



# PROJET DE FIN D'ETUDE

CONCEPTION ET RÉALISATION D'UN SMART OFFICE QUI  
CONTRÔLE DES LUMIÈRES, DES VOLETS ET DU  
VENTILATEUR..

LICENCE EN SCIENCES MATHÉMATIQUES ET INFORMATIQUE

**PRÉALISÉ PAR :**  
HAJAR BOUYOUS  
HAFSA EL KHATTABI

**ENCADRE PAR :**  
Pr. Zahour Omar

---

# REMERCIEMENT

Nous sommes reconnaissants envers Dieu qui nous a donné la patience et le courage nécessaires pour traverser ces années d'études.

Nous aimerais également exprimer notre gratitude envers toutes les personnes qui ont contribué à la réalisation de ce projet ainsi qu'à la réussite de cette année universitaire.

Tout d'abord, nous sommes reconnaissants envers le corps professoral et administratif de l'Université des sciences Ben Msik, qui nous a offert un enseignement d'une richesse et d'une qualité remarquables et qui s'efforce constamment de fournir aux étudiants une formation à jour.

Nous aimerais remercier sincèrement M. Benlamher El Habib, Mme Elfiala Sanaa, Mme Sara Ouahabi, M. zahour et Mr Abdeljalil El Hassani comme co-encadrant qui nous ont guidés dans la réalisation de ce projet. Nous sommes reconnaissants pour leur aide et leurs conseils précieux tout au long de ce projet.

Nous n'oublions pas non plus nos parents pour leur contribution, leur soutien et leur patience.

En résumé, nous adressons nos sincères remerciements à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste projet.

---

---

# Sommaire

Liste des figures.....	1
Liste des tableaux.....	2
Résumer.....	3
Abstract.....	4
Introduction générale.....	5
Chapitre 1:Présentation de cahier des charges.....	7
Introduction.....	8
1-Contexte du projet.....	9
2-Objectifs du projet.....	9
3-Description fonctionnelle du projet .....	10
4-Description technique du projet.....	11
5-Planification du projet.....	12
5.1-l'équipe de travail.....	12
5.2-Les différents phases du projet.....	12
5.3-Les phases de réalisation de notre projet.....	12
5.4-Diagramme de Gantt.....	13
Conclusion.....	14
chapitre 2:Etude du projet .....	15
1-Introduction sur l'internet des objets.....	16
2- l'histoire de l'internet des objets.....	17
3-Principe de fonctionnement de l'internet des objets.....	18
4-Domaines d'application de l'internet des objets.....	18
4.1-les appareils intelligents.....	19
4.2-Les maisons intelligentes.....	19
4.3-Les usines et la fabrication intelligente.....	19
4.4- Le système de santé électronique.....	20
4.5- Les villes intelligentes.....	20
conclusion.....	21
chapitre 3:état de l'art.....	22
introduction.....	23
1-Etude de l'existant.....	24
1.1- Zennio Z41.....	24
1.2-Coviva Hager.....	25
1.3-KNX Schneider.....	26
2-Analyses et Solution propose.....	28
conclusion.....	30
Chapitre4:choix de matériel.....	31
introduction.....	32
1- Choix de la carte programmable.....	33
1.1- Carte RASPBERRY_PI.....	33

---

---

1.2- La cartes FPGA.....	34
1.2.1- Spécifications techniques.....	35
1.2.2- Caractéristiques.....	36
1.3- Carte ARDUINO.....	36
1.3.1-Arduino Méga 2560.....	37
1.3.2-Caractéristiques.....	37
2- Choix de la carte programmable.....	38
3-Choix des composants du système.....	40
3.1-Shield Ethernet W5100.....	40
3.1.1-caractéristiques.....	40
3.2-Module de Relais à 8 canaux.....	41
3.3-ventilateur 12V.....	41
3.3.1-caractéristiques.....	42
3.4-lampe.....	42
3.5-Servomoteurs.....	43
3.5.1-Caractéristiques.....	44
3.6-L'ESP8266.....	45
3.6.1-Caractéristiques.....	45
conclusion.....	47
Chapitre 5:Conception, analyse et spécification.....	48
introduction.....	49
1-Identification des acteurs.....	50
2-Besoins fonctionnels .....	50
2.1-Responsable de sécurité.....	50
2.2-Responsable de direction(admin).....	51
3-Besoins non fonctionnels.....	51
4-Spécification des besoins.....	52
4.1-Description textuelle du cas d'utilisation "S'authentifier ".....	52
4.2- Description textuelle du cas d'utilisation "s'inscrire " .....	53
4.3- Description textuelle du cas d'utilisation "Gérer lumière " .....	54
4.4-Description textuelle du cas d'utilisation "Gérer le volet " .....	54
4.5- Description textuelle du cas d'utilisation "Gestion d'aération ".....	55
4.6-Description textuelle du cas d'utilisation "Gestion de climatisation " .....	55
4.7-Diagramme de cas d'utilisation.....	56
4.7.1- Diagramme de cas d'utilisation du système.....	56
4.8- Diagramme de séquence.....	57
4.8.1- l'authentification.....	57
4.8.2- Allumage de lampe.....	58

---

---

4.8.3- Gestion d'aération et de climatisation.....	58
4.8.4-Gestion de volet.....	59
4.9-Diagramme de classe.....	60
4.10-Diagramme de flux de données.....	62
conclusion.....	64
Chapitre 6:Etude des solutions logicielles.....	65
introduction.....	66
1-Plateforme de programmation Arduino.....	67
1.1-Présentation du plateforme.....	67
1.2-Structure d'un programme en Arduino.....	68
2-Le simulateur fritzing.....	69
2.1-Présentation du simulateur.....	69
2.2-Les avantages de fritzing.....	70
3-Java.....	71
3.1-Java swing.....	71
4-My SQL.....	72
5-XAMPP.....	73
conclusion.....	74
Chapitre 7:Phase de la réalisation des systèmes IOT.....	75
1-Présentation du système Smart Home.....	76
1.1-Matériels nécessaire.....	77
1.2-Simulation.....	77
1.3-Déscription de code.....	78
1.3.2-Configuration.....	78
1.3.3-Traitement.....	78
1.3.4-Résultat.....	79
2-Présentation de l'application.....	80
2.1-logo de l'application.....	80
2.2-Page d'authentification.....	80
2.3-Page d'inscription.....	81
2.4-Page d'accueil.....	82
2.5-page d'administration.....	83
Conclusion.....	84
Conclusion générale.....	85

---

---

# Liste des figures

Figure 1 :Diagramme de Gantt .....	13
Figure 2 :architecture évolutive de l'IoT.....	17
Figure 3:Zennio Z41.....	25
Figure 4:Coviva Hager.....	26
Figure 5:KNX Schneider.....	28
Figure 6:Carte RASPBERRY_PI.....	33
Figure 7:La cartes FPGA.....	34
Figure 8:Arduino Méga 2560.....	37
Figure 9:Shield Ethernet W5100.....	40
Figure 10:Module de Relais à 8 canaux.....	41
Figure 11:ventilateur 12V.....	42
Figure 12:lampe.....	43
Figure 13:Servomoteurs.....	45
Figure 14:ESP8266.....	46
Figure 15: Diagramme de cas d'utilisation du système.....	56
Figure 16:diagramme de séquence "l'authentification".....	57
Figure 17:diagramme de séquence "Allumage de lampe" .....	58
Figure 18:diagramme de séquence " Gestion d'aération et de climatisation.....	59
Figure 19:diagramme de séquence " Gestion de volet".....	60
Figure 20:diagramme de classe.....	62
Figure 21:Diagramme de flux de données.....	63
Figure 22 : Espace de développement Arduino.....	67
Figure 23 : Interface de Fritzing.....	69
Figure 24 : Simulateur de système.....	77
Figure 25 : Code de déclaration des variables de Smart office.....	78
Figure 26 :Code de configuration des composants de système Smart home.....	78
Figure 27 :Code de Traitement de système Smart home.....	78
Figure 28 :Résultat de système Smart home.....	79
Figure 29:logo d'application.....	80
Figure 30:Page d'authentification de l'application.....	81
Figure 31:Page d'inscription de l'application.....	82
Figure 32:Page d'accueil de l'application.....	83
Figure 33:Page d'administration de l'applicatio.....	84

---

# Liste des tableaux

Tableau1:Description technique du projet.....	11
Tableau 2:comparatif entre des systèmes existants.....	29
Tableau 3: Tableau des cartes programmables.....	39
Tableau 4:Description textuelle du cas d'utilisation "S'authentifier ".....	53
Tableau 5:Description textuelle du cas d'utilisation "s'inscrire " .....	54
Tableau 6:Description textuelle du cas d'utilisation "Gérer lumière " .....	54
Tableau 7:Description textuelle du cas d'utilisation "Gérer le volet " .....	54
Tableau 8: Description textuelle du cas d'utilisation "Gestion d'aération " .....	55
Tableau 9: Description textuelle du cas d'utilisation "Gestion de climatisation" ..	55



# Résumé

---

Ce rapport présente en détail notre projet de fin d'études : une maison intelligente. Notre objectif était de créer des systèmes indépendants pour gérer l'éclairage, les fenêtres et la climatisation/ventilateurs. Pour faciliter le contrôle de ces systèmes, nous avons développé une application intuitive permettant de tout contrôler en un seul clic.



---

# Abstract

This report presents in detail our graduation project: a smart home. Our goal was to create independent systems to manage lighting, windows and air conditioning/fans. To facilitate the control of these systems, we have developed an intuitive application allowing you to control everything with a single click.

# Introduction générale

Au cours des dernières années, les avancées technologiques ont connu une grande évolution dans tous les domaines. Cette évolution technologique a pour objectif de faciliter l'utilisation et la communication entre l'être humain et les objets qu'il utilise au quotidien.

La vie quotidienne, en particulier le travail, exige de l'être humain des efforts considérables. Afin que ces efforts soient maintenus à un rythme régulier sans diminution de cadence, il est nécessaire pour lui de disposer d'un environnement extra-professionnel offrant un confort et une sécurité optimaux, afin de maintenir une cadence de travail régulière et efficace. Ce cadre de vie devient une nécessité vitale pour certaines catégories de personnes vulnérables telles que les handicapés et les personnes âgées. Pour cela, l'IOT propose des solutions répondant à ce besoin, offrant une multitude de possibilités comme par exemple le déclenchement de la climatisation et du chauffage et l'ouverture automatique des rideaux des fenêtres... Par une simple pression d'un bouton ou via un smartphone ou une tablette, l'humain peut contrôler de nombreuses fonctionnalités dans leur maison. Sans l'intervention physique de l'humaine. Notre projet consiste à réaliser une maison intelligente, comportant un certain nombre de fonctionnalités visant l'optimisation du confort dans l'habitat.

Notre rapport est structuré en six chapitres détaillant les étapes de réalisation de notre système. Voici un aperçu de chaque chapitre : Dans le premier chapitre, nous présenterons le cahier des charges de notre projet, détaillant les exigences et les spécifications nécessaires. Le deuxième chapitre sera consacré à notre vision du sujet proposé, où nous partagerons nos idées et notre compréhension du problème. Le troisième chapitre sera dédié à l'introduction de l'idée du projet, mettant en lumière les plateformes existantes qui proposent des concepts similaires. Dans le quatrième chapitre du rapport, nous explorerons différents modèles de cartes programmables, en décrivant leurs caractéristiques, dans le but de sélectionner la plus optimisée pour notre projet. Le cinquième chapitre se concentrera sur les besoins de l'utilisateur, en décrivant en détail les objectifs du projet et en les analysant. Nous mettrons en évidence les aspects à prendre en compte pour répondre aux attentes des utilisateurs. Dans le sixième chapitre, nous présenterons la conception de notre système, décrivant son architecture et ses composants clés. Enfin, dans le dernier chapitre, nous détaillerons l'environnement matériel et logiciel utilisé pour la réalisation de notre système, en fournissant des informations sur les ressources technologiques exploitées.

# CHAPITRE 1

## PRESENTATION DE CAHIER DES CHARGES

# Introduction

Ce chapitre a pour objectif de contextualiser notre projet dans son ensemble. Pour ce faire, nous allons présenter en détail son cahier des charges. Nous débuterons par une introduction au contexte dans lequel s'inscrit notre projet. Ensuite, nous énoncerons clairement ses objectifs et ses fonctionnalités, en les décrivant de manière concise et précise. Nous aborderons également la planification du projet, en définissant les étapes clés et les délais à respecter.

## 1-Contexte du projet :

smart home se situe dans le domaine des maisons intelligentes et de l'Internet des objets (IoT). Avec l'avancement des technologies de communication et de l'automatisation, les maisons intelligentes sont devenues une réalité accessible et attrayante pour de nombreux propriétaires.

Dans ce contexte, le projet vise à répondre aux besoins des occupants en matière de confort, de sécurité et d'efficacité énergétique. Les maisons intelligentes offrent des fonctionnalités avancées qui permettent aux utilisateurs de contrôler divers aspects de leur environnement domestique à l'aide de technologies connectées.

Le projet est motivé par les avantages potentiels d'une maison intelligente, tels que la facilité d'utilisation, l'économie d'énergie, l'amélioration de la sécurité et le confort personnalisé. En offrant un contrôle centralisé et convivial de l'éclairage, de la climatisation et des fenêtres, le projet vise à améliorer la qualité de vie des occupants tout en réduisant leur consommation d'énergie.

Le contexte du projet comprend également l'évolution rapide des technologies de l'IoT, des capteurs, des systèmes de communication sans fil et des applications mobiles. Ces avancées technologiques offrent des opportunités pour créer des systèmes de smart home plus sophistiqués, intégrés et faciles à utiliser.

## 2:Objectifs du projet :

Les objectifs et les avantages que notre projet vise sont les suivants :

- Améliorer le confort.
- Optimiser l'efficacité énergétique.
- Offrir une automatisation intelligente.
- Renforcer la sécurité.
- Faciliter la gestion à distance.
- Fournir une expérience utilisateur conviviale.

### **3-Description fonctionnelle du projet :**

Le projet de smart home vise à développer un système intégré qui permet le contrôle intelligent de l'éclairage, de la climatisation et des fenêtres d'une maison, en utilisant une application mobile conviviale. L'objectif principal est d'améliorer le confort et l'efficacité énergétique de la maison, tout en offrant une expérience utilisateur pratique et personnalisée.

