

**Mise en place d'une plateforme de télécommande des
équipements électrique à distance
« Smart House »**

Réalisé par :

Wisseem HENI

Imen Hmaied

UNIVERSITÉ VIRTUELLE DE TUNIS

MÉMOIRE DU PROJET DE FIN D'ÉTUDES

Pour l'obtention de la

**Licence Appliquée en Sciences et Techniques de l'Information et de
Communication**

Encadré par :

Mr Arafet BOUSSAID

Organisme d'accueil :



Année Universitaire 2010-2011

Remerciements

C'est avec plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à l'égard de tous ceux qui nous ont aidés à la réalisation de notre projet de fin d'étude.

Nous tenons en premier lieu à exprimer notre gratitude envers notre encadreur de l'entreprise « **Tunisie Cartes** », où nous avons effectué notre projet de fin d'études, Monsieur *Boussaid ARAFET*, qui nous a permis de mener à terme ce travail, par ses soutiens, ses précieux conseils et ses bienveillances.

Nous adressons tous nos remerciements à notre professeur et coordinateur de LASTIC3 Monsieur *Chérif ADNANE* et tous nos enseignants pour leurs précieux conseils, leurs soutiens et leurs remarques pertinentes qui nous ont permis de mener au terme ce projet.

N'oublions pas l'administration de l'Université Virtuelle de Tunis qui fut assez visionnaire pour nous offrir tous les moyens de réussite.

Que les membres de jury trouvent nos profondes gratitude pour l'honneur qu'ils nous font en assistant à notre soutenance.

Table des matières

Introduction générale.....	1
Chapitre I : Présentation Générale.....	3
Introduction	4
I. Cadre du projet	4
II. Présentation de la société d'accueil	5
III. Cahier de charge proposé	7
III.1. Contexte.....	7
III.2. Les objectifs à réaliser dans notre projet	8
III.3. Contraintes.....	8
III.4. Structure générale du système	9
IV. Chronogramme	10
Conclusion.....	10
Chapitre II : Etude préalable.....	11
Introduction	12
I. Le système domotique	12
I.1. Qu'est ce que la domotique ?	12
I.2. Pourquoi la domotique ?	13
I.2.1. Le confort	13
I.2.2. La sécurité	13
I.2.3. La gestion d'énergie	14
II. Le service des messages courts (SMS)	14
II.1. Format d'un message court.....	15
II.2. Caractéristiques.....	16
III. Présentation d'un modem GSM	16
III.1. Les commandes Hayes pour modem (Commandes AT)	17
III.1.1. Principes généraux.....	18
III.1.2. Fonctionnement	18
III.1.3. Commandes dédiés au service SMS	19

III.1.4. Interface modem GSM	20
IV. Présentation du Microcontrôleur	20
IV.1. Caractéristiques principales d'un microcontrôleur	21
IV.1.1. Les avantages d'un microcontrôleur	21
IV.1.2. Contenu d'un microcontrôleur	21
IV.2. Le PIC	21
IV.2.1. Définition	22
IV.2.2. Architecture	22
IV.2.3. Identification des PICs	22
IV.2.4. Organisation d'un PIC.....	23
IV.3. Le microcontrôleur « PIC 16F877 »	24
IV.3.1. Désignation.....	24
IV.3.2. Caractéristiques	25
IV.3.3. Mémoires du PIC 16F877	26
IV.3.3.1. Mémoire vive RAM	26
IV.3.3.2. Mémoire morte FLASH	26
IV.3.3.3. Mémoire EEPROM.....	26
IV.3.4. Brochage.....	26
IV.3.4.1. Ports d'entrée/sortie.....	27
IV.3.4.2. Interruption RB0/INT.....	28
IV.3.5. Module de commande	28
IV.3.5.1. Relation entre (PIC, Sorties numériques).....	28
IV.3.5.2. Module de contrôle d'état	29
IV.3.5.3. Module de mesure de tension.....	30
IV.3.5.5. La liaison RS232	31
IV.3.5.5. Présentation du MAX232.....	31
IV.3.2. Le QUARTZ	33
Conclusion.....	33
Chapitre III : Etude conceptuelle	34
Introduction	35
I. Les besoins fonctionnels.....	35
I.1. Authentification.....	35
I.2. Administration.....	35

II.	Conception du système	36
II.1.	Méthodes de conception	36
II.1.1.	Méthodes fonctionnelles	36
II.1.2.	Méthodes orientées objets.....	36
II.2.	Conception de l'application	36
II.2.1.	Choix du modèle de conception.....	37
II.2.2.	Langage UML.....	37
II.2.2.1.	Diagrammes de cas d'utilisation.....	38
II.2.2.2.	Diagrammes de classe.....	39
II.2.2.3.	Diagrammes de séquences	39
II.2.3.	Conception de notre application	39
II.2.3.1.	Représentation des diagrammes de cas d'utilisation	39
II.2.3.2.	Représentation des diagrammes de séquences	42
II.2.3.3.	Représentation du diagramme de navigation.....	45
	Conclusion.....	47
	Chapitre IV : Implémentation de l'application « Smart House ».....	48
	Introduction	49
I.	Environnement logiciel	49
I.1.	Choix du langage Java	49
I.1.1.	Java 2 Micro Edition	49
I.1.1.1.	L'architecture J2ME	50
I.1.1.2.	Les configurations et les profils	50
I.1.2.	Netbeans	52
I.2.	ISIS.....	52
I.3.	ARES.....	52
I.4.	PIC C Compiler.....	53
I.5.	PICPgm	53
II.	Implémentation	54
II.1.	Réalisation de la carte électronique	54
II.1.1.	Schéma fonctionnel de la carte	54
II.1.2.	Simulation de la carte	57
II.1.3.	Routage de la carte.....	58
II.1.4.	Programmeur du PIC	59

II.2. Réalisation de l'application mobile	60
II.2.1. Programmation d'une classe MIDlet	60
II.2.2. Présentation des interfaces de l'application mobile.....	62
II. Tests et Résultats	65
Conclusion Générale	67
Bibliographie & Netographie	68
Liste des Acronymes	70
Annexes	72

Liste des figures

Figure 1.1: Caméra de vidéo de surveillance	6
Figure 1.2: Structure générale du Système « Smart House ».....	9
Figure 2.1: Format d'un message court.....	15
Figure 2.2: Carte PC Card / PCMCIA Card.....	17
Figure 2.3: Modem GSM	17
Figure 2.4: Schéma du fonctionnement des commandes AT.....	18
Figure 2.5: Communication en Usart Terminal	20
Figure 2.6: Le Microcontrôleur PIC16F877-20	24
Figure 2.7: Structure interne du PIC	25
Figure 2.8: Brochage du Microcontrôleur PIC 16F877	27
Figure 2.9: Relation entre (PIC, Sorties Numériques)	28
Figure 2.10: Commande de sortie numérique	28
Figure 2.11: Schéma interne de relais	29
Figure 2.12: Relation entre (PIC, Entrées Numériques)	39
Figure 2.13: Relation entre (PIC, Entrées Analogiques).....	30
Figure 2.14: Diviseur de tension	30
Figure 2.15: Câble RS232	31
Figure 2.16: Structure interne et externe de MAX232.....	32
Figure 2.17: Liaison RS232 entre l'ordinateur et la carte.....	32
Figure 2.18: Schéma de QUARTZ.....	33
Figure 3.1: Diagramme de cas d'utilisation général de notre système domotique	40
Figure 3.2: Diagramme de cas d'utilisation détaillé de notre système domotique	41
Figure 3.3: Diagramme de cas d'utilisation de «Piloter les différents équipements électriques».....	42

Figure 3.4: Diagramme de séquence « Demande d'accès à la plateforme domotique ».....	43
Figure 3.5: Diagramme de séquence « Activer un équipement électrique »	44
Figure 3.6: Diagramme de séquence « Consulter l'état d'un équipement électrique »	45
Figure 3.7: Diagramme de navigation de l'application mobile de notre système.....	46
Figure 4.1 : Architecture de la plateforme J2ME.....	49
Figure 4.2: Schéma fonctionnel de la carte	54
Figure 4.3 : Schéma du bloc.....	55
Figure 4.4: schéma de simulation de la carte électronique en ISIS	56
Figure 4.5 : schéma de routage de la carte sur ARES	57
Figure 4.6: Schéma de la carte en 3D.....	58
Figure 4.7: Programmeur de PIC	59
Figure 4.8: Cycle de vie d'un « MIDlet »	61
Figure 4.9: La page d'accueil de « Smart House »	61
Figure 4.10: Interface d'authentification.....	61
Figure 4.11: Interface de la liste des équipements électriques	62
Figure 4.12: Interface de l'équipement 'Climatiseurs'	62
Figure 4.13: Interface pour activer le climatiseur du Salon	62
Figure 4.14: Interface de l'équipement 'Portes'	63
Figure 4.15: Interface pour ouvrir la porte principale.....	63
Figure 4.16: Interface de l'équipement 'Fenêtres'	63
Figure 4.17: Interface de l'équipement 'Eclairage'	63
Figure 4.18: Interface de l'équipement 'Arrosage'	64
Figure 4.19: Interface de l'équipement 'Système Alarme'	64

Introduction Générale

Avec la diversité des moyens de communication humaine, les technologies de l'information et des télécommunications sont devenues une condition suffisante pour assurer une communication illimitée avec tous les habitants de la planète. Nous pouvons alors nous demander quelle sera la prochaine étape ?

La communication homme-machine ou machine-machine peut être considérée comme étant un nouveau type de dialogue possible. En effet, depuis une dizaine d'années les appareils deviennent intelligents, agissent selon le profil des utilisateurs et sont capables de prendre des décisions de manière autonome.

En pensant à la technologie de télécommunication GSM, il nous vient immédiatement à l'esprit la communication vocale, l'envoi et la réception des SMS et des MMS, l'internet mobile, et tous ce qui fonctionne en utilisant cette technologie moderne introduite dans notre vie quotidienne.

Nous pouvons donc imaginer, en se servant de ces services fournis à travers la technologie GSM, que la maison communicante fera partie de notre futur. Qui ne voudrait pas d'une maison qui prend en charge les tâches domestiques, qui assure à ses occupants plus de confort, de sécurité et de bien-être, qui leur facilite la vie et répond à leurs besoins et désirs avant même qu'ils aient été formulés ? Tout ceci est du domaine de la domotique. Par domotique, il est possible de contrôler et de commander des systèmes à distance en ayant recours au réseau GSM. Des applications peuvent être utilisées dans divers domaines comme le contrôle et la commande à distance des machines, des systèmes d'alarme et de surveillance, de commander des portes et des fenêtres ou d'allumer des lampes...

C'est dans ce cadre que se situe notre projet de fin d'études intitulé « Mise en place d'une plateforme de télécommande des équipements électriques à distance 'Smart House' ». Ce projet a pour objectif de développer un système ou une plateforme d'aide à l'administration des équipements domestiques. Ce système permet le pilotage et la surveillance des différents dispositifs disponibles dans la maison ainsi que la description des services fournis et les actions qu'on peut les invoquer.

Pour se faire, nous avons divisé le travail en quatre parties. Premièrement, nous avons commencé par une présentation générale de notre projet dans laquelle nous allons présenter la société d'accueil, le cadre de projet et les objectifs à réaliser. Ensuite, dans le deuxième chapitre, nous avons fait une étude détaillée sur la domotique, le service des messages courts et le modem GSM. Le troisième chapitre est la représentation des besoins et des exigences qui ont incité au développement de cette application ainsi que la conception que nous avons adopté pour sa réalisation. La réalisation de notre application sera présentée, dans le quatrième chapitre, dans lequel nous présenterons l'environnement de développement et les divers composants implémentés dans l'architecture de notre système.

Nous finirons ce rapport par une conclusion générale récapitulative des différentes phases de notre travail, signalant les côtés bénéfiques du projet et énonçant les perspectives du travail élaboré.

Chapitre I

Présentation Générale

- Cadre du projet
- Présentation de la société d'accueil
- Cahier des charges proposé
- Chronogramme

Présentation Générale

Introduction :

L'objectif de ce chapitre introductif est de mettre notre travail dans son contexte général. Tout d'abord, nous commençons par faire une présentation succincte du sujet en détaillant son cadre et ses fonctionnalités. Ensuite, nous présentons l'organisme d'accueil. Enfin, nous détaillons le cahier des charges proposé et le chronogramme suivi tout au long de la réalisation de ce projet.

I. Cadre du projet :

Notre projet « Smart House » a été réalisé dans le but de répondre à un ensemble de besoins qui spécifient précisément les services demandés et attendus par l'utilisateur. Ces services, qui sont regroupés sous le terme "domotique", concernent principalement le confort (commande à distance d'appareils ou équipements,..), la sécurité (protection contre les intrusions, détection d'incendie, l'économie d'énergie (gestion du climatiseur, d'éclairage..).

En effet, notre système permet de piloter de façon simple et confortable l'ensemble de équipements électriques notamment l'éclairage, les climatiseurs, les ouvrants, l'arrosage et le système alarme.

De plus l'utilisateur a besoin d'un tel système de pilotage, lorsqu'il est engagé dans son travail. Par exemple, lorsqu'il part le matin de son logement, il peut oublier de désactiver le climatiseur ou d'ouvrir les fenêtres. En outre, lorsque l'utilisateur sort en déplacement inattendu, il peut oublier aussi d'activer le système alarme et il ne peut pas faire l'arrosage. Donc, le but de notre application est de surmonter ces problèmes en offrant le service « Smart House » avec lequel il peut par une simple application de son téléphone portable commander les équipements électriques de son domicile à distance.

Dans le cadre du projet de fin d'étude nous nous proposons de réaliser une carte électronique de commande connectée via un port série RS232 avec un modem GSM, ainsi une application téléphonique permettant le contrôle et la commande des équipements électriques à distance.

L'utilisation de cette carte est facile dont l'utilisateur pourra, à l'aide de son téléphone portable, commander, contrôler ou surveiller des équipements électriques, à travers une application installée sur le mobile permettant l'envoi d'un simple SMS codé qui contient une information ou un ordre, ou bien la réception d'un message de compte rendu qui lui informe sur l'état récent du système surveillé.

II. Présentation de la société d'accueil:

Notre projet de fin d'études a été effectué au sein de l'entreprise « Tunisie Cartes » qui se situe à Ariana.

Tunisie cartes, est une société de mise en place des systèmes de sécurité domicile et de géo-localisation distribuant et installant un concept innovant dans le domaine des nouvelles technologies en matière de sécurité. Ainsi elle rassemble de nombreux savoir-faire en matière de sécurité, protection, conception, installation, maintenance et dépannage. Après une étude sur place ou sur plan, Tunisie cartes proposera des solutions adaptées aux besoins des clients.

Elle utilise une large gamme de matériel de sécurité de marques reconnues pour leur fiabilité en détection (vol, intrusion, incendie), automatisme de portails, surveillance vidéo, sonorisation.

Tunisie cartes offre aussi des solutions de gestions de sécurité domicile a distance a travers plusieurs produits innovants qui offrent plusieurs avantages telles que :

❖ VIDEO SURVEILLANCE A DISTANCE:

- Depuis n'importe quel PC dans le monde, visualisez en direct vos établissements par Internet, et consultez les films enregistrés. Aucun besoin de logiciel particulier, de licence coûteuse, ou d'abonnement.

- Avec un simple téléphone 3G, sans PC, gardez le contact visuel avec votre entreprise.

- Recevez un email ou un SMS d'alerte en cas d'intrusion. La vidéo surveillance vous évitera souvent de vous déplacer.

❖ CAMERAS DE VIDEO SURVEILLANCE :

En fonction du site et des besoins des clients, Tunisie carte équipe les locaux de caméras de surveillance, ou utilise celles existantes. Sa large gamme permet de couvrir chaque situation (contre-jour, obscurité, extérieurs, caméras pilotables, anti vandale, Infra rouge...)

A DISTANCE

A distance depuis un PC connecté à Internet ou depuis un téléphone 3G, vous visualisez votre entreprise en direct et consultez les enregistrements de vidéosurveillance.



Figure 1.1: Caméra de vidéo de surveillance

III. Cahier de charge proposé :

La réalisation du système va consister à mettre en œuvre un microcontrôleur PIC, interfacé avec le modem GSM, et pilotant une sortie de commande de puissance tout-ou-rien (type relais électromécanique).

Afin de le concrétiser on va passer par les étapes suivantes :

- Etude sur le modem GSM et le microcontrôleur PIC.
- Conception du système.
- Réalisation de l'application.
- Test du fonctionnement de l'application.

III.1. Contexte :

Avec le développement des équipements électriques du logement, un nombre de plus en plus grand des systèmes électriques permettent de piloter de façon simple et confortable l'ensemble de ces équipements notamment l'éclairage, les climatiseurs, les ouvrants, l'arrosage et le système alarme. On appelle ces systèmes « la domotique ».

C'est dans ce contexte que se déroule notre projet de fin d'étude « Smart House » qui consiste à mettre en place une plateforme de télécommande des équipements électriques à distance en utilisant un modem GSM pour l'envoi et la réception des SMS à partir d'une application installée sur téléphone portable.

En effet, le système doit permettre à l'utilisateur la commande et le contrôle des équipements électriques (Climatiseur, éclairage, système alarme...), la surveillance à distance en récupérant à tout instant l'état de son système via l'envoi d'un SMS.

Notre système électronique est composé d'une carte d'interface et un modem GSM. La liaison entre ces deux équipements se fait à travers une communication série RS232.

III.2. Les objectifs à réaliser dans notre projet :

Notre projet de fin d'études a pour objectifs de :

- Présenter une large idée sur le service SMS et ses différents composants pour passer en deuxième lieu à présenter l'entité intervenante dans la transmission de données qui est le modem GSM avec toutes ses particularités.
- Développer une application mobile créée en J2ME (Java 2 Micro Edition) afin de piloter notre système. Cette application constitue d'une interface affichant les équipements électriques suivants :
 - ✓ Les climatiseurs
 - ✓ Les portes : La porte principale et la porte garage
 - ✓ Les Fenêtres
 - ✓ L'arrosage
 - ✓ Le système alarme
 - ✓ L'éclairage
- Présenter une large idée sur le Microcontrôleur PIC 16F877-20 qui est le cœur de la carte électronique afin de bien expliquer sa construction dans le chapitre de réalisation.

III.3. Contraintes :

Ce projet de fin d'études est à réaliser durant une période de quatre mois. Au bout de laquelle un rapport bien rédigé ainsi qu'une application répondant aux différents besoins qui seront présentés dans la partie « Spécification fonctionnelle et non fonctionnelle » du rapport.

