

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



*Institut Supérieur des Sciences
Appliquées et de Technologie de Sousse
Université de Sousse*



DEPARTEMENT INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ETUDES

En vue de l'obtention du:

Diplôme National d'Ingénieur Informatique

Option: Informatique Industrielle

Système Embarqué Orienté vers les Applications Domotiques

Elaboré par :

Hafedh Ben Hassen

Soutenu le 17/06/2015 devant le jury :

Président : M. Khaled Ben Khalifa ISSAT de Sousse

Examinateur : M. Chokri Souani ISSAT de Sousse

Encadreur : M. Bairem Ben Abdallah ISSAT de Sousse

Année Universitaire: 2014/2015

DEDICACES

Je dédie ce travail

À mes chers parents, À ma famille

À tous ceux qui je sont chers

« Que dieu vous garde »

REMERCIEMENTS

« La Louange est à Dieu seul »

Je tiens à exprimer en premier lieu ma profonde gratitude et ma vive reconnaissance à **M. Bayrem Ben Abdallah** mon encadreur, enseignant à l’Institut Supérieur des sciences Appliquées et de Technologie de Sousse, d’avoir accepté de diriger ce travail.

J’adresse également, mes sincères remerciements à **M. Belgacem HAMDI**. Il s’est toujours montré à l’écoute et très disponibles tout au long de la réalisation de ce projet, ainsi pour l’inspiration, l’aide et le temps qu’ils ont bien voulu me consacrer.

Toute ma gratitude et mon Amour pour mes parents pour leur patience et soutien sans failles tout au long de mes études.

Enfin, je tiens à remercier toutes les personnes qui j’ont, de près ou de loin, d’une manière ou d’une autre, aidé à mener à bien ce travail.

RESUME

Ce travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme national d'ingénieur en informatique industrielle à l'institut supérieur des sciences appliquées et de technologie de Sousse (ISSATSo).

L'objectif de ce projet est de développer un système de gestion de la maison. Ce système permet de commander une multitude d'équipements (lampes, volets, portes) dans la maison et surveiller la température et l'humidité et détecter la fuite de gaz.

Via ce système, de n'importe endroit dans votre maison ou dehors, vous pourrez utiliser un navigateur web sur votre ordinateur ou Samrtphone/Tablet pour voir l'état/contrôler à distance les équipements reliés au système dans votre maison.

Pour la réalisation de ce projet, nous avons utilisé une carte Raspberry Pi comme micro-ordinateur et Python, PHP et Javascript comme langages de programmation et UML comme langage de modélisation.

Mots clés: Système Embarqué, Domotique, Contrôle, Raspberry Pi, Application Web.

SOMMAIRE

Dédicaces	i
Remerciements	ii
Résumé	iii
Sommaire.....	iv
Liste des Figures	vii
Liste des Tableaux.....	ix
Introduction Générale	1
CHAPITRE 1: Etat de l'art	3
1.1 Introduction :	3
1.2 Les systèmes embarqués :	3
1.2.1 <i>Généralités et avantages des systèmes embarqués</i>	3
1.2.2 <i>Domaine d'application des systèmes embarqués</i> :.....	4
1.2.3 <i>Architecture d'un système embarqué</i> :.....	5
1.2.4 <i>Evolution logiciels embarqués</i> :.....	5
1.3 La domotique :	6
1.3.1 <i>Définition</i> :.....	6
1.3.2 <i>Les applications de la domotique</i> :.....	7
1.3.3 <i>Différents types des systèmes domotiques</i> :.....	10
1.4 Solution proposée :	11
1.5 Conclusion.....	11
CHAPITRE 2: Contexte du Projet et étude Préalable	12
2.1 Introduction	12
2.2 Contexte du Projet :.....	12
2.3 Cahier des Charges :	12
2.3.1 <i>Objectifs</i> :.....	12
2.3.2 <i>Besoins Fonctionnels</i> :	13
2.3.3 <i>Besoins non fonctionnels</i> :	13

2.3.4	<i>Budget de temps</i>	14
2.4	Conclusion.....	14
CHAPITRE 3: Conception et Réalisation Partie Hardware		15
3.1	Introduction	15
3.2	Choix de plateforme :	15
3.2.1	<i>Arduino UNO</i> :	15
3.2.2	<i>BeagleBone</i> :.....	16
3.2.3	<i>Raspberry Pi</i> :.....	17
3.2.4	<i>Comparaison entre les trois plateformes</i>	18
3.2.5	<i>Plateforme choisi</i> :.....	19
3.3	Présentation et configuration du Raspberry Pi :.....	19
3.3.1	<i>Autres caractéristiques</i> :.....	19
3.3.2	<i>Accessoires du Raspberry Pi</i> :	19
3.3.3	<i>Pins GPIO</i> :.....	20
3.3.4	<i>Installation Raspbian sur la carte SD</i> :.....	21
3.3.5	<i>Premier démarrage du Raspberry Pi</i>	23
3.3.6	<i>Installation du serveur web et base de donne</i> :.....	26
3.3.7	<i>Installation WebIOPi</i> :.....	27
3.3.8	<i>Multiplexage des GPIO du Raspberry Pi</i> :	28
3.4	Schéma synoptique :	29
3.5	Conception et Réalisation du Module de contrôle :	29
3.5.1	<i>Carte à relais</i> :	29
3.5.2	<i>Conception du module de retour d'état</i> :	32
3.6	Autre composant utilisé (capteur de température) :	35
3.7	Conclusion.....	35
CHAPITRE 4: Conception et Réalisation Partie Software		36
4.1	Introduction	36
4.2	Etude conceptuelle :	36
4.2.1	<i>Langage de modélisation UML</i> : [1].....	36

<i>4.2.2 Diagramme de cas d'utilisations (Use Case) :</i>	37
<i>4.2.3 Diagramme de Classe:</i>	43
<i>4.2.4 Diagramme de séquence:</i>	44
4.3 Réalisation :	47
<i>4.3.1 Les outils de développement :</i>	47
<i>4.3.2 Interfaces de l'application :</i>	51
4.4 Conclusion :	58
Conclusions et Perspectives	59
Liste des acronymes	60
Références bibliographiques	61

LISTE DES FIGURES

<i>Figure 1 : La Domotique.....</i>	1
<i>Figure 2 : Architecture d'un système embarqué.....</i>	5
<i>Figure 3 : Les applications de la domotique.....</i>	7
<i>Figure 4 : Système EIB/KNX.....</i>	11
<i>Figure 5 : Arduino UNO.....</i>	16
<i>Figure 6 : BeagleBone Black</i>	16
<i>Figure 7 : Raspberry Pi.....</i>	17
<i>Figure 8 : Pins GPIO.....</i>	20
<i>Figure 9 : Choix de la carte SD dans la liste déroulante.....</i>	21
<i>Figure 10: Choix de l'image de l'OS.....</i>	22
<i>Figure 11 : Choix des options de la carte bootable</i>	22
<i>Figure 12 : Progression de l'écriture de l'image Raspbian sur la carte.....</i>	23
<i>Figure 13 : Configuration Raspberry.....</i>	23
<i>Figure 14 : Console de connexion.....</i>	25
<i>Figure 15 : Interface graphique Raspberry.....</i>	25
<i>Figure 16 : Test serveur web.....</i>	26
<i>Figure 17 : Test serveur webiopi.....</i>	27
<i>Figure 18 : Fiche technique du PCF8574P.....</i>	28
<i>Figure 19 : Schéma synoptique</i>	29
<i>Figure 20 : Schéma de simulation</i>	31
<i>Figure 21 : Routage de la carte</i>	31
<i>Figure 22 : Typon</i>	32
<i>Figure 23 : visualisation en 3D</i>	32
<i>Figure 24 : Simulation</i>	33
<i>Figure 25 : Routage de module de retour d'état.....</i>	34
<i>Figure 26 : Visualisation en 3D du module.....</i>	34
<i>Figure 27 : sonde numérique DS18B20.....</i>	35
<i>Figure 28 : Les trois vues de modélisation UML</i>	37

<i>Figure 29 : Diagramme de cas d'utilisation général.....</i>	38
<i>Figure 30 : Diagramme de Cas d'utilisation de package « Gestion des comptes utilisateurs» ..</i>	39
<i>Figure 31 : Diagramme de Cas d'utilisation de package « Gestion de la maison».....</i>	41
<i>Figure 32 : Diagramme des classes</i>	43
<i>Figure 33 : Diagramme de séquence« Authentification ».....</i>	45
<i>Figure 34 : Diagramme de séquence« Allumer/Eteindre lampe ».....</i>	46
<i>Figure 35 : Bootstrap.....</i>	47
<i>Figure 36 : Dreamweaver.....</i>	47
<i>Figure 37 : StarUML</i>	48
<i>Figure 38 : Apache HTTP Server.....</i>	48
<i>Figure 39 : MySQL.....</i>	49
<i>Figure 40 : phpMyAdmin</i>	49
<i>Figure 41 : Webiopi</i>	50
<i>Figure 42 : Python</i>	50
<i>Figure 43 : Interface d'accueil.....</i>	51
<i>Figure 44 : Interface d'authentification.....</i>	52
<i>Figure 45 : Interface ajout utilisateur.....</i>	53
<i>Figure 46: Interface modification mot de passe.....</i>	53
<i>Figure 47 : Interface suppression utilisateur.....</i>	54
<i>Figure 48 : Interface liste des utilisateurs.....</i>	54
<i>Figure 49 : Interface de contrôle des volets.....</i>	55
<i>Figure 50 : Interface d'ouverture de porte</i>	56
<i>Figure 51 : Interface de contrôle des lampes</i>	56
<i>Figure 52 : Interface détection de fuit de gaz.....</i>	57
<i>Figure 53 : Interface consultation température.....</i>	57
<i>Figure 54 : Interface consultation humidité</i>	58

LISTE DES TABLEAUX

<i>Tableau 1 : Budget de temps.....</i>	14
<i>Tableau 2 : Comparaison entre les trois plateformes.....</i>	18
<i>Tableau 3 : Liste des composants : carte à relais</i>	30
<i>Tableau 4 : Liste des composants : Module de retour d'état</i>	33
<i>Tableau 5 : Description de diagramme de cas d'utilisation général</i>	38
<i>Tableau 6 : Description de diagramme de Cas d'utilisation de package « Gestion des comptes utilisateurs»</i>	40
<i>Tableau 7 : Description de diagramme de Cas d'utilisation de package « Gestion de la maison».....</i>	42
<i>Tableau 8 : Description détaillé du diagramme de séquence« Authentification ».....</i>	45
<i>Tableau 9: Description détaillé du diagramme de séquence« Allumer/Eteindre lampe »</i>	46
<i>Tableau 10 : Les outils de développement.....</i>	50

INTRODUCTION GENERALE

Les progrès technologiques de l'informatique, des télécommunications et de l'électronique ont permis le développement des systèmes de transmission, des commandes à distance et ont offert de nouveaux services pour les occupants de logements. Ces services regroupés sous le terme « domotique », concernent principalement :

Le confort : Commande à distance d'appareils ou équipements.

L'économie d'énergie : Connaître vos consommations pour appliquer un modèle économique adapté.

La sécurité : Alarme incendie, alarme intrusion, surveillance vidéo...

L'aide à l'autonomie : Le contrôle distant de votre habitat depuis un terminal informatique.

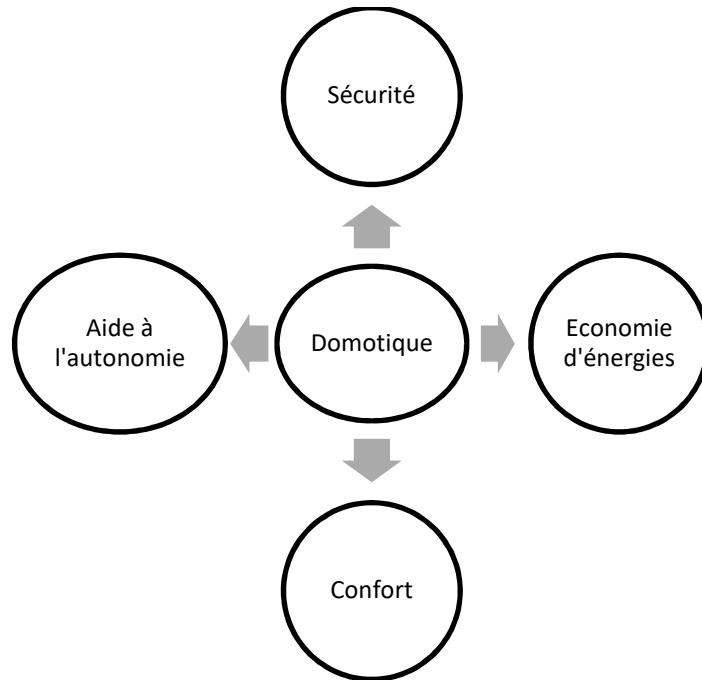


Figure 1 : La Domotique

Comme schématisé dans la figure ci-dessus, la domotique est l'ensemble des techniques et technologies permettant d'automatiser et d'améliorer les tâches au sein d'une maison ou d'un immeuble. À chaque besoin correspond une domotique spécifique, en effet, on peut la programmer selon les installations pour lesquelles on souhaite avoir une gestion simplifiée. Cette technologie apporte plus de confort dans l'éclairage, l'énergie, les ouvertures et fermetures ...

Ce travail sera organisé en quatre chapitres, nous présenterons en premier lieu l'état de l'art.

Le deuxième chapitre sera consacré pour le contexte et les objectifs du projet, son cahier des charges et ses besoins fonctionnels et non fonctionnels.

Le troisième chapitre fera l'objet de « l'étude conceptuelle et la réalisation de la partie Hardware », où l'on précisera l'aspect fonctionnel de notre solution.

Enfin, le dernier chapitre sera consacré à « l'étude conceptuelle et la réalisation de la partie Software », où l'on précisera les étapes essentielles pour réaliser une application Web .Puis, on clôturera notre rapport de fin d'études par une conclusion générale ainsi que les perspectives envisageables pour ce projet.

CHAPITRE 1: ETAT DE L'ART

1.1 *Introduction :*

Dans ce chapitre nous allons présenter l'état de l'art du domaine des systèmes embarqués et de la domotique. Nous allons mettre l'accent, en particulier sur ce qui existe sur le marché et ce qui motive les consommateurs dans leurs choix. Ensuite, nous allons faire, sommairement une étude critique de l'existant. Enfin, la dernière section fera l'objet de la solution que nous avons adoptée avec l'argumentation de ce choix.

1.2 *Les systèmes embarqués :*

1.2.1 *Généralités et avantages des systèmes embarqués*

Un système embarqué est tout système conçu pour résoudre un problème ou une tache spécifique qui utilise un microprocesseur combiné avec d'autres matériels et logiciels.

Il contient des parties matérielles et logicielles. Il contrôle dans certaines applications l'environnement extérieur en utilisant des capteurs et des actionneurs.

Les différents types des systèmes embarqués sont multiples on cite quatre groupes :

- Groupe dédié au calcul généraliste.
- Groupe dédié au contrôle de systèmes.
- Groupe pour le traitement du signal.
- Groupe pour les réseaux et communications.

- En général, les avantages des systèmes embarqués sont :
- La fiabilité
 - Faible consommation d'énergie
 - Utilisation efficace de la mémoire
 - La concurrence
 - Vivacité : le programme ne doit pas se terminer ou se bloquer dans l'attente d'événement qui ne va jamais s'effectuer
 - Hétérogénéité :
 - Différents style de programmation
 - Les interactions avec l'environnement extérieur
 - Réactivité : réaction continue avec l'environnement à la vitesse d changement de l'environnement.

1.2.2 Domaine d'application des systèmes embarqués :

Il existe 4 grandes classes :

- Calcul généraliste** : Similaire aux applications bureau mais embarqué (assistant personnel, téléphone portable, Consoles de jeux vidéo ...)
- Contrôle de systèmes** : Moteur, voiture, avion, processus chimique, nucléaire...
- Traitements du signal** : Compression vidéo, radar, flux de données, etc.
- Réseaux et communications** : Transmission de données, commutation, routage, téléphone, Internet, etc.

