

الجمهورية الجزائرية الديمقراطية الشعبية

*République Algérienne Démocratique et Populaire*

وزارة التعليم العالي و البحث العلمي

*Ministère de l'enseignement supérieur et de la recherche scientifique*

Université 20 Août 1955- Skikda

Sciences des Faculté

Département d'Informatique



جامعة 20 أوت 1955 سكيكدة

كلية العلوم

قسم إعلام آلي

*Mémoire de fin d'études*

*En vue de l'obtention du diplôme de Master en Informatique*

*Option : Réseaux et Systèmes distribués RSD*

## *Thème*

# *Conception et simulation d'un système IoT Pour Smart Home*

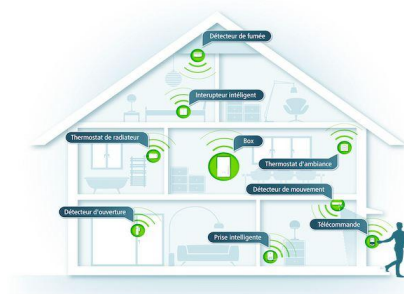
**Réalisé par :**

Nafir Maroua

Lekoui Amani

**Encadré par :**

Ms. Bourrich.H



2021/2022

## **REMERCIEMENTS**

*Nous tenons à remercier au premier lieu **ALLAH**, le tout puissant qui nous a donné la force la patience et la volonté pour terminer ce travail. Alhamdulillah.*

*Nous voudrions vraiment remercier notre encadreur*

***Ms H.Bourrich**, pour ses avis compétents, sa disponibilité, son soutien, et sa patience. Nous sommes très reconnaissantes de tous ce que vous avez faits.*

*Nous tenons à citer dans ces remerciements les membres du jury qui ont bien voulu examiner et juger notre travail.*

*A tous les enseignants de département informatique et surtout le chef du département **Mr Benoudina Lazhar**.*

*Nous remercions chaleureusement nos parents de fond de coeur pour nous avoir accompagnés, aidé, soutenu moralement et financièrement.*

*Nos remerciements vont aussi à toutes ceux et celles qu'ils ont contribué de près ou de loin à la réalisation de ce modeste travail.*

*merci*



## *Dédicaces*

*Je dédie ce modeste travail à : l'homme de ma vie, mon exemple éternel, mon soutien moral, et source de joie et de bonheur, à toi mon papa.*

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que tu as consenti pour mon instruction et mon bien être.*

*A la lumière de mes jours, la source de mes efforts, la flamme de mon coeur, ma vie et mon bonheur ; maman que j'adore. Je te remercie pour tout le soutien et l'amour que tu me portes depuis mon enfance et j'espère que ta bénédiction m'accompagne toujours.*

*A ma chère sœur Ikram et mon agréable frère Achraf, Je vous souhaite une vie pleine de bonheur et de réussite, et que Dieu te protège.*

*A mon encadreur, Tous les mots ne sauraient exprimer la gratitude, Le respect, la reconnaissance, que j'ai à l'égard du Ms Bourrich.H, Merci pour votre confiance, votre patience et votre aide.*

*A celles que j'aime beaucoup mes amies Maria, Djihane, Ryem et surtout A mon binôme, Amani. Que je te souhaite beaucoup de réussite dans sa vie.*

*A toutes les personnes qui ont participé à l'élaboration de ce travail, à tous ceux que j'ai omis de citer.*

**Nafir Maroua**





## *Dédicaces*

*Je dédie ce travail à ma famille adorée*

*Mes chers parents pour leur soutien, patience, sacrifice et amour, vous  
méritez tous éloges*

*Vous qui avez fait de moi ce que je suis maintenant*

*J'espère être comme vous l'avez souhaité*

*Que Dieu vous garde et vous bénisse*

*À l'âme de mon grand-père qui m'a laissé triste*

*Pour la santé de ma grand-mère, qui l'a toujours trouvée propice à l'idée  
de terminer mes études et son intérêt à atteindre les meilleures notes*

*A mes chères sœurs et unique frère Akram sans oublier le mari de ma  
chère sœur.*

*À mon âme sœur Maïssa et ma chère amie Maroua*

*Les deux que je trouve toujours lors de mes journées difficiles*

*À tous ceux qui m'ont aidé à arriver ici aujourd'hui*

*Enfin et surtout, mon chère encadreur, Bourrich.H, merci beaucoup pour  
vos efforts, votre patience et votre confiance*

*Lekoui Amani*



## ملخص

قام الإنسان القديم بإنشاء أكواخًا (بيوت الأشجار والجلود) لحمايته من العوامل الطبيعية والحيوانات المفترسة لتلبية احتياجاته الأمنية، ثم تطورت هذه الأكواخ وفقًا لاحتياجات الإنسان، وتوفر مواد البناء حتى ظهور البيوت الحالية. أصبحت المنازل في الوقت الحاضر مجهزة للغاية بالأجهزة الإلكترونية والكهربائية مع ظهور آلات التشغيل الآلي للمنزل، ومع الانتشار الهائل للإنترنت، مع كل هذه العوامل التكنولوجية والبنى التحتية، تطورت حاجيات الإنسان وأصبح يفكر في إنشاء منازل آليّة تمامًا توفر له بيئة أكثر راحة وأمان وبهذا ظهر مصطلح "المنازل الذكية".

ركز هذا المشروع على تصميم منزل ذكي يتم التحكم فيه تلقائيًا بواسطة جهاز كمبيوتر (أو هاتف) ولوحة اردوينو.

الهدف من هذا المشروع هو إنشاء منزل ذكي يوفر الطاقة ويوفر أعلى درجات الحماية ويحترم مبدأ الراحة ويساعد كبار السن والأشخاص ذوي الإعاقة على العيش باستقلالية قدر الإمكان وتجنب العزلة الاجتماعية وتحويل حياتهم إلى نوع من المشاركة وليس الاعتماد على المساعدة. فتطور مفهوم الأجهزة المدمجة وإنترنت الأشياء والحوسبة السحابية، جعل المنازل الذكية تكتسب ميزات جديدة مثل: البيت الذكي التفاعلي، وتميزت محتويات هذه المنازل باستقلالية القرار والذكاء والتفاعل مع البيئة من خلال مستشعرات وأجهزة إلكترونية دقيقة مدمجة.

**الكلمات الرئيسية:** المنزل الذكي، النظام المدمج، إنترنت الأشياء، الأردوينو، الاستشعار.

## Résumé

L'homme ancien a établi des huttes (maisons d'arbres et de cuir) pour se protéger des facteurs naturels et des prédateurs pour répondre à ses besoins de sécurité, puis ces huttes se sont développées en fonction des besoins humains, et de la disponibilité des matériaux de construction jusqu'à l'émergence des maisons actuelles. Les maisons d'aujourd'hui sont largement équipées d'appareils électroniques et électriques avec l'avènement des machines domotiques, et avec la diffusion massive d'Internet, avec tous ces facteurs et infrastructures technologiques, les besoins humains se sont développés et il pense à créer des maisons entièrement automatisées qui offrent lui avec un environnement plus confortable et sûr. Le couplage de ces technologies avec l'intelligence artificielle se donné naissance à un nouveau modèle de maison dite maison intelligente.

Ce projet s'est concentré sur la conception d'une maison intelligente qui pourrait être contrôlée automatiquement par un ordinateur (ou un téléphone) et une carte Arduino.

L'objectif de ce projet est de créer une maison intelligente qui économise l'énergie, offre le plus haut degré de protection, respecte le principe de confort, aide les personnes âgées et les personnes handicapées à vivre de la manière la plus autonome

possible, évite l'isolement social et transforme leur vie en une forme de participation et non de dépendance vis-à-vis de l'assistance. Le développement du concept d'appareils embarqués, de l'Internet des objets et du cloud computing, a permis aux maisons intelligentes d'acquérir de nouvelles fonctionnalités telles que : La maison intelligente interactive et le contenu de ces maisons se caractérisent par l'indépendance de la prise de décision, l'intelligence et l'interaction avec l'environnement grâce à des capteurs intégrés et des dispositifs microélectroniques.

**Mots clé :** Maison intelligente, Système embarqué, Internet des objets, Arduino, Capteur.

### *Abstract*

In antiquity, human beings created shelters of woods, stones, and mud to protect themselves from natural factors and predatory animals. Then they developed them according to their needs and available materials until they got to their current shape. Nowadays, houses became equipped with a lot of electronic devices, and the Internet has become indispensable in all homes.

These caused humans to think of creating fully automated homes to provide a safer and more comfortable lifestyle. This is where the term smart homes were introduced.

This project focused on designing a smart home that could be automatically controlled by a computer (or phone) and an Arduino board.

The objective of this project is to create a smart home that saves energy, offers the highest degree of protection, respects the principle of comfort, helps the elderly and people with disabilities to live as independently as possible, avoids social isolation and transforms their life into a form of participation and not dependency on assistance. The development of the concept of embedded devices, the Internet of Things and cloud computing, has enabled smart homes to acquire new features such as: The interactive smart home and the content of these homes are characterized by independence decision-making, intelligence and interaction with the environment through integrated sensors and microelectronic devices.

**Keywords:** Smart home, Embedded system, Internet of things, Arduino, Sensor.

## *Table des matières*

REMERCIEMENTS.....	I
Dédicaces .....	II
Résumé.....	IV
Table des matières .....	VI
Table des figures.....	X
Liste des tableaux.....	XII
Introduction générale .....	1

## *CHAPITRE 01*

### *Les systèmes embarqués*

1. Introduction.....	3
2. Historique.....	3
3. Définitions .....	4
4. Composition d'un système embarqué.....	5
5. Caractéristiques des systèmes embarqués.....	6
6. Architecture d'un système embarqué .....	6
7. Les approches de développement de systèmes embarqués.....	7
7.1. Développement classique .....	7
7.2. Le Co-design.....	7
8. Contraintes des systèmes embarqués .....	8
9. Domaines d'application des systèmes embarqués .....	9
9.1. Systèmes de chauffage central.....	9
9.2. Systèmes GPS.....	9
9.3. Dispositifs médicaux .....	10
9.4. Systèmes automobiles.....	10
9.5. Distributeurs automatiques de billets .....	10
10. Conclusion .....	11

## CHAPITRE 02

### *L'IoT pour la maison intelligente*

1.	Introduction.....	12
2.	Définition.....	12
3.	Le fonctionnement de l'IoT .....	12
4.	Architecture, éléments et protocoles IoT .....	13
4.1.	Architecture IoT à trois couches.....	13
4.2.	Architecture IoT à cinq couches .....	18
4.3.	Architecture IoT à sept couches .....	20
5.	Applications .....	23
5.1.	Les maisons intelligentes (Smart home).....	23
5.2.	Les villes intelligentes (Smart cities).....	24
5.3.	La santé (Smart health).....	24
5.4.	L'agriculture .....	25
5.5.	Le transport.....	26
6.	La maison intelligente (Smart home).....	26
6.1.	Le confort .....	27
6.2.	Les économies d'énergie .....	27
6.3.	Communication .....	28
6.4.	La sécurité.....	28
7.	Que trouve-t-on dans une maison intelligente ? .....	29
7.1.	Éclairage intelligent.....	29
7.2.	Prises électriques .....	29
7.3.	Chauffage intelligent .....	30
7.4.	Smart Lock .....	30
7.5.	Hydrao First.....	31
7.6.	Caméra intelligente.....	31
8.	Les inconvénients de maison intelligente .....	31
9.	Conclusion .....	32

## **CHAPITRE 03**

### ***Conception & Simulation***

1.	Introduction.....	33
2.	Architecture globale du système.....	33
3.	UML-RT .....	34
4.	Conception.....	37
4.1.	Diagramme de cas d'utilisation .....	37
4.2.	Diagramme de structure (Modèle/Top capsule) .....	38
4.3.	Définition des protocoles.....	38
4.4.	Définition des machines à états .....	39
5.	Simulation.....	40
5.1.	Démarche de la Simulation.....	42
5.2.	Schéma globale de la simulation .....	44
6.	Conclusion .....	46

## **CHAPITRE 04**

### ***Implémentation***

1.	Introduction.....	47
2.	Disposition du système (system layout) .....	47
3.	Matérielle utilisés.....	47
3.1.	la carte Arduino YUN.....	47
3.2.	Les capteurs .....	49
3.3.	Climatiseur (Ventilateur).....	50
3.4.	Breadboard (plaque d'essai) .....	51
3.5.	LED .....	51
3.6.	Buzzer.....	52
3.7.	Fils de connexion.....	52
3.8.	Câble USB .....	52
4.	Plateformes logicielles .....	53

4.1.	Arduino IDE .....	53
4.2.	Mosquitto (MQTT Broker).....	54
4.3.	Telegraf.....	55
4.4.	InfluxDB .....	56
4.5.	Grafana .....	56
5.	Explication de code.....	57
6.	Le projet de stage .....	62
6.1.	Le code .....	62
6.2.	Les étapes .....	64

## Table des figures

<b>Figure 1.1</b> : La convergence des disciplines. ....	5
<b>Figure 1.2</b> : Composition d'un système embarqué. ....	5
<b>Figure 1.3</b> : Architecture d'un système embarqué. ....	6
<b>Figure 1.4</b> : Développement classique. ....	7
<b>Figure 1.5</b> : Etapes du Co-desing. ....	8
<b>Figure 1.6</b> : chauffage central. ....	9
<b>Figure 1.7</b> : Systèmes GPS. ....	9
<b>Figure 1.8</b> : Les dispositifs médicaux. ....	10
<b>Figure 1.9</b> : Systèmes automobiles. ....	10
<b>Figure 1.10</b> : Distributeurs automatiques de billets. ....	11
<b>Figure 2.1</b> : Architecture IoT à trois couches. ....	13
<b>Figure 2.2</b> : Architecture du WSN. ....	14
<b>Figure 2.3</b> : Système RFID. ....	14
<b>Figure 2.4</b> : Topologies ZigBee. ....	15
<b>Figure 2.5</b> : Topologies BLE. ....	15
<b>Figure 2.6</b> : Architecture 6LoWPAN. ....	16
<b>Figure 2.7</b> : Architecture LoRaWAN. ....	17
<b>Figure 2.8</b> : Architecture CoAP. ....	17
<b>Figure 2.9</b> : Architecture MQTT. ....	18
<b>Figure 2.10</b> : Architecture IoT à cinq couches. ....	18
<b>Figure 2.11</b> : Architecture IoT à sept couches. ....	20
<b>Figure 2.12</b> : Quelques applications d'IoT. ....	23
<b>Figure 2.13</b> : Une maison intelligente (Smart Home). ....	23
<b>Figure 2.14</b> : Infographie de Smart Cities. ....	24
<b>Figure 2.15</b> : Moniteur de santé Healthpatch. ....	25
<b>Figure 2.16</b> : Agriculture intelligente (Système d'arrosage). ....	25
<b>Figure 2.17</b> : véhicule intelligent. ....	26
<b>Figure 2.18</b> : Une maison intelligente (Smart Home). ....	27
<b>Figure 2.19</b> : Éclairage intelligent. ....	29
<b>Figure 2.20</b> : Prise électrique. ....	30
<b>Figure 2.21</b> : Chauffage intelligent. ....	30
<b>Figure 2.22</b> : Verrouillage intelligent. ....	30

<b>Figure 2.23</b> : Hydrao First.....	31
<b>Figure 2.24</b> : Caméra intelligente. ....	31
<b>Figure 3.1</b> : Architecture globale du système. ....	34
<b>Figure 3.3</b> : Représentation graphique du concept de comportement d’UML-RT.....	35
<b>Figure 3.2</b> : Représentation graphique des concepts de structure d’UML-RT.....	35
<b>Figure 3.4</b> : Diagramme de cas d’utilisation du système.....	37
<b>Figure 3.5</b> : Top capsule : SmartHome.....	38
<b>Figure 3.6</b> : Machine à état du contrôleur.....	39
<b>Figure 3.7</b> : Machine à état d’une lampe d’objet contrôlée.....	39
<b>Figure 3.8</b> : Machine à état d’un capteur de mouvement.....	39
<b>Figure 3.9</b> : ISIS (PORTEUS).....	40
<b>Figure 3.10</b> : Composants Blynk.....	41
<b>Figure 3.11</b> : COMPIM.....	42
<b>Figure 3.12</b> : Paramètres COMPIM.....	42
<b>Figure 3.13</b> : Spécifiez le type de devise.....	43
<b>Figure 3.14</b> : Émulation de ports série virtuels.....	43
<b>Figure 3.15</b> : connecter un appareil à COM4 à Blynk-cloud.....	44
<b>Figure 3.16</b> : Schéma globale de la simulation.....	44
<b>Figure 3.17</b> : Model de capteur de mouvement(PIR).....	45
<b>Figure 3.18</b> : Lampe et moteur utilisée comme un climatiseur.....	45
<b>Figure 3.19</b> : Capteur DHT 22 (humidité et température).....	46
<b>Figure 4.1</b> : Disposition du système.....	47
<b>Figure 4.2</b> : Carte Arduino YUN.....	48
<b>Figure 4.3</b> : les différentes LEDs de la carte Arduino.....	48
<b>Figure 4.4</b> : les trois boutons de réinitialisation de la carte Arduino.....	48
<b>Figure 4.5</b> : Capteur de température et humidité DHT22.....	50
<b>Figure 4.6</b> : Capteur de mouvement PIR.....	50
<b>Figure 4.7</b> : Ventilateur a 12V.....	50
<b>Figure 4.8</b> : Breadboard.....	51
<b>Figure 4.9</b> : LEDs.....	51
<b>Figure 4.10</b> : Module buzzer.....	52
<b>Figure 4.11</b> : Câbles de prototypage.....	52
<b>Figure 4.12</b> : Câble USB.....	52

<b>Figure 4.13</b> : Composants de l'écran principal de l'IDE Arduino. ....	53
<b>Figure 4.14</b> : Arduino IDE. ....	54
<b>Figure 4.15</b> : Logo de Telegraf. ....	54
<b>Figure 4.16</b> : Logo de Telegraf. ....	55
<b>Figure 4.17</b> : Présentation du système de plug-in Telegraf. ....	55
<b>Figure 4.18</b> : Logo d'InfluxDB. ....	56
<b>Figure 4.19</b> : Écran d'accueil de Grafana. ....	56
<b>Figure 4.20</b> : Logo de Grafana. ....	57
<b>Figure 4.21</b> : Le modèle de circuit électronique du projet sous Proteus. ....	62

### ***Liste des tableaux***

<b>Tableau 3.1</b> : Capteurs et actionneur du système. ....	33
<b>Tableau 3.2</b> : Définitions des concepts d'UML-RT. ....	36
<b>Tableau 3.3</b> : Description textuelle de cas d'utilisation. ....	37
<b>Tableau 3.4</b> : Définition des protocoles. ....	38

## *Introduction générale*

La maison est l'endroit où les gens passent la plupart de leur temps, en particulier les personnes âgées, d'où le grand impact du logement sur la qualité de vie. Le sentiment de confort et de sécurité dans l'habitat est un élément d'une grande importance.