## 4-Description technique du projet :

<i>Matériels</i>	<i>Description</i>
<i>Arduino</i>	L'Arduino Mega est une carte de développement électronique puissante et polyvalente basée sur le microcontrôleur ATmega2560. Elle offre une grande capacité de stockage et de nombreuses broches d'E/S.
<i>Lampe</i>	Une lampe est un dispositif d'éclairage utilisé pour produire de la lumière à partir d'une source électrique
<i>Résistance</i>	Une résistance est un composant électrique passif qui limite le flux de courant électrique dans un circuit en opposition à la tension appliquée
<i>Les câbles</i>	Les câbles sont des fils conducteurs recouverts d'isolants utilisés pour transmettre l'électricité, les signaux ou les données entre différents appareils ou composants d'un système électrique ou électronique.
<i>breadboard</i>	est une plaque de prototypage électronique utilisée pour connecter facilement et temporairement des composants électroniques sans avoir besoin de souder, permettant ainsi de réaliser des montages et des expérimentations rapidement.
<i>l'ESP8266</i>	L'ESP8266 est un microcontrôleur WiFi intégré largement utilisé pour la connectivité Internet des objets (IoT), offrant une puissance de traitement et une connectivité sans fil à faible coût dans un petit format.
<i>fan</i>	est un appareil électrique qui utilise des pales rotatives pour créer un courant d'air, procurant ainsi une ventilation et un refroidissement dans diverses applications.
<i>servo-moteur</i>	Un servo-moteur est un dispositif de commande mécanique qui utilise un moteur électrique et un système de feedback pour effectuer des mouvements précis et contrôlés, couramment utilisé dans les applications nécessitant un positionnement précis, telles que la robotique et l'automatisation.

Tableau 1:Description technique du projet

## 5-Planification du projet :

### 5.1-l'équipe de travail:

La réussite d'un projet repose sur plusieurs piliers essentiels, notamment une équipe de travail compétente, une organisation efficace et des choix technologiques appropriés. Dans notre cas, notre équipe est composée de deux étudiants en 3ème année de licence en informatique, ayant déjà obtenu un Deug en Mathématiques Informatique à la Faculté des Sciences Ben M'Sik.

### 5.2-Les différents phases du projet:

Le cycle de vie du projet que nous avons défini pour notre projet se compose de quatre étapes principales :

- l'étape de cadrage,
- l'étape de conception et de planification,
- l'étape de réalisation du projet
- l'étape de clôture.

Chacune de ces étapes est composée de plusieurs phases et tâches spécifiques pour assurer la réussite du projet.

### 5.3-Les phases de réalisation de notre projet :

Phase 1 : Recherches.

Phase 2 : Réalisation du Cahier des charges.

Phase 3 : Recherches complémentaires.

Phase 4 : Étude de la partie matérielle et logicielle.

Phase 5 : Réalisation des systèmes IoT.

Phase 6 : Réalisation du Rapport.

Phase 7 : Préparation de la Présentation.

## 5.4-Diagramme de Gantt:

Dans le cadre de notre projet, nous avons utilisé le Diagramme de Gantt pour effectuer la planification. Le Diagramme de Gantt est un outil visuel qui nous a permis de représenter les différentes tâches du projet sur une ligne de temps. Chaque tâche est affichée sous forme de barre horizontale, avec sa date de début et sa date de fin prévues. Cela nous a donné une vue d'ensemble claire des séquences des tâches et des dépendances entre elles. Grâce au Diagramme de Gantt, nous avons pu organiser efficacement les activités du projet, identifier les jalons importants, suivre l'avancement des tâches et anticiper d'éventuels retards. Il a également facilité la communication et la coordination entre les membres de l'équipe en offrant une représentation visuelle facile à comprendre. En somme, le Diagramme de Gantt s'est révélé être un outil précieux pour la planification et la gestion de notre projet.

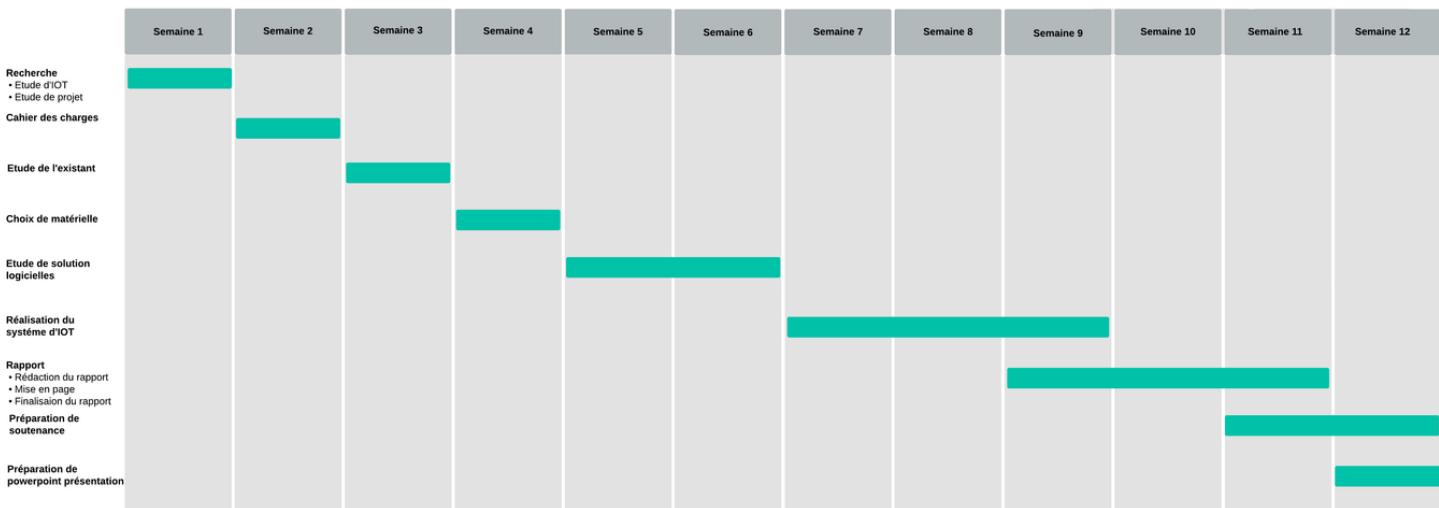


Figure 1 :Diagramme de Gantt

# conclusion

Dans ce chapitre nous avons abordé le contexte global du projet en développant un cahier des charges, en présentant les objectifs à atteindre, en planifiant le projet et en détaillant la structure générale du système Internet des objets (IOT).

# CHAPITRE 2

## ETUDE DU PROJET

## 1-Introduction :

L'Internet des objets (IdO) est une technologie émergente qui englobe un réseau d'objets connectés capable de communiquer entre eux via le réseau Internet, en utilisant des technologies telles que l'informatique omniprésente, la technologie de détection, et les protocoles tels que l'IPv6 et d'autres protocoles. Les objets connectés peuvent être des actionneurs et des capteurs, qui sont connectés à tout moment et n'importe où. On prévoit que le nombre d'appareils IdO triplera dans le monde entre 2018 et 2025, atteignant 25 milliards. Les domaines d'application de l'IdO sont nombreux, allant des villes intelligentes, la fabrication, les transports, la santé, l'agriculture, etc. Des plates-formes matérielles et logicielles ont été développées pour répondre aux demandes croissantes d'applications dans l'industrie et la vie quotidienne. Ces plates-formes sont constituées de différents types de réseaux hétérogènes tels que WSN, MCN et WMN, qui collectent des données à partir des objets intelligents. Cependant, la quantité croissante de données produites par le réseau IdO pose des problèmes de sécurité, de scalabilité, d'hétérogénéité et d'interopérabilité. L'optimisation du réseau IdO est donc cruciale pour minimiser l'interférence des flux de trafic des différents services. Une approche d'allocation de ressources est nécessaire pour gérer les services et les affecter aux ressources disponibles, compte tenu de plusieurs paramètres tels que l'hétérogénéité, la bande passante des interfaces des passerelles, la date limite de l'application IdO, la consommation d'énergie, etc.



Figure 2 :architecture évolutive de l'IoT

## 2- L'histoire de l'internet des objets :

L'Internet des Objets (IoT) est une avancée technique qui permet l'intégration naturelle de la technologie dans les objets de la vie quotidienne. Ce développement ouvre de nouvelles perspectives dans la création d'objets interconnectés, permettant l'interaction entre le monde physique et le monde virtuel. Parmi les domaines qui bénéficient de cette avancée technologique, on peut citer la domotique, la ville intelligente, la sécurité, l'e-santé, etc. Cependant, le développement de l'IoT est confronté à des problèmes qui nécessitent des études approfondies pour atteindre ses objectifs.

Le terme "Internet des Objets" a été utilisé pour la première fois en 1999 par Kevin Ashton, et cette technologie promet de changer radicalement notre vie quotidienne grâce à la connectivité des objets. L'IoT représente une avancée majeure dans la réalisation de la vision de l'informatique ubiquitaire, où la technologie devient active dans l'environnement des utilisateurs et intégrée naturellement à l'intérieur des objets du quotidien. Les appareils spécialisés et simples d'utilisation, tels que les liseuses numériques, les télévisions et montres connectées, les ordinateurs de bord et les téléphones intelligents, peuvent communiquer entre eux via plusieurs types de réseaux sans fil.

### 3-Principe de fonctionnement de l'internet des objets :

L'Internet des objets repose sur l'utilisation de capteurs et d'objets connectés qui sont placés dans ou sur des infrastructures physiques. Ces capteurs collectent des données qui sont ensuite transmises par un réseau sans fil à des plateformes IoT.

Les objets connectés interagissent avec leur environnement à l'aide de capteurs tels que la température, la vitesse, l'humidité et les vibrations. Chaque objet connecté possède sa propre carte d'identité, qui le rend unique et identifiable, généralement par une adresse IP. Cette identité numérique permet de localiser l'objet et de lui donner des instructions à partir d'un ordinateur ou d'un téléphone portable.

Les instructions envoyées par le capteur sont transmises à l'objet via un canal de communication tel que le Wi-Fi ou le Bluetooth. Par exemple, si vous êtes en route pour rentrer chez vous en plein été et que vous remarquez que la température de votre maison est élevée, vous pouvez ordonner à votre ventilateur de se mettre en marche en appuyant simplement sur un bouton depuis votre téléphone, pour trouver une température agréable à votre arrivée. La technologie IoT nous permet ainsi de contrôler les objets de notre environnement à distance, de manière pratique et facile.

### 4-Domaines d'application de l'internet des objets :

L'Internet des objets (IoT) est un concept utilisé dans divers domaines tels que la santé, l'environnement, l'industrie, le transport et le militaire. Voici quelques exemples d'application :

#### **4.1-les appareils intelligents :**

L'Internet des objets permet aux appareils intelligents de se connecter entre eux via Internet. Ces objets contiennent des données similaires qui sont utilisées pour stocker et gérer les paramètres donnés. Grâce à cela, vous pouvez contrôler vos appareils intelligents à distance à partir d'applications installées sur votre téléphone portable ou tablette. Cette technologie vous permet donc de piloter facilement vos objets connectés et d'adapter leur fonctionnement à vos besoins et préférences.

#### **4.2-Les maisons intelligentes :**

La domotique est la technologie qui permet à la maison d'être intelligente, avec une connectivité constante à Internet. La maison connectée offre de nombreux avantages pour la vie quotidienne, car elle permet de contrôler l'éclairage, la température, les systèmes de sécurité, les appareils ménagers et bien plus encore, depuis une application mobile ou une commande vocale. Les derniers développements en matière de maison intelligente permettent également de personnaliser les paramètres de chaque pièce pour répondre aux préférences individuelles de chaque membre de la famille.

#### **4.3-Les usines et la fabrication intelligente :**

La fabrication intelligente utilise des capteurs connectés à l'Internet des objets pour surveiller et améliorer les performances des machines de production. Ces données sont collectées pour améliorer la sécurité des travailleurs et préserver l'environnement.

#### **4.4- Le système de santé électronique :**

La santé est l'un des secteurs les plus impactés par la révolution des objets connectés. La santé connectée regroupe l'ensemble des moyens et services en lien avec la santé qui utilisent les nouvelles technologies. L'internet des objets trouve donc tout son intérêt dans le domaine médical, permettant ainsi d'améliorer les soins et le développement de la médecine.

#### **4.5- Les villes intelligentes :**

Une ville intelligente (ou smart city) est une ville qui utilise les technologies de l'information et de la communication (TIC) pour améliorer les services urbains ou réduire les coûts. Les technologies utilisées incluent la surveillance de la ville, la gestion des déchets, la mobilité urbaine, l'éclairage public, l'efficacité énergétique et la sécurité publique. Certaines des villes les plus avancées en matière de ville intelligente sont Séoul, New York, Tokyo, Shanghai, Singapour, Amsterdam et Dubaï.

# conclusion

Ce chapitre explore l'essence de l'IoT, en présentant une introduction captivante, une rétrospective historique et les fonctionnalités clés qui en font une révolution technologique majeure.

# CHAPITRE 3

## ETAT DE L'ART

# Introduction

Dans ce chapitre, nous allons présenter les systèmes les plus populaires sur le marché pour le contrôle à distance des maisons. Nous les évaluerons en termes de prix, d'efficacité et d'équipements afin d'aider à faire le meilleur choix. Ensuite, nous procéderons à une comparaison entre les différentes techniques citées.

## 1-Etude de l'existant :

Dans cette partie, nous allons examiner plusieurs applications qui ont des objectifs similaires aux nôtres. Étant donné que la plupart des applications de domotique sont payantes, nous allons présenter trois exemples de systèmes : Zennio Z41 et Coviva Hager et KNX Schneider, afin de mieux comprendre les différences et les similitudes entre eux.

### 1.1- Zennio Z41 :

Zennio Z41 est qu'il s'agit d'un panneau capacitif tactile qui peut être facilement intégré dans n'importe quel type d'environnement.

Zennio Z41 est un système de domotique, également appelé contrôleur domestique intelligent, qui permet de contrôler à distance différents équipements électriques et électroniques d'une maison ou d'une entreprise, tels que les lumières, les stores, les systèmes de climatisation, etc. Le Zennio Z41 permet de gérer ces équipements à partir d'une application mobile ou d'un ordinateur, offrant ainsi une plus grande flexibilité et une facilité d'utilisation accrue. Il est fabriqué par la société espagnole Zennio, spécialisée dans les systèmes de domotique et les solutions de contrôle d'accès.