1.2.3 Architecture d'un système embarqué :

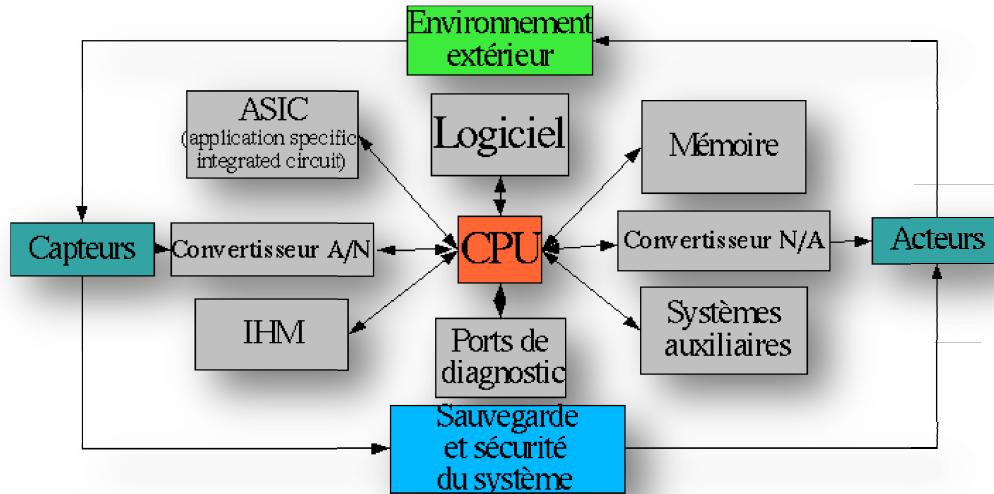


Figure 2 : Architecture d'un système embarqué

En entrée on trouve des capteurs généralement analogiques couplés à des convertisseurs analogiques/numériques. On retrouve en sortie des actionneurs généralement analogiques couplés à des convertisseurs numériques/analogiques. Au cœur du système, on trouve le calculateur qui s'appuie quelques fois sur des circuits du type FPGA ou ASIC. Les systèmes les plus simples comportent uniquement des boutons, des LED. Les systèmes les plus complexes peuvent avoir un écran tactile. En ce qui concerne le système auxiliaire, il peut s'agir de module qui gère la dissipation de chaleur par exemple. Le port de diagnostic permet de transférer des informations telles que les problèmes détectés dans le système.

1.2.4 Evolution logiciels embarqués:

Les systèmes et logiciels embarqués ont fortement évolué depuis leur apparition dans les années 60 avec quatre phases principales comme représentées ci-dessous :

- Années 1960 - 70 : Réalisation en logiciel de fonctions auparavant réalisées en matériel.

-Années 1970 - 80 : Ajouts de nouvelles fonctionnalités (non réalisables en matériel).

-Années 1980 - 2000 : Diversité des offres de logicielles propriétaires dédiée

-Années 2000 - : Croissance de l'offre des logiciels libres.

Un logiciel embarqué est open source, fiable, ne fini Jamais, se caractérise par une Faible consommation d'énergie, utilisation efficace de la mémoire, vivacité, utilise de différents style de programmation.

Donc les utilisateurs ont tendance à les utiliser de plus en plus et dans de divers domaines.

Parmi les logiciels embarqués on trouve :

- Debian « wheezy » est recommandé par la fondation Raspberry Pi avec sa version dédiée Raspbian.
- Fedora « Raspberry Pi Fedora remix ».
- ArchLinux est fonctionnel avec sa version ARM pour Raspberry Pi...

1.3 La domotique :

1.3.1 Définition :

La domotique est le domaine technologique qui traite de l'automatisation du domicile, d'où l'étymologie du nom qui correspond à la contraction des termes "maison" (en latin "domus") et «automatique». Elle consiste à mettre en place des réseaux reliant différents types d'équipement (électroménager, équipement domotique, etc.) dans la maison. Ainsi, elle regroupe tout un ensemble de services permettant l'intégration des technologies modernes et intelligentes dans la maison.

La domotique vise quatre grands secteurs qui sont :

-La santé (télésanté, télémédecine...).

-La sécurité (mise en place d'alarmes, de caméras IP ou d'autres équipements permettant la télésurveillance).

-Le confort de vie (planification automatique de tâches par exemple : préparer le café tous les matins, mettre l'alarme tous les soirs, etc).

-Les économies d'énergie (régulation du chauffage, lancement des tâches couteuses en énergie pendant les heures creuses de la STEG, ...).

1.3.2 Les applications de la domotique :

La domotique a connu une grande progression durant les dernières années. Elle est utilisée dans la plupart des secteurs de la vie jusqu'aux petites tâches dans un domicile.

La figure ci-dessus montre ce qu'offre un système domotique au client.

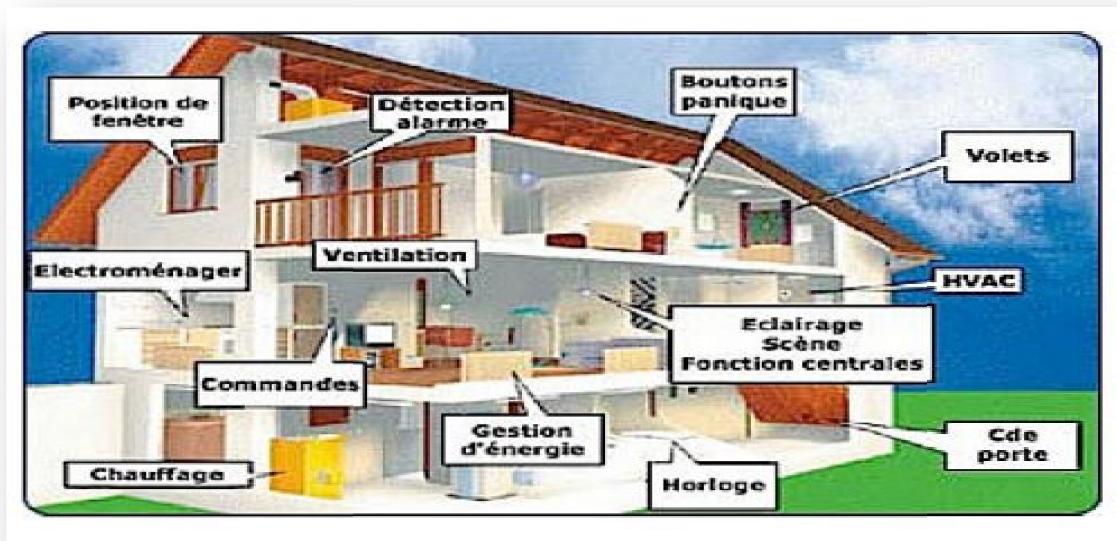


Figure 3 : Les applications de la domotique

Les services offerts par un système domotique sont les suivants :

-Service au sein de la maison: positionnement du store, éclairage des chambres, chauffage...

-Service à l'extérieur: télésurveillance, télésanté...

Les différents services peuvent être regroupés en quatre axes: Confort, sécurité, économie d'énergie et santé.

1.3.2.1 La sécurité :

Lorsqu'une personne quitte son domicile, elle souhaite qu'elle reste surveillée. On cherche à éviter les intrusions, les tentatives de violations. On a également d'autres types de problèmes : incendie, inondation, fuite de gaz...

La domotique fait appel à un panel très large de détecteurs, de capteurs et de caméras. On peut citer :

- Capteur de mouvements.
- Capteur d'ouverture de porte et fenêtre.
- Détecteur de fumée / feu.
- Détecteur d'humidité / inondation.
- Détecteur de pluie, de vent et de tempêtes.
- Caméras de vidéosurveillance.

Une autre forme de sécurité des demeures est le fait d'être capable d'agir sur elles à distance. Il va ainsi être possible de bloquer ou libérer les serrures à distance. Ici encore, on peut être alerté en cas de problèmes.

1.3.2.2 Le confort et bien-être : (Well-being at home)

Il existe de nombreuses applications domotiques pour le confort à domicile. Certaines actions peuvent être programmées. On peut par exemple programmer la préparation du café à la même heure tous les matins. On peut également citer tous les équipements électroménagers: tondeuse à gazon automatique, aspirateur qui détecte la saleté, ouverture et fermeture de portes de garages, etc. Dans ce domaine, on profite des contrôles à distance pour mieux maîtriser des appareils audio-visuels. On peut aussi utiliser cette application pour des fonctions domestiques plus avancées :

- Commander l'ouverture/fermeture des volets.

-Allumer la pièce sombre dès qu'une personne y entre.

-La gestion d'un centre medias.

Une application domotique peut passer du confort à l'obligation surtout pour les personnes âgées et handicapés.

1.3.2.3 L'économie d'énergie :

Un système domotique dispose de l'état énergétique d'un domicile et agit d'une manière intelligente pour l'économie d'énergie. La solution pour les économies de l'énergie et de minimiser le gaspillage; par exemple, on peut se trouver dans une pièce et voir si les lampes sont allumées dans les autres locaux sur un simple LED. Ainsi, on pourra piloter l'énergie pour qu'elle soit attribuée juste au besoin. Ces actions peuvent être gérées manuellement ou automatiquement par le système en place:

- Ne chauffer que les pièces occupées
- Gérer efficacement l'éclairage de façon automatique ; par exemple il n'y a plus de mouvement pendant une certaine durée de temps.
- Utiliser des lampes économiques.

1.3.2.4 La santé :

Cette application de la domotique est essentiellement prévue pour le suivi de personnes fragiles (personnes âgées, malades, handicap lourd, malentendants ou sourds...). On peut imaginer qu'un équipement installé sur une personne ou dans son domicile contrôle un certain nombre de paramètres: son rythme cardiaque, sa température corporelle, son taux de glycémie pour le cas d'un diabète, la qualité de l'air... On peut également imaginer un capteur sensoriel qui permet de détecter la perte de connaissance.

En fonction des résultats mesurés par ces détecteurs, il va être possible de remonter des alertes vers des organismes compétents (services d'urgence, médecins, les proches, etc.).

1.3.3 Différents types des systèmes domotiques :

Il existe de nombreux systèmes domotiques différents qui sont : Systèmes filaires, systèmes sans fil, systèmes radios.

1.3.3.1 Systèmes sans fil :

➤ Système Z-wave

Le Z-wave est un des systèmes sans fil utilisé pour rassembler tous les périphériques électroniques de la maison en un seul réseau. De plus ce système se base sur l'utilisation de protocole sans fil, dans ce cadre on peut citer la bande 868.42 à l'Amérique.

➤ Système Wifi

Le mot WI-FI est une abréviation de wireless fidelity qui dispose d'une fréquence au alentour de 2.4 Ghz qui peut être soit en infrastructure mode, soit en ad-hoc mode (peer-to-peer technologie) ou bien point-to-Multipoint. Il est bien sécurisé et il est utile pour se connecter à internet.

1.3.3.2 Systèmes filaires :

Pour les systèmes filaires on trouve : Système EIB/KNX et système X10.

➤ Le protocole de communication CPL :

CPL utilise le réseau électrique existant. Utiliser la domotique courant porteuse revient à transformer son habitat en maison communicante par le biais d'une installation domotique. La domotique CPL est aussi connue sous l'acronyme de X10, qui est un protocole de communication et de contrôle de plusieurs appareils domotiques.

➤ Système EIB/KNX :

EIB/KNX est l'intégration du bus EIB (Europen Instalation Bus) avec le protocole KNX.

Le système EIB/KNX est un système de communication normalisé qui a été créé par l'association KONNEX .Il est très utilisé dans la fabrication de réseaux domotiques.

Le système EIB/KNX utilise pour la communication entre tous les équipements de la maison une simple armoire électronique.

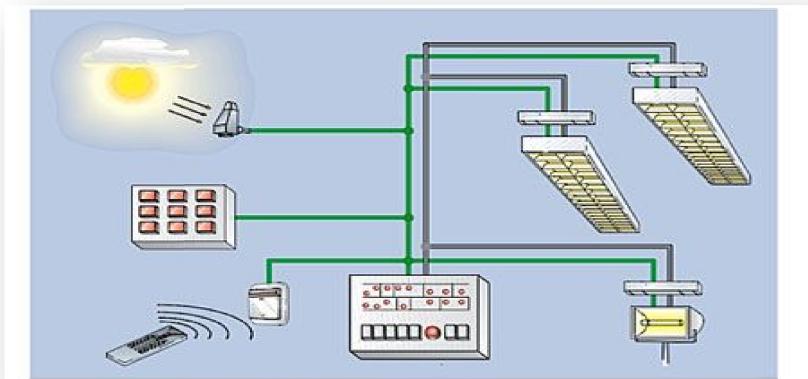


Figure 4 : Système EIB/KNX

1.4 Solution proposée :

Les solutions présentes nécessitent un grand nombre de périphériques (module X10, interrupteur CPL...). En plus, nous devons maîtriser les protocoles de communication associés à ces solutions et acquérir les équipements nécessaires. Ce qui augmente considérablement le coût et la complexité de la solution domotique.

Ainsi, on a choisi de développer une nouvelle solution câblée en utilisant uniquement un nano-ordinateur qui est de faible coût et simple à utiliser et des périphériques simples de type relais, capteur de température...

1.5 Conclusion

Dans ce chapitre nous avons présenté l'état de l'art du domaine des systèmes embarqués et de la domotique. Le chapitre suivant sera dédié à la cadre globale du projet, le cahier des charges ainsi que ses besoins fonctionnels, ses besoins non fonctionnels et son Puget de temps.

CHAPITRE 2: CONTEXTE DU PROJET ET ETUDE PREALABLE

2.1 *Introduction*

Dans ce chapitre, nous présenterons d'abord le cadre global du projet. Puis le cahier des charges de notre projet de fin d'études ainsi que ses besoins fonctionnels, ses besoins non fonctionnels et son budget de temps.

2.2 *Contexte du Projet :*

Dans le cadre de la formation d'ingénieur en informatique industrielle à l'Institut Supérieur des Sciences Appliquées et de la Technologie de Sousse (ISSATSo), nous avons eu l'occasion d'effectuer notre projet de fin d'études pour l'obtention du diplôme d'ingénieur national en Informatique Industrielle.

Généralement, ce projet vise à compléter notre formation universitaire acquise, durant trois ans, au sein de cet établissement, et de nous introduire dans la vie professionnelle grâce à une mise en pratique de nos connaissances, à l'utilisation des compétences acquises et une mise à épreuve de notre esprit d'ingénieur.

Le projet consiste à concevoir et réaliser un système intelligent de gestion de différents équipements de la maison localement et à distance a fin d'assurer le contrôle et la consultation des différents équipements, consultation de la température et l'humidité de la maison et la détection de fuit de gaz.

2.3 *Cahier des Charges :*

2.3.1 *Objectifs:*

Ce projet a plusieurs objectifs :

- Contrôler à distance les lampes
- Contrôler à distance les volets
- Contrôler à distance la porte

- Consulter à distance température
- Consulter à distance humidité
- Contrôler à distance la fuite de gaz

2.3.2 Besoins Fonctionnels :

- Le contrôle des équipements à distance : L'utilisateur peut utiliser son navigateur web soit avec un téléphone ou un ordinateur via une application «Web» pour commander à distance et gérer les équipements dans son domicile. Ces équipements sont les lampes, les volets et la porte.
- La consultation de la température et l'humidité de la maison à distance.
- La détection de fuite de gaz à distance.

2.3.3 Besoins non fonctionnels :

- Facilité d'utilisation : L'application doit avoir une interface simple pour permettre à n'importe quel utilisateur, qui en a besoin, de l'utiliser.
- Rapidité du temps de réponse : Les tâches doivent être exécutées dans un délai raisonnable.
- Modularité du code : Le code de l'application doit être modulaire et évolutif afin de favoriser l'intégration de nouvelles fonctionnalités.
- Disponibilité et fiabilité : Le système doit répondre à toute intervention instantanée de l'utilisateur. Aucun élément ne peut occasionner une panne de tout le système.

2.3.4 Budget de temps

	Mars		Avril		Mai		Juin
Étude des besoins et spécification	■	■					
Recherche et documentation	■	■	■	■	■	■	■
Proposition de différentes solutions et méthodes de travail			■	■			
Élection de la solution à adopter				■			
Réalisation de manipulations sur la plateforme utilisée				■	■		
Travail sur la carte électronique : Montage, Routage et soudure...					■	■	
Travail sur l'application web					■	■	
Élaboration du rapport						■	■

Tableau 1 : Budget de temps

2.4 Conclusion

Dans ce chapitre, nous avons commencé par présenter le cadre du projet puis nous avons décrit le système réalisé durant ce projet.

Le chapitre suivant sera consacré à une étude plus détaillée de notre projet.

CHAPITRE 3: CONCEPTION ET REALISATION

PARTIE HARDWARE

3.1 *Introduction*

La réalisation pratique est l'aboutissement logique d'un travail de conception bien fait. Dans ce chapitre nous allons en premier lieu justifier le choix de la plateforme utilisée. Ensuite, nous allons citer des configurations nécessaires pour le démarrage et l'utilisation de Raspberry. Enfin, nous passerons à décrire brièvement les différentes étapes de réalisation du projet à partir d'un schéma synoptique.

