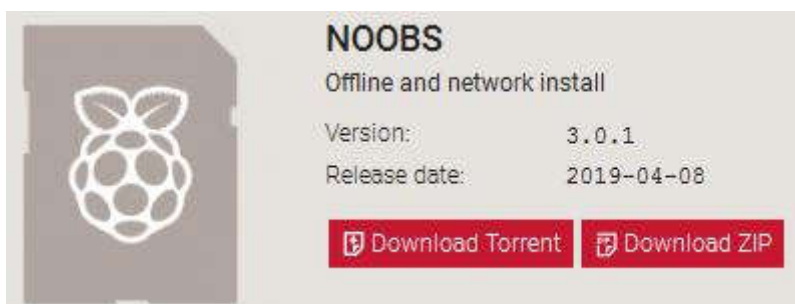


*Figure II.18 Interface du SDFormatter*

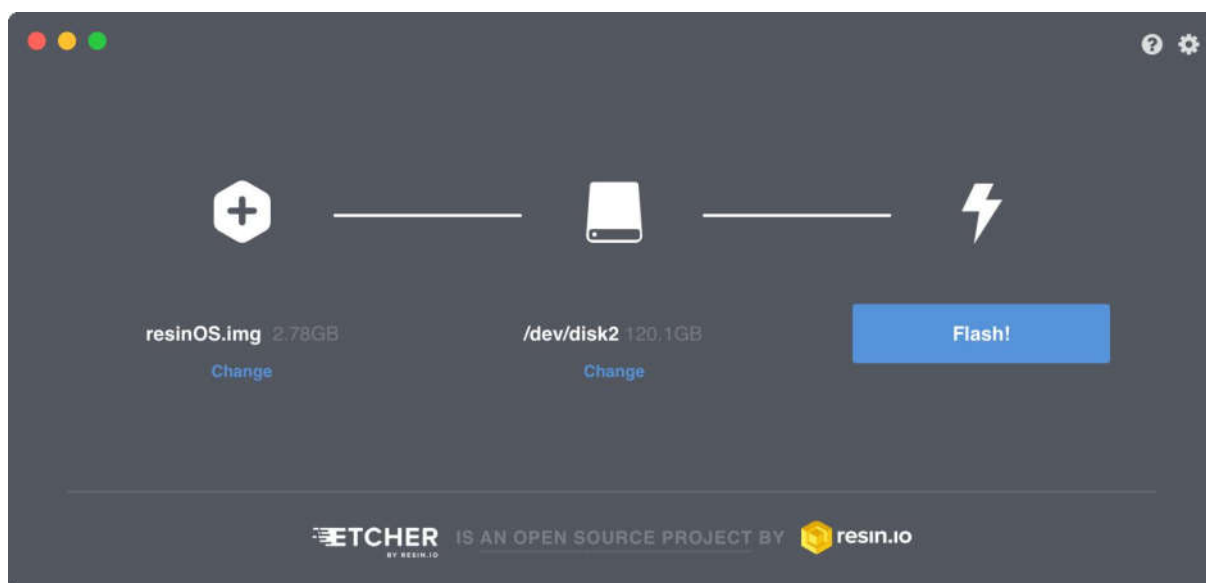
#### Installer Raspbian ou Noobs sur une carte SD

Noobs (New Out Of the Box Software) est un système de récupération et d'installation rapide du Raspberry Pi contenant le système Raspbian

Pour avoir Noobs il faut accéder au site officiel de la Raspberry Pi -- rubrique Download. <https://www.raspberrypi.org/downloads/>. Et téléchargez NOOBS (version offline)



Une fois téléchargé, il faut décompresser le fichier .zip et le copier directement sur la carte microSD. Quel que soit l'environnement de travail (PC ou Mac), il est préférable d'utiliser le logiciel Open Source **Etcher**. La carte SD est automatiquement éjectée après la vérification de la copie



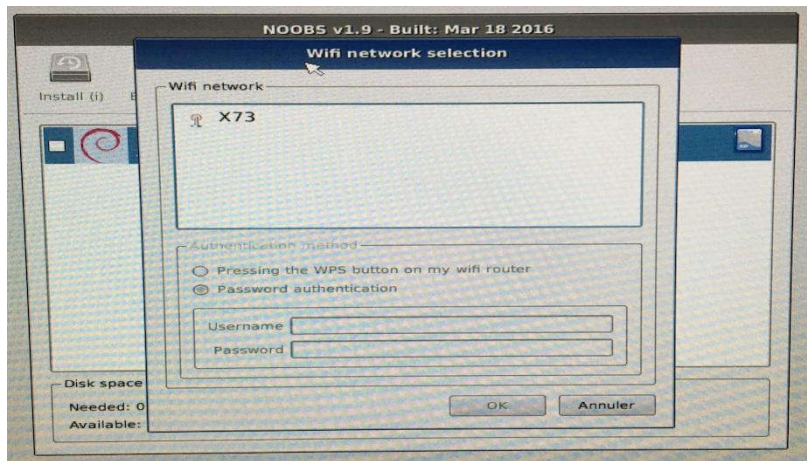
*Figure II.19 Interface du logiciel Etche*

### **Premier démarrage du Raspberry Pi sous Noobs**

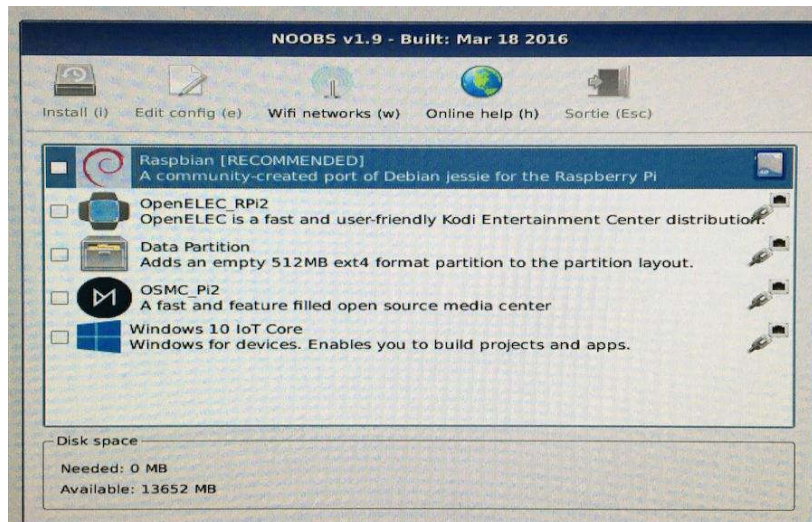
C'est le moment de démarrer le Pi. On doit insérer la carte SD dans le lecteur et le mettre sous tension. Par le choix de Noobs comme installateur, nous avons droit à une interface graphique qui va nous permettre de choisir la version de l'OS désirée, on peut changer la langue de l'interface en sélectionnant le français dans le menu en bas de l'écran.



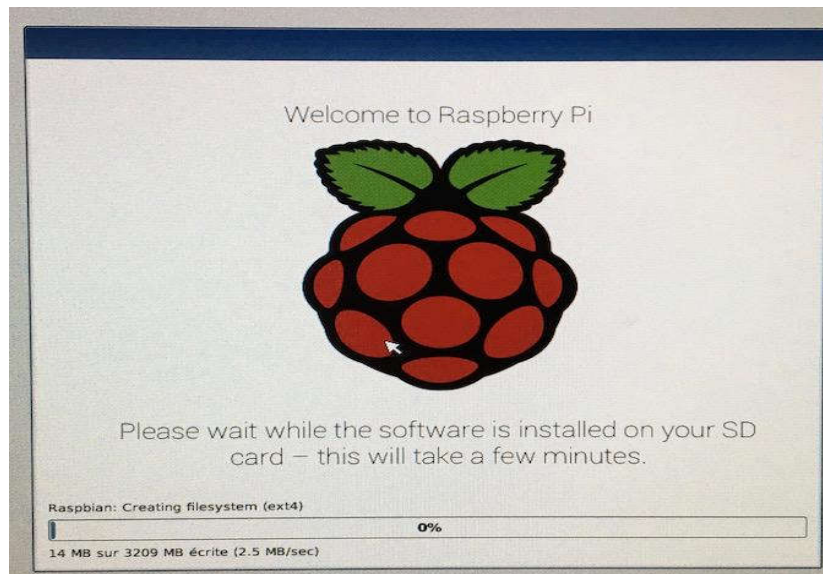
Figure II.20 Liste des distributions de NOOBS Après avoir choisi la langue; nous allons nous connecter à notre réseau Wifi.



Une fois connecté à internet, Noobs actualise la liste des distributions disponibles. Il faut au moins avoir Raspbian, OpenELEC, Data Partition, OSMC et Windows 10 IoT Core dans la liste. On sélectionne Raspbian puis on clique sur Installer.



La durée d'installation va dépendre de notre débit internet.



Une fois l'installation terminée, le Raspberry Pi doit redémarrer.



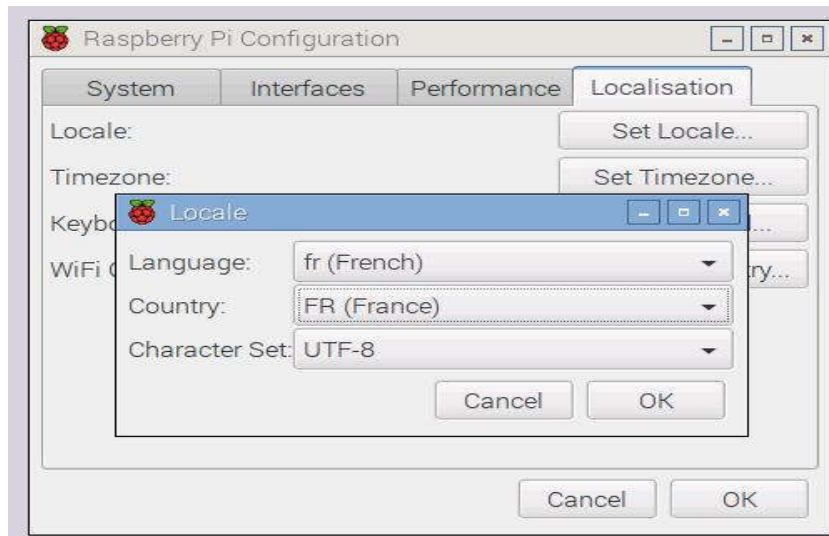
### Ajuster les réglages

Au premier démarrage du Raspberry Pi 3, aucun utilitaire de configuration n'est proposé pour configurer le système (clavier, heure...). Il faut donc procéder aux réglages uns à uns et mettre à jour le système pour profiter de toutes les nouveautés du Pi 3.

### Configurer l'interface et le clavier en français

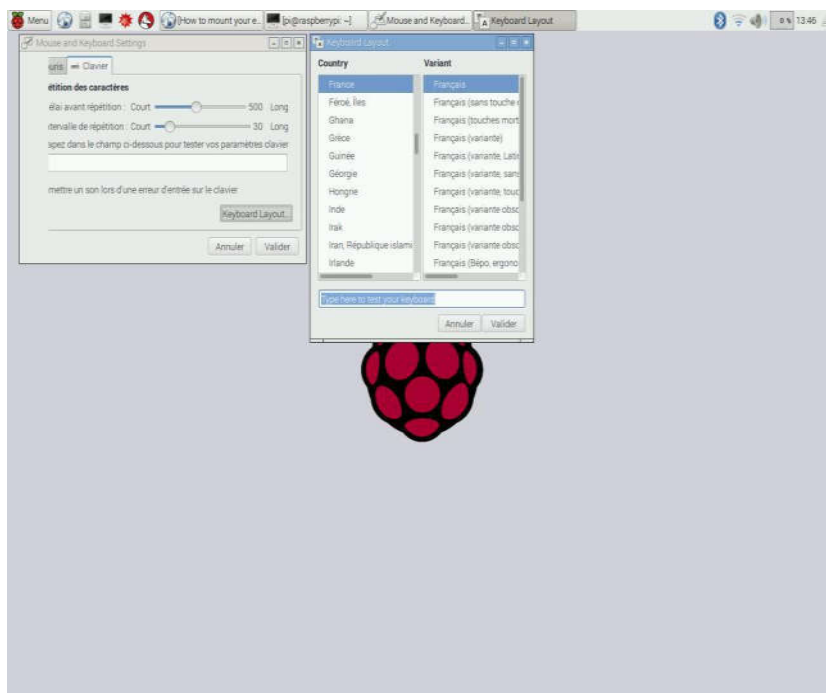
Au premier démarrage, le Pi est configuré en anglais et le clavier en qwerty ce qui n'est pas très pratique. Pour tout mettre en français, on fait Menu puis Preferences --> Raspberry Pi Configuration --> onglet Localisation.