III.4. Structure générale du système :

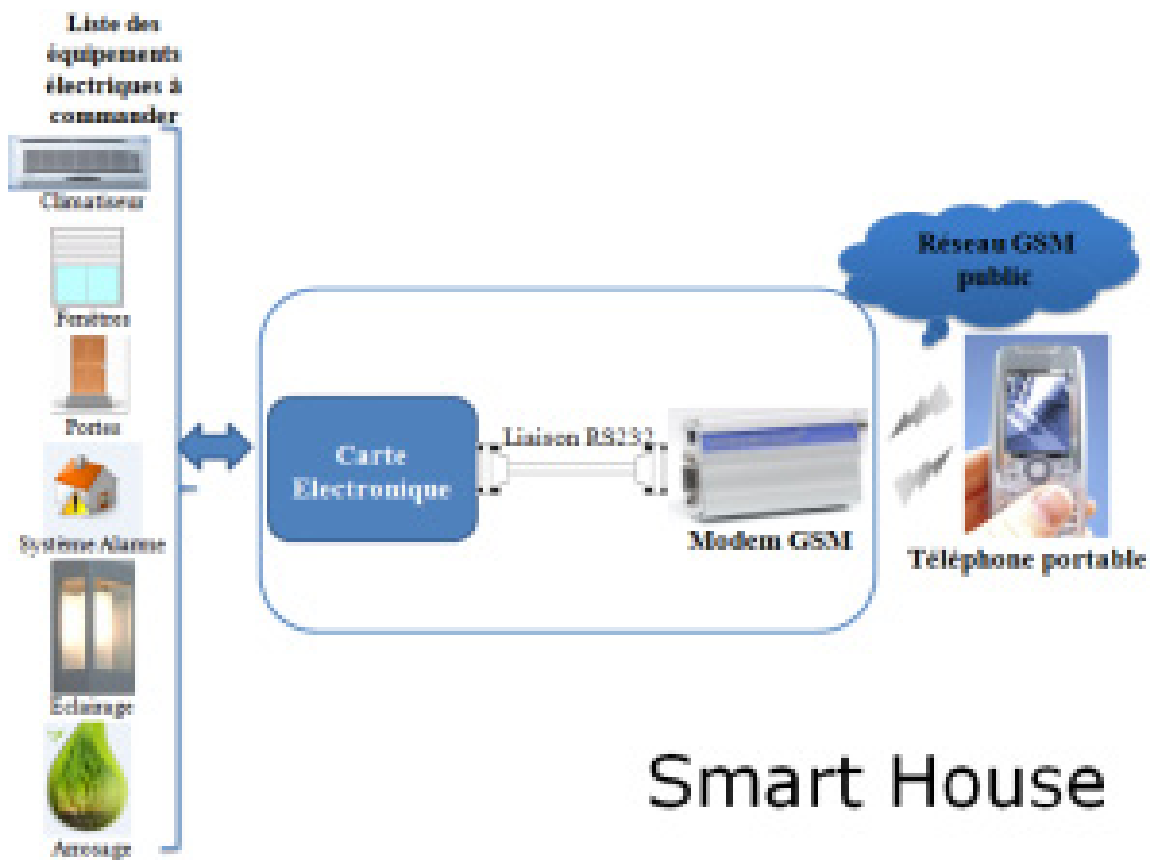


Figure 1.2: Structure générale du Système « Smart House »

Le fonctionnement du système est simple, il suffit de piloter à l'aide d'une application installée sur le téléphone portable, qui envoie un SMS crypté contenant la tâche à effectuer vers le modem GSM pour commander ou contrôler les entrées ou les sorties, en cas d'erreur le modem GSM nous envoie un SMS d'erreur.

Il existe deux méthodes d'envoi et de réception des SMS (Short Message Service). Soit en utilisant le mode texte, soit en utilisant le mode PDU (Protocol Description Unit). Nous avons opté, pour notre application, pour le premier mode du fait qu'il est plus simple à utiliser. Le mode texte se base sur les commandes AT. Donc avec ces commandes, on peut lire, envoyer, effacer et recevoir des SMS en mode texte.

IV. Chronogramme :

Pour la réalisation de notre projet de fin d'études, il nous a été nécessaire de suivre une démarche bien déterminée. Le chronogramme ci-dessous donne une idée approximative sur le déroulement de ce projet.

Tableau 1: Chronogramme suivi pour la rédaction du rapport

Tâche	2011																						
	Février				Mars				Avril				Mai				Juin						
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4			
Cahier des charges			■	■																			
Etude préalable			■	■	■	■																	
Conception					■	■	■	■	■														
Réalisation & Tests									■	■	■	■	■	■									
Rédaction de Rapport					■	■	■	■	■	■	■	■	■	■	■								

Conclusion :

Tout au long de ce chapitre, nous avons pu situer le cadre général de notre projet de fin d'études, à savoir la présentation des ses objectifs, la société d'accueil, le cahier des charges proposé ainsi que le chronogramme suivi pour la réalisation des différentes étapes du projet.

Dans le chapitre suivant, nous donnerons une idée sur « le système domotique », une présentation sur le modem GSM et ses différents composants ainsi que la carte électronique.

Chapitre II

Etude préalable

- La domotique
- Le service des messages courts 'SMS'
- Présentation du modem GSM
- Présentation du Microcontrôleur

Etude préalable

Introduction :

La réalisation de notre projet nécessite une étude approfondie sur certaines notions qui touchent non seulement le cadre général du projet, mais aussi son implémentation. Pour bien assimiler ces différentes notions, nous détaillons, dans un premier lieu, la définition de la domotique ainsi que ses avantages. Ensuite, nous présenterons les méthodes d'envoi et de réception des messages SMS et l'entité intervenante dans la transmission de données qui est le modem GSM avec toutes ses particularités. Nous terminerons ce chapitre avec une présentation détaillée sur le microcontrôleur.

I. Le système domotique :

Avec le développement des équipements électriques du logement, un nombre de plus en plus grand des systèmes électriques permettent de piloter de façon simple et confortable l'ensemble de ces équipements notamment l'éclairage, le climatiseur, les ouvrants, l'arrosage et le système alarme. On appelle ces systèmes « la domotique ».

Les progrès technologiques, notamment de l'informatique, de la télécommunication et de l'électronique ont permis le développement de systèmes de transmission, des commandes à distance et favorisé l'éclosion d'une offre abondante de nouveaux services pour les occupants des logements.

I.1. Qu'est ce que la domotique ?

La domotique rassemble les technologies de l'informatique, des télécommunications l'électronique et de utilisées dans les domiciles. Elle vise à assurer des fonctions de sécurité, de confort, de gestion d'énergie et de communications. Les appareils de la maison sont intégrés au sein des systèmes qui doivent communiquer entre eux afin de gérer des automatismes [1].

La domotique s'agit d'un système électrique qui permet de communiquer avec des télécommandes ou des boutons poussoirs afin de rendre le contrôle de la maison plus facile. Ils permettent de commander d'un simple geste une ou plusieurs actions (Exemples : baisse du chauffage, réglage de l'éclairage, descente des volets motorisés...).

Une application domotique nécessite, pour son fonctionnement, de pouvoir capter une information et d'actionner par la suite une commande. Pour cela, il faut pouvoir faire transférer l'information entre différents dispositifs.

Le système permet de programmer des automatismes sources de confort de sécurité et d'économie d'énergie : l'ouverture et la fermeture automatique des volets, l'arrosage automatique en fonction du climat, la mise en sécurité des stores en cas de vent violent, la gestion de l'éclairage en fonction de la présence et de la luminosité, etc.

1.2. Pourquoi la domotique ?

La domotique est l'automatisation des fonctions techniques à des fins d'économie d'énergie, d'augmentation du confort, de la sécurité et de la mobilité pour les constructions nouvelles comme pour les rénovations [2].

I.2.1. Le confort :

L'accroissement du niveau de confort des habitations a été le premier objectif de la domotique. Les fonctions de commande à distance simples et qui agissent sur différents types d'appareils sont maintenant banalisées.

En effet, il est possible d'activer à distance des fonctions qui a pour but de recréer une ambiance ou un état prédéfinis dans la maison. Il est donc facile d'imaginer un nombre illimité des fonctions qui pourraient faciliter le confort quotidien dans la maison (par exemple la cafetière s'allume et les volets s'ouvrent à 7h tous les matins).

I.2.2. La sécurité :

En cas de menace pour la sécurité de la maison, tout composant domotique est capable d'émettre un message sur l'installation qui sera repris et traité par un module spécialisé pour la surveillance. Ce module peut alors déclencher n'importe quel composant présent dans l'installation afin de simuler une présence (lumière qui s'allume, télévision, musique) ou bien renforcer la sécurité (verrouillage de toutes les serrures, déclenchement des alarmes). Ces

actions peuvent se faire selon un choix particulier, selon une durée ou un nombre de détections ou bien directement par téléphone ou par un ordinateur à distance.

Ces actions peuvent être aussi diverses que [3]:

- Enclenchement de certaines lumières intérieurs pour simuler une présence (des centaines de combinaisons aléatoires sont possibles)
- Enclenchement d'un signal acoustique destiné à décourager les « visiteurs »
- Création et envois d'un SMS sur votre portable
- Composition d'un numéro de téléphone afin d'activer un service de sécurité

I.2.3. La gestion d'énergie :

Un système domotique peut diminuer de 40% à 70% la facture d'énergie du domicile sans toucher au confort de vie. La domotique va influencer la dépense d'énergie sur deux éléments :

- ✓ La suppression de la consommation électrique inutile : L'économie d'énergie, c'est avant tout supprimer une énergie dépensée alors qu'elle n'est pas utilisée. En effet, le gaspillage d'énergie peut être limité avec des produits domotiques afin de ne pas avoir de lampe oubliée à la cave pendant plusieurs jours, un éclairage surdimensionné ou une lampe allumée en plein jour.
- ✓ La distribution de chaleur dans les pièces : Une régulation « intelligente » de l'énergie dans les pièces est essentielle, non seulement pour les factures d'énergie, mais aussi pour le confort de vie. Le réglage des températures est simple et visuel, chaque pièce peut bénéficier d'un réglage qui lui est propre (absence/présence, jour/nuit) et une commande à distance par téléphone est possible afin de mettre la maison sur « confort » ou bien « économie » lors de l'absence des habitants.

II. Le service des messages courts (SMS) :

Le service des messages courts (SMS) consiste à émettre et à recevoir des messages alphanumériques avec uniquement un terminal mobile (portable GSM).

Les messages courts SMS sont spécifiés par l'ETSI. Il y a deux façons de transmettre un message SMS, soit en mode PDU (Protocol Description Unit) soit en mode TEXT. Le mode TEXT, non supporté par la plupart des téléphones portables et les modules GSM, permet d'envoyer des SMS sans codage préalable à l'étape de numérisation. Par contre, en mode

PDU, qui est le mode de base, le message à envoyer est codifié en une suite de caractères hexadécimaux avant de le transformer en un train binaire dans le but d'augmenter le nombre de caractères qu'un SMS peut comporter. Différents types de codage sont utilisés pour passer du mode PDU en mode TEXT. Le plus répandu est celui nommé « 7-bits GSM alphabet » qui offre le maximum de caractères à envoyer (160 caractères). Pour ce type de codages chaque caractère est codé sur sept bits.

II.1. Format d'un message court :

Un message court est composé de 160 caractères codés au maximum par l'ASCII 7 bits, issus de la couche applicative SM-AL (Short Message Application Layer). Son format est défini par la recommandation ETSI 3.40 de GSM et un en-tête doit être ajouté pour préciser l'adresse de destination du message court. La couche SM-TL (Short Message Application Layer) pourra alors le transporter. Le format d'un message court est présenté par la figure suivante :

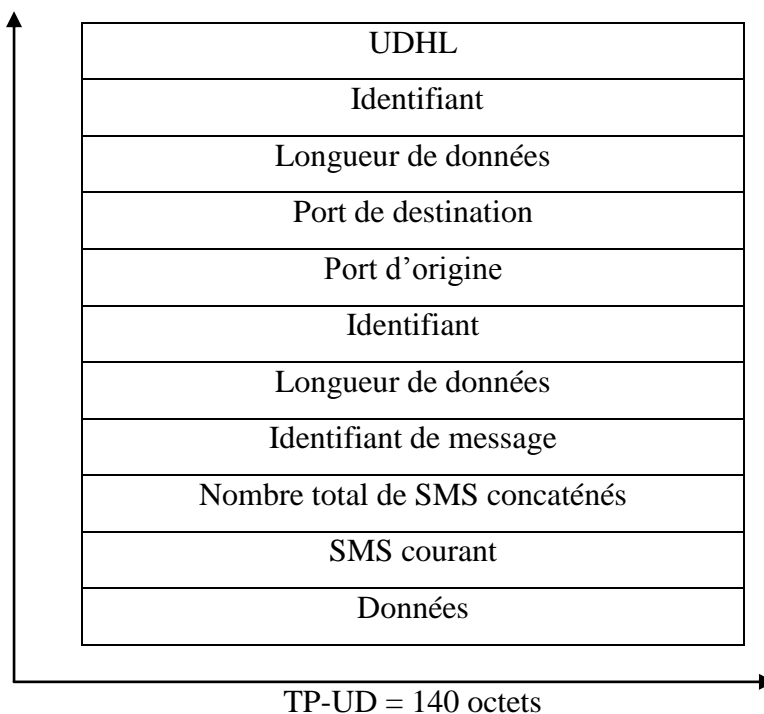


Figure 2.1: Format d'un message court

I.2. Caractéristiques :

Le service SMS connaît, de nos jours, un succès considérable. Ce succès démontre l'intérêt particulier que les consommateurs lui prêtent et ce en adoptant spontanément ce média. Ceux-ci en ont fait un véritable phénomène de société disposant par exemple de son propre vocabulaire. Ce sont aujourd'hui plus de 15 milliards des SMS qui sont échangés chaque mois dans le monde. Parmi les caractéristiques du SMS, on peut citer :

- **Au point de vue technique :** ce service utilise le canal de signalisation SS7 (Signalisation Sémaphore n°7) qui permet de véhiculer les données de contrôle et qui est parallèle au canal des données sur lequel circulent les communications vocales. Ce canal de signalisation offre des capacités suffisantes pour permettre l'acheminement de mini messages texte pour les abonnés. Toutefois, cette capacité excédentaire est limitée à 128 octets.
- **Au point de vue économique :** eux-mêmes surpris par le succès d'un produit dont ils n'avaient pas dans un premier temps spécifiquement assuré la promotion. Le SMS est aujourd'hui une source importante de revenus pour les opérateurs.

III. Présentation d'un modem GSM :

Un modem GSM est un modem sans fil qui fonctionne avec un réseau GSM. Il se comporte comme n'importe quel modem. La principale différence entre les deux modems étant que le modem envoie et reçoit les données par l'intermédiaire d'une ligne téléphonique fixe alors que le modem GSM envoie et reçoit les données par l'intermédiaire des ondes radio.

Un modem GSM peut être un appareil externe ou une carte PC Card / PCMCIA Card. En effet, un modem GSM externe est généralement connecté à un ordinateur via un câble série ou un câble USB. Il peut se présenter aussi sous la forme d'une carte PC Card / PCMCIA Card conçue pour être utilisée avec un ordinateur portable.

Comme un téléphone mobile GSM, le modem GSM nécessite une carte SIM d'un opérateur GSM afin d'assurer son fonctionnement. Comme il a été mentionné dans les sections précédentes, les ordinateurs utilisent des commandes AT pour contrôler les modems.

Les deux modems GSM et Dial-up ont en commun un ensemble de commandes AT standard. On peut donc utiliser un modem GSM comme un modem dial-up.

Outre les commandes AT standard, les modems GSM soutiennent une longue série de commandes AT lesquelles sont définies dans les normes GSM. Avec l'étendue commandes AT, on peut réaliser des fonctions telles que :

- La lecture, l'écriture et la suppression des SMS.
- L'envoi des SMS.
- Le suivi de la puissance du signal.
- Le contrôle de l'état et le niveau de la charge de la batterie.
- La lecture, l'écriture et la recherche des entrées du répertoire.

Le nombre des messages SMS peuvent être traités par un modem GSM à la minute est très faible (environ six à dix SMS par minute).



Figure 2.2: carte PC Card / PCMCIA Card



Figure 2.3: Modem GSM

III.1. Les commandes Hayes pour modem (Commandes AT) :

Durant plusieurs années, les modems Hayes ont été un standard. Mais comme le champ des fabricants de modems a augmenté, beaucoup ont adhéré, plus ou moins, au standard Hayes. Ce qui suit est une liste partielle de l'ensemble de commandes Hayes qui sont appelées les commandes « AT ». Pour la manipulation d'un modem GSM, nous avons besoin d'utiliser un ensemble de commandes Hayes. [4]

III.1.1. Principes généraux :

Ces commandes commencent toujours par la séquence AT à l'exception de la commande de répétition de la dernière commande (A/). Le modem comprend indifféremment les commandes en majuscules ou en minuscules. Chaque commande doit se terminer par un caractère de fin de ligne.

La commande de répétition (A/) ne nécessite pas de caractère de fin. Le caractère back space (08H) permet d'annuler, lors de l'envoi d'une commande le dernier caractère envoyé au modem.

La longueur maximale d'une chaîne de commande est de 128 caractères y compris le AT et le retour chariot. S'il y a plus de 128 caractères, le modem renvoie un message d'erreur et n'exécute pas la commande. Si le modem détecte une erreur dans la chaîne, il interprète la chaîne jusqu'à la détection d'erreur, il envoie un message d'erreur sans traiter les commandes pouvant se trouver derrière la commande ayant occasionnée l'erreur (voir Annexe A).

III.1.2. Fonctionnement :

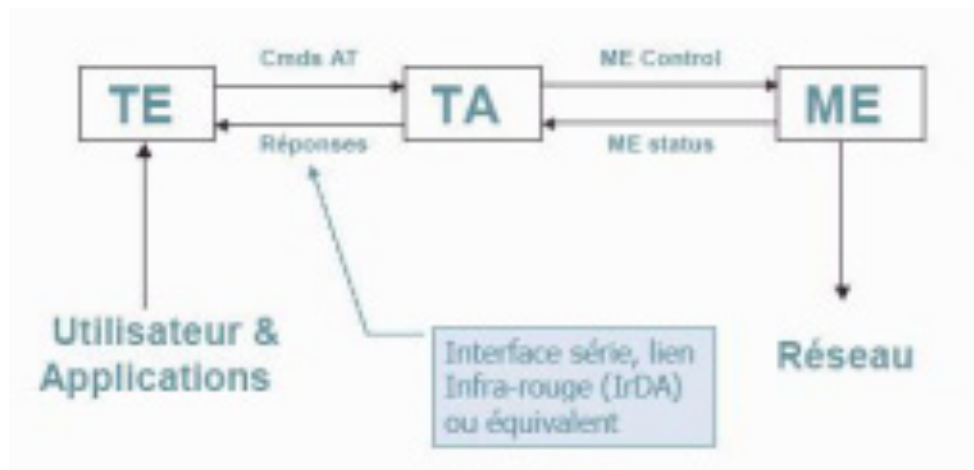


Figure 2.4: Schéma du fonctionnement des commandes AT [5]

ME (Mobile Equipment) : téléphone portable

TE (Terminal Equipment) : peut être un ordinateur ou un microcontrôleur

TA (Terminal Adaptateur) : assure la liaison entre le ME et le TE

TA et ME forment une seule entité, par exemple un téléphone portable standard ou un terminal GSM contient dans son boîtier à la fois le TA et le ME.

Le TE forme une entité à part, par exemple il peut s'agir d'un ordinateur qui dispose d'un port série ou un circuit électronique basé sur un microcontrôleur qui implante un port série.

III.1.3. Commandes dédiées au service SMS :

Pour une utilisation standard et une gestion de messagerie, les commandes AT du tableau suivant peuvent être utilisées :

Tableau 2: Commandes AT dédiées au service SMS

AT+CSMS	Sélection du service de messagerie
AT+CPMS	Sélection de ta zone mémoire pour le stockage des SMS
AT+CMGF	Sélection du format du SMS (PDU ou TEXT)
AT+CSCA	Définition de l'adresse du centre de messagerie
AT+CSDH	Affiche en mode TEXT le paramétrage des SMS
AT+CSAS	Sauvegarde du paramétrage
AT+CRES	Restauration du paramétrage par défaut
AT+CNMI	Indication concernant un nouveau SMS
AT+CMGL	Liste les SMS stockés en mémoire
AT+CMGR	Lecture d'un SMS
AT+CMGS	Envoie un SMS
AT+CMSS	Envoie d'un SMS stocké en mémoire
AT+CMGW	Écriture d'un SMS.
AT+CMGD	Efface un SMS

Le caractère « ? » est souvent remplacé par un chiffre ou par un nombre selon l'utilisation.