Ces dernières années, Ce sujet est devenu très intéressant et demeure une question importante du futur. Le développement des technologies domotiques, après les grandes évolutions technologiques dans le domaine des réseaux de communication, des systèmes embarqués et de l'intelligence artificielle, a fait du concept de maison intelligente une réalité après avoir été un fantasme ou plutôt un rêve.

La conception d'un nouveau système de contrôle de maison intelligente permettra principalement d'améliorer la vie quotidienne des personnes fragiles et plus ou moins autonomes (par exemples, les personnes âgées, les personnes ayant des difficultés de se déplacer ou des problèmes de vue, etc.).

Cette thèse est organisée en quatre chapitres dont les thèmes sont donnés ci-dessous :

➤ Le premier chapitre:

Dans ce chapitre, nous avons fait une étude approfondie des systèmes embarqués, nous avons évoqué l'histoire des systèmes embarqués avec une collection de quelques définitions des chercheurs précédents, en plus de ses composants, limitations, types, architecture, et enfin ses domaines d'application.

➤ Le deuxième chapitre :

Nous avons divisé ce chapitre en deux parties :

- La première partie, nous avons parlé de l'interconnexion des choses car c'est l'outil qui relie les systèmes embarqués et la maison intelligente, où nous avons abordé son fonctionnement, son architecture et ses domaines d'application.
- La deuxième partie, nous avons étudié la maison intelligente en termes de confort, d'économies d'énergie, de communication, ainsi que de sécurité, tout en évoquant certains appareils connectés contrôlables, tels que l'éclairage, le chauffage, les caméras intelligentes, etc.

➤ Le troisième chapitre :

Dans ce chapitre, nous avons modélisé le projet, on peut donc simplifier le projet par des diagrammes utilisant uml-rt qui basé sur le logiciel papyrus pour faciliter et simplifier le fonctionnement de la maison intelligente pour le continent.

Quant aux simulations, c'est grâce au logiciel ISIS (Proteus), en connectant plusieurs accessoires (lumineux, sensibles au mouvement, etc.) pour contrôler le travail de la maison intelligente.

➤ Le quatrième chapitre :

Dans ce chapitre, nous avons décrit les différents outils et les langages de programmation utilisés ainsi que le programme et la manière dans le système est réalisée.

# *Chapitre 01:*

## *Les systèmes embarqués*

## ***1. Introduction***

Les systèmes embarqués font désormais partie intégrante de nos vies, envahissant de nombreux domaines tels que l'apprentissage automatique et les applications d'intelligence artificielle. Le matériel et les logiciels embarqués jouent un rôle important dans l'alimentation des voitures, des appareils électroménagers, des appareils médicaux, des kiosques interactifs et d'autres équipements que nous utilisons au quotidien. Dans ce chapitre, nous présenterons les bases des systèmes embarqués avec des explications pour vous aider à voir comment cette technologie affecte tous les aspects de la vie moderne.

## ***2. Historique***

- L'un des premiers systèmes embarqués modernes reconnus a été l'Apollo Guidance Computer, développé par Charles Stark Draper au MIT Instrumentation Laboratory. Au début du projet, l'ordinateur de guidage Apollo était considéré comme l'élément le plus risqué du projet Apollo car il utilisait les circuits intégrés monolithiques nouvellement développés pour réduire la queue. L'un des premiers systèmes embarqués produits en série était l'ordinateur de guidage Autonetics D-17 pour le missile Minuteman, sorti en 1961.
- Lorsque le Minuteman II est entré en production en 1966, le D-17 a été remplacé par un nouvel ordinateur qui était le premier haut volume d'utilisation des circuits intégrés.
- En 1978, la National Engineering Manufacturer Association a publié une "norme" pour les microcontrôleurs programmables, y compris presque tous les contrôleurs informatisés, tels que les ordinateurs à carte unique, les contrôleurs numériques et les contrôleurs basés sur les événements.
- Au début des années 1980, les composants du système de mémoire, d'entrée et de sortie avaient été intégrés dans la même puce que le processeur formant un microcontrôleur [1].

### 3. Définitions

#### *Définition 1 :*

Un système embarqué est un système inclus dans un autre système, il est défini comme un système électronique et informatique autonome souvent temps réel, spécialisé dans une tâche bien précise. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires [2].

#### *Définition 2 :*

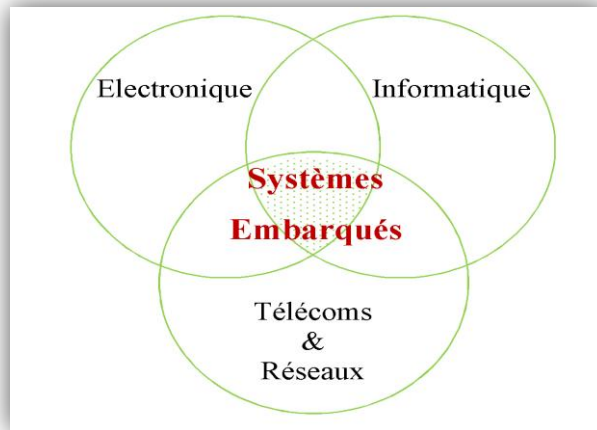
Un système embarqué est un système électronique et informatique autonome, souvent temps réel est constituée de puce électronique sur laquelle fonctionne un logiciel dédié à l'exécution de fonction spécifique , le tout étant destiné à être intégré dans des sous-ensembles, équipements, appareils et produits divers [3].

#### *Définition 3 :*

Selon Perry Xiao (2018), il s'appelle embarqué parce que le système informatique est embarqué à un périphérique matériel. Il est caractérisé par un faible cout, faible consommation d'énergie ainsi que par sa taille petite. Il est conçu pour effectuer des tâches spécifiques et souvent un système en temps réel [4].

#### *Définition 4 :*

De nombreux chercheurs s'accordent sur les propos qu'un système embarqué est un système électronique et informatique autonome, qui est dédié à une tâche bien précise. Il ne possède généralement pas des entrées/sorties standards et classiques comme un clavier ou un écran d'ordinateur. Le système matériel et l'application sont intimement liés, étant donné que le logiciel embarqué est généralement enfoui, noyé dans le matériel, le matériel et le logiciel ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur PC. Le terme système embarqué peut bien être utilisé pour désigner le matériel embarqué ou bien le logiciel embarqué, ou les deux réunis [5].

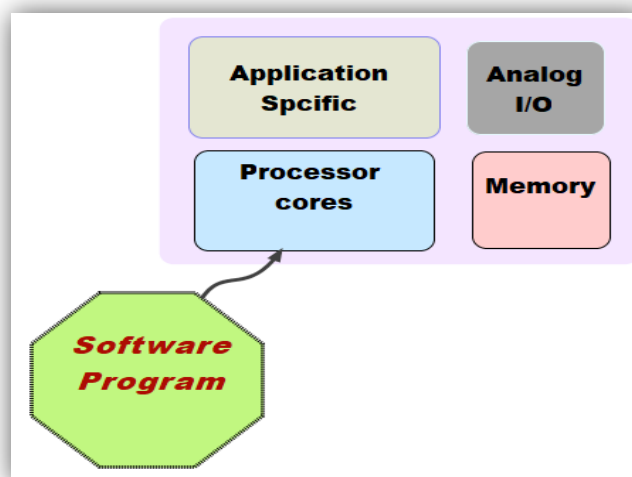


*Figure 1.1* : La convergence des disciplines.

## ***4. 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.



*Figure 1.2* : Composition d'un système embarqué.



## 7. Les approches de développement de systèmes embarqués

### 7.1. Développement classique

Les systèmes mixtes sont traités depuis longtemps et souvent d'une unique manière : dès les différents besoins identifiés, on sépare ce qui deviendra la partie logicielle et la partie matérielle du système. Cette séparation était nécessaire car les compétences à mettre en œuvre étaient différentes (informatique, électronique numérique et analogique). Ainsi une équipe "logiciel" s'occupera de la première des parties tandis qu'une équipe "matériel" concevra l'équipement électronique sur lequel, en fin du processus de production, le logiciel sera intégré.

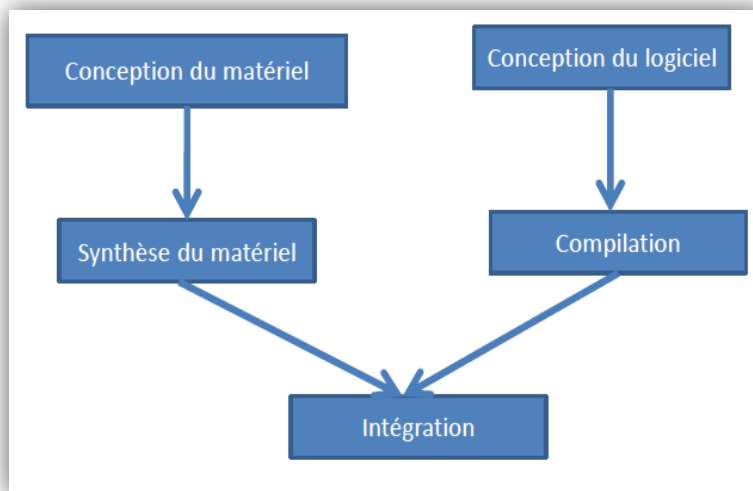


Figure 1.4 : Développement classique.

### 7.2. Le Co-design

Les méthodes Co-design proposent en fait un processus de conception unifiée des deux différentes parties que l'on appelle conception conjointe matérielle/logicielle. Ces méthodes tentent de repousser les nombreuses limites présentées par la démarche classique de conception.

Nous appelons Co-design une méthode unique de conception, couvrant l'intégralité du cycle de vie du système mixte, en unifiant la conception des parties logicielles et matérielles.

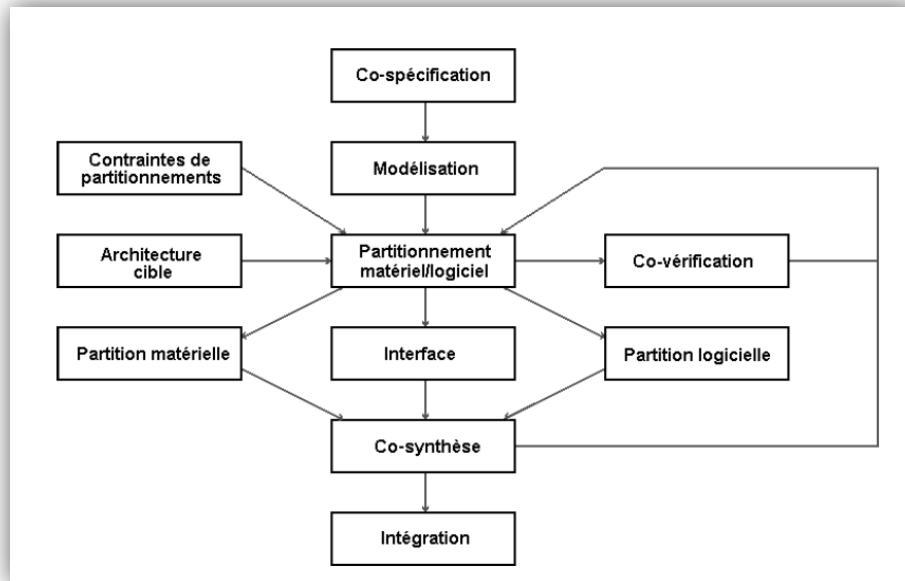


Figure 1.5 : Etapes du Co-design.

## 8. Contraintes des systèmes embarqués

- **De coût** : le prix de revient doit être le plus faible possible surtout s'il est produit en grande série.
- **D'espace compté** : ayant un espace mémoire limité de l'ordre de quelques **GO** maximum (bien que la taille vienne à être de moins limitée grâce à la miniaturisation des éléments). Il convient de concevoir des systèmes embarqués qui répondent au besoin au plus juste pour éviter un surcoût.
- **De puissance de calcul** : il convient d'avoir la puissance de calcul juste nécessaire pour répondre aux besoins et aux contraintes temporelles de la tâche prédéfinie. Ceci en vue d'éviter un surcoût de l'appareil et une consommation excédentaire d'énergie (courant électrique).
- **Une sûreté de fonctionnement** : qui demande aux systèmes critiques de fournir des résultats exacts et pertinents.
- **Une sécurité indispensable** : pour assurer la confidentialité des données utilisées, notamment pour les systèmes employés au service de la santé.
- **Temporel** : dont les temps d'exécution et l'échéance temporelle d'une tâche sont déterminés (les délais sont connus ou bornés a priori). Cette dernière contrainte fait que généralement de tels systèmes ont des propriétés temps réel.

## 9. Domaines d'application des systèmes embarqués

Les systèmes embarqués sont désormais utilisés dans des applications diverses tels que le transport, dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation...Voici quelques exemples concrets d'applications de systèmes embarqués :

### 9.1. Systèmes de chauffage central

Les systèmes de chauffage central convertissent l'énergie chimique en énergie thermique qui est ensuite transmise à de nombreux espaces du bâtiment. Si le système de chauffage central n'est pas équipé de contrôleurs de température, il peut surchauffer une pièce tout en laissant une autre pièce froide. Des commandes de thermostat pratiques vous permettent de régler la température à un niveau confortable et de réaliser d'importantes économies d'énergie.



*Figure 1.6* : chauffage central.

### 9.2. Systèmes GPS

Le GPS est un système de navigation qui utilise des satellites et des récepteurs pour synchroniser les données de localisation, d'heure et de vitesse. Le récepteur dispose d'un système intégré pour faciliter l'application du GPS.



*Figure 1.7* : Systèmes GPS.

### 9.3. *Dispositifs médicaux*

Une nouvelle catégorie de dispositifs médicaux utilise des systèmes embarqués pour aider à traiter les patients qui ont besoin d'une surveillance fréquente. Ces systèmes sont dotés de capteurs qui recueillent des données relatives à la santé des patients, qui sont envoyées à un nuage où un médecin peut consulter les données du patient sur son appareil sans fil.



*Figure 1.8* : Les dispositifs médicaux.

### 9.4. *Systèmes automobiles*

Les systèmes embarqués des véhicules sont conçus et installés pour améliorer la sécurité des véhicules. Les industries automobiles font un effort supplémentaire pour équiper les voitures de systèmes et de capteurs de technologie de pointe, afin que le conducteur puisse contrôler le régulateur de vitesse adaptatif, alerter en cas de dysfonctionnement...



*Figure 1.9* : Systèmes automobiles.

### 9.5. *Distributeurs automatiques de billets*

Un distributeur automatique de billets (DAB) est une machine informatisée utilisée dans le secteur bancaire qui communique avec un ordinateur bancaire hôte via un réseau.



*Figure 1.10* : Distributeurs automatiques de billets.

### ***10. Conclusion***

Dans ce chapitre, nous avons présenté les bases des systèmes embarqués. Ce dernier peut ou non se connecter à Internet, s'il est connecté à Internet, on l'appelle l'Internet des Objets. Donc, dans le chapitre suivant, nous présenterons la définition de l'Internet des Objets, son fonctionnement, son architecture et ces applications, en particulier la maison intelligente.

***Chapitre 02:***

***L'IoT pour la maison  
intelligente***

## **1. Introduction**

Dans les années 90, le monde a connu un grand boom dans le domaine de l'Internet, de sorte qu'Internet n'est pas seulement un lien entre les ordinateurs, mais plutôt un lien entre les objets physiques, la révolution Internet a créé un nouveau terme appelé l'Internet des objets (The Internet of things IoT), Dans ce chapitre, nous allons essayer d'aborder cette technologie en exposant les principaux points qui l'incorpore (son définition, son fonctionnement, son architecture et ses applications), et parmi ses applications nous aborderons en particulier la maison intelligente.

## **2. Définition**

L'Internet des Objets (IDO) est l'acronyme de « Internet of Things (IoT)» en anglais. Le terme IoT est apparu la première fois en 1999 dans un discours de Kevin ASHTON, un ingénieur britannique. Il servait à désigner un système où les objets physiques sont connectés à Internet ou un réseau local. Il s'agit également de systèmes capables de créer et de transmettre des données afin de créer de la valeur pour ses utilisateurs à travers divers services (agrégation, analyse, etc.)[6].

## **3. Le fonctionnement de l'IoT**

Pour assurer le bon fonctionnement de l'Internet des Objets (IoT), il existe plusieurs étapes à respecter. Perry Xiao (2018) les a résumés dans cinq étapes notamment [4]:

- Tout d'abord, chaque « objet » sur l'Internet des Objets doit avoir une identité unique. Grâce à l'évolution de l'adresse IP (Internet Protocol) et ses générations on peut fournir des milliers de différentes adresses IP au point que nous nous devrions pouvoir attribuer un identifiant unique à chaque objet physique de la planète.
- Deuxièmement, les « objets » doivent communiquer entre eux. Grâce aux nouvelles technologies sans fil qui existe et qui rendent les communications possibles, telles que le Wifi, LoRaWAN, les communications à faible champ (Bluetooth), la communication en champ proche (NFC), la RFID, ainsi que les technologies ZigBee, Z Wave.
- Troisièmement, afin d'obtenir des informations sur l'environnement, chaque « objet » doit avoir des capteurs. Il existe de nombreux capteurs notamment les capteurs de température, d'humidité, de lumière, de mouvement, de pression, infrarouge, à ultrasons, etc. Les nouveaux capteurs deviennent de plus en plus petits, économiques et durables.

- Quatrièmement, grâce au microcontrôleur que les « objets » doivent avoir, on peut gérer les capteurs, les communications ainsi que l'exécution des tâches. Il existe de nombreux microcontrôleurs pouvant être utilisés dans IoT selon le besoin.
- Enfin, afin de pouvoir exploiter la puissance de calcul, le stockage d'un serveur informatique, ainsi l'analyse et l'affichage des données, les services Cloud sont recommandés afin de pouvoir voir ce qui se passe et d'agir via des applications téléphoniques. Beaucoup de grandes entreprises y travaillent déjà, telles que Watson, IBM, la plate-forme Google Cloud, Azure, Microsoft, Oracle Cloud, etc.

