

République Algérienne Démocratique et Populaire
Ministère De L'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri De Tizi-Ouzou
Faculté De Génie Electrique et Informatique
Département d'Electronique



Mémoire de fin d'études

*En vue de l'obtention du diplôme MASTER Académique en Electronique
Option : Réseaux et Télécommunication*

Thème :

**Conception, étude et réalisation d'un système
domotique à base d'une carte Arduino Méga et d'une
carte Raspberry pi 2**

Présenté par :

ABBA Ahmed

Mémoire soutenu publiquement le 12/07/2016 devant le jury composé de:

Mr M.LAGHROUCHE

Mr H.HAMICHE

Mr K.Bennamane

Mr R.ZIRMI

Année Universitaire : 2015/2016

Remerciement

Je tiens à remercier énormément mon promoteur: Mr Hamiche enseignant à l'université MOULOUD MAMMERI de TIZI-OUZOU pour toute l'aide déterminante et l'intérêt qu'il m'a apporté tout au long de ce mémoire de fin d'étude.

Je tiens à remercier aussi Mr Laghrouche, les membres du laboratoire I2CSP à leur tête Mr Aidene et Mr Mouloudj pour leurs aides .

Pour finir je tiens à exprimer ma reconnaissance à tous les enseignants du département d'électronique et à tous les membres du jury pour avoir accepté de juger ce travail.

Dédicace

Je dédie ce travail à mes très chers parents qui m'ont élevé, soutenu et qui se sont sacrifiés pour moi tout au long de leurs vies malgré les difficultés. Mon chère père qui a toujours été mon exemple avec son courage, sa détermination et sa logique sans failles. A ma maman qui est la plus gentille et affectueuse des mères au monde qui a toujours été là pour moi et m'a donné la meilleure éducation. A ma chère grand-mère paternelle que je considère comme ma seconde mère. A mes trois autres grands parents hélas, que je n'est pas trop connu et que j'aurais aimé connaître un peu plus .

Je remercie infiniment Emilya qui a toujours été derrière moi durant ce travail qui est comme ma binôme. Je remercie Dieu d'avoir fait croiser nos chemins car après l'avoir connu ma vie est devenu un véritable paradis. J'ai jamais été aussi heureux de ma vie, tu m'a couvert de tendresse et d'amour au quotidien.

Je remercie aussi toute ma famille tantes, oncles, cousins et cousines pour leurs soutiens apportés chaque jour.

Je remercie également mes chers amis qui ont toujours été là pour moi et qui m'ont toujours apporté leurs aides et encouragement .

Je rend hommage a mes amis morts durant cette accident tragique à Tlemcen. On vous oublie pas, vous gardez une place spéciale dans nos cœurs.

Je remercie Dieu de m'avoir gardé en vie durant ce tragique accident me permettant de me retrouver ici en bonne santé parmi vous.

SOMMAIRE

Introduction générale.....	1
-----------------------------------	----------

Chapitre I :Généralités sur la domotique et la vidéosurveillance

I.1.Introduction.....	3
I.2. La domotique.....	3
I.2.1. Définition.....	3
I.2.2. Pourquoi la domotique ?.....	3
I.2.3. Technique de la domotique.....	4
I.2.4. Fonctionnement d'un système domotique.....	5
I.2.4.1. Les capteurs.....	5
a - Types de capteurs.....	6
b - Les familles de capteurs.....	7
I.2.4.2. Les actionneurs.....	7
I.2.4.3. La relation capteurs et actionneurs.....	7
I.2.5. Les fonctions de la domotique.....	8
I.2.5.1. Alarme intrusion.....	8
I.2.5.2. Gestion d'énergie.....	9
I.2.5.3. Scénarisation.....	9
I.2.5.4. Sécurité.....	9
I.2.5.5. Confort.....	9
I.2.6. Les courants porteurs.....	10
I.3. La vidéosurveillance.....	11
I.3.1.Introduction	11
I.3.2.Domaine d'application de la vidéosurveillance	11
I.3.3.Différents type de systèmes de vidéosurveillance.....	12
I.3.3.1.Vidéosurveillance en circuit fermé.....	12
I.3.3.2.Vidéosurveillance avec magnétoscopes.....	12
I.3.3.3.Vidéosurveillance analogique avec enregistreur numériques.....	12
I.3.3.4.Vidéosurveillance analogique avec enregistreur numériques réseau....	13
I.3.3.5.Vidéosurveillance sur IP avec serveur vidéo.....	13
I.3.3.6.Vidéosurveillance sur IP avec caméra IP.....	14
I.3.3.7.Vidéosurveillance avec webcam.....	15
I.4. Conclusion.....	16

Chapitre II :Acquisition, transfert et stockage d'images

II.1. Introduction	17
II.2. Acquisition d'images.....	17
II.2.1. Balayage entrelacé.....	17
II.2.1. Balayage progressif.....	17
II.3. Compression d'images.....	18

II.3.1. Normes de compression des images fixes.....	18
II.3.1. 1. Compression JPEG.....	18
II.3.1. 2. Compression JPEG 2000.....	19
II.3.2. Normes de compression des vidéos.....	20
II.3.2. 1. Vidéo obtenue par une suite d'images JPEG ou Motion JPEG (M-JPEG).....	20
II.3.2.2. Le protocole H.263.....	20
II.3.2.3. MPEG.....	20
II.3.2.4. MPEG-4 (Part 10).....	22
II.3.2.5. Avantages et inconvénients respectifs des normes Motion JPEG, MPEG-2 et MPEG-4.....	22
II.4. Résolution	23
II.4.1. Résolutions NTSC et PAL.....	23
II.4.2. Résolutions VGA.....	23
II.4.3. Résolutions MPEG.....	24
II.4.4. Evolution des cameras vers la résolution méga pixel.....	24
II.5. Les réseaux IP (Internet Protocol).....	24
II.5.1. Ethernet.....	24
II.5.1.1. Principe de transmission	24
II.5.2. Réseaux sans fils.....	25
II.5.2.1. WI-FI.....	26
II.5.2.2. Les différentes normes Wifi.....	26
II.5.2.3. Broadband Wireless Access.....	27
II.5.2.4. Sécurité sur les réseaux sans fil.....	27
II.6. Transmission de données.....	27
II.6.1. Les adresses IP.....	27
II.6.1.1. IPv4.....	27
II.6.1.2. IPv6.....	28
II.6.2. Modes de transmission de la vidéo sur IP.....	28
II.6.2.1. Diffusion individuelle (Unicast).....	28
II.6.2.2. Multidiffusion (Multicast).....	28
II.6.2.3. Radiodiffusion (Broadcast).....	28
II.6.3. Protocoles de transport de données pour la vidéo sur IP.....	28
II.6.4. Le streaming vidéo.....	30
II.6.4.1. FTP (File Transfer Protocol).....	30
II.6.4.2. HTTP (HyperText Transfer Protocol).....	31
II.6.4.3. Le protocole RTP.....	32
II.6.4.4. RTCP (Real-time Transfer Control Protocol).....	32
II.7. Bande passante.....	32
II.7.1. Réseaux commutés.....	33
II.7.2. Bande passante élargie.....	33
II.7.3. Fréquence évolutive.....	33
II.8. Stockage.....	33
II.8.1. Espace disque nécessaire.....	33

II.8.1.1. JPEG/Motion JPEG.....	34
II.8.1.2. MPEG-4.....	34
II.9.Redondance.....	34
II.9.1. RAID (Redundant Array of Independent Disks).....	35
II.9.2. Réplication des données	35
II.9.3. Sauvegarde sur bandes magnétiques.....	35
II.9.4. Mise en cluster de serveurs.....	35
II.9.5. Dédoublment des réceptionnaires vidéos	36
II.10.Sécurité de données	36
II.11.Visualisation via l'interface web.....	36
II.12.Conclusion	37

Chapitre III :Conception et étude du système

III.1. Introduction.....	38
III.2. Conception générale du système.....	38
III.3. Description détaillée du système.....	38
III.4. Carte de contrôle et sécurité du domicile.....	40
III.4.1. Carte de commande Arduino.....	40
III.4.1.1. Description générale de la carte arduino Méga.....	40
III.4.1.2. Le microcontrôleur ATmega2560.....	41
III.4.1.3. Communication avec l'extérieur de la carte Arduino.....	43
III.4.2. Composants utilisé avec la carte Arduino.....	44
III.4.2.1. Clavier matriciel.....	44
III.4.2.2. Afficheurs alphanumériques à cristaux liquides.....	44
III.4.2.3. Capteur de distance à Ultrasons HC-SR.....	46
III.4.2.4. La fonction du capteur ultrason dans notre système.....	47
III.4.2.5. Le bipper.....	48
III.4.2.6. Le servomoteur.....	48
III.4.2.7 La LED.....	50
III.4.2.8 La liaison Bluetooth.....	51
III.5. Carte de commande d'acquisition vidéo.....	52
III.5.1. La carte Raspberry Pi.....	52
III.5.2. Connectiques et ports disponibles sur la Raspberry Pi 2.....	53
III.5.3. Préparation de la carte SD.....	54
III.5.4. Les ports GPIO.....	54
III.5.5. Dimensions de la Raspberry Pi 2.....	55
III.5.6. La liaison avec la camera.....	56
III.5.7. La Raspberry Pi 2 sur le réseau.....	57
III.5.8. Communication série Raspberry Pi.....	57
III.6. Arduino et la réalisation pratique	57

III.7. Conclusion.....	58
------------------------	----

Chapitre IV : Réalisation expérimentale et description des outils logiciels

IV.1. Introduction.....	59
IV.2. Le logiciel Arduino.....	59
IV.2.1. Introduction.....	59
IV.2.2. Composition du logiciel Arduino.....	59
IV.2.3. Description des menus.....	61
IV.2.3 .1. Le menu File.....	61
IV.2.3.2. Le menu Edit (Editer).....	62
IV.2.3.3. Le menu Sketch.....	63
IV.2.3.4. Le menu Tools.....	63
IV.2.3.5. Le menu Help.....	64
IV.2.4. Présentation des boutons.....	64
IV.2.5. Les librairies.....	65
IV.2.6. Le terminal série.....	65
IV.2.7. Ouvrir un programme et le téléverser.....	67
IV.3. Le langage Python.....	71
IV.3.1. Syntaxe du Python.....	71
IV.4. Le Raspbian.....	73
IV.4.1. L'utilité de Raspbian avec la Raspberry Pi2.....	73
IV.4.2. Debian Wheezy.....	73
IV.4.3. Le terminal de la Raspberry pi 2.....	74
IV.4.4. Les commandes de la Raspberry.....	74
IV.4.5. Capture d'image avec fswebcam de la Raspberry PI 2.....	75
IV.4.6. Serveur HTTP ou Serveur WEB.....	77
IV.4.7. La Raspberry Pi 2 comme serveur web.....	77
IV.4.8. Serveur Apache.....	78
IV.4.8.1.Définition.....	78
IV.4.8.2.Installation du serveur Apache avec Raspbian.....	78
IV.4.8.2.1.Installation d'Apache.....	78
IV.4.6.2.2.Vérification du fonctionnement d'Apache.....	79
IV.4.9. Serveur PHP.....	79
IV.4.6.1.Introduction.....	79
IV.4.6.2. Installer PHP.....	79
IV.4.6.3. Vérifier que PHP fonctionne.....	79
IV.4.6.4. Sécurisation de notre page web avec le PHP.....	80
IV.5. Application Android.....	82
IV.5.1.définition.....	82
IV.5.2.Les différents autres systèmes d'exploitation mobiles sur le marché.....	82
IV.5.2.1.Symbian OS.....	82

IV.5.2.2.IOS.....	83
IV.5.2.3.BlackBerry OS.....	83
IV.5.2.4.Windows Phone.....	84
IV.5.3.Les avantages d'une application mobile.....	84
IV.5.4. App Inventor.....	85
IV.5.5. Commande vocal Google.....	85
IV.6. Conclusion.....	86
 Conclusion générale.....	 87
Annexe A : Détail sur la carte Arduino Méga.....	88
Annexe B : Programme Arduino du système conçu.....	92
Bibliographie.....	104

Listes des figures

Chapitre I :Généralités sur la domotique et la vidéosurveillance

Figure 1.1: Illustration d'une maison automatisée.....	3
Figure 1.2: Illustration sur le rôle de l'unité centrale dans une smart home.....	4
Figure 1.3: Illustration d'un écran de contrôle (tablette).....	5
Figure 1.4: Schéma fonctionnel d'un capteur.....	6
Figure 1.5: Illustration sur les différentes fonctions de la domotique.....	10
Figure 1.6: Schéma d'un système vidéosurveillance avec magnétoscope.....	12
Figure 1.7: Schéma d'un système vidéosurveillance analogique avec enregistreur numérique.....	13
Figure 1.8: Schémas d'un système vidéosurveillance analogique avec enregistreur numérique réseau	13
Figure 1.9: Schéma d'un système vidéosurveillance sur IP avec serveur vidéo.....	14
Figure 1.10: Schéma d'un système vidéosurveillance sur IP avec camera IP.....	15
Figure 1.11: Illustration de la vidéosurveillance avec une webcam.....	15

Chapitre II : Acquisition, transfert et stockage d'images

Figure 2.1: Différence entre le balayage entrelacé et balayage progressif.....	17
Figure 2.2: Différence entre un niveau de compression élevé et bas.....	19
Figure 2.3: Différence entre la compression JPEG et JPEG2000.....	19
Figure 2.4: Principe d'une séquence trois images avec JPEG.....	20
Figure 2.5: Principe du codage MPEG.....	21
Figure 2.6: Comparaison du rapport fréquence d'image et bande passante pour les normes M-JPEG et MPEG-4.....	23
Figure 2.7 : Connexion FTP entre un serveur et un client.....	30
Figure 2.8 : Communication entre un navigateur et un serveur.....	31

<u>Figure 2.9 :</u> Répartition des données sur plusieurs disques selon la méthode RAID.....	35
<u>Figure 2.10 :</u> Figure explicative sur la technique de réplication de données.....	35
<u>Figure 2.11:</u> Schéma explicative sur mise en cluster des serveurs.....	36

Chapitre III :Conception et étude du système

<u>Figure 3.1:</u> Schéma synoptique représentatif du système.....	39
<u>Figure 3.2:</u> Illustration de la carte Arduino Méga ADK.....	40
<u>Figure 3.3:</u> Schéma électrique et illustration réel du CPU ATMega256.....	41
<u>Figure 3.4 :</u> Illustration des différentes composant de la carte Arduino Méga.....	43
<u>Figure 3.5 :</u> Clavier matriciel ainsi que son schéma électrique.....	45
<u>Figure 3.6 :</u> L'afficheur LCD ainsi que sa connexion avec l'Arduino.....	45
<u>Figure 3.7 :</u> Fonctionnement du module ultrason.....	46
<u>Figure 3.8 :</u> Chronogramme de fonctionnement du capteur ultrason.....	47
<u>Figure 3.9:</u> Le module ultrason et sa connexion avec l'Arduino Méga.....	48
<u>Figure 3.10:</u> Illustration d'un bipper.....	48
<u>Figure 3.11:</u> Les composition interne d'un servomoteur.....	49
<u>Figure 3.12:</u> Illustration d'un servomoteur ainsi que sa connexion avec l'Arduino Méga.....	50
<u>Figure 3.13:</u> Illustration de différente LED ainsi que leur schéma électrique.....	51
<u>Figure 3.14:</u> Logo du Bluetooth.....	51
<u>Figure 3.15:</u> Le module Bluetooth ainsi que sa connexion avec l'Arduino Méga.....	52
<u>Figure 3.16:</u> Illustration de la Raspberry Pi 2.....	53
<u>Figure 3.17:</u> Schéma explicant les diffrentes fonctions des ports GPIO ainsi que leurs schéma électrique.....	55
<u>Figure 3.18:</u> Les dimensions de la Raspberry Pi 2.....	55

Figure 3.19: La carte Raspberry avec le module Raspicam	56
Figure 3.20: La carte Raspberry avec une webcam embarquée.....	56
Figure 4.21: Figure de la maquette conçue.....	57

Chapitre IV : Réalisation expérimentale et description des outils logiciels

Figure 4.1: Logo Arduino.....	59
Figure 4.2: Approche et utilisation du logiciel.....	60
Figure 4.3: Contenu du menu " File ".....	62
Figure 4.4: Contenu du menu "Edit".....	62
Figure 4.5: Contenu du menu "Sketch".....	63
Figure 4.6: Contenu du menu "tools".....	64
Figure 4.7: Contenu du menu "Help".....	64
Figure 4.8: Présentation des boutons.....	64
Figure 4.9: Présentation du terminal série.....	66
Figure 4.10: Ouverture du terminal série.....	66
Figure 4.11: Ajustage du débit de la communication série.....	67
Figure 4.12: Ouvrir le programme Blink.....	67
Figure 4.13: Contenu du programme Blink.....	68
Figure 4.14: Choix de la carte Arduino.....	68
Figure 4.15: Choix du port de connexion de la carte.....	69
Figure 4.16: Envoi du programme Blink.....	69
Figure 4.17: Fin de l'upload.....	70
Figure 4.18: Logo du Python.....	71
Figure 4.19 : Aperçu de l'interface Raspbian Wheezy.....	74
Figure 4.20 : Illustration du terminal de la Raspberry PI 2.....	74
Figure 4.21 : confirmation installation PHP.....	80
Figure 4.22 : Logo Android.....	82
Figure 4.23 : Logo Symbian OS.....	83
Figure 4.24 : Logo IOS.....	83
Figure 4.25: Logo BlackBerry.....	84
Figure 4.26 : Logo Windows Phone.....	84

<u>Figure 4. 27 :</u> interface du site App Inventor.....	85
<u>Figure 4. 28 :</u> interface graphique de notre application Android	86

Listes des tableaux

Chapitre I :Généralités sur la domotique et la vidéosurveillance

<u>Tableau 1.1:</u> Exemples d'actionneurs et de capteurs.....	8
---	---

Chapitre II : Acquisition, transfert et stockage d'images

<u>Tableau 2.1:</u> Variantes des technologies Ethernet suivant le type et le diamètre des câbles.....	25
<u>Tableau 2.2:</u> Les couches basses du modèle OSI pour la norme 802.11.....	26
<u>Tableau 2.3:</u> Protocole TCP/IP et port utilisés dans la vidéo sur IP.....	29

Chapitre III :Conception et étude du système

<u>Tableau 3.1 :</u> Caractéristiques du module et son fonctionnement.....	46
---	----

Chapitre IV : Réalisation expérimentale et description des outils logiciels et tests

<u>Tableau 4.1 :</u> Tableau comparatif entre la syntaxe du C et celle du Python....	77
---	----

Introduction générale

Introduction Générale

Ces dernières années, on a pu voir une évolution ascendante des technologies et de la télécommunication et vidéosurveillance. En effet, le confort et la sécurité des individus est devenu primordial dans le quotidien. Par conséquent, les inventions et les innovations ne cessent d'apparaître afin de faciliter la vie, ménager les efforts et surtout renforcer la sécurité. Cette dernière occupe une place importante pour la protection des domiciles, des personnes, des biens et des patrimoines. Ainsi la problématique est de trouver des solutions qui permettent de répondre à ces exigences.

Ce besoin croissant en confort et en sécurité conjugué aux avantages offerts par les systèmes domotiques et de vidéosurveillance sur IP, représente un moyen efficace pour le contrôle entier du domicile à distance, avec une sécurité renforcée et surtout la visualisation en temps réel des événements au sein de la demeure.

Dans le cadre de notre travail, les aspects liés à la technologie de la domotique et de la vidéosurveillance sur IP vont être étudiés. En effet, il s'agit de mettre en œuvre un système domotique utilisant une carte Arduino Méga, avec possibilité de contrôle via une application Android, incluant aussi un système de vidéosurveillance avec possibilité de visualisation en temps réel à distance réalisée avec une carte de type Raspberry PI 2.

Le système est composé principalement d'un digicode qui permet d'activer et de désactiver le système d'alarme avec un code bien précis. Si une infraction est détectée grâce aux capteurs lorsque le système d'alarme est activé, l'alarme émet un signal sonore. Une fois le système d'alarme désactivé, on peut accéder aux fonctions de contrôle via une application Android d'un Smartphone, par exemple l'allumage automatique des lumières avec commande vocale. En parallèle, le système de vidéosurveillance se charge de l'acquisition d'images puis le transfert des images prises dans une page web sécurisée à laquelle peut accéder avec un Nom d'utilisateur et un Mot de passe. Ainsi, tout appareil connecté au réseau ayant le Nom d'utilisateur et le Mot de passe, peut visualiser les images capturées.

Le travail développé ici, consiste d'une part à offrir des solutions adéquates et les résolutions des inconvénients journaliers. D'autre part, il doit répondre aux critères de fiabilité et d'efficacité, en accédant aux nouvelles technologies matérielles et logicielles évoluées. Ces dernières années des cartes de développement ont vu le jour dont les cartes Arduino Méga, Raspberry PI 2 et la technologie du Bluetooth.

Le mémoire est structuré en quatre chapitres. Dans le premier seront abordé les généralités et les définitions de la domotique et de la vidéosurveillance.

Dans le deuxième chapitre seront étudiés l'acquisition et le transfert ainsi que le stockage d'images.

Dans le troisième chapitre sera décortiqué la conception, étude et réalisation du système ainsi que la définition de chaque composant utilisé pour son élaboration.

Pour finir dans le quatrième et dernier chapitre, sera présenté la description des outils logiciels utilisés pour la conception du système.

Enfin, on termine par une conclusion générale et quelques perspectives.

Chapitre I

Généralités sur la domotique et la vidéosurveillance

I.1.Introduction:

La technologie ne cesse de se développer à chaque jour qui passe, elle nous procure quotidiennement du confort et de la sécurité . Dans ce chapitre sera abordé d'une part, la définition globale de la domotique ainsi que les techniques utilisées et ses offres dans la vie. D'autre part , la vidéosurveillance ainsi que les différents domaines de la vidéosurveillance et pour finir les différents systèmes disponibles .

I.2. La domotique [1]

I.2.1.Définition:

Associant le mot "domos" (maison en latin) et "informatique", la domotique, composante essentielle du confort électrique, regroupe l'ensemble des techniques visant à l'intégration des systèmes automatiques et à la communication dans l'habitat. Elle repose sur différentes fonctions (sécurité, confort, gestion d'énergie...), conçues pour offrir une meilleure qualité de vie, auxquelles il convient d'ajouter les applications liées à la communication (téléphonie, services Internet, réseaux informatiques). Cette technologie s'articule aujourd'hui autour d'un ensemble de solutions simples, pratiques et modulables, permettant d'automatiser les gestes quotidiens en fonction des besoins et des attentes de l'utilisateur final. Elle couvre les automatismes, la gestion des flux, la communication et les réseaux multiservices. La figure ci-dessous illustre une maison automatisée.

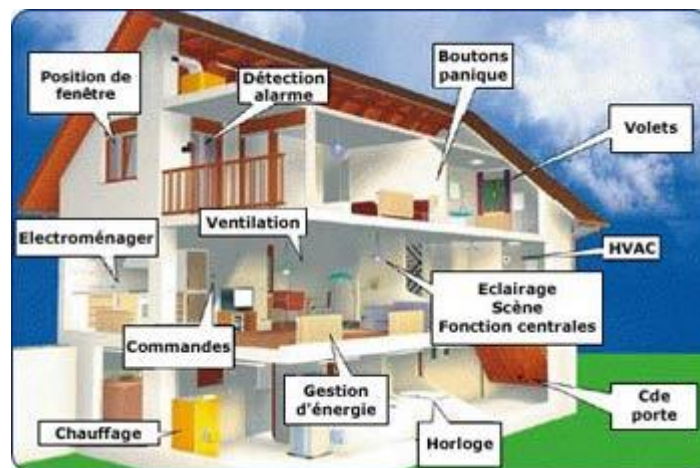


Figure 1.1: Illustration d'une maison automatisée

I.2.2.Pourquoi la domotique?

La domotique a pour but d'accroître l'économie d'énergie, le confort, la flexibilité, la communication et la sécurité dans l'habitation. La domotique peut libérer les gens de nombreuses activités routinières et faire en sorte que les conditions de vie soient optimales.

L'équipement technique de l'habitation fera en sorte de répondre de façon optimale aux attentes, sans qu'il faille effectuer des adaptations ou réglages.

I.2.3.Technique de la domotique [1]

La domotique est basée sur la mise en réseau par une « centrale de commande » des différents appareils électriques de la maison.

- **La centrale de commande** : Programmable et contenant des modules embarqués (passerelles domestiques) ou une interface micro-informatique (écran tactile, serveur, etc.) elle joue le rôle d'une « intelligence » centralisée et d'interface homme-machine centralisée pour l'utilisateur ou des services distants de contrôle. Elle tend à devenir plus réactive aux changements du contexte. Pour cela, elle réunit ou remplace divers appareils (programmeur/régulateur de chauffage, centrale d'alarme, système de centralisation des persiennes électriques, contacteur jour/nuit du cumulus, délesteur, programmeurs horaires, systèmes d'arrosage automatiques, etc.), qui peuvent fonctionner et interagir de manière *asynchrone*.(Voir figure ci-dessous)

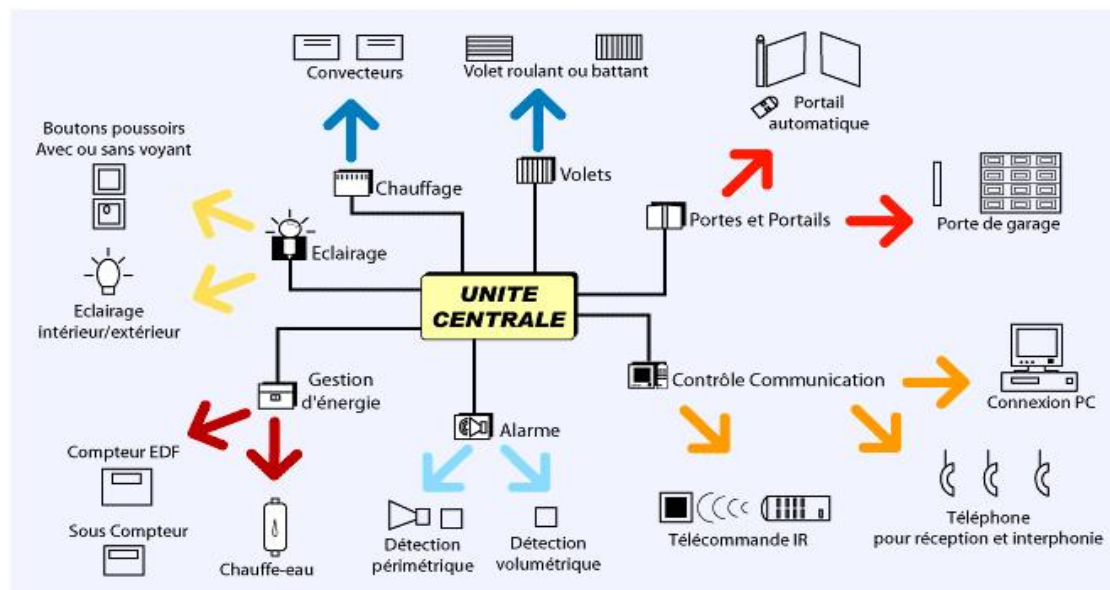


Figure 1.2: Illustration sur le rôle de l'unité centrale dans une smart home

- **L' écran de contrôle** : il est fixe dans le domicile, où il peut être émulé à distance (via le réseau ADSL de la maison puis l'internet), permettant le pilotage de la maison à distance pour tout ou partie des fonctions domotiques. L'interface distante peut être par exemple un ordinateur de poche, un téléphone portable ou Smartphone, une tablette tactile, une télécommande (universelle ou non), une interface sur télévision connectée, un écran + souris, etc. (Voir figure 1.3)



Figure 1.3: Illustration d'un écran de contrôle (tablette)

- **Le pilotage à distance** Un système domotique permet la communication non seulement à l'intérieur de la maison, mais aussi à l'extérieur. La technologie Internet interviendra de plus en plus pour la commande à distance par certains utilisateurs. Vous ne devez même pas être à la maison pour commander vos appareils. Un simple coup de fil ou un SMS vous permettra par exemple de régler le chauffage à distance,

I.2.4.Fonctionnement d'un système domotique

I.2.4.1.Les capteurs [2]

C'est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. On fait souvent (à tort) la confusion entre capteur et transducteur : le capteur est au minimum constitué d'un transducteur.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même, disposant d'un affichage ou d'un système de stockage des données. Le capteur, lui, en est dépourvu.

Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation. (Ceci est illustré par la figure 1.4)

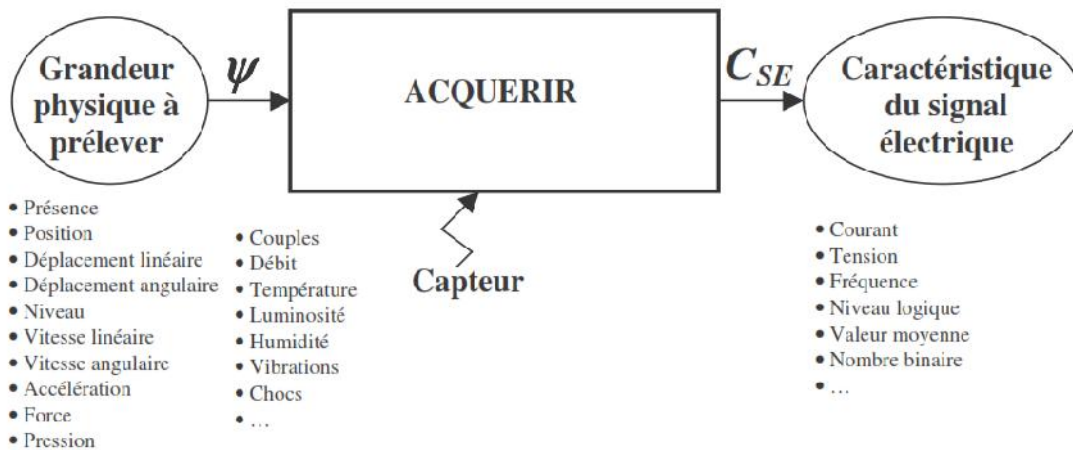


Figure 1.4: Schéma fonctionnel d'un capteur

a - Types de capteurs

• Capteurs actifs

On parle de capteur actif lorsque le phénomène physique qui est utilisé pour la détermination de la mesure effectue directement la transformation en grandeur électrique. C'est la loi physique elle-même qui relie la mesure et grandeur électrique de sortie.

Un capteur actif fonctionne assez souvent en électromoteur et dans ce cas, la grandeur de sortie est une différence de potentiel.

Le nombre des lois physiques permettant une telle transformation est évidemment limité, on peut donc recenser facilement les capteurs actifs (dont le nombre est fini). Toutefois, les domaines d'application sont eux très étendus.

• Capteurs passifs

Il s'agit généralement d'impédances (résistance, inductance, capacité) dont l'un des paramètres déterminants est sensible à la grandeur mesurée.

La variation d'impédance résulte :

- D'une variation de dimension du capteur (capteurs de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile) .
- D'une déformation résultant d'une force ou d'une grandeur s'y ramenant (pression accélération). Exemples : armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable .

b .Les familles de capteurs

- **Capteurs TOR**

Ce type de capteur permet de détecter un événement ou un objet lié au fonctionnement du système technique. Le signal électrique en sortie de ce capteur est de type logique (signal acceptant deux niveaux : niveau logique 0 ou niveau logique 1) .

- **Capteurs Analogiques**

La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peut être du type :

- sortie tension
- sortie courant
- règle graduée, cadran, jauge (avec une aiguille ou un fluide) ;
- etc.

- **Capteurs Numériques**

La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type :

- train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise
- code numérique binaire

I.2.4.2. Les actionneurs

Dans une machine, un actionneur est une outils qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système. Dans les définitions de l'automatisme, l'actionneur appartient à la partie opérative. [2]

I.2.4.3. La relation capteurs et actionneurs

Dans une installation domotique, on parle d'actionneurs et de capteurs qui interagissent. Les actionneurs sont les composantes qui reçoivent les messages (instructions) et qui, en fonction de ceux-ci, entreprennent ou non une action. Les capteurs font en sorte qu'un ou plusieurs actionneurs reçoivent l'instruction.

Actionneurs et capteurs communiquent via un 'bus': les capteurs placent l'information sur le bus et transmettent ainsi leurs instructions aux acteurs. Le principe est très simple. Un capteur transmet un signal, une alarme ou une instruction au système domotique. La programmation détermine quel acteur doit entrer en action. Le système transmet automatiquement l'instruction à l'acteur qui fait le nécessaire sur la base de celle-ci. [2]

Exemples d'actionneurs et de capteurs:

Actionneurs	Capteurs
Relais	Interrupteurs et boutons poussoirs
Extincteur à eau automatique	Détecteurs de fumée
Chauffage	Thermostats
Eclairage	Capteur d'intensité lumineuse
Alarme	Détecteurs de présence
Compresseur	Capteur seuil de pression pneumatique
Ouverture porte motorisé	Détecteur de proximité
Régulateur de tension	Capteur de surtension

Tableau 1.1:Exemples d'actionneurs et de capteurs

I.2.5.Les fonctions de la domotique [3]

Plusieurs actions peuvent être réalisées grâce aux technologies intégrées dans la domotique et on trouve :

I.2.5.1.Alarme intrusion

Contrairement à un système d'alarme traditionnel, une centrale domotique agit sur toute l'installation électrique de l'habitation. Elle dissuade les intrus en simulant une présence par l'allumage aléatoire des éclairages, de la radio ou l'ouverture des volets durant la journée,...

Les accès à un logement ou à un commerce sont contrôlés et enregistrés et, si une intrusion est détectée, la centrale prend les mesures qui s'imposent :

- sirène
- allumage de tous les éclairages de la maison
- appel d'un centre de surveillance, d'un voisin ou d'un téléphone mobile

À partir d'un téléphone, il est possible d'écouter et de s'adresser directement aux intrus grâce au hautparleur de la centrale.

I.2.5.2.Gestion d'énergie

La programmation des seuils de température est l'une des principales sources d'économie. La programmation peut être journalière ou hebdomadaire. Des sondes de température renseignent la centrale sur les valeurs de température dans chaque pièce. Elle tient compte de la présence d'un occupant et s'adapte automatiquement à son emploi du temps :

- extinction des éclairages inutiles
- réglage de l'intensité lumineuse en fonction de l'activité, ...

Un simple appel téléphonique ou email suffit pour augmenter le chauffage ou allumer la climatisation en prévision de l'arrivée des occupants.

I.2.5.3.Scénarisation

Au moment de quitter un habitat ou un commerce, la mise en fonction de l'alarme déclenche une série de contrôles et d'actions, (centralisation des commandes) :

- fermeture de toutes les lumières
- coupure de l'arrivée de gaz
- vérification de la fermeture de toutes les fenêtres
- allumage de la lumière extérieure durant quelques minutes s'il fait nuit, ...

I.2.5.4.Sécurité

Le système domotique raccordé aux détecteurs permet de détecter les fuites de gaz ou les incendie. Dans ce cas la centrale intervient instantanément pour couper les alimentations, ouvrir les stores, éteindre les flammes , appeler les numéros d'urgence ou faire retentir la sirène si l'occupant est présent.

I.2.5.5.Confort

Le fait d'automatiser la maison permet a un véritable apport sur le confort qu'on y retrouve. il n'est plus nécessaire de déplacer ou de faire un effort physique afin de effectuer une tâche . maintenant on peut exécuter une multitude de tâches en restant sur place et cela grâce a une télécommande ou un Smartphone .Parmi ces actions on trouve par exemple l'ouverture automatique des volets ou l'allumage automatique des lumières . (La figure 1.5 illustre les différentes fonctions de la domotique)

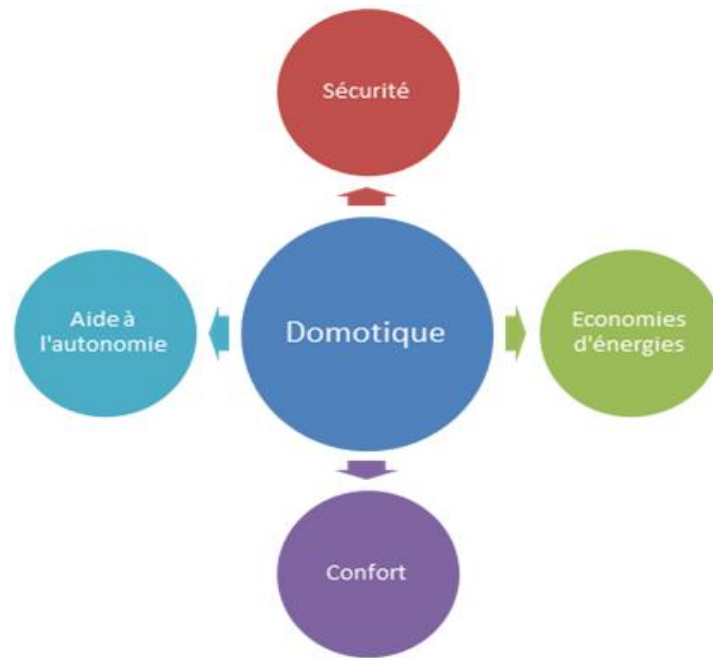


Figure 1.5: Illustration sur les différentes fonctions de la domotique

I.2.6. Les courants porteurs

Le principe de base du réseau CPL (Courant Porteur en Ligne) est d'utiliser les circuits de distribution électrique du domicile pour véhiculer des données et des commandes.

On distingue généralement les 3 grands types de courant porteur suivants:

- **Courant porteur domestique** : cette technologie est conçue pour piloter l'éclairage, le chauffage, les automatismes, les prises de courant et la sécurité .
- **Courant porteur informatique** : ce type de courant porteur en ligne autorise le transport des données informatiques, permettant ainsi de constituer un véritable réseau local reliant ordinateurs, imprimantes, accès Internet, serveur multimédia, écran tactile, point d'accès Wifi, etc.
- **Courant porteur audiovisuel** : cette technologie permet de distribuer l'image et le son dans la maison (standard Home Plug)

Ces trois types de CPL utilisant des fréquences différentes, il est possible de les faire cohabiter.

La technologie CPL est particulièrement intéressante dans le cas des logements anciens, toutefois les désavantages en sont :

- risque de manque de fiabilité en raison de la faible immunité aux parasites sur les lignes
- bus dédié davantage à la commande (manque de retour d'information)
- bus essentiellement propriétaires (peu d'interopérabilité)

I.3. La vidéosurveillance [4]

I.3.1.Introduction :

La vidéosurveillance consiste à placer des caméras de surveillance dans un lieu public ou privé pour visualiser et/ou enregistrer en un endroit centralisé tous les flux de personnes au sein d'un lieu ouvert au public pour surveiller les allées et venues, prévenir les vols, agressions, fraudes et gérer les incidents et mouvements de foule.

Au début des années 2000, les caméras font leur apparition en nombre important dans de nombreuses villes européennes. Londres est réputée comme étant la ville où la vidéosurveillance est la plus importante. L'utilisation de la vidéosurveillance fait débat en matière de sécurité et de la vie privée.

I.3.2.Domaine d'application de la vidéosurveillance

Elle est employée dans de nombreuses situations, généralement pour des raisons de sécurité :

- Dans le cadre de la sécurité routière, au moyen de caméras spécialisées ou des capteurs à proximité voire même noyés dans la chaussée permettent d'évaluer la densité du trafic, les ralentissements qui peuvent en découler, la présence de personnes sur les bandes d'arrêt d'urgence, etc.
- Pour la surveillance des machines : divers capteurs permettent d'évaluer l'état de la machine, ces informations peuvent alors être envoyées à un poste de surveillance. L'épuisement de consommables, une anomalie de fonctionnement ou même un acte de malveillance serait alors détecté à distance ;
- Dans le cadre de la prévention de la délinquance (avec notamment la vidéosurveillance) ;
- Pour la surveillance de lieux sensibles (banques, centrales nucléaires, etc.) et d'habitations, afin de prévenir les intrusions, les cambriolages et les actes de vandalisme ;
- Dans le cadre de la télémédecine, et en particulier pour la surveillance des patients à distance .
- Pour la surveillance à distance des enfants et des personnes vulnérables .

I.3.3.Différents type de systèmes de vidéosurveillance

La technologie des systèmes de vidéosurveillance existe déjà depuis près de 60 ans . Au début ces systèmes là étaient analogique , puis ils ont évolué progressivement vers le numérique . Puis, de nos jours la technologie vidéosurveillance s'est complètement métamorphosé.

1.3.3.1.Vidéosurveillance en circuit fermé

Appelé aussi CCTV (Closed-Circuit television), cette technique utilise des caméras analogique avec des sorties coaxiales reliées a un ou plusieurs écrans . mais l'inconvénient de cette technique c'est que quelqu'un doit être constamment devant l'écran afin de surveiller et d'analyser ce qui se passe. Ceci est le tout premier système vidéosurveillance utilisé.

1.3.3.2.Vidéosurveillance avec magnétoscopes

Cette technique utilise des caméras analogique à sorties coaxiales reliées à un magnéscope VCR (Video Cassette Recording) pour l'enregistrement sur bande magnétique . un quad ou multiplexeur peut être connecté entre la caméra et le magnéscope permet ce qui permet d'enregistrer le contenu de plusieurs caméras sur un même magnéscope. Dans cette technique les séquences vidéo ne sont pas compressées .Mais l'inconvénient de cette technique la c' est le besoin permanent de changer les cassette vu que celle-ci dur que 8 heures. (Ceci est illustrées par la figure ci-dessous)

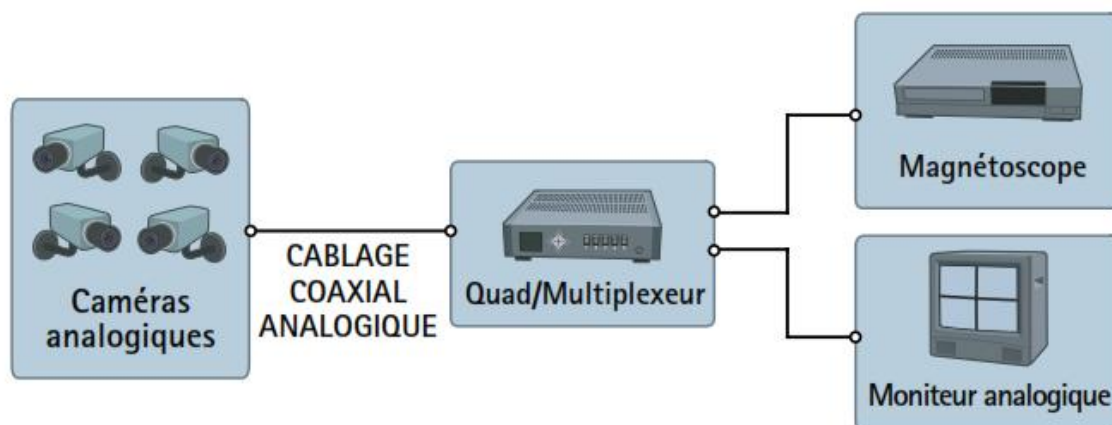


Figure 1.6: Schéma d'un système vidéosurveillance avec magnéscope

1.3.3.3. Vidéosurveillance analogique avec enregistreur numériques

Dans ce cas on utilise toujours des caméras analogiques à sorties coaxiales reliées à un enregistreur numérique DVR (Digital Video Recorder) permettant l'enregistrement numérique des séquences vidéos sur disque dur .Avec cette petite révolution ,on a plus besoin de changer la cassette a chaque fois vue que les disques durs ont une mémoire assez importante . Le passage de l'analogique au numérique permet d'avoir une qualité constante de l'image et de meilleur qualité. (Ceci est illustrées par la figure 1.)

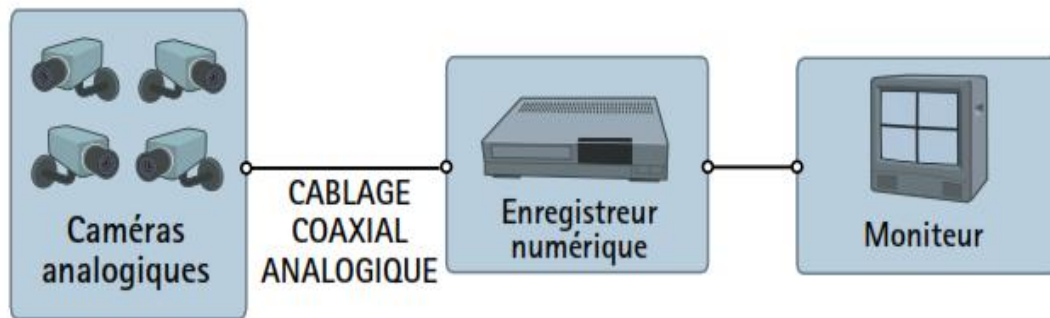


Figure 1.7: Schéma d'un système vidéosurveillance analogique avec enregistreur numérique

I.3.3.4.Vidéosurveillance analogique avec enregistreur numériques réseau

C'est un système de vidéosurveillance analogique utilisant un enregistreur numérique (DVR) réseau est un système en partie numérique comprenant un enregistreur numérique réseau connecté via un port Ethernet. La vidéo étant numérisée et compressée sur l'enregistreur numérique, les images peuvent être transportées sur un réseau informatique à des fins de surveillance sur PC distant. Certains systèmes permettent à la fois la visualisation des séquences en direct et des séquences enregistrées, d'autres se limitent aux images enregistrées. Cette avancée a permis la visualisation des vidéo à distance ainsi que le contrôle du système à distance. (Voir la figure ci-dessous)

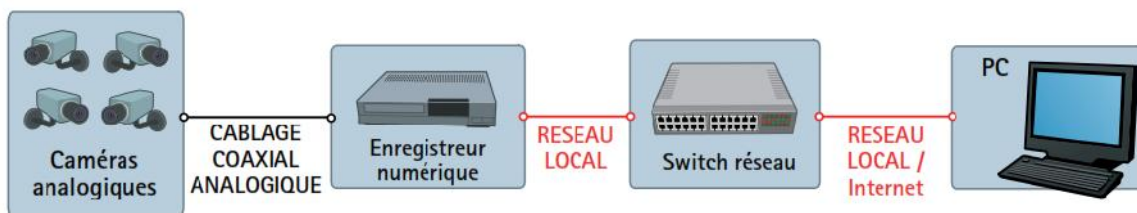


Figure 1.8: Schémas d'un système vidéosurveillance analogique avec enregistreur numérique réseau

I.3.3.5.Vidéosurveillance sur IP avec serveur vidéo

Cette technique associe un serveur vidéo et un commutateur réseau et un PC équipé d'outils de gestion vidéo la caméra est ainsi branché sur le serveur vidéo, celui-ci assure une numérisation et une compression des séquences vidéo. Par contre, le serveur vidéo est

connecté sur le réseau qui transporte la vidéo vers un PC ou serveur via commutateur réseau.

Ce système présente plusieurs avantages :

- recours a un réseau standard et à un serveur informatique standard pour l'enregistrement et le traitement vidéo.
- possibilité d'enregistrement hors ligne.
- Système évolutif pouvant être élargie par ajout de nouvelles cameras réseau. (Ceci est illustrées par la figure ci-dessous)

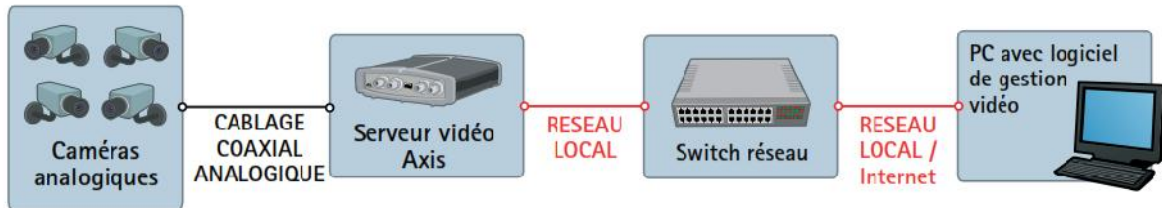


Figure 1.9: Schéma d'un système vidéosurveillance sur IP avec serveur vidéo

➤ Serveur vidéo

La particularité des serveurs vidéo c'est qu'ils s'intègrent facilement avec tout les systèmes vidéosurveillance existant déjà et cité auparavant , et cela en numérisant les signaux anal alogique et en distribuant les images numériques directement sur réseau IP.

Par conséquent , les utilisateurs peuvent visualiser les images en temps réel sur un navigateur web sur un ordinateur du réseau ,ou a n'importe lequel endroit.

I.3.3.6.Vidéosurveillance sur IP avec caméra IP

Une caméra réseau associe une caméra et un ordinateur. Permettant la numérisation et la compression vidéo, elle est en outre équipée d'un connecteur réseau. La vidéo est acheminée par réseau IP via les commutateurs réseau, pour être enregistrée sur un PC/serveur standard à l'aide d'outils de gestion vidéo. Il s'agit d'un système de vidéo sur IP à part entière, doublé d'un système entièrement numérique n'utilisant aucun composant analogique. Les systèmes de vidéo sur IP reposant sur l'utilisation de caméras réseau présentent les avantages suivants :

- Caméras haute résolution (méga pixels) .
- Qualité constante de l'image .
- Fonction d'alimentation par câble Ethernet (Power over Ethernet) et réseau sans fil .

- Fonctions panoramique/inclinaison/zoom, audio, entrées et sorties numériques sur IP .
- Grandes flexibilité et évolutivité . (Ceci est illustrées par la figure ci-dessous)

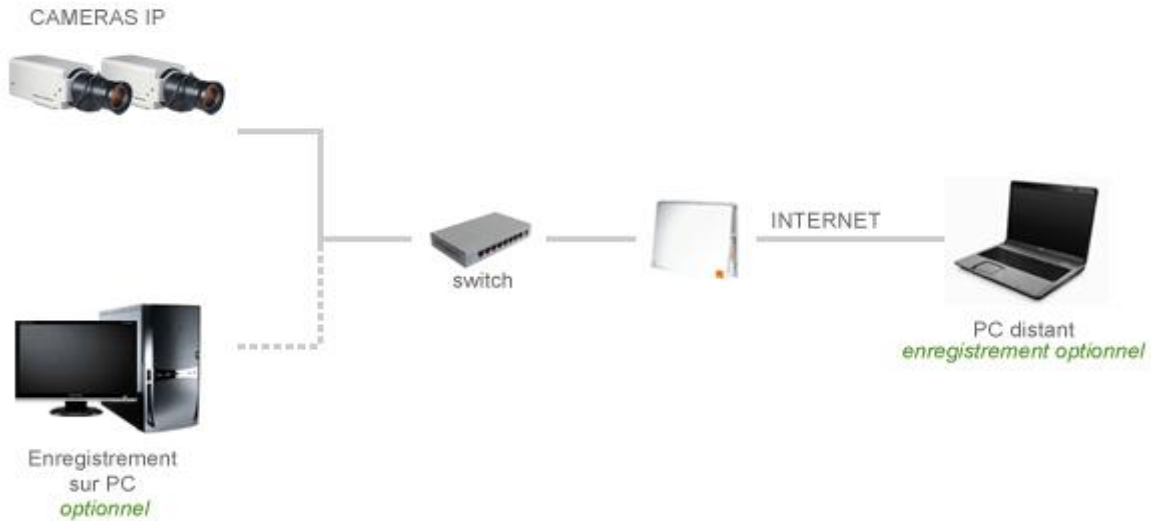


Figure 1.10: Schéma d'un système vidéosurveillance sur IP avec camera IP

I.3.3.7.Vidéosurveillance avec webcam

Cette technique utilise une webcam ou une petite camera connectée a un ordinateur, elle peut être intégrée a un PC ou ajoutée .Cette dernière peut afficher les images capturés d'un espace situé a proximité de l'ordinateur pour surveiller par exemple les enfants jouant dans autre pièce.

On peut aussi faire de la vidéo surveillance à distance , et cela en utilisant un logiciel pour visualiser en temps réel en temps réel et n'importe qu'elle endroit . (Ceci est illustrées par la figure ci-dessous)



Figure 1.11: Illustration de la vidéosurveillance avec une webcam

I.4. Conclusion :

Dans ce chapitre, en premier lieu on a eu un aperçu sur la domotique ainsi que les différents avantages offerts par celle-ci, on remarque qu'elle propose énormément de solution et d'avantage dans notre vie quotidienne sur tout les plans et cela grâce au différentes technologie de pointe employé dans ces systèmes là .

En deuxième lieu, on a pu voir les notions liées au différents systèmes de vidéosurveillances ayant vu le jour ainsi, on conclue que la vidéosurveillance sur IP reste incontestablement la meilleur technique et moyen efficace pour la sécurisation des domiciles et la protection des biens .

Chapitre II

**Acquisition, transfert et
stockage d'images**

II.1. Introduction

La vidéo sur IP est devenue importante et omniprésente dans notre vie ainsi que dans de nombreux secteurs, et cela grâce à ses fonctions avancées dont la surveillance et la protection des personnes et des biens . Dans ce chapitre sera abordé : l'acquisition de l'image ,les techniques de compression d'image et vidéo ,les différents protocoles de transport de données ainsi que la sécurité et le stockage de données.

II.2. Acquisition d'images [5]

Deux techniques permettent de produire des images vidéo : le balayage entrelacé et le balayage progressif. Le choix de l'une ou l'autre de ces techniques dépend de l'application qui sera faite du système vidéo et de son but, mais surtout de la nécessité ou non de saisir le mouvement et de visualiser le détail des objets en mouvement.

II.2.1. Balayage entrelacé

Les images obtenues par balayage entrelacé font appel à des techniques mises au point pour les téléviseurs à tube cathodique comportant 576 lignes horizontales visibles pour un écran standard. L'entrelacement consiste à diviser ce total en lignes paires et impaires et à les rafraîchir ensuite alternativement à une cadence de 30 par seconde. L'écart minime entre le rafraîchissement des lignes paires et impaires crée une légère distorsion.

Les effets de l'entrelacement peuvent être compensés par la technique du désentrelacement . Cette technique là est le processus par lequel des images vidéo entrelacées sont converties sous forme non entrelacée, en éliminant en partie les distorsions vidéo afin d'améliorer la qualité d'affichage.

II.2.1. Balayage progressif

Cette technique scanne toute l'image ligne par ligne, tous les seizièmes de seconde. En d'autres termes, les images capturées ne sont pas divisées en zones séparées comme dans le balayage entrelacé. Les images sont placées dans un ordre croissant sur une seule ligne à la fois, cette technique peut s'avérer décisive lorsqu'il s'agit d'afficher le détail d'une séquence vidéo. Cependant cette technique requiert un écran de très bonne qualité.(La figure suivante illustre la différence entre Différence entre le balayage entrelacé et balayage progressif)

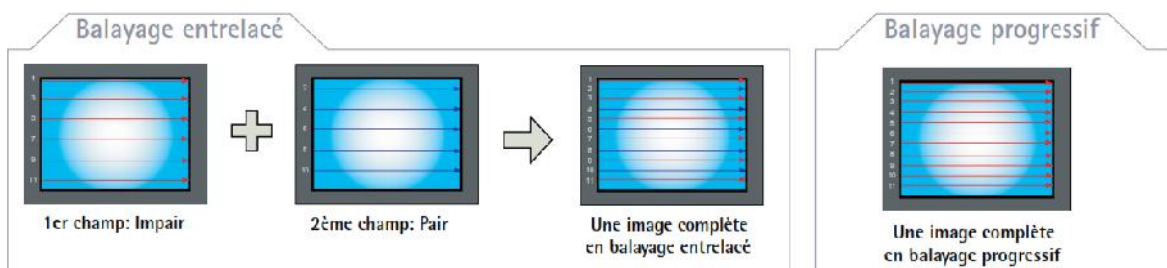


Figure 2.1: Différence entre le balayage entrelacé et balayage progressif

II.3. Compression d'images [5]

La compression des images et des données vidéo peut suivre deux approches différentes : Sans perte ou avec perte. Dans le cas d'une compression sans perte, chaque pixel est maintenu intact. L'image obtenue après compression est donc identique à l'original. Cependant la réduction des données est très limitée. Le format de compression "sans perte" bien connu est le format GIF. Du fait de son faible taux de compression, ce format ne convient guère aux solutions de vidéo sur IP nécessitant l'archivage et la transmission de quantités importantes d'images. C'est pour ça qu'on utilise la compression avec perte dont le principe fondamental est de réduire les éléments invisibles à l'œil humain et d'accroître ainsi considérablement le taux de compression.

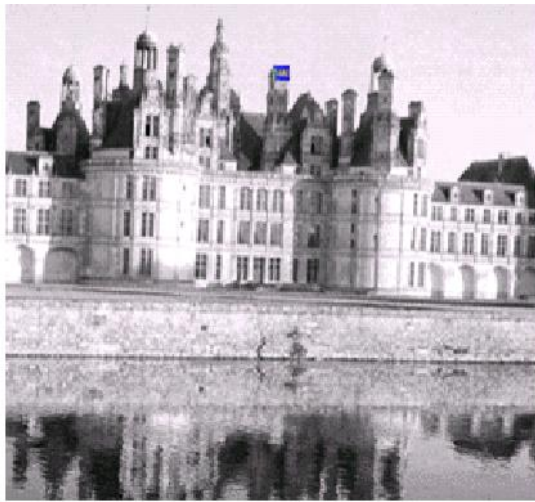
II.3.1. Normes de compression des images fixes

Toutes les normes de compression des images fixes ont la particularité de se concentrer sur une seule image à la fois. La norme la plus connue et la plus répandue en la matière est JPEG (acronyme de Joint Photographic Experts Group) .

II.3.1. 1. Compression JPEG

Le mode de compression JPEG a été normalisé au milieu des années 1980, il est possible de décompresser et de visualiser des images à l'aide d'un navigateur web standard. JPEG permet d'obtenir le degré de compression souhaité : le taux de compression est paramétrable. La compression sélectionnée est directement liée à la qualité de l'image voulue. Outre le degré de compression, l'image elle-même influence également le taux de compression obtenu.