Un clique sur **Set Locale...** --> Choisir fr (French) en face de Language et Fr (France) en face de Country



Notre Pi doit redémarrer pour prendre en compte les modifications.

Une fois redémarré, on fait Menu -> Préférences -> Clavier et souris -> onglet clavier et un clique sur Keyboard Layout...



Dans la liste qui s'affiche, choisir France en dessous de la colonne Country et on cherche le clavier qui s'approche le plus du notre dans la colonne Variant.

Valider ; le clavier est maintenant en AZERTY.



### Réglage de l'heure

Le Raspberry Pi n'a aucune batterie lui permettant de conserver le réglage de la date et de l'heure. Le Pi ira se mettre à l'heure tout seul sur internet. Pour que cela fonctionne, on doit configurer le fuseau horaire qui correspond à notre région dans l'Utilitaire de Configuration du Raspberry Pi (Menu -> Préférences) en allant dans l'onglet Localisation.

### Activer le WiFi

L'activation du WiFi est automatique si vous installez Raspbian à l'aide de Noobs ou depuis le dépôt officiel.

### Mise à jour et amélioration

On doit ouvrir le terminal des commande linux et taper les commandes suivantes :

```
S # sudo apt-get update E
```

```
S # sudo apt-get upgrade E
```

*sudo* : est l'abréviation de substitute user do, en français : faire en se substituant à l'utilisateur Cela peut prendre entre 10 mn et 1h selon le débit internet

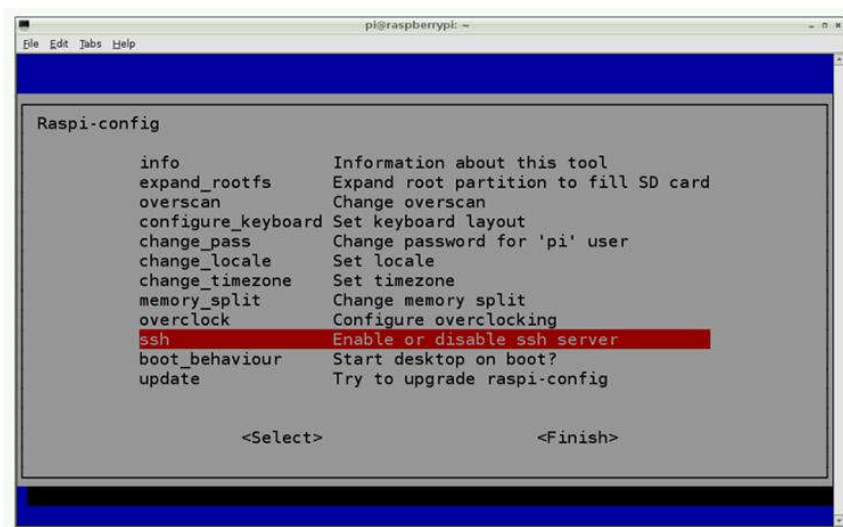
### Connexion SSH :

Grâce à ce protocole sécurisé, on peut nous connecter à notre Raspberry et effectuer ce qu'on veut depuis notre poste Windows/Max/Linux sans avoir besoin de brancher un écran sur la Raspberry, simplement :

- Tapé la commande

```
# sudo raspi-config
```

- Activer le serveur SSH



- Sous Windows, installer «Putty» qui est un client SSH
- Ensuite, il suffit d'entrer l'adresse IP du Raspberry dans Putty pour pouvoir se connecter

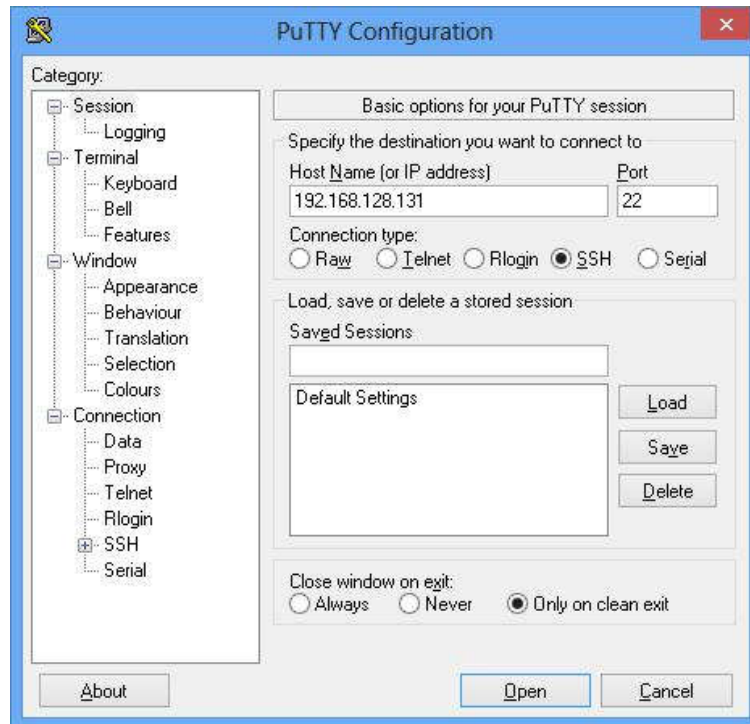


Figure II.21 Interface de PuTTY Configuration

### E. Éteindre le Raspberry Pi :

Il n'existe pas de bouton d'alimentation sur le Raspberry Pi, Pour éteindre correctement le système d'exploitation, allez dans le menu Logout de l'interface graphique et sélectionnez shutdown. On peut aussi éteindre le Pi depuis la ligne de commande en tapant :

```
S # pi@raspberrry ~ $ sudo shutdown -h now
```

Ou bien

```
S # pi@raspberrry ~ $ sudo halt
```

Remarque : Veuillez bien arrêter le système d'exploitation, Si vous coupez le courant sans arrêter le système correctement, vous risquez d'endommager le Raspberry.

### F. Programmation du RPi B+

Le langage que nous allons utiliser s'appelle Python. Il a le grand avantage d'être facile à apprendre.

#### Présentation du python :

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par Guido van Rossum, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au Centrum voor Wiskunde en Informatica aux Pays-Bas, jusqu'à se voir associer une

organisation à but non lucratif particulièrement dévouée, la Python Software Foundation, créée en 2001.

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Dès l'instant de son installation, on 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.

Python est un langage de programmation interprété, c'est-à-dire que les instructions que vous lui envoyez sont transcrites en langage machine au fur et à mesure de leur lecture. D'autres langages (comme le C / C++) sont appelés langages compilés car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine.

En contrepartie, un langage compilé se révélera bien plus rapide qu'un langage interprété (la traduction à la volée de votre programme ralentit l'exécution),

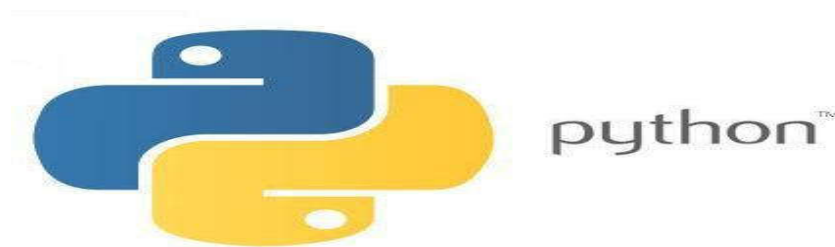


Figure II.22 Slogan de langage Python

### Programmation des entrées / sorties avec python :

On peut utiliser Les broches marquées GPIO comme broches d'entrée / sortie. Dans ce sens on va utiliser plusieurs langages de programmation capable de contrôler ces broches comme le C, Java, Bash... mais dans notre projet on a choisi le python pour examiner ces broches.

3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Figure II.23 Le port GPIO.BCM

Le module GPIO est installé par défaut sur les versions les plus récentes de Raspberry, Mais pour les versions plus anciennes, on doit probablement l'installer et effectuer une mise à jour. Le Raspberry Pi autorise deux numérotations : celle de la sérigraphie du connecteur de la carte (**GPIO.BOARD**), ou la numérotation électronique de la puce (**GPIO.BCM**). Mais dans notre projet on a choisi GPIO.BCM.

### **II.5.2. La Carte Arduino :**

C'est une plateforme basée sur une interface entrée/sortie simple, elle possède un microcontrôleur facile à programmer ainsi que de nombreuses entrées-sorties. Elle a été destinée à l'origine principalement mais pas exclusivement à la programmation multimédia interactive en vue de spectacles ou d'animations artistiques, Plusieurs cartes Arduino existent et qui se différencient par la puissance du microcontrôleur ou par la taille et la consommation de la carte. Le choix du type de la carte Arduino s'effectue en fonction des nécessités du projet.

Il y a trois types de cartes :

- Les cartes officielles : qui sont fabriquées en Italie par le fabricant officiel: *Smart Projects*.
- Les cartes compatibles : qui ne sont pas fabriqués par *Smart Projects*, mais qui sont totalement compatibles avec les Arduino officielles.
- Les autres : fabriquées par diverses entreprises et commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino, ...).

#### **II.5.2.1. Les différentes carte Arduino**

Plusieurs versions des cartes de type Arduino ont été produites et vendues dans le monde, leurs apparitions dans l'ordre chronologique est :

1. Serial Arduino - 2. Arduino Extreme - 3. Arduino Mini - 4. Arduino Nano - 5. Lilypad Arduino - 6. Arduino NG - 7. Arduino NG plus - 8. Arduino Bluetooth(BT) - 9. Arduino Diecimilia - 10. Arduino Duemilanove - 11. Arduino Mega - 12. Arduino Uno - 13. Arduino Mega2560 - 14. Arduino Ethernet - 15. Arduino Leonardo - 16. Arduino DUE - 17. Arduino Esplora - 18. Arduino MKR

Pour notre projet nous avons utilisé une carte Arduino UNO

**II.5.2.2. La carte Arduino UNO :** est une carte à microcontrôleur basée sur le ATmega328 Programmable, c'est la pièce maîtresse de tout circuit électronique pour les débutants.

Elle est dotée:

- de 14 entrées/sorties (dont 6 fournissent la sortie PWM)
- 6 entrées analogiques
- un cristal à 16 MHz
- une connexion USB
- une prise jack d'alimentation
- un en tête ICSP
- une fonction reset.

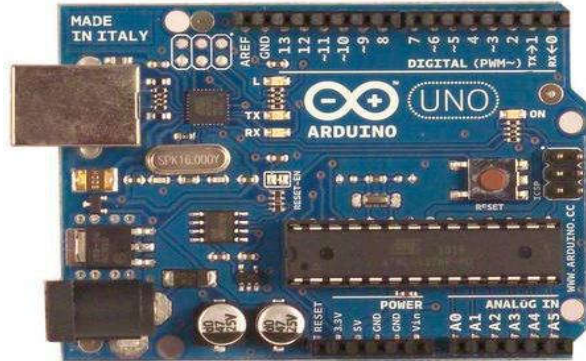


Figure II.24 Carte Arduino UNO

Brochage interne :



Figure II.25 Brochage interne d'une carte Arduino UNO

Brochage externe :

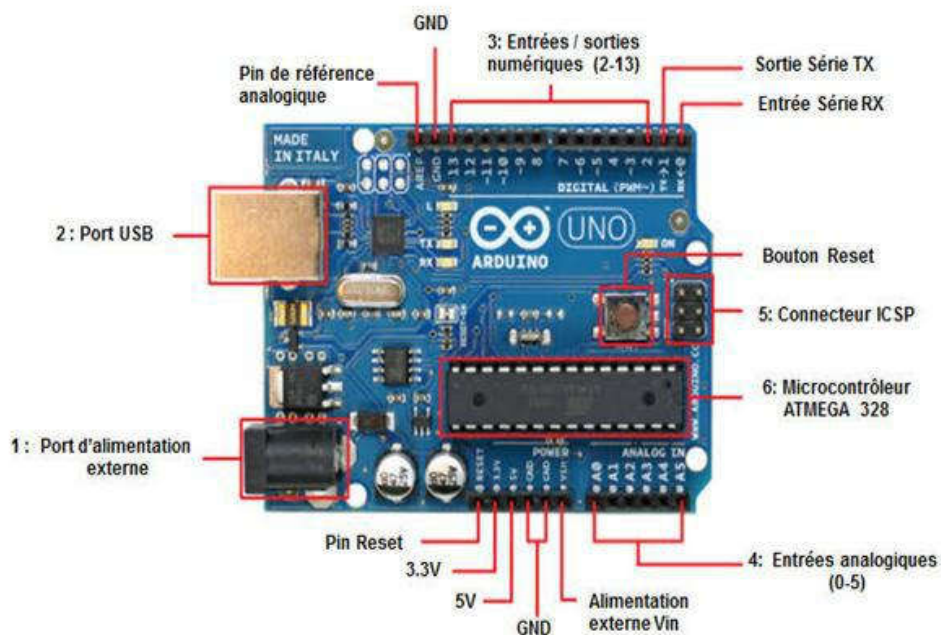


Figure II.26 Brochage externe de Arduino UNO

1 : La carte a besoin d'une alimentation externe de fonctionnement comprise entre 6 et 20V. On conseille en général de l'alimenter plutôt entre 7V et 12V pour garder une marge en basse tension et éviter que le circuit ne chauffe trop (car le régulateur de tension disperse toute surtension en chaleur). cette tension doit être continue et peut par exemple être fournie par un pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte.

