

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira Bejaïa

Faculté des sciences exactes

Département informatique



**En vue de l'obtention du diplôme de MASTER en Informatique option
administration et sécurité des réseaux**

Thème :

**Conception et réalisation d'un réseau de capteur sans fil
applicable à la domotique**

Encadré par :

Mr. LARBI Ali

Mme MEZEGHARENE Wahiba

Réalisé par :

MOKHTARI Soumia

Devant le jury composé de :

Mr ATMANI Mouloud

Mr AISSANI Sofiane

Année universitaire 2020/2021

Remerciements

*Je tiens à remercier en premier lieu ALLAH,
le tout puissant qui m'a donné la force, le courage, la patience et
la volonté nécessaires pour accomplir ce travail.
Alhamdulillah.*

*je rends grâce et je remercie chaleureusement mon promoteur
Mr A.larbi et ma Co-promotrice Mme W.mezegharene , pour leur
bonne volonté, leur avis compétents, leur disponibilité, leur
soutien, leur patience et leur qualités humaines. Comme je
souligne que c'est
« Tout à leur honneur ».*

*Je témoigne pour leur grand mérite et leur bonne réputation
pour lesquels je déclare toute mon admiration.*

*Je remercie fortement Mr M.Atmani pour son soutien, ses
conseils et son orientation qui ont été capital dans ma
démarche.*

*Je remercie également l'école « STEM » qui a grandement
contribué à la réalisation de mon projet fin d'étude.*

*Je rends honneur à
tous les membres du jury pour leur acceptation de faire partie
de cette commission a l'effet d'apprécier et d'évaluer ce travail.*

*Je remercie ma mère de m'avoir donné l'amour de la lecture, le
savoir vivre et la culture dès ma tendre enfance. Je remercie
chaleureusement mon père pour tous ses encouragements, sa
tendresse, sa compréhension et ses sacrifices. Merci aussi à tous
les membres de ma famille d'avoir cru en moi.*

*mes pensées vont aussi à toutes ceux et celles qui ont contribué de
près ou de loin à la réalisation de mon travail.*

Dédicace

Je dédie ce travail

A ma chère, douce et tendre mère,

celle qui a illuminé mes chemins les plus obscurs, celle qui s'est donnée corps et âme pour me voir réussir, celle qui a toujours su dessiner sur mon visage le sourire. Aucune dédicace ne saurait exprimer mon respect, mon amour inconditionnel et éternel et ma considération pour les sacrifices que tu as consenti pour mon instruction et mon bien être.

A mon cher père, le plus doux de tous,

Je te remercie pour tout le soutien et l'amour que tu me portes depuis mon enfance et j'espère que ta bénédiction m'accompagnera pour toujours.

A ma chère sœur et son mari

Je vous remercie pour le soutien que vous m'avez apporté toutes ces longues années, vous étiez mon refuge, et l'épaule sur laquelle je me reposais.

A mon petit bout

Ma source d'énergie et de tendresse, ma boule de douceur.

A mon cher frère.

A ma belle Serine, mon pilier sur lequel je m'adossais étant épuisée.

A ma chère et unique meilleure amie Tania, la plus belle âme que j'ai jamais croisée de toute mon existence, personne ne pourra occuper ta place dans mon cœur. Je dédie ce mémoire à notre amitié éternelle

A mes collègues Dalía, Sylía je n'oublierais jamais nos souvenirs et folies partagés.

A ma cher tante Fahima , une femme admirable et exemplaire.

A ma chère amie psychologue clinicienne feriel MEBARKI qui m'a toujours soutenue.

A tout membre de ma famille.

A mon oncle défunt KHALI ZOUBIR .

Table des matières

Table des matières	I
Liste des figures	VI
Liste des tableaux	VII
Liste des abréviations	VIII
Introduction générale	1

CHAPITRE 1

Généralités sur les réseaux de capteur sans fil

I.1	Introduction	5
I.2	Les réseaux de capteurs	5
I.2.1	Définition d'un capteur	5
I.2.2	Définition d'un réseau de capteurs sans fil	6
I.2.3	Architecture d'un réseau de capteurs sans fil.....	6
I.2.4	Architecture d'un nœud capteur.....	8
I.3	Domaines d'applications des RCSFs.....	11
I.3.1	Le domaine militaire	11
I.3.2	L'industrie :	12
I.3.3	Le domaine médical	13
I.3.4	Les domaines urbains et domotique.....	13
I.4	Contraintes de conception des RCSFs:.....	14
I.4.1	Durée de vie du réseau	14
I.4.2	Ressources limitées	14
I.4.3	Bande passante limitée.....	14

I.4.4	Facteur d'échelle	14
I.4.5	Topologie dynamique	15
I.4.6	Mise en veille	15
I.4.7	Évolution des réseaux capteurs sans fil	16
I.5	Systèmes embarqués pour les capteurs.....	17
I.5.1	TinyOS	17
I.5.2	Contiki.....	18
I.5.3	MANTIS OS	18
I.5.4	Nano-RK.....	19
I.5.5	Lite Os.....	19
I.6	Standards de communications sans fil.....	21
I.6.1	La norme IEEE 802.15.1 / Bluetooth.....	21
I.6.2	La norme Wibree (Ultra Low Power Bluetooth)	21
I.6.3	La norme IEEE 802.15.3 / UWB (Ultra Wide Band)	21
I.6.4	La norme IEEE 802.15.4 / Zigbee	22
I.6.5	La norme IEEE 802.15.6	22
I.6.6	La norme IEEE 802.11x/WiFi	22
I.6.7	Choix de la norme	23
I.7	Conclusion.....	25

CHAPITRE 2

Preambule à la conception de l'application

II.1	Introduction	27
II.2	Définitions de la domotique	27
II.3	Description de l'application de gestion de la domotique	28
II.3.1	Exigences de l'application	28
II.4	Architecture et scénarios retenus	29

II.4.1	Architecture	29
II.4.2	Scenarios retenus	31
II.5	Internet des objets	37
II.5.1	Objets connectés	37
II.6	Plateforme et outils pour l'internet de objets	38
II.6.1	Zetta	38
II.6.2	Arduino	39
II.6.3	ChosesBoard	40
II.6.4	Node-red	41
II.6.5	Plateforme Kaa IoT	42
II.6.6	DSA	43
II.7	Description du projet	44
II.7.1	Diagramme prévisionnel d'avancement de travail	44
II.7.2	Analyse fonctionnelle	45
II.8	Conclusion	46

CHAPITRE 3

Conception et réalisation du projet

III.1	Introduction	49
III.2	Partie hardware	49
III.2.1	Présentation des cartes Arduino	49
III.2.2	Plateforme de programmation Arduino	53
III.2.3	Choix de matériel	54
III.3	Partie software	63
III.3.1	Organigramme de la partie commande	63
III.3.2	Développement de l'application android	69

III.3.3	Explication et démarche	71
III.3.4	Présentation des fonctions de système	72
III.3.5	Synchronisation de l'application Android avec la page web de commande.....	75
III.4	Conclusion	76
Conclusion générale et perspectives		78
Bibliographie		
Annexes		
Résumé		

Liste de figures

Figure I -1 : Réseaux de capteur sans fil.....	6
Figure I -2 : Ensemble des nœuds composant un réseau de capteurs sans fil.....	9
Figure I -3: Architecture d'un nœud capteur.....	8
Figure I -4 : Service militaire utilisant les RCSFs.....	12
Figure I -5 : les RCSFs dans la domotique.....	13
Figure I-6 : Evolution des capteurs.....	16
Figure II-1 Plan 3D de la domotique.....	30
Figure II-2 Vue globale de l'application domotique.....	31
Figure II-3 Scénarios retenus.	32
Figure II-4 Détecteur de fumée.	33
Figure II-5 Alarme piscine.	33
Figure II-6 Alarme anti-intrusion.	34
Figure II-7 : Moniteur d'énergie.	35
Figure II-8 : LED ampoule sans fil.	37
Figure II-9 : Thermostat Qivivo.	38
Figure II-10 : La lampe DAL (premier objet connecté).	38
Figure II-11 : diagramme prévisionnel d'état d'avancement	45
Figure II-12 : Diagramme de pieuvre.....	46
Figure III-1 : carte Arduino LEONARDO.	50
Figure III-2 : carte Arduino NANO	50
Figure III-3 : carte Arduino UNO.	51
Figure III-4 : carte Arduino MEGA 2560.	52
Figure III-5 : Composants de l'écran principal de l'IDE Arduino.	53
Figure III-6 : Structure générale d'un sketch Arduino.....	54
Figure III-7 : Capteurs DHT11.....	53
Figure III-8 : Capteurs DHT22.....	55
Figure III-9 : Diagramme de câblage.	56
Figure III-10 Vue prototypage du montage.....	Error! Bookmark not defined.
Figure III-11 : déroulement du protocole communication.	57
Figure III-12 : Image réelle du capteur de gaz/fumée MQ-2.	58
Figure III-13 : Image réelle du servomoteur.	59

Figure III-14 : Image réelle d'une LED.	60
Figure III-15 : Image réelle du capteur flamme.	60
Figure III-16: image réelle du LDR.	61
Figure III-17 : Image réelle du module Wifi ESP8266 – ESP-MCU.....	62
Figure III-18 : organigramme de fonctionnement global de la commande à distance.....	64
Figure III-19: Organigramme de la fonction de détection de fuite de gaz ou d'incendie.	65
Figure III-20 : Organigramme de la fonction d'allumage automatique.	66
Figure III-21 : Organigramme de gestion de luminosité.....	68
Figure III-22 : Organigramme fonction des ouvrants.	68
Figure III-23: Interface de design d'AppInventor.	69
Figure III-24: interface de programmation.....	70
Figure III-25 : fenêtre d'identification	71
Figure III-26 : Fenêtre principale de l'application « my house ».....	72
Figure III-27: message d'alerte d'incendie.....	73
Figure III-28 : message d'alerte fuite de gaz.	73
Figure III-29: interface de gestion de luminosité.	73
Figure III-30: interface de gestion des ouvrants.....	74

Liste des tableaux

Tableau I-1 : Comparaison entre les caractéristiques de quelques systèmes d'exploitation...	20
Tableau I-2 : comparaison entre les différentes technologies sans fil.....	24
Tableau III-1 : tableau comparatif entre le DHT22 et le DHT11.....	55
Tableau III-2 : format d'une trame de communication.....	57

SIGLES ET ABRÉVIATIONS

RCSF	réseaux de capteurs sans fil.
GPRS	general packet radio service.
WI-FI	wireless fidelity.
CAN	control area network.
DSP	digital signal processors.
FPGA	fiels programmable gate array.
OS	operating system.
IEEE	institute of electrical ans electronics engineers.
CMOS	COMPLEmentarity metal-oxide-semiconductor.
WMSNS	wireless multimedia sensor.
DSSS	direct sequence spread spectrum r networks.
WBAN	wireless body area networks.
UWB	ultra web band.
RCF	request for comments.
MANTIS	multimodal networks of in-situ micro sensor.
NESC	network embedded system C.
API	Application Programming Interface.
DSA	Digital Signature Algorithm.
IDE	Integrated Development Environment.
USB	Universal Serial Bus.
SRAM	Static Random Access Memory.
EEPROM	Electrically-erasable programmable read-only memory.
RAM	Random Access Memory.
GND	ground.
LPG	Gaz de pétrole liquéfié.
RCO	Ratio of Correct Object.

LED	Light-Emitting Diode.
IR	Infrarouge.
LDR	light dependent resistor.
ESP	Stabilisateur Électronique Programmé.
MCU	Microcontroller Unit.
TCP	Transmission Control Protocol.
IP	Internet Protocol.
VCC	Voltage Courant continu.
XML	Extensible Markup Language.
HTTP	Hypertext Transfer Protocol.
SDK	Software Development Kit.
JSON	JavaScript Object Notation.
MAC	Media Access Control.
WIFI	Wireless Fidelity.

Introduction Générale

INTRODUCTION GÉNÉRALE

Les évolutions technologiques permanentes des systèmes de communication sans fil se sont succédés ces dernières décennies permettant l'émergence de besoins croissants des utilisateurs en termes d'accessibilité, de débits, quantité de données et de consommation d'énergie. Ces technologies sont en innovation permanente dans le but d'améliorer la connectivité des usagers mais aussi de connecter des milliards d'objets entre eux. Ces objets connectés sont des éléments physiques/ numériques autonomes, capables de communiquer entre eux créant ainsi une révolution technologique qui apporte des innovations plus ambitieuses dans différents domaines d'application. Ce qui constitue une révolution pour laquelle on estime à 80 milliards d'objets connectés à l'horizon de 2021 [34]. L'intelligence embarquée dans des objets assure leur connectivité, et répond à un besoin de contrôle ou de surveillance dans différents domaines d'applications telle la médecine, l'industrie, l'environnement, la domotique et autres.

Il est possible aujourd'hui de gérer la plupart des fonctions à partir d'un Smartphone, en intervenant sur toute la partie électronique d'une habitation. Dans ce contexte d'habitat intelligent, plusieurs études portent sur l'utilisation des réseaux de capteurs sans fil. Ces réseaux sont caractérisés par leur facilité de déploiement et leur auto-organisation. Ils sont constitués de composants intelligents, miniatures, multifonctionnels, à faible coût appelés « capteurs » (ou nœuds capteurs), aussi connus sous le nom de « motes », ou « sensors ». Ces nœuds sont caractérisés par des capacités d'acquisition, de traitement et de communication de l'information. Ils sont équipés d'une interface radio sans fil ou infrarouge qui leur permet de communiquer leurs observations et de coopérer entre eux.

Notre travail est une contribution à la mise en œuvre d'un réseau ambiant de capteurs sans fil au sein d'un habitat. Le réseau repose sur un déploiement précis, dans des zones définies, de capteurs disposant d'une certaine autonomie énergétique. Cela nécessite de développer un modèle et une architecture de communication adaptés. Pour y parvenir, la connaissance préalable de l'impact des caractéristiques de l'environnement sur la

communication inter-capteurs est indispensable. Notre application qui permet de prendre le contrôle et l'automatisation de la majorité des ouvrants et appareils de la maison impose une conception du réseau qui doit s'adapter aux besoins dans une ambiance intelligente.

Le mémoire est organisé en 3 chapitres, le premier fait objet d'une description globale sur la technologie des réseaux de capteurs sans fil, le second inclut la démarche que nous avons menée afin d'aboutir à une solution qui permet de réaliser notre projet. Le troisième aborde le développement et l'intégration d'une technologie ARDUINO propre au projet, ce chapitre est consacré à la conception et la réalisation de notre projet.

PROBLEMATIQUE

L'habitat du futur est une préoccupation actuelle qui a plusieurs objectifs. Un aménagement adapté est souvent nécessaire. Cet aménagement peut prendre la forme d'une amélioration de l'accessibilité de l'habitat, d'une meilleure sécurisation des lieux ou d'une meilleure assistance et aide à l'autonomie. Les technologies domotiques sont largement disponibles aujourd'hui, mais elles ne sont pas toujours facilement utilisables; améliorer leur accessibilité, par exemple par le contrôle de certains équipements à travers des techniques d'interaction tactile ou vocale, permet à l'utilisateur un contrôle des installations et un meilleur bien être. Dans cet objectif, il est possible d'équiper l'environnement de capteurs connectés en réseau, pour acquérir une meilleure connaissance de l'environnement domotique mais également pour donner à l'utilisateur la possibilité de piloter son habitat via des commandes tactiles à travers une tablette ou un smartphone. Cette connaissance peut aussi permettre la détection de comportements anormaux pour déclencher des alertes et prévenir les aidants, mais aussi de proposer une interaction à l'utilisateur selon le contexte ambiant remonté par le réseau de capteurs.

L'habitat doit pouvoir permettre d'offrir des outils qui assurent ce confort et bien être en offrant à l'utilisateur la possibilité de contrôler son environnement de manière permanente mais également d'alerter en cas de danger et de prévenir les risques survenue d'une perte d'autonomie ou de dépendance. Pour cela, il convient de proposer des solutions qui offrent le pilotage de certaines fonctionnalités de la maison à distance et de prévenir en cas de danger ou d'alerte. Certains de ces facteurs peuvent avoir une incidence considérable avec des

conséquences potentiellement lourdes sur la qualité de vie. Il est donc nécessaire de mettre en place des solutions à faible coût, flexibles, ouvertes, d'installation aisée pour collecter des données pertinentes représentatives de l'activité de vie de l'occupant et de son environnement. La solution que nous proposons est basée sur la mise en place, d'un ensemble de modules capteurs communicants sans fil à basse consommation, configurables et facilement interfaçables avec des équipements domotiques, et d'un système d'interaction tactile reposant sur une interface utilisateur simple et intuitive.

DESCRIPTION DE LA SOLUTION

La solution technique est constituée d'un réseau de modules génériques, intégrant des capteurs ou actionneurs sans fil, basés sur une architecture matérielle et logicielle Arduino. Arduino est une plateforme de prototypage électronique open-source simple d'utilisation pour faciliter le développement et le prototypage rapide des objets connectés.

*

Chapitre I :
**Généralités sur les réseaux de capteur
sans fil**

Chapitre I

I.1 Introduction

Les réseaux de capteurs sans fil (RCSF) sont des réseaux ad hoc généralement constitués d'entités autonomes miniaturisées appelés nœuds capteurs pouvant communiquer entre eux par liaison radio. En effet, ceux-ci sont constitués d'un ensemble de petits appareils, ou capteurs, possédant des ressources particulièrement limitées mais qui leur permettent d'acquérir des données sur leur environnement immédiat, de les traiter et de les communiquer.

Un réseau de capteurs peut être mis en place dans le but de surveiller une zone géographique plus ou moins étendue pour détecter l'apparition de phénomènes ou mesurer une grandeur physique.

Après cette courte introduction, nous aborderons dans ce premier chapitre le concept des réseaux de capteurs sans fil ainsi que leurs applications, contraintes et spécificités.

I.2 Les réseaux de capteurs

I.2.1 Définition d'un capteur

Un capteur sans fil est un dispositif électronique capable de mesurer une valeur physique (la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la position, la pression, la présence d'un gaz, la vision (capture d'image), le son, etc...), et de la communiquer à un centre de contrôle via une station de base. Un capteur a pour tâche de transformer la mesure physique observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible par un système d'information.

La notion de capteur a évolué avec le temps puisque leur domaine d'application s'est élargi. Les premiers capteurs n'étaient dédiés qu'à un unique type de mesure, les capteurs contemporains sont la combinaison de plusieurs dispositifs capables de mesurer différentes grandeurs physiques. En outre leurs possibilités de mesures multiples, les capteurs actuels se sont vus gérer des fonctionnalités qui leur permettent, en plus de l'enregistrement et de la

détection d'événements mesurables, le traitement de ces données et leur communication vers un autre dispositif. On parle alors de capteur intelligent [1].

I.2.2 Définition d'un réseau de capteurs sans fil

Un réseau de capteurs sans fil est un système distribué de grande échelle mettant en communication un grand nombre d'entités autonomes communément appelés « capteurs sans fil », ou simplement « capteurs » ces capteurs forment donc les nœuds du réseau. Dans un scénario d'application classique, plusieurs nœuds capteurs sont déployés dans une zone géographique appelée zone de captage afin de surveiller un phénomène et récolter ses données d'une manière autonome. Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées avec un routage multi sauts vers un nœud collecteur appelé puits (ou sink) qui va transmettre, via internet ou satellite, ces informations à l'utilisateur du réseau (voir Figure I-1). Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisément le type de données requises, puis récolter les données environnementales captées par le biais du nœud collecteur.

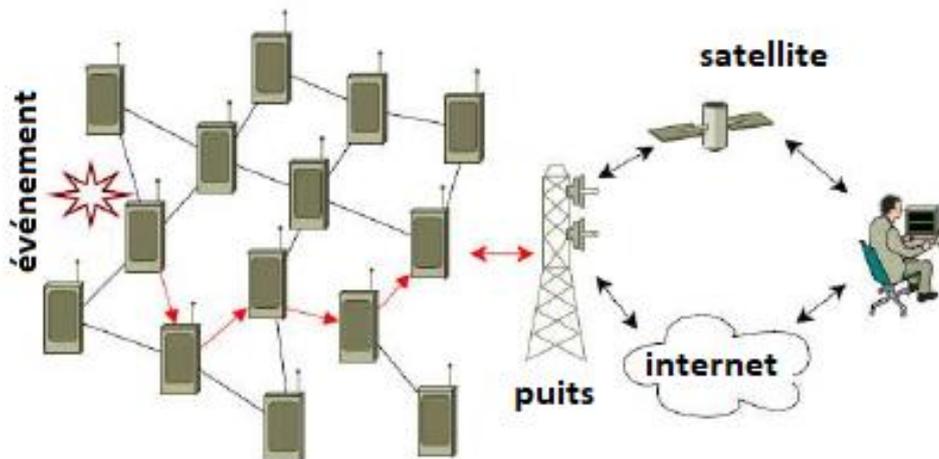


Figure I-2 : Réseaux de capteur sans fil.

I.2.3 Architecture d'un réseau de capteurs sans fil

Un réseau de capteurs sans fil générique est composé d'un grand nombre de nœuds capteurs dispersés dans le terrain d'intérêt appelé « champ de captage ». Les nœuds ont la possibilité de collecter périodiquement les données sur le phénomène surveillé et d'envoyer les rapports de captage à un nœud spécial appelé puits (Sink), comme le montre la Figure I-2. Ce nœud est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur

les types de données requise aux capteurs via des messages de requêtes. Un réseau de capteurs peut contenir plusieurs nœuds puits diffusant des intérêts différents [2].

Un réseau de capteurs sans fil est constitué d'un ensemble de terminaux (nœuds) qui communiquent par voie hertzienne. On peut distinguer trois types de nœuds:

- Les nœuds sources (nœuds puits ou coordinateurs)
- Les nœuds routeurs.
- Les nœuds terminaux.

I.2.3.1 Un nœud puits

C'est un nœud régulier doté d'un convertisseur série connecté à une seconde unité de communication (GPRS, Wi-Fi, etc.). La seconde unité de communication fournit une retransmission transparente des données provenant de nœuds capteurs à un utilisateur final ou d'autres réseaux comme internet.

I.2.3.2 Un nœud routeur

C'est un nœud régulier permettant de relayer le trafic dans le réseau sur le même canal de communication. Pour optimiser certains paramètres comme la durée de vie du réseau ou le délai de livraison des données, certains travaux se sont focalisés sur l'architecture (plat, hiérarchique, multiniveaux) des RCSF. Ces architectures définissent le plus souvent les rôles joués par les nœuds dans un RCSF.

I.2.3.3 Nœud terminal

Son rôle principal est de détecter les phénomènes physiques ou physiologiques se produisant dans son environnement immédiat afin de les transmettre, directement ou via multiples sauts, à un utilisateur final.

Notons qu'un même nœud peut jouer alternativement tous les rôles. Les nœuds peuvent être tous identiques en termes de puissance de calcul, d'énergie. En général, certains d'entre eux ont un rôle plus spécifique. C'est le cas des puits ou certain relais qui peuvent jouer le

rôle de concentrateurs. Le réseau de capteur sans fil peut avoir différentes topologies (voir figure I-2) :

- Etoile
- Maillé
- Arbre

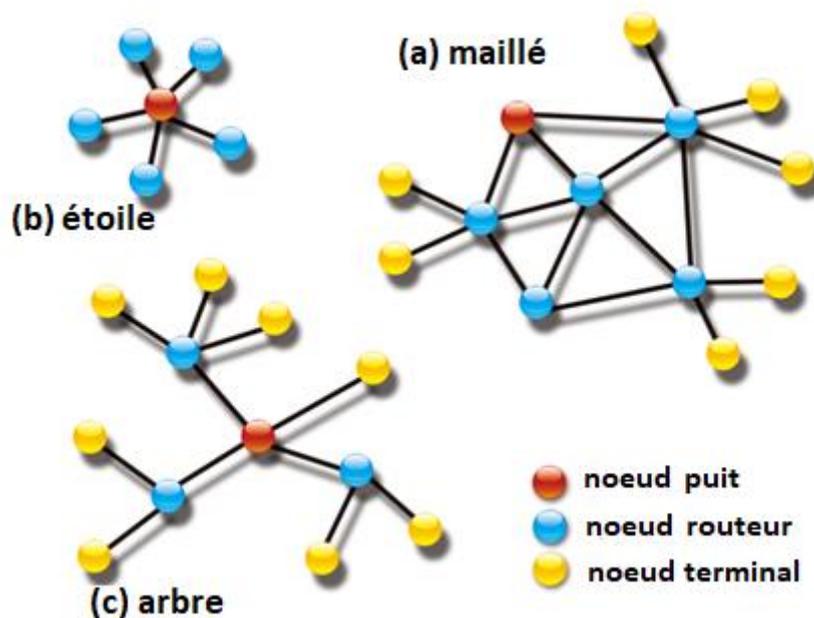


Figure I-2 : Ensemble des nœuds composant un réseau de capteurs sans fil [2].