Caractéristiques:

- Ecran tactile de 4,1"
- 16 fonctions en toute libertés configurables à accès direct
- Régulation température ambiante
- Thermostats autonomes
- 4 pages dédiés, à paramétrier en tant que : minuteries, scénarios, régulation du climat
- 4 pages génériques : alarmes, Template, paramétrage, outils
- Application non payante pour les systèmes Apple pour l'accessibilité à distance.



Figure 3:Zennio Z41

### 1.2-Coviva Hager :

Coviva est une solution proposée par la société Hager pour la gestion intelligente des installations électriques résidentielles. Elle permet de connecter et de contrôler à distance différents équipements électriques tels que l'éclairage, le chauffage, la ventilation, les volets roulants, etc. à l'aide d'un smartphone, d'une tablette ou d'un ordinateur. Coviva permet également de programmer des scénarios d'utilisation pour une gestion économe de l'énergie et un confort accru.

Caractéristiques:

- Interface utilisateur conviviale et intuitive
- Possibilité de personnalisation et d'ajout de nouvelles fonctions
- Compatible avec les protocoles de communication les plus courants tels que Zigbee, Z-Wave, KNX, etc.
- Alarme SEPIO (déTECTEURS de mouvement, détECTEURS de gaz fumés ou d'inondations, détECTEUR d'ouverture capTEURS de déplacements, sirènes, ...)
- Système radio pour éclairage et ouverture

- Chauffage d'eau chaude et électrique : piloter le chauffage en local à distance ; gestion par radiateur ou par zone.
- Chauffage d'eau chaude et électrique : piloter le chauffage en local à distance ; gestion par radiateur ou par zone.

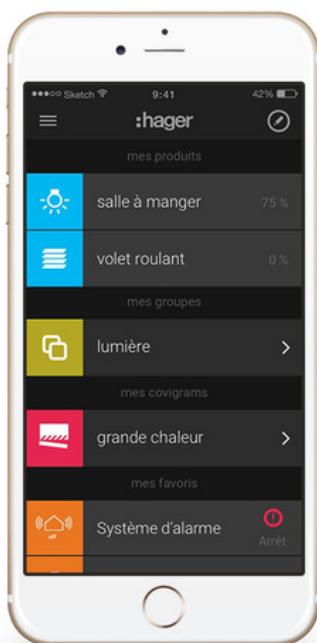


Figure 4:Coviva Hager

### 1.3-KNX Schneider :

KNX Schneider est un système de gestion de bâtiments automatisé développé par l'entreprise Schneider Electric. Le système utilise la technologie KNX (Konnex) pour permettre la communication entre différents équipements électriques et électroniques dans une maison ou un bâtiment.

Le système KNX Schneider permet de contrôler diverses fonctions telles que l'éclairage, le chauffage, la climatisation, les stores et volets, les systèmes de sécurité, etc. Le système est très flexible et peut être configuré pour répondre aux besoins spécifiques de chaque bâtiment.

En outre, Schneider Electric propose une large gamme d'équipements KNX compatibles avec le système, notamment des capteurs, des actionneurs, des thermostats, des interrupteurs et des boutons poussoirs, des détecteurs de mouvement, des caméras de surveillance, etc.

Le système KNX Schneider est une solution performante pour améliorer l'efficacité énergétique et la sécurité d'un bâtiment, tout en offrant un confort accru aux occupants.

Caractéristiques :

- Possibilité de réaliser jusqu'à 20% d'économies chaque année sur la climatisation et le chauffage
- L'éclairage peut être optimisé en évitant le gaspillage de lumière inutile, notamment dans les salles inoccupées ou lors d'oublis d'extinction des lumières. Grâce à des capteurs de présence et des systèmes de commande automatisés, il est possible de réguler efficacement l'éclairage en fonction des besoins réels des occupants, ce qui permet de réduire significativement la consommation d'énergie et les coûts associés,
- Les volets et les stores peuvent s'adapter aux conditions extérieures grâce à un système automatisé qui les fait réagir aux rayons du soleil, au vent ou à la lumière. Cela permet de réguler la température et la luminosité à l'intérieur du bâtiment de manière efficace et économique.



Figure 5:KNX Schneider

## 2-Analyses et Solution propose :

Caractéristiques	Zennio Z41	Coviva Hager	KNX Schneider
Environnement	Pro	Résidentiel	Résidentiel et Pro
Ecran tactile	Oui	Non	Non
Prix	490,90 \$	349 \$	455,60 \$
Température	Oui	Oui	Oui

<b>Humidité</b>	Oui	Non	Non
<b>Volets</b>	Oui	Oui	Oui
<b>Eclairage Auto</b>	Oui	Oui	Oui
<b>Régulation Climat</b>	Oui	Oui	Oui
<b>Volume de Fenêtre</b>	Oui	Oui	Oui
<b>Application Mobile</b>	Oui	Oui	Non
<b>Multiplateforme</b>	Non	Oui	Non

**Tableau 2:comparative entre des systèmes existants.**

Notre solution s'inscrit dans la même lignée que les exemples concrets que nous avons évoqués, mais elle se distingue par des procédés différents et une approche novatrice. Nous avons mené une analyse approfondie des applications similaires sur le marché, ce qui nous a permis de mettre en évidence des points forts et des faiblesses de ces solutions existantes. Grâce à cette démarche, nous avons identifié des opportunités d'amélioration pour notre propre projet et nous sommes en mesure de proposer une solution encore plus performante et adaptée aux besoins spécifiques de nos clients. Nous sommes convaincus que notre remède répondra aux attentes les plus exigeantes et apportera une valeur ajoutée significative à tous ceux qui l'adopteront.

# Conclusion

Grâce à cette section, nous avons pu établir un cadre clair pour notre programme et le situer dans son contexte global. Cette analyse nous a été d'une grande utilité pour avancer dans le développement de notre projet. Nous sommes désormais en mesure de passer à l'étape suivante qui consiste à sélectionner les équipements les plus adaptés à nos besoins, ce que nous aborderons dans la partie suivante. Cette approche méthodique et rigoureuse nous permettra d'optimiser notre solution et de répondre aux attentes les plus exigeantes de nos utilisateurs.

# CHAPITRE 4

## CHOIX DE MATERIEL

# Introduction

Au cours de ce chapitre, nous avons examiné différentes alternatives de cartes programmables ainsi que leurs spécifications techniques, tout en étudiant également les composants du système pertinents pour notre projet, afin de choisir celle qui répond le mieux aux exigences de notre projet.

## 1- Choix de la carte programmable :

### 1.1- Carte RASPBERRY\_PI :

La carte Raspberry Pi est un petit ordinateur monocarte qui peut rappeler la carte Arduino, également largement utilisée dans les projets électroniques open-source. La carte Raspberry Pi a été publiée sous une licence open-source, permettant aux utilisateurs de modifier et d'améliorer le design de la carte pour leurs propres projets.

Physiquement, la carte Raspberry Pi est une carte-mère seule avec un processeur ARM11 à 700 MHz de fréquence. Elle dispose de 1 ou 4 prises USB, d'une prise RJ45 pour la connexion Ethernet et de 256 Mo de RAM pour la version originale, bien que les versions plus récentes disposent de 2 Go de RAM. Cette configuration matérielle permet à la carte Raspberry Pi d'être utilisée pour une variété de projets, allant de la robotique à la domotique en passant par les projets d'automatisation industrielle.



Figure 6:Carte RASPBERRY\_PI

## 1.2- La cartes FPGA :

Les cartes FPGA sont des composants électroniques programmables et reconfigurables qui sont utilisés pour la conception de circuits intégrés. Elles sont construites à partir de matrices de portes programmables, de blocs de mémoire et de circuits de communication, qui peuvent être programmés pour effectuer des fonctions logiques spécifiques.

Contrairement aux circuits intégrés traditionnels, les cartes FPGA peuvent être reprogrammées à tout moment pour répondre aux besoins spécifiques des concepteurs de circuits. Cette flexibilité permet aux ingénieurs et aux développeurs de concevoir des circuits personnalisés pour des applications spécifiques.

Les cartes FPGA sont utilisées dans de nombreux domaines, tels que la conception de systèmes informatiques, la communication, la vidéo numérique et la cryptographie. Elles offrent une grande puissance de traitement et une grande flexibilité, ce qui les rend idéales pour les applications nécessitant des mises à jour fréquentes ou des exigences de traitement complexes.

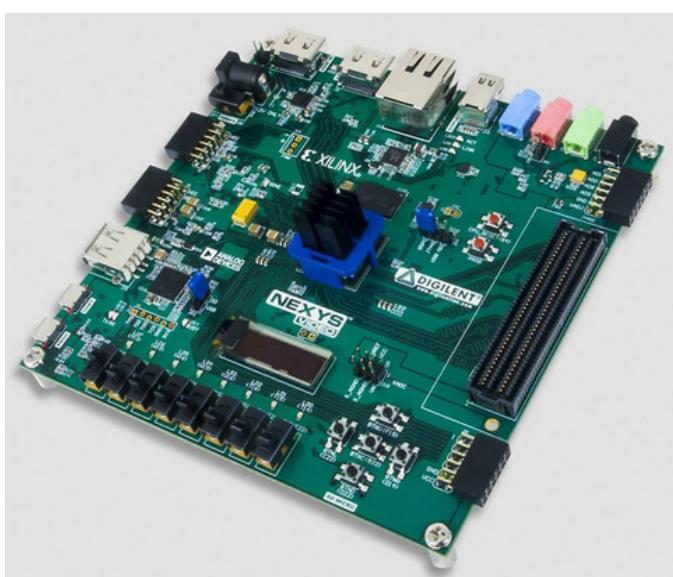


Figure 7:La cartes FPGA

### 1.2.1- Spécifications techniques :

Les cartes FPGA sont des circuits électroniques programmables qui sont constitués de trois éléments principaux. Tout d'abord, il y a la CLB, ou "Configurable Logic Bloc", qui est une matrice de blocs logiques configurables permettant d'effectuer des opérations logiques complexes. Ensuite, il y a des blocs d'entrées/sorties reconfigurables, qui permettent à la carte FPGA de communiquer avec d'autres composants électroniques. Enfin, il y a un réseau d'interconnexions programmables, qui permet de connecter les différents blocs et éléments de la carte.

Il existe plusieurs fabricants de cartes FPGA, tels que Actel, Xilinx et Altera. Ces cartes sont très utilisées dans de nombreux domaines, notamment dans les systèmes embarqués, la conception de circuits électroniques et la recherche scientifique. Les avantages des cartes FPGA sont leur grande flexibilité, leur puissance de calcul et leur capacité à être reprogrammées pour de nouvelles applications. :

#### FPGA Spartan 3E :

La carte FPGA Spartan-3E-Digilent est un dispositif de développement autonome destiné à offrir une plate-forme très performante pour la création de conceptions ciblant l'FPGA Spartan-3E de Xilinx. Cette carte dispose d'une puce FPGA Spartan-3E qui offre une capacité de 500 000 portes logiques, avec un processeur RISC de 32 bits et des interfaces DDR. Elle est également dotée d'une mémoire flash sur Xilinx, d'interfaces USB et JTAG pour la programmation, ainsi que de nombreuses options de configuration FPGA via le module intégré Strata-Flash et la mémoire flash série de ST-Micro-electronics. Cette carte est compatible avec toutes les versions des logiciels Xilinx ISE, y compris la version gratuite Web-Pack.

Elle est livrée avec un câble USB et un bloc d'alimentation pour le développement, sans coûts cachés. L'FPGA Spartan-3E est également compatible avec le kit de développement Micro-Blaze et Pico-Blaze pour les conceptions embarquées.

### 1.2.2- Caractéristiques :

#### FPGA Xilinx-XC3S500E

- Mémoire flash sur plateforme Xilinx-XCF04 pour stocker les configurations FPGA.
- 32Mo Micron DDR-SDRAM Ø 16Mo Numonyx-Strata-Flash Ø 2Mo de mémoire flash série de ST-Micro-electronics
- Alimentation à technologie linéaire .
- Circuit intégré de gestion de l'alimentation Texas-Instruments-TPS75003 .
- SMSC-LAN-83C185-Ethernet-PHY .

### 1.3- Carte ARDUINO :

La carte Arduino est une carte électronique open-source, dont l'architecture est librement disponible, permettant aux utilisateurs de construire des systèmes électroniques personnalisés pour un large éventail d'applications, y compris la domotique, la robotique, l'automatisation industrielle, et bien plus encore. Elle est équipée d'un microcontrôleur programmable qui peut être utilisé pour lire et produire des signaux électriques, permettant aux utilisateurs de contrôler des périphériques tels que des capteurs, des actionneurs, des moteurs, des afficheurs et des écrans tactiles. La carte Arduino est facile à utiliser, peu coûteuse et largement disponible, ce qui en fait une solution idéale pour les projets électroniques DIY.

### 1.3.1-Arduino Méga 2560:

La carte Arduino-Méga-2560 est une carte électronique polyvalente qui utilise le microcontrôleur ATmega-2560 pour offrir des capacités d'E/S numériques et analogiques étendues. Avec ses 54 broches numériques, 16 entrées analogiques et une gamme de connecteurs d'entrée/sortie, elle est idéale pour une variété d'applications, notamment la robotique, l'automatisation industrielle, la surveillance de l'environnement et bien d'autres encore. Elle est facile à alimenter avec une batterie ou un adaptateur, et peut également être connectée directement à un ordinateur via un câble USB pour une programmation simple et pratique.

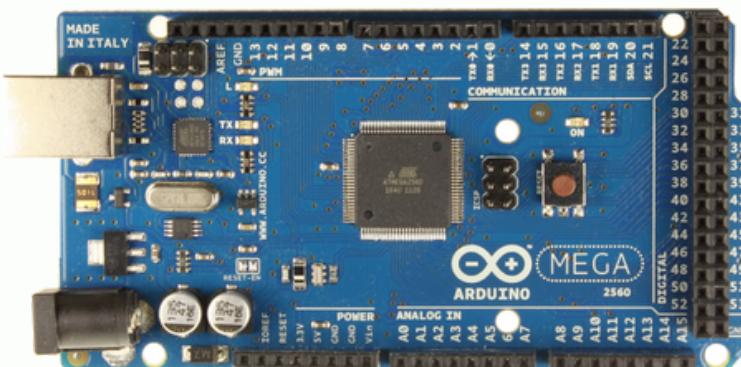


Figure 8:Arduino Méga 2560

### 1.3.2-Caractéristiques:

La carte Arduino Mega 2560 est une plate-forme de développement électronique qui est basée sur le microcontrôleur ATmega2560 d'Atmel. Elle offre 54 broches numériques d'entrée/sortie, dont 15 peuvent être utilisées comme sorties PWM et 16 entrées analogiques. La carte fonctionne sous une tension de 5V et peut être alimentée par une source de 7 à 12V. Elle dispose également d'une mémoire flash de 128 Ko, d'une SRAM de 8 Ko et d'une EEPROM de 4 Ko. Le taux d'horloge de la carte est de 16 MHz.

