

الجمهورية الجزائرية الديمقراطية الشعبية

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

وزارة التعليم العالي والبحث العلمي

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



## **MEMOIRE**

Présenté pour l'obtention du **diplôme** de **MASTER**

**En :** Télécommunications

**Spécialité :** Réseaux Mobiles et Services de Télécommunications

**Par :**

**Mr: BOUHARAOUA Abderrahim**

**Mr: BOUKLI HACENE Mohammed Imad**

**Sujet**

***Automatisation d'une maison intelligente  
via une application Android***

Soutenu publiquement, le 11 / 06 / 2017, devant le jury composé de :

M.KADRI benamar	MCA	Univ. Tlemcen	Président
M. HADJILA Mourad	MCA.	Univ. Tlemcen	Directeur de mémoire
M.MERZOUGUI Rachid	MCA	Univ. Tlemcen	Examineur

*Je voudrais dédier de tout mon cœur ce modeste travail :  
A mes très chers parents en signe de ma profonde et affectueuse reconnaissance pour  
leur amour sans mesure, tous les sacrifices, les soutiens, les tolérances et les  
encouragements qu'ils ont bien voulu consentir pour moi. Tous les mots restent faibles  
pour leur exprimer mes sentiments et qu'ils acceptent seulement ces lignes en guise de  
témoignage.*

*A mon binôme Boukli Hacene Mohammed Imad.  
A ma tante qui a toujours cru en moi et m'a soutenue dans les meilleurs et les pires  
moments, je lui exprime mon infinie reconnaissance.  
A mon petit cher frère Ali, mes sœur Nour El Houda, Ibtissem, Nihad et son mari Tayeb.  
A toute les familles Bouharaoua et Bensabeur.  
A mes amis Yassine, Mohammed, Lotfi, Amine le pompier, Mehdi.  
A mes collègues de la classe RMST.  
Enfin je dédie ce mémoire à tous ceux qui m'aiment et que j'aime.*

**Abderrahim**

*Je dédie ce modeste travail et ma profonde gratitude à ma mère et mon père  
pour l'éducation qu'ils m'ont prodigué ; avec tous les moyens et au prix de toutes les  
sacrifices qu'ils ont consentis à mon égard,  
pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance.*

*A mon binôme Bouharaoua Abderrahim  
A mes frères et ma sœur et leurs enfants  
et à qui je leur souhaite beaucoup de réussite dans leur vie ;  
A ma tante et Mon cousin Moncef qui m'ont soutenue dans les meilleurs et les pires  
moments.*

*Ma gratitude la plus profonde à mes amis et frères présents,  
A tout le corps enseignant, administratif, et le personnel de la faculté de technologie.  
A tous mes collègues du M2 RMST 2016/2017.*

**Mohammed Imad**

# REMERCIEMENTS

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

Au terme de ce travail, on tient à remercier particulièrement Mr Hdjila Morad, notre encadreur notre enseignant Maitre de conférence à l'université Abou-bekr-Belkaid de Tlemcen pour son encadrement, Sa disponibilité, ses critiques et ses remarques pertinentes. Il nous a conseillé et guidé du début du projet à sa fin.

On exprime notre profonde reconnaissance et nos sincères remerciements également à Mr. S.Boukli Hacene Maitre de conférence et directeur de Laboratoire de recherche de département d'informatique à l'université de Sidi Belabbes on aimerait lui adresser nos plus vifs remerciements pour ses conseils avisés ,son aide pour la rédaction de ce mémoire par éditeur de texte Latex et surtout pour réaliser notre maison intelligente avec la carte Raspberry Pi.

On voudrait aussi adresser nos sincères remerciements à tous les enseignants de la spécialité RMST Pour leurs enseignements et les cours intéressants qu'ils nous ont donné pendant notre scolarité à l'Université Abou Bekr Belkaid.

Et finalement on tient à remercier chaleureusement et respectivement tous ceux qu'ont contribués de près ou de loin à la réalisation de ce modeste projet de fin d'étude de licence.

Tlemcen, le 01 Mai 2017.

# TABLE DES MATIÈRES

REMERCIEMENTS	iii
TABLE DES MATIÈRES	iv
LISTE DES FIGURES	ix
LISTE DES TABLEAUX	xii
LISTE DES ABRÉVIATION	xiii
INTRODUCTION GÉNÉRALE	1
<b>I INTRODUCTION À L'INTERNET DES OBJETS</b>	<b>3</b>
I.1 HISTORIQUE DE L'INTERNET DES OBJETS . . . . .	5
I.2 QU'EST-CE QU'UN OBJET ? . . . . .	5
I.3 TYPE D'OBJET . . . . .	5
I.3.1 Objet passif . . . . .	6
I.3.2 Objet actif . . . . .	6
I.4 QU'EST-CE QUE L'INTERNET DES OBJETS ? . . . . .	6
I.5 OBJECTIF D'INTERNET DES OBJETS . . . . .	6
I.6 PROBLÉMATIQUES POSÉES PAR L'INTERNET DES OBJETS . . . . .	7
I.7 APPLICATION DE L'INTERNET DES OBJETS . . . . .	7
I.7.1 Ville intelligente . . . . .	8
I.7.2 Les Réseaux Intelligents . . . . .	9
I.7.3 Les environnements intelligents . . . . .	10
I.7.4 La Médecine Intelligente (Smart eHealth) . . . . .	10
I.7.5 Agriculture Intelligente . . . . .	11
I.7.6 La domotique . . . . .	11
I.7.7 Industrie Intelligente . . . . .	12
I.8 SÉCURITÉ ET VIE PRIVÉE . . . . .	12
I.9 VISIONS DE L'INTERNET DES OBJETS . . . . .	13
I.9.1 La vision orientée objet . . . . .	14
I.9.2 La vision orientée Internet . . . . .	14
I.9.3 La vision orientée sémantique . . . . .	15
I.10 CONCLUSION . . . . .	15
<b>II LA DOMOTIQUE</b>	<b>16</b>
II.1 DÉFINITION . . . . .	17
II.2 PRINCIPE DE FONCTIONNEMENT . . . . .	17
II.3 LE MARCHÉ DE LA DOMOTIQUE . . . . .	17
II.4 LES FONCTIONS DE LA DOMOTIQUE . . . . .	18
II.4.1 La fonction de sécurité . . . . .	19
II.4.2 La fonction de surveillance . . . . .	19
II.4.3 La fonction de gestion d'énergie . . . . .	20

II.4.4	La fonction de scénarisation . . . . .	20
II.4.5	La fonction de communication . . . . .	20
II.4.6	La fonction de confort . . . . .	21
II.5	LES TECHNOLOGIES UTILISÉES POUR LA DOMOTIQUE . . . . .	21
II.5.1	La technologie bus filaire . . . . .	21
II.5.1.1	EIB (European Installation Bus) /Konnex . . . . .	22
II.5.1.2	Lonworks . . . . .	22
II.5.2	Le courant porteur en ligne (CPL) . . . . .	22
II.5.2.1	Courant porteur domestique . . . . .	22
II.5.2.2	Courant porteur informatique . . . . .	23
II.5.2.3	Courant porteur audiovisuel . . . . .	23
II.5.3	La radiofréquence . . . . .	23
II.5.4	La mixité des supports . . . . .	24
II.6	MIXITÉ ENTRE LA TECHNOLOGIE RADIO ET FILAIRE . . . . .	24
II.7	LES MATÉRIELS DOMOTIQUES . . . . .	25
II.8	CONCLUSION . . . . .	26
III	PRÉSENTATION DE RASPBERRY PI . . . . .	27
III.1	CHOIX DE LA SOLUTION . . . . .	28
III.1.1	Carte Arduino . . . . .	28
III.1.2	Carte STM32 . . . . .	28
III.1.3	Raspberry Pi . . . . .	29
III.1.4	Beaglebone . . . . .	29
III.1.5	Analyse du tableau comparatif . . . . .	30
III.2	RASPBERRY PI . . . . .	30
III.2.1	Présentation du Raspberry . . . . .	30
III.2.2	Les modèles de Raspberry Pi . . . . .	30
III.2.2.1	Modèles A . . . . .	30
III.2.2.2	Modèles B . . . . .	31
III.2.2.3	Modèle 0 . . . . .	32
III.2.3	Raspberry Pi Type B + . . . . .	33
III.2.4	Les composants de Raspberry Pi . . . . .	33
III.2.4.1	Processeur ARM . . . . .	33
III.2.4.2	Mémoire vive RAM . . . . .	34
III.2.4.3	Une connectique variée . . . . .	34
III.2.4.4	Système d'exploitation . . . . .	35
III.2.4.5	Boîtier . . . . .	36
III.2.4.6	Adaptateur Wi-Fi . . . . .	36
III.2.5	Connexion de l'ensemble des éléments . . . . .	37
III.2.6	Les différentes utilisations du Raspberry Pi . . . . .	37
III.2.6.1	Les serveurs . . . . .	37
III.2.6.2	L'utilisation MultiMedia . . . . .	38
III.2.6.3	La domotique . . . . .	38
III.2.6.4	Les systèmes embarqués . . . . .	38
III.2.6.5	Un ordinateur, tout simplement . . . . .	38
III.2.7	Préparation du Raspberry Pi . . . . .	38
III.2.7.1	Installation de l'OS « Raspbian » . . . . .	38
III.2.7.2	Démarrage du système . . . . .	39
III.2.7.3	Configurer le Pi . . . . .	40
III.2.7.4	Éteindre le Raspberry Pi . . . . .	41
III.2.7.5	Connexion à distance . . . . .	41
III.2.7.6	Réservation d'adresse IP . . . . .	43

III.2.8	Programmer pour Raspberry Pi . . . . .	43
III.2.8.1	Présentation du python . . . . .	44
III.2.8.2	Programmation des entrées / sorties avec python . . . . .	44
III.3	CONCLUSION . . . . .	45
<b>IV</b>	<b>RÉALISATION DE LA MAISON INTELLIGENTE</b>	<b>46</b>
IV.1	COMPOSANTS NÉCESSAIRES DE RÉALISATION . . . . .	47
IV.2	LE RASPBERRY PI COMME SERVEUR WEB, POURQUOI ? . . . . .	47
IV.2.1	Installation du serveur Apache . . . . .	47
IV.2.1.1	Apache, c'est quoi ? : . . . . .	47
IV.2.1.2	Installation d'Apache : . . . . .	48
IV.2.2	Installation de PHP sur le Raspberry . . . . .	48
IV.2.2.1	Qu'est-ce que PHP ? . . . . .	48
IV.2.2.2	Installer PHP . . . . .	49
IV.2.2.3	PHP contrôle les GPIOs . . . . .	49
IV.2.3	Une base de données MySQL pour notre serveur . . . . .	49
IV.2.3.1	Un SGBD c'est quoi ? Pourquoi MySQL ? . . . . .	49
IV.2.3.2	Installer MySQL . . . . .	50
IV.2.3.3	Gérer facilement les bases de données avec PHPMyAdmin . . . . .	50
IV.3	CONFIGURER LES COMPOSANTS AVEC RASPBERRY PI . . . . .	51
IV.3.1	Les LEDs . . . . .	51
IV.3.1.1	LED rouge . . . . .	51
IV.3.1.2	LED vert . . . . .	52
IV.3.1.3	LED jaune . . . . .	53
IV.3.1.4	Page php (led.php) de contrôle des leds à distance (WiFi) . . . . .	53
IV.3.2	Les moteurs . . . . .	54
IV.3.2.1	Ventilateur . . . . .	55
IV.3.2.2	La porte . . . . .	57
IV.3.3	Le capteur de température . . . . .	60
IV.3.3.1	La configuration du 1-wire . . . . .	60
IV.3.4	Le détecteur de gaz . . . . .	62
IV.3.4.1	Script python MQ-2.py : . . . . .	62
IV.3.4.2	Script php : . . . . .	63
IV.3.5	la vidéo-surveillance avec Raspberry Pi : . . . . .	63
IV.3.5.1	Définition : . . . . .	63
IV.3.5.2	Installation de motion sur le Raspberry Pi : . . . . .	63
IV.3.5.3	Configurer Motion sur notre Raspberry Pi : . . . . .	63
IV.4	CONCLUSION . . . . .	65
<b>V</b>	<b>CONTRÔLE DE LA MAISON INTELLIGENTE</b>	<b>66</b>
V.1	COMMUNICATION CLIENT/SERVEUR . . . . .	68
V.1.1	Création de la BDD . . . . .	68
V.1.2	Création de la table admin . . . . .	68
V.1.3	Création de la table user . . . . .	68
V.2	CONCEPTION D'APPLICATION . . . . .	69
V.2.1	Projet de l'application . . . . .	69
V.2.2	Présentation des interfaces graphique et classes java . . . . .	70
V.2.2.1	la classe Smarthouse.java . . . . .	70
V.2.2.2	la classe MD5hash.java . . . . .	70
V.2.2.3	la classe Connect.java . . . . .	70
V.2.2.4	la classe HttpDo.java . . . . .	71

V.2.2.5	La classe Identification.java	72
V.2.2.6	La classe Accueil.java	73
V.2.2.7	La classe ControlHouse.java	74
V.2.2.8	La classe Register.java	76
V.3	CONCEPTION DU SITE	78
V.3.1	Script de configuration (db.php)	79
V.3.2	Page de démarrage du site (index.php)	79
V.3.2.1	Vue	79
V.3.2.2	Code source	79
V.3.3	Page d'accueil (accueil.php)	80
V.3.3.1	Vue	80
V.3.3.2	Code source	80
V.3.4	Page d'affichage de température (sensor.php)	81
V.3.4.1	Vue	81
V.3.4.2	Code source	81
V.3.5	Page de streaming vidéo	82
V.3.5.1	Vue	82
V.3.5.2	Code source	82
V.3.6	Page de contrôle de porte (door.php)	82
V.3.6.1	Vue	82
V.3.6.2	Code source	83
V.3.7	Page de contrôle des LEDs (light.php)	83
V.3.7.1	Vue	83
V.3.7.2	Code source	84
V.3.8	Page de contrôle du ventilateur (fan.php)	84
V.3.8.1	Vue	84
V.3.8.2	Code source	85
V.4	PRÉSENTATION D'ANDROID	85
V.4.1	La création d'Android	85
V.4.2	Les avantages de Google Android	86
V.4.3	Les inconvénients de Google Android	86
V.4.4	Versions d'Android	86
V.4.4.1	Versions disparues	86
V.4.4.2	Versions existantes	86
V.5	PHP ET MySQL	87
V.5.1	Page de configuration (db_config.php)	87
V.5.2	Script d'authentification des utilisateurs (authentification.php)	87
V.6	CONCEPTION DE NOTRE APPLICATION ANDROID	88
V.6.1	Répertoire complet de l'application	88
V.6.2	Fichier AndroidManifest.xml	88
V.6.3	Interface authentification	89
V.6.3.1	Vue	89
V.6.3.2	Classe MainActivity.java	89
V.6.3.3	Explication du code java	90
V.6.4	Interface accueil (welcome.xml)	91
V.6.5	Interface de contrôle (dashboard_layout.xml)	91
V.6.6	Interface capteur	92
V.6.6.1	Vue (sensor.xml)	92
V.6.6.2	Classe Sensor.java	93
V.6.6.3	Explication du code java	93
V.6.7	Interface camera (camera.xml)	93
V.6.7.1	Vue (camera.xml)	94

V.6.7.2	Explication du code java . . . . .	94
V.6.8	Interface portail . . . . .	94
V.6.8.1	Vue (door.xml) . . . . .	94
V.6.8.2	Classe Door.java . . . . .	95
V.6.8.3	Explication du code java . . . . .	95
V.6.9	Interface light . . . . .	95
V.6.9.1	Vue (Light.xml) . . . . .	95
V.6.9.2	Classe light.java . . . . .	95
V.6.9.3	Explication du code java . . . . .	96
V.6.10	Interface fan . . . . .	97
V.6.10.1	Vue (fan.xml ) . . . . .	97
V.6.10.2	Classe fan.java . . . . .	97
V.6.10.3	Explication du code java . . . . .	98
V.7	CONCLUSION . . . . .	98
	CONCLUSION	<b>99</b>
	BIBLIOGRAPHIE	<b>100</b>
	ANNEXE	<b>101</b>
	RÉSUMÉ	<b>103</b>
	ABSTRACT	<b>103</b>



# LISTE DES FIGURES

I.1	Quelques exemples sur les objets connectés . . . . .	6
I.2	Prévision de nombre d'objets connectés vers 2020 . . . . .	6
I.3	Domaines d'applications de l'Internet des Objets . . . . .	8
I.4	Les Systèmes de Transport Intelligents (ITS) . . . . .	9
I.5	Smart Grids . . . . .	9
I.6	Système de Détection des Incendies . . . . .	10
I.7	Médecine Intelligente . . . . .	10
I.8	Capteur de température et de la vitesse du vent et taux d'humidité . . . . .	11
I.9	Smart Home . . . . .	11
I.10	Usine Intelligente . . . . .	12
I.11	Vision de l'Internet des Objets . . . . .	14
II.1	Circulation d'information en domotique . . . . .	17
II.2	Gamme de vente d'équipements domotiques . . . . .	18
II.3	Différentes fonctions de la domotique . . . . .	19
II.4	Exemple de la technologie Bus . . . . .	22
II.5	Exemple d'une installation CPL . . . . .	23
II.6	Installation réseaux sans fil en domotique . . . . .	24
II.7	Mixité entre la technologie radio et filaire . . . . .	24
II.8	Quelques équipements utilisés en domotique . . . . .	25
III.1	Carte Arduino UNO . . . . .	28
III.2	Carte STM32 . . . . .	28
III.3	Carte Raspberry Pi . . . . .	29
III.4	Carte Beaglebone . . . . .	29
III.5	Modèle A . . . . .	31
III.6	Modèle A+ . . . . .	31
III.7	Modèle B1 . . . . .	31
III.8	Modèle B1+ . . . . .	32
III.9	Modèle B2 . . . . .	32
III.10	Modèle B3 . . . . .	32
III.11	Modèle 0 . . . . .	33
III.12	Les composants standard d'un Raspberry . . . . .	33
III.13	Bloc d'alimentation USB. . . . .	34
III.14	Le port GPIO . . . . .	35
III.15	Logo du système d'exploitation Raspbian et Windows 10 IoT. . . . .	36
III.16	Boitier de Raspberry . . . . .	36
III.17	Adaptateur Wi-Fi . . . . .	37
III.18	Système Raspberry Pi . . . . .	37
III.19	Installation du Raspbian avec Win32 Disk Imager . . . . .	39
III.20	Le menu principal de l'outil raspi-config . . . . .	39
III.21	Écran de démarrage Raspberry . . . . .	41
III.22	Installation xrdp réussite . . . . .	42

III.23	Outil de connexion à distance . . . . .	42
III.24	Fenêtre d'authentification . . . . .	42
III.25	Bureau Raspberry . . . . .	43
III.26	Réservation de l'adresse 192.168.0.100 au Raspberry . . . . .	43
III.27	Slogan de langage Python . . . . .	44
IV.1	Logo d'Apache . . . . .	47
IV.2	Logo PHP . . . . .	49
IV.3	Logo MYSQL . . . . .	50
IV.4	Branchement LED rouge . . . . .	51
IV.5	Branchement LED vert . . . . .	52
IV.6	Circuit LEDs avec Raspberry . . . . .	54
IV.7	Pont H . . . . .	54
IV.8	Brochage du Pont H . . . . .	55
IV.9	Ventilateur . . . . .	56
IV.10	Circuit ventilateur avec Raspberry . . . . .	56
IV.11	Moteur pas à pas . . . . .	58
IV.12	Circuit moteur pas à pas avec Raspberry . . . . .	58
IV.13	Câblage de la sonde DS18B20 avec Raspberry . . . . .	60
IV.14	Capteur MQ-2 . . . . .	62
IV.15	Teste de camera de surveillance . . . . .	64
V.1	Architecture client serveur . . . . .	68
V.2	Projet Smarthouse . . . . .	69
V.3	Interface authentification . . . . .	72
V.4	Interface accueil . . . . .	73
V.5	Quit item . . . . .	73
V.6	Fenêtre information . . . . .	74
V.7	Interface de contrôle . . . . .	74
V.8	Interface utilisateur . . . . .	76
V.9	Interface recherche . . . . .	77
V.10	Répertoire du site . . . . .	78
V.11	La page index . . . . .	79
V.12	Page d'accueil . . . . .	80
V.13	Page d'affichage de température . . . . .	81
V.14	Page de streaming vidéo . . . . .	82
V.15	Page de contrôle de porte . . . . .	82
V.16	Page de contrôle des LEDs . . . . .	83
V.17	Page de contrôle du ventilateur . . . . .	84
V.18	Android . . . . .	85
V.19	Communication Android PHP MySQL . . . . .	87
V.20	Répertoire de l'application SmartHouse . . . . .	88
V.21	Vue activity_main.xml . . . . .	89
V.22	Vue Accueil . . . . .	91
V.23	Vue page de control . . . . .	92
V.24	Vue page capteurs . . . . .	92
V.25	Vue page de lancement de camera . . . . .	94
V.26	Vue page de control du portail . . . . .	94
V.27	Vue page de control des LEDs . . . . .	95
V.28	Vue page de control de ventilateur . . . . .	97
V.29	La fiche technique de Raspberry Pi . . . . .	101
V.30	La fiche technique de la carte mère . . . . .	101
V.31	Vue 1 . . . . .	102

V.32	Vue 2	.....	102
V.33	Vue 3	.....	102

# LISTE DES TABLEAUX

I.1	Résumé de l'impact de l'échelle sur l'Internet des objets . . . . .	7
I.2	Résumé des problèmes de sécurité et de vie privée dans l'Internet des objets	12
III.1	Comparaison technique : Arduino Uno, Beaglebone, Raspberry Pi etSTM32	30

# LISTE DES ABRÉVIATION

<i>BDD</i>	Base Des Données
<i>CPL</i>	Courant porteur en Ligne
<i>DC</i>	Direct Courant
<i>DECT</i>	Digital Enhanced Cordless Telecommunication
<i>EIB</i>	European Installation Bus
<i>EJP</i>	Effacement des Jours de Pointe
<i>FTP</i>	File Transfert Protocole
<i>GPIO</i>	General purpose Input/Output
<i>HD</i>	High Dification
<i>HDMI</i>	High Dification Multimedia Interface
<i>HTTP</i>	HyperText Transfert Protocol
<i>IDE</i>	Integrated Development Environment
<i>IdO</i>	Internet des Objets
<i>IOS</i>	Iphone OS
<i>IoT</i>	Internet of Things
<i>IP</i>	Internet Protocol
<i>ITS</i>	Itelligent Trasportation System
<i>JDBC</i>	Java DataBase Connectivity
<i>LCD</i>	Liquid Crystal Display
<i>LED</i>	Light-Emetting Diod
<i>M2M</i>	Machine to Machine
<i>MAC</i>	Media Access Control
<i>MYSQL</i>	My Structured Query Language
<i>NFC</i>	Near Flied Communication
<i>OS</i>	Oparating System
<i>OSI</i>	Open System Interconnexion
<i>PDO</i>	PHP Data Object
<i>PHP</i>	Hypertext Preprocessor
<i>PME</i>	Petit et Moyen Entreprise
<i>PWM</i>	Pulse Width Modulation

<i>RAM</i>	Random Access Memory
<i>RCA</i>	Radio Corporation of America
<i>RFID</i>	Radio Frequency IDentification
<i>SGBD</i>	Système de Gestion de Base Données
<i>SSH</i>	Serveur Secure Shell
<i>STI</i>	Système de transport Intelligent
<i>TIC</i>	Technologie de Information et de Communication
<i>USB</i>	Universal Serial Bus
<i>WiFi</i>	wireless Fidelity
<i>WSAN</i>	Wireless Sensor and Actuator Network
<i>X3D</i>	Extensible 3D

=====

# INTRODUCTION GÉNÉRALE

LES progrès continus des technologies de la microélectronique et des réseaux de capteurs permettent maintenant d'envisager le déploiement de services sécurisés et optimisés distribués sur des réseaux d'objets communicants intelligents interconnectés : c'est la vision de l'Internet des Objets. Nous assistons actuellement au déploiement d'une nouvelle génération d'objets interconnectés dotés de capacités de communication de détection et d'activation (réseaux de transport d'information sans fil, RFID, WSN, etc.) pour de nombreuses applications. Ainsi l'interconnexion d'objets dotés de capacités avancées de traitement va conduire à une révolution en termes de création et de disponibilité de service et va profondément changer notre façon d'agir sur notre environnement. En bref le monde physique fusionne progressivement avec le monde virtuel.

L'Internet des Objets, ou Internet of Things (IoT) en anglais, transformera l'ensemble de la société, y compris nous-mêmes. À première vue, cette affirmation peut paraître exagérée, mais pensez à l'impact qu'a déjà eu Internet sur l'enseignement, les communications, les entreprises, la science, les organismes publics et les hommes. Internet est sans nul doute l'une des inventions les plus importantes et les plus significatives de toute l'histoire de l'humanité.

Dites-vous maintenant que l'IoT représente la prochaine évolution d'Internet et permettra d'améliorer considérablement sa capacité à rassembler, analyser et restituer des données que nous pourrons ensuite transformer en informations, en connaissances et enfin en savoir. Dans ce contexte, l'importance de l'IoT paraît évidente.

Des projets IoT déjà en cours promettent de combler les écarts de richesse, d'améliorer la distribution des ressources mondiales aux populations défavorisées et de nous aider à comprendre notre planète, ce qui nous permettra d'adopter un comportement plus proactif au lieu de simplement réagir aux événements, parmi ces projets on trouve un domaine très intéressant celui de la domotique qui rend la vie humaine plus simple et confortable.

La gestion intelligente de l'habitat, et plus généralement du bâtiment, est au centre de plusieurs débats : environnement, sécurité, confort. Il existe aujourd'hui des architectes qui proposent une approche écodomotique, où l'écologie se marie à la technologie. Les constructeurs rivalisent d'innovations pour concilier bien-être et performance dans votre quotidien.

Si, par ses divers moyens de contrôle et de régulation, la domotique est bien une des clés de l'optimisation énergétique de votre domicile, elle en possède aussi beaucoup d'autres. Avec une installation domotique, vous pourrez aujourd'hui avoir une maison vivante et économe. Rendre votre maison intelligente vous assurera un résultat de basse consommation évident. Votre habitat sera protégé, car la domotique veillera à la sécurité de votre lieu de vie.

Elle nous offre aussi un bien-être sur-mesure, avec un confort en permanence. La solution domotique pour les personnes âgées est, par exemple, une extension d'avenir de ce type d'installation.

C'est par cette idée qu'on a pensé de faire l'étude de notre projet de fin d'étude sur un des domaines de l'internet des objets c'est la domotique là où nous avons décidé de

réaliser un prototype d'une maison intelligente avec la fameuse carte Raspberry Pi qui ne cesse de donner des solutions facile pour le domaine de la domotique.

### **ORGANISATION DU DOCUMENT :**

Notre mémoire est réparti essentiellement en cinq chapitres :

Le premier chapitre est réservé pour une introduction à l'internet des objets. Nous donnons un petit aperçu sur la définition de l'objet connecté, les réseaux d'internet des objets, les différentes applications pour ces réseaux.

Dans le deuxième chapitre, nous allons aborder le domaine de la domotique qui fait l'objet de notre projet de fin d'études là où nous allons voir qu'est ce que c'est cette science, ses technologies, ses services qui rendent la vie humaine révolutionnaire.

Dans le troisième chapitre, nous allons parler de notre solution pour installer un système domotique utilisant la fameuse carte électronique Raspberry Pi en faisant une présentation exhaustive sur ses composants, ses domaines d'utilisation et sa préparation. Nous allons faire aussi un petit aperçu sur le langage python pour contrôler les broches de cette carte.

Dans le quatrième chapitre, nous allons voir en détails la réalisation électronique de notre maison intelligente accompagnée de la programmation des objets reliés à notre carte Raspberry à savoir les LEDs, les moteurs, les capteurs et aussi la webcam.

Finalement, le dernier chapitre est divisé en 3 parties. La première partie fera l'objet de détails et explications des différentes étapes de conception et d'implémentation de l'application desktop qui est gérée par l'administrateur pour créer des comptes utilisateurs. La deuxième partie est consacré àù contrôle de notre Smarthouse avec le web là où on explique le développement de notre site web avec PHP5 . Et finalement dans la troisième partie, nous allons voir comment contrôler notre maison intelligente à travers une application Android qui constitue le but de notre travail.



# INTRODUCTION À L'INTERNET DES OBJETS

# I

## SOMMAIRE

I.1	HISTORIQUE DE L'INTERNET DES OBJETS . . . . .	5
I.2	QU'EST-CE QU'UN OBJET ? . . . . .	5
I.3	TYPE D'OBJET . . . . .	5
I.3.1	Objet passif . . . . .	6
I.3.2	Objet actif . . . . .	6
I.4	QU'EST-CE QUE L'INTERNET DES OBJETS ? . . . . .	6
I.5	OBJECTIF D'INTERNET DES OBJETS . . . . .	6
I.6	PROBLÉMATIQUES POSÉES PAR L'INTERNET DES OBJETS . . . . .	7
I.7	APPLICATION DE L'INTERNET DES OBJETS . . . . .	7
I.7.1	Ville intelligente . . . . .	8
I.7.2	Les Réseaux Intelligents . . . . .	9
I.7.3	Les environnements intelligents . . . . .	10
I.7.4	La Médecine Intelligente (Smart eHealth) . . . . .	10
I.7.5	Agriculture Intelligente . . . . .	11
I.7.6	La domotique . . . . .	11
I.7.7	Industrie Intelligente . . . . .	12
I.8	SÉCURITÉ ET VIE PRIVÉE . . . . .	12
I.9	VISIONS DE L'INTERNET DES OBJETS . . . . .	13
I.9.1	La vision orientée objet . . . . .	14
I.9.2	La vision orientée Internet . . . . .	14
I.9.3	La vision orientée sémantique . . . . .	15
I.10	CONCLUSION . . . . .	15

L'INTERNET des objets (ou IdO) se traduit à l'heure actuelle par l'accroissement du nombre d'objets connectés, c'est-à-dire d'appareils possédant une identité propre et des capacités de calcul et de communication de plus en plus sophistiquées : téléphones, montres, appareils ménagers, etc. Ces objets embarquent un nombre grandissant de capteurs et d'actionneurs leur permettant de mesurer l'environnement et d'agir sur celui-ci, faisant ainsi le lien entre le monde physique et le monde virtuel.

Dans les années à venir, votre voiture, votre bureau, votre domicile et tous les appareils électroménagers qu'ils contiennent, y compris les compteurs d'électricité, de gaz et d'eau, l'éclairage public, le système d'arrosage automatique, les pèse-personnes, les tensiomètres et même les murs seront connectés à l'IdO.

Dans ce chapitre, on va aborder en détails le monde de l'Internet des objets, sa vision, son architecture avec les différentes applications associées ainsi on va parler des capteurs et actionneurs qui font une pièce primordiale dans les réseaux d'Internet des Objets.

## I.1 HISTORIQUE DE L'INTERNET DES OBJETS

L'Internet des Objets est une concrétisation technique de l'informatique ubiquitaire où la technologie est intégrée naturellement aux objets du quotidien. Très prometteur, ce concept ouvre la voie vers une multitude de scénarios basés sur l'interconnexion entre le monde physique et le monde virtuel : domotique, e-santé, ville intelligente, logistique, sécurité, etc. Cependant, comme d'autres concepts prometteurs, celui-ci fait face à un certain nombre de problématiques techniques et non techniques qui nécessitent d'être étudiées pour permettre à l'Internet des Objets d'atteindre son plein potentiel.

L'Internet des Objets est un concept concrétisant la vision de l'informatique ubiquitaire telle qu'imaginée en 1991 par Mark Weiser, où la technologie deviendra efficace peu à peu dans l'environnement des utilisateurs, intégrée naturellement à l'intérieur des objets du quotidien. La technologie n'est plus alors représentée par un objet unique, l'ordinateur personnel, mais se présente au contraire sous la forme d'appareils spécialisés et simples d'emploi, capables de communiquer au travers de plusieurs types de réseaux sans fil : liseuses numériques, télévisions et montres connectées, ordinateurs de bord, téléphones intelligents, etc.