### 4. Architecture, éléments et protocoles IoT

#### 4.1. Architecture IoT à trois couches

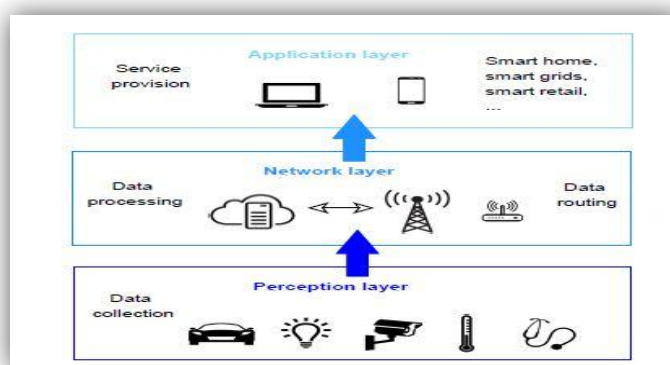


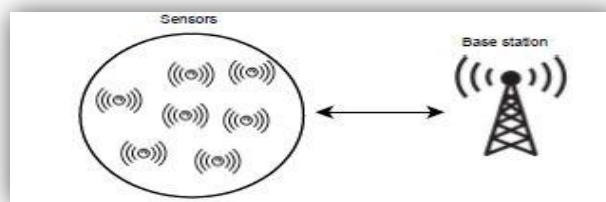
Figure 2.1 : Architecture IoT à trois couches.

##### ➤ Couche de perception

La couche de perception comprend différents appareils IoT physiques, il est responsable de l'interaction entre les appareils et de la collecte des données IoT. La collecte de données est effectuée à l'aide d'appareils intelligents tels que des étiquettes et des capteurs d'identification par radiofréquence (RFID).

##### • Capteurs sans fil

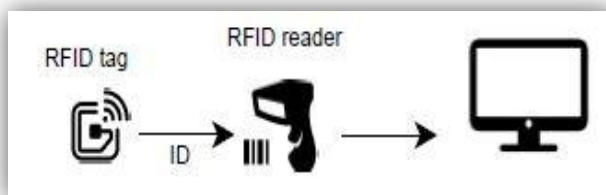
Les capteurs sans fil jouent un rôle essentiel dans l'IoT en fournissant des services de détection et de communication. Un réseau de capteurs sans fil (WSN) se compose d'un grand nombre de capteurs intelligents déployés dans des environnements distants pour détecter et collecter des données telles que la température, l'humidité, les vibrations, etc. Les données détectées sont transmises via un ou plusieurs sauts à une passerelle/station de base, comme le montre la figure 2.2.



**Figure 2.2 :** Architecture du WSN.

- **Identification par radiofréquence (RFID)**

La technologie RFID est un élément majeur de l'IoT en raison de son identification, suivi et surveillance des objets. Un système RFID se compose d'un transpondeur de signal radio (étiquette) qui stocke une identité unique d'objet et d'un lecteur d'étiquette qui identifie l'objet par des ondes radio. Le lecteur d'étiquette transfère le numéro d'identification à un ordinateur pour suivre et surveiller l'objet, comme illustré à la figure 2.3.



**Figure 2.3 :** Système RFID.

➤ **Couche réseau**

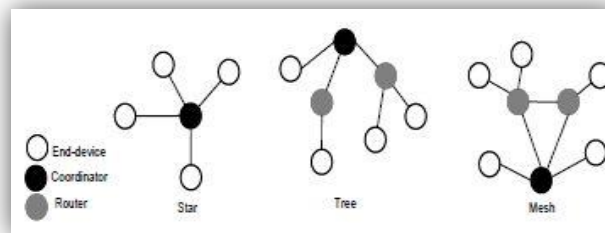
La couche réseau traite les données collectées fournies par la couche perception et stocke ou envoie les données à la couche application. Il s'agit de la couche la plus importante de l'architecture IoT car elle intègre diverses technologies de communication qui permettent la connectivité des appareils IoT. Les technologies de communication largement utilisées comprennent ZigBee, BLE, 6LoWPAN et LoRaWAN.

- **ZigBee**

ZigBee est une technologie de communication sans fil conçue pour les communications à courte portée. Il peut être utilisé dans les maisons intelligentes, les compteurs intelligents et les soins de santé intelligents. La pile de protocoles ZigBee comprend des couches physiques (PHY) et de contrôle d'accès au support (MAC) basées sur la norme IEEE 802.15.4, une couche réseau (NWK) et une couche application (APP). Un réseau ZigBee peut avoir une topologie en étoile, arborescente ou maillée et chaque réseau a un nœud coordinateur (nœud de confiance) qui gère le

## Chapitre 02 : L'IoT pour la maison intelligente

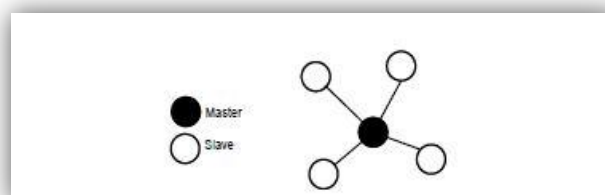
réseau et maintient la sécurité entre les appareils. Dans le réseau en étoile, les terminaux sont directement connectés au coordinateur tandis que dans les réseaux arborescents ou maillés, des routeurs intermédiaires sont utilisés pour étendre le réseau, comme le montre la figure 2.4. La couche NWK assure le routage des données à l'aide d'algorithmes de cluster-tree et d'algorithmes ad hoc modifiés à vecteur de distance à la demande (AODV). Un appareil ZigBee ne peut communiquer qu'avec un autre appareil ZigBee et, par conséquent, son interopérabilité est limitée.



*Figure 2.4* : Topologies ZigBee.

- **BLE**

Le BLE est une technologie de communication à courte portée qui réduit la consommation d'énergie par rapport au Bluetooth classique. Il est largement utilisé dans les systèmes de véhicules IoT. BLE a une pile de protocoles composée d'une couche PHY, d'une couche MAC, d'un protocole de contrôle et d'adaptation de liaison logique (L2CAP) et d'un protocole d'attribut (ATT). Le BLE adopte une topologie en étoile comprenant des dispositifs maîtres et esclaves, comme illustré à la Figure 2.5. Chaque nœud esclave est associé à un seul nœud maître. Le nœud maître est chargé d'initier la communication et de fournir une table de planification en fonction de l'accès multiple par répartition dans le temps (TDMA).



*Figure 2.5* : Topologies BLE.

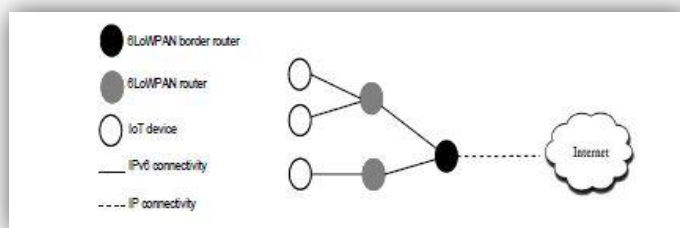
- **6LoWPAN**

6LoWPAN combine la dernière version du protocole Internet (IPv6) et le réseau personnel sans fil à faible consommation (LoWPAN). Il permet aux appareils IoT aux capacités limitées de transmettre des données via des canaux sans fil à l'aide d'IPv6. Il

## Chapitre 02 : L'IoT pour la maison intelligente

convient aux appareils à ressources limitées car il réduit les coûts de transmission, prend en charge la mobilité, etc. Les cas d'utilisation les plus courants de 6LoWPAN sont la maison intelligente, l'agriculture intelligente et l'IoT industriel. Par rapport à ZigBee, un appareil 6LoWPAN peut communiquer avec un autre appareil 6LoWPAN ou un appareil IEEE 802.15.4. Il peut également communiquer avec un réseau IP tel que le Wi-Fi, comme illustré à la Figure 2.6. La spécification de 6LoWPAN définit une pile de protocoles complète qui se compose des couches PHY et MAC basées sur la norme IEEE 802.15.4, la couche NWK, la couche transport et la couche APP.

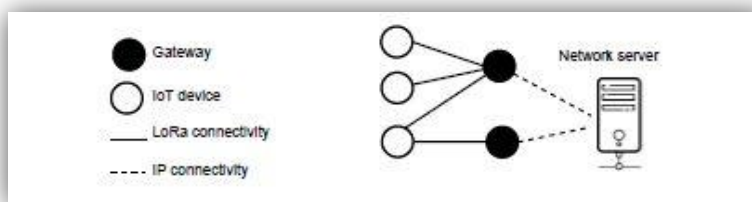
Le routage au sein du réseau 6LoWPAN utilise le protocole de routage pour les réseaux à faible puissance et avec perte (RPL). RPL prend en charge les communications point à point, point à multipoint et multipoint à point. Il est basé sur le graphe acyclique direct (DAG). À partir de DAG, RPL crée un arbre de graphe acyclique direct orienté vers la destination (DODAG) qui contient une racine du nœud feuille à la racine.



**Figure 2.6 :** Architecture 6LoWPAN.

- **LoRaWAN**

LoRaWAN est un protocole de communication longue portée conçu pour les applications IoT à faible consommation et évolutives. Comme le montre la figure 2.7, un réseau LoRaWAN se compose de terminaux, de passerelles et d'un serveur unique dans une topologie en étoile ou en étoile. Les périphériques finaux peuvent communiquer avec une ou plusieurs passerelles à l'aide du schéma ALOHA via des liaisons à un saut. Les passerelles sont connectées au serveur de réseau via le protocole Internet. Les communications sont bidirectionnelles et initiées par le terminal.



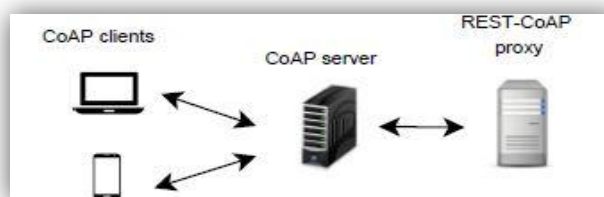
**Figure 2.7 :** Architecture LoRaWAN.

### ➤ *Couche applicative*

La couche application reçoit les données de la couche réseau et fournit les services requis aux utilisateurs IoT. Il prend en charge une grande variété d'applications telles que la maison intelligente, la vente au détail intelligente, les réseaux intelligents, etc. Les protocoles d'application les plus courants sont le protocole d'application contraint (CoAP) et le transport de télémétrie par file d'attente de messages (MQTT).

#### • *CoAP*

Étant donné que les appareils IoT sont limités en ressources, le protocole HTTP n'est pas adapté aux appareils à faible consommation en raison de sa complexité. CoAP a été conçu pour inclure des fonctionnalités HTTP dédiées aux appareils IoT. Comme le montre la Figure 2.8, CoAP est un protocole de messagerie basé sur une architecture REST (Representational State Transfer). Il fournit des fonctionnalités qui ne sont pas disponibles sur HTTP telles que la notification push (c'est-à-dire que le serveur envoie une notification à l'appareil) et la découverte de ressources (c'est-à-dire que le serveur peut stocker la liste des appareils).



**Figure 2.8 :** Architecture CoAP.

#### • *MQTT*

MQTT est un protocole de messagerie léger qui fournit la connectivité des réseaux et des utilisateurs avec des applications. Il est basé sur une architecture de publication/abonnement où le système se compose de trois composants principaux : les éditeurs, les abonnés et un courtier, comme le montre la figure 2.9. Dans le contexte de l'IoT, les éditeurs sont des appareils embarqués qui envoient des données au courtier et les abonnés sont des serveurs d'applications.

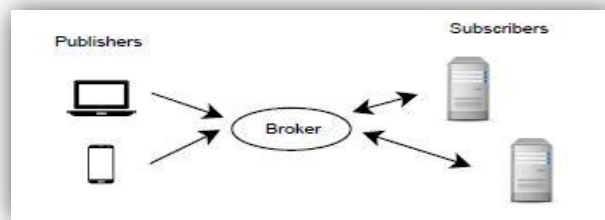


Figure 2.9 : Architecture MQTT.

### 4.2. Architecture IoT à cinq couches

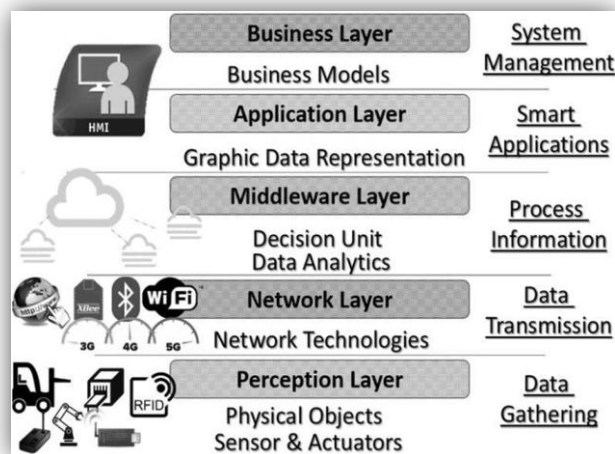


Figure 2.10 : Architecture IoT à cinq couches.

#### ➤ Technologies de la couche de perception (Perception Layer)

La couche de perception (également connue sous le nom de couche d'objets) est la première couche (à partir du bas, comme illustré à la figure 2.10) du modèle IoT. Il contient plusieurs types d'appareils physiques chargés de collecter des données et d'agir de cette manière, à savoir les températures, les emplacements, les identificateurs d'objets et les mesures d'humidité. La consommation d'énergie et la capacité de communication comme unidirectionnelle ou bidirectionnelle sont des aspects importants de cette couche [7].

#### ➤ Technologies de la couche réseau (Network Layer)

La couche réseau est la deuxième couche. Elle est considérée comme la couche infrastructure car ses technologies transforment les capteurs traditionnels référencés dans la couche perception en nœuds intelligents et connectés. Les technologies de la couche réseau permettent aux nœuds d'être reconnaissables sur Internet ou sur n'importe quel réseau local, ce qui leur permet de communiquer en toute sécurité les uns avec les autres. Les différentes technologies de cette couche se retrouvent également dans les trois premières couches de la suite IP (TCP/IP). En raison des

capacités conditionnées de la plupart des nœuds IoT, des techniques de routage évolutives et efficaces sont nécessaires pour assurer l'interopérabilité entre les appareils IoT. Faites attention au fait que de nombreuses technologies IoT sont effectivement utilisées dans les WSN ou les communications machine à machine (M2M), mais ont été améliorées pour répondre aux exigences de l'IoT. Les différents appareils modernes utilisent plus d'une technologie ; comme dans les montres intelligentes, qui ont souvent Wi-Fi, Bluetooth, Li-Fi et NFC [7].

➤ ***Technologies intergiciels (Middleware Layer)***

La couche middleware est le cœur de l'environnement IoT. Il peut être mappé à la couche application dans la suite IP (TCP/IP). Les technologies de cette couche sont fréquemment prises en charge par les plates-formes IoT. Cette couche permet d'identifier et de demander des services sur la base de noms et d'adresses et permet aux programmeurs de communiquer avec divers objets, quelle que soit la configuration matérielle spécifique. Cette couche traite également les données reçues, émet des jugements et fournit les services requis [7].

➤ ***Technologies de la couche d'application (Application Layer)***

Cette couche est chargée de fournir les services demandés aux utilisateurs IoT via une interface simple sans savoir comment les demandes de service sont traitées dans les couches sous-jacentes. Les utilisateurs de l'IoT peuvent accéder à un service (par exemple, suivre et gérer des véhicules ou lire ou régler les conditions de température à distance) en utilisant de nombreuses plates-formes (par exemple, des montres intelligentes, des ordinateurs portables et des Smartphones) via des portails ou des applications Web [7].

➤ ***Technologies de la couche métier (Business Layer)***

Dans cette couche, la couche d'application disparate, les données de service et les données environnementales IoT, à savoir les modèles commerciaux, les graphiques et les organigrammes, sont accessibles. Cet accès aide les administrateurs dans la conception, la surveillance, l'analyse, la mise en œuvre, l'évaluation et le développement des systèmes IoT, car la sortie de chacune des couches déjà mentionnées est analysée dans cette couche pour améliorer les services et protéger la confidentialité des utilisateurs [7].

### 4.3. Architecture IoT à sept couches

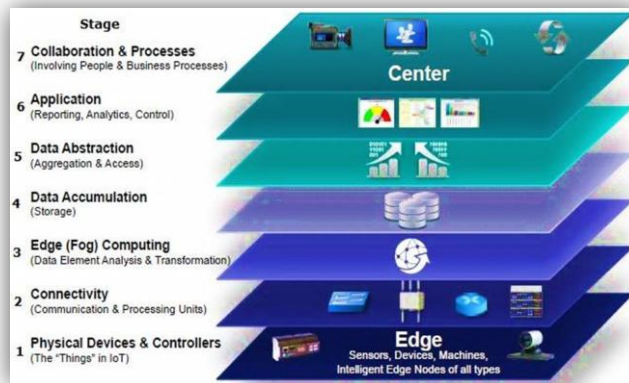


Figure 2. 11 : Architecture IoT à sept couches.

➤ **Les périphériques physiques et les contrôleurs (Physical Devices & Controllers)**

Le modèle de référence IoT commence par des appareils physiques et des contrôleurs qui peuvent contrôler divers appareils. Ce sont les «choses» de l'IoT, et elles contiennent une large gamme d'appareils terminaux qui envoient et reçoivent des informations. Les appareils sont divers et il n'y a pas de réglementation sur l'emplacement, la taille, le facteur de forme ou le débit. Moins d'appareils auront la taille d'une puce de silicium et peu seront aussi gros que des véhicules. L'IoT doit supporter toute la gamme [7].

➤ **La connectivité (The Connectivity)**

La connectivité est la fonction la plus essentielle de la prochaine étape est la transmission d'informations fiable et rapide. Le modèle de référence IoT concerne les communications et les traitements à effectuer par les réseaux existants. Le modèle de référence IoT n'a pas besoin ou n'indique pas la création d'un réseau divers, il s'appuie sur des réseaux existants. Au fur et à mesure que les dispositifs de première étape prolifèrent, la manière dont ils interagissent avec l'équipement de connectivité de deuxième étape peut changer. Sans se soucier des détails, les appareils de premier étage communiquent via le système IoT en interagissant avec les équipements de connectivité de deuxième étage [7].

➤ **L'informatique de périphérie (The Edge « Fog » Computing)**

Le traitement des données de l'étape miniature au niveau de la passerelle et des nœuds de capteur est connu sous le nom d'informatique de pointe. Les fonctions de la troisième étape sont motivées par la nécessité de convertir les flux de données du

réseau en informations appropriées pour le stockage et le traitement d'étape supérieure à la quatrième étape. Cela signifie que les activités de la troisième étape se concentrent sur l'analyse et la transformation de données à grand volume. Étant donné que les données sont généralement soumises à l'équipement de mise en réseau de deuxième étape de connectivité par des dispositifs dans des unités miniatures, le traitement de troisième étape est effectué paquet par paquet. Ce traitement est fini, compte tenu du fait qu'il n'y a que la compréhension des unités de données et non des sessions ou des transactions [7].