2 : Port USB permet de communiquer avec la carte et de l'alimenter en 5V.

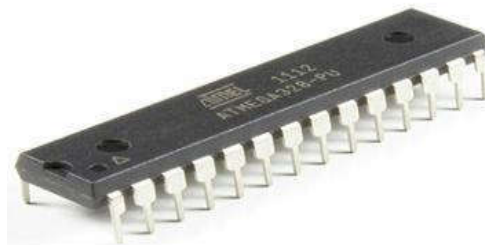
3 : Les entrée/sorties numériques 4 entrée/sorties numériques dont 6 peuvent assurer une sortie PWM peuvent actionner de nombreux composants (led, transistor,...etc) mais elles ne peuvent pas fournir beaucoup de courant (40mA pour une carte Arduino Uno). Pour piloter des circuits de plus forte puissance, il faut passer par des transistors ou des relais.

4 : Les entrées analogiques lui permettent de mesurer une tension variable (entre 0 et 5V) qui peut provenir de capteurs 6 ou d'interfaces diverses

5: Connecteur In-Circuit Serial Programming ICSP pour le téléchargement du programme. 6 : Microcontrôleur ATMEL ATmega328.

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleur **ATmega328**.

C'est un microcontrôleur ATMEL de la famille AVR8bits.



*Figure II.27–le microcontrôleur de Arduino UNO*

Les principales caractéristiques de ce microcontrôleur sont :

**FLASH**: mémoire programme de 32ko

**SRAM** : données (volatiles) 2ko

**EEPROM** : données (non volatiles) 1ko

**Digital I/O** (entrées-sorties Tout Ou Rien) : 3 ports PortB, PortC, PortD (soit 23 broches en tout I/O)

**Timers/Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits), Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)

**Plusieurs broches multifonctions** : certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation

**PWM**=6brochesOC0A(PD6),OC0B(PD5),OC1A(PB1),OC1B(PB3),OC2A(PB3),OC2B(PD3)

**Analog to Digital Converter** (résolution 10bits) : 6 entrées multiplexées ADC0(PC0) àADC5(PC5)

**Gestion bus I2C** (TWI TwoWire Interface) : le bus est exploité via les broches SDA(PC5)/SCL(PC4).

**Port série (USART)** : émission/réception série via les broches TXD(PD1)/RXD(PD0)

**Comparateur Analogique** : broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption

**WatchdogTimer programmableGestion d'interruptions** (24 sources possibles

(cfinterruptvectors)) : en résumé :

- Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3)
- Interruptions sur changement d'état des broches PCINT0 à PCINT23
- Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables)
- Interruption liée au comparateur analogique
- Interruption de fin de conversion ADC
- Interruptions du port série USART
- Interruption du bus TWI (I2C)

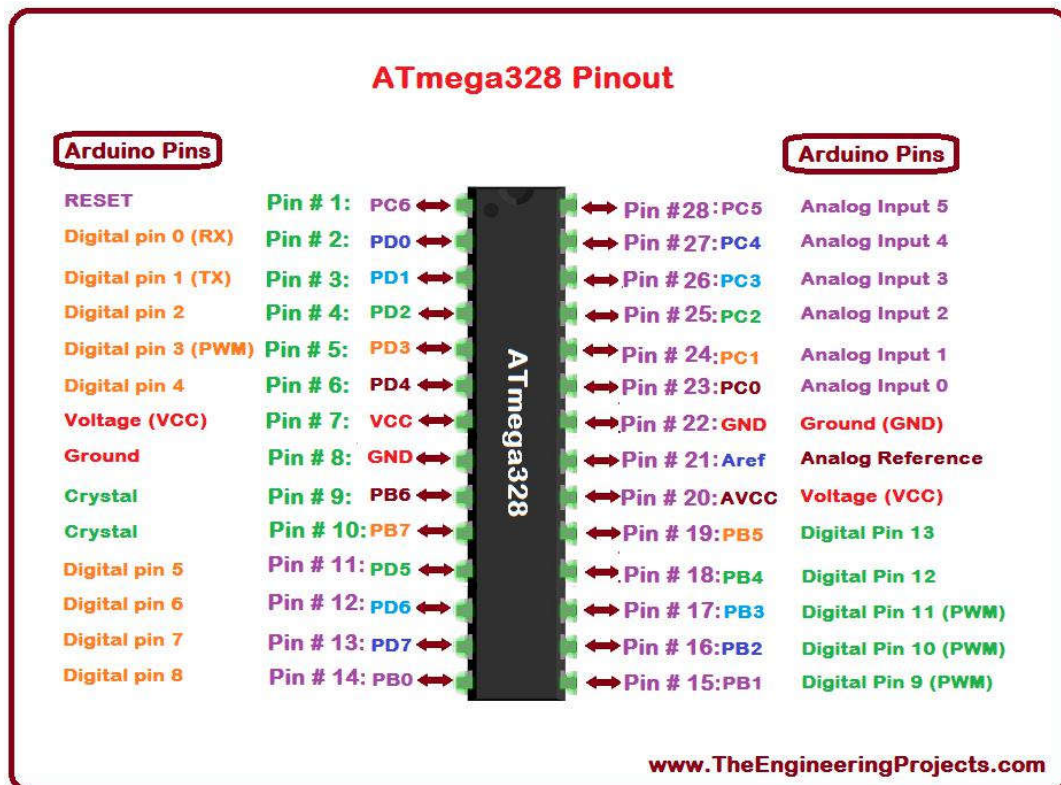
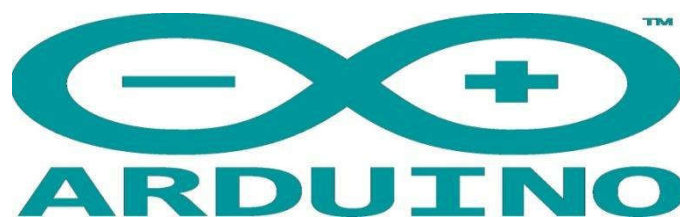


Figure II.28 les pins du microcontrôleur d'un Arduino UNO

### II.5.2.2. programmation de la carte Arduino Uno :

Afin de fonctionner, l'Arduino Uno doit être tout d'abord programmé, un programme destiné à ce type de carte s'appelle un sketch. la mise en œuvre est relativement simple :

- Tout d'abord, il faut télécharger l'environnement de développement Arduino IDE. Ce logiciel va d'une part installer tous les pilotes nécessaires. Il pourra être utilisé avec n'importe quel type de carte nativement compatible Arduino, mais également avec d'autres types de cartes dérivées, comme les NodeMCU, qui couplent un circuit wifi et un Arduino Nano. Il permet d'autre part de vérifier l'intégrité du code avant de l'envoyer à la carte.
- Brancher la carte à l'aide d'un câble USB et indiquer au logiciel le port COM sur lequel l'Arduino est branché.
- Télécharger les éventuelles bibliothèques nécessaires au sketch (comme par exemple la bibliothèque wifi en cas d'utilisation avec un circuit wifi ESP8266),
- Entrer le code du programme (langage C),
- Compiler et téléverser le code sur l'Arduino.



*Figure II.29 Slogan du logiciel arduino*

### II.5.3. Module GSM A6 :

Le module A6 GSM / GPRS est une carte de développement de base mini GSM / GPRS basée sur le module GPRS A6. Il prend en charge le réseau bi-bande GSM / GPRS, disponible pour la transmission à distance de données de messages GPRS et SMS. supporte les appels vocaux et SMS, faire et répondre à des appels téléphoniques en utilisant un casque et un microphone à électret, envoyer et recevoir des données GPRS (TCP / IP, HTTP, etc.).

La carte présente une taille compacte et une faible consommation de courant. Avec la technique d'économie d'énergie, la consommation de courant est aussi basse que 3 mA en mode veille. Il communique avec le microcontrôleur via le port UART et prend en charge les commandes, notamment les commandes GSM 07.07, GSM 07.05 et Ai-Thinker améliorées.





*Figure II.30 Module GSM A6*

Elle présente les caractéristiques suivantes :

- Fréquence de fonctionnement: réseau quadri-bande, 850/900/1800/1900MHz
- Tension de fonctionnement: 4.8-9VDC (régulateur de tension de bord)
- Interface Micro USB intégrée pour l'alimentation externe
- Courant de fonctionnement: 2A max
- Mode veille: 5 mA
- Support de carte Micro SIM
- Interface de communication: port série TTL / port série RS232
- Débit en bauds: 115200bps et il peut également être défini par la commande AT
- Tension logique d'interface: 3.3V
- Faire et répondre à des appels téléphoniques en utilisant un casque et un microphone
- Envoyez et recevez des messages SMS
- Envoyer et recevoir des données GPRS (TCP / IP, HTTP, etc.).
- Être utilisé pour tester le module Ai-Thinker GPRS A6
- Pas de la broche: 2.54mm
- Interface d'antenne embarquée: SMA et IPX
- Compatible avec les modules GSM GPRS SIM800L / SIM900
- Fréquence: 780 MHz ~ 960 MHz. 1710 MHz ~ 2170 MHz

### **II.5.3.1. Généralités sur les commandes AT**

Les commandes AT sont définies dans la norme GSM 07.07 (pour les SMS - GSM 07.05).

AT est l'abréviation de ATtention. Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes ASCII). Les commandes permettent la gestion complète du mobile.

Trois entités sont définies :

- TE : Terminal Equipment : envoi et affiche les commandes
- TA : Terminal Adaptator : interface entre l'utilisateur et le mobile
- ME : Mobile Equipment.

Types de AT commandes et réponses:

Type de la commande AT	Syntaxe	Fonction
Commande de teste	AT+CXXX=?	L'équipement mobile renvoie la liste des paramètres et des plages de valeurs définis avec la commande d'écriture correspondante ou par des processus Internes
La commande de lecture	AT+CXXX?	Cette commande renvoie la valeur actuellement définie du paramètre ou des paramètres.
La commande de définition	AT+CXXX=<...>	Cette commande définit les valeurs des paramètres définissables par l'utilisateur.
La commande d'exécution	AT+CXXX	La commande d'exécution lit les paramètres invariables déterminés par des processus internes

### II.5.3.2. Schéma de fonctionnement

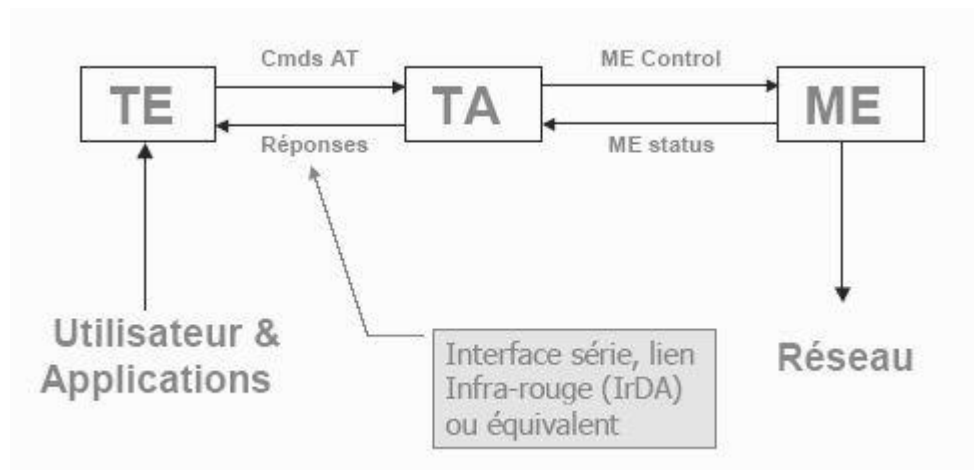


Figure II.31 Fonctionnement d'une commande AT

### II.5.3.3. Commandes de base

En général, il faut taper les commandes AT en MAJUSCULES. La commande AT tout cours doit donner la réponse "OK"

ATI [<valeur>] : Affichage des paramètres du TA.