À l'origine, le terme Internet des objets a été utilisé pour la première fois en 1999 par Kevin Ashton pour décrire des objets équipés de puces d'identification par radiofréquence (ou puce RFID). Chaque objet, identifié de manière unique et universelle, peut alors être rattaché à un ensemble d'informations le concernant, ces dernières étant lisibles par d'autres machines. Le concept a toutefois évolué avec le temps et s'est généralisé vers une approche consistant à connecter un très grand nombre d'objets du quotidien au réseau Internet, les dotant ainsi d'une identité propre et leur permettant, entre autres, d'offrir des services et de collecter des informations de manière autonome [Billet 2015].

## I.2 QU'EST-CE QU'UN OBJET ?

Un objet est, avant toute chose, une entité physique ; par exemple, un livre, une voiture, une machine à café électrique ou un téléphone mobile. Un objet connecté est un matériel électronique qui peut communiquer avec un Smartphone, une tablette tactile et/ou un ordinateur, ses caractéristiques pouvant évoluer au cours du temps (position, niveau de batterie, etc.). Il peut envoyer et recevoir des informations, via une liaison sans fil, Bluetooth ou WiFi, etc.

Autres définitions s'accordent à dire qu'un objet possède des capacités de calcul, d'acquisition (capteur) et d'action (actionneur). Cependant, cette définition exclut les objets inertes identifiés par RFID. En effet, une puce RFID classique ne peut pas être considérée comme un dispositif de calcul, celle-ci se résumant à un identifiant stocké dans une mémoire. De même, si l'on considère les cas extrêmes, un simple code-barres peut être employé pour identifier un objet, ce dernier étant exempté d'une quelconque partie électronique.

L'intérêt principal d'un objet connecté est l'interactivité, la possibilité de récupérer des informations, ou d'envoyer des statistiques, de garder le contact, etc.

## I.3 TYPE D'OBJET

Il est raisonnable de considérer que l'Internet des objets est composé d'objets actifs, capables d'accomplir des calculs, d'effectuer des mesures sur l'environnement ou d'influer sur celui-ci, et d'objets passifs qui n'ont pas d'autres aptitudes que celles d'être suivis et détectés par des objets actifs [Billet 2015].



FIGURE I.1 – Quelques exemples sur les objets connectés

### I.3.1 Objet passif

Par extension, l'identité d'un **objet passif** n'est pas directement stockée dans celui-ci, à l'exception de l'identifiant, et nécessite l'utilisation d'une infrastructure tierce capable de stocker ces informations [Billet 2015].

### I.3.2 Objet actif

Au contraire, un **objet actif** peut stocker tout ou partie de son identité et échanger directement ces informations avec d'autres objets actifs [Billet 2015].

## I.4 QU'EST-CE QUE L'INTERNET DES OBJETS ?

Un ensemble d'objets physiques disposant de capteurs et possédant une adresse IP qui se connectent en reçoivent et envoient des données via un réseau de communication. L'IoT s'inscrit dans une migration qui s'opère alors que le monde évolue des communications M2M (Machine-to-Machine) de l'IoT vers l'Internet of Things et, probablement vers « l'Internet of Everything ».

Cette évolution est induite par trois moteurs :

- la prolifération des objets connectés.
- la croissance exponentielle des applications multiplateformes.
- la maturité des technologies réseau capables de connecter des milliards de dispositifs, de manière économique et sans efforts.

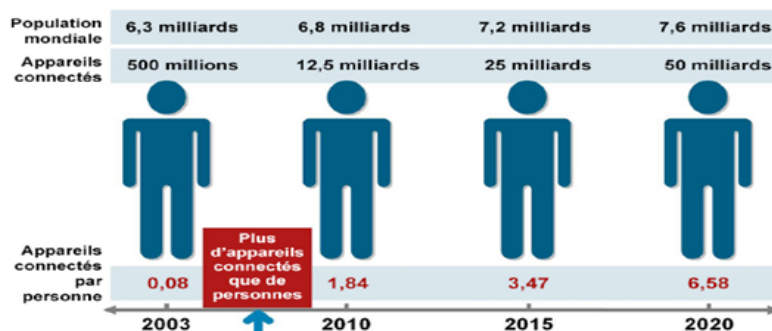


FIGURE I.2 – Prévision de nombre d'objets connectés vers 2020

## I.5 OBJECTIF D'INTERNET DES OBJETS

L'objectif ambitieux derrière l'Internet des Objets d'après le concept d'objet étant posé, les différentes définitions présentées et la littérature que nous avons étudiée à ce

sujet nous conduisent à conclure que l'Internet des objets est une infrastructure globale permettant notamment :

- aux objets actifs d'échanger des informations ou d'en collecter à propos des objets passifs, ce qui rejoint les définitions d'un réseau M2M mondial ;
- de stocker et de rendre accessible l'identité des objets, les informations produites par ces derniers et toutes les connaissances nécessaires pour que les objets gagnent en autonomie, ce au travers de représentations, de structures et de formats de données manipulables par les machines ;
- offrir les abstractions nécessaires aux êtres humains pour interagir avec ces machines, et par extension avec le monde physique, aussi simplement qu'avec le monde virtuel que nous connaissons aujourd'hui.

## I.6 PROBLÉMATIQUES POSÉES PAR L'INTERNET DES OBJETS

Pour permettre à l'Internet des objets d'atteindre son plein potentiel et concrétiser les scénarios qui en découlent, plusieurs aspects doivent être étudiés. En effet, les concepts prometteurs imaginés par la communauté scientifique nécessitent de résoudre un certain nombre de problématiques : grande échelle, hétérogénéité, impact du monde physique (grands flux continus de données, variabilité de l'environnement), sécurité des biens et des personnes, et respect de la vie privée des utilisateurs [Billet 2015].

De toutes les difficultés posées par l'Internet des objets, son échelle mondiale et le très grand nombre d'objets impliqués sont les plus importants, et ce même si l'on exclut.

Adressage et nommage	Le très grand nombre d'objets nécessite (i) un espace d'adressage important et (ii) augmente significativement la quantité d'informations que les serveurs de noms doivent stocker pour assurer leur rôle d'association entre les noms d'objet et leurs adresses.
Découverte	Enregistrer et rechercher des objets par leur nom ou par leurs caractéristiques est fondamental pour la réalisation des scénarios où les objets ne sont pas connus à l'avance. Cependant, stocker et parcourir l'ensemble des objets de manière centralisée n'est pas envisageable à une telle échelle.
Accès	La multitude d'objets différents complexifie le rapatriement des données (data collection), tandis que les grands volumes de données complexifient leur traitement (data mining et data aggregation)

TABLE I.1 – Résumé de l'impact de l'échelle sur l'Internet des objets

## I.7 APPLICATION DE L'INTERNET DES OBJETS

L'Internet des Objets est influencé par les différents scénarios d'utilisation qui sont considérés aujourd'hui par les mondes scientifique et industriel. Quelques exemples courants sont présentés dans la figure I.3 à titre d'illustration.

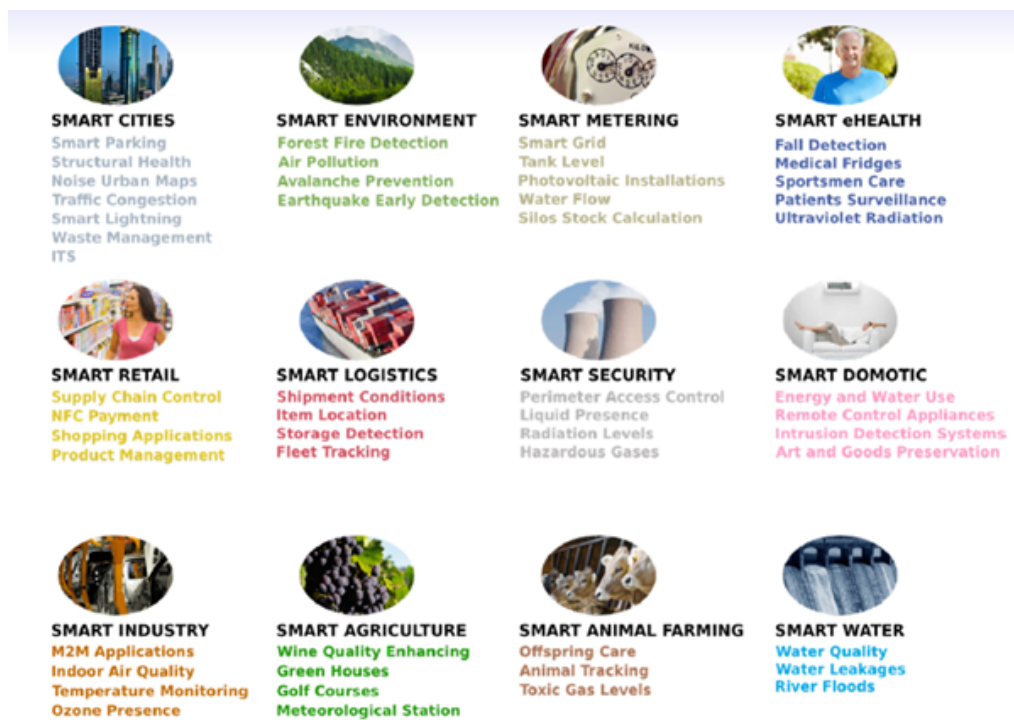


FIGURE I.3 – Domaines d'applications de l'Internet des Objets

Chacun d'entre eux possède ses propres objectifs et ses propres contraintes, d'où l'intérêt de considérer l'Internet des objets non pas sous le prisme d'une application spécifique, mais comme la combinaison des besoins d'une multitude de scénarios. Ces scénarios peuvent prendre place dans des espaces privés, au sein desquels les interactions avec l'extérieur sont strictement contrôlées, ou dans des espaces publics ouverts à tous (par exemple, la détection des places de parking vides), avec toute la gamme de nuances qui peut exister entre les deux.

Parmi les domaines d'application d'Internet des objets, on peut citer :

### I.7.1 Ville intelligente

Une ville intelligente ou encore « smart city » en Anglais, désigne une ville utilisant les technologies de l'information et de la communication (TIC) pour « améliorer » la qualité des services urbains ou encore réduire ses coûts. Ce concept désigne un type de développement urbain apte à répondre à l'évolution ou l'émergence des besoins des institutions, des entreprises et des citoyens, tant sur le plan économique, social, qu'environnemental.

Une ville intelligente serait capable de mettre en oeuvre une gestion des infrastructures (d'eau, d'énergies, d'information et de télécommunications, de transports, de services d'urgence, d'équipements publics, de bâtiments, de gestion et tri des déchets, etc.) communicantes, adaptables, durables et plus efficaces, automatisées pour améliorer la qualité de vie des citoyens, dans le respect de l'environnement.

Un des aspects d'une ville intelligente est les **Systèmes de Transport Intelligents** (STI) (en anglais **Intelligent Transportation System** (ITS)) qui sont les applications des nouvelles technologies de l'information et de la communication au domaine des transports. On les dit "Intelligents" parce que leur développement repose sur des fonctions généralement associées à l'intelligence : capacités sensorielles, mémoire, communication, traitement de l'information et comportement adaptatif. On trouve les STI dans plusieurs champs d'activité : dans l'optimisation de l'utilisation des infrastructures de transport, dans l'amélioration de la sécurité (notamment de la sécurité routière) et de la sûreté ainsi que dans le développement des services [[wikipedia\\_Ville\\_intelligente](https://fr.wikipedia.org/wiki/Ville_intelligente)].



FIGURE I.4 – Les Systèmes de Transport Intelligents (ITS)

### I.7.2 Les Réseaux Intelligents

Les Réseaux Intelligents ou « **Smart Grids** » sont des réseaux d'électricité qui, grâce à des technologies informatiques, ajustent les flux d'électricité entre fournisseurs et consommateurs.

En collectant des informations sur l'état du réseau, les Smart Grids contribuent à une adéquation entre production, distribution et consommation.

Il est nécessaire de différencier Smart Grids et compteur communicant ou « Smart Meter », qui renseigne le consommateur sur sa demande en électricité. « Smart grids » est une appellation générale pour l'ensemble des technologies et des infrastructures « intelligentes » installées. Chez le particulier, le compteur communicant est une première étape dans la mise en place des Smart Grids [[wikipedia\\_Réseau\\_intelligent](http://wikipedia_Réseau_intelligent)].

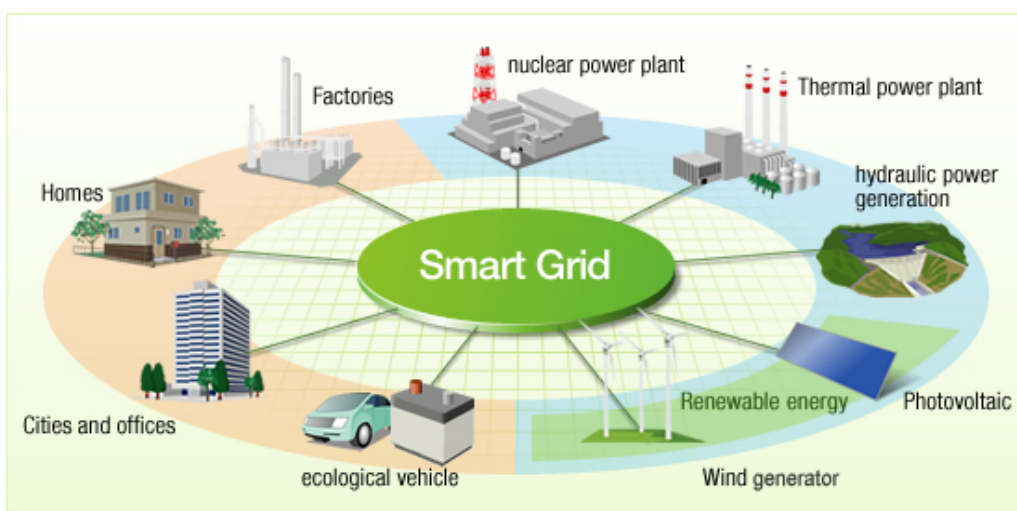


FIGURE I.5 – Smart Grids

Les réseaux intelligents peuvent être définis selon quatre caractéristiques en matière de :

**Flexibilité** : ils permettent de gérer plus finement l'équilibre entre production et consommation ;

**Fiabilité** : ils améliorent l'efficacité et la sécurité des réseaux ;

**Accessibilité** : ils favorisent l'intégration des sources d'énergies renouvelables sur l'ensemble du réseau ;

**Économie** : ils apportent, grâce à une meilleure gestion du système, des économies d'énergie et une diminution des coûts (à la production comme à la consommation).

### I.7.3 Les environnements intelligents

Les environnements intelligents sont des environnements dans lesquels des détecteurs et actionneurs ont été intégrés, pour réagir à des événements et s'adapter aux personnes présentes. Par exemple, une maison intelligente peut adapter la température et l'éclairage en fonction de la santé, de l'humeur et des préférences des personnes et animaux à l'intérieur de chaque pièce.

Un exemple sur ce concept intelligent c'est le **système de détection des incendies de forêts**. Ce système est basé sur des capteurs Libelium, permettant de détecter des incendies de forêt. L'objectif étant de fournir une infrastructure de surveillance de l'environnement, avec une capacité de gestion d'alerte rapide [[wikipedia\\_Système\\_de\\_détection\\_incendie](#)].

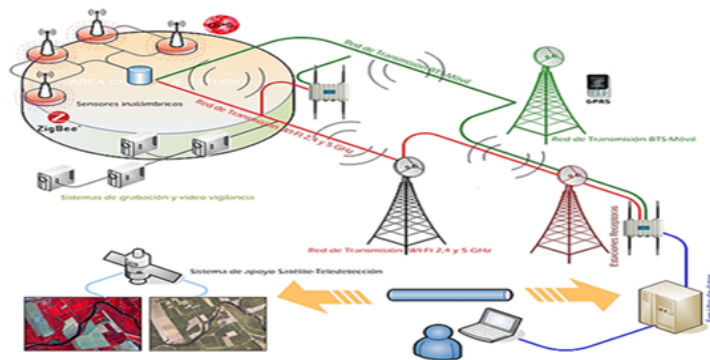


FIGURE I.6 – *Système de Détection des Incendies*

### I.7.4 La Médecine Intelligente (Smart eHealth)

Le domaine médical peut aussi intégrer des applications pertinentes, comme par exemple, l'aide à la médication et le suivi des patients à distance (rythme cardiaque, pression du sang, etc.), l'identification des allergies et des médicaments administrés aux patients, la localisation des docteurs et des patients dans l'hôpital, etc. [[smarthealthit.org](#)].

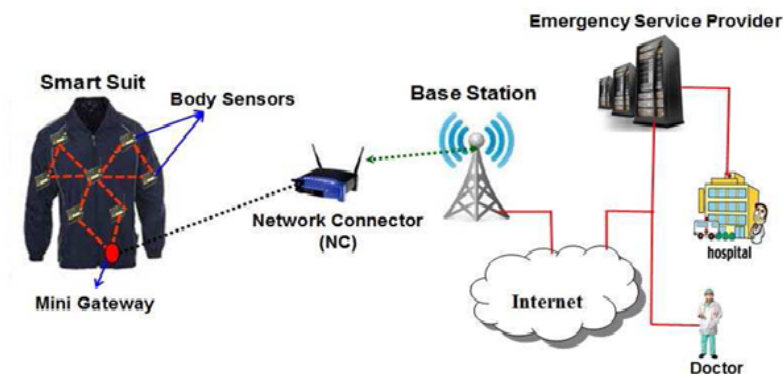


FIGURE I.7 – *Médecine Intelligente*



### I.7.5 Agriculture Intelligente

L'agriculture intelligente a pour objet de renforcer la capacité des systèmes agricoles, de contribuer à la sécurité alimentaire en intégrant le besoin d'adaptation et le potentiel d'atténuation dans les stratégies de développement de l'agriculture durable.

Cet objectif requiert l'utilisation de nouvelles technologies, telles que l'imagerie satellitaire et l'informatique. Il s'appuie sur des moyens de localisation dans la parcelle dont le système de positionnement par satellite de type GPS, il s'appuie aussi sur l'utilisation des capteurs qui vont s'occuper de récolter les informations utiles sur l'état du sol, taux d'humidité, taux des sels minéraux, etc. et envoyer ces informations au fermier pour prendre les mesures nécessaires garantissant la bonne production [[fao.org](http://fao.org)].



FIGURE I.8 – Capteur de température et de la vitesse du vent et taux d'humidité

### I.7.6 La domotique

La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les bâtiments, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc.). La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc [[maison-et-domotique.com](http://maison-et-domotique.com)].



FIGURE I.9 – Smart Home

### I.7.7 Industrie Intelligente

Le concept d'industrie de quatrième génération correspond à une nouvelle façon d'organiser les moyens de production : l'objectif est la mise en place d'usines dites « **intelligentes** » « **smart factories** » capables d'une plus grande adaptabilité dans la production et d'une allocation plus efficace des ressources, ouvrant ainsi la voie à une nouvelle révolution industrielle. Ses bases technologiques sont l'Internet des Objets et les systèmes cyber-physiques.

L'usine intelligente se caractérise par une communication continue et instantanée entre les différents outils et postes de travail intégrés dans les chaînes de production et d'approvisionnement. L'utilisation de capteurs communicants apporte à l'outil de production une capacité d'autodiagnostic et permet ainsi son contrôle à distance tout comme sa meilleure intégration dans le système productif global [[vinci energies.com](http://vinci.energies.com)].

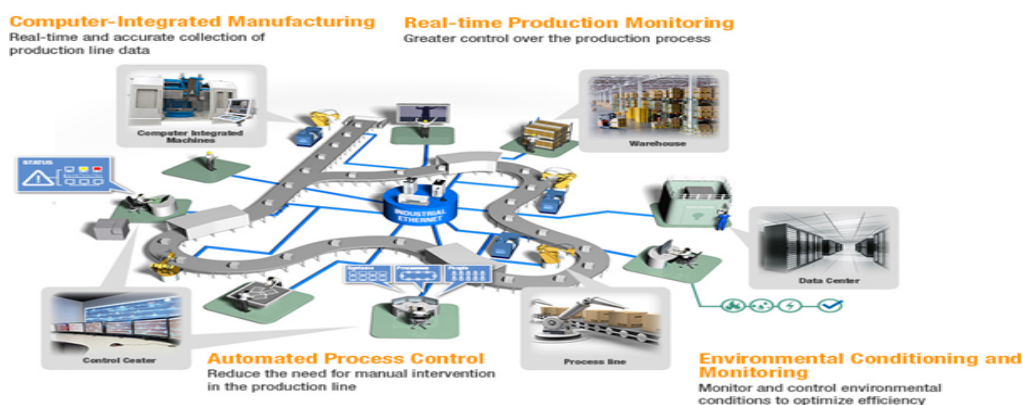


FIGURE I.10 – Usine Intelligente

### I.8 SÉCURITÉ ET VIE PRIVÉE

Si l'Internet des Objets introduit plusieurs cas d'utilisation prometteurs, il pose aussi plusieurs problèmes quant à la sécurité et à la vie privée, comme résumé dans le tableau suivant :

<p>Vulnérabilité</p>	<p>L'impossibilité d'utiliser les technologies modernes de sécurité (chiffrement, authentification, échange de clé, signature, etc.) rend les objets vulnérables aux attaques informatiques ce qui, en raison de leurs capacités à influencer sur le monde physique, représente un danger pour les biens et les personnes.</p>
<p>Surveillance de masse</p>	<p>En s'intégrant naturellement à l'environnement, les objets peuvent mesurer n'importe quelle information sans qu'il soit possible de savoir où, quand et par qui sont collectées ces données. En effet, même en l'absence d'intentions malveillantes de la part des propriétaires d'objets, l'utilisation massive des réseaux sans fil facilite les écoutes clandestines. En fin, l'identification unique des objets permet de dresser des profils très détaillés de leurs propriétaires et de les suivre à grande échelle.</p>

TABLE I.2 – Résumé des problèmes de sécurité et de vie privée dans l'Internet des objets

En effet, les objets étant des entités physiques possédant les capacités nécessaires pour influencer sur leur environnement, les dégâts provoqués par une attaque informatique dans un tel contexte ont un impact bien plus important que ceux causés par les intrusions, les défigurations de site, les vols de données ou les dénis de service que nous connaissons à l'heure actuelle. À titre d'exemple, on citera Stuxnet 8, un ver sophistiqué, découvert en 2010, qui s'attaque spécifiquement aux systèmes de contrôle et d'acquisition de données (supervisory control and data acquisition, ou SCADA) utilisés, entre autres, pour contrôler des infrastructures industrielles (privées, militaires et civiles) de premier plan : centrales nucléaires, oléoducs et gazoducs, aéroports, etc.

Concrètement, l'Internet des Objets souffre de plusieurs vulnérabilités inhérentes aux technologies utilisées. En effet, de nombreux objets possèdent des capacités matérielles réduites, voire quasiment inexistantes, et, de fait, ne peuvent pas directement mettre en oeuvre des techniques de sécurisation modernes, ces dernières nécessitant des ressources de calcul conséquentes.

De la même façon, le coût énergétique élevé des communications sans fil complexifie les processus d'authentification des objets au travers d'infrastructures dédiées, du fait de l'échange de multiples messages entre les appareils et les serveurs d'authentification. L'échange de clés entre les différents objets pose des problèmes similaires, aussi bien en matière de calculs (génération des clés) qu'en matière de communication (échange des clés). Pour finir, la préservation de l'intégrité des données, c'est-à-dire la garantie qu'un adversaire ne puisse pas modifier de données sans que cela soit détecté par le système, est difficile à assurer en l'absence de chiffrement fort. Par ailleurs, les objets sont vulnérables physiquement, étant le plus souvent déployés dans un environnement sans surveillance particulière. Un attaquant pourrait, notamment, altérer leur mémoire en accédant directement à leur matériel ou aux interfaces d'entrée-sortie disponibles. En outre, l'utilisation des technologies de communication sans fil facilite d'une part la mise en place d'écoutes clandestines et, d'autre part, les attaques par déni de service en perturbant le signal radio.

Ainsi, l'Internet des objets pose des problèmes de respect de la vie privée même pour ceux qui n'en utilisent pas les services, car la simple présence d'un individu dans l'environnement entraîne implicitement la collecte d'informations. Couplé à l'identification unique des objets, cela rend possible la construction de profils extrêmement complets à propos des utilisateurs et des objets qu'ils possèdent ou transportent sur eux. Ces profils uniques peuvent ensuite être utilisés pour suivre les individus en temps réel et à très grande échelle, en se basant sur la détection des objets et sur les différents capteurs placés dans l'environnement. Il faut avoir conscience que ce problème de collecte à l'insu des utilisateurs ne peut pas être résolu uniquement avec des mesures techniques.

## I.9 VISIONS DE L'INTERNET DES OBJETS

L'Internet des Objets émerge des travaux menés dans plusieurs domaines de recherche dont les différentes « visions » convergent vers un même but. Dans leur état de l'art, Atzori et coll. identifient trois visions majeures : une vision orientée objet, une vision orientée Internet et une vision orientée sémantique. Comme montré sur la Figure I.11, chaque vision possède ses concepts et ses technologies clés [Billet 2015].

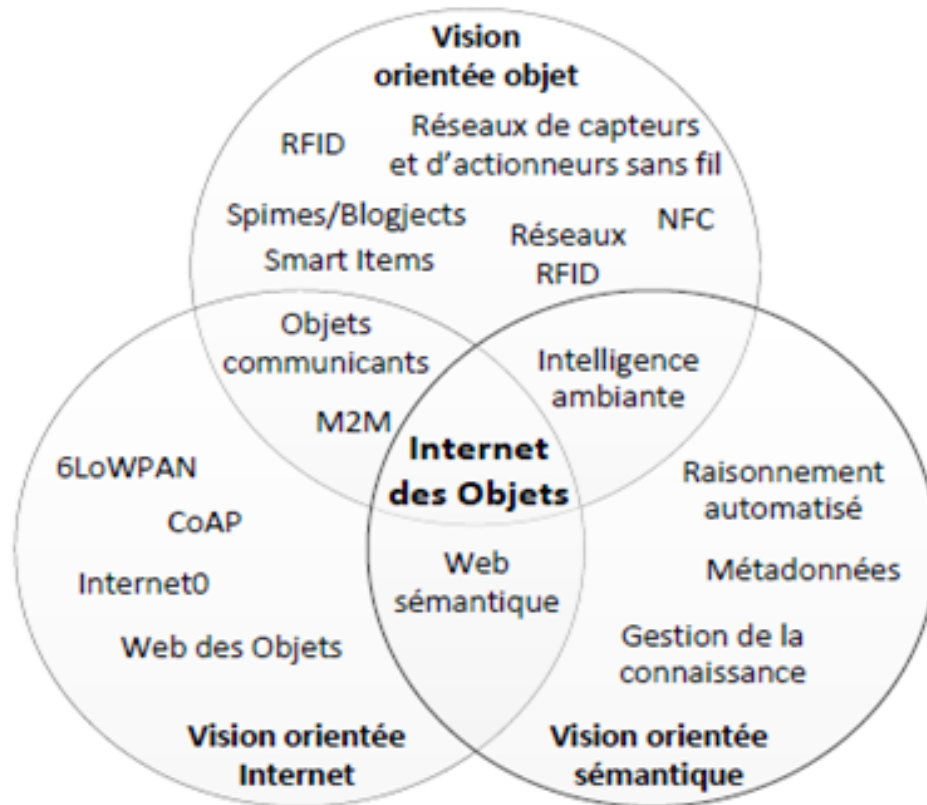


FIGURE I.11 – Vision de l'Internet des Objets

### I.9.1 La vision orientée objet

Cette vision met l'accent sur les objets physiques et les systèmes qui y sont embarqués, avec notamment les problématiques suivantes :

- identifier les objets de manière unique et, de fait, les doter d'une identité propre ;
- permettre aux objets d'acquérir des informations sur leur environnement en utilisant des capteurs, pour ensuite échanger et traiter ces informations afin d'influer sur cet environnement au travers d'actionneurs.

Cette vision est principalement issue des travaux sur les réseaux d'objets identifiés par radiofréquence (RFID), la communication en champ proche (Near Field Communication, ou NFC) et les réseaux de capteurs et d'actionneurs sans fil (Wireless Sensor and Actuator Network, ou WSAN). L'Internet des objets est vu comme le concept émergent de la mise en réseau des différents objets dans le but de faciliter les interactions entre les êtres humains et les objets, ainsi que les interactions entre objets eux-mêmes.

### I.9.2 La vision orientée Internet

Elle suggère que cette interconnexion devrait se faire spécifiquement au travers du réseau Internet, considérant que celui-ci est déjà un « réseau de réseaux » qui connecte un très grand nombre de machines au moyen de protocoles standards conçus pour de nombreux usages. Ainsi, une seule couche réseau, au sens OSI du terme, est utilisée pour connecter tous types d'objets, simplifiant de fait la fabrication et le déploiement de ces derniers, tout en assurant une certaine interopérabilité. Cette vision s'attache donc à étudier comment le protocole IP (Internet Protocol) peut être adapté pour des systèmes embarqués caractérisés par de faibles ressources matérielles, notamment au travers de nouveaux standards. Par extension, si l'on connecte les objets physiques au

réseau Internet, il devient possible de les intégrer aussi au Web pour bénéficier de ses avantages (Web des objets) [Billet 2015].

### I.9.3 La vision orientée sémantique

Elle se détache des problématiques techniques propres aux objets physiques et aux réseaux permettant de les faire communiquer, pour se concentrer sur la représentation, l'organisation et le stockage des données relatives à l'Internet des Objets, notamment :

- les identités de chaque objet physique (caractéristiques, états passés et présents, etc.) et les relations qui peuvent unir ces objets entre eux ;
- les flux d'informations acquis par ces objets sur leur environnement, par exemple au moyen de capteurs [[wikipedia\\_Domotique](#)].

### I.10 CONCLUSION

On a vu dans ce chapitre qu'est-ce que l'Internet des Objets leur objectif ainsi quelques domaines d'application.

On a constaté que le réseau de capteurs sans fil est vulnérable à cause de l'absence d'une infrastructure de sécurité solide. Dans le prochain chapitre, on va parler d'un des domaines de l'**Internet des Objets** qui est la domotique, ses services et les différentes technologies utilisées.

# LA DOMOTIQUE

# II

## SOMMAIRE

II.1	DÉFINITION . . . . .	17
II.2	PRINCIPE DE FONCTIONNEMENT . . . . .	17
II.3	LE MARCHÉ DE LA DOMOTIQUE . . . . .	17
II.4	LES FONCTIONS DE LA DOMOTIQUE . . . . .	18
II.4.1	La fonction de sécurité . . . . .	19
II.4.2	La fonction de surveillance . . . . .	19
II.4.3	La fonction de gestion d'énergie . . . . .	20
II.4.4	La fonction de scénarisation . . . . .	20
II.4.5	La fonction de communication . . . . .	20
II.4.6	La fonction de confort . . . . .	21
II.5	LES TECHNOLOGIES UTILISÉES POUR LA DOMOTIQUE . . . . .	21
II.5.1	La technologie bus filaire . . . . .	21
II.5.2	Le courant porteur en ligne (CPL) . . . . .	22
II.5.3	La radiofréquence . . . . .	23
II.5.4	La mixité des supports . . . . .	24
II.6	MIXITÉ ENTRE LA TECHNOLOGIE RADIO ET FILAIRE . . . . .	24
II.7	LES MATÉRIELS DOMOTIQUES . . . . .	25
II.8	CONCLUSION . . . . .	26

L'ÉVOLUTION des technologies et des automatismes du bâtiment ont permis à la domotique de faire ses premiers pas.

Pleine de promesses quant aux économies d'énergie, à la gestion intelligente des automatismes du bâtiment ou dans l'aide des personnes en perte d'autonomie, la domotique peut aussi être source d'interrogation pour les installateurs électriciens.

Ce chapitre a pour but :

- De définir la domotique et ses fonctionnalités
- De citer les diverses technologies employées en domotique
- De citer quelques équipements utilisés en domotique.

## II.1 DÉFINITION

La domotique, du latin « domus » signifiant maison, est l'ensemble des technologies de l'électronique, de l'information et des télécommunications permettant d'automatiser des bâtiments individuels ou collectifs [sirlan.com ].

## II.2 PRINCIPE DE FONCTIONNEMENT

Le principe de la domotique consiste à faire communiquer ensemble et entre eux les équipements électriques d'un bâtiment. On parle alors de bâtiment intelligent ou de bâtiment communicant.