:

I.2.4 Architecture d'un nœud capteur

Un nœud capteur contient quatre composants de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie. Il peut contenir également, suivant son domaine d'application, des éléments supplémentaires tels qu'un système de localisation, ou bien un système générateurs d'énergie (voir figure I-3).

- L'unité de captage englobe généralement deux sous-unités, le capteur lui-même en plus d'un convertisseur analogique-numérique qui transforme les signaux analogiques produits par les capteurs, et basés sur le phénomène observé en signal digitale, ce dernier est transmis par la suite à l'unité de traitement.

- L'unité de traitement, généralement associée à une petite unité de stockage, exécute les procédures permettant au nœud de collaborer avec les autres nœuds du réseau pour donner, enfin, le résultat de la tâche assignée au réseau.

La connexion du nœud au réseau est gérée par l'unité de transmission. L'unité de contrôle d'énergie, cependant, constitue l'un des systèmes les plus importants dans un nœud capteur

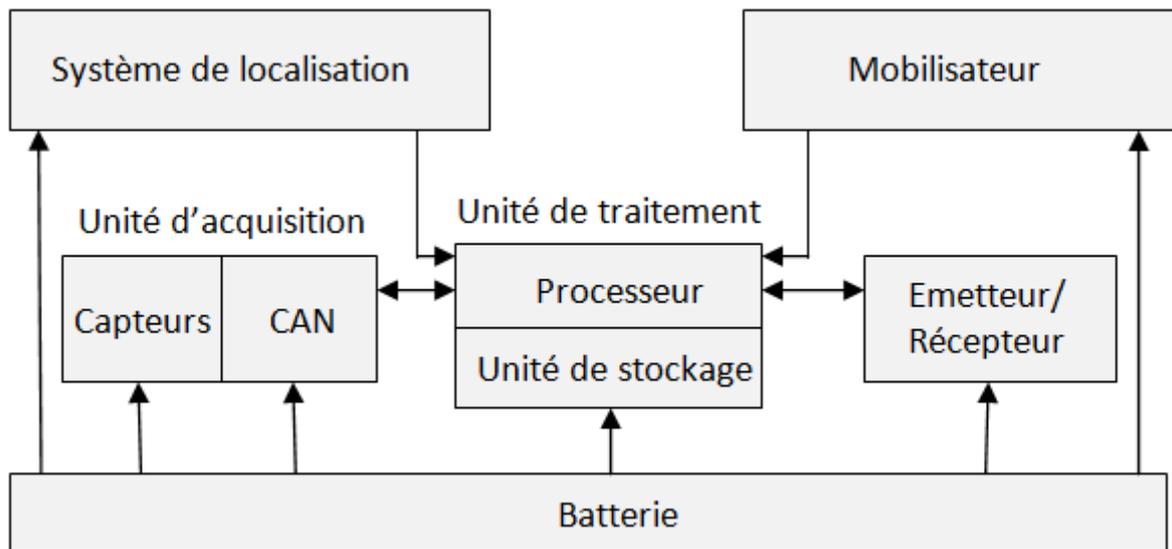


Figure I -3: Architecture d'un nœud capteur [3].

I.2.4.1 Unité d'énergie

Après leur déploiement, les nœuds d'un réseau de capteurs sont généralement inaccessibles, de ce fait, la durée de vie du réseau dépend complètement de celle de la source d'énergie du nœud capteur. Celle-ci est influencée considérablement par la contrainte de taille des nœuds. La source d'énergie est généralement une batterie [3], pour alimenter tous ses composants. Les batteries utilisées sont soit rechargeables ou non. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs et donc d'un réseau de capteurs.

I.2.4.2 Unité de captage

La fonction principale de l'unité de captage est de capturer ou mesurer les données physiques à partir de l'objet cible. Il est composé de deux sous-unités : le récepteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique (CAN). Ce dernier transforme ces signaux en données numériques et les transmet à l'unité de traitement. Un capteur peut avoir un ou plusieurs unités de captage [4].

I.2.4.3 Unité de traitements (processeur)

Elle est composée d'une interface d'acquisition et d'une interface pour l'unité de transmission ainsi qu'un processeur et un système d'exploitation spécifique. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission. Les types de processeurs qui peuvent être utilisés dans un capteur incluent le Microcontrôleur, les DSP (Digital Signal Processors), les FPGA (Field Programmable Gate Array) et les ASIC (Application Specific Integrated Circuit). Parmi toutes ces alternatives, le Micro-contrôleur a été le processeur le plus utilisé pour les capteurs à cause de sa flexibilité à être relié à d'autres composants (comme par exemple l'unité de communication), à son bon prix et sa faible consommation énergétique [5], [6].

I.2.4.4 Unité de communication

Les communications basées sur les composants de type radio-fréquence nécessitent des circuits de modulation, démodulation, filtrage, et multiplexage, ce qui implique la complexité de ce type de nœud et l'augmentation de leur coût de production. De plus, et puisque les antennes utilisées par ces nœuds sont très proches du sol, la perte du signal transmis entre eux peut être très élevée. Toutefois, ce mode de communication reste le mode préféré par la plupart des projets de recherche menés sur les réseaux de capteurs, car les paquets échangés dans ces réseaux sont de petites tailles, et ils sont transmis à un faible débit, la possibilité de réutilisation de fréquence est également considérable à cause de la petite distance entre les nœuds. Toutes ces caractéristiques favorisent l'utilisation des composants de transmission radio dans les réseaux de capteurs, mais la réalisation de tels composants avec une faible consommation d'énergie constitue, jusqu'à présent, un défi technique important.

Les technologies commercialisées disponibles telle que Bluetooth ne permettent pas encore une telle possibilité.

I.2.4.5 Unité de stockage (Mémoire)

Elle inclut la mémoire de programme et la mémoire de données. La taille de cette mémoire est souvent limitée essentiellement par les considérations économiques et elle est en continuelle amélioration au fil des années [7].

I.3 Domaines d'applications des RCSFs

Les réseaux de capteurs sans fil peuvent inclure différents types de capteurs selon la nature des signaux, à titre d'exemples, taux sismique, thermique, visuel, infrarouge, acoustique et radar, qui peuvent surveiller une grande variété de conditions ambiantes incluant : la température, l'humidité, le mouvement véhiculaire, l'état de foudre, la pression, les niveaux de bruit, la présence ou l'absence de certains genres d'objets, les caractéristiques courantes d'un objet, telles que la vitesse, la direction et la taille.

Les nœuds capteurs peuvent être employés pour la capture continue, la détection d'événements, l'identification d'événements et la commande locale des déclencheurs. Le raccordement sans fil des nœuds de micro-capteur permet un large éventail d'applications, essentiellement dans le domaine militaire et environnemental.

I.3.1 Le domaine militaire

L'exploitation militaire est l'une des principales applications des réseaux de capteurs. Dans ce contexte, l'emploi des réseaux de capteurs peut aller des surveillances de routine des périmètres, jusqu'à assister des attaques aériennes ou terrestres et conduire des opérations d'espionnage. Comme exemple d'application dans ce domaine, on peut penser à un réseau de capteurs déployé sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (Détection d'agents chimiques, biologiques ou de radiations).

Comme ces réseaux sont basés sur le déploiement dense d'un grand nombre de capteurs jetables et à coût réduit, la destruction de certains capteurs n'affectera pas une opération

militaire entreprise, ce qui rend le concept de réseau de capteurs une meilleure approche pour les champs de bataille. De plus, il existe d'autres applications militaires aux quelles les réseaux de capteurs peuvent être appliqués :

- Le contrôle des forces, équipement et munition
- Reconnaissance et surveillance du champ de bataille
- Ciblage
- Estimation des dégâts...etc.



Figure I -4 : Service militaire utilisant les RCSFs.

I.3.2 L'industrie :

Les réseaux de capteurs possèdent également d'autres applications dans le domaine commercial. Parmi ces applications, on peut énumérer: la surveillance de l'état du matériel, la gestion des inventaires, le contrôle de qualité des produits, la construction des espace d'achat intelligents, le contrôle de l'environnement dans les bâtiments administratives, le contrôle des robots dans les environnement de fabrications automatiques, les jouets interactifs, les musées interactifs, le contrôle et l'automatisation des processus d'usinage, le diagnostic des machines, le transport, la détection et la surveillance des vols de voitures, le dépistage des véhicules, l'instrumentation des chambres blanches consacrées aux traitements des semi-conducteurs, Etc...

I.4 Contraintes de conception des RCSFs:

La conception et la réalisation de plusieurs paramètres. Ces facteurs servent comme directives pour le développement des algorithmes et protocoles utilisés dans les RCSF.

I.4.1 Durée de vie du réseau

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier nœud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années.

I.4.2 Ressources limitées

En plus de l'énergie, les nœuds capteurs ont aussi une capacité de traitement et de mémoire limitée. En effet, les industriels veulent mettre en œuvre des capteurs simples, petits et peu coûteux.

I.4.3 Bande passante limitée

Afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds, les capteurs opèrent à bas débit. Typiquement, le débit utilisé est de quelques dizaines de Kb/s. Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes.

I.4.4 Facteur d'échelle

Le nombre de nœuds déployés pour une application peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec des densités de capteurs très grandes. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que la station de base soit équipée de mémoire suffisante pour stocker les informations reçues [12], [13].

I.4.5 Topologie dynamique

La topologie des réseaux de capteurs peut changer au cours du temps pour les raisons suivantes [12], [13]:

- Les nœuds capteurs peuvent être déployés dans des environnements hostiles (champ de bataille par exemple), la défaillance d'un nœud capteur est, donc très probable.
- Un nœud capteur peut devenir non opérationnel à cause de l'expiration de son énergie.
- Dans certaines applications, les nœuds capteurs et les stations de base sont mobiles.

I.4.6 Mise en veille

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données [14]. Avec cette technique, les nœuds intermédiaires agrègent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données.

I.4.7 Évolution des réseaux capteurs sans fil

Au fil des années, avec la disponibilité du matériel et dispositifs de prix abordable, des capteurs de plus en plus sophistiqués ont été proposés, permettant de capturer des données scalaires ou encore multimédias dans le but de satisfaire une grande panoplie d'applications sur les RCSF.

Dans ce qui suit, nous présentons une évolution au cours du temps des capteurs les plus utilisés ces dernières années.

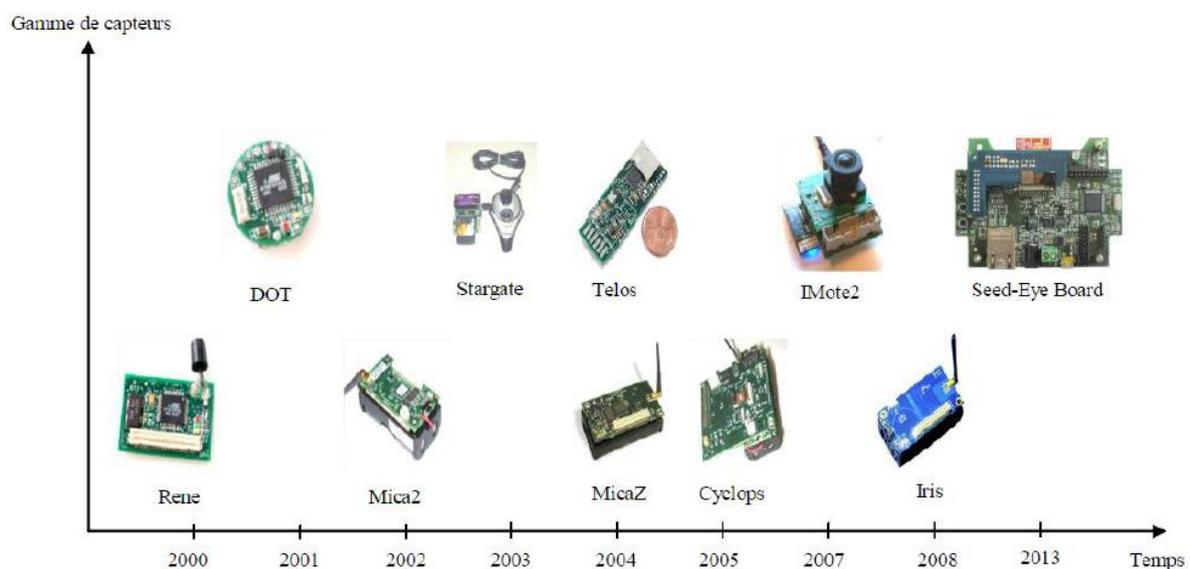


Figure I-6 : Evolution des capteurs.

Comme l'illustre la figure I-6, plusieurs capteurs ont été proposés par différentes universités comme celle de Berkeley qui a présenté plusieurs travaux sur les RCSF.

Les plus utilisés sont les capteurs fabriqués par l'entreprise Xbow (aussi appelé Crossbow) qui est née au sein de l'université californienne.

Ces capteurs conçus pour collecter des données scalaires telles que la température et l'humidité se basent sur le composant Chipcon CC2420 qui est devenu le standard au niveau des modules de transmission utilisant le protocole de communication IEEE 802.15.4 [15].

Avec les exigences des applications critiques de surveillance, une nouvelle catégorie de capteurs dotés de capacités multimédias a vu le jour, Cyclops et Stargate sont deux

exemples concrets de ce type de capteurs. Les capteurs conçus initialement pour collecter des données scalaires peuvent contribuer à la mise en œuvre des capteurs multimédias. Pour la transmission sans fils, par exemple, l'émetteur/récepteur des capteurs Mica2 ou MicaZ peut être utilisé via l'interface du capteur d'images Cyclops afin de transmettre les images capturées. Plus récemment, le capteur Seed-Eye a vu le jour, développé par la société Evidence Embedding Technology, il a été conçu pour l'implémentation des RCMSF à faible coût, il est constitué de plusieurs interfaces de communication telles que Ethernet, IEEE 802.15.4/Zigbee et USB. Il contient de plus une camera CMOS pour la capture d'images.

I.5 Systèmes embarqués pour les capteurs

Les avancées technologiques récentes ont permis de faire embarquer des systèmes d'exploitation (OS : Operating System) au sein des capteurs, mais leurs fonctionnalités restent toutefois limitées. Les systèmes d'exploitation pour les réseaux de capteurs sans fil sont des interfaces informatiques spécifiques destinées au fonctionnement des capteurs dans les réseaux. Le rôle du système d'exploitation pour un capteur en réseau est d'être l'interface entre les ressources matérielles et les applications distribuées. Il doit fournir une variété de services systèmes basiques comme la gestion de l'allocation des ressources sur les périphériques de matériels divers et la gestion et la planification des tâches. Le but du système d'exploitation est de faciliter la programmation des applications, mais aussi d'optimiser les utilisations des ressources. Il existe plusieurs systèmes d'exploitation pour les réseaux de capteurs sans fil comme : TinyOS, Contiki, MANTIS OS, LiteOS, Nano-RK [16], [17], [18], [19]. Il y a certaines caractéristiques qui font la différence entre ces systèmes d'exploitation [16], par exemple : l'architecture, le modèle de programmation, la gestion de la mémoire, le langage de programmation.

I.5.1 TinyOS

TinyOS est un système d'exploitation open source pour les réseaux de capteurs sans fil qui trouve sa genèse au sein du laboratoire d'informatique de l'université de Berkeley et qui a été l'un des premiers systèmes d'exploitation conçus pour les réseaux de capteurs miniatures. Et le plus répandu. Il est capable d'intégrer très Rapidement les innovations en relation avec l'avancement des applications et des réseaux eux-mêmes tout en minimisant la taille du code source en raison des problèmes inhérents de mémoire dans les réseaux de capteurs. La librairie de TinyOS comprend des protocoles réseau, des applications de services distribués,

des pilotes de capteurs et des outils d'acquisition des données. La contrainte énergétique due à l'autonomie des capteurs implique l'utilisation de puissance de calcul réduite. Cela entraîne le développement de logiciels contraint par la capacité de la mémoire et par la rapidité d'exécution. Les applications pour TinyOS sont écrites en langage de programmation NesC (Network Embedded System C), une extension du langage programmation C. L'utilisation du langage NesC permet l'optimisation du code et par suite réduit l'usage de la mémoire à accès aléatoire (RAM) [16].

I.5.2 Contiki

Contiki est également un système d'exploitation open source. C'est un système configurable modulaire pour les réseaux de capteurs. L'architecture hybride du noyau Contiki autorise deux modes de fonctionnement : soit multitâche, soit basé sur les événements. Contiki est un système d'exploitation conçu pour prendre le moins de place possible, avec une faible empreinte mémoire. Pour cela, le code est écrit en langage C. Un système utilisant Contiki contient des processus, qui peuvent être des applications ou des services, c.à.d. un processus proposant des fonctionnalités à une ou plusieurs applications. La communication entre processus se fait par l'envoi d'évènements. Le noyau Contiki reste, nativement, un système d'exploitation basé sur les événements. Pour obtenir le mode multitâche, une bibliothèque doit être installée. Les fonctions associées à cette bibliothèque n'accèdent pas directement à l'ensemble des ressources du capteur sans fil. Elles doivent, dans certains cas, faire appel à la partie du noyau dédié à la gestion des événements. Cette structure à deux niveaux a pour conséquence une dégradation des performances du système quand le mode multitâche est activé [17].

I.5.3 MANTIS OS

MANTIS (Multimodal NeTworks of In-situ micro Sensor) OS apparue en 2005, a été conçue par l'université du Colorado [22]. C'est un système d'exploitation léger et multitâche pour les capteurs adapté aux applications où plusieurs traitements, chacun associé à un ou plusieurs processus, sont en concurrence pour accéder aux ressources du capteur sans fil.

Il dispose d'un environnement de développement Linux et Windows. La programmation d'application sur MANTIS OS se fait en langage C. Son empreinte mémoire est faible : 500 octets en mémoire RAM et 14 kilo-octets en mémoire flash. C'est un système modulaire dont le noyau supporte également des entrées/sorties synchrones et un ensemble de primitives de concurrence [18].

I.5.4 Nano-RK

Nano-RK est un système d'exploitation temps réel multitâche pour capteurs en réseau. Il supporte le multi-hop réseau. Il possède une faible empreinte mémoire [19].

I.5.5 Lite Os

LiteOS fournit un environnement comparable àUNIX, adapté aux capteurs en réseau. LiteOS possède un système de gestion des fichiers et des commandes en mode terminal distante semblable aux commandes Unix pour gérer les capteurs. Le noyau supporte le chargement dynamique et l'exécution multitâche d'applications. Un langage de programmation orienté objet C++ est disponible pour développer des programmes et permettre leur déploiement sur les capteurs [19].

Le Tableau I-1 fait une comparaison entre les caractéristiques de quelques systèmes d'exploitation.

OS/ Caractéristiques	Architecture	Modèle de programmation	Gestion de la mémoire	Langage de programmation
TinyOS	Monolithique	Evènementielles	Mémoire statique	NesC
Contiki	Modulaire	Evènementielles et multitâches	Mémoire dynamique	C
MANTIS	Sous forme de couches	Multitâches	Mémoire dynamique	C
Nano-RK	Monolithique	Evènementielles et multitâches	Mémoire dynamique	C
LiteOS	Modulaire	Evènementielles et multitâches	Mémoire dynamique	LiteC++

Tableau I-1 : Comparaison entre les caractéristiques de quelques systèmes d'exploitation.

I.6 Protocoles de communications sans fil

Le médium utilisé par les réseaux de capteurs sans fil est l'onde radio. Parmi les grandes normes radios qui ont été utilisées pour des applications à bases de réseaux de capteurs nous citons:

I.6.1 La norme IEEE 802.15.1 / Bluetooth

Initialement, la norme Bluetooth a été proposée pour transmettre la voix et les données. Elle avait pour objectif préalable de permettre des communications sur de courtes distances avec un débit de communication limitée. Ses caractéristiques ont ainsi retenu l'attention des développeurs de capteurs. Par exemple les capteurs BtNode sont conçus pour une communication de type Bluetooth. Pour autant, le protocole Bluetooth n'est pas le protocole le plus utilisé dans les réseaux de capteurs, bien qu'il puisse répondre en partie aux problèmes de préservation de l'énergie, car il est gravement handicapé par la taille limitée du réseau qu'il peut former (8 nœuds, 1 maître et 7 esclaves).

I.6.2 La norme Wibree (Ultra Low Power Bluetooth)

Elle est considérée comme une version allégée de la norme Bluetooth fonctionnant dans la bande de fréquence des 2,4 GHz. Wibree n'utilise pas de sauts de fréquences. Cette norme prend en charge une topologie en étoile avec un maître et sept esclaves [20]. Afin de réduire la consommation d'énergie de Bluetooth, Wibree utilise une puissance de transmission et un débit symbole faibles. La consommation d'énergie de Wibree est l'équivalent de 10% de celle d'une connexion par Bluetooth. Sa limite principale est la faible portée de communication: 5 à 10 m.

I.6.3 La norme IEEE 802.15.3 / UWB (Ultra Wide Band)

Cette norme utilise des signaux radio envoyés avec une intensité très faible et des impulsions très courtes [21]. Elle opère dans la bande de fréquence de 3,1GHz à 10,6 GHz. UWB est conçue pour remplacer la norme Bluetooth afin d'offrir plus de bande passante, moins d'interférences avec les autres technologies et un délai plus court. UWB est utilisée pour les transmissions à haut débit avec une consommation électrique (proche de 400 mW). Cette technologie offre des avantages par rapport à Bluetooth. Elle consomme 50 fois moins d'énergie pour transmettre un bit par rapport à Bluetooth. [22], aujourd'hui, le standard IEEE 802.15.3 est devenu le candidat le plus intéressant pour fournir la qualité de service dans les

réseaux WMSNs (Wireless MultiMedia Sensor Networks). L'inconvénient majeur de la technologie UWB est sa faible portée de communication (environ 10 m).

I.6.4 La norme IEEE 802.15.4 / Zigbee

Elle est conçue pour être utilisée dans les communications à très faible puissance et sur des distances réduites. Cette technologie est utilisée dans les réseaux de capteurs sans fil. Par rapport à Bluetooth, cette technologie fournit une faible latence ; une couche physique « DSSS : Direct Sequence Spread Spectrum » permet aux nœuds de basculer en mode sommeil sans perdre la synchronisation. Le protocole Zigbee est basé sur le standard déploiement de réseau dense à plus de 65000 nœuds avec une portée de l'ordre de 100 mètres pour un débit de 250 Kbits/s. Ces caractéristiques en font aujourd'hui le principal protocole utilisé dans les réseaux de capteurs.

I.6.5 La norme IEEE 802.15.6

Cette norme de courte portée est utilisée par des objets ou dispositifs à ultra basse consommation, placés sur ou à proximité d'un corps humain. Elle permet un débit maximal de 10 Mbits/s. Cette norme combine des caractéristiques de sécurité, de fiabilité, de qualité de service, de basse consommation d'énergie et de protection contre les interférences, ce qui la rend adaptées de multiples applications de réseaux radio corporels (WBAN, Wireless Body Area Networks). La norme IEEE 802.15.6 définit une couche MAC unique et trois couches physiques différentes utilisables en fonction des applications visées.

I.6.6 La norme IEEE 802.11x/WiFi

Le protocole de communication WiFi est le protocole le plus utilisé pour toutes les applications sans fil. Il offre une large bande passante (de 11 à 320 Mbits/s) ce qui a permis de démocratiser l'utilisation de la technologie sans-fil dans les réseaux classiques WLANs. Les premiers capteurs sans-fil ont eu recours à ce protocole pour permettre la communication entre nœuds. Cependant, le standard de communication WiFi n'apparaît plus actuellement comme une solution viable pour les réseaux de capteurs sans fil, du fait d'un besoin énergétique trop important pour son utilisation. La durée de vie des capteurs sans fil alimentés par des piles ne dépasse que rarement quelques heures. C'est pourquoi, les applications de capteurs à base de communication sans fil WiFi sont très peu répandues.