ATZ [<valeur>] : Chargement des paramètres par défaut mémorisés par l'utilisateur.

AT&F [<valeur>] : Chargement des paramètres par défaut mémorisés par le fabricant.

ATE [<valeur>] : Echo des commandes AT (déf. 1).

ATQ [<valeur>] : Suppression du résultat (déf. 0).

#### II.5.3.4. Commandes de gestion d'un appel

- AT+CSTA = [<type>] : Sélection du type des numéros de téléphone.
  - 145 : Numérotation internationale (avec+).
  - 129 : Les autres cas.
- ATD<option> : Envoi d'un appel vocal ou autre.
- ATT : Numérotation fréquentielle.
- ATP : Numérotation par impulsions.
- ATA : Décroche la ligne lors d'un appel en cours.
- ATH : Raccroche la communication en cours.
- AT+CHUP : Version GSM de la commande H, (cf. +CVHU).

#### II.5.3.5. Les options de la commande D

- Les caractères utilisables : 1 2 3 4 5 6 7 8 9 0 \* # + A B C
- Les caractères non utilisés : D , T P ! W @
- Les caractères spéciaux :
  - Le caractère ; lance un appel vocal
  - Le caractère > recherche un numéro dans l'annuaire.
  - I ou i : supplementary service subscription, cf. +CLIR.
  - G ou g : supplementary service information, cf. +CCUG.

#### II.5.4. Capteur de mouvement PIR

Les capteurs PIR vous permettent de détecter un mouvement, presque toujours utilisé pour détecter si une personne est entrée ou non dans la plage des capteurs. Ils sont petits, peu coûteux, de faible puissance, faciles à utiliser et ne s'usent pas. Pour cette raison, ils se trouvent couramment dans les appareils et les gadgets utilisés dans les maisons ou les entreprises. Ils sont souvent appelés capteurs PIR, "infrarouge passif", "pyroélectrique" ou "mouvement IR".



Figure II.32 Capteur PIR

### II.5.4.1. Principe de fonctionnement du capteur PIR

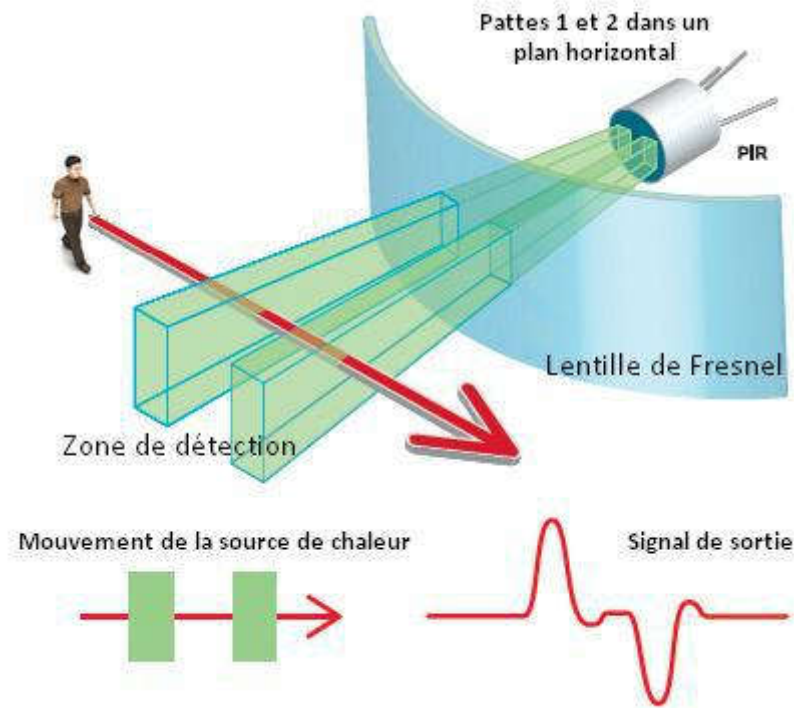


Figure II.33 Fonctionnement d'un capteur PIR

Un capteur PIR détecte les modifications de la quantité de rayonnement infrarouge qui l'atteint, qui varient en fonction des caractéristiques de température et de la surface des objets situés devant le capteur [18].

Lorsqu'un objet, tel qu'un humain, passe devant l'arrière-plan, tel qu'un mur, la température à cet endroit du champ de vision du capteur augmentera de la température ambiante à la température corporelle, et ainsi de suite [18].

Le capteur convertit la modification résultante du rayonnement infrarouge entrant en une modification de la tension de sortie, et cela déclenche la détection. Les objets de température similaire mais de caractéristiques de surface différentes peuvent également avoir un motif d'émission infrarouge différent, et leur déplacement par rapport à l'arrière-plan peut également déclencher le détecteur [18].

### II.5.5. L'afficheur I2C LCD :

Cet afficheur LCD va permettre d'afficher deux lignes de 16 caractères, il dispose en plus d'un rétro-éclairage RGB, ce qui permet de choisir la couleur du fond. La connexion s'effectue à travers une liaison Grove avec seulement 4 fils (GND, DATA, SCL, SDA), idéal pour ne pas utiliser trop de pins sur une carte Arduino ou Raspberry Pi.



Figure II.34 L'afficheur I2C LCD

### II.5.5.1. Caractéristiques d'un I2C LCD

Tension d'entrée : 5V  
CGROM: 10880 bits  
CGRAM: 64 \* 8 bits  
Rétro éclairage RVB  
Polices anglaise et japonaise incluses  
Communication I2C  
Reset automatique à l'allumage  
Interface I2C (adresse 0x27)  
Caractères blancs sur fond bleu  
Contraste ajustable via potentiomètre  
Dimensions: 80 x 38 x 18 mm

### II.5.5.2. Pourquoi choisir I2C dans notre projet :

Si une interface lisible simple est nécessaire dans un projet. L'affichage à cristaux liquides alphanumérique peut être le plus préféré en raison de divers facteurs, tels que le faible coût, facilement programmable, n'ayant aucune limitation pour l'affichage de caractères spéciaux et même personnalisés.

Mais pour l'utiliser directement avec un microcontrôleur, il faut disposer d'au moins 6 broches - RS, EN, D7, D6, D5 et D4 (avec la broche R / W mise à la terre en permanence pour l'opération d'écriture). Pour un projet complexe regroupant un grand nombre de modules, on peut être confronté à une pénurie de broches pour notre microcontrôleur. Dans de telles situations, l'écran LCD I2C est pratique car il n'utilise que 2 broches (SDA et SCL) pour communiquer avec la MCU, ce qui permet d'économiser au moins 4 broches. Si on possède déjà des périphériques I2C, ce module LCD n'utilise plus de broches du tout.

### II.5.5.3. Sac à dos LCD I2C :

On peut l'utiliser avec des modules LCD dotés d'une interface compatible HD44780 avec différentes tailles d'écran. La clé est que l'écran LCD doit avoir les pads d'interface dans une seule rangée de seize, de sorte qu'il corresponde aux broches du sac à dos.

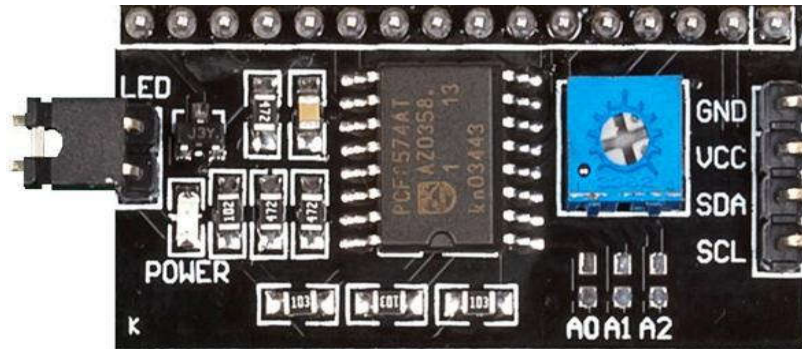


Figure II.35 Sac à dos LCD I2C

Comme indiqué dans l'image ci-dessus, le sac à dos comporte 4 broches, à savoir GND, Vcc, SDA et SCL. En outre, le sac à dos a les caractéristiques clés suivantes :

- Le cavalier LED doit être en position pour allumer le rétro éclairage
- Le potentiomètre bleu sur le bord du sac à dos règle le contraste. Il faut l'ajuster sur une position où les caractères sont clairs et où l'arrière-plan ne comporte pas de cases derrière les caractères.
- Les trois broches d'adresse matérielle A0, A1 et A2 permettent à huit périphériques d'être sur le même bus I2C. Vous pouvez modifier l'adresse de l'appareil en fonction de l'état de chacune de ces broches.

Addicore I2C Backpack Addresses				
PCF8574AT	PCF8574T	Solder Jumpers		
I2C Address	I2C Address	A0	A1	A2
0x3F	0x27			
0x3E	0x26	bridged		
0x3D	0x25		bridged	
0x3C	0x24	bridged	bridged	
0x3B	0x23			bridged
0x3A	0x22	bridged		bridged
0x39	0x21		bridged	bridged
0x38	0x20	bridged	bridged	bridged

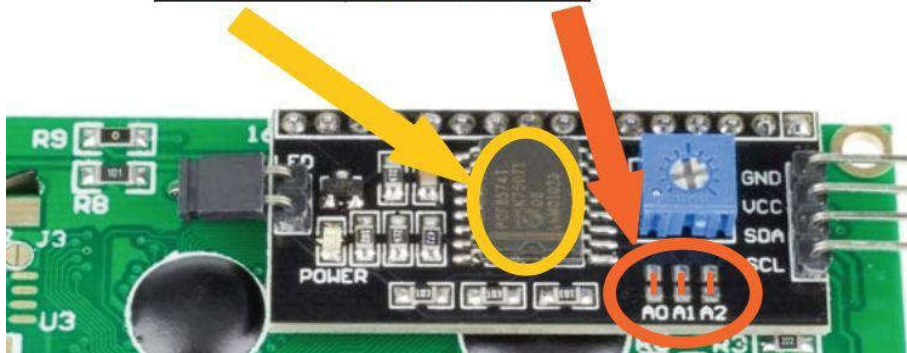


Figure II.36 Adresses I2C

L'adresse par défaut du périphérique est 0X3F (avec PCF8574AT). Les modules de sac à dos centrés autour de PCF8574 / PCF8574T (package SO16 de PCF8574 dans le package DIP16) ont une adresse esclave par défaut de  $0 \times 27$ . Pour ceux avec une puce PCF8574A / PCF8574AT, l'adresse d'esclave par défaut passera à 0x3F

Une fois que toutes les connexions nécessaires sont établies, on peut communiquer avec l'écran LCD en utilisant I2C.

### **II.6. Conclusion**

Dans ce chapitre nous avons présenté la partie théorique des différents modules constituant notre circuit électronique qui consiste un système de sécurité. Nous avons organisé notre chapitre par la description de la carte Arduino puis le GSM A6 ensuite le capteur de mouvement et le Raspberry Pi ainsi le langage python et on a terminé par l'afficheur LCD I2C. Dans le chapitre suivant, nous présenterons nos résultats de simulation à base de la carte Arduino et Raspberry Pi, et la partie expérimentale

# **Chapitre III :**

# **Conception et**

# **Analyse**



### Introduction

Pour qu'un système de surveillance embarqué en temps réel soit utilisé pour une surveillance et une alerte efficaces, le système doit savoir au moins trois fonctions qui sont les suivantes: détection, traitement d'image, et mécanisme d'alerte.

Ce système de sécurité basé sur Raspberry Pi est donc composé de principalement deux parties.

Ce sont: le matériel de conception et les logiciels de conception

### III.1. Matériel de conception (Installation et configuration des modules du système).

L'ensemble des modules du système est composé de :

- Un Raspberry Pi 3 Model B,
- Un Arduino UNO
- Capteur de mouvement PIR,
- Un Module GSM/GPRS A6
- Un afficheur LCD 2 X 16
- Une Camera USB

### III.2. Mettre à jour le système de la Raspberry Pi

Pour mettre à jour Raspbian vers une version compatible avec la Raspberry Pi 3, il faut directement chercher dans ce que l'on appelle les « dépôts » officiels de Raspbian les mises à jour pour les logiciels déjà installés.