L'installation domotique peut être pilotée localement ou à distance depuis votre Smartphone, un écran tactile ou encore un ordinateur.

La domotique permet de superviser, de coordonner et de programmer les fonctions du bâtiment afin de répondre à vos attentes en termes de sécurité, de confort, de gestion d'énergie et de communication. Elle participe également à l'aide au maintien à domicile des personnes âgées ou handicapées en facilitant leur quotidien.

Le schéma simplifié, ci-dessous, permet de mieux comprendre la circulation des informations dans une maison « intelligente ».

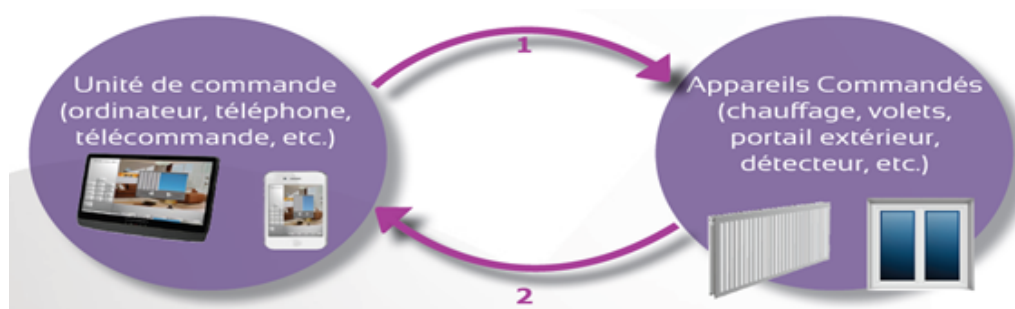


FIGURE II.1 – Circulation d'information en domotique

**Sens 1 :** Circuit de commande, transmission des informations pour la réalisation d'une tâche à partir du tableau électrique, le WiFi, etc.

**Sens 2 :** Informations sur l'état des appareils

## II.3 LE MARCHÉ DE LA DOMOTIQUE

Le marché de la domotique rassemble des acteurs provenant d'univers différents en raison d'une grande diversité d'équipements domotiques. Les acteurs historiques du secteur, à savoir les fabricants de matériels électriques (Schneider, Hager, Legrand, Delta Dore, Theben, ABB) côtoient désormais les opérateurs télécoms et les sociétés informatiques qui proposent des offres permettant d'automatiser une partie des équipements du logement. Ces solutions dites « partielles » sont confrontées aujourd'hui à des solutions globales, totalement intégrées au bâti, et proposées par les fabricants de matériels électriques. L'offre domotique est ainsi devenue plus riche fonctionnellement et surtout plus fiable.

D'un positionnement de niche depuis la fin des années 2000, le marché tend à se positionner sur une dimension de masse.

En effet, comme représenté par le schéma ci-dessous, le haut de gamme et le moyen de gamme représentaient 60 % des ventes en équipement domotique en 2010 [sirlan.com ].

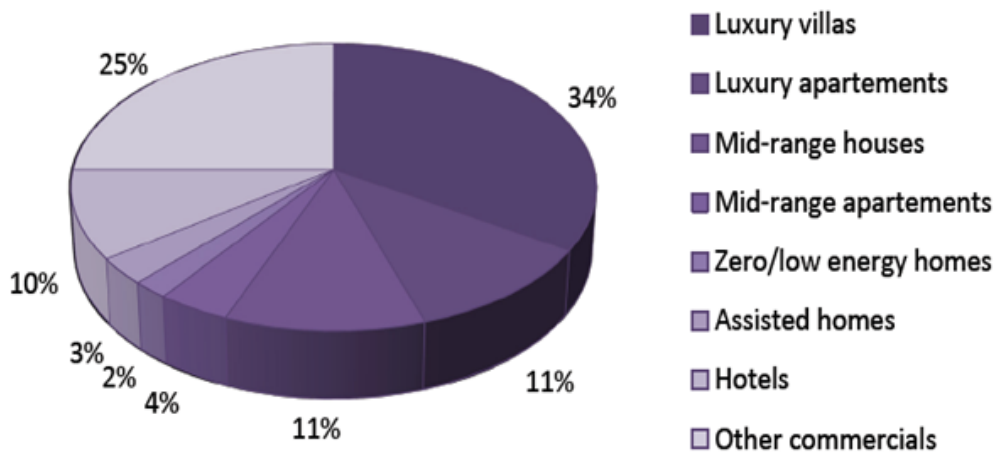


FIGURE II.2 – Gamme de vente d'équipements domotiques

La croissance attendue pour les prochaines années est en forte augmentation grâce à la conjugaison de plusieurs facteurs : baisse des prix permettant de toucher significativement le moyen de gamme, développement d'outils performants de configuration facilitant la mise en œuvre par les installateurs non spécialisés. Egalement, la généralisation du haut débit ouvre la voie à une offre plus attrayante avec la transmission d'images et l'utilisation d'outils ludiques sur Smartphones et Tablettes.

Toutes les conditions sont donc réunies pour faire du marché de la domotique un marché de masse. De plus, avec le vieillissement de la population, le maintien des personnes âgées à domicile génère un besoin d'automatisme et de communication, et joue en faveur d'une domotique généralisée.

La tendance des prix devrait continuer à la baisse et, en conséquence accélérer la démocratisation de la domotique.

Les attentes utilisateurs sont mieux prises en compte :

- Recherche du confort et de la sécurité ;
- Aide aux personnes dépendantes, puisque le maintien à domicile est cité par 79 % des Français comme une action prioritaire de l'état face à la dépendance ;
- Développement durable, une des préoccupations majeures de notre société, traduite par la RT2012 de la loi Grenelle I concernant l'amélioration de l'efficacité énergétique des bâtiments.

Parallèlement, le marché de la domotique doit aussi relever un certain nombre de défis dont :

- Consolider l'offre des constructeurs en améliorant la qualité et en diminuant les coûts.
- Structurer la filière et ses différents acteurs, tels que les architectes et les installateurs. Ils ont besoin de travailler d'avantage.

## II.4 LES FONCTIONS DE LA DOMOTIQUE

L'utilisation de plus en plus importante des Smartphones et des Tablettes contribue à favoriser l'acceptation de la domotique au sein de l'habitat.

Les domaines d'application sont au cœur de la vie quotidienne. Les fonctions suivantes peuvent être réalisées grâce aux technologies intégrées dans la domotique :

- Sécurité,
- Surveillance,
- Gestion de l'énergie,
- Scénarisation des actions,
- Communication,



- Confort.



FIGURE II.3 – Différentes fonctions de la domotique

### II.4.1 La fonction de sécurité

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 à l’hautparleur de la centrale. En termes de sécurité, la domotique permet entre autres de :

- Prévenir les risques provenant de l’extérieur (intrusion, cambriolage ...) comme ceux provenant de l’intérieur (accidents domestiques) ;
- Surveiller à distance vos locaux ou votre habitation depuis votre Smartphone, une Tablette ou un ordinateur ;
- Être averti d’un incident (alarmes techniques) par SMS et/ou par e-mail.

### II.4.2 La fonction de surveillance

Différents capteurs détectent les anomalies :

- Inondation,
- Incendie,
- Fuite de gaz,
- Arrêt du congélateur,
- Coupure de courant,
- Vent ou pluie.

La centrale intervient instantanément pour couper les alimentations, remonter les stores, couvrir la piscine, appeler les numéros d’urgence ou faire retentir la sirène si l’occupant est présent.

### II.4.3 La fonction de 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 suffit pour augmenter le chauffage en prévision de l'arrivée des occupants.

Le délestage permet la souscription d'un abonnement d'une puissance inférieure et la gestion des différentes formules jour/nuit, (Tempo, EJP, ...)

Suivre sa consommation d'énergie afin de contrôler ses dépenses, optimiser ses consommations tout en gardant un certain confort (en gérant la température de son intérieur par zone et selon son occupation) sont autant d'atouts qui vont permettre une meilleure gestion des dépenses énergétiques.

L'augmentation des coûts de l'énergie aussi bien que l'émergence des préoccupations écologiques sont des enjeux importants de notre société actuelle. La domotique propose ainsi de réduire les consommations énergétiques des bâtiments en adaptant ces consommations aux modes de vie des occupants et à l'environnement extérieur. Cela comprend la régulation de l'éclairage et du chauffage, le traitement de l'air, l'optimisation des ouvrants, la programmation horaire, les commandes à distance, les interrupteurs automatiques pour l'éclairage d'un escalier ou d'un couloir, l'ouverture ou la fermeture d'un volet selon l'ensoleillement...

### II.4.4 La fonction de 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, ...

À partir d'un bouton unique, tous les éclairages du living seront ajustés pour le dîner, une soirée télévision ou la création d'une ambiance lumineuse adaptée à l'activité de l'occupant.

### II.4.5 La fonction de communication

Aujourd'hui, une centrale domotique sait communiquer

- Par téléphone,
- Par ordinateur (Internet).

Ceci permet à une personne de recevoir l'état de son installation et d'émettre des alertes et piloter sa maison de n'importe quel endroit du monde, de son bureau ou de sa voiture.

Lire vos films sur le support de votre choix, mettre en place un système multi room et pouvoir diffuser le son et l'image dans votre logement sont des exemples de scénarios qu'offre la domotique. Les équipements vidéo, home-cinéma, réseau téléphonique et internet sont intégrés dans toutes les pièces de votre habitation.

Il est possible de gérer et diffuser ses bibliothèques de musiques et de vidéos dans différentes pièces, de sauvegarder ses données informatiques, d'avoir accès à distance

à ses ordinateurs, de faciliter la mobilité et le télétravail. Ces systèmes sont en général indépendants et peuvent être pilotés par les fonctions domotiques.

### II.4.6 La fonction de confort

Ouvrir le portail sans descendre de voiture, allumer automatiquement la lumière en ayant les bras chargés de paquets, arroser automatiquement le jardin, ouvrir ou fermer les volets ou les stores, programmer une ambiance sonore dans la maison, disposer de la télévision dans plusieurs pièces, déclencher la préparation du café avant le réveil, adapter la température aux conditions extérieures, ... autant de fonctionnalités qui peuvent devenir indispensables.

Un émetteur radio, une télécommande infrarouge, un téléphone, une horloge ou un détecteur : différents dispositifs de commande peuvent agir sur le même appareil et un même détecteur peut engendrer des actions différentes (ex : commander un éclairage à partir d'une télécommande tout en conservant les fonctions de l'interrupteur mural).

Un détecteur de présence peut donner l'alarme en cas d'absence de l'occupant mais allumera la lumière dans l'autre cas.

## II.5 LES TECHNOLOGIES UTILISÉES POUR LA DOMOTIQUE

Généralement, une installation domotique peut être conçue sur trois principaux types de technologie. Ces technologies peuvent cohabiter, être superposées suivant l'évolution de l'installation dans le temps.

### II.5.1 La technologie bus filaire

La technologie bus filaire est souvent utilisée dans la construction ou la rénovation de bâtiments en raison de l'installation d'un bus filaire.

Cette technologie veille à ce que tous les composants communiquent entre eux avec le même langage afin qu'ils puissent échanger des informations, les analyser et les traiter.

L'information circule dans les deux sens : une unité d'entrée envoie des informations aux récepteurs de sortie chargés de faire effectuer une tâche précise à des équipements de l'installation électrique (éclairage, ouvrants, chauffage, alarmes). Ces derniers envoient ensuite des informations concernant leur état vers la ou les unités d'entrée.

L'installation de ce dispositif est composée de deux réseaux :

- un réseau bus filaire reliant les capteurs (détecteurs, interrupteurs, sondes) aux actionneurs (éclairage, ouvrants, chauffage, produits de puissance),
- un réseau d'alimentation reliant les actionneurs au courant fort.

Ce type d'installation présente pour les utilisateurs plusieurs avantages :

- la réduction massive du câblage : un seul câble en général pour tous les équipements au lieu d'un par équipement,
- une meilleure fiabilité de la transmission des informations et de l'installation,
- une supervision en local ou à distance,
- une évolutivité de l'installation à tout moment.

Le schéma, ci-dessous, rend compte de l'installation avec une technologie BUS :

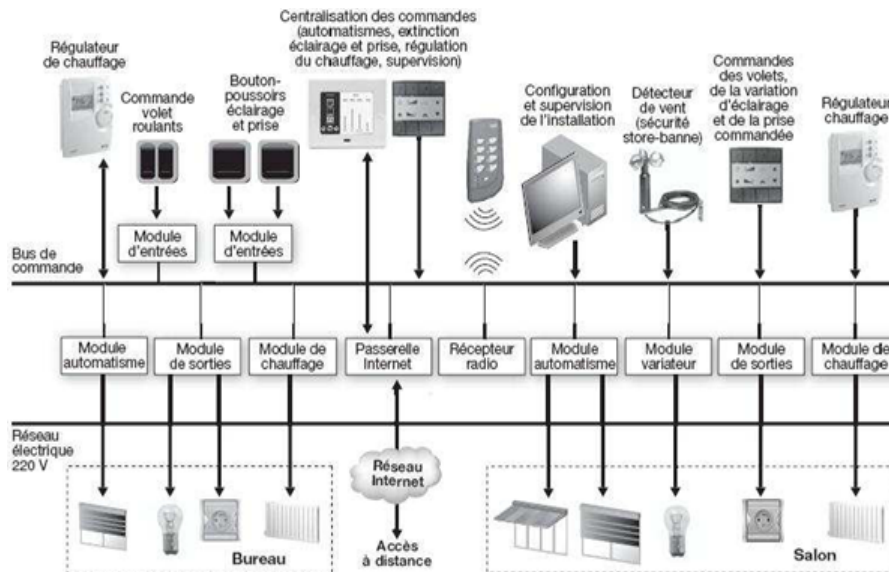


FIGURE II.4 – Exemple de la technologie Bus

Le bus filaire est constitué d'une ou plusieurs paires de conducteurs torsadés, permettant la circulation de données à haut débit.

Il s'agit de la solution la plus fiable et la plus performante. La mise en œuvre de ce type de bus dans le domaine du logement neuf est facilité par l'implantation du pré câblage préconisé par la norme C 15-100.

Les deux types de bus de commande standard de cette catégorie sont les suivants :

#### II.5.1.1 EIB (European Installation Bus) /Konnex

Ce protocole a été retenu par plus de 150 constructeurs et s'implante peu à peu dans l'habitat résidentiel.

#### II.5.1.2 Lonworks

Très sophistiqué, il est essentiellement utilisé dans les projets d'immeubles de bureaux ou d'hôpitaux, par exemple. Contrairement à l'EIB/Konnex, qui est relativement centralisé, Lonworks permet à tous les produits reliés au réseau de communiquer directement entre eux.

### II.5.2 Le courant porteur en ligne (CPL)

La technologie du courant porteur en ligne (CPL) permet le transfert et l'échange d'informations et de données en passant par le réseau électrique existant.

L'installation est composée d'émetteurs et de récepteurs connectés au réseau électrique qui communiquent entre eux. Le principe de base du réseau CPL (Courant Porteur en Ligne) est d'utiliser les circuits de distribution électrique du bâtiment pour véhiculer des données et des commandes.

On distingue généralement les deux grands types de courant porteur suivants [[scribd.com](http://scribd.com)].

#### II.5.2.1 Courant porteur domestique

Cette technologie est conçue pour piloter l'éclairage, le chauffage, les automatismes, les prises de courant et la sécurité (ex : norme X10, In One de Legrand, X2D) [[scribd.com](http://scribd.com)].

### II.5.2.2 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 ordinateur, imprimantes, accès Internet, serveur multimédia, écran tactile, point d'accès WiFi, etc [[scribd.com](http://scribd.com)].

### II.5.2.3 Courant porteur audiovisuel

Cette technologie permet de distribuer l'image et le son dans la maison (standard HomePlug AV). Ces trois types de CPL utilisent 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)



FIGURE II.5 – Exemple d'une installation CPL

L'avantage d'une installation utilisant cette technologie est qu'elle ne nécessite pas de travaux particuliers.

### II.5.3 La radiofréquence

Avec la technologie radio, la transmission d'informations s'effectue sans fil. Elle convient particulièrement aux travaux de rénovation légère étant donné qu'elle est souvent utilisée en complément d'une installation filaire traditionnelle.

En utilisant les ondes radio, l'émetteur (une commande sans fil) peut ainsi piloter un récepteur (interrupteur, prise ...). L'avantage de la radiofréquence est qu'elle permet de faire évoluer une installation électrique sans grands travaux.

On cite comme exemples de technologies radiofréquence :

- KNX radio fréquence, Zigbée, Encan...
- Solutions constructeurs telles que X3D de Delta Dore et MyHome RF de Legrand
- **Télécommande** : infrarouge ou radio
- **Réseau informatique** : WiFi, Bluetooth, ...
- **Téléphone sans fil** : DECT

- Réseau audiovisuel : diffusion de la télévision dans toutes les pièces

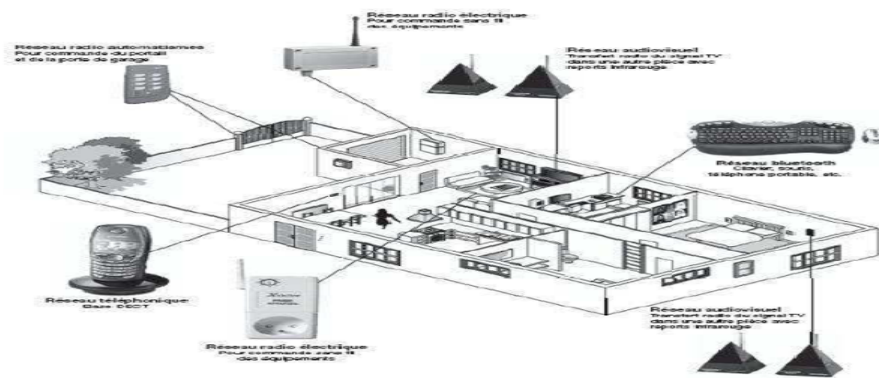


FIGURE II.6 – Installation réseaux sans fil en domotique

### II.5.4 La mixité des supports

Un bus domotique peut utiliser plusieurs vecteurs au sein d'un même réseau :

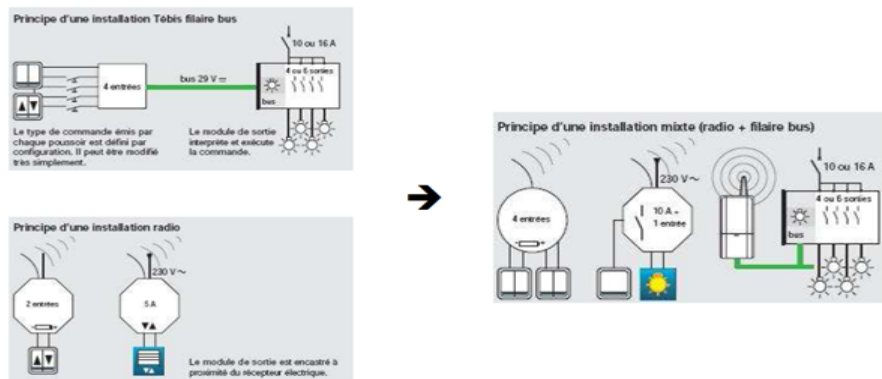


FIGURE II.7 – Mixité entre la technologie radio et filaire

## II.6 MIXITÉ ENTRE LA TECHNOLOGIE RADIO ET FILAIRE

Il y a plusieurs manières pour un utilisateur pour commander sa maison intelligente ça peut se faire par :

- Un simple navigateur web peut être l'interface humaine pour toutes ces applications.
- Une application (App) sur les OS Android, Apple ou Windows est également courante.
- Télécommande à radio fréquences.
- Télécommande à rayons infrarouges.
- Commande par ordinateur.

L'interface homme-machine peut également ne pas se baser sur un écran et un clavier, il peut s'agir :

- De la voix (par reconnaissance vocale, et synthèse vocale).
- De murs « tactiles » ou autre surfaces rendues tactiles ou clavier.
- De badge (RFID, NFC, propriétaires).
- De mouvement de la main (par reconnaissance de forme/mouvement, type Kinect).
- Des objets divers, voire insolites (Karotz).

L'objectif de la maison communicante est aussi de minimiser le temps que l'humain doit passer devant l'interface homme machine. Plusieurs technologies existent pour déclencher des automatismes (par programmation volontaire, par des algorithmes évolués tels l'apprentissage des modes de vie, par de très nombreuses techniques algorithmique, d'apprentissage supervisé ou non) [scribd.com].

## II.7 LES MATÉRIELS DOMOTIQUES

Il y a plusieurs équipements et composants utilisés dans la domotique et voici quelques-uns figurant dans l'image suivante :



FIGURE II.8 – Quelques équipements utilisés en domotique

## II.8 CONCLUSION

On a vu dans ce chapitre la domotique, ses fonctionnalités et son impact sur la vie de l'individu.

On a vu aussi que ses équipements domotiques peuvent être installés selon trois manières : sans fil, filaire ou par CPL selon le besoin voulu.

Avec la domotique, on peut assurer la sécurité de notre vie privée en utilisant une caméra de surveillance.

Dans le chapitre suivant, on va voir en détails c'est quoi la carte électronique Raspberry Pi, ses composants, son domaine d'utilisation et comment contrôler ses broches avec le langage python.



# PRÉSENTATION DE RASPBERRY PI

# III

## SOMMAIRE

III.1 CHOIX DE LA SOLUTION . . . . .	28
III.1.1 Carte Arduino . . . . .	28
III.1.2 Carte STM32 . . . . .	28
III.1.3 Raspberry Pi . . . . .	29
III.1.4 Beaglebone . . . . .	29
III.1.5 Analyse du tableau comparatif . . . . .	30
III.2 RASPBERRY PI . . . . .	30
III.2.1 Présentation du Raspberry . . . . .	30
III.2.2 Les modèles de Raspberry Pi . . . . .	30
III.2.3 Raspberry Pi Type B + . . . . .	33
III.2.4 Les composants de Raspberry Pi . . . . .	33
III.2.5 Connexion de l'ensemble des éléments . . . . .	37
III.2.6 Les différentes utilisations du Raspberry Pi . . . . .	37
III.2.7 Préparation du Raspberry Pi . . . . .	38
III.2.8 Programmer pour Raspberry Pi . . . . .	43
III.3 CONCLUSION . . . . .	45

**D**ANS ce chapitre, on va présenter la carte électronique Raspberry Pi comme un module de notre projet de domotique. On va voir c'est quoi la carte Raspberry Pi, ses composants, ses fonctionnalités aussi on parle du langage python son importance et sa simplicité pour contrôler les GPIO du Raspberry.

### III.1 CHOIX DE LA SOLUTION

Durant notre recherche sur la partie électronique qui va être implémentée pour automatiser notre maison intelligente, nous avons constaté qu'au cours des dernières années, de nombreuses cartes de développement modulaire sont apparues. Chaque carte présente des avantages et des inconvénients. Elles se différencient par leurs fonctionnalités, leurs complexités et leurs prix [Aissia & Chaieb 2015]. On peut citer parmi ces cartes :

#### III.1.1 Carte Arduino

c'est un circuit intégré à base d'un microcontrôleur programmable qui peut analyser et produire des signaux électriques de façon à effectuer des tâches précises [Aissia & Chaieb 2015].



FIGURE III.1 – Carte Arduino UNO

#### III.1.2 Carte STM32

c'est un circuit intégré à base d'un microcontrôleur composé d'un processeur ARM, une mémoire RAM et une interface de débogage [Aissia & Chaieb 2015].



FIGURE III.2 – Carte STM32

### III.1.3 Raspberry Pi

C'est un mini-ordinateur qui exécute le système d'exploitation Linux [Aissia & Chaieb 2015].



FIGURE III.3 – Carte Raspberry Pi

### III.1.4 Beaglebone

C'est une mini-carte électronique qui possède les fonctionnalités d'un ordinateur basique.

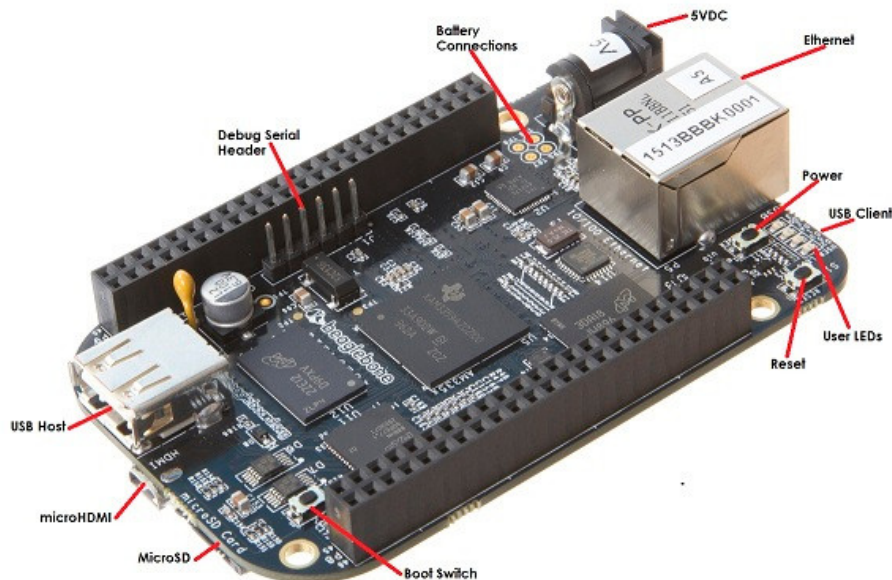


FIGURE III.4 – Carte Beaglebone

Le tableau suivant nous donne une comparaison technique entre ces cartes :

Nom de la carte	Arduino UNO	Beaglebone	Raspberry Pi B+	STM 32
Naissance	2005	2008	2012	2007
Prix (dollars \$)	30	90	Moins de 40	10
Processeur	ATmeg328 8bits	Sitara335x sur un cotex M8	BCM2835 sur un ARM11-cpu intégrée	cotex
RAM	2 Ko	256 Mo	512 Mo	Max 128 Ko
Mémoire Flash	32 Ko	MicroSD-4G	MicroSD-32G	Max 2048 Ko
Consommation (w)	0.5	2.5	3.5	basse
Ethernet	NON	OUI	OUI	NON
USB	NON	1	4	Micro-USB
Sortie vidéo	NON	NON	HDMI,RCA	NON
système d'exploitation	RTOS	Linux	Linux	RTOS

TABLE III.1 – Comparaison technique : Arduino Uno, Beaglebone, Raspberry Pi etSTM32

### III.1.5 Analyse du tableau comparatif

- Nous remarquons qu'en terme de mémoire vive ou mémoire externe, la capacité de Raspberry est plus importante que les autres cartes.
- Le Raspberry possède des connectiques (HDMI, port Ethernet, port USB ...) de plus par rapport au Beaglebone, STM32 et Arduino UNO.
- Le Raspberry et le Beaglebone se différencient par leur exécution du système Linux qui facilite les communications avec le monde extérieur.
- L'Arduino a une puissance faible devant les autres cartes.

En tenant compte du tableau III.1, on déduit que le Raspberry représente un outil plus performant et mieux adapté à nos besoins. Le Raspberry possède des connectiques convenables au fonctionnement de notre système. De plus, son rapport performance-prix est plus qu'abordable [Aissia & Chaieb 2015].

## III.2 RASPBERRY PI

### III.2.1 Présentation du Raspberry

Raspberry est une carte mère d'un mini-ordinateur qui peut être branchée à n'importe quel périphérique (souris, clavier...). Cette carte est fabriquée pour aider à étudier les ordinateurs et pour représenter un moyen d'apprentissage de la programmation informatique en plusieurs langages (python, scratch...). Elle est aussi capable de lire les vidéos à haute définition et même à installer des jeux vidéo. L'intérêt d'utiliser le Raspberry Pi est sa capacité d'interaction avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...) et des autres logiciels compatibles [raspberrypis 2017].

### III.2.2 Les modèles de Raspberry Pi

#### III.2.2.1 Modèles A

il y a deux types A et A+.

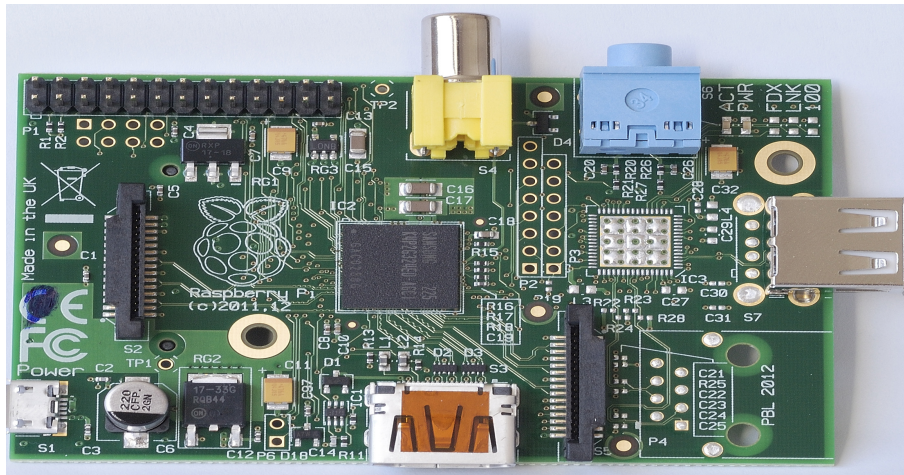


FIGURE III.5 – *Modèle A*

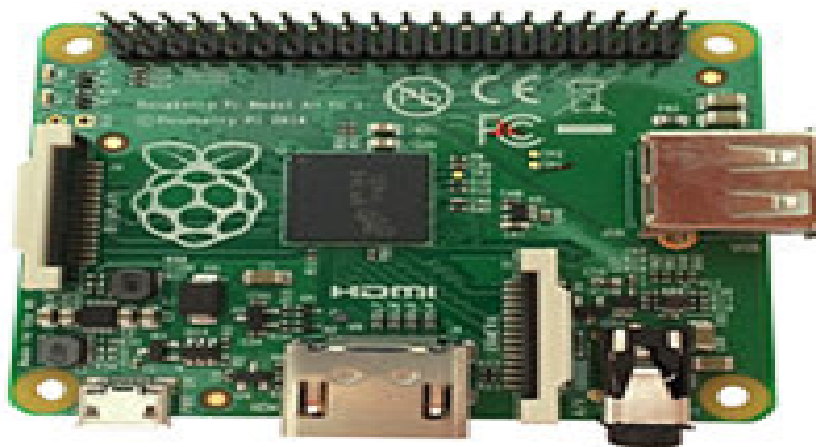


FIGURE III.6 – *Modèle A+*

### III.2.2.2 Modèles B

il y a 4 types B1 et B1+, B2, B3.