Les deux images ci-dessous illustrent le rapport entre taux de compression et qualité d'image pour une scène donnée, selon deux degrés de compression différents:



*Niveau de compression : “bas”
Ratio de compression : 1:16
6% de la taille de fichier original
Pas de dégradation visible de la
la qualité de l'image*



*Niveau de compression : “élevé”
Ratio de compression : 1:96
1% de la taille de fichier original
dégradation de la qualité de
l'image*

Figure 2.2: Différence entre un niveau de compression élevé et bas

II.3.1. 2. Compression JPEG 2000

JPEG2000 est une autre norme utilisée pour la compression d'images fixes. Cette norme s'adresse principalement aux applications médicales et au monde de la photographie fixe. À des taux de compression peu élevés, la qualité JPEG2000 est similaire à la qualité JPEG. En revanche, quand on passe à des taux beaucoup plus élevés, JPEG2000 s'avère légèrement supérieur .



JPEG



JPEG 2000



JPEG



JPEG 2000

Figure 2.3: Différence entre la compression JPEG et JPEG2000

II.3.2. Normes de compression des vidéos [6]

II.3.2. 1. Vidéo obtenue par une suite d'images JPEG ou Motion JPEG (M-JPEG)

Une camera saisit des images individuelles et les compresse au format JPEG. par exemple, 30 images individuelles par seconde puis les envoyer sur réseau sous forme de flux continu pouvant être lu sur un poste de visualisation. À une fréquence de l'ordre de 16 images par seconde ou plus, l'utilisateur perçoit une vidéo en mouvement. Cette méthode appelée Motion JPEG ou M-JPEG. Chaque image individuelle étant totalement compressée en JPEG, une qualité identique est assurée pour toutes les images, en fonction du taux de compression sélectionné. La figure ci-dessous illustre un exemple de séquence de trois images au format JPEG :



Figure 2.4: Principe d'une séquence trois images avec JPEG

II.3.2.2. Le protocole H.263

La technique de compression H.263 est conçue pour une transmission vidéo à débit fixe. L'inconvénient du débit fixe est que l'image perd de sa qualité lorsque les objets sont en mouvement. La norme H.263 était initialement destinée aux applications de vidéoconférence et non à la surveillance où les détails ont plus d'importance que la régularité du débit.

II.3.2.3. MPEG

La norme MPEG (fondée par le *Motion Picture Experts Group* à la fin des années 1980) est la plus connue des techniques de transmission directe audio et vidéo. Cette section, se limitera à la partie vidéo de la norme MPEG.

Le principe de base du MPEG consiste à comparer entre elles deux images compressées destinées à être transmises sur le réseau. La première des deux images servira de trame de référence. Sur les images suivantes, seuls seront envoyées les zones qui diffèrent de la référence. L'encodeur réseau reconstruit alors toutes les images en fonction de l'image de référence et de la "plage de différence". Bien que plus complexe que la technique Motion JPEG, la compression vidéo MPEG produit de plus petits volumes de données à transmettre via le réseau. (La figure 2.5 illustre le principe du codage MPEG)

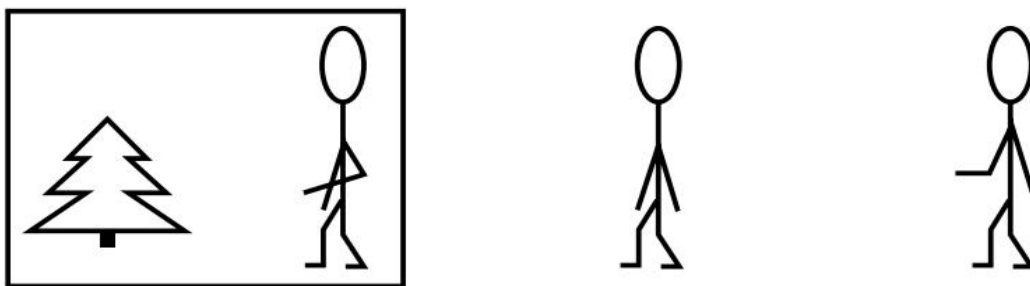


Figure 2.5: Principe du codage MPEG

MPEG est en réalité bien plus complexe que l'ébauche ci-dessus. Cette méthode implique bien souvent des techniques ou des outils supplémentaires permettant de gérer certains paramètres tels que la prédiction du mouvement dans une scène ou l'identification des objets. Il existe aussi différentes normes MPEG :

- **MPEG-1**, lancée en 1993 et destinée à l'archivage des données vidéos numériques sur CD. La plupart des encodeurs et des décodeurs MPEG-1 sont conçus pour un débit d'environ 1,5 Mbit/s en résolution CIF. MPEG-1 met surtout l'accent sur le maintien d'un débit relativement constant, au détriment de la qualité d'image, laquelle est variable et comparable à la qualité vidéo VHS. En MPEG-1, la fréquence d'image est plafonnée à 25 (PAL)/30 (NTSC) images par seconde.
- **MPEG-2**, approuvée en 1994, était destinée à la vidéo numérique de qualité supérieure (DVD), à la télévision haute définition (HDTV), aux supports d'enregistrement interactifs (ISM), aux systèmes d'émission vidéo numérique (DBV) et à la télévision par câble (CATV). Le format MPEG-2 visait à accroître la technique de compression de la norme MPEG-1 afin de couvrir des images plus grandes et de meilleure qualité, mais aux dépens d'un taux de compression plus faible et d'un débit d'images plus rapide. La fréquence est plafonnée à 25 (PAL)/30 (NTSC) images par seconde, tout comme en MPEG-1.
- **MPEG-4** représente une évolution substantielle par rapport au format MPEG-2. Les outils permettant de réduire le débit d'images de manière à atteindre une certaine qualité pour une application ou une scène déterminée sont beaucoup plus nombreux en MPEG-4. En outre, la fréquence n'est plus limitée à 25 ou 30 images par seconde. Il est à souligner cependant que la plupart des outils actuels permettant de réduire le débit ne concernent que les applications en temps réel. Ceci est dû au fait que ces outils requièrent des capacités telles que les durées d'encodage et de décodage (temps de latence) les rendent quasiment impossibles à utiliser à d'autres fins que pour l'encodage de films en studio, de films d'animation, etc. En réalité, la majorité des outils MPEG-4 destinés aux applications en temps réel sont les mêmes que ceux qui existent pour les formats MPEG-1 et MPEG-2.

II.3.2.4. MPEG-4 (Part 10)

Les deux comités à l'origine des normes H.263 et MPEG-4 se sont récemment regroupés pour élaborer une nouvelle génération de normes de compression : la norme AVC (Advanced Vidéo Coding) ou H.264 ou MPEG-4 Part 10, dont le but est d'atteindre des taux de compression très élevés. Cette norme entend offrir une bonne qualité d'image vidéo, mais selon des débits considérablement moins élevés que par les normes précédentes, et sans pour autant accroître la complexité au point de rendre le concept impraticable ou trop onéreux à mettre en œuvre.

II.3.2.5. Avantages et inconvénients respectifs des normes Motion JPEG, MPEG-2 et MPEG-4 [7]

M-JPEG (Motion JPEG) est utilisée dans de nombreux systèmes et représente souvent un choix judicieux vu sa simplicité. Le délai entre la capture de l'image par la caméra, l'encodage, le transfert sur le réseau, le décodage et l'affichage final à l'écran sont limités. En conclusion M-JPEG produit un temps de latence plus faible du fait de sa simplicité (compression des images et images individuelles entières). Il convient donc aux applications de traitement d'image et, notamment, à la détection des mouvements et la localisation des objets. Toutes les résolutions d'images disponibles en pratique, depuis les images adaptées aux téléphones portables (QVGA) jusqu'aux images vidéo en taille réelle (4CIF) ou supérieure (méga pixels), sont possibles en M-JPEG. Quels que soient le mouvement et la complexité de l'image, le système garantit la qualité tout en permettant de choisir entre une qualité supérieure (faible compression) et inférieure (compression élevée).

Quant aux fichiers images, leurs petites tailles ne réclament qu'un faible débit et n'utilise qu'une faible bande passante. La fréquence d'image peut facilement être réglée afin de limiter la bande passante utilisée sans pour autant nuire à la qualité de l'image.

Cependant, le format Motion JPEG génère des volumes de données images assez importants à destination du réseau. Sur ce plan, MPEG présente l'avantage d'envoyer sur le réseau un volume moins important de données par unité de temps (débit) que M-JPEG, sauf à des fréquences peu élevées, comme décrit ci-dessous. Lorsque la bande passante disponible est limitée ou si la vidéo doit être enregistrée à fréquence élevée alors que l'espace de stockage est limité, MPEG s'avère plus indiqué. MPEG offre une qualité d'image relativement élevée pour un débit plus faible. Néanmoins, cette consommation moins importante de la bande passante se fait au prix d'encodages et de décodages plus complexes, qui affectent à leur tour les temps de latence par rapport à M-JPEG.

Le graphe ci-dessous compare les bandes passantes respectives des normes M-JPEG et MPEG-4 pour une même scène en mouvement. Pour de faibles fréquences MPEG-4 consomme juste un petit peu plus que M-JPEG en bande passante. Par contre à des fréquences plus élevées les images MPEG-4 requièrent une bande passante plus faible que les images au format M-JPEG. Ceci est illustré par le graphe suivant :

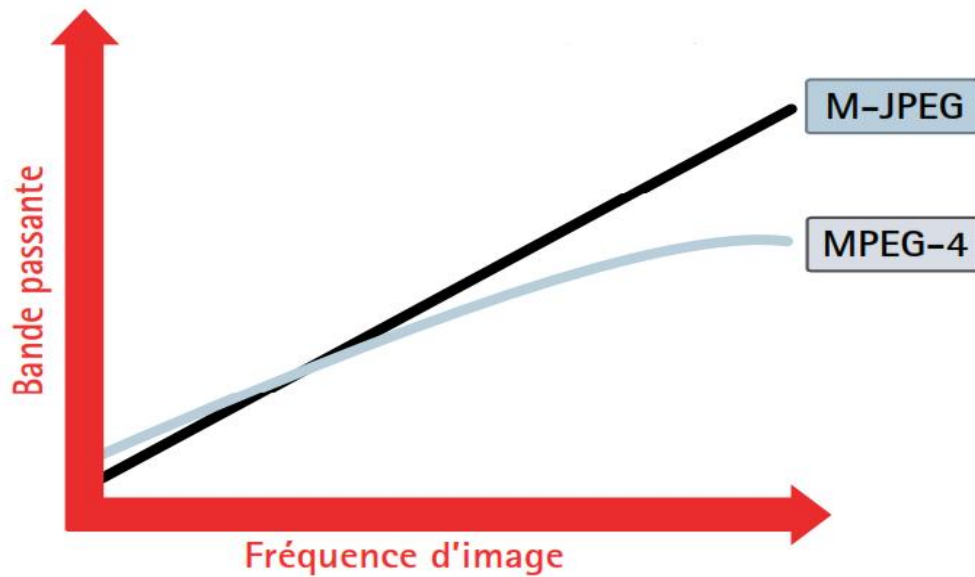


Figure 2.6: Comparaison du rapport fréquence d'image et bande passante pour les normes M-JPEG et MPEG-4

II.4. Résolution

Une image est formée d'un ensemble de points appelés PIXELS (Picture Element). L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image. On appelle définition le nombre de pixels (de points) constituant l'image, c'est-à-dire le nombre de colonnes de l'image que multiplie son nombre de ligne .

II.4.1. Résolutions NTSC et PAL

NTSC offre une résolution de 480 lignes, pour une fréquence de rafraîchissement de 60 champs entrelacés par seconde ,soit 30 images par seconde . Ce modele est utilisé en Amérique du Nord et au Japon. La résolution PAL procure 576 lignes, pour une fréquence de rafraîchissement de 50 champs entrelacés par seconde soit 25 images complètes par seconde, elle est généralement utilisé en Europe.

II.4.2. Résolutions VGA

VGA est l'abréviation de Vidéo Graphics Array, un système d'affichage graphique initialement prévu pour les ordinateurs et mis au point par IBM. La résolution est de 640x480 pixels, soit un format semblable aux formats NTSC et PAL

II.4.3. Résolutions MPEG

La résolution MPEG concerne généralement l'une des résolutions suivantes :

- 704x576 pixels (PAL 4CIF)
- 704x480 pixels (NTSC 4CIF)
- 720x576 pixels (PAL ou D1)
- 720x480 pixels (NTSC ou D1)

II.4.4. Evolution des cameras vers la résolution méga pixel

La résolution maximale en NTSC et en PAL sur caméra analogique, après numérisation du signal vidéo sur enregistreur numérique ou sur serveur vidéo, est de 400000 pixels (704x576 = 405 504). 400 000 correspond à 0,4 méga pixel.

De nos jours, les caméras réseaux permettent dorénavant des résolutions supérieures. Un format méga pixel bien connu est le format 1280x1024, qui offre une résolution de 1,3 méga pixel, soit 3 fois plus qu'une caméra analogique. On trouve également des caméras de 2 et de 3,5 méga pixels. Ces caméras proposent différents rapports largeur-hauteur.

II.5. Les réseaux IP (Internet Protocol) [8]

Le protocole Internet (Internet Protocol, ou IP) est le protocole de communication entre ordinateurs le plus utilisé de nos jours. Il sert de base à la communication par Internet, par messagerie, web et multimédia. L'une des raisons de son succès tient à son adaptabilité. IP convient en effet aussi bien aux infrastructures de petite taille qu'aux entreprises de grande envergure et est accepté par un nombre croissant d'équipements et de technologies puissantes, économiques et éprouvées.

II.5.1. Ethernet

Dans le domaine de la bureautique, les ordinateurs utilisent largement la technologie TCP/IP et sont généralement connectés entre eux via un réseau Ethernet qui permet de bénéficier d'un réseau rapide, moyennant un coût raisonnable. La plupart des ordinateurs actuels intègrent en standard une interface Ethernet ou acceptent facilement une carte d'interface réseau (NIC) Ethernet.

II.5.1.1. Principe de transmission

Tous les ordinateurs d'un réseau Ethernet sont reliés à une même ligne de transmission, et la communication se fait à l'aide d'un protocole appelé *CSMA/CD* (*Carrier Sense Multiple Access with Collision Detect* ce qui signifie qu'il s'agit d'un protocole d'accès multiple avec surveillance de porteuse et détection de collision).

Avec ce protocole toute machine est autorisée à émettre sur la ligne à n'importe quel moment et sans notion de priorité entre les machines. Cette communication se fait de façon simple :

- Chaque machine vérifie qu'il n'y a aucune communication sur la ligne avant d'émettre
- Si deux machines émettent simultanément, alors il y a collision (c'est-à-dire que plusieurs trames de données se trouvent sur la ligne au même moment)
- Les deux machines interrompent leur communication et attendent un délai aléatoire, puis la première ayant passé ce délai peut alors réémettre.

Ce principe est basé sur plusieurs contraintes :

- Les paquets de données doivent avoir une taille maximale
- il doit y avoir un temps d'attente entre deux transmissions

On distingue différentes variantes de technologies Ethernet suivant le type et le diamètre des câbles utilisés :

Sigle	Dénomination	Câble	Connecteur	Débit	Portée
10Base2	Ethernet mince (thin Ethernet)	Câble coaxial (50 Ohms) de faible diamètre	BNC	10 Mb/s	185m
10Base5	Ethernet épais (thick Ethernet)	Câble coaxial de gros diamètre (0.4 inch)	BNC	10Mb/s	500m
10Base-T	Ethernet standard	Paire torsadée (catégorie 3)	RJ-45	10 Mb/s	100m
100Base-TX	Ethernet rapide (Fast Ethernet)	Double paire torsadée (catégorie 5)	RJ-45	100 Mb/s	100m
100Base-FX	Ethernet rapide (Fast Ethernet)	Fibre optique multimode du type (62.5/125)		100 Mb/s	2 km
1000Base-T	Ethernet Gigabit	Double paire torsadée (catégorie 5e)	RJ-45	1000 Mb/s	100m
1000Base-LX	Ethernet Gigabit	Fibre optique monomode / multimode		1000 Mb/s	550m /10000m
1000Base-SX	Ethernet Gigabit	Fibre optique multimode		1000 Mbit/s	550m
10GBase-SR	Ethernet 10Gigabit	Fibre optique multimode		10 Gbit/s	500m
10GBase-LX4	Ethernet 10Gigabit	Fibre optique multimode		10 Gbit/s	500m

Tableau 2.1: Variantes des technologies Ethernet suivant le type et le diamètre des câbles

II.5.2. Réseaux sans fils

Les réseaux sans fil sont basés sur une liaison utilisant des ondes radioélectriques (radio et infrarouges) en lieu et place des câbles habituels. Il existe plusieurs technologies se distinguant d'une part par la fréquence d'émission utilisée ainsi que le débit et la portée des transmissions. Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu.

II.5.2.1.WI-FI

Le Wifi, connu sous la norme *IEEE 802.11* qui permet de créer des réseaux locaux sans fils à haut débit pour peu que l'ordinateur à connecter ne soit pas trop distant par rapport au point d'accès. Dans la pratique, le Wifi permet de relier des ordinateurs portables, des ordinateurs de bureau, des assistants personnels (PDA) ou tout type de périphérique à une liaison haut débit (11 Mbps ou supérieur) sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert.

La norme 802.11 s'attache à définir les couches basses du modèle OSI (Open Systems Interconnection) pour une liaison sans fil utilisant des ondes électromagnétiques, c'est-à-dire :

- la couche physique (notée parfois *couche PHY*), proposant trois types de codages de l'information.
- la couche liaison de données, constitué de deux sous-couches : le contrôle de la liaison logique (Logical Link Control, ou LLC) et le contrôle d'accès au support (Media Access Control, ou MAC)

La couche physique définit la modulation des ondes radioélectriques et les caractéristiques de la signalisation pour la transmission de données, tandis que la couche *liaison de données* définit l'interface entre le bus de la machine et la couche physique, notamment une méthode d'accès proche de celle utilisée dans le standard Ethernet et les règles de communication entre les différentes stations. La norme 802.11 propose en réalité trois couches physiques, définissant des modes de transmission alternatifs :

Couche Liaison de données(MAC)	802.2		
	802.11		
Couche Physique(PHY)	DSSS	FHSS	Infrarouges

Tableau 2.2: Les couches basses du modèle OSI pour la norme 802.11

II.5.2.2.Les différentes normes Wifi

802.11a

Norme opérant sur la fréquence des 5 GHz pour une capacité réelle allant jusqu'à ~24 Mbps et jusqu'à 30 m en situation d'extérieur. Le nombre de produits pris en charge est cependant limité. Le débit théorique est de 54 Mbps.

802.11b

Cette norme offre une capacité réelle allant jusqu'à 5 Mbps et jusqu'à 100 m en situation d'extérieur. Elle opère sur la fréquence des 2,4 GHz. Le débit théorique est de 11 Mbps.

802.11g

Norme la plus utilisée, elle représente une amélioration par rapport à la norme 802.11b. Capacité réelle atteint 24 Mbps et jusqu'à 100 m en situation d'extérieur. Elle opère sur la fréquence des 2,4 GHz avec un débit théorique de 54 Mbps.

802.11n

C'est la nouvelle génération de la norme LAN 802.11 sans fil. Le débit effectif dépasse les 100 Mbps.

II.5.2.3 Broadband Wireless Access

IEEE 802.16, ou WiMAX, est une norme conçue pour l'accès aux réseaux Wifi métropolitains utilisant une architecture de type point-à-multipoint. Elle définit l'utilisation de la bande passante entre les bandes de fréquences des 10 GHz et des 66 GHz, et sous les 11GHz. 802.16 accepte des débits très élevés, tant en amont qu'en aval, à partir de postes pouvant être situés à 50 km et permettant dès lors la prise en charge de services tels que la téléphonie sur IP.

II.5.2.4. Sécurité sur les réseaux sans fil

La communication sans fil implique que toute personne munie d'un appareil sans-fil et présente dans la zone couverte par le réseau est désormais à même d'intervenir sur le réseau et d'utiliser les services partagés. D'où la nécessité de sécuriser le système.

II.6. Transmission de données [8]

II.6.1. Les adresses IP

II.6.1.1. IPv4

Une adresse IP (adresse de protocole Internet) est un numéro unique permettant aux équipements de s'identifier et de communiquer entre eux sur les réseaux utilisant la norme liée au protocole Internet. Une adresse IP se compose de quatre nombres séparés par un point. Elle se subdivise encore en une partie réseau et une partie hôte. La séparation entre ces deux parties est déterminée par la longueur du masque réseau ou du préfixe.

II.6.1.2. IPv6

IPv6, ou protocole Internet version 6, est conçu comme une mise à jour révolutionnaire du protocole Internet. À ce titre, il devrait continuer à coexister pendant un certain temps encore avec l'ancien IPv4. IPv6 a pour but d'assurer la croissance constante d'Internet, tant au niveau du nombre d'hôtes connectés que du volume total d'échanges de données.

La différence entre IPv4 et à IPv6 tient à l'allongement des adresses IP, qui passe de 32 à 128 bits. Cette extension anticipe l'énorme croissance future d'Internet, qui devra permettre un nombre illimité de réseaux et de systèmes. IPv6 devrait ainsi permettre d'attribuer une adresse spécifique à chaque téléphone ou autre appareil mobile.

II.6.2. Modes de transmission de la vidéo sur IP

Il existe trois différents modes de transmission des données sur un réseau informatique :

II.6.2.1. Diffusion individuelle (Unicast)

L'émetteur et le récepteur communiquent entre eux via une liaison point-à-point. Les paquets de données sont adressés à un seul récepteur. Aucun autre ordinateur du réseau n'a besoin de traiter ces informations.

II.6.2.2. Multidiffusion (Multicast)

C'est une communication entre un seul émetteur et plusieurs récepteurs sur un réseau. Les technologies de multidiffusion permettent de réduire le trafic sur le réseau lorsque plusieurs récepteurs souhaitent visualiser une même source en même temps. Un seul flux d'information peut ainsi être envoyé à des centaines de récepteurs.

La différence majeure par rapport à la diffusion individuelle est que le flux vidéo n'est envoyé qu'une seule fois. La multidiffusion est fréquemment utilisée en conjonction avec les transmissions RTP.

II.6.2.3. Radiodiffusion (Broadcast)

C'est type de transmission de un à tous. Sur un réseau LAN, le Broadcast est en principe limité à certains segments spécifiques et ne s'applique pas en pratique aux transmissions vidéo sur IP.

II.6.3. Protocoles de transport de données pour la vidéo sur IP [8]

Le protocole le plus courant pour la transmission des données sur réseaux informatiques est la suite TCP/IP qui sert de "transporteur" pour de nombreux autres protocoles, en particulier pour le HTTP (*Hyper Text Transfer Protocol*) permettant de consulter les pages des serveurs dans le monde entier via Internet.

L'IP utilise deux protocoles de transport : le protocole TCP (Transmission Control Protocol) et le protocole UDP (User Datagram Protocol). Le protocole TCP offre un canal de transmission fiable, qui repose sur la notion de connexions. Il prend en charge le processus de séparation de volumes de données importants en paquets plus petits, adaptés à la configuration physique du réseau, et veille à ce que les données envoyées à un bout parviennent bien à l'autre bout. Le protocole UDP, est protocole dit "sans connexion". Il ne garantit pas la livraison physique des données envoyées et laisse donc à l'application le soin de vérifier et de contrôler les erreurs.

En général, le protocole TCP est utilisé lorsque la fiabilité de la communication a priorité sur la latence du transport. La fiabilité obtenue par retransmission peut cependant causer des délais importants. UDP ne permet pas, en revanche, la retransmission des données perdues, il ne produit dès lors pas non plus de délais supplémentaires.

Protocole	Protocole de transport	Port	Utilisation courante	Utilisation en vidéo sur IP
FTP File Transfer Protocol	TCP	21	Transfert de fichiers sur Internet/intranet	Transfert d'images ou de vidéo de la caméra réseau/serveur vidéo vers un serveur FTP ou une application
SMTP Simple Mail Transfer Protocol	TCP	25	Protocole pour l'envoi de messages e-mail	Une caméra réseau/serveur vidéo peut envoyer des images ou notifications d'alarme à l'aide de son client e-mail intégré
HTTP Hyper Text Transfer Protocol	TCP	80	Utilisé pour le web, par ex. pour accéder à des pages de serveurs web	La manière la plus courante pour transmettre des flux vidéo d'une caméra réseau/serveur vidéo. L'appareil de vidéo sur IP agit comme un serveur web, rendant la vidéo disponible à l'utilisateur ou au serveur applicatif
HTTPS Hypertext Transfer Protocol over Secure Socket Layer	TCP	443	Utilisé pour accéder à des pages web de façon sécurisée à l'aide de l'encryptage	La transmission vidéo sécurisée depuis les caméras réseau/serveurs vidéo peut aussi être utilisée pour authentifier la caméra à l'aide des certificats numériques X.509
RTP Real Time Protocol	UDP/TCP	Non défini	Format de paquets avec encryptage RTP pour fournir de l'audio et de la vidéo sur Internet. Souvent utilisé dans les systèmes de media streaming ou de vidéoconférence.	Une méthode courante pour transmettre des flux de vidéo sur IP MPEG. La transmission peut être individuelle (unicast) ou multiple (multicast)
RTP Real Time Protocol	UDP/TCP	Non défini	Format de paquets avec encryptage RTP pour fournir de l'audio et de la vidéo sur Internet. Souvent utilisé dans les systèmes de media streaming ou de vidéoconférence.	Une méthode courante pour transmettre des flux de vidéo sur IP MPEG. La transmission peut être individuelle (unicast) ou multiple (multicast)
RTSP Real Time Streaming Protocol	TCP	554	Utilisé pour configurer et contrôler les sessions multimédia par RTP	

Tableau 2.3: Protocole TCP/IP et port utilisés dans la vidéo sur IP

II.6.4. Le streaming vidéo

Le streaming est la diffusion d'une vidéo d'un serveur vers un client à travers le réseau internet. En effet, en émission, le serveur segmente la vidéo en paquets susceptibles d'être diffusés sur le réseau. Ces paquets seront ensuite assemblés par le client dans le but de reconstituer la vidéo. A la différence d'un simple transfert de fichier, la vidéo est jouée au fur et à mesure que les paquets arrivent.

il y a actuellement trois protocoles de base permettant de faire du streaming:

II.6.4.1. FTP (File Transfer Protocol)

Le File Transfer Protocol (protocole de transfert de fichiers), ou FTP, est un protocole de communication dédié à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers depuis ou vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou modifier des fichiers sur cet ordinateur. La variante sécurisée de FTP avec les protocoles SSL ou TLS s'appelle FTPS.

FTP obéit à un modèle client-serveur, c'est-à-dire qu'une des deux parties, le client, envoie des requêtes auxquelles réagit l'autre, appelé serveur. En pratique, le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé serveur FTP, qui rend publique une arborescence de fichiers similaire à un système de fichiers Unix. Pour accéder à un serveur FTP, on utilise un logiciel client FTP (possédant une interface graphique ou en ligne de commande). Le protocole, qui appartient à la couche session du modèle OSI et à la couche application du modèle ARPA, utilise une connexion TCP.

Lors d'une connexion FTP, deux canaux de transmission sont ouverts :

- Un canal pour les commandes (canal de contrôle)
- Un canal pour les données

La figure ci-dessous illustre une connexion FTP entre un serveur et un client :

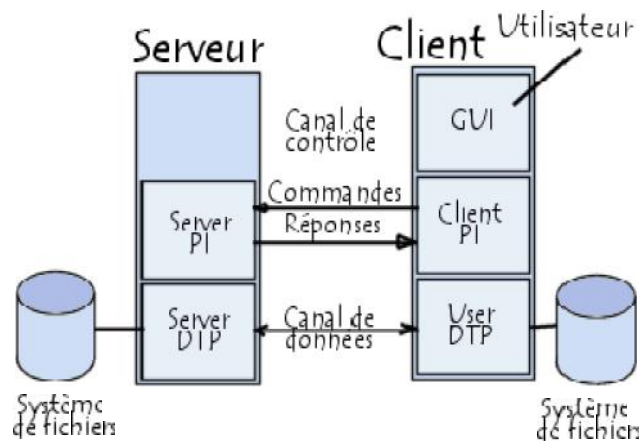


Figure 2.7 : Connexion FTP entre un serveur et un client

Ainsi, le client comme le serveur possèdent deux processus permettant de gérer ces deux types d'information :

- Le DTP (Data Transfer Process) est le processus chargé d'établir la connexion et de gérer le canal de données. Le DTP côté serveur est appelé SERVER-DTP, le DTP côté client est appelé USER-DTP.
- Le PI (Protocol Interpreter) est l'interpréteur de protocole permettant de commander le DTP à l'aide des commandes reçues sur le canal de contrôle. Il est différent sur le client et sur le serveur :
 - ✓ Le SERVER-PI est chargé d'écouter les commandes provenant d'un USER-PI sur le canal de contrôle sur un port donné, d'établir la connexion pour le canal de contrôle, de recevoir sur celui-ci les commandes FTP de l'USER-PI, d'y répondre et de piloter le SERVER-DTP.
 - ✓ Le USER-PI est chargé d'établir la connexion avec le serveur FTP, d'envoyer les commandes FTP, de recevoir les réponses du SERVER-PI et de contrôler le USER-DTP si besoin

II.6.4.2. HTTP (HyperText Transfer Protocol)

Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML). La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME.

Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web (appelé d'ailleurs *http* sur les machines UNIX).

Communication entre navigateur et serveur

La communication entre le navigateur et le serveur se fait en deux temps ,elle est illustrée par la figure ci-dessous :

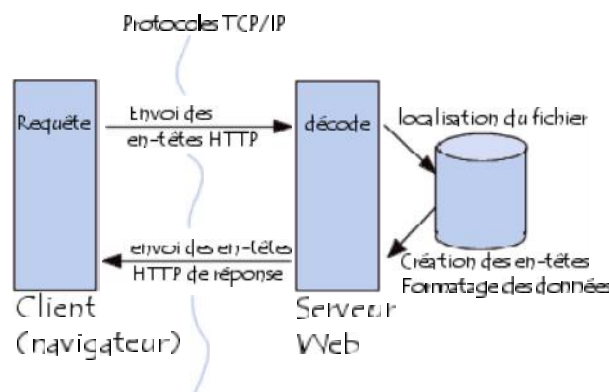


Figure 2.8 :Communication entre un navigateur et un serveur

- Le navigateur effectue une **requête HTTP** .
- Le serveur traite la requête puis envoie une réponse **HTTP** .

En réalité la communication s'effectue en plus de temps si on considère le traitement de la requête par le serveur. Etant donné que l'on s'intéresse uniquement au protocole HTTP, le traitement du côté serveur ne sera pas explicité dans le cadre de cet article... Si ce sujet vous intéresse, référez-vous à l'article sur le traitement des CGI.

Remarque :

HTTPS (avec S pour *secured*, soit « sécurisé ») est la variante du HTTP *sécurisée* par l'usage des protocoles SSL ou TLS.

II.6.4.3. Le protocole RTP [9]

Le protocole RTP (Real Time protocole) permet d'envoyer les paquets en temps réel sur le réseau . Les paquets sont marqués temporellement de manière à être réordonné par le client afin d'afficher la vidéo de manière cohérente .

Il permet des services de types unicast ou multicast sans garantie de qualité de service (QoS). Par opposition à http et FTP qui fonctionnent au dessus de TCP(mode connecté), RTP fonctionne au dessus d'UDP (mode non connecté).

II.6.4.4. RTCP (Real-time Transfer Control Protocol)

Le protocole RTCP est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session. C'est un protocole de contrôle des flux RTP, permettant de véhiculer des informations basiques sur les participants d'une session, et sur la qualité de service qui sert par exemple de contrôler les propriétés temps réel du contenu délivré (la diffusion de données en direct) ou le contrôle de données préenregistrées.

II.7. Bande passante

La vidéo sur IP utilise la bande passante en fonction de sa configuration. L'usage de la bande par une caméra dépend de plusieurs facteurs :

- La taille de l'image
- Le taux de compression
- La fréquence d'images (nombre d'images par seconde)
- La complexité de la scène

Il existe de nombreuses manières de tirer pleinement parti d'un système vidéo sur IP et de gérer au mieux la consommation de la bande passante. Voici les techniques employées :

II.7.1. Réseaux commutés

Grâce au “switching”, méthode très répandue aujourd’hui, le même réseau physique (ordinateurs et système de vidéo sur IP) peut être séparé en deux réseaux autonomes. Même en restant physiquement connecté, cette méthode le divise de manière logique en deux réseaux virtuels et indépendants.

II.7.2. Bande passante élargie

Le prix des commutateurs Gigabits et des routeurs baissant continuellement, les réseaux de grande capacité sont de plus en plus abordables. En augmentant la bande passante, la tendance est aux réseaux plus rapides et les systèmes de surveillance à distance sont donc de plus en plus attrayants et rentables.

II.7.3. Fréquence évolutive

La configuration de 25 images par seconde enregistrées en permanence pour toutes les caméras offre un niveau de qualité bien supérieur à ce dont la plupart des applications ont besoin. Grâce aux capacités de configuration et à l’intelligence intégrée de la caméra réseau ou du serveur vidéo, la fréquence peut être diminuée en conditions normales (1 à 5 images/sec), réduisant ainsi la consommation de bande passante. En cas d’alarme ,déclenchée par exemple par le détecteur de mouvements, la fréquence d’enregistrement peut adopter automatiquement un niveau supérieur.

II.8. Stockage [10]

L’émergence des systèmes de vidéo sur IP implique une utilisation de plus en plus importante d’espace disque. Ceci pose un certain nombre de questions, notamment celle de savoir quel espace disque sera nécessaire et comment assurer un stockage sûr.

Facteurs à prendre en compte dans la détermination des besoins de stockage :

- Nombre de caméras.
- Nombre d’heures d’enregistrement quotidien par la caméra .
- Durée de conservation souhaitée des données.
- Détection des mouvements (événements) uniquement, ou enregistrement continu.
- Autres paramètres, comme par exemple la fréquence, la compression, la qualité d’image et la complexité demandées.

II.8.1. Espace disque nécessaire

le stockage de donnée sur le disque dépend de la norme de compression vidéo utilisé :

II.8.1.1. JPEG/Motion JPEG

Dans le cas d'une solution JPEG/Motion JPEG impliquant la réception de fichiers individuels, les besoins de stockage peuvent varier en fonction de la fréquence, de la résolution et de la compression mises en œuvre.

Calcul :

Taille de l'image x nombre d'images par seconde x 3600s = Ko par heure / 1000 = Mo par heure .

Mo par heure x nombre d'heures de fonctionnement par jour / 1000 = Go par jour

Go par jour x durée de conservation = besoin de stockage

Caméra	Résolution	Taille de l'image (Ko)	Images par seconde	Mo/heure	Heures de fonctionnement	Go/jour
No. 1	CIF	13	5	234	8	1,9
No. 2	CIF	13	15	702	8	5,6
No. 3	4CIF	40	15	2160	12	26

Total pour les 3 caméras sur 30 jours de stockage = 1002 Go

II.8.1.2. MPEG-4

En MPEG-4, les images parviennent suivant un flux continu et non pas sous forme de fichiers individuels. Les besoins de stockage dépendent du débit, c'est-à-dire de la quantité de données vidéo transmises. Le débit est quant à lui le résultat d'une fréquence déterminée, selon une résolution et un taux de compression donnés et en fonction du degré de mouvement de la scène.

Calcul :

Taille de l'image x nombre d'images par seconde x 3600s = Ko par heure / 1000 = Mo par heure

Mo par heure x nombre d'heures de fonctionnement par jour / 1000 = Go par jour

Go par jour x durée de conservation = besoin de stockage

Caméra	Résolution	Taux de bits (kBit/s)	Images par seconde	Mo/heure	Heures de fonctionnement	Go/jour
No. 1	CIF	13	5	234	8	1,9
No. 2	CIF	13	15	702	8	5,6
No. 3	4CIF	40	15	2160	12	26

Total pour les 3 caméras sur 30 jours de stockage = 204 Go

II.9.Redondance [10]

La redondance consiste à installer un équipement en double en vue de pallier l'éventuelle déficience de l'un des deux espaces de stockage.

II.9.1. RAID (Redundant Array of Independent Disks) :

Cette méthode consiste essentiellement à répartir les données sur plusieurs disques durs, de telle manière qu'en cas d'échec sur l'un des disques, les données puissent être récupérées sur un autre disque. (Voir figure ci-dessous)

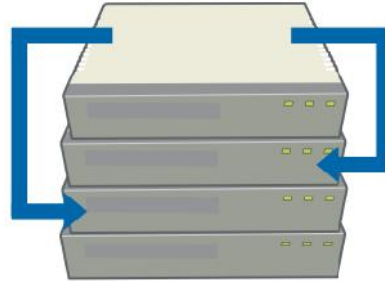


Figure 2.9 : Répartition des données sur plusieurs disques selon la méthode RAID

II.9.2. Réplication des données

C'est une technique commune à de nombreux systèmes réseau : les serveurs de fichiers sur le réseau sont configurés de manière à ce que les données soient répliquées entre les différents serveurs. .(Voir figure ci-dessous)

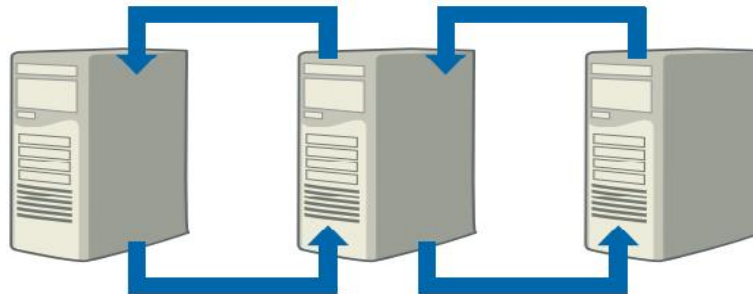


Figure 2.10 : Figure explicative sur la technique de réplication de données

II.9.3. Sauvegarde sur bandes magnétiques

Il s'agit d'une méthode alternative ou complémentaire. Il existe toute une panoplie de logiciels et d'équipements prévus à cet effet. En général, pour faire face aux vols éventuels ou aux incendies, les politiques de sauvegarde préconisent en outre de stocker les bandes à l'extérieur.

II.9.4. Mise en cluster de serveurs

Il existe de nombreuses méthodes de mise en cluster des serveurs. L'une des plus fréquentes pour les serveurs de bases de données et les serveurs de messagerie, concerne deux

serveurs utilisant un même dispositif de stockage. Le système appelé système RAID peut palier en cas de problèmes sur l'un des deux serveurs, l'autre (configuré de façon identique) reprend en charge l'application. Souvent, les deux serveurs partagent la même adresse IP, rendant ainsi la procédure de basculement automatique totalement transparente pour l'utilisateur. (Voir figure ci-dessous)

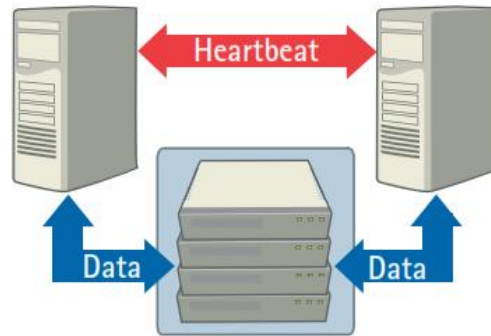


Figure 2.11:Schéma explicative sur mise en cluster des serveurs

II.9.5. Dédoublément des réceptionnaires vidéos

C'est Une méthode courante qui vise à garantir la récupération des données en cas de problèmes . L'archivage externe de la vidéo réseau consiste à envoyer la vidéo simultanément vers deux serveurs différents, situés en des lieux différents. Ces serveurs peuvent bien sûr disposer des techniques RAID, fonctionner en clusters ou répliquer leurs données sur d'autres serveurs éventuellement encore plus éloignés.

II.10.Sécurité de données

Tout système de vidéosurveillance pose certaines questions majeures quant au respect de la vie privée. L'intelligence vidéo et les caméras réseaux peuvent apaiser certaines inquiétudes à ce sujet. Contrairement aux caméras en circuit fermé qui ne transmettent qu'un seul flux vidéo pouvant être intercepté, une caméra réseau est capable de crypter la vidéo qu'elle transmet sur le réseau, de manière à ce qu'elle ne puisse être ni visualisée, ni manipulée d'aucune façon. Le système peut également être configuré pour authentifier la connexion avec des certificats de cryptage qui acceptent uniquement une caméra réseau spécifique, ce qui permet d'éliminer la possibilité de piratage de la ligne.

II.11.Visualisation via l'interface web

La vidéo peut être visualisée en tout point du réseau à condition d'avoir accès à un navigateur web. Chaque caméra intègre un serveur web disposant d'une adresse IP. Pour visualiser les images sur PC, il suffit donc d'ouvrir le navigateur web et de saisir l'adresse IP de la caméra dans la zone d'adresse. Une fois que l'ordinateur a établi la connexion, la page d'accueil de la caméra réseau s'affiche automatiquement dans le navigateur web.

II.12.Conclusion

Dans ce chapitre on a donnée quelque notions sur les différents bloc qui constituent un système de télésurveillance ainsi que certaines contraintes engendrées lors d'émission de vidéo sur internet.

Cependant la télésurveillance sur IP reste la technique la plus utilisée d'une part , parce qu'elle est accessible à tout le monde avec un cout abordable. D'autre part , elle propose les meilleurs solutions, car elle dispose de technologies avancées et d'un niveau de sécurité assez élevée .

Chapitre III

**Conception, étude et
réalisation expérimentale du
système**

III.1. Introduction

Dans ce chapitre sera présenté le système conçu, passant par l'explication de son fonctionnement de manière détaillée, le schéma synoptique du système, et la représentation des deux éléments de base le constituant dont la carte Arduino Méga et la carte Raspberry Pi 2 ainsi que les différents composants et modules embarquée sur ces deux cartes. Pour finir le module Bluetooth qui assure la communication sans fils entre ces deux cartes.

III.2. Conception générale du système

Le système conçu et réalisé est constitué de deux cartes principales et essentielles, la carte Arduino qui va s'occuper d'une part du système de sécurité, et d'autre part de la partie domotique.

La carte Raspberry PI avec caméra embarquée, dont la fonction principale est d'effectuer une acquisition vidéo puis un traitement d'images et à la fin notre vidéo est transmise via internet accessible sur une page web sécurisé depuis n'importe qu'elle endroit.

III.3. Description détaillée du système

Le système est composé essentiellement de :

- **Carte de contrôle et sécurité du domicile** : Elle est structurée autour d'une carte Arduino Méga qui s'occupe de la domotique. Cette dernière s'occupe de la sécurité du domicile avec un accès via digicode ainsi que de l'allumage automatique des lumières avec commande vocale via une application Android sur notre Smartphone. Le module Bluetooth Il se charge de la liaison entre l'application de notre Smartphone et la carte Arduino, afin de contrôler l'ensemble du domicile via un téléphone portable. (Voir figure schéma synoptique bloc 1)
- **Carte de commande d'acquisition vidéo** : La carte Raspberry pi 2, se charge de l'acquisition vidéo ainsi qu'une compression d'images sont effectués. Immédiatement les images sont transmises sur notre page web, afin qu'on puisse y accéder en temps réel. (Voir figure schéma synoptique bloc 2)

Le schéma synoptique du système est illustré par la figure suivante :

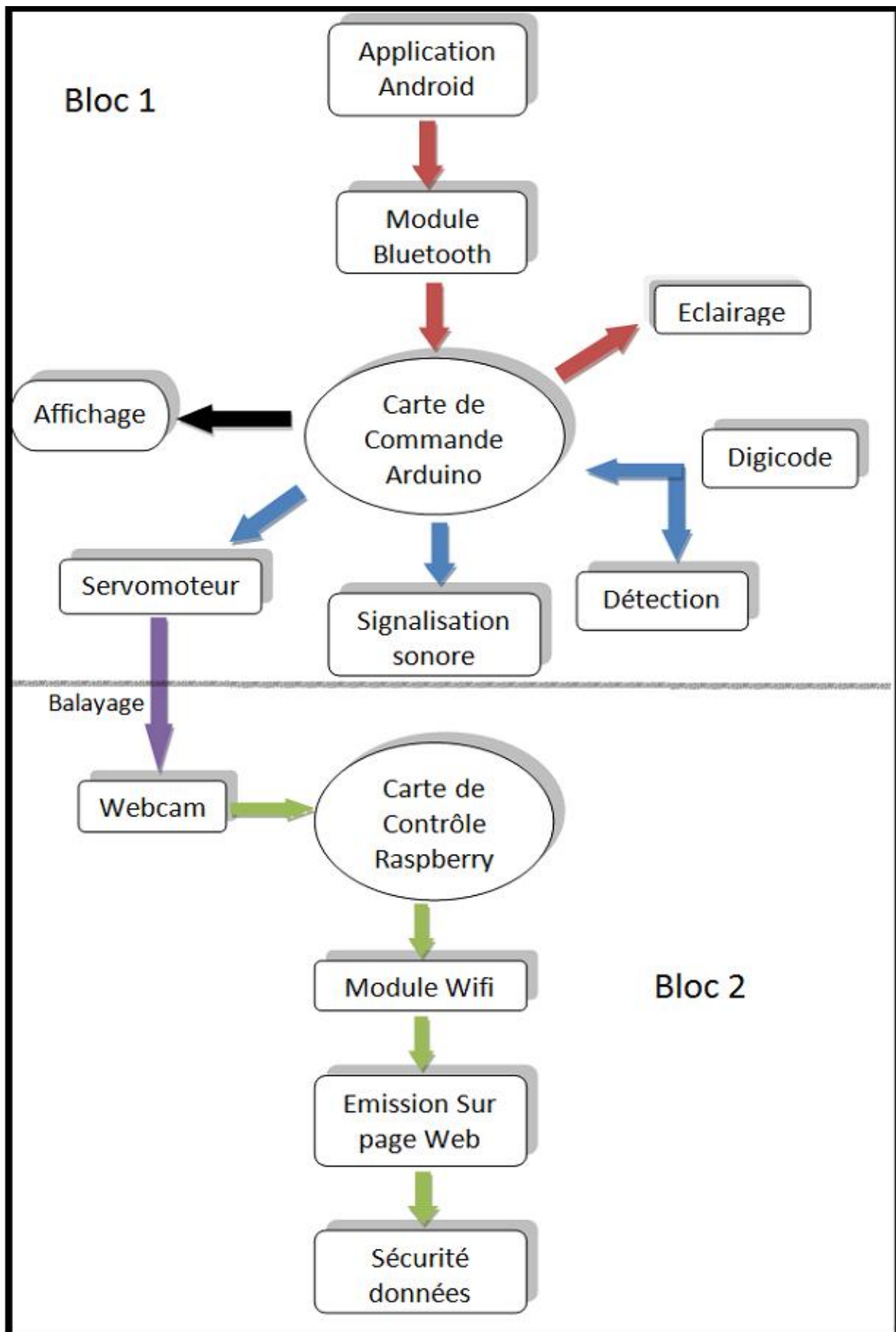


Figure 3.1: Schéma synoptique représentatif du système

III.4. Carte de contrôle et sécurité du domicile

III.4.1. Carte de commande Arduino

III.4.1.1. Description générale de la carte Arduino Méga [11]

La carte Arduino méga de la société Arduino est une carte électronique dotée d'un microcontrôleur ATMEL de référence ATmega2560. Elle dispose d'une interface hôte USB pour se connecter aux téléphones fonctionnant sous Android.

Le microcontrôleur ATmega2560 est un microcontrôleur 8bits de famille AVR dont la programmation peut être réalisée en langage C. Cette carte dispose de 54 entrées/sorties numérique dont 15 peuvent être utilisées en PWM (Pulse Width Modulation), 16 entrées analogique/numérique de 10 bits de résolution, 4 ports de communication série, 1 résonateur céramique (Quartz) de 16 Mhz, un connecteur ICSP(In-Circuit Serial Programmions) qui permet la programmation du microcontrôleur sur le circuit sans avoir à l'enlever, 1 connecteur jack pour une alimentation extérieur, un bouton reset pour mettre le processeur a zéro.

L'avantage de cette carte est qu'elle n'a pas besoin de pilote pour faire la conversion FTDI USB/Série, elle a juste un petit microcontrôleur ATmega 16U2 programmé comme convertisseur USB/série.

Arduino permet un environnement de développement utilisant des outils open source. Les programmes sont chargés dans le microcontrôleur grâce au port USB. En plus de cela, il y a plusieurs bibliothèques de fonctions utilisables. Ces fonctions sont proposées pour l'exploitation d'entrées-sorties : exploitation des bus I2C, génération de signaux PWM, utilisation des port de communication séries....etc. (Voir figure ci-dessous)



Figure 3.2: Illustration de la carte Arduino Méga ADK

III.4.1.2. Le microcontrôleur ATmega2560 [11]

Le Atmel ATMEGA2560-16AU est un microcontrôleur 8 bits CMOS basse puissance basée sur architecture RISC améliorée des AVR. En exécutant des instructions puissantes en un seul cycle d'horloge, le ATMEGA2560-16AU atteint des débits approchant 1Mbit/S par MHz permettant aux concepteurs de système d'optimiser la consommation d'énergie par rapport à la vitesse de traitement. (La figure ci-dessous illustre un CPU ATmega2560 ainsi que son schéma électrique)

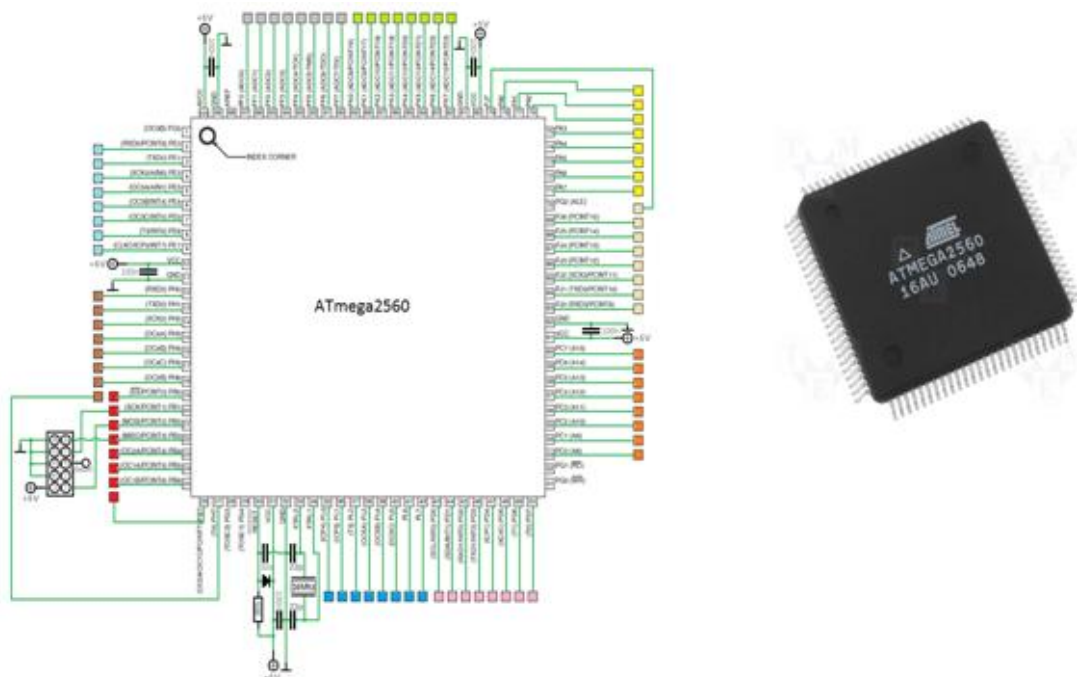


Figure 3.3 : Illustration du CPU ATmega2560 ainsi que son schéma électrique

Les principales caractéristiques sont :

- **FLASH** = mémoire programme de 256 KB dont **8 KB** sont utilisés par le bootloader.
- **SRAM** = données (volatiles) 8Ko
- **EEPROM** = données (non volatiles) 4Ko.
- **Tension d'alimentation interne** = 5V.
- **tension d'alimentation (recommandée)** = 7 à 12V, limites = 6 à 20 V.
- **Courant max par broches E/S** = 40 mA.
- **Courant max sur sortie 3,3V** = 50mA.
- **Fréquence horloge** = 16 MHz.

- **Dimensions** = 101.52 mm en longueur et de 53.3mm en largeur .
- **Timers/Counters : Timer0 et Timer2** (comptage 8 bits), **Timer1** (comptage 16bits).
Chaque timer peut être utilisé pour générer deux signaux **PWM**.
- **Plusieurs broches multifonctions** : certaines broches peuvent avoir plusieurs fonctions différentes choisies par l'utilisateur.
- **PWM** = 15 broches.
- **Analog to Digital Converter (résolution 10bits)** = 16 entrées multiplexées ADC0(PC0) à ADC15(PC15)
- **Gestion bus I2C (TWI Two Wire Interface)**
- **Port série (USART)** = émission/réception série via les broches TXD/RXD
- **Comparateur Analogique** = certaines broches peuvent déclencher des interruptions.
- **Watchdog Timer programmable** : l'ATMéga possède un compteur dit de chien de garde programmable pour générer des interruptions à la fin de son comptage, et il peut être utilisé comme étant un simple compteur.
- **Gestion d'interruptions:**
 - ✓ Interruptions liées aux entrées numériques.
 - ✓ 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 I2C.

Les détails de la carte Arduino Méga sont résumés en Annexe A.

Voir la figure suivante qui détermine des différentes composantes de la carte Arduino Méga

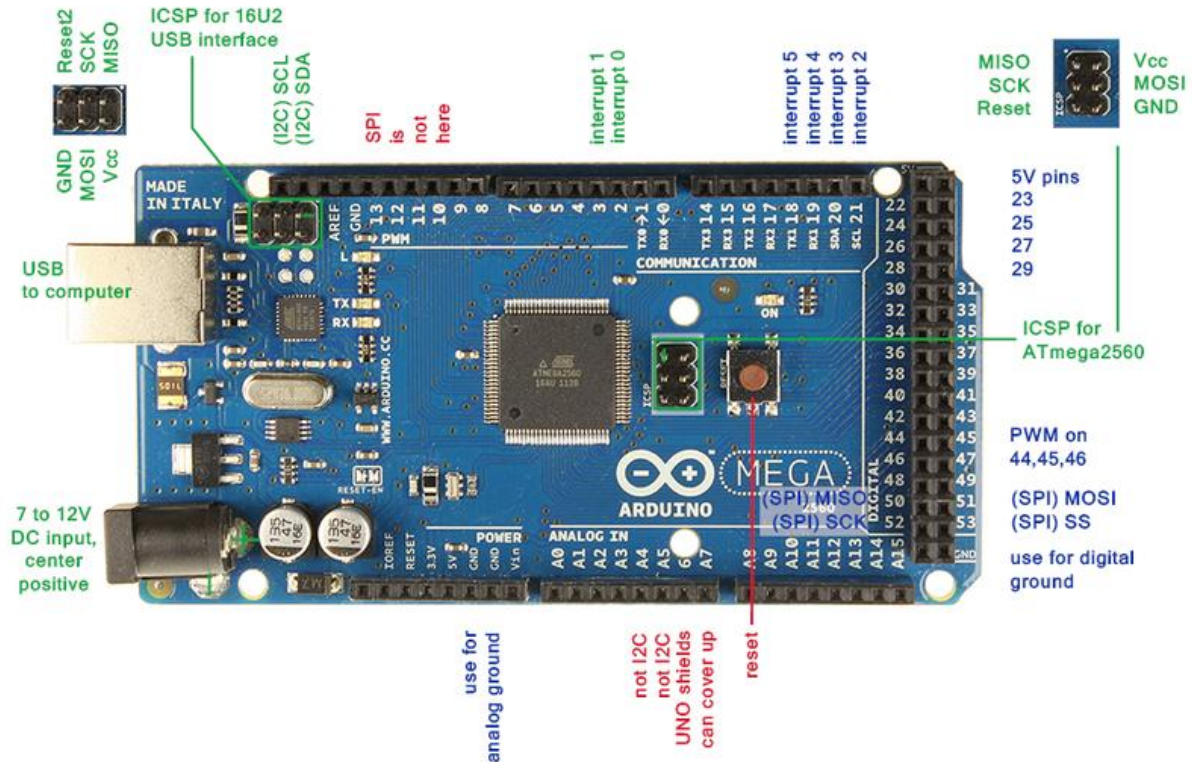


Figure 3.4: Illustration des différentes composantes de la carte Arduino méga

III.4.1.3. Communication avec l'extérieur de la carte Arduino [11]

La carte Arduino Mega2560 dispose de moyens efficaces et faciles pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UART (Universal Asynchronous Receiver Transmitter) ou émetteur-récepteur asynchrone universel pour la communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LED de RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte.

L'ATmega2560 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

- l'ide Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.
- Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

III.4.2. Composants utilisés avec la carte Arduino

III.4.2.1. Clavier matriciel

Le clavier numérique est plus aisé et plus pratique à utiliser et présente la communication Homme-Machine.

Le code étant numérique, il suffit de choisir un clavier 16 touches, est équipé de touche numérique 0 à 9, * et # et des lettres alphabétiques A,B,C,D.

Le clavier matriciel est implémenté sur 8 broches de la carte Arduino. Cette application pourra servir pour le codage des serrures ainsi que l'activation et la désactivation du système d'alarme dont le but de protéger le domicile.