### ➤ *L'accumulation de données (The Data Accumulation)*

Les données envoyées sur Internet via des passerelles par les nœuds capteurs sont obtenues et stockées dans une base de données sur le Cloud et les systèmes de mise en réseau sont conçus pour déplacer les données de manière fiable. Avant la quatrième étape, les données sont déplacées via le réseau au rythme et à l'organisation déterminés par les appareils produisant les données. Le modèle événementiel est piloté. Comme défini ci-dessus, les dispositifs de premier étage ne contiennent pas eux-mêmes de capacités informatiques. Cependant, peu d'activités de calcul pourraient se produire à la deuxième étape, à savoir la traduction de protocole ou l'application de la politique de sécurité du réseau. Des tâches de calcul supplémentaires peuvent être effectuées lors de la troisième étape, à savoir l'inspection des paquets. La conduite de tâches de calcul aussi près que possible de la périphérie de l'IoT, avec des systèmes divers, répartis sur divers domaines de gestion représente un exemple de fog computing. Dans la quatrième étape, convertit les données basées sur les événements en traitement basé sur les requêtes. Il s'agit d'une étape décisive pour combler les différences entre le monde des applications en temps différé et le monde des réseaux en temps réel [7].

### ➤ *L'abstraction des données (The Data Abstraction)*

L'objectif principal est d'extraire les données obligatoires et significatives de toutes les données collectées. Les systèmes IoT devront évoluer au niveau de l'entreprise ou même au niveau mondial et auront besoin de plusieurs systèmes de stockage pour accueillir les données des appareils IoT et les données des systèmes HRMS, CRM, ERP et autres d'entreprise conventionnels. Les fonctions d'abstraction de données de la cinquième étape sont axées sur le rendu des données et leur stockage de manière à permettre le développement d'applications simples et performantes [7].

### ➤ *L'application (The Application)*

L'analyse des données et la réponse en fonction des données pour contrôler les actionneurs au niveau des nœuds de capteurs est l'une des principales applications de l'architecture IoT. La sixième étape est le niveau de l'application, où l'interprétation de l'information se produit. Le logiciel à ce niveau interagit avec la cinquième étape et les données au repos, il n'a donc pas à fonctionner à la vitesse du réseau. Le modèle de référence IoT ne définit pas de manière rigide une application. Les applications varient en fonction des marchés verticaux, de la nature des données de l'appareil et des besoins de l'entreprise. Peu d'applications se concentreront sur la surveillance des données de l'appareil. Rares sont ceux qui se concentrent sur le contrôle des appareils et peu combinent les données des appareils et des autres appareils. Les applications de surveillance et de contrôle délimitent de nombreux modèles d'application, serveurs d'applications, hyperviseurs, multi-threading, multi-tenant, modèles de programmation et piles logicielles, conduisant à des discussions sur les systèmes d'exploitation, la mobilité, etc [7].

### ➤ *La collaboration et les processus (The Collaboration and Processes)*

Il s'agit des sept étapes du modèle de référence IoT. L'interaction et l'implication humaines dans le scénario IoT sont considérées comme les parties les plus négligées. Non seulement l'appareil doit être suffisamment intelligent pour effectuer certaines tâches, mais il doit également avoir des interactions intuitives avec l'humain. L'une des principales différences entre l'Internet des objets (IoT) et l'IoT est que l'IoT contient des personnes et des processus. Cette dissemblance devient particulièrement nette à sept étapes. Le système IoT et les informations qu'il crée n'ont que peu de valeur s'ils ne produisent pas d'action, ce qui nécessite souvent des processus et des personnes. Applications pour exécuter le logique métier pour responsabiliser les gens.

Les gens utilisent les applications et les données associées pour leur besoin spécifique. Souvent, plusieurs personnes utilisent la même application à des fins diverses. L'objectif n'est donc pas l'application, c'est de donner aux gens les moyens de faire leur travail de manière supérieure. Les gens doivent pouvoir communiquer et collaborer, parfois en utilisant l'Internet conventionnel, pour rendre l'IoT avantageux [7].

### 5. Applications

Les applications potentielles de l'Internet des Objets sont nombreuses et variées, pénétrant pratiquement tous les domaines de la vie quotidienne des individus, des entreprises et de la société dans son ensemble, tels que :

- La maison intelligente (Smart home)
- La santé (Smart health)
- Ville intelligente (Smart city)
- L'agriculture
- Le Transport

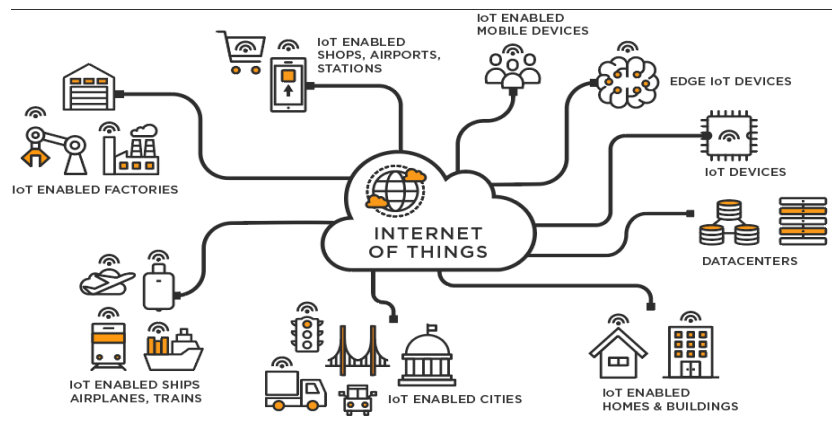


Figure 2.12 : Quelques applications d'IoT.

#### 5.1. Les maisons intelligentes (Smart home)

Une maison intelligente est une maison qui est contrôlée et gérée par un ensemble de boutons et de télécommandes, ou d'applications qui utilisent des protocoles Internet sur des réseaux locaux sans fil. Nous expliquerons ce domaine en détail par la suite.



Figure 2.13 : Une maison intelligente (Smart Home).



le taux respiratoire, la température de la peau, la posture du corps, la détection des chutes et les lectures d'activité à distance. Cela peut alerter les médecins sur les problèmes de santé potentiels avant qu'ils ne surviennent, même lorsque leurs patients ne sont pas au bureau [9].

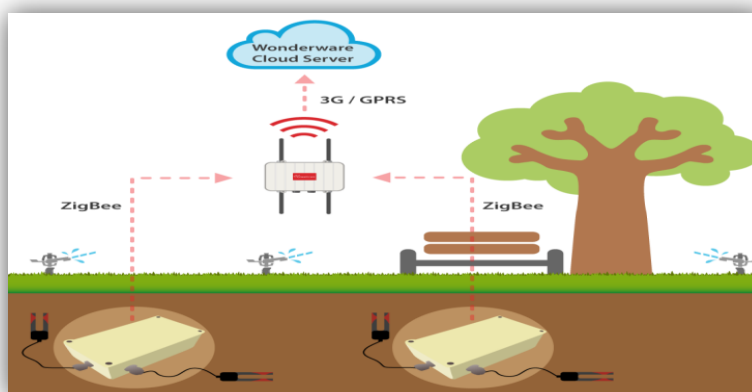


*Figure 2.15 : Moniteur de santé Healthpatch.*

### 5.4. L'agriculture

L'agriculture intelligente vise à renforcer la capacité des systèmes agricoles, à contribuer à la sécurité alimentaire.

Cet objectif a finalement été atteint grâce à l'utilisation de nouvelles technologies, comme les systèmes de positionnement par satellite comme le GPS, et aussi par l'utilisation de capteurs qui se chargeront de recueillir des informations utiles sur l'état du sol, le taux d'humidité, le taux de sel minéral, etc. Et transmettre ces informations à l'agriculteur afin qu'il prenne les mesures nécessaires pour assurer une bonne production [10].



*Figure 2.16 : Agriculture intelligente (Système d'arrosage).*

### 5.5. *Le transport*

Depuis la création de l'IoT en 1999, le nombre des véhicules intelligents sont en croissance, presque Tous les véhicules vendus aujourd'hui dans le monde renferment déjà des capteurs et de moyens de communication pour traiter la congestion du trafic, la sécurité, la pollution et le transport efficace des marchandises, etc. L'objectif est qu'une voiture soit capable de communiquer de façon autonome avec d'autres véhicules ou une centrale de surveillance pour prévenir les accidents et réduire les coûts d'assurance.

Des applications Smartphone (comme Waze) sont déjà très répandues dans le monde pour avertir les usagers de l'application en temps réel sur la présence d'un radar de vitesse mobile, d'accidents ou de ralentissement sur les autoroutes voire de proposer un itinéraire plus rapide à l'aide de Google Maps. Certaines voitures sont également équipées de la fonction appel SOS, si la voiture subit un accident, elle appelle automatiquement les secours, fournit sa localisation et la possibilité de communiquer avec les usagers. Les constructeurs automobiles travaillent aussi sur des projets de véhicules autonomes (sans conducteur) capables de se déplacer d'un point A à un point B sans aucune intervention humaine [10].



*Figure 2.17* : véhicule intelligente.

## 6. *La maison intelligente (Smart home)*

« Smart Home » ou une maison intelligente ou encore la domotique est un ensemble de techniques permettant de contrôler les appareils électriques utilisés dans les domiciles (éclairage, chauffage, audio-visuel, électroménager, etc.), d'automatiser et de programmer l'habitat. C'est un nouveau concept récent de plus en plus utilisé, il est issu de l'application à l'habitat de la programmation informatique, en profitant des technologies d'informatiques, des télécommunications, l'électronique. Toutes ces techniques et systèmes doivent être interopérables pour une gestion optimale.

## Chapitre 02 : L'IoT pour la maison intelligente

La maison intelligente a pour rôle d'assurer des fonctions de confort, d'économies d'énergie, de communication et sécurité.



*Figure 2.18* : Une maison intelligente (Smart Home).

### 6.1. Le confort

Bien sûr, le fait d'automatiser sa maison a un véritable apport sur le confort qu'on y trouve. Aujourd'hui, une maison intelligente est capable de savoir quand vous rentrez à la maison (grâce à votre Smartphone par exemple), et donc d'ouvrir le portail avant même que vous n'arriviez. Les volets peuvent s'ouvrir et se fermer au rythme du soleil, et peuvent même aller jusqu'à s'adapter à la saison et la température pour laisser entrer la lumière et la chaleur du soleil l'hiver, ou au contraire conserver le frais l'été en fermant les volets des fenêtres exposées au soleil. De la même façon, votre maison sait quand vous êtes présent, et peut ainsi adapter elle-même votre chauffage pour que la maison soit toujours à la température idéale pour vous.

Il est même possible de diffuser automatiquement votre Play List musicale préférée à votre réveil, ou quand vous rentrez à la maison. Pendant ce temps, un robot peut passer l'aspirateur dans la maison à votre place, et le système d'arrosage automatique arrosera votre jardin, tout en tenant compte des prévisions météo des prochains jours, histoire de ne pas arroser inutilement.

### 6.2. Les économies d'énergie

En gérant les volets selon la saison, ainsi que le chauffage, le système domotique vous permet d'économiser de l'énergie, et donc de l'argent, même si au départ on ne recherchait que le confort en plus. La consommation d'énergie peut être suivie très

finement, qu'il s'agisse de votre consommation d'électricité, d'eau, ou même de gaz.

Vous partez de la maison ? Le simple fait d'activer l'alarme en partant va passer le chauffage en mode éco, et éteindre toutes les lampes et les appareils restés en veille, réduisant ainsi votre consommation d'énergie en votre absence. Et ceci sans aucune action de votre part. C'est cela la maison intelligente !

### **6.3. Communication**

Un système domotique permet la communication non seulement à l'intérieur de la maison, mais aussi à l'extérieur. La technologie Internet interviendra de plus en plus pour la commande à distance par La certains utilisateurs. Vous ne devez même pas être à la maison pour commander vos appareils. Un simple coup de fil ou un sms vous permettra par exemple de régler le chauffage à distance, d'activer une simulation de présence ou de lancer le lave-vaisselle ou le lave-linge. Pratique, non?

### **6.4. La sécurité**

Les automatismes que nous avons vus plus haut peuvent tout à fait contribuer à la sécurité de vos biens, en réalisant ce qu'on appelle une simulation de présence : même en votre absence, les volets continuent de s'ouvrir, de la musique peut être diffusée dans la maison, et des lumières allumées aléatoirement. Ainsi, de l'extérieur, il devient très difficile de savoir si la maison est inoccupée, ce qui dissuade de nombreux cambrioleurs.

Une détection de fuite d'eau peut couper automatiquement l'arrivée d'eau afin d'éviter de gros dégâts.

Mais la sécurité, c'est également la sécurité des personnes : en cas de détection d'incendie, par exemple, il est tout à fait possible d'ouvrir automatiquement les volets, déverrouiller les portes, et éclairer le chemin de la sortie pour faciliter l'évacuation. Exemple d'une mise en scène d'un « simple » détecteur de fumée, couplé à un système domotique.

Dans le cas de personnes dépendantes, telles que des seniors ou des personnes à mobilité réduite, il est également possible par exemple de prévenir un proche ou les services de secours en cas de chute ou tout évènement anormal.

### 7. *Que trouve-t-on dans une maison intelligente ?*

Ces dernières années, le concept de "maison intelligente" a connu une montée en puissance. Le terme « domotique » est désormais entré dans le langage de la vie quotidienne, et on l'entend partout ! C'est un ensemble d'appareils connectés qui peuvent contrôler les différents éléments de l'espace de vie tels que l'éclairage, le chauffage, caméras intelligentes et vidéosurveillance, etc.

#### 7.1. *Éclairage intelligent*

En vous procurant un système d'éclairage intelligent, vous obtenez le contrôle sur vos lumières intérieures et extérieures à partir de votre téléphone intelligent. Ces ampoules intelligentes écoénergétiques vous permettent de contrôler l'ambiance de chacune des pièces de votre maison en fonction du moment de la journée, sans vous soucier de votre consommation d'électricité. Via l'application mobile, vous aurez une variété d'options d'optimisation d'éclairage: programmation selon un horaire régulier, tamisation après une certaine heure, éteindre les lumières en plein jour, allumer les lumières à votre arrivée, changer la couleur des lumières en soirée, etc.



*Figure 2.19 : Éclairage intelligent.*

#### 7.2. *Prises électriques*

Des prises électriques intelligentes sont également disponibles et compatibles avec les systèmes domotiques, pour vous permettre de contrôler les appareils électriques qui y sont branchés. Vous devez connecter les prises électriques à votre Wifi via l'application mobile associée, puis voir directement sur votre téléphone portable le statut énergétique de vos appareils à la maison. Vous pouvez également configurer un horaire pour chacune des entrées de la prise, pour limiter votre consommation d'électricité dans des moments stratégiques.



*Figure 2.20* : Prise électrique.

### 7.3. *Chauffage intelligent*

Les thermostats intelligents adaptent votre système de chauffage selon la température extérieure, le moment de la journée, le type de chauffage et le type de logement, pour vous offrir un confort optimal, tout en économisant de l'argent. Il vous suffit de connecter votre thermostat intelligent à votre Wi-Fi pour obtenir le contrôle total sur la température de votre maison, en configurant seulement quelques paramètres.



*Figure 2.21* : Chauffage intelligent.

### 7.4. *Smart Lock*

Avec cette serrure intelligente, vous n'aurez jamais besoin de clés, il se déverrouille automatiquement lorsque vous arrivez à la maison et se verrouille derrière vous lorsque vous fermez la porte. Un clavier optionnel signifie que vous pouvez définir un code pour ouvrir votre porte au cas où vous ne possédez pas de votre téléphone avec vous [9].



*Figure 2.22* : Verrouillage intelligent.

### 7.5. *Hydrao First*

Un pommeau de douche connecté dont la vocation est de vous faire réaliser des économies d'eau. Grâce à un jeu de LED colorées, vous pourrez adapter votre consommation d'eau lors de la douche, l'application connectée vous permettra de savoir combien d'économies ont été effectuées [9].



*Figure 2.23* : Hydrao First.

### 7.6. *Caméra intelligente*

Les caméras intelligentes permettent de surveiller votre appartement ou votre maison à distance via un Smartphone. Si vous n'avez pas le temps de tout contrôler en direct, les caméras intelligentes stockent les enregistrements vidéo, qui sont ensuite disponibles sur la carte mémoire, le Cloud ou le téléphone mobile si nécessaire.



*Figure 2.24* : Caméra intelligente.

## 8. *Les inconvénients de maison intelligente*

- ***Prix élevé***

Certains systèmes peuvent coûter relativement cher. Les options sont infinies, mais toutes ne sont pas aussi abordables.

- ***Décentralisation***

Les marques sont très nombreuses. Ce qui est un désavantage. Car toutes ces applications ne communiquent pas toujours parfaitement entre elles. Vous devrez donc gérer souvent différentes applications pour contrôler toute votre habitation.

- ***Chronophage***

Il faut du temps pour apprendre à tirer un maximum de profit de chaque application domotique.

- ***Dépendance d'internet***

L'exigence de base pour le système de maison intelligente est l'Internet, sans une bonne et forte connexion Internet, vous ne serez pas capable de prendre le contrôle.

### ***9. Conclusion***

Dans ce chapitre, nous avons introduit le concept d'Internet des Objets en présentant les principaux points qu'il comprend, ainsi que le concept de maisons intelligentes, ses objectifs, ses avantages et ses inconvénients, et à la fin nous avons mentionné quelques exemples. Le chapitre suivant sera consacré à la conception et à la simulation de notre système.

***Chapitre 03 :***

***Conception & Simulation***

### 1. Introduction

Après avoir expliqué le concept et le but des maisons intelligentes, ainsi que les exemples de ce projet dans le chapitre précédent. Nous pouvons maintenant passer à la phase conception et modélisation du système qui représente une activité clé dans le processus de développement de notre projet de fin d'études. En effet, elle formalise et détaille ce qui a été ébauché au cours de préliminaire, et permet de dégager l'étude fonctionnelle du système. Elle permet ainsi d'obtenir une idée sur ce que va réaliser le système, ainsi que la simulation qui permet de simuler les schémas avant de passer à la réalisation le système.

### 2. Architecture globale du système

La figure 3.1 représente l'architecture globale du système, cette dernière montre les différents composants du système dont le noyau et le **contrôleur**, ainsi que les liens entre ces composants, qui sont en réalité des liens de communication via lesquelles on échange généralement deux types d'informations :

- Les données (*Data*) récoltés par les capteurs et envoyés au contrôleur (*Controller*).
- Les réactions du système aux informations réceptionnées qui représente l'état du système, ces réactions sont matérialisées par un ensemble d'actions à exécuter envoyés par le contrôleur aux actionneurs.