FIGURE III.7 – *Modèle B1*

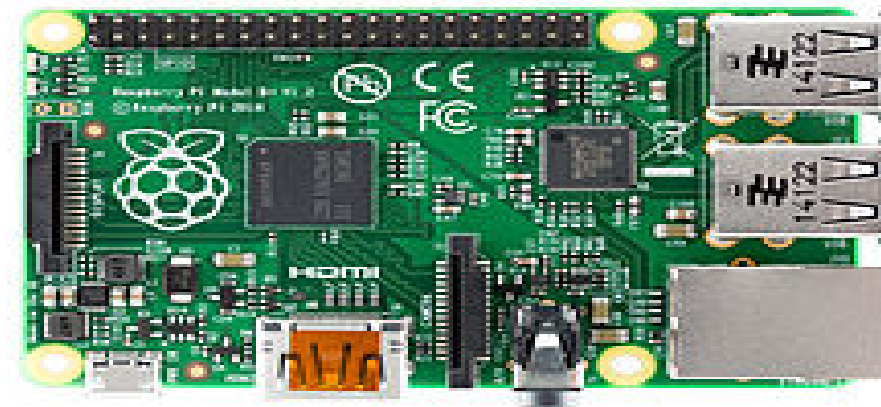


FIGURE III.8 – *Modèle B1+*

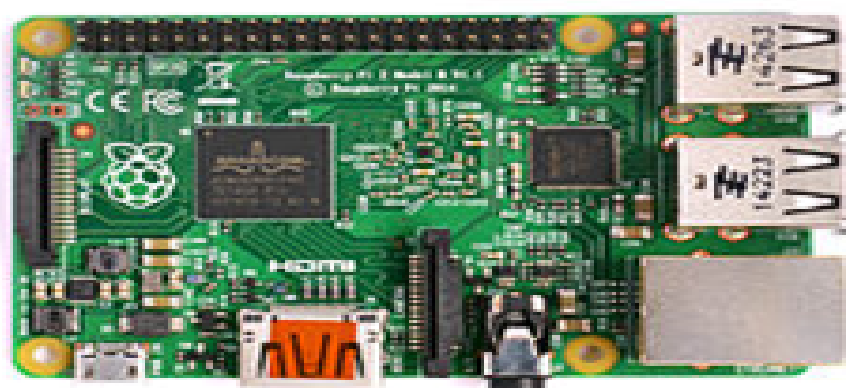


FIGURE III.9 – *Modèle B2*



FIGURE III.10 – *Modèle B3*

### III.2.2.3 **Modèle 0**

La figure III.11 représente le modèle 0

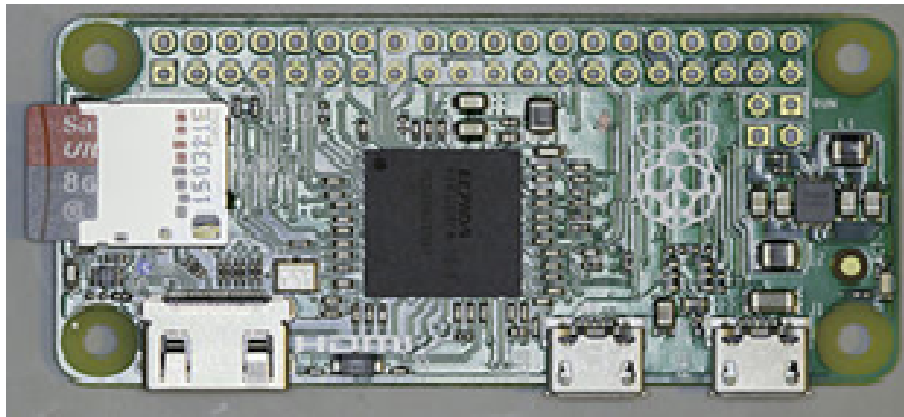


FIGURE III.11 – Modèle 0

### III.2.3 Raspberry Pi Type B +

Pour notre projet, nous avons opté pour Raspberry Pi B+ pour beaucoup de raisons notamment sa puissance et sa rapidité par rapport au Raspberry Pi Type B. Raspberry Pi Type B+ est la carte mère idéale pour un système d'architecture ARM. Equipée d'un puissant processeur ARM1176JZF-S core de 700 MHz, d'une mémoire RAM intégrée de 512 Mo et d'un contrôleur graphique Broadcom VideoCore III capable de décoder les flux vidéo HD 1080p, la carte mère Raspberry Pi Type B+ est une solution performante et abordable pour tous types de systèmes compacts ou embarqués (MediaCenter, domotique, affichage dynamique, robotique...etc.) [[rasberrypis 2017](#)].

### III.2.4 Les composants de Raspberry Pi

La figure suivante représente un Raspberry standard :

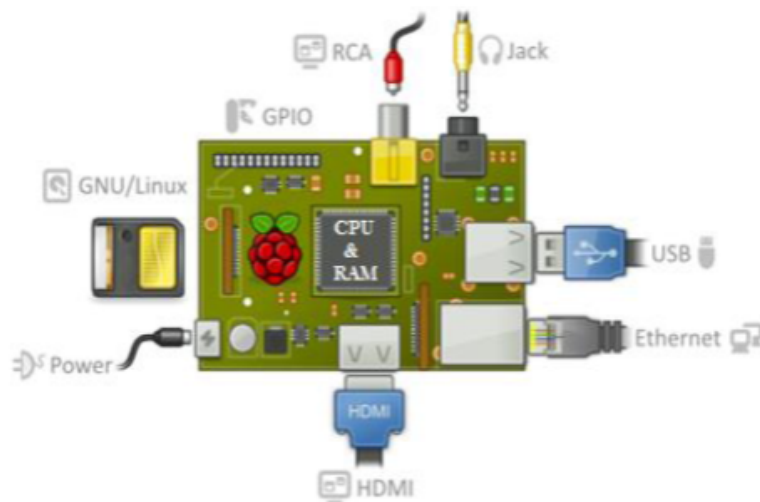


FIGURE III.12 – Les composants standard d'un Raspberry

#### III.2.4.1 Processeur ARM

Les architectures ARM sont des architectures de processeurs, à faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd ». Raspberry Pi B+ possède un processeur intégré Quad-core ARM Cortex-A7 900 MHz (Broadcom BCM2836) [[Aissia & Chaieb 2015](#)].

### III.2.4.2 Mémoire vive RAM

C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement, elle a une capacité de **512Mo** [Aissia & Chaieb 2015].

### III.2.4.3 Une connectique variée

1. **HDMI** : « High Definition Multimedia Interface » permet de relier le Raspberry Pi à un dispositif compatible : écran LCD ou un vidéoprojecteur...
2. **Port USB 2.0** : Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques, et il y a 4 ports USB. Le Raspberry Pi fonctionne avec pratiquement n'importe quels claviers et souris USB. Vous pouvez aussi utiliser la plupart des claviers et des souris sans fil (ceux qui marchent avec un dongle que l'on branche sur un port USB).
3. **Port Ethernet** : C'est un port qui correspond au protocole international Ethernet de réseau local à commutation de paquets.
4. **Prise RCA** : « Radio Corporation of America » est un connecteur électrique utilisé dans le domaine audio/vidéo via jack.
5. **Un slot les cartes SD** : Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe. Vous pouvez utiliser votre propre carte SD dans le Raspberry Pi, mais elle devra être préparée avec une image disque d'un système d'exploitation.
6. **Une prise jack** : C'est une connectique audio-vidéo sa dimension est égale à 3.5 mm.
7. **Bloc d'alimentation** : La figure ci-dessous représente le bloc d'alimentation USB classique avec un câble USB-A vers micro-USB. L'alimentation doit être capable



FIGURE III.13 – Bloc d'alimentation USB.

de fournir au moins 700 mA, mais 1 A donne au Raspberry Pi un peu de marge qui sera utilisée par les périphériques connectés à ses ports USB. Si vous regardez attentivement les spécifications de l'alimentation, vous devriez être capable de déterminer sa capacité. Il arrive que la capacité soit exprimée en watts (W); si tel est le cas, elle doit être au moins de 3 W. Si elle indique 5 W, cela correspond à 1 A.

8. **GPIO** : Les broches GPIO « General Purpose Input/Output » du Raspberry Pi vous permettent de contrôler d'autres composants électroniques ainsi que des interfaces telles que des LED, des moteurs et des relais. Ces diverses interfaces sont généralement regroupées sous le terme de « sorties ». Pour ce qui est des « entrées »,



voire Raspberry Pi peut lire et interpréter l'état de boutons, d'interrupteurs, de capteurs de température, de lumière, de mouvement ou de proximité, etc., la liste est très longue. En utilisant les interfaces numériques asynchrones (port série vu au chapitre précédent) et synchrones (SPI et I2C), les applications potentielles sont encore plus nombreuses. Le Raspberry B+ comporte 40 broches.

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

FIGURE III.14 – Le port GPIO

#### III.2.4.4 Système d'exploitation

La plupart des systèmes qui fonctionnent sur Raspberry Pi sont des versions du système d'exploitation Linux. Parce que Linux est open source, les développeurs peuvent l'adopter pour des buts spécifiques. La distribution recommandée est Raspbian. C'est un système d'exploitation libre basé sur Debian optimisé pour le matériel Raspberry Pi. Cependant Raspbian fournit plus qu'un simple système d'exploitation : il est livré avec plus de 35.000 paquets, des logiciels précompilés qui nous facilitent le développement soft. Il y a autre système d'exploitation crée par Microsoft spécialement pour le domaine des IoT Windows 10 IoT Core qui peut être installé sur notre carte Raspberry Pi B+ [Aissia & Chaieb 2015].



FIGURE III.15 – Logo du système d'exploitation Raspbian et Windows 10 IoT.

#### III.2.4.5 Boîtier

Le Raspberry Pi est livré nu, sans protection. Cela permet d'abaisser son prix, mais cela le rend vulnérable à la casse. Il est par conséquent préférable d'acheter un boîtier dès que vous le pouvez ou bien d'en réaliser un vous-même. La figure III.16 illustre un exemple d'une boîtiers prêt à l'emploi qui sont commercialisés. Le choix du boîtier dépend de ce vous envisagez de faire avec votre Raspberry Pi [Aissia & Chaieb 2015].



FIGURE III.16 – Boîtier de Raspberry

#### III.2.4.6 Adaptateur Wi-Fi

Aucun des deux modèles de Raspberry Pi ne prend en charge le Wi-Fi. Si vous souhaitez connecter en Wi-Fi votre Raspberry Pi au réseau, vous avez deux possibilités. La première est d'utiliser un adaptateur sans fil USB qui se branche sur un port USB (figure III.17). Avec un peu de chance, Linux devrait le reconnaître et permettre de vous connecter immédiatement (ou vous indiquer ce que vous devez faire pour vous connecter).

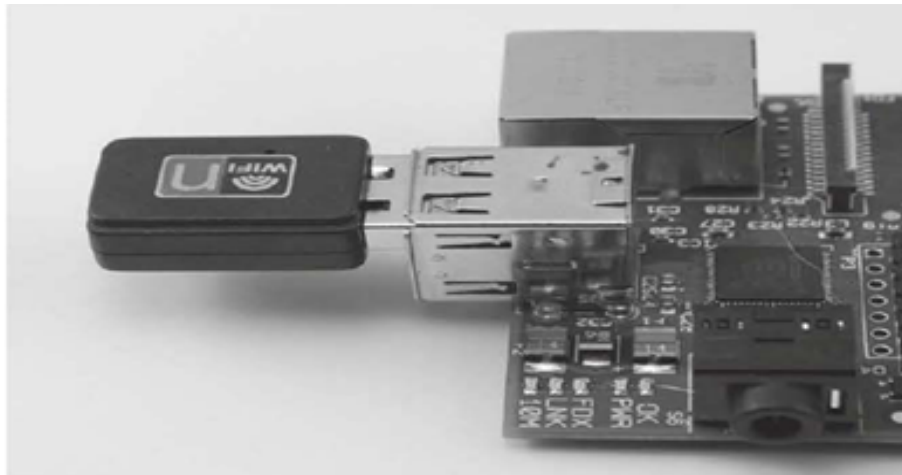


FIGURE III.17 – Adaptateur Wi-Fi

### III.2.5 Connexion de l'ensemble des éléments

Une fois que vous avez tous les éléments dont vous avez besoin, vous pouvez tous les brancher et démarrer votre Raspberry Pi pour la première fois. La figure III.18 illustre le branchement de chacun des éléments [Aissia & Chaieb 2015].

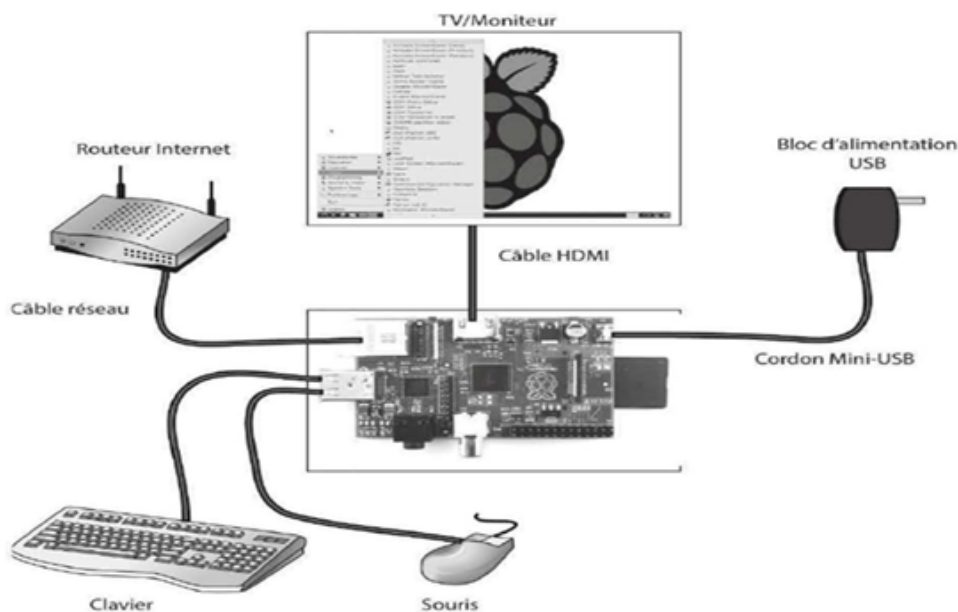


FIGURE III.18 – Système Raspberry Pi

### III.2.6 Les différentes utilisations du Raspberry Pi

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

#### III.2.6.1 Les serveurs

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

### 1. Le serveur web :

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

### 2. Le serveur de stockage :

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

### 3. Le serveur de service :

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

#### III.2.6.2 L'utilisation MultiMedia

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

#### III.2.6.3 La domotique

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

#### III.2.6.4 Les systèmes embarqués

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

#### III.2.6.5 Un ordinateur, tout simplement

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

### III.2.7 Préparation du Raspberry Pi

#### III.2.7.1 Installation de l'OS « Raspbian »

Le Raspberry Pi est initialement sans système d'exploitation. D'abord, nous avons téléchargé Raspbian à partir du site officiel : [raspberrypi.org/downloads](http://raspberrypi.org/downloads). Puis, nous l'avons installé sur la carte mémoire par le biais d'un logiciel d'écriture d'image : « Win32 Disk Imager ».

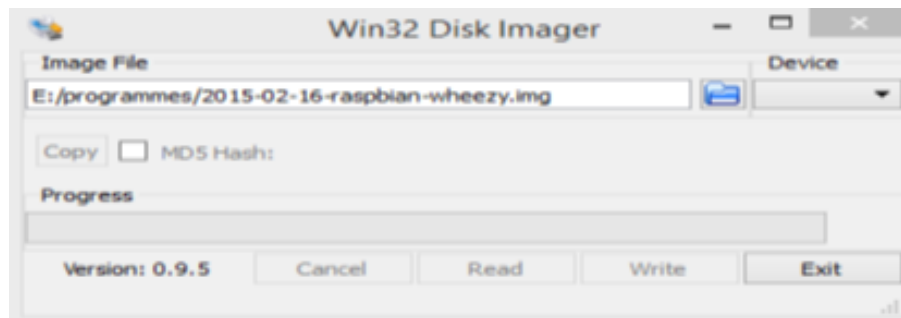


FIGURE III.19 – Installation du Raspbian avec Win32 Disk Imager

Après avoir installé le système d’exploitation, nous avons exécuté les commandes suivantes dans une session de terminal pour le mettre à jour :

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

### III.2.7.2 Démarrage du système

Suivez ces étapes pour démarrer votre Raspberry Pi pour la première fois.

1. Insérez la carte SD dans son support.
2. Branchez un clavier USB et une souris au Raspberry Pi.
3. Connectez la sortie HDMI à votre téléviseur ou moniteur.
4. Branchez l’alimentation du Pi.

Si tout se passe correctement, vous verrez une liste de messages défiler à l’écran. Ces messages de log montrent tous les processus du système d’exploitation qui se lancent au démarrage de votre Pi. Vous verrez l’interface réseau s’initialiser et les différents périphériques seront détectés. Ces messages peuvent être examinés par la suite, en saisissant la commande **dmesg** sur la ligne de commande du **bash**. Au tout premier démarrage de votre Pi, l’outil **raspi-config** se lancera automatiquement pour personnaliser votre carte (figure III.20) . La configuration par défaut devra être adaptée pour que votre Raspberry Pi fonctionne conformément à vos attentes. Si besoin, vous pourrez relancer ultérieurement cet outil de configuration en saisissant la commande suivante : [Aissia & Chaieb 2015]

**sudo raspi-config**

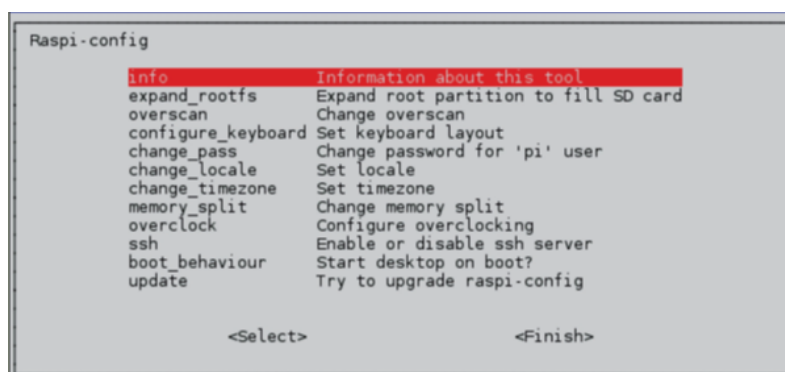


FIGURE III.20 – Le menu principal de l’outil raspi-config

### III.2.7.3 Configurer le Pi

Nous allons à présent passer en revue les différents paramètres de base et distinguer ceux qui sont essentiels de ceux qui pourront être réglés par la suite. Dans l'outil **raspi-config**, utilisez les flèches de direction du clavier haut et bas pour naviguer dans la liste. La barre d'espace permet de sélectionner une option et la touche Tab de passer d'un champ à un autre ou de déplacer le curseur en bas de l'écran pour quitter [[htpc\\_raspberrypi 2017](#)]. Les options disponibles sont les suivantes :

- **expand\_rootfs** : vous devez toujours choisir cette option, qui agrandit le système de fichiers Linux, afin d'utiliser l'intégralité d'une carte SD. Par exemple, si vous disposez d'une carte SD de 4 Go, Linux pourra accéder à 2 Go supplémentaires.
- **Overscan** : pour commencer, laissez cette option désactivée. Si vous possédez un moniteur haute définition, il est possible que du texte dépasse de votre écran. Pour résoudre ce problème, activez l'option overscan et modifiez les valeurs pour que l'image s'adapte à la taille de votre écran. Entrez des valeurs positives si l'image sort de l'écran, des valeurs négatives si des marges noires apparaissent sur les côtés.
- **keyboard** : par défaut, les paramètres du clavier sont réglés pour un clavier britannique (QWERTY). Pour que les bonnes lettres soient saisies lorsque vous tapez sur les touches correspondantes, vous devez choisir le type de clavier approprié. Heureusement, Linux supporte énormément de langues et de dispositions de touches. **Remarque** : vos paramètres de localisation (option `change_locale` ci-après) peuvent affecter les paramètres du clavier.
- **password** : nous vous conseillons de modifier le mot de passe par défaut (raspberry) pour le rendre plus fiable.
- **change\_locale** : si vous ne vous trouvez pas en Angleterre, vous devrez modifier la valeur de cette option pour activer votre langue et vos préférences d'encodage des caractères. Le paramètre par défaut est l'anglais (en) britannique (UK) avec un encodage de caractères en UTF-8 (ce qui donne en\_GB.UTF-8). Il est conseillé de sélectionner fr\_FR.UTF-8 si vous êtes en France.
- **change\_timezone** : modifiez cette option pour sélectionner le fuseau horaire correspondant au lieu où vous vous trouvez.
- **memory\_split** : cette option nous permet de changer la quantité de mémoire allouée à l'affichage. Le partage par défaut convient pour le moment.
- **overclock** : nous avons la possibilité de faire tourner le processeur à des vitesses supérieures à 700 MHz. Pour notre premier démarrage, conservez les paramètres par défaut. Si par la suite on doit modifier ce paramètre, essayant les valeurs Medium ou Modest. Le mode Turbo permet de monter à 1 GHz avec un dissipateur.
- **ssh (Serveur Secure Shell)** : cette option active l'accès au shell sécurisé, ce qui nous permet de nous connecter à notre Raspberry Pi en passant par le réseau. Activant cette option, elle est vraiment très pratique.
- **boot\_behaviour** : cette option est sélectionnée par défaut et active le démarrage automatique de l'environnement de bureau graphique. Si nous sélectionnons No, le Pi restera en mode Texte. Nous pourrions toujours lancer l'interface graphique manuellement, comme suit : [[htpc\\_raspberrypi 2017](#)]

```
raspberrypi login : pi
```

```
Password :raspberry (ou le mot de passe que vous venez de paramétrer)
```

```
pi@raspberrypi ~ $ startx
```

Une fois la configuration terminée, sélectionnant Finish (accédez avec la touche Tab). Selon les options que nous aurons modifiées, il se peut que l'outil **raspi-config** redémarre le système. Sinon, on sera redirigé vers la ligne de commande sur laquelle vous saisirez :

```
pi@raspberrypi ~ $ sudo reboot
```

afin que notre Pi redémarre et prenne en compte nos nouveaux paramètres [[htpc\\_raspberrypi 2017](#)].



FIGURE III.21 – Écran de démarrage Raspberry

#### III.2.7.4 Éteindre le Raspberry Pi

Il n'existe pas de bouton d'alimentation sur le Raspberry Pi, quoique les nouvelles cartes soient équipées d'une broche permettant d'accueillir un bouton de réinitialisation. Pour éteindre correctement le système d'exploitation, allez dans le menu Logout de l'interface graphique et sélectionnez **shutdown**. On peut aussi éteindre le Pi depuis la ligne de commande en tapant :

```
pi@raspberrypi ~ $ sudo shutdown -h now ou sudo Ralt
```

**Remarque :** Veillez bien à arrêter le système d'exploitation comme indiqué ici et à ne pas débrancher sauvagement votre Pi. Si vous coupez le courant sans arrêter le système correctement, vous risquez d'endommager la carte SD [[htpc\\_raspberrypi 2017](#)].

#### III.2.7.5 Connexion à distance

Pour pouvoir se connecter à distance à notre Raspberry, il faut le connecter à un réseau local et faire appel au protocole SSH.

##### 1. Connexion SSH :

Le serveur SSH est activé par défaut sur le Raspberry Pi. Sous Linux, le client étant intégré dans la plupart des distributions. Par contre, sous Windows, nous devons installer « Putty » qui est un client SSH et Telnet.

Ensuite, il suffit d'entrer l'adresse IP du Raspberry dans Putty pour pouvoir se

connecter. Afin d'accéder à l'interface graphique, nous avons utilisé l'outil préinstallé sur le PC Windows « Connexion Bureau à distance ».

2. **Connexion Bureau à distance de Windows :**

Avant de pouvoir se connecter au bureau du Raspberry, il faut installer sur ce dernier un petit programme. Pour ce faire, tapant sur notre Raspberry (connecté à Internet) la commande suivante : **sudo apt-get install xrdp**

Continuons en appuyant sur 'Y'. Une fois l'installation terminée, nous avons la ligne suivante qui devrait s'afficher.

```
done (done) .
[ ok ] Starting Remote Desktop Protocol server : xrdp sesman
pi@raspberrypi ~$
```

FIGURE III.22 – Installation xrdp réussite

Elle signifie que le protocole est installé et démarré. On peut maintenant passer à la configuration du PC pour se connecter au bureau du Raspberry. Maintenant, on ouvre l'outil « Connexion Bureau à distance » de Windows comme le montre l'image suivante :

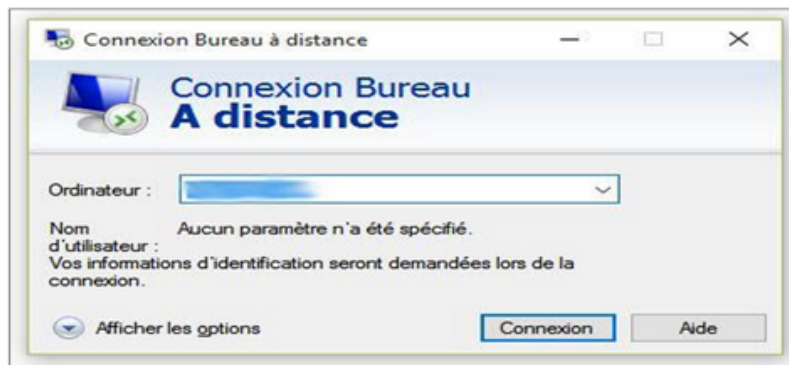


FIGURE III.23 – Outil de connexion à distance

On tape l'adresse IP du Raspberry, on clique sur connexion pour avoir la fenêtre suivante :

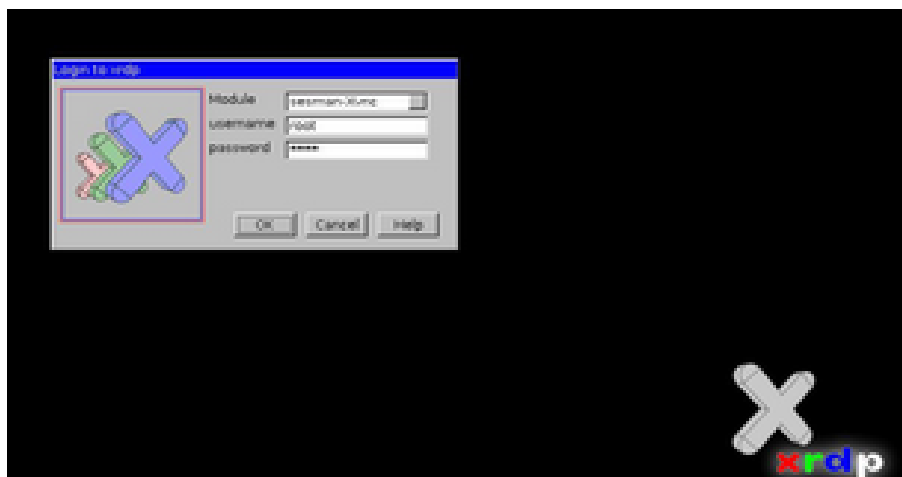


FIGURE III.24 – Fenêtre d'authentification

Une fois connecté, nous pouvons nous servir du bureau de notre Raspberry sur l'ordinateur [[htpc\\_raspberrypi 2017](#)].



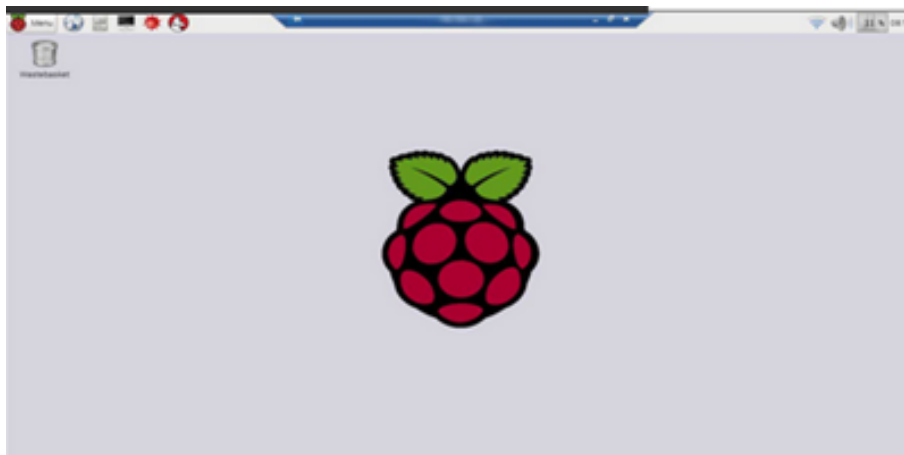


FIGURE III.25 – Bureau Raspberry

### III.2.7.6 Réserveation d'adresse IP

Un serveur DHCP distribue une adresse IP aux clients selon la disponibilité des adresses de la plage définie. Cependant, il est possible de faire en sorte qu'un hôte ait toujours la même adresse IP attribuée, c'est ce que l'on appelle de la réserveation d'adresse. On doit donc indiquer dans le serveur DHCP que telle adresse IP est réservée à telle adresse MAC [[httpc\\_raspberrypi 2017](#)].

Address Reservation				
ID	MAC Address	Reserved IP Address	Status	Modify
1	B8-27-EB-C1-44-D0	192.168.0.100	Enabled	<a href="#">Modify</a> <a href="#">Delete</a>

FIGURE III.26 – Réserveation de l'adresse 192.168.0.100 au Raspberry

Ou bien on alloue une adresse IP statique à notre Raspberry pour une liaison filaire par la configuration suivante :

- On ouvre le fichier de configuration des cartes réseaux :
- Puis indiquons à notre carte réseau que nous souhaitons passer en static :

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

```
auto etho
iface etho inet static
address 192.168.1.20
netmask 255.255.255.0
gateway 192.168.1.1
```

### III.2.8 Programmer pour Raspberry Pi

Le moment est venu de commencer à créer nos propres programmes pour le Raspberry Pi. Le langage que nous allons utiliser s'appelle Python. Il a le grand avantage d'être facile à apprendre tout en étant assez puissant pour créer des programmes intéressants [[site officiel de Raspberry 2017](#)].

### III.2.8.1 Présentation du python

Python est un langage de programmation, dont la première version est sortie en 1991. Créé par Guido van Rossum, il a voyagé du Macintosh de son créateur, qui travaillait à cette époque au Centrum voor Wiskunde en Informatica aux Pays-Bas, jusqu'à se voir associer une organisation à but non lucratif particulièrement dévouée, la Python Software Foundation, créée en 2001 [[site officiel de Raspberry 2017](#)].



FIGURE III.27 – Slogan de langage Python

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Dès l'instant où vous l'installez sur votre ordinateur, vous disposez de nombreuses fonctionnalités intégrées au langage. Ainsi, il existe ce qu'on appelle des bibliothèques qui aident le développeur à travailler sur des projets particuliers.

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

En contrepartie, un langage compilé se révélera bien plus rapide qu'un langage interprété (la traduction à la volée de votre programme ralentit l'exécution), bien que cette différence tende à se faire de moins en moins sentir au fil des améliorations [[site officiel de Raspberry 2017](#)].

### III.2.8.2 Programmation des entrées / sorties avec python

Les broches marquées GPIO peuvent être utilisées comme broches d'entrée / sortie. En d'autres termes, n'importe quelle broche peut être programmée comme une entrée ou une sortie. Dans ce sens on va utiliser plusieurs langages de programmation capable de contrôler ces broches comme le C, Java, Bash... mais dans notre projet on a opté pour le python pour contrôler ces broches.

Le module GPIO est installé par défaut sur les versions les plus récentes de la distribution Raspbian Linux. Mais pour les versions plus anciennes, on doit probablement l'installer et effectuer une mise à jour. Pour cela on exécute la commande suivante :

```
"sudo apt-get install python-rpi.gpio"
```

Ensuite on exécute la commande suivante pour la mise à jour :

```
"sudo apt-get update"
```

**Remarque :** avant d'utiliser les broches, vous devez indiquer au module GPIO comment votre code y accèdera. Le Raspberry Pi autorise deux numérotations : celle de la

sérigraphie du connecteur de la carte (**GPIO.BOARD**), ou la numérotation électronique de la puce (**GPIO.BCM**). À nous de choisir celle qu'on veut.

Voici maintenant un petit programme **led.py** pour contrôler une LED :

```
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BCM)
4 GPIO.setup(18, GPIO.OUT) # broche 18 comme sortie
5 print "LED on"
6 GPIO.output(18, GPIO.HIGH) # led allumee
7 time.sleep(1) #attendre pour une seconde
8 print "LED off"
9 GPIO.output(18, GPIO.LOW) # led etteinte
```

Pour tester le programme on exécute la commande suivante sur la console :

```
"sudo python led.py"
```

### III.3 CONCLUSION

Dans ce chapitre dédié à la présentation du Raspberry Pi, nous avons présenté les divers composants de cette carte, son utilité spécialement pour notre projet sur la domotique, ensuite on a vu comment utiliser les broches pour contrôler les composants électroniques à l'aide de langage de programmation python.

Dans le chapitre suivant, nous allons expliquer notre projet pratique qui est la réalisation d'un prototype d'une maison intelligente.

# RÉALISATION DE LA MAISON INTELLIGENTE

# IV

## SOMMAIRE

IV.1	COMPOSANTS NÉCESSAIRES DE RÉALISATION . . . . .	47
IV.2	LE RASPBERRY PI COMME SERVEUR WEB, POURQUOI ? . . . . .	47
IV.2.1	Installation du serveur Apache . . . . .	47
IV.2.2	Installation de PHP sur le Raspberry . . . . .	48
IV.2.3	Une base de données MySQL pour notre serveur . . . . .	49
IV.3	CONFIGURER LES COMPOSANTS AVEC RASPBERRY PI . . . . .	51
IV.3.1	Les LEDs . . . . .	51
IV.3.2	Les moteurs . . . . .	54
IV.3.3	Le capteur de température . . . . .	60
IV.3.4	Le détecteur de gaz . . . . .	62
IV.3.5	la vidéo-surveillance avec Raspberry Pi : . . . . .	63
IV.4	CONCLUSION . . . . .	65

**C**E chapitre est consacré à notre réalisation d'un prototype d'une maison intelligente équipée de 3 LED qui jouent le rôle des lampes, d'un capteur de température, d'un capteur de gaz, d'une caméra de surveillance, d'une porte et d'un ventilateur .