La carte est livrée avec un câble USB pour la programmation et la connexion à un ordinateur, ainsi qu'un connecteur d'alimentation et un connecteur ICSP pour la programmation en circuit. Elle est compatible avec l'environnement de développement intégré (IDE) Arduino et peut être programmée en utilisant la bibliothèque Arduino.

## 2- Choix de la carte programmable :

Nous allons réaliser un comparatif entre trois types de cartes selon différents critères pour déterminer la plus adaptée à nos besoins. Nous allons notamment nous baser sur le prix, l'environnement de développement intégré (IDE) et le mode de programmation. Ce tableau comparatif nous permettra de choisir la carte la plus appropriée pour notre projet.

Caractéristiques	Raspberry Pi	FPGA Spartan 3E	Arduino Mega 2560
Type de dispositif programmable	Microcontrôleur	FPGA	Microcontrôleur
Processeur	Broadcom ARM	Xilinx Spartan 3E	Atmel AVR
Fréquence du processeur	1.2 GHz	50 MHz	16 MHz
RAM	Jusqu'à 8 Go	512 Ko	8 Ko

Entrées/Sorties	GPIO, USB, HDMI, Ethernet, etc.	Broches d'E/S configurables par l'utilisateur	Broches d'E/S
Langages de programmation	Python, C, C++, Scratch, etc.	VHDL, Verilog, etc.	Arduino C/C++
Environnement de développement intégré (IDE)	Raspberry Pi OS, Thonny, etc.	Xilinx ISE, Vivado, etc.	Arduino IDE
Coût	Moyen-élévé(50\$)	Élevé(400\$)	Faible(16\$)

**Tableau 3: Tableau des cartes programmables..**

La carte Arduino offre une solution peu coûteuse et facile à utiliser pour les projets électroniques DIY. Son IDE convivial et son langage de programmation simplifient la tâche de travailler avec des microcontrôleurs, ce qui la rend populaire auprès des étudiants, des enseignants et des amateurs. En outre, la grande communauté d'utilisateurs de la carte Arduino offre un support et des ressources supplémentaires pour les débutants. Après avoir examiné différentes options, nous avons choisi la carte Arduino Mega 2560 pour notre projet en raison de ses nombreuses broches d'entrée/sortie, de sa mémoire flash importante et de sa fréquence d'horloge élevée, ce qui en fait une solution puissante pour les projets de grande envergure.

### 3-Choix des composants du système:

Maintenant que nous avons opté pour la carte Arduino Mega 2560, il est temps de sélectionner le matériel compatible qui répondra à nos besoins spécifiques pour notre projet.

#### 3.1-Shield Ethernet W5100 :

Le Shield Ethernet W5100 est un module compatible avec Arduino qui intègre le contrôleur Ethernet Wiznet W5100. Il permet à une carte Arduino de se connecter à Internet ou à un réseau local en utilisant le protocole Ethernet.

Le Wiznet W5100 est une puce de contrôleur Ethernet largement utilisée qui fournit une pile de protocoles TCP/IP pour la communication réseau. En utilisant le Shield Ethernet W5100, une carte Arduino acquiert la capacité d'envoyer et de recevoir des données via une connexion Ethernet.

##### 3.1.1-caractéristiques:

- Prise en charge des connexions Ethernet 10/100.
- Intégration d'une pile de protocoles TCP/IP pour gérer la communication réseau.
- Compatibilité avec l'Arduino Uno et d'autres cartes compatibles Arduino.
- Connexion facile en utilisant le format de shield Arduino.
- Emplacement intégré pour carte MicroSD pour stocker des données.



Figure 9:Shield Ethernet W5100

### 3.2-Module de Relais à 8 canaux:

Le module de Relais à 8 canaux est une carte d'interface de relais qui permet de contrôler différents équipements avec une tension allant jusqu'à 220 V. Il peut être contrôlé directement par un µ-contrôleur comme l'Arduino, l'A.V.R, le P.I.C, l'A.R.M ou l'A.P.I. Ce module de relais est activé par une tension de 5 V et dispose d'une LED rouge qui indique l'état de travail.

Le Relais à 8 canaux est très utilisé dans le secteur industriel, le contrôle M.C.U, le contrôle P.L.C et la domotique. Dans notre projet, nous avons choisi d'utiliser un module de relais à 8 canaux pour contrôler l'éclairage, la climatisation, le chauffage et la ventilation. Ce module est facile à intégrer avec l'Arduino et permet un contrôle précis et fiable des équipements électriques.

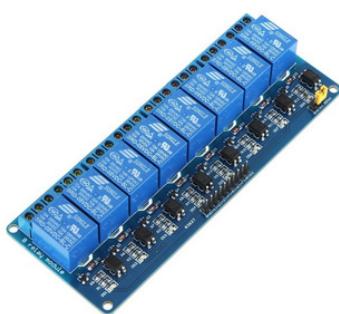


Figure 10:Module de Relais à 8 canaux

### 3.3-ventilateur 12V:

Un ventilateur 12V est un appareil électronique alimenté par une tension de 12 volts et conçu pour générer un flux d'air. Il est polyvalent et utilisé dans divers domaines tels que l'automobile, l'informatique et les systèmes de refroidissement électronique.

### 3.3.1-caractéristiques:

- Alimentation : Fonctionne avec une tension de 12 volts.
- Taille et forme : Disponible dans différentes tailles et formes pour s'adapter à différentes applications.
- Flux d'air : Génère un débit d'air mesuré en m<sup>3</sup>/min ou CFM.
- Vitesse de rotation : Peut avoir plusieurs vitesses de rotation réglables en RPM.
- Niveau sonore : Émet un niveau sonore mesuré en décibels (dB).
- Contrôle et connectivité : Certains modèles offrent des fonctionnalités de contrôle de la vitesse ou une connectivité avec d'autres appareils.
- Durabilité : Construit pour être durable et résistant avec des matériaux robustes et des caractéristiques de protection.



Figure 11:ventilateur 12V

### 3.4-lampe:

Une lampe est un appareil d'éclairage utilisé pour illuminer un espace spécifique. Elle est généralement constituée d'une source de lumière, d'un abat-jour pour diffuser la lumière et d'un support ou d'un pied pour la maintenir en place. Les lampes sont disponibles dans une variété de styles et de designs pour s'adapter à différents décors et préférences esthétiques.

Elles peuvent être utilisées dans les maisons, les bureaux ou les espaces publics pour créer une ambiance lumineuse agréable et fonctionnelle. Certaines lampes offrent également des fonctionnalités supplémentaires, telles que des interrupteurs tactiles, des réglages d'intensité lumineuse ou des options de connectivité. Que ce soit pour éclairer une pièce, créer une atmosphère chaleureuse ou fournir un éclairage ciblé, une lampe est un élément essentiel de l'éclairage domestique et professionnel.



Figure 12:lampe

### 3.5-Servomoteurs:

Pour permettre la commande des volets des fenêtres, nous avons décidé d'utiliser des servomoteurs pour leur simplicité d'utilisation, leur flexibilité et leur capacité à fournir un couple acceptable. Les servomoteurs sont conçus pour actionner les parties mobiles d'un système et peuvent être facilement commandés en vitesse ou en position. Ces moteurs sont équipés d'un potentiomètre rotatif qui sert de détecteur de rotation, permettant ainsi une commande précise des mouvements des volets.

C'est un ensemble électronique et mécanique contenant :

- Un petit moteur à courant continu avec Un réducteur baissant la vitesse et hausser le couple 23.
- Un potentiomètre qui produit une tension qui est proportionnelle à l'angle de l'axe.
- Un axe sortant du boîtier avec différents bras ou roues
- Un mécanisme d'asservissement.

### 3.5.1-Caractéristiques:

Le servomoteur Tower Pro SG90 est un petit moteur électrique à courant continu utilisé pour les applications de contrôle de positionnement. Voici ses principales caractéristiques :

- Couple de sortie : 1,5 kg.cm
- Tension de fonctionnement : 4,8V à 6V
- Vitesse de fonctionnement : 0,12 s/60 degrés à 4,8V
- Angle de rotation : 180 degrés
- Poids : 9g
- Dimensions : 22mm x 11.5mm x 27mm
- Type de connecteur : JR (compatible avec les connecteurs Futaba)
- Contrôle de positionnement précis
- Faible consommation d'énergie
- Idéal pour les applications de contrôle de mouvement à petite échelle
- Compatible avec la plupart des microcontrôleurs, y compris les cartes Arduino et Raspberry Pi.



Figure 13: Servomoteurs

### 3.6-L'ESP8266 :

L'ESP8266 est un microcontrôleur à faible coût et à faible consommation d'énergie qui intègre la connectivité Wi-Fi. Il est largement utilisé dans le domaine de l'IoT (Internet des objets) en raison de ses performances fiables et de sa facilité d'utilisation. Voici une description brève de l'ESP8266 et de ses caractéristiques :

L'ESP8266 est un système-sur-puce (SoC) développé par le fabricant chinois Espressif Systems. Il combine un microcontrôleur 32 bits, une connectivité Wi-Fi intégrée, une mémoire flash et d'autres fonctionnalités nécessaires pour la communication sans fil.

#### 3.6.1-Caractéristiques:

- Connectivité Wi-Fi : L'ESP8266 offre une connectivité Wi-Fi intégrée, ce qui permet aux appareils de se connecter à des réseaux sans fil et d'échanger des données sur Internet.
- Microcontrôleur puissant : Il est basé sur une architecture 32 bits avec une fréquence d'horloge pouvant atteindre 160 MHz, ce qui lui confère une puissance de traitement suffisante pour de nombreuses applications IoT.

- Mémoire intégrée : L'ESP8266 est équipé d'une mémoire flash intégrée pour stocker le code du programme, les données et les paramètres de configuration.
- Interfaces d'E/S : Il dispose de broches d'E/S (entrée/sortie) pour interagir avec des composants externes tels que des capteurs, des actionneurs et d'autres périphériques.
- Protocoles de communication : L'ESP8266 prend en charge plusieurs protocoles de communication, tels que TCP/IP, UDP, HTTP et MQTT, permettant une intégration aisée avec les réseaux et les services IoT.
- Programmation facile : Il peut être programmé à l'aide de langages de programmation courants tels que C/C++ ou via des environnements de développement intégrés (IDE) tels que l'IDE Arduino.
- Faible consommation d'énergie : L'ESP8266 est conçu pour une faible consommation d'énergie, ce qui le rend adapté aux applications alimentées par batterie ou soucieuses de l'efficacité énergétique.
- Écosystème de développement : L'ESP8266 bénéficie d'une communauté active de développeurs et d'un vaste écosystème de bibliothèques et de ressources disponibles en ligne.



Figure 14:ESP8266

# conclusion

Dans cette section, nous avons procédé à une évaluation approfondie de différentes cartes programmables et des composants nécessaires, en analysant méticuleusement leurs fonctionnalités. Notre objectif était de choisir la solution la mieux adaptée à notre projet. Dans la suite de notre rapport, nous allons explorer en détail les besoins spécifiques des utilisateurs. Cette approche nous permettra d'acquérir une compréhension approfondie des objectifs du projet et de les analyser en profondeur, afin de développer une conception sur mesure répondant pleinement à ces besoins.

# CHAPITRE 5

## CONCEPTION, ANALYSE ET CONCEPTION

# introduction

Au cours de cette section, nous allons fournir des détails sur les exigences de l'utilisateur du système après avoir examiné les options de matériel dans la section précédente. Cette partie aura pour objectif de clarifier davantage l'objectif de notre projet, d'analyser les besoins fonctionnels et non fonctionnels ainsi que les acteurs impliqués. Par la suite, nous présenterons les diagrammes des cas d'utilisation, les scénarios et les diagrammes de classe qui décriront les fonctionnalités offertes par le système.

## 1-Identification des acteurs:

Les utilisateurs du système se divisent en deux catégories :

Les utilisateurs , qui sont les propriétaires de la maison et qui interagiront avec le système via des smartphones, tablettes, ordinateurs ou autres périphériques connectés. Ces utilisateurs pourront demander des services spécifiques à leur maison, tels que la gestion de l'éclairage, de la climatisation et des volets.

Les administrateurs, qui auront la responsabilité de gérer les comptes d'utilisateurs et de superviser les activités du système. Ces administrateurs seront en mesure d'ajouter ou de supprimer des utilisateurs, de réinitialiser les mots de passe, de définir les autorisations d'accès et de surveiller les activités des utilisateurs.

Chaque utilisateur du système aura un rôle bien défini et des autorisations spécifiques, afin de garantir que le système soit utilisé de manière appropriée et sécurisée. Les besoins fonctionnels et non fonctionnels de chaque utilisateur seront pris en compte dans la conception et la mise en œuvre de l'application.

Analyse des besoins:

## 2-Besoins fonctionnels:

Notre application de gestion de maison intelligente offrira une variété de fonctionnalités pour répondre aux besoins des différents utilisateurs :

### 2.1-Responsable de sécurité:

**Authentification:** l'utilisateur pourra se connecter à l'application en utilisant ses identifiants personnels.

**Contrôle des lampes:** l'utilisateur pourra allumer/éteindre les lumières de la maison à tout moment.

**Authentification:** l'utilisateur pourra se connecter à l'application en utilisant ses identifiants personnels.

**Gestion de l'aération et de la climatisation :** l'utilisateur pourra contrôler le système de ventilation et de climatisation et les régler en fonction de la température ambiante.

**Gestion des volets de fenêtre :** l'utilisateur pourra contrôler les volets de fenêtre .

**Surveillance des propriétés de la maison:** l'utilisateur pourra consulter les données de température, d'humidité, de gaz et de fumée détectées par les capteurs de la maison.

## 2.2-Responsable de direction(admin) :

**Authentification :** l'administrateur pourra se connecter à l'application en utilisant un compte dédié créé lors de l'installation de l'application.