III.2. Interfaçage modem GSM :

Pour communiquer avec le GSM, on utilise le port COM (1 ou 2) du PC. Dès que ce dernier est ouvert, on peut interroger facilement le modem du GSM en utilisant les commandes AT et connaître presque tous les informations et de transférer des données.

On interroge le modem GSM par la commande « AT+CGMM » pour l'identification du modèle, la commande « AT+CGMI » pour l'identification du constructeur et la commande « AT+CGMR » pour connaître le numéro de série du téléphone.

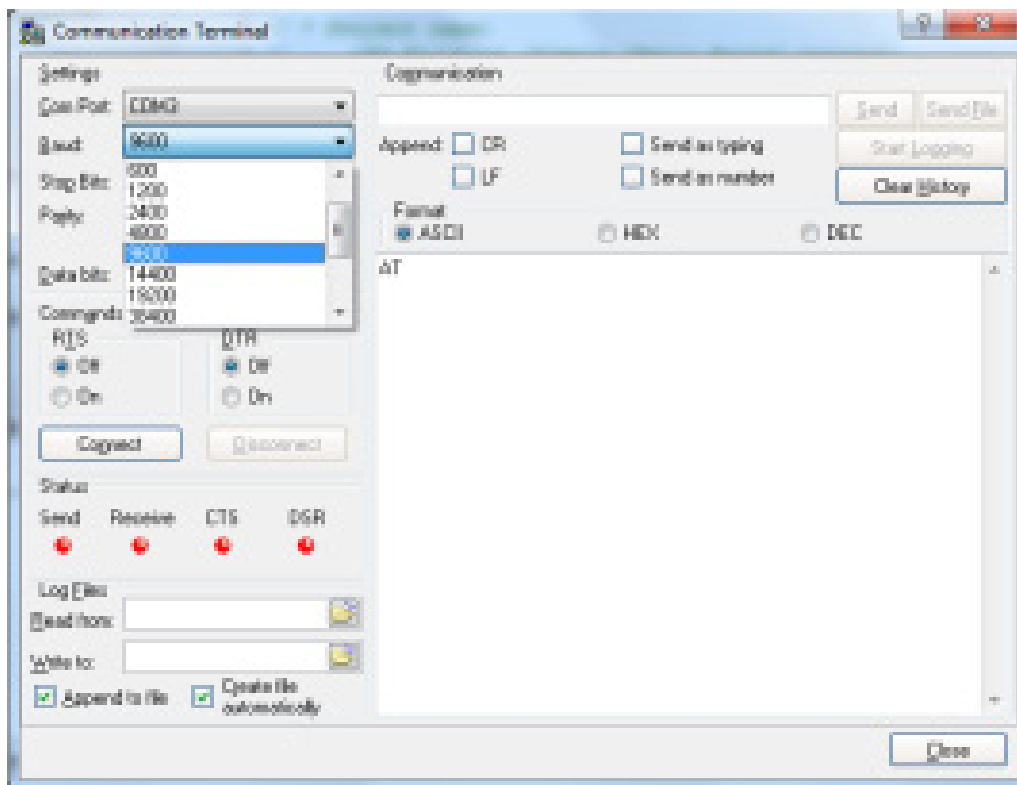


Figure 2.5: communication en Usart Terminal

IV. Présentation du microcontrôleur :

Un microcontrôleur est un circuit central de systèmes électroniques intégrant un microprocesseur et des fonctions supplémentaires (mémoire, mémoire non-volatile, Interfaces analogiques). [6]

IV.1. Caractéristiques principales d'un microcontrôleur :

Le microcontrôleur possède plusieurs caractéristiques notamment :

- De nombreux périphériques d'E/S.
- Une mémoire de programme.
- Une mémoire vive (en général de type SRAM).
- Éventuellement une mémoire EEPROM destinée à la sauvegarde par programme des données à la coupure de l'alimentation.
- Un processeur 8 ou 16 bits.
- Faible consommation électrique.

IV.1.1. Les avantages d'un microcontrôleur :

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts est bien réelle. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants des circuits intégrés en ce domaine depuis quelques années. En effet, le microcontrôleur est moins cher que les autres composants qu'il remplace. D'une autre part, le microcontrôleur diminue les coûts de main d'œuvre.

IV.1.2. Contenu d'un microcontrôleur :

Un circuit microcontrôleur doit contenir dans un seul boîtier tous les éléments de base qu'on verra par la suite. En effet, pour l'analyse des divers systèmes réalisés avant l'avènement des microcontrôleurs, les fabricants des circuits intégrés ont affiné un peu la définition de ce qu'il fallait intégrer pour arriver à un schéma type analogue.

IV.2. Le PIC :

Les microcontrôleurs PIC (ou PICmicro dans la terminologie du fabricant) forment une famille de microcontrôleurs de la société Microchip. Ces microcontrôleurs sont dérivés du PIC1650 développé à l'origine par la division microélectronique de General Instruments.

Le nom PIC n'est pas officiellement un acronyme, bien que la traduction en « Peripheral Interface Controller » (contrôleur d'interface périphérique) soit généralement admise. Cependant, à l'époque du développement du PIC1650 par General Instruments, PIC était un acronyme de « Programmable Intelligent Computer » ou « Programmable Integrated Circuit ».

IV.2.1. Définition:

Les PIC intègrent une mémoire de programme, une mémoire de données, des ports d'entrée-sortie (numériques, analogiques, MLI, UART, bus I²C, etc.), et même une horloge, bien que des bases de temps externes puissent être employées. Certains modèles disposent de port et unités de traitement de l'USB.

Les Pics sont très performants de par leur vitesse d'exécution, et peu couteux, les microcontrôleurs PIC se sont imposés avec succès depuis une dizaine d'années et se retrouvent aujourd'hui dans de nombreux appareils très divers (programmateurs domestiques ou d'appareils électroménager; Télécommande; thermostats électroniques...).

IV.2.2. Architecture :

Les PIC se conforment à l'architecture Harvard : ils possèdent une mémoire de programme et une mémoire de données séparées. La plupart des instructions occupent un mot de la mémoire de programme. La taille de ces mots dépend du modèle de PIC, tandis que la mémoire de données est organisée en octets.

Les PIC sont des processeurs dits RISC, c'est-à-dire processeur à jeu d'instruction réduit. Plus le nombre d'instructions est réduit, plus le décodage est rapide, et plus le composant fonctionne vite. Cependant, il faut plus d'instructions pour réaliser une opération complexe.

Un cycle d'instruction d'un PIC dure 4 temps d'horloge. La plupart des instructions durent un cycle, sauf les sauts qui durent deux cycles. On atteint donc des vitesses élevées.

Avec un quartz de 4 MHz (ou l'horloge interne), on obtient donc 1 000 000 de cycles/seconde, or, comme le PIC exécute pratiquement 1 instruction par cycle, hormis les sauts, cela donne une puissance de l'ordre de 1 MIPS (1 million d'instructions par seconde).

Les PIC peuvent être cadencés à 20 MHz (série PIC16), 40 MHz (série PIC18), et 48 MHz (exemple : PIC18F2550 — PIC avec USB) et 64 MHz (exemple : PIC18F25K20 — PIC en 3,3 V).

IV.2.3. Identification des Pics :

Nous présenterons les méthodes utilisées pour identifier les références des circuits pic, ce qui nous renseigne déjà sur un certain nombre de paramètres importants.

Une référence de microcontrôleur Micro chip est toujours de cette forme: xx (L) XXyy –zz Expliqué dans le tableau suivant :

Tableau 3: Identification du PIC

XX	FAMILLE DE COMPOSANTE (12-14-16-17-18)
L	Tolérance plus importante de la plage de tension
XX : Type de mémoire de programme	C - EPROM or EEPROM CR - PROM F - FLASH
yy	Identification : type (84, 877...)
Zz	Vitesse maximum du quartz

IV.2.4. Organisation d'un PIC :

Un microcontrôleur se présente sous la forme d'un circuit intégré réunissant tous les éléments d'une structure à base de microprocesseur.

Voici généralement ce que l'on trouve à l'intérieur d'un tel composant:

- Un microprocesseur (C.P.U.).
- Une mémoire de donnée (RAM et EEPROM).
- Une mémoire programme (ROM, OTPROM, UVPROM ou EEPROM).
- Des interfaces parallèles pour la connexion des entrées / sorties.
- Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités.
- Des timers pour générer ou mesurer des signaux avec une grande précision temporelle.
- Des convertisseurs analogique / numérique pour le traitement des signaux.

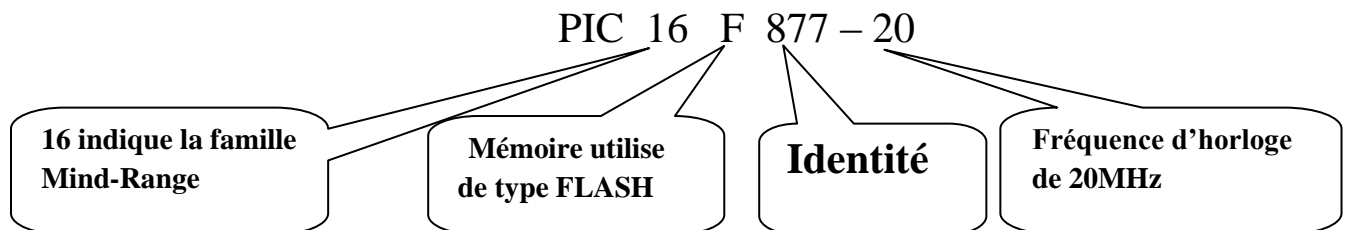
IV.3. Le microcontrôleur « PIC 16F877 »:

Nous allons maintenant s'intéresser à la structure interne du PIC 16F877, avec lequel nous avons réalisé notre projet de fin d'études. Ce 16F877 est un microcontrôleur de MICROCHIP, fait partie intégrante de la famille des Mid-Range (16) dont la mémoire programme est de type flash (F). (Voir figure 2.6)



Figure 2.6: Le Microcontrôleur PIC16F877-20

IV.3.1. Désignation :



Le numéro 16 signifie qu'il fait partie de la famille "MID-RANGE". C'est un microcontrôleur de la famille 8 bits. Cela veut dire que l'ALU (Arithmetic and Logic Unit ou Unit Arithmétique et Logique en français) traite naturellement des mots de 8 bits maximum.

La lettre F indique que la mémoire programme de ce PIC est de type "Flash". Chaque ligne de mémoire est un mot de 14 bits.

Les trois derniers chiffres permettent d'identifier précisément le PIC, ici c'est un PIC de type 877.

La référence 16F877 peut avoir un suffixe du type "-XX" dans lequel XX représente la fréquence d'horloge maximale que le PIC peut recevoir. [7]

IV.3.2. Caractéristiques :

Les principales caractéristiques de ce microcontrôleur sont [8]:

- Fréquence fonctionnement élevée, jusqu'à 20 MHz
- Une mémoire morte de type FLASH de 8 kmots (1mot = 14 bits), elle est réinscriptible à volonté.368 octets de RAM
- Une mémoire EEPROM pour sauver des paramètres de 256 octets.
- Une mémoire vive de 368 octets.
- 32 Entrées et Sorties bidirectionnelles.
- 8 convertisseurs A/N 10 bits
- Liaison SSP
- Bus I2C
- 3 Temporisateurs : TIMER0 (8 bits avec pré diviseur), TIMER1 (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ) et TIMER2 (8bits avec pré diviseur et post diviseur).
- Une tension d'alimentation entre 2 et 5,5 V.

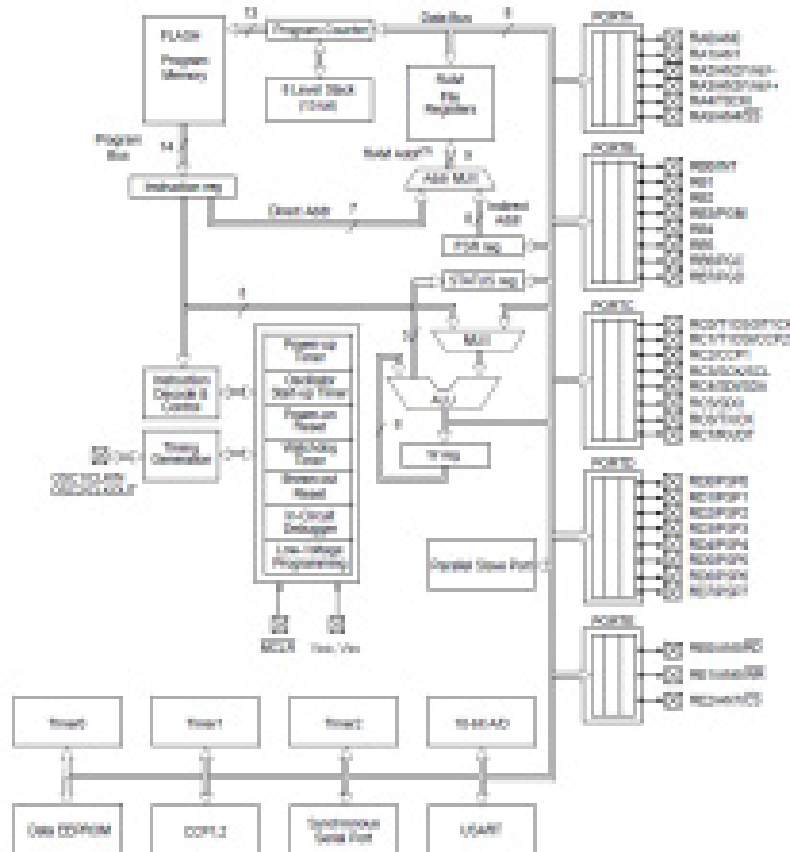


Figure 2.7: Structure interne du PIC

IV.3.3. Mémoires du PIC 16F877 :

Le PIC 16F877 dispose de trois types de mémoires :

IV.3.3.1. Mémoire vive RAM :

C'est de la mémoire d'accès rapide, mais labile (c'est-à-dire qu'elle s'efface lorsqu'elle n'est plus sous tension); cette mémoire contient les registres de configuration du PIC ainsi que les différents registres de données. Elle comporte également les variables utilisées par le programme.

IV.3.3.2. Mémoire morte FLASH :

C'est la mémoire programme proprement dite. Chaque case mémoire unitaire fait 14 bits. La mémoire FLASH est un type de mémoire stable, réinscriptible à volonté. Cette mémoire qui a fait le succès de microprocesseur PIC. Dans le cas du 16F877, cette mémoire FLASH fait 8 Kmots. Lorsque l'on programme en assembleur, on écrit le programme directement dans cette mémoire.

III.3.3.3. Mémoire EEPROM :

Elle est de 256 octets, électriquement effaçable, réinscriptible et stable. Ce type de mémoire est d'accès plus lent, elle est utilisée pour sauver des paramètres. L'adresse relative de l'accès EEPROM est comprise entre 0000 et 00ff, ce qui nous permet d'utiliser qu'un registre de huit bits pour définir cette adresse.

IV.3.4. Brochage :

Le boîtier du PIC 16F877 décrit par la figure 2.8 comprend 40 pins : 33 pins d'entrées/sorties, 4 pins pour l'alimentation, 2 pins pour l'oscillateur et un pin pour le reset (MCLR).

La broche MCLR sert à initialiser le microcontrôleur en cas de la mise sous tension, de remise à zéro externe, de chien de garde et en cas de la baisse de tension d'alimentation. Les broches VDD (broches 11 et 32) et VSS (broches 12 et 31) servent à alimenter le PIC.

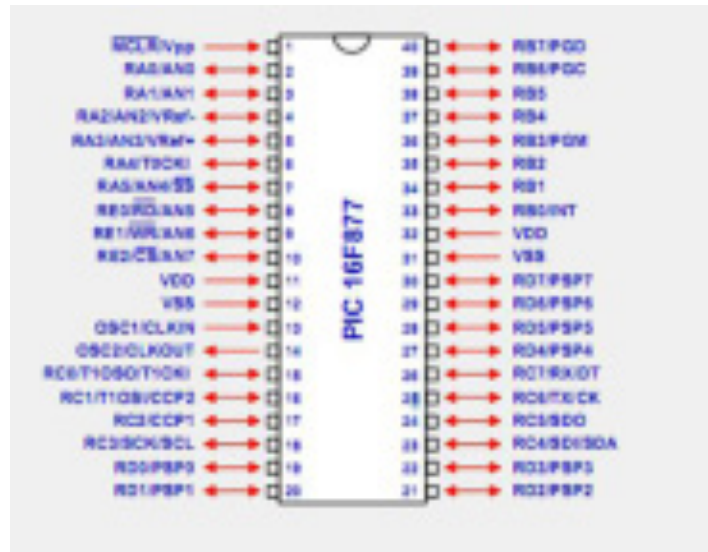


Figure 2.8 Brochage du Microcontrôleur PIC 16F877

Les courants véhiculés dans le PIC sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles.

IV.3.4.1. Ports d'entrée/sortie :

Les Pics 16F877 contiennent les 5 ports suivants :

- Port A : 6 pins I/O numérotées de RA0 à RA5.
- Port B : 8 pins I/O numérotées de RB0 à RB7.
- Port C : 8 pins I/O numérotées de RC0 à RC7.
- Port D : 8 pins I/O numérotées de RD0 à RD7.
- Port E : 3 pins I/O numérotées de RE0 à RE2.

Tous ces ports se trouvent dans la banque 0, mais tous leurs registres se trouvent dans la banque 1, pour déterminer les modes des ports (I/O), il faut sélectionner leurs registres TRISX:

- Le positionnement d'un bit à « 1 » place le pin en entrée.
- Le positionnement de ce bit à « 0 » place le pin en sortie.

IV.3.4.2. Interruption RB0/INT :

Cette broche a une double fonction. En faite, elle peut être utilisée comme une broche standard RB0 ou comme une entrée d'interruption INT.

Si cette broche est utilisée comme une entrée d'interruption externe, elle doit être maintenue à un niveau haut par l'intermédiaire de résistances de 10 kΩ pour ne pas déclencher d'interruptions imprévues, cela permet aussi de relier plusieurs sources d'interruptions sur une même ligne.

IV.3.5. Module de commande :

IV.3.5.1. Relation entre (PIC, Sorties Numériques) :

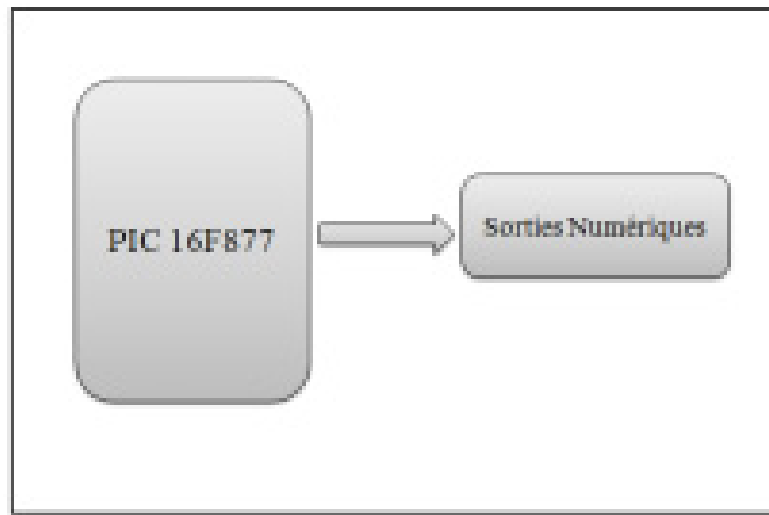


Figure 2.9: Relation entre (PIC, Sorties Numériques)

Le PIC reçoit une commande numérique sous forme d'un mot binaire (0/1) qui définit l'état de système commandé à partir de PC et l'enregistre périodiquement à l'aide d'un langage de programmation.

