République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université 20 Août 1955 de Skikda.
Faculté des Sciences.
Département de l'Informatique.

Mémoire de Fin d'Etudes Pour l'Obtention du Diplôme de Master

Spécialité : GLAA

N d'ordre : .....

Modélisation et Simulation d'un Système IoT Pour Smart Home

Préparé Par :
DJABER Marouane

Dirigé par :
Dr.SEDDARI Noureddine

Session 2022/2023

Dédicaces

Je dédie ce modeste travail à ma mère qui m'a donné la vie et qui incarne

la tendresse et le sacrice pour mon bonheur et ma réussite. A mon père

A ma soeur et à mon frère imran que j'aime.

À ma chère épouse. À mes chers petits neveux et nieces. À tous mes amis.

A toute ma famille, qu'elle soit pe tite ou grande, ainsi qu'à mes chers amis allali arabi , saif mabrouk , laarid chouaib , bahloul mossab , bizou ,raouf , amin , je vous suis profondément re connaissant pour votre soutien indéfectible et votre amour inconditionnel.

Vos dédicaces,

Remerciements

En premier lieu, ma profonde gratitude et mes sincères remerciements vont à mon encadrant **Dr SEDDARI Noureddine** pour m'avoir dirigé, conseillé et encouragé tout au long de la réalisation de ce travail et surtout pour sa patience et sa compréhension afin de donner le fruit de ce travail.
Mes remerciements vont aussi aux membres du jury,le Docteur **LAAROUM Toufik** ,maître de conférences A de l'université de Skikda, ainsi que pour Mme **Kerroui Sara** de l'université de Skikda, pour m'avoir fait l'honneur de participer à mon jury de soutenance en tant qu'examinateurs.

ملخص

في الماضي، كان البشر يقومون ببناء الكوخ لضمان سلامتهم من العناصر الطبيعية والمفترسين. مع مرور الوقت، تطورت هذه الكوخ إلى منازل تتناسب مع احتياجات البشر وتوافر مواد البناء، مما أدى إلى ولادة المنازل التي نعرفها اليوم. المنازل الحديثة مجهزة بالأجهزة الإلكترونية والكهربائية، ومع ظهور التحكم المنزلي الآلي والإنترنت، تطورت أيضًا احتياجات البشر. وبالتالي، ظهرت فكرة المنازل المتكاملة تمامًا والتي توفر بيئة مريحة وآمنة. تجمع هذه التقنيات مع الذكاء الاصطناعي أدت إلى ظهور المنازل الذكية.

يهدف هذا المشروع إلى تصميم منزل ذكي يتم التحكم فيه تلقائيًا عن طريق جهاز كمبيوتر أو هاتف محمول باستخدام لوحة أردوينو. الهدف هو إنشاء منزل يحسن استخدام الطاقة، ويوفر مستوى عالٍ من الأمان، ويعطي الأولوية للراحة، ويسهل استقلال الأشخاص المسنين وذوي الإعاقة، ويشجع المشاركة الاجتماعية بدلاً من الاعتماد على المساعدة. بفضل التقدمات في الأجهزة المدمجة وإنترنت الأشياء والحوسبة السحابية، تستطيع المنازل الذكية توفير ميزات جديدة مثل التفاعل مع البيئة من خلال الاستشعارات المدمجة والأجهزة الميكروإلكترونية، بالإضافة إلى اتخاذ قرارات مستقلة وذكية.

كلمات مفتاحية : منازل، ذكاء اصطناعي، لوحة أردوينو، راحة، مشاركة اج.

Résumé

Dans le passé, les humains construisaient des huttes pour assurer leur sécurité contre les éléments naturels et les prédateurs. Au fil du temps, ces huttes ont évolué en fonction des besoins humains et de la disponibilité des matériaux de construction, donnant naissance aux maisons que nous connaissons aujourd'hui. Les maisons modernes sont équipées d'appareils électroniques et électriques, et avec l'avènement de la domotique et d'Internet, les besoins humains ont également évolué. Ainsi, l'idée de maisons entièrement automatisées, offrant un environnement plus confortable et sûr, a émergé. La combinaison de ces technologies avec l'intelligence artificielle conduit à l'émergence des maisons intelligentes.

Ce projet vise à concevoir une maison intelligente contrôlée automatiquement par un ordinateur ou un téléphone, en utilisant une carte Arduino. L'objectif est de créer une maison qui optimise l'utilisation de l'énergie, offre un haut niveau de sécurité, privilégie le confort, facilite l'autonomie des personnes âgées et handicapées, et favorise la participation sociale plutôt que la dépendance à l 'assistance. Grâce aux avancées dans les appareils embarqués, l'Internet des objets et le cloud computing, les maisons intelligentes sont capables de nouvelles fonctionnalités telles que l'interaction avec l'environnement grâce à des capteurs intégrés et des dispositifs microélectroniques, ainsi que la prise de décisions indépendantes et intelligentes.

Mots clés : maisons intelligente, intelligence artificielle,carte Arduino,confort,internet des objets, participation sociale,capteurs intégrés,décisions intelligentes.

Abstract

In the past, humans used to build huts to ensure their safety against natural elements and predators. Over time, these huts evolved based on human needs and the availability of construction materials, giving rise to the houses we know today. Modern houses are equipped with electronic and electrical appliances, and with the advent of home automation and the Internet, human needs have also evolved. Thus, the idea of fully automated houses that provide a more comfortable and secure environment has emerged. The combination of these technologies with artificial intelligence has led to the emergence of smart homes.

This project aims to design a smart home that is automatically controlled by a computer or a phone, using an Arduino board. The objective is to create a home that optimizes energy usage, provides a high level of security, prioritizes comfort, facilitates the independence of elderly and disabled individuals, and promotes social participation rather than dependence on assistance. Thanks to advancements in embedded devices, the Internet of Things, and cloud computing, smart homes are capable of new functionalities such as interacting with the environment through integrated sensors and microelectronic devices, as well as making independent and intelligent decisions.

Keywords : : houses, artificial intelligence, Arduino board,IOT, comfort, social participation, integrated sensors, intelligent decisions.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

La maison est l'endroit où les individus passent la majeure partie de leur temps, notamment les personnes âgées, ce qui souligne l'impact considérable du logement sur la qualité de vie. L'aspect essentiel d'un habitat est de procurer un sentiment de confort et de sécurité.

Ces dernières années, ce domaine a suscité un vif intérêt et reste une question cruciale pour l'avenir. Grâce aux avancées technologiques majeures dans les domaines des réseaux de communication, des systèmes embarqués et de l'intelligence artificielle, le concept de maison intelligente est passé du stade de fantasme ou de rêve à une réalité concrète.

La conception d'un nouveau système de contrôle pour une maison intelligente aura pour objectif principal d'améliorer le quotidien des individus fragiles et ayant une certaine dépendance (par exemple, les personnes âgées, les personnes à mobilité réduite ou atteintes de problèmes de vision, etc.).
Cette thèse est organisée en quatre chapitres dont les thèmes sont donnés ci-dessous :

- Le premier chapitre :
  Ce chapitre de notre étude s'est concentré sur les systèmes embarqués et leur interconnexion. Nous l'avons divisé en deux parties distinctes.
  - La première partie traite de l'historique des systèmes embarqués, en fournissant des définitions de chercheurs antérieurs. Nous explorons également les composants, les limitations, les différents types d'architecture, ainsi que les domaines d'application de ces systèmes.
  -La deuxième partie est dédiée à l'interconnexion des choses, qui constitue le lien entre les systèmes embarqués et les maisons intelligentes. Nous examinons le fonctionnement de cette interconnexion, ainsi que ses divers domaines d'application.
- Le deuxième chapitre :
  Dans ce chapitre, nous avons examiné la notion de maison intelligente en mettant l'accent sur le confort, les économies d'énergie, la communication et la sécurité. Nous avons également abordé divers appareils connectés qui peuvent être contrôlés, tels que l'éclairage, le chauffage, les caméras intelligentes, et ainsi de suite. De plus, nous avons mentionné d'autres domaines liés à notre projet, tels que les villes intelligentes, l'agriculture, la santé, les transports, et bien d'autres.

- Le troisième chapitre :
  Dans ce chapitre, nous avons effectué la modélisation du projet en utilisant UML-RT avec le logiciel Papyrus afin de simplifier et de faciliter le fonctionnement de la maison intelligente pour le continent. Pour les simulations, nous avons utilisé le logiciel ISIS (Proteus), en interconnectant différents accessoires tels que des dispositifs lumineux et des capteurs de mouvement, pour contrôler le fonctionnement de la maison intelligente de manière efficace.
- Le quatrième chapitre :
  Dans ce chapitre, nous avons présenté les différents outils et langages de programmation utilisés, ainsi que le programme et la mise en œuvre du système.

Chapitre 1

Les systèmes embarqués

1.1 Introduction

Nous sommes envahis d'appareils, de plus en plus intelligents et de plus en plus connectés, qui collectent de nombreuses informations sur leur environnement et leurs utilisateurs. Attachés à ces appareils, des systèmes embarqués, sorte d'ordinateur, interagissent avec l'appareil et l'utilisateur afin de leurs rendre des services. Ce premier chapitre présente une vue d'ensemble des systèmes embarqués, depuis le hardware les constituant jusqu'au software les contrôlant.

1.2 Historique des systèmes embarqués

Les premiers systèmes embarqués sont apparus en 1971 avec l'apparition de l'Intel 4004. L'Intel 4004 développé en 1971, le premier microprocesseur, était le premier circuit intégré incorporant tous les éléments d'un ordinateur dans un seul boîtier : unité de calcul, mémoire, contrôle des entrées / sorties. Alors qu'il fallait auparavant plusieurs circuits intégrés différents, chacun dédié à une tâche particulière, un seul microprocesseur pouvait assurer autant de travaux différents que possible. Très rapidement, des objets quotidiens tels que fours à micro-ondes, télévisions et automobiles à moteur à injection électronique ne tardèrent pas à être équipés de microprocesseurs. Ce sont alors les débuts de l'informatique embarquée. [1]

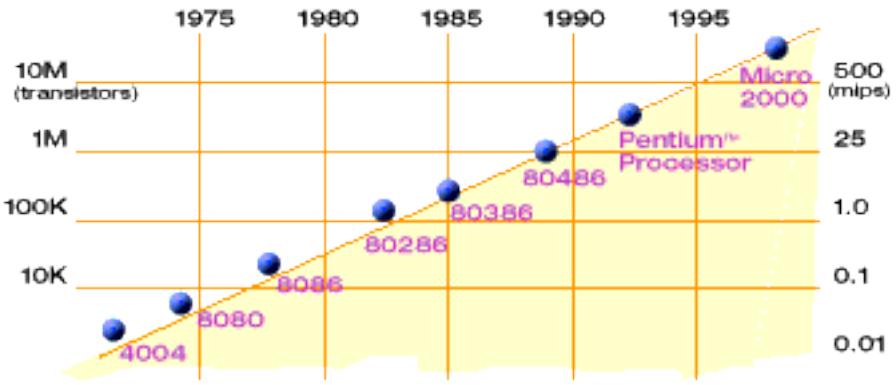

FIGURE 1.1 – Histoire des micro-processeurs

1.3 Définition des systèmes embarqués

1.3.1 Définition

Un système embarqué un ensemble autonome d'appareils informatiques et électroniques qui sont intégrés à un dispositif spécifique et exécutent une tâche définie. Cette tâche est généralement effectuée en temps réel. Le terme "système embarqué" englobe à la fois le matériel et le logiciel utilisé dans ce contexte [2].

Le logiciel d'un système embarqué est exécuté sur différentes plateformes, telles qu'un microcontrôleur, un microprocesseur ou occasionnellement un FPGA. Le choix de la plateforme dépend à la fois des besoins en puissance de calcul et des interfaces requises, telles que les bus de communication, l'écran ou les mesures analogiques[3]

Les systèmes embarqués sont souvent soumis à des contraintes strictes en termes d'espace mémoire et de taille. Par conséquent, ils doivent être capables d'accomplir leur tâche avec précision, dans des délais sélectionnés, en optimisant simultanément le rapport entre l'encombrement, la mémoire, la consommation d'énergie et les coûts..

1.3.2 Définition 2

Un système embarqué est un système électronique et informatique autonome qui ne dispose pas des interfaces d'entrée et de sortie standard telles qu'un clavier ou un écran d'ordinateur. Il se distingue également par le fait qu'il n'est pas directement visible en tant qu'entité distincte, mais est intégré à un équipement qui remplit une autre fonction. On peut également dire que le système est "enfoui", ce qui reflète plus précisément le terme anglais "embedded" [4] .

Les systèmes informatiques embarqués font partie intégrant de notre quotidien et englobent diverses applications telles que la gestion des ascenseurs, les autoradios, les

calculateurs d'airbag, les distributeurs de boissons, les routeurs Internet, les téléphones mobiles, les distributeurs de billets et les consoles de jeux.

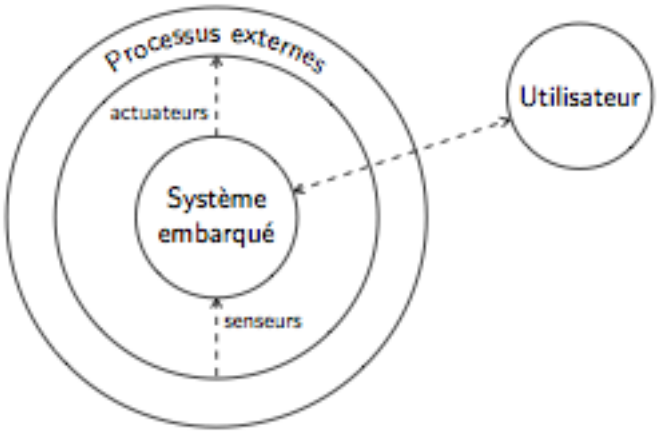

FIGURE 1.2 – Le rôle central des systèmes embarqués

Les systèmes informatiques embarqués jouant un rôle essentiel dans diverses applications telles que la gestion des ascenseurs, les autoradios, les calculateurs d'airbag, les distributeurs de boissons, les routeurs Internet, les téléphones mobiles, les distributeurs de billets et les consoles de jeux. Ils sont intégrés de manière transparente dans ces dispositifs pour offrir des fonctionnalités avancées et améliorer notre expérience utilisateur. [4]

1.4 Composition d'un système embarqué

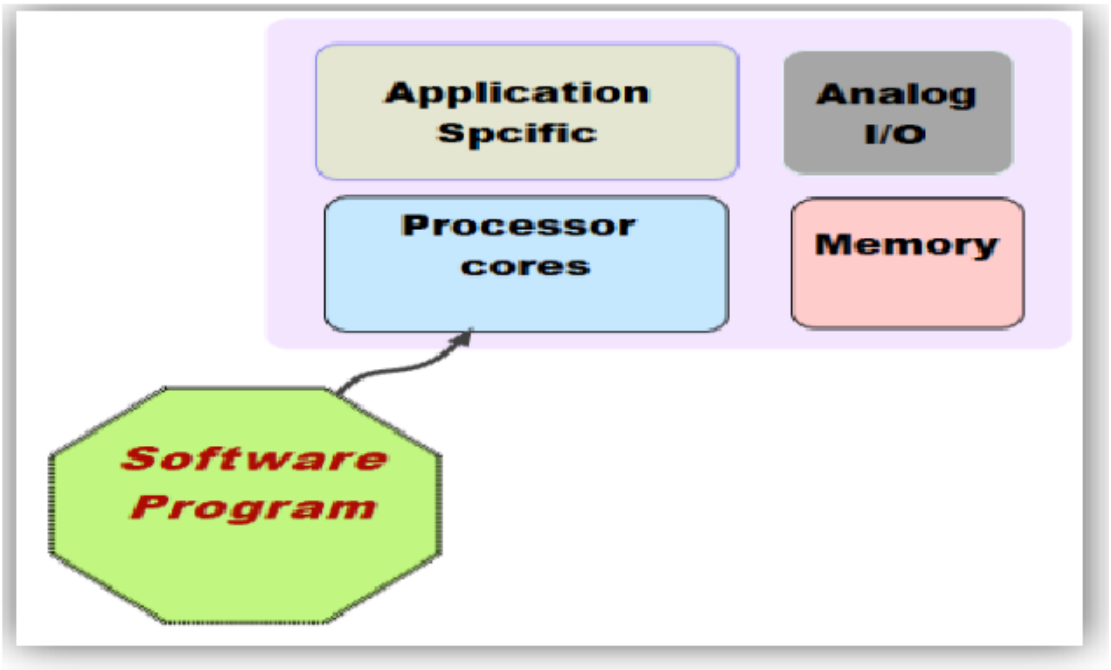

FIGURE 1.3 – Composition d'un système embarqué.