3.2 *Choix de plateforme :*

Durant la phase de proposition de différentes solutions, nous avons trouvé plusieurs plateformes adoptables à notre projet. Nous citons :

- Arduino UNO
- BeagleBone
- Raspberry Pi

3.2.1 *Arduino UNO :*

L'Arduino est un concept à la fois :

- matériel : une carte d'entrées/sorties avec un microcontrôleur programmable.
- logiciel : du C++ simplifié et une interface utilisateur simple

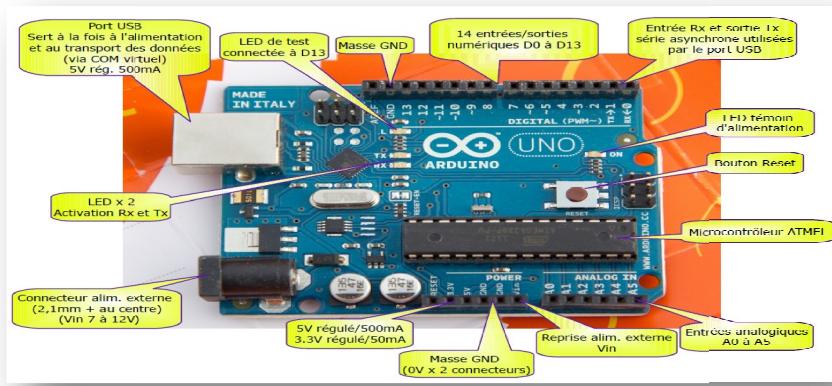


Figure 5 : Arduino UNO

3.2.2 BeagleBone :

Il ne s'agit pas d'une carte avec microcontrôleur (Atmel sur Arduino, ...) mais d'une carte avec un micro-processeur. Cela signifie que cette carte fonctionne avec un système d'exploitation.

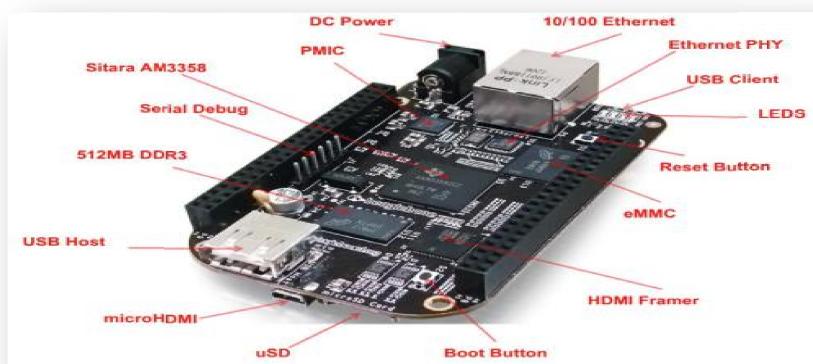


Figure 6 : BeagleBone Black

3.2.3 Raspberry Pi :

Raspberry Pi est un micro-ordinateur de la taille d'une carte de crédit disposant d'un processeur ARM 32bits. Cette carte est essentiellement un système sur puce et des ports de connexion. Elle se connecte sur un téléviseur ou sur n'importe quel autre dispositif d'affichage disposant d'une entrée.

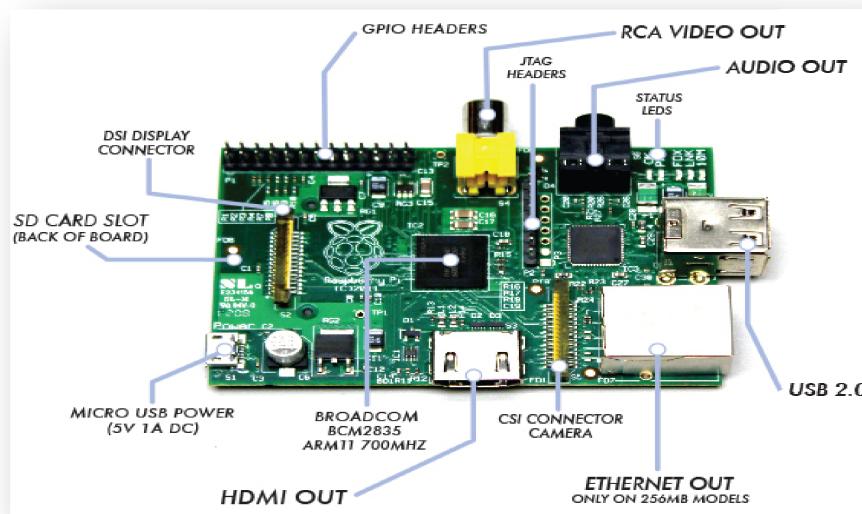


Figure 7 : Raspberry Pi

3.2.4 Comparaison entre les trois plateformes

	Arduino Uno	BeagleBone	Raspberry Pi (Model B)
Naissance	2005	2008	2008
Prix	30\$	45\$	40\$
Taille	45.43*32.24 mm	86.36*53.34 mm	85.60*53.98 mm
Processeur	ATmega328 8bits 16MHz	Cortex A8 720 Mhz	ARM11 à 700 Mhz
Mémoire	2Ko RAM, 1Ko EEPROM	512Mo	512 Mo
Mémoire Flash	32Ko	Sur MicroSD	Sur carte SD
Tension d'entrée	7v-12v	5v-3.3v	5v
Consommation	42mA (0.5W)	210 à 450mA (2.5W)	700 mA(3.5W)
Ethernet	Non	10/100 Ethernet	10/100 Ethernet
USB	Non	1 USB 2.0	2 USB 2.0
Sortie vidéo	Non	Micro HDMI	RCA et HDMI
Développement	Langage de programmation arduino	Environnement BoneScript,Langage Python,Scratch	Langage Python,Scratch
Disponibilité en Tunisie	OUI	Rarement	OUI

Tableau 2 : Comparaison entre les trois plateformes

3.2.5 Plateforme choisi :

On remarque d'après le tableau ci-dessus que chaque plateforme a des avantages par rapport aux autres.

La carte Arduino admet une mémoire de petite taille et un processeur de faible fréquence de calcul par rapport aux deux autres plateformes, de plus elle ne possède pas de port Ethernet.

La Raspberry Pi et la BeagleBone ont presque les mêmes caractéristiques et satisfont toutes les deux nos besoins sauf que la Raspberry Pi est moins chère et qu'elle est disponible partout en Tunisie.

Nous avons choisi donc la carte Raspberry Pi.

3.3 Présentation et configuration du Raspberry Pi :

3.3.1 Autres caractéristiques :

De plus des caractéristiques citées précédemment, la Raspberry Pi possède :

- Une prise femelle Micro-USB pour l'alimentation
- Une embase de GPIO et bus série.
- Une empreinte d'embase pour la connexion de caméra

3.3.2 Accessoires du Raspberry Pi :

Pour démarrer la Raspberry Pi il nous faut plusieurs accessoires :

- Chargeur Micro USB 5V – 1.5A.
- Carte SD (4Go suffisent mais 8Go sont préférables).
- Câble HDMI ou Câble audio/vidéo RCA.
- Clavier et souris USB.

Il existe d'autres accessoires que nous pouvons utiliser selon le besoin, nous citons par exemple :

- Un câble réseaux RJ45/ Dongle WIFI.

- Un boitier.

3.3.3 Pins GPIO :

La carte Raspberry Pi est équipée par des entrées et sorties numériques appelées GPIO (General Purpose Input and Output) qui sont contrôlées par le processeur ARM et fonctionnent en 3,3v sur Raspberry Pi. Elle possède 26 pins GPIO qui permettent de connecter une multitude d'équipements et périphériques : LED, transistor, radio, afficheur LCD ...



Figure 8 : Pins GPIO

3.3.4 Installation Raspbian sur la carte SD :

3.3.4.1 Outils nécessaires :

Pour pouvoir utiliser le micro-Pc Raspberry Pi, il faut tout d'abord installer le système d'exploitation convenable sur une carte SD en utilisant le logiciel «Linux live USB Creator» pour parcourir l'image du système déjà téléchargée à partir du site <http://www.raspberrypi.org/>.

Dans notre projet, nous avons utilisé le système d'exploitation «Raspbian Wheezy».

3.3.4.2 Etapes d'installation :

Une fois tous ces éléments récupérés, nous avons commencé par lancer Lili USB Craton, et dans le cadre « Étape 1 », nous avons choisi notre carte SD dans la liste déroulante.



Figure 9 : Choix de la carte SD dans la liste déroulante

Puis, dans le cadre « Étape 2 », nous avons cliqué sur « IMG/IMG/ZIP », et nous avons choisi notre img.

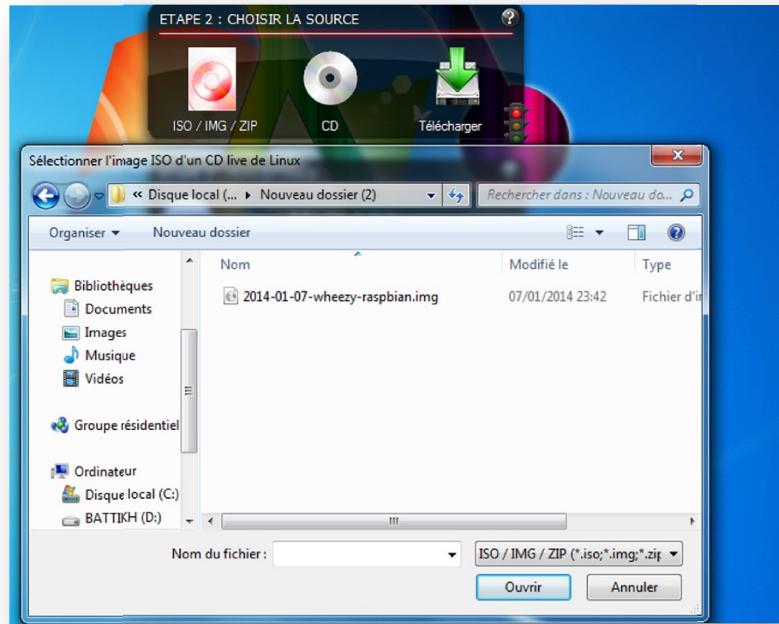


Figure 10: Choix de l'image de l'OS

Dans le cadre « Étape 4 », nous avons coché uniquement « Formater la clef en FAT32 ».

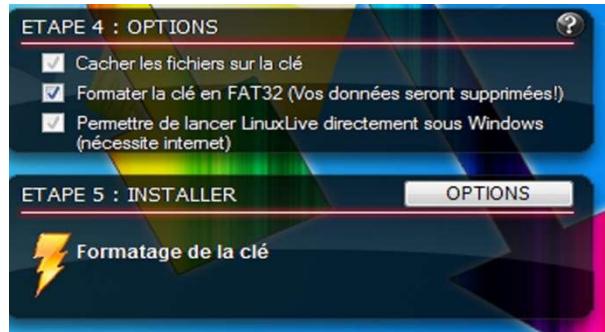


Figure 11 : Choix des options de la carte bootable

Enfin, nous avons cliqué sur l'éclair du cadre « Étape 5 », et nous avons validé le message d'avertissement pour lancer l'écriture de la carte.

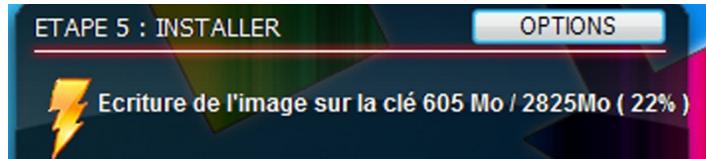


Figure 12 : Progression de l'écriture de l'image
Raspbian sur la carte

Une fois cette écriture finie, nous avons pu éjecter proprement la carte SD.

3.3.5 Premier démarrage du Raspberry Pi

Après avoir branché notre Raspberry Pi, l'interface ci-dessous apparaît



Figure 13 : Configuration Raspberry

On se retrouve face à un menu plutôt fourni permettant de configurer Raspbian :

- info : donne simplement des informations sur l'écran de paramétrage.
Cela n'est pas vraiment utile.

- `expand_rootfs` : par défaut, la distribution utilisera seulement 2 GB d'espace sur la carte SD. Si vous avez une carte avec plus d'espaces (4Go ou +), vous pouvez indiquer par cette commande à Raspbian d'utiliser tout l'espace disponible sur la carte.
- `overscan` : utile pour modifier l'affichage.
- `configure_keyboard` : pour configurer le clavier.
- `change_pass` : pour changer le mot de passe de l'utilisateur par défaut (pi)
- `change_locale` : pour définir le jeu de caractères à utiliser
- `change_timezone` : pour afficher l'heure en fonction de sa zone géographique
- `memory_split` : pour répartir la mémoire physique entre le GPU (processeur graphique) et le CPU (processeur). Par défaut, le GPU utilise 64Mo.
- `overclock` : permet d'augmenter la performance des processeurs et de la RAM en augmentant leur cadence.
- `ssh` : pour activer le SSH. Déjà activé par défaut.
- `boot_behaviour` : pour indiquer si l'on souhaite utiliser le Pi en mode graphique ou en mode texte. Le mode graphique est activé par défaut.
- `update` : pour mettre à jour le système

Une fois que nous avons terminé, il ne nous reste plus qu'à sélectionner « Finish ».

```

Debian GNU/Linux wheezy/sid raspberrypi tty1

raspberrypi login: pi
Password:
Last login: Tue Aug 21 21:24:50 EDT 2012 on tty1
Linux raspberrypi 3.1.9+ #168 PREEMPT Sat Jul 14 18:56:31 BST 2012 armv6l

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Type 'startx' to launch a graphical session

pi@raspberrypi ~ $

```

Figure 14 : Console de connexion

En tapant le login (pi) et le mot de passe (raspberry) par défaut suivi de la commande « startx » l'interface graphique du Raspberry Pi se lance.

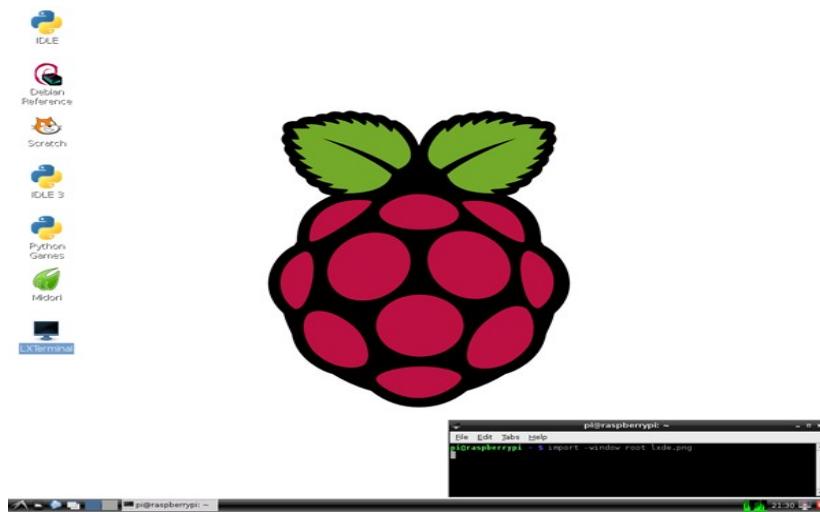


Figure 15 : Interface graphique Raspberry

3.3.6 Installation du serveur web et base de donne :

Nous avons installé sur la Raspberry Pi les éléments :

- Apache : Le serveur web HTTP.
- PHP : Le langage de script pour rendre un site dynamique.
- MYSQL : Le système de base de données.

En tapant la commande suivante :

```
pi@raspberrypi ~ $sudo apt-get install apache2 php5 mysql-server  
libapache2-mod-php5 php5-mysql
```

Une fois que l'installation est terminée, nous commencerons par vérifier que notre serveur fonctionne bien. Pour cela nous ouvrons le navigateur du Raspberry Pi et nous tapons dans la barre d'adresse : <http://localhost>.