Le tableau 3.1 recense les capteurs et les actionneurs du système :

<i>Capteurs</i>	<i>Actionneurs</i>
Capteur de température et d'humidité	Climatiseur
Capteur de mouvement	Lampe
	Sirène

**Tableau 3.1** : Capteurs et actionneur du système.

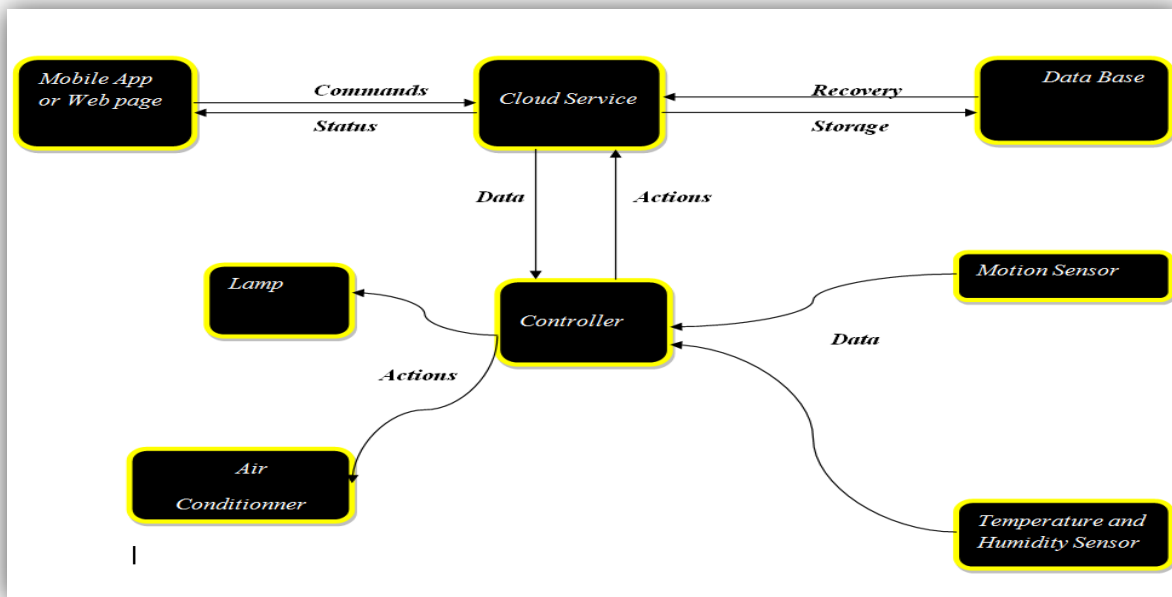
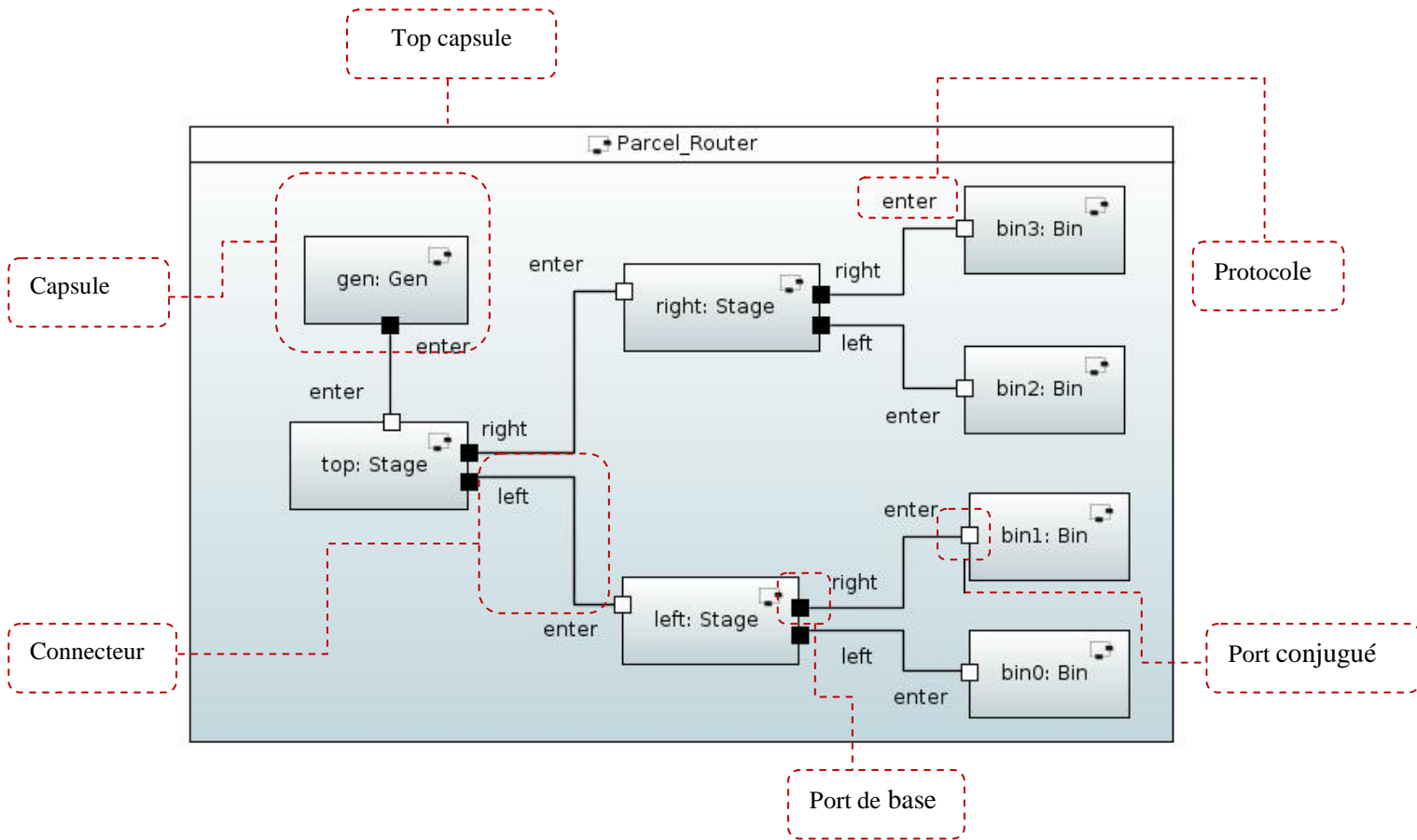


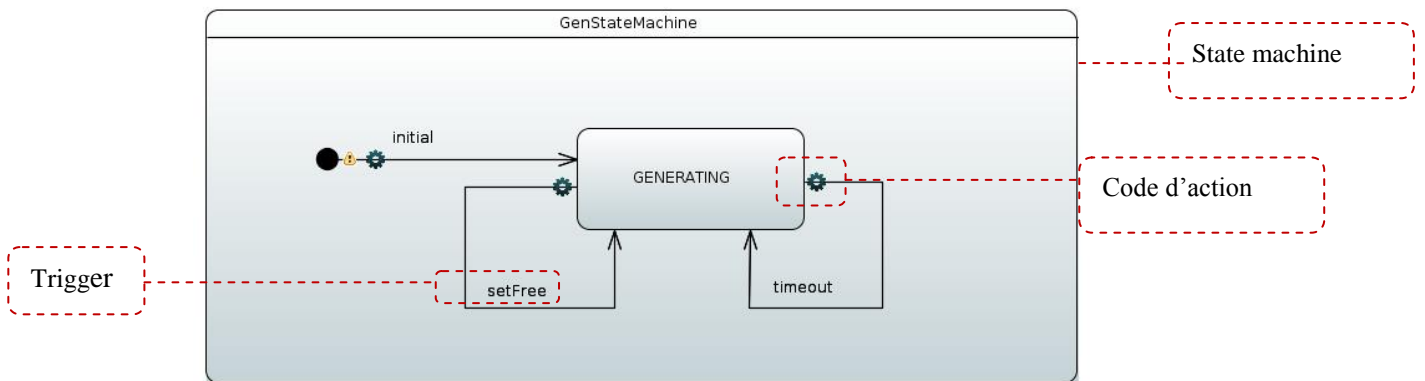
Figure 3.1 : Architecture globale du système.

### 3. UML-RT

L'UML-RT (UML-Real Time) [11] 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. Les concepts UML-RT sont hérités de ceux définis dans le langage de Modélisation Orientée Objet Temps Réel (ROOM) [12] 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. 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 l'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.

## Chapitre 03 : Conception & Simulation

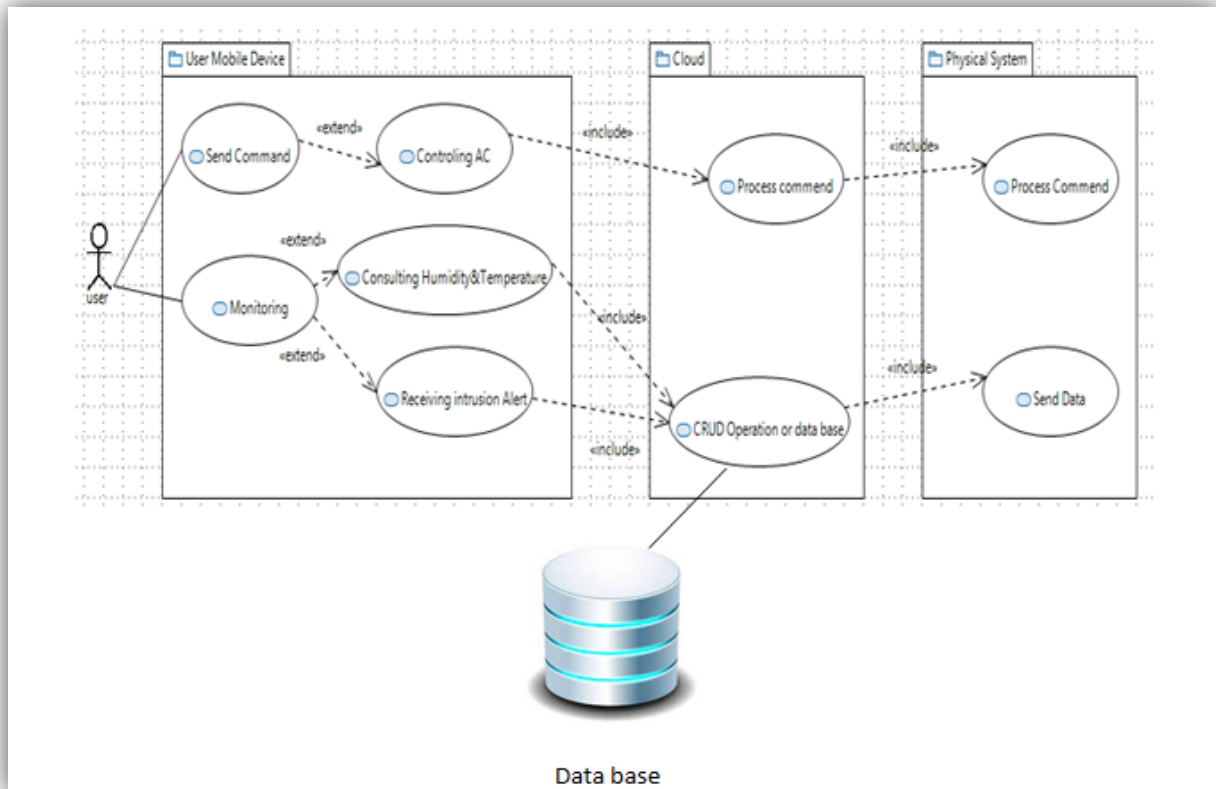
<i>Concept</i>	<i>Définition</i>
<i>Modèle</i>	<ul style="list-style-type: none"> <li>• Collection de définitions de capsules ;</li> </ul> <p>Capsule "Top" contenant la collection de capsule instances (parties).</p>
<i>Capsule</i>	<ul style="list-style-type: none"> <li>• Peut contenir des attributs, ports, ou autres instances de capsules (parts).</li> </ul> <p>Son comportement est défini par une machine à états.</p>
<i>Port</i>	<ul style="list-style-type: none"> <li>• « Boundary objects » possédé par une capsule ;</li> <li>• Typé par un protocole ;</li> <li>• Possède une opération « send » :               <ul style="list-style-type: none"> <li>- portName.msg(arg1,...,argn).send()</li> </ul> </li> <li>• Peut être de base : direction des messages comme déclarée dans le protocole.</li> <li>• Conjugué : direction des messages comme déclarée dans le protocole est renversée.</li> </ul>
<i>Machine à état</i>	<ul style="list-style-type: none"> <li>• Transition : déclenché par les messages entrants ;</li> </ul> <p>Code d'action : peut contenir des instructions d'envoi qui envoient des messages sur certains ports.</p>
<i>Protocole</i>	<ul style="list-style-type: none"> <li>• Fourni des types pour les ports ;</li> <li>• Défini :               <ul style="list-style-type: none"> <li>- Les messages d'entrées représentant les services fournis par la capsule possédant le port ;</li> <li>- Les messages de sortie représentant les services requis par la capsule possédant le port ;</li> <li>- Les messages entrants /sortant</li> <li>- un message peu transmettre des données ;</li> </ul> </li> </ul>
<i>Connecteur</i>	<ul style="list-style-type: none"> <li>• Connecte deux ports ;</li> <li>• Les ports doivent être compatibles : les deux sont des instances du même protocole.               <ul style="list-style-type: none"> <li>- Soit asymétriques l'un de base et l'autre conjugué ;</li> <li>- Soit symétriques dans le cas de messages entrants /sortants.</li> </ul> </li> </ul>

**Tableau 3.2 :** Définitions des concepts d'UML-RT.

## 4. Conception

### 4.1. Diagramme de cas d'utilisation

La figure 3.3 représente le diagramme de cas d'utilisation décrivant les fonctionnalités du système :



**Figure 3.4 :** Diagramme de cas d'utilisation du système.

- **Définition des acteurs :**

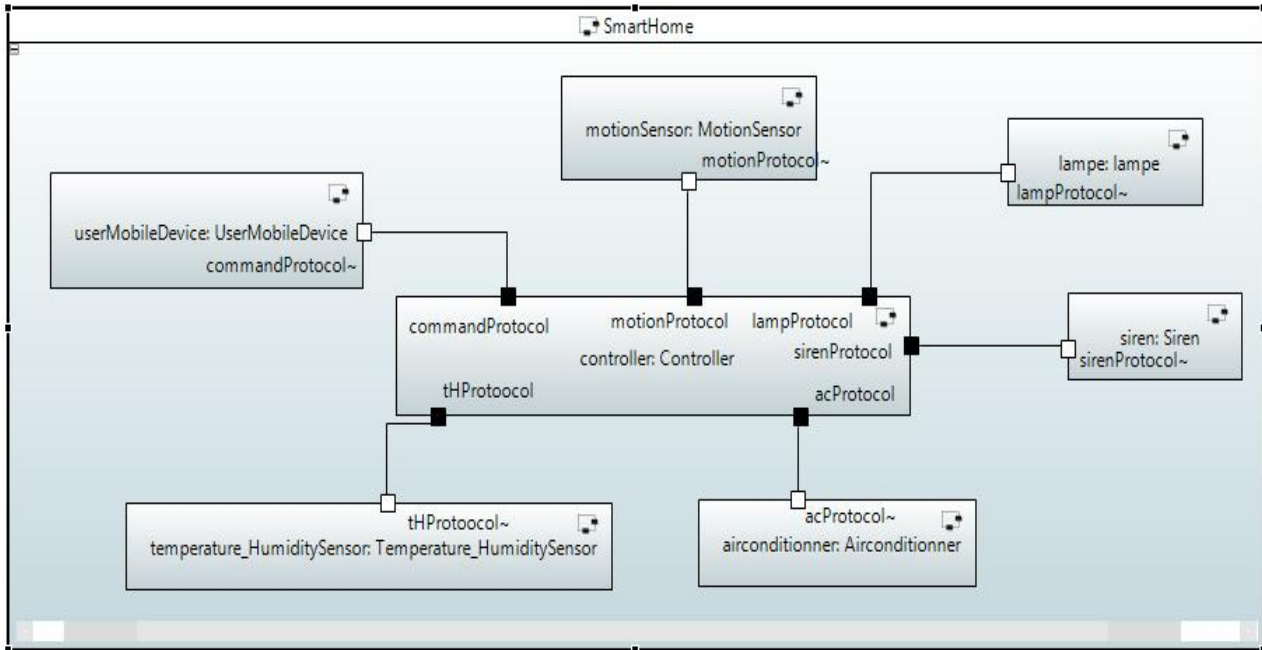
Notre système comporte un seul acteur qui est l'utilisateur du système qui interagit avec le système via une application mobile ou une page web.

- **Définition des cas d'utilisation:**

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

**Tableau 3.3 :** Description textuelle de cas d'utilisation.

## 4.2. Diagramme de structure (Modèle/Top capsule)



**Figure 3.5** : Top capsule : SmartHome.

## 4.3. Définition des protocoles

Protocole	Messages	
	Entrants	Sortants
<b>AcProtocol</b>	AcOn () AcOff ()	StarteAc () StopAc ()
<b>CommandProtocol</b>	ShowTemp () ShowHumid () ShowAc () ShowLamp () SetAc (Status : OnOff)	TempNot (Temperature : String) HumidNot (Humidity : String) AcState () LampState () IntrusionNot (IntrusionNotification : String)
<b>LampProtocol</b>	LampOn () LampOff ()	TurnOn () TurnOff ()
<b>MotionProtocol</b>	MotionState (Motion : EBoolean)	MotionState ()
<b>SirenProtocol</b>	Active () Inactive ()	ActivateSiren () DesactivateSiren ()
<b>THProtocol</b>	TempState (Temperature : Real) HumidState (Humidity : Real)	TempState () HumidState ()

**Tableau 3.4** : Définition des protocoles.

## 4.4. Définition des machines à états

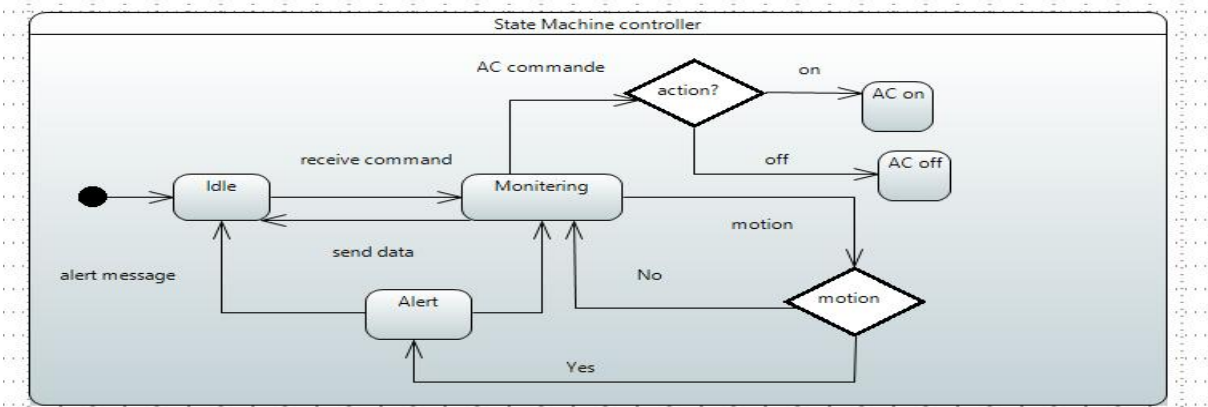


Figure 3.6 : Machine à état du contrôleur.