Un système embarqué est composé d' :

- Une partie matérielle utilisée pour la performance
  - Microprocesseur, microcontrôleurs, DSP.
  - Mémoires.
  - Interfaces d'entrées /sorties.
- Une partie logicielle
  - Programmes.

1.5 Caractéristiques des systèmes embarqués

Le système est caractérisé par sa petite taille, ce qui permet de réduire son encombrement, ainsi que par une consommation d'énergie relativement faible. En pratique, il doit faire face à plusieurs contraintes, notamment en termes de [5] :

- Espace mémoire limité, généralement de quelques gigaoctets maximum.
- Puissance de calcul adaptée aux besoins spécifiques du système.
- Autonomie, c'est-à-dire la capacité à fonctionner pendant une durée sans nécessiter de recharge ou de remplacement de la batterie.
- Respect des délais d'exécution requis pour réaliser les tâches en temps voulu.
- Sécurité, en assurant la protection des données et la prévention des accès non autorisés.
- Fiabilité, afin que le système puisse fonctionner de manière stable et sans défaillances.

En résumé, l'objectif est de concevoir un système embarqué qui offre le meilleur compromis en termes de dimensions, de consommation d'énergie, de puissance de calcul, de coût et de fiabilité.

1.6 Architecture d'un système embarqué

L'architecture d'un système embarqué se définie par le schéma suivant [6] :

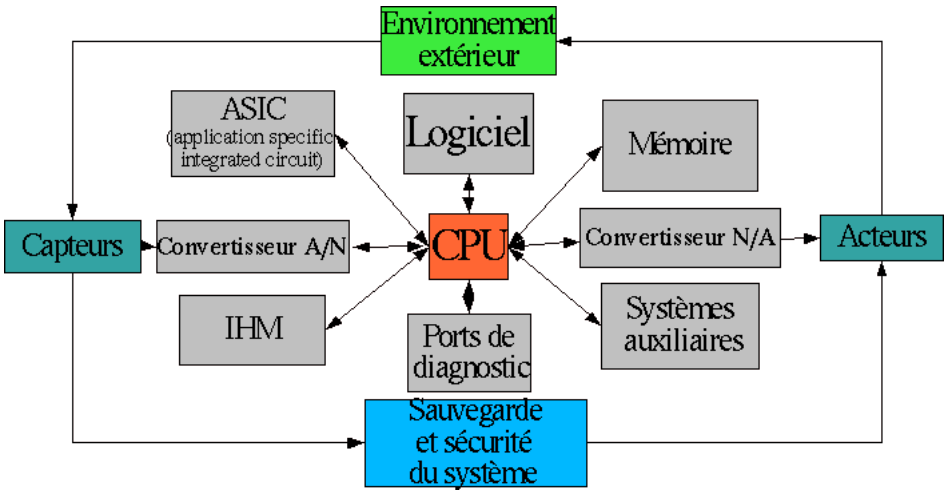

FIGURE 1.4 – Architecture d'un système embarqué [6]

L'architecture des systèmes embarqués peut varier en fonction des besoins spécifiques de chaque système. Dans de nombreux systèmes embarqués autonomes et indépendants, il n'y a pas de systèmes auxiliaires supplémentaires. En revanche, l'architecture de base est la plupart du temps composée d'une unité centrale de traitement (CPU), d'un ssytème d'exploitation qui réside parfois uniquement en un logiciel spécifique (ex : routeur), ou une boucle d'execution (ex : ABS). De même l'interface IHM n'est pas souvent existante, mais est souvent utile pour reconfigurer le système ou verifier son comportement.

Le fonctionnement du sytème se résume ainsi :

— Il reçoit des informations de l'environnement extérieur qu'il converti en signal numérique

— L'unité de traitement composée du CPU, de la mémoire, du logiciel, de l'ASIC et eventuellement de système externes traite l'information

— Le traitement génère eventuellement une sortie qui est envoyée vers la sortie, les systèmes auxiliaire, les ports de monitoring ou l'IHM

1.7 Le développement de systèmes embarqués

Pour la réalisation d'un système embarqué, plusieurs composants matériels sont nécessaires. Voici quelques exemples [7] :

— La documentation (datasheet) sur les composants utilisés. C'est la première source d'informations pour le développement.

— L'outillage de base de l'électronicien (fer à souder, insoleuse...)

— Les outils d'analyse temporelle : oscilloscope, analyseur logique...

— Des composants de base (résistances, condensateurs...)

— Un microprocesseur ou un microcontrôleur

— Un compilateur croisé (dit aussi en anglais cross-compiler)

— Un programmateur de microcontrôleur ou un programmateur in-situ

— Un émulateur in-circuit ou ICE (In Circuit Emulator). Considéré comme l'équipement roi pour le debug matériel et logiciel (possibilité de déverminer au niveau du source du logiciel), cependant il reste coûteux.

— Une sonde JTAG

— Ingénierie des systèmes : approche multidisciplinaire pour définir, développer et déployer des systèmes embarquant des technologies numériques.

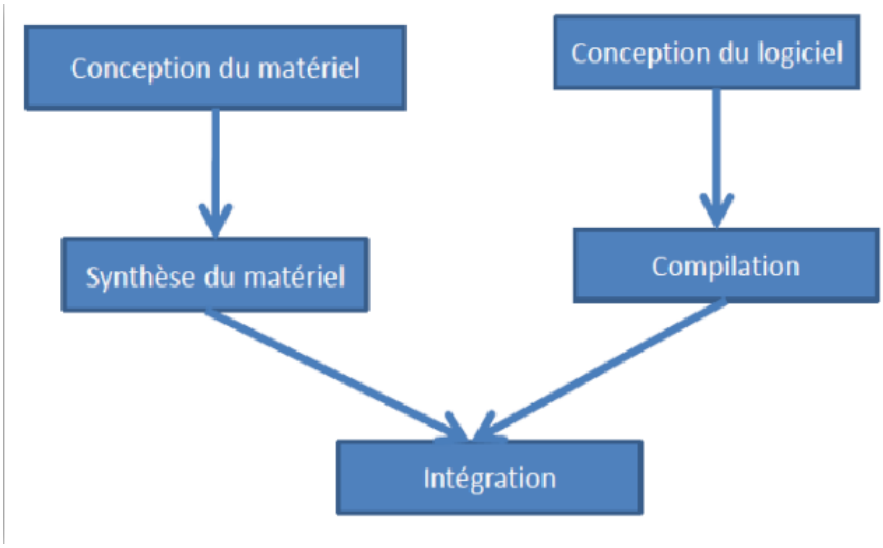

FIGURE 1.5 – Développement classique.

1.8 Contraintes des systèmes embarqués

Les systèmes embarqués exécutent des tâches prédéfinies et ont un cahier des charges contraignant à remplir, qui peut être d'ordre [8] :

— les coûts de production.L'objectif principal lors du développement de systèmes embarqués est de minimiser les coûts de production, en particulier pour les grandes séries, tout en maintenant la qualité requise.

— Il est essentiel de concevoir des systèmes embarqués avec une utilisation optimale de l'espace mémoire limité, de l'ordre de quelques Mo maximum, afin de répondre précisément aux besoins et d'éviter tout surcoût.

— De puissance de calcul.Il est important d'optimiser la puissance de calcul des systèmes embarqués en fonction des besoins et des contraintes temporelles spécifiques de la tâche afin de prévenir tout surcoût matériel et de minimiser la consommation excessive d'énergie électrique.

— De consommation énergétique la plus faible possible,Il est essentiel de minimiser la consommation d'énergie des systèmes embarqués en utilisant des batteries, des

panneaux solaires ou même des piles à combustible dans certains prototypes, afin d'assurer une efficacité énergétique maximale.

— Temporel, Ces systèmes déposés sont généralement caractérisés par des propriétés temporelles réelles, où les temps d'exécution et les échéances temporelles des tâches sont déterminés et bornés a priori, ce qui nécessite une gestion déterminée des contraintes temporelles.

— La sûreté de fonctionnement est une contrainte primordiale pour ces systèmes embarqués, en particulier pour ceux qui sont considérés comme critiques, car toute défaillance pourrait mettre en danger des vies humaines ou compromettre des investissements importants. Ainsi, ces systèmes doivent fournir des résultats précis, pertinents et dans les délais attendus par les utilisateurs, sans aucune marge d'erreur.

— La sécurité constitue une autre contrainte essentielle pour ces systèmes embarqués, en particulier lorsqu'ils manipulent des informations confidentielles telles que des données médicales ou des informations personnelles. Il est crucial de garantir la confidentialité, l'intégrité et la protection de ces informations, notamment dans le cas de systèmes permettant aux personnels l'acquisition et la transmission à distance de données sensibles, afin de préserver la vie privée des utilisateurs et de prévenir tout accès non autorisé.

1.9 Domaines d'application des systèmes embarqués

Les systèmes embarqués se trouvent dans divers domaines tels que [8] :

— transport : Automobile, Aéronautique (avionique), etc.

— astronautique : fusée, satellite artificiel, sonde spatiale, etc.

— militaire : missile.

— télécommunication : Set-topbox, téléphonie, routeur, pare-feu, serveur de temps, téléphone portable, etc.

— électroménager : télévision, four à micro-ondes

— impression : imprimante multifonctions, photocopieur, etc.

— informatique : disque dur, Lecteur de disquette, etc.

— multimédia : console de jeux vidéo, assistant personnel

— guichet automatique bancaire (GAB)

— équipement médical

— automate programmable industriel, contrôle-commande

— jeux : consoles

— métrologie

1.10 Conclusion

En conclusion, les systèmes embarqués sont des composants essentiels dans de nombreux domaines, offrant des fonctionnalités avancées et une automatisation intelligente. Leur optimisation en termes de coût, d'espace, de puissance de calcul, de consommation énergétique et de sécurité reste un défi majeur. Cependant, grâce à des approches ingénieuses, à une conception précise et à une expertise multidisciplinaire, il est possible de réduire ces contraintes et d'offrir des systèmes embarqués plus efficaces, fiables et économiques. Cette recherche constante de réduction permettra de répondre aux besoins croissants des utilisateurs tout en ouvrant la voie à de nouvelles opportunités d'innovation et de développement dans un large éventail de secteurs industriels

Chapitre 2

L'IoT pour la maison intelligente

2.1 Introduction

Ce chapitre explore l'Internet des Objets (IoT) appliqué à la maison intelligente. L'IoT permet de connecter et de contrôler différents objets et équipements domestiques via internet, offrant ainsi de nouvelles possibilités en termes de confort, d'efficacité énergétique, de sécurité et de commodité. Nous examinerons la définition de l'IoT et son fonctionnement, ainsi que les avantages et les applications de cette technologie dans le contexte de la maison intelligente.

2.2 Historique

Les constructions ont évolué en fonction des besoins de l'homme, l'homme de l'histoire et nomade et se déplacent au gré des saisons et des migrations animales, pour se mettre à l'abri, il fabrique des huttes faites de branchages d 'Ossements et de peau.

FIGURE 2.1 – Maisons rondes de bois

Il y a environ 12 mille ans les hommes inventent l'élevage et l'agriculture, n'ayant plus besoin de se déplacer pour trouver leur nourriture, ils bâtissent des habitats fixes et se

regroupent dans les premiers villages dont les maisons sont rondes construite en bois ou en terre et recouvertes de feuillages

C'est en Mésopotamie il y a cinq mille ans que naissent les premières villes progressivement les maisons deviennent rectangulaire ou carrés formé ou pratiques pour être cloisonné en différents espaces et permettent d'assembler les maisons les unes contre les autres autour de petites rues.

FIGURE 2.2 – Modèle d'une maison gauloise

2.3 Définition et fonctionnement de l'IoT

L'IoT (Internet des Objets) est un réseau de dispositifs interconnectés qui collectent, échangent et partagent des données via internet. Les objets connectés sont équipés de capteurs et de technologies de communication pour permettre la collecte des données, qui sont ensuite transmises et traitées pour prendre des décisions ou effectuer des actions automatisées [9].

2.4 Architecture, éléments et protocoles IoT

Dans cette section, nous aborderons l'architecture, les éléments et les protocoles de l'IoT (Internet des Objets). Ces aspects sont essentiels pour comprendre comment les dispositifs connectés interagissent et communiquent au sein de l'écosystème de l'IoT. L'architecture de l'IoT peut varier en fonction des besoins spécifiques d'une application ou d'un système. Cependant, une architecture couramment utilisée est celle à trois couches. Cette architecture comprend [10] :

FIGURE 2.3 – Architecture IoT à trois couches

La couche de perception

Cette couche est composée des capteurs et des dispositifs qui collectent les données physiques de l'environnement. Les capteurs mesurent divers paramètres tels que la température, l'humidité, la luminosité, la présence, etc [11].