Nous devrons avoir le résultat présenté par la figure suivante :

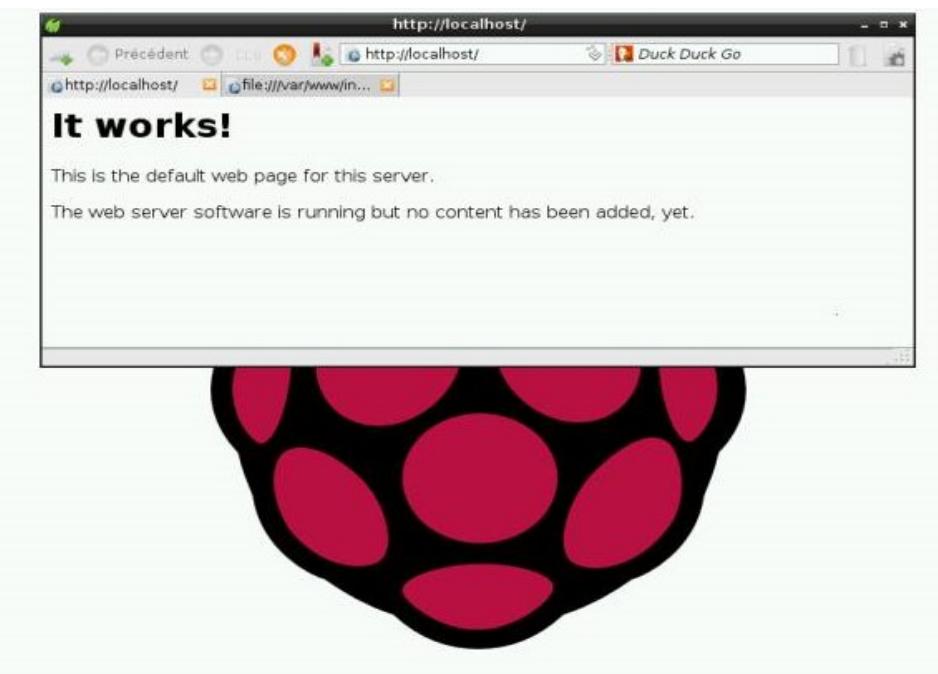


Figure 16 : Test serveur web

3.3.7 Installation WebIOPi :

Pour installer sur la Raspberry Pi le serveur WebIOPi en tapant la commande suivante :

```
→ $ wget http://webiopi.googlecode.com/files/WebIOPi- 0.6.0.tar.gz
→ $ tar xvzf WebIOPi-0.6.0.tar.gz
→ $ cd WebIOPi-0.6.0
→ $ sudo ./setup.sh
```

Pour mettre en service le WebIOPi à chaque démarrage de la carte en fait la commande suivante :

```
→$ sudo update-rc.d webiopi defaults
```

Une fois que l'installation est terminée, on vérifie que notre serveur fonctionne bien. Pour cela nous ouvrons le navigateur du Raspberry Pi et nous tapons dans la barre d'adresse : <http://localhost:8000/>.

Nous devrons avoir le résultat présenté par la figure suivante :

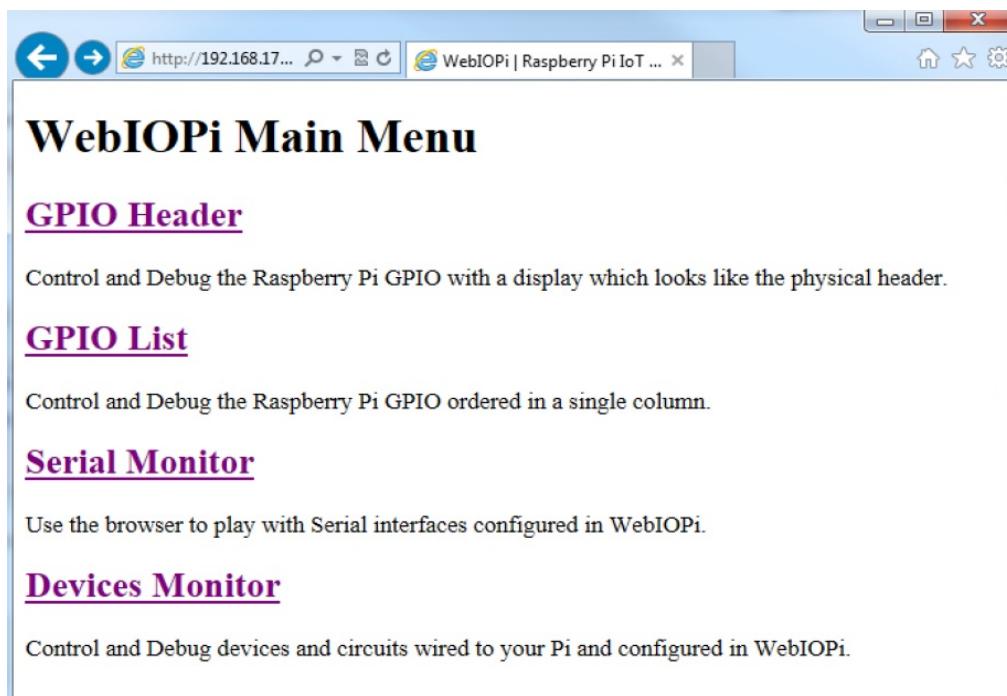


Figure 17 : Test serveur webiopi

3.3.8 Multiplexage des GPIO du Raspberry Pi :

La Carte Raspberry Pi possède 17 GPIO's configurable en entrée ou sortie. Dans un bâtiment nous pouvons trouver même une vingtaine d'équipements que nous ne pouvons pas contrôler en utilisant les GPIO de Raspberry, c'est pourquoi nous avons adopté la solution de multiplexage en utilisant le circuit intégré PCF8574P.

La puce PCF8574P permet d'ajouter 8 GPIO de plus et se connecte à Raspberry Pi en utilisant des GPIO's spéciaux dédiés au protocole I2C (pin 3 : SDA, pin 5 : SCL).

En utilisant cette méthode nous obtiendrons alors 23 GPIO (8+15).

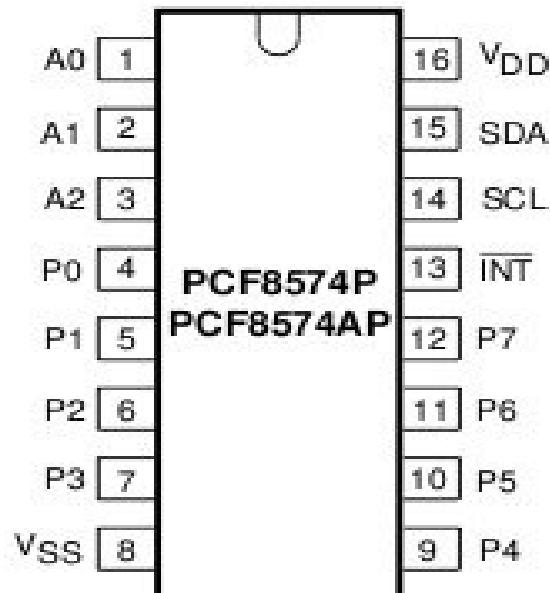


Figure 18 : Fiche technique du PCF8574P

3.4 Schéma synoptique :

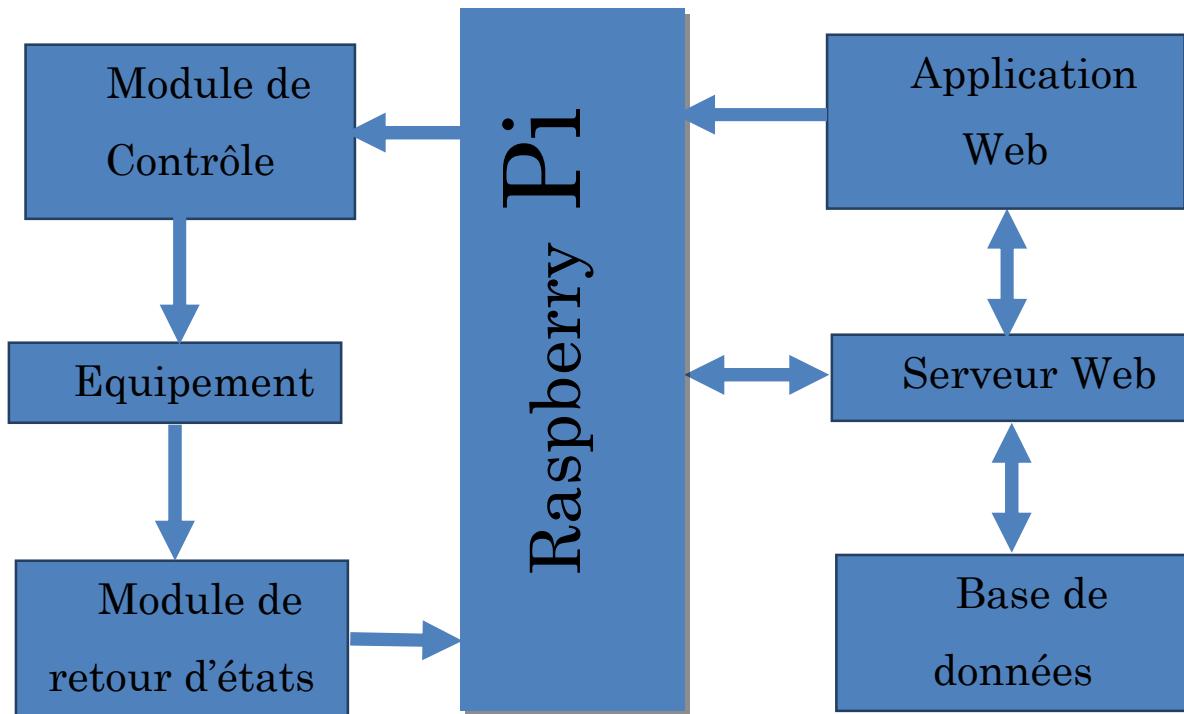


Figure 19 : Schéma synoptique

3.5 Conception et Réalisation du Module de contrôle :

3.5.1 Carte à relais :

Pour la commande automatique de l'équipement nous avons utilisé un relais piloté via un transistor de commande, une diode de roue libre pour la protection du relais et une résistance liée à la base du transistor pour le protéger contre la surintensité.

3.5.1.1 Liste des composants :

N°	Désignation	Référence	Valeur
1	Relais	SRD-05VDC-SL-C	5v
2	Transistor	2N2222	
3	Diode	1N4001	
4	Résistance		1K

Tableau 3 : Liste des composants : carte à relais

- Relais : Un relais est un commutateur électromécanique. Il se compose de deux parties : Une première partie de faible puissance contrôlée par Raspberry Pi et une deuxième partie est le commutateur qui peut supporter une puissance élevée.
- Transistor : Le relais nécessite 5v pour commuter tandis que le port GPIO de Raspberry Pi ne délivre que 3.3v. C'est pourquoi nous avons utilisé un transistor. Lorsque le transistor est saturé, le relais va s'alimenter directement du 5v de Raspberry Pi.

3.5.1.2 Etapes de réalisation de la carte à relais :

➤ Outils Utilisés :

Durant la phase de réalisation de notre carte à relais nous avons utilisé PROTEUS 8 qui est composé de deux logiciels principaux : ISIS, permettant la création des schémas et la simulation électrique, et ARES, dédié pour le routage des cartes électroniques.

➤ **Schema de simulation**

La figure ci-dessous représente le schéma de simulation de notre carte à relais :

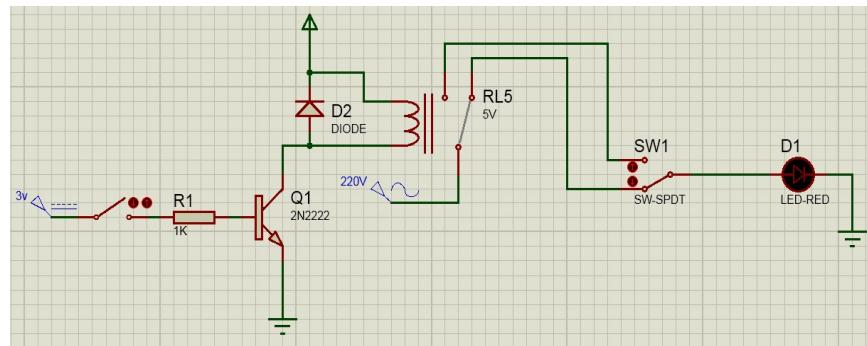


Figure 20 : Schéma de simulation

➤ **Routage de la carte :**

La figure ci-dessous représente le schéma de routage de notre carte à relais :

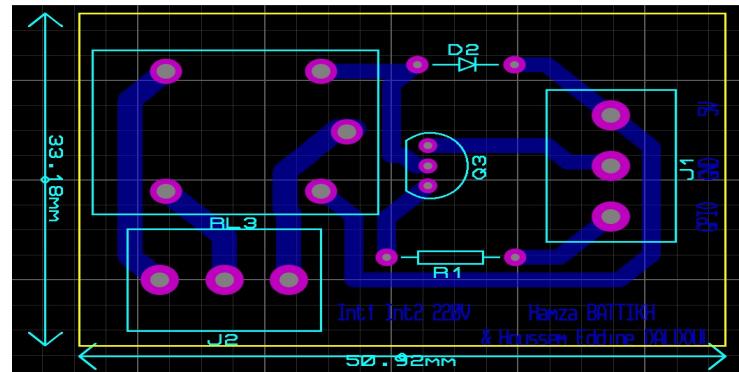


Figure 21 : Routage de la carte

➤ **Typon de la carte :**

Ce schéma représente le typon de la carte du système d'insolation :

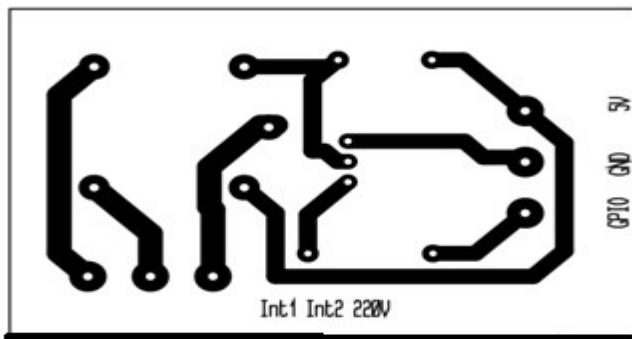


Figure 22 : Typon

➤ **Visualisation 3D :**

La visualisation en 3D de la carte est représentée par la figure ci-dessous :

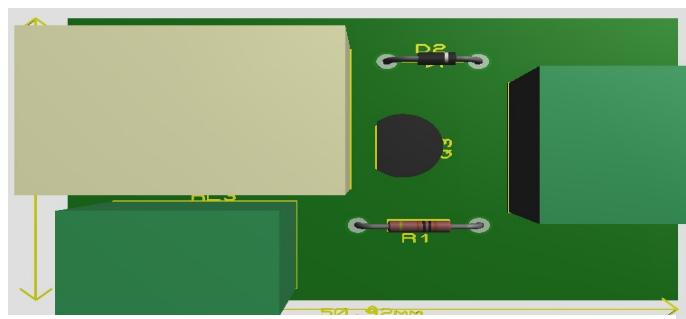


Figure 23 : visualisation en 3D

3.5.2 Conception du module de retour d'état :

Puisque nous contrôlons manuellement et automatiquement l'équipement, nous ne pouvons pas récupérer son état réel. C'est pourquoi nous avons trouvé la solution d'intégrer un module de retour d'état pour chaque équipement.

Ce module est relié à un chargeur qui délivre une tension de 5V. Puisque la carte raspberry ne supporte pas plus que 2.2V lorsqu'il est configuré comme "INPUT" le module présenté baisse la tension de 5 à 2V.

3.5.2.1 Liste des composants :

N°	Désignation	Référence	Valeur
1	Diode zener	1N4370A	
2	Résistance		100 Ω
3	Résistance		1K Ω
4	Connecteur USB		
5	Diode led		
6	Connecteur 4 pin		

Tableau 4 : Liste des composants : Module de retour d'état

3.5.2.2 Etapes de réalisation du module de retour état :

➤ Schema de simulation

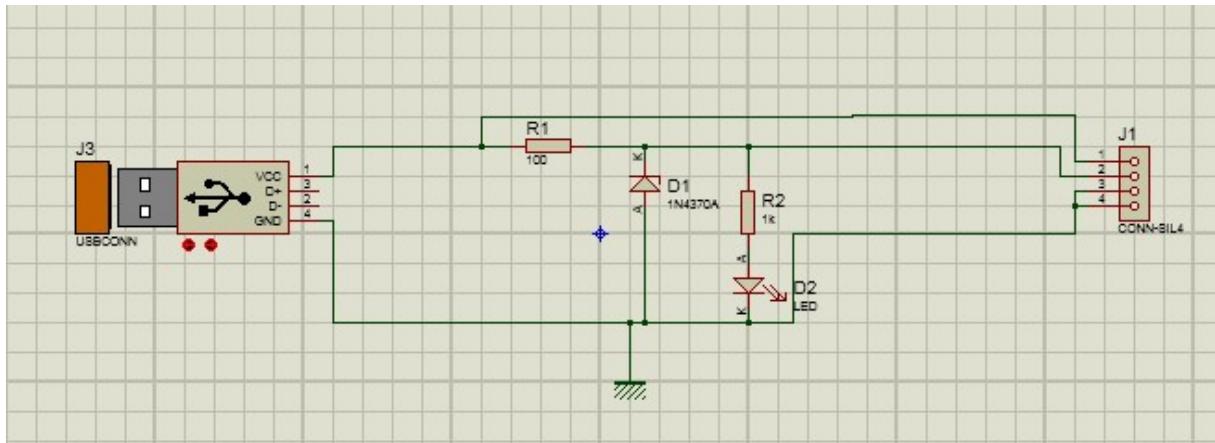


Figure 24 : Simulation

La diode zener sert à faire passer une tension de 2V à partir d'une tension stabilisée de 5V arrivé d'un chargeur et il est protégé par la résistance R1.