I.6.7 Choix de la norme

Le choix d'une technologie de communication sans fil dépend des services proposés, ainsi que des besoins du concepteur du réseau. Certains paramètres comme la consommation d'énergie, le débit, la durée de vie de la pile, la portée et le nombre de nœuds supportés doivent être pris en compte. Dans le Tableau I-2, nous faisons une comparaison entre les protocoles de communications cités ci-dessus [25].

technologies	Bluetooth	UWB	Zigbee	WIFI	WBAN
Norme IEEE	802.15.1	802.15.3	802.15.4	802.11.x	802.15.6
Nombre de nœud maximum	8	128	65000	32	256
Durée de vie moyenne de la pile	Plusieurs jours	Plusieurs minutes	Plusieurs mois à plusieurs années	Plusieurs minutes à plusieurs heures
Débit théorique maximum	Bluetooth Low Energy : 1Mbit/s Bluetooth 3.0 + High Speed : 3-24Mbits/s	110-480 Mbits/s	20 Kbit/s - 40 Kbit/s	11-320 Mbit/s	10 Mbit/s
Bande de fréquence	2.4 GHz	3.1-10.6 GHz	868 MHz	2.4 – 5 GHz	-----
			2.4 GHz (global)		
Portée théorique maximum	10m	Inferieur a 10 m	10-100m	10-100m	5-10 m
Consommation d'Energie	100-200 mW	400 mW Pour 200 Mbit/s	30 mW	750-2000 mW	Jusqu'à 50 mW

Tableau I - 2 : comparaison entre les différentes technologies sans fil.

Pour les équipements personnels et domestiques équipés de petits émetteurs radios à faible consommation, ZigBee est le protocole le plus approprié.

I.7 Conclusion

Le succès qu'ont connu les réseaux de capteurs est dû à plusieurs facteurs. La taille réduite des nœuds permet un déploiement aisé dans différents environnements tels que les bâtiments, les champs et même sous la mer. L'intelligence implantée, sur les capteurs leur permet de s'auto-organiser et de démarrer le réseau sans difficultés majeures. Les réseaux peuvent comporter un nombre de nœuds allant de quelques capteurs pour atteindre des centaines voire des milliers.

Dans ce chapitre, nous avons présenté les généralités sur les réseaux de capteurs sans fil, et leurs caractéristiques. Dans les chapitres qui suivront nous allons présenter la conception de notre propre réseau de capteurs en détaillant la structure et le fonctionnement général du système ainsi que le matériel nécessaire pour la réalisation de ce dernier.

Chapitre II

Préambule à la conception

II Chapitre

II.1 Introduction

Les réseaux de capteurs sans fil (RCSF) représentent une révolution technologique qui change radicalement la façon de concevoir les systèmes domotiques de très grandes échelles. Ils promettent un meilleur confort et bien-être ainsi qu'une surveillance permanente à distance de son domicile.

L'objectif de ce deuxième chapitre est de présenter le périmètre de cette application particulière du RCSF en nous focalisant sur les différentes exigences de l'application ainsi que sur les contraintes auxquelles est soumis le réseau de capteurs. Nous exposons dans cette deuxième partie, les caractéristiques et les exigences de l'application que nous avons recensées, puis nous présentons les caractéristiques et facteurs de conception du réseau. Nous donnons également une synthèse des différentes approches. Nous concluons ce chapitre avec une synthèse sur les différents éléments retenus ainsi que la méthodologie de travail de conception adoptée.

II.2 Définitions de la domotique

Le mot domotique vient de domus qui signifie «domicile » et du suffixe –tique qui fait référence à la technique. La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans le bâtiment, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison. La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort, de sécurité et de communication [22].

A l'origine, la domotique avait donc pour but d'automatiser sa maison.

II.3 Description de l'application de gestion de la domotique

Dans ce domaine de l'habitat intelligent, plusieurs applications bénéficient des avantages inhérents à l'utilisation des réseaux de capteurs sans fil. On peut citer : la télésurveillance et la récupération continue de données physiologiques, scalaires et quantiques. Toutefois cette application doit respecter un certain nombre d'exigences et de contraintes.

II.3.1 Exigences de l'application

Plusieurs conditions doivent être remplies afin de mettre en place ce RCSF qui répond aux besoins de la domotique. Ces exigences sont les suivantes :

II.3.1.1 Prise en compte des caractéristiques de l'environnement

Parmi les caractéristiques de l'habitat que le réseau RCSF doit prendre en compte, on trouve : son architecture, ses dimensions, les obstacles, le type de matériaux de construction, le nombre de personnes à surveiller, etc.

II.3.1.2 Gestion de la mobilité

Le réseau RCSF doit fournir un certain niveau de mobilité adaptée aux besoins des aspects domotiques. Cet aspect concerne la mobilité des capteurs embarqués sur le corps d'une personne, dont il faut maintenir la connectivité et la couverture du réseau quand la personne se déplace dans son habitation [23, 24].

II.3.1.3 Respect de la vie privée :

Les données multimédias, doivent être protégées et sécurisées. La personne habilitée à la gestion de sa domotique à distance doit avoir la possibilité de contrôler et de paramétrer les données qui doivent être transmises vers l'extérieur de l'habitation.

II.3.1.4 Sécurisation des données :

Comme dans de nombreux types de réseaux sans fil, la sécurité des données est un aspect très important. Actuellement, il existe de nombreux algorithmes et techniques de cryptages efficaces qui sécurisent les échanges entre les nœuds capteurs (distribution aléatoire des clés, protocole de routage sécurisé, etc.) [25].

II.3.1.5 Faible coût de déploiement

Le coût de mise en place d'un réseau de capteurs doit être maîtrisé. Ce coût inclut celui des capteurs environnementaux, ainsi que le coût d'installation et de maintenance du réseau [26] auxquels vient s'ajouter le coût lié au service.

Il existe d'autres exigences de l'application comme la facilité d'installation, la flexibilité et la robustesse du réseau, la convivialité et l'ergonomie (facilité d'utilisation, interface graphique adaptée à la personne), l'efficacité et l'interopérabilité.

Après avoir étalé les exigences principales de l'application, nous présentons dans la section suivante l'architecture et les scénarios retenus dans cette domotique.

II.4 Architecture et scénarios retenus

II.4.1 Architecture

La Figure II-2 montre une vue globale de l'architecture de la maison, la Figure III-1 indique une vue 3D de de cette dernière. L'application inclut l'usage d'un ensemble de capteurs hétérogènes qui permettent l'observation de l'état de l'environnement.

On trouve également deux types de communication :

II.4.1.1 La communication intra-habitat

Elle regroupe la transmission (via une technologie sans fil adéquate) des données recueillies de la maison vers une station de base. Les données peuvent être visualisées grâce à une interface graphique (sur smartphone par exemple).

II.4.1.2 La communication entre la station de base et le monde extérieur

La station de base est interconnectée à un lien de communication externe qui peut être de type : Internet, téléphonique, satellite, etc. Les informations recueillies par cette station de base sont transférées à destination vers l'interface graphique.

Notre contexte d'étude est focalisé sur les communications à l'intérieur de la domotique.



Figure II-1 Plan 3D de la domotique.

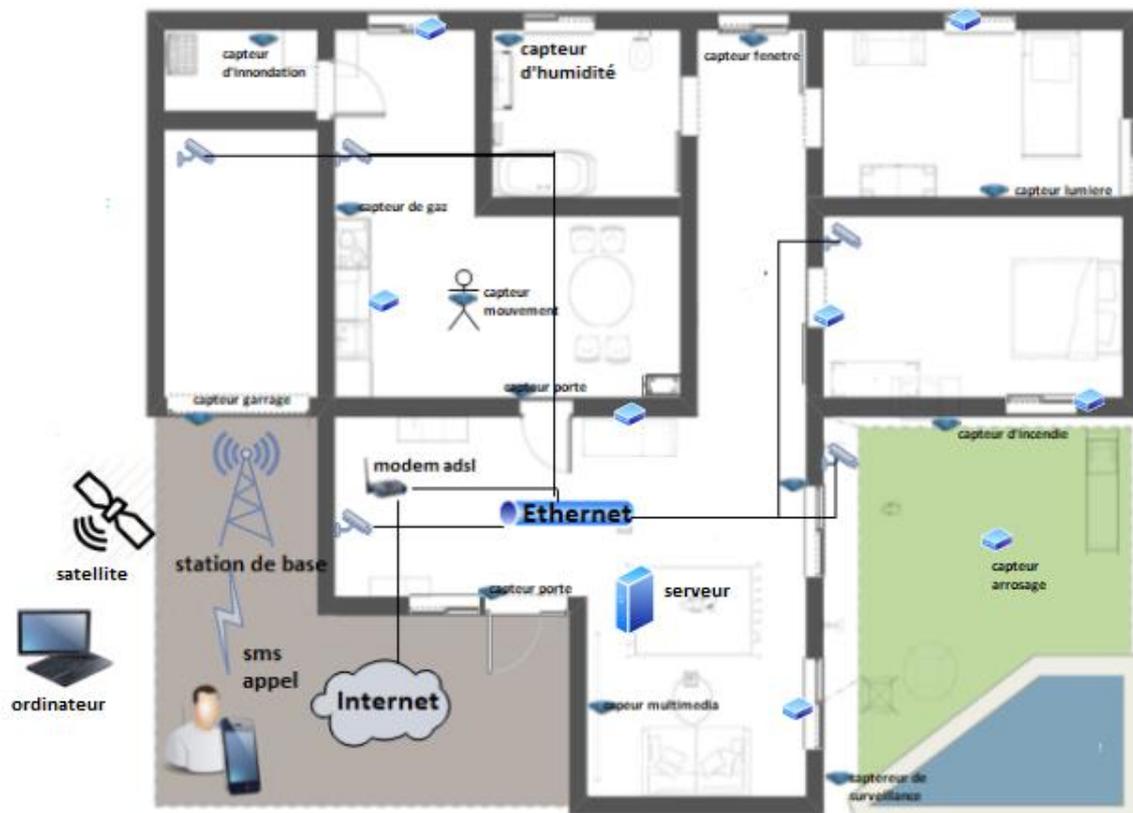


Figure II-2 Vue globale de l'application domotique.

II.4.2 Scenarios retenus

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

Les domaines d'application sont au cœur de la vie quotidienne. Les fonctions suivantes peuvent être réalisées grâce aux technologies intégrées dans la domotique (voir figure III-3)

:

- Sécurité ;
- Surveillance ;
- Gestion de l'énergie ;
- Scénarisation des actions ;
- Communication ;
- Confort.

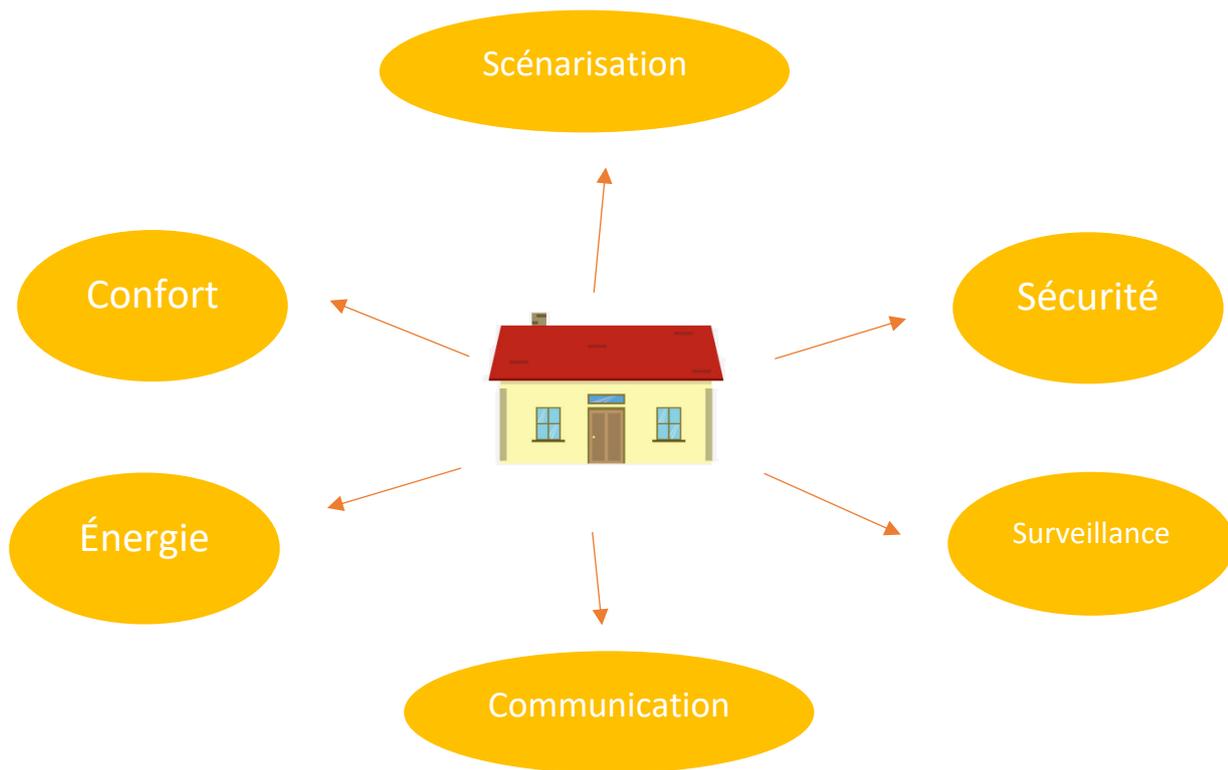


Figure II-3 Scénarios retenus.

II.4.2.1 La fonction de sécurité

Un des domaines d'application de la domotique est la sécurité des biens et des personnes par des systèmes d'alarme qui préviennent d'une part des risques techniques (pannes ou dysfonctionnements des appareils) et d'autre part des éventuelles intrusions dans la maison (cambriolage) en général on trouve :

Alarmes techniques : les alarmes techniques (voir Figure II-4) sont basées sur des capteurs capables de détecter différents incidents tels que des dégagements toxiques, incendie, fuite d'eau, fuite de gaz, etc. Ces différents capteurs sont raccordés à une centrale d'alarme. Les sécurités anti-noyade des piscines font également partie de ces systèmes d'alarme. Ainsi que certains détecteurs de pannes sur les équipements domestiques (chaudière par exemple).



Figure II-5 Alarme piscine [27].



Figure II-6 Détecteur de fumée [27].

Alarmes anti-intrusion : Ce sont en général des capteurs sur les portes (détection d'ouverture) ou dans les pièces (détection de présence) qui sont reliés eux aussi à une centrale d'alarme (voir Figure II-6). Ces capteurs peuvent être couplés avec un réseau de caméras numériques de surveillance. Lors d'une intrusion, un message d'alerte peut être envoyé par e-mail ou sur un téléphone portable [27].



Figure II-7 Alarme anti-intrusion.

II.4.2.2 La fonction de gestion d'énergie

La programmation des seuils de température est l'une des principales sources d'économie. Cette programmation peut être journalière ou hebdomadaire. Des sondes de température renseignent la domotique sur les valeurs de température dans chaque pièce. Dans un souci de confort un simple clic suffirait pour augmenter le chauffage et préparer l'habitat en prévision de l'arrivée des occupants. Ainsi les capteurs s'organisent d'une manière autonomes et assure :

- l'extinction des éclairages inutiles.
- le réglage de l'intensité lumineuse en fonction de l'activité, ...
- la régulation de l'éclairage et du chauffage.
- l'optimisation des ouvrants.
- l'ouverture ou la fermeture d'un volet selon l'ensoleillement...

Des moniteurs d'énergie peuvent contrôler à tout moment à distance votre consommation énergétique, ainsi que celle de vos appareils Électroménagers.

La domotique propose ainsi de réduire les consommations énergétiques en adaptant ces consommations aux modes de vie des occupants et à l'environnement extérieur.



Figure II-8 : Moniteur d'énergie.

II.4.2.3 La fonction de scénarisation

Au moment de quitter un habitat, les capteurs de mouvements, qui ne détectent plus aucun mouvement dans l'habitat déclenche une série de contrôles et d'actions :

- Extinction de toutes les lumières.
- Coupure de l'arrivée de gaz.
- Vérification de la fermeture de toutes les fenêtres.
- Allumage de la lumière extérieure durant quelques minutes s'il fait nuit.

II.4.2.4 La fonction de communication

Aujourd'hui, une centrale domotique sait communiquer via

- téléphone.
- interfaces tactiles.
- microphones permettant une activation par commande vocale associée à des logiciels de reconnaissance vocale.

Ceci permet à une personne de recevoir l'état de son installation et d'émettre des alertes et piloter sa maison à distance.

La centrale domotique offre aussi la gestion des multimédias, les équipements vidéo, home-cinéma, réseau téléphonique et internet sont intégrés dans toutes les pièces de l'habitat.

Il est possible de gérer et diffuser ses bibliothèques de musiques et de vidéos dans différentes pièces, de sauvegarder ses données informatiques, d'avoir accès à distance à ses ordinateurs, de faciliter la mobilité et le télétravail. Ces systèmes peuvent être pilotés via une interface tactile [28].

II.4.2.5 La fonction de surveillance

Les différents capteurs peuvent détecter différentes anomalies au sein de l'habitat

- Inondation,
- Incendie,
- Fuite de gaz (butane(CH₄), monoxyde de carbone (CO)...),
- Arrêt du congélateur,
- Coupure de courant,
- Vent ou pluie.

Lors d'une anomalie les capteurs interviennent instantanément pour couper les alimentations, remonter les stores, couvrir la piscine, appeler les numéros d'urgence.

La domotique trouve aussi de nouvelles applications dans le domaine de la santé. En installant des systèmes domotiques dans les maisons des personnes en situation d'handicap, atteintes de maladies neurodégénératives telles que la maladie d'Alzheimer ou encore des personnes âgées, il est possible de les aider dans leur quotidien en automatisant le plus possible des tâches considérées comme complexes.

II.4.2.6 Domotique pour le confort

La domotique permet d'améliorer le confort d'usage. Depuis l'application installée sur Smartphone, l'utilisateur final peut décider de l'heure d'ouverture des volets, de la température des pièces selon l'heure de la journée. Des capteurs installés détectent la présence des individus et peuvent ainsi donner le signal pour :

- allumer ou éteindre les lumières dans une pièce,
- activer la température optimale,
- Arroser le jardin
- Gérer les volets roulants

Les Gestions d'éclairage, chauffage, volets roulants, s'exécutent par une simple action d'une commande, toutes ces tâches sont simplifiées grâce à l'application.



Figure II-9 : LED ampoule sans fil.

II.5 Internet des objets

IOT est l'acronyme « d'internet of things ». Ou internet des objets en français, IOT est l'extension d'internet qui n'était qu'un module virtuel et qui n'interagissait pas avec le monde physique à des entités et des emplacements existants sur terre. Les données générées par ces entités (objets) sont échangées via internet afin d'être exploitées dans divers domaines : la santé, la domotique, l'agriculture.....etc. Les géants de l'informatique parlent de maisons connectées, de villes intelligentes, et de véhicules autonomes. [28]

II.5.1 Objets connectés

Un objet connecté est un objet électronique relié à Internet et capable de communiquer des informations, apportant ainsi un service ou une valeur ajoutée. Le premier objet connecté était la lampe DAL (Figure II-8) [28], sensible au toucher et au bruit, cette lampe communiquait des informations sur la météo, la pollution...les fonctions proposées aujourd'hui vont beaucoup plus loin que la simple annonce de météo. Prenons exemple sur le thermostat Qivivo (Figure II-9), qui permet non seulement de piloter son chauffage à distance, mais également d'obtenir un diagnostic de sa consommation d'énergie, des conseils d'optimisation et même d'être mis en relation avec des professionnels afin d'étudier les possibilités d'amélioration de son logement.

Les usages ont donc su se développer et, aujourd’hui, les objets connectés sont partout, ils ont particulièrement appropriés dans certains domaines, tel que la domotique [28].



Figure II-11 : La lampe DAL (premier objet connecté).



Figure II-10 : Thermostat Qivivo.

II.6 Plateforme et outils pour l’Internet de objets

Préambule

Une plateforme c’est un ensemble d’outils logiciels ou systèmes d’exploitation destinés au stockage et au partage de contenus virtuels. Nous allons exposer dans ce qui suit un ensemble de plateformes et outils open source destiné pour l’internet des objets. Ces dernières offrent la possibilité de contact, transfert de données et de synchronisation des objets internet.

II.6.1 Zetta

Il s’agit d’une plateforme basée sur une API basée sur Node.js. Destinée à la création de serveurs pour l’Internet des objets qui peuvent tourner sur des ordinateurs décentralisés ou sur le Cloud. Zetta combine API REST, WebSockets qui sont parfaits pour connecter plusieurs appareils avec des applications en temps réel.

Cette plateforme a de nombreux avantages :

- Il peut fonctionner sur le cloud, sur un PC ou même sur des cartes de développement modestes.
- Interface facile et programmation nécessaire pour contrôler les capteurs, les actionneurs et les contrôleurs.

- Permet aux développeurs d'assembler des applications pour smartphone, des applications pour appareil et des applications cloud.
- Il est développé pour les applications en temps réel et gourmandes en données.
- Transforme n'importe quelle machine en API



II.6.2 Arduino

Arduino une plateforme basée sur une interface entrée/sortie (analogique et numérique) qui permet de traiter l'information provenant des capteurs pour commander des actionneurs et peut être utilisée pour construire des objets interactifs indépendants, ou bien peut être connectée à un ordinateur pour communiquer avec des logiciels. C'est une technologie qui fait associer un environnement de développement avec un circuit électronique à base d'un microcontrôleur AVR, distribué sous la licence du matériel libre (les schémas électriques sont disponibles gratuitement, voir Annexe C). La carte électronique peut être programmée pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le contrôle des appareils domestiques ou le pilotage d'un robot, etc.

(Pour plus de détails voir chapitre 3)



Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique, les principaux avantages de l'électronique programmée sont :

- Simplifie grandement les schémas électroniques.
- Diminue le coût de la réalisation.
- La charge de travail à la conception d'une carte électronique.
- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Macintosh et Linux.
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.
- Logiciel et matériel open source et extensible.
- Nombreux conseils, tutoriaux et exemples en ligne (forums, site perso, etc.).
- Existence de « shield » (boucliers en français).

II.6.3 ChosesBoard

Il est destiné à la collecte, au traitement, à la visualisation et à la gestion des appareils. Il respecte tous les protocoles IoT standard tels que CoAP, MQTT et HTTP aussi rapidement que les déploiements cloud et sur site. Il crée des flux de travail basés sur les événements du cycle de vie de conception, les événements d'API REST et les demandes RPC.

Il a plusieurs avantages notamment :

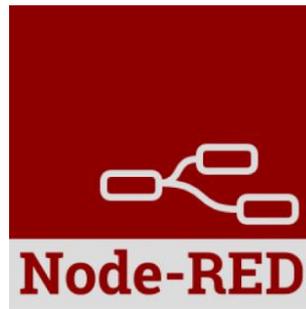
- Une plateforme stable qui allie évolutivité, production et tolérance aux pannes.
- Contrôle facile de tous les appareils connectés dans un système exceptionnellement sécurisé
- Transforme et normalise les entrées de l'appareil et facilite les alarmes pour générer des alertes sur tous les événements de télémétrie, les restaurations et l'inactivité.
- Gère des millions d'appareils en même temps.
- Aucun moment de défaillance unique, car chaque nœud du bundle est exact.



II.6.4 Node-red

Node-RED est un outil de programmation graphique pour modéliser les applications IoT sous forme de flux de données. Il permet l'interconnexion des objets physiques, les API, les environnements Cloud et des services web. Il fournit un éditeur basé sur un navigateur qui facilite le câblage des flux à l'aide de la large gamme de nœuds de la palette qui peuvent être déployés dans son environnement d'exécution en un seul clic.