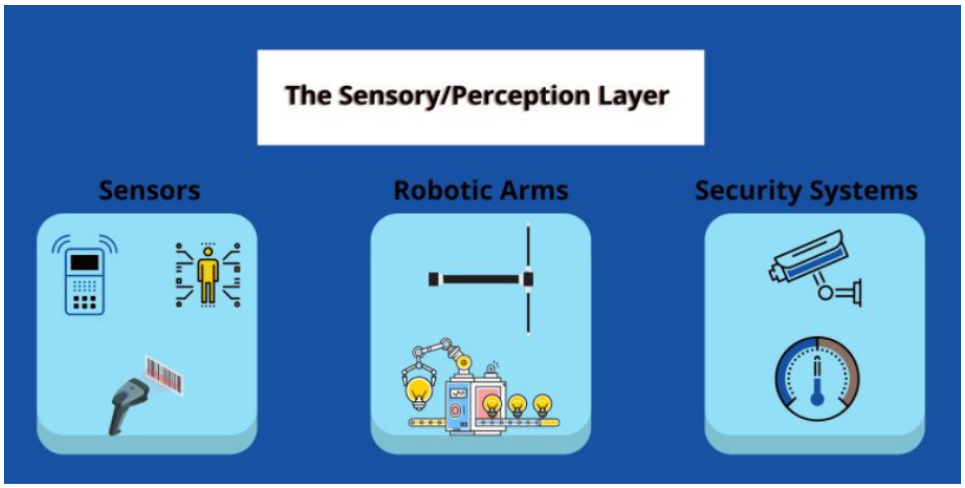

FIGURE 2.4 – Couche de perception[5]

La couche de réseau

Cette couche assure la connectivité entre les capteurs et les dispositifs. Elle utilise différents protocoles de communication tels que le Wi-Fi, le Bluetooth, Zigbee, LoRa, ou des réseaux cellulaires pour transmettre les données collectées [12].

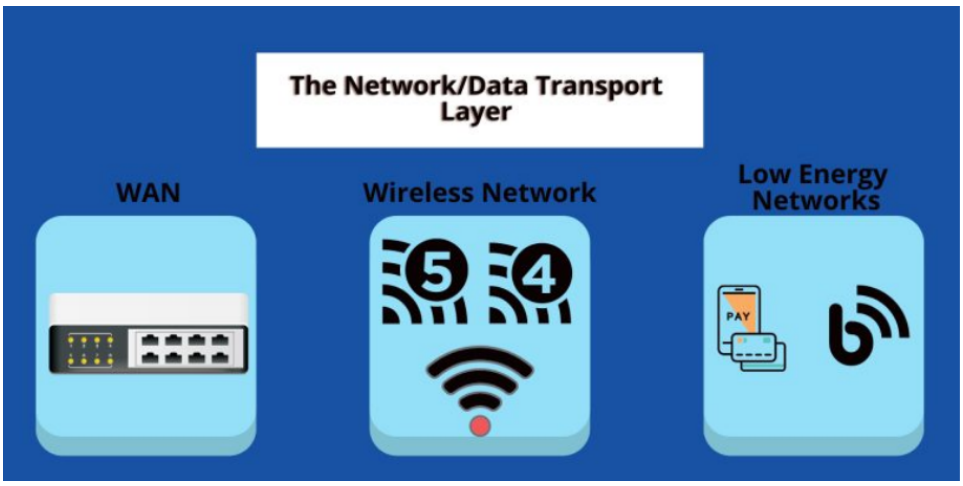

FIGURE 2.5 – Couche reseau

La couche d'application

Les éléments clés de l'IoT comprennent les capteurs, les actionneurs, les passerelles et les dispositifs de traitement des données. Les capteurs collectent les données, les actionneurs permettent d'effectuer des actions physiques en réponse aux données collectées, les passerelles assurent la connectivité entre les dispositifs et les réseaux, et les dispositifs de traitement des données traitent et analysent les données collectées [13].

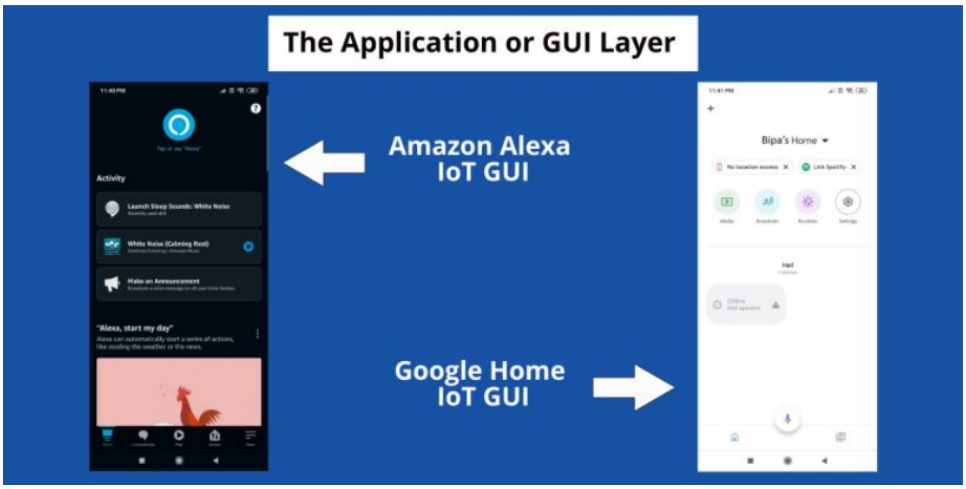

FIGURE 2.6 – Couche application

En ce qui concerne les protocoles de communication, il en existe plusieurs qui sont utilisés dans l'IoT. Certains protocoles couramment utilisés incluent MQTT (Message Queuing Télémétrie Transport), CoAP (Constrained Application Protocol), HTTP (Hypertext Transfer Protocol), et Zigbee, pour n'en citer que quelques-uns. Ces protocoles facilitent la transmission des données entre les dispositifs de l'IoT de manière efficace et sécurisée.

Comprendre l'architecture, les éléments et les protocoles de l'IoT est crucial pour

concevoir et mettre en œuvre des systèmes IoT efficaces et interopérables. Cela permet de créer des réseaux d'objets connectés fonctionnant de manière fluide et offrant des services intelligents dans divers domaines.

2.5 Différentes architectures IoT

Dans cette section, nous examinerons différentes architectures couramment utilisées dans le domaine de l'IoT (Internet des Objets). Ces architectures offrent des structures organisées pour la conception et le déploiement de systèmes IoT.

Architecture IoT à trois couches

Cette architecture est basée sur trois couches principales : la couche de perception, la couche de réseau et la couche d'application. La couche de perception comprend les capteurs et les dispositifs qui collectent les données physiques. La couche de réseau assure la connectivité entre les dispositifs IoT et permet la transmission des données. Enfin, la couche d'application gère le traitement et l'analyse des données collectées pour fournir des fonctionnalités spécifiques [14].

Architecture IoT à cinq couches

Cette architecture étend l'architecture à trois couches en ajoutant deux nouvelles couches intermédiaires. La couche de gestion de la donnée est responsable de la collecte, du stockage et de la gestion des données IoT. La couche d'intelligence inclut des algorithmes d'apprentissage automatique et d'analyse de données pour extraire des informations précieuses des données collectées.

Architecture IoT à sept couches

Cette architecture est plus détaillée et permet une meilleure modularité et évolutivité des systèmes IoT. Les sept couches sont les suivantes : perception, transport, traitement de données, gestion de la plate-forme, gestion de la sécurité, gestion de l'application et interface utilisateur.

Chacune de ces architectures a ses avantages et ses applications spécifiques. Le choix de l'architecture dépend des besoins du système IoT, de la complexité des données à gérer et de la portée des fonctionnalités souhaitées.

Il convient de noter que ces architectures servent de guides généraux et peuvent être adaptées ou personnalisées en fonction des exigences spécifiques de chaque projet ou

application IoT. L'objectif principal est de fournir une structure solide et cohérente pour le déploiement réussi de systèmes IoT efficaces et évolutifs[15].

2.6 Applications de l'IoT

Dans cette section, nous explorerons différentes applications de l'IoT (Internet des Objets) dans divers domaines.

2.6.1 Définition de la maison intelligente

Les maisons intelligentes seront probablement les applications IoT les plus populaires. La maison intelligente, ou domotique, est une extension de l'automatisation du bâtiment, avec laquelle nous pouvons surveiller et contrôler le chauffage, la ventilation et la climatisation (CVC), l'éclairage [16] , les appareils électroménagers, les systèmes de sécurité à bande. En connectant tous les appareils électroménagers, nous pouvons automatiser de nombreuses routines quoti diennes, comme allumer et éteindre automatiquement les lumières et le chauffage, démarrer ou arrêter la cuisson et le lavage, etc. Avec le réseau intelligent et les compteurs intelligents, nous pouvons réduire les consommations d'énergie et les factures de services publics, et avec les systèmes de sécurité, nous pouvons rendre la maison plus sûre en détectant automatiquement et en dissuadant, espérons-le, les intrusions en utilisant divers capteurs infrarouges, de mouvement, sonores, de vibration ainsi que des systèmes d'alarme [17].

Une maison intelligente peut également rendre les personnes âgées et les personnes handicapées plus confortables et plus sûres à la maison. Avec l'IoT, nous pouvons collecter et analyser les données des personnes âgées et handicapées pour diagnostiquer les maladies, prédire les risques potentiels, identifier ou prévenir les accidents tels que les chutes, ouvrir ou verrouiller la porte (ou les fenêtres) à distance, et laisser les membres de la famille les surveiller à distance [17]. Avec l'IoT, il est également possible de rapprocher les personnes âgées et handicapées du monde extérieur et de réduire leur sentiment de solitude

Le marché de la maison intelligente devrait avoir une valeur marchande de plus de 53 milliards de dollars en 2022 [18] .

2.6.2 Maison intelligente (Smart home)

L'IoT offre de nombreuses possibilités pour rendre nos maisons plus intelligentes et connectées. Les systèmes domotiques basés sur l'IoT permettent de contrôler à distance l'éclairage, le chauffage, la sécurité, les appareils électroménagers, et bien plus encore. Des capteurs intelligents surveillent également l'environnement domestique, détectant la

présence, la température, l'humidité, et permettant une automatisation personnalisée pour le confort et l'efficacité énergétique [19].

FIGURE 2.7 – Smart home

2.6.3 Villes intelligentes (Smart cities)

:

L'IoT contribue à la transformation des villes en environnements intelligents et durables. Des capteurs et des dispositifs connectés surveillent et gèrent les infrastructures urbaines telles que l'éclairage public, la gestion des déchets, les transports, la sécurité, les réseaux d'eau et d'énergie. Cela permet d'optimiser l'utilisation des ressources, d'améliorer la qualité de vie des habitants et de faciliter la prise de décision basée sur les données [20].

FIGURE 2.8 – Smart cities

2.6.4 Santé intelligente (Smart health)

L'IoT est utilisé pour révolutionner le domaine de la santé. Des dispositifs portables, tels que les montres connectées et les capteurs de suivi de la santé, collectent des données

sur les signes vitaux, les niveaux d'activité, le sommeil, etc. Ces données peuvent être surveillées à distance par les professionnels de la santé, permettant un suivi continu, une détection précoce des problèmes de santé et une gestion personnalisée des soins de santé [21].

FIGURE 2.9 – Smart health

2.6.5 Agriculture intelligente

L'IoT est appliqué dans l'agriculture pour améliorer l'efficacité et la productivité. Des capteurs intelligents surveillent les conditions du sol, de l'humidité, de la météo, et fournissent des informations précises sur les besoins des cultures. Cela permet une irrigation précise, une gestion optimisée des pesticides et des fertilisants, et une planification agricole basée sur les données, favorisant ainsi une agriculture plus durable et rentable.

FIGURE 2.10 – Agriculture intelligente

2.6.6 Transport intelligent

:

L'IoT est utilisé pour optimiser les systèmes de transport en rendant les véhicules et les infrastructures de transport plus connectés et intelligents. Des capteurs intégrés dans les

véhicules collectent des données sur le trafic, la géolocalisation, les conditions routières, et permettent une gestion en temps réel du trafic, des systèmes de stationnement intelligents, une navigation assistée, une sécurité renforcée, et une réduction des émissions de CO2 [22].

FIGURE 2.11 – Transport intelligent

Ces applications ne sont qu'un aperçu des nombreuses possibilités de l'IoT. Cette technologie continue d'évoluer rapidement, offrant de nouvelles opportunités pour améliorer différents secteurs de notre vie quotidienne et de l'infrastructure de nos sociétés.

2.7 Fonctionnalités de la maison intelligente

2.7.1 Confort et commodité

La maison intelligente offre des fonctionnalités qui améliorent le confort et la commodité de ses occupants. Cela peut inclure le contrôle automatisé de l'éclairage, de la climatisation, des appareils électroménagers et des divertissements, ainsi que la possibilité de programmer des scénarios personnalisés selon les préférences individuelles [23].

2.7.2 Économies d'énergie

Les systèmes de la maison intelligente permettent une gestion énergétique efficace, ce qui entraîne des économies d'énergie. Les capteurs intelligents surveillent la présence des occupants, ajustent la température et l'éclairage en conséquence, et optimisent l'utilisation des ressources énergétiques [23].

2.7.3 Communication et connectivité

La maison intelligente facilite la communication et la connectivité entre les appareils et les occupants. Des assistants vocaux, des applications mobiles et des interfaces utilisateur intuitives permettent de contrôler et de surveiller les équipements de la maison intelligente à distance [23].

2.7.4 Sécurité et surveillance

Les systèmes de sécurité avancés, tels que les caméras de surveillance, les détecteurs de mouvement, les alarmes et les serrures intelligentes, renforcent la sécurité et la surveillance de la maison. Les propriétaires peuvent surveiller leur propriété et recevoir des notifications en cas d'activités suspectes ou d'incidents potentiels [23].

2.8 Équipements de la maison intelligente

2.8.1 Éclairage intelligent

Les ampoules intelligentes permettent de contrôler l'éclairage à distance, d'ajuster l'intensité lumineuse et de programmer des scénarios d'éclairage personnalisésfigure : 2.13.

FIGURE 2.12 – Éclairage intelligent

2.8.2 Prises électriques intelligentes

Les prises intelligentes permettent de contrôler et de surveiller les appareils électriques connectés, offrant la possibilité d'allumer/éteindre à distance, de définir des minuteries et de surveiller la consommation d'énergie figure : 2.14.

FIGURE 2.13 – Transport intelligent

2.8.3 Chauffage intelligent

Les systèmes de chauffage intelligents ajustent automatiquement la température en fonction des préférences et des habitudes des occupants,

ce qui permet des économies d'énergie tout en maintenant le confort figure : 2.15.

FIGURE 2.14 – Chauffage intelligent

2.8.4 Serrure connectée (Smart Lock)

Les serrures intelligentes offrent un accès sécurisé à la maison sans clé physique, permettant de verrouiller/déverrouiller à distance et de suivre les activités d'accès figure : 2.16.

FIGURE 2.15 – Smart Lock

2.8.5 Dispositif Hydrao First

Il s'agit d'un pommeau de douche intelligent qui surveille la consommation d'eau et fournit des indicateurs visuels pour encourager l'économie d'eau figure : 2.17.