Pour chercher ces mises à jour, il suffit d'utiliser la commande suivante :

```
# sudo apt-get update
```

Cette commande va récupérer la liste des mises à jour à effectuer, une fois la commande terminée, nous allons télécharger et installer ces mises à jour sur la Raspberry Pi à l'aide de la commande :

```
# sudo apt-get upgrade
```

Comme la Raspberry Pi va devoir récupérer et installer toutes les mises à jour des logiciels installés sur Raspbian, cela peut-être un peu long,

### III.3. Activer la camera USB (webcam)



Figure III.1 Camera USB (Webcam)

Une webcam, est une caméra conçue pour être utilisée comme un périphérique d'ordinateur, Pour pouvoir utiliser et activer notre camera USB il faut utiliser une simple et petite application qui est la *fswebcam*

Il est préférable de vérifier si notre webcam est acceptable avec RPi, pour cela il faut connectez la Webcam au port USB du RPi et taper la commande :

```
# lsusb
```

La liste des périphériques USB est affichée comme suite :

```
pi@raspberrypi:~$ lsusb
Bus 001 Device 005: ID 1e4e:0109 Cubeternet
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast
Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~$
```

Si la sortie de la commande ne répertorie pas la webcam, il est possible que le Raspberry Pi ne fournisse pas assez d'énergie pour cette Webcam. Dans ce cas, il faut essayer d'utiliser une ligne d'alimentation distincte pour la Webcam, telle qu'un concentrateur d'alimentation USB,

### III.3.1. Serveur webcam avec *fswebcam*

*fswebcam* est le moyen le plus simple et le plus rapide de capturer des images uniques avec une webcam. Les paramètres de *fswebcam* sont très complets, et permettent entre autre de régler la capture à un intervalle de temps spécifié, de mettre un ou des bandeaux contenant du texte, supprimer l'image, etc...

L'installation de *fswebcam* est en utilisant la commande :

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

Après avoir installé *fswebcam* sans erreur, on peut capturer assez facilement en tapant la commande suivante:

```
# fswebcam nom_capture.jpg
```

La commande peut bien sûr être détaillée par des paramètres personnalisés, par exemple :

- **-r** : résolution de hauteur/largeur
- **-S** : nombre d'image à écarter pour que la webcam puisse capturer une image stable et bien éclairée
- **--flip** : retournement d'image, par défaut horizontal
- **--jpeg** : qualité de la compression jpeg
- **--shadow** : ajoute une ombre portée au texte de bas de page
- **--title**, **--subtitle**, **--info** : différentes zones du pied de page
- **--save** : chemin et nom du fichier où seront sauvegardées les images  
Le nom du fichier suffit pour enregistrer l'image.
- **-q** : laisse le processus en cours d'exécution en arrière-plan

- **-l** : prend un instantané à chaque intervalle de temps (en secondes)
- **--top-banner** : Cette option sert à placez la bannière en haut de l'image.
- **--bottom-banner** : Cette option sert à placez la bannière au bas de l'image.  
C'est la valeur par défaut.

Pour notre travail on a utilisé la commande :

```
# fswebcam -d /dev/video0 -r 1920X1080
```

Où `/dev/video0` présente le *videodevice* ou le chemin de la webcam à utiliser.

Cette commande à pour rôle de prendre une capture avec une webcam d'une résolution de 1920 par 1080 et sauvegardée dans le dossier `/home/pi/captures` et sous le nom `photo.jpg`

Pour pouvoir appeler cette commande à partir d'un programme python il faut utiliser le module `OS` qui fournit une manière portable d'utiliser les fonctionnalités dépendantes du système d'exploitation, son usage nécessite d'importer sa bibliothèque comme suivant:

```
import os  
os.system('fswebcam -d /dev/video0 -r 1920X1080 -save /home/pi/captures/photo.jp')
```

```
pi@raspberrypi:~ $ fswebcam -d /dev/video0 -r 1920x1080.jpg  
--- Opening /dev/video0...  
Trying source module v4l2...  
/dev/video0 opened.  
No input was specified, using the first.  
Adjusting resolution from 384x288 to 352x288.  
--- Capturing frame...  
GD Error: gd-jpeg: JPEG library reports unrecoverable error: Not a JPEG file: st  
arts with 0x0a 0x28Captured frame in 0.00 seconds.  
--- Processing captured image...  
Writing JPEG image to '-d'.  
Writing JPEG image to '/dev/video0'.  
GD Error: gd-jpeg: JPEG library reports unrecoverable error: Output file write e  
rror --- out of disk space?Writing JPEG image to '-r'.  
Writing JPEG image to '1920x1080.jpg'.  
pi@raspberrypi:~ $
```

Figure III.2 capture par fswebcam



Figure III.3 Image capturée et sauvegardée

### III.4. Configuration du capteur de mouvement PIR

Le capteur de mouvement principal a été utilisé pour contrôler l'ensemble du système. La plage de détection est de 7 mètres sur 140 degrés. Il a un délai de 16 secondes mais réglable. La température ambiante est 253K-323K. Il va être alimenté directement du Pi par la broche d'alimentation 5V CC. Sa sortie est connectée comme entrée à la broche GPIO 18 BCM programmable comme le montre la figure suivante

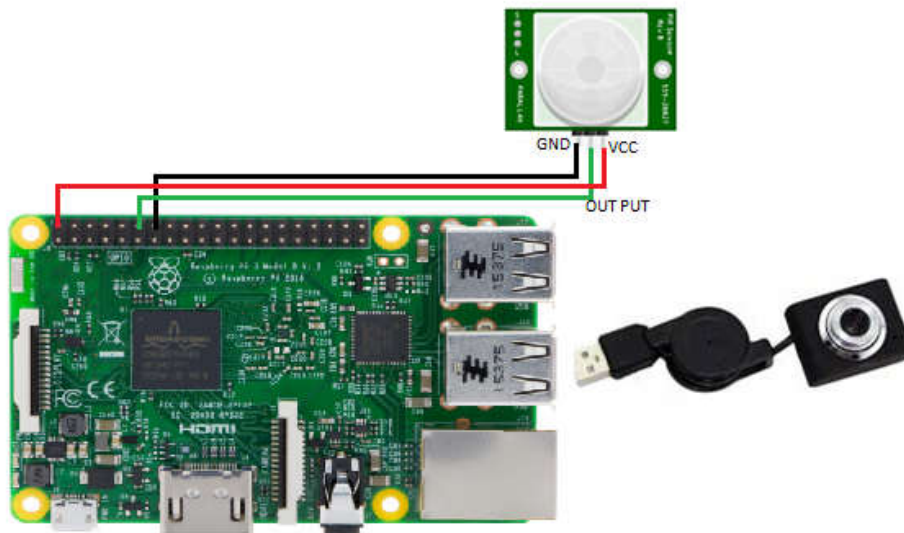


Figure III.4 Câblage du Capteur PIR avec Raspberry Pi

Notre objectif est de lire la sortie du capteur de mouvement PIR appliquée sur la GPIO 18. Pour cela il faut une initialisation et une configuration de notre système de sécurité qui implique l'importation des bibliothèques et de packages Python nécessaires. Ces bibliothèques sont prédéfinies et aident au bon fonctionnement des modules interfacés.

L'initialisation des broches GPIO est effectuée à l'aide de la numérotation des canaux BCM. Le GPIO 18 du capteur PIR a été défini en mode lecture (Entrée) alors que le GPIO 17 qui d'éclanche l'alerte a été configuré en mode lecteur / écriture (Sortie)

Pour utiliser la numérotation GPIO.BCM ou GPIO.BOARD on doit utiliser la commande :

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setmode(GPIO.BOARD)
```

Pour lire la valeur de n'importe quelle broche GPIO, il faut écrire simplement :

```
GPIO.input (pin)
```

Pour gérer un canal de broche GPIO il faut écrire :

```
GPIO.output (pin, statut)
```

Cette séquence d'événements d'initialisation et de configuration peut être bien élaborée à l'aide du diagramme ci-dessous.

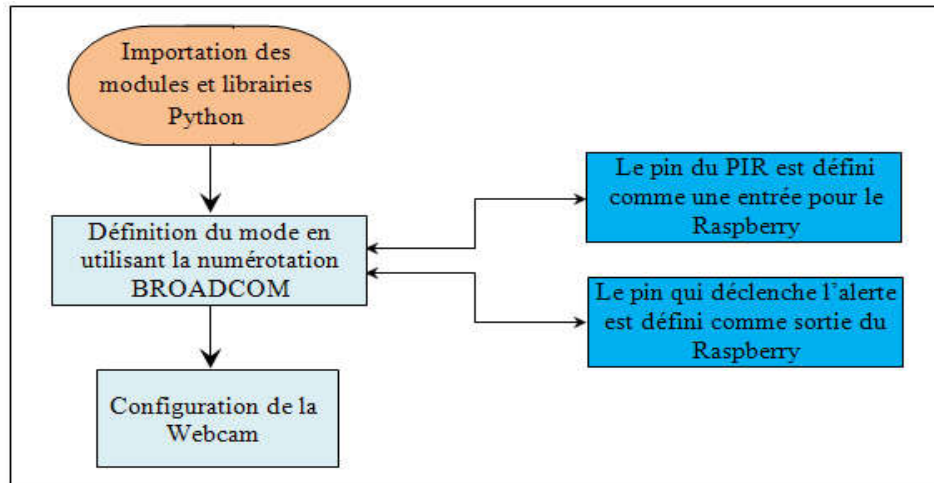


Figure III.5 : Initialisation et configuration du système

Le capteur émet un signal numérique à état haut (5V) lorsqu'il détecte un mouvement, cette valeur haute doit provoquer tous le reste avec le programme **mouvement.py** suivant :

```

import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(18, GPIO.IN)           #Lire la sortie du capteur PIR
GPIO.setup(17, GPIO.OUT)         #Le pin 17 va déclencher l'envoi d'alerte

while True:
    i=GPIO.input(18)
    if i==0:                       #Sortie du capteur PIR est à état bas
        print ('Aucun mouvement')
        GPIO.output(17, 0)         #Maitre la sortie GPIO 17 à état bas
        time.sleep(0.1)
    elif i==1:                     #Sortie du capteur PIR est à état haut
        print ('Mouvement detecte')
        GPIO.output(17, 1)         #Maitre la sortie GPIO 17 à état haut
        time.sleep(0.1)
    time.sleep(0.1)
    
```

### III.5. Connexion du LCD I2C avec Raspberry Pi

Connecter un écran LCD à un Raspberry Pi va pimenter presque n'importe quel projet, mais connectez simplement un écran LCD avec I2C utilise seulement deux broches (quatre si on compte les câbles Vcc et de terre).

#### III.5.1. Activer l'I2C sur le Pi

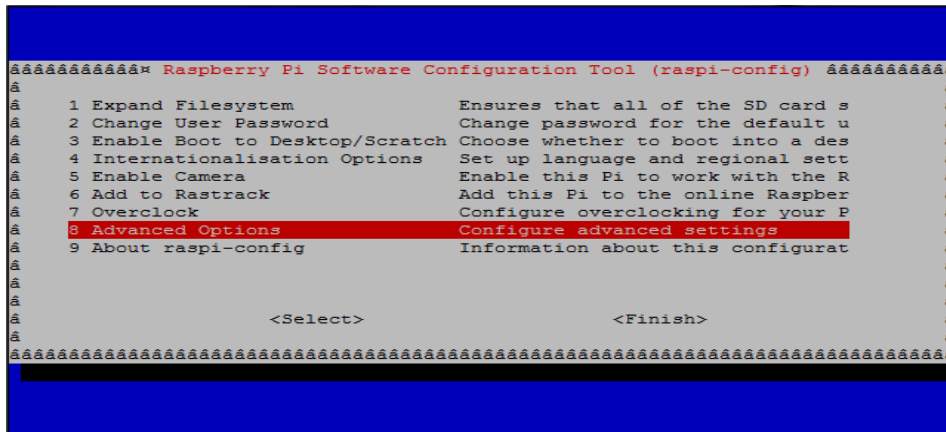
Avant de commencer la programmation, il faut s'assurer que le module I2C est activé sur le Pi et installer quelques outils qui faciliteront son utilisation

#### Activer I2C dans Raspi-Config

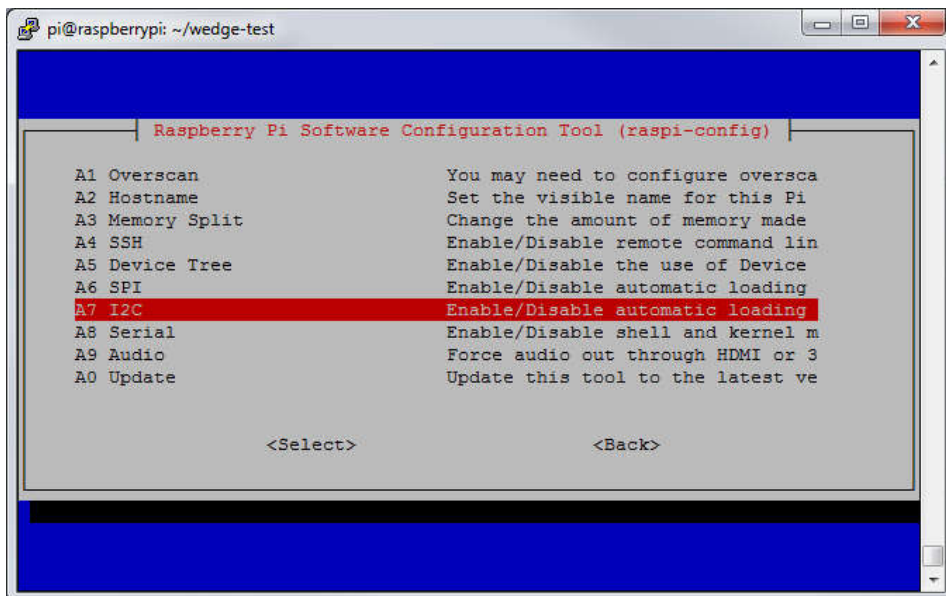
Tout d'abord, on doit nous connecter à notre Pi et entrez la commande suivante pour accéder au menu de configuration