- La communication bidirectionnelle prend en charge les transmissions depuis/vers les capteurs lors du déclenchement de l'évènement défini par l'utilisateur
- assure une flexibilité, une couverture des zones aveugles, et une convergence des nœuds dans le réseau
- environnement de travail simple d'utilisation



II.6.5 Plateforme Kaa IoT

Il s'agit d'une plate-forme middleware polyvalente, flexible et prête à la production pour la création de solutions IoT de bout en bout, d'applications connectées et d'appareils intelligents. Il donne un moyen complet d'effectuer une communication efficace, traite et interopère des capacités dans les appareils connectés et intelligents.

Il monte de petites startups à une grande entreprise et contient des modèles de déploiement avancés pour les solutions IoT multi-cloud. Il est principalement basé sur des microservices flexibles et se conforme facilement à pratiquement tous les besoins et applications, il a également d'autres fonctionnalités telles que :

- Facilite l'interopérabilité entre les appareils.
- Exécute le contrôle des périphériques en temps réel, l'approvisionnement et la structure des périphériques distants.
- Crée des services cloud pour des produits intelligents
- Gère une quantité infinie d'appareils connectés
- Collecte et analyse les données des capteurs



II.6.6 DSA

L'acronyme DSA signifie architecture de services distribués qui est destiné à la mise en œuvre de la communication, de la logique et des efforts inter-appareils à chaque étape de l'infrastructure IoT. Il permet la coopération entre les appareils de manière distribuée et met en place un ingénieur réseau pour partager les fonctionnalités entre des systèmes informatiques discrets.

DSA fournit de nombreux avantages dont :

- L'accès au réseau auto-organisé assure la réparation du réseau.
- En termes d'énergie, la durée de vie de la batterie peut être prolongée de 5 à 10 ans.
- C'est une version plusieurs protocoles, elle répond aux différents besoins et scénarios d'utilisateurs.

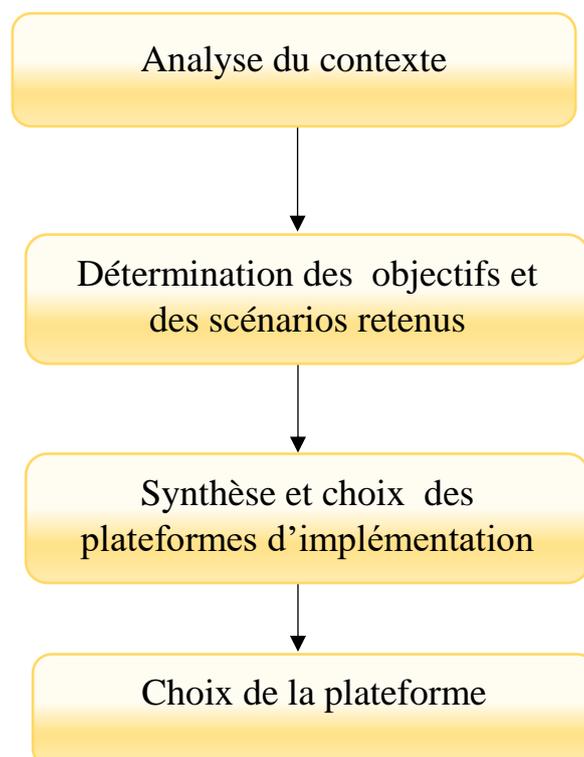


II.7 Description du projet

La réalisation de ce projet se divise en deux parties, la partie soft et partie hard. Nous allons détailler ces deux parties dans le chapitre 3.

II.7.1 Diagramme prévisionnel d'avancement de travail

Durant l'élaboration de ce projet de fin d'études, nous avons conçu un diagramme prévisionnel pour une meilleure organisation et suivi du travail, synthétisant la démarche suivie afin d'aboutir à la réalisation du projet (Figure II-11).



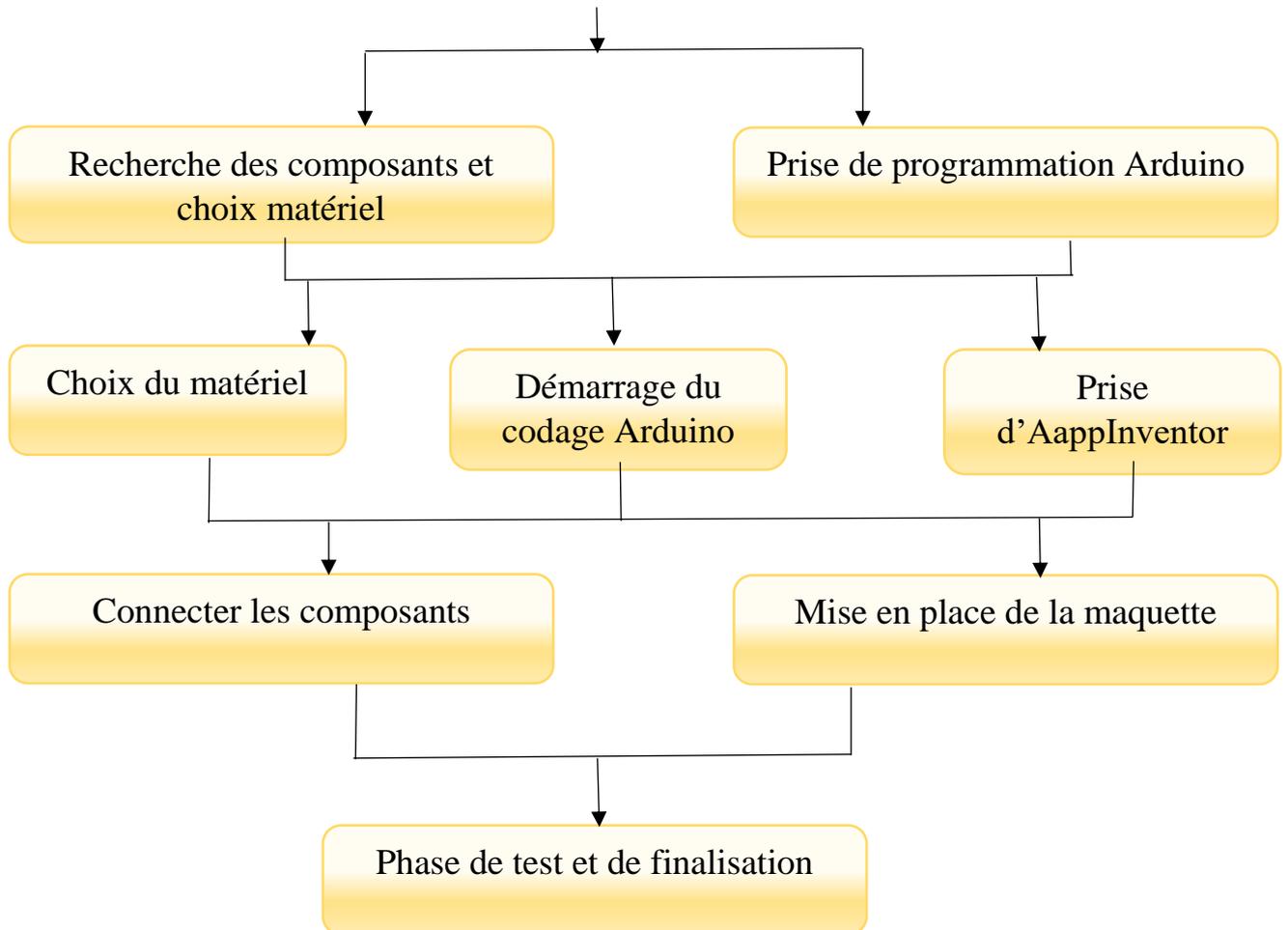
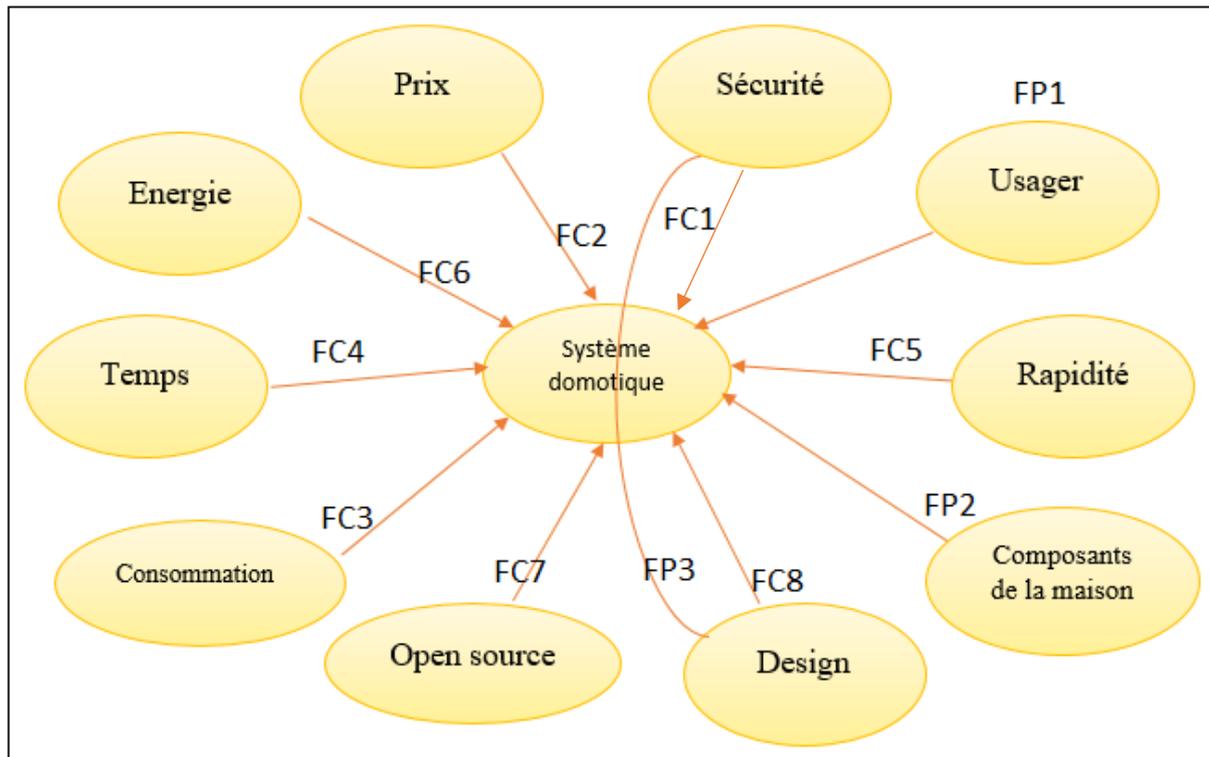


Figure II-12 : diagramme prévisionnel d'état d'avancement

II.7.2 Analyse fonctionnelle

Pour notre cas, l'analyse fonctionnelle étudie donc le système domotique sous son aspect fonctionnel. L'outil que nous avons choisi pour faire l'analyse fonctionnelle c'est le diagramme de pieuvre nommé aussi diagramme d'interaction, ce diagramme permet de définir le lien entre le système et son environnement (figure II-12).



FC : fonctions complémentaires, FP : fonctions principales.

Figure II-13 : Diagramme de pieuvre.

Les fonctions principales:

FP1 : le système permet de commander un ensemble d'éléments dans la maison.

FP2 : le système informe l'utilisateur de l'état des E/S de système.

FP3 : le système permet de faire des tâches automatiquement.

FC1 : il doit répondre aux mesures de sécurité.

FC2 : il doit respecter le budget.

FC3 : il doit consommer moins d'énergie.

FC4 : il doit respecter le délai.

FC5 : la réponse de système doit être rapide et en temps réel.

FC6 : il doit être alimenté par de l'énergie.

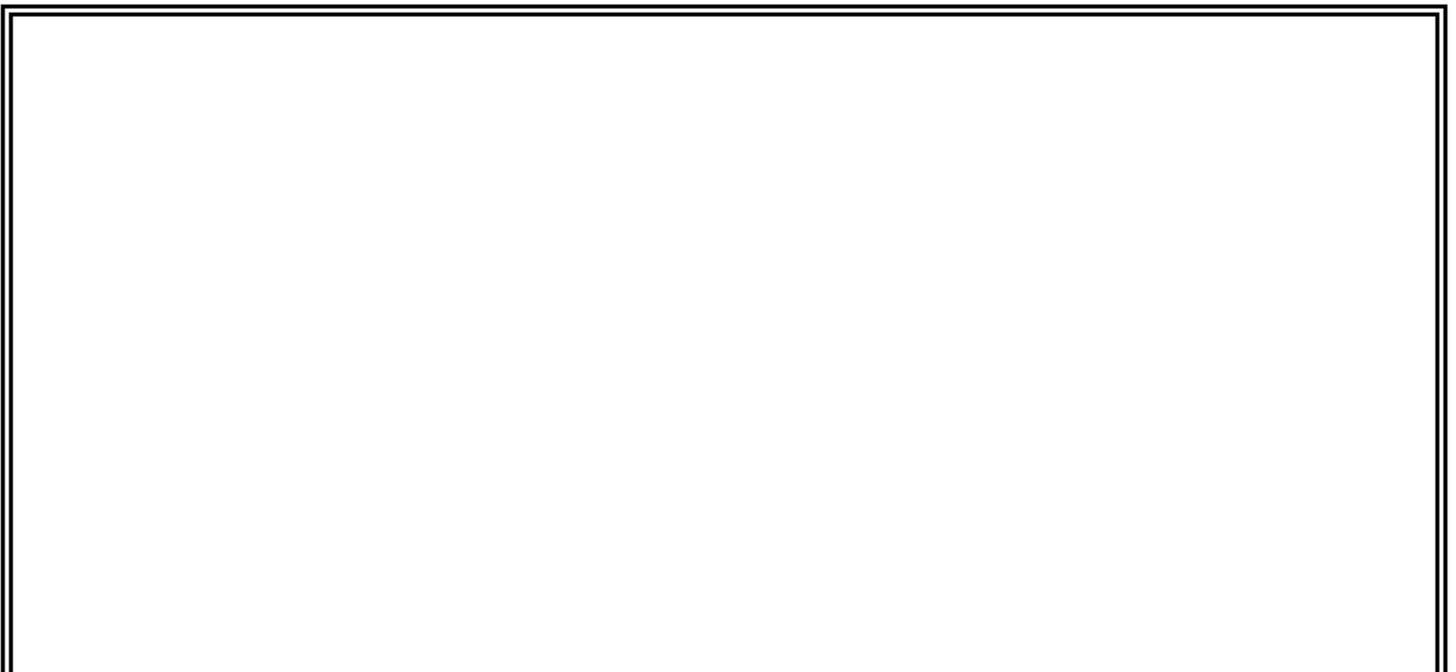
FC7 : pouvoir être fabriqué à partir de matériaux open source.

FC8 : l'interface doit être simple, légère et agréable.

II.8 Conclusion

Dans ce chapitre, nous avons présenté la démarche suivie qui a permis de concevoir ce réseau capteur sans fil applicable à la domotique. Nous avons exposé les différentes

caractéristiques et exigences de l'application, Nous avons présenté une vue globale de la technologie de la domotique automatisée et intelligente et les différentes technologies disponibles pour implémenter les objets internet. Dans le chapitre qui suivra nous allons présenter le matériel utilisé ainsi que la démarche suivie pour la réalisation de notre projet.



Chapitre III :

Conception et Réalisation

III Chapitre

III.1 Introduction

Après avoir comparé et analysé les différentes plateformes conçues pour implémenter la technologie des IOT nous avons choisi la plateforme Arduino pour des raisons de simplicité de réalisation hardware et software. Dans ce chapitre, nous expliquerons les différents capteurs et composants qui seront utilisés ainsi que leur mode de fonctionnement, ensuite nous procéderons au branchement qui sera placé dans la maquette et enfin nous détaillerons le développement de l'application Android responsable de la gestion des fonctions intelligentes de l'habitat.

III.2 Partie hardware

III.2.1 Présentation des cartes Arduino

Il existe plusieurs éléments qui persistent dans la conception d'un réseau domotique, ces éléments appelés objets connectés peuvent être combinés avec un seul élément central responsable de la décision et de la commande de ces objets. Ce dernier est la carte Arduino dotée d'un module Ethernet, qui permet à son tour d'accéder à notre système Arduino à distance. Arduino est une carte électronique dans laquelle se trouve un microcontrôleur pouvant programmer grâce au langage Arduino IDE pour analyser et produire des signaux électriques de manière assez simple afin d'effectuer diverses tâches [29]. Elle est basée sur un microcontrôleur de la compagnie ATMEL, c'est un outil qui va permettre la création des systèmes électroniques plus ou moins complexes, elle est programmée en langage Arduino C. il existe plusieurs cartes Arduino, la différence consiste en leurs taille, puissance de microcontrôleurs et consommation. .

III.2.1.1 La carte LEONARDO

La carte Arduino LEONARDO est basée sur un microcontrôleur ATmega32u4 cadencé à 16 MHz permettant la gestion du port USB par un seul processeur Avec 32 ko de mémoire flash, 2,5 ko de mémoire SRAM, 1 ko de mémoire EEPROM et 20 broches d'entrées/sorties dont 7 sont des sorties PWM ainsi que 12 entrées analogiques. C'est la carte

qui est prévue de succéder à la carte Arduino UNO en présentant des caractéristiques équivalentes mais avec une ergonomie revue et une stabilité plus éprouvée. [30]

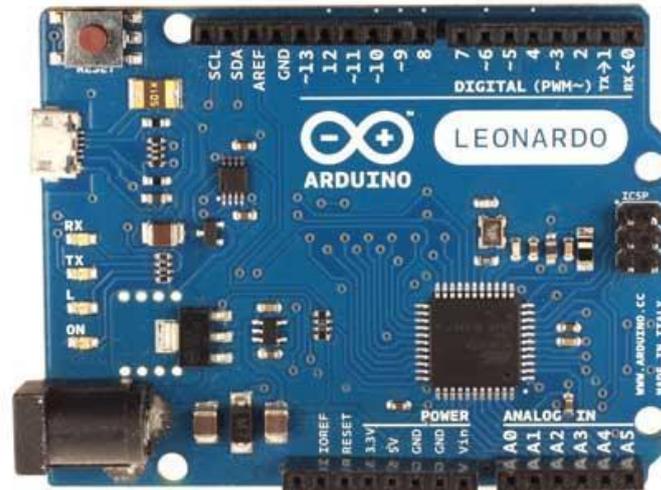


Figure III-1 : carte Arduino LEONARDO.

III.2.1.2 La carte NANO

C'est l'une des plus petites cartes de microcontrôleur de la gamme Arduino. Sa taille et son poids réduits lui permettent d'être utilisée dans des espaces limités. Elle possède un microcontrôleur ATmega328 à 16 MHz, comprend 32 Ko de mémoire flash, 1 Ko d'EEPROM, 2Ko de RAM, 14 entrées /sorties numériques, 6 entrées analogiques et une tension de fonctionnement de 5V. [31]

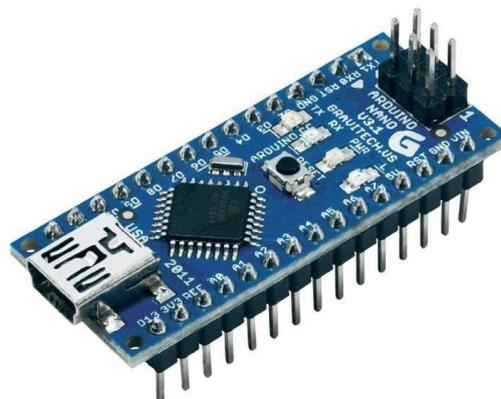


Figure III-2 : carte Arduino NANO

III.2.1.3 La carte UNO

Il s'agit du modèle le plus répandu. Livré avec une puce ATmega328 qui convient aux tâches relativement simples en raison de sa mémoire flash et SRAM et également un nombre d'entrées/sorties limité qui est 14 entrées/sorties numériques dont 6 PWN et 6 entrées analogiques. L'un des avantages de ce microcontrôleur, est sa facilité d'utilisation, la carte se connecte facilement à un ordinateur via un câble USB fourni. Elle fonctionne à une fréquence de 16 MHz, elle comprend 32 Ko de mémoire programme, 2 Ko de RAM, 1 Ko d'EEPROM et un rail d'alimentation d 5V et 3,3 V. [31] [32]



Figure III-3 : carte Arduino UNO.

III.2.1.4 La carte MEGA 2560

La carte MEGA2560 est la plus utilisée après l'UNO, grâce au nombre important d'entrées/sorties qu'elle propose, estimées à 54 broches numériques dont 14 peuvent être utilisées comme sorties PWM et 16 entrées analogiques, elle offre une configuration plus complète que le modèle UNO et permet une importante marge de manœuvres et de montages plus poussés. Et sa connexion à l'ordinateur via port USB, permet une programmation complète et facile par l'interface du logiciel. Elle fonctionne grâce à un microcontrôleur ATmega2560, avec une fréquence d'horloge de 16 mhz et une tension de 5 V, sa tension d'entrée recommandée est comprise entre 7 et 12V et la tension limite est entre 6V et 20V. Dotée d'une mémoire flash de 256 Ko pour stocker les codes dont 4 Ko sont utilisés par

bootloader, d'une SRAM de 8 Ko ce qui la rend relativement rapide et une EEPROM de 4 Ko. Toutes ces caractéristiques lui permettent d'exploiter des algorithmes plus complexes. [29] [31] [32].

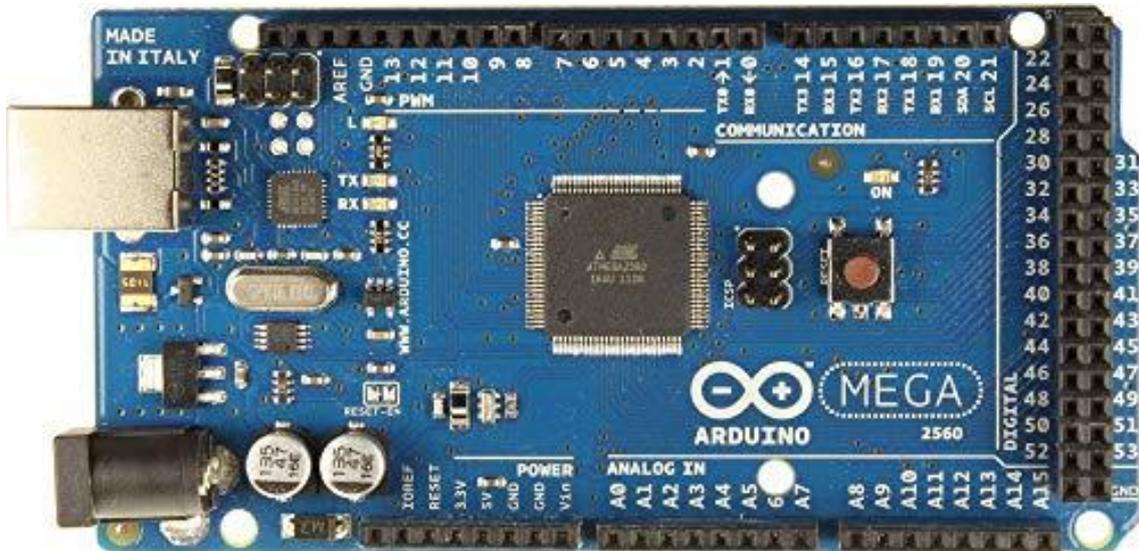


Figure III-4 : carte Arduino MEGA 2560.

III.2.1.5 Choix de la carte de développement

Parmi toutes ces cartes, nous avons choisis la carte Arduino UNO pour de nombreuses raisons.

Critères de choix de la carte de développement

Le choix de la carte Arduino UNO n'est pas fait au hasard, mais il a été pour les raisons suivantes :

- Une carte électronique programmable et un logiciel gratuit.
- logiciel et matériel Open source.
- Environnement de programmation simple et clair.
- Un prix si réduit étant donné l'étendue des applications possibles.
- Une compatibilité sur toutes les plateformes (linux, windows, ...).
- Il s'agit d'une carte très bien documentée.
- Une infinité d'applications possibles.

III.2.2 Plateforme de programmation Arduino

L'interface de l'IDE Arduino est plutôt simple, elle offre cinq zones, chacune de ces zones a une fonctionnalité bien précise.