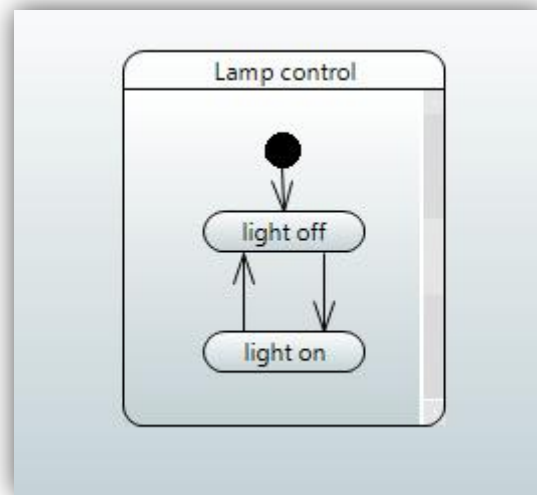


Figure 3.7 : Machine à état d'une lampe d'objet contrôlée.

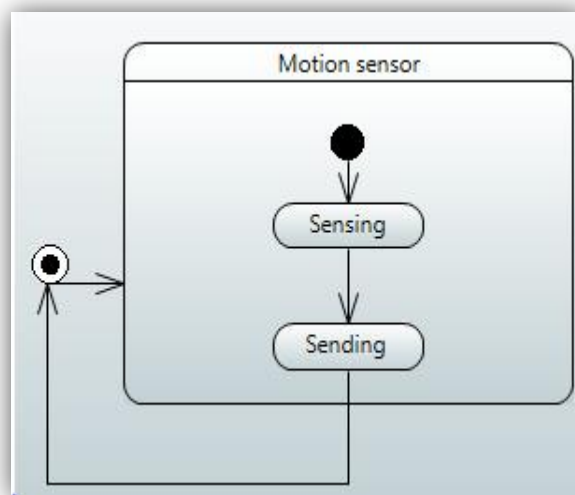


Figure 3.8 : Machine à état d'un capteur de mouvement.

### 5. Simulation

Avant de passer à la réalisation de notre système, on va procéder à la simulation du système qui sert en premier à la vérification de la partie matérielle du système, ainsi que son intégration avec la partie logicielle. Pour réaliser cette tâche on va exploiter les logiciels suivants :

#### ✓ Simulation avec ISIS (Proteus)

PROTEUS est une suite logicielle permettant la conception assistée par ordinateur électronique éditée par la société Labcenter Electronics. Il est composé de deux logiciels principaux: ISIS, permettant entre autres la création de schémas et la simulation électrique, et ARES, dédié à la création de circuits imprimés.

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 détecter certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits [13].

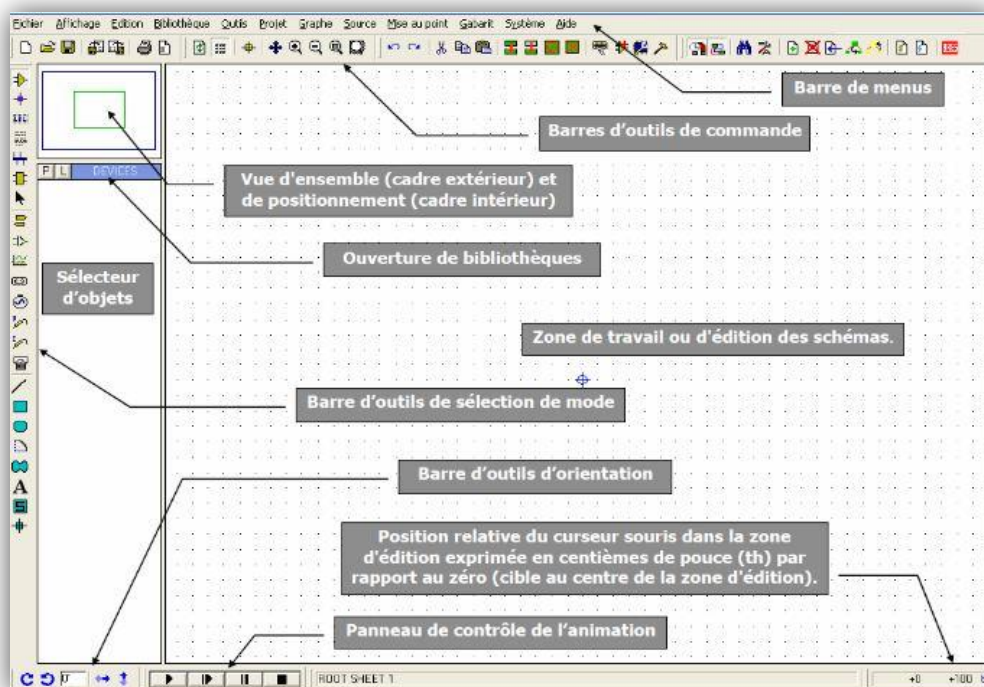


Figure 3.9 : ISIS (PORTEUS).

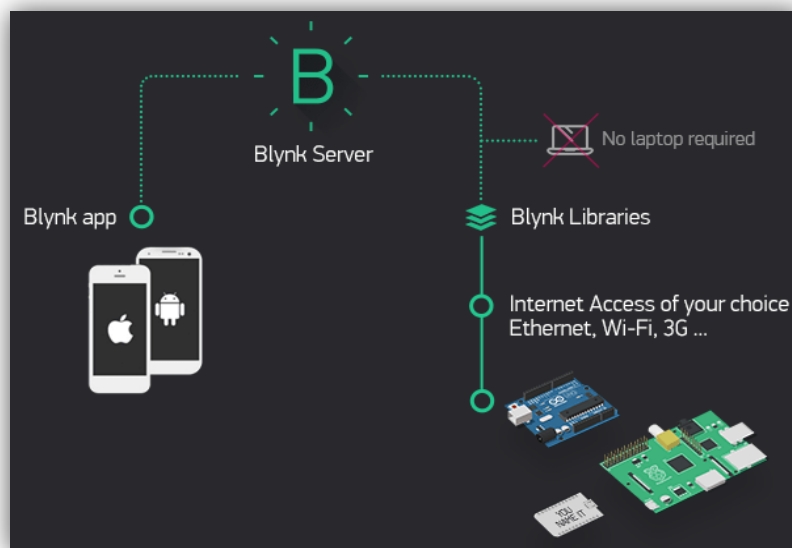
### ✓ *Blynk*

Blynk est une plate-forme IoT pour Smartphones iOS ou Android utilisée pour contrôler Arduino et GSM/GPRS via Internet. Cette application permet de créer une interface graphique ou interface homme-machine (IHM) en compilant et en fournissant l'adresse appropriée sur les widgets disponibles.

Blynk a été conçu pour l'IoT. Il peut contrôler le matériel à distance, il peut afficher les données des capteurs, il peut stocker des données, les visualiser et faire bien d'autres choses. Il y a trois composants principaux dans la plate-forme [14]:

- **Application Blynk** : Elle vous permet de créer des interfaces pour les applications à l'aide de divers widgets fournis.
- **Blynk Server** : Il est responsable de toutes les communications entre le Smartphone et le matériel. Vous pouvez ensuite utiliser le Blynk Cloud ou exécuter un serveur Blynk privé localement. C'est open source, pourrait facilement gérer des milliers d'appareils.
- **Bibliothèques Blynk** : Il permet la communication, pour toutes les plates-formes matérielles populaires, avec le serveur et traite toutes les commandes entrantes et sortantes.

Le processus qui se produit lorsque quelqu'un appuie sur le bouton dans l'application Blynk est que les données se déplacent vers Blynk Cloud, où les données trouvent par magie leur chemin vers le matériel qui a été installé.



**Figure 3.10** : Composants Blynk.

### 5.1. Démarche de la Simulation

#### ➤ Etape 01 : Connecter Proteus à l'ordinateur

Le protocole de communication UART (Universal asynchronous receiver-transmitter) a été suivi pour connecter la simulation Proteus à l'application Blynk.

L'Arduino UNO reçoit les données transmises par le capteur. Après traitement des données, l'Arduino UNO les envoie aux applications Blynk, ainsi, il est indispensable d'établir une connexion entre le circuit Proteus et le port série de l'ordinateur à l'aide du COMPIM, qui est présenté sur la figure 3.11

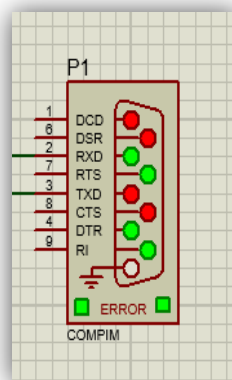


Figure 3.11 : COMPIM.

Le COMPIM est le composant utile pour connecter le port physique de l'ordinateur avec le port virtuel. Ce composant doit fixer les paramètres pour être compatible avec le port physique, comme illustré à la figure 3.12. Le port COM3 a été sélectionné avec le Baud Rate 9600, donc le port virtuel à besoin du même Baud Rate sélectionné.

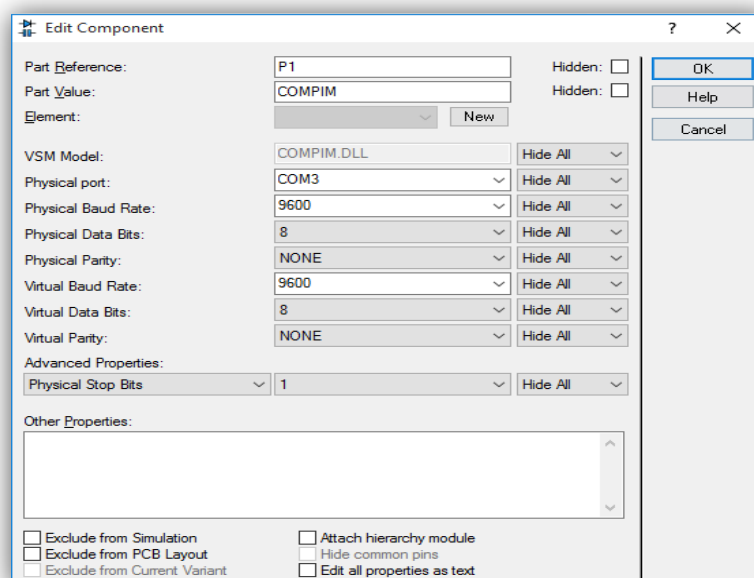
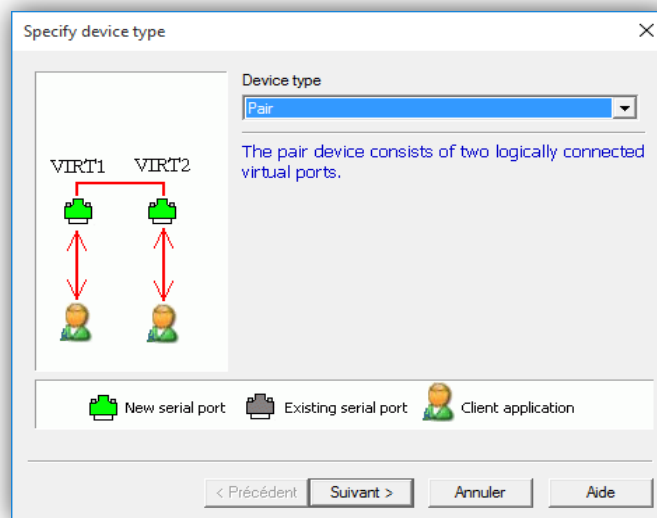


Figure 3.12 : Paramètres COMPIM.

### ➤ *Etape 02 : Connecter Proteus à l'application Blynk*

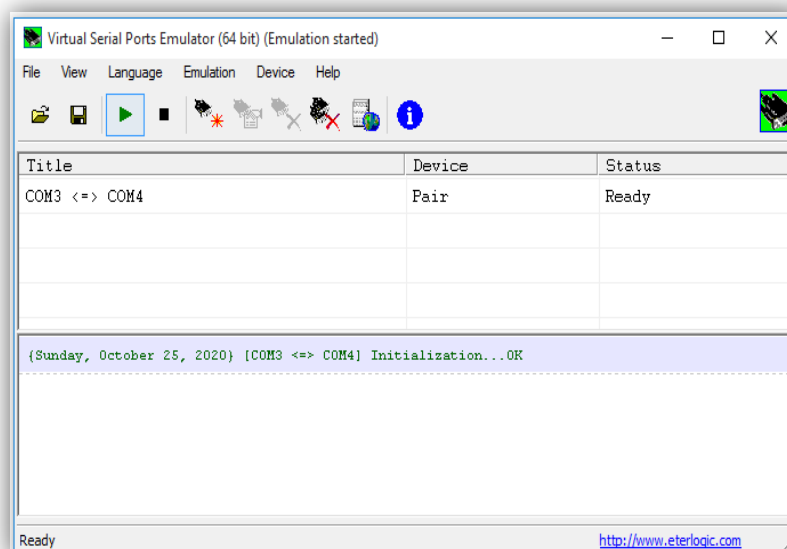
Dans la première étape, l'application logicielle VSPE (Virtual Serial Ports Emulator) partage les données du port série physique pour l'application Blynk dans le Smartphone, comme suit : Lancez l'application et créez un nouveau dispositif.

Ensuite, spécifiez le type de dispositif, le dispositif de paire se compose de deux ports virtuels connectés logiquement, comme illustré à la figure 3.13.



**Figure 3.13 :** Spécifiez le type de devise.

Ensuite, reliez le port physique de l'ordinateur COM3 considéré comme un port série virtuel 1 avec COM4 considéré comme un port série virtuel 2, comme on peut le voir sur la figure 3.14.



**Figure 3.14 :** Émulation de ports série virtuels.



## Chapitre 03 : Conception & Simulation

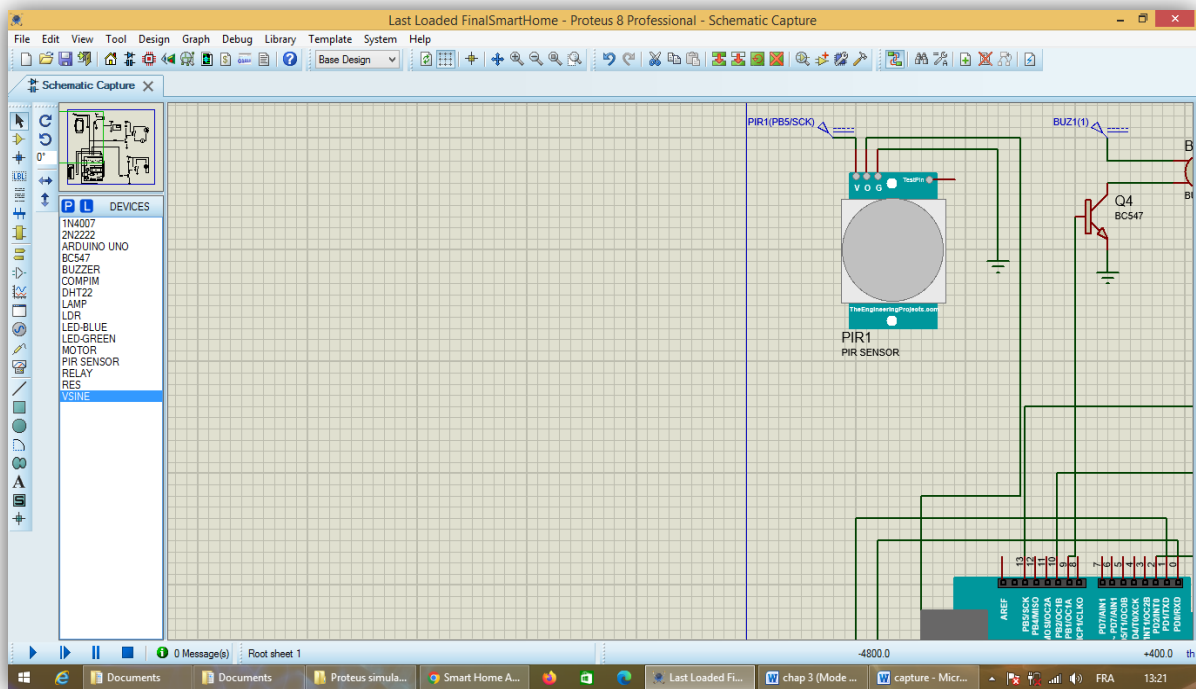


Figure 3.17 : Model de capteur de mouvement(PIR).

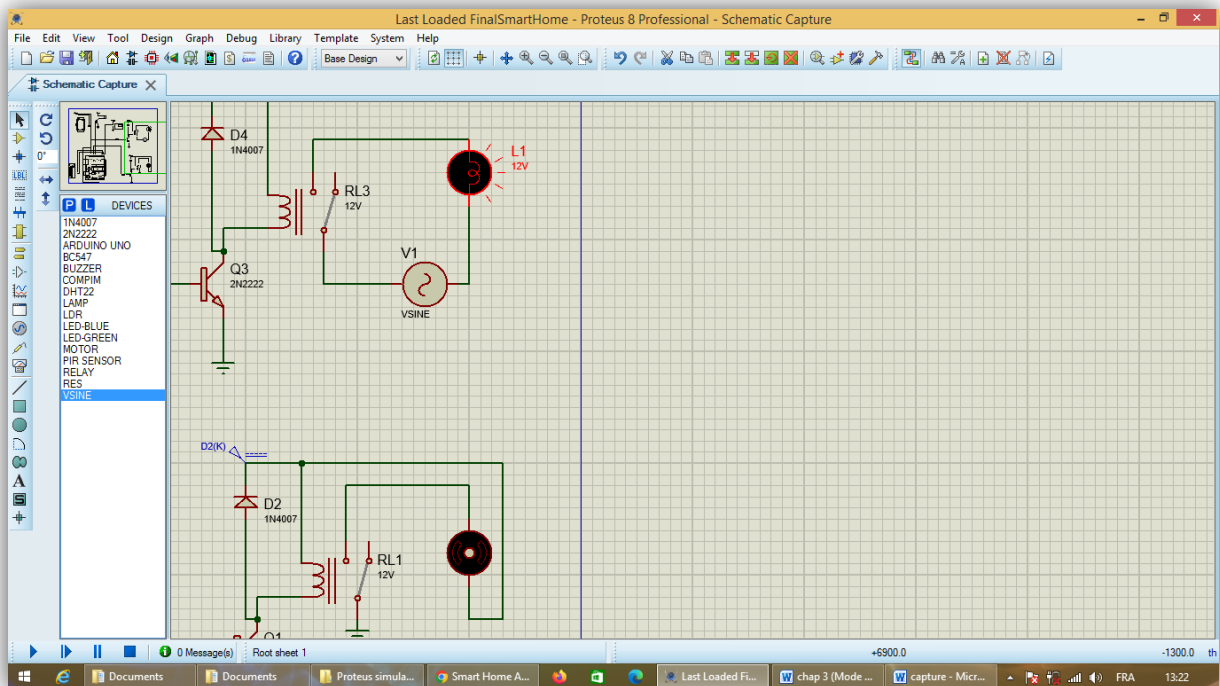


Figure 3.18 : Lampe et moteur utilisée comme un climatiseur.