Dans ce chapitre, nous détaillons les tâches que nous avons pu réaliser étape par étape ainsi que les différents tests de validation.

## IV.1 COMPOSANTS NÉCESSAIRES DE RÉALISATION

- Carte Raspberry Pi B+.
- Carte SD 16 Go.
- Planche à pain.
- Câbles jumpers M/M et M/F.
- Câble Ethernet (RJ45).
- Sonde DS18B20 (capteur de température).
- Capteur de GAZ MQ-2.
- 2 Ponts H L293D.
- 1 Moteur DC (utilisé comme un ventilateur).
- 1 Moteur pas à pas (utilisé comme une porte).
- Webcam utilisée comme caméra de surveillance.
- 3 LEDs (rouge, jaune, bleu).
- 4 Résistances (dont 3 résistances de 100  $\Omega$  , une de 4.7k $\Omega$ ).

## IV.2 LE RASPBERRY PI COMME SERVEUR WEB, POURQUOI ?

Pour pouvoir accéder à notre Raspberry à distance via WiFi et contrôler les GPIOs, on a besoin de rendre notre carte un serveur Web capable d'héberger un ou plusieurs sites Internet et assurer donc la communication avec un client et répondre à ses demandes grâce au protocole réseau http.

Le plus souvent, un serveur Web utilise d'autres logiciels qui fonctionnent en collaboration avec le serveur HTTP (Apache, IIS, Lighthttp), comme le serveur de base de données (MySQL, Oracle), et pour communiquer avec un serveur web, on a besoin d'un langage de programmation serveur comme PHP.

Dans notre projet, on va opter pour installer la combinaison (Apache, PHP, MySQL).

### IV.2.1 Installation du serveur Apache

#### IV.2.1.1 Apache, c'est quoi ? :

Tout d'abord, nous allons installer Apache, qui est le serveur web en tant que tel. Quand on parle de serveur web, on pense souvent à la machine, mais ce terme désigne aussi le logiciel qui permet à la machine d'analyser les requêtes d'un utilisateur (sous forme http), et de retourner le fichier correspondant à la requête (ou une erreur si le fichier n'est pas trouvé, ou la requête est mal formulée). Dans le cadre d'Apache, c'est donc du logiciel que l'on parle.



FIGURE IV.1 – Logo d'Apache

À l'heure actuelle, Apache est le serveur web le plus utilisé, avec environ 60 % de parts de marché. Apache possède même sa propre licence, utilisée par de nombreux autres projets. De plus, l'utilisation massive d'Apache (devenu le standard des serveurs web), couplée à sa forte popularité, à amener à une formidable abondance de documentation, de cours, et autres livres traitant de son utilisation, depuis l'installation jusqu'à la sécurisation.

Que ce soit pour la Raspberry Pi et Raspbian, ou pour une machine plus généraliste, Apache est donc un choix sûr, et les compétences que nous pourrions acquérir sur le sujet nous seront toujours utiles.

### IV.2.1.2 Installation d'Apache :

Avant d'installer le serveur, assurons nous d'avoir une machine bien à jour. Pour cela, on exécute les commandes suivantes : **sudo aptitude update** ensuite **sudo aptitude upgrade**.

Une fois le Raspberry Pi est mis à jour, nous allons installer le serveur Apache par la commande suivante : **sudo aptitude install apache2**.

Une fois l'installation terminée, nous pouvons tester qu'Apache fonctionne correctement en nous rendant sur l'adresse du Raspberry.

Pour cela, il faut tenter d'accéder au Raspberry depuis le port 80 (ce port n'étant pas encore ouvert depuis l'extérieur, il faudra le faire depuis le Raspberry lui-même). Il nous suffit d'ouvrir le navigateur web du Raspberry (par défaut Midori sous Raspbian), et d'aller à l'adresse « `http://127.0.0.1` ». On obtient alors une page avec un message du genre « It works ! » et plein d'autre texte.

Vous pouvez dès à présent utiliser votre Raspberry pour faire un site en HTML, CSS et JavaScript pur, en interne.

Cependant, nous voudrions sans doute rapidement permettre des interactions entre le site et l'utilisateur. Par exemple permettre à l'utilisateur de s'inscrire, etc. Pour cela, vous allez avoir besoin de PHP.

## IV.2.2 Installation de PHP sur le Raspberry

### IV.2.2.1 Qu'est-ce que PHP ?

Tout d'abord, il faut savoir que le PHP est un langage interprété. Et comme dans le cas des serveurs, l'acronyme PHP peut avoir plusieurs sens. En fait, quand l'on parle de PHP, on peut parler soit du langage, soit de l'interpréteur.

Ici, quand nous parlons d'installer PHP, cela signifie que nous allons installer l'interpréteur, afin d'utiliser le langage.

PHP (le langage cette fois) est principalement utilisé pour rendre un site dynamique, c'est-à-dire que l'utilisateur envoie des informations au serveur qui lui renvoie les résultats modifiés en fonction de ces infos. A contrario, 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.

PHP est libre, et maintenu par la fondation PHP, ainsi que l'entreprise Zend, et diverses autres entreprises (il est à noter que Zend est aussi l'auteur du célèbre framework Zend PHP, très utilisé et reconnu dans le monde de l'entreprise). PHP est un des langages de programmation les plus utilisés, et c'est même le plus utilisé pour la programmation destinée au web, avec environ 79 % de parts de marché .



FIGURE IV.2 – Logo PHP

#### IV.2.2.2 Installer PHP

On va encore une fois faire appel à l'administrateur pour installer PHP avec la ligne de commande : **sudo aptitude install php5**

Maintenant pour savoir si PHP fonctionne correctement, ce n'est pas très compliqué, et la méthode est relativement proche de celle employée pour Apache. Nous allons en premier lieu supprimer le fichier « index.html » dans le répertoire « /var/www/html » par la commande suivante :

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

Puis créez un fichier « index.php » dans ce répertoire, avec cette ligne de commande : **echo "<? 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, par taper l'adresse « http://127.0.0.1 ».

#### IV.2.2.3 PHP contrôle les GPIOs

Nous utilisons dans notre projet le PHP pour contrôler les entrées et sorties de notre maison intelligente. Pour cela, on héberge dans le serveur Apache dans le dossier /var/www/html/ nos scripts PHP. Chaque script a un rôle pour contrôler une entrée ou sortie reliée avec Raspberry Pi en l'occurrence les LEDs, les moteurs, les capteurs.

PHP est capable d'exécuter un script python grâce à la fonction **exec()**. Mais avant cela, il faut donner les droit d'exécution des scripts pythons par la commande suivante : **sudo chmod +x script.py**.

### IV.2.3 Une base de données MySQL pour notre serveur

#### IV.2.3.1 Un SGBD c'est quoi ? Pourquoi MySQL ?

Maintenant que nous avons mis en place PHP5, nous allons sans doute vouloir stocker les informations des comptes de nos utilisateurs qui vont utiliser l'application Android pour contrôler notre maison intelligente. Pour cela, on utilise le plus souvent des bases de données. Nous allons donc mettre en place un SGBD (Système de Gestion de Bases de Données), à savoir MySQL.

MySQL est un SGBD libre, puissant, massivement utilisé (environ 56 % de parts de marché des SGBD libres). Là encore, MySQL est tellement un incontournable du développement, quelle qu'en soit la branche, que vous devez absolument l'apprendre et le maîtriser.



FIGURE IV.3 – Logo MYSQL

### IV.2.3.2 Installer MySQL

Pour ce faire, nous allons installer `mysql-server` et `php5-mysql` (qui servira de lien entre php et mysql) par la commande : **`sudo aptitude install mysql-server php5-mysql`**

Lors de l'installation de `mysql-server`, il nous est demandé un mot de passe pour le compte administrateur MySQL (`root`), attention à bien le retenir, car il sera utilisé plus tard.

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

**`mysql –user=root –password=notre mot de passe`**

Nous devrions alors avoir un message du type « Bienvenue sur MySQL! ». Nous avons donc maintenant un serveur web, relié à PHP et à MySQL. Soit tout ce qu'il faut.

### IV.2.3.3 Gérer facilement les bases de données avec PHPMyAdmin

Toutefois, nous pourrions vouloir une interface un peu plus simple pour administrer nos bases de données qu'une simple console MySQL. Pour cela, nous pouvons installer PHPMyAdmin.

#### 1. PHPMyAdmin, c'est quoi ?

PHPMyAdmin est une application développée en PHP, et qui vise à fournir une interface simplifiée pour MySQL.

Elle nous permet, par exemple, de voir rapidement et de façon lisible le contenu de votre base de données, ou de la manipuler sans avoir besoin de faire vous-mêmes vos requêtes MySQL.

#### 2. Installer PHPMyAdmin

L'installation de PHPMyAdmin n'est pas du tout obligatoire. Nous ferons ici une installation sans paramètres de sécurité particuliers.

L'installation de PHPMyAdmin se fait très simplement, via le gestionnaire de paquets, en utilisant la commande suivante : **`sudo aptitude install phpmyadmin`**

PHPMyAdmin nous posera plusieurs questions concernant ses réglages, choisissons donc un serveur Apache, puis utilisons le choix par défaut pour la base (`dbconfig-common`). Pour le mot de passe `root`, il s'agit de celui que vous aviez utilisé pour MySQL.

Nous devons également modifier le fichier de configuration d'Apache pour inclure notre installation PHPMyAdmin. On l'ouvre avec le programme `nano` en tapant la commande suivante : **`sudo nano /etc/apache2/apache2.conf`**

Maintenant au bas de ce fichier entrons la ligne suivante :



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

Une fois terminé, nous sauvegardons et nous quittons en appuyant sur CTRL + X puis sur Y. Maintenant, redémarrons simplement le service Apache en entrant la commande suivante : **sudo /etc/init.d/apache2 restart**.

Pour vérifier le bon fonctionnement de PHPMyAdmin, nous allons simplement tenter d’y accéder, en utilisant l’adresse de notre Raspberry suivi de /phpmyadmin. Par exemple, en local ce sera « http://127.0.0.1/phpmyadmin ».

### IV.3 CONFIGURER LES COMPOSANTS AVEC RASPBERRY PI

#### IV.3.1 Les LEDs

Les ports GPIO du Raspberry peuvent fournir au max 3.3 V et un courant de 50 mA.

- LED rouge, orange, jaune (ambre) : 1.8V à 2V pour environ un courant de 20 mA.
- LED verte standard (vert clair) : 1.8V à 2.2V pour environ un courant de 20 mA.

Le calcul de la résistance sera le suivant :  $R = (U_{alim} - U_{led}) / I$  où :

- **R** correspond à la valeur de la résistance en Ohms ( $\Omega$ ).
- **U<sub>alim</sub>** correspond à la tension d’alimentation (dans notre cas se sera 3.3 V).
- **U<sub>led</sub>** correspond à la tension à fournir sur la LED.
- **I** correspond au courant nécessaire pour la LED, donc 0.02 A.

Donc, afin de pouvoir brancher ce montage sur une sortie GPIO du Raspberry, la formule sera la suivante :

$$R = (3.3 - 1.8) / 0.02 = 75 \Omega$$

Donc, notre résistance doit être supérieure à 75  $\Omega$ , alors dans notre projet nous avons pris une résistance R=100  $\Omega$ .

##### IV.3.1.1 LED rouge

On branche le pole positif de la LED avec la pin 11 du Raspberry et le pole négatif avec la masse qui est la pin 6 du Raspberry, comme montre le schéma suivant :

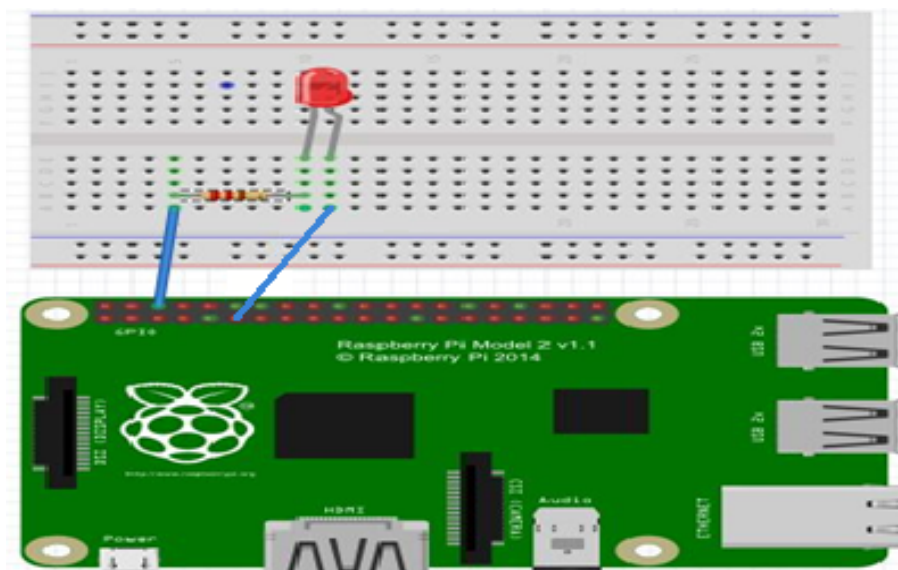


FIGURE IV.4 – Branchement LED rouge

#### 1. Code python pour allumer la led (rougeOn.py) :

```
1 | #!/usr/bin/python
```

```

2 # Import required Python libraries
3 import RPi.GPIO as GPIO
4 GPIO.setmode(GPIO.BOARD)
5 # init list with pin numbers
6 GPIO.setup(11, GPIO.OUT)
7 GPIO.output(11,1)
8

```

2. code python pour éteindre la led (rougeOff.py) :

```

1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(11, GPIO.OUT)
5 GPIO.output(11,0)
6

```

IV.3.1.2 LED vert

On la branche cette fois-ci avec la pin 12. Pour le programme python, il reste le même pour la LED rouge à l'exception du numéro de broche qui doit changer.

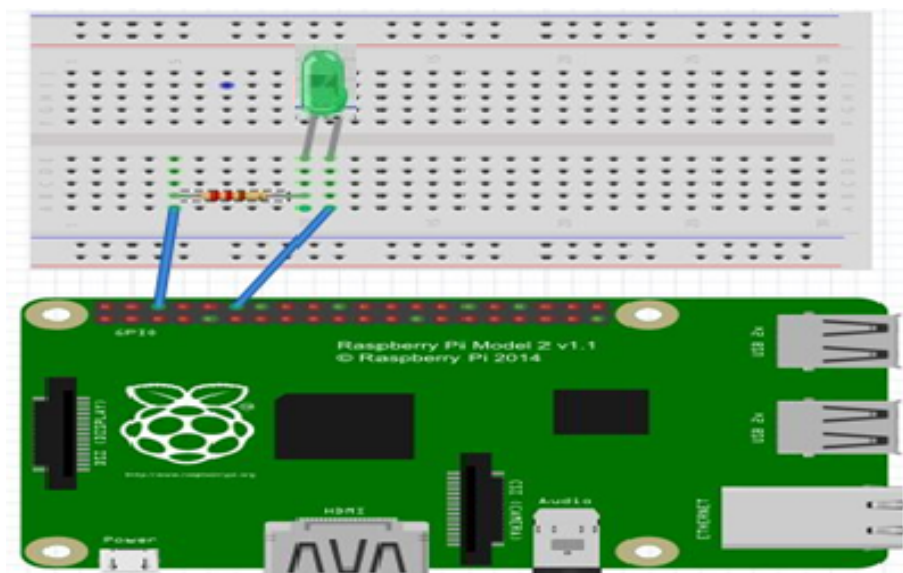


FIGURE IV.5 – Branchement LED vert

1. Code python pour allumer la led (vertOn.py) :

```

1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(12, GPIO.OUT)
5 GPIO.output(12,1)
6

```

2. Code python pour éteindre la led (vertOff.py) :

```

1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(12, GPIO.OUT)

```

```
5 GPIO.output(12,0)
6
```

### IV.3.1.3 LED jaune

On la branche cette fois-ci avec la pin 13. Pour le programme python, il reste le même comme pour la led rouge à l'exception du numéro de broche qui doit changer.

#### 1. code python pour allumer la led (jauneOn.py) :

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(13, GPIO.OUT)
5 GPIO.output(13,1)
6
```

#### 2. code python pour éteindre la led(jauneOff.py) :

```
1 #!/usr/bin/python
2 import RPi.GPIO as GPIO
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(13, GPIO.OUT)
5 GPIO.output(13,0)
6
```

### IV.3.1.4 Page php (led.php) de contrôle des leds à distance (WiFi)

Cette page PHP va pouvoir allumer ou éteindre la led choisie selon un caractère reçu de la part de notre application Android tel que chaque cas de contrôle correspond à un caractère bien précis. Par exemple, si on reçoit « rc », on exécute le programme python qui éteint la LED rouge grâce à la fonction prédéfinie en PHP « exec ».

```
1 <?php
2 if (isset($_POST["order"])){
3 $o=$_POST["order"];
4 if ($o=="vo")
5 {
6 exec("sudo python /home/pi/Desktop/code/led/greenOn.py");
7 }
8 if ($o=="vc" )
9 {
10 exec("sudo python /home/pi/Desktop/code/led/greenOff.py");
11 }
12 if ($o=="ro")
13 {
14 exec("sudo python /home/pi/Desktop/code/led/redOn.py");
15 }
16 if ($o=="rc" )
17 {
18 exec("sudo python /home/pi/Desktop/code/led/redOff.py");
19 }
20 if ($o=="jo")
21 {
22 exec("sudo python /home/pi/Desktop/code/led/yellowOn.py");
23 }
24 if ($o=="jc" )
25 {
```

```

26 | exec("sudo python /home/pi/Desktop/code/led/yellowOff.py");
27 | }
28 | }
29 | ?>
    
```

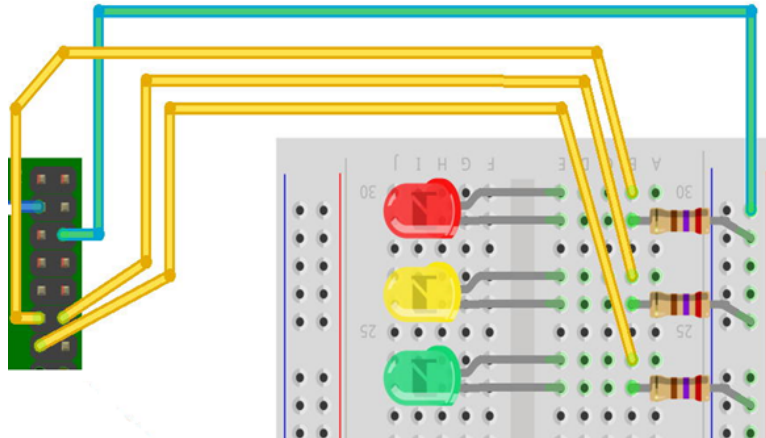


FIGURE IV.6 – Circuit LEDs avec Raspberry

### IV.3.2 Les moteurs

Commençons par une mise en garde : il est hors de question de brancher directement le moteur aux pins GPIO du Raspberry Pi. Les courants assez intenses qui sont nécessaires pour faire tourner le moteur pourraient s'avérer très néfastes pour la santé de notre Raspberry Pi. Nous allons plutôt contrôler le moteur par l'intermédiaire d'un circuit intégré spécialement conçu pour ce genre d'utilisation : le L293D (le pont H).



FIGURE IV.7 – Pont H

Grâce à ce circuit intégré, l'intensité du courant traversant les fragiles pins du Raspberry Pi sera beaucoup plus faible que celle qui traversera le moteur. De plus, un moteur en rotation agit un peu comme un générateur de courant : le L293D comporte des diodes intégrées qui protégeront notre Raspberry Pi contre ce courant parasite.

Et voici le brochage du L293D :

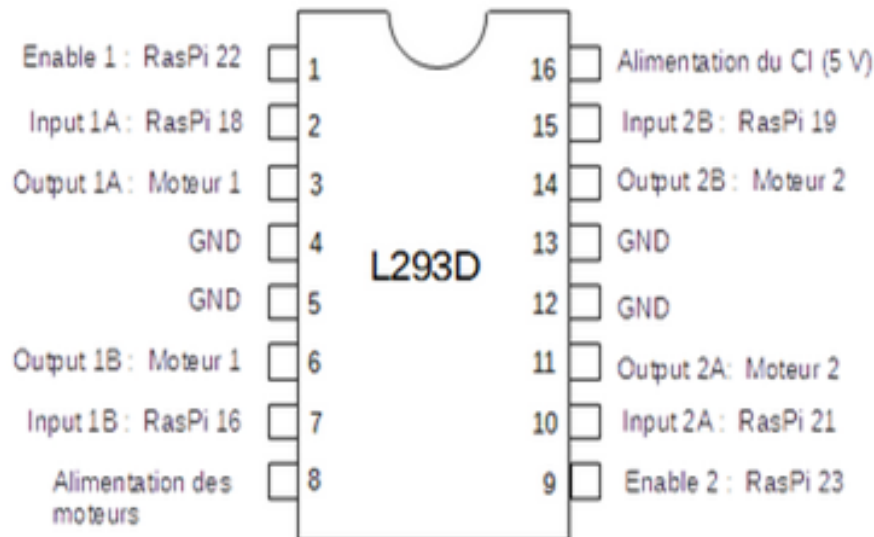


FIGURE IV.8 – Brochage du Pont H

Les pins 4, 5, 12 et 13 du L293D doivent toutes être reliées à la masse (les pins GND) du Raspberry Pi (la pin numéro 6, par exemple). Comme tous les circuits intégrés, le L293D doit être alimenté, et c'est la pin 16 qui est désignée pour recevoir cette alimentation. Une des sorties 5 V du Raspberry Pi (la pin numéro 2, par exemple) fera parfaitement l'affaire. Cette tension sert uniquement à alimenter le circuit intégré : elle n'est pas transmise au moteur.

Pour alimenter le moteur, on utilise la pin 8 du L293D. Mais attention : il n'est pas du tout conseillé d'utiliser les pins du Raspberry Pi pour cette fonction. Nous allons donc utiliser une alimentation autonome comme, par exemple, des piles (la tension de votre assemblage de piles sera choisie de façon à ne pas dépasser ce que peut tolérer votre moteur). La borne positive de l'alimentation du moteur doit être reliée à la pin 8 du L293D, et la borne négative doit être reliée à la masse (du L293D et du Raspberry Pi).

On peut brancher deux moteurs DC à un même L293D : les pins situés du côté gauche du CI sont réservées au premier moteur, et les pins du côté droit sont pour le deuxième (s'il y a lieu).

#### IV.3.2.1 Ventilateur

Notre ventilateur est un moteur DC avec 2 fils. Ce moteur lui-même se branche aux pins 3 et 6 du L293D. La pin 1 du L293D est la pin "enable" (activation) du premier moteur. Le moteur tournera à sa vitesse maximale si cette pin est soumise au signal logique "HAUT", et il ne tournera pas si elle se trouve au signal logique "BAS". Si on veut que le moteur tourne à une vitesse réduite, il s'agit de soumettre cette pin à un signal qui alterne rapidement entre "HAUT" et "BAS" : c'est ce qu'on appelle la modulation de largeur d'impulsion, ou PWM pour Pulse Width Modulation. Relions donc la pin 1 du L293D à la pin 22 du Raspberry Pi.



FIGURE IV.9 – Ventilateur

Il reste deux autres pins du L293D à raccorder au Raspberry Pi : le moteur tournera dans un sens lorsque la pin 2 sera soumise au signal logique "HAUT" et la pin 7 sera soumise au signal logique "BAS", et tournera dans l'autre sens si la pin 2 est à "BAS" pendant que la pin 7 est à "HAUT". Branchant la pin 2 du L293D à la pin 18 du Raspberry Pi, et la pin 7 du L293D à la pin 16 du Raspberry Pi.

Voici le schéma du circuit :

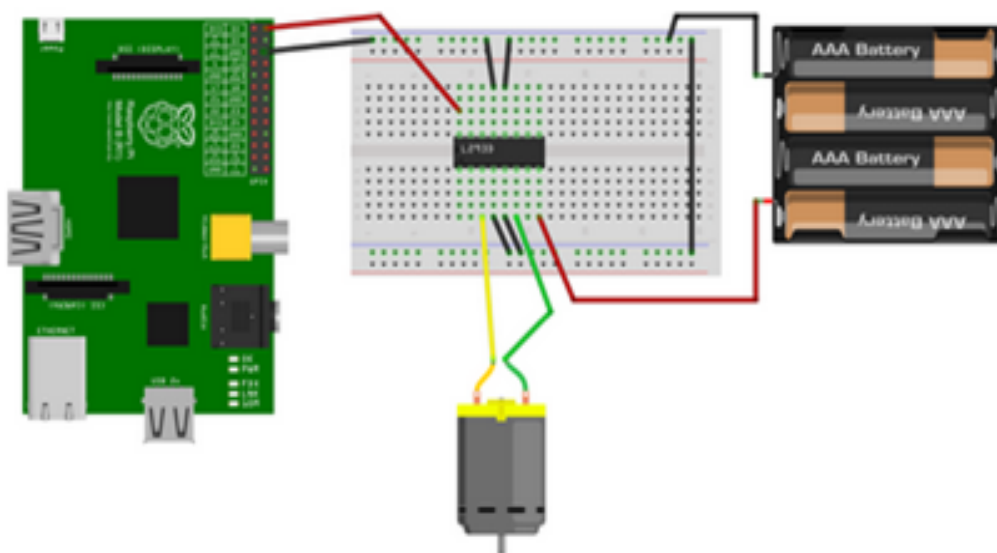


FIGURE IV.10 – Circuit ventilateur avec Raspberry

1. Programme python qui lance le ventilateur (fanStart.py) :

```

1 import RPi.GPIO as GPIO
2 from time import sleep
    
```

```

3  GPIO.setmode(GPIO.BOARD)
4  GPIO.cleanup()
5  Moteur1A = 16    # premiere sortie du premier moteur, pin 16
6  Moteur1B = 18    # deuxieme sortie de premier moteur, pin 18
7  Moteur1E = 22    # enable du premier moteur, pin 22
8  GPIO.setup(Moteur1A,GPIO.OUT)
9  GPIO.setup(Moteur1B,GPIO.OUT)
10 GPIO.setup(Moteur1E,GPIO.OUT)
11 GPIO.output(Moteur1E,GPIO.HIGH)
12 GPIO.output(Moteur1B,GPIO.HIGH)
13 GPIO.output(Moteur1A,GPIO.LOW)
14

```

2. Programme python qui arrête le ventilateur (fanStop.py) :

```

1  import RPi.GPIO as GPIO
2  from time import sleep
3  GPIO.setmode(GPIO.BOARD)
4  GPIO.cleanup()
5  Moteur1A = 16    # premiere sortie du premier moteur, pin 16
6  Moteur1B = 18    # deuxieme sortie de premier moteur, pin 18
7  Moteur1E = 22    # enable du premier moteur, pin 22
8  GPIO.setup(Moteur1A,GPIO.OUT)
9  GPIO.setup(Moteur1B,GPIO.OUT)
10 GPIO.setup(Moteur1E,GPIO.OUT)
11 GPIO.output(Moteur1E,GPIO.LOW)
12 GPIO.cleanup()
13

```

3. Script php qui contrôle le ventilateur a distance(fan.php) :

```

1  <?php
2  if (isset($_POST["order"])){
3  $o=$_POST["order"];
4  if ($o=="fo")
5  {exec("sudo python /home/pi/Desktop/code /fan/fanStart.py");}
6  if ($o=="fc" )
7  {exec("sudo python /home/pi/Desktop/code/fan/fanStop.py");}
8  ?>
9

```

IV.3.2.2 La porte

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

Ce moteur uni-polaire dispose de deux emplacements/trous de montage solidaires du moteur. Le moteur ne dispose pas seulement de 32 pas (11.25 degrés/pas) par révolution mais il intègre un engrenage réducteur ayant un 1/16 (1/16.032 pour être précis).

Cela signifie que ce moteur dispose vraiment de 32\*16.032 pas par révolution = 513 pas !

Nous allons ajouter ce deuxième moteur pas à pas. Le moteur se branche aux pins 11 et 14 du L293D. On peut ensuite brancher la pin 9 du L293D à la pin 23 du Raspberry Pi, la pin 10 du L293D à la pin 21 du Raspberry Pi, et la pin 15 du L293D à la pin 19 du Raspberry Pi.

Voici le circuit :



FIGURE IV.11 – Moteur pas à pas

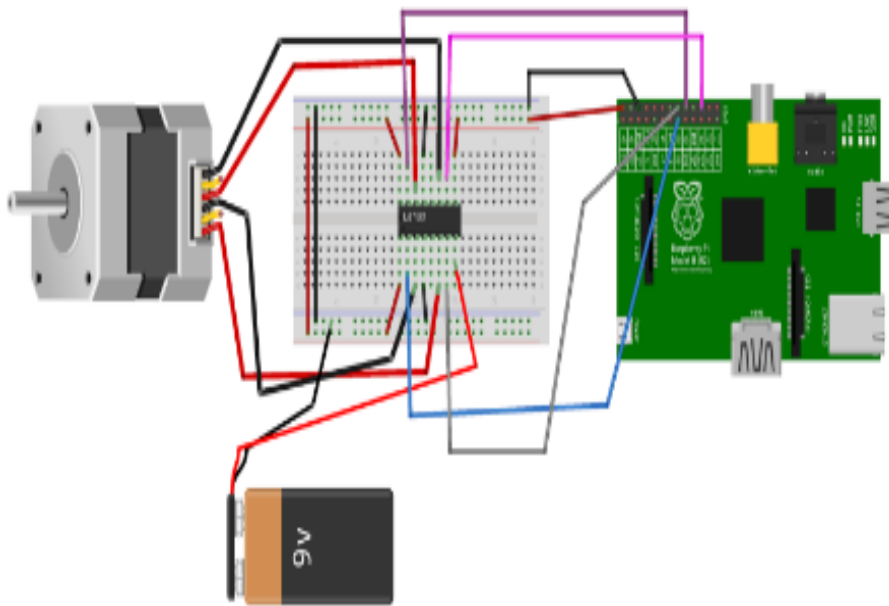


FIGURE IV.12 – Circuit moteur pas à pas avec Raspberry

- La pin 1 du L293D est branchée au 5 V du Raspberry Pi
- La pin 2 du L293D est branchée à la pin 15 du Raspberry Pi
- La pin 3 du L293D est branchée à une extrémité du premier bobinage du moteur
- La pin 4 ou la pin 5 du L293D est branchée à la masse (GND) du Raspberry Pi
- La pin 6 du L293D est branchée à l'autre extrémité du premier bobinage du moteur
- La pin 7 du L293D est branchée à la pin 29 du Raspberry Pi
- La pin 8 du L293D est branchée à la borne positive de l'alimentation du moteur (la borne négative de cette alimentation est reliée à la masse GND).
- La pin 9 du L293D est branchée au 5 V du Raspberry Pi
- La pin 10 du L293D est branchée à la pin 31 du Raspberry Pi
- La pin 11 du L293D est branchée à l'extrémité du deuxième bobinage du moteur
- La pin 12 ou la pin 13 du L293D est branchée à la masse (GND) du Raspberry Pi
- La pin 14 du L293D est branchée à l'autre extrémité du deuxième bobinage du moteur
- La pin 15 du L293D est branchée à la pin 32 du Raspberry Pi
- La pin 16 du L293D est branchée au 5 V du Raspberry Pi

1. Programme python qui ouvre la porte (doorOpen.py) :