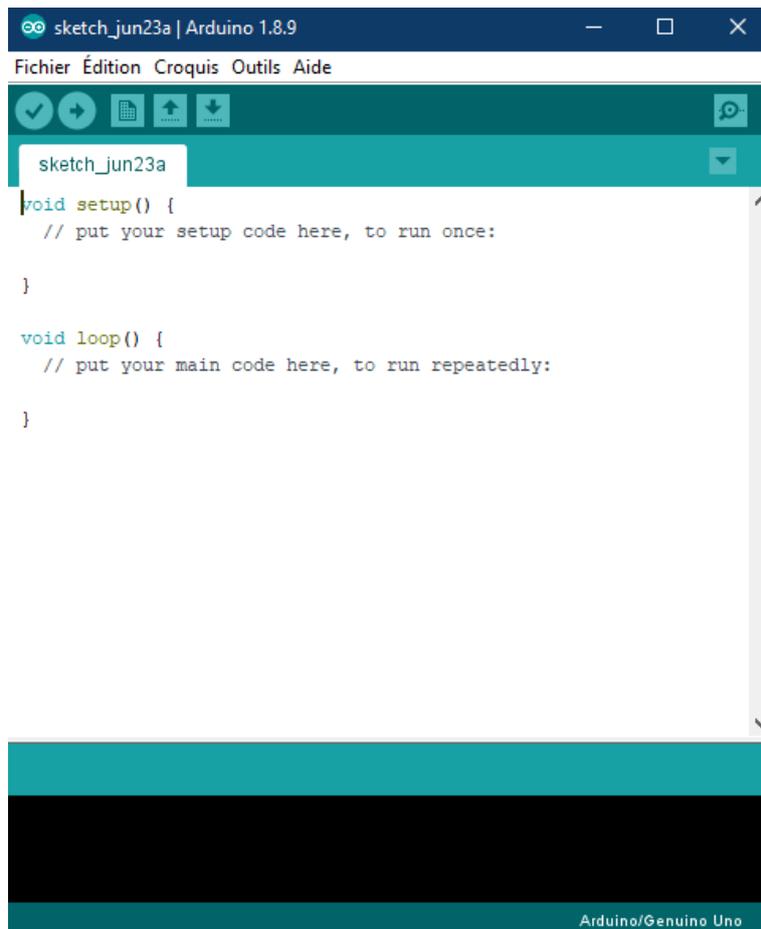


Figure III-5 : Composants de l'écran principal de l'IDE Arduino.

- Une barre de menus.
- Une barre d'icônes : la série des icônes sous la barre de menu permet de réaliser diverses actions (vérifier le code, televerser le code, ouvrir un nouveau projet, ouvrir un projet de programme Arduino existant, sauvegarder le projet, ouvrir le moniteur série pour communiquer avec l'Arduino)
- Une barre d'onglets : Chaque fichier du projet est présenté dans cet IDE par un onglet.

- zone d'édition : l'endroit où le code source doit être saisi, la structure générale d'un sketch Arduino est représentée par l'organigramme de la figure III-6.
- Une zone d'information et de statut : cette zone affiche diverses informations sur le programme, les erreurs de compilation, le transfert de programme ...

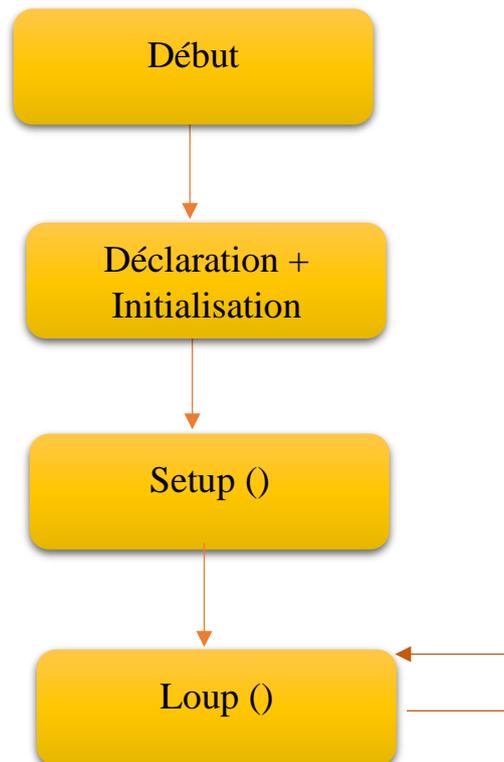


Figure III-6 : Structure générale d'un sketch Arduino.

III.2.3 Choix de matériel

III.2.3.1 Les capteurs DHT11 et DHT22 :

Ces capteurs sont bifonctionnels car ils peuvent mesurer le degré d'humidité ainsi que la température d'un environnement bien défini. Ce sont des capteurs basiques à faible coût et assez simple à utiliser. La différence entre ces deux capteurs est la capacité de mesurer selon l'écart de température, le DHT22 est bien plus précis, et accepte une plus grande plage de température.

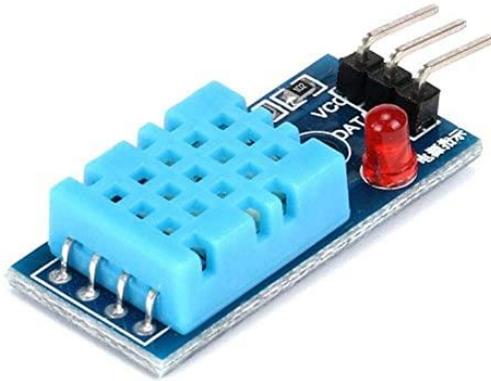


Figure III-7 : Capteurs DHT11

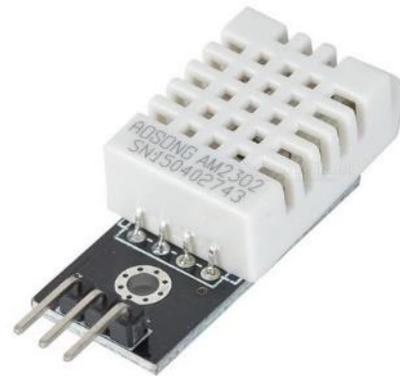


Figure III-8 : Capteurs DHT22

Le capteur DHT22 est capable de mesurer des températures de -40 à $+125^{\circ}\text{C}$ avec une précision de $\pm 0.5^{\circ}\text{C}$ et des taux d'humidité relative de 0 à 100% avec une précision de $\pm 2\%$. Une mesure peut être réalisée toutes les 500 millisecondes (soit deux fois par seconde).

Le capteur DHT11 est lui capable de mesurer des températures de 0 à $+50^{\circ}\text{C}$ avec une précision de $\pm 2^{\circ}\text{C}$ et des taux d'humidité relative de 20 à 80% avec une précision de $\pm 5\%$. Une mesure peut être réalisée toutes les secondes.

Le DHT22 et le DHT11 sont tous les deux compatibles 3.3 volts et 5 volts (le fabricant recommande cependant de toujours alimenter le capteur en 5 volts pour avoir des mesures précises). Ils ont aussi le même câblage et le même protocole de communication.

Pour résumer, voici les caractéristiques des deux capteurs sous forme de tableau comparatif :

	DHT22	DHT11
Humidité (relative %)	$0 \sim 100 \%$	$20 \sim 80\%$
Précision (humidité)	$\pm 2\%$ ($\pm 5\%$ aux extrêmes)	$\pm 5\%$
Température	$-40 \sim +150^{\circ}\text{C}$	$0 \sim +50^{\circ}\text{C}$
Précision (température)	$\pm 0.5^{\circ}\text{C}$	$\pm 2^{\circ}\text{C}$
Fréquence mesure max	2Hz (2 mesures par seconde)	1Hz (1 mesure par seconde)
Tension d'alimentation	$3 \sim 5$ volts	$3 \sim 5$ volts

Tableau III-1 : tableau comparatif entre le DHT22 et le DHT11.

III.2.3.2 Câblage d'un capteur DHTxx

Le câblage des capteurs DHT22 et DHT11 est le même. Les capteurs DHTxx communiquent avec le microcontrôleur via une unique broche d'entrée / sortie. Le brochage du capteur est le suivant :

- La broche n°1 est la broche d'alimentation (5 volts ou 3.3 volts).
- La broche n°2 est la broche de communication. Celle-ci doit impérativement être reliée à l'alimentation via une résistance de tirage de 4.7K ohms.
- La broche n°4 est la masse du capteur (GND).

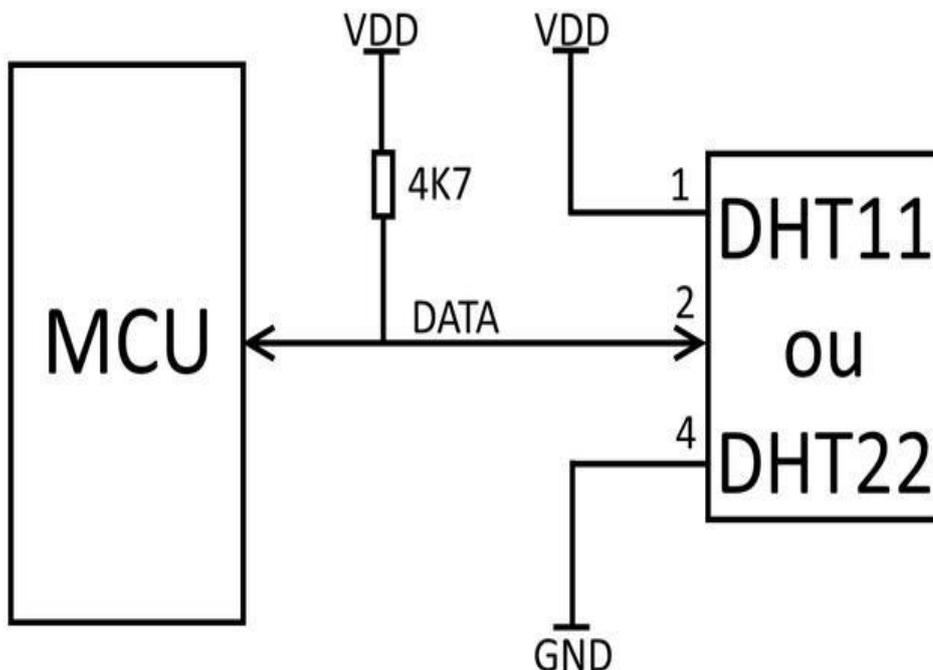


Figure III-9 : Diagramme de câblage.

III.2.3.2.1 Le protocole de communication

Comme précisé précédemment, les capteurs DHTxx ont la particularité de communiquer avec le microcontrôleur via une unique broche d'entrée / sortie. Il s'agit simplement d'un protocole de communication utilisant un seul fil et nécessitant des timings très précis.

Le déroulement de communication :

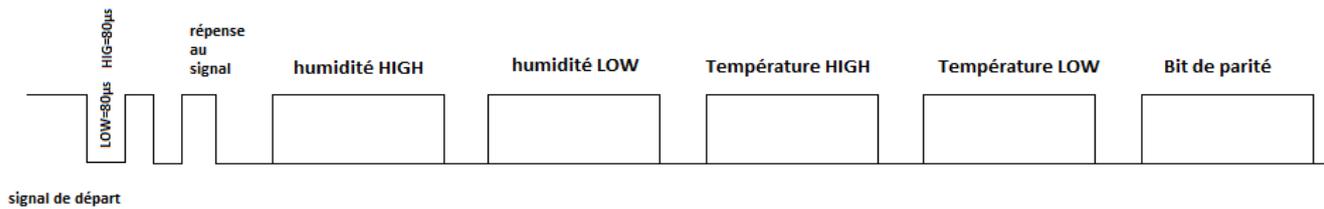


Figure III-10 : déroulement du protocole communication.

III.2.3.2.2 Format d'une trame de communication

	Protocole bus simple
Signal du début	Le microprocesseur de la carte Arduino réduit la ligne de donnée pendant 800 µs et suscite le capteur à préparer les données.
Reponse au signal	Le capteur de donnée réduit le HIGH à 80 µs, puis le LOW à 80 µs pour recevoir le signal de départ.
Format de donnée	Quand le signal de départ est reçu, le capteur va transmettre les données par une suite de 40 bits
Humidité	La valeur d'humidité est codée sur 16 bits
température	La valeur de la température est codé sur 16 bits dont (15 bits pour la HIGH/ LOW température) le bit numéro 16 indique la nature de la température, si le bit=1 il indique que la valeur est négative si le bit=0 il indique que la valeur est positive.
Bit de parité	Bit de parité = humidité HIGH+ humidité LOW+ température HIGH+ température LOW

Tableau III-2 : format d'une trame de communication.

III.2.3.2.3Détails du format d'une trame de communication

La communication avec un capteur DHTxx se fait en 3 étapes :

- Tout d'abord, le microcontrôleur maître (la carte Arduino dans notre cas) réveille le capteur en plaçant la ligne de données à LOW pendant au moins 800µs (au moins 18ms pour le DHT11). Durant ce laps de temps, le capteur va se réveiller et préparer

une mesure de température et d'humidité. Une fois le temps écoulé, le maître va libérer la ligne de données et passer en écoute.

- Une fois la ligne de données libérée, le capteur répond au maître (pour montrer qu'il est bien réveillé) en maintenant la ligne de données à LOW pendant 80 μ s puis à HIGH pendant 80 μ s.
- Le capteur va ensuite transmettre une série de 40 bits (5 octets). Les deux premiers octets contiennent la mesure de l'humidité. Les deux octets suivants contiennent la mesure de la température et le cinquième octet contient une somme de contrôle qui permet de vérifier que les données lues sont correctes.

III.2.3.3 Capteur de Gaz MQ2

Le capteur de gaz MQ2 est utilisé pour la détection des fuites de gaz pour les équipements des marchés de grandes consommations et industriels. Ce capteur est conçu pour détecter le LPG, i-butane, propane, méthane, alcool, hydrogène et la fumée. Il est conçu pour un usage intérieur à température ambiante. Il doit être alimenté en 5 V pour que le capteur physico-chimique puisse atteindre sa température de fonctionnement. Il a une grande sensibilité et un temps de réponse rapide. Il dispose d'une sortie analogique et d'un réglage de la sensibilité par potentiomètre. [19]



Figure III-10 : Image réelle du capteur de gaz/fumée MQ-2.

Caractéristiques :

- Alimentation : 5V.
- Type d'interface : Analogique.
- Connectique : 1-sortie 2-GND 3-VCC.
- Dimension 40*20 mm

III.2.3.4 Servomoteur

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à 3 fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de position sous forme d'un signal codé en largeur d'impulsion plus communément appelés PWM (Pulse Width Modulation ou Modulation de Largeur d'Impulsion) ou RCO (Rapport Cyclique d'Ouverture). Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.



Figure III-11 : Image réelle du servomoteur.

Caractéristiques :

- Alimentation : 4,8 à 6V
- Angle de rotation : 180°
- Couple : 1,3 kg.cm
- Vitesse : 0,12 sec/ 60°
- Dimensions : 23,2 X 12,5 X 22 mm
- Poids : 9g

III.2.3.5 Les LEDs

Une diode électroluminescente est un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique. Ces LED feront office d'éclairage dans notre maquette et serviront d'indicateur lumineux pour nos circuits.



Figure III-12 : Image réelle d'une LED.

III.2.3.6 Capteur flamme

Le capteur est utilisé pour détecter le feu et permet de mesurer des longueurs d'ondes sur une plage, ce capteur réagit en présence d'une flamme.

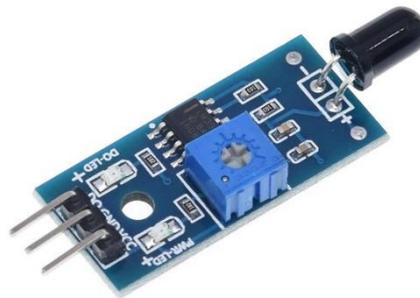


Figure III-13 : Image réelle du capteur flamme.

Caractéristiques :

- haute sensibilité du récepteur IR
- extrêmement sensibles aux ondes entre 760-1100nm
- voyant d'alimentation
- voyant comparateur de sortie

- sortie analogique quantité
- seuil de renversement du niveau de sortie électrique
- seuil réglé par potentiomètre
- Gamme de détection d'angle: environ 60 degrés
- Alimentation: 0-15 V DC.

III.2.3.7 LDR (Photosensible)

LDR (résistance variant à la lumière, ou photorésistance) agit comme une résistance variable, dont la résistance varie en fonction de la lumière perçue. Le module capteur LDR génère une sortie 5V dans l'obscurité et 0V en pleine lumière, La sensibilité du LDR peut être ajustée via un potentiomètre ajustable placé sur le module [32].

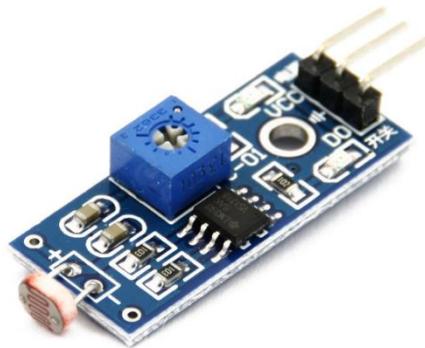


Figure III-14: image réelle du LDR.

Caractéristiques :

- tension de Fonctionnement: 3.5 V, 5 V;
- forme de sortie: numérique de commutation spectacle (0 et 1);
- trou de vis fixe, installation facile;
- petit PCB Taille: 3.2 cm * 1.7 cm;
- la sortie du comparateur, le signal propre, bonne vague, la capacité de conduire est forte, pour plus que 15 mA;
- avec réglable potentiomètre pour régler la luminosité de la lumière.

III.2.3.8 Module de connexion Wifi ESP8266

Les cartes wifi basées sur le microcontrôleur ESP8266 sont programmables comme les cartes Arduino et peuvent communiquer par wifi avec d'autres appareils (ordinateurs, Smartphones, etc.). Il existe plusieurs modèles : l'ESP-D1, l'ESP-03, l'ESP-12 ... etc. L'ESP-MCU, le modèle que nous allons utiliser dans notre travail est présenté dans la figure III-17.

Le module Wifi ESP8266 est un réseau autonome avec une pile de protocoles TCP/IP intégrée qui peut donner accès au réseau Wifi. L'ESP8266 est capable d'héberger une application ou de décharger toutes les fonctions de réseau Wifi d'un autre processeur d'application. Chaque module ESP8266 est préprogrammé avec un microprogramme de configuration de commande AT (voir l'**annexe B** pour la liste des commandes de l'ESP8266), ce qui signifie que vous pouvez simplement le brancher sur votre appareil Arduino et obtenir autant de fonctionnalités Wifi. Le module ESP8266 est un tableau extrêmement rentable avec un ensemble de commandes. [30]



Figure III-15 : Image réelle du module Wifi ESP8266 – ESP-MCU.

Mise sous tension

- Alimenter le module en reliant Vcc et Rx Tx, et câbler GND;
- La LED rouge doit être allumée et le rester;
- La LED bleue peut clignoter au démarrage;
- Vérifiez si votre module ne présente pas son réseau dans la liste des réseaux WIFI (par exemple un nouveau réseau ESP_99b22. [30]

Caractéristiques d'ESP8266

- Norme sans fil: IEEE 802.11.
- Gamme de fréquence: 2.4 GHz ~ 2.5 GHz.
- Tension de fonctionnement: 3.3v ou 5v.
- Courant de fonctionnement moyenne 80MA.

III.3 Partie software

Dans cette section, nous décrivons les organigrammes de programmation pour représenter graphiquement l'enchaînement des opérations et de décisions effectuées par les différents scénarios de notre programme.

III.3.1 Organigramme de la partie commande

Pour envoyer des commandes au système pour qu'il fait une certaine action (fermeture de la porte, ouverture d'un volet ...), le programme exécuté dans l'Arduino fonctionne de la même manière pour n'importe quel organe commandé.

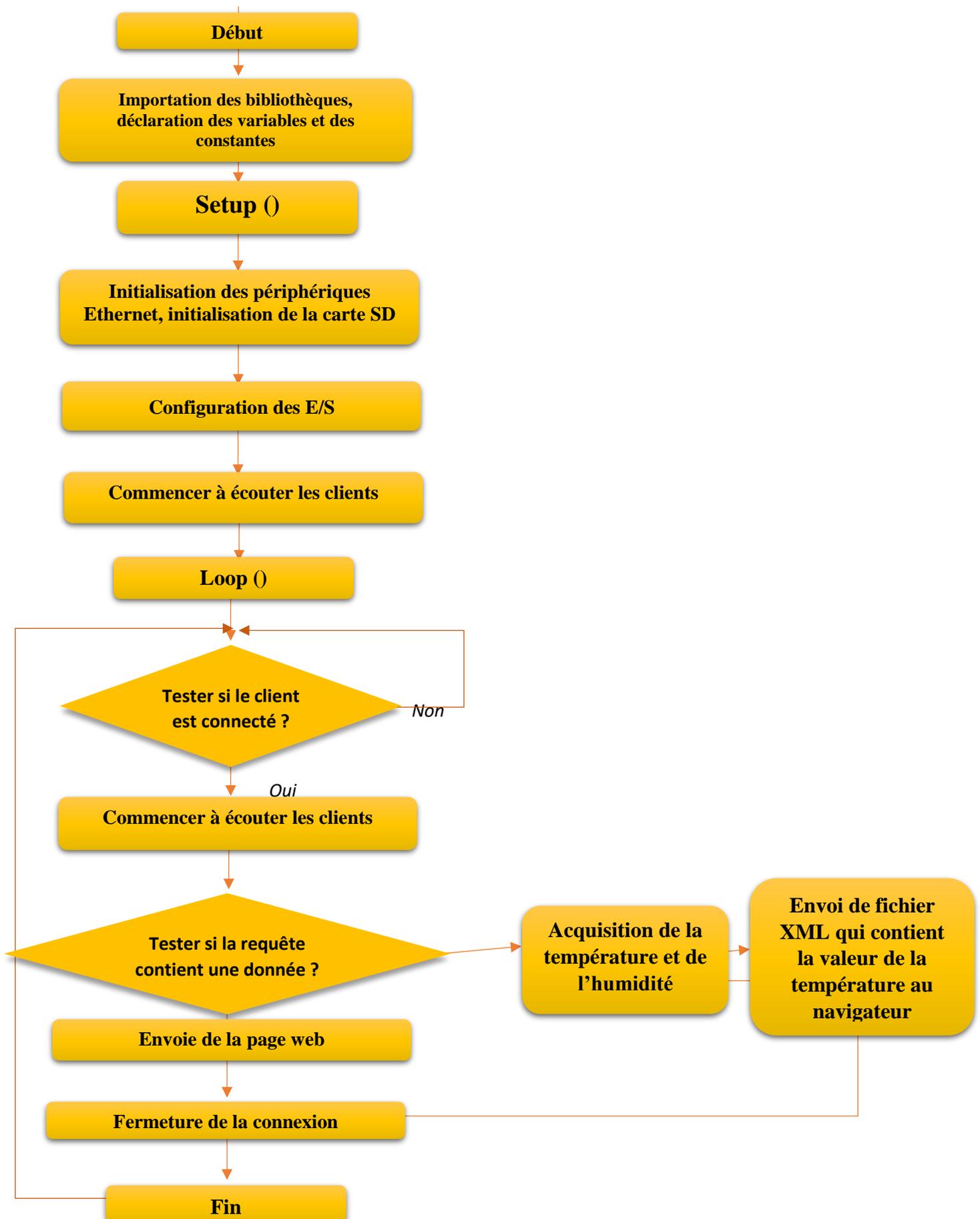


Figure III-16 : organigramme de fonctionnement global de la commande à distance.