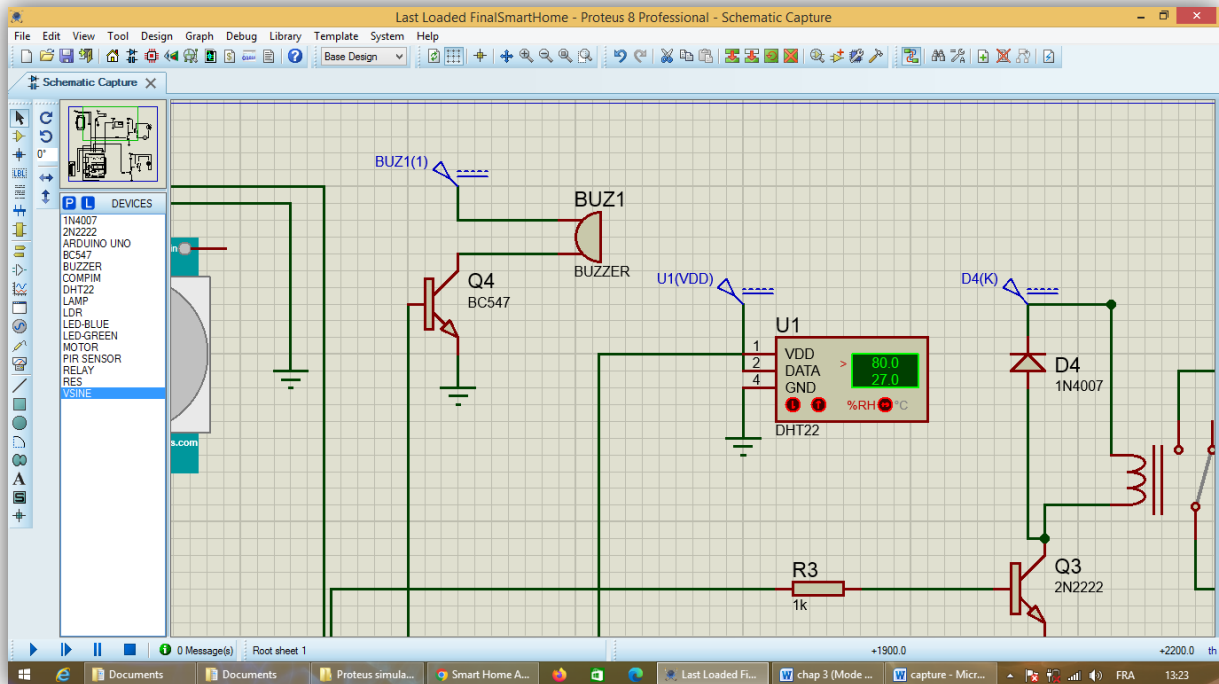


Figure 3.19 : Capteur DHT 22 (humidité et température).

### 6. Conclusion

Dans ce chapitre nous avons faits une modélisation et simulation des éléments avec des périphériques basés sur Internet of Things ces derniers utilise un réseau internet. Nous avons également présenté le programme UML-RT pour concevoir diagramme de cas d'utilisation d'une maison intelligente et Enfin pour assimilé nous avons utilisé le logiciel ISIS (Proteus).

***Chapitre 04 :***

***Implémentation***

## 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.

## 2. Disposition du système (system layout)

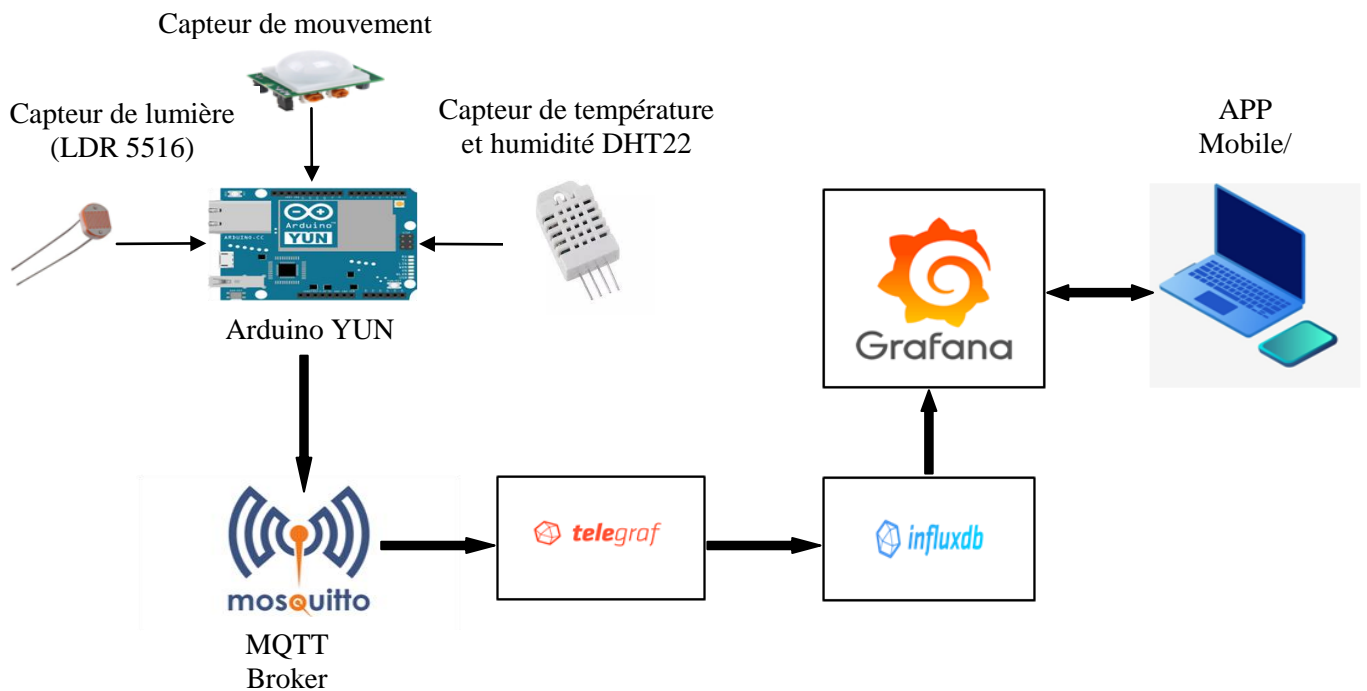
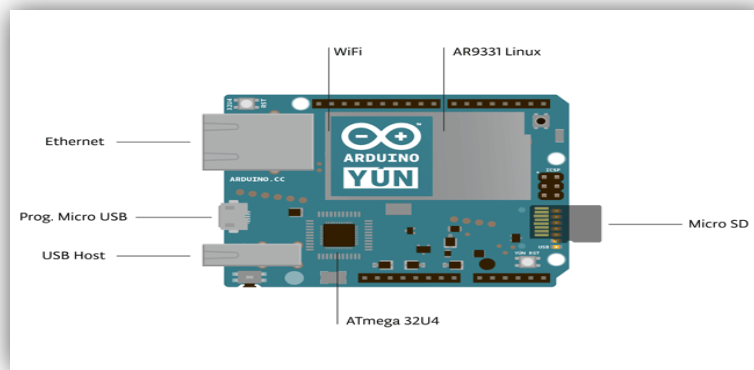


Figure 4.1 : Disposition du système.

## 3. Matérielle utilisés

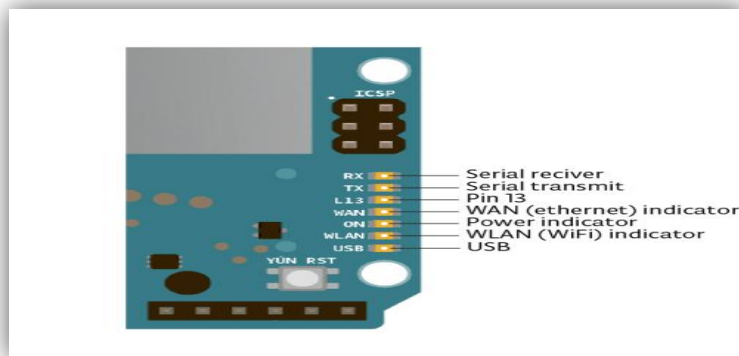
### 3.1. la carte Arduino YUN

La carte Arduino YUN est une carte similaire à la carte Arduino Leonardo mais qui possède des fonctionnalités avancées permettant de réaliser des prototypes d'objets connectés. La carte YUN possède une gestion intégrée des liaisons Ethernet et Wi-Fi, un port USB-A, un logement pour carte Micro SD, 20 broches d'E/S numériques (dont 7 peuvent servir de sorties MLI/PWM, et 12 d'entrées analogiques), un oscillateur à quartz de 16 MHz, un connecteur micro USB, une embase ICSP, et 3 boutons d'initialisation (reset) [15].



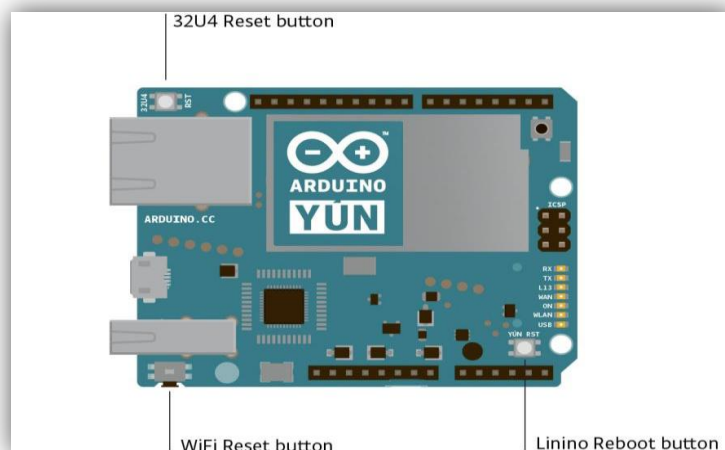
**Figure 4.2 :** Carte Arduino YUN.

- On retrouve aussi plusieurs LEDs de statuts sur le YUN indiquant :  
L'alimentation (ON), la connexion Wifi(WLAN), la connexion Ethernet (WAN) et la connexion USB (USB). De plus, le pin 13 (L13) est connecté à l'une des LEDs de statut [15].



**Figure 4.3 :** les différentes LEDs de la carte Arduino.

- Les trois boutons de réinitialisation avec des fonctions différentes sur la carte sont représentés ainsi [15]



**Figure 4.4 :** les trois boutons de réinitialisation de la carte Arduino.

### 3.2. 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 :  $s = \mathbf{F}(\mathbf{m})$  où  $s$  est la grandeur de sortie ou la réponse du capteur [16].

#### 1.1.1. Les caractéristiques d'un capteur

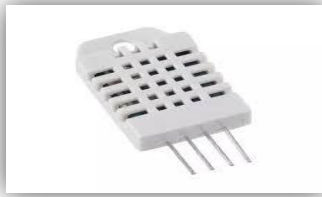
- Etendue de mesure : Valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : Plus petite variation de grandeur mesurable par le capteur.
- Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : Aptitude du capteur à donner une mesure proche 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.

#### 1.1.2. 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.

##### ***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 4.5* : Capteur de température et humidité DHT22.

### ***Capteur de mouvement PIR***

Un capteur de mouvement PIR (Passive Infrared Sensor) est un capteur électronique capable de détecter un mouvement dans son champ de vision grâce aux rayons infrarouges. Aussi appelé capteur thermoélectrique, Ils sont très souvent utilisés dans les systèmes d'alarme ou de détection de présence pour son faible coût et son efficacité [17].



*Figure 4.6* : Capteur de mouvement PIR.

### ***3.3. 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 ( $27^{\circ}\text{C}$ ) 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 [18]. Nous avons utilisé de petits ventilateurs comme indiqué sur la figure 4.7.



*Figure 4.7* : Ventilateur a 12V.

### 3.4. 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 entre eux à l'intérieur de la breadboard.

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

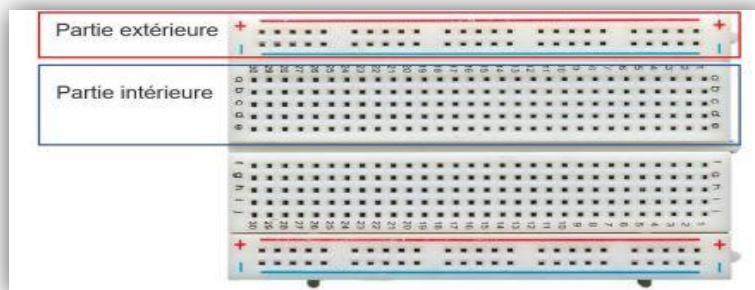


Figure 4.8 : Breadboard.

### 3.5. LED

Une LED (en français : DEL : diode électroluminescente) est un composant optoélectronique, 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 [19].

La LED possède 2 broches :

Le long: positive.

Le petit: négative.



Figure 4.9 : LEDs.

### 3.6. *Buzzer*

C'est un composant électromécanique ou piézoélectrique, qui produit un son distinct lorsqu'on lui applique une tension, la fréquence de ce signal sonore est alors relative au signal appliqué. Il ya deux types de buzzer, actif et passif, la différence entre eux, c'est que les buzzer actif fonctionne en tension continu, alors que les buzzer passif fonctionne seulement en tension alternatif [17].



*Figure 4.10* : Module buzzer.

### 3.7. *Fils de connexion*

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



*Figure 4.11* : Câbles de prototypage.

### 3.8. *Câble USB*

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



*Figure 4.12* : Câble USB.

### 4. Plateformes logicielles

Cette partie est dédiée à la représentation des plateformes informatiques utilisées dans le développement de notre système.

#### 4.1. Arduino IDE

Arduino se compose à la fois d'une carte de circuit physique programmable (souvent appelée microcontrôleur) et d'un logiciel, ou IDE (Integrated Développement Environnement) qui fonctionne sur l'ordinateur, utilisé pour écrire et télécharger du code informatique sur le tableau physique. Elle offre cinq zones, chacune de ces zones à une fonctionnalité bien précise [17].

- Une barre de menus.
- Une barre d'icônes : la série des icônes sous la barre de menu permet de réaliser diverses actions (vérifier le code, télé verser le code, ouvrir un nouveau projet, ouvrir un projet de programme Arduino existant, sauvegarder le projet, ouvrir le moniteur série pour communiquer avec l'Arduino)
- Une barre d'onglets : Chaque fichier du projet est présenté dans cet IDE par un onglet.
- Une zone d'édition : l'endroit où le code source doit être saisi.
- Une zone d'information et de statut : cette zone affiche divers information sur le programme, les erreurs de compilation, le transfert de programme ...

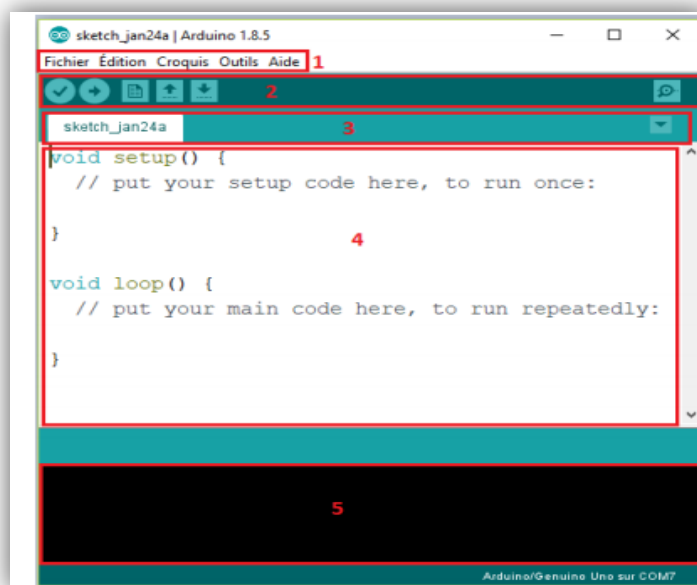


Figure 4.13 : Composants de l'écran principal de l'IDE Arduino.

## Chapitre 04 : Implémentation

---

Il se connecte au matériel Arduino et Genuino pour télécharger des programmes et communiquer avec eux. Le langage de programmation utilisé est le C++, lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties.



*Figure 4.14* : Arduino IDE.

### **4.2. Mosquitto (MQTT Broker)**

Eclipse Mosquitto est un courtier de messages open source (sous licence EPL / EDL) qui implémente les versions 5.0, 3.1.1 et 3.1 du protocole MQTT. Mosquitto est léger et convient à tous les appareils, des ordinateurs mono cartes basse consommation aux serveurs complets.

Le protocole MQTT fournit une méthode légère d'exécution de la messagerie à l'aide d'un modèle de publication / abonnement. Cela le rend adapté à la messagerie Internet des objets, par exemple avec des capteurs de faible puissance ou des appareils mobiles tels que des téléphones, des ordinateurs intégrés ou des microcontrôleurs.

Le projet Mosquitto fournit également une bibliothèque C pour l'implémentation des clients MQTT, et les très populaires clients MQTT en ligne de commande `mosquitto_pub` et `mosquitto_sub`.

Mosquitto fait partie de la Fondation Eclipse, est un projet [iot.eclipse.org](http://iot.eclipse.org) et est sponsorisé par [cedalo.com](http://cedalo.com).

Nous avons choisi Mosquitto pour sa simplicité d'implémentation, il suffit juste d'installer le package et ajouter la règle comme cité avant ensuite on l'exécute sur le serveur [20].



*Figure 4.15* : Logo de Telegraf.

### 4.3. Telegraf

Le projet Telegraf est un greffon logiciel permettant les interactions entre MQTT et InfluxDB. Il est subdivisé en deux catégories [21]:

- **Telegraf Input** : ce greffon permet de lire les messages sur des topics MQTT et d'injecter les données dans MQTT. Les messages collectés par ce greffon sont attendus à un format donné.
- **Telegraf Output** : publie vers MQTT des données insérées dans une base de données InfluxDB.



Figure 4.16 : Logo de Telegraf.