**Gestion des comptes utilisateurs :** l'administrateur pourra créer, modifier et supprimer des comptes d'utilisateurs pour donner accès à l'application à d'autres personnes de la maison. Notre application offrira également une compatibilité avec une variété de dispositifs connectés pour permettre une gestion efficace et intuitive de la sécurité, du confort et de l'efficacité énergétique de la maison.

## 3-Besoins non fonctionnels:

Le projet présente plusieurs exigences dans sa phase de réalisation, notamment les besoins non fonctionnels suivants :

**Accessibilité :** L'application doit être facilement accessible et utilisable pour tous les utilisateurs sans aucun prérequis particulier. Elle doit être intuitive et conviviale pour une expérience utilisateur agréable.

**Simplicité :** L'application doit être une application à interface utilisateur légère qui peut être utilisée facilement par tout utilisateur, avec ou sans connaissance technique.

Un guide d'utilisation complet doit être fourni pour aider les utilisateurs à comprendre les fonctionnalités de l'application.

**Maintenance** : Le code source du programme doit être bien documenté avec des commentaires appropriés pour permettre une maintenance facile et rapide par les développeurs, y compris ceux qui n'ont pas travaillé sur le projet initial.

**Performances** : Les performances de l'application sont primordiales, toute instruction traitée ne doit pas prendre plus de 3 secondes pour s'exécuter. Cela garantira que l'application est rapide et efficace, ce qui contribuera à la satisfaction des utilisateurs.

#### 4-Spécification des besoins:

Au cours de cette phase, nous avons utilisé le formalisme UML pour représenter notre système, en créant le diagrammes de cas d'utilisation pour donner une vue d'ensemble des fonctionnalités proposées, ainsi que des diagrammes de séquence pour chaque cas d'utilisation, afin de décrire les différents scénarios possibles. Ces diagrammes nous permettent de mieux comprendre les interactions entre les différents acteurs du système et les différentes fonctionnalités proposées, et de mieux structurer notre projet en fonction des besoins de nos utilisateurs.

##### 4.1-Description textuelle du cas d'utilisation "S'authentifier "

Cas d'utilisation	S'authentifier
Acteur	Utilisateur et administrateur

<b>Objectif</b>	Permet à un utilisateur ou à l'administrateur d'accéder à son propre espace.
<b>Pré condition</b>	L'utilisateur possède un compte
<b>Post condition</b>	Permet à l'acteur d'accéder à son propre espace
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système invite l'acteur à entrer son login et son mot de passe.</li> <li>2. L'acteur saisit le login et le mot de passe</li> <li>3. Le système vérifie les paramètres</li> <li>4. Le système affiche l'espace correspondant à l'acteur.</li> <li>5. L'instance de cas d'utilisation se termine</li> </ol>

Tableau 4:Description textuelle du cas d'utilisation "S'authentifier "

#### 4.2- Description textuelle du cas d'utilisation "s'inscrire "

<b>Cas d'utilisation</b>	s'inscrire
<b>Acteur</b>	Utilisateur et administrateur
<b>Objectif</b>	Permet à un utilisateur d'être un membre
<b>Pré condition</b>	Consulter l'application
<b>Post condition</b>	Créer un compte dans l'application

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. l'utilisateur choisit de s'inscrire</li> <li>2. Le système affiche le formulaire correspondant</li> <li>3. l'utilisateur remplit le formulaire</li> <li>4. Le système vérifie les données saisies</li> <li>5. Le système affiche l'espace du membre</li> <li>6. L'instance de cas d'utilisation se termine</li> </ol>
-------------------------	---

Tableau 5:Description textuelle du cas d'utilisation "s'inscrire "

#### 4.3- Description textuelle du cas d'utilisation "Gérer lumière "

<b>Cas d'utilisation</b>	Gérer lumière
<b>Acteur</b>	Utilisateur et administrateur
<b>Objectif</b>	Permet à un utilisateur ou à l'administrateur de contrôler la lumière dans son propre espace
<b>Pré condition</b>	S'authentifier
<b>Post condition</b>	Allumer ou éteindre la lampe
<b>Scénario nominal</b>	l'utilisateur clique sur le bouton on s'il souhaite allumer la lumière ou sur le bouton off pour éteindre la lumière.

Tableau 6:Description textuelle du cas d'utilisation "Gérer lumière "

#### 4.4-Description textuelle du cas d'utilisation "Gérer le volet "

<b>Cas d'utilisation</b>	Gérer le volet
<b>Acteur</b>	Utilisateur et administrateur
<b>Objectif</b>	Permettre à un utilisateur ou à l'administrateur de gérer le volet de fenêtre dans leur propre espace personnel.
<b>Pré condition</b>	S'authentifier
<b>Post condition</b>	Ferme ou ouvrir les fenêtres
<b>Scénario nominal</b>	l'utilisateur clique sur le bouton on s'il souhaite ouvrir la fenêtre ou sur le bouton off pour fermer la fenêtre

**Tableau 7: Description textuelle du cas d'utilisation "Gérer le volet "**

#### **4.5- Description textuelle du cas d'utilisation "Gestion d'aération "**

<b>Cas d'utilisation</b>	Gestion d'aération
<b>Acteur</b>	Utilisateur et administrateur
<b>Objectif</b>	Permettre à un utilisateur ou à l'administrateur de gérer l'aération dans leur propre espace personnel.
<b>Pré condition</b>	S'authentifier
<b>Post condition</b>	arrêter ou mettre en marche l'aération
<b>Scénario nominal</b>	l'utilisateur clique sur le bouton on s'il souhaite mettre en marche l'aération ou sur le bouton off pour arrêter le déclenchement de l'aération

**Tableau 8: Description textuelle du cas d'utilisation "Gestion d'aération "**

#### **4.6-Description textuelle du cas d'utilisation "Gestion de climatisation "**

<b>Cas d'utilisation</b>	Gestion de climatisation
<b>Acteur</b>	Utilisateur et administrateur
<b>Objectif</b>	Permettre à un utilisateur ou à l'administrateur de gérer l'aération dans leur propre espace personnel.
<b>Pré condition</b>	S'authentifier
<b>Post condition</b>	arrêter ou mettre en marche la climatisation
<b>Scénario nominal</b>	l'utilisateur clique sur le bouton on s'il souhaite mettre en marche la climatisation ou sur le bouton off pour arrêter le déclenchement de la climatisation

**Tableau 9: Description textuelle du cas d'utilisation "Gestion de climatisation "**

## 4.7-Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation en UML sont utilisés pour modéliser le comportement d'un système et capturer ses exigences. Ils mettent en évidence les interactions entre le système et ses utilisateurs, en identifiant les acteurs et en définissant leurs rôles. Ils permettent également de spécifier les obligations de chaque utilisateur dans le système. Ci-dessous se trouve une figure représentant un diagramme de cas d'utilisation avec les différents utilisateurs, leurs rôles et les obligations qui leur sont assignées dans nos systèmes.

### 4.7.1- Diagramme de cas d'utilisation du système:

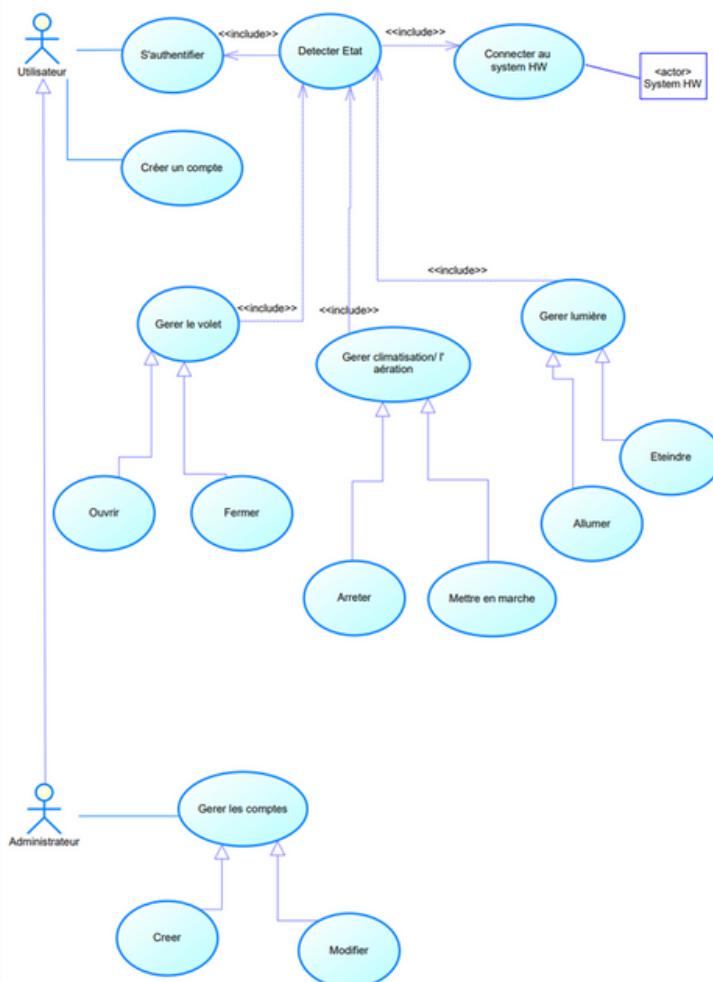


Figure 15: Diagramme de cas d'utilisation du système

## 4.8- Diagramme de séquence:

Un diagramme de séquence est utilisé dans le domaine du génie logiciel pour représenter les interactions de processus organisées dans une séquence temporelle. Il décrit les processus impliqués et la séquence de messages échangés entre ces processus pour exécuter une fonctionnalité donnée.

### 4.8.1- l'authentification:

La figure ci-dessous présente un diagramme de séquence qui permet de mieux comprendre le processus d'authentification.

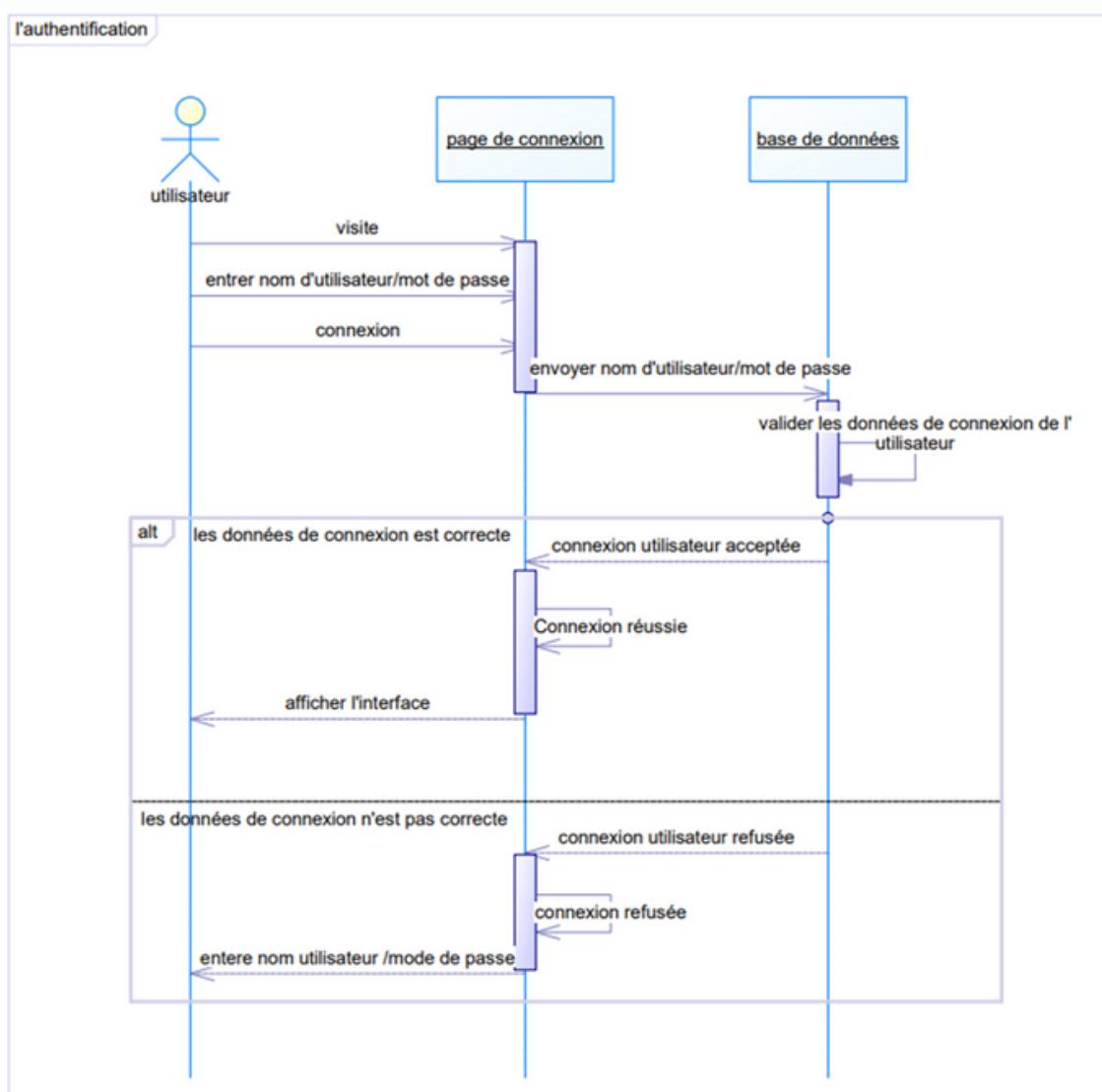


Figure 16:diagramme de séquence "l'authentification"

### 4.8.2- Allumage de lampe:

La figure ci-dessous présente un diagramme de séquence qui facilite la compréhension du processus d'allumage d'une lampe.