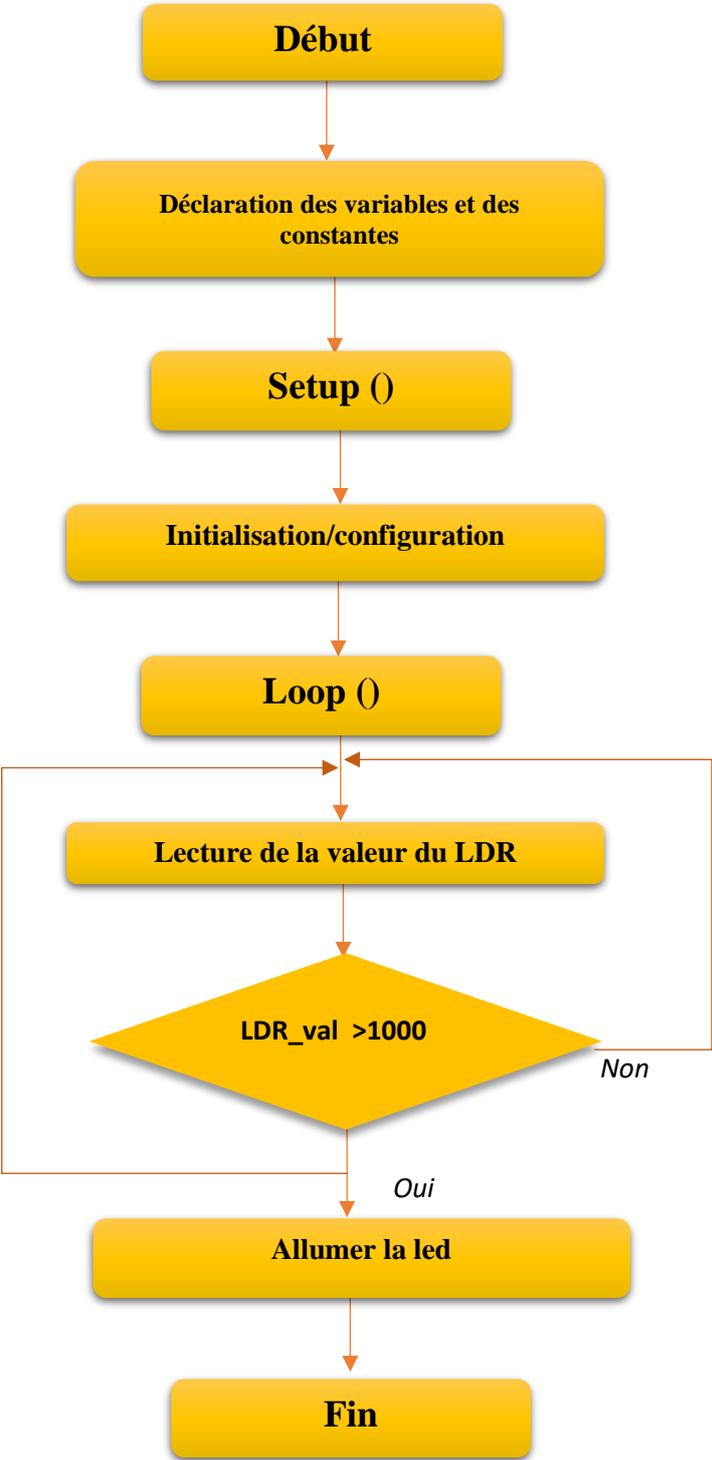


Figure III-17: Organigramme de la fonction de détection de fuite de gaz ou d’incendie.

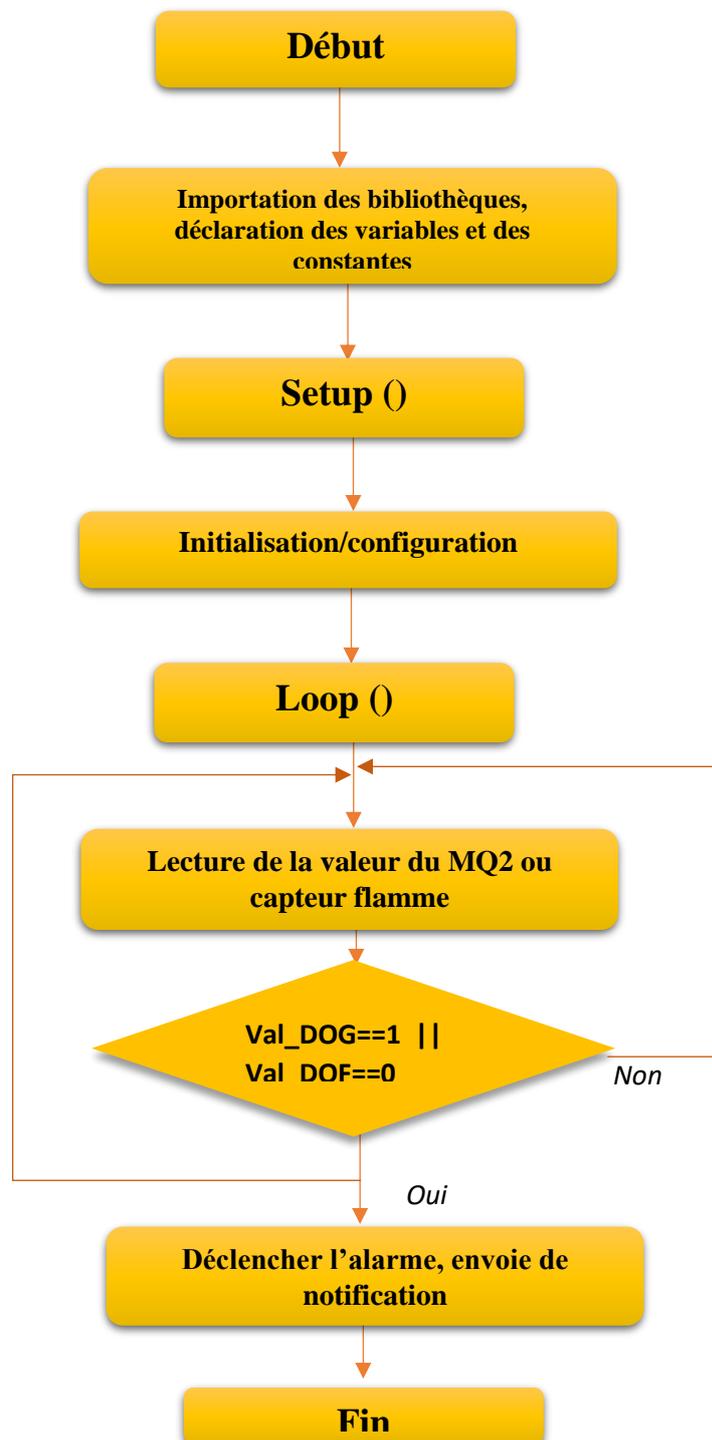


Figure III-18 : Organigramme de la fonction d'allumage automatique.

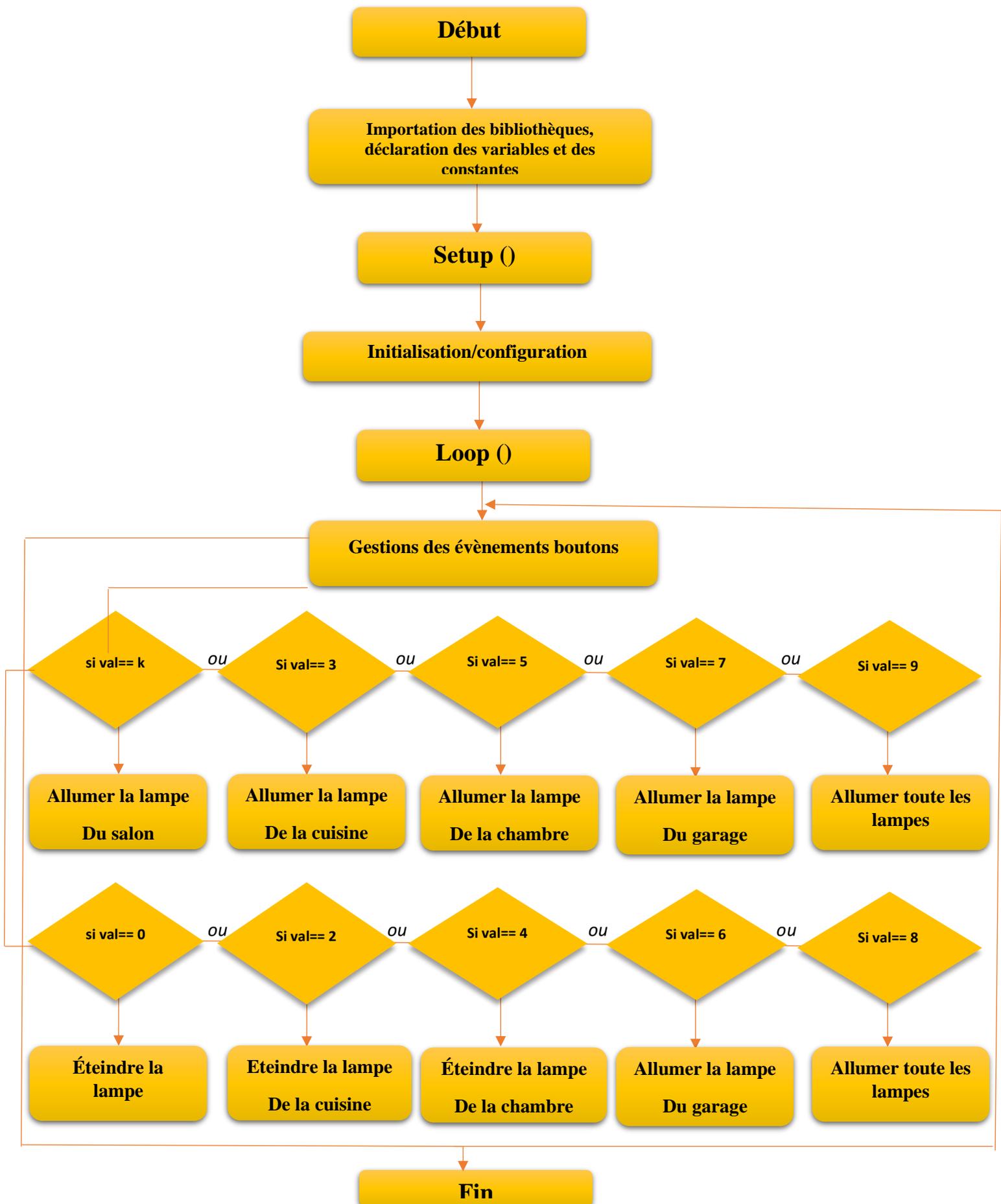


Figure III-19 : Organigramme de gestion de luminosité.

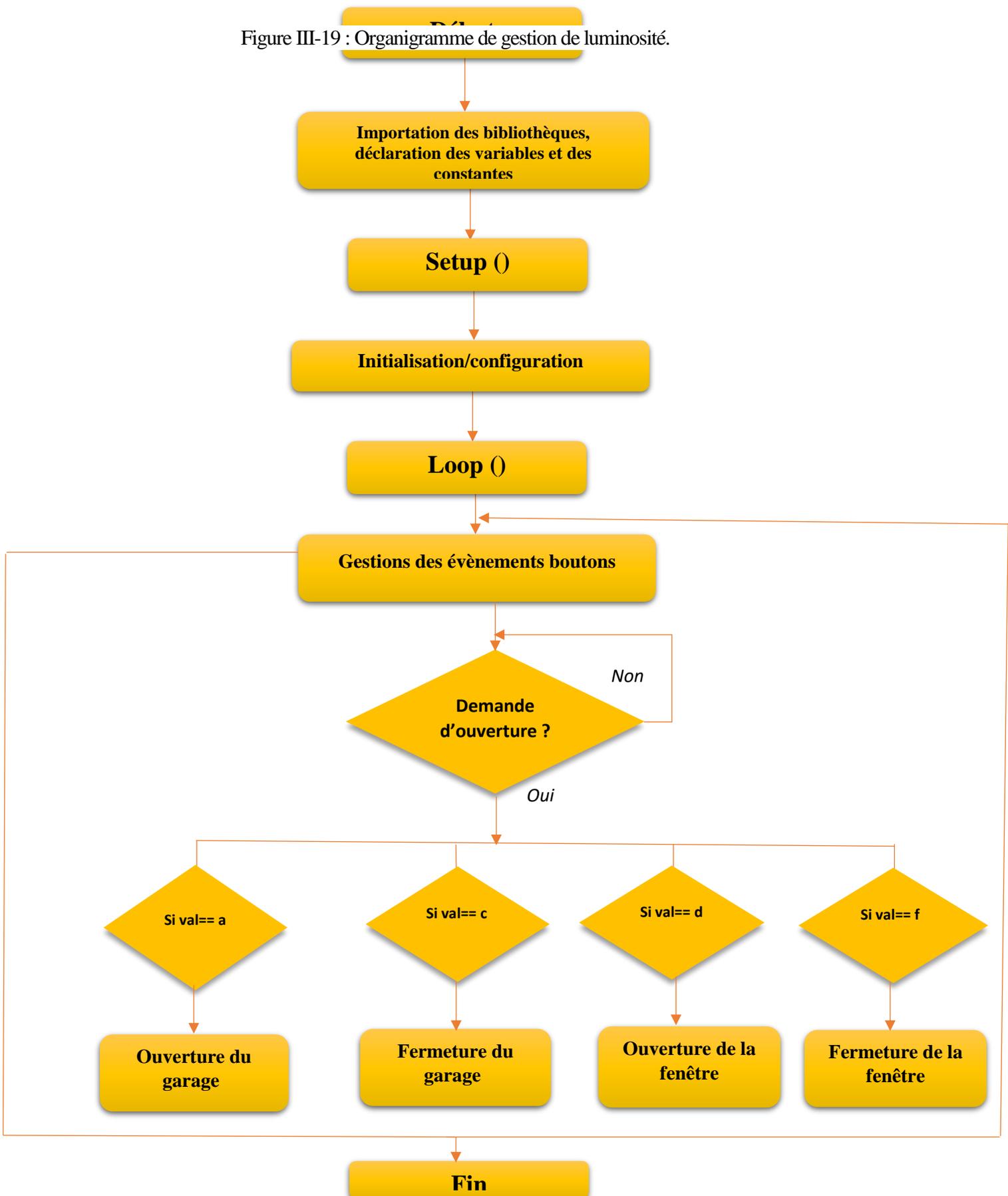


Figure III-20 : Organigramme fonction des ouvrants.

III.3.2 Développement de l'application android

Android est un système d'exploitation développé initialement pour les Smartphones. Il utilise un noyau Linux qui est un système d'exploitation libre pour PC et intègre tout utilitaire et périphérique nécessaires. Il est optimisé pour les outils Gmail. L'androïde est libre et gratuit par conséquent il a été rapidement adopté par les fabricants.

Toujours dans l'optique de garantir le confort et la sécurité des habitants, la conception d'une application sur smartphone ou tablette est préférable. Nous avons choisi App Inventor qui est un logiciel visuel open source en ligne, pour sa facilité d'utilisation. Il s'agit d'un logiciel qui fonctionne avec des blocs à imbriquer comme un puzzle, ces derniers sont préprogrammés.

Le logiciel possède deux interfaces. La première est celle du Design où on construit l'interface graphique de l'application, celle-ci se divise en quatre parties comme indiqué dans la figure III-23.

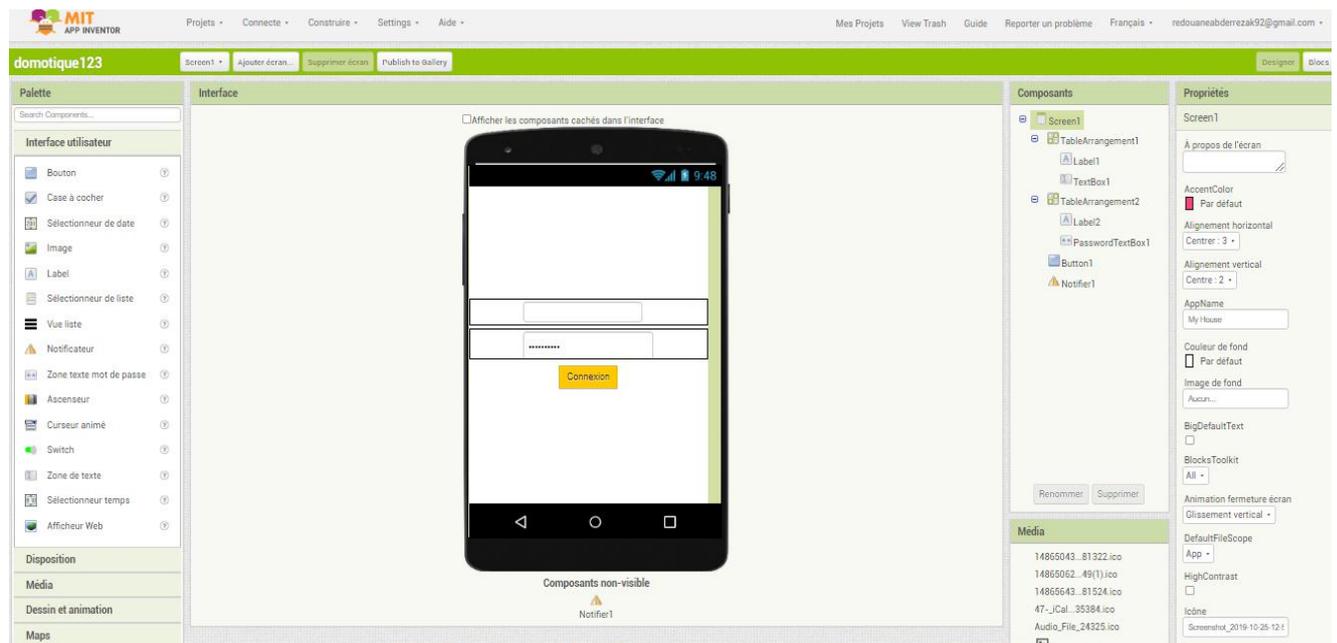


Figure III-21: Interface de design d'AppInventor.

Le « view » : représente un téléphone et donc dessus nous pouvons voir à quoi ressemblera l'application, il regroupe les différents composants et écrans utilisés.

La palette : qui contient les composants pré-codés et préinstallés, ces derniers peuvent être visibles comme invisibles sur l'écran.

Les composants : dans cette partie on retrouve les composants sélectionnés pour notre application dans la partie précédente.

Propriétés : cette partie nous permet de modifier et personnaliser les composants utilisés.

La seconde interface (figure III-24) est celle de la programmation ou nous allons pouvoir programmer les différents éléments choisis. Et cela grâce des blocs codés au préalable ou bien des blocs associés aux composants que nous avons choisis.

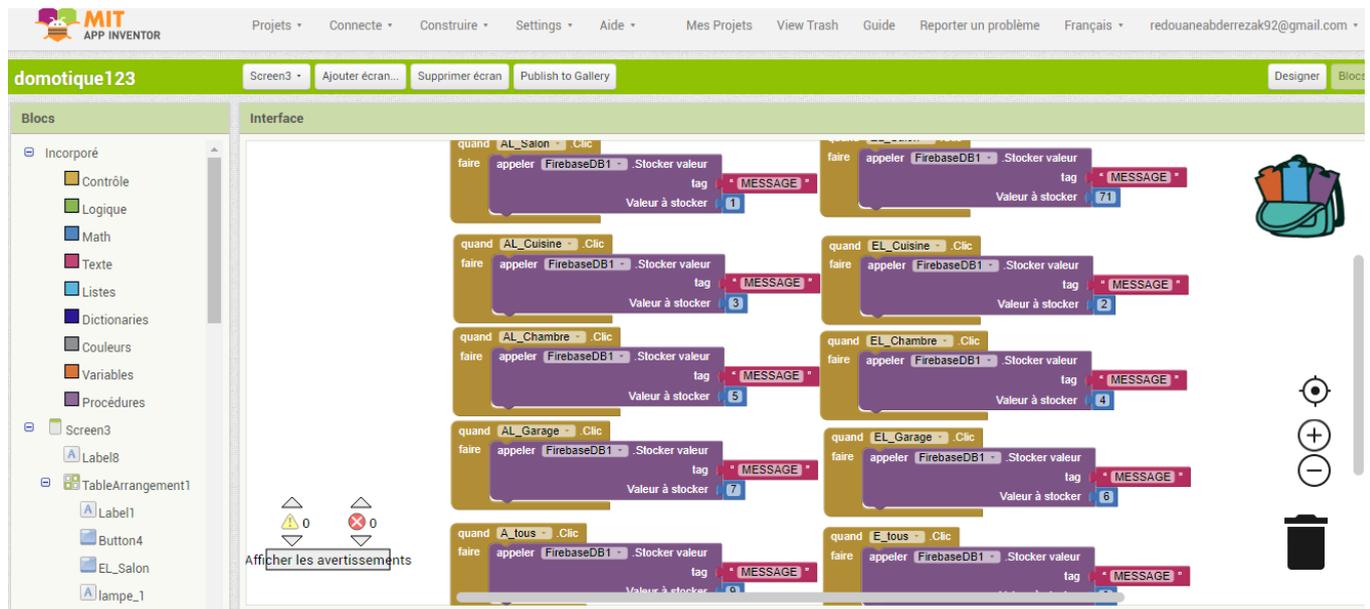


Figure III-22: interface de programmation.

III.3.3 Explication et démarche

L'application que nous avons conçue se connecte à un module wifi (le module ESP2286 expliqué précédemment) afin d'accéder aux fonctionnalités de cette application il faut d'abord s'identifier a traves un identifiant et un mot de passe (figure III-25).



Figure III-23 : fenêtre d'identification

III.3.4 Présentation des fonctions de système

Une fois l'identification terminée, on bascule vers la fenêtre principale (figure III-26).

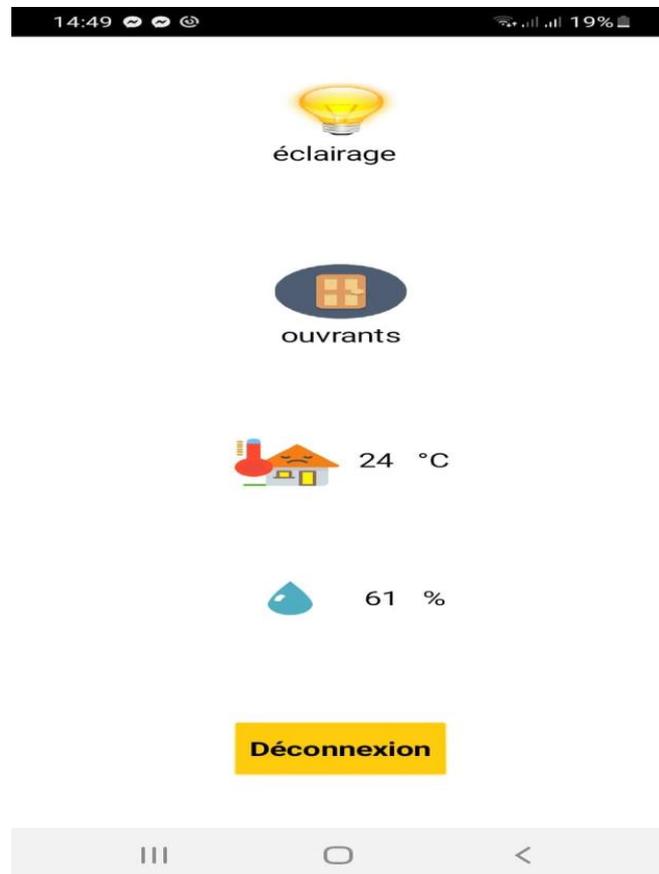


Figure III-24 : Fenêtre principale de l'application « my house ».

III.3.4.1 Fonction de sécurité

Cette fonction permet de détecter s'il y a eu des fuites de gaz dans la cuisine à l'aide du capteur MQ-2, ou bien s'il y a eu un incendie à proximité de la maison via le capteur flamme, une fois la menace détectée une alarme se déclenche et un message s'affiche sur l'écran de l'utilisateur afin de l'informer et l'inciter à appeler les pompiers, le volet de la cuisine s'ouvre automatiquement lors de la détection du GPL (figure III-27, III-28).



Figure III-26 : message d'alerte fuite de gaz.



Figure III-25: message d'alerte d'incendie.

III.3.4.2 Fonction d'éclairage

- Eclairage d'intérieur

La fonction d'éclairage de chaque pièce de la maison est assurée via l'application en appuyant sur les ampoules allumées/éteintes dans l'interface graphique du smartphone. (Figure III-29).

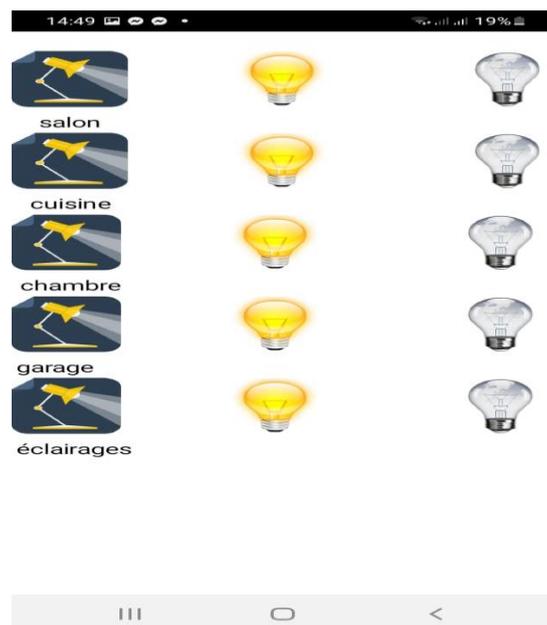


Figure III-27: interface de gestion de luminosité.