```

1  #!/usr/bin/env python
2  import RPi.GPIO as GPIO
    
```



```

3 import time
4 GPIO.setmode(GPIO.BOARD)
5 pin_bobine_1_1 = 15
6 pin_bobine_1_2 = 29
7 pin_bobine_2_1 = 32
8 pin_bobine_2_2 = 31
9 # configuration des pins en sortie
10 GPIO.setup(pin_bobine_1_1, GPIO.OUT)
11 GPIO.setup(pin_bobine_1_2, GPIO.OUT)
12 GPIO.setup(pin_bobine_2_1, GPIO.OUT)
13 GPIO.setup(pin_bobine_2_2, GPIO.OUT)
14 def marche_arriere(attente, nombre_de_pas):
15     for i in range(0, nombre_de_pas):
16         prochainStep(3 - (i % 4))
17         time.sleep(attente)
18     def reglage_pins(pin1, pin2, pin3, pin4):
19         GPIO.output(pin_bobine_1_1, pin1)
20         GPIO.output(pin_bobine_1_2, pin2)
21         GPIO.output(pin_bobine_2_1, pin3)
22         GPIO.output(pin_bobine_2_2, pin4)
23     def prochainStep(numero):
24         if (numero == 0):
25             reglage_pins(1, 0, 1, 0)
26         if (numero == 1):
27             reglage_pins(0, 1, 1, 0)
28         if (numero == 2):
29             reglage_pins(0, 1, 0, 1)
30         if (numero == 3):
31             reglage_pins(1, 0, 0, 1)
32         attente = 20
33         nombre_de_pas -= 160
34         marche_arriere(int(attente) / 1000.0, -int(nombre_de_pas))
35     GPIO.cleanup()
36

```

## 2. Programme python qui ferme la porte :

```

1 #!/usr/bin/env python
2 import RPi.GPIO as GPIO
3 import time
4 GPIO.setmode(GPIO.BOARD)
5 pin_bobine_1_1 = 15
6 pin_bobine_1_2 = 29
7 pin_bobine_2_1 = 32
8 pin_bobine_2_2 = 31
9 # configuration des pins en sortie
10 GPIO.setup(pin_bobine_1_1, GPIO.OUT)
11 GPIO.setup(pin_bobine_1_2, GPIO.OUT)
12 GPIO.setup(pin_bobine_2_1, GPIO.OUT)
13 GPIO.setup(pin_bobine_2_2, GPIO.OUT)
14 def marche_avant(attente, nombre_de_pas):
15     for i in range(0, nombre_de_pas):
16         prochainStep(i % 4)
17         time.sleep(attente)
18     def reglage_pins(pin1, pin2, pin3, pin4):
19         GPIO.output(pin_bobine_1_1, pin1)
20         GPIO.output(pin_bobine_1_2, pin2)
21         GPIO.output(pin_bobine_2_1, pin3)
22         GPIO.output(pin_bobine_2_2, pin4)
23     def prochainStep(numero):
24         if (numero == 0):
25             reglage_pins(1, 0, 1, 0)
26         if (numero == 1):

```

```

27  reglage_pins(0, 1, 1, 0)
28  if (numero == 2):
29  reglage_pins(0, 1, 0, 1)
30  if (numero == 3):
31  reglage_pins(1, 0, 0, 1)
32  attente =20
33  nombre_de_pas =160
34  marche_avant(int(attente) / 1000.0, int(nombre_de_pas))
35  GPIO.cleanup()
36

```

### 3. Script php qui contrôle la porte à distance (door.php) :

```

1
2  <?php
3  if (isset($_POST["order"]) )
4  {$o=$_POST["order"];
5  if ($o=="do"){exec("sudo python /home/pi/Desktop/code/door/doorOpen.py");}
6  if ($o=="dc" )
7  {exec("sudo python /home/pi/Desktop/code/door/doorClose.py");}
8  ?>
9

```

### IV.3.3 Le capteur de température

Dans notre maison intelligente, on souhaite surveiller la température temporaire. Pour cela, je m'aide d'un Raspberry Pi en WiFi et d'une sonde 1-wire **DS18B20**.

Cette sonde utilise le bus 1-wire et se connecte directement sur le GPIO en 3,3v et peut mesurer une température allant de -55 °C à +125 °C, avec une précision de ± 0,5 °C (entre -10 °C et +85 °C). La résolution programmable est de 9 à 12 bits en utilisant une résistance de 4.7kΩ qui fait très bien l'affaire.

Voici le schéma de câblage d'un DS18B20 :



FIGURE IV.13 – Câblage de la sonde DS18B20 avec Raspberry

#### IV.3.3.1 La configuration du 1-wire

Souvent, le Raspberry ne détecte pas la sonde de température. Pour qu'il la détecte, il faut faire des ajouts dans deux fichiers système :

Ajouter les lignes suivantes au fichier qui se trouve dans /etc/modules :

**w1-therm**

**w1-gpio pullup=1**

Ajouter la ligne suivante au fichier qui se trouve dans /boot/config.txt :

**dtoverlay=w1-gpio**

Pour une action immédiate, on tape les commandes suivantes :

**sudo modprobe w1-gpio**

**sudo modprobe w1-therm**

Pour récupérer l'ID de notre sonde, on tape la commande suivante :

**ls /sys/bus/w1/devices/**

**Résultat : 28-00141389deff**

Et voilà **28-00141389deff** correspond à l'ID de notre sonde DS18B20. On va récupérer la température avec PHP en utilisant le script suivant :

### 1. Script python temper.py :

```

1  import os,time
2  temp_sensor='/sys/bus/w1/devices/28-00141389deff/w1_slave'
3  def temp_raw():
4  f=open(temp_sensor,'r')
5  lines = f.readlines()
6  f.close()
7  return lines
8  def read_temp():
9  lines = temp_raw()
10 while lines[0].strip()[-3:] != 'YES':
11     time.sleep(0.2)
12     lines = temp_raw()
13     temp_output = lines[1].find("t=")
14     if temp_output != -1:
15         temp_string = lines[1].strip()[temp_output+2:]
16         temp_c = float(temp_string) / 1000.0
17         return temp_c
18     while True:
19         temp=read_temp()
20         time.sleep(1)
21         if temp > 20:
22             print('warning !temperature at home is high')
23             os.system('sudo python /home/pi/Desktop/code/fan/fanStart.py')
24         else :
25             os.system('sudo python /home/pi/Desktop/code/fan/fanStop.py')
26

```

### 2. Script php temp.php :

```

1  <?php
2  // Fichier à lire
3  $file = "/sys/bus/w1/devices/28-00141389deff/w1_slave";
4  // Lecture ligne par ligne
5  $lines = file($file);
6  // R'\{e}cup\{e}re la 2nd ligne
7  $temp = explode(' ', $lines[1]);
8  // Formatage de la temp\{e}rature
9  $temp = number_format($temp[1]/1000,2, '.', '');
10 // On affiche la temp\{e}rature sous format json
11 $reponse=array();
12 $reponse["temperature"]=$temp;
13 echo json_encode($reponse);
14 ?>
15

```

### IV.3.4 Le détecteur de gaz

Les **capteurs MQ** sont des capteurs physicochimiques permettant de détecter une grande variété des gaz, polluants et fumées dans l'atmosphère. Le MQ-2 est un capteur qui permet de détecter du gaz ou de fumée à des concentrations de 300 ppm à 10000 ppm. Après calibration, le MQ-2 peut détecter différents gaz comme le GPL (LPG), l'isobutane, le propane, le méthane, l'alcool, l'hydrogène ainsi que les fumées. Il est conçu pour un usage intérieur à température ambiante.

Alimenté en 5V, il dispose d'une sortie analogique et d'un réglage de la sensibilité par potentiomètre. Le détecteur de fumée MQ-2 déclare la fumée par le niveau de tension qu'elle produit. Plus il y a de fumée, plus la tension est élevée. À l'inverse, moins la fumée à laquelle elle est exposée, moins elle émet de tension. Le MQ-2 dispose également d'un potentiomètre intégré pour ajuster la sensibilité à la fumée.



FIGURE IV.14 – Capteur MQ-2

La broche de sortie numérique est reliée avec la pin 38 du Raspberry, elle est true s'il y a du gaz détecté et automatiquement la porte s'ouvre par mesure de sécurité et elle est false dans le cas contraire.

#### IV.3.4.1 Script python MQ-2.py :

Ce script python est un système de sécurité pour notre maison intelligente de telle sorte que si le capteur détecte du gaz alors la porte s'ouvre automatiquement.

```

1 import time, sys
2 import RPi.GPIO as GPIO
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setup(38, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
5 def action(pin):
6 fichier = open("/home/pi/Desktop/gaz.txt","w")
7 fichier.write("Sensor detected smoke")
8 fichier.close()
9 os.system("sudo python /home/pi/Desktop/code/door/doorOpen.py")
10 return
11 GPIO.add_event_detect(38, GPIO.RISING)
12 GPIO.add_event_callback(38, action)
13 try:
14 while True:
15 time.sleep(2)
16 except KeyboardInterrupt:
17 GPIO.cleanup()
18 sys.exit()

```

#### IV.3.4.2 Script php :

```
1 <?php
2 // Fichier à lire
3 $monfichier = fopen('/home/pi/Desktop/gaz.txt', 'r+');
4 //récupérer la ligne
5 $ligne = fgets($monfichier);
6 //effacer contenu
7 ftruncate($monfichier, 0);
8 //fermer
9 fclose($monfichier);
10 // On code le message sous format json
11 $reponse=array();
12 $reponse["gaz"]=$ligne;
13 echo json_encode($reponse);
14 ?>
```

#### IV.3.5 la vidéo-surveillance avec Raspberry Pi :

On essaie dans notre projet faire la surveillance vidéo pour la sécurité de notre maison intelligente accessible depuis partout. Pour ce faire, on aura besoin de notre carte Raspberry Pi, d'une connexion Internet et d'une webcam. Pour réaliser tout ça, on a besoin d'installer le logiciel Motion pour Raspbian, qui se chargera de transformer notre Raspberry Pi en véritable serveur webcam.

##### IV.3.5.1 Définition :

Motion est un logiciel très simple d'utilisation, disponible pour Raspbian et parfaitement adapté à la vidéo-surveillance avec le Raspberry Pi.

La fonction première de Motion est de faire de la vidéo-surveillance avec diffusion en ligne, en temps réel, via une ou plusieurs caméras ! Motion permet également de faire de la « motion capture », c'est-à-dire, que Motion est capable de comparer la dernière image prise avec la nouvelle pour faire de la détection de mouvements.

##### IV.3.5.2 Installation de motion sur le Raspberry Pi :

Maintenant que vous savez ce qu'est Motion, nous allons voir comment l'installer sur votre Raspberry Pi pour le transformer en caméra de surveillance.

Dans un premier temps, on s'assure que le Raspberry Pi est bien à jour à l'aide des deux commandes suivantes :

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

Une fois ces commandes terminées et le Raspberry Pi à jour, installons maintenant Motion. Il nous suffit alors d'utiliser la ligne de commande suivante :

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

##### IV.3.5.3 Configurer Motion sur notre Raspberry Pi :

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

Dans un premier temps, assurons-nous que Motion tourne en permanence. Pour cela, éditez le fichier « /etc/default/motion » et modifiez la ligne « start\_motion\_daemon » en :

```
start_motion_daemon=yes
```

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

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

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

- On active le daemon : **daemon on**
- On utilise le fichier /tmp/motion.log pour stocker les logs :  
**logfile /tmp/motion.log**
- On règle la résolution (adapter la résolution à celle de votre caméra. Ici, nous avons réglé pour la caméra du Raspberry pi) :  
**width 1280**  
**height 720**
- On règle le nombre d'images prises par seconde : **framerate 2**
- On règle le nombre d'images enregistrées avant et après un mouvement (ici 2 avant 2 après) : **pre\_capture 2 post\_capture 2**
- On règle le code, pour une lecture sur de nombreux médias :  
**ffmpeg\_video\_codec msmpeg4**
- On autorise la lecture du flux depuis internet pour permettre l'accès en ligne à la vidéo-surveillance : **stream\_localhost off**

Enfin, nous avons donné les droits nécessaires au fichier pour que le Raspberry Pi puisse les exécuter :

```
sudo chmod 777 /etc/motion.conf // Tout le monde peut lire, écrire et exécuter
```

Voilà, notre Raspberry Pi est maintenant équipé pour faire un système de vidéo-streaming efficace. Par défaut, Motion diffuse la vidéo sur le port 8081.

Pour accéder à la surveillance vidéo de notre Raspberry Pi en local host, on tape l'adresse `http://127.0.0.1:8081`, tandis que si on veut accéder au réseau local, on tape l'adresse IP de notre Raspberry Pi `192.168.1.20`, ou encore via Internet grâce au DDNS.

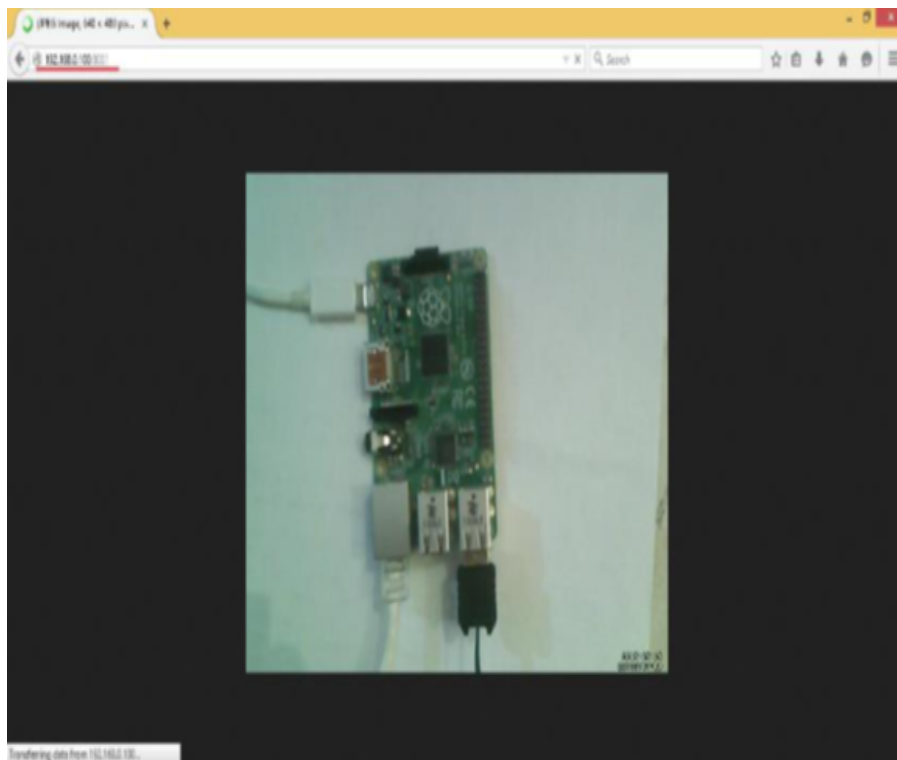


FIGURE IV.15 – Teste de camera de surveillance

## IV.4 CONCLUSION

Ce chapitre était un véritable test pour nous dans le domaine des systèmes embarqués là où nous avons appris comment contrôler les actionneurs, les capteurs, les LEDs grâce au mini-pc Raspberry.

On a vu qu'on peut faire de la vidéo surveillance grâce au Motion qui rend notre carte Raspberry un vrai serveur webcam.

Dans le chapitre suivant, on verra comment contrôler notre maison intelligente à distance avec une application Android via WiFi.

# CONTRÔLE DE LA MAISON INTELLIGENTE



## SOMMAIRE

V.1	COMMUNICATION CLIENT/SERVEUR . . . . .	68
V.1.1	Création de la BDD . . . . .	68
V.1.2	Création de la table admin . . . . .	68
V.1.3	Création de la table user . . . . .	68
V.2	CONCEPTION D'APPLICATION . . . . .	69
V.2.1	Projet de l'application . . . . .	69
V.2.2	Présentation des interfaces graphique et classes java . . . . .	70
V.3	CONCEPTION DU SITE . . . . .	78
V.3.1	Script de configuration (db.php) . . . . .	79
V.3.2	Page de démarrage du site (index.php) . . . . .	79
V.3.3	Page d'accueil (acceuil.php) . . . . .	80
V.3.4	Page d'affichage de température (sensor.php) . . . . .	81
V.3.5	Page de streaming vidéo . . . . .	82
V.3.6	Page de contrôle de porte (door.php) . . . . .	82
V.3.7	Page de contrôle des LEDs (light.php) . . . . .	83
V.3.8	Page de contrôle du ventilateur (fan.php) . . . . .	84
V.4	PRÉSENTATION D'ANDROID . . . . .	85
V.4.1	La création d'Android . . . . .	85
V.4.2	Les avantages de Google Android . . . . .	86
V.4.3	Les inconvénients de Google Android . . . . .	86
V.4.4	Versions d'Android . . . . .	86
V.5	PHP ET MySQL . . . . .	87
V.5.1	Page de configuration (db_config.php) . . . . .	87
V.5.2	Script d'authentification des utilisateurs (authentification.php) . . . . .	87
V.6	CONCEPTION DE NOTRE APPLICATION ANDROID . . . . .	88
V.6.1	Répertoire complet de l'application . . . . .	88
V.6.2	Fichier AndroidManifest.xml . . . . .	88
V.6.3	Interface authentification . . . . .	89
V.6.4	Interface accueil (welcome.xml) . . . . .	91
V.6.5	Interface de contrôle (dashboard_layout.xml) . . . . .	91
V.6.6	Interface capteur . . . . .	92
V.6.7	Interface camera (camera.xml) . . . . .	93
V.6.8	Interface portail . . . . .	94
V.6.9	Interface light . . . . .	95



V.6.10 Interface fan . . . . .	97
V.7 CONCLUSION . . . . .	98

**C**E chapitre est destiné pour la création des applications desktop, Android, Site web PHP qui servent à contrôler notre maison intelligente à distance via WiFi.

Ce chapitre est composé de trois parties. Dans la première partie, on verra comment contrôler la maison intelligente à travers une application desktop. Quant au deuxième et troisième parties, elles sont réservées respectivement au contrôle de notre maison intelligente via un site web et une application android.

On verra aussi comment administrer les comptes des utilisateurs : création, mise à jour, suppression et recherche.

Nous allons illustrer donc les différentes interfaces graphiques de notre application ainsi quelques codes java et Android importants.

Nous allons illustrer dans les différentes pages du site avec le code source PHP de chaque page.

## V.1 COMMUNICATION CLIENT/SERVEUR

Le terme client/serveur décrit une relation possible entre deux applications logicielles dans lequel le client effectue une demande de service du serveur et le serveur lui donne ce service.

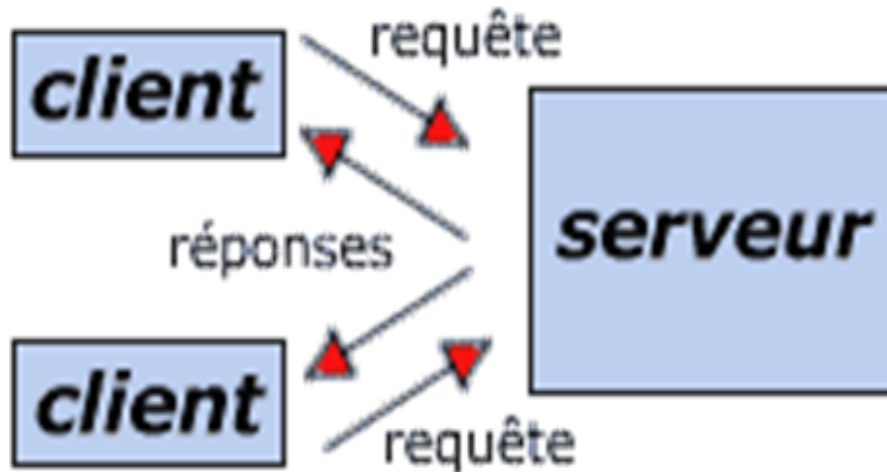


FIGURE V.1 – Architecture client serveur

Dans notre projet, il s’agit bien d’une application Desktop cliente qui demande un service au serveur web Apache installé sur Raspberry qui consiste à allumer une LED, avoir la température de la maison intelligente, ouvrir la porte etc..

Pour cela, nous allons héberger des scripts PHP qu’on en a déjà abordé dans le chapitre précédent dans notre serveur Apache dans le dossier « `/var/www/html/smarthouse/desktopApp/` ».

Nous allons aussi utiliser les bases de données avec le SGBD MySQL. Nous allons donc créer deux tables dans notre base de données une pour l’administrateur et une autre pour les utilisateurs.

On utilisera la fonction de hachage md5 pour enregistrer les condensés des mots de passes.

### V.1.1 Création de la BDD

Nous allons créer une base de données nommée smarthouse en exécutant la requête suivante sur le SGBD MySQL :`create database 'smarthouse' ;`

### V.1.2 Création de la table admin

```

1 USE 'smarthouse';
2 CREATE TABLE IF NOT EXISTS 'admin' (
3 'id' int(3) NOT NULL AUTO_INCREMENT, 'username' varchar(50) NOT NULL, 'password' varchar
  (100) NOT NULL, PRIMARY KEY ('id') );

```

### V.1.3 Création de la table user

```

1 USE 'smarthouse';
2 CREATE TABLE IF NOT EXISTS 'user' (
3 'id' int(3) NOT NULL AUTO_INCREMENT, 'username' varchar(50) NOT NULL, 'password' varchar
  (100) NOT NULL, PRIMARY KEY ('id') );

```

## V.2 CONCEPTION D'APPLICATION

Notre application desktop est gérée par l'administrateur de la maison intelligente celui qui va créer des comptes pour tous les utilisateurs qui veulent contrôler notre Smarthouse via le web ou via Android qu'on en parlera dans ce chapitre.

### V.2.1 Projet de l'application

Notre application desktop est développée par java sous l'IDE netbeans. Voici notre projet Smarthouse dans la figure suivante :

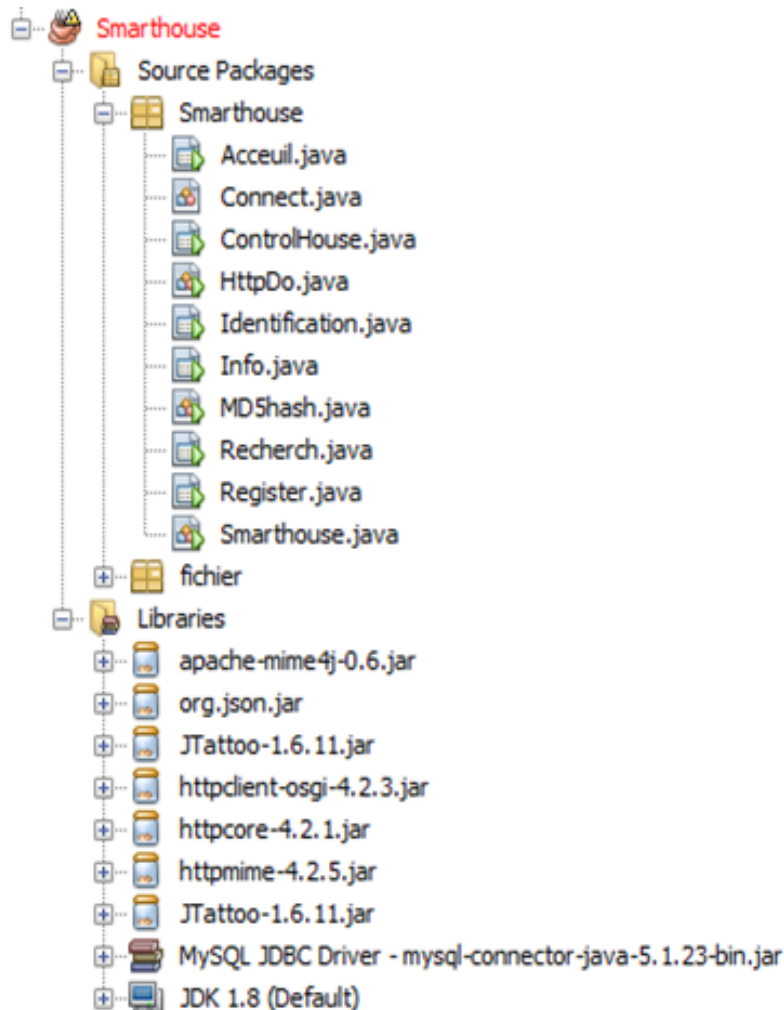


FIGURE V.2 – *Projet Smarthouse*

Donc nous avons deux package : le package Smarthouse contient les classes java et l'autre package fichier contient les images et icones utilisés dans notre application. Pour le dossier Librairies c'est là où on importe les différentes bibliothèques nécessaires pour notre application :

- Les bibliothèques apache-mime4j-0.6.jar et org.json.jar httpclient-osgi-4.2.3.jar httpcore-4.2.1.jar httpmime-4.2.5.jar sont utilisés pour assurer le bon fonctionnement de protocole http pour communiquer à distance avec notre carte Raspberry.
- La bibliothèque MySQL JDBC Driver est utilisée pour manipuler les bases de données.
- Pour la dernière bibliothèque Jtatto-1.6.11.jar nous l'avons importé pour améliorer l'apparence de notre application en choisissant un thème par la méthode prédéfi-

nie en java suivante : `UIManager.setLookAndFeel("com.jtattoo.plaf.aero.AeroLookAndFeel");`

## V.2.2 Présentation des interfaces graphique et classes java

### V.2.2.1 la classe Smarthouse.java

C'est la classe principale qui inclut la méthode main qui lance l'application.

```

1 public class Smarthouse {
2     public static void main(String[] args) throws ClassNotFoundException,
        InstantiationException, IllegalAccessException,
        UnsupportedLookAndFeelException {
3     try {
4         UIManager.setLookAndFeel("com.jtattoo.plaf.aero.AeroLookAndFeel");
5     } catch (ClassNotFoundException | InstantiationException | IllegalAccessException |
        UnsupportedLookAndFeelException ex) {}
6     new Identification().setVisible(true);}

```

Dans cette classe, on applique notre merveilleux thème graphique pour toute l'application et aussi on appelle l'interface graphique d'authentification qui est Identification.java

### V.2.2.2 la classe MD5hash.java

Cette classe est chargée de hacher les mots de passe d'administrateur et des utilisateurs qui seront enregistrés dans la base de données ou utilisés pour l'authentification des comptes.

```

1 public class MD5hash {
2     public static String getMD5Hash(String string){
3     try {
4         MessageDigest md5 = MessageDigest.getInstance("MD5");
5         md5.update(string.getBytes());
6         byte[] digest = md5.digest();
7         string = byteArrayToHexString(digest);}
8     catch (NoSuchAlgorithmException e1) {
9         e1.printStackTrace(); } return string;}
10    private static String byteArrayToHexString(byte[] bArr){
11        StringBuffer sb = new StringBuffer();
12        for (int i = 0; i < bArr.length; i++) {
13            int unsigned = bArr[i] & 0xff;
14            if (unsigned < 0x10)
15                sb.append("0");
16            sb.append(Integer.toHexString(unsigned)); }
17        return sb.toString();}

```

### V.2.2.3 la classe Connect.java

Cette classe est chargée d'établir une connexion avec notre base de données à distance qui se trouve sur le serveur MySQL Raspberry.

On charge le pilote par la méthode `Class.forName()` ensuite on établit la connexion avec la méthode `DriverManager.getConnection()` tout en précisant l'url de base de données en l'occurrence `jdbc:mysql://192.168.1.20:3306/smarthouse` nom d'utilisateur et mot de passe du serveur MySQL.

```

1 public class Connect {
2     String urlPilote="com.mysql.jdbc.Driver" ;
3     String urlBD = "jdbc:mysql://192.168.1.20:3306/smarthouse";
4     static Connection con;

```

```

5 public Connect(){
6   try{
7     Class.forName(urlPilote);
8     System.out.println("chargement du pilote");
9   } catch(ClassNotFoundException e){
10    System.out.println(e);}
11   try{
12     con=DriverManager.getConnection(urlBD, "root", "111");
13     System.out.println("connexion a la base de donnees");
14   } catch(SQLException ex){
15     System.out.println(ex);}
16   Connection ObttenirConnexion(){
17     return con;}

```

#### V.2.2.4 la classe HttpDo.java

Cette classe est chargée de l'envoi des requêtes http via la méthode POST ou GET de ce protocole. C'est grâce à cette classe qu'on va communiquer avec notre serveur web et donner l'ordre au serveur PHP d'exécuter les scripts python qu'ils s'agissent d'allumer une LED ou ouvrir une porte par exemple.