FIGURE 2.16 – Dispositif Hydrao First

2.8.6 Caméra de surveillance intelligente

Les caméras de surveillance intelligentes permettent une surveillance en temps réel, des enregistrements vidéo, des notifications d'activité et des fonctionnalités de détection de mouvement figure : 2.18.

FIGURE 2.17 – Caméra de surveillance intelligente

2.9 Les inconvénients de la maison intelligente

présente également quelques inconvénients qu'il convient de prendre en considération :

2.9.1 Dépendance aux technologies

La maison intelligente repose sur des dispositifs électroniques et des réseaux connectés, ce qui rend les occupants dépendants de leur bon fonctionnement. En cas de panne de courant, de dysfonctionnement du réseau ou de problèmes techniques, certaines fonctionnalités peuvent ne pas être accessibles.

2.9.2 Confidentialité et sécurité des données

Les dispositifs connectés collectent et traitent souvent des données personnelles et sensibles. Il existe des préoccupations concernant la confidentialité et la sécurité de ces données, notamment la possibilité de piratage ou de violation de la vie privée. Il est essentiel de prendre des mesures appropriées pour protéger les données dans un environnement de maison intelligente.

2.9.3 Complexité de l'installation et de la configuration

Mettre en place une maison intelligente peut être complexe et nécessiter des compétences techniques. L'installation et la configuration des différents équipements et dispositifs peuvent prendre du temps et nécessiter une certaine expertise, ce qui peut être décourageant pour certains utilisateurs.

2.9.4 Coût élevé

Les équipements et les dispositifs de la maison intelligente peuvent être coûteux, surtout si l'on souhaite mettre en place un système complet avec de nombreuses fonctionnalités. Les coûts d'achat, d'installation et de maintenance doivent être pris en compte avant de se lancer dans l'adoption de la maison intelligente.

Il est important de peser les avantages et les inconvénients avant de décider d'implémenter une maison intelligente. Il est également recommandé de faire appel à des professionnels pour l'installation et de prendre des mesures de sécurité appropriées pour protéger les données personnelles.

2.10 Conclusion

En conclusion, la maison intelligente offre de nombreuses opportunités et avantages, mais il est essentiel de prendre en compte les inconvénients potentiels et de faire des choix éclairés en matière d'adoption de la technologie de la maison intelligente. Une planification minutieuse, une sécurité adéquate et une compréhension des besoins individuels sont essentielles pour tirer le meilleur parti de la maison intelligente.

Chapitre 3

Conception et Simulation

3.1 Introduction

Ce chapitre se concentre sur la phase cruciale de conception de notre projet de maisons intelligentes. Après avoir exposé les concepts et objectifs dans le chapitre précédent, nous aborderons maintenant la planification et la modélisation du système.

Cette étape est essentielle pour formaliser et détailler les idées préliminaires, en mettant en évidence l'étude fonctionnelle approfondie du système.

Nous développons des modèles et des schémas détaillés, tout en utilisant des simulations pour évaluer et optimiser les configurations.

Ce chapitre jettera les bases solides nécessaires à la réalisation d'un système performant et fonctionnel de maisons intelligentes. ‘

3.2 Architecture globale du système

La Figure 3.1 présente l'architecture globale du système, mettant en évidence les différents composants tels que le noyau et le contrôleur, ainsi que les liens de communication qui les reposent. Ces liens de communication permettent l'échange de deux types d'informations principales :

- Les données (Data) requises par les capteurs et transmises au contrôleur (Controller). Ces données peuvent inclure des mesures physiques, des signaux provenant de l'environnement ou d'autres informations pertinentes
- Les réactions du système aux informations reçues, qui représentent l'état du système à un moment donné. Le contrôleur envoie alors un ensemble d'actions à effectuer aux actionneurs, qui sont des composants responsables de modifier l'environnement ou de prendre des mesures physiques spécifiques en réponse aux instructions reçues.

Capteurs	Actionneurs
Capteur de température et d'humidité	Climatiseur
Capteur de mouvement	Lampe
Capteur de mouvement	Sirène

TABLE 3.1 – Capteurs et actionneur du système

Cette architecture permet une interaction dynamique entre les différents composants du système embarqué, permettant ainsi le contrôle et la gestion des données et des actions. Cette approche est essentielle pour atteindre les objectifs de performance, de fiabilité et de réactivité requis par le système embarqué

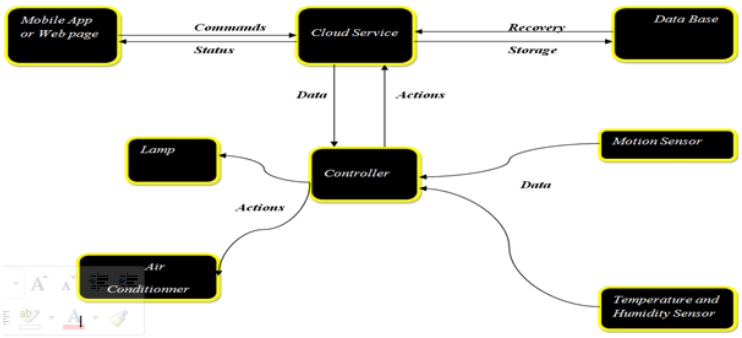

FIGURE 3.1 – Architecture globale du système

3.3 UML-RT

L'UML-RT (UML-Real Time) est un profil temps réel qui vise à simplifier la spécification d'architecture logicielle de plus en plus complexe pour les systèmes embarqués temps réel et de créer des systèmes réactifs qui interagissent avec leur environnement [24].‘

Les concepts UML-RT sont hérités de ceux définis dans le langage de Modélisation Orientée Objet Temps Réel (ROOM) et représenté à l'aide des mécanismes d'extensibilité UML. UML-RT permet à la fois la modélisation de la structure et la modélisation du comportement des systèmes temps réel [25]. ‘

La partie structurelle est représentée à l'aide de diagrammes de structure, tandis que la partie comportementale est représentée à l'aide de diagrammes de machine à 'état. Les concepts fondamentaux d'UML-RT sont des capsules, qui sont entités actives encapsulées pouvant s'exécuter en parallèle. Une capsule est une classe active qui définit un comportement concurrent (similaire à un thread). ‘

Son comportement peut être spécifié avec une machine à états (hiérarchique). Une capsule a une interface composée d'un ou plusieurs ports qui sont typés par un protocole.

Un protocole définit l'ensemble des messages qui peuvent être envoyés et reçus par le port et traités par sa machine d'état. ‘

Une capsule peut également avoir une structure interne constituée de parties statiques ou dynamiques qui contiennent des instances d'autres capsules. Les connecteurs sont utilisés pour relier leurs ports. Un récapitulatif sur les concepts de UML-RT est présenté dans le tableau 3.2.

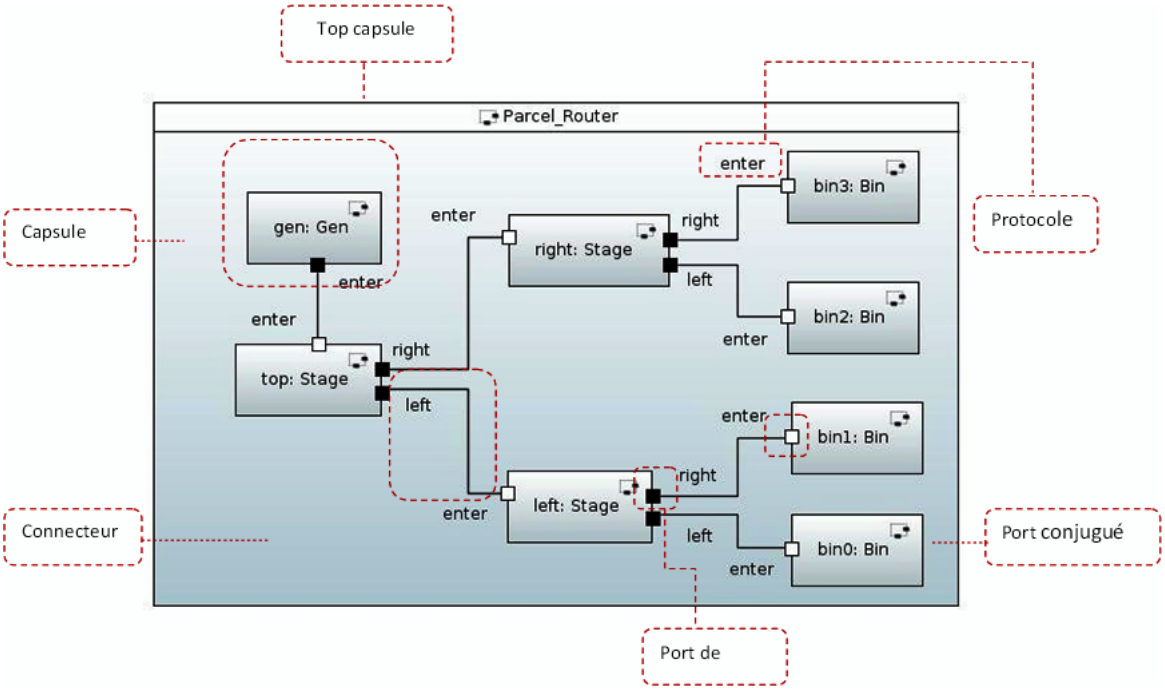

FIGURE 3.2 – Représentation graphique des concepts de structure d'UML-RT

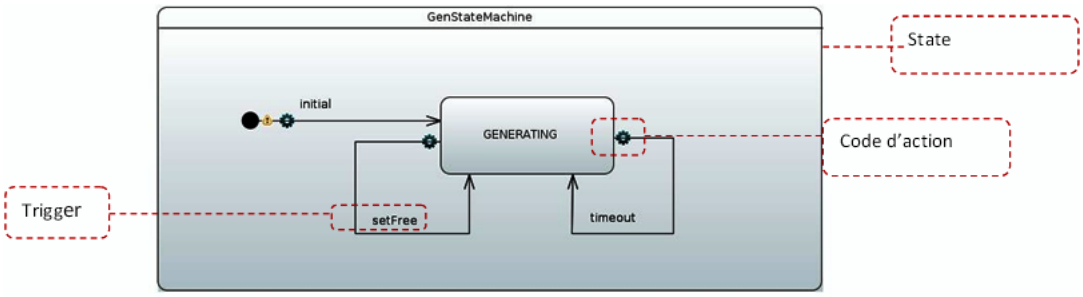

FIGURE 3.3 – Représentation graphique du concept de comportement d'UML-RT.

Concept	**Définition**
Modèle	Collection de définitions de capsules ;Capsule "Top" contenant la collection de capsules instances (parties).
Capsule	Peut contenir des attributs, des ports ou d'autres instances de capsules (parts). Son comportement est défini par une machine à états.
Port	« Objets frontières » possédé par une capsule ; Typé par un protocole ; Possède une opération « send » :- portName.msg(arg1,...,argn).send()Peut être de base : direction des massages comme déclarée dans le protocole. Conjugué : direction des massages comme déclarée dans le protocole estrenversée.
Machine à l'état	Transition : autorisé par les messages entrants ;Code d'action : peut contenir des instructions d'envoi qui envoient des messages sur certains ports.
protocole	Fourni des types pour les ports ; Défini : Les messages d'entrées représentant les services fournis par la capsule possédant le port ; Les messages de sortie représentant les services requis par la capsule possédant le port ; -Les messages entrants /sortant -un message peu transmettre des données ;

TABLE 3.2 – Définitions des concepts d'UML-RT.

3.4 Conception

3.4.1 Diagramme de cas d'utilisation

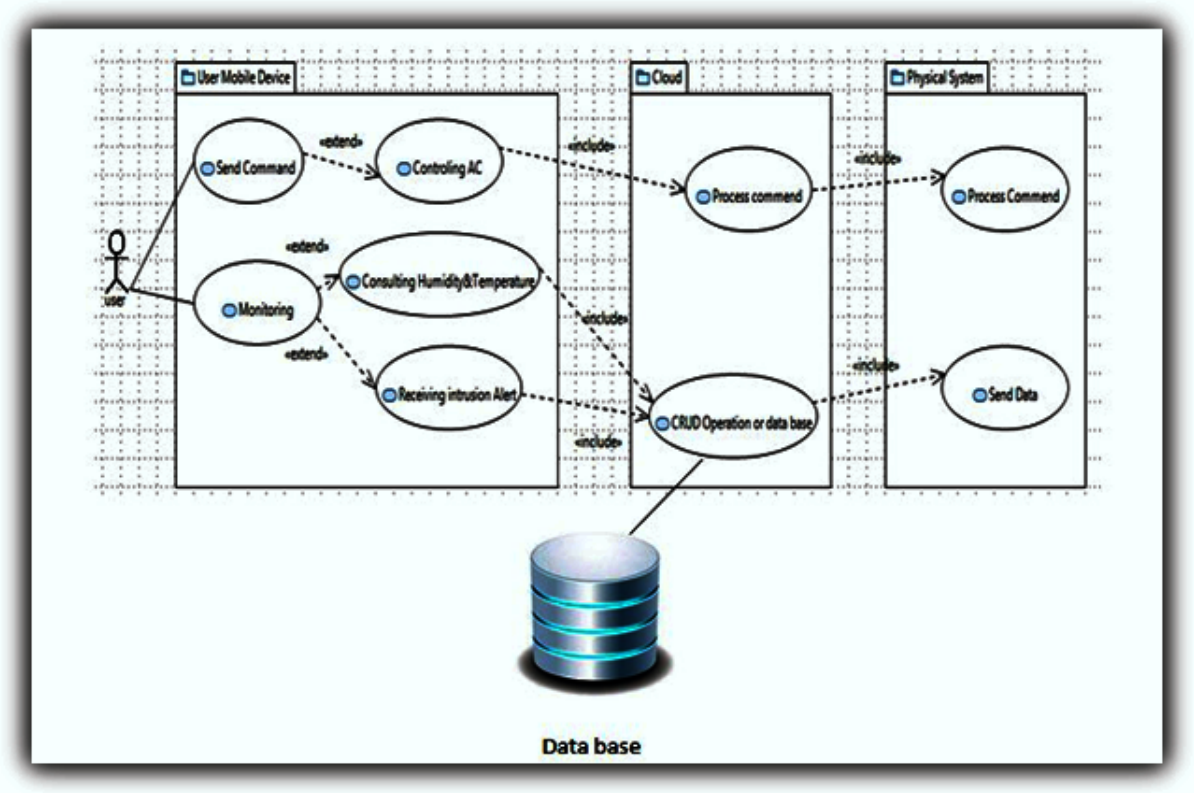

FIGURE 3.4 – Diagramme de cas d'utilisation du système.

La Figure 3.4 présente un diagramme de cas d'utilisation qui offre une représentation visuelle des fonctionnalités du système.

Ce diagramme permet de décrire les interactions entre les utilisateurs (acteurs) et le système

3.4.2 Définition des acteurs :

Notre système comporte un seul acteur qui est l'utilisateur du système qui interagit avec le système via une application mobile ou une page web.