A l'aide d'un programme mis au point, le PIC va alors modifier l'état de 8 sorties selon la variation de ces valeurs.

Chaque sortie est commandée par un relais 12V, un transistor et une diode.

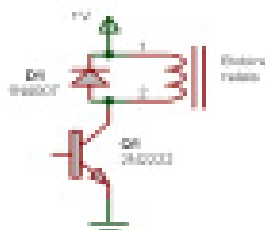


Figure 2.10: Commande de sortie numérique

En effet, un relai est un appareil composé d'une bobine (électroaimant) qui agit sur un ou plusieurs contacts, lorsqu'elle est parcourue par un courant électrique.

C'est un phénomène électrique (courant ou tension) qui contrôle la commutation On / Off d'un élément électrique (relais statique).

Comme la commande peut être réalisée sous faible puissance (faible tension, faible courant), et que la partie coupure peut commuter des puissances importantes, on peut dire que ce composant est un amplificateur de courant.

Nous présentons dans la figure suivante, la structure interne d'un relais ainsi nous ferons une étude plus détaillée sur les relais dans « l'Annexe B ».

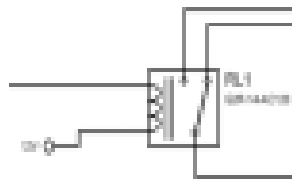


Figure 2.11: Schéma interne de relais

IV.3.5.2. Module de contrôle d'état :

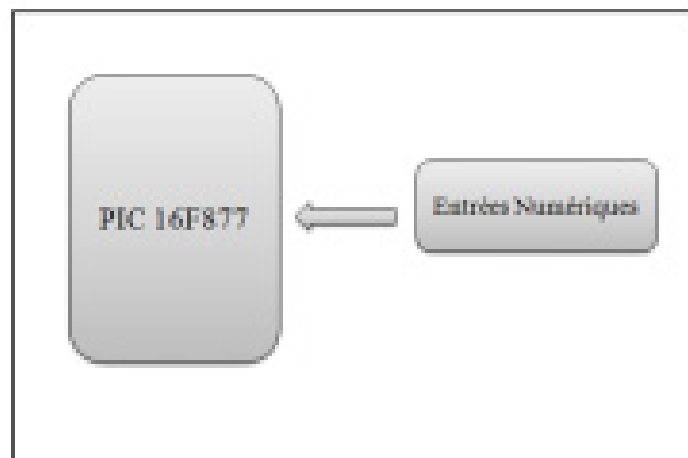


Figure 2.12 : Relation entre (PIC, Entrées Numériques)

Le PIC reçoit un mot binaire qu'il donne l'état du système (0 : non active /1 : active) et l'enregistre périodiquement à l'aide d'un langage de programmation.

I

V.3.5.3. Module de mesure de tension :

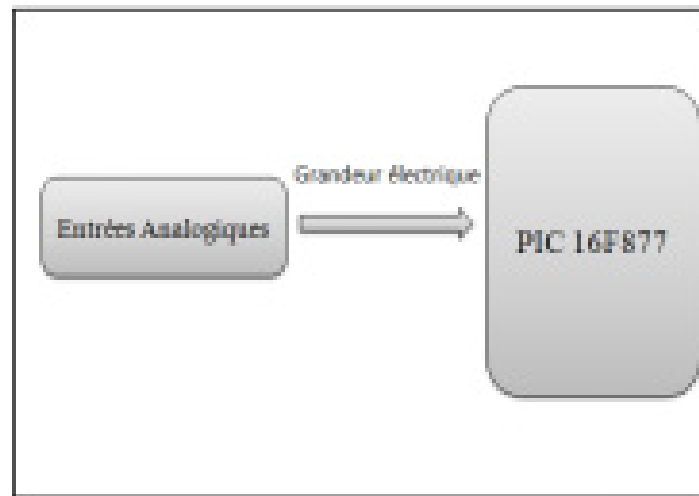


Figure 2.13: Relation entre (PIC, Entrées Analogiques)

Le PIC reçoit des grandeurs électriques telles que la tension qu'on va la mesurer et l'enregistrer dans sa mémoire périodiquement à l'aide d'un langage de programmation. Cette fonction consiste à mesurer la tension analogique aux bornes d'un système électrique.

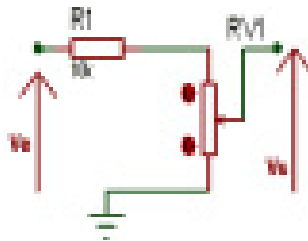


Figure 2.14: Diviseur de tension

Le microcontrôleur mesure des tensions de 5V maximum donc on a utilisé la règle de diviseur de tension pour diviser la tension à 5V.

Quelque soit la tension à mesurer on a utilisé un potentiomètre pour régler la tension à mesurer à 5V comme montre la figure 2.15.

Choix des résistances :

$$V_s = V_e \cdot \frac{R_2}{R_1 + R_2} \quad \longleftrightarrow \quad R_2 = \frac{V_s \cdot R_1}{V_e - V_s}$$

IV. 3.5.4. La liaison RS232 :

La liaison RS232 est une liaison série qui permet de transmettre des informations sans avoir à transmettre l'horloge de synchronisation. Elle utilise des tensions de fonctionnement non compatibles avec la logique 5V car elle fonctionne sur les niveaux +12V ou -12V. Son principal intérêt est un nombre de fils réduit, puisque trois suffisent à la transmission d'informations : un fil pour émettre, un autre pour recevoir et un dernier pour la masse.

Pour notre projet, nous avons dû prévoir la connexion du port série du PC à la plaquette grâce à une prise DB9.

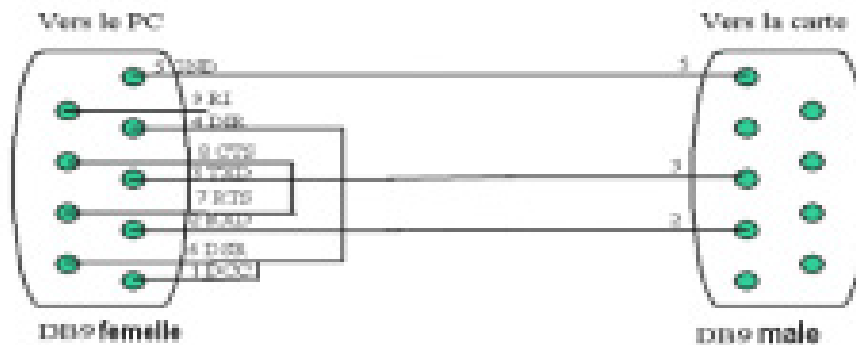


Figure 2.15: Câble RS232

Le boîtier ne peut qu'envoyer ou recevoir des données codées entre 0 et 5V du fait de l'alimentation du PIC et des différents composants qui l'entourent. Or, comme nous l'avons remarqué la liaison RS232 utilise un codage sur les niveaux +12 et -12V. Il faut donc insérer un composant nommé MAX232 en amont du connecteur DB9 afin de convertir les signaux en 0-5V.

IV.3.5.5. Présentation du MAX232 :

Le Max 232 est un standard depuis longtemps, il permet de réaliser des liaisons RS232 et des interfaces de communications, il amplifie et met en forme deux entrées et deux sorties TTL/MOS vers deux entrées et deux sorties RS232. [9]

Le MAX232 est un circuit intégré créé par le constructeur MAXIM. Il se présente sous la forme d'un boîtier DIL 16 (2* 8 broches) et s'alimente sous 5V.

Le MAX232 sert d'interface entre une liaison série TTL dont les niveaux sont 0 et 5V et une liaison RS232 qui a pour niveaux -12V et +12V.

Nous présenterons la structure interne et externe ainsi que le brochage d'un MAX232 dans la figure 2.16.

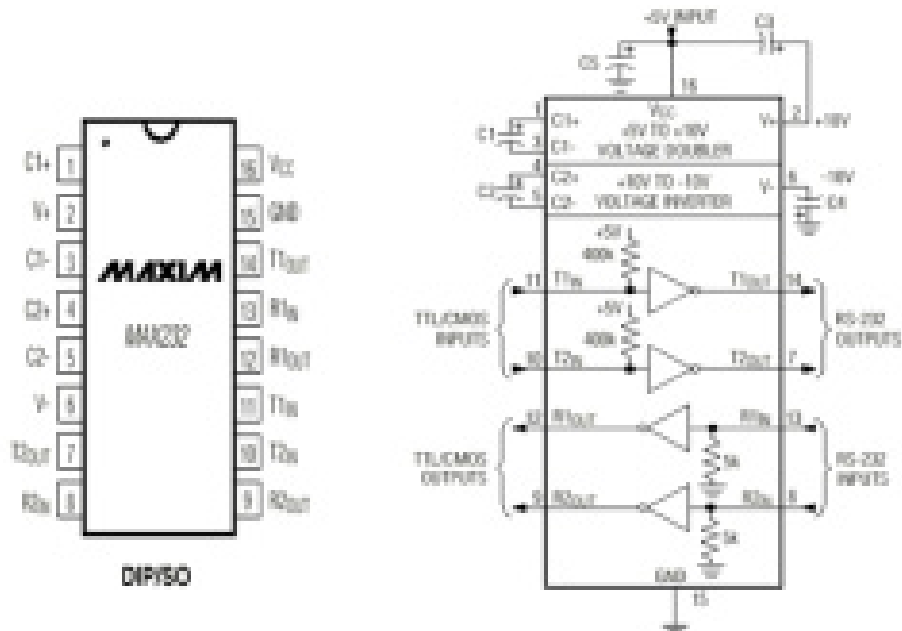


Figure 2.16: Structure interne et externe de MAX232

Nous pouvons remarquer ci-dessus qu'il y a deux entrées TTL/CMOS ainsi que 2 entrées RS232 avec leurs sorties correspondantes.

Dans notre cas il permet de réaliser la liaison RS232 entre l'ordinateur et la carte.

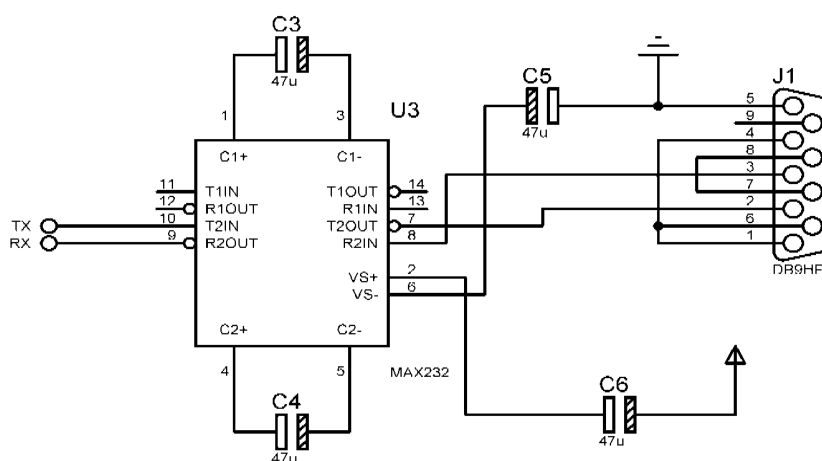


Figure 2.17: Liaison RS232 entre l'ordinateur et la carte

IV.3.5.6. Le QUARTZ :

Le quartz est composé de silice SiO_2 , qui est une matière minérale, une fois taillé en fine lamelle. Il présente la particularité d'être piézo-électrique. Généralement incolore on peut le trouver dans la nature mais on l'obtient maintenant surtout par synthèse dans l'industrie.



Figure 2.18: Schéma de QUARTZ

Après avoir fait une étude sur quelques composants de notre carte électronique dans ce chapitre, nous présenterons aussi un petit aperçu, dans l'annexe C, sur les résistances et les diodes que nous avons utilisées dans la carte.

Conclusion :

Dans ce chapitre, nous avons décrit les bases théoriques des modules à développer dans notre projet. Tout d'abord nous avons fait une étude sur « la domotique », ensuite nous avons présenté un large panorama sur le service SMS et l'entité intervenante dans la transmission de données qui est le modem GSM. Enfin, nous avons présenté aussi une étude sur les composants de notre carte électronique essentiellement le microcontrôleur 16F877-20.

Ce chapitre relève ainsi une utilité majeure pour ce qui suit puisqu'il détaille des notions exploitées au sein de la partie réalisation de notre projet.

Chapitre III

Etude Conceptuelle

- Les besoins fonctionnels
- Conception du système
 - Méthodes de conception
 - Conception de notre application

Etude conceptuelle

Introduction

En se basant sur l'étude préalable faite précédemment, nous allons exprimer les besoins fonctionnels dans une première partie afin de déterminer les caractéristiques que doivent satisfaire les composants du système pour aboutir à la réalisation de notre projet. Ensuite, nous allons aborder la partie conception qui présente l'étape la plus importante pour le développement de n'importe quel projet.

I. Les besoins fonctionnels :

La spécification fonctionnelle décrit les fonctions principales de l'application créée qui doivent répondre à nos besoins dégagés dans l'étude faite précédemment et sont résumées ci dessous:

I.1. Authentification :

Les utilisateurs du système doivent s'authentifier à travers un login et un mot de passe, pour pouvoir accéder à l'interface mobile des services offerts par notre système domotique.

I.2. Administration :

Dans notre projet nous allons développer un système qui permet d'administrer un réseau domotique à distance par un mobile.

L'administration à distance sera appliquée aux différents équipements électriques domotiques et au niveau de la base de données. En effet, notre système devra être capable:

- De visualiser les actions que l'utilisateur peut les invoquer.
- De permettre à l'utilisateur de consulter les états des équipements électriques à distance.
- De permettre à l'utilisateur de piloter les différents équipements électriques à distance.
- De permettre à l'utilisateur d'être notifié sur les différents changements dans son réseau.

II. Conception du système :

Une fois l'analyse des besoins et la spécification des exigences du projet sont élaborées, on aborde la partie conception qui constitue la phase la plus importante dans le cycle de développement d'un projet puisqu'elle permet de traduire l'ensemble des exigences exposées en une solution.

Tout d'abord, nous donnons une idée sur les méthodes de conceptions utilisées dans une application. Ensuite nous présenterons la conception de notre système domotique après avoir précisé la méthode utilisée.

II.1. Méthodes de conception :

Une méthode de conception définit une démarche reproductible qui vise l'obtention des résultats fiables. D'une manière générale, les méthodes permettent de construire des modèles à partir d'éléments de modélisation qui constituent des concepts fondamentaux pour la représentation des systèmes ou des phénomènes.

Les méthodes définissent également une représentation souvent graphique qui permet d'une part de manipuler aisément les modèles, et d'autre part de communiquer et d'échanger l'information entre les différents intervenants. Une bonne représentation recherche l'équilibre entre la densité d'information et la lisibilité [6].

II.1.1. Méthodes fonctionnelles

Elle consiste à définir les fonctions des composantes d'un système et leurs relations fonctionnelles. Le système est conçu d'un point de vue fonctionnel, en partant d'une vue de haut niveau, qu'on affine successivement afin d'obtenir une conception plus détaillée. L'état du système est centralisé et partagé par les fonctions qui agissent sur cet état.

II.1.2. Méthodes orientées objets

C'est une méthode de conception qui permet de programmer en termes d'objets. Le système est vu comme un ensemble d'objets, plutôt que comme un ensemble de fonctions. L'état du système est centralisé, et chaque objet gère l'information concernant son propre état. Les objets disposent d'un ensemble d'attributs qui définissent leurs états, et d'un ensemble d'opérations qui permettent d'agir sur ces attributs [7].

II.2. Conception de l'application :

La conception de l'application vise principalement à préciser le modèle d'analyse de telle sorte qu'il peut être implémenté avec les composants de l'architecture. Cette opération représente la phase la plus complexe du projet.

II.2.1 Choix du modèle de conception :

Dans le cas de notre projet, on a choisi l'approche objet pour la conception de l'application. En effet, l'approche objet est une idée qui a plusieurs avantages dont on cite [7]:

- Le système développé est plus facile à maintenir du fait que les objets sont indépendants. Ils peuvent être modifiés. Mais, le fait de modifier l'implémentation d'un objet ou de lui ajouter des services ne doit pas affecter les autres objets du système.
- Les objets sont considérés comme des composants réutilisables appropriés vu leur indépendance. On peut alors développer des conceptions à l'aide des objets créés dans une autre conception.
- Pour certaines classes du système, il existe une correspondance claire entre les entités du monde réel (tels que les composants matériels) et les objets du système qui le contrôlent ce qui permet d'améliorer la compréhension de la conception.

Pour la modélisation de notre application, on a choisi le langage UML (Unified Modeling Language) qui permet de modéliser un problème de façon standard.

II.2.2 Langage UML :

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

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage [6].

UML unifie aussi les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression des besoins jusqu'à le codage. C'est un langage formel possédant les caractéristiques suivantes :

- ✓ Il n'est pas une notation fermée ; elle est extensible, générique et configurable par l'utilisateur.

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

UML s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel mais, on va représenter seulement ceux qui sont utilisés dans notre projet et qui sont :

- ✓ Les diagrammes de cas d'utilisation.
- ✓ Les diagrammes de classes.
- ✓ Les diagrammes de séquence.

II.2.2.1. Diagrammes de cas d'utilisation

Les cas d'utilisation permettent de modéliser et de structurer les interactions entre les utilisateurs au sens large, appelés acteurs et un système.

Les cas d'utilisation représentent un moyen d'analyse des besoins utilisateurs et permettent de relier les actions faites par un utilisateur avec les réactions attendues d'un système. Plus précisément, un cas d'utilisation unitaire est une abstraction d'un ensemble de scénarios concrets effectués sur l'initiative d'un type d'utilisateurs [6].

Les éléments de base des diagrammes de cas d'utilisation sont :

- ✓ Les acteurs: ils représentent un rôle joué par une entité externe (utilisateurs humain, dispositifs matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut modifier et/ou consulter directement l'état du système, en émettant et/ou en recevant des messages susceptible d'être porteurs de données.

- ✓ Cas d'utilisation: il représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteur/système et apporte une valeur ajoutée « notable » à l'acteur concerné.

II.2.2.2. Diagrammes de classes

Une classe d'objets est représentée par un rectangle comprenant trois parties : nom de la classe, attributs et opérations (ou méthodes). Les listes des attributs et des opérations sont toutefois optionnelles suivant le degré de détail recherché dans un diagramme : ces parties peuvent être vides ou même absentes.

Les attributs et les opérations possèdent une visibilité (notamment publique ou protégée) qui est indiquée par un symbole précédant leurs noms : si la forme d'une clef est dessinée, l'accès est protégé, il est réduit à la classe courante et à ses sous-classes.

II.2.2.3. Diagrammes de séquences

Un diagramme de séquence montre chronologiquement (de haut en bas) les interactions entre un ensemble d'objets. Chaque objet dispose d'une ligne de vie (ligne verticale). Sur ces lignes de vie, des périodes d'activités sont indiquées par des rectangles fins qui sont superposés en cas d'appel récursif [6].