```

1 public class HttpDo {
2   static String url;static String resultat;
3   static private final String USER_AGENT = "Mozilla/5.0";static String json = "";
4   private static final String TAG_TEMPERATURE= "temperature";
5   static int success;static String temp;
6   static public void sendGet(String ur) throws Exception {
7     url=ur;
8     URL obj = new URL(url);
9     HttpURLConnection con = (HttpURLConnection) obj.openConnection();
10    con.setRequestMethod("GET");
11    con.setRequestProperty("User-Agent", USER_AGENT);
12    int responseCode = con.getResponseCode();
13    BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
14    String inputLine;
15    StringBuffer response = new StringBuffer();
16    while ((inputLine = in.readLine()) != null)
17    {
18      response.append(inputLine);}
19    in.close();
20    resultat=response.toString();
21  }
22  static public void sendPost(String url,String ord) throws Exception {
23    HttpClient client = new DefaultHttpClient();
24    HttpPost post = new HttpPost(url);
25    post.setHeader("User-Agent", USER_AGENT);
26    List<NameValuePair> urlParameters = new ArrayList<NameValuePair>();
27    urlParameters.add(new BasicNameValuePair("order",ord));
28    post.setEntity(new UrlEncodedFormEntity(urlParameters));
29    HttpResponse response = client.execute(post);
30    BufferedReader rd = new BufferedReader(new
31    InputStreamReader(response.getEntity().getContent()));
32    StringBuffer result = new StringBuffer();
33    String line = "";
34    while ((line = rd.readLine()) != null) {
35      result.append(line);
36    }
37    resultat=result.toString();
38  }
39  static public void sendPost(String url) throws Exception {
40    HttpClient client = new DefaultHttpClient();
41    HttpPost post = new HttpPost(url);
42    post.setHeader("User-Agent", USER_AGENT);

```

```

43 List<NameValuePair> urlParameters = new ArrayList<NameValuePair>();
44 post.setEntity(new UrlEncodedFormEntity(urlParameters));
45 HttpResponse response = client.execute(post);
46 BufferedReader rd = new BufferedReader( new InputStreamReader(response.getEntity().
    getContent()));
47 StringBuffer result = new StringBuffer();
48 String line = "";
49 while ((line = rd.readLine()) != null)
50 {
51     result.append(line);
52 }
53 JSONObject Jobj;
54 json= result.toString();
55 Jobj = new JSONObject(json);
56 temp = Jobj.getString(TAG_TEMPERATURE);
57 }
58 public String getTemp()
59 {return temp;
60 }
61 }
    
```

### V.2.2.5 La classe Identification.java

Cette classe est la première classe qui s’affiche voici l’interface graphique :

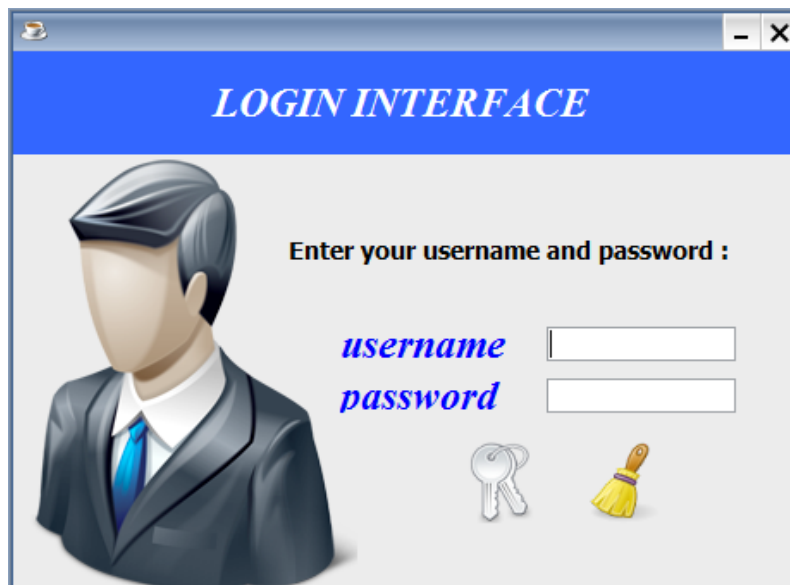


FIGURE V.3 – Interface authentification

Dans cette interface, nous avons deux boutons. Lorsqu’on clique sur le bouton nettoyer, les champs se vident. Quand on clique sur le bouton valider, on appelle une méthode pour faire une vérification avec les informations introduites et celles enregistrées dans la table admin dans la base de données. Voici le code :

```

1 public void verifier(){
2     String req= "SELECT * FROM admin ";
3     try { stat=connecter.ObttenirConnexion().createStatement();
4         ResultSet res = stat.executeQuery(req);
5         boolean f=true;
6         if( i<2){us =userfield.getText();
7         pass = MD5hash.getMD5Hash(String.valueOf((pw.getPassword())));
8         while(res.next()){
9             String o=res.getString("username");
10            String p=res.getString("password");
    
```

```

11 if (us.equals(o)& p.equals(pass)) {Accueil k=new Accueil();
12 k.setVisible(true);dispose(); f=false; i=2;} if(f==true){
13 JOptionPane.showMessageDialog(this, "Password or username incorrect","erreur", JOptionPane.
    ERROR_MESSAGE); i++; userfield.setText("");
14 pw.setText(""); }else if(i>=2) {
15 JOptionPane.showMessageDialog(this, " Sorry you have exceeded authorized attempts !","
    information",JOptionPane.INFORMATION_MESSAGE);
16 System.exit(0);}catch (SQLException e) {
17 JOptionPane.showMessageDialog(this, e); }}

```

On teste 3 fois l'authentification sinon après 3 tentatives fausses alors l'application se ferme.

### V.2.2.6 La classe Accueil.java



FIGURE V.4 – Interface accueil

C'est l'interface accueil après avoir eu une authentification réussite, alors cette interface contient une barre des menus composée de certains onglets.

- Pour quitter, on clique sur Item Quit qui exécute l'instruction suivante : `System.exit(0);`

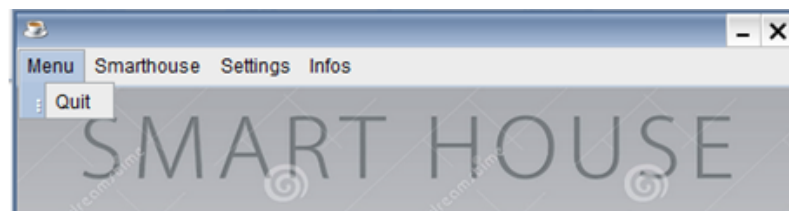


FIGURE V.5 – Quit item

- Pour l'onglet Smarthouse, on accède à l'interface control lorsqu'on clique sur Item Control : `new ControlHouse().setVisible(true); this.hide();`
- Pour l'onglet Settings, on accède à l'interface administration des comptes utilisateur. Lorsqu'on clique sur Item User :
- Pour l'onglet Infos, on affiche une petite fenêtre qui indique quelques informations sur l'application.

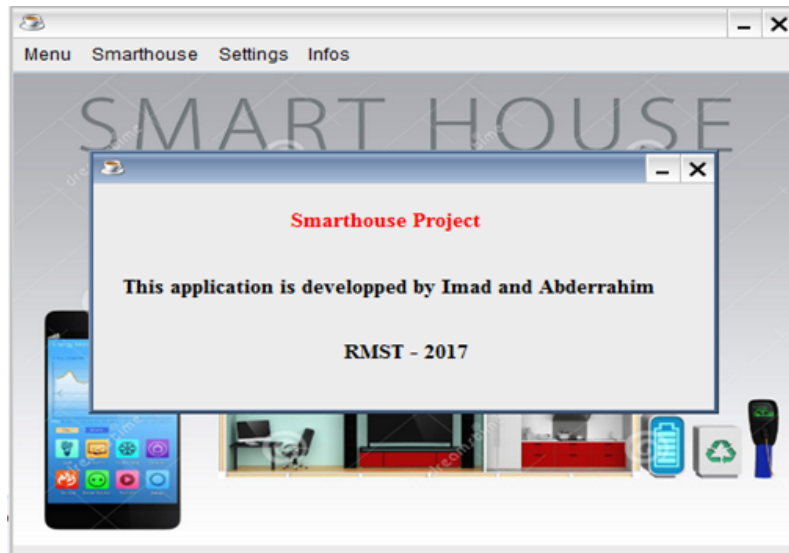


FIGURE V.6 – Fenêtre information

### V.2.2.7 La classe ControlHouse.java

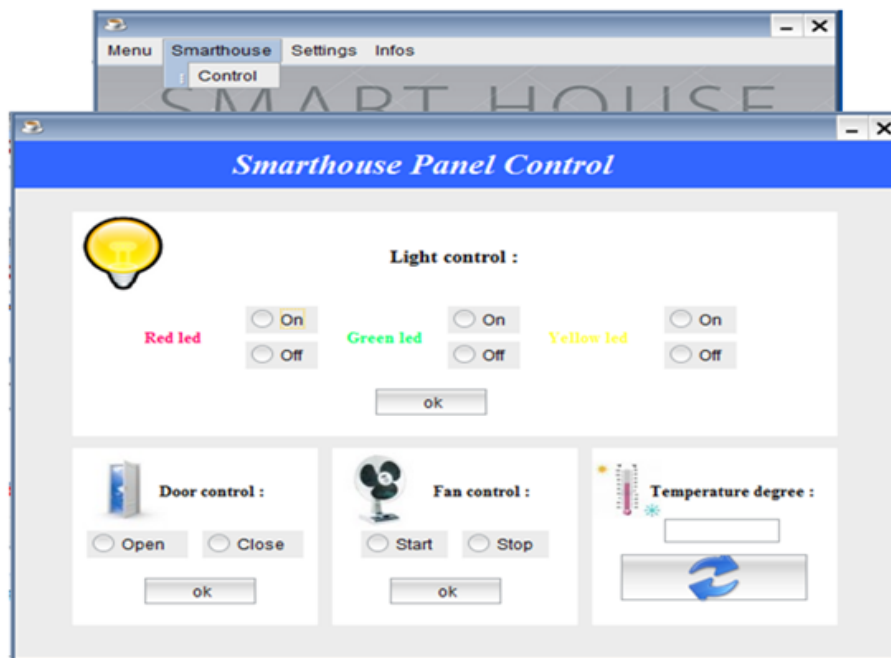


FIGURE V.7 – Interface de contrôle

Dans cette interface, on affiche le degré de température récupéré depuis le capteur de température. Pour cela, on a créé une méthode. Voici son code java :

```

1 private void dispTemp(){
2     try {
3         String url="http://192.168.1.20/smarthouse/desktopApp/temp.php";
4         new HttpDo().sendPost(url);
5         temp.setText(new HttpDo().getTemp());
6     } catch (Exception ex) {
7         Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);
8     }
9 }

```

On peut aussi allumer les LEDs rouge, jaune et verte. Cela se fait par l'envoi d'un caractère qui indique l'ordre, par exemple, pour allumer la LED rouge on envoie **ro** par



la méthode post du protocole http ensuite le script PHP reçoit ce paramètre et exécute le script python responsable de cet ordre. Voici le code java qui contrôle les LEDs :

```

1 String url="http://192.168.1.20/smarthouse/desktopApp/led.php";
2 if(ron.isSelected()) {
3   try {
4     new HttpDo().sendPost(url,"ro");
5   } catch (Exception ex) {
6     Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);}
7   if(roff.isSelected()) {
8     try {
9       new HttpDo().sendPost(url,"rc");
10    } catch (Exception ex) {
11      Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);}
12    if(gon.isSelected()) {
13      try {
14        new HttpDo().sendPost(url,"go");
15      } catch (Exception ex) {
16        Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);} }
17    if(goff.isSelected()) {
18      try {
19        new HttpDo().sendPost(url,"gc");
20      } catch (Exception ex) {
21        Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);} }
22    if(yon.isSelected()) {
23      try {
24        new HttpDo().sendPost(url,"yo");
25      } catch (Exception ex) {
26        Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);} }
27    if(yoff.isSelected()) {
28      try {
29        new HttpDo().sendPost(url,"yc");
30      } catch (Exception ex) {
31        Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);
32      }}

```

– Pour la porte, par le même principe, on envoie un caractère au serveur et voici le code java :

```

1   String url="http://192.168.1.20/smarthouse/desktopApp/door.php";
2   if(doorOpen.isSelected()) {
3     try {
4       new HttpDo().sendPost(url,"do");}
5     catch (Exception ex) {
6       Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);}
7     if(doorClose.isSelected()) {
8       try {new HttpDo().sendPost(url,"dc");} catch (Exception ex) {
9         Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);}
10
11

```

– Pour le ventilateur voici le code java :

```

1   String url="http://192.168.1.20/smarthouse/desktopApp/fan.php";
2   if(fanStart.isSelected()) {
3     try {
4       new HttpDo().sendPost(url,"mo");} catch (Exception ex) {
5       Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);}
6     if(fanStop.isSelected()) {
7       try {
8         new HttpDo().sendPost(url,"mc");} catch (Exception ex) {
9         Logger.getLogger(ControlHouse.class.getName()).log(Level.SEVERE, null, ex);}
10
11

```

V.2.2.8 La classe Register.java

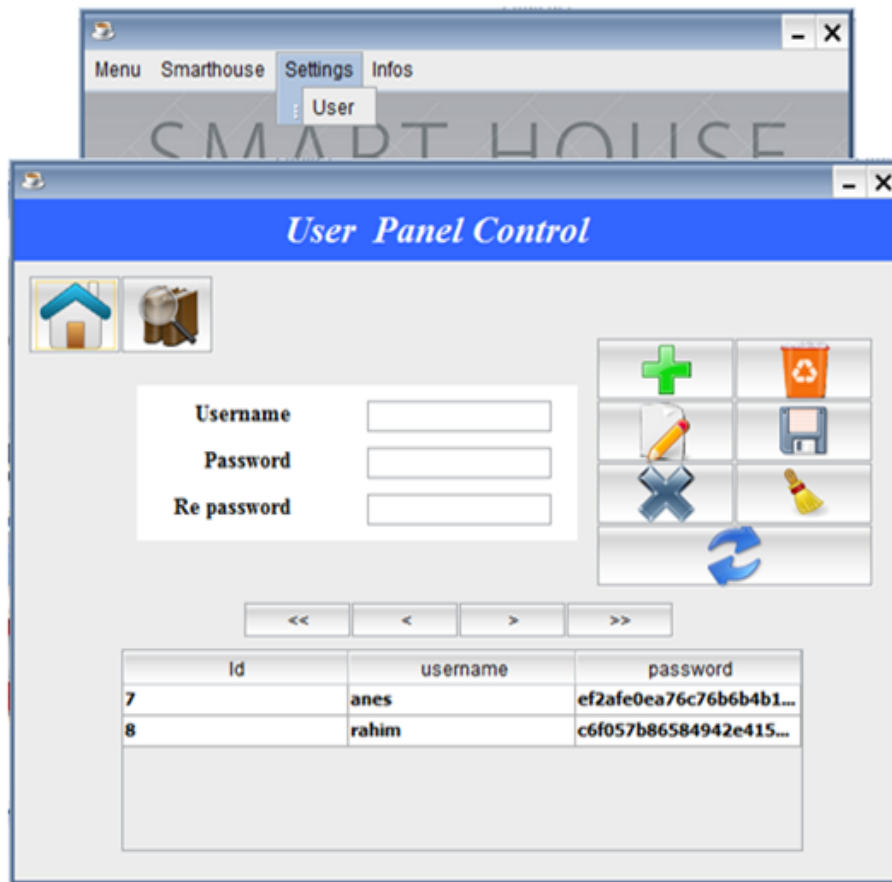


FIGURE V.8 – Interface utilisateur

- Dans cette interface, on peut créer, supprimer et même mettre à jour les comptes utilisateurs. Ci-dessous la méthode qui sert à ajouter un nouveau compte :

```

1 public void ajouterDB(){
2     String user= userfield.getText();
3     String val = null;
4     val=MD5hash.getMD5Hash(Arrays.toString(passwordfield.getPassword()));
5     if(TestUser(user)==1){
6         JOptionPane.showMessageDialog(null,"username already used ! choose another one");
7         viderField();
8     }else {
9         String requete="INSERT INTO user(username,password) VALUES ("'+user+"','"+val+"'");
10        try {
11            stat=connecter.ObttenirConnexion().createStatement();
12            stat.executeUpdate(requete);
13            JOptionPane.showMessageDialog(null,"query executed successfully");
14            this.viderField();
15        } catch (SQLException et){
16            JOptionPane.showMessageDialog(this, et );
17        }
18    }

```

- Voici maintenant la méthode pour modifier un compte :

```

1 public void modifierBD(){
2     try{
3         int a= tableclient.getSelectedRow();
4         int c =Integer.valueOf(tableclient.getValueAt(a, 0).toString());
5         String user= userfield.getText();

```

```

6   String val ;
7   val=MD5hash.getMD5Hash(String.valueOf(passwordfield.getPassword()));
8   String req="UPDATE user set  username="+user+" ,password="+val+" WHERE id="+ c;
9   stat=connecter.ObttenirConnexion().createStatement();
10  stat.executeUpdate(req);
11  JOptionPane.showMessageDialog(this, "update successful" );}
12  catch(SQLException e){
13  JOptionPane.showMessageDialog(this, e );}
14
15

```

– Et pour la suppression voici la méthode :

```

1   public void deleteDB(){
2   int a=0;
3   if(d==1){
4   a= Integer.parseInt(tableclient.getValueAt(tableclient.getSelectedRow(), 0).
5   toString());}
6   String req2="DELETE FROM 'user' WHERE id =" +a;
7   try {
8   int option = JOptionPane.showConfirmDialog(null,"Are you sure you want delete it ?", "
9   confirmation", JOptionPane.YES_NO_OPTION,JOptionPane.QUESTION_MESSAGE);
10  if(option == JOptionPane.OK_OPTION){
11  stat=connecter.ObttenirConnexion().createStatement();
12  stat.executeUpdate(req2);
13  ((DefaultTableModel)tableclient.getModel()).removeRow(tableclient.
14  getSelectedRow());
15  tableclient.setModel(model);
16  } else
17  JOptionPane.showMessageDialog(this,"operation canceled","information",JOptionPane.
18  INFORMATION_MESSAGE);
19  stat.close();
20  }
21  catch(SQLException et){
22  JOptionPane.showMessageDialog(this,"choose an user to delete please"+et);
23  }}

```

– Un clic sur le bouton recherche fait apparaître la fenêtre suivante :

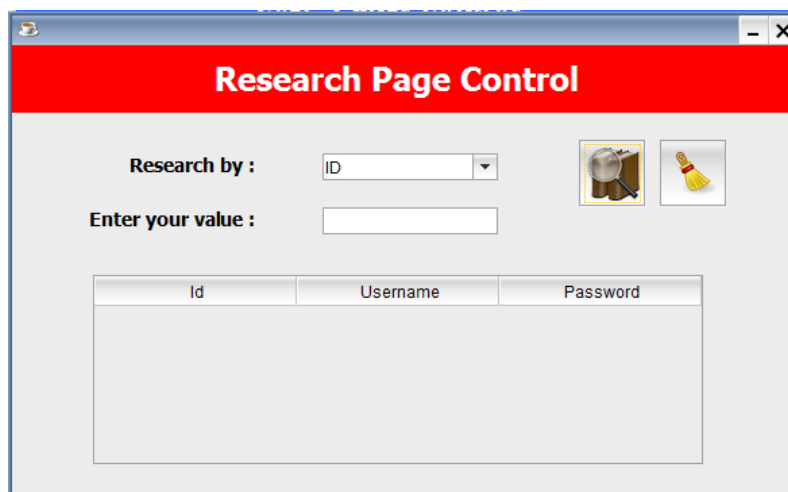


FIGURE V.9 – Interface recherche

On peut faire la recherche d'un utilisateur soit en utilisant l'id ou bien le champ username. Voici le code java :

```

1 private void RechUser(){
2 clearTable();
3 valrech=valeurField.getText();

```

```

4 String req1 = null;
5 String z=comboRech.getItemAt(comboRech.getSelectedIndex()).toString();
6 if( z.equals("ID")){
7     int d = 0;
8     int c =Integer.parseInt(valrech);
9     req1="SELECT * from 'user' WHERE 'id'="+c;
10    try {
11        stat=connecter.ObttenirConnexion().createStatement();
12        ResultSet resultat=stat.executeQuery(req1);
13        while(resultat.next()){
14            model.addRow(new Object[]{resultat.getString("id"),resultat.getString("username"),
15                resultat.getString("password")});
16            d=Integer.parseInt(resultat.getString("id"));
17            if(c!=d){JOptionPane.showMessageDialog(this, "cet ID n'Ã©xiste pas" ); }
18            resultat.close();
19            catch(SQLException et){
20                JOptionPane.showMessageDialog(this, et );}}
21            if( z.equals("Username")){
22                String c =""+valrech+"";
23                req1="SELECT * from 'user' WHERE 'username'="+c;
24                String d =null;
25                try {
26                    stat=connecter.ObttenirConnexion().createStatement();
27                    ResultSet resultat=stat.executeQuery(req1);
28                    while(resultat.next()){
29                        model.addRow(new Object[]{resultat.getString("id"),resultat.getString("username"),
30                            resultat.getString("password")});
31                        d=resultat.getString("username"); }
32                        JOptionPane.showMessageDialog(this, "ce nom n'Ã©xiste pas" ); } resultat.close(); }
33                    catch(SQLException et){
34                        JOptionPane.showMessageDialog(this, et ); } }

```

### V.3 CONCEPTION DU SITE

Notre site est composé de 4 dossiers et 10 scripts PHP selon la figure ci-dessous tel que le dossier css comprend le fichier **app.css** qui assure le design de notre site ainsi le dossier polices contient les fichiers de police d'écriture sur le site. Pour le dossier house, il contient les images utilisées dans le site et le dernier dossier inc ça contient le script de configuration et les headers et footers des autres pages du site.

Sachant qu'on va utiliser la classe PDO pour se connecter à la base de données smarthouse et ainsi pour l'enregistrement et l'authentification des comptes utilisateurs.

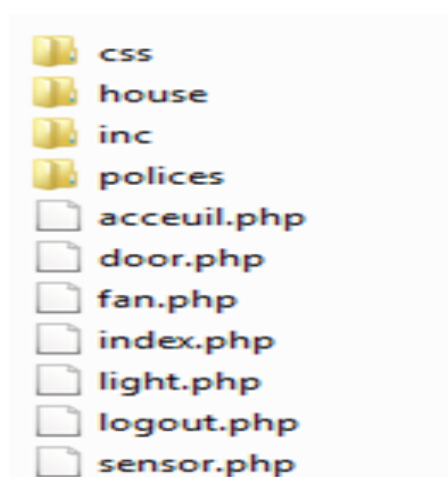


FIGURE V.10 – Répertoire du site

### V.3.1 Script de configuration (db.php)

```

1 <?php
2 $pdo = new PDO('mysql:dbname=smarthouse;host=localhost', 'root', '111');
3 $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
4 $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);

```

### V.3.2 Page de démarrage du site (index.php)

Le fichier index est la page HTML que les utilisateurs verront en premier en accédant à notre site. Il est donc vital qu'elle soit bien remplie, avec tous les liens vers les autres pages du site. C'est le fichier par défaut du site, qui doit se trouver sur la racine du site. Lorsqu'on tape le nom de notre site, (http://192.168.1.20/smarthouse/website/). La page qui s'ouvre ne se nommera pas http://192.168.1.20/smarthouse/website/ , mais http://192.168.1.20/smarthouse/website/index.php.

#### V.3.2.1 Vue

FIGURE V.11 – La page index

#### V.3.2.2 Code source

```

1 <?php
2 session_start ();
3 if(empty($_SESSION['login']) || empty($_SESSION['pwd'])) {
4 } else { header('Location: accueil.php'); }
5 if(!empty($_POST)){ $errors = array();require_once 'inc/db.php';
6 if(empty($_POST['username']) || empty($_POST['password'])) {
7 $errors['e'] = "fill up all fields please ";} if(empty($errors)){
8 $req = $pdo->prepare("SELECT * FROM user WHERE username = :username ");
9 $user = htmlspecialchars(stripslashes($_POST['username']));
10 $pass = htmlspecialchars(stripslashes($_POST['password']));

```





### V.3.5 Page de streaming vidéo

#### V.3.5.1 Vue



FIGURE V.14 – Page de streaming vidéo

#### V.3.5.2 Code source

Le code source de cette page est assuré par le serveur de webcam motion, il suffit juste de taper l'url : <http://192.168.1.20:8081>.

### V.3.6 Page de contrôle de porte (door.php)

#### V.3.6.1 Vue



FIGURE V.15 – Page de contrôle de porte



### V.3.6.2 Code source

```

1 <?php require 'inc/header3.php'; ?>
2 <h1 align="center" >Door control
3 </h1>
4 <h1 align="center"> 
5 </h1>
6 <?php
7 if (isset($_POST['do']))
8 {
9   exec("sudo python /home/pi/Desktop/code/door/doorOpen.py");
10 }
11 if (isset($_POST['dc']))
12 {
13   exec("sudo python /home/pi/Desktop/code/door/doorClose.py");};>
14 <form method="post" align="center">
15 <H3>Door :
16 </h3>
17 <button class="btn" name="do">Open
18 </button>&nbsp;
19 <button class="btn" name="dc">Close
20 </button>
21 </form>

```

### V.3.7 Page de contrôle des LEDs (light.php)

#### V.3.7.1 Vue



FIGURE V.16 – Page de contrôle des LEDs

### V.3.7.2 Code source

```

1 <?php require 'inc/header3.php'; ?>
2 <h1 align="center" >Light control </h1>
3 <h1 align="center" > </br>
4 <?php
5 if (isset($_POST['LightON']))
6 {exec("sudo python /home/pi/Desktop/code/led/redOn.py");}
7 if (isset($_POST['LightOFF']))
8 {exec("sudo python /home/pi/Desktop/code/led/redOff.py");}
9 if (isset($_POST['go']))
10 {exec("sudo python /home/pi/Desktop/code/led/greenOn.py");}
11 if (isset($_POST['goff']))
12 {exec("sudo python /home/pi/Desktop/code/led/greenOff.py");}
13 if (isset($_POST['yo']))
14 {exec("sudo python /home/pi/Desktop/code/led/yellowOn.py");}
15 if (isset($_POST['yoff']))
16 {exec("sudo python /home/pi/Desktop/code/led/yellowOff.py");}?>
17 <form method="post" align="center">
18 <H3 ><FONT color="red">Red light :</FONT> </h3>
19 <button class="btn" name="LightON">Light ON</button>&nbsp;
20 <button class="btn" name="LightOFF">Light OFF</button><br>
21 <H3 ><FONT color="green">Green light :</FONT> </h3>
22 <button class="btn" name="go">Light ON</button>&nbsp;
23 <button class="btn" name="goff">Light OFF</button><br>
24 <H3 ><FONT color="yellow">Yellow light :</FONT> </h3>
25 <button class="btn" name="yo">Light ON</button>&nbsp;
26 <button class="btn" name="yoff">Light OFF</button><br></form>

```

### V.3.8 Page de contrôle du ventilateur (fan.php)

#### V.3.8.1 Vue



FIGURE V.17 – Page de contrôle du ventilateur

### V.3.8.2 Code source

```
1 <?php require 'inc/header3.php'; ?>
2 <h1 align="center" >Fan control </h1>
3 <h1 align="center" > 
4 </h1>
5 <?php
6 if (isset($_POST['fs']))
7 {
8   exec("sudo python /home/pi/Desktop/code/fan/fanStart.py");
9 }
10 if (isset($_POST['fstp']))
11 {
12   exec("sudo python /home/pi/Desktop/code/fan/fanStop.py");
13 }
14 ?>
15 <form method="post" align="center">
16 <H3>Fan :</h3>
17 <button class="btn" name="fs">Start</button>&nbsp;
18 <button class="btn" name="fstp">Stop</button><br><br>
19 </form>
```

## V.4 PRÉSENTATION D'ANDROID

Avec l'explosion des ventes de Smartphones ces dernières années, Android a pris une place importante dans la vie quotidienne. Ce système d'exploitation permet d'installer des applications de toutes sortes : jeux, bureautique, multimédia, etc. Pour cette raison, on a opté dans notre projet pour utiliser une application mobile Android pour commander notre maison intelligente.



FIGURE V.18 – *Android*

### V.4.1 La création d'Android

À l'origine, « Android » était le nom d'une PME américaine, créée en 2003 puis rachetée par Google en 2005, qui avait la ferme intention de s'introduire sur le marché des produits mobiles.

C'est pourquoi est créée en novembre de l'année 2007 l'Open Handset Alliance et qui comptait à sa création 35 entreprises évoluant dans l'univers du mobile, dont Google.

Android est à l'heure actuelle le système d'exploitation pour Smartphones et tablettes le plus utilisé. Les prévisions en ce qui concerne la distribution d'Android sur le marché sont très bonnes. Bientôt, il se trouvera dans certains téléviseurs et les voitures. Android sera partout.

### V.4.2 Les avantages de Google Android

- Open source.
- Gratuit (ou presque).
- Facile à vendre Play Store.
- Facile à développer API disponibles.
- Flexible : il s'adapte à beaucoup de structures différentes. Les Smartphones, les tablettes, les TV et même système embarqué (Raspberry Pi).
- Plusieurs applications peuvent être lancées à la fois.
- De constantes mises à jour sont proposées, ce qui montre une amélioration du système.

### V.4.3 Les inconvénients de Google Android

- Vous êtes connecté en "mode continu" et donc en permanence.
- Les téléphones Android possèdent de faible autonomie.
- Possibilité de chauffage de votre téléphone.
- Certaines applications sont incompatibles à des versions de Google Android.
- Bug de certaines applications.

### V.4.4 Versions d'Android

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en septembre 2008.

Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités.

#### V.4.4.1 Versions disparues

- Versions 1.0 Android 1.0 (23 septembre 2008)
- Versions 1.1 Banana Bread (9 février 2009)
- Versions 1.5 Cupcake (30 avril 2009)
- Versions 1.6 Donut (15 septembre 2009)

#### V.4.4.2 Versions existantes

- Versions 2 Eclair (26 octobre 2009)
- Version 2.2 Froyo (20 mai 2010)
- Versions 2.3 Gingerbread (6 décembre 2010)
- Versions 3 Honeycomb (22 février 2011)
- Versions 4 Ice Cream Sandwich (19 octobre 2011)
- Versions 4.1 Jelly Bean (9 juillet 2012)
- Versions 4.4 KitKat (3 septembre 2013)
- Versions 5 Lollipop (3 novembre 2014)
- Versions 6 Marshmallow (fin mai 2015)
- Versions 7 Nougat (21 septembre 2016)

## V.5 PHP ET MySQL

Pour faire des connexions avec le serveur apache de notre Raspberry, nous allons envoyer des requêtes grâce au protocole **HTTP** du système **Android** tout en utilisant du script **PHP** pour interagir avec MySQL selon l'architecture client-serveur suivante :

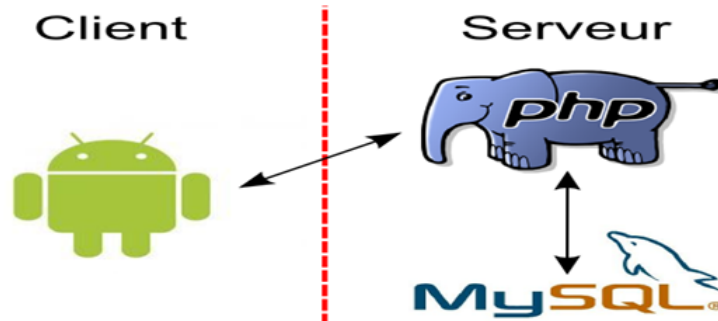


FIGURE V.19 – Communication Android PHP MySQL

Comme on peut aussi commander notre maison intelligente à distance allumer des LEDs ouvrir la porte démarrer le ventilateur pour tout ça nous allons héberger les scripts PHP qu'on a vu au chapitre 4 dans notre serveur Apache dans le dossier « `/var/www/html/smarthouse` ».

### V.5.1 Page de configuration (db\_config.php)

```

1 <?php/* * All database connection variables */
2 define('DB_USER', "root"); // db user
3 define('DB_PASSWORD', "111"); // db password
4 define('DB_DATABASE', "smarthouse"); // database name
5 define('DB_SERVER', "localhost"); // db server?>

```

### V.5.2 Script d'authentification des utilisateurs (authentification.php)

```

1 <?php
2 $response = array();
3 if (isset($_GET["username"]) && isset($_GET["password"]) )
4 {
5 $u= $_GET["username"];
6 $p = md5($_GET["password"]);
7 $con = mysqli_connect(DB_SERVER, DB_USER, DB_PASSWORD, DB_DATABASE);
8 $req="select * from user where username=$u and password=$p";
9 $res = mysqli_query($con, $req);
10 if (mysqli_num_rows($res) > 0)
11 {
12 $response["success"] = 1; echo json_encode($response);
13 }
14 else {
15 $response["success"] = 0; $response["message"] = "No user found";
16 echo json_encode($response);
17 }
18 }
19 else {
20 $response["success"] = 0; $response["message"] = "Required field(s) is missing";
21 echo json_encode($response);
22 }
23 ?>

```

## V.6 CONCEPTION DE NOTRE APPLICATION ANDROID

Nous allons présenter dans cette partie étape par étape notre application Android que l'on a nommée SmartHouse servant à contrôler notre maison intelligente.

### V.6.1 Répertoire complet de l'application

Notre projet d'application est composé de 13 classes java et neuf fichiers xml et douze icônes. Notre application est développée par l'environnement de développement intégré eclipse illustré par la figure suivante :

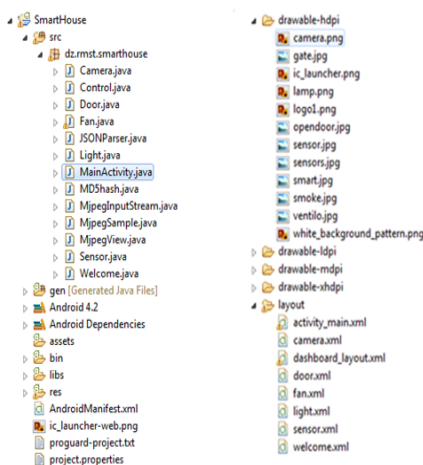


FIGURE V.20 – Répertoire de l'application SmartHouse

### V.6.2 Fichier AndroidManifest.xml

Dans ce fichier, on déclare nos classes java qui héritent de la classe prédéfinie en Android « **Activity** » et on déclare aussi les permissions nécessaires pour le bon fonctionnement de notre application en l'occurrence la permission INTERNET car on travaille en réseaux.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 package="dz.rmst.smarthouse"
4 android:versionCode="1"
5 android:versionName="1.0" >
6 <uses-sdk
7 android:minSdkVersion="14" android:targetSdkVersion="17" />
8 <uses-permission android:name="android.permission.INTERNET"/>
9 <application
10 android:allowBackup="true"
11 android:icon="@drawable/ic_launcher"
12 android:label="@string/app_name"
13 android:theme="@style/AppTheme" >
14 <activity
15 android:name="dz.rmst.smarthouse.MainActivity"
16 android:label="@string/app_name" >
17 <intent-filter><action android:name="android.intent.action.MAIN" />
18 <category android:name="android.intent.category.LAUNCHER" />
19 </intent-filter>
20 </activity>
21 <activity android:name=".Control" > </activity>
22 <activity android:name=".Welcome" ></activity>
23 <activity android:name=".Camera" > </activity>
24 <activity android:name=".Light" > </activity>

```

```

25 <activity android:name=".Sensor" > </activity>
26 <activity android:name=".Door" > </activity>
27 <activity android:name=".Fan" > </activity>
28 <activity android:name=".MjpegSample" >
29 </activity>
30 </application>
31 </manifest>

```

### V.6.3 Interface authentication

Commençons par illustrer la vue de notre interface qui est le fichier `activity_main.xml`.