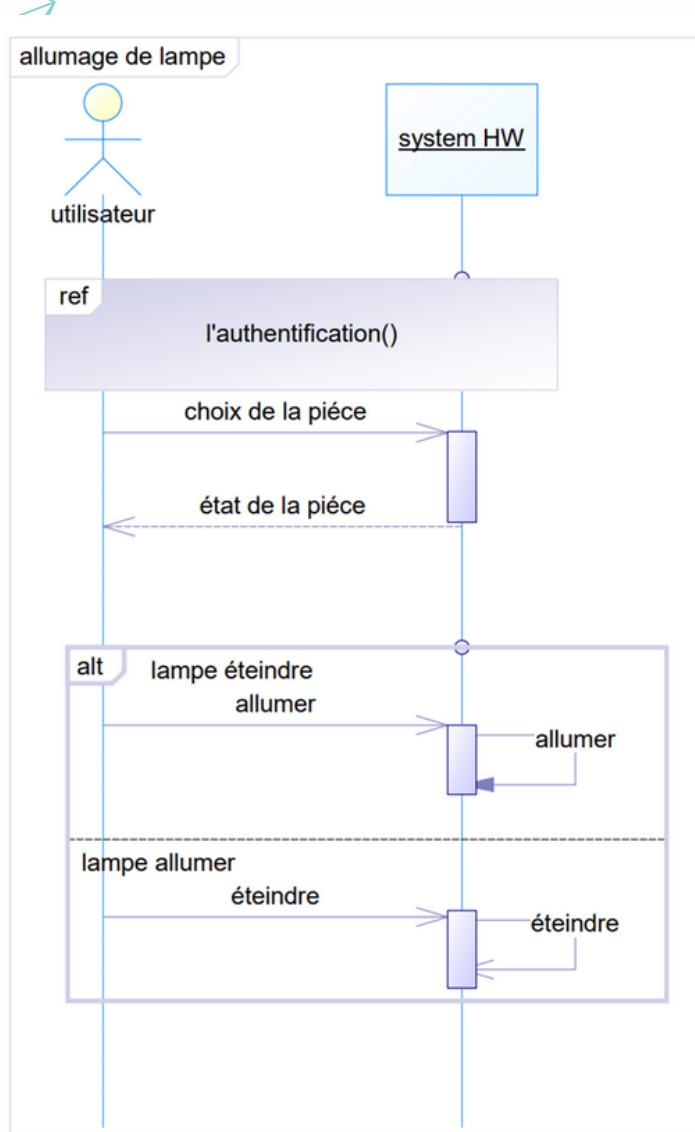


Figure 17:diagramme de séquence "Allumage de lampe"

### 4.8.3- Gestion d'aération et de climatisation :

La figure ci-dessous présente un diagramme de séquence qui facilite la compréhension du processus de Gestion d'aération et de climatisation .

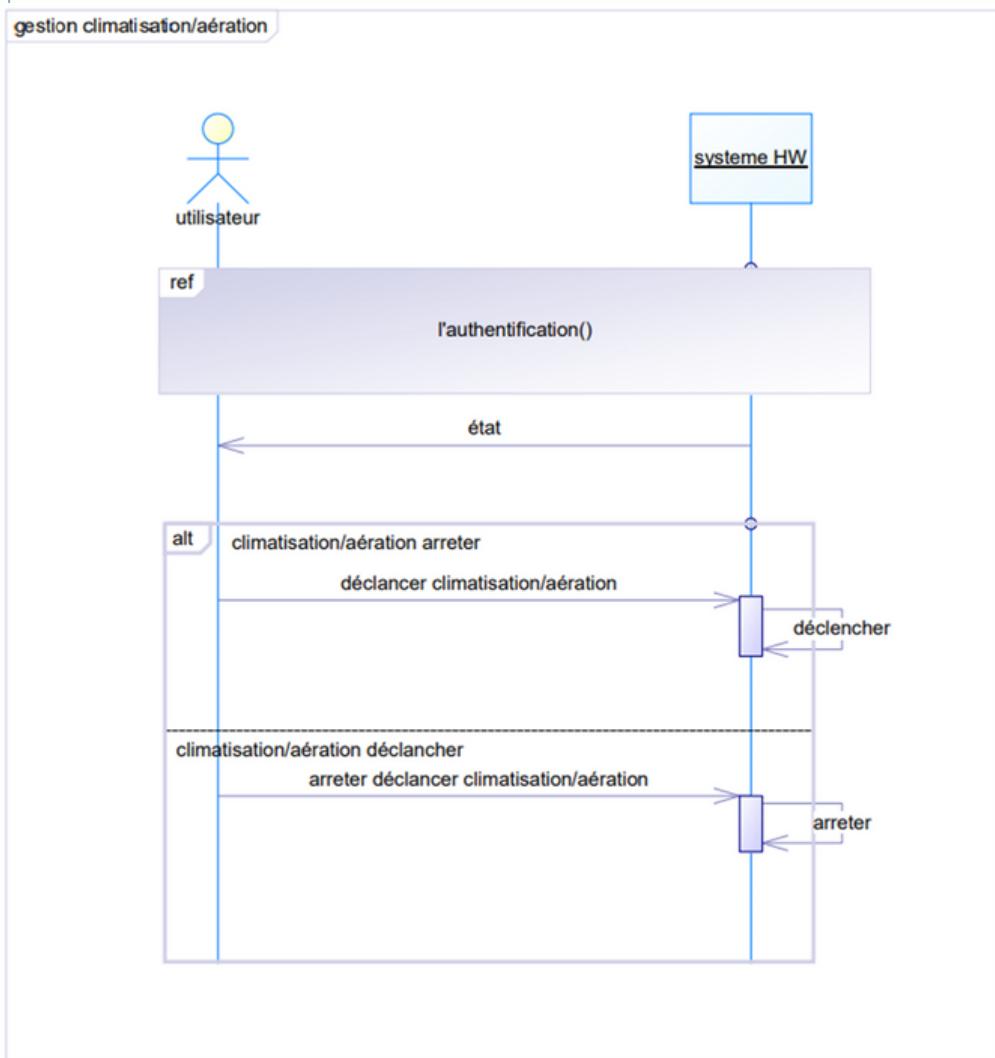


Figure 18:diagramme de séquence " Gestion d'aération et de climatisation"

#### 4.8.4-Gestion de volet:

Le diagramme de séquence représente de manière concise et visuelle le processus de gestion des volets. Voici une description du processus :

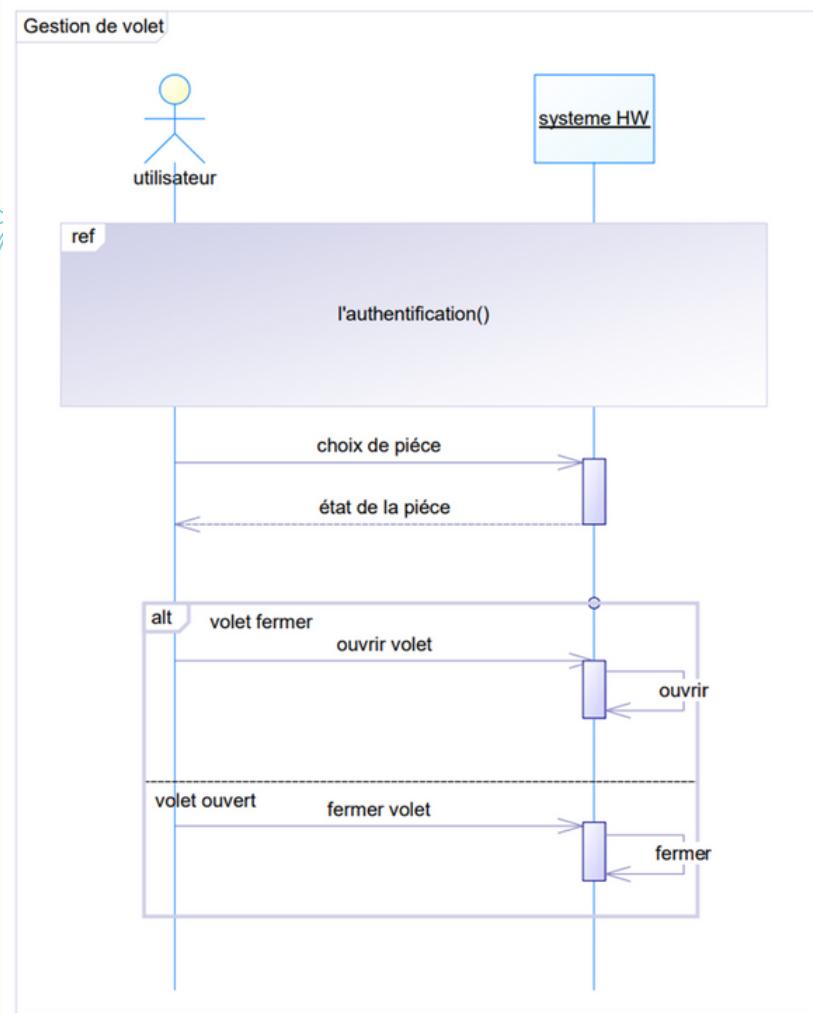


Figure 19:diagramme de séquence " Gestion de volet"

#### 4.9-Diagramme de classe :

Le diagramme de classes est une représentation graphique utilisée dans le langage de modélisation unifié (UML) pour décrire la structure statique d'un système. Il permet de visualiser et de modéliser les différentes classes du système, en montrant leurs attributs, leurs opérations (ou méthodes) et les relations entre les objets.

En étudiant les systèmes existants et en analysant les diagrammes de cas d'utilisation, nous avons identifié les classes principales qui sont présentées dans la figure ci-dessous. Cette représentation nous permet d'avoir une vision plus claire du système étudié. Notre application comprend les classes suivantes :

- **Classe Utilisateur** : contient toutes les informations sur les utilisateurs de l'application.
- **Classe Administrateur** : Représente un utilisateur ayant des priviléges étendus dans l'application, Ils peuvent effectuer des tâches telles que la création et la suppression des utilisateurs.
- **Maison** : La classe "Maison" est une entité qui contient toutes les informations et fonctionnalités liées à une maison contrôlable.
- **Application** : La classe "Application" est une entité logicielle qui facilite l'interaction entre les utilisateurs et un système matériel spécifique. Elle fournit une interface utilisateur intuitive et regroupe les fonctionnalités essentielles pour répondre aux besoins des utilisateurs. Son rôle principal est de permettre aux utilisateurs de contrôler, et interagir avec le système matériel de manière conviviale.
- **Classe Lampe** : La classe "Lampe" regroupe toutes les informations pertinentes concernant une lampe spécifique dans le système. Elle contient des détails tels que l'état de la lampe (allumée ou éteinte)
- **Classe Volet** : La classe "Volet" rassemble toutes les informations essentielles concernant un volet spécifique dans le système. Elle englobe des détails tels que l'état actuel du volet (ouvert ou fermé).
- **Classe Climatisateur** : La classe "Climatiseur" encapsule toutes les informations essentielles relatives à un climatiseur spécifique dans le système. Elle comprend des détails tels que l'état de fonctionnement (allumé ou éteint).
- **Classe Aération** : La classe "Aération" rassemble toutes les informations pertinentes concernant le système d'aération spécifique dans le système. Elle comprend des détails tels que l'état de fonctionnement (activé ou désactivé).

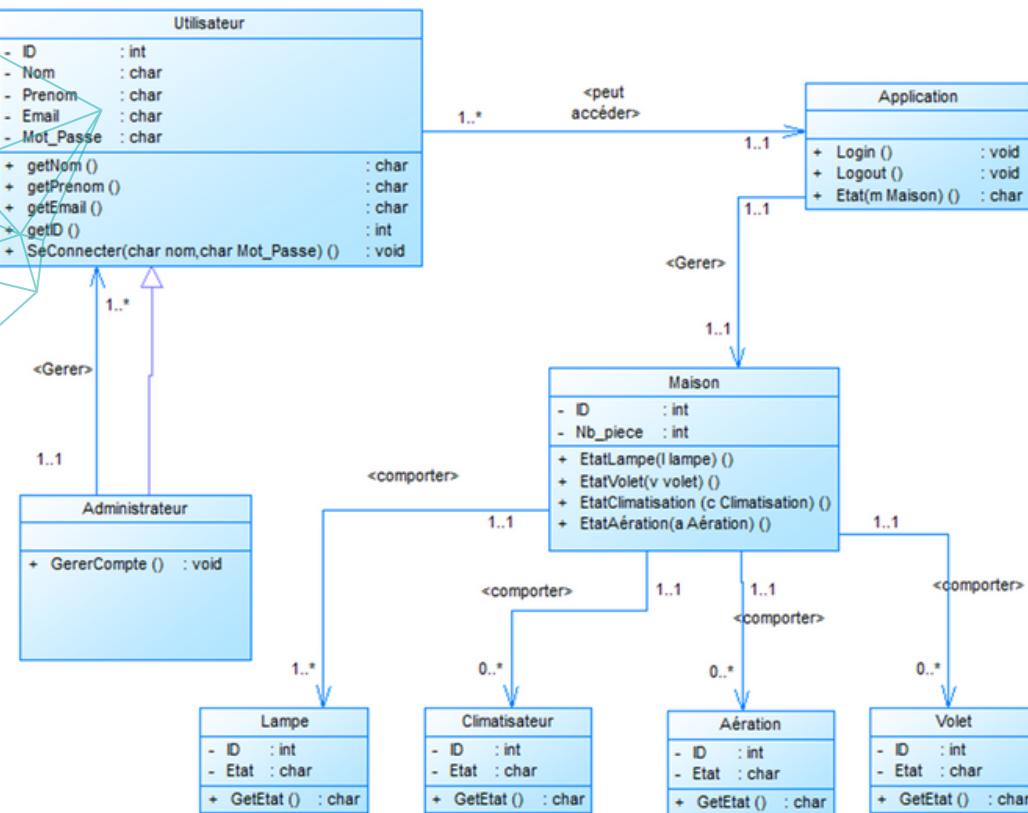


Figure 20:diagramme de classe

#### 4.10-Diagramme de flux de données :

Comme la programmation sur ARDUINO est séquentielle et ne suit pas la programmation orientée objet, nous avons la liberté de choisir une représentation pour ce paquetage. Nous avons choisi d'utiliser un diagramme de contexte, également appelé un diagramme de flux de données, niveau 0, comme représentation pour la solution.

Ce choix de représentation est dû au fait que le diagramme de contexte est plus simple et plus facile à comprendre que les diagrammes UML plus complexes. Il permet également de visualiser les différents flux de données et les interactions entre les différents composants du système.

Le diagramme de contexte, niveau 0, que nous avons représenté ci-dessous , nous permettra de visualiser les différents acteurs impliqués dans le système, les flux de données entrants et sortants, ainsi que les différents processus et sous-processus impliqués dans l'application.