```
# sudo raspi-config
```

Puis flèche vers le bas et sélectionnez «Paramètres avancés» ou « Advanced Options »



Maintenant la flèche vers le bas et sélectionnez “I2C Activer / Désactiver le chargement automatique



Choisissez «Oui» à l'invite suivante, quittez le menu de configuration et redémarrez le Pi pour activer les paramètres.

Il faut ensuite modifier le fichier de configuration du module. Pour cela on utilise la commande suivant:

```
$ sudo nano / etc / modules
```

Puis ajoutez les deux lignes suivantes dans le fichier de modules si elles n'existent pas

```
i2c-bcm2708
i2c-dev
```

Ensuite, Ctrl X et Oui pour enregistrer le fichier.

**Installez les bibliothèques smbus et i2c python:**

On utilise les commandes:

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y python-smbus i2c-tools
```

```
$ sudo reboot
```

Après avoir redémarré le système il faut taper la commande suivante pour vérifier l'installation du logiciel:

```
$ Ismod | grep i2c
```

Il faut avoir `i2c_bcm2708` dans une liste, cela signifie que la bibliothèque a été installée avec succès. Ensuite il faut effectuer un teste en exécutant l'une des commandes suivantes dans le terminal:

```
$ sudo i2cdetect -y 1
```

Ou bien

```
$ sudo i2cdetect -y 0
```

Qui va afficher :

```
pi@raspberrypi:~$ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  1d  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70: UU  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Le signe “- -” dans la liste sans aucun chiffre signifie que la connexion au circuit est incorrecte ou que le logiciel n'est pas correctement installé.

**III.5.2. Connexion I2C Raspberry Pi**

Le câblage de cet afficheur avec notre Raspberrv est comme suivant :

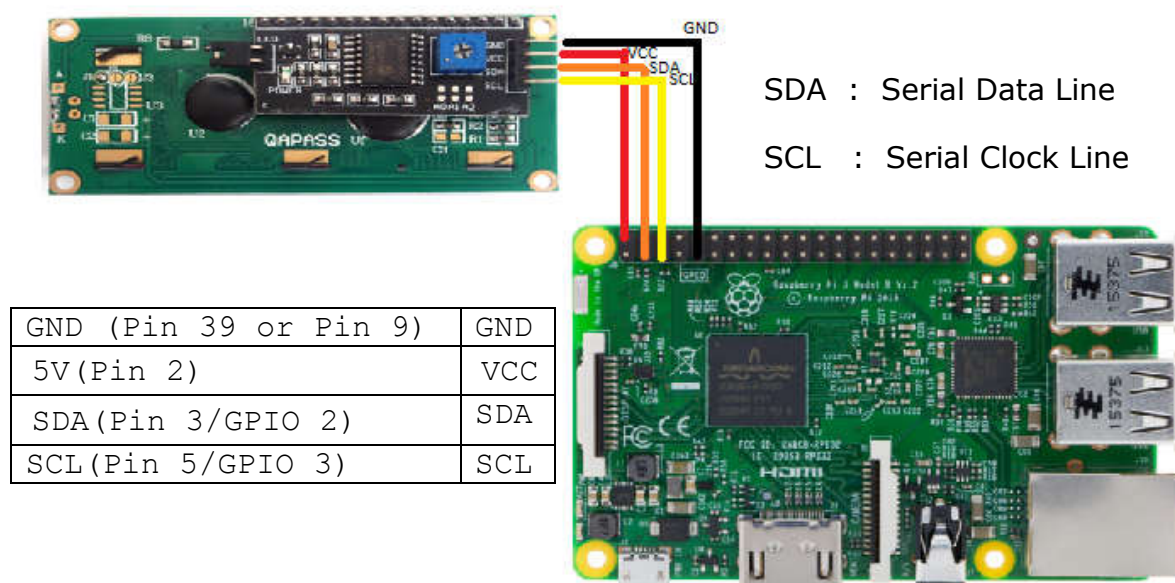


Figure III.6 Câblage d'afficheur I2C avec Raspberry Pi

Notre programme python permettant un affichage de l'état de détection du mouvement et de l'envoi d'alerte `i2c_display.py` est le suivant :

```
import smbus      # sous ensemble de protocole i2c
import time      # pour gérer les tâches liées au temps

I2C_ADDR = 0x27 # l'adresse du I2C, en cas d'erreur on change sur 0x3f
LCD_WIDTH = 16  # le Maximum de caractères par ligne est 16

LCD_CHR = 1     # Mode : Envoie de données
LCD_CMD = 0     # Mode : Envoie de commande

LCD_LINE_1 = 0x80 # Adresse LCD RAM pour la 1er ligne
LCD_LINE_2 = 0xC0 # Adresse LCD RAM pour la 2eme ligne
LCD_LINE_3 = 0x94 # Adresse LCD RAM pour la 3eme ligne
LCD_LINE_4 = 0xD4 # Adresse LCD RAM pour la 4eme ligne

LCD_BACKLIGHT = 0x08 # On #LCD_BACKLIGHT = 0x00 # Off

ENABLE = 0b00000100 # Activer le bit

E_PULSE = 0.0005
E_DELAY = 0.0005

bus = smbus.SMBus(1) # Rev 2 Pi uses 1 Ouvrir l'interface i2c

def lcd_init():

    # Initialisation d'affichage
    lcd_byte(0x33,LCD_CMD) # Initialisation 110011
    lcd_byte(0x32,LCD_CMD) # Initialisation 110010
    lcd_byte(0x06,LCD_CMD) # Direction du curseur 000110
    lcd_byte(0x0C,LCD_CMD) # Affichage On ,Curseur Off, Blink Off 001100
    lcd_byte(0x28,LCD_CMD) # Longueur de données, nombre de lignes 101000
    lcd_byte(0x01,LCD_CMD) # Effacer l'affichage 000001
    time.sleep(E_DELAY)

def lcd_byte(bits, mode):
    # Send byte to data pins
    # bits = the data
    # mode = 1 for data
    #       0 for command
    bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
    bits_low = mode | ((bits<<4) & 0xF0) | LCD_BACKLIGHT

    # bits hauts
    bus.write_byte(I2C_ADDR, bits_high)
    lcd_toggle_enable(bits_high)

    # bits bas
    bus.write_byte(I2C_ADDR, bits_low)
    lcd_toggle_enable(bits_low)

def lcd_toggle_enable(bits):

    time.sleep(E_DELAY)
    bus.write_byte(I2C_ADDR, (bits | ENABLE))
    time.sleep(E_PULSE)
    bus.write_byte(I2C_ADDR, (bits & ~ENABLE))
    time.sleep(E_DELAY)
```



```
def lcd_string(message,line):
    # Envoyer les characters à l'afficheur

message = message.ljust(LCD_WIDTH," ")
    lcd_byte(line, LCD_CMD)

for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)

def main():
    # block du programme principale
    # Initialiser l'afficheur

lcd_init()

    # Afficher Mouvement détecté
    lcd_string("mouvement",LCD_LINE_1)
    lcd_string("detecte",LCD_LINE_2)
    time.sleep(2)

    # Afficher email envoye
    lcd_string("email envoye",LCD_LINE_1)

    # Afficher sms envoye
    lcd_string("sms envoye",LCD_LINE_2)
    time.sleep(2)

    lcd_string("",LCD_LINE_1)
    lcd_string("",LCD_LINE_2)
    lcd_byte(0x01, LCD_CMD)

if __name__ == '__main__':
try:
    main()
    except KeyboardInterrupt:
        pass

finally:
    lcd_byte(0x01, LCD_CMD)
```

### III.6. Connexion du Module A6 GSM avec Arduino

Le module A6 GSM est connecté avec Arduino à l'aide de 4 pins :

- U\_TX et U\_RX avec les pins 0 et 1 (RX et TX) de la carte Arduino,
- VCC du module A6 GSM avec PWR,
- GND du module avec GND de la carte.

Le pin 2 de la carte Arduino est configuré comme input et connecté avec le GPIO 17 qui renvoie une valeur haute si le PIR détecte un mouvement; donc on n'aura pas un appel que si le mouvement est détecté et l'entrée PIN2 est mise à 1.

Le câblage du A6 GSM avec la carte arduino est présenté comme suite :

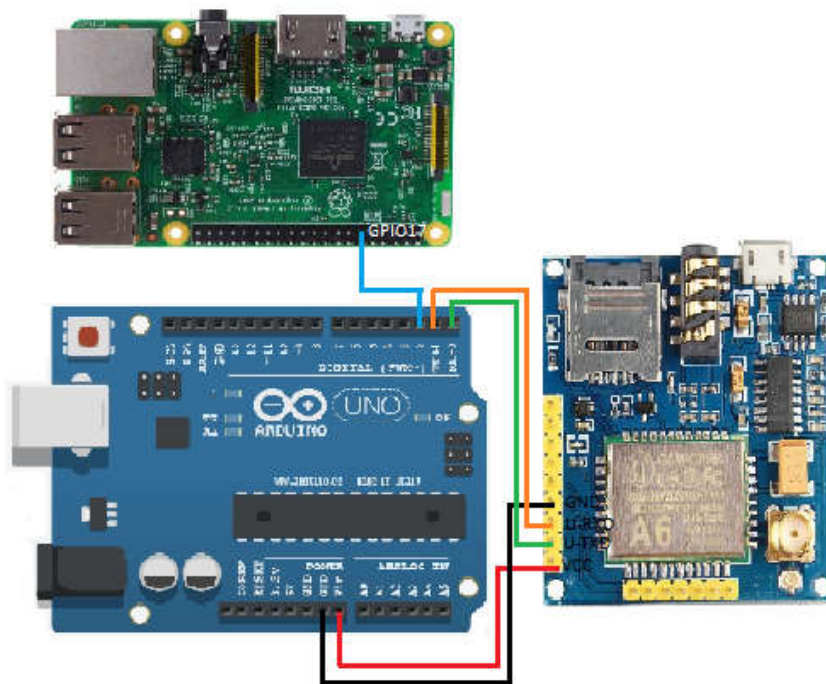


Figure III.7 Connexion du module A6 GSM avec Arduino

Le programme Arduino permettant d'effectuer un appel avec le module A6 GSM suivant la condition de détection de mouvement est :

```

phone_no[]="entrer le numéro de téléphone";
    int RP= 2;
    int val=0;
void setup() {
  pinMode(RP, INPUT);
  Serial.begin(9600);
  delay(200);
}
void loop() {
  val=digitalRead(RP);
  if (val==HIGH){
    Serial.println("AT");
    delay(1000);
    Serial.print("ATD");
    Serial.println(phone_no);
    delay(20000);
    Serial.println("ATH");
    val=0;
  }
}
    