- **Eclairage d'extérieur**

L'ampoule extérieure s'allume automatiquement en fonction de la luminosité ambiante de l'environnement et ça grâce au capteur de lumière avec photorésistance.

III.3.4.3 Fonction d'acquisition de température et d'humidité

La fonction de l'acquisition de la température et d'humidité est réalisée via le capteur DHT22 par la suite les valeurs seront affichées dans le smartphone. (Voir figure III-26).

III.3.4.4 Fonction d'ouverture du garage et des volets

La commande d'ouverture du portail du garage et du volet de la cuisine est réalisée à distance via l'application de commande en agissant sur le contrôle du servomoteur pour faire ouvrir/fermer le portail du garage et la porte principale (voir figure III-30).

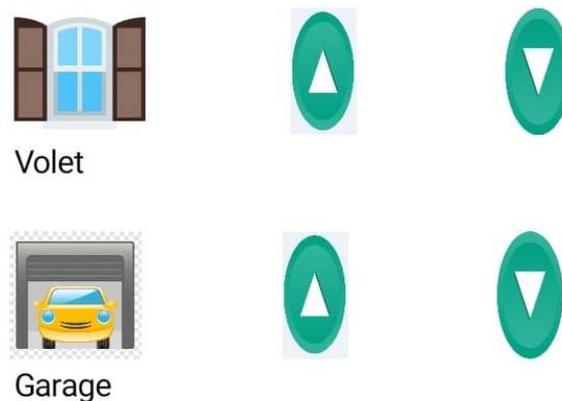


Figure III-28: interface de gestion des ouvrants.

Pour afficher les valeurs des différents capteurs en temps réel, nous allons extraire ces valeurs de FirebaseDB, cette procédure est détaillée dans la section qui suit.

III.3.5 Synchronisation de l'application Android avec la page web de commande

Pour que l'application Android puisse suivre le changement des états effectués par la page de commande, nous avons utilisé une base de données commune entre la page web et l'application Android, qui est le *Firestore database*. Au lieu de requêtes HTTP classiques, la base de données *Firestore* utilise la synchronisation des données chaque fois que les données sont modifiées.

III.3.5.1 Firestore

La base de données *Firestore* est une base de données hébergée dans le *Cloud*. Les données sont stockées en tant que JSON et synchronisées en temps réel sur chaque client connecté. Lorsque nous créons des applications multiplateformes avec SDK iOS, Android et JavaScript, tous les clients partagent une instance de base de données en temps réel et reçoivent automatiquement les mises à jour avec les données les plus récentes.

La base de données en temps réel *Firestore* est accessible directement depuis un appareil mobile ou un navigateur Web. Il n'y a pas besoin d'un serveur d'applications. La sécurité et la validation des données sont disponibles via les règles de sécurité de base de données *Firestore*, règles basées sur des expressions qui sont exécutées lors de la lecture ou de l'écriture de données.

III.3.5.2 La commande via internet

Pour connecter Arduino au réseau mondial, il faut utiliser l'IP public de modem, pour le connaître, il suffit de consulter les sites suivants :

- Mon-ip.com
- Votreip.free.fr

Une fois l'adresse IP public récupérée, il convient de paramétrer le box (modem ou router) de sorte que quand on se connecte sur l'adresse public et sur certain port, le box dirige automatiquement la connexion vers Arduino (qui sera en mode serveur).

La méthode de paramétrage se diffère d'un box à un autre, mais le principe général est le suivant :

- Associer l'adresse IP local de l'Arduino (192.168.X.X) à un nom.

- Associer cette adresse avec l'adresse MAC de l'Arduino, donc la rendre statique.
- Relier l'adresse locale avec l'adresse publique du box et un numéro de port.

III.4 Conclusion

Dans ce chapitre nous avons présenté et expliqué les deux parties hardware et software qui ont contribué à la réalisation de notre projet et nous avons défini les composants qui constituent notre maquette. Nous avons pu les tester et les faire fonctionner, en plus c'était l'occasion pour nous de nous perfectionner dans la programmation et de nous initier dans l'électronique. C'était une véritable découverte étant donné que l'univers des Webmaster nous était totalement inconnu jusqu'alors.

Conclusion Générale

CONCLUSION GÉNÉRALE

Dans ce mémoire, nous avons présenté un type particulier des réseaux ad hoc qui est le réseau de capteur sans fil avec ses caractéristiques, domaines d'applications. Nous nous sommes focalisés sur un domaine particulier en l'occurrence la domotique. Afin de concrétiser notre projet nous avons, en premier lieu, exposé tous les besoins et toutes les contraintes que nous avons rencontrés lors de l'étude. Après plusieurs recherches nous nous sommes amenés à choisir la plateforme Arduino pour développer et réaliser ce projet.

Ce projet se présente sous deux aspects : l'aspect hardware qui consiste à concevoir un prototype d'une maison miniature, sur laquelle nous avons testé les organigrammes de commande, l'aspect software qui consiste à développer une application Android qui permet de contrôler les dispositifs domestiques à distance.

Nous avons choisi la carte Arduino UNO comme unité de commande pour relier tout phénomène physique, son rôle est de traiter les données délivrées par les capteurs et commander les différents actionneurs utilisés. Nous avons développé l'interface de commande Android sous l'environnement MIT App Inventor. Nous avons préparé les icônes et les labels des commandes pour permettre au système de superviser les résultats attendus programmés dans la carte Arduino. Afin de relier ces deux parties hardware et software nous avons utilisé le module ESP8266 qui se connecte au système réseau internet à travers la connexion sans fil (WIFI).

Une telle réalisation n'est pas dénuée de difficultés, il est à noter que nous nous sommes confrontés à plusieurs contraintes et problèmes dans la partie connexion sans fil. Cependant, malgré toutes ces difficultés, les résultats obtenus à travers cette étude, permettent d'ouvrir la porte à d'autres études. Nous espérons que ce mémoire pourrait faire objet de référence aux personnes souhaitant développer et réaliser des projets de synthèse à base d'une carte Arduino.

Pour perspectives d'amélioration, plusieurs voies peuvent être envisagées notamment la gestion des multimédias, l'arrosage automatique, surveillance interne et externe avec cameras. Comme nous pouvons utiliser le module GSM pour contrôler les fonctions domotiques.

Bibliographie

BIBLIOGRAPHIE

- [1] Vernon S. Somerset, "Intelligent and Biosensors", Edited by Vernon S. Somerset, Intech, January 2010.
- [2] F. Brissoud, D. Charpentier, A. Barros et C. Bérenguer, « capteurs intelligents : nouvelles technologies et nouvelles problématiques pour la sûreté de fonctionnement », Maitrise des Risques et de sûreté de fonctionnement, Lambda-Mu 16, Avignon : France(2008).
- [3] S. Sudevalayam, P. Kulkarn, "Energy Harvesting Sensor Nodes: Survey and Implications", IEEE Communications Surveys & Tutorials, Vol. 13, No. 3, pp. 443-461, september 2011.
- [4] J. P. Lynch, A. Sundararajan, K. H. Law, A. S. Kiremidjian, E. Carryer, "Embedding damage detection algorithms in a wireless sensing unit for operational power efficiency", Smart Materials and Structures, Vol. 13, No. 4, pp. 800-810, April 2004.
- [5] H. Qi, Y. Xu, X. Wang, "Mobile-agent-based collaborative signal and information processing in sensor networks", Proceedings of the IEEE, Vol. 91, No. 8, pp. 1172-1183, August 2003.
- [6] Ben L. Titzer et J. Palsberg, "Nonintrusive Precision Instrumentation of Microcontroller Software", ACM, New York, NY, ETATS-UNIS, Vol. 40, No. 7, pp. 59-68, July 2005.
- [7] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, P. Corke, "Data collection, storage, and retrieval with an underwater sensor network", In Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM, pp. 154-165, November 2005.

[11] F.Kristianto Karnadi, Zhi Hai Mo et Kun-chan Lan, "Rapid Generation of Realistic Mobility Models for VANET, Wireless Communications and Networking Conference, WCNC 2007, IEEE, pp. 2506- 2511. 2007

[12] L. Campelli, M. Cesana, R. Fracchia, "Evaluation of Integrated Routing/MAC Solutions for the Diffusion of Warning Messages in VANETs", Journal of Networks (Academy Publisher), Vol. 2, No. 6, pp. 13-23, December 2007.

[13] Yazeed Al-Obaisat, Robin Braun "On Wireless Sensor Networks: Architectures, Protocols, Applications, and Management" Institute of Information and Communication Technologies University of Technology, Sydney, Australia, 2009.

[14] H. Cam, S. Ozdemir, P. Nair, and D. Muthuavinashippan, "ESPDA: Energy-Efficient and Secure Pattern Based Data Aggregation for Wireless Sensor Networks," in press, IEEE Sensor, Toronto, Canada, 2003.

[15] D.Martins, Sécurité dans les réseaux de capteurs sans fil Sténographie et réseaux de confiance, Thèse de doctorat, L'UFR des Sciences et Techniques de l'Université de Franche-Comté, Novembre 2010.

[16] M.O. Farooq and T. Kunz, "Operating systems for wireless sensor networks: A Survey", Sensors, Volume 11, (Issue 6), pages: 5900–5930, 2011.

[17] A. Dunkels, B. Gronvall and T. Voigt, "Contiki - a light weight and flexible operating system for tiny networked sensors", IEEE 29th International Conference on Local Computer Networks, pages: 455-462, November 2004.

[18] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "Mantis OS: An embedded

multithreaded operating system for wireless micro sensor platforms”, *Mobile Networks and Applications Journal*, Volume 10 (Issue 4), pages: 563-579, 2005.

[19] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks”, *Book: Ambient Intelligence*, Chapter 7, pages:115-148, 2005.

[20] M.Hamdi, N.Boudriga, and Mohammad S. Obaidat. Whomoves : An optimized broadband sensor network for military vehicle tracking. *Int. J. Communication Systems*, 21(3) :277–300, 2008.

[21] David M. Davenport, Budhaditya Deb, and Fergus J. Ross. Wireless propagation and coexistence of medical body sensor networks for ambulatory patient monitoring. In *Sixth International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009, Berkeley, CA, USA, 3-5 June 2009*, pages 41–45. IEEE Computer Society, 2009.

[22] Serge Darrieumerlou, " le guide de la maison et des objets connectés", Edition Eyrolles, 2016.

[23] Z. Tang and W. Dargie. “A Mobility-Aware Medium Access Control Protocol for Wireless Sensor Networks”, the fifth IEEE international workshop on Heterogeneous, Multi-Hop, Wireless and Mobile Networks, Miami, Florida, 6 December, 2010.

[24] Y. Zatout, R. Kacimi, J. Llibre, E. Campo, “Mobility-aware Protocol for Wireless Sensor Networks in Health-care Monitoring,” *Fifth International Workshop on Personalized Networks (Pernets 2011)*, Las Vegas, Nevada, USA, January 2011.

[25] S. Bao, Y. Zhang, L. Shen, "Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems," in: 27th Annual International Conference of the IEEE EMBS, 2005

[26] D. Surie, O. Laguionie, T. Pederson, "Wireless sensor networking of everyday objects in a smart home environment," ISSNIP, Sydney, December 15-18, 2008.

[28] Boudellal. M, "Smart home - Habitat connecté, 361 installations domotiques et multimédia". Dunod, 2014.

[29] S.MONK. « The tab book of arduino projects. ». MC graw hill education, 2014.

[30] A.ELHAMMOUMI et M.SLIMANI. «Conception et réalisation d'une maison domotique intelligente, My Smart Home ». Mémoire de Master. Maroc, Université Hassan 1er settat, 2016.

[31] J.Lechalupé. « Cours d'initatiton Arduino ». Cours. Université Paul Sabatier, 2014.

[32] <https://sites.google.com/site/arduinoencore/esp8266>.

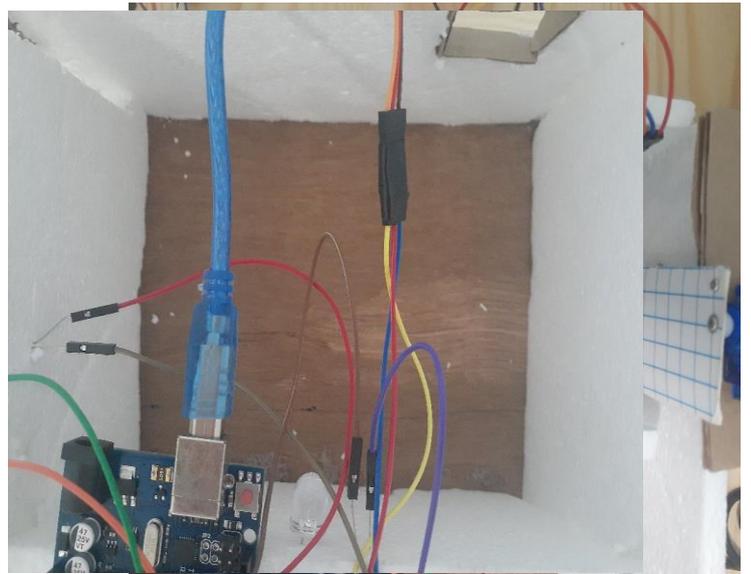
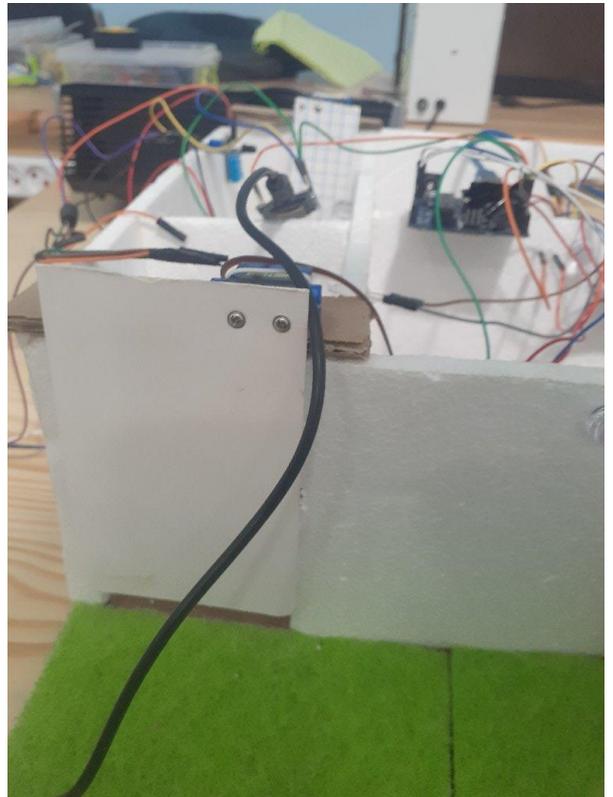
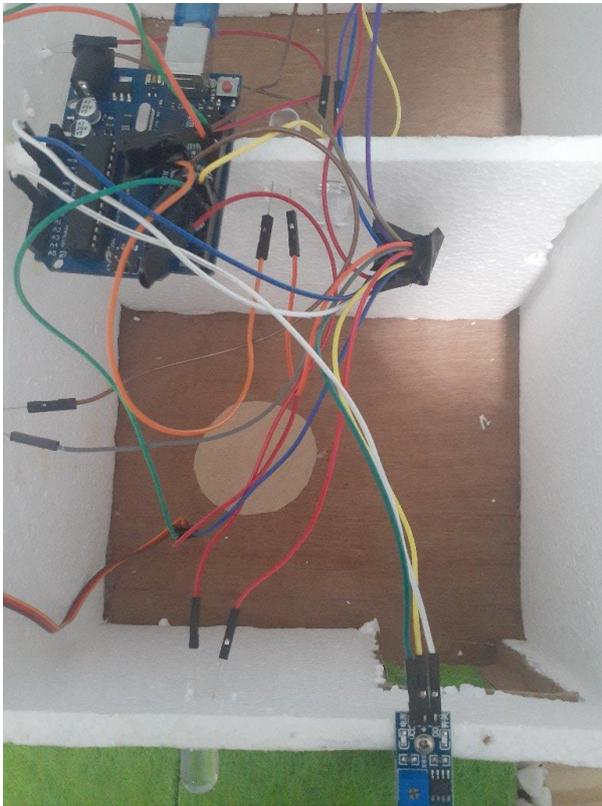
[33] Lechalupé. J, "cours d'initiation à Arduino", Université Paul Sabatier, Mai 2014.

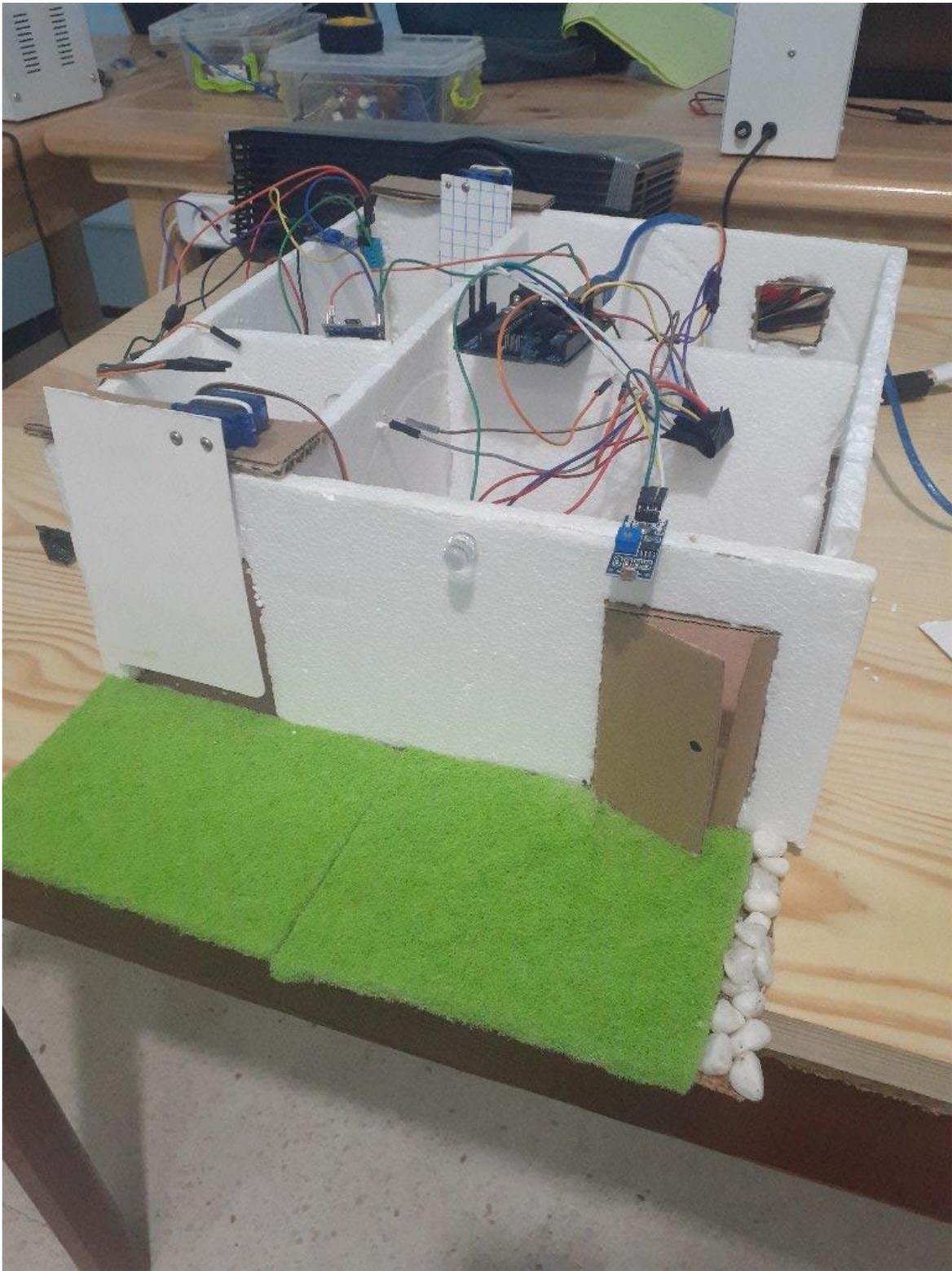
[34] P. Gautier, Objets « connectés », objets « communicants »... ou objets « acteurs », <http://www.refondation.org/blog/2385/internet-des-objets-objets-connectes-objets-communicants-ou-objets-acteurs> [archive].