La figure ci-dessous illustre le flux de données entre le système Arduino et les différents composants du système :

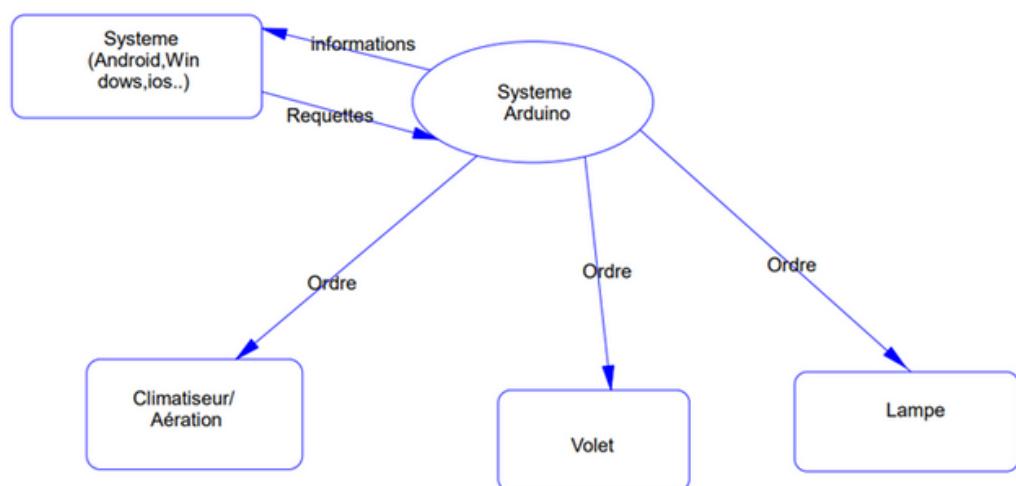


Figure 21:Diagramme de flux de données

# conclusion

Dans cette section, nous avons réalisé une analyse et une conception détaillée de notre système. Cela inclut la compréhension des interactions entre l'utilisateur et les différents composants du système à travers l'utilisation de cas d'utilisation et de scénarios. Cette approche nous permet de mieux comprendre le fonctionnement de notre application et de décrire les flux d'actions entre les utilisateurs et les différentes fonctionnalités offertes par le système.

# **CHAPITRE 6**

## **ETUDE DES SOLUTIONS LOGICIELLES**

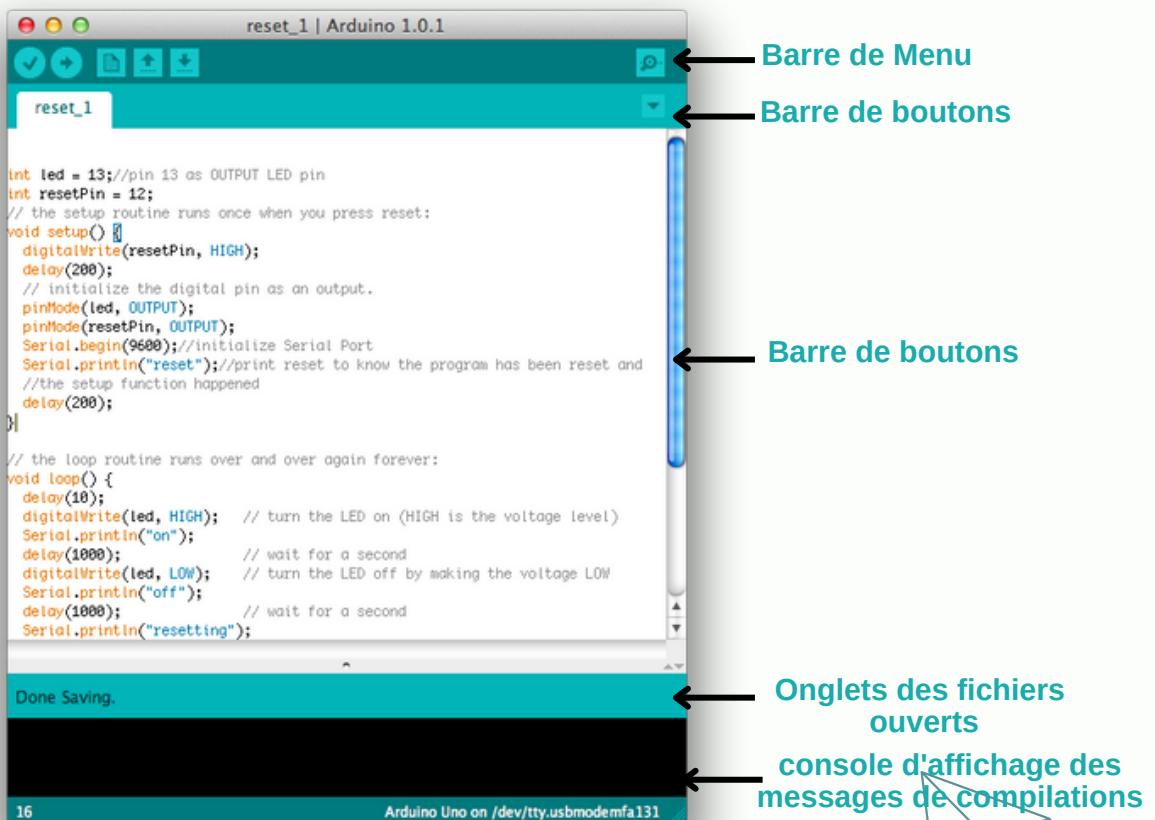
# introduction

Cette section est consacrée à l'exposition des plateformes informatiques employées pour la conception de nos systèmes d'IoT.

## 1-Plateforme de programmation Arduino:

## **1.1-Présentation du plateforme :**

L'IDE Arduino, ou environnement de développement intégré Arduino, est une application polyvalente conçue pour plusieurs systèmes d'exploitation tels que Windows, macOS et Linux. Programmé en C et C++, cet environnement est spécialement conçu pour écrire et télécharger des programmes sur des cartes Arduino compatibles. Il offre également la possibilité, grâce à des cœurs tiers, de programmer d'autres cartes de développement provenant de différents fabricants.



## Figure 22 : Espace de développement Arduino

Ce environnement de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino présente les caractéristiques suivantes :

- Une interface graphique (GUI) avec une barre de menus, commune à de nombreux logiciels.
- Une barre de boutons qui donne un accès direct aux fonctions essentielles du logiciel, facilitant ainsi son utilisation.
- Un éditeur de code avec coloration syntaxique, vous permettant d'écrire vos programmes Arduino. Il comprend également des onglets de navigation pour faciliter la gestion de plusieurs fichiers.
- Une zone de messages qui affiche l'état des actions en cours, vous informant ainsi des opérations en cours d'exécution.
- Une console texte qui affiche les messages relatifs au résultat de la compilation du programme.

## 1.2-Structure d'un programme en Arduino :

Le langage Arduino tire son inspiration de plusieurs langages, notamment le C, le C++, le Java et le Processing. Il présente une structure spécifique propre à l'informatique embarquée. Chaque programme Arduino se compose des éléments suivants:

- Un en-tête déclaratif.
- Une section de configuration qui ne s'exécute qu'une seule fois, définie par la fonction `setup()`.
- Une boucle sans fin, représentée par la fonction `loop()`, dans laquelle le programme s'exécute en continu. Cette partie constitue le cœur du programme.

Un programme Arduino est une séquence d'instructions élémentaires, écrites sous forme de texte, ligne par ligne. La carte Arduino lit et exécute ces instructions dans l'ordre défini par les lignes de code.

Pour compiler le programme dans l'environnement de développement Arduino, il faut le télécharger sur la carte Arduino via la connexion USB.

## 2-Le simulateur fritzing:

### 2.1-Présentation du simulateur:

Fritzing est un logiciel de conception électronique open-source avec un simulateur intégré. Il permet aux utilisateurs de créer des schémas électroniques, des prototypes et des circuits virtuels. Son interface conviviale facilite le glisser-déposer des composants sur une plaque de prototypage virtuelle. Le simulateur permet de tester et de vérifier le fonctionnement des circuits avant la construction physique. Fritzing est idéal pour les débutants, les amateurs et les étudiants qui souhaitent apprendre et expérimenter l'électronique, ainsi que pour documenter et partager des projets électroniques.

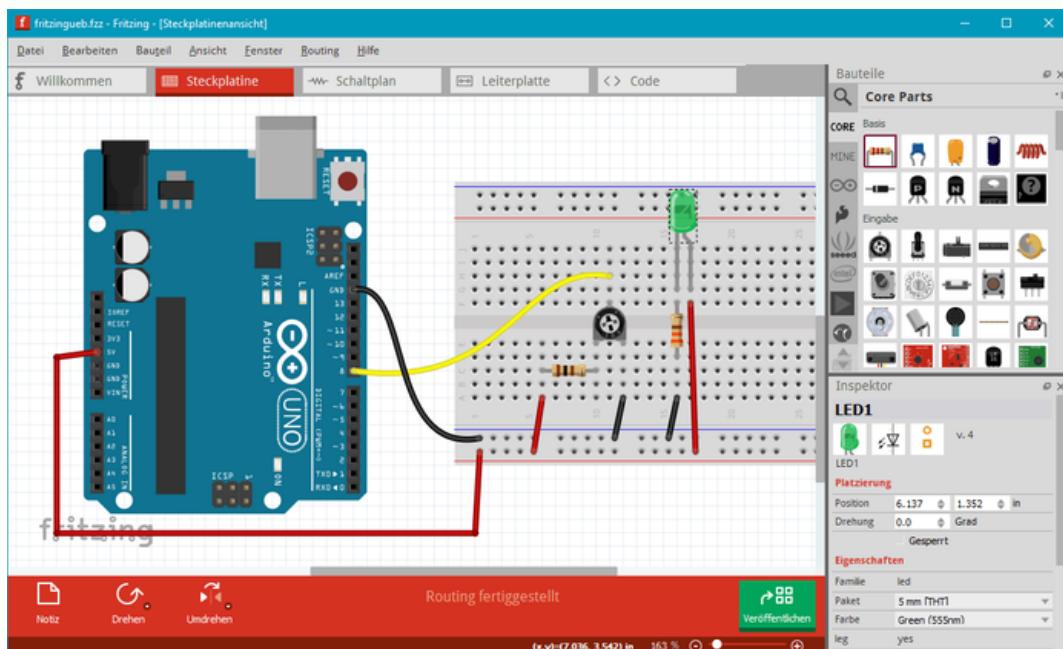


Figure 23 : Interface de Fritzing

## 2.2-Les avantages de fritzing:

Voici les avantages de Fritzing :

1. **Facilité d'utilisation** : Fritzing offre une interface conviviale et intuitive, ce qui le rend accessible même aux débutants en électronique.
2. **Simulation intégrée** : Le simulateur intégré permet de tester et de vérifier le fonctionnement des circuits sans avoir à les construire physiquement.
3. **Schémas visuels** : Fritzing permet de créer des schémas électroniques visuels, ce qui facilite la compréhension et la manipulation des composants.
4. **Plaque de prototypage virtuelle** : Il offre une plateforme virtuelle pour faire glisser et déposer les composants, simulant ainsi la création de prototypes électroniques.
5. **Documentation et partage** : Fritzing permet de documenter et de partager facilement les projets électroniques, ce qui favorise la collaboration et l'apprentissage communautaire.
6. **Bibliothèque étendue** : Il dispose d'une bibliothèque de composants électroniques étendue, offrant aux utilisateurs un large choix de composants pour leurs conceptions.
7. **Open-source** : Fritzing est un logiciel open-source, ce qui signifie qu'il est gratuit à utiliser et qu'il bénéficie de contributions et d'améliorations constantes de la part de la communauté des utilisateurs.

En résumé, Fritzing offre une interface conviviale, un simulateur intégré, des schémas visuels, une plaque de prototypage virtuelle, des fonctionnalités de documentation et de partage, une bibliothèque de composants étendue, le tout dans un logiciel open-source, en faisant un outil précieux pour les débutants et les amateurs en électronique.

### 3-Java :



Java est un langage de programmation polyvalent et orienté objet. Il a été développé par Sun Microsystems (maintenant Oracle) et est devenu l'un des langages les plus populaires et largement utilisés dans le domaine du développement logiciel.

Java est réputé pour sa portabilité, ce qui signifie qu'il peut être exécuté sur différentes plates-formes, telles que Windows, macOS et Linux, sans nécessiter de modifications significatives du code source.

Il est utilisé pour développer une variété d'applications, allant des applications de bureau traditionnelles aux applications mobiles, en passant par les applications Web et les systèmes embarqués. Il offre une grande flexibilité grâce à sa vaste bibliothèque standard et à sa capacité à prendre en charge des frameworks et des technologies supplémentaires.

#### 3.1-Java swing:

Java Swing est une bibliothèque de composants graphiques pour le développement d'interfaces utilisateur dans les applications Java. Il permet de créer des fenêtres, des boutons, des menus et bien d'autres éléments d'interface. Swing utilise le modèle MVC pour séparer la logique de l'application de sa présentation graphique. Il offre une grande personnalisation des composants, une flexibilité de disposition et de style, et est utilisé pour développer des applications graphiques multiplateformes.

Java Swing est largement utilisé pour créer des applications de bureau, des systèmes de gestion et des jeux, entre autres.

#### 4-MySQL :



MySQL est un système de gestion de base de données relationnelle (SGBDR) largement utilisé et open-source. Il permet de stocker, organiser et interroger des données de manière efficace. MySQL est réputé pour sa facilité d'utilisation, sa performance élevée et sa fiabilité.

Il utilise le langage SQL (Structured Query Language) pour gérer les données et offre des fonctionnalités avancées telles que la prise en charge des transactions, la réPLICATION des données, les index, les vues et les déclencheurs. Il est compatible avec de nombreux langages de programmation et peut être utilisé dans diverses applications et industries. MySQL est souvent utilisé dans le développement web, les applications d'entreprise, les systèmes de gestion de contenu (CMS) et les applications en ligne. Grâce à sa nature open-source, il est constamment amélioré et bénéficie d'une communauté active de développeurs.

## 5-XAMPP :



XAMPP est un ensemble de logiciels open-source qui fournit un environnement de développement web local tout-en-un. L'acronyme XAMPP signifie Apache, MySQL, PHP et Perl, qui sont les principaux composants inclus dans le package.

XAMPP est conçu pour être facile à installer et à utiliser, permettant aux développeurs de créer un serveur web local sur leur propre machine. Il fournit une plateforme de développement complète pour créer et tester des sites web dynamiques et des applications web.

# conclusion

Dans cette section, nous examinerons différentes solutions logicielles qui peuvent améliorer notre projet et répondre à nos besoins de manière plus efficace.