```

L'instruction **Serial.println("AT");** permet d'initialiser de configurer le débit automatique,  
 L'instruction **Serial.print("ATD");** permet de composer le numéro de téléphone à l'aide de la commande ATD et l'instruction **Serial.println("ATH");** permet de raccrochez l'appel après un délai de 10 secondes,

### III.7. La génération et l'envoi du courrier électronique

L'envoi d'e-mails avec un Raspberry Pi sous programme Python peut s'avérer très utile si on souhaite envoyer une alerte, recevoir un ensemble de résultats ou savoir si un ensemble de conditions a été atteint ... etc.

Pour ce faire, il faut créer un script python qui utilise la bibliothèque `smtplib` du protocole SMTP (Simple Mail Transfer Protocol) qui gère l'envoi des courriers électroniques et leur routage entre les serveurs de messagerie.

L'installation du SMTP est effectuée à l'aide de la commande :

```
# sudo apt-get install ssmtp  
smtplib est importée par l'instruction
```

```
import smtplib
```

`smtplib` définit un objet de session client SMTP pouvant être utilisé pour envoyer des messages à n'importe quel ordinateur internet doté d'un démon écouteur SMTP ou ESMTP.

Pour obtenir plus qu'un simple courrier électronique, nous aurons besoin d'inclure des modules supplémentaires comme *email.mime.multipart* et *email.mime.image*

Pour envoyer un simple email sans objet, nous pouvons utiliser le code suivant

```
import smtplib  
  
server = smtplib.SMTP('smtp.gmail.com', 587)  
server.starttls()  
server.login('FromUser@gmail.com', 'Password')  
msg = "Un simple teste email"  
server.sendmail('FromUser@gmail.com', 'ToUser@gmail.com', msg)  
server.quit()
```

Dans cet exemple, on utilise le serveur smtp gmail avec le port 587. Pour l'utilisation d'une adresse Outlook ou Yahoo, il faut remplacer `smtp.gmail.com` par l'un des éléments suivants:

`smtp-mail.outlook.com`

`smtp.mail.yahoo.com`

Pour utiliser ce programme il faut remplacer *FromUser@gmail.com* par l'adresse email utilisée pour envoyer l'alerte et 'Password' par son mot de passe, ainsi que *ToUser@gmail.com* par l'adresse de la personne qu'on veut alerter

#### Email avec une ligne d'objet

Un simple courrier électronique peut suffire à alerter mais si on souhaite que le courrier électronique ait l'air un peu plus professionnel, on peut lui ajouter un objet en ajoutant la bibliothèque `MIMEMultipart` qui permet de définir les différentes parties d'un email comme suivant :

```
import smtplib
from email.mime.multipart import MIMEMultipart

fromaddr = 'FromUser@gmail.com'
toaddr = 'ToUser@gmail.com'

msg = MIMEMultipart()
msg['From'] = fromaddr
msg['To'] = toaddr
msg['Subject'] = 'Teste Alerte'
body = 'Un simple teste email'

server = smtplib.SMTP('smtp.gmail.com', 587)
server.ehlo
server.starttls()
server.login(fromaddr, "Passw")
text = msg.as_string()
server.sendmail(fromaddr, toaddr, text)
server.quit()
```

### Email avec une pièce jointe:

Pour aller plus loin si on souhaite inclure une pièce jointe dans notre courrier électronique, et qui est la photo de l'intrus dans notre projet il faut utiliser le package MIME (Multipurpose Internet Mail Extension) et importer la bibliothèque MIMEImage utilisé pour créer des objets de message MIME de type image.

MIME prend en charge les caractères qui ne font pas partie du code ASCII, les pièces jointes non textuelles (programmes audio, vidéo et d'application), ... etc. Il étend ainsi le format d'un email.

Donc pour envoyer un email avec une la photo capturée de l'intrus comme pièce jointe il faut utiliser le code suivant et qui est dans notre projet le programme PFE.py

```
from datetime import datetime
import os
import smtplib
from email.MIMEText import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage

Os.system('fswebcam -d /dev/video0 -r 1920x1080 --save home/pi/motion.jpg')

Fonc_time=datetime.now().strftime('%a %d %b')

to='nedromhafsa@gmail.com'
Sender='sidahmedsoltani19@gmail.com'
Pasw='mot de passe de l'expéditeur'
Body='mouvement detecte'
Subject='motion captur;' +fonc_time

att=open('/home/pi/motion.jpeg', 'rb')
ima=MIMEImage(att.read())
att.close()
ima.add_header('content-ID','<image>')
msg.attach(ima)
```

```
msg=MIMEMultipart()
msg('subject')=subject
msg('from')=sender
msg('to')=to
msg.preamble= 'photo' +fonc_time

ses=smtplib.SMTP('smtp.gmail.com',587)
ses .starttls()
ses.login(sender,pasw)
ses.sendmail(sender,to,msg.as_string(),body)

print 'mouvement detecte'
print 'sms envoyer'
print 'email envoyer'
```

L'instruction suivante est utilisée pour effectuer une capture de taille 1920x1080 et la sauvegarder dans le dossier /home/pi/ et sous le nom capture.jpg'

```
os.system('fswebcam -d /dev/video0 -r 1920x1080 --save /home/pi/capture.jpg')
```

Et la suivante pour afficher la date et le temps exacte de la capture dans l'objet de l'email

```
f time = datetime.utcnow().strftime('%a %d %b %H:%M:%S')
```

### III.8 Conception du système

Pour notre projet pour envoyer cet email en fonction des autres paramètres du mouvement on doit appliquer l'algorithme suivant :

#### Algorithme :

- 1- Vérifiez l'état de la broche GPIO 18 du PIR configurée en entrée pour le Raspberry. Si la broche est à état bas, la broche de sortie GPIO 17 doit rester à état bas et le système est inactif. Sinon si la broche devient soudainement à état haut: détection d'un mouvement et la broche GPIO 17 est à état haut.
- 2- Lorsque la valeur de la broche GPIO 17 configurée en sortie est à état haut (détection d'un mouvement), cette instance va appeler la fonction qui démarre la webcam.
- 4- La camera prend une capture de la vidéo et la sauvegarde dans un fichier.
- 5- Le système vérifie si Internet est activé sur le Raspberry Pi.
- 6- Si Internet est disponible, envoyez un courrier électronique à un hôte de messagerie prescrit. S'il n'y a pas d'Internet, attendez 2 secondes, puis affichez un message d'erreur et réinitialisez la broche 18 du capteur PIR à état bas. Cela va renvoyer le programme à la boucle principale.

#### III.8.1. Organigramme du système de sécurité basé sur Raspberry Pi

L'organigramme présenté ci-dessous illustre la série d'événements à partir d'un événement d'intrusion jusqu'au moment où il envoie une alerte.

C'est un organigramme qui doit être implémenté à l'aide d'un script Python.

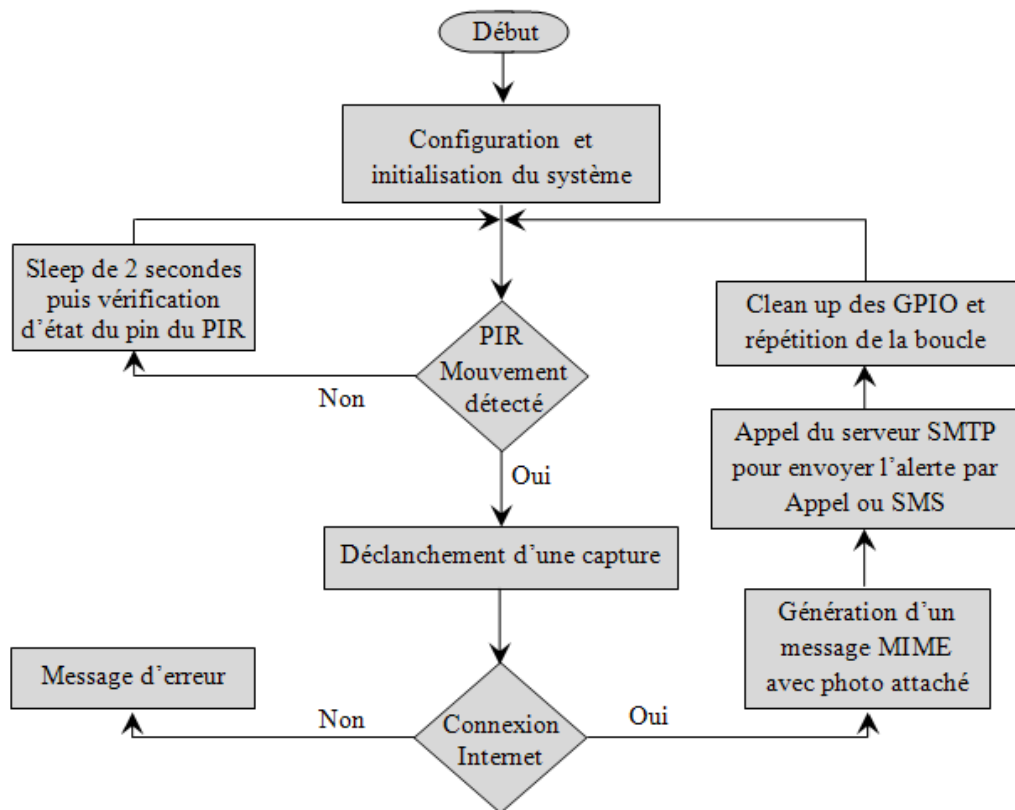


Figure III.8 Organigramme du système

La figure suivante présente le câblage des différents modules constituant ce système :

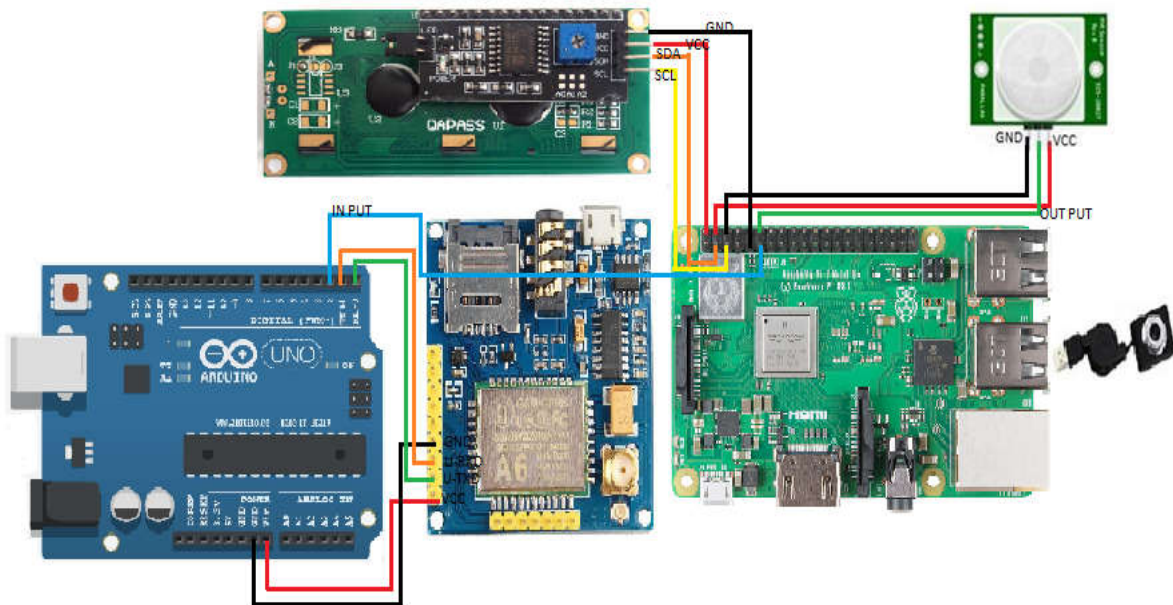


Figure III.9 Circuit globale du système

### III.8.2. Développer un code source complet

Pour pouvoir développer le script Python qui exécute l'algorithme défini dans l'organigramme, Les opérations suivantes ont été effectuées à la CLI du Raspberry Pi:

- Démarrage du Pi création d'un répertoire avec la commande `mkdir`
- À l'intérieur du répertoire, un fichier a été créé à l'aide de la commande `touch` et rendu exécutable à l'aide de

```
# sudo chmod + x(nom du fichier)
```

La commande `nano` a ensuite été utilisée pour ouvrir l'éditeur et le code Python complet y a été écrit par l'instruction :

```
# sudo nano nom_fichier.py
```

Le script a été exécuté à l'aide de la commande suivante:

```
# sudo python nom_fichier.py
```

Donc le script complet de notre système est le suivant :

```
import RPi.GPIO as GPIO
import os
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

Pir_sensor=18
Led=17
GPIO.setup(pir_sensor, GPIO.IN , GPIO.PUD_DOWN)
Current_state =0
GPIO.setup(led,GPIO.OUT)