II.2.3. La conception de notre application :

Pour la conception de notre application, nous avons utilisé la technique UML. A partir de la définition des besoins, on identifiera les acteurs et leurs interactions avec le système, ce qui permet de déduire assez facilement le diagramme de cas d'utilisation général.

Le diagramme de cas d'utilisation général sera spécifié par d'autres cas d'utilisation. Ces diagrammes nous permettront dans une deuxième étape de réaliser le diagramme de classe du système et dans la dernière étape, les cas d'utilisation seront illustrés à l'aide de diagrammes de séquences.

II.2.3.1 Représentation des diagrammes de cas d'utilisation

Dans la sous section suivante, à partir de la définition des besoins, on identifiera les acteurs et leurs interactions avec le système, ce qui permet de déduire assez facilement le diagramme de cas d'utilisation général.

- **Présentation de l'acteur :**

Notre projet de fin d'études possède un seul type d'acteurs qui est l'utilisateur à distance.

_ *Utilisateur distant* : cet acteur a le droit de se servir de notre plateforme domotique à distance en utilisant son téléphone portable.

Nous procédons par la suite à la représentation des diagrammes de cas d'utilisation UML pour mettre en évidence les relations existantes entre les acteurs et notre système domotique.

- **Cas d'utilisation général de notre système domotique :**

A ce stade là, nous présentons le diagramme de cas d'utilisation général de notre système domotique qui sera affiché dans la figure suivante.

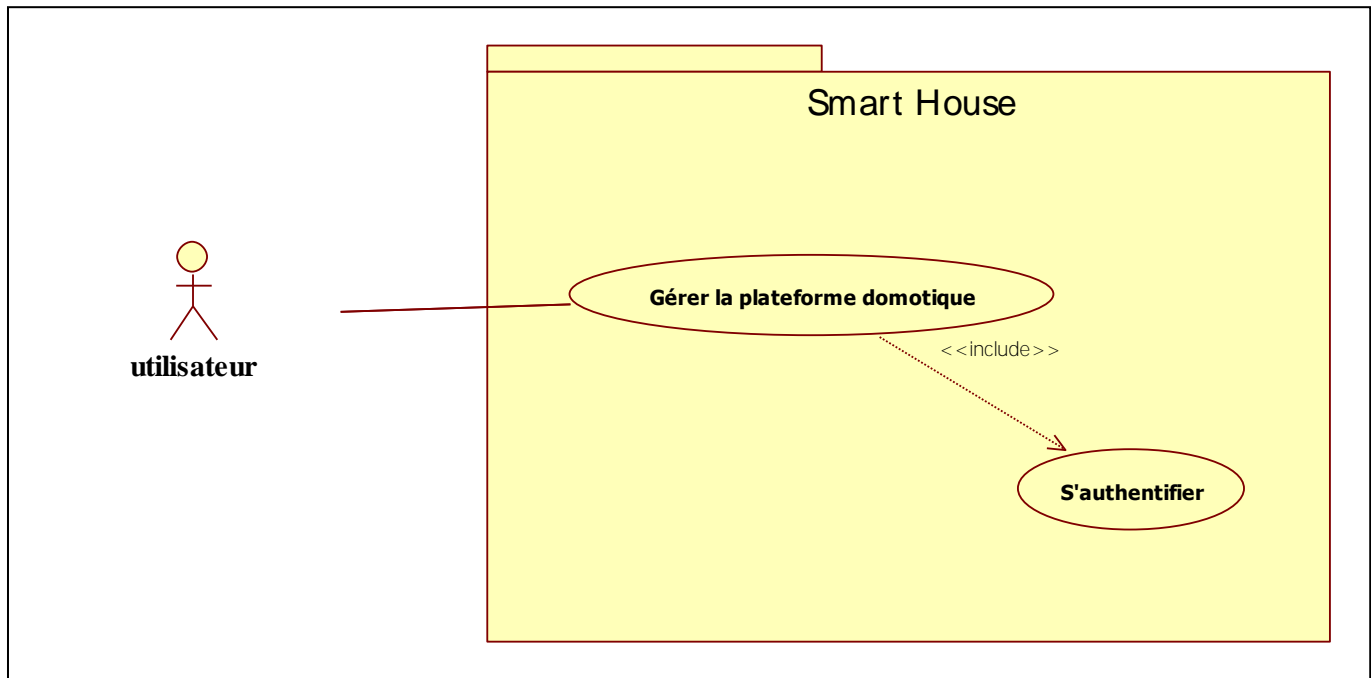


Figure 3.1: Diagramme de cas d'utilisation de notre système domotique

L'utilisateur doit s'authentifier afin de pouvoir gérer la liste des équipements électriques soit à distance soit localement.

Ce diagramme de cas d'utilisation général sera spécifié par d'autres cas d'utilisation. Ces diagrammes nous permettront dans une deuxième étape de réaliser le diagramme de classe du système et dans la dernière étape, les cas d'utilisation seront illustrés à l'aide de diagrammes de séquences qui seront présentés dans la sous-section suivante.

- **Cas d'utilisation détaillé :**

Le fonctionnement de notre système domotique peut être décrit par le cas d'utilisation de la figure suivante :

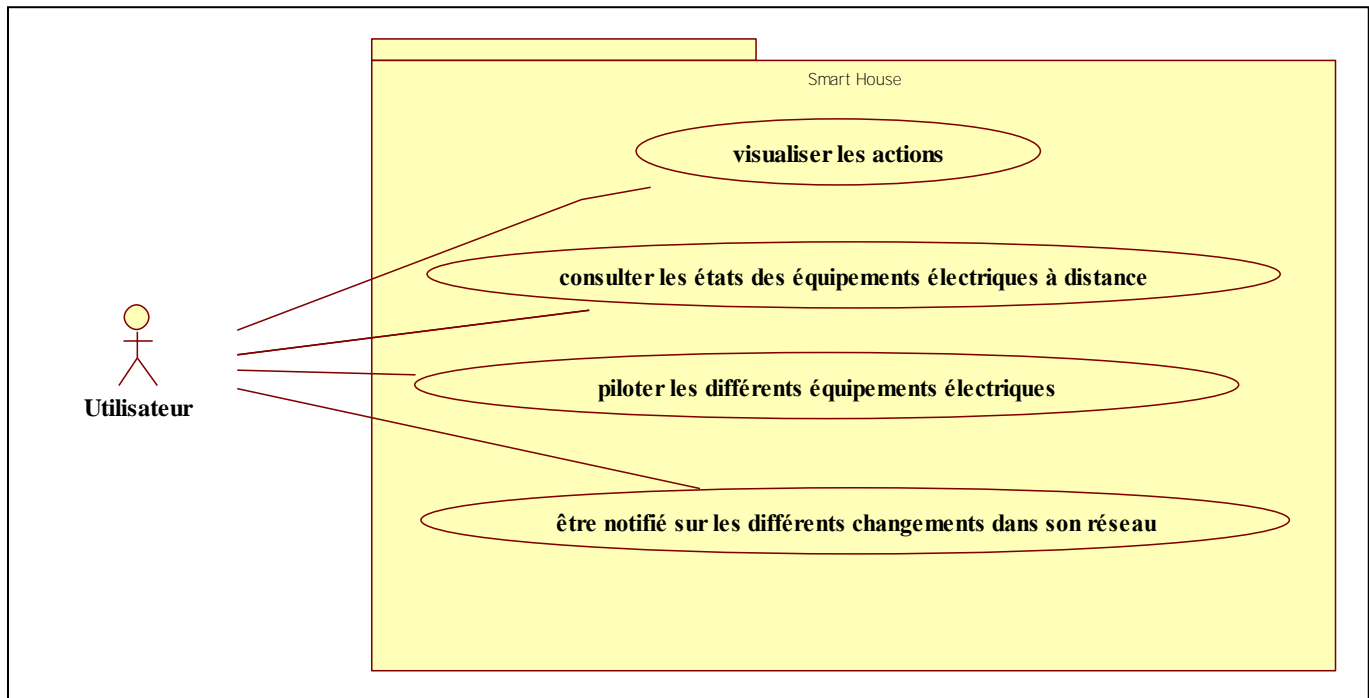


Figure 3.2: Diagramme de cas d'utilisation détaillé de notre système domotique

La figure précédente résume les cas d'utilisation de l'utilisateur distant « gérer la plateforme domotique » détaillé ci-dessous.

- «visualiser les actions que l'utilisateur peut les invoquer» : les actions que l'utilisateur peut visualiser les actions « ON », « OFF » et « Automatique » pour chacun des équipements électriques du système domotique à distance.
- «consulter les états des équipements électriques à distance» : l'utilisateur peut visualiser l'état de chaque équipement électrique du système domotique.
- «piloter les différents équipements électriques à distance» : l'utilisateur peut démarrer ou arrêter ou mettre « Automatique » chacun des équipements électriques du système domotique à distance.
- «être notifié sur les différents changements dans son réseau».

- **Cas d'utilisation de « Piloter les différents équipements électriques » :**

Le fonctionnement de pilotage de la plateforme domotique peut être décrit par le cas d'utilisation de la figure 3.3. En effet, l'utilisateur distant a le choix de démarrer, arrêter ou mettre « Automatique » les équipements électriques de notre système domotique.

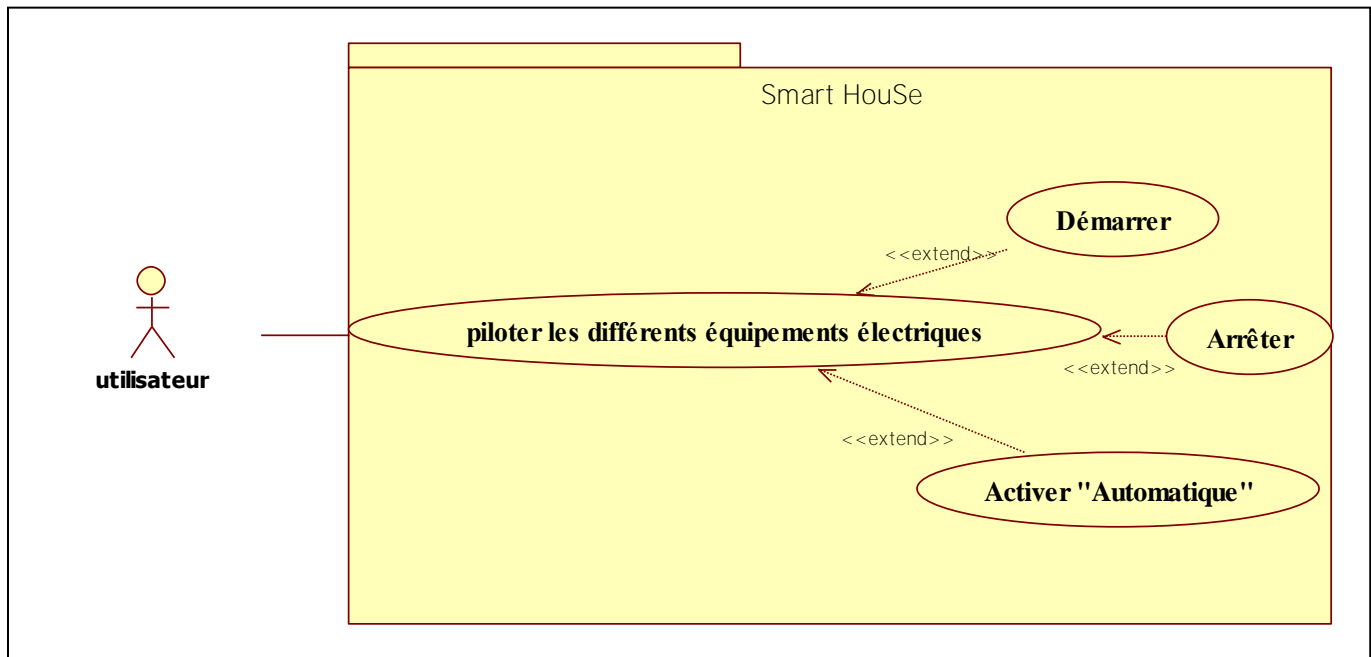


Figure 3.3: Diagramme de cas d'utilisation de « Piloter les différents équipements électriques »

II.2.3.2. Représentation des diagrammes de séquences

Les diagrammes de séquence permettent de décrire les interactions entre les objets pour chaque cas d'utilisation. Dans notre cas, ces diagrammes sont liés aux diagrammes de cas d'utilisation représentés auparavant.

- **Demande d'accès de l'utilisateur distant :**

Quand un utilisateur veut commander son réseau domotique à distance, il peut accéder à l'interface de la plateforme dans son téléphone portable et il saisit son login et son mot de passe. Les entrées seront vérifiées dans l'application mobile. Si tout est bien, l'interface de pilotage des équipements électriques sera affichée. La figure suivante présentera le diagramme de séquence « Demande d'accès de l'utilisateur distant ».

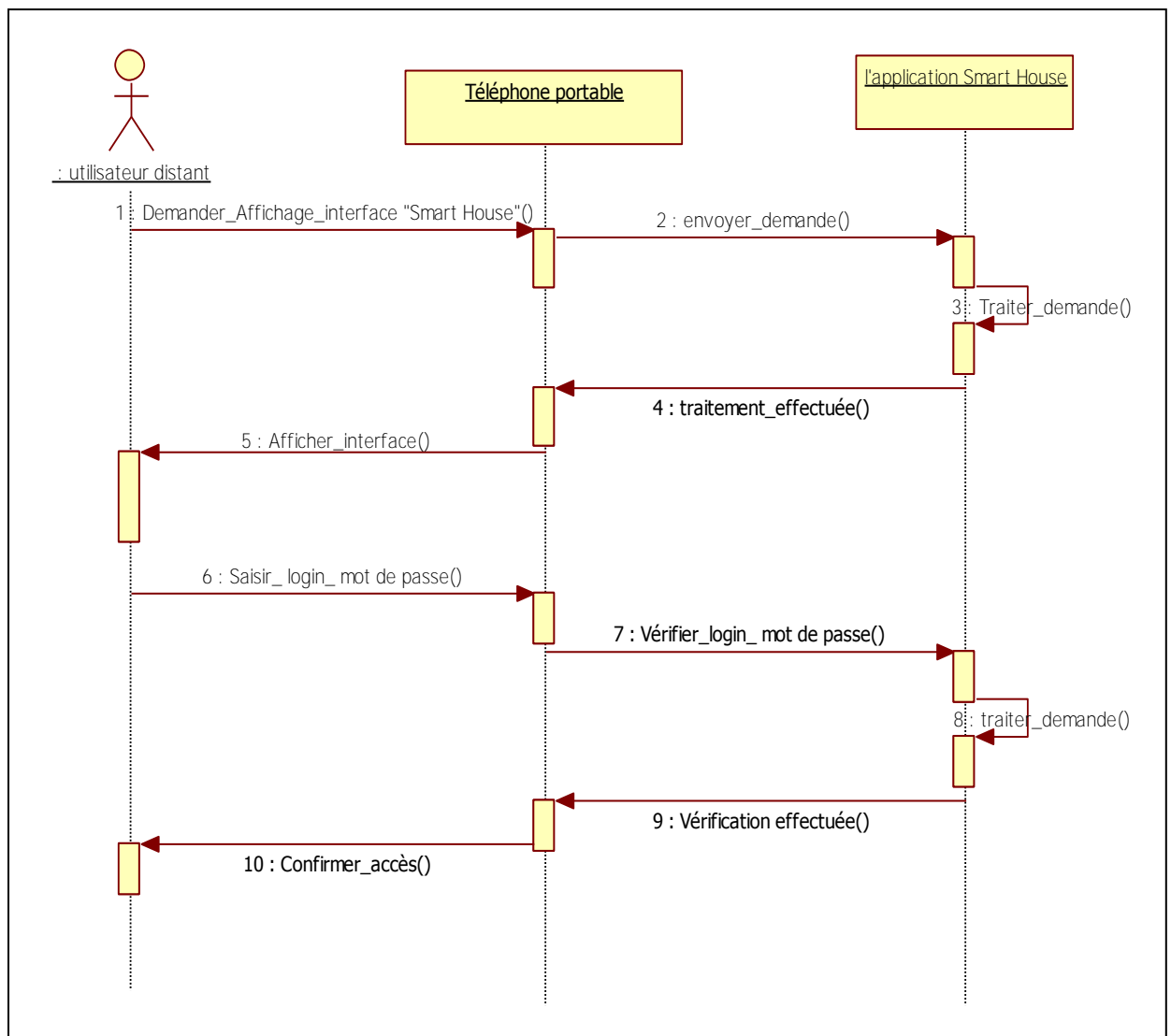


Figure 3.4: Diagramme de séquence « Demande d'accès à la plateforme domotique »

- **Activer un équipement électrique :**

Pour activer un équipement électrique, l'utilisateur doit tout d'abord saisir son login et son mot de passe. Ensuite, il demande d'afficher l'interface de la liste des équipements électriques et il choisit l'équipement qu'il veut l'activer. Après que les actions associées à cet équipement sont affichées, il saisit le bouton d'activation « ON ». Enfin, le message « équipement activé » sera affiché après le traitement de la demande sera effectué par notre application « Smart House ».

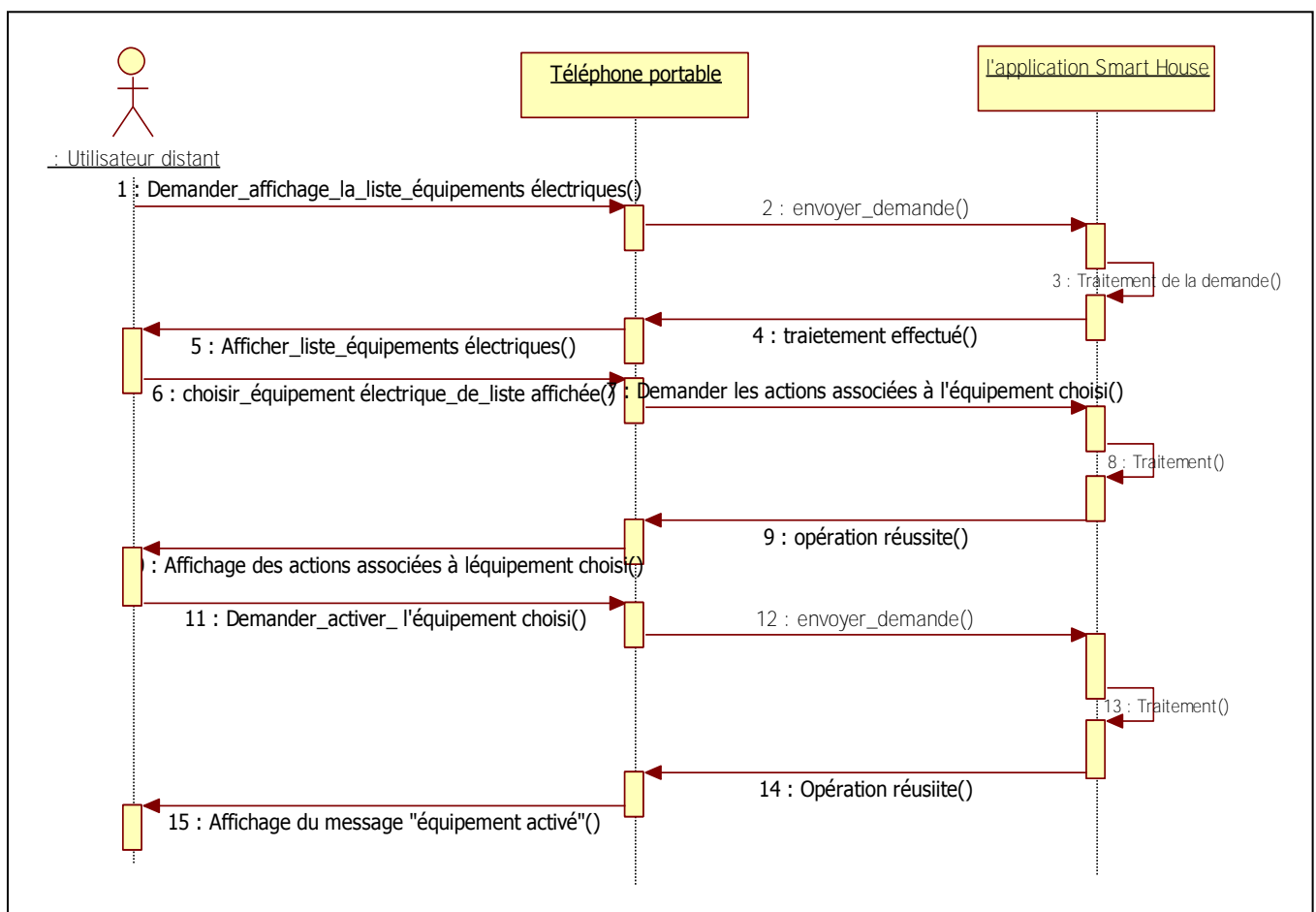


Figure 3.5: Diagramme de séquence « Activer un équipement électrique »

- **Consulter l'état d'un équipement électrique:**

L'utilisateur peut savoir l'état d'un équipement disponible sur le réseau domotique. Tout d'abord, il choisit l'équipement de la liste affichée (Climatiseur, Portes, Fenêtres, Système Alarme, Arrosage, Eclairage) et après l'affichage des actions qu'il peut invoquer, il valide son choix d'actualiser l'état de cet équipement.