Dans notre système la touche '*' sert à effacer et la touche '#' sert à confirmer le code inséré, les autres touches restantes font partie du code 4 caractères à introduire. (La figure suivante illustre parfaitement un clavier matriciel ainsi que son schéma électrique)

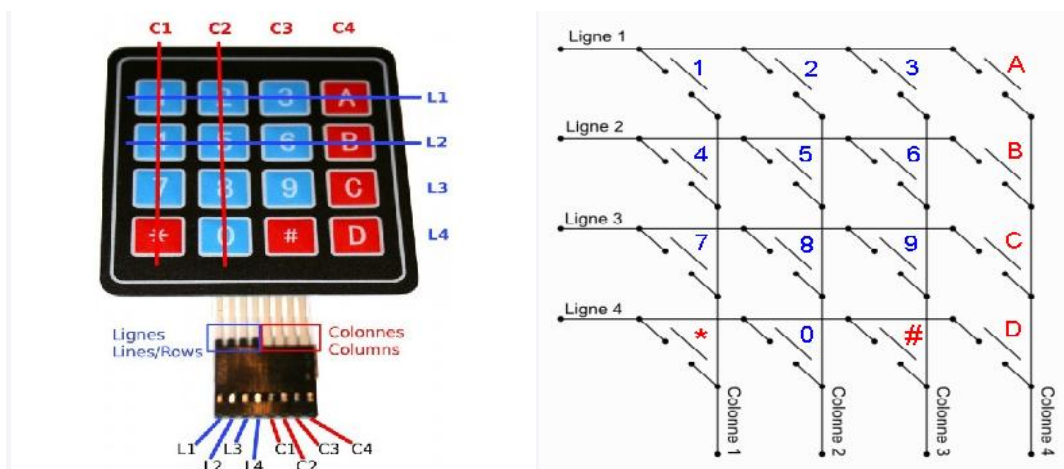


Figure 3.5 : Clavier matriciel et son schéma électrique

III.4.2.2. Afficheurs alphanumériques à cristaux liquides [12]

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

L'afficheur est constitué de deux lames de verre, distantes de 20 μm environ, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli

de cristal liquide normalement réfléchissant (pour les modèles réflectifs). L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réflectif ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Les modèles transmissifs fonctionnent différemment. Normalement opaque au repos, le cristal liquide devient transparent lorsqu'il est excité. Pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétro éclairés.

L'afficheur utilisé dispose de :

- 4 lignes de 20 caractères.
- Une RAM (DDRAM : DATA RAM) de 80 caractères correspondants.
- Une RAM (CGRAM : CHARACTER GRAPHIC RAM) permettant de créer de nouveaux caractères.
- Le registre d'instruction *I R*: (Instruction Register) :C'est le registre de contrôle, suivant la valeur introduit, l'afficheur exécute des opérations de configurations. Il permet aussi de positionner le curseur parmi les 32 adresses de l'afficheur.
- Le registre de données *D R*: (*Data Register*) : Suivant la valeur mise dans l'afficheur , ce dernier peut monter un caractère (Code *ASCII* ou spécifiques) ou créer une ligne d'une matrice d'un nouveau caractère.

Cet afficheur nécessite une alimentation de 5V pour pouvoir alimenter son pilote interne et ainsi permettre l'affichage des caractères sur l'écran. La résistance variable permet de modifier le contraste à tout moment. Ce LCD possède huit bits de données, mais seules quatre sont utilisées. Le composant propose en effet un mode de transmission sur 4 bits en passant ces caractères en deux temps et économise ainsi quatre lignes de données sur le microcontrôleur.

En effet, l'afficheur LCD permet de manière très rapide de révéler n'importe quelle information qui pourrait être utile, et permet créer une petite interface utilisateur efficace. (Voir figure ci-dessous)

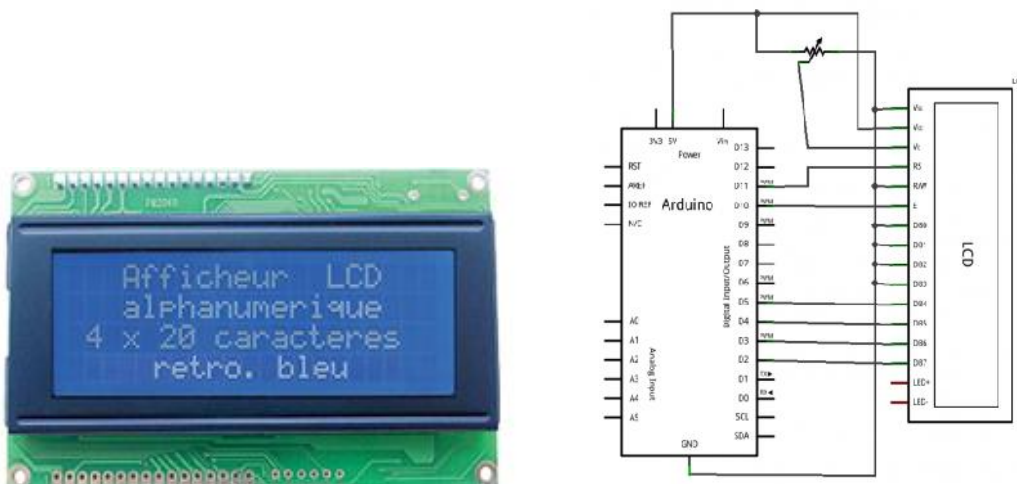


Figure 3.6 : L'afficheur LCD et le schéma de connexion avec l'Arduino

III.4.2.3. Capteur de distance à Ultrasons HC-SR04

C'est un capteur muni d'une paire d' « yeux » : un émetteur qui émet un son à une fréquence de 40 kHz (domaine des ultrasons) et un récepteur qui collecte le son répercuté par l'obstacle. (La figure suivante illustre le fonctionnement du module ultrason)

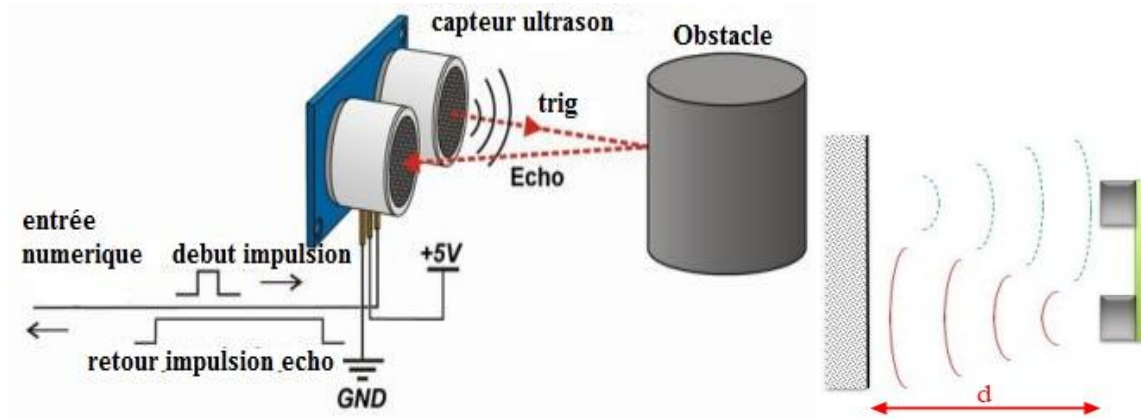
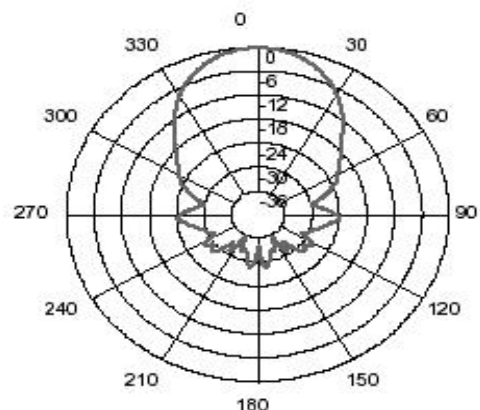


Figure 3.7 : Fonctionnement du module ultrason

La distance à l'obstacle peut alors être calculée par le temps mis par le son pour faire l'aller-retour. La vitesse du son est à peu près stable. Pendant l'intervalle de temps t , le son parcourt deux fois la distance d .
vitesse du son = 340 m/s dans l'air.

$$d = \frac{\text{vitesse du son} \times t}{2}$$

Specifications	
Frequency	40kHz
Max Range	4 meters
Min Range	3 centimeters
Input Trigger	10uSec minimum, TTL level pulse
Echo Pulse	Positive TTL level signal, proportional to range



Voici les caractéristiques du module et son fonctionnement détaillé dans ce tableau:

Tableau 3.1 : caractéristiques du module et son fonctionnement

Pour l'utilisation du capteur ultra-son il faut lui envoyer une demande de mesure, que l'on appelle *Trigger*. C'est une impulsion de 5 volts dont la durée est de 10 μ s (microsecondes) minimum. Par précaution, il est conseillé d'envoyer un signal *Trigger* dont la durée est entre 12 et 15 μ s.

Dès qu'il reçoit ce signal, le trigger envoie un train d'ultrasons de 40 kHz (huit signaux de courtes périodes). Dès l'envoi du dernier ultrason, le SRF-05 envoie un signal sur la broche *Echo* qu'il maintient à 5 volts jusqu'à ce qu'il reçoive un retour des ultrasons. À ce moment-là, il redescend le signal *Echo* à la masse (0 volt).

Nous avons donc un signal *Echo* dont la durée est proportionnelle à la distance entre le capteur et l'objet. Ce dernier *Echo* redescend à la masse au bout de 30 ms si aucun obstacle n'est détecté (*time-out*).

Afin d'éviter des résultats faussés par des mesures lancées quasi simultanément, il est recommandé d'attendre 50 ms entre deux demandes de mesure, ce qui permet de déclencher jusqu'à 20 mesures par seconde d'utilisation au maximum.

Le principe du fonctionnement est résumé sur le chronogramme constructeur ci-dessous :

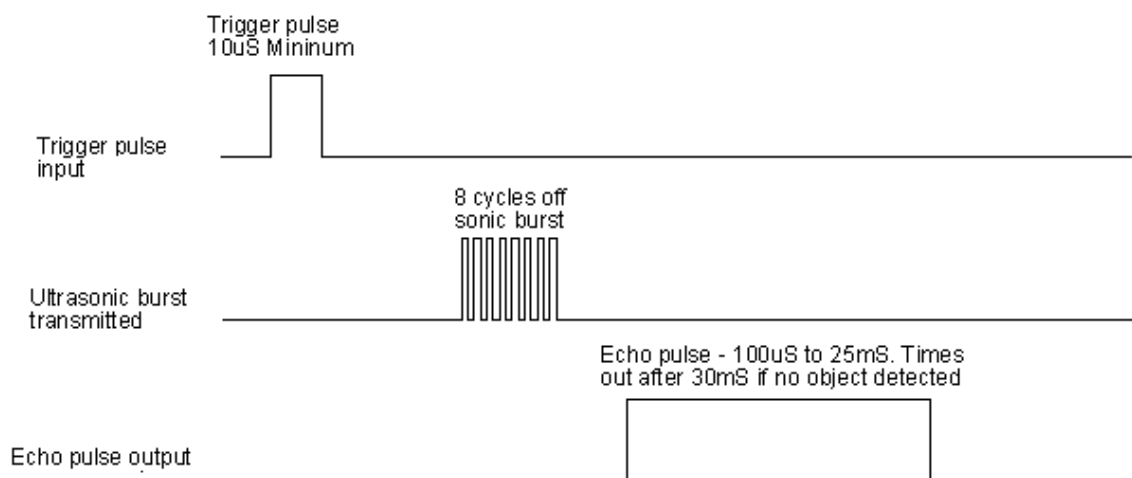


Figure 3.8 : Chronogramme de fonctionnement du capteur ultrason

III.4.2.4. La fonction du capteur ultrason dans le système étudié

Lors de l'activation de l'alarme les capteurs de chaque pièce permettent d'effectuer des mesures. On définit la mesure maximale à 30Cm (qui est le mur). Si un obstacle est détecté à une distance inférieure à cette dernière, cela implique qu'il y a une intrusion dans le domicile.

Par conséquent le système d'alarme est enclenché. (La figure suivante illustre le module ultrason ainsi que sa connexion avec l'Arduino méga)

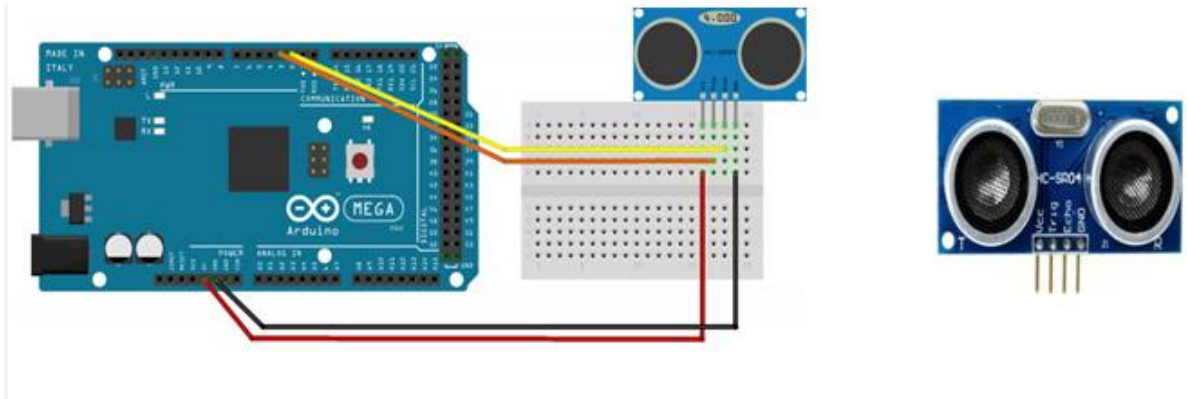


Figure 3.9 : Le module ultrason et sa connexion avec l'Arduino méga

III.4.2.5. Le bipper

Un bipper ou buzzer (voir figure 3.11) est un élément électromécanique ou piézoélectrique qui produit un son caractéristique quand on lui applique une tension : le bip. Certains nécessitent une tension continue, d'autres nécessitent une tension alternative.

Dans notre cas c'est un buzzer à courant continu de 6V à fréquence variable. Ce bipper là est activé par un signal d'horloge externe : la fréquence du signal sonore est alors relative au signal appliqué. Cette fréquence est généralement comprise entre 2 kHz et 4 kHz. Ce type de bippers peut générer des bips graves, aigus, plus graves ou plus aigus.



Figure 3.10: Illustration d'un bipper

III.4.2.6. Le servomoteur [12]

Un servomoteur est un système qui a pour but de produire un mouvement précis en réponse à une commande externe. C'est un actionneur (système produisant une action) qui mélange l'électronique, la mécanique et l'automatique.

Les servomoteurs répandus dans la robotique mobile dits de modélisme par leur taille, en effet leur petite taille les destine à une utilisation dans des modèles réduits nécessitant un

encombrement réduit, ils sont composés de trois parties : d'un moteur électrique, d'un ensemble d'engrenage, et d'une carte électronique pour la commande et l'asservissement du moteur. (La figure suivante définit la composition interne d'un servomoteur)

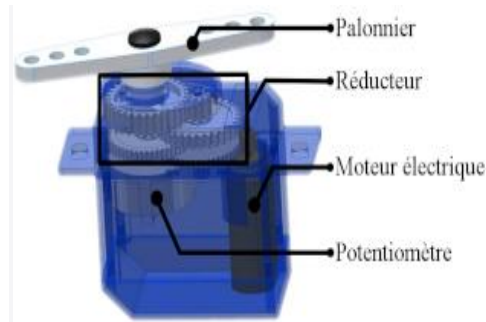


Figure 3.11: La composition interne d'un servomoteur

Ce dernier est un moteur qui est utilisé pour maintenir un axe de rotation dans certaines positions bien précises. C'est un moteur électrique associé à une électronique de commande pilotée par une impulsion de durée variable qui va déterminer la position fixe de l'axe de sortie.

Sur un servomoteur, l'axe peut prendre une position comprise entre 0° et 180° selon la durée de l'impulsion reçue ($1000\mu s$ pour 0° , $1500\mu s$ pour 90° et $2000\mu s$ pour 180°), ces valeurs pouvant varier d'un servomoteur à l'autre.

- **Branchage d'un servomoteur**

Typiquement, un servomoteur présente 3 broches :

- un fil noir à connecter à la masse.
- un fil rouge à connecter au +.
- un fil blanc à connecter à la broche de commande par impulsion.

On peut utiliser un servomoteur directement en +5V et avec une entrée numérique PWM sur notre Arduino (Voir la figure ci-dessous).

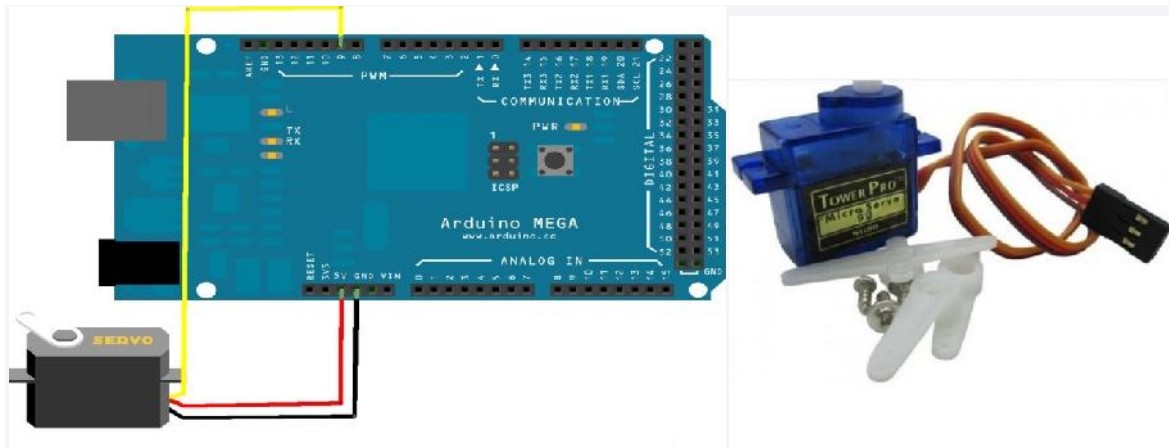


Figure 3.12: Illustration d'un servomoteur ainsi que le schéma de connexion avec l'Arduino méga

Remarque : Dans notre système, le servomoteur sert à balayer la caméra afin de suivre les zones où il y a des intrusions.

III.4.2.7. La LED :

Une **diode électroluminescente** (*light emitting diode*, **LED**), est un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.

Une diode électroluminescente ne laisse passer le courant électrique que dans un seul sens (le sens passant, comme une diode classique, l'inverse étant le sens bloquant) et produit un rayonnement monochromatique ou polychromatique non cohérent à partir de la conversion d'énergie électrique, lorsqu'un courant la traverse. Ceci est illustré par la figure ci-dessous.

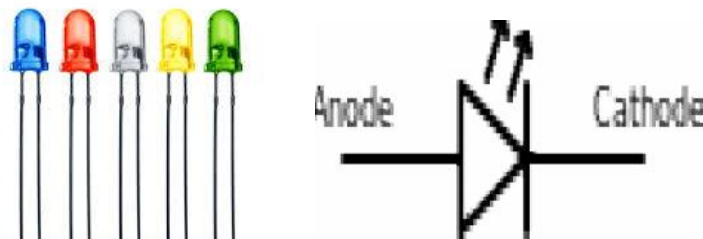


Figure 3.13: Illustration de différentes LED ainsi que leur schéma électrique

III.4.2.8. La liaison Bluetooth

a. Définition

Le **Bluetooth** est un standard de communication permettant l'échange bidirectionnel de données à très courte distance et utilisant des ondes radio UHF. Son objectif est de simplifier les connexions entre les appareils électroniques en supprimant des liaisons filaires. Elle peut remplacer par exemple les câbles entre ordinateurs, tablettes, téléphones mobiles entre eux ou avec des imprimantes, scanners, claviers, souris, manettes de jeu vidéo, téléphones portables, PDA, systèmes et kits mains libres micro et/ou écouteurs, autoradios, appareils photo numériques, lecteurs de code-barres et bornes publicitaires interactives.

Le standard Bluetooth, à la manière du wifi utilise la technique FHSS (Frequency Hopping Spread Spectrum, en français étalement de spectre par saut de fréquence ou étalement de spectre par évaison de fréquence), consistant à découper la bande de fréquence (2.402 - 2.480 GHz) en 79 canaux (appelés hops ou sauts) d'une largeur de 1MHz, puis de transmettre en utilisant une combinaison de canaux connue des stations de la cellule.

Ainsi, en changeant de canal jusqu'à 1600 fois par seconde, le standard Bluetooth permet d'éviter les interférences avec les signaux d'autres modules radio.(La figure suivante illustre Logo du Bluetooth)



Figure 3.14 : Logo du Bluetooth

b. Le module Bluetooth

Le module Bluetooth HC-06 ajoute une fonction de communication par Bluetooth à la carte Arduino. Il peut ainsi communiquer sans fil à moyenne distance avec n'importe quel autre dispositif Bluetooth (ordinateur, téléphone mobile, second module Bluetooth sur une autre carte Arduino ou Raspberry ...).

Caractéristiques:

- Module de type JY-MCU / HC-06 Master
- Paramétrage par défaut du port série: 9600, N, 8, 1.
- Le baud rate est paramétrable de 4800 à 1382400 (par commande AT, uniquement si le module n'est pas associé, et maxi 115200 pour pouvoir l'utiliser avec une carte Arduino).

- Alimentation de 3.3 à 5V DC
- LED indicatrice : statut de connexion.
- Fonctionnement Bluetooth sur la bande 2.4 GHz, modulation GFSK.
- Le module est apparié avec un mot de passe (1234) modifiable.
- VCC → +5V
- GND → GND
- TXD → Pin 0 (RX) vert
- RXD → Pin 1 (TX) noir

Remarque :

Il faut croiser Rx et Tx entre le module HC-06 et l'Arduino.

L'utilisation de Pin0 et Pin1 sur la carte Arduino ne permet pas d'utiliser le serial monitor du PC en même temps, ni de téléverser (donc penser à débrancher ces 2 fils pour téléverser le code par le câble USB).

La figure suivante illustre le module Bluetooth et sa connexion avec l'Arduino :

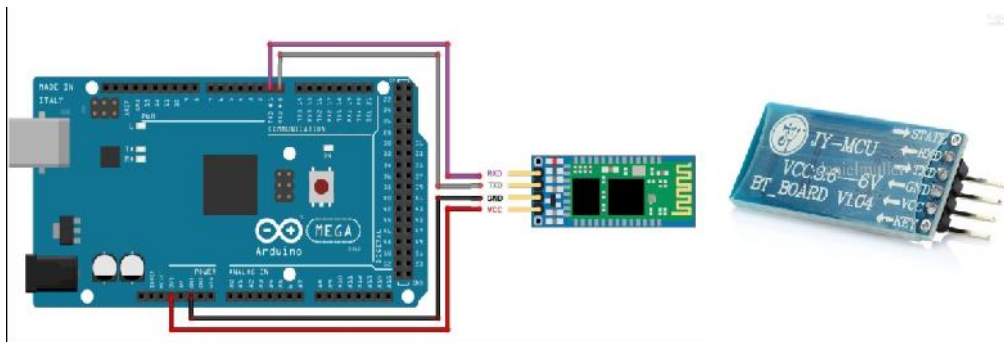


Figure 3.15: Le module Bluetooth ainsi que sa connexion avec l'Arduino

III.5. Carte de commande d'acquisition vidéo

III.5.1. La carte Raspberry Pi [13]

La Raspberry Pi est un nano-ordinateur mono-carte à processeur ARM conçu par le créateur de jeux vidéo David Braben, dans le cadre de sa fondation Raspberry Pi. La Raspberry Pi est un petit ordinateur sous le système d'exploitation Linux sur carte SD destiné à des applications d'informatique embarquée. Le cœur de l'ordinateur est un FPGA (Broadcom 2835) intégrant un processeur ARMv7 cadencé à 900MHz avec 1Ghz de RAM et de nombreux périphériques.

La Raspberry Pi peut être directement connectée à une IHM classique, souris / clavier / écran HDMI ou vidéo composite. Cependant comme tout ordinateur Linux, elle peut intégrer ses propres outils de développement et une interface homme machine reposant sur SSH contrôlable depuis un autre ordinateur par Ethernet ou WIFI.

Le connecteur d'extension supporte les entrées/sorties parallèles ainsi que la plupart des bus de communication.

C'est un support particulièrement économique et puissant qui peut être facilement mis en œuvre dans de petits systèmes nécessitant un accès au monde physique par des capteurs / actionneurs disposants d'interfaces numériques. (La figure suivante montre une illustration détaillé de la Raspberry PI 2

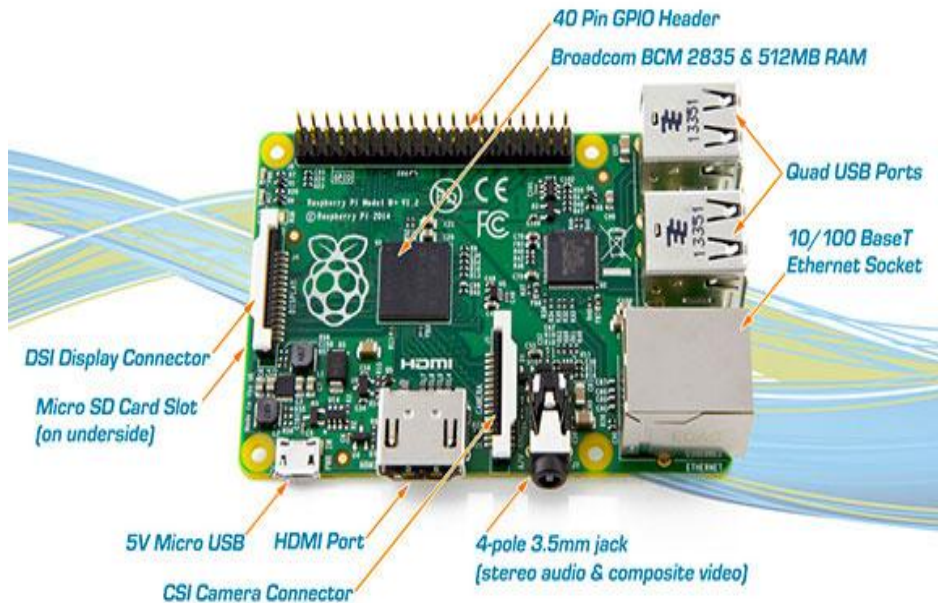


Figure 3.16: Illustration détaillé de la Raspberry PI 2

III.5.2. Connectiques et ports disponibles sur la Raspberry Pi 2[13]

- Double abaisseur de tension d'alimentation (buck) pour le 3,3 V et le 1,8 V
- Le 5V dispose d'une protection contre l'inversion de polarité, d'un fusible de 2A et d'une protection pour les branchements à chaud
- Nouvelle puce contrôleur USB / Ethernet
- 4 ports USB au lieu de 2 ports
- 40 broches GPIO dont 2 broches d'identification d'EEPROM
- La sortie vidéo composite (NTSC/PAL) intégrée au jack audio de 3,5mm
- Support de carte Micro SD.
- Quatre trous de montage en disposition rectangulaire
- Connecteur d'alimentation micro USB.
- Le port HDMI pour l'affichage sur un moniteur de haute définition.
- Connecteur CSI pour module caméra et d'affichage DSI.

III.5.3. Préparation de la carte SD

La Raspberry Pi n'a pas de disque dur pour le stockage de masse, mais un lecteur de carte mémoire SD (*Secure Digital*) telles celles que l'on peut trouver équipant les appareils photo numériques.

Avant de démarrer la Raspberry Pi, il faut préparer une carte SD et y installer la totalité du **système d'exploitation** (ou OS pour *Operating System*).

La Raspberry Pi est conçue pour faire fonctionner le système d'exploitation GNU **Linux**. La philosophie open source de Linux a permis de porter rapidement l'OS à l'architecture matérielle de la Raspberry Pi.

Il existe à l'heure actuelle plusieurs variantes de Linux utilisables avec le circuit de la Raspberry Pi, appelées **distributions Linux** (Raspbian, Arch, Pidora, etc.). Par la suite, nous utiliserons une distribution Debian portée et optimisée pour Raspberry Pi nommée Raspbian.

III.5.4. Les ports GPIO (General Purpose Input / Output) [13]

Ce sont des ports d'entrée/sortie très utilisés dans le monde des microcontrôleurs, en particulier dans le domaine de l'électronique embarquée. Le connecteur GPIO supporte les GPIO (entrées/sorties binaires) mais également les sorties PWM, les périphériques de communication (UART, I2C, SPI) et les alimentations 5v et 3V3.