While True :
    Try :
        current_state = GPIO.input(pir_sensor)
        if current_state == 1 :
            print('motion detecte')
            GPIO.output(led,True)
            os.system('sudo python pfe.py') # Envoie d'email

            print ('appel')
            time sleep(2)
            GPIO.output(led,False)
            os.system('sudo python i2c_display.py') #Affichage LCD I2C

            current_state == 0 :
                time.sleep(4)
        except KeyboardInterrupt : print('conexion non disponible')
            GPIO.cleanup()
```

Le programme a été divisé en 3 parties :

1<sup>ère</sup> partie : il y a trois importations qui initialisent le fonctionnement du GPIO, les fonctions de gestion du temps et la moyenne portable OS, ensuite, il y a la configuration de la

numérotation du port GPIO dans le mode BCM et on termine par éviter l'affichage des messages d'erreur inutile

2<sup>ème</sup> partie : déclarer 2 broches, une pour l'entrée connectée au GPIO18, et l'autre broche qu'on a nommé led pour la sortie connectée au GPIO 17 puis Configurer l'instance GPIO avec GPIO 18 et marquez-la comme périphérique d'entrée ensuite, nous annonçons la variable que nous devons utiliser dans le programme c'est *current\_state* qui sera utilisé pour enregistrer l'état de sensibilité, Nous fixons la valeur initiale à 0. Configurer led comme périphérique de sortie.

3<sup>ème</sup> partie : tester les signaux provenant du PIR, si le signal est à état haut un message sera affiché sur l'écran (motion détecté), la GPIO 17 est à état haut et provoque l'exécution du programme de l'envoi d'email, après, un autre message sera affiché (appel) pour nous informer du passage de l'appel lorsque le signal est arrivé à la PIN2 de la carte Arduino. Après 2 seconds on a l'exécution du programme de l'afficheur LCD I2C, la valeur *current\_state* revient à 0 et après 4 seconds la sortie du PIR est réinitialisée.

Et à la fin de programme on fait l'extinction de la GPIO 17 à l'arrêt du programme

Le montage expérimental est montré dans la figure suivante



Figure III.10 Boîtier du système de sécurité





Figure III.11 Boîtier du système de sécurité

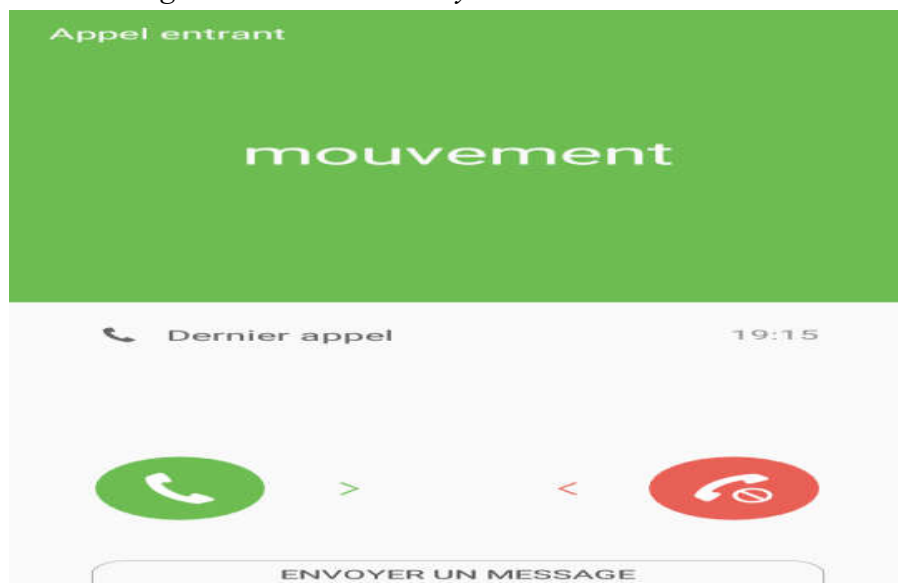


Figure III.12 Réception de l'appel

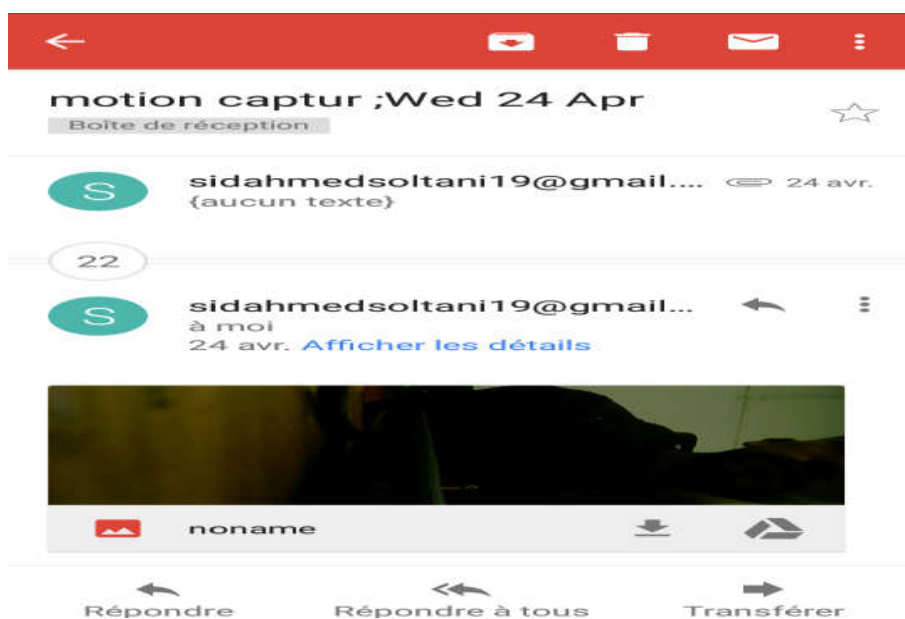


Figure III.13 Réception de l'email

### **III. 9. Conclusion**

Le projet a conçu et mis en œuvre un système de sécurité basé sur le Raspberry Pi. Les aspects du système sont les suivants: détection de mouvement à l'aide d'un capteur PIR, capture d'une photo à partir d'une vidéo à l'aide d'une caméra USB et envoi d'une alerte par courrier électronique.

Dans ce dernier chapitre nous avons présenté ce système avec ses différentes parties et leur principe de fonctionnement ainsi que sa réalisation pratique.

## Conclusion Générale

---

### Conclusion Générale :

Ce travail illustre le fonctionnement d'un système de vidéosurveillance intelligent à faible coût à base d'un Raspberry Pi 3B. Les aspects de ce système sont: la détection du mouvement à l'aide d'un capteur de mouvement PIR, capture d'une photo de l'intrus à l'aide d'une caméra USB et l'envoi d'une alerte par courrier électronique et par appel téléphonique à l'aide d'un module GSM A6 utilisé pour interfacer la fonctionnalité GSM / GPRS avec le processeur Raspberry Pi.

Nous avons rencontré quelques obstacles dans ce travail, tels que l'existence d'un défaut de certains composants, le manque des instruments dans les laboratoires, d'où la nécessité de les chercher ailleurs, le flux d'Internet faible au niveau des laboratoires, le langage de programmes que nous n'avions pas connus auparavant. Mais nous avons tiré parti de nouvelles informations et avons travaillé avec de nombreux programmes (Python, Linux, Arduino et Raspbian). Et nous nous sommes familiarisés avec les différents types de systèmes de sécurité.

Cependant, ce projet peut être amélioré, nous pouvons ajouter plus de paramètres à l'utilisateur pour lui donner plus de contrôle, comme par exemple le Node JS ou Open CV. Enfin, nous espérons que notre étude sera utilisée dans d'autres projets et développée.

## Références Bibliographique

- [1] Raspberry Pi for Begginers, 2014th ed. London UK.: Imagine Pulishing Ltd.
- [2] P. S. Dhake and B. Sumedha S., “Embedded Surveillance System Using PIR Sensor.,” vol. No. 02, no. 3, 2014.
- [3] J. D., “Real Time Embedded Network Video Capture And SMS Alerting system,” Jun. 2014.
- [4] S. Sneha, “IP Camera Video Surveillance using Raspberry Pi.,” Feb. 2015.
- [5] F. C. Mahima and A. Prof. Gharge, “Design and Develop Real Time Video Surveillance System Based on Embedded Web Server Raspberry PI B+ Board. International Journal of Advance Engineering and Research Development (Ijaerd), NCRRET.,” pp. 1–4, 2015.
- [6] J. G. J, “Design and Implementation of Advanced ARM Based Surveillance System Using Wireless Communication.,” 2014.
- [7] P. Sanjana, J. S. Clement, and S. R., “Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor.,” 2014.
- [8] U. Kumar, R. Manda, S. Sai, and A. Pammi, “Implementation Of Low Cost Wireless Image Acquisition And Transfer To Web Client Using Raspberry Pi For Remote Monitoring. International Journal of Computer Networking, Wireless and Mobile Communications (IJCNWMC).,” vol. No. 4, no. 3, pp. 17–20, 2014.
- [9] “The History of Security \_ PerspecSys.com.htm.”
- [10] A.-D. Osama, “Cisco IP Video Surveillance Introduction,” Cisco Expo, 2009.
- [11] “What is a security system and how does it work \_ SafeWise.htm.” .
- [12] T.K. Hareendran, “GSM Home Security Alarm System With Arduino,”Library Security System, 2014. .
- [13] R. Verman, “Distance Education In Technological Age,” Anmol Publ. Pvt Ltd, p. 166, 2005.
- [14] “Television Rides Wires,” Pop. Sci., no. February, p. 179, 1949..
- [15] “IP Surveillance,” IT Encyclopedia. .
- [16] S. Prasad, P. Mahalakshmi, A. J. C. Sunder, and R. Swathi, “Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor,” Int. J. Comput. Sci. Inf. Tech., vol. 5, no. 6, 2014.
- [17] G. Honey, Intruder alarms, 2nd ed. Oxford□ ; Burlington, MA: Newnes, 2003.
- [18] “How Infrared Motion Detector Components Work,” Glolab Corporation., 2013.

## Résumé :

Notre projet de fin d'étude consiste en un système de sécurité intelligent et à faible coût, qui permet de surveiller des biens envoyer des informations et des alertes à distance

La notification d'alerte est envoyée à l'utilisateur lorsque Raspberry Pi détecte un mouvement de corps vivant dans une zone de surveillance. Le Raspberry Pi transmet la capture de la video en temps réel via Internet et envoie un message de notification à l'utilisateur via le module GSM. Le Raspberry Pi ayant toutes ces interfaces utilisant des ports de communication tels que GPIO, Ethernet et USB.

Les étapes de conception de ce système sont mentionnées de manière simple et détaillé avec sa méthode de fonctionnement et explication des codes des programmes en plus de la réalisation pratique qui montre le bon fonctionnement de notre système.

À travers ce projet, une gigantesque expansion est anticipée avec de nouveaux domaines d'applications de la surveillance, la sécurité et villes intelligentes.

**Mots-clés** Surveillance intelligente, Raspberry Pi, Arduino, GSM A6, PIR, Webcam et afficheur LCD

## Abstract :

Our end of study project consists of an intelligent security system, that allows monitoring places and sending information remotely and sending alert notification with Email.

The alert notification is sent to user, when Raspberry Pi detects living body or motion of living body in surveillance area. Raspberry Pi transmits the real-time video through internet as well as it sends notification message to the user though GSM module. The Raspberry Pi having these all interfaces using communication ports like GPIO, Ethernet and USB.

Through this project, a huge expansion is anticipated with new areas of applications of surveillance, security and smart cities

Keywords: Smart Surveillance, Raspberry Pi, Arduino, GSM A6, PIR, Webcam and Display.

## ملخص:

يتكون مشروع نهاية الدراسة لدينا من نظام امني، إنه نظام ذكي للمراقبة وهو نظام يسمح لك بمراقبة المباني وإرسال المعلومات عن بعد ، في هذا المشروع قمنا بربط أربعة أجهزة مختلفة تستخدم لغات برمجة مختلفة (الراسبييري باي ، اردوينو ، نظام الاتصالات المتنقلة و حساس الحركة) إضافة إلى كاميرا ويب و شاشة عرض. يتم ذكر مراحل تصميم هذا النظام بطريقة بسيطة ومفصلة مع طريقة تشغيله وشرح رموز البرامج بالإضافة إلى الإدراك العملي الذي اظهر حسن سير نظامنا. من خلال هذا المشروع، من المتوقع التوسع الهائل في مجالات جديدة من تطبيقات المراقبة والامن والمدن الذكية. الكلمات المفتاحية: المراقبة الذكية،الراسبييري باي، اردوينو، نظام الاتصالات المتنقلة و حساس الحركة ،كاميرا ويب و شاشة عرض.