La diode led est allumée lorsqu'il y a une tension sur le pin de sortie de 2V.

➤ **Routage de la carte :**

La figure ci-dessous représente le schéma de routage de notre module :

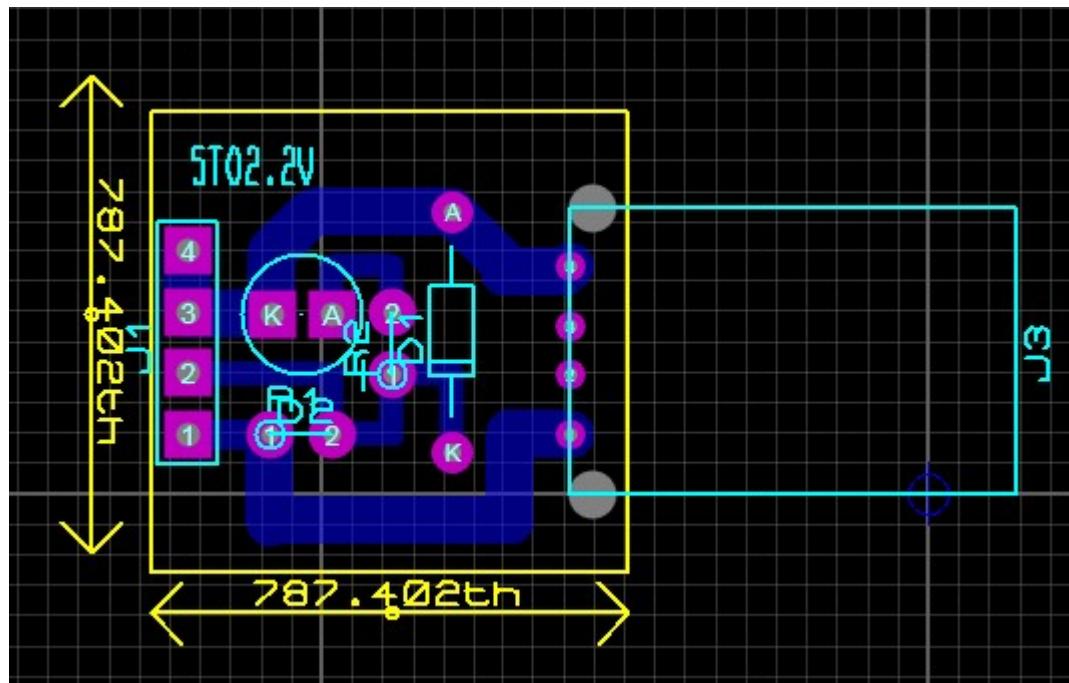


Figure 25 : Routage de module de retour d'état

➤ **Visualisation 3D :**

La visualisation en 3D du module est représentée par la figure ci-dessous :

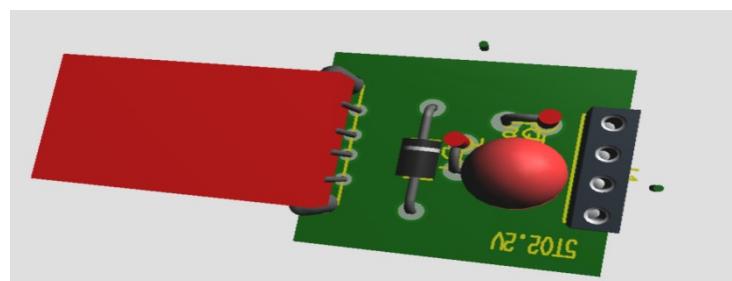


Figure 26 : Visualisation en 3D du module

3.6 Autre composant utilisé (capteur de température) :

Il faut savoir que le GPIO du Raspberry Pi ne dispose pas d'entrées analogiques, c'est pour cela il faut choisir un capteur numérique de température comme la sonde numérique DS18B20. Elle est une sonde numérique qu'utilise un bus/protocole particulier qui permet de communiquer l'information à travers un seul câble (par extension, une seule broche du port GPIO du Pi) ; il s'agit du bus 1-Wire.

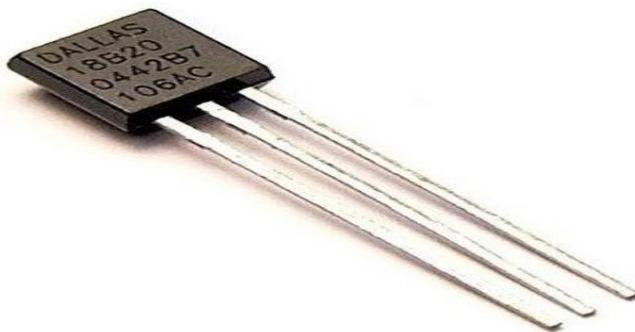


Figure 27 : sonde numérique DS18B20

3.7 Conclusion

Dans ce chapitre nous avons justifié le choix de Raspberry. Nous avons aussi présenté notre plateforme et cité quelques configurations nécessaires dont nous aurons besoin tout au long du projet. Par la suite nous avons présenté le schéma synoptique de notre projet et, enfin, la conception et la réalisation du module de contrôle, la conception du module de retour et on a parlé du capteur de température.

CHAPITRE 4: CONCEPTION ET REALISATION

PARTIE SOFTWARE

4.1 *Introduction*

Ce chapitre a pour objectifs d'exposer la réalisation de la partie logicielle de notre projet présentée sous forme d'une application web qui permet principalement de faire contrôler et consulter des équipements dans une maison à distant. Cette application communique avec la carte Raspberry Pi et avec le serveur web.

Pour ce faire, nous avons divisé ce chapitre en deux parties. La première partie contiendra l'étude conceptuelle de l'application. La deuxième sera consacrée pour la réalisation pratique.

4.2 *Etude conceptuelle :*

La conception est une phase primordiale et importante avant la réalisation, quel que soit le projet ou l'application. Un modèle ou un diagramme présente une réalité qui facilite la compréhension du système étudié puisqu'il réduit sa complexité. Nous avons utilisé UML comme méthode de modélisation.

4.2.1 *Langage de modélisation UML : [1]*

Le langage UML (Unified Modeling Langage), est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ».

Il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. Sa notation graphique permet d'exprimer visuellement une solution proposée évitant toute sorte d'ambiguïté et d'incompréhension. Ce langage décrit neuf diagrammes regroupés selon trois vues de modélisation présentées dans la figure ci-dessous :

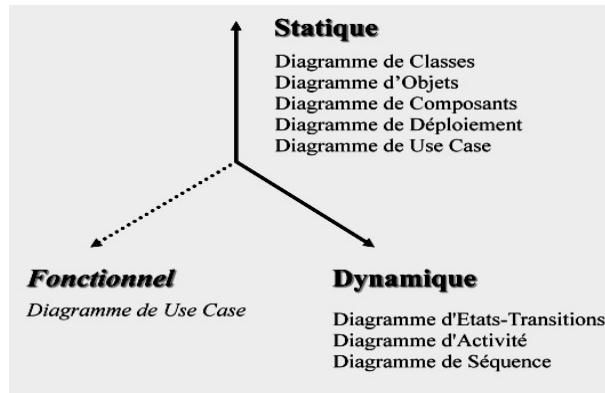


Figure 28 : Les trois vues de modélisation UML

4.2.2 **Diagramme de cas d'utilisations (Use Case) :**

4.2.2.1 Définition:

Un cas d'utilisation (Use Case) modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur [3].

4.2.2.2 Modélisation :

➤ Diagramme de Cas d'utilisation général

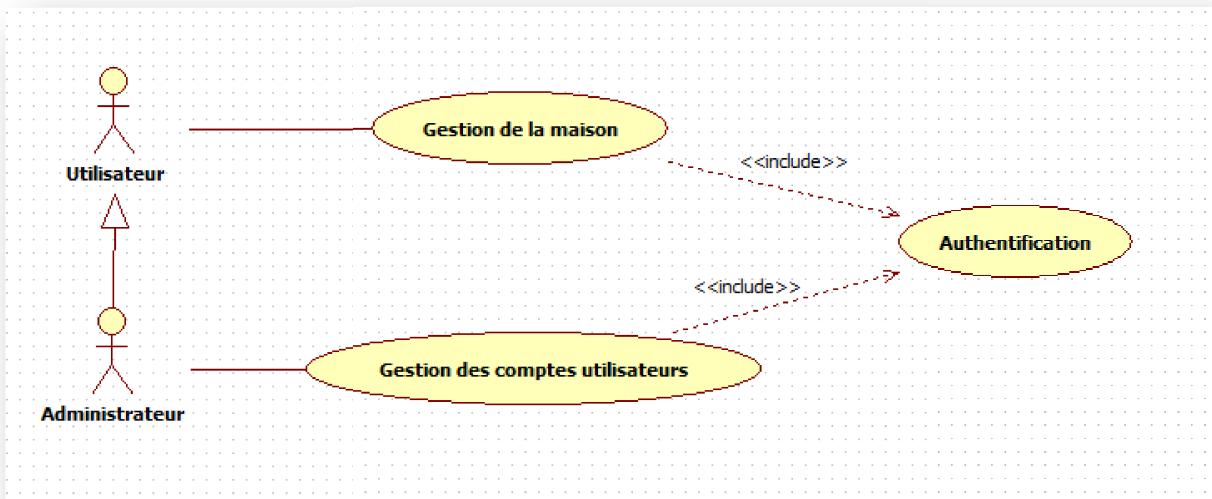


Figure 29 : Diagramme de cas d'utilisation général

Cas d'utilisation	Acteur	Description
Gestion de la maison	Utilisateur	Chaque utilisateur peut consulter et contrôler tous les équipements de la maison.
Gestion des comptes utilisateurs	Administrateur	L'administrateur peut ajouter, modifier, supprimer et consulter les utilisateurs.

Tableau 5 : Description de diagramme de cas d'utilisation général

- Diagramme de Cas d'utilisation de package « Gestion des comptes utilisateurs »

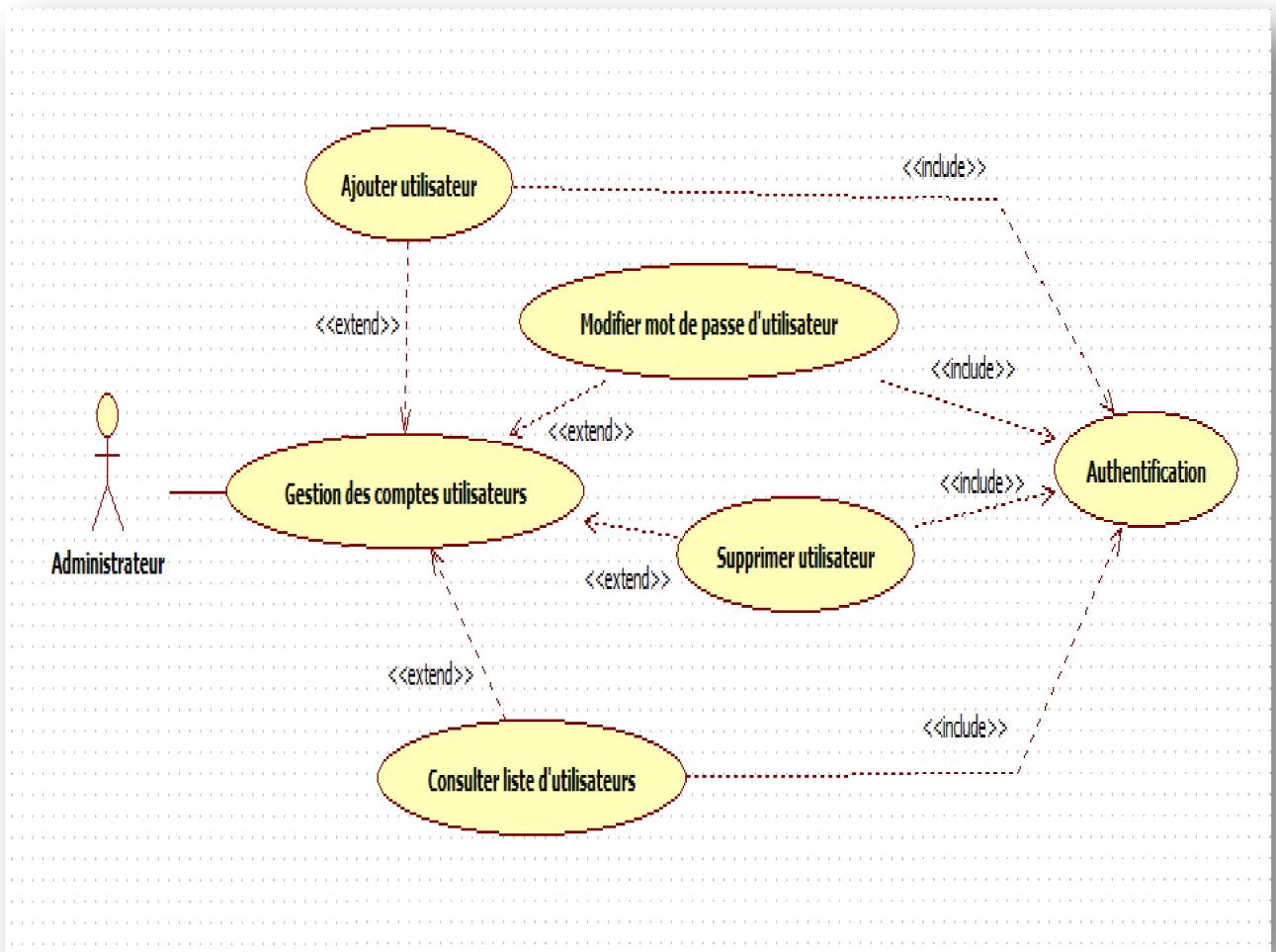


Figure 30 : Diagramme de Cas d'utilisation de package « Gestion des comptes utilisateurs»

Cas d'utilisation généralisé	Cas d'utilisation détaillé	Condition	Acteur	Description
Gestion des comptes utilisateurs	Ajouter utilisateur	Authentification	Administrateur	L'administrateur peut ajouter des comptes de nouveaux utilisateurs.
Gestion des comptes utilisateurs	Modifier mot de passe d'utilisateur	Authentification	Administrateur	L'administrateur peut modifier les mots des passes des comptes utilisateurs.
Gestion des comptes utilisateurs	Supprimer utilisateur	Authentification	Administrateur	L'administrateur peut supprimer des comptes utilisateurs.
Gestion des comptes utilisateurs	Consulter liste d'utilisateurs	Authentification	Administrateur	L'administrateur peut consulter la liste des comptes utilisateurs existants.