Les broches peuvent avoir des fonctions différentes suivant qu'elles soient activées en tant que GPIO ou périphérique de communication. Certaines possèdent des résistances de pull up donnant un bit à 0 dominant et un bit à 1 récessif. (La figure suivante illustre un schéma expliquant les différentes fonctions des ports GPIO ainsi que le schéma électrique)

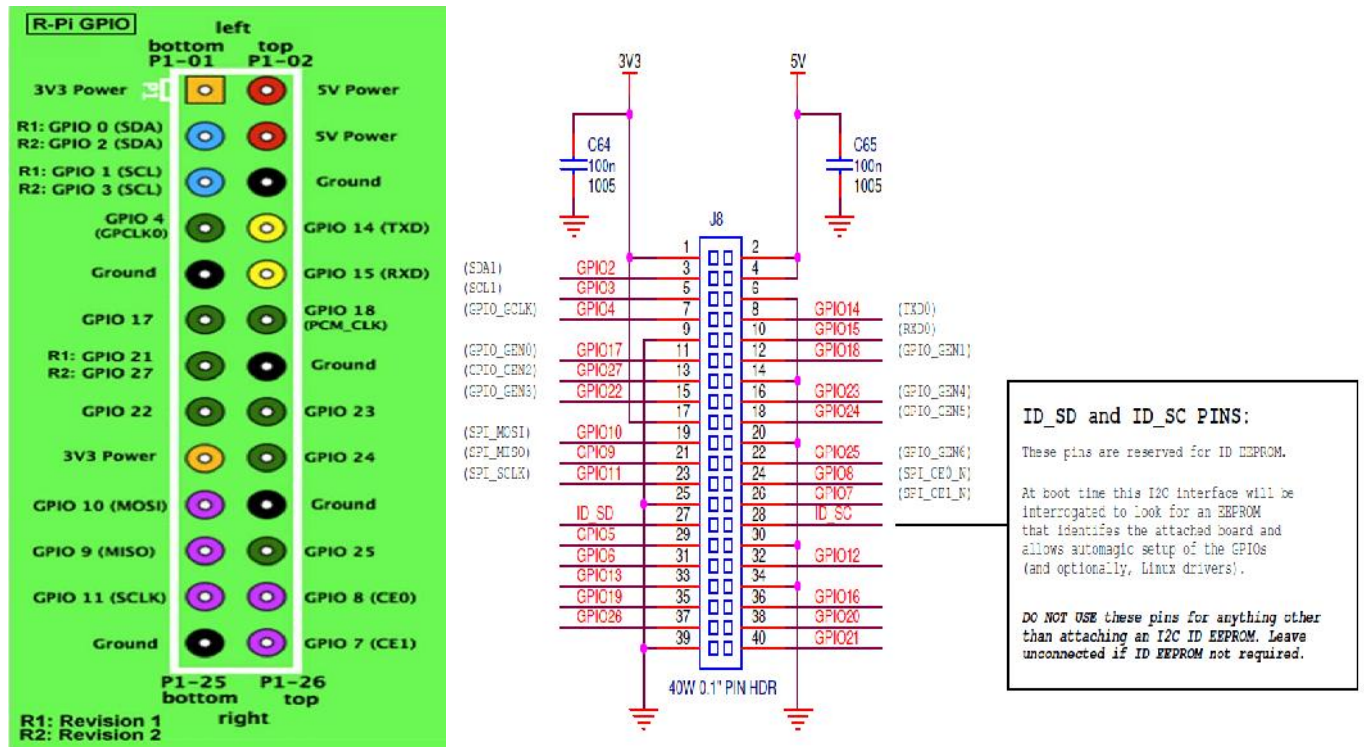


Figure 3.17: Schéma expliquant les différentes fonctions des ports GPIO ainsi que schéma électrique

III.5.5. Dimensions de la Raspberry Pi 2 [13]

La Raspberry Pi2 dispose de 85,60 mm de longueur et de 53,98 mm de largeur avec un poids de 45 g .Elle est dotée aussi de Quatre trous de vis permettant à la carte d'être fixée sur une surface ou dans un boîtier afin de l'embarquer sur un système. (Voir figure ci-dessous)

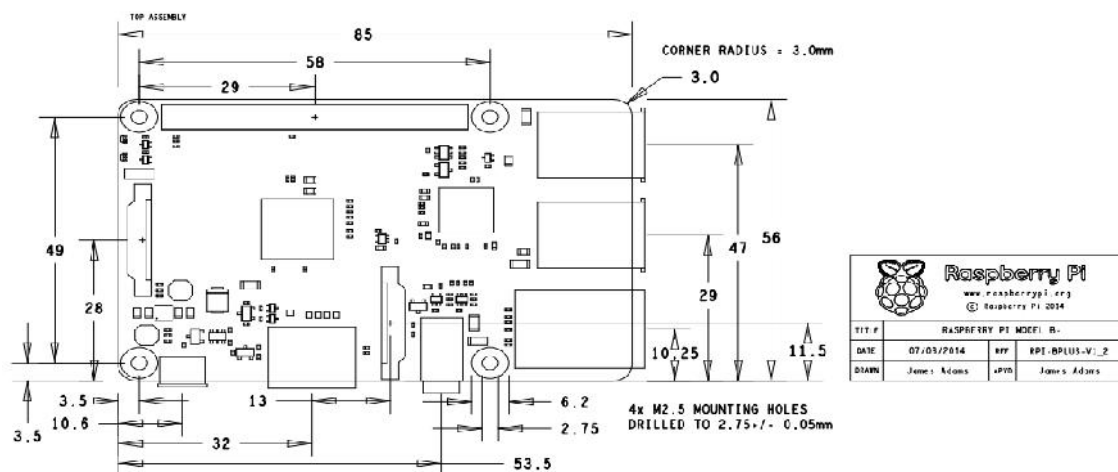


Figure 3.18: Les dimensions de la Raspberry Pi 2

III.5.6. La liaison avec la camera

La raspicam prend en charge 1080 p @ 30 fps, 720 p à 60 images par seconde et 640 x 480 p enregistrement vidéo 60/90 aussi est pris en charge par la dernière version de Raspbian, système d'exploitation recommandé pour la Raspberry Pi. (Voir figure suivante)



Figure 3.19: La carte Raspberry avec le module Raspicam

Vu l'absence du module raspicam on relie la raspberry a une webcam a haute résolution. (voir illustration ci-dessous)



Figure 3.20: La carte Raspberry avec une webcam embarquée

III.5.7. La Raspberry Pi 2 sur le réseau

Pour connecter la Raspberry Pi au réseau, on profite du fait que le réseau où les réseaux sont connectés à la Raspberry PI 2 profite d'un service DHCP (*Dynamics Host Configuration Protocol*) dont la fonction est de distribuer des adresses IP aux hôtes connectés.

Le client DHCP est activé par défaut sur la distribution Linux proposée et installée sur la carte SD de sorte qu'on a plus qu'à connecter la Raspberry Pi à une prise Ethernet ou avec un module wifi.

III.5.8. Communication série de la Raspberry Pi

Les signaux numériques envoyés par la Raspberry Pi (via la GPIO nommée Tx) sont compatibles avec le protocole RS-232 (dont les niveaux de tension sont adaptés, si le niveau logique « 0 » est normalement à +12 V et le niveau logique « 1 » à -12 V dans le protocole RS-232, ils sont adaptés à des niveaux de tension compatibles avec le processeur Broadcom à respectivement 0 V et 3,3 V).

III.6. Arduino et la réalisation pratique

Un programme à base du langage Arduino a été développé afin que le système fonctionne convenablement et avec fiabilité. Le programme conçu détaillé est reporté en Annexe B.

La figure ci-dessous permet de voir la maquette conçue, qui est une maison fonctionnant selon un système domotique.



Figure 4.21: Figure de la maquette conçue

III.7. Conclusion

Dans ce chapitre sont décrites les deux cartes de programmation qui sont la carte Arduino Méga et la Raspberry PI 2. Les composants utilisés dans chaque carte de programmation sont décortiqués et définis ainsi que le principe de fonctionnement et la liaison Bluetooth.

Chapitre IV

Description des outils logiciels du système

IV.1. Introduction :

Après avoir conçu le système, la réalisation pratique fera l'objet de ce chapitre. Il se présente sous forme logicielle, commençant par le langage de programmation Arduino, ensuite la présentation du langage de programmation orienté objet, Python, l'environnement de développement Raspbian Wheezy, le serveur apache, le PHP et pour finir Android .

IV.2. Le logiciel Arduino

IV.2.1. Introduction

Une équipe de développeurs composée de *Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti* a imaginé un projet répondant au doux nom de **Arduino** et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques. Vocabulaire commun propre au domaine de l'électronique et de l'informatique est résumé pour plus de compréhension.

Le système Arduino permet de réaliser un grand nombre de réalisations, qui ont une application dans tous les domaines. L'étendue de l'utilisation de l'Arduino est gigantesque.

Voici quelques exemples:

- contrôler les appareils domestiques
- fabriquer un robot
- faire un jeu de lumières
- communiquer un ordinateur
- télécommander un appareil mobile (modélisme)

(La figure ci-dessous illustre le logo Arduino)



Figure 4.1: Logo Arduino

IV.2.2. Composition du logiciel Arduino

Le logiciel permet de programmer la carte Arduino. Il offre une multitude de fonctionnalités et il est composé de :

- **Une BARRE DE MENUS** : comme pour tout logiciel une interface graphique (GUI) .

Chapitre IV: Description des outils logiciels du système

- Une **BARRE DE BOUTONS** : qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation.
- Un **EDITEUR** (à coloration syntaxique) pour écrire le code de vos programmes, avec onglets de navigation.
- Une **ZONE DE MESSAGES** qui affiche et indique l'état des actions en cours.
- Une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme.

La figure suivante illustre la composition d'une fenêtre du logiciel Arduino

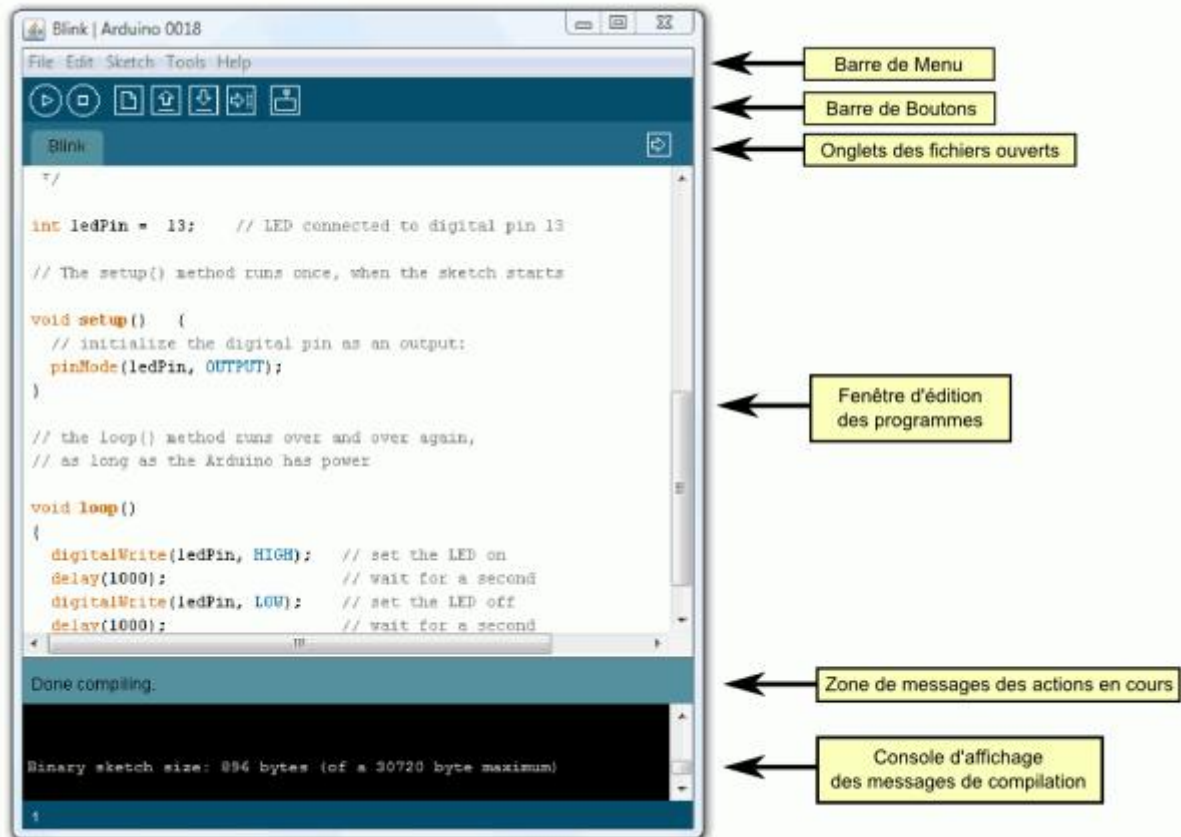


Figure 4.2: Approche et utilisation du logiciel

Le code écrit avec le logiciel Arduino est appelé un programme (ou une séquence - sketch en anglais) :

- Ces programmes sont écrits dans l'**éditeur de texte**. Celui-ci a les fonctionnalités usuelles de copier/coller et de rechercher/remplacer le texte.
- la **zone de messages** donne l'état de l'opération en cours lors des sauvegardes, des exportations et affiche également les erreurs.
- La **console texte** affiche les messages produits par le logiciel Arduino, incluant des messages d'erreur détaillés et d'autres informations utiles.
- la **barre de boutons** permet de vérifier la syntaxe et de transférer les programmes, créer, ouvrir et sauver le code, et ouvrir le moniteur série.
- la **barre des menus** permet d'accéder à toutes les fonctionnalités du logiciel Arduino.

IV.2.3. Description des menus

Il ya cinq principales commandes dans le menu :

- **File** (Fichier)
- **Edit** (Editer)
- **Sketch** (Programme ou Séquence)
- **Tools** (Outils)
- **Help** (Aide)

IV.2.3 .1. Le menu File

C'est principalement ce menu qui va être utilisé le plus. Il dispose d'un certain nombre de fonctions qui vont être très utiles. Voici quelques options :

- **Carnet de croquis** : ce menu regroupe les fichiers qu'on aurait pu faire jusqu'à maintenant (et s'ils sont enregistré dans le dossier par défaut du logiciel)
- **Exemples (exemples)** : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existants ; avec ça, on peut aider/inspirer pour créer des programmes ou tester de nouveaux composants.
- **Téléverser** : Permet d'envoyer le programme sur la carte Arduino.
- **Téléverser avec un programmeur** : Idem que si dessus, mais avec l'utilisation d'un programmeur (vous n'en n'aurez que très rarement besoin).
- **Préférences** : Quelques paramètres du logiciel peuvent être réglés ici.
- **New (nouveau)** : Permet de créer un nouveau programme. Quand on appuie sur ce bouton, une nouvelle fenêtre, identique à celle de la figure 4.2
- **Open... (ouvrir)** : Avec cette commande, un programme existant peuvent être ouvert.
- **Save / Save as... (enregistrer / enregistrer sous ...)** : Enregistre le document en cours.
- **Exemples (exemples)** : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existants. (Voir la figure 4.3)

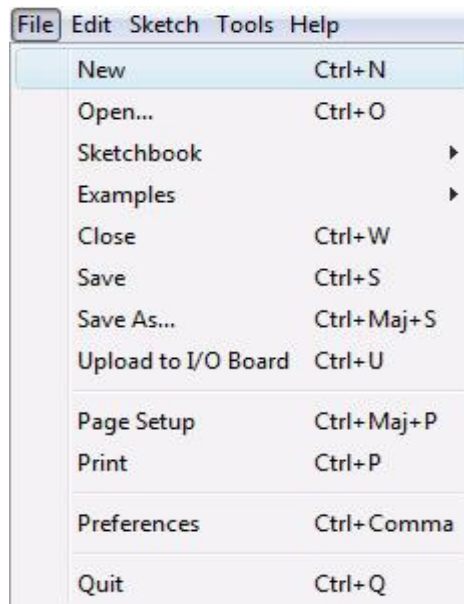


Figure 4.3: Contenu du menu " File "

IV.2.3.2. Le menu Edit (Editer)

- Copy for forum (copier pour le forum) : Copie le code du programme dans le presse-papier dans un format approprié pour poster sur le forum, avec coloration syntaxique complète.
- Copy as HTML (copier en tant qu'HTML) : Copie le code de votre programme dans le presse-papier en tant qu'HTML, au format adapté pour être intégré dans des pages web. (Voir figure ci-dessous)

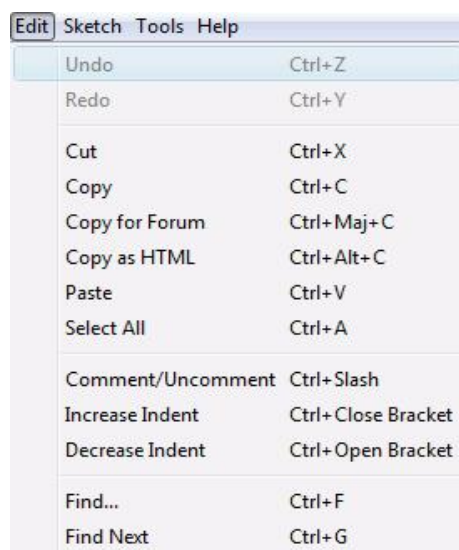


Figure 4.4: Contenu du menu "Edit"

IV.2.3.3. Le menu Sketch

- **Verify/Compile (Vérifier/compiler)** : Vérifie le code à la recherche d'erreurs.
- **Import Library (Importer la librairie)** : Ajoute une librairie au programme en insérant l'instruction **#include** dans le code.
- **Show Sketch Folder (Montrer le répertoire du programme)** : Ouvre le répertoire courant du programme sur le bureau.
- **Add File** : Ajouter un fichier . (Voir figure ci-dessous)

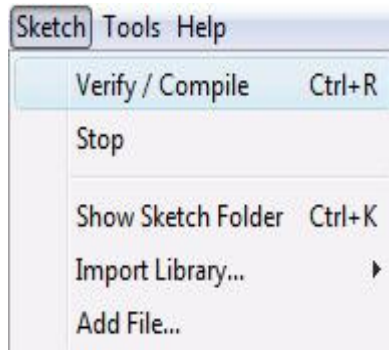


Figure 4.5: Contenu du menu "Sketch"

IV.2.3.4. Le menu Tools

- **Auto Format (Mise en forme Automatique)** : Cette fonction formate le code, c'est à dire ajuste le code de façon à ce que les accolades soient alignées et que les instructions entre les accolades soient décalées.
- **Board (Carte)** : sélectionne la carte Arduino utilisée.
- **Serial Port (Port Série)** : Ce menu contient tous les ports séries (réels ou virtuels) présents sur l'ordinateur. Il est automatiquement mis à jour à chaque fois que le niveau supérieur du menu outils.
- **Burn Bootloader (Graver le bootloader)** : Cette fonctionnalité permet de graver le bootloader dans le microcontrôleur sur une carte Arduino. Ceci n'est pas nécessaire pour une utilisation normale de la carte Arduino (le bootloader est déjà gravé dans la carte à l'achat) mais peut être utile à l'achat d'un nouvel ATmega (qui est livré normalement sans bootloader). (Voir la figure suivante)

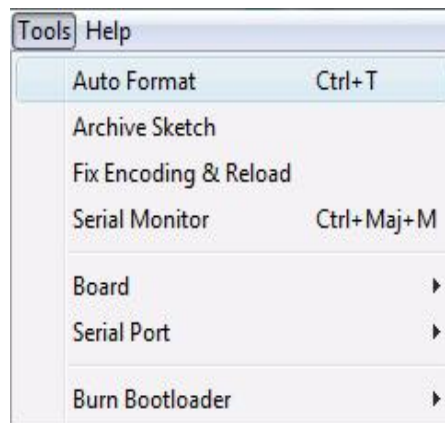


Figure 4.6: Contenu du menu "tools"

IV.2.3.5. Le menu Help

- Propose tout ce qui peut être utile pour l'aider lors de l'écriture des programmes, notamment un lien vers la référence du langage en anglais.

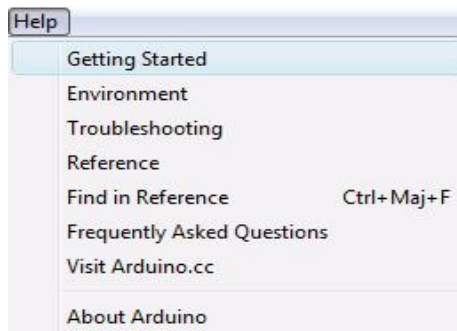


Figure 4.7: Contenu du menu "Help"

IV.2.4. Présentation des boutons

Chaque bouton, encadré en rouge et numéroté de 1 à 6 présente une fonction bien déterminée. (Ceci est illustré par la figure ci-dessous)

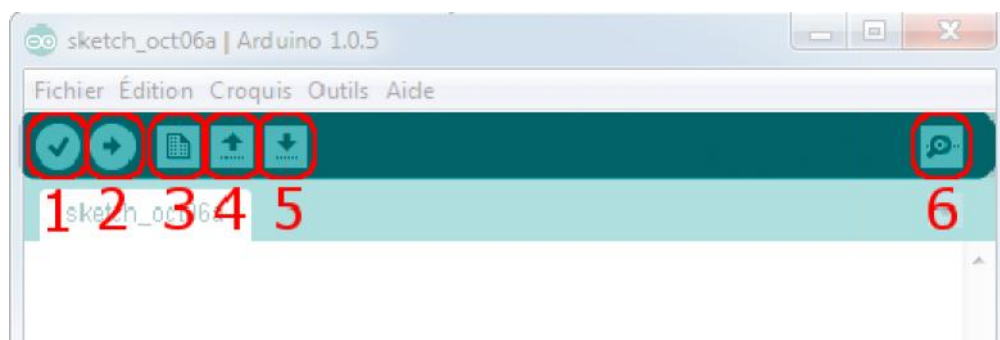


Figure 4.8: Présentation des boutons

- **Bouton 1** : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme
- **Bouton 2** : Charge (téléverse) le programme dans la carte Arduino
- **Bouton 3** : Crée un nouveau fichier
- **Bouton 4** : Ouvre un fichier
- **Bouton 5** : Enregistre le fichier
- **Bouton 6** : Ouvre le moniteur série

IV.2.5. Les librairies

Les librairies fournissent des fonctions nouvelles qui peuvent être utilisées dans les programmes, par exemple un matériel précis (un afficheur LCD...) ou manipuler des données. Pour utiliser une librairie, il faut la sélectionner depuis le menu **Sketch > Import Library** (Programme > Importer Librairie). Cela insèrera une ou plusieurs instructions **#include** au début du programme et compilera la librairie avec le programme.

Puisque les librairies sont transférées dans la carte avec le programme, elles augmentent la quantité de mémoire utilisée. Si un programme ne nécessite plus d'une librairie, il faut effacer simplement l'instruction **#include** correspondante au début du code.

Il y a toute une liste de librairies dans la référence du langage Arduino. Certaines librairies sont incluses dans le logiciel Arduino (la librairie Serial pour les communications série notamment). D'autres librairies peuvent être téléchargées depuis différentes sources (voir notamment : <http://www.arduino.cc/playground/Main/InterfacingWithHardware>)

IV.2.6. Le terminal série

Le **TERMINAL SERIE** (figure 4.9) permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permettant d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique. C'est un élément essentiel pour améliorer, tester et corriger les programmes.

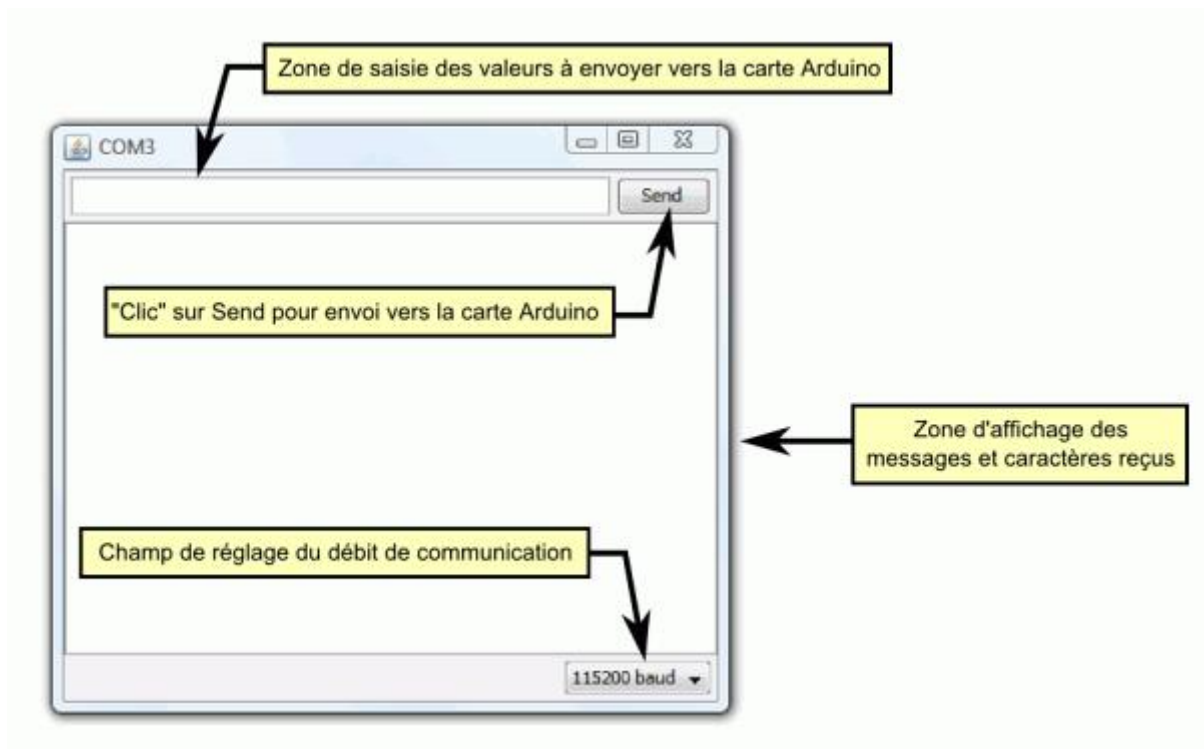


Figure 4.9: Présentation du terminal série

Pour ouvrir le terminal série: Il faut cliquer sur le serial monitor (figure ci-dessous)

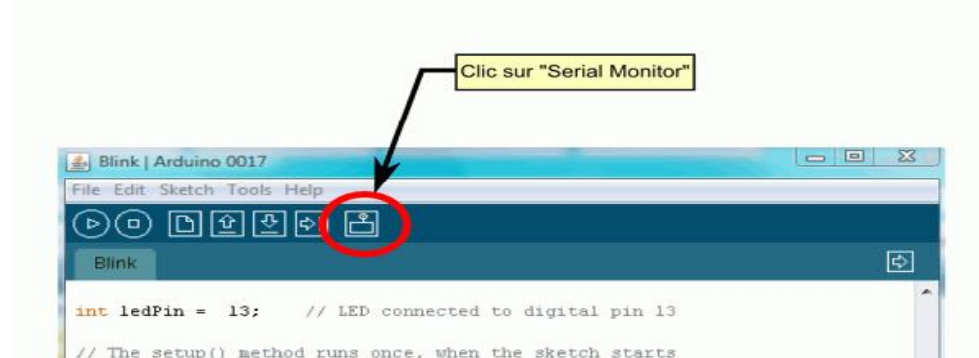


Figure 4.10: Ouverture du terminal série

Pour régler le débit de communication sur la même valeur que celle utilisée par le programme avec lequel programmer la carte Arduino :

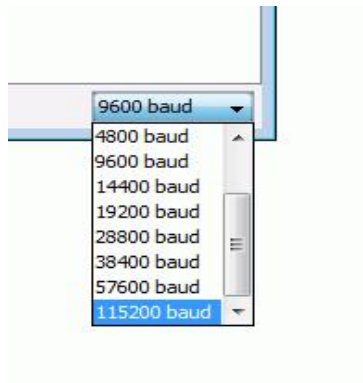


Figure 4.11: Ajustage du débit de la communication série

IV.2.7. Ouvrir un programme et le téléverser

1ere étape: Lancer un programme de base

Nous allons choisir un exemple simple qui consiste à faire clignoter une LED, son nom est *Blink* et il est dans la catégorie *Basics* :

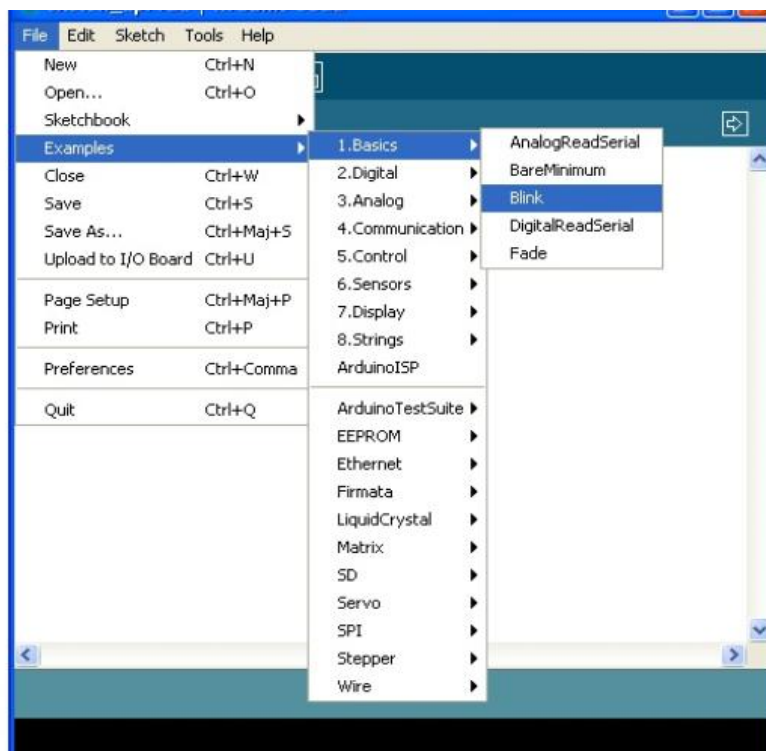


Figure 4.12: Ouvrir le programme Blink

Après avoir cliqué sur *Blink*, une nouvelle fenêtre va apparaître. Elle va contenir le programme *Blink*. L'ancienne fenêtre peut être fermée puisqu'elle n'est plus utile.