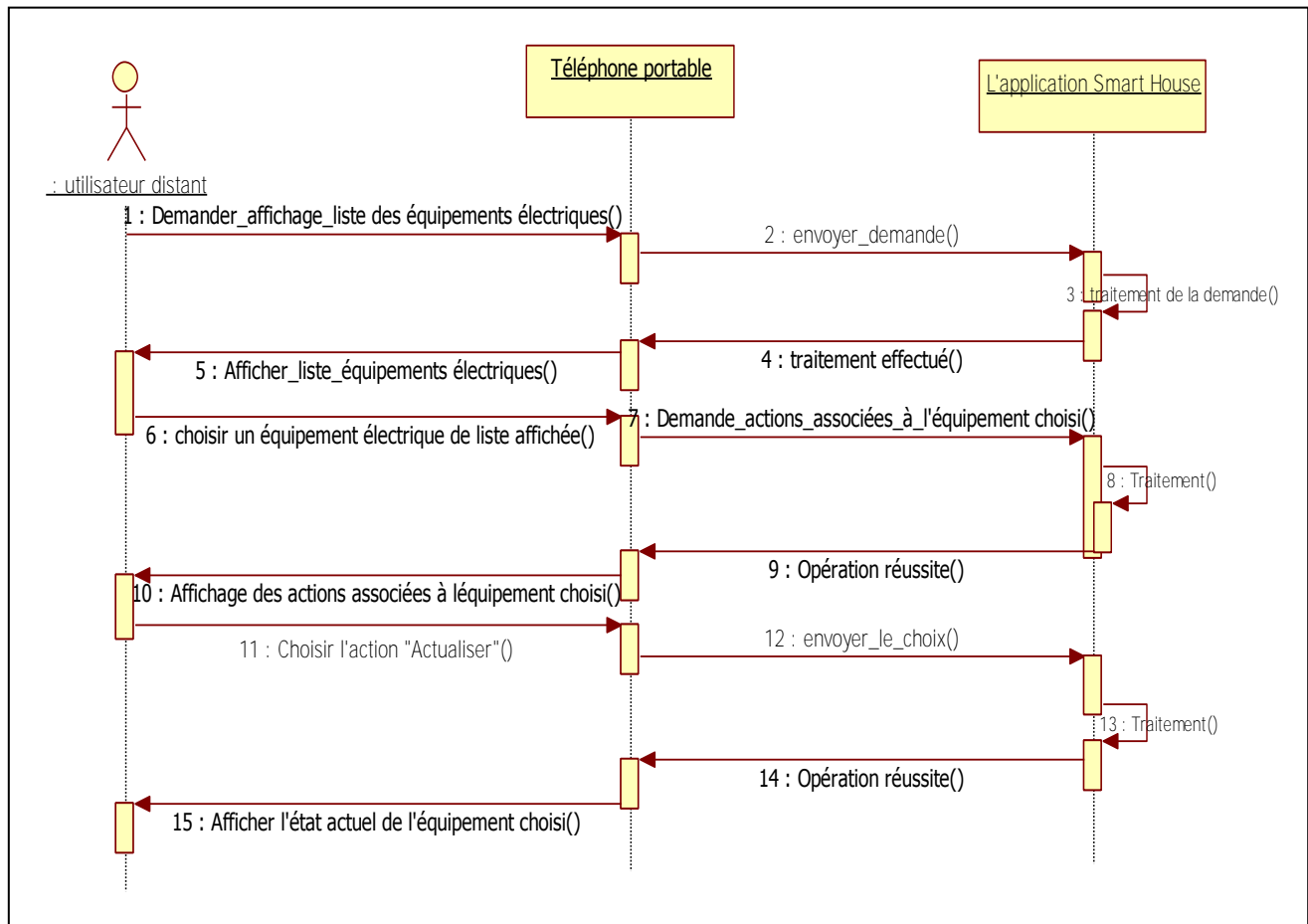


Figure 3.6: Diagramme de séquence « Consulter l'état d'un équipement électrique »

II.2.3.3. Représentation du diagramme de navigation

Bien que ce diagramme ne fasse pas parti des diagrammes UML de base, nous avons préféré de réaliser un diagramme de navigation de notre application mobile qui consiste en de nombreuses pages liées entre elles. Donc, ce diagramme nous a permis de clarifier la navigation et l'interface qui doivent être implémentées.

Conclusion :

Tout au long de ce chapitre, nous avons exposé les différents besoins auxquels doit répondre l'application à réaliser ainsi que la solution que nous allons adopter pour la satisfaction de ces exigences stipulées. Dans le chapitre suivant, nous allons aborder l'implémentation de cette solution proposée tout en évoquant l'ensemble des choix techniques et logiciels qui nous ont aidés dans la réalisation de cette tâche.

Chapitre IV

Implémentation de l'application « Smart House »

- Environnement logiciel
- Réalisation de la carte électronique
- Réalisation de l'application mobile
- Tests et résultats

Implémentation de l'application

« Smart House »

Introduction :

La réalisation est la phase la plus importante après celle de la conception. Le choix des outils de développement détermine énormément le coût en temps de programmation, ainsi que la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi en des composants logiciels formant notre système. Dans un premier volet, nous allons procéder à la spécification de l'environnement matériel et logiciel utilisé dans notre projet. Dans un deuxième volet nous expliquerons nous intéresserons à décrire les différents étapes de notre système « Smart House » et dans un dernier volet nous présenterons les tests les résultats de l'application.

I. Environnement logiciel :

Avant de commencer la description de l'implémentation de notre système, nous présenterons les logiciels et les langages que nous avons recours dans notre travail afin de réaliser notre système « Smart House ».

I.1. Choix du langage Java :

Java est un pur langage orienté objet, ce qui signifie que le niveau le plus externe de la structure des données est l'objet. Il n'y a pas de constante, de variable ni de fonction indépendante en java, on accède à toute chose via les classes et les objets. En plus, Java permet une large panoplie de fonctionnalités intégrées : xml, documentation, databases, tomcat ...

En fait, une application écrite en java peut être utilisée soit sur un PC, sur un Macintosh, sur une station Sun utilisant UNIX comme système d'exploitation, etc. Les applications développées en java ne sont pas dépendantes du système d'exploitation et comme les communications dans un système domotique peuvent s'établir entre divers types d'équipements, Java semble le langage le plus approprié.

I.1.1. Java 2 Micro Edition :

J2ME ou **Java 2 Micro Edition** est l'édition de la plateforme Java 2 à destination de

l'électronique grand public et des systèmes embarqués, et plus précisément dans les applications mobiles. Des plates-formes Java compatibles avec J2ME sont embarquées dans de nombreux téléphones portables et PDA. Une plate-forme J2ME est structurée en un modèle en couches :

- une « machine virtuelle » capable d'exécuter une application Java, elle pourra être allégée afin de consommer plus ou moins de ressources (**KVM** ou Kilobyte Virtual Machine, **CVM** ou C-Virtual Machine, ...) ;
- une « configuration », une API donnant accès aux fonctions de base du système ;
- d'un « profil », une API donnant accès aux fonctions spécifiques de la plate forme.

1.1.1.1. L'architecture J2ME :

La technologie J2ME se compose d'une machine virtuelle et d'un jeu d'APIs appropriées pour fournir des environnements d'exécution sur mesure aux terminaux mobiles. Les deux types de composants principaux de la technologie J2ME sont les configurations et les profils.

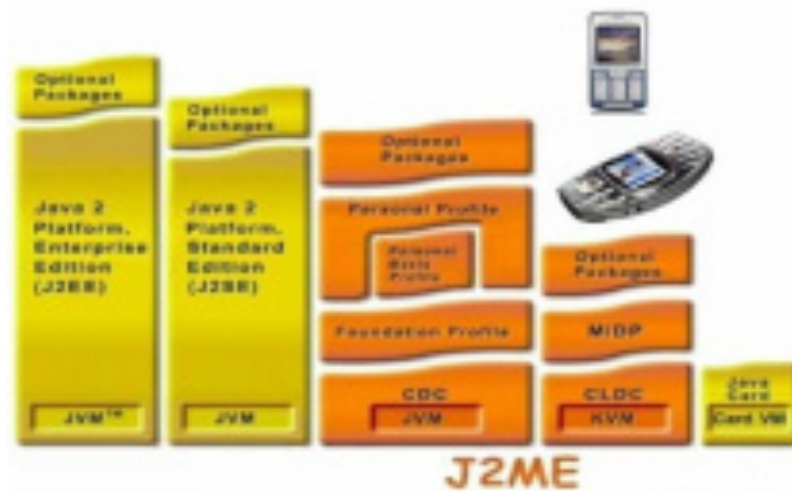


Figure 4.1 : Architecture de la plateforme J2ME

1.1.1.2. Les configurations et les profils :

Les appareils mobiles possibles sont de nature très différentes, J2ME définit alors deux types de spécifications fonctionnant conjointement, les configurations et les profils.

- **Les configurations :**

Une configuration définit une plate-forme minimale en terme de services concernant un ou plusieurs profils donnés. Elle est composée d'une machine virtuelle, des bibliothèques du noyau, de classes et d'APIs. Actuellement, il y a deux configurations J2ME

possibles : « Connected Limited Device Configuration (CLDC) » et « Connected Device Configuration (CDC) ».

Les configurations sont empilables, de sorte que ce qui est prévu pour s'exécuter sur une configuration CLDC peut s'exécuter sur une configuration CDC, pas contre la réciproque n'est pas vraie.

CLDC (Connected Limited Device Configuration) concerne les appareils possédant des ressources faibles (moins de 512 KB de RAM, faible vitesse de processeur, connexion réseau limitée et intermittente) et une interface utilisateur réduite (par exemple un téléphone mobile ou PDA 'bas de gamme'). Elle s'utilise dans une machine virtuelle KVM (Kuai VM de Sun) (=V avec limites).

CDC (Connected Device Configuration) concerne les appareils possédant des ressources plus importantes (au moins 2Mb de RAM, processeur 32bits, une meilleure connexion au réseau), par exemple un Set Top Box ou certain PDA 'haut de gamme'. Elle s'utilise sur une machine virtuelle CVM (CVM=JVM (Sun) mais plus compact). [8]

- **Les Profils :**

Pour assurer sa capacité à fournir une solution adaptée à chaque catégorie de terminaux, J2ME permet la définition de profils spécifiques. Ces profils sont des spécifications définis par les acteurs industriels de chaque catégorie de terminaux selon le Sun's Java Community ProcessSM Program et leur implémentation du profil est à la charge du fabricant de l'appareil.

Un profil est donc nécessaire pour fournir un environnement de fonctionnement complet pour un type d'appareil donné. Il fourni généralement l'interface utilisateur, les méthodes d'entrées-sorties et le mécanisme de persistance pour un groupe d'appareil. Il doit se suffire à lui même et ne pas nécessiter l'ajout d'autres classes java. Toutefois certains profils peuvent être créés pour répondre à certaines capacités ou services d'appareils (RMI, multimédia,...). Un appareil peut donc supporter plusieurs profils. Sun propose deux **profils** de référence J2ME : le profil Foundation et le profil Mobile Information Device Profile (MIDP).

Le profil **Foundation** est destiné à la configuration CDC. Les développeurs qui utilisent ce profil ont accès à une implémentation complète des fonctionnalités de J2SE.

Le profil **MIDP** est destiné à la configuration CLDC. Il prend en charge un nombre limité des classes de J2SE et définit des classes d'entrée/sortie et d'interface spécialisées pour

une configuration CLDC.

I.1.2. Netbeans :

La Plateforme NetBeans apporte aux applications bureautiques les mêmes avantages que l'architecture J2EE apporte aux applications coté-serveur:

- **Un contexte de déploiement runtime** pour des fonctionnalités arbitraires qui simplifient le développement.
- **Une boîte à outils** qui permet de gagner beaucoup de temps en développement et d'effort.
- **Un ensemble d'abstractions** qui permet aux développeurs de se concentrer sur le business logique, et non de réécrire de la logique de routine et des composants requis par la plupart des applications.
- **Un ensemble de Standards** pour rehausser et renforcer la consistance et l'interopérabilité entre les applications et les systèmes d'exploitation.

En effet, NetBeans propose des modèles de projets prêts à l'emploi:

- Pour des applications standards (pour les lignes de commandes, interfaces graphiques,...) ;
- Pour des applications Web. La version 4.1 fournira des modèles pour J2EE, comme EJB et web-services ;
- Une entrée "sample", qui contient des exemples de code.

I.2. ISIS :

L'ISIS est un logiciel professionnel, utilisé dans l'électronique pour simuler des circuits et créer des typons. Il est également capable de simuler le fonctionnement du PIC avec tous les périphériques de la carte de commande [9].

L'utilisation du logiciel « ISIS » permet de mieux visualiser le bon déroulement du système ainsi que d'avoir une idée claire sur la partie matérielle et la conception des circuits imprimés. Il nous permet de limiter les essais réels.

I.3. ARES :

C'est un logiciel permettant le routage des cartes électroniques en mode automatique ou manuel. Il est possible d'utiliser ARES sans avoir créé au préalable un schéma dans ISIS [10].

Cette fonctionnalité permet de réaliser des circuits de faible complexité en plaçant les composants et en traçant les pistes directement sur ARES. Une fois les connections établies, il est possible d'effectuer un routage automatique des pistes.

Avec ce logiciel nous pouvons également créer de nouveaux boitiers et les placer dans une bibliothèque.

1.4. PIC C Compiler:

Ce compilateur C est destiné à tous les programmeurs graves de la PIC qui souhaitent la commodité d'un langage de haut niveau ainsi que la vitesse de l'assembleur. Grâce à ce compilateur C nous n'avons plus à nous soucier de ROM et RAM paging, nous pouvons appeler à une profondeur limitée par la RAM non pas par l'appel sur 8 niveaux de pile [11].

Le PIC C Compiler gère l'un des actuels 14 bits PIC et futurs dispositifs peuvent être ajoutés par des modifications aux fichiers d'initialisation. Tous les dispositifs sont gérés par la norme C-tête des fichiers.

1.5. PICPgm:

Le PICPgm Programmeur est un logiciel gratuit et simple pour les microcontrôleurs PIC de Microchip. Le logiciel de programmation est disponible avec une interface utilisateur graphique (GUI) et une interface de ligne de commande. Le logiciel de programmation fonctionne sur Windows 9x/ME/2k/XP/Vista, 64 bit version Windows XP x64 x64/Vista et maintenant aussi sur Linux (actuellement la version ligne de commande seulement) [12].

Pour l'interface du PC vers le microcontrôleur PIC, un matériel de programmation est nécessaire. PICPgm supporte un grand nombre de programmes différents qui peuvent être connecté au PC via le port Centronics (port imprimante), port série COM ou USB. Jusqu'à présent, de nombreux dispositifs PIC différents sont supportés ainsi.

Le programmeur a les caractéristiques suivantes [12]:

- exécute la commande version en ligne disponible sur 9x/ME/2k/XP/Vista Windows et Linux (pour WIN 2K/XP/Vista un include pilote est utilisé pour accéder au port parallèle) ;
- détecte automatiquement le dispositif PIC et utilise la méthode appropriée de programmation ;
- flash Programmes de mémoire (code), EEPROM interne et les bits de configuration à partir des données du fichier HEX ;

- Conçu pour permettre aux In-System-programmation et de programmation basse tension (pas de tension de programmation nécessaire!);
- Beaucoup de microcontrôleurs PIC appuyé ;
- Support de programmeurs populaires comme JDM, Tait, programmeur AN589 Microchip, El Cheapo, ...

II. Implémentation :

Après avoir présenté les langages et les logiciels que nous avons utilisé dans notre système « Smart House », nous allons, dans cette section, décrire toutes les étapes nécessaires dans la réalisation de notre projet qui se divise en deux grandes parties : la réalisation de l'application mobile en Java et la réalisation de la carte électronique qui est la partie la plus importante.

II.1. Réalisation de la carte électronique :

Après avoir bien étudié les différents éléments constituant notre carte électronique dans le chapitre « Etude préalable », nous pouvons maintenant passer à la réalisation physique de cette carte de commande.

II.1.1. Schéma fonctionnel de la carte:

Tout d'abord, nous allons rappeler le schéma fonctionnel et le schéma de bloc de notre carte électronique avant de passer à présenter sa réalisation physique.

En effet, dans la figure 4.2, notre carte électronique contient plusieurs modules de fonctions différentes que nous avons détaillés dans le chapitre « Etude préalable ». Ces modules sont :

- Module de traitement de l'information;
- Module de commande;
- Module de contrôle;
- Module de mesure;
- Module de communication.

En outre, la carte électronique possède 2 entrées analogiques (avec alimentation 5V), 4 entrées numériques et 9 sorties numériques (avec alimentation 24V).