Tableau 6 : Description de diagramme de Cas d'utilisation de package « Gestion des comptes utilisateurs»

➤Diagramme de Cas d'utilisation de package «Gestion de la maison»

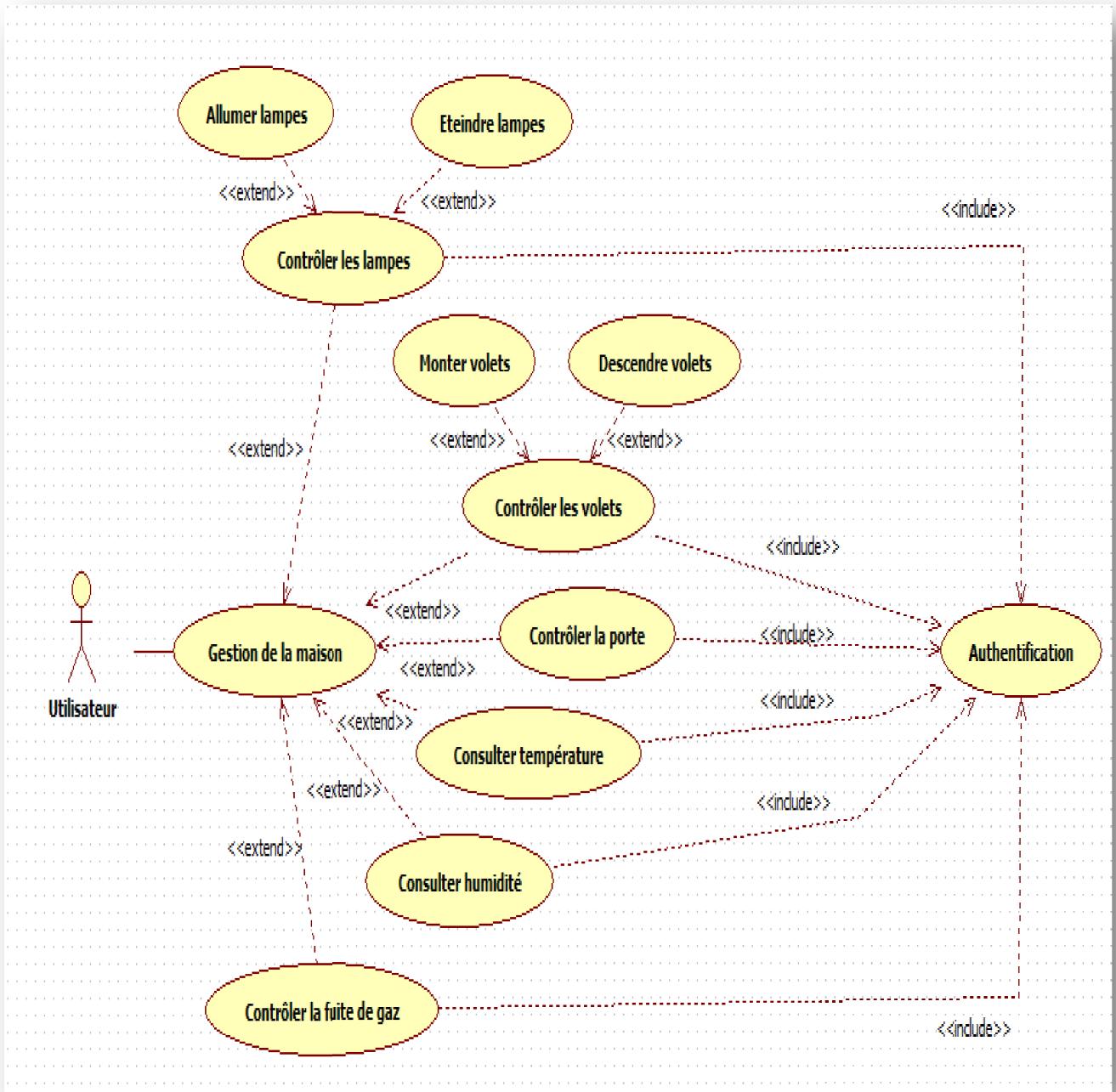


Figure 31 : Diagramme de Cas d'utilisation de package « Gestion de la maison»

Cas d'utilisation généralisé	Cas d'utilisation détaillé	Condition	Acteur	Description
Gestion de la maison	Contrôler les lampes	Authentification	Utilisateur	Chaque utilisateur peut allumer ou éteindre les lampes.
Gestion de la maison	Contrôler les volets	Authentification	Utilisateur	Chaque utilisateur peut Monter ou descendre les volets.
Gestion de la maison	Contrôler la porte	Authentification	Utilisateur	Chaque utilisateur peut ouvrir la porte de la maison.
Gestion de la maison	Consulter température	Authentification	Utilisateur	Chaque utilisateur peut consulter la température de la maison.
Gestion de la maison	Consulter humidité	Authentification	Utilisateur	Chaque utilisateur peut consulter l'humidité de la maison.
Gestion de la maison	Contrôler la fuite de gaz.	Authentification	Utilisateur	Chaque utilisateur peut Contrôler la fuite de gaz dans la maison.

Tableau 7 : Description de diagramme de Cas d'utilisation de package « Gestion de la maison»

4.2.3 Diagramme de Classe:

4.2.3.1 Définition :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier. [3]

4.2.3.2 Modélisation :

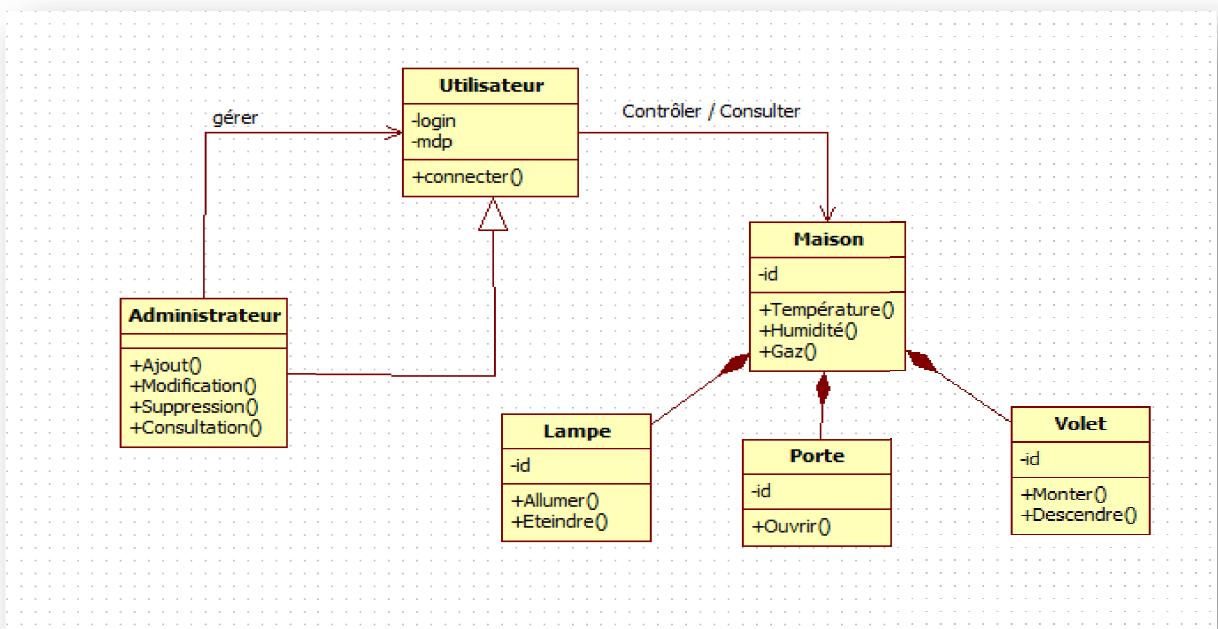


Figure 32 : Diagramme des classes

4.2.3.3 Description du diagramme de classes :

L'administrateur est un utilisateur particulier qui permet d'ajouter, modifier, supprimer et consulter des utilisateurs.

Chaque utilisateur possède un login et un mot de passe qui permet de contrôler et consulter la maison.

La maison est composée par des lampes, des volets et une porte et on peut mesurer la température et l'humidité et détecter la fuite de gaz aussi.

Chaque lampe possède un identifiant unique et on peut allumer ou éteindre cette lampe.

Chaque volet possède un identifiant unique et on peut monter ou descendre ce volet.

La porte possède un identifiant unique et on peut ouvrir la porte seulement.

4.2.4 Diagramme de séquence:

4.2.4.1 Définition :

Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but était de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages. [3]

4.2.4.2 Modélisation :

- Diagramme de séquence « Authentification » :

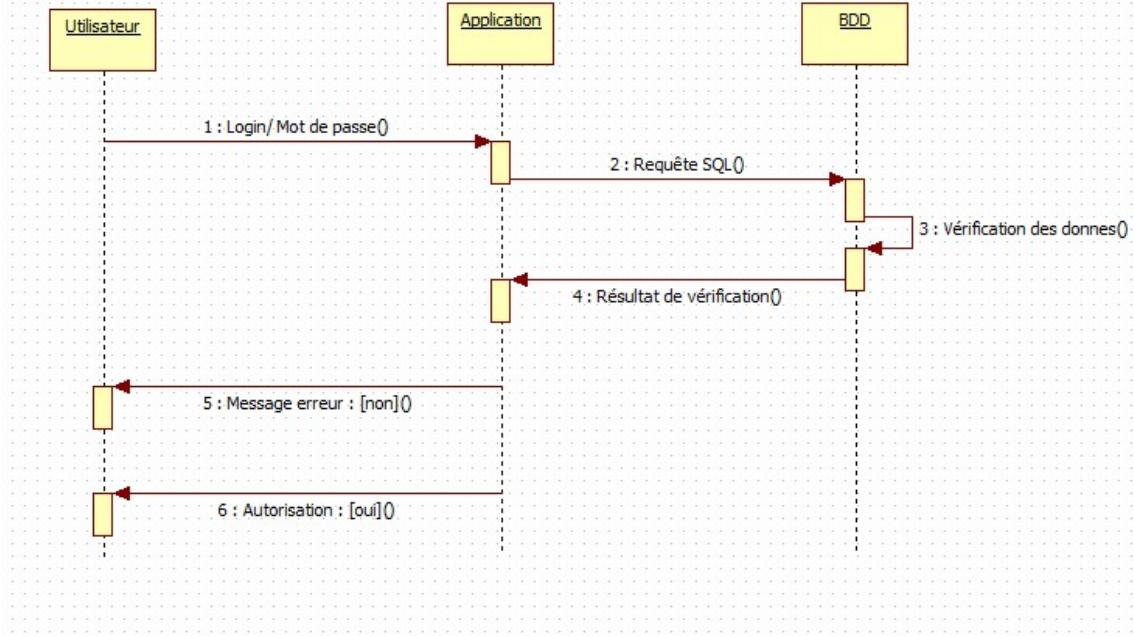


Figure 33 : Diagramme de séquence« Authentification »

Séquence Système	Authentification
Acteur Principal	Utilisateur / Administrateur
But	L'administrateur ou l'utilisateur connecter a l'application web.
Description	<ol style="list-style-type: none"> 1. L'utilisateur saisir login et mot de passe pour connecter. 2. L'application envoyer une requête SQL à la base de données contient l'information d'authentification. 3. Vérification des données. 4. Retour de résultat au système. 5. Si retour non on affiche message d'erreur. 6. Si retour oui on accédée a l'application.

Tableau 8 : Description détaillé du diagramme de séquence« Authentification »

➤ Diagramme de séquence « Allumer/Eteindre lampe» :

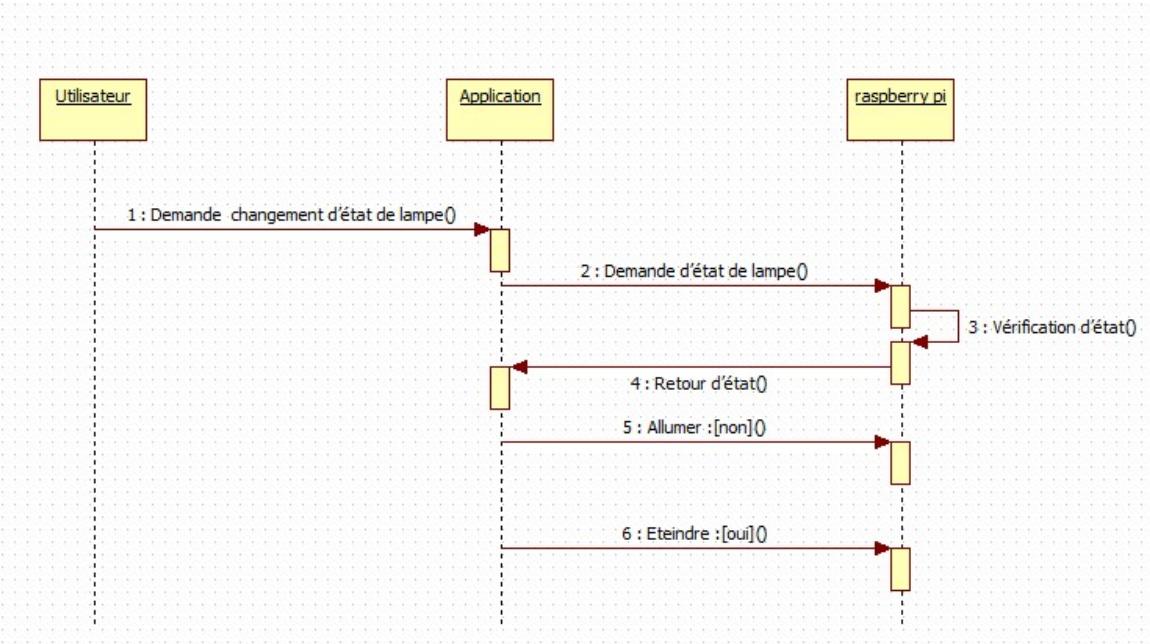


Figure 34 : Diagramme de séquence« Allumer/Eteindre lampe »

Séquence Système	Allumer/Eteindre lampe
Acteur Principal	Utilisateur
But	l'utilisateur allumer ou atteindre les lampes.
Description	<p>1. L'utilisateur faire une demande de changement d'état de lampe.</p> <p>2. L'application envoyer une demande de vérification d'état de lampe à la carte Raspberry Pi.</p> <p>3. La carte vérifier l'état de lampe.</p> <p>4. La carte retourner l'état de lampe à l'application.</p> <p>5. Si retour non l'application envoyer à la carte une demande d'allumer la lampe.</p> <p>6. Si retour oui l'application envoyer à la carte une demande d'éteindre la lampe.</p>

Tableau 9: Description détaillé du diagramme de séquence« Allumer/Eteindre lampe »

4.3 Réalisation :

4.3.1 Les outils de développement :

Les principaux outils de développement que nous avons utilisés dans la cadre de notre projet sont récapitulés dans le tableau ci-dessous.

Besoins	Outils
Interface web	 <p>Figure 35 : Bootstrap</p> <p>Bootstrap Twitter est une collection d'outils qui facilite la création d'un site web. Ce framework est composé entre autre de code HTML, Javascript et CSS et permet donc de créer des formulaires designés très facilement ainsi que l'implémentation d'éléments dynamiques comme un carrousel ou une modal. [3]</p>
Environnement de développement PHP	 <p>Figure 36 : Dreamweaver</p> <p>Adobe Dreamweaver est un programme de développement de site Web créé à l'origine par Macromedia. Il est maintenant développé par Adobe Systems, qui a acquis Macromedia en 2005. Dreamweaver est disponible pour les systèmes d'exploitation Windows et Mac. Les versions récentes de Dreamweaver ont intégré du soutien pour les</p>

	<p>différentes technologies Web telles que JavaScript, CSS, et divers langages de script côté serveur ainsi que les framework, y compris ASP, ColdFusion et PHP. [5]</p>
Conception de l'application Web	 <p>Figure 37 : StarUML</p> <p>StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL [2].</p>
Serveur web et base de données	 <p>Figure 38 : Apache HTTP Server</p> <p>Le logiciel libre APACHE HTTP SERVER (Apache) est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache. [6]</p>



Figure 39 : MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. [7]



Figure 40 : phpMyAdmin

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL. [8]

Interaction entre Raspberry Pi et l'application web	 <p>Figure 41 : Webiopi</p> <p>Webiopi est un serveur web spécial créé en langage python et JavaScript, il s'agit donc d'un service réseau, qui permet de faire la communication entre l'application web et la carte Raspberry Pi. [9]</p>
Langage de programmation Raspberry Pi	 <p>Figure 42 : Python</p> <p>Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. [10]</p>

Tableau 10 : Les outils de développement

4.3.2 Interfaces de l'application :

Nous avons choisi de créer une application web qui permet principalement de faire contrôler et consulter des équipements dans une maison à distance. Cette application communique avec la carte Raspberry Pi et avec le serveur web.

La page principale est l'interface d'accueil de cette application. Elle représente la première confrontation entre l'utilisateur et l'application. Cette page permet de choisir le langage de l'application.

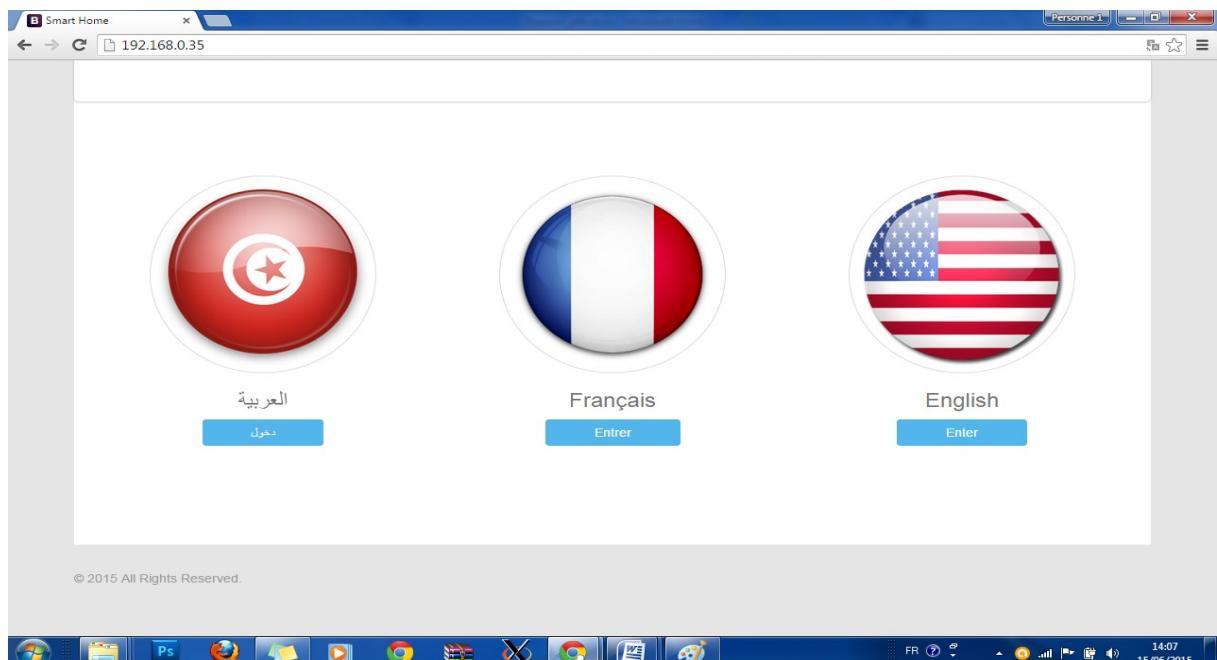


Figure 43 : Interface d'accueil

Après le choix de la langue on fait l'authentification pour connecter à l'application soit comme administrateur ou soit utilisateur.