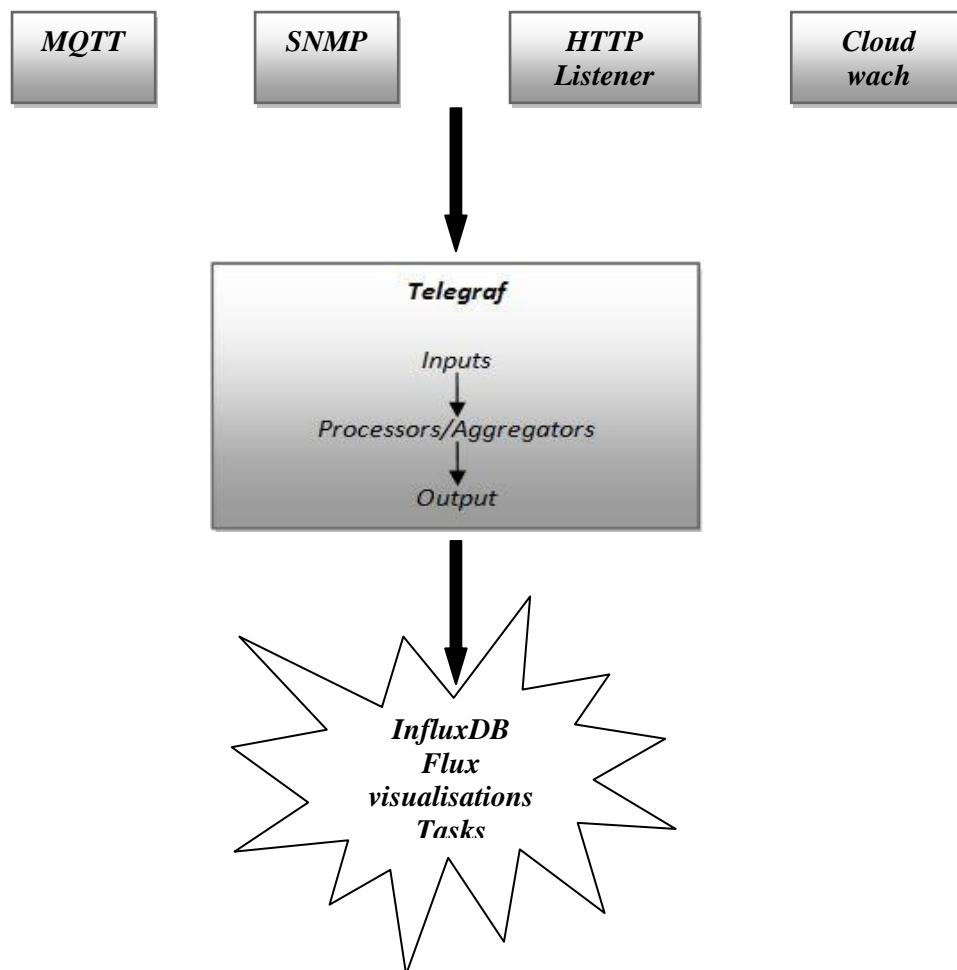


Figure 4.17 : Présentation du système de plug-in Telegraf.

### 4.4. InfluxDB

Collecter des données sur un broker MQTT permet d'en disposer pour d'autres types d'applications. Un excellent exemple est l'injection dans une base de données de type TimeSeries (spécialisée dans la gestion des grandes séries de données).

InfluxDB est un excellent outil open source pour gérer les données de type TimeSeries. Ces séries de données seront ensuite exploitées avec l'outil de représentation graphique Grafana.

InfluxDB est une base de données de type TimeSeries, un type assez récent dans l'histoire des bases de données.

Ce type de base de données dispose de points communs avec les bases de données traditionnelles (ex : SQLite), mais également des caractéristiques spécifiques très appréciées dans le domaine des objets connectés.

Enfin, une base de données TimeSeries n'est pas une base de données relationnelle. Elle ne se crée donc pas de la même façon [21].



Figure 4.18 : Logo d'InfluxDB.

### 4.5. Grafana

Grafana est un excellent outil open source orienté sur la visualisation graphique des données.



Figure 4.19 : Écran d'accueil de Grafana.

## Chapitre 04 : Implémentation

---

Cette approche « graphique » permet de naviguer naturellement dans un historique de données et d'évaluer globalement une situation d'un simple coup d'œil. En effet, il est plus simple d'évaluer la tendance d'une courbe que de lire une colonne de chiffres contenant ces mêmes informations.

Grafana est une solution de type Web, son administration et la visualisation des données s'effectuent depuis un simple navigateur Internet [21].



*Figure 4.20* : Logo de Grafana.

### 5. Explication de code

Dans la première étape de la programmation, `#include`, qui est utilisé pour inclure des bibliothèques extérieures dans l'esquisse. Cela donne accès à un grand nombre de bibliothèques C standard, ainsi qu'à des bibliothèques écrites spécialement pour Arduino. Il est nécessaire d'inclure la bibliothèque pour connecter la partie matérielle avec la carte Arduino. Les bibliothèques utilisées dans ce projet sont présentées dans le Code\_1 :

Code 01

```
#include <Wire.h>
#include <DHT.h>
#include <YunClient.h>
#include <PubSubClient.h>
#include <Bridge.h>
```

Dans le Code 02, les variables globales ont été déclarées afin d'être utilisées plus tard dans le programme dans le stockage des données acquises du capteur. De plus, il est important de définir le type, le nom et la valeur initiale de la variable :

## Chapitre 04 : Implémentation

---

### Code 02

```
// Définition de constants
#define DHTPIN 7 // Le pin de connexion du capteur
#define DHTTYPE DHT22 // Le type de capteur DHT 22
#define LED 5 //définir le numéro de pin du led
#define BuzzerPin D2 //définir le numéro de pin du buzzer
(alarme)
float h; // pour stocké la valeur de l'humidité
float t; // pour stocké la valeur de la température

// les topic pour la température et de l'humidité
const char* topic_t="capteur/temperature";
const char* topic_h="capteur/humidité";
```

Après la déclaration des variables, il est indispensable de définir L'adresse du Broker, l'instance de la classe EthernetClient, de la classe Mqtt, et de la classe DHT, Comme expliqué dans le code\_3 :

### Code 03 :

```
// L'adresse du Broker
const char* server = "192.168.1.240";
// L'instance de la classe EthernetClient
YunClient YClient;
// L'instance de la classe MQTT
PubSubClient mqtt(YClient);
// L'instance de la classe DHT
DHT dht(DHTPIN, DHTTYPE);
```

Dans le Code 04, on se trouve les chaînes de caractères pour transmettre des données avec MQTT :

### Code 04 :

```
// Les chaîne de caractère pour transmettre des données avec MQTT
char hum[10],temp[10];
```

## Chapitre 04 : Implémentation

---

La fonction "callback()" a été écrite pour lire les commandes d'envoi du moniteur série et contrôler l'allumage et l'extinction de la LED comme indiqué dans le Code 05 :

Code 05 :

```
void callback(char *topic, byte *payload, unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");
  String message;
  for (int i = 0; i < length; i++) {
    message = message + (char) payload[i]; // convert *byte to string}
  Serial.print(message);
  if (message == "on") { digitalWrite(LED, LOW); } // LED on
  if (message == "off") { digitalWrite(LED, HIGH); } // LED off
  Serial.println();
}
```

Ensuite, pour des bousions de compatibilité avec les anciennes versions de l'IDE nous mettons les déclarations de fonctions que nous allons utiliser cela avant la fonction "setup ()" :

Code 06 :

```
// compatibilité avec les anciennes version de l'IDE
void reconnect(void);
void pubCapteur(void);
```

## Chapitre 04 : Implémentation

---

Code 07 est la fonction "setup ()", cette fonction définit l'état initial de l'Arduino au démarrage, elle initialise les bibliothèques et la connexion au Broker :

Code 07 :

```
void setup()
{
  // Initialisation de la liaison série
  Serial.begin(9600);
  // Initialisation du capteur DHT22
  dht.begin();
  // Initialisation de la liaison entre le MCU et la SoC de l'arduino Yùn
  Bridge.begin();
  // Initialisation de la connexion au Broker
  mqtt.setServer(server, 1883);
  mqtt.setCallback(callback);
  delay(1500);
  // publish and subscribe
  mqtt.publish(topic, "HELLO");
  mqtt.subscribe(topic);
}
```

Nous arrivons maintenant au cœur de notre programme "loop()", cette fonction pour vérifier si la connexion au Broker est réalisée envoyer les données, Sinon se reconnecté au Broker comme indiqué dans le Code 08 :

Code 08 :

```
void loop(){
  if(mqtt.connected()){
    // Si la connexion au Broker est réalisé envoyer les données
    pubCapteur();
    mqtt.loop();
  }else {
    // Sinon se reconnecté au Broker
    reconnect();
  }
}
```

## Chapitre 04 : Implémentation

---

Dans le Code 09 la fonction pubCapteur , elle commence par lire les valeurs de température et d'humidité , puis la conversion des valeurs capturé en chaine de caractères par la fonction décimale to string, enfin les publie dans les topics qui leur correspondent :

Code 09 :

```
void pubCapteur(void){
  // récupération des valeurs depuis le capteur
  h = dht.readHumidité();
  t = dht.readTemperature();
  // Conversion des valeurs capturé en chaine de caractères
  dtostrf(h, 7, 4, hum);
  dtostrf(t, 7, 4, temp);
  // Publication des donnée sur la Broker
  mqtt.publish(topic_h,hum);
  mqtt.publish(topic_t,temp);
  delay(10000); }
```

Enfin, pour la fonction " reconnect ()" elle boucle tant que la connexion avec le broker n'est pas rétablie et si la connexion n'est pas établie elle nous donne une indication de la cause de cette déconnexion via la méthode state :

Code 10 :

```
void reconnect(void) {
  // Boucle jusqu'à ce que nous soyons reconnectés
  while (!mqtt.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Tentative de connexion
    if (mqtt.connect("oha")) {
      Serial.println("connected");
      // Une fois connecté, publiez une annonce...
    } else {
      Serial.print("failed, rc=");
      Serial.print(mqtt.state());
      Serial.println(" try again in 5 seconds");
      // Attendre 5 secondes avant de réessayer
      delay(5000); }
  }
```

### 6. Le projet de stage

Nous avons écrit un programme pour contrôler des leds et des moteurs avec le WIFI mobile.

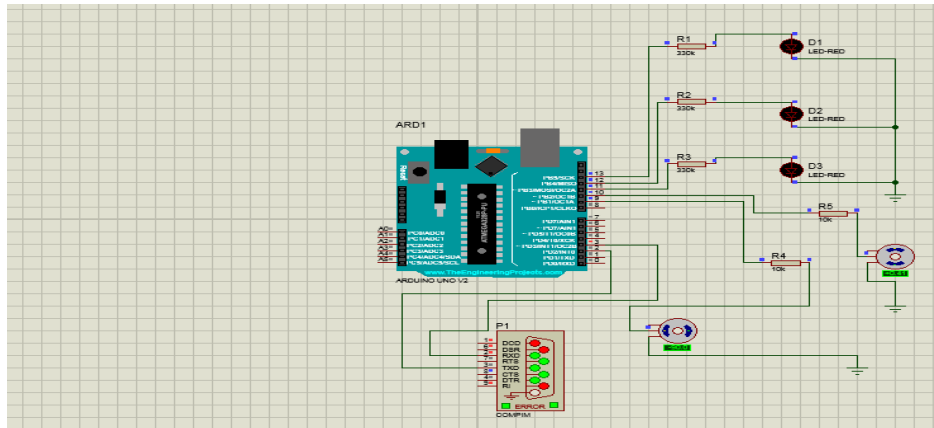


Figure 4.21 : Le modèle de circuit électronique du projet sous Proteus.

#### 6.1. Le code

```
#include <BlynkSimpleStream.h><Servo.h>

Servo myservo;

Servo myservoe;

// create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).

char auth[] = "3NyiiB8i6r6EWvRfuS43nSb8NGexA9KH";

void setup()

{

  Serial.begin(9600);

  Blynk.begin(auth, Serial);

  myservo.attach(9); // attaches the servo on pin 9 to the servo object

  myservoe.attach(10);

  pinMode(11, OUTPUT);

  pinMode(12, OUTPUT);

  pinMode(13, OUTPUT);
```

## Chapitre 04 : Implémentation

---

```
}  
BLYNK_WRITE(V1) //Button Widget is writing to pin V1  
{  
  int pinData = param.asInt();  
  if(pinData==1){  
    digitalWrite(11, HIGH);  
  }else{  
    digitalWrite(11, LOW);  
  }  
}  
  
void loop()  
{  
  Blynk.run();  
  
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees  
    // in steps of 1 degree  
    myservo.write(pos);      // tell servo to go to position in variable 'pos'  
    delay(15);               // waits 15ms for the servo to reach the position  
  }  
  
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees  
    myservo.write(pos);      // tell servo to go to position in variable 'pos'  
    delay(15);               // waits 15ms for the servo to reach the position  
  }  
  
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees  
    // in steps of 1 degree  
    myservoe.write(pos);     // tell servo to go to position in variable 'pos'  
    delay(15);               // waits 15ms for the servo to reach the position  
  }  
  
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees  
    myservoe.write(pos);     // tell servo to go to position in variable 'pos'  
    delay(15);               // waits 15ms for the servo to reach the position  
  }  
}
```

```
}  
digitalWrite(11, HIGH);  
delay(2000);  
digitalWrite(11, LOW);  
delay(1000);  
// LED 2  
digitalWrite(12, HIGH);  
delay(1000);  
digitalWrite(12, LOW);  
delay(2000);  
//LED 3  
digitalWrite(13, HIGH);  
delay(1000);  
digitalWrite(13, LOW);  
delay(2000);  
}
```

### 6.2. Les étapes

1. Téléchargé l'application BLYNK IOT
2. Téléchargé la bibliothèque et installation sur Arduino
3. Création de la plateforme sur l'application plus le patch de connexion

```
char auth[] = "3NyiiB8i6r6EWvRfuS43nSb8NGexA9KH";
```

4. Envoyer sur la boîte email
5. Puis jouer avec les commandes

#### **Les équipements sur PROTEUS :**

- ARDUINO UNO
- COMPIN
- LED
- MOTEUR PWMSERVO
- RÉSISTANCE

## *Conclusion générale*

Grâce aux progrès des outils informatiques et à la recherche sur les systèmes embarqués et l'IoT, les contributions produites par cette recherche ont grandement amélioré l'industrie de la maison intelligente.

Dans ce travail, nous avons conçu et mis en œuvre un système efficace de gestion de certains appareils électroménagers (lumières, climatiseurs, ...) basé sur l'Internet des objets.

L'objectif principal de ce travail est d'éviter les pertes de puissance et de réduire la consommation d'énergie des appareils électriques en éteignant les appareils lorsqu'ils ne sont pas utilisés.

Les travaux futurs sur un système de gestion de l'énergie domestique pourraient se dérouler de plusieurs manières. La chose la plus évidente est d'installer un wattmètre Arduino pour mesurer l'énergie consommée dans la maison avant et après l'installation de notre système afin de démontrer la faisabilité et l'efficacité de notre solution. Nous suggérons et encourageons l'utilisation d'énergies renouvelables, comme les panneaux solaires..., à la maison pour améliorer l'efficacité énergétique et réduire les émissions de dioxyde de carbone.

# *Bibliographie*

- [2] Dyhia Hales, G. C. (2017-2018). Etude de l'implémentation de ARM Cortex M3 dans un système embarqué. Tizi-Ouzou, Génie Electronique.
- [3] AICHOUCHI Manar El Houda, C. N. (2018, Juillet). Conception et réalisation d'un système embarquée pour mesurer les paramètres d'une ligne téléphonique. BLIDA, ELECTRONIQUE.
- [4] Laabidine, L. M. (2019, Juillet). Plateforme de développement pour l'internet des objets (IdO). Guelma, Informatique.
- [5] Abdelaziz, K. (s.d.). Le développement d'un système embarqué implémentant le Peer-to-Peer pour la TVoIP. Setif, Informatique.
- [6] BENARIB Hadil, B. T. (2020, septembre 16). Sécurisation d'une Smart Home par Reconnaissance Vocale et Faciale. Bordj Bou Arreridj, Electronique.
- [7] Yusuf Perwej, M. A. (2019, 2 27). An Extended Review on Internet of Things (IoT) and its Promising Applications. New York, USA, Computer Science.
- [9] ABDERRAHMEN, R. (2018, Juin). Développement d'un système d'IoT (Internet of Things) pour le Smart Lighting sous la Plateforme IBM. Oum El Bouaghi, Mathématiques et Informatique.
- [10] M. HAOUA Zakaria, M. M. (2018-2019). Vers des Bâtiments Intelligents pour l'élevage de volailles. BLIDA, Informatique.
- [11] B. Selic, G. Gullekson, and P. T. Ward, Real-time object-oriented modeling. John Wiley & Sons New York, 1994, vol. 2.
- [12] Selic, B., Limited, O.: Real-time object-oriented modeling (ROOM). In: Proceedings of the 2Nd IEEE Real-Time Technology and Applications Symposium (RTAS '96), Ser. RTAS '96, pp. 214. IEEE Computer Society, Washington, DC (1996).
- [13] SADI, A. (2018, 9 27). Conception et simulation d'une carte d'acquisition de données météorologique en temps réels à base d'un microcontrôleur 18F45K22. TIZI-OUZOU, Electronique.
- [14] Elmehri, M. A. (2020). Design of a Control System for an Indirect Solar Dryer. Mechanical Engineering.

[15] IZZA, K. (2016, 7 11). CONCEPTION ET REALISATION D'UN DATA-LOGGER A 8 VOIES A BASE D'UNE CARTE ARDUINO YUN. TIZI-OUZOU, Génie électrique.

[17] YAHY Amina, K. L. (2018, 10 2). Contrôle et suivi d'une maison intelligente via internet. bouira, Génie électrique.

[18] Linda BOUDJADJA, M. K. (2019-2020). Conception & Simulation de Fonctionnement d'une Maison Intelligente. Jijel, Electronique.

[19] ADDOU Asmaa, B. N. (2019 -2020). Réalisation d'une maison intelligente à base d'Arduino. Ain Témouchent, Electricue.

[20] EL-AMINE, A. T. (2017). Usage du protocole MQTT dans une application de suivi. BISKRA, informatique.

## *Webographie*

[1] <https://www.panadisplay.com/info/embedded-system-history-33071082.html>

[8] <https://www.unge.net/pdfs/706-villeintelligente.pdf>

[16] [http://lgt.garnier.free.fr/espace\\_ee\\_fichiers/capteurs/capteurs.pdf](http://lgt.garnier.free.fr/espace_ee_fichiers/capteurs/capteurs.pdf)

[21] <https://www.editions-eni.fr/open/mediabook.aspx?idR=18038fc4ea1c30eaea9496c9a3ee4337>