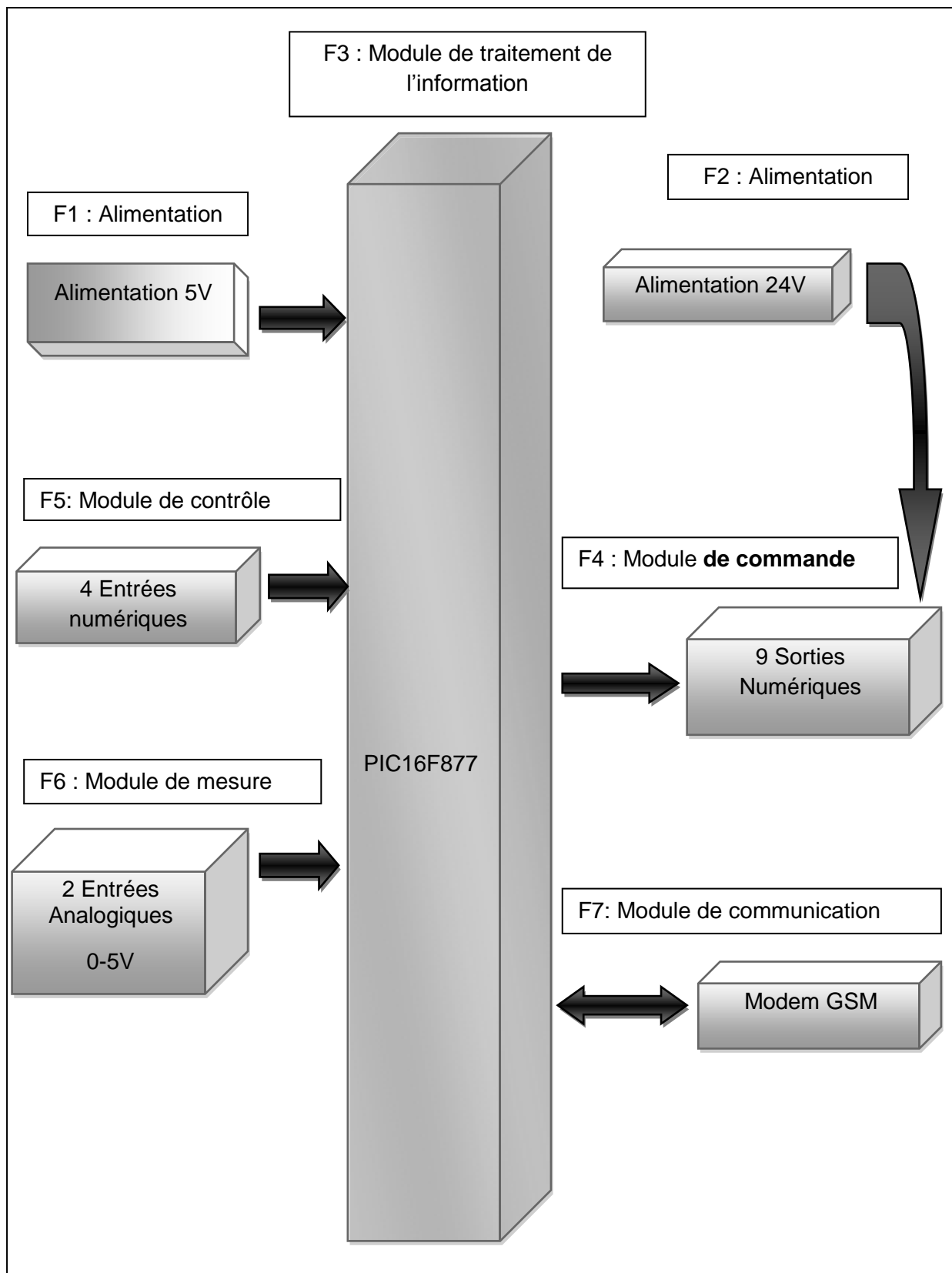


Figure 4.2: Schéma fonctionnel de la carte

Dans la figure suivante, nous avons présenté les composants de notre carte électronique : les entrées analogiques et numériques, les sorties numériques, notre microcontrôleur PIC 16F877, la prise DB9 et le MAX232.

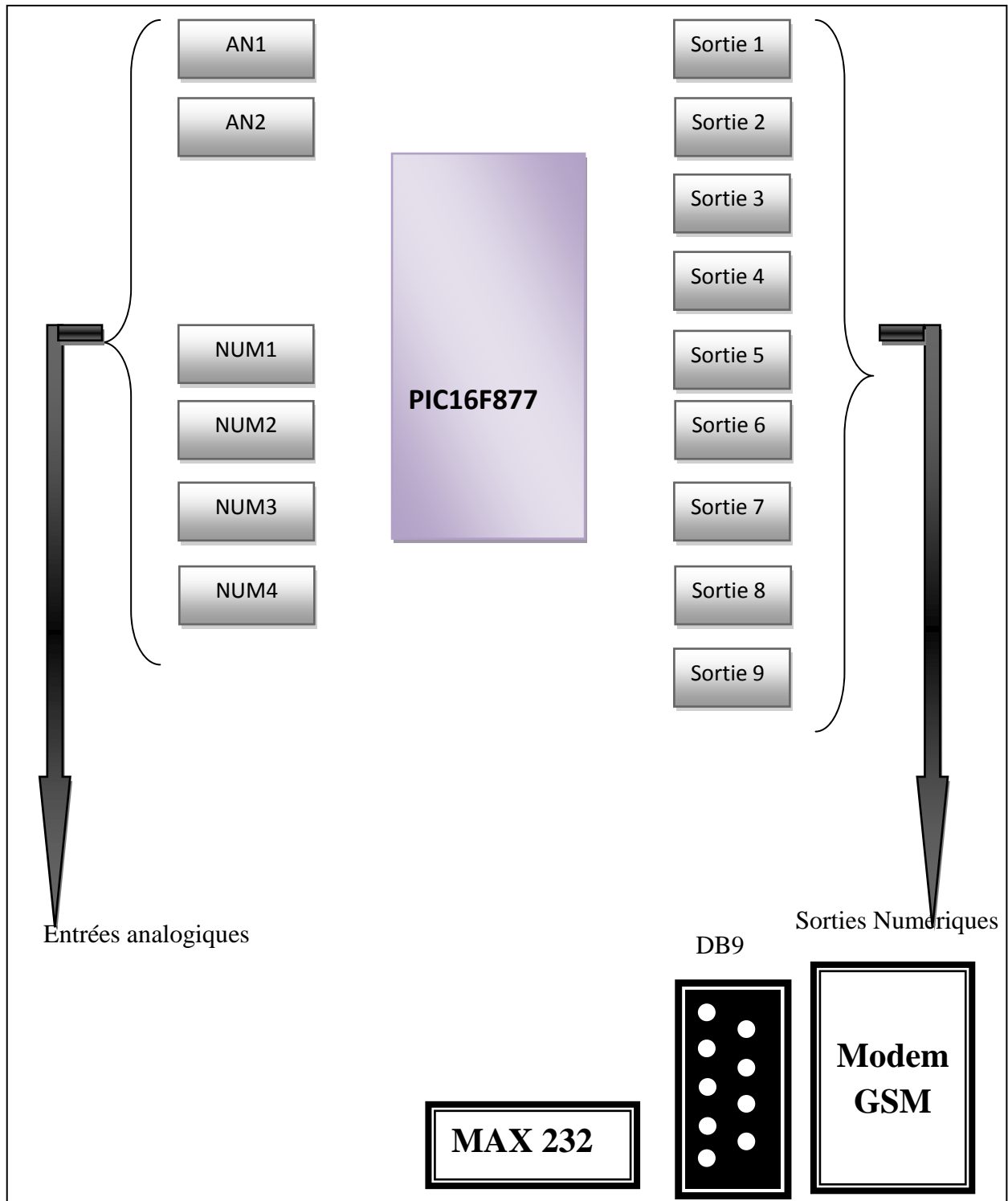


Figure 4.3 : Schéma du bloc

II.1.2. Simulation de la carte:

Pour la simulation de la carte électronique, nous avons utilisé le logiciel ISIS qui permet simuler le fonctionnement des microcontrôleurs PIC avec tous les périphériques de la carte de commande.

En effet, nous avons utilisé ce logiciel afin de mieux visualiser le bon déroulement du système ainsi que d'avoir une idée claire sur la partie matérielle et la conception des circuits imprimés.

La figure suivante (figure 4.2) présentera un imprime écran de la dernière étape de la simulation de notre carte électronique en utilisant le logiciel ISIS.

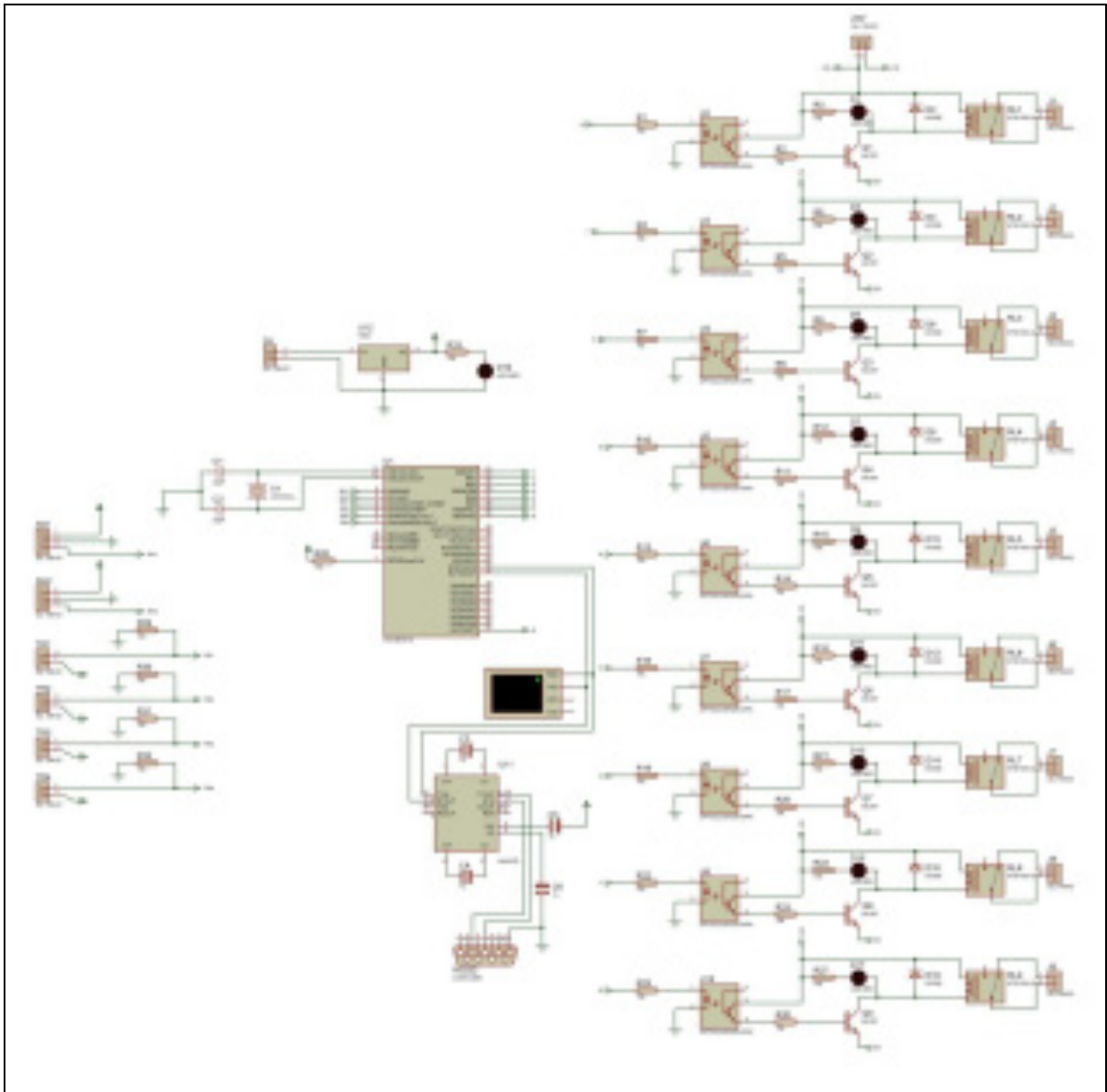


Figure 4.4: schéma de simulation de la carte électronique en ISIS

II.1.3. Routage de la carte:

Après avoir fait la simulation de la carte électronique, nous passons maintenant à l'étape suivante qui est le routage dont nous avons utilisé le logiciel ARES.

Nous présenterons, dans la figure suivante, un imprime écran du routage de la carte électronique avec le logiciel ARES.

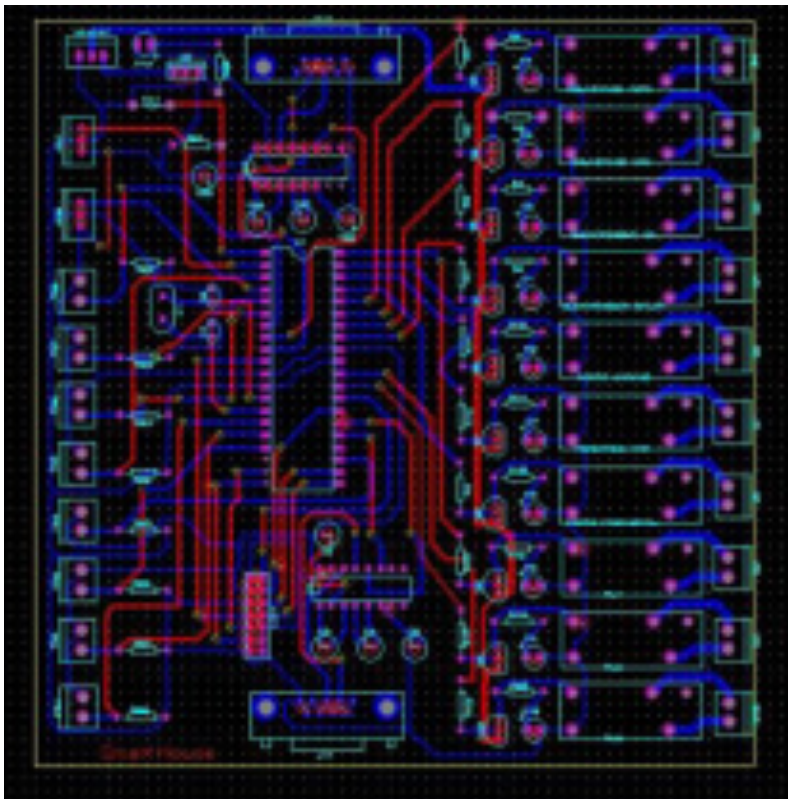


Figure 4.5 : schéma de routage de la carte sur ARES

Nous pouvons aussi, en utilisant le logiciel de routage ARES, afficher la carte électronique en 3D que nous présenterons dans la figure 4.4.

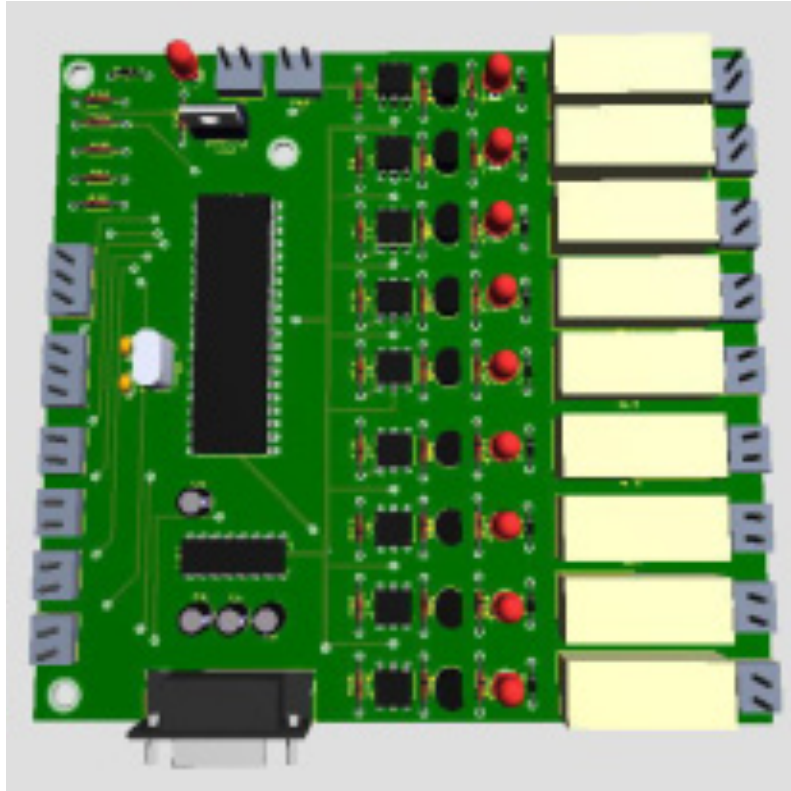


Figure 4.6: Schéma de la carte en 3D

II.1.4. Programmeur du PIC :

Après avoir réalisé la programmation du PIC en langage C, nous devons avoir un programmeur de PIC, qui nous permet d'implémenter notre programme dans les registres de PIC.

Pour cela nous avons utilisé un programmeur de PIC universel dont nous allons implémenter notre programme en C à l'aide du logiciel PIC C Compiler.

- **Présentation du programmeur de PIC :**

Ce programmeur permet de programmer une large gamme de PIC (12c508, 16f877....) aussi que la plus part des mémoires EEPROM série. Il se connecte simplement à l'interface parallèle d'un PC avec un port COM.

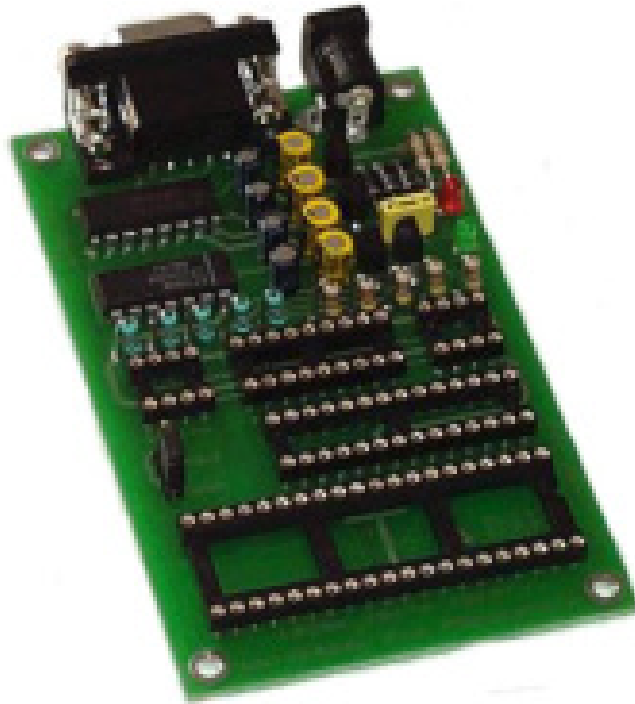


Figure 4.7: Programmeur de PIC

II.2. Réalisation de l'application mobile:

Nous avons créé notre application mobile avec J2ME (Java 2 Micro Edition) qui est une plateforme spécialisée pour les petits périphériques (mobiles et embarqués). En effet, nous avons utilisé le profil MIDP dans notre application.

En outre, toute application MIDP doit dériver d'une classe spéciale appelée « **MIDlet** » qui définit et contrôle le cycle de vie d'une application. Cette classe se trouve dans le package « *javax.microedition.midlet* ».

II.2.1. Programmation d'une classe « MIDlet » :

Le MIDP introduit un nouveau modèle, construit à la base de CLDC, qui permet à la multitude des applications de Java de se démarrer concurremment sur KVM et partager les données.

Premièrement, ce qu'on fait, c'est la création de la classe qui réalise une classe abstraite '*java.microedition.midlet.Midlet*'. Pour réaliser cette classe, il est indispensable de réaliser 3 méthodes abstraites '*startApp*', '*pauseApp*' et '*destroyApp*'. Ces 3 méthodes déterminent le cycle de vie du « MIDlet ».

Mais tout d'abord, quelques mots sur l'histoire de la création de l'objet « MIDlet ». Dans le modèle MIPD, le système éveille dans le public constructeur sans arguments pour créer un objet « MIDlet ».