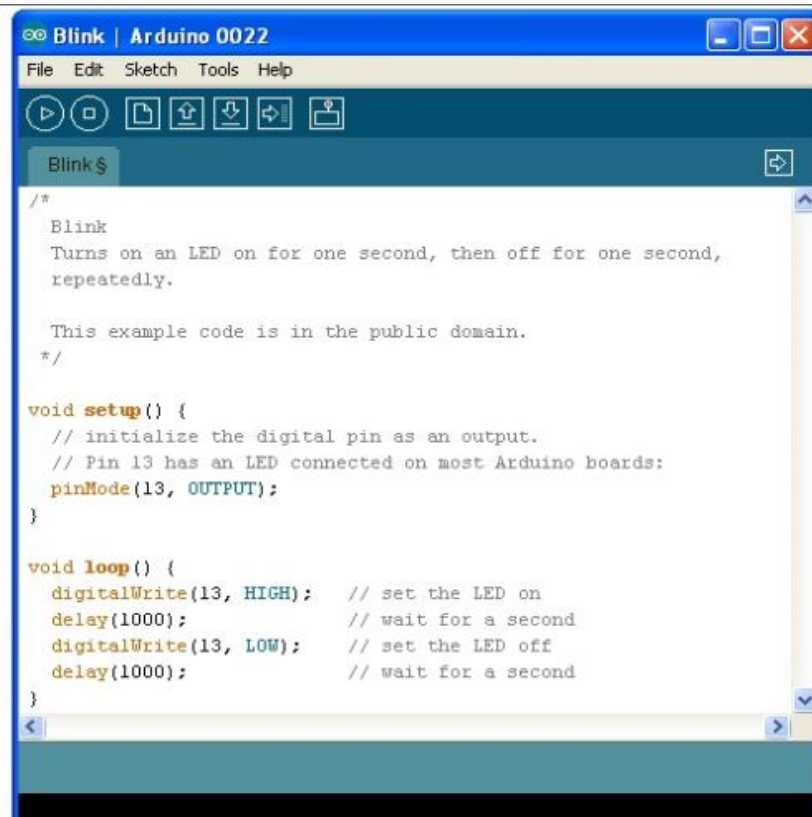


Figure 4.13: Contenu du programme Blink

2ème étape: Choisir la carte de programmation

Avant d'envoyer le programme *Blink* vers la carte, il faut donner au logiciel le nom de la carte et sur quel port elle est branchée. Et cela en allant sur " *Tools*" puis dans "*Board* " (" carte" en français). On vérifie que c'est bien le nom " Arduino Méga ADK" qui est coché..

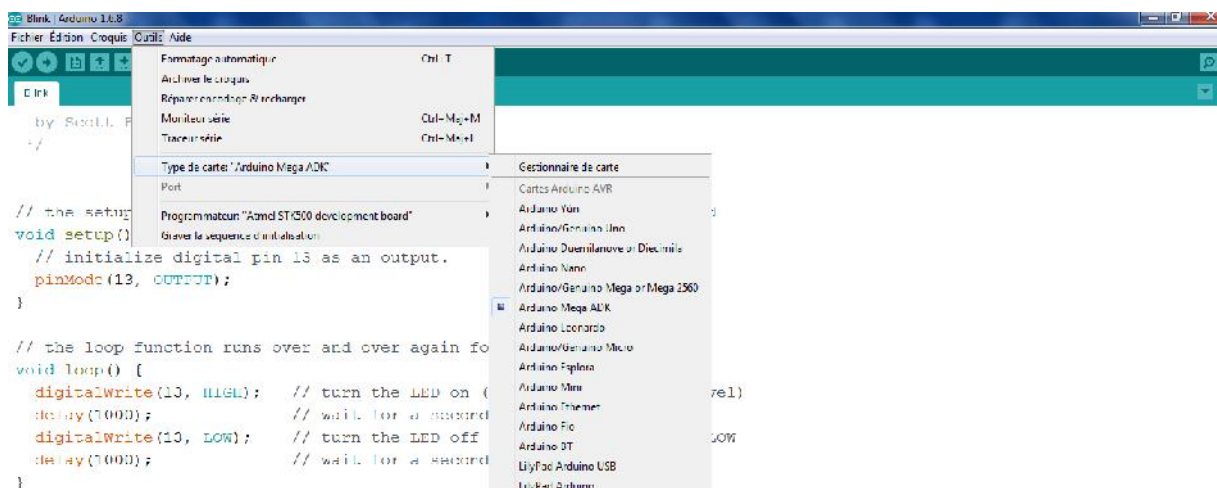


Figure 4.14: Choix de la carte Arduino

3ème étape: Choisir le port de connexion de la carte

Il faut partir dans le menu *Tools*, puis *Serial port*, ou on choisit le port COMX, X étant le numéro du port qui est affiché. Dans notre cas, il s'agit de COM5 :

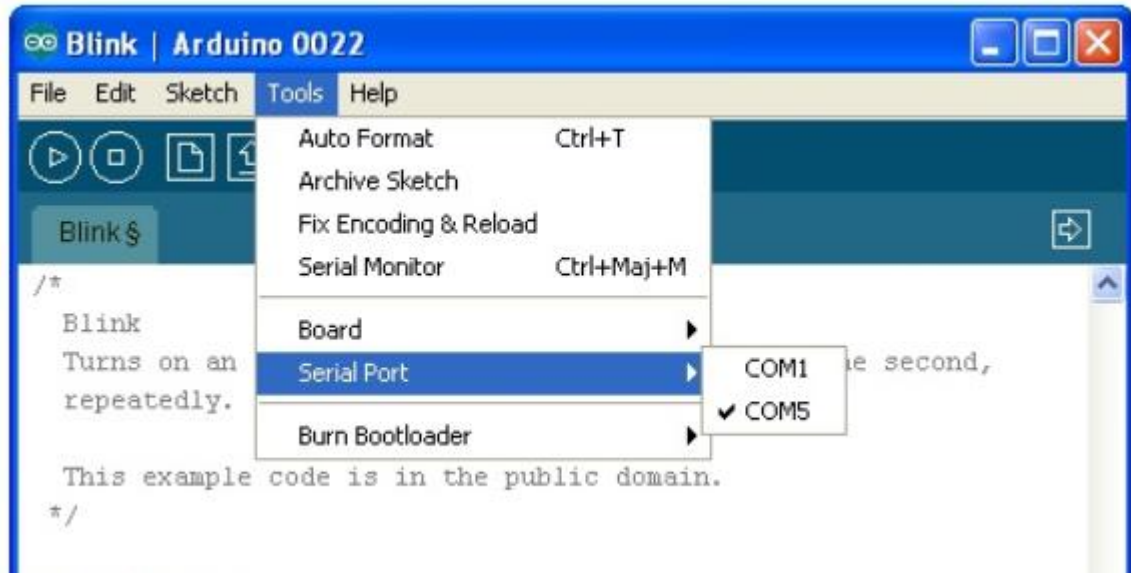


Figure 4.15: Choix du port de connexion de la carte

4ème étape: Téléverser le programme

Après les étapes passées le programme est envoyé dans la carte. Pour ce faire, il suffit de cliquer sur le bouton *Upload* (ou "Téléverser" en Français)

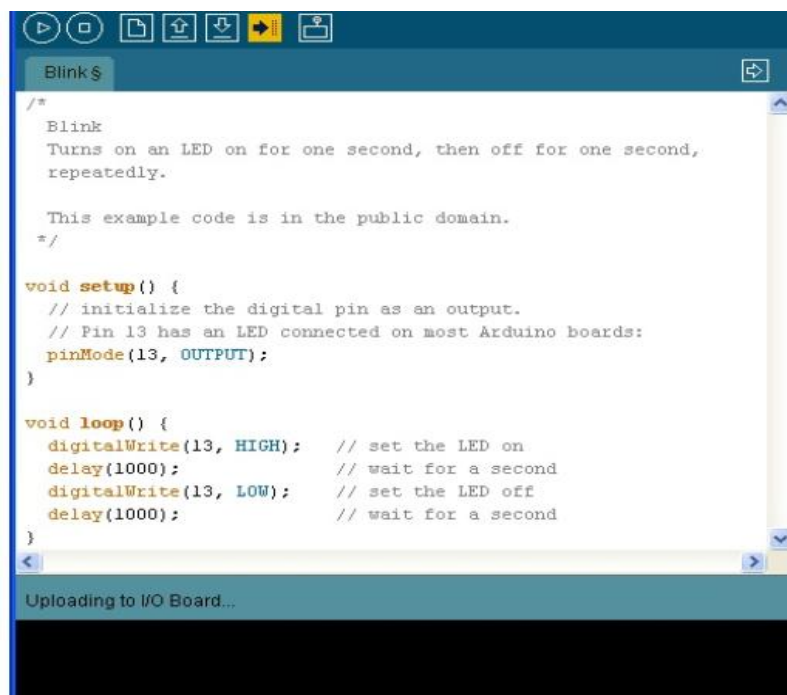
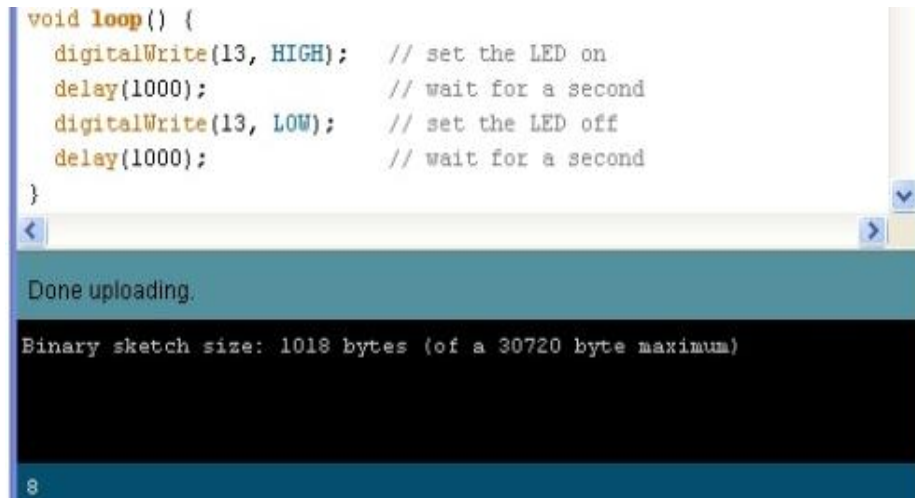


Figure 4.16: Envoi du programme Blink

Chapitre IV: Description des outils logiciels du système

En bas dans l'image, le texte : " *Uploading to I/O Board...* " est affiché, cela signifie que le logiciel es t en train d'envoyer le programme dans la carte. A la fin, il affiche un autre message :



```
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW);  // set the LED off  
  delay(1000);           // wait for a second  
}
```

Done uploading.

Binary sketch size: 1018 bytes (of a 30720 byte maximum)

8

Figure 4.17: Fin de l'upload

Le message afficher : " *Done uploading* " signale que le programme à bien été chargé dans la carte. Si la carte fonctionne, la LED sur la carte clignote.

IV.3. Le langage Python [14]

Python est un langage de programmation objet, multi paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions . Il est ainsi similaire à Perl , Ruby , Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix en passant par GNU/Linux, Mac OS, ou encore Android, i OS, et aussi avec Java ou encore NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Son langage a une syntaxe claire et permet une initiation aisée aux concepts de base de la programmation.

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



Figure 4.18: Logo du Python

IV.3.1. Syntaxe du Python

Python a été conçu pour être un langage lisible. Il vise à être visuellement épuré. Il possède moins de constructions syntaxiques que de nombreux langages structurés tels que C, Perl, ou Pascal. Les commentaires sont indiqués par le caractère croisillon (#).

Chapitre IV: Description des outils logiciels du système

Les blocs sont identifiés par l'indentation, au lieu d'accolades comme en C ou C++ ; ou de `begin ... end` comme en Pascal ou en Ruby. Une augmentation de l'indentation marque le début d'un bloc, et une réduction de l'indentation marque la fin du bloc courant

Fonction factorielle en C	Fonction factorielle en Python
<pre>/* Fonction factorielle en C */ int factorielle(int x) { if (x < 2) { return 1; } else { return x * factorielle(x-1); } }</pre>	<pre># Fonction factorielle en Python def factorielle(x): if x < 2: return 1 else: return x * factorielle(x-1)</pre>

Tableau 4.1 :Tableau comparatif entre la syntaxe du C et celle du Python

NB : l'indentation pourrait être modifiée ou supprimée dans la version en C sans modifier son comportement, mais une indentation correcte permet de détecter plus aisément des erreurs en cas d'imbrication de plusieurs blocs et facilite donc l'élimination de ces erreurs. C'est pourquoi il est préférable d'indenter convenablement les programmes en C.

Les mots-clés sont fournis dans la liste `keyword.kwlist` du module `keyword`. Les mots-clés de Python 2.7.5 sont les suivants: `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `exec`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `not`, `or`, `pass`, `print`, `raise`, `return`, `try`, `while`, `with`, `yield`.

À partir de Python 3.0, `print` et `exec` ne sont plus des mots-clés du langage, mais des fonctions du module `builtins`. Sont ajoutés aux mots-clés : `True`, `False`, `None` et `nonlocal`. Les trois premiers étaient déjà présents dans les versions précédentes, mais ils ne sont plus modifiables (auparavant, l'affectation `True = 1` était possible). `Nonlocal` a été introduit par le PEP 3104, et permet, dans une fonction définie à l'intérieur d'une autre fonction, de modifier une variable d'un niveau supérieur de portée. Avant cela, seules les variables locales à la fonction, et globales (niveau module) étaient modifiables.

IV.4. Le Raspbian

Le Raspbian est un système d'exploitation libre basé sur la distribution GNU/Linux Debian, et optimisé pour le plus petit ordinateur du monde, la Raspberry Pi.

Raspbian ne fournit pas simplement un système d'exploitation basique, il est aussi livré avec plus de 35 000 paquets, c'est-à-dire des logiciels pré-compilés livrés dans un format optimisé, pour une installation facile sur la Raspberry Pi via les gestionnaires de paquets. La première version des 35 000 paquets Raspbian, optimisés pour la Raspberry Pi, a été achevée en Juin 2012.

IV.4.1. L'utilité de Raspbian avec la Raspberry Pi2

La Raspberry Pi est une carte assez puissante, mais de puissance inférieure à celle d'un ordinateur moderne. Il est préférable, donc d'installer un système optimisé pour la Raspberry. Raspbian a été créé dans cette optique, et il est donc tout particulièrement adapté à la Raspberry. Il est donc la distribution la plus utilisée pour les Raspberry.

IV.4.2. Debian Wheezy :

Debian (/de.bjan/) est une organisation communautaire et démocratique, dont le but est le développement de systèmes d'exploitation basés exclusivement sur des logiciels libres.

Chaque système, lui-même nommé Debian, réunit autour d'un noyau de système d'exploitation de nombreux éléments pouvant être développés indépendamment les uns des autres, pour plusieurs architectures matérielles. Ces éléments, programmes de base complétant le noyau et logiciels applicatifs, se présentent sous forme de « paquets » qui peuvent être installés en fonction des besoins (voir Distribution des logiciels). L'ensemble système d'exploitation plus logiciels s'appelle une distribution.

On assimile généralement ces systèmes d'exploitation au système **Debian GNU/Linux**, la distribution GNU/Linux de Debian, car jusqu'en 2009 c'était la seule branche parfaitement fonctionnelle. Mais d'autres distributions Debian sont en cours de développement en 2013 : Debian GNU/Hurd et Debian GNU/kFreeBSD. La version Debian « Squeeze » est la première à être distribuée avec le noyau kFreeBSD en plus du noyau Linux.

Debian est utilisée comme base de nombreuses autres distributions telles que Knoppix et Ubuntu .

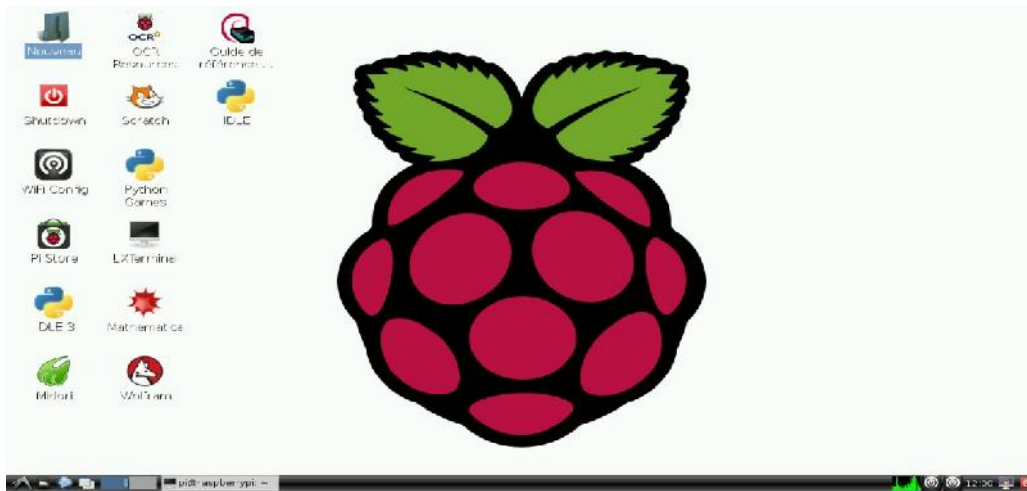


Figure 4.19 : Aperçu de l'interface Raspbian Wheezy

IV.4.3. Le terminal de la Raspberry pi 2

L'**ouverture du terminal** permet de saisir des commandes, en cliquant sur l'icône LXTerminal qui est sur le bureau ou dans la catégorie << Accessoires >> dans le menu des programmes . (Voir figure ci-dessous)



Figure 4.20 : Illustration du terminal de la Raspberry PI 2

IV.4.4. Les commandes de la Raspberry

Les commandes les plus utilisés sur la Raspberry Pi 2 :

- **man <nom de la commande>** : permet de consulter le manuel associé à la commande passée en paramètre (man = manual).
- **whereis <nom de la commande>** : permet de connaître quel script est exécuté lors de l'appel à la commande.

Les commandes du système

- **top** : permet de lister les processus du système.
- **kill** : permet de terminer un processus.
- **df** : permet de connaître l'espace disponible sur les disques.
- **uname** : permet d'afficher des informations importantes du système (version du système...).
- **su** : permet de changer d'utilisateur courant (su = switch user).

Les commandes fichiers et répertoires

- **pwd** : permet de savoir dans quel répertoire on se trouve .
- **mkdir <nom du répertoire>** : permet de créer un nouveau répertoire dans le répertoire courant
- **ls** : permet de lister les fichiers et répertoires contenus dans le répertoire courant.
- **touch <nom du fichier>** : permet de créer un nouveau fichier vide.
- **chmod et chown** : permettent de changer les permissions sur les fichiers et les répertoires.
- **cp <fichier source> <répertoire de destination>** : permet de copier le fichier source et de le coller dans le répertoire de destination (cp = copy).
- **cat <nom du fichier>** : permet de lire le contenu d'un fichier
- **grep <mots à trouver> <fichiers cibles>** : permet de rechercher une chaîne de caractères dans un ou plusieurs fichiers.
- **find** : permet de rechercher des fichiers ou des répertoires selon leur nom.

Les commandes pour le téléchargement

- **wget <adresse de la ressource>** : permet de télécharger la ressource hébergée à l'adresse définie.
- **curl** : permet de télécharger mais également de transmettre un fichier vers un serveur distant.
- **ftp** : permet de se connecter à l'aide du protocole FTP à un serveur distant.

Les commandes pour le réseau

- **ping <ip ou nom de domaine>** : permet de vérifier si la machine cible est bien accessible par le réseau.
- **ifconfig** : permet d'afficher des informations concernant les interfaces réseaux. On l'utilise fréquemment pour trouver son IP.

IV.4.5. Capture d'image avec fswebcam de la Raspberry PI 2

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

La liste de configuration de fswebcam est longue, Il est possible de simplifier le processus en faisant appelle à un script bash à chaque fois qu'il est nécessaire. L'installation est simple :

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

Après que fswebcam est installé , on exécute la commande suivante :

```
fswebcam -r 1280x720 -S 5 --jpeg 95 --font sans:15 --shadow --title "Route63.free.fr" --  
subtitle "WeatherCam @ Lempdes - Puy-De-Dôme - France" --info "Proudly Powered by  
Raspberry Pi" --save home.jpg -q -l 60
```

Chapitre IV: Description des outils logiciels du système

La commande contient des détails pour personnaliser les paramètres de capture avec une webcam, 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
- **-q** : laisse le processus en cours d'exécution en arrière-plan
- **-l** : prend un instantané à chaque interval de temps (en secondes)

et bien d'autres, pour cela *man fswebcam*

La commande étant assez longue on peut utiliser un script de configuration listant tous les paramètres à appliquer lors de l'exécution du programme. Pour cela, il faut créer un fichier de configuration dans le répertoire du user "pi" :

```
nano .fswebcam.conf
```

Le contenu du fichier de configuration est énuméré ci-dessous :

```
resolution 320x240
jpeg 95
delay 1
font sans:15
title "WEB domotique"
subtitle "... "
info " "
timestamp "%d-%m-%Y %H:%M:%S (%Z)"
save /home/pi/test.jpg
exec /home/pi/fswebcam.sh
loop 1
```

fswebcam permet aussi de lancer un script, fswebcam.sh : connexion à mon ordinateur principal par FTP, dépôt du dernier instantané, et cloture de la session FTP.

Pour démarrer fswebcam en utilisant le fichier de configuration :

```
fswebcam -c .fswebcam.conf
```

Pour stopper fswebcam :

```
pkill fswebcam
```

Les images prises sont émises directement vers une page web. Cette dernière est actualisée automatiquement lors de la capture des images par notre webcam, pour la visualisation des images en streaming :

index.html

```
<!doctype html>
<html lang="fr">
  <head>
    <meta http-equiv="Refresh" content="1; url=index.html">
    <title>Web domotique !</title>
    <style>h1, p{ text-align: center}</style>
    <meta charset="UTF-8">
  </head>

  <body>
    <h1>Hello world !</h1>
    <p>Cette page est actualisé toutes les <b>1</b> secondes.</p>
  </body>
</html>
```

IV.4.6. Serveur HTTP ou Serveur WEB

Le serveur Web s'inscrit dans l'architecture informatique dite client-serveur. Son rôle est de " servir " des pages Web sur la demande d'un internaute utilisant un logiciel client Web dont la dénomination usuelle est navigateur Web. Pour cela, il met en œuvre le protocole correspondant, HTTP.

Un ordinateur faisant office de serveur Web ou exécutant, entre autres processus serveurs, un processus serveur Web, doit être relié par une connexion permanente à Internet, avec une adresse IP fixe, de sorte que les sites Web qu'il sert soient disponibles en permanence.

On peut également parler de serveur Web pour désigner un serveur de pages HTML ne fonctionnant que sur un réseau local, sans liaison directe à Internet. Mais si l'on veut être rigoureux, il vaut mieux dans ce cas parler de serveur HTTP.

Les moyennes et grandes entreprises disposent en général de leur propre serveur Web, tandis que les travailleurs indépendants et les particuliers font héberger leurs pages Web . Une solution intermédiaire est également proposée par les Fournisseurs de Services Internet (FSI), notamment à l'intention des PME : l'hébergement d'un ordinateur serveur Web entièrement réservé à l'entreprise.

Un serveur web est donc un « simple » logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole.

IV.4.7. La Raspberry Pi 2 comme serveur web

Plutôt que de faire appel à des prestataires de services spécialisés dans l'hébergement web on utilise la Raspberry Pi comme serveur web.

Tout d'abord, d'un point de vue économique, les services d'hébergement web ne sont pas gratuits et qu'il faut payer chaque mois ou chaque année. Contrairement la Raspberry demande juste une connexion. De plus, en choisissant la Raspberry, il y a la

possibilité de modifier les services d'utilisateurs à souhait (exemples : la taille du disque, l'hébergement de Database, etc.). Ceci n'est en général pas le cas chez les hébergeurs spécialisés, qui vendent souvent des hébergements mutualisés avec une faible capacité de configuration

IV.4.8. Serveur Apache

IV.4.8.1.Définition

Le logiciel libre **Apache HTTP Server (Apache)** est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web.

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc. Néanmoins, que l'existence de nombreux modules Apache rend complexe la configuration du serveur web.

Les possibilités de configuration d'Apache sont une fonctionnalité phare. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. Cette caractéristique est notamment utile aux hébergeurs qui peuvent ainsi servir les sites de plusieurs clients à l'aide d'un seul serveur HTTP. Pour les clients, cette fonctionnalité est rendue visible par le fichier .htaccess.

IV.4.8.2.Installation du serveur Apache avec Raspbian

IV.4.8.2.1.Installation d'Apache

Avant d'installer le serveur, il faut avoir une machine bien à jour. Pour ce faire les droits administrateurs doivent être possédés soit en étant connecté en root, soit via la commande sudo.

```
sudo aptitude update
```

```
sudo aptitude upgrade
```

Une fois la Raspberry Pi à jour, le serveur Apache sera installé :

```
sudo aptitude install apache2
```

Voici les droits au dossier d'apache qui vous permettent d'administrer facilement les sites. Les commandes sont les suivantes :

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

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


IV.4.6.8.2.Vérification du fonctionnement d'Apache

Une fois l'installation terminée, il faut tester le fonctionnement d'Apache en nous rendant sur l'adresse de la Raspberry.

Pour cela, il faut tenter d'accéder à la Raspberry depuis le port 80 (ce port n'étant pas encore ouvert depuis l'extérieur, il faudra le faire depuis la Raspberry elle-même). Il suffit d'ouvrir le navigateur web de la Raspberry (par défaut Midori sous Raspbian), et d'aller à l'adresse « `http://127.0.0.1` ».

On doit alors obtenir une page avec un message du genre « *It works !* » et plein d'autre texte.

Si il n'y a pas d'interface graphique sur la Raspbian, ou qu'on utilise le SSH pour se connecter à la Raspberry, on peut utiliser la commande suivante :

```
wget -O verif_apache.html http://127.0.0.1
```

IV.4.9. Serveur PHP

IV.4.9.1.Introduction

PHP: Hypertext Preprocessor, plus connu sous son sigle **PHP** (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook et Wikipédia, etc. Il est considéré comme la base de la création des sites Internet dits dynamiques, c'est-à-dire que l'utilisateur envoie des informations au serveur qui lui renvoie les résultats modifiés en fonction de ses infos. Par contre un site statique ne s'adapte pas aux informations fournies par un utilisateur. Il est enregistré sous forme de fichier une fois pour toute, et livrera toujours le même contenu.

IV.4.9.2. Installer PHP

On fait appel à l'administrateur pour installer php5 avec la ligne de commande.

```
sudo aptitude install php5
```

IV.4.9.3. Vérifier que PHP fonctionne

Pour savoir si PHP fonctionne correctement, la méthode est relativement proche de celle employée pour Apache.

On supprime le fichier « `index.html` » dans le répertoire « `/var/www/html` ».

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

Puis on crée un fichier « `index.php` » dans ce répertoire, avec cette ligne de commande

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

À partir de là, le fonctionnement est le même que pour la vérification d'Apache. On tente d'accéder à notre page, et on va avoir le résultat de cette image .


PHP Version 5.4.4-14+deb7u7	
	
System	Linux ajaniserveur 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64
Build Date	Dec 12 2013 08:42:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525 NTS

Figure 4.21 : confirmation installation PHP

IV.4.9.4. Sécurisation de notre page web avec le PHP

Afin de garantir la confidentialité et l'intégrité des images envoyées par l'utilisateur et reçues du serveur. On doit sécuriser la page en mettant un nom d'utilisateur et un mot de passe pour valider l'identité du visiteur. Ceci est fait grâce au programme suivant :

```
<?php
if(!isset($_GET['login']) && !isset($_GET['motdepasse']))
{
    header('Location: index.php');
}
else
{
    // On va vérifier les variables
    if(!preg_match('/^[[:alnum:]]+$/ ', $_GET['login']) or
!preg_match('/^[[:alnum:]]+$/ ', $_GET['motdepasse']))
    {
        echo 'Vous devez entrer uniquement des lettres ou des
chiffres <br/>';
        echo '<a href="index.php"
temp_href="index.php">Réessayer</a>';
        exit();
    }
    else
    {
        // require('config.php'); // On réclame le fichier
        $login = $_GET['login'];
        $motdepasse = $_GET['motdepasse'];

        $NU = "XXXXXXXXX";
        $PW = "YYYYYYYYY";
        if(($login == $NU) && ($motdepasse == $PW))
        {
            require('webCAM.php');
            exit();
        }
        else{
            require('index.php');
            exit();
        }
    }
}
?>
```

IV.5. Application Android

IV.5.1.Définition

Android est un système d'exploitation mobile pour Smartphones, tablettes tactile, Smartwatches (version Wear) et terminaux mobiles. C'est un système open source utilisant le noyau Linux. (Logo figure 4.22)

Android de Google Inc. fut développé par une petite startup qui fut acheté par Google qui poursuit activement son développement. Android est distribué sous licence open source, est une variante de Linux. Google a lancé Open Handset Alliance qui regroupe des grands constructeurs et développeurs de logiciels (tel qu'Intel, HTC, ARM, Samsung, Motorola and eBay). Ce système est assez nouveau auprès des programmeurs. Il a eu douze versions, chacune portant un « nom de code » spécifique. En septembre 2014, la part de marché mondiale d'Android est passée à 85 %.