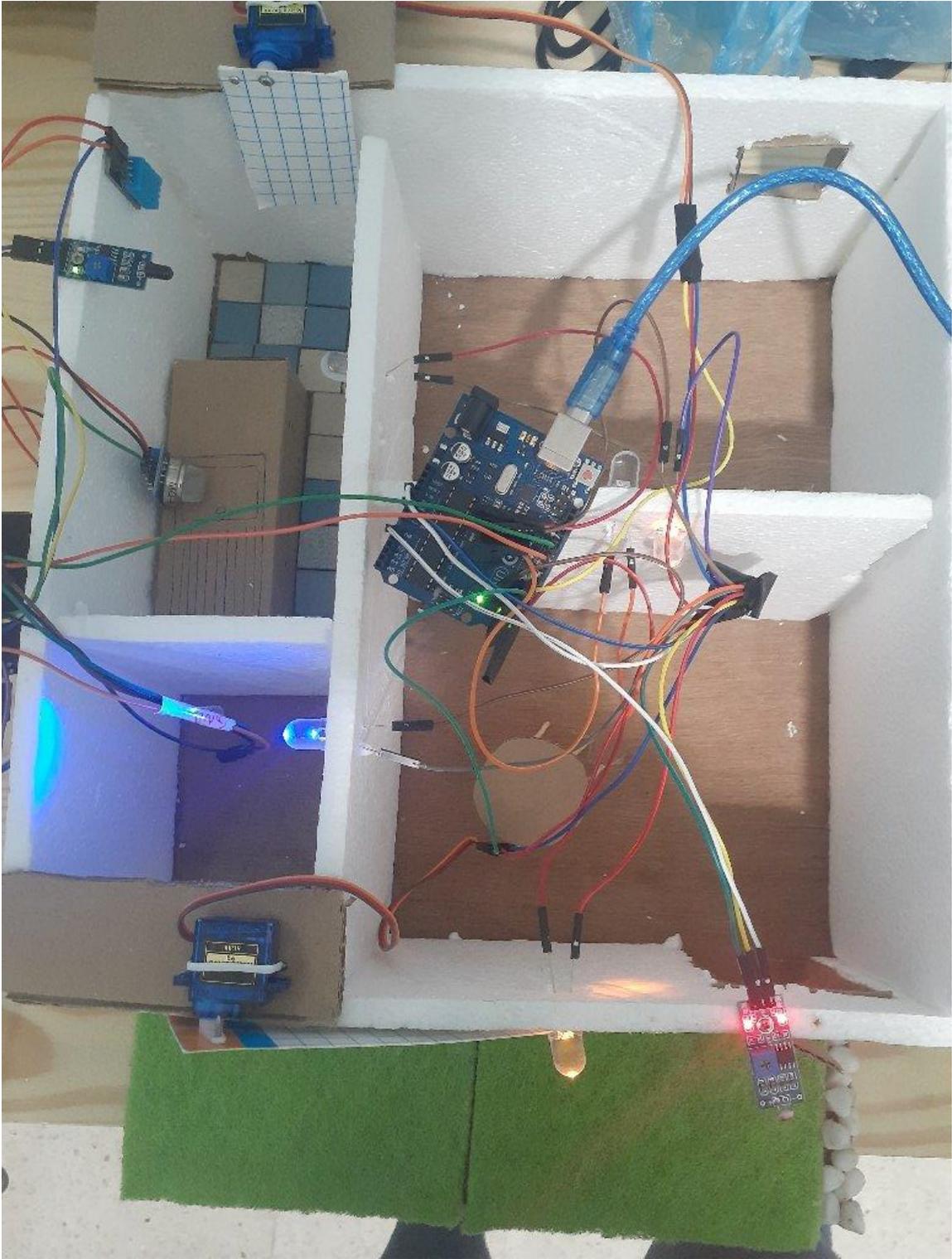
Annexes

ANNEXE

Annexe A







Annexe B

a. Le code televerser sur la carte Arduino

```
#include <Servo.h>

#include <SoftwareSerial.h>

SoftwareSerial myserial(10,11);

Servo myservoG;

Servo myservoC;

const int LDRpin = A0;

int valeurSurA0;

const int LEDpin = 4;

char messageRecu = ' ';

int ledg =8 ;int leds = 12;int ledc =13 ;//int ledch = 13;

void setup() {

  myservoC.write(70);

  myservoG.write(33);

  pinMode (ledg, OUTPUT);pinMode (leds, OUTPUT);pinMode (ledc,
OUTPUT); pinMode(LEDpin,OUTPUT);

  Serial.begin(9600);

  myserial.begin(9600);

  myservoG.attach(5);

  myservoC.attach(6);

}
```

```

void loop() {
while(myserial.available())
{
delay(3);
messageRecu = myserial.read();

Serial.print("messageRecu = ") ; Serial.println(messageRecu) ;

if (messageRecu =='0' ) {digitalWrite(leds,LOW); }
if (messageRecu =='k' ) {digitalWrite(leds,HIGH); }

if (messageRecu =='2' ) {digitalWrite(ledc,LOW); }
if (messageRecu =='3' ) {digitalWrite(ledc,HIGH); }

if (messageRecu =='4' ) {digitalWrite(9,LOW); }
if (messageRecu =='5' ) {digitalWrite(9,HIGH); }

if (messageRecu =='6' ) {digitalWrite(ledg,LOW); }
if (messageRecu =='7' ) {digitalWrite(ledg,HIGH); }

if (messageRecu =='8' )
{digitalWrite(ledg,LOW);digitalWrite(ledc,LOW);digitalWrite(leds,LOW);digitalWrite(9,LOW); }

if (messageRecu =='9' )
{digitalWrite(ledg,1);digitalWrite(ledc,1);digitalWrite(leds,1);digitalWrite(9,1);
}

if (messageRecu =='a' ) {myservoG.write(130); }
if (messageRecu =='b' ) {}

```

```
if (messageRecu == 'c' ) { myservoG.write(33);}

if (messageRecu == 'd' ) { myservoC.write(170);}
if (messageRecu == 'e' ) { }
if (messageRecu == 'f' ) { myservoC.write(70); }

}
valeurSurA0 = analogRead(LDRpin);
Serial.println(valeurSurA0);
  if (valeurSurA0 >1000) {digitalWrite(LEDpin ,HIGH);}
  else {digitalWrite(LEDpin ,LOW);}
  delay (1000);
}
```

b. Le code televerser sur le module ESP8266

```
#include <FirebaseArduino.h>
#include <FirebaseHttpClient.h>
#include <FirebaseObject.h>
#include <SerialTransceiver.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiServer.h>
#include <WiFiUdp.h>

#ifndef UNIT_TEST
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>
SoftwareSerial HC06(D0,D1);
Set these to run example.

#define FIREBASE_HOST "*****"
#define FIREBASE_AUTH "*****"

#define WIFI_SSID "soumia"
#define WIFI_PASSWORD "*****"
```

```
int message;
int temp;
int temp_p = 0;
char MessageSend = '0';
char MessageSend1 = '0';
int etatTAG=70;
char messageRecu_room;
int lampe1 ;
int in1 =16;int in2 = 5;int in3 = 4;int in4 = 0;int inar = 2;
int Val_DOF;
int PIN_DOF = D7;
int Val_DOG;
int PIN_DOG = D6;
#include "DHT.h"
#define DHTPIN D8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  pinMode(4, OUTPUT);
  HC06.begin(9600);
  Serial.begin(9600);
  pinMode(D7,INPUT);//FLAMMME
  pinMode(D6,INPUT);//GAZ
  pinMode(5, OUTPUT);//IN2 D1
  pinMode(4,OUTPUT);//IN3 D2
  pinMode(0, OUTPUT);//IN4 D3
  pinMode(2, OUTPUT);//INAR D4
```

```
connect to wifi.//WiFi.disconnect();// delay(3000);
Serial.println("START");
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("connecting");
while (WiFi.status() != WL_CONNECTED) {//Serial.print(".");
  delay(500);}
Serial.println();
Serial.print("connected: ");
Serial.println((WiFi.localIP().toString()));
while (!Serial) // Wait for the serial connection to be established.
  delay(50);
  Serial.println();
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.setString("LAMPE1", "70");
dht.begin();
}
void loop() {
  float h = dht.readHumidity();
  Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  set value
  Firebase.setFloat("Temp",t);
  Serial.println(t);
  Firebase.setFloat("Hum",h);
  Serial.println(h);
```

```
if (outputValueF == 0) { Firebase.setFloat("fire",0);}
if (outputValueF ==1 ) { Firebase.setFloat("fire",1);}

message = Firebase.getString("MESSAGE").toInt();
Serial.println(message);
Serial.println(message);
if (message!=etatTAG && message!= 0
){Serial.print("OUI");Firebase.setString("MESSAGE","70");
```

```
////////////////////////////////////LAMPE
```

```
if (message==71){MessageSend ='0'; }
if (message==1){MessageSend ='k'; }
if (message==2){MessageSend ='2'; }
if (message==3){MessageSend ='3';}
if (message==4){MessageSend ='4'; }
if (message==5){MessageSend ='5';}
```

```
if (message==6){MessageSend ='6'; }
if (message==7){MessageSend ='7'; }
if (message==8){MessageSend ='8'; }
if (message==9){MessageSend ='9'; }
```

```
////////////////////////////////////GARAGE
```

```
if (message==21){MessageSend ='a';} //H
if (message==20){MessageSend ='b'; }//S
if (message==19){MessageSend ='c'; } // B
```

```
////////////////////////////////////VOLET
```

```

if (message==18){MessageSend ='d'; }//H
if (message==17){MessageSend ='e';} //S
if (message==16){MessageSend ='f'; }//B
  HC06.write(MessageSend);
  delay(10);
}

//Val_DOG = digitalRead(PIN_DOG);//Serial.println(Val_DOG);
Val_DOF = digitalRead(PIN_DOF);//Serial.println(Val_DOF);
Val_DOG = digitalRead(PIN_DOG);
if (Val_DOF==0 ||| Val_DOF==0
)
{ Firebase.setFloat("fire",1);HC06.write('d');
}
  if (Val_DOF==1||| Val_DOF==0
)
{ Firebase.setFloat("fire",0);
}
Val_DOF = digitalRead(PIN_DOF);//Serial.println(Val_DOF);
if (Val_DOG==0 ||| Val_DOF==0
)
{ Firebase.setFloat("gaz",1);HC06.write('d');
}

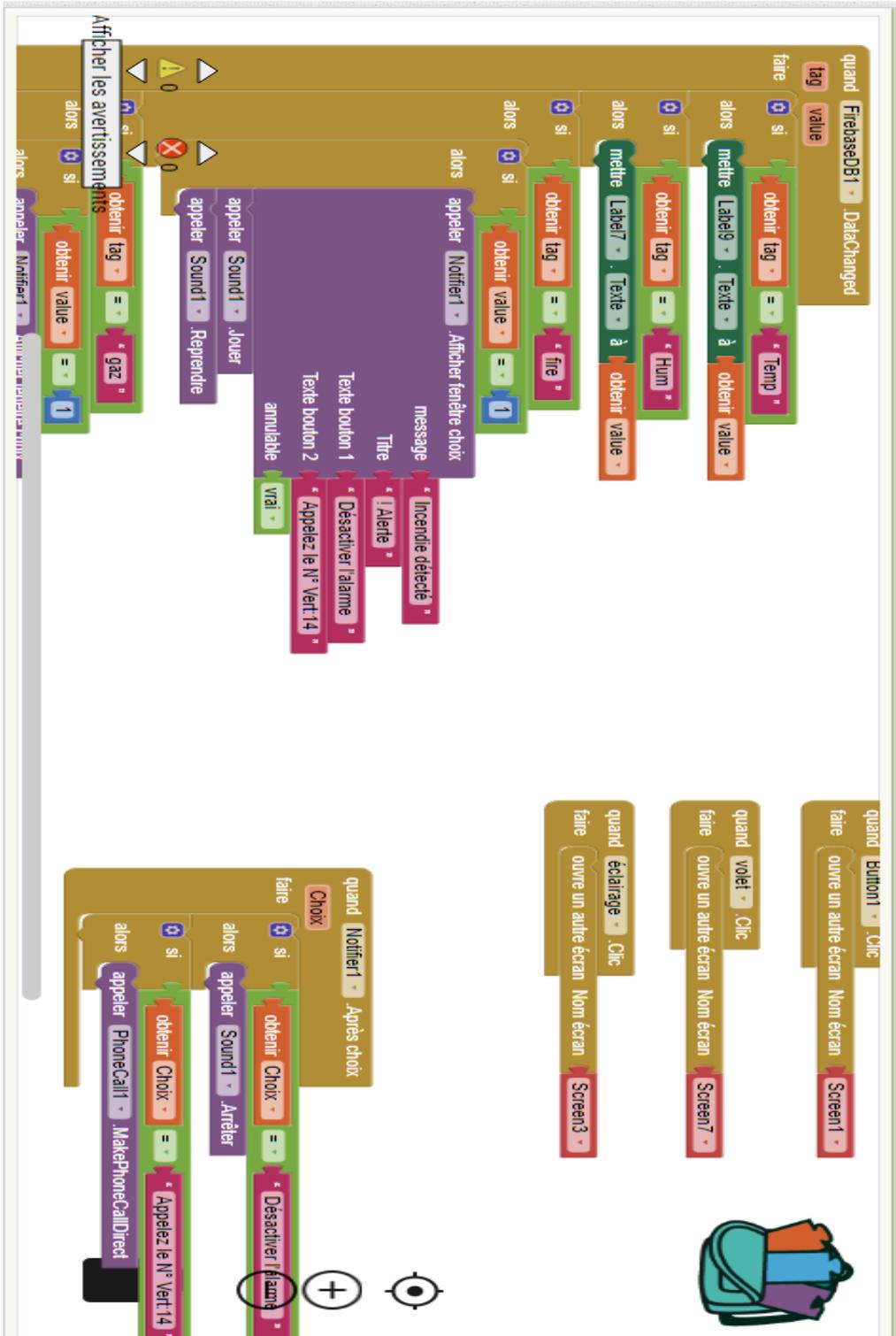
  if (Val_DOG==1||| Val_DOF==0
)
{ Firebase.setFloat("gaz",0);
}

```

```
}  
  
//while(Serial.available())// lorsque un message arrive sur le port série ..  
//{  
//delay(3);// attend un peu que le message arrive  
// messageRecu_room = Serial.read();  
//  
// // Serial.print("messageRecu_room = ") ; Serial.println(messageRecu_room  
// );  
//  
// if ( messageRecu_room=='1'  
// ){Firebase.setFloat("ROOM",1);delay(800);Firebase.setFloat("ROOM",0);}  
//  
// if ( messageRecu_room=='2'  
// ){Firebase.setFloat("ROOM",2);delay(800);Firebase.setFloat("ROOM",0);}  
//  
//  
// if ( messageRecu_room=='3'  
// ){Firebase.setFloat("ROOM",3);delay(800);Firebase.setFloat("ROOM",0);}  
//  
// if ( messageRecu_room =='4'  
// ){Firebase.setFloat("ROOM",4);delay(800);Firebase.setFloat("ROOM",0);}  
// }  
  
}
```

Annexe C

Programmation sur AppInventor

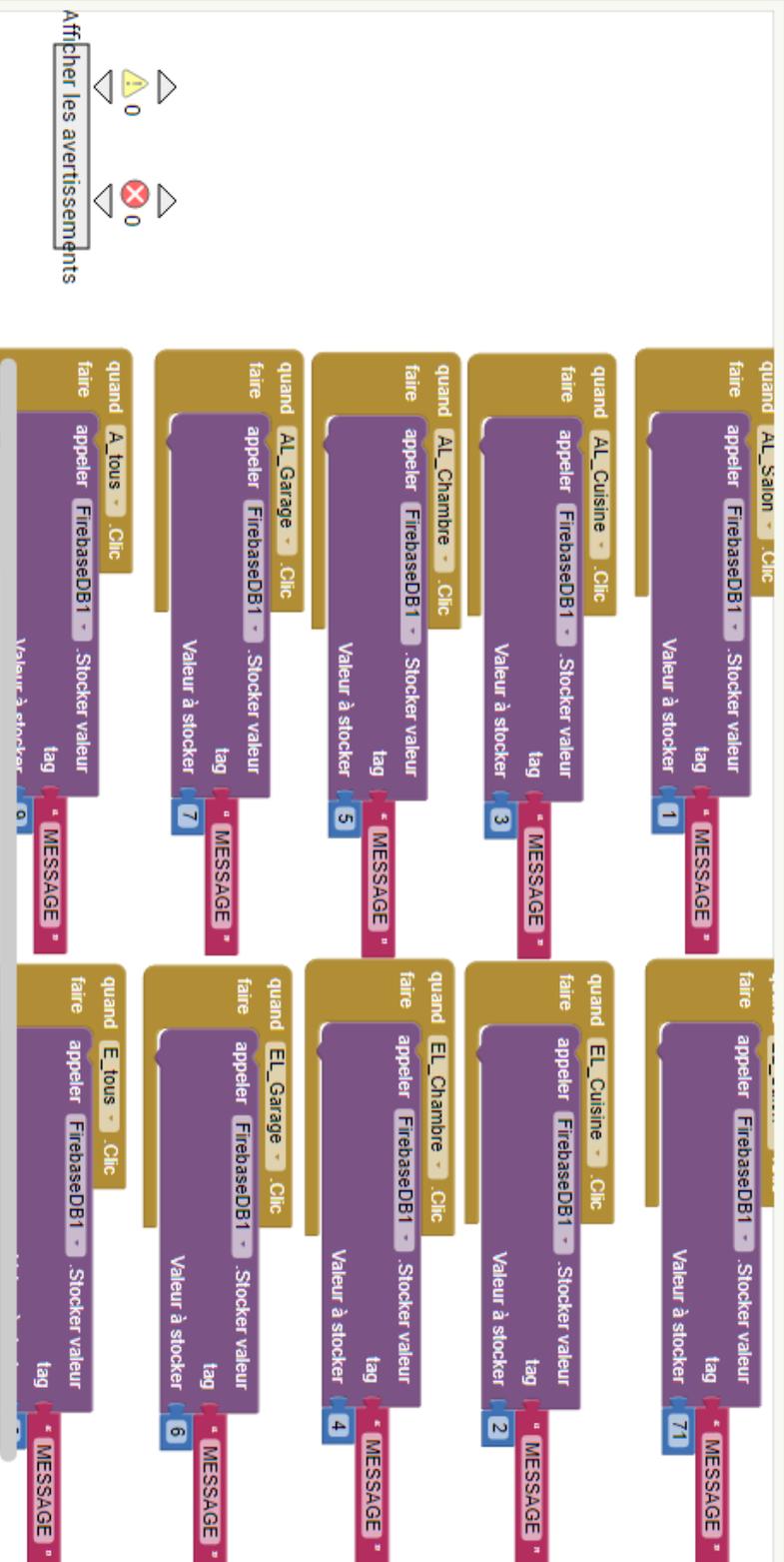


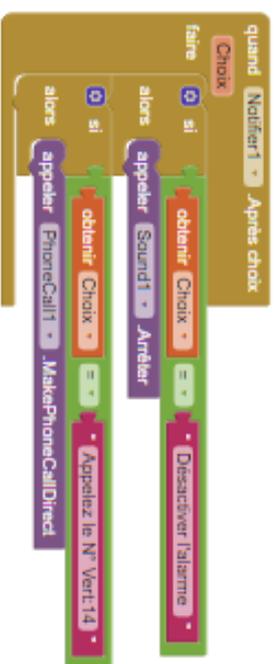
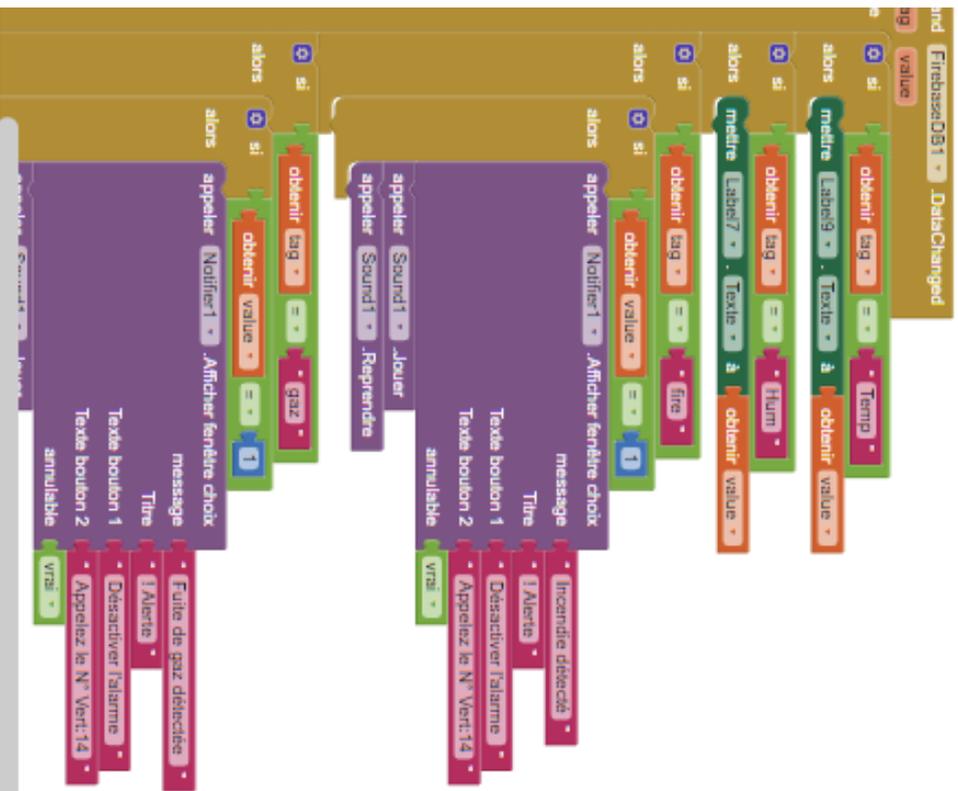
domotique123

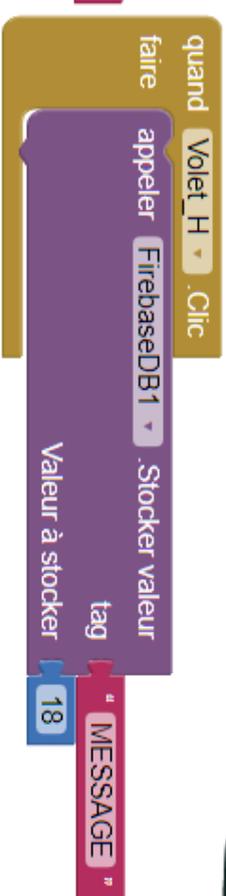
Blocs

- Incorpore
 - Contrôle
 - Logique
 - Math
 - Texte
 - Listes
 - Dictionnaires
 - Couleurs
 - Variables
 - Procédures
- Screen3
 - Label8
 - TableauArrangement1
 - Label1
 - Button4
 - EL_Salon
 - lampe_1

Interface

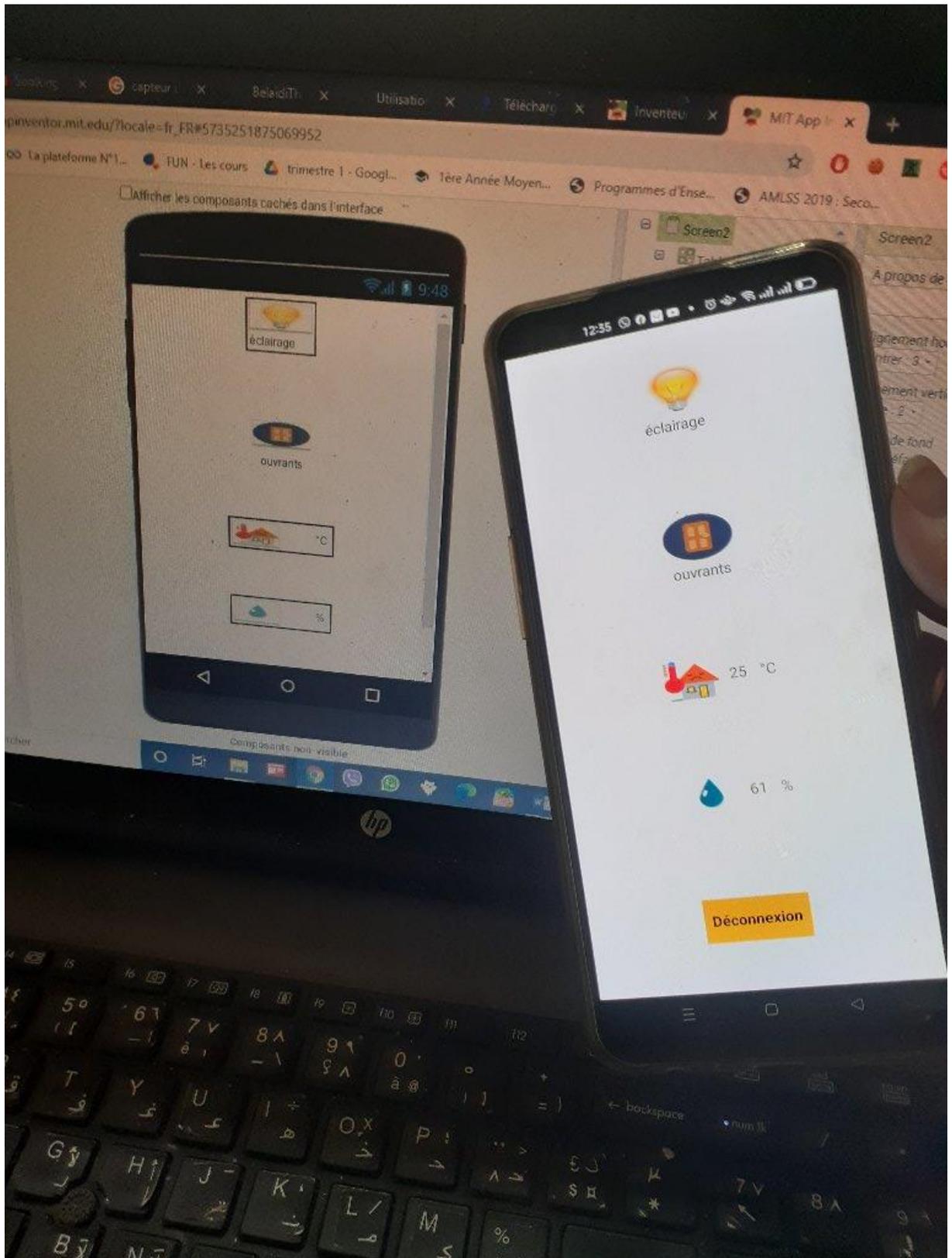






Afficher les avertissements





Résumé

L'internet s'est développé sensiblement surtout ces derniers temps. De nos jours l'utilisation de l'internet n'est pas limitée à la gestion des réseaux, mais aussi s'est étendue à la gestion des objets, ce qu'est appelé Internet des objets, parmi les domaines les plus en vue de l'utilisation de cette nouvelle technologie est le domaine de domotique.

Le but principal de ce travail est la réalisation d'un système de commande à distance (sous réseau Internet) à base d'un Arduino UNO pour commander des installations électriques pour la domotique, afin de faciliter les tâches répétitives du quotidien.

Les mots clés : Arduino, Domotique, Réseau de capteurs sans fil, Conception, Protocole, UNO.

Abstract

The Internet has developed significantly in recent times. Nowadays the use of the Internet is not limited to the management of networks, but also extended to the management of objects, and that is what it is called the Internet of things, Among the areas most prominent in the use of this new technology is the domain of home automation.

The main purpose of this work is the realization of a remote control system (under Internet network) based on an Arduino UNO to control electrical installations for home automation, to facilitate the repetitive tasks of everyday life.

Keywords: Arduino, Home automation, Wireless sensor network, Design, Protocol, UNO.