3.4.3 Définition des cas d'utilisation :

Cas d'utilisation	Identification
Send command	L'utilisateur envoie une commande pour allumer ou éteindre le climatiseur.
Monitoring	Surveillance de la température et de l'humidité ainsi que les tentatives d'intrusion.

TABLE 3.3 – Description textuelle de cas d'utilisation

3.5 Diagramme de structure (Modèle/Top capsule)

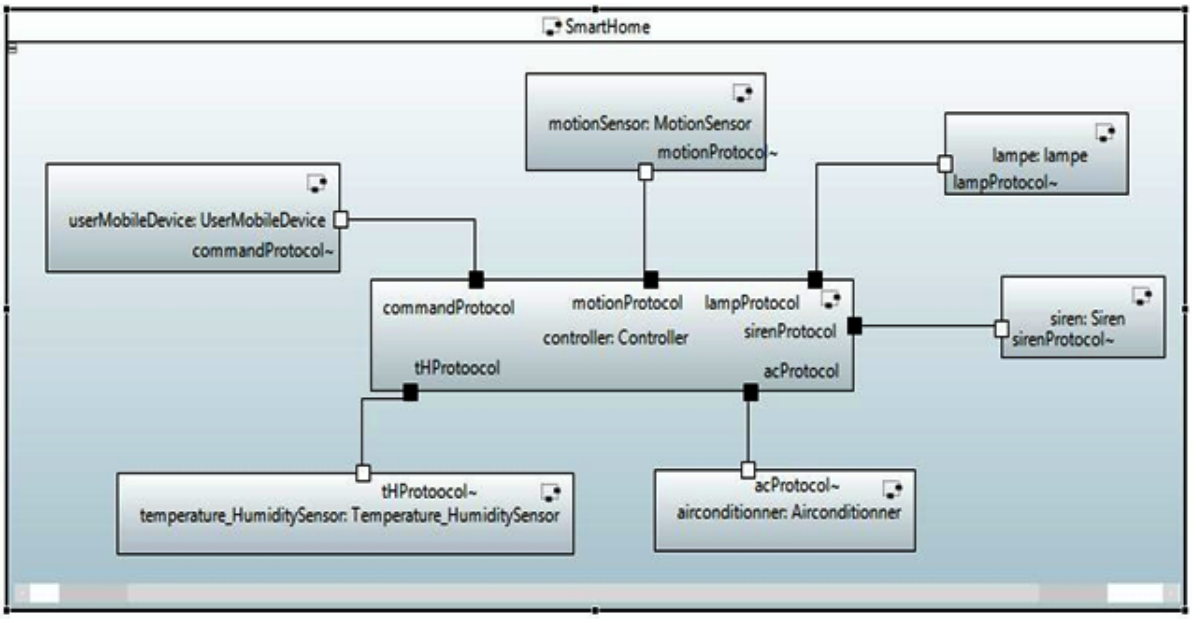

FIGURE 3.5 – Top capsule : SmartHome.

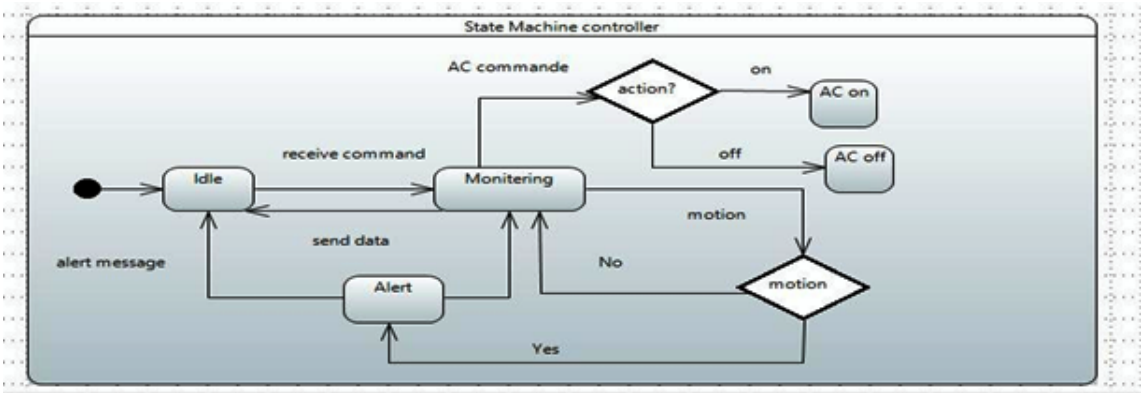

FIGURE 3.6 – Machine à état du contrôleur

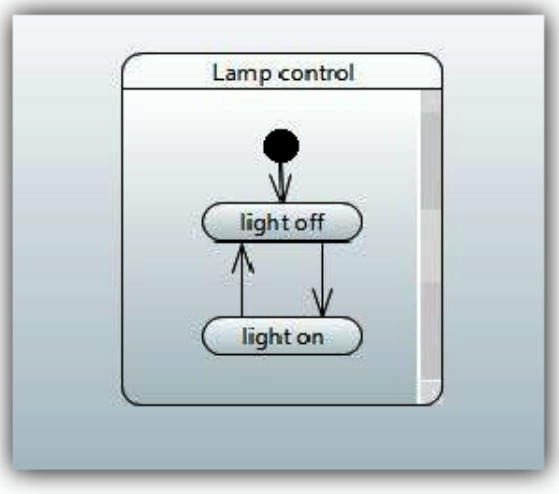

FIGURE 3.7 – Machine à état d'une lampe d'objet contrôlée

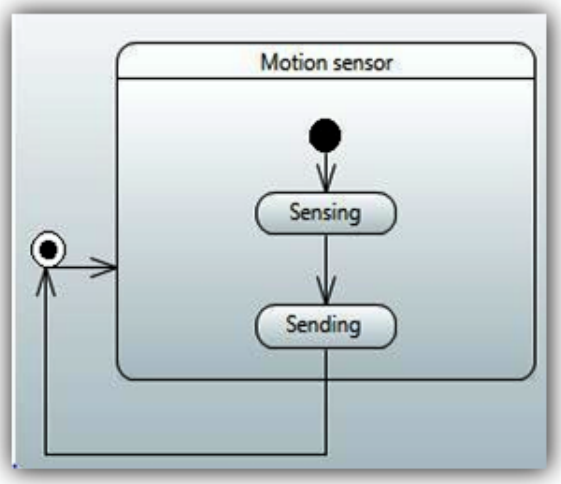

FIGURE 3.8 – Machine à état d'un capteur de mouvement

3.6 Simulation

Avant d'entreprendre la mise en œuvre de notre système, nous allons procéder à la simulation du système, ce qui constitue une étape essentielle pour vérifier la composante matérielle ainsi que son intégration avec la partie logicielle. Afin d'accomplir cette tâche, nous ferons usage des logiciels suivants

3.6.1 Simulation avec ISIS (Proteus)

PROTEUS, édité par Labcenter Electronics, est une suite logicielle spécialisée dans la conception assistée par ordinateur pour l'électronique. Cette suite comprend deux logiciels principaux : ISIS, qui facilite la création de schémas et la simulation électrique, ainsi que ARES, dédié à la conception de circuits imprimés. La fonctionnalité phare d'ISIS réside

dans la conception de schémas électriques, mais il offre également la possibilité de simuler ces schémas afin de détecter d'éventuelles erreurs dès la phase de conception. De plus, le logiciel permet de contrôler l'aspect graphique des circuits, ouvrant ainsi la voie à leur utilisation dans des documentations techniques [26].

Le logiciel ISIS de PROTEUS est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisé dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits figure 3.9

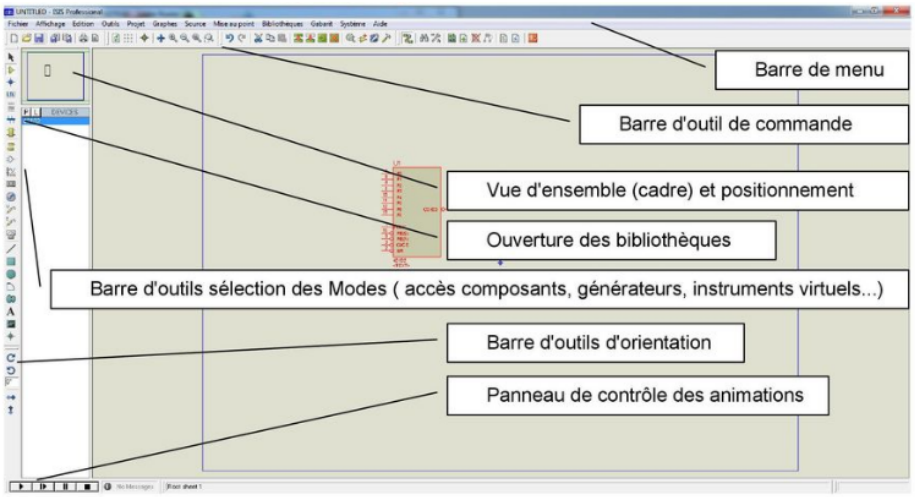

FIGURE 3.9 – présontation de l'interface ISIS (PROTERS)

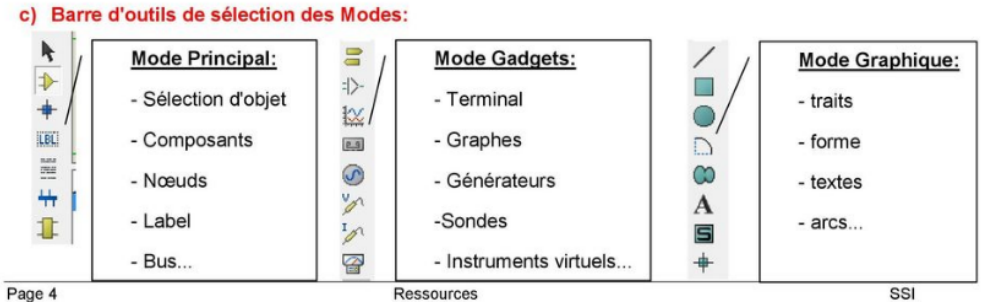

FIGURE 3.10 – présontation de l'interface Barre d'outil ISIS (PROTERS)

3.6.2 Les organigrammes de notre système

Avant de passer à la programmation, nous devons réaliser un organigramme qui explique le déroulement des différentes séquences.

L'organigramme est une représentation schématique et graphique normalisée de l'enchaînement des opérations et des décisions effectuées par un programme d'ordinateur des liens fonctionnels organisationnels d'un programme, etc.

Nous avons adopté pour construire notre programme d'un" maison intelligent '' 6 organigramme :

1. Organigramme éclairage automatique, voir figure 3.11
2. Organigramme fonction vérification de code, voir figure 3.12
3. Organigramme fonction open door, voir figure 3.13
4. Organigramme fonction RFID, voir figure 3.14
5. Organigramme Fonction gaz, voir figure 3.15
6. Organigramme principale, voir figure 3.16