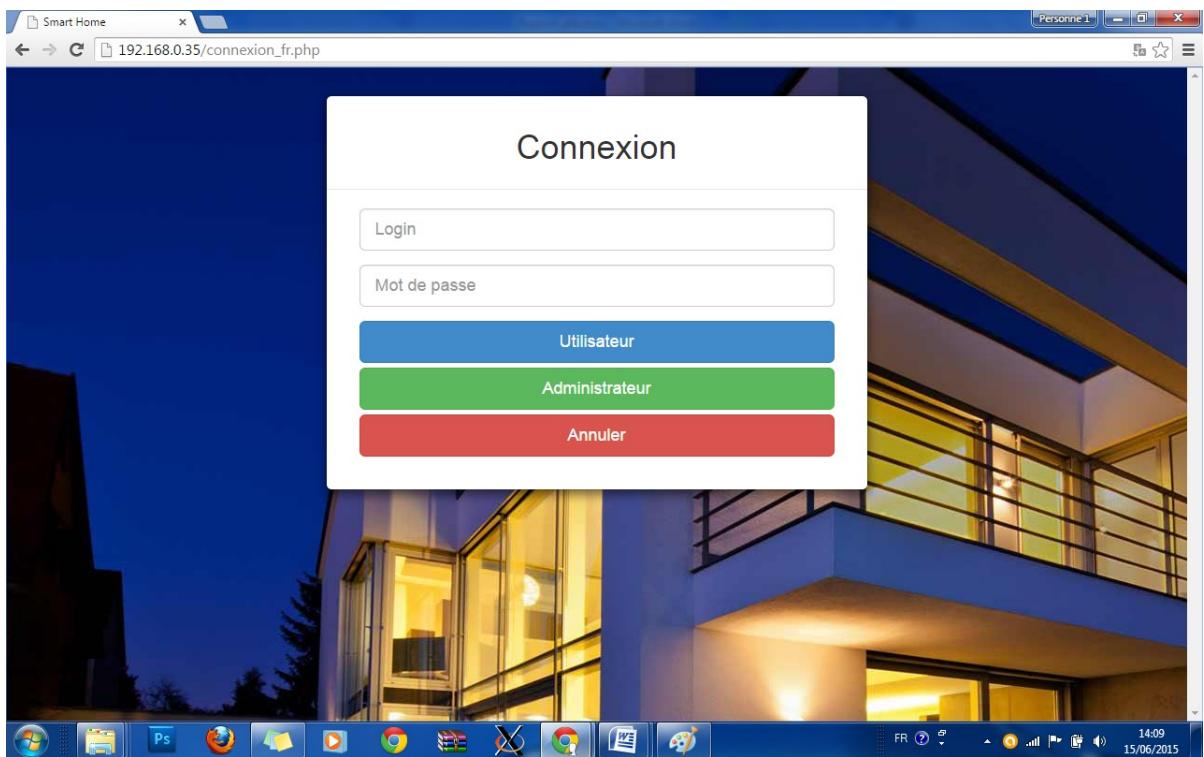


Figure 44 : Interface d'authentification

4.3.2.1 Tâche Administrateur :

Après l'authentification en tant qu'administrateur, ce dernier, comme le montre la figure ci-dessous peut :

- Ajouter utilisateur
- Modifier mot de passe d'utilisateur
- Supprimer utilisateur
- Consulter la liste des utilisateurs

Pour ajouter un utilisateur à cette application on doit saisir le login et mot de passe et choisir le type soit un simple utilisateur soit un administrateur et on clique sur le bouton Ajouter.

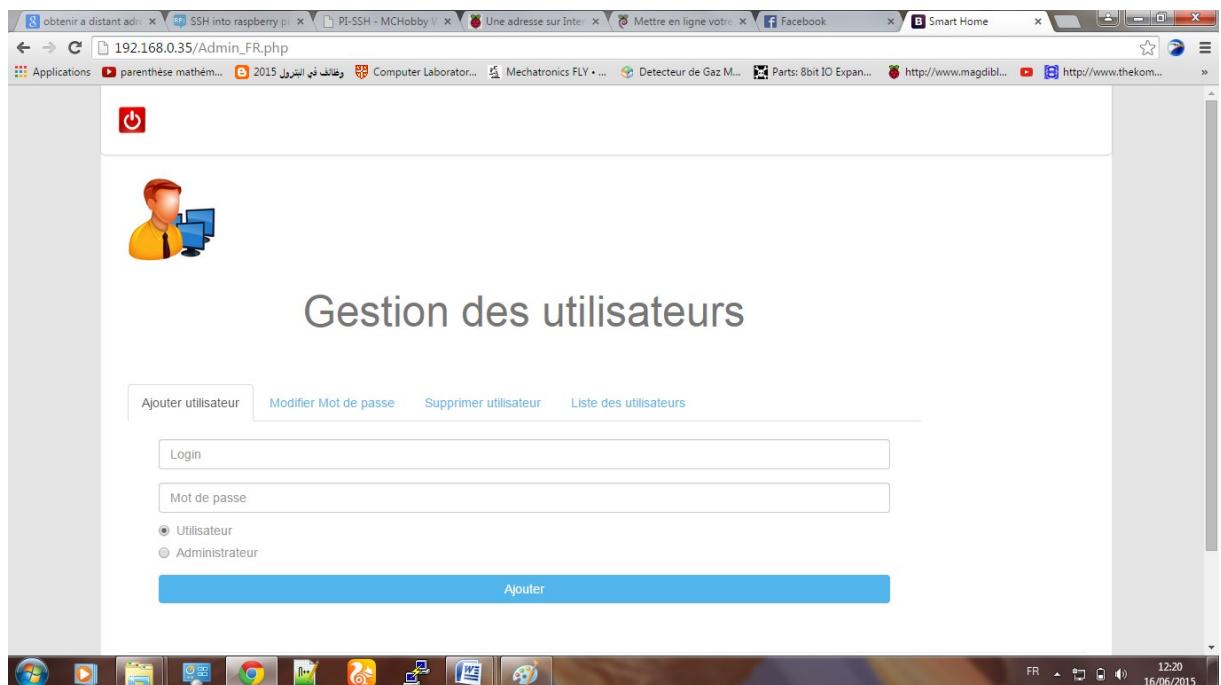


Figure 45 : Interface ajout utilisateur

Pour modifier le mot de passe d'un utilisateur on doit choisir le login de l'utilisateur puis tapez le nouveau mot de passe et on clique sur le bouton Modifier.

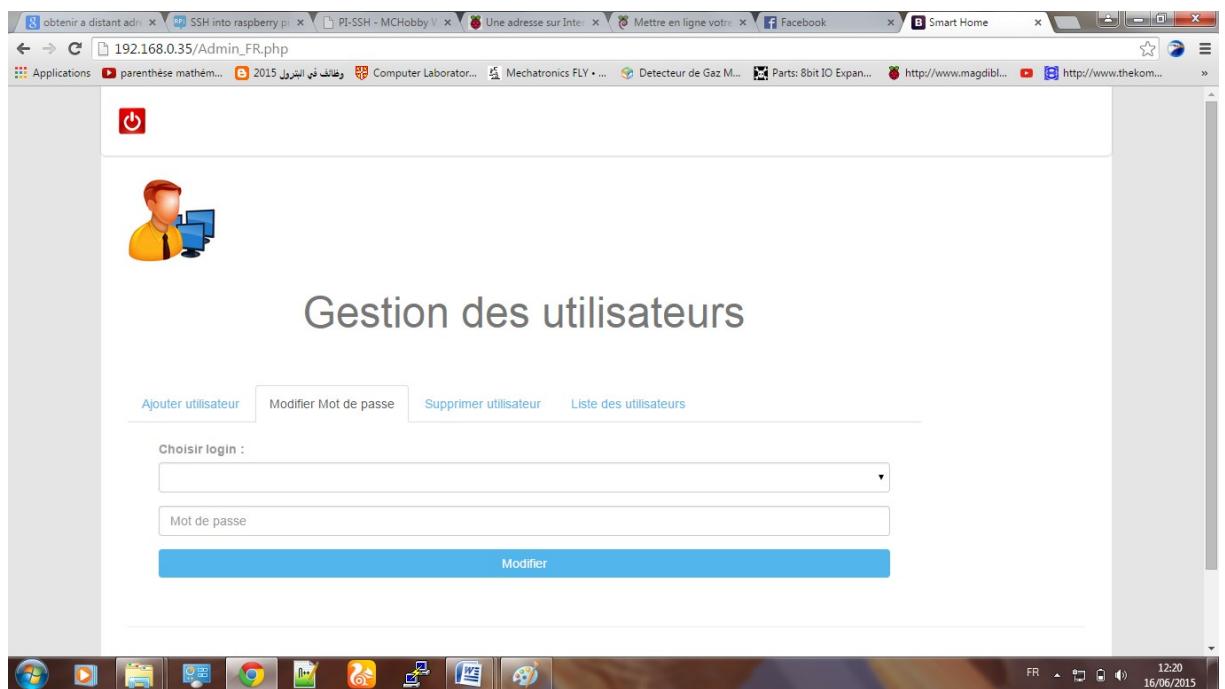


Figure 46: Interface modification mot de passe

Pour supprimer un utilisateur on doit choisir le login d'utilisateur puis on clique sur le bouton supprimer.

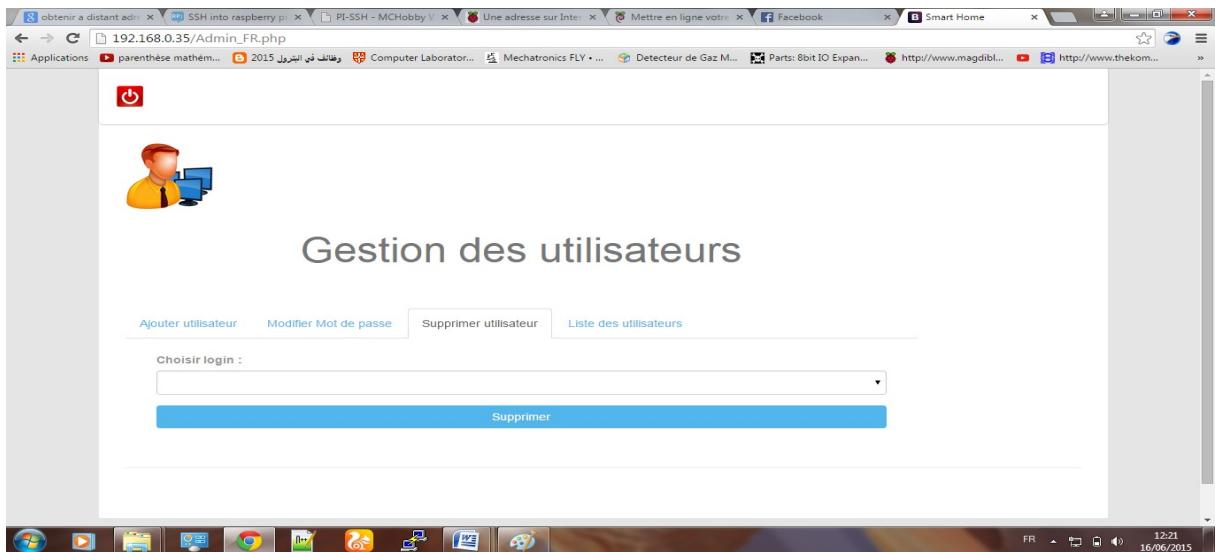


Figure 47 : Interface suppression utilisateur

Pour consulter la liste des utilisateurs de cette application on doit cliquer sur l'onglet Liste des utilisateurs.



Figure 48 : Interface liste des utilisateurs

4.3.2.2 Tâche Utilisateur :

Après l'authentification en tant qu'utilisateur, ce dernier, comme le montre la figure ci-dessous peut :

- Contrôler les volets
- Contrôler la porte
- Contrôler les lampes
- Contrôler la fuite de gaz
- Consulter température
- Consulter humidité

A travers cette interface, l'utilisateur peut monter ou descendre tous les volets.

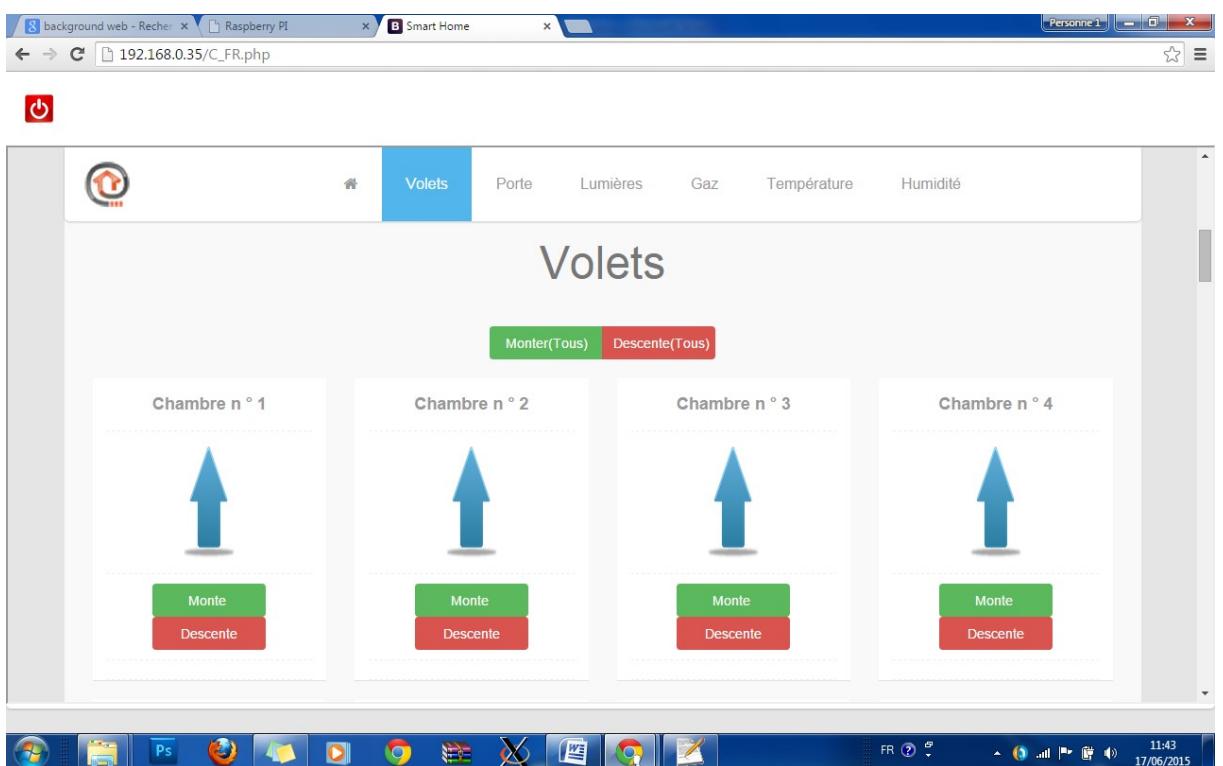


Figure 49 : Interface de contrôle des volets

À travers cette interface, l'utilisateur ouvrira la porte.