#### V.6.3.1 Vue

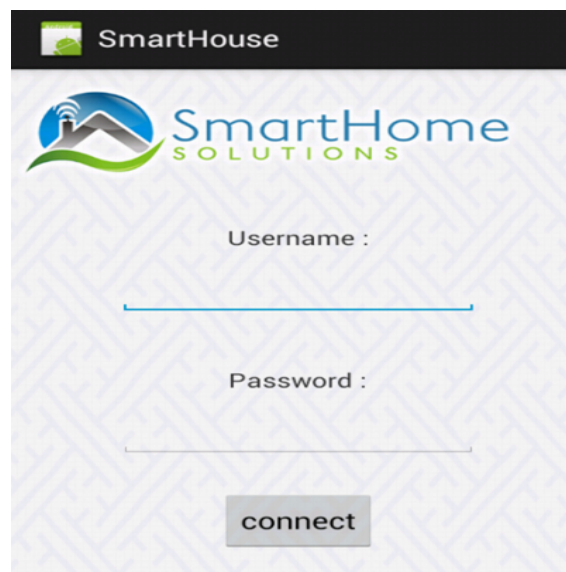


FIGURE V.21 – Vue `activity_main.xml`

#### V.6.3.2 Classe `MainActivity.java`

```

1 package dz.rmst.smarthouse;
2 import java.util.ArrayList;
3 import java.util.List;
4 import org.apache.http.NameValuePair;
5 import org.apache.http.message.BasicNameValuePair;
6 import org.json.JSONException;
7 import org.json.JSONObject;
8 import android.app.Activity;
9 import android.app.ProgressDialog;
10 import android.content.Intent;
11 import android.os.AsyncTask;
12 import android.os.Bundle;
13 import android.util.Log;
14 import android.view.Menu;
15 import android.view.View;
16 import android.widget.Button;
17 import android.widget.EditText;
18 import android.widget.Toast;
19 public class MainActivity extends Activity {
20     Button login;

```

```

21 static EditText user,password;
22 JSONParser jParser = new JSONParser();
23 private ProgressDialog pDialog;
24 private static String url_auth = "http://192.168.1.20/smarthouse/authentication.php";
25 private static final String TAG_SUCCESS = "success"; int success;
26 @Override
27 protected void onCreate(Bundle savedInstanceState) {
28     super.onCreate(savedInstanceState);
29     setContentView(R.layout.activity_main);
30     login = (Button) findViewById(R.id.connect); user = (EditText) findViewById(R.id.user);
31     password = (EditText) findViewById(R.id.password);
32     login.setOnClickListener(new View.OnClickListener() {
33         @Override
34         public void onClick(View v) {
35             if ("".equals(user.getText().toString()) || "".equals(password.getText().toString()) ){
36                 Toast.makeText(getApplicationContext(), "fill up all fields.", Toast.LENGTH_SHORT).show(); }
37             else {new Authentication().execute();} } });
38     class Authentication extends AsyncTask<String, String, String > {
39         @Override
40         protected void onPreExecute() {
41             super.onPreExecute();pDialog = new ProgressDialog(MainActivity.this);
42             pDialog.setMessage("Authentification is running . Please wait...");pDialog.setIndeterminate(false
43             );
44             pDialog.setCancelable(false);pDialog.show();}
45         protected String doInBackground(String... args) {
46             String u=user.getText().toString();String p=password.getText().toString();
47             List<NameValuePair> params = new ArrayList<NameValuePair>();
48             params.add(new BasicNameValuePair("username",u));
49             params.add(new BasicNameValuePair("password", MD5hash.getMD5Hash (p)));
50             JSONObject json = jParser.makeHttpRequest(url_auth, "GET", params);
51             try {
52                 success = json.getInt(TAG_SUCCESS);
53                 catch (JSONException e) {e.printStackTrace();}
54                 return null;}
55         protected void onPostExecute(String file_url) {
56             pDialog.dismiss();
57             if (success == 1) {
58                 Intent intent = new Intent(MainActivity.this,Welcome.class);startActivity(intent);
59                 finish();}
60             else {
61                 Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();} }
62         @Override
63         public boolean onCreateOptionsMenu(Menu menu) {
64             getMenuInflater().inflate(R.menu.activity_main, menu);
65             return true;}

```

### V.6.3.3 Explication du code java

Au niveau de cette interface. On fait une authentification avec un nom d'utilisateur et un mot de passe c.à.d. on interroge notre base de données de tester le couple username password s'il est correct on accède à la page d'accueil de notre application par contre, si on n'a pas de compte on demande auprès de l'administrateur de nous créer un compte.

Notre classe java inclut une classe **Authentification** qui hérite de la classe **AsyncTask <String, String, String >**. Alors une classe AsyncTask permet de réaliser des tâches de manière asynchrone, à la manière de la classe **Thread**.

L'avantage de l'AsyncTask est sa simplicité d'utilisation et d'implémentation. Le Thread secondaire est créé automatiquement et la communication entre les différents Thread est simplifiée. Les trois paramètres attendus lors de la déclaration sont des types génériques dont voici la signification :

- Le premier est le type des paramètres fournis à la tâche.



- Le second est le type de données transmises durant la progression du traitement.
- Enfin, le troisième est le type du résultat de la tâche.

Une AsyncTask doit obligatoirement implémenter la méthode **doInBackground**. C'est elle qui réalisera le traitement de manière asynchrone dans un Thread séparé.

Les méthodes **onPreExecute** (cette méthode est appelée avant le traitement), **onProgressUpdate** (appelée lorsque nous souhaitons afficher sa progression) et **onPostExecute** (appelée après le traitement) sont optionnelles.

Dans la méthode **doInBackground**, on procède à envoyer le couple (username, password) comme des paramètres à notre serveur apache à travers la méthode **GET** du Protocol **HTTP** en spécifiant l'url de script php ici <http://192.168.1.20/smarthouse/authentification.php>, de telle sorte que le mot de passe soit haché avec la fonction md5 est envoyé. Ensuite, on exécute l'instruction suivante : `JSONObject json = jParser.makeHttpRequest(url_auth, "GET", params);` Et on reçoit à l'issue de cette requête une réponse depuis le serveur sous format json de type entier : `int success = json.getInt(TAG_SUCCESS)`. Si la réponse est égale à 1 cela signifie une authentification réussite et on accède à l'interface accueil de notre application sinon l'authentification est échouée.

#### V.6.4 Interface accueil (welcome.xml)

C'est la vue Accueil de notre application on clique sur le bouton Enter pour accéder à l'interface de contrôl ou le bouton Disconnect pour se déconnecter.

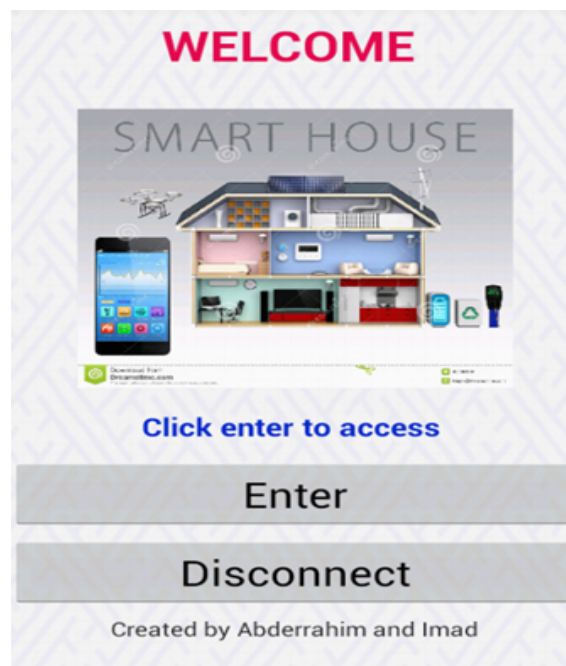


FIGURE V.22 – Vue Accueil

#### V.6.5 Interface de contrôle (dashboard\_layout.xml)

Dans cette interface, on choisit quel appareil on veut contrôler dans notre maison intelligente à distance via WiFi.

- Lorsqu'on clique sur le bouton Sensors ça nous redirige vers l'interface qui affiche le degré de température.
- Pour accéder au vidéo streaming, on clique sur le bouton Camera.
- Pour contrôler la porte, on clique sur le bouton Door.

- Même chose pour contrôler les LEDs, on clique sur le bouton Light et pour contrôler le ventilateur, on clique sur le bouton Fan.



FIGURE V.23 – Vue page de control

## V.6.6 Interface capteur

### V.6.6.1 Vue (sensor.xml)

Dans cette interface, on peut connaître le degré de la température de notre maison intelligente en temps réel et aussi s’il y a eu du gaz détecté, ça sera affiché sur TextView dans cette interface.

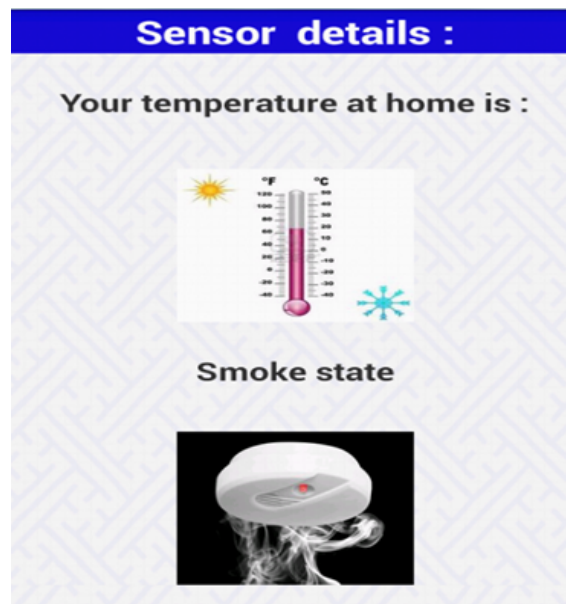


FIGURE V.24 – Vue page capteurs

### V.6.6.2 Classe Sensor.java

```

1 public class Sensor extends Activity {
2     TextView t=null;
3     TextView g=null;
4     private ProgressDialog pDialog;
5     JSONParser jParser = new JSONParser();
6     private static String url_sensor = "http://192.168.1.20/smarthouse/temp.php";
7     private static String url_gaz = "http://192.168.1.20/smarthouse/gaz.php";
8     private static final String TAG_TEMPERATURE = "temperature";
9     private static final String TAG_GAZ = "gaz";
10    JSONArray temperature = null;
11    String tem,gaz;
12    String order;
13    @Override
14    protected void onCreate(Bundle savedInstanceState) {
15        super.onCreate(savedInstanceState);
16        requestWindowFeature(Window.FEATURE_NO_TITLE);
17        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
18        WindowManager.LayoutParams.FLAG_FULLSCREEN);
19        setContentView(R.layout.sensor);
20        t = (TextView) findViewById(R.id.txTemperature);
21        g = (TextView) findViewById(R.id.txGaz);
22        new LoadTemperature().execute(); }
23    class LoadTemperature extends AsyncTask<String, String, String> {
24        @Override
25        protected void onPreExecute() {
26            super.onPreExecute();pDialog = new ProgressDialog(Sensor.this);
27            pDialog.setMessage("Loading. Please wait...");pDialog.setIndeterminate(false);
28            pDialog.setCancelable(false);pDialog.show();}
29        protected String doInBackground(String... args) {
30            List<NameValuePair> params = new ArrayList<NameValuePair>();
31            JSONObject json = jParser.makeHttpRequest(url_sensor, "GET", params);
32            try {
33                tem=json.getString(TAG_TEMPERATURE);gaz=json2.getString(TAG_GAZ);
34            } catch (JSONException e)
35            {e.printStackTrace();} return null;}
36        protected void onPostExecute(String file_url) {
37            pDialog.dismiss();
38            t.setText(tem);
39            g.setText(gaz); }}}

```

### V.6.6.3 Explication du code java

Dans cette classe, on envoie une requête http par la méthode GET et on récupère la valeur de la température mesurée par le capteur de température sous format json puis en les affiche sur TextView au niveau du fichier xml et aussi un message d'alarme si le capteur de gaz détecte de fumée dans la maison.

### V.6.7 Interface camera (camera.xml)

On clique sur le bouton Start pour lancer le streaming vidéo de notre camera de surveillance, mais pour ce faire, on avait besoin de créer trois classes java afin d'accéder à notre camera à distance et afficher le flux sur écran :

**MjpegInputStream.java , MjpegView.java et MjpegSample.java .**

### V.6.7.1 Vue (camera.xml)

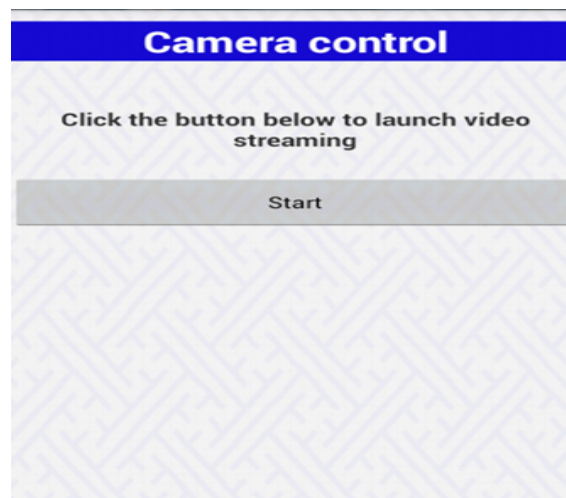


FIGURE V.25 – Vue page de lancement de camera

### V.6.7.2 Explication du code java

Chaque classe a un rôle bien défini tel que la classe **MjpegView** qui hérite de la classe **SurfaceView** c'est celle-ci qui s'occupe de l'affichage de la vidéo (dimension largeur et hauteur, position ..etc.) sur notre Smartphone tandis que la classe **MjpegInputStream** héritant de la classe **DataInputStream** s'occupe de la récupération de flux vidéo de notre caméra de surveillance depuis le Raspberry Pi grâce au logiciel motion qui diffuse la vidéo en temps réel. Il reste la dernière classe **MjpegSample** c'est la classe qui hérite de la classe **Activity** dans laquelle on déclare l'url de la vidéo-surveillance en l'occurrence <http://192.168.1.20:8081> ; on l'appelle lorsqu'on clique sur le bouton Start de la vue **camera.xml** pour démarrer le vidéo streaming.

## V.6.8 Interface portail

### V.6.8.1 Vue (door.xml)



FIGURE V.26 – Vue page de control du portail

### V.6.8.2 Classe Door.java

```

1 public class Door extends Activity {
2     boolean state=false;private ProgressDialog pDialog;
3     JSONParser jsonParser = new JSONParser();
4     private static final String url_door = "http://192.168.1.20/smarthouse/door.php";
5     private static final String TAG_CMD = "order";
6     String order;boolean check=false;
7     private Switch mySwitch;
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10        super.onCreate(savedInstanceState);
11        requestWindowFeature(Window.FEATURE_NO_TITLE);
12        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
13        WindowManager.LayoutParams.FLAG_FULLSCREEN);
14        setContentView(R.layout.door);
15        mySwitch = (Switch) findViewById(R.id.doorSwitch);
16        mySwitch.setChecked(false);
17        mySwitch.setOnCheckedChangeListener(new OnCheckedChangeListener() {
18            @Override
19            public void onCheckedChanged(CompoundButton buttonView,boolean isChecked) {
20                if(isChecked){order="dc";new Doorcontrol().execute();
21                }else{order="do";
22                new Doorcontrol().execute();  } }); order="";}

```

### V.6.8.3 Explication du code java

Dans cette classe à l’instar des autres classes, on envoie un caractère via la méthode POST, si le caractère est égal à « do » cela veut dire qu’on donne un ordre pour ouvrir la porte, et s’il est égal à « dc » alors qu’il s’agit de fermer la porte.

## V.6.9 Interface light

### V.6.9.1 Vue (Light.xml)



FIGURE V.27 – Vue page de control des LEDs

### V.6.9.2 Classe light.java

```

1 public class Light extends Activity {
2     private ProgressDialog pDialog;
3     JSONParser jsonParser = new JSONParser();
4     private static final String url_light = "http://192.168.1.20/smarthouse/led.php";
5     private static final String TAG_SUCCESS = "success";
6     private static final String TAG_CMD = "order";
7     String order;
8     boolean check=false;
9     private Switch sred,syellow,sgreen;
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        requestWindowFeature(Window.FEATURE_NO_TITLE);
14        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
15        WindowManager.LayoutParams.FLAG_FULLSCREEN);
16        setContentView(R.layout.light);
17        syellow = (Switch) findViewById(R.id.yellowSwitch);
18        sred = (Switch) findViewById(R.id.redSwitch);
19        sgreen = (Switch) findViewById(R.id.greenSwitch);
20        sred.setChecked(false);
21        syellow.setChecked(false);
22        sgreen.setChecked(false);
23        syellow.setOnCheckedChangeListener(new OnCheckedChangeListener() {
24            @Override
25            public void onCheckedChanged(CompoundButton buttonView,boolean isChecked) {
26                if(isChecked){
27                    order="jo";
28                    new Lightcontrol().execute();
29                }else{
30                    order="jc";
31                    new Lightcontrol().execute();    }});
32        sred.setOnCheckedChangeListener(new OnCheckedChangeListener() {
33            @Override
34            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
35                if(isChecked){
36                    order="ro";
37                    new Lightcontrol().execute();
38                }
39                else{
40                    order="rc";
41                    new Lightcontrol().execute();}}});
42        sgreen.setOnCheckedChangeListener(new OnCheckedChangeListener() {
43            @Override
44            public void onCheckedChanged(CompoundButton buttonView,boolean isChecked) {
45                if(isChecked){
46                    order="go";
47                    new Lightcontrol().execute();
48                }
49                else{
50                    order="gc";
51                    new Lightcontrol().execute();
52                }});
53            order="";}

```

### V.6.9.3 Explication du code java

Voici une partie de la cette classe Light.java qui vise à envoyer un caractère via la méthode POST du protocole http en spécifiant l'url <http://192.168.1.20/smarthouse/led.php> pour exécuter un script php (led.php) et se fera avec l'appelle de l'instruction : **new Lightcontrol().execute()**. Alors, pour allumer la LED rouge par exemple, on envoie le caractère « ro » au serveur apache ensuite le script php reçoit le caractère et exécute le

script python qui allume la LED rouge par la fonction `exec()`. Et pour éteindre la LED, on envoie un caractère « rc » ; c'est par la même manière qu'on contrôle les autres LEDs de smarthouse.

## V.6.10 Interface fan

### V.6.10.1 Vue (fan.xml)

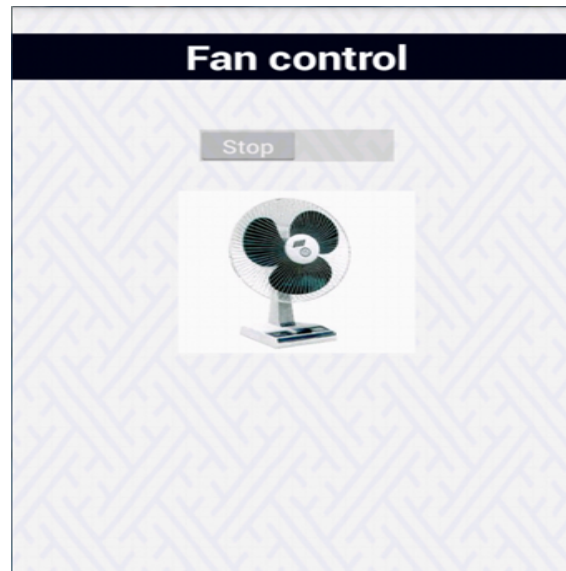


FIGURE V.28 – Vue page de control de ventilateur

### V.6.10.2 Classe fan.java

```

1 public class Fan extends Activity {
2     JSONParser jsonParser = new JSONParser();
3     private static final String url_door = "http://192.168.1.20/smarthouse/fan.php";
4     private static final String TAG_SUCCESS = "success";
5     private static final String TAG_CMD = "order";
6     String order;
7     boolean check=false;
8     private Switch mySwitch;
9     private TextView switchStatus;
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        requestWindowFeature(Window.FEATURE_NO_TITLE);
14        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
15        WindowManager.LayoutParams.FLAG_FULLSCREEN);
16        setContentView(R.layout.fan);
17        switchStatus = (TextView) findViewById(R.id.statusfan);
18        mySwitch = (Switch) findViewById(R.id.fanSwitch);
19        mySwitch.setChecked(false);
20        mySwitch.setOnCheckedChangeListener(new OnCheckedChangeListener() {
21            @Override
22            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
23                if(isChecked){
24                    order="mo";
25                    new Fancontrol().execute();}
26                else{
27                    order="mc";
28                    new Fancontrol().execute(); }));
29                order="";}

```

### V.6.10.3 Explication du code java

Dans cette classe, on envoie un caractère de type string via la méthode POST du protocole http en spécifiant l'url <http://192.168.1.20/smarthouse/fan.php>. Alors pour démarrer le ventilateur, on envoie le caractère « **mo** » au serveur apache ensuite le script fan.php reçoit le caractère et exécute le script python responsable de démarrage du ventilateur et pour l'arrêter, on envoie le caractère « **mc** ».

## V.7 CONCLUSION

Dans ce chapitre, on vu les différentes interfaces de nos applications Desktop, Android, site web et comment contrôler notre maison intelligente à distance via WiFi.

Nous avons vu comment se connecter à la base de données et ajouter, supprimer et modifier un utilisateur et même rechercher.

On a appris comment utiliser le protocole http en java pour communiquer avec un serveur web et envoyer des caractères via la méthode POST et GET.

Nous avons appris comment créer un site web avec le langage PHP qui prend 80% de part de marché du web au monde.

On a utilisé la fonction de hachage md5 pour donner plus de sécurité au mot de passe de l'utilisateur.



# CONCLUSION

Au bout de notre cursus en master Telecom, nous avons été chargés de réaliser un projet de fin de l'année. Notre travail s'est basé sur le développement d'une application sur les technologies mobiles (Smartphone) et la réalisation et la conception d'une maison intelligente. Ce projet était alors une occasion d'apprendre à travailler en binôme d'une façon autonome et efficace. Ceci nous a amené à découvrir une nouvelle plateforme de développement et à enrichir nos connaissances théoriques ainsi que pratiques et notre expérience dans le domaine des IoT. Ce dernier qui est devenu un grand domaine de recherche et un grand marché de travail au monde.

Ce travail nous a également permis de nous initier dans le domaine de la programmation mobile, en particulier, la programmation sous le système Android ainsi sur les systèmes embarqués. De plus, la réalisation d'un tel projet nous a permis de nous familiariser avec plusieurs outils et langages de programmation tels que Python, Java, PHP et MySQL. On a appris à écrire notre mémoire avec le fameux éditeur de texte libre Latex, on a fait des efforts considérables pour le terminer à terme.

Cela n'empêche pas d'avoir rencontré plein de contraintes et d'obstacles notamment le manque de composants électroniques au niveau du laboratoire ce qui nous a gêné pendant la réalisation de notre maison intelligente, ainsi comme la carte Raspberry est encore nouvelle dans le domaine des systèmes embarqués en Algérie on n'a pas trouvé de l'aide c.à.d qu'on a compté seulement sur la documentation sur internet, on a rencontré aussi des problèmes de programmation et configuration des différents modules électroniques car cela était notre première expérience dans ce domaine.

Finalement, on espère par notre travail, apporter une validation pratique de ces techniques et donner une bonne cause pour mieux explorer ce domaine d'internet des objets.

# BIBLIOGRAPHIE

- [Aissia & Chaieb 2015] Yasser Ben Aissia et Ghassene Chaieb. Gateway d'un système de monitoring d'un malade à un serveur d'urgence. Master's thesis, Ecole Nationale d'Ingénieurs de Tunis, 2015.
- [Billet 2015] Benjamin Billet. *Système de gestion de ux pour l'Internet des objets intelligents*. PhD thesis, l'université de Versailles Saint-Quentin-En-Yvelines, 2015.
- [fao.org ] fao.org. *definition de lagriculture intelligente face au climat*. <http://www.fao.org/climatechange/epic/notre-action/definition-de-lagriculture-intelligente-face-au-climat/fr/WMcGffk97IU> (13/01/2017 à 6:20)
- [htpc\_raspberrypi 2017] htpc\_raspberrypi. <http://www.htpc.com/raspberrypi>. <http://www.htpc.com.au/index.php/raspberrypi/raspberrypi>, 2017.
- [maison-et domotique.com ] maison-et domotique.com. *la domotique cest quoi*. <http://www.maison-et-domotique.com/47895-la-domotique-cest-quoi/>.
- [raspberrypis 2017] raspberrypis. <http://raspberrypis.net/>. <http://raspberrypis.net/>, 2017. (18/02/2017 à 8:28)
- [scribd.com ] scribd.com. *Domotique.pdf*. <https://fr.scribd.com/doc/298047952/46-Domotique.pdf>. (25/03/2017 à 19:15)
- [sirlan.com ] sirlan.com. *livreblanc.pdf*. [www.sirlan.com/livreblanc.pdf](http://www.sirlan.com/livreblanc.pdf).
- [site officiel de Raspberry 2017] site officiel de Raspberry. *raspberrypi\_raspbian : site officiel de Raspberry*. <https://www.raspberrypi.org/documentation/raspbian> : site officiel de Raspberry, 2017. (29/03/2017 à 15:26)
- [smarthealthit.org ] smarthealthit.org. *platform-for-healthcare*. <http://smarthealthit.org/an-app-platform-for-healthcare/about/>.
- [vinci energies.com ] vinci energies.com. *pour une industrie intelligente*. <http://www.vinci-energies.com/cest-deja-demain/pour-une-industrie-intelligente/lusine-du-futur-sera-autonome/>. (08/02/2017 à 18 :28)
- [wikipedia\_Domotique ] wikipedia\_Domotique. *Domotique*. <https://fr.wikipedia.org/wiki/Domotique>. (08/02/2017 à 10 :12)
- [wikipedia\_Réseau\_intelligent ] wikipedia\_Réseau\_intelligent. *Réseau intelligent*. [https://fr.wikipedia.org/wiki/Réseau\\_intelligent](https://fr.wikipedia.org/wiki/Réseau_intelligent). (05/02/2017 à 00:04)
- [wikipedia\_Système\_de\_détection\_incendie ] wikipedia\_Système\_de\_détection\_incendie. *détection incendie*. [https://fr.wikipedia.org/wiki/Système\\_de\\_détection\\_incendie](https://fr.wikipedia.org/wiki/Système_de_détection_incendie). (31/01/2017 à 21:58)
- [wikipedia\_Ville\_intelligente ] wikipedia\_Ville\_intelligente. *Ville intelligente*. [https://fr.wikipedia.org/wiki/Ville\\_intelligente](https://fr.wikipedia.org/wiki/Ville_intelligente). (29/01/2017 à 17:26)

# ANNEXE

## LA FICHE TECHNIQUE DE RASPBERRY PI

FICHE TECHNIQUE		0803/PTT   FICHE TECHNIQUE   ARM   HAUT DÉBIT
<a href="#">Sélectionnez une option</a>		
<b>INFORMATIONS GÉNÉRALES</b>		
Désignation	Raspberry Pi Model B+	
Marque	Raspberry	
Modèle	RASPBERRY-PI-B+	
<b>PROCESSEUR</b>		
Support du processeur	N/A	
<b>CHIPS ET</b>		
Nom courant du chip set	N/A	
<b>GRAPHIQUE</b>		
Contrôleur graphique intégré	0x1	
<b>CONNECTIQUE</b>		
Connecteur(s) USB 0 (Max)	4	
Connecteurs panneau arrière	1 x Carque (jack 3.5mm Femelle), 1 x RJ45 Femelle, 4 x USB 2.0, 1 x HDMI Femelle, 1 x Vidéo composite (RCA Femelle)	
Porte USB	USB 2.0	
<b>CARACTÉRISTIQUES PHYSIQUES</b>		
Format de carte mère	Spécifique	
<b>EQUIPEMENT</b>		
Fonctionnalité d'Overclocking avancée	Non	
Utilisation	Mini-mémoire	
<b>GARANTIES</b>		
Garantie commerciale		
Garantie légale	Voir les modalités	
Raspberry Pi Model B+ référencé le : 25/01/2014   Dernière modification le : 25/03/2016		

FIGURE V.29 – La fiche technique de Raspberry Pi

### Carte mère Raspberry Pi Type B+

- Carte mère **Raspberry Pi Type B+**
- **Processeur intégré** ARM1176JZF-S core (ARM11) **700 MHz**
- **RAM : 512 Mo**
- **GPU Broadcom VideoCore IV**
- Lecteur de cartes **Micro SD**
- Ports : **HDMI, 4x USB, RJ45, jack 3.5 mm, vidéo composite RCA**

FIGURE V.30 – La fiche technique de la carte mère

## NOTRE MAISON INTELLIGENTE



FIGURE V.31 – Vue 1

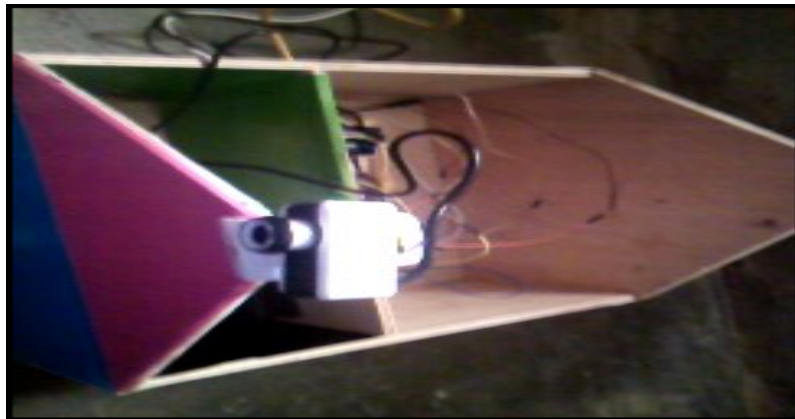


FIGURE V.32 – Vue 2



FIGURE V.33 – Vue 3



**Titre** Automatisation d'une maison intelligente via une application android

**Résumé** Le concept de l'Internet des Objets vise à intégrer le monde virtuel de l'Internet avec le monde physique des objets intelligents communicants, tels que les capteurs et les actionneurs, afin d'assurer une meilleure accessibilité et exploitation des ressources du monde réel. À travers un tel paradigme, une gigantesque expansion de l'Internet d'aujourd'hui est anticipée avec de nouveaux domaines d'applications comprenant la surveillance, la sécurité, la santé, les maisons et villes intelligentes, l'agriculture de précision, et les systèmes de logistique et de transportation intelligents. Dans ce contexte, l'objectif de ce projet de fin d'étude consiste à réaliser une maison intelligente équipée des LEDs, d'une porte, de quelques capteurs de température de fumée et d'une caméra de surveillance grâce au fameux mini-pc Raspberry Pi et contrôlé à distance par une application Android.

**Mots-clés** Internet des Objets, Raspberry Pi, Réseau de capteurs et d'actionneurs, Android, Python, PHP, JSON, java, MySQL.

**Title** Automatisation of a smart house using android application

**Abstract** The concept of the Internet of Things aims at integrating the virtual world of the Internet with the physical world of communicating intelligent things, such as sensors and actuators, in order to ensure better accessibility and exploitation of the resources of the real world. Through such a paradigm, today's massive expansion of the Internet is anticipated with new areas of application including surveillance, security, health, smart homes and cities, precision agriculture, and intelligent logistics and transportation systems. In this context, the aim of this end-of-study project is to produce a smart house equipped with LEDs, a door, some smoke and temperature sensors and a surveillance camera thanks to the famous mini-pc Raspberry Pi and remotely controlled by an Android application.

**Keywords** Internet of Things, Raspberry Pi, Network of sensors and actuators, Android, Python, PHP, JSON, java, MySQL.

**العنوان** أتمتة منزل ذكي عن طريق تطبيق أندرويد

**الملخص** مفهوم إنترنت الأشياء هو دمج العالم الافتراضي للإنترنت مع العالم المادي من الكائنات الذكية المترابطة مثل أجهزة الاستشعار والمحركات، لضمان تحسين إمكانية الحصول عليها واستغلال موارد العالم الحقيقي. من خلال مثل هذا النموذج، من المتوقع أن يكون هناك توسعاً هائلاً للإنترنت مع مجالات جديدة من التطبيقات بما في ذلك المراقبة والأمن والصحة والمنازل والمدن الذكية، الزراعة الدقيقة، والخدمات اللوجستية وأنظمة النقل الذكية

وفي هذا السياق، فإن الهدف من هذه الدراسة النهائية للمشروع هو صناعة منزل ذكي مجهز بمصابيح و باب و بعض أجهزة استشعار درجة الحرارة والدخان وكاميرا للمراقبة وذلك بفضل الجهاز الشهير الذي هو عبارة عن كمبيوتر صغير يدعى بالراسبيري بي والتحكم فيها عن بعد عن طريق تطبيق الأندرويد

**الكلمات المفتاحية** إنترنت الأشياء، راسبيري باي، شبكة الحساسات و المحركات، أندرويد، باين، ب أش ب، جيسن، جافا، ماي أسكول