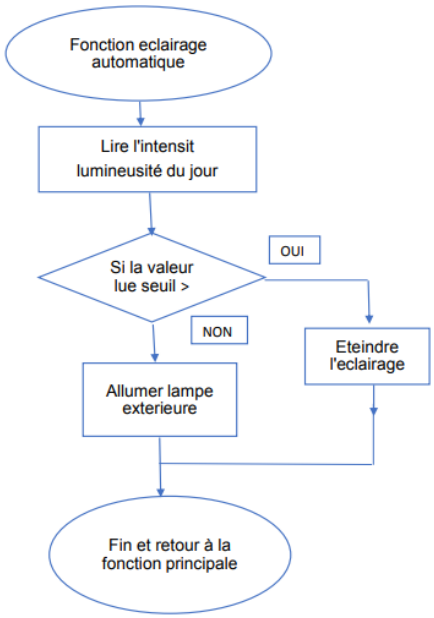

FIGURE 3.11 – Organigramme éclairage automatique, voir

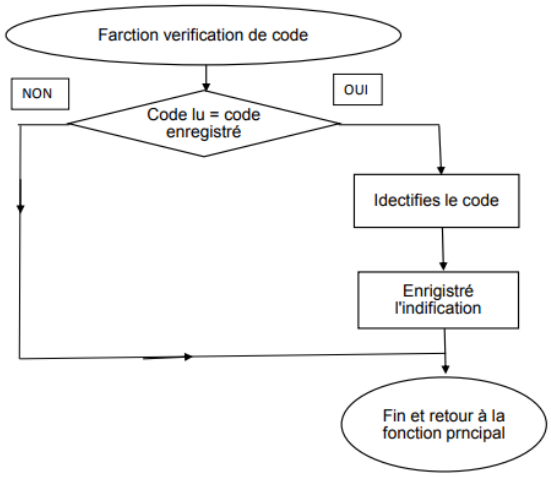

FIGURE 3.12 – Organigramme fonction vérification de code

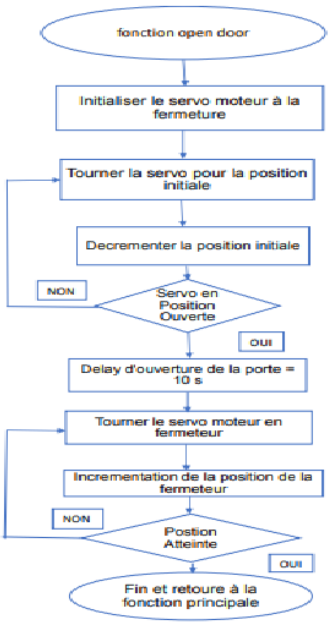

FIGURE 3.13 – Organigramme fonction open door

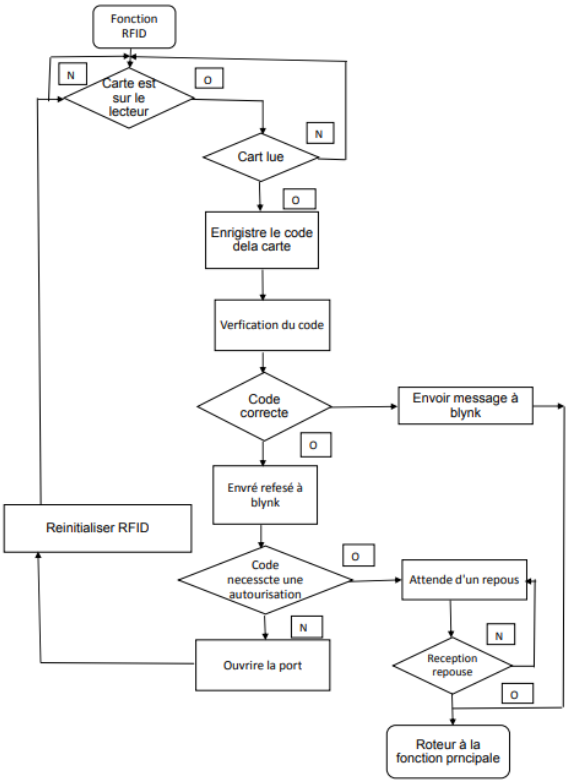

FIGURE 3.14 – Organigramme fonction RFID

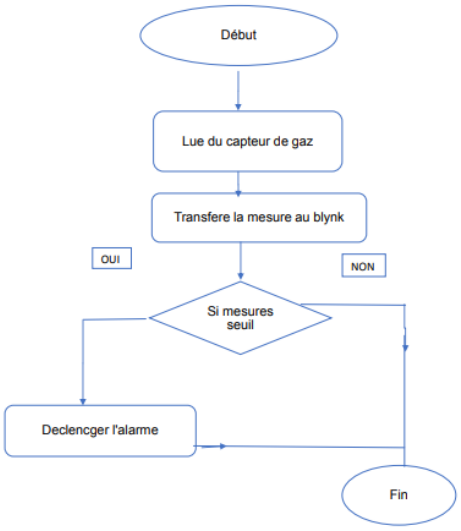

FIGURE 3.15 – Organigramme Fonction gaz

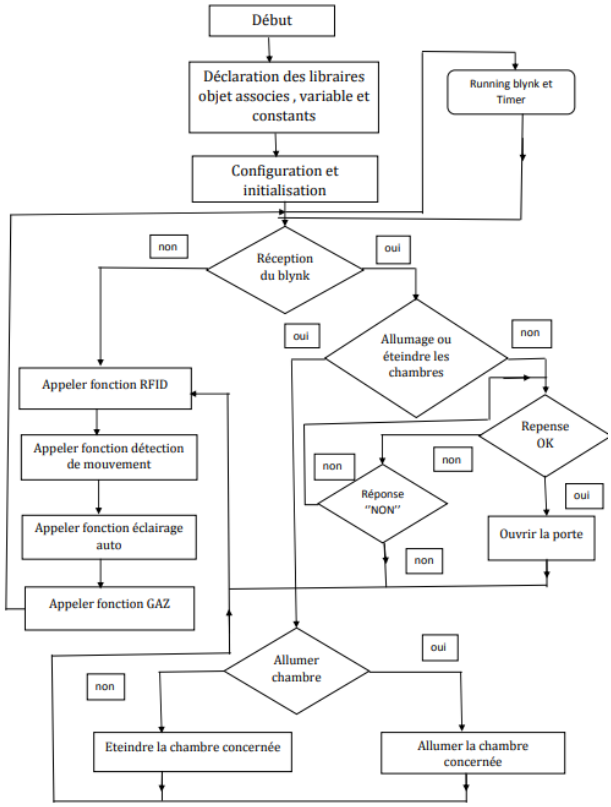

FIGURE 3.16 – Organigramme principale

3.7 Schéma globale de la simulation

La figure 3.17 montre le schéma global de branchement de la carte Arduino avec les différents capteurs via le logiciel ISIS (Proteus).

les figures 3.18 , 3.19 et 3.20 illustrent chaque detail de la partie du shema globale (PIR, lampe et climatiseur, DHT)

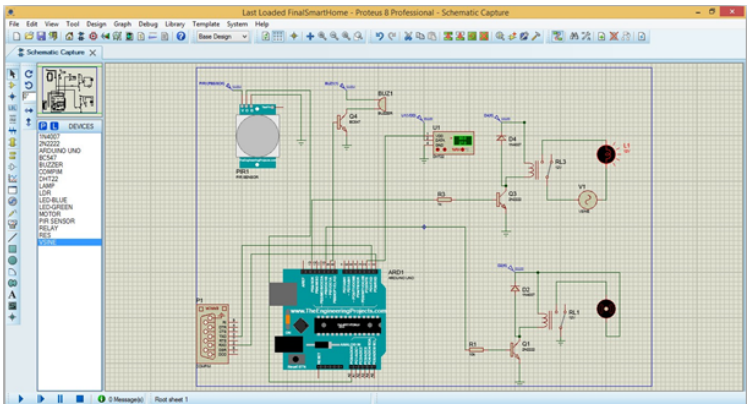

FIGURE 3.17 – Schéma globale de la simulation.

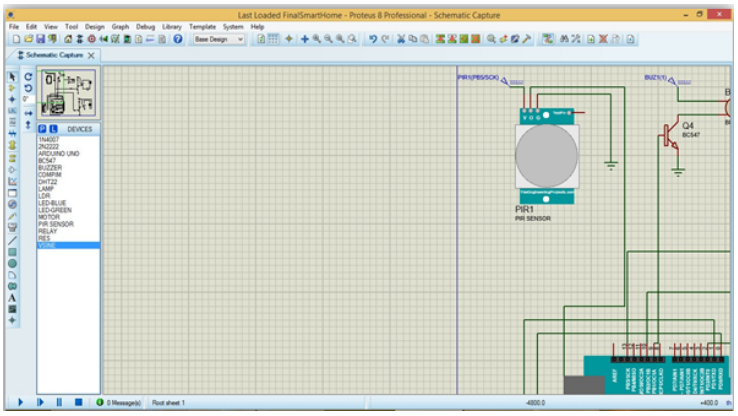

FIGURE 3.18 – Model de capteur de mouvement(PIR).

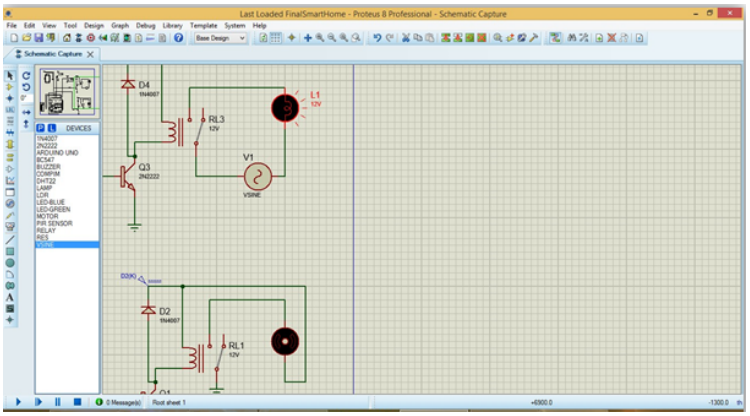

FIGURE 3.19 – Lampe et moteur utilisée comme un climatiseur

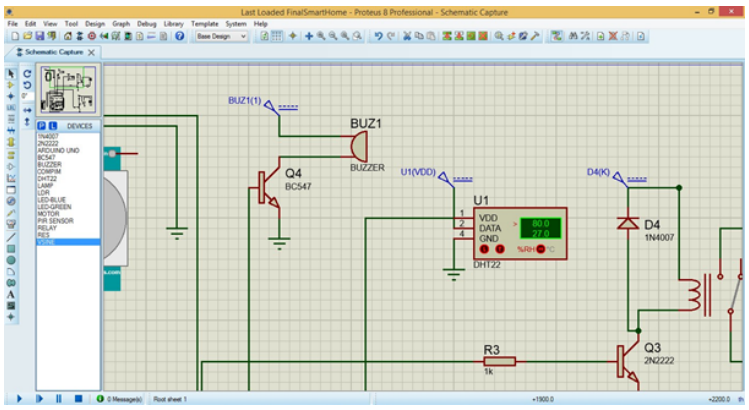

FIGURE 3.20 – Capteur DHT 22 (humidité et température).

3.8 Conclusion

Dans ce chapitre nous avons faits une modélisation et simulation des éléments avec des périphériques en utilisant l'Internet of Things. Nous avons également présenté le modèle UML-RT et enfin pour simulation nous avons utilisé le logiciel ISIS (Proteus).

Chapitre 4

Implémentation

4.1 Introduction

Après avoir terminé la conception du projet, on passe maintenant à son implémentation. Ce chapitre décrit les différents outils et les langages de programmation utilisés, ainsi que le programme et la manière dont le système est réalisé.

4.2 Matérielle utilisés

4.2.1 la carte Arduino UNO

L'Arduino Uno est un microcontrôleur programmable qui permet, comme son nom l'indique, de contrôler des éléments mécaniques : systèmes, lumières, moteurs, etc. Cette carte électronique permet donc à son utilisateur de programmer facilement des choses et de créer des mécanismes automatisés, sans avoir de connaissances particulières en programmation. Il est un outil pensé et destiné aux inventeurs, artistes ou amateurs qui souhaitent créer leur propre système automatique en le codant de toute pièce, elle est devenue le symbole de l'univers Arduino Figure 4.1.

FIGURE 4.1 – La carte arduinoUno [27]

Il vous suffit de connecter votre carte électronique sur votre ordinateur (Windows, Mac ou Linux) et vous pouvez commencer à programmer à partir du logiciel Arduino. Votre seule limite est votre imagination ! L'ArduinoUno dispose de caractéristiques techniques suffisantes pour bien commencer la programmation : 14 broches d'entrées/sorties digitales, dont six sont utilisables en PWM, de 6 broches d'entrées analogiques, d'une connectique USB, d'une connectique d'alimentation, d'un port ICSP et d'un bouton RESET Figure 4.2.

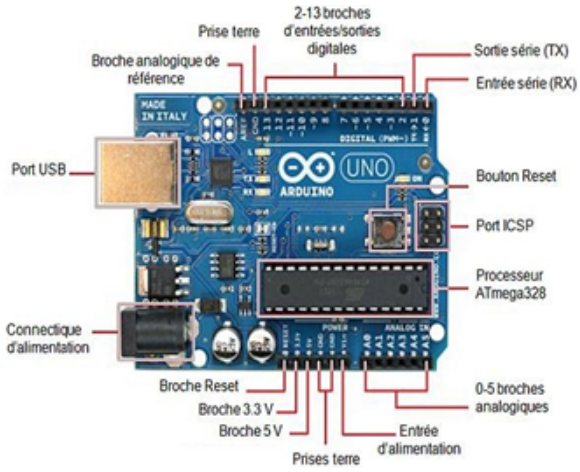

FIGURE 4.2 – Description des entrées/sorties de la carte ArduinoUno [28]

4.2.2 Les entrées / sorties d'Arduino UNO :

Les entrées analogiques A0 à A5 :

Contrairement aux entrées/sorties numériques qui ne peuvent prendre que deux états HAUT et BAS, ces six entrées peuvent admettre un millier de valeurs (1024 exactement) analogiques comprises entre 0 et 5 Volts. Nous pourrons donc avoir des valeurs de tension précises à 5 mV près (5V/1024) Figure 4.3.

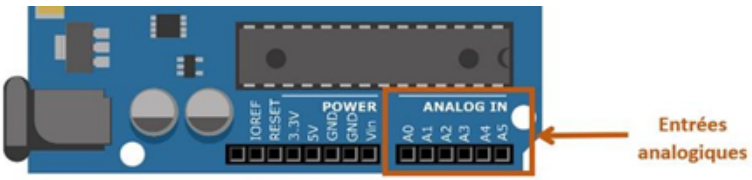

FIGURE 4.3 – Les entrées/sorties analogiques

Les entrées/sorties numériques : de D0 à D13 :

Chacun des connecteurs D0 à D13 peut être configuré par programmation en entrée ou en sortie, nous pouvons donc avoir par exemple les connecteurs 2 et 3 configurés comme des entrées et les connecteurs 7, 8 et 9 configurés comme des sorties. Il est par conséquent

possible de connecter côte à côte des capteurs logiques (interrupteurs par exemple) aux connecteurs 2 et 3 et des actionneurs aux connecteurs 7, 8 et 9 Figure 4.4.

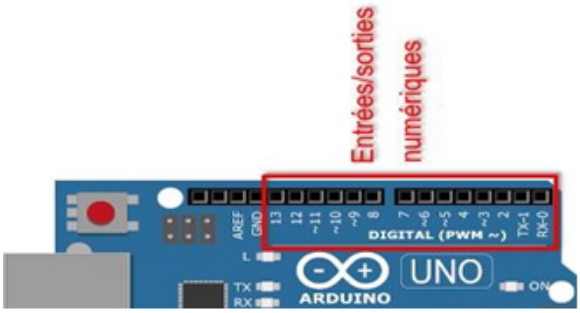

FIGURE 4.4 – Les entrées/sorties numériques

4.2.3 ESP8266 / ESP-07 (WI-FI)

Définition :

c'est un module émetteur-récepteur série wi-fi, basé sur soc ESP8266. le SOC a intégré la pile de protocole TCP / IP. c'est l'interface de communication série TTL et ses paramètres peuvent être définis par la commande AT. il est largement utilisé dans les réseaux, projet de maison intelligente lorsqu'il est connecté au routeur wifi. il peut être utilisé pour la surveillance à distance des appareils ménagers, de la température et de l'humidité de la chambre, pour contrôler les appareils ménagers et la voiture intelligente à l'aide du téléphone mobile.

FIGURE 4.5 – Esp8266 / ESP-07 (WIFI)

Les caractéristiques de ESP8266 :

— arduino UNO R3 et la carte compatible peuvent se connecter directement à ce module.

— Tension De Travail : 4.5V - 5.5v (régulateur À Bord 3.3V LDO)

— Courant De Travail : 240ma (max)

— débit en bauds du port série : 115200 (valeur par défaut), modifiable en d'autres valeurs par la commande AT

— Format De Communication Série : 8N1

— bouton de réinitialisation intégré

— commutateur intégré pour sélectionner le mode UART ou le mode de mise à jour du firmware

— type d'antenne : une antenne en céramique intégrée est disponible et vous pouvez également utiliser une antenne externe.

— Mode Réseau Sans Fil : Station / softap / Softap + Station

— Critères De Connexion Sans Fil : 802.11 B / g / n

— Wi-fi @ 2.4 Ghz, Support Du WPA / Mode de sécurité WPA2

— il est très facile de concevoir des projets avec la commande AT

louloi

4.2.4 Bluetooth (HC-06)

Modules HC-05 et HC-06

Les modules HC-05 (maître) et HC-06 (esclave) sont des circuits extrêmement fréquents et bon marché, parfaitement adaptés à une utilisation avec Arduino.

Mode maitre : l'association (ou appairage) avec un autre périphérique doit se faire depuis le module.

Mode esclave : l'association avec un autre périphérique doit se faire depuis l'autre périphérique.