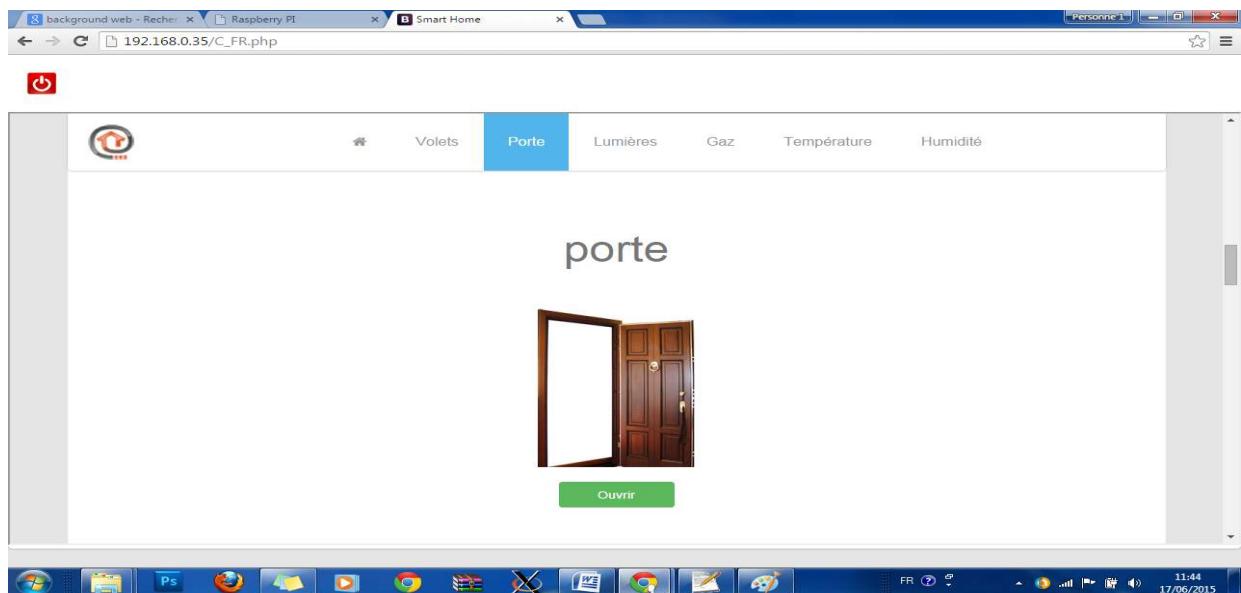


Figure 50 : Interface d'ouverture de porte

À travers cette interface, l'utilisateur peut allumer ou éteindre tous les lampes.

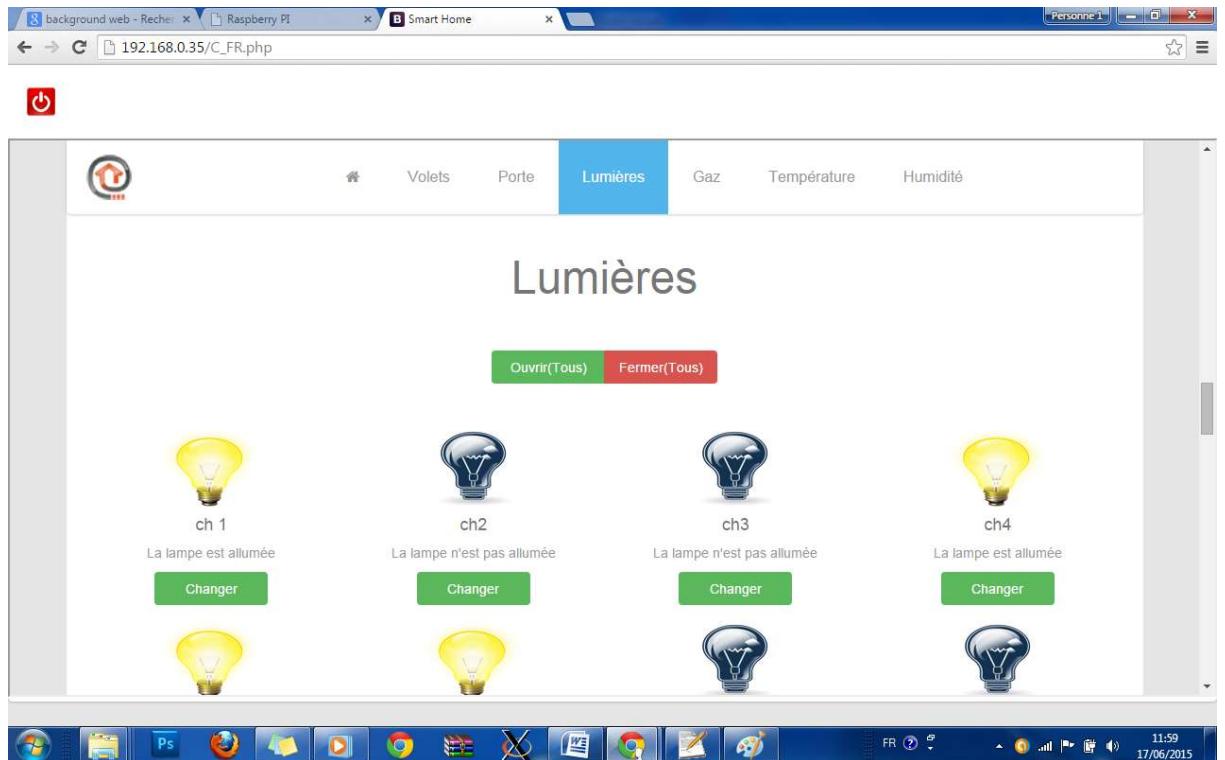


Figure 51 : Interface de contrôle des lampes

À travers cette interface, l'utilisateur peut détecter la fuite de gaz.

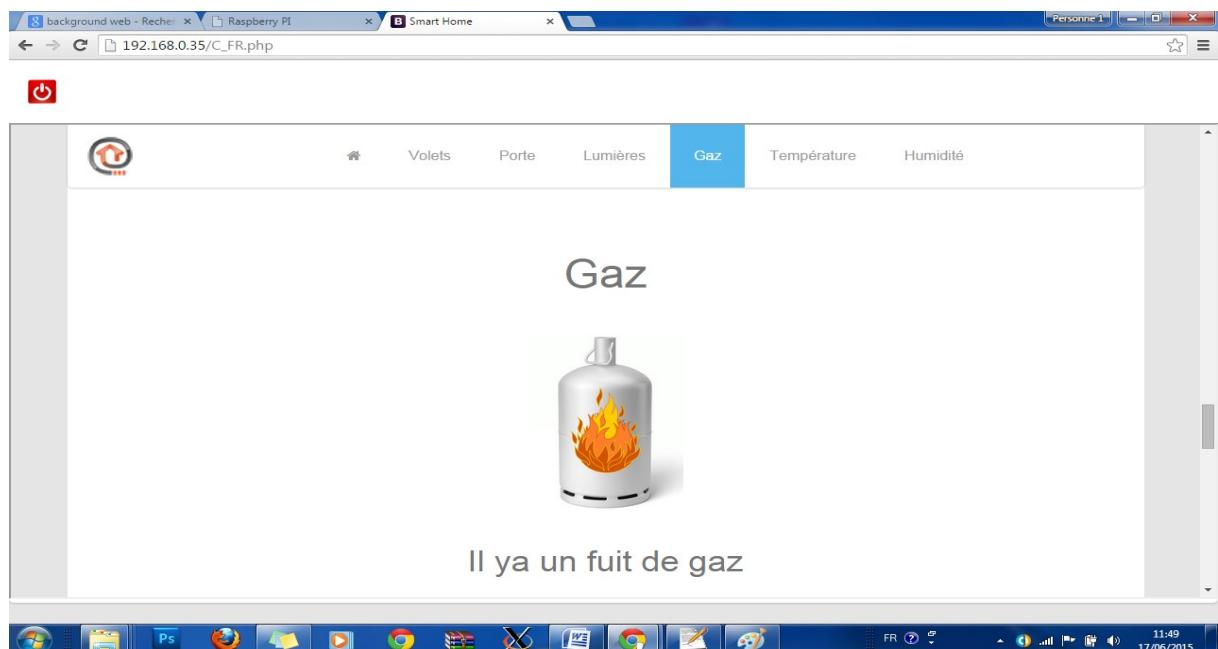


Figure 52 : Interface détection de fuit de gaz

À travers cette interface, l'utilisateur peut consulter la température.

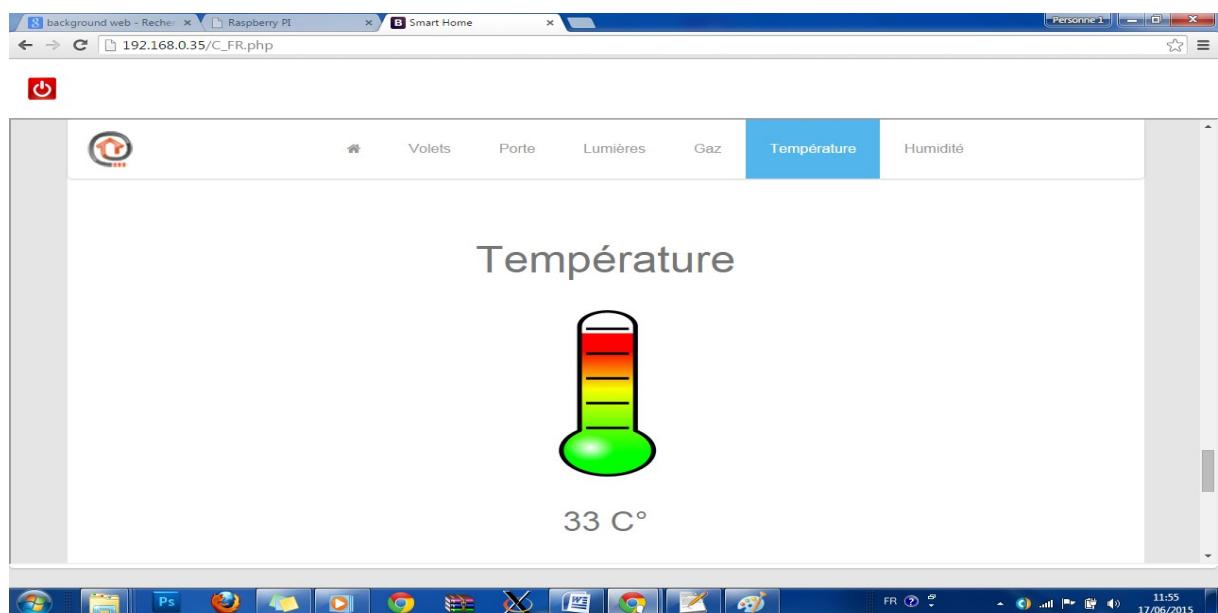


Figure 53 : Interface consultation température

À travers cette interface, l'utilisateur peut consulter l'humidité.

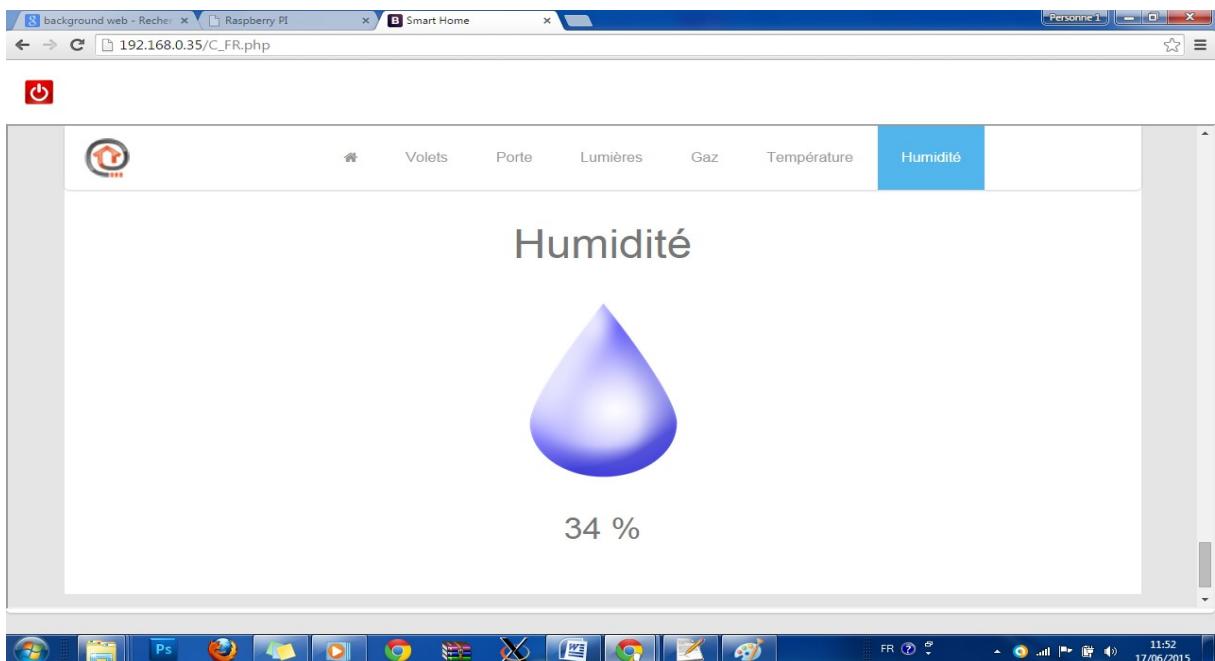


Figure 54 : Interface consultation humidité

4.4 Conclusion :

Dans ce chapitre nous avons décrit l'étude conceptuelle de notre application. Nous avons justifié par la suite le choix des différents outils nécessaires à l'implémentation de notre application. Enfin, nous avons présenté quelques interfaces de réalisation de notre application. À ce stade du travail, nous pouvons affirmer que l'essentiel est présent.

CONCLUSIONS ET PERSPECTIVES

La domotique offre aujourd’hui de véritables bénéfices en termes de sécurité, d’assistance à l’autonomie des personnes âgées ou handicapées, de consommation raisonnée de notre énergie ou encore d’optimisation des systèmes de communication dans le logement. Pour cela, nous avons développé un système de contrôle en domotique.

L’application web que nous avons développée comme projet de fin d’études a pour objectif de commander une multitude d’équipements (lampes, volets, porte) dans la maison qui aide à l’économie de l’énergie, donner une idée sur la température et l’humidité et détecter la fuite de gaz.

À travers les différents chapitres de ce rapport, nous avons essayé de détailler la démarche suivie tout au long de la réalisation de notre application afin de donner une idée plus claire sur l’objectif et les apports de notre travail.

Ce projet a été pour nous d’une part une opportunité d’enrichir nos connaissances dans le monde de l’informatique et l’électronique par la découverte et l’apprentissage de nouvelles technologies surtout dans le domaine du développement embarqué.

Enfin nous voulons signaler que le présent travail reste encore ouvert pour plusieurs autres extensions ou améliorations telles que : la communication sans fil entre la Raspberry Pi et les relais, gérer la consommation d’électricité et une caméra de surveillance accessible à partir de l’application web.

LISTE DES ACRONYMS

- GPIO: General Purpose Input and Output
- Pi: Picademy
- SCL: Slave Clock Line
- SDA: Slave Data Line
- UML: Unified Modeling Language

REFERENCES BIBLIOGRAPHIQUES

- [1] <http://abgattal.info/UML/DCU.pdf>
- [2] <http://fr.wikipedia.org/wiki/StarUML>
- [3] <http://wwwdfr.ensta.fr/Cours/docs/IN204/uml.pdf>
- [4] <http://creersonsiteweb.net/page-apprendre-bootstrap-twitter>
- [5] <http://www.livre-dreamweaver.com/information.html>
- [6] http://fr.wikipedia.org/wiki/Apache_HTTP_Server
- [7] <http://fr.wikipedia.org/wiki/MySQL>
- [8] <http://fr.wikipedia.org/wiki/PhpMyAdmin>
- [9] <https://code.google.com/p/webiopi/>
- [10] [http://fr.wikipedia.org/wiki/Python_\(langage\)](http://fr.wikipedia.org/wiki/Python_(langage))

Résumé

Ce travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du diplôme national d'ingénieur en informatique industrielle à l'institut supérieur des sciences appliquées et de technologie de Sousse (ISSATSo).

L'objectif de ce projet est de développer un système de gestion de la maison. Ce système permet de commander une multitude d'équipements (lampes, volets, portes) dans la maison et surveiller la température et l'humidité et détecter la fuite de gaz.

Via ce système, de n'importe endroit dans votre maison ou dehors, vous pourrez utiliser un navigateur web sur votre ordinateur ou Samrtphone/Tablet pour voir l'état/contrôler à distance les équipements reliés au système dans votre maison.

Pour la réalisation de ce projet, nous avons utilisé une carte Raspberry Pi comme micro-ordinateur et Python, PHP et Javascript comme langages de programmation et UML comme langage de modélisation.

Mots clés: Système Embarqué, Domotique, Contrôle, Raspberry Pi, Application Web.