# CHAPITRE 7

## PHASE DE LA REALISATION DES SYSTEMES IOT

# introduction

Dans cette section finale, nous avons mis en place un programme de commande qui offre au responsable de sécurité la possibilité de contrôler l'ensemble du système à distance de manière automatique et conviviale. Cette phase de réalisation vise à concrétiser les étapes techniques précédentes en mettant en œuvre les fonctionnalités nécessaires.

## 1-Présentation du système Smart Home :

la première partie de notre projet "SmartHome" comprend un circuit de commande ESP-32, une lampe, un ventilateur et un servo-moteur. Ces éléments sont contrôlés par le circuit de commande, offrant une automatisation et une personnalisation des fonctions d'éclairage, de ventilation et de contrôle supplémentaire dans un environnement résidentiel intelligent.

### 1.1-Matériels nécessaire:

Le matériel utilisé pour ce projet comprend Arduino, des câbles, un ventilateur, une lampe, des résistances, un servo-moteur, une breadboard et ESP-32. Ces composants jouent un rôle essentiel dans la réalisation du système, offrant la connectivité, le contrôle et les fonctionnalités nécessaires pour créer un environnement intelligent et automatisé.

### 1.2-Simulation:

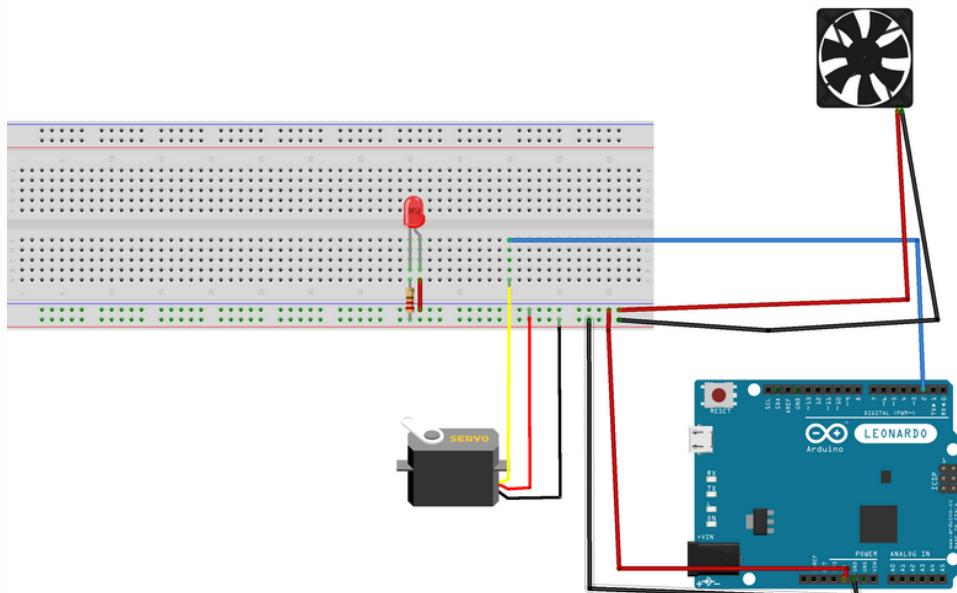


Figure 24 : Simulateur de système

### 1.3-Description de code:

La deuxième partie de notre projet se concentre sur la réalisation d'une application développée avec la technologie Java Swing. Cette application permet d'établir une communication avec le matériel afin de contrôler les lampes, les volets, l'aération et la climatisation.

```
1 #include <Servo.h>
2
3 Servo myservo; // create servo object to control a servo
4 int pos = 0;
5 char state;
```

Figure 25 : Code de déclaration des variables de Smart home

#### 1.3.2-Configuration:

```
6 void setup()
7 {
8     pinMode(7,OUTPUT);
9     pinMode(6,OUTPUT);
10    pinMode(2,OUTPUT);
11    myservo.attach(4);
12    pinMode(13,OUTPUT);
13    Serial.begin(9600);
14 }
15
```

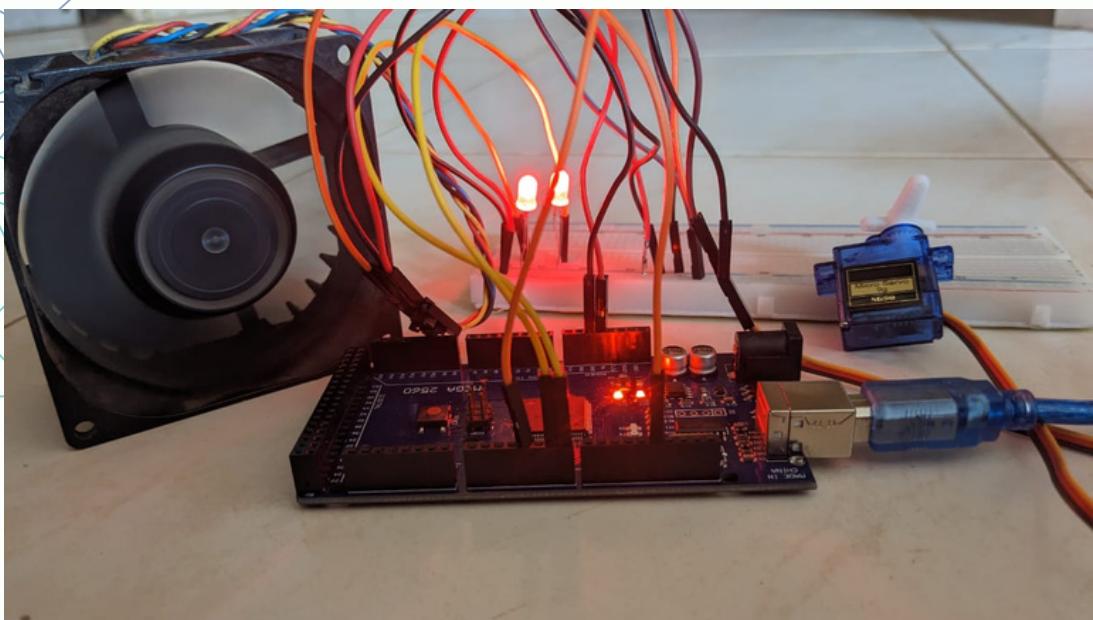
Figure 26 :Code de configuration des composants de système Smart home

#### 1.3.3-Traitement :

```
16 void loop(){
17     if(Serial.available()){
18         state=Serial.read();
19         if(state=='1')digitalWrite(7, HIGH);
20         else if(state=='a')digitalWrite(7, LOW);
21         else if(state=='2')digitalWrite(6, HIGH);
22         else if(state=='b')digitalWrite(6, LOW);
23         else if(state=='3')digitalWrite(12, HIGH);
24
25         else if(state=='d')digitalWrite(12, LOW);
26
27         else if(state=='v'){
28             for (pos = 0; pos <= 180; pos += 1) {
29                 myservo.write(pos);
30                 delay(15);
31             }
32             else if(state=='o'){
33                 for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
34                     myservo.write(pos); // tell servo to go to position in variable 'pos'
35                     delay(15); // waits 15ms for the servo to reach the position
36                 }
37                 else if(state=='c')digitalWrite(13, HIGH);
38
39                 else if(state=='e')digitalWrite(13, LOW);
40             }
41 }
```

Figure 27 :Code de Traitement de système Smart home

### 1.3.4-Résultat :

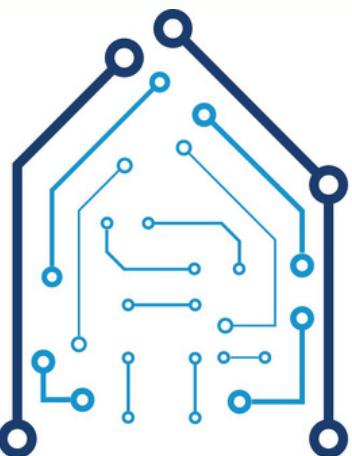


**Figure 28 :Résultat de système Smart home**

## 2-Présentation de l'application:

La deuxième partie de notre projet se concentre sur la réalisation d'une application développée avec la technologie Java Swing. Cette application permet d'établir une communication avec le matériel afin de contrôler les lampes, les volets, l'aération et la climatisation.

### 2.1-logo de l'application:



# CAPITAN

Figure 29:logo d'application

### 2.2-Page d'authentification:

Login Account

Username

Password

**Login**

D'ont have an account? [Sign up](#)

Figure 30:Page d'authentification de l'application

page d'authentification permettant aux utilisateurs de se connecter à l'application. La page devrait comporter des champs pour saisir le nom d'utilisateur et le mot de passe. Lorsque l'utilisateur clique sur le bouton de connexion, vérifiez les informations d'identification fournies. Si les informations sont correctes, l'utilisateur doit être redirigé vers l'application principale. Sinon, affichez un message d'erreur indiquant que les informations d'identification sont invalides.

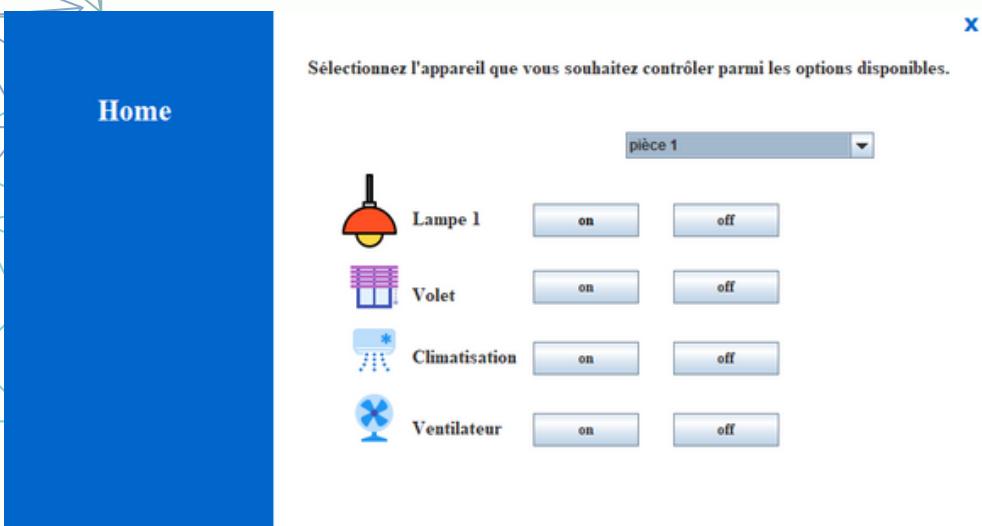
### 2.3-Page d'inscription:

The figure consists of two side-by-side screenshots of a web application. The left screenshot shows a 'Sign up' form with four input fields: 'Username', 'Email', 'Phone', and 'Password', each with a placeholder text ('Username', 'Email', 'Phone', '.....'). Below the fields is a blue 'Sign up' button. The right screenshot shows a blue-themed success message box with the text 'Hello, Friend!' at the top and 'Enter your details and start journey with us' below it.

Figure 31:Page d'inscription de l'application

page d'inscription permettant aux utilisateurs de créer un compte pour accéder à l'application. La page comporter des champs pour saisir les informations nécessaires telles que le nom, l'adresse e-mail et le mot de passe.

## 2.4-Page d'accueil:



**Figure 32:Page d'accueil de l'application**

La page d'accueil (home) de notre application est soigneusement conçue pour offrir une expérience conviviale et intuitive aux utilisateurs. Elle joue un rôle central en leur permettant d'accéder facilement à différentes fonctionnalités clés. Sur cette page, les utilisateurs ont la possibilité de contrôler les lampes dans les différentes chambres de leur choix. Ils peuvent facilement allumer ou éteindre les lumières, ajuster leur intensité et même changer leur couleur si elles sont équipées de cette fonctionnalité. Sur cette page, les utilisateurs ont la possibilité de contrôler les lampes dans les différentes chambres de leur choix. Ils peuvent facilement allumer ou éteindre les lumières. De plus, la page d'accueil offre aux utilisateurs la capacité de contrôler les fenêtres de manière pratique. Ils peuvent ouvrir ou fermer les fenêtres à distance, les utilisateurs peuvent également contrôler les climatiseur et les ventilateur à partir de la page d'accueil.

## 2.5-page d'administration:

The screenshot shows a web-based administration page titled "Admin Page". On the left side, there is a vertical list of input fields with labels: "id", "username", "email", "phone", and "password", each accompanied by a corresponding text input box. To the right of these fields is a large, empty rectangular area with a light gray background, which is likely a placeholder for a data grid or list. At the top right of this area is a blue button labeled "load user data". Below the input fields, there is a row of three blue buttons: "update", "create", and "delete". At the bottom left of the page is another blue button labeled "Home".

**Figure 33:Page d'administration de l'application**

La page d'administration (admin) de notre application est spécialement conçue pour offrir aux administrateurs un accès complet et centralisé à tous les aspects de notre système. Cette page est réservée aux utilisateurs disposant des droits d'administration et leur permet de gérer et de contrôler diverses fonctionnalités et paramètres.

Sur la page d'administration, les administrateurs peuvent gérer les utilisateurs en créant de nouveaux comptes, en modifiant les informations existantes et en supprimant des comptes si nécessaire.

# conclusion

Dans cette partie, nous allons examiner en détail les différents matériaux que nous avons utilisés, en fournissant une description détaillée du montage. De plus, nous présenterons l'application que nous avons créée à l'aide de Java, offrant à l'utilisateur la possibilité de communiquer avec le système.

# Conclusion générale

Dans notre rapport détaillé, nous avons présenté les différentes étapes de conception et de développement de notre projet de smart office. Notre objectif principal était de créer des systèmes IoT permettant de contrôler avec précision l'allumage des lampes, l'ouverture/fermeture des fenêtres, ainsi que la mise en marche ou l'arrêt de la climatisation et du ventilateur.

La collaboration au sein de notre équipe, ainsi qu'avec nos encadrants et co-encadrants, a été d'une grande valeur tout au long de cette expérience enrichissante.

Cependant, nous sommes conscients qu'il reste des possibilités d'amélioration pour ce projet. Par exemple, l'intégration de technologies de machine learning pourrait nous permettre d'effectuer des prévisions basées sur les données collectées. De plus, l'ajout de nouveaux systèmes IoT pourrait étendre les fonctionnalités de notre smart office. Nous sommes convaincus que l'incorporation de ces améliorations transformera notre smart office en une installation encore plus efficace et bénéfique pour les utilisateurs. Nous sommes impatients de poursuivre le développement et le perfectionnement de ce projet, en tirant parti des avancées technologiques et en répondant aux besoins croissants du domaine de l'IoT.