Fonctionnement

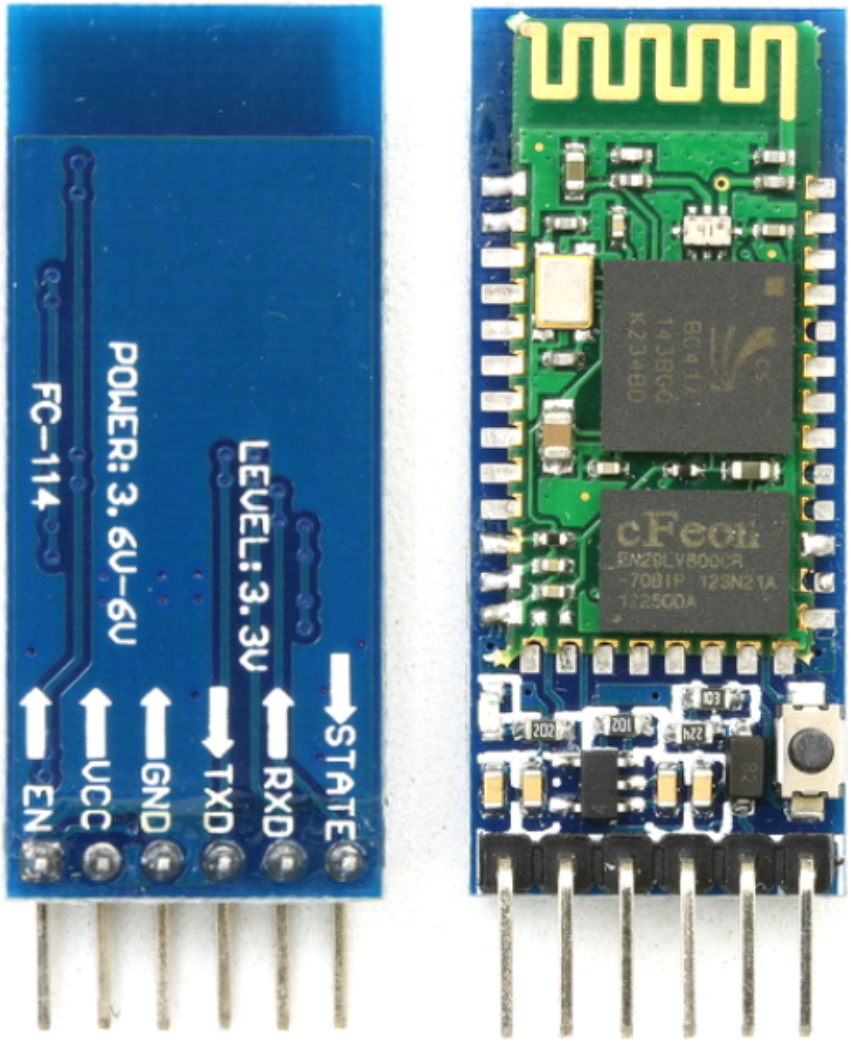

FIGURE 4.6 – Bluetooth (HC-05/HC-06)

4.3 Partie software

4.3.1 Le logiciel Arduino

Comme on a précisé avant, pour utiliser l'Arduino, il faut le programmé. Dans un premier temps, on a installé le logiciel d'Arduino, d'après le lien https ://www.arduino.cc/ Une fois cela fait, on a lancé le logiciel, Le code que vous rentrerez sur l'éditeur de texte vierge de votre logiciel Arduino, ça sera la manière de faire comprendre au microcontrôleur ce que vous souhaitez instaurer comme programme.Vous pourrez directement constater la simplicité de l'interface de ce dernier :

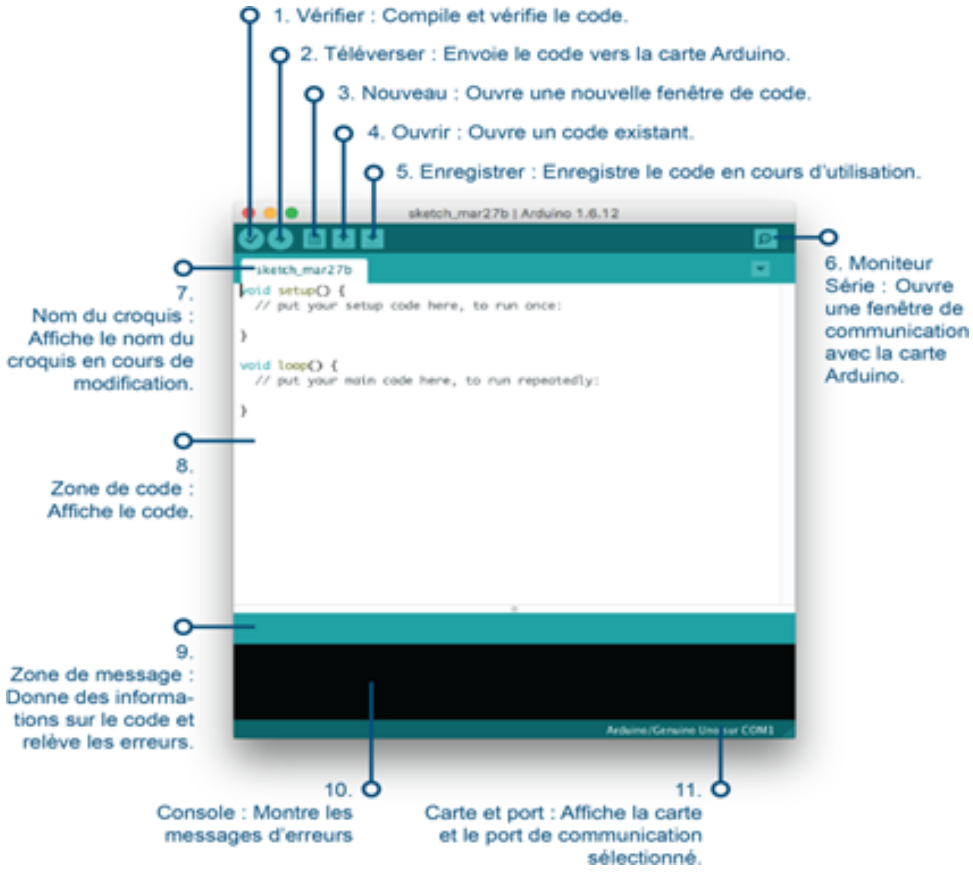

FIGURE 4.7 – Fenêtre du logiciel Arduino [29]

4.3.2 L'application Blynk

Blynk est conçu pour l'Internet des Objets (iot). Avec lui il est possible de contrôler un hardware à distance, afficher des données de capteur, stocker des données, les visualiser... Les hardwares les plus utilisés avec la plateforme Blynk sont les ESP8266, le Raspberry pi et l'Arduino.

Il y a trois composants majeurs dans la plateforme :

— **Application Blynk :** vous permet de créer de fantastiques interfaces pour vos projets en utilisant différents widgets que nous fournissons.

— **Serveur Blynk :** responsable de toutes les communications entre le smartphone et le hardware. Vous pouvez utiliser notre CloudeBlynk ou faire tourner votre Serveur privé Blynk localement. C'est open-source, ça peut facilement gérer des milliers de périphériques et peut même être démarré sur une Raspberry Pi.

— **Bibliothèque Blynk** pour toutes les plateformes hardware populaires - active la communication avec le serveur et traite toutes les commandes entrantes et sortantes. Maintenant imaginez : chaque fois que vous pressez un Bouton sur l'application Blynk, le Cloud trouve magicalement un chemin vers votre hardware. Ça marche de la même manière dans l'autre sens et tout se déroule en un clignotement.

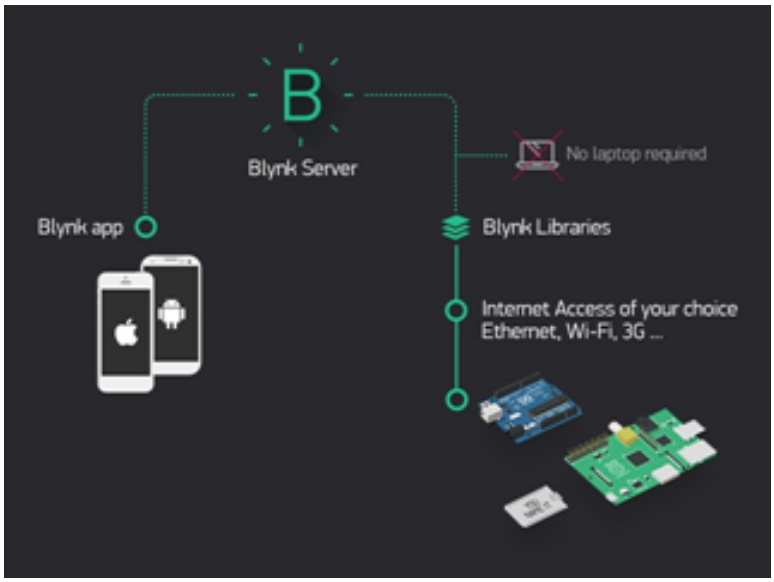

FIGURE 4.8 – Le Chemin d'application BLYNK vers le hardware.[30]

4.4 Partie hardware

4.4.1 Les capteurs

Un capteur est un dispositif permettant de détecter, en vue de le quantifier et de le représenter, un phénomène physique sous la forme d'un signal, généralement électrique. Le capteur se différencie du détecteur et du senseur par sa possibilité de délivrer une grandeur physique directement utilisable pour une mesure ou une commande.

Le capteur est caractérisé par sa fonction : s = F(m) où s est la grandeur de sortie ou la réponse du capteur

Les caractéristiques d'un capteur

- Etendue de mesure : Valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : Plus petite variation de grandeur mesurable par le capteur.
- Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- Rapidité : Temps de réaction du capteur. La rapidité est liée à la bande passante.
- Linéarité : représente l'écart de sensibilité sur l'étendue de mesure.

Choix de Capteurs

Une maison intelligente vise à fournir des fonctions de sécurité et de confort, elle doit donc être en mesure, au minimum, d'obtenir des relevés de température et de détecter s'il existe un risque d'incendie dû à la détection de fumée ou à la présence de gaz inflammable. Nous avons donc besoin de plusieurs capteurs pour notre bouclier, tels que le capteur de détection des gaz inflammables et de la fumée ainsi que le capteur de température et d'humidité afin que le système puisse contrôler le climat à l'intérieur de la maison.

4.4.2 Capteur de température et humidité DHT22

On a utilisé le capteur de température et humidité DHT22, qui communique avec un microcontrôleur via un port série. Le capteur est calibré et ne nécessite pas de composants supplémentaires pour pouvoir être utilisé. Il est capable de mesurer des températures de -40 à +80°C avec une précision de +/- 0,5 °C et des taux d'humidité relative de 0 à 100 RH avec une précision de +/- 2 RH.

FIGURE 4.9 – Capteur de température et humidité DHT22.

4.4.3 Climatiseur (Ventilateur)

La ventilation est un élément essentiel à l'intérieur de la maison, la solution est de programmer une température qui s'adapte à chaque pièce et notamment en fonction de moments de la journée, nuit et jour. Si le seuil de température (27°C) est dépassé par exemple, un ventilateur est activé pour simuler le fonctionnement du climatiseur. De plus, si du gaz et de la fumée sont détectés, un ventilateur est activé dans la direction opposée pour évacuer la fumée . Nous avons utilisé de petits ventilateurs comme indiqué sur la figure 4.9

FIGURE 4.10 – Ventilateur a 12V.

4.4.4 Breadboard (plaque d'essai)

La plaque d'essai est une plaque en plastique isolant et pleine de trous, son objectif est de pouvoir faire des connexions simplement entre des composants (résistances, LEDs, Capacités, etc . . .) sans souder. Elle est idéale pour tester un circuit ou réaliser un montage temporaire. Il en existe de plusieurs types, mais le principe reste le même. La breadboard possède des petits trous pour insérer les composants. Les trous, regroupés par lignes de 5 sont reliés entres eux à l'intérieur de la breadboard.

Certaines breadboard possèdent en plus, de chaque coté, 2 colonnes destinées à connecter les bornes + et – de l'alimentation

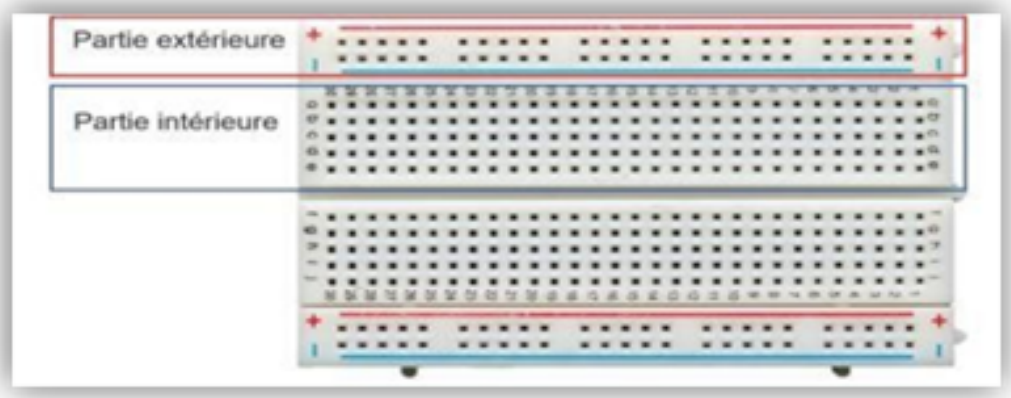

FIGURE 4.11 – Breadboard

4.4.5 LED

Une LED (en français : DEL : diode électroluminescente) est un composant opto-électronique, sa facilité de montage sur un circuit imprimé, sa faible consommation, sa résistance mécanique, sa petite taille, sa longue durée de vie et d'autres caractéristiques font la LED un composant de plus en plus inévitable .

La LED possède 2 broches :

Le long : positive.

Le petit : négative.

FIGURE 4.12 – Led

4.4.6 Fils de connexion

Fils de connexion (Câbles de prototypage) utilisés pour connecter électriquement des composants.

FIGURE 4.13 – Câbles de prototypage

4.4.7 Câble USB

Un câble USB utilisé pour connecter l'Arduino à l'ordinateur et/ou à la source d'alimentation.

FIGURE 4.14 – Câbles de usb

4.5 Explication de code arduino

Dans la phase initiale de la programmation, l'instruction **includeest** utilisée pour insérer des bibliothèques externes dans le code source. Cela permet d'accéder à un large éventail de bibliothèques C standard ainsi qu'à des bibliothèques demandées développées pour Arduino. En les incluant, vous pouvez tirer parti de fonctionnalités prédéfinies et faciliter le développement en utilisant des fonctionnalités déjà implémentées plutôt que de les coder vous-même.