Figure 4.22 : Logo Android

IV.5.2.Les différents autres systèmes d'exploitation mobiles sur le marché :

Il existe sur le marché des dizaines de systèmes d'exploitation différents : Symbian OS de Nokia, ios d'Apple, BlackBerry OS de RIM, Windows Phone de Microsoft, Bada de Samsung et Android de Google...etc

IV.5.2.1.Symbian OS :

Le Symbian OS est développé par la société éponyme qui est une propriété exclusive de Nokia. Bien que cette plateforme soit créée par la participation de plusieurs fabricants tels que Samsung ou Sony Ericsson, ce système est fortement connoté Nokia, ce qui est un frein à son adoption par d'autres constructeurs. Il est récemment passé en open source. C'est un système libre, open source se base sur un noyau Symbian (Voir figure suivante logo symbian).



Figure 4.23 : Logo Symbian OS

IV.5.2.2.IOS

IOS (*Intarnetwork Operating System*), qui était nommé iPhone OS, se trouve non seulement sur les différentes générations de iPhone mais également sur d'autres produits de Apple iPad et iPod touch. Il est dérivé de Mac OS X dont il partage les fondations : kernel, les services Unix et Cocoa. Pour Apple, le succès est considérable : début 2009, il n'y avait pas moins de 5 millions de téléchargements par jour. Donc, il s'agit du concurrent numéro un pour Android. (Voir figure suivante logo IOS)



Figure 4.24 : Logo IOS

IV.5.2.3.BlackBerry OS

Le système d'exploitation BlackBerry (logo figure 4.23) est la plate-forme exclusive mobile développée par RIM (*Research In Motion*) exclusivement pour ses Smartphones BlackBerry et les appareils mobiles. RIM utilise ce système d'exploitation pour soutenir des fonctions spécialisées, notamment le trackball de la marque, molette, le trackpad et l'écran tactile.



Figure 4.25 : Logo BlackBerry

IV.5.2.4.Windows Phone

Windows Mobile, est l'OS (système d'exploitation) mobile de Microsoft. C'est une évolution de Windows Pocket PC, ancêtre de Windows CE. Cet OS a réussi au fil des années à s'octroyer une part de marché honorable. Son succès est dû à son affiliation à la famille d'OS Windows, ultra-dominante sur le bureau. L'autre avantage est la facilité de développement apportée grâce à l'environnement cliquodrome de Visual Studio qui a su faire venir au développement mobile les développeurs VB (*Visual Basic*).



Figure 4.26 :Logo Windows Phone

IV.5.3.Les avantages d'une application mobile :

Une application mobile revêt de nombreux avantages comparés à un site web mobile, en voici 5 principaux :

- Un confort d'usage et une expérience utilisateur inégalée.
- L'accès direct aux contenus de l'application mobile via l'icône présent sur le Dashboard du téléphone ou de la tablette (mode d'accès sans URL).
- Un fonctionnement en mode déconnecté.
- elle permet d'utiliser et d'intégrer toutes les **fonctionnalités téléphone** (accéléromètre, gyroscope, GPS, caméra...), ce qui n'est pas forcément le cas des WebApps.
- L'implémentation de fonctionnalités natives comme par exemple le notifications «PUSH ».

IV.5.4. App Inventor

On a pu créer notre application grâce à App Inventor pour Android qui est une application développée par Google. Elle est actuellement entretenue par le Massachusetts Institute of Technology (MIT). Cette application est disponible sur le site internet www.appinventor.mit.edu.

Elle simplifie le développement des applications sous Android et le rend accessible même pour les novices et ceux qui ne sont pas familiers avec les langages de programmation. Elle est basée sur une interface graphique similaire à Scratch et à celle de StarLogo TNG. Grâce à son interface entièrement graphique et à l'absence totale de ligne de code, elle est particulièrement adaptée à l'initiation de la programmation. (Voir figure suivante : l'interface du site App Inventor)

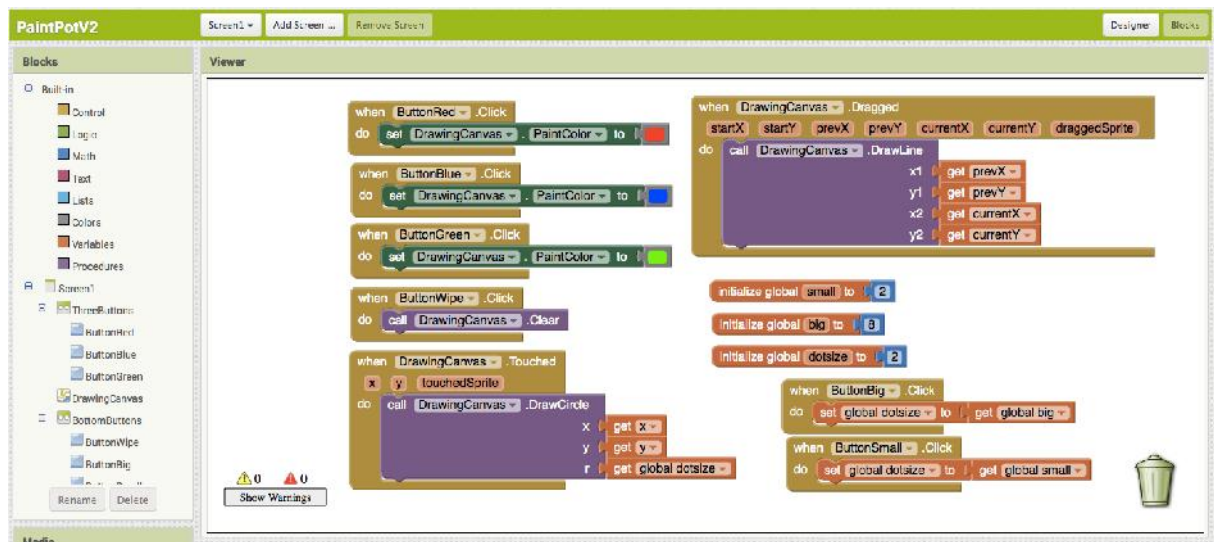


Figure 4. 27 : interface du site App Inventor

IV.5.5. Commande vocale Google

Dans l'application créé la commande vocale Google a pu être intégrée afin de commander le domicile sans avoir à appuyer sur aucun bouton et pour cela le Smartphone doit être connecté à internet. La commande vocale lancée va être traduite par la base de données en ligne de Google. Une fois reconnue la commande est retransmise vers la carte arduino pour être exécutée.

(Voir figure suivante l'interface graphique de l'application Android conçu)



Figure 4. 28 : Interface graphique de notre application Android

IV.6. Conclusion

Dans ce chapitre, on a pu faire connaissance avec les différents langages de programmation et logiciel, à commencer par le logiciel et le langage Arduino avec qui la partie domotique du domicile a pu être mise en œuvre. Ensuite, on a pu se familiariser avec la Raspberry PI 2 et son système d'exploitation qui est le Raspbian et ses commandes principales ainsi que le langage python qui est le langage de programmation de la Raspberry PI 2. De plus, on a installé les différents serveurs pour la mise en place du système. Pour finir une application Android a été créée pour le contrôle entier du domicile via un Smartphone.

Conclusion générale

Conclusion Générale

Après avoir étudié les systèmes domotiques et la vidéosurveillance, un système a été mis en œuvre afin d'optimiser au maximum le contrôle et la sécurité d'un domicile et faciliter la vie au quotidien. De ce fait, la solution proposée est intéressante et peut devenir une nécessité vu l'énorme apport dont on peut bénéficier.

Cette approche de la domotique et de la vidéosurveillance basée sur les nouvelles technologies, offre une grande flexibilité d'utilisation vu la facilité, la fiabilité et la rapidité de contrôle des objets simultanément et cela à travers différents sites distants.

A travers le travail présenté, on a pu mettre en œuvre les acquis relatifs aux systèmes temps réels, la programmation orientée objet et la gestion des réseaux qui constituent les principaux axes d'études de notre formation.

Passant par une conception électronique illustrée, par des modules, éléments, composants électroniques très utilisés dans des projets technologiques destinés aux différents domaines de l'industrie et du développement durable, ont été intégrés au système. De nouvelles versions des langages orientés objets, comme le Python, l'environnement de développement du système qui est le Raspbian wheezy, conçu pour le produit Raspberry PI, les différents serveurs Web et la programmation Arduino ont permis la programmation et le contrôle du projet développé.

Cette réalisation répond aux exigences d'une maison moderne et intelligente, elle se caractérise par sa facilité d'utilisation et son aspect sécuritaire.

Cette conception est basée sur deux cartes de développement numériques destinées à la programmation et aux tests, qui peuvent être remplacés par des éléments matériels plus adaptés au monde extérieur. Comme perspective, une alimentation du domicile par panneaux solaires est envisageable ainsi qu'un contrôle plus global et plus approfondie de la maison.

Enfin, espérant que ce travail servira comme documentation de référence pour les promotions à venir.

Annexe A : Détails sur la carte Arduino Méga

I. Alimentation de la carte Arduino

La carte Arduino Méga peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) soit à l'aide d'une alimentation externe. L'alimentation externe (non-USB) peut être soit adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus).

L'adaptateur secteur peut être connecté en branchant une prise de 2.1mm, dédié du pôle positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelée GND (masse ou 0V) et Vin (tension positive en entrée) du connecteur d'alimentation. La carte peut aussi fonctionner avec une alimentation externe de 6 à 20V. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si plus de 12V sont utilisés, le régulateur de tension pourrait chauffer et endommager la carte. Ainsi, la plage idéale recommandée pour alimenter la carte Méga est entre 7V et 12V.

Il est à noter qu'il est strictement dangereux d'utiliser une alimentation externe via la prise jack et avoir le câble USB connecté.

Les broches d'alimentation sont les suivantes :

- **VIN** : La carte peut être alimentée à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, on peut accéder à la tension d'alimentation sur cette broche. (à distinguer du 5V de la connexion USB ou autre source 5V régulée).
- **5V** : un régulateur de tension est intégré dans la carte permettant de faire fonctionner le microcontrôleur et les autres composants de la carte. Cette broche fournit 5v soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB, qui elle aussi fournit une tension 5V régulé.
- **3V3** : une alimentation de 3.3V est disponible. Cette tension est fournie par un régulateur de 3.3V intégré à la carte Arduino. Ceci est très intéressant pour les circuits externes utilisant cette tension au lieu de 5V, avec une intensité maximale allant jusqu'à 50mA.

II. Protection du port USB contre la surcharge en intensité

L'un des avantages de la carte Arduino Méga est qu'elle possède un poly-fusible remplaçable en cas de surcharge en intensité, ce qui protège le port USB de l'ordinateur sachant que les ports USB sont généralement limité à 500mA en intensité. Si plus de 500mA

Annexe A : Détail sur la carte Arduino Méga

sont appliqués au port USB, le fusible de la carte coupe automatiquement la connexion et le court-circuit ou la surcharge est stoppé immédiatement.

Généralement la plupart des ordinateurs ont leur propre protection interne. Cependant, le fusible de la carte sert comme second niveau de protection pour le port.

III. Gestion des mémoires dans la carte Arduino

L'ATMega 2560 a 256ko de mémoire FLASH pour stocker le programme dont 8ko sont utilisés par le bootloader. Ce microcontrôleur la possède d'une part de 8ko de mémoire SRAM appelé mémoire volatile, et d'autre part de 4ko de mémoire d'EEPROM connu sous le nom de mémoire non volatile. Cette dernière est accessible en lecture grâce à la librairie EEPROM.

Remarque: Le bootloader est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

III.4.6. Les entrées / sorties numériques

les 54 broches numériques de la carte Arduino Méga, fonctionnant en 5V, peuvent être utilisées soit comme entrées, ou comme sorties numériques, en utilisant les instructions `pinMode()`, `digitalWrite()`, et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne « résistance de rappel » (*pull-up*) (déconnectée par défaut) de 20-50 kOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

Il y a entre ces broches celles qui ont des fonctionnalités en plus :

➤ Communication Série:

il y a 4 ports séries pour la carte Arduino Méga qui est l'un de ces principaux avantages

- * Port Série numéro 0: broche 0 (RX) et 1 (TX).
- * Port Série numéro 1: broche 19 (RX) et 18 (TX).
- * Port Série numéro 2: broche 17 (RX) and 16 (TX).
- * Port Série numéro 3: broche 15 (RX) and 14 (TX).

La communication série utilise pour recevoir la broche RX et pour la broche TX pour transmettre les données séries de niveau TTL. Les broches (RX) et (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB/série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

La communication série est appelée grâce à l'instruction `Serial.print()` du langage Arduino.

- **Interruptions Externes:** les Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). Ces broches

Annexe A : Détail sur la carte Arduino Méga

peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.

- **Impulsion PWM (largeur d'impulsion modulée):** les Broches allant de 0 à 13 fournissent une impulsion PWM de 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI (Interface Série Périphérique):** Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Uno, Duemilanove et Diecimila.
- **Le port I2C:** les Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C`.
- **LED:** Une LED est incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsqu'elle est au niveau BAS, elle est éteinte.

IV. Les entrées analogiques

La carte Mega2560 ADK dispose de 16 entrées analogiques, chacune pouvant fournir une mesure de résolution à 10 bits. C'est à dire qu'il ya 1024 niveaux, soit de 0 à 1023. on accède a ces entrées à l'aide de la fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023). Il est aussi possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

Remarque : il est a notées que les broches analogiques peuvent être utilisées en tant que broches numériques.

V. Les autres entrées

Il y a deux autres broches disponibles sur la carte :

- **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V), utilisée avec l'instruction `analogReference()`. Elle est utilisée pour comparer la valeur d'une tension d'entrée par rapport à la valeur d'une tension de référence choisie.
- **Reset** : Mettre cette broche au niveau BAS entraine la réinitialisation du microcontrôleur. Cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte. (voir figure suivante des différentes composantes de la carte Arduino méga)

Annexe A : Détail sur la carte Arduino Méga

VI. Dimension de la carte Arduino méga

La Méga dispose de 101.52 mm en longueur et de 53.3 mm en largeur avec un poids de 36g. Avec le connecteur USB et le connecteur d'alimentation jack, elle s'étend au-delà des dimensions de la carte. Quatre trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier afin de l'embarquer sur un système. (Voir la figure suivante) :

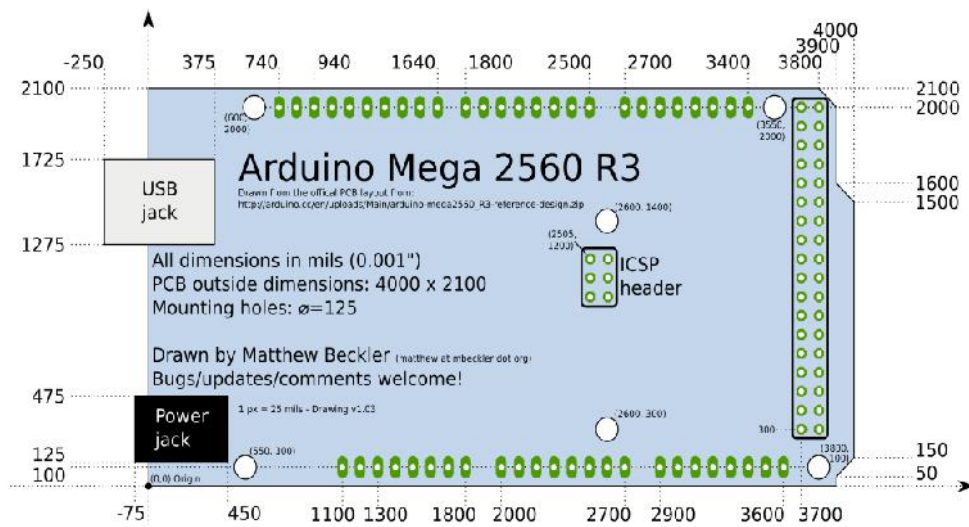


Figure 1: Les dimensions de l'Arduino méga

Programme développé de la carte Arduino Méga

```
#include <LiquidCrystal.h>

//      RS  E  D4  D5  D6  D7
LiquidCrystal lcd(45,44, 43, 42, 41, 40);

#####

#####

#include <Servo.h>

Servo myservo;

#include <Keypad.h>

const byte ROWS = 4; //nombre de lignes
const byte COLS = 4; //nombre de colonnes

char keys[ROWS][COLS] = {

  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}

};

byte rowPins[ROWS] = {22, 23, 24, 25}; //entrées numériques où sont branchées les lignes
byte colPins[COLS] = {26, 27, 28, 29}; //entrées numériques où sont branchées les colonnes


Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

#####

#####

int ACTIVE_ALARME=0;

char KEY1='1',KEY2='2',KEY3='3',KEY4='4';

int cmpMP=0;

int LED_SYS_A_ON=46,LED_SYS_A_OFF=47,LED_SYS_G=12,LED_SYS_D=11;
```

Annexe B : Programme Arduino

```
int Bipper=10;
int TrigG=8,EchoG=9,TrigD=2,EchoD=3;
int ALARM_ON=0;
int ValeurG=0,ValeurD=0;

#####

String chaine;

int LED_SALON = 34, LED_CHAMBRE = 35;

#####
#####
#####

void setup() {
  lcd.begin(20, 4);
  Serial.begin(9600);

  pinMode(LED_SYS_A_ON,OUTPUT);pinMode(LED_SYS_A_OFF,OUTPUT);pinMode(LED_SYS_G,OUTPUT);pinMode(LED_SYS_D,OUTPUT);
  pinMode(9,OUTPUT);

  pinMode(TrigD, OUTPUT);
  pinMode(EchoD, INPUT);
  pinMode(TrigG, OUTPUT);
  pinMode(EchoG, INPUT);

  pinMode(LED_SALON,OUTPUT);
  pinMode(LED_CHAMBRE,OUTPUT);

  myservo.attach(5);
}

#####
```


Annexe B : Programme Arduino

```
//#####  
void loop() {  
  if(ACTIVE_ALARME==0){  
    moteur(90);  
    digitalWrite(LED_SYS_A_ON,LOW);digitalWrite(LED_SYS_A_OFF,HIGH);  
    digitalWrite(LED_SYS_G,LOW);digitalWrite(LED_SYS_D,LOW);  
    alarmeOFF();  
  }  
  if(ACTIVE_ALARME==1){  
    moteur(90);  
    digitalWrite(LED_SYS_A_ON,HIGH);digitalWrite(LED_SYS_A_OFF,LOW);  
    alarmeON();  
  }  
}  
  
//#####  
//#####  
void alarmeOFF(){  
  debut:  
  ALARM_ON=0;  
  lcd.clear();  
  lcd.setCursor(0,0);lcd.print("#####");  
  if(ACTIVE_ALARME==1){  
    lcd.setCursor(0,1);lcd.print("# ALARME ON  #");  
  }else{  
    lcd.setCursor(0,1);lcd.print("# ALARME OFF  #");  
  }  
  lcd.setCursor(0,2);lcd.print("# Mot de passe:  #");  
  lcd.setCursor(0,3);lcd.print("#####");  
  do{
```

```
char key = keypad.getKey();

if ((key != NO_KEY)&&(key != '#')&&(cmpMP < 4)){
  cmpMP++;bip(Bipper,900,50);
  lcd.setCursor(14+cmpMP,2);lcd.print('*'); //changé 'key' par '*'
  switch(cmpMP){
    case 1: KEY1 = key;break;
    case 2: KEY2 = key;break;
    case 3: KEY3 = key;break;
    case 4: KEY4 = key;break;
  }
}

if (key == '*'){
  cmpMP=0;
  lcd.setCursor(0,2);lcd.print("# Mot de passe:  #");
}

if ((cmpMP >= 4)&&(key == '#')){
  if((KEY1=='1')&&(KEY2=='2')&&(KEY3=='3')&&(KEY4=='4')){
    ACTIVE_ALARME=1;
    lcd.clear();
    lcd.setCursor(0,0);lcd.print("#####");
    lcd.setCursor(0,1);lcd.print("# MOT de PASSE #");
    lcd.setCursor(0,2);lcd.print("# CORRECTE #");
    lcd.setCursor(0,3);lcd.print("#####");
    delay(500);
    bip(Bipper,880,100);bip(Bipper,880,100);
```

```
    cmpMP=0;
  }else{
    lcd.clear();
    lcd.setCursor(0,0);lcd.print("#####");
    lcd.setCursor(0,1);lcd.print("#  MOT DE PASSE  #");
    lcd.setCursor(0,2);lcd.print("#  INCORRECTE  #");
    lcd.setCursor(0,3);lcd.print("#####");
    delay (1000);
    bip(Bipper,1000,200);bip(Bipper,1000,200);bip(Bipper,1000,200);

    cmpMP=0;
  }

}

//
//PARTIE
DOMOTIQUE//#####
#####
DOMOTIQUE();
//
}while (ACTIVE_ALARME==0);
}

//#####
//#####

void alarmeON(){
  debut:
  lcd.clear();
  lcd.setCursor(0,0);lcd.print("#####");
  if(ACTIVE_ALARME==1){
```

Annexe B : Programme Arduino

```
lcd.setCursor(0,1);lcd.print("# ALARME ON  #");
}else{
lcd.setCursor(0,1);lcd.print("# ALARME OFF  #");
}

lcd.setCursor(0,2);lcd.print("# Mot de passe:  #");
lcd.setCursor(0,3);lcd.print("#####");
do{

char key = keypad.getKey();

if ((key != NO_KEY)&&(key != '#')&&(cmpMP < 4)){
cmpMP++;bip(Bipper,900,50);
lcd.setCursor(14+cmpMP,2);lcd.print('*'); //changé 'key' par '*'
switch(cmpMP){
case 1: KEY1 = key;break;
case 2: KEY2 = key;break;
case 3: KEY3 = key;break;
case 4: KEY4 = key;break;
}
}

if (key == '*'){
cmpMP=0;
lcd.setCursor(0,2);lcd.print("# Mot de passe:  #");
}

if ((cmpMP >= 4)&&(key == '#')){
if((KEY1=='1')&&(KEY2=='2')&&(KEY3=='3')&&(KEY4=='4')){
ACTIVE_ALARME=0;
```

Annexe B : Programme Arduino

```
    lcd.clear();

    lcd.setCursor(0,0);lcd.print("#####");

    lcd.setCursor(0,1);lcd.print("#  MOT de PASSE  #");

    lcd.setCursor(0,2);lcd.print("#  CORRECTE  #");

    lcd.setCursor(0,3);lcd.print("#####");

    delay(500);

    bip(Bipper,880,100);bip(Bipper,880,100);


    cmpMP=0;

    }else{

        lcd.clear();

        lcd.setCursor(0,0);lcd.print("#####");

        lcd.setCursor(0,1);lcd.print("#  MOT DE PASSE  #");

        lcd.setCursor(0,2);lcd.print("#  INCORRECTE  #");

        lcd.setCursor(0,3);lcd.print("#####");

        delay(1000);

        bip(Bipper,1000,200);bip(Bipper,1000,200);bip(Bipper,1000,200);


    cmpMP=0;

    goto debut;

    }

}

#####

SYSTEM_CONTROLE();

#####

}while (ACTIVE_ALARME==1);

}

#####
```

Annexe B : Programme Arduino

```
//#####  
void bip(int PIN,int FRQ,int TIME){  
  pinMode(PIN,OUTPUT);  
  tone(PIN,FRQ);  
  delay(TIME);  
  noTone(PIN);  
  delay(50);  
}  
  
//#####  
//#####  
int CapteurG(){  
  digitalWrite(TrigG,LOW);  
  delayMicroseconds(5);  
  digitalWrite(TrigG, HIGH);  
  delayMicroseconds(10);  
  
  long temps = pulseIn(EchoG, HIGH);  
  long distance = int(0.017 * temps);  
  
  return distance;  
}  
  
//#####  
//#####  
int CapteurD(){  
  digitalWrite(TrigD,LOW);  
  delayMicroseconds(5);  
  digitalWrite(TrigD, HIGH);  
  delayMicroseconds(10);
```

Annexe B : Programme Arduino

```
long temps = pulseIn(EchoD, HIGH);

long distance = int(0.017 * temps);

return distance;
}

#####
#####

void SYSTEM_CONTROLE(){
if(ALARM_ON==0){
    ValeurG = CapteurG();
    ValeurD = CapteurD();
}

int DistanceOBSTACLE = 20;// 20cm

if(((ValeurD > 5)&&(ValeurD < DistanceOBSTACLE))||((ValeurG > 5)&&(ValeurG <
DistanceOBSTACLE))){

//if((ValeurD > 5)&&(ValeurD < DistanceOBSTACLE)){

if((ValeurD > 5)&&(ValeurD < DistanceOBSTACLE)){

    moteur(45);

    digitalWrite(LED_SYS_D,HIGH);

}

if((ValeurG > 5)&&(ValeurG < DistanceOBSTACLE)){

    moteur(135);

    digitalWrite(LED_SYS_G,HIGH);

}

ALARM_ON=1;

}

if(ALARM_ON==1){

bip(Bipper,1500,100);
```

```
}  
  
}  
  
#####  
#####  
  
void moteur(int degre){  
  int frq= map(degre,0,180,500,2400);  
  myservo.writeMicroseconds(frq);  
}  
  
#####  
  
#####  
  
void DOMOTIQUE()  
{  
  int N=0;  
  while(Serial.available())  
  {  
    delay(15);  
    char c = Serial.read();  
    chaine = chaine+c;  
  }  
  
  if (chaine.length() >0){  
    Serial.println(chaine);  
    if(chaine == "LED_SALON_ON") N = 1 ;  
    if(chaine == "LED_SALON_OFF") N = 2 ;  
    if(chaine == "LED_CHAMBRE_ON") N = 3 ;  
    if(chaine == "LED_CHAMBRE_OFF") N = 4 ;
```



```
chaîne = "" ;
```

```
}
```

```
switch(N)
```

```
{
```

```
case 1: SALON_ON(); break;
```

```
case 2: SALON_OFF(); break;
```

```
case 3: CHAMBRE_ON(); break;
```

```
case 4: CHAMBRE_OFF(); break;
```

```
}
```

```
}
```

```
//#####
```

```
void SALON_ON()
```

```
{
```

```
digitalWrite(LED_SALON,HIGH);
```

```
}
```

```
void SALON_OFF()
```

```
{
```

```
digitalWrite(LED_SALON,LOW);
```

```
}
```

```
//#####
```

```
void CHAMBRE_ON()
```

```
{
```

```
digitalWrite(LED_CHAMBRE,HIGH);
```

```
}
```

```
void CHAMBRE_OFF()
```

```
{
```

```
digitalWrite(LED_CHAMBRE,LOW);
```

```
}
```

Bibliographie

Bibliographie

- [1] Méziane Boudellal, "Smart home Habitat connecté, installation domotique et multimedia" Edition Dunod , 2014.
- [2] George Asch, " Les capteurs en instrumentation industrielle" Edition Dunod, 2010.
- [3] Cédric Locqueneux , "Le guide de la maison et des objets connectés " Edition Eyrolles, 2016.
- [4] Lacène Beddiaf, "Vidéosurveillance : Principes et technologies" Edition Dunod, 2008.
- [5] Phillipe bellaiche, "les secrets de l'image video" 10ème Edition Eyrolles, 2015.
- [6] <<Guide technique, technique de la vidéo sur IP>> page 21 disponible sur le site :
www.axis.com
- [7] Philipe Gasser, "MPEG-4", créative commons, cours, MSH, Paris nord, plateforme Arts, Sciences et Technologies, 2005.
- [8] Sylvain Caicoya , Jean-Georges Saury, "TCP/IP - Le guide complet", Edition Micro Application, 2013
- [9] Nico Van Haute, Julien Barascud et Jean-Roland Conca, <<Les protocoles RTP et RTCP>>, document disponible sur : www.commentcamarche.net.
- [10] <<Guide technique, technique de la vidéo sur IP>> disponible sur le site :
www.axis.com
- [11] Eskimon & olyte, "Arduino pour bien commencer en électronique et en programmation", 2012 openclassrooms.
- [12] Eskimon & olyte , "Arduino : Premiers pas en informatique embarquée", Le blog d' Eskimon ,Edition 19 Juin 2014
- [13] Matt Richardson "A la découverte de la Raspberry PI", édition Eyrolles, Serial maker, 2014
- [14] Gérard Swinnen " Apprendre à programmer avec PYTHON".

Sites Web :

- [1] www.arduino.cc
- [2] www.abcelectronique.com
- [4] www.fr.wikipedia.org
- [5] www.fr.openclassrooms.com
- [6] www.developpez.net
- [7] www.sourceforge.net
- [8] www.github.com
- [9] www.microsoft.com/france/visual-studio.com
- [10] www.fritzing.org
- [11] www.mon-club-elec.fr
- [12] www.raspbian-france.fr
- [13] www.the-raspberry.com
- [14] www.pihomeserver.fr
- [15] www.framboise314.fr
- [16] www.axis.com
- [17] www.securite01.com
- [18] www.rince.fr
- [19] www.commentcamarche.net
- [20] www.appinventor.mit.edu
- [21] www.insctructable.com