La méthode '*startApp*' est appelée par le système pour démarrer ou redémarrer le « MIDlet ». Son but est la préparation du « MIDlet » au travail, par exemple, la répartition des ressources et la création de l'interface d'utilisateur nécessaire. La méthode '*startApp*' peut être finie par 2 façons :

- Transient : ce n'est pas un cas fatal, le « MIDlet » peut « dire » au système qu'il a été lancé encore une fois plus tard. Pour cela le « MIDlet » fait lancer *MIDletStateChangeException* ;
- Non-transient : c'est le cas quand il arrive quelque chose d'extraordinaire qui n'a pas été planifié, un problème, par exemple, une *Error* ou *RuntimeException*. Dans ce cas, le « MIDlet » devra se détruire à l'aide de la méthode '*destroyApp*' ;

La méthode '*pauseApp*' est appelée par le système pour suspendre l'activité du « MIDlet ». Dans cette situation, le « MIDlet » doit s'arrêter et rendre disponibles les ressources dont il n'a plus besoin. Ce dernier est très important car les ressources de KVM sont assez limitées. La méthode '*pauseApp*', en général, doit fonctionner en même temps que la méthode '*startApp*'.

La dernière méthode '*destroyApp*' s'éveille par le système pour « annoncer » au « MIDlet », qu'il sera détruit et qu'il se prépare à la procédure: fermer les ressources et sauvegarder l'information nécessaire.

Maintenant, passons à l'étude du cycle de vie du midlet. Quand il est en veille, il peut avoir l'un des 3 états suivants :

- **Paused State :**

Dans l'état « Paused State », le midlet peut demeurer, premièrement, s'il vient d'être créé, et la méthode '*startApp*' n'a pas encore été mise en circuit/service, et deuxièmement, en résultat des appels à '*pauseApp*' ou des méthodes '*notifyPaused*'. Dès que le « MIDlet » aura retrouvé cet état, il devra rendre disponibles les ressources dont il n'a plus besoin. Mais il reste toujours « en vie » car il peut toujours recevoir des messages asynchrones comme, par exemple, celui du Timer. Cependant, il faut bien noter que le système peut ne pas réaliser cet état. À titre d'exemple prenons une situation où le « MIDlet » a été démarré, et en même temps le mobile a reçu un appel. Dans ce cas, le mobile peut simplement « tuer » la machine virtuelle si ce mobile ne peut pas réaliser l'état « Paused ».

- **Active State :**

Le « MIDlet » étant en état Active après la mise en service de la méthode *'startApp'*, ou bien, après avoir quitté l'état « Paused » en résultat de l'appel à *'resumeRequest'*.

- **Destroyed State :**

L'état de « Destroyed State », le « MIDlet » se retrouve après l'appel à *'destroyApp'* ou *'notifyDestroyed'*. Étant dans cet état, le « MIDlet » ne peut pas passer à un autre état. Le passage d'un état à un autre se produit par le système ainsi que par le « MIDlet » même.



Figure 4.8: Cycle de vie d'un « MIDlet »

II.2.2. Présentation des interfaces de l'application mobile :

Dans la suite, nous allons vous présenter des exemples d'imprimés écran des interfaces mobiles en utilisant Java. Ces images, présentent quelques cas de figures en se basant sur le diagramme de navigation présenté dans le chapitre précédent « Etude conceptuelle ».

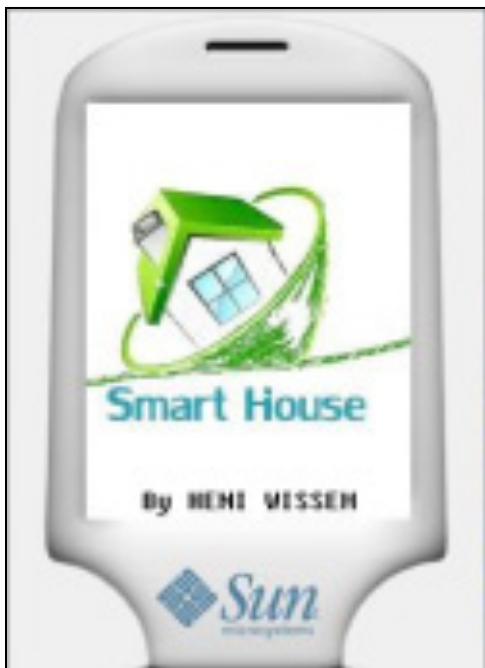


Figure 4.9: La page d'accueil de « Smart House »



Figure 4.10 : Interface d'authentification



Figure 4.11: Interface de la liste des équipements électriques



Figure 4.12: Interface de l'équipement 'Climatiseurs'



Figure 4.13: Interface pour activer le climatiseur du Salon



Figure 4.14: Interface de l'équipement 'Portes'

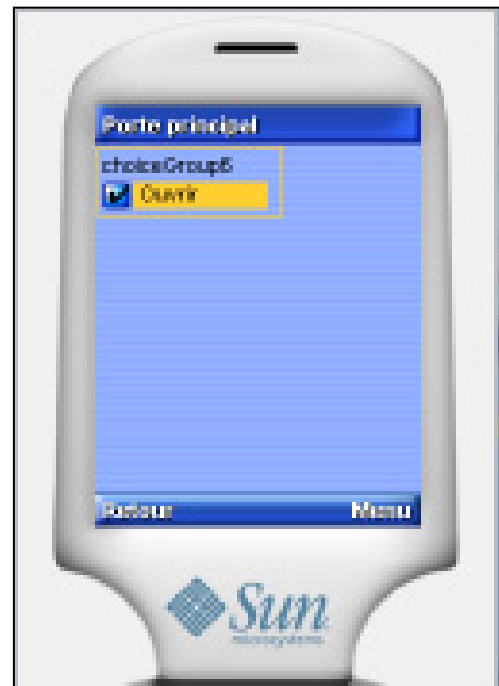


Figure 4.15: Interface pour ouvrir la porte principale



Figure 4.16: Interface de l'équipement 'Fenêtres'



Figure 4.17: Interface de l'équipement 'Eclairage'



Figure 4.18: Interface de l'équipement 'Arrosage'



Figure 4.19: Interface de l'équipement 'Système Alarme'

III. Tests et résultats :

Dans cette sous section, nous allons présenter une figure de notre carte électronique après sa réalisation physique. Cette figure nous montre les différents composants de notre carte que nous avons présenté dans le schéma fonctionnel: le microcontrôleur PIC 16F877, les entrées analogiques et numériques, les sorties numériques, les diodes, les résistances...

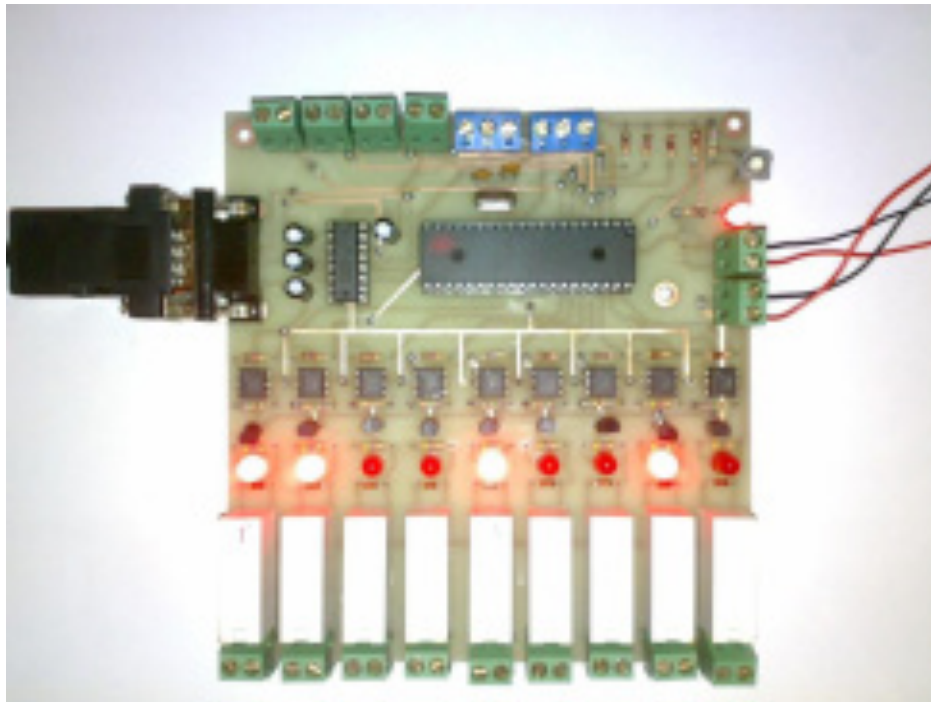


Figure 4.20 : Notre carte électronique

Conclusion :

Au cours de ce chapitre, nous avons décrit toutes les étapes nécessaires de l'implémentation de notre système « Smart House » y compris la présentation de l'environnement logiciel, la réalisation de la carte électronique et de l'application mobile afin d'aboutir à son fonctionnement.

Comme nous l'avons mentionné, notre système « Smart House » offre à l'utilisateur de l'application les fonctionnalités nécessaires pour piloter son réseau domotique afin d'assurer le bon fonctionnement du système.

Conclusion Générale

Avec le grand progrès des technologies de la communication, de l'électronique et de l'informatique qui sont réunies pour former un seul domaine : la domotique, l'utilisateur est devenu capable de communiquer avec les différents équipements domestiques.

Dans ce cadre, nous avons essayé de développer un système domotique, qui permet aux utilisateurs de piloter et de surveiller les dispositifs domestiques localement ou à distance, en utilisant un modem GSM pour l'envoi et la réception des SMS à partir d'un téléphone portable.

Au terme de ce travail élaboré dans le cadre de notre projet de fin d'études, nous considérons que ce projet nous a été bénéfique vu qu'il nous a permis de consolider nos connaissances vers la conception d'une application qui sera utile dans le domaine de la domotique. En effet, l'apport de notre projet se résume surtout dans la découverte d'un nouveau domaine, la domotique, qui est un domaine vaste et innovant et la familiarisation avec les techniques de développement qui nous ont permis d'améliorer nos compétences et nos acquis en ce qui concerne la programmation orientée Objet « Java ».

En outre, dans le cadre de ce projet, nous avons eu l'occasion de bien étudier le service de messagerie SMS et ses différents composants ainsi que détailler les méthodes d'envoi et de réception des messages SMS à partir d'un PC et l'entité intervenante dans la transmission de données qui est le modem GSM avec toutes ses particularités. En plus, nous nous sommes familiarisés avec les différents composants de notre carte électronique essentiellement le microcontrôleur PIC 16F877.

En perspectives, nous pouvons signaler que ce travail n'est qu'une simple application dans le domaine de la domotique, il peut être plus autonome, plus pratique, et assez évolutif vu le pas géant et les progrès réalisés dans les domaines de la technologie et de la communication à notre époque.

Liste des Acronymes

API: Application Programming Interface

ASCII: American Standard Code for Information Interchange

CDC : Connected Device Configuration

CLDC: Connected Limited Device Configuration

CPU: Central Processing Unit

CVM: C-Virtual Machine

EEPROM: Electrically Erasable and Programmable Read-Only Memory

ETSI: European Telecommunications Standards Institute

GSM: Global System for Mobile Communications

GUI: Graphic User Interface

J2ME: Java 2 Micro Edition

J2SE: Java 2 Standard Edition

KVM: Kilobyte Virtual Machine

MIDP : Mobile Information Device Profile

MMS: Multimedia Messaging Service

OMG : Object Management Group

OTPROM: One Time Programmable Read Only Memory

PCMCIA: Personal Computer Memory Card International Association

PDA: Personal Digital Assistant

PIC: Peripheral Interface Controller

RAM: Random Access Memory

RISC: Reduced Instruction Set Computing

ROM : Read-Only Memory

RS232: Recommended Standard 232

SM-AL: Short Message Application Layer

SMS: Short Message Service

SS7: Signalisation Sémaphore n°7

SSP: Synchronous Serial Port

UDP: Protocol Description Unit

UDHL: User Data Header Length

UML: Unified Modeling Language

UPVROM: Ultraviolet Programmable Read Only Memory

USB: Universal Serial Bus

Annexe A :

La méthode d'envoi et de réception des SMS: Mode TEXT

En mode TEXT même chose que le mode PDU il faut savoir quels sont les types de mémoires disponibles sur votre téléphone.

- Il faut s'assurer que le mode TEXT est supporté par le module GSM:

AT+CMGF= ?

- Le module retourne la liste des modes qu'il supporte :

+CMGF : (0,1)

- Activons donc le mode TEXT

AT+CMGF=1

OK

Pour envoyer un SMS il faut indiquer le numéro du SMSC que l'on souhaite utiliser. Ce paramètre est existant déjà dans la mémoire du mobile, il correspond à celui de l'opérateur auquel vous avez souscrit votre abonnement.

0: mode **PDU** (AT+CMGF=0)

1: mode **TEXT** (AT+CMGF=1)

- Pour s'en assurer demandons les paramètres associés à la commande +CSCA :

AT+CSCA ?

+CSCA : "num"

- Nous pouvons modifier ce paramètre :

AT+CSCA="+num"

OK

- Il faut entrer le numéro de téléphone du destinataire du message :

AT+CMGS=" num"

- Entrer le texte du message et valider la saisie par l'action simultanée des touches [CTRL] et [Z] .Qui provoque l'envoi du SMS sur le réseau GSM.

AT+CMGS=" num"

> Test PFE

- Si tout s'est bien déroulé, le module GSM doit retourner la réponse suivante :

+CMGS : 0

OK

Annexe B :

Les Relais



Un relais qui est présenté dans la figure précédente est un appareil dans lequel un phénomène électrique (courant ou tension) contrôle la commutation On / Off d'un élément mécanique (on se trouve alors en présence d'un relais électromécanique) ou d'un élément électronique (on a alors affaire à un relais statique).

C'est en quelque sorte un interrupteur que l'on peut actionner à distance, et où la fonction de coupure est dissociée de la fonction de commande. La tension et le courant de commande (partie "Commande"), ainsi que le pouvoir de commutation (partie "Puissance") dépendent du relais, il faut choisir ces paramètres en fonction de l'application désirée.

Dans le cas des « **relais électromécaniques** », la bobine peut générer de fortes surtensions au moment où le courant cesse de la traverser. Cette surtension (qui peut atteindre plusieurs dizaines de volts ou plus de 100 volts, même avec une alimentation de 12V) peut détruire le transistor ou la porte logique qui la commande. Pour éviter tout risque de destruction de la commande électronique qui précède le relais, il est d'usage de placer une diode dite de roue libre, en parallèle sur la bobine du relais. Cette diode doit être câblée en inverse, c'est à dire cathode vers le pôle le plus positif de l'alimentation (si vous la branchez à l'envers, elle grillera instantanément dès la première activation de la commande).

Avantages du relais électromécanique :

- Capacité de commuter aussi bien des signaux continus qu'alternatifs sur une large gamme de fréquences.
- Fonctionnement avec une dynamique considérable du signal commuté.
- Aucun ajout de bruit ou de distorsion.
- Résistance de contact fermé très faible (il est moins facile de trouver des valeurs aussi faibles avec des composants électroniques).

- Résistance de contact ouvert très élevée (il est moins facile de trouver des valeurs aussi élevées avec des composants électroniques).
- Très grande isolation entre circuit de commande (bobine) et circuit commuté (contacts).
- Possibilité de résoudre des problèmes d'automatisme de façon parfois plus simple qu'avec un circuit électronique.

Annexe C :

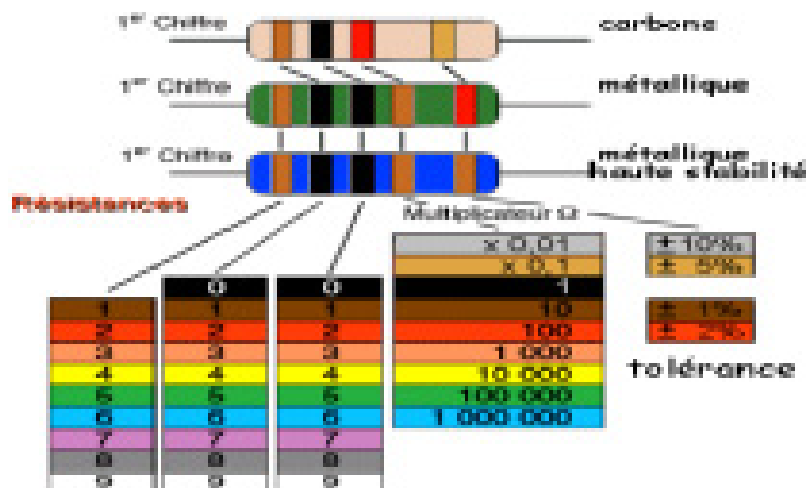
Les Résistances et les Diodes

I. Les Résistances

- Description :

La résistance (Résistor) est l'élément le plus simple, très utilisé en électronique. C'est un composant dit passif, il conduit l'électricité avec un effet résistif. Il est bidirectionnel, il n'y a pas de sens obligatoire du passage du courant.

- Symboles :



Les valeurs des différentes résistances du montage

II. Les diodes

- Description :

Les D.E.L (Diode Electro Luminescente) ou en Anglais : L.E.D (Light Emitting Diode) éclairent lorsqu'elles sont parcourues par un courant de l'anode vers la cathode.

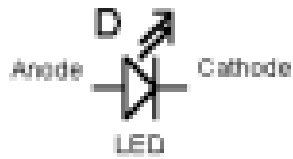


Schéma interne de Diode

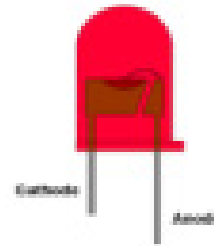


Schéma de Diode

- Utilisations :

L' avantage d' utiliser des leds est qu' elles ne s' usent pas , elles sont moins chères que des voyants, elles consomment moins d' énergie .Mais l' inconvénient est qu'elles ne peuvent fonctionner qu'avec une faible tension , et qu' elles n' éclairent pas beaucoup par rapport aux ampoules classiques .



Bibliographie :

Mémoire:

- Plate-forme domotique (SUP'COM)

Réalisé par : Aymen KORCHED

Encadré par : M. Zied CHOUKAIR
Mme. Asma BELTAIFA

Livre :

- Pascal Roque, « les cahiers de programmeur UML », Edition Eyrolles, 2001



Neto graphie :

[1]: Hager Group, 2001. <http://la.domotique.online.fr/index.htm>

[2]: <http://www.domo-energie.com/fr/page.asp?Id=58>

[3]: <http://www.domo-energie.com/fr/page.asp?Id=28>

[4]: <http://docs.kde.org/stable/fr/kdenetwork/kppp/appendix-hayes-commands.html>

[5]: http://www.technologuepro.com/gsm/commande_at.htm

[6]: <http://www.dicodunet.com/definitions/materiel/microcontrolleur.htm>

[7]: <http://fr.wikipedia.org/wiki/16F877>

[8]: <http://fr.wikipedia.org/wiki/16F877>

[9]: <http://etronics.free.fr/dossiers/analog/analog62/MAX232.htm>

[10]: <http://www.oboulo.com/effet-piezoelectrique-mesure-vibrations-leur-amortissement-17537.html>

[11]: Pascal Roque, « les cahiers de programmeur UML », Edition Eyrolles, 2001

[12] : Pierre-Alain Muller, «Modélisation Objet avec UML », Edition Eyrolles, 1998

[13] : http://www.supcom.mincom.tn/~tebourbi/cours/psm/J2ME_tebourbi.pdf

[14] : [http://fr.wikipedia.org/wiki/Isis_\(homonymie\)](http://fr.wikipedia.org/wiki/Isis_(homonymie))

[15] : Carlos Valente, « Notice ARES », Version 5.20 'Française'

[16] : http://www.e-radiocontrol.com.ar/?download=FED_PIC_C_Manual.pdf

[17] : <http://members.aon.at/electronics/pic/picpgm/>