La bibliothèque incluse est nécessaire pour interfacer la partie matérielle avec la carte Arduino. Les bibliothèques utilisées dans ce projet sont fournies dans Code Reassemble :

```
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <DHT.h>
```

FIGURE 4.15 – Code1

Dans le Code 02, les variables globales ont été ralenties pour être utilisées plus tard dans le programme lors du stockage des données acquises à partir des capteurs. Aussi, il est important de définir le type, le nom et la valeur initiale de la variable :

```
#define redPin 7
#define bluePin 8
#define greenPin 9
#define MQ135 A0
int standardAir =200;   // The standard of clean air.
float Temperature, Humidity;
SoftwareSerial bluetooth(2, 4); // RX, TX

LiquidCrystal_I2C lcd(0x27,16,2);
DHT dht(10, DHT11);
```

FIGURE 4.16 – Code2

La fonction setup() est une fonction spéciale du code Arduino qui n'est appelée qu'une seule fois au démarrage du programme. Principalement utilisé pour la configuration initiale nécessaire avant le démarrage du programme principal.

```
void setup() {
  bluetooth.begin(9600); // start the bluetooth uart at 9600 which is its default
  Serial.begin(9600);
  dht.begin();          // Initialize DHT sensor
  lcd.init();           // Initialize LCD screen
  lcd.backlight();      // Turn on LCD light
  pinMode(redPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(MQ135,INPUT);


}
```

FIGURE 4.17 – Code3

Ces quelques lignes de code servent à lire les valeurs de température et d'humidité du capteur DHT. Ces valeurs peuvent ensuite être utilisées dans le programme pour afficher des données sur l'écran LCD, envoyer des données via Bluetooth ou faire tout ce qui est nécessaire pour le projet.

```
Temperature = dht.readTemperature();
Humidity = dht.readHumidity();
```

FIGURE 4.18 – Code4

Les lignes de code suivantes sont utilisées pour afficher les valeurs de température et d'humidité sur l'écran LCD et les envoyer via Bluetooth :

```
lcd.setCursor(0,0);
lcd.print("Temp: " + String(Temperature)); // Print Temperature on the screen
lcd.setCursor(0,1);
lcd.print("Humidity: " + String(Humidity));// Print the Humidity on the screen
bluetooth.print(String(Temperature) + ":" + String(Humidity) + ":"); // Send temp and humidity to android via bluetooth
```

FIGURE 4.19 – Code5

Les lignes de code suivantes sont utilisées pour lire la valeur du capteur de gaz (gaz sensor), comparer cette valeur à un seuil prédéfini et envoyer un message via Bluetooth si la valeur dépasse ce seuil :

```
// Read the value of Gaz sensor.
int gazValue =analogRead(MQ135);
Serial.println(gazValue);
if(gazValue > standardAir)    // If Gaz detected over passed the standard clean air then send "gaz_detected" to android
{
  bluetooth.print("gaz_detected");
}
```

FIGURE 4.20 – Code6

Les lignes de code suivantes sont utilisées pour lire les données du module Bluetooth et contrôler les broches de sortie en fonction des commandes reçues.

Ces lignes de code permettent de recevoir des commandes via Bluetooth, de les interpréter et de piloter des broches de sortie (LED) en fonction des commandes. Cela vous permettra de contrôler vos lumières connectées à Arduino à partir de n'importe quel appareil connecté par Bluetooth.

```
    bluetooth.print("gaz_detected");
  }

  if(bluetooth.available()){


    char msg = bluetooth.read();

    switch(msg){
      case '1':
        digitalWrite(redPin, HIGH);
      break;
      case '2':
        digitalWrite(redPin, LOW);
      break;
      case '3':
        digitalWrite(bluePin, HIGH);
      break;
      case '4':
        digitalWrite(bluePin, LOW);
      break;
      case '5':
        digitalWrite(greenPin, HIGH);
      break;
      case '6':
        digitalWrite(greenPin, LOW);
      break;
```

FIGURE 4.21 – Code7

delay(500) ;ajoute une pause de 500 millisecondes (ou 0,5 seconde) dans l'exécution du programme. Cela signifie que le programme va attendre pendant 500 millisecondes

```
delay(500);
```

FIGURE 4.22 – code8

4.5.1 Explication de code android

Ce code suivant est pour une activité dans une application Android appelée "Gaz-WarningActivity". Il affiche une alerte pour une alarme de gaz et joue un son continu jusqu'à ce que l'activité soit arrêtée. L'activité contient un bouton "btnDone" qui permet de fermer l'activité lorsque l'utilisateur appuie dessus. Lorsque l'activité démarre, le son de l'alarme de gaz est activé en boucle. Lorsque l'activité s'arrête, le son est arrêté et libéré.

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    supportRequestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.activity_gaz_warning);
    setFinishOnTouchOutside(false);

    btnDone =(Button) findViewById(R.id.btn_warning_done);
    btnDone.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { finish(); }
    });

    mediaPlayer = MediaPlayer.create( context: this, R.raw.fire_alarm);
    mediaPlayer.setLooping(true);

}
```

FIGURE 4.23 – code9

FIGURE 4.24 – code10

```java
@Override
protected void onStop() {
    super.onStop();
    if (mediaPlayer != null) {
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
    MainActivity.isGazAlarmDisplayed = false;
}

@Override
protected void onStart() {
    super.onStart();
    if (mediaPlayer != null)
        mediaPlayer.start();
    MainActivity.isGazAlarmDisplayed = true;
}
```

FIGURE 4.25 – code11

Ce code suivant représente une classe appelée "ConnectedThread" dans une application Android utilisant Bluetooth. Elle est responsable de la communication avec un périphérique distant via une socket Bluetooth.

```
@Override
public void run() {
    byte[] buffer = new byte[1024];  // buffer store for the stream
    int bytes; // bytes returned from read()
    // Keep listening to the InputStream until an exception occurs
    while (true) {
        try {
            // Read from the InputStream
            bytes = mmInStream.available();
            if(bytes != 0) {
                buffer = new byte[1024];
                SystemClock.sleep( ms: 500); //pause and wait for rest of data. Adjust this depending
                bytes = mmInStream.available(); // how many bytes are ready to be read?
                bytes = mmInStream.read(buffer,  off: 0, bytes); // record how many bytes we actually
                mHandler.obtainMessage(MainActivity.MESSAGE_READ, bytes,  arg2: -1, buffer)
                        .sendToTarget(); // Send the obtained bytes to the UI activity
            }
        } catch (IOException e) {
            e.printStackTrace();

            break;
        }
    }
}
```

FIGURE 4.26 – code12

Ce code suivant XML représente la mise en page d'une vue dans une activité Android. Il utilise un CardView pour afficher une alerte de fuite de gaz.

```
@Override
public void run() {
    byte[] buffer = new byte[1024];  // buffer store for the stream
    int bytes; // bytes returned from read()
    // Keep listening to the InputStream until an exception occurs
    while (true) {
        try {
            // Read from the InputStream
            bytes = mmInStream.available();
            if(bytes != 0) {
                buffer = new byte[1024];
                SystemClock.sleep( ms: 500); //pause and wait for rest of data. Adjust this depending
                bytes = mmInStream.available(); // how many bytes are ready to be read?
                bytes = mmInStream.read(buffer,  off: 0, bytes); // record how many bytes we actually
                mHandler.obtainMessage(MainActivity.MESSAGE_READ, bytes,  arg2: -1, buffer)
                        .sendToTarget(); // Send the obtained bytes to the UI activity
            }
        } catch (IOException e) {
            e.printStackTrace();

            break;
        }
    }
}
```

FIGURE 4.27 – code13

ce code suivant XML représente la structure d'une mise en page LinearLayout contenant des boutons, des cartes représentant des pièces, des informations de température, d'humidité et d'état, ainsi qu'une liste de périphériques. Il est utilisé pour créer une interface utilisateur dans une application Android pour contrôler des dispositifs et afficher des informations liées à des pièces spécifiques.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingLeft="16dp"
    android:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingBottom="16dp"
    tools:context="com.mcuhq.simplebluetooth.MainActivity">
```

FIGURE 4.28 – code14

```xml
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp"
    android:gravity="center">
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center">
        <ImageButton
            android:id="@+id/btn_bth"
            android:layout_width="60dp"
            android:layout_height="60dp"
            android:tint="@color/bthOff"
            android:src="@drawable/ic_bluetooth_disabled"/>
    </LinearLayout>
```

FIGURE 4.29 – code15

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="2dp">
    <ImageView
        android:id="@+id/iv_room1"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:src="@drawable/ic_light_on"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Room 1"
        android:textSize="18sp"
        android:layout_gravity="center"
        android:textColor="@android:color/black"/>
</LinearLayout>
```

FIGURE 4.30 – code16

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="0.1"
        android:ellipsize="end"
        android:maxLines="1"
        android:text="Temperature:"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/tv_temp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.9"
        android:ellipsize="end"
        android:maxLines="1"
```

FIGURE 4.31 – code17

```
    <ListView
        android:id="@+id/devices_list_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:choiceMode="singleChoice" />
</LinearLayout>
```

FIGURE 4.32 – code18

4.6 conclustion

En conclusion, ce chapitre offre une vue d'ensemble détaillée des différents composants matériels et logiciels nécessaires pour mettre en place un système de maison intelligente. Il fournit des informations sur les fonctionnalités et l'utilisation de chaque composant, ainsi que des explications sur la manière dont ils interagissent les uns avec les autres.

Cette compréhension approfondie est essentielle pour assurer le succès de l'implémentation d'une maison intelligente.

Conclusion générale et perspectives

Au fil des avancées technologiques et des recherches sur les systèmes embarqués et l'Internet des objets (IoT), l'industrie de la maison intelligente a bénéficié considérablement des contributions issues de ces travaux de recherche.

Dans le cadre de ce projet, nous avons entrepris la conception et la mise en œuvre d'un système de gestion efficace pour certains appareils électriques tels que les lumières et les climatiseurs, en utilisant les principes de l'Internet des objets.

L'objectif principal de notre travail était de prévenir les pertes d'énergie et de réduire la consommation électrique des appareils en éteignant ces derniers lorsqu'ils ne sont pas utilisés. Cela permet non seulement d'économiser de l'énergie, mais également de contribuer à la réduction des coûts énergétiques pour les utilisateurs.

En envisageant les travaux futurs dans le domaine de la gestion de l'énergie domestique, différentes approches peuvent être explorées. Une démarche évidente consisterait à installer un wattmètre Arduino, qui permettrait de mesurer la consommation énergétique de la maison avant et après l'implémentation de notre système. Cette mesure comparative permettrait de démontrer la faisabilité et l'efficacité de notre solution.

Par ailleurs, nous encourageons vivement l'adoption d'énergies renouvelables, comme les panneaux solaires, au sein des foyers. Cela permettrait d'améliorer l'efficacité énergétique globale du système et de réduire les émissions de dioxyde de carbone, contribuant ainsi à la préservation de l'environnement.

pour conclure, ce projet a permis de concevoir et de mettre en œuvre un système de gestion efficace pour les appareils électriques dans le contexte d'une maison intelligente. Les travaux futurs devraient se concentrer sur la mesure précise de la consommation d'énergie, ainsi que sur l'adoption de sources d'énergie renouvelables, afin de promouvoir une gestion énergétique durable et respectueuse de l'environnement.

Bibliographie

[1] *L'histoire des systèmes embarqués - Optima Junior Entreprise*, https://optimaje.com/blog/2019/01/lhistoire-des-systemes-embarques/.

[2] *Définition | Système embarqué | Futura Tech*, https://www.futura-sciences.com/tech/definitions/technologie-systeme-embarque-15282/.

[3] *FPGA et microcontrôleur : lequel est le mieux adapté à vos besoins ?* https://pcbassemblyfrance.com/fpga-et-microcontroleur.html.

[4] *Système embarqué : définition, caractéristiques et fonctionnement*, https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1513911-systeme-embarque-definition-caracteristiques-et-fonctionnement/.

[5] *Système embarqué : définition, caractéristiques et fonctionnement*, https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1513911-systeme-embarque-definition-caracteristiques-et-fonctionnement/.

[6] *Acrhitecture d'un système embarqué*, http://igm.univ-mlv.fr/~dr/XPOSE2002/SE/architecture.html.

[7] *Développement système embarque - Anthemis Technologies*, https://www.anthemis-technologies.com/developpement-systeme-embarque.

[8] *Système embarqué - Domaines d'applications*, https://www.techno-science.net/glossaire-definition/Systeme-embarque-page-2.html.

[9] *Qu'est-ce que l'Internet of Things (IoT) ? | Oracle France*, https://www.oracle.com/fr/internet-of-things/what-is-iot/.

[10] *IoT Architecture : Considérations sur la topologie et l'informatique de périphérie | Digi International*, https://fr.digi.com/blog/post/iot-architecture-topology-and-edge-compute.

[11] *Les différentes couches d'une infrastructure IoT - Silicon*, https://www.silicon.fr/hub/hpe-intel-hub/les-differentes-couches-dune-infrastructure-iot.

[12] *Les différentes couches d'une infrastructure IoT - Silicon*, https://www.silicon.fr/hub/hpe-intel-hub/les-differentes-couches-dune-infrastructure-iot.

[13] *Apprenez l'architecture de l'Internet des objets (IoT) en 5 minutes ou moins [+ Cas d'utilisation] - Geekflare*, https://geekflare.com/fr/iot-architecture/.

[14] *Architecture IdO à 3 couches – StackLima*, https://stacklima.com/architecture-iot-a-3-couches/.

[15] *Problèmes de sécurité et de confidentialité pour une maison intelligente basée sur l'IoT | Publication de conférence IEEE | IEEE Xplorer*, https://ieeexplore.ieee.org/document/7973622.

[16] *Utilisation des technologies cloud pour les données domestiques à grande échelle dans la ville intelligente | Actes de la 4e conférence internationale IEEE 2012 sur la technologie et la science du cloud computing (CloudCom)*, https://dl.acm.org/doi/10.1109/CloudCom.2012.6427546.

[17] *Système intelligent de sécurité et de domotique basé sur l'IoT | Publication de conférence IEEE | IEEE Xplorer*, https://ieeexplore.ieee.org/document/7813916.

[18] *Les technologies de la maison intelligente en Europe : une revue critique des concepts, avantages, risques et politiques - ScienceDirect*, https://www.sciencedirect.com/science/article/pii/S1364032119308688.

[19] *Smart Home : Ce qu'il faut savoir sur la maison intelligente - Batiadvisor*, https://batiadvisor.fr/smart-home/.

[20] *Définition : qu'est-ce qu'une smart city ? | Plateforme Smart City Smart-City*, https://smart-city.cerema.fr/territoire-intelligent/definition-smart-city.

[21] *Smart Health : des services de santé "smart" | Le blog de Cellenza*, https://blog.cellenza.com/actualite-cellenza/smart-health-des-services-de-sante-smart/.

[22] *Une introduction au transport intelligent : Avantages et exemples | Digi International*, https://fr.digi.com/blog/post/introduction-to-smart-transportation-benefits.

[23] *Maison intelligente et connectée : 6 fonctionnalités à avoir chez soi - Le Reflet du Lac*, https://www.lerefletdulac.com/publireportages/maison-intelligente-et-connectee-6-fonctionnalites-a-avoir-chez-soi/.

[24] *Modélisation Orientée Objet Temps Réel - ScienceDirect*, https://www.sciencedirect.com/science/article/pii/S1474667017463464.

[25] *Real-Time Object-Oriented Modeling - ScienceDirect*, https://www.sciencedirect.com/science/article/pii/S1474667017463464.

[26] *Mémoire 2022.pdf*, https://pmb-int.univ-temouchent.edu.dz/memoire.

[27] *Editions ENI - Livres, vidéos et e-formations informatiques pour tous les niveaux*, https://www.editions-eni.fr/.

[28] *Accès gratuit à la 1ere Bibliothèque Informatique en français - Editions ENI*, https://www.editions-eni.fr/open/.

[29] *Comment programmer une carte Arduino ? - Playhooky*, https://www.playhooky.fr/technologie/arduino/.

[30] *Documentation Française de Blynk*, https://booteille.github.io/blynk-docs-fr/.