



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option :

Réseaux et Systèmes Distribués (R.S.D)

Thème

Mise en place et exploitation d'une infrastructure d'internet des objets

Réalisé par :

- HAMLILI Moussa
- TAHRI Mohammed Yacine

Présenté le 22 Juin 2020 devant le jury composé de :

- DIB-MALTI Djawida (Encadrant)
- ABDELDJELIL Hanane (Encadrant)
- SELADJI Yasmine (Président)
- SETTOUTI Ahmed Khalid Yassine (Examineur)

Dédicaces

À nos très chers parents

Nous vous devons ce que nous sommes aujourd'hui, grâce à votre amour, votre patience et vos innombrables sacrifices.

Que ce modeste travail, soit pour vous une petite compensation et reconnaissance pour tout ce que vous avez fait.

Que Dieu, vous préserve et vous procure santé et longue vie afin que nous puissions à notre tour vous combler.

À nos très chers frères et sœurs

Aucune dédicace ne pourrait exprimer assez profondément ce que nous ressentons envers vous.

Nous vous dirons tout simplement, un grand merci, nous vous aimons.

À nos très chers ami(e)s

En témoignage de l'amitié sincère qui nous lie et les bons moments passés ensemble. Nous vous dédions ce travail en vous souhaitons un avenir radieux et plein de réussites.

Remerciement

En premier lieu nous remercions DIEU tout puissant de nous avoir donné la patience, la santé et la volonté pour achever ce travail.

Et nos remerciements vont tout particulièrement à nos parents, pour leur soutien et leur patience, nous remercions aussi le président de jury madame SELADJI Yasmine ainsi que notre examinateur monsieur SETTOUTI Ahmed Khalid Yassine qui ont accepté d'examiner et de participer à notre soutenance de mémoire.

Nous aimerions adresser plus qu'un merci pour nos deux encadreurs madame MALTI Djawida et madame ABDELJELIL Hanane qui ont su partager leur savoir-faire, leurs connaissances et leur temps pour nous porter aide pendant et hors leurs heures de travail.

Résumé

Le concept de l'Internet des Objets (IoT) a attiré beaucoup d'attention ces dernières années en plusieurs domaines. L'IoT consiste à relier des objets connectés via internet en utilisant des protocoles de communication afin d'échanger des informations et transmettre des données entre ces objets. Dans ce travail, nous avons mis en place une infrastructure qui inclut toutes les couches d'une architecture IoT. Nous avons utilisé cette infrastructure pour développer et déployer deux systèmes IoT : un système pour la surveillance de distances et un système pour la surveillance de température.

Abstract

The concept of the Internet of Things (IoT) has attracted much attention over the last few years in several areas. The IoT consists of linking connected objects via the Internet by using communication protocols in order to exchange information and transmit data between these objects. In this context we established an infrastructure that includes all the layers of an IoT architecture. This infrastructure is used to develop two IoT systems : a system for distance monitoring and a system for temperature monitoring.

ملخص

جذب مفهوم إنترنت الأشياء (IoT) الكثير من الاهتمام على مدى السنوات القليلة الماضية في العديد من المجالات. تتميز إنترنت الأشياء بربط الأشياء عبر الإنترنت باستخدام بروتوكولات الاتصال من أجل تبادل المعلومات ونقل البيانات بين هذه الأجهزة. في هذا السياق، قمنا بإنشاء بنية تحتية تشمل كل الطبقات لمعمارية إنترنت الأشياء. تُستخدم هذه البنية التحتية لتطوير نظامين من إنترنت الأشياء : نظام لمراقبة المسافات و نظام لمراقبة درجة الحرارة.

Table de matière

Introduction Générale	9
Chapitre 1 : Concepts fondamentaux en IoT.....	12
1. Introduction.....	13
2. Définitions	13
2.1. IoT	13
2.2. Objet connecté	14
3. Les couches d'un modèle IoT	14
4. La couche perception	15
4.1. Les capteurs	16
4.2. Les actionneurs.....	17
4.3. Les cartes électroniques	18
5. La couche réseau.....	20
5.1. Les réseaux de communication	21
5.1.1. Réseau longue portée	22
5.1.2. Réseau courte portée	23
5.2. Les passerelles IoT	25
6. La couche middleware	25
6.1. Base de données.....	26
6.2. Cloud Computing.....	26
6.3. Big Data.....	26
7. La couche application.....	26
7.1. Application de bureau	27
7.2. Application WEB	27
7.3. Application Mobile	27
8. Protocoles d'application pour l'IoT	27
8.1. MQTT (Message Queuing Telemetry Transport).....	27
8.2. AMQP (Advanced Message Queuing Protocol).....	28
8.3. CoAP (Constrained Application Protocol).....	29
8.4. XMPP (Extensible Messaging and Presence Protocol)	29
9. Domaines d'applications de l'IoT	30
9.1. La domotique.....	30

9.2. Le transport	30
9.3. La santé.....	30
9.4. L'industrie	31
10. Conclusion	31
Chapitre 2 : Mise en place de l'infrastructure IoT	32
1. Introduction.....	33
2. La couche perception	34
3. La couche réseau	34
4. La couche middleware.....	34
5. La couche application	36
6. Exemples de systèmes IoT	36
6.1. Lumière intelligente.....	37
6.2. Alarme fumée intelligente.....	37
7. Conclusion	37
Chapitre 3 : Exemples de systèmes IoT implémentés.....	38
1. Aperçu	39
2. Système IoT de surveillance de distances	40
3. Système IoT de surveillance de températures.....	42
4. Conclusion	44
Conclusion Générale.....	45
Annexes	48
1. Branchement des capteurs.....	49
2. Branchement des actionneurs	50

Table de Figures

Figure 1 : Aperçu sur l'architecture IoT en 4 couches	14
Figure 2 : Exemples de capteurs	16
Figure 3 : Exemples d'actionneurs	17
Figure 4 : Les composants d'une carte électronique	18
Figure 5 : Exemples de cartes électroniques	20
Figure 6 : Comparaison des technologies de communication	21
Figure 7 : Architecture d'un réseau LPWAN	22
Figure 8 : Réseaux de communication courte portée.....	23
Figure 9 : Aperçu sur les passerelles IoT	25
Figure 10 : Principe de fonctionnement de MQTT	28
Figure 11 : Principe de fonctionnement d'AMQP	28
Figure 12 : Principe de fonctionnement de COAP	29
Figure 13 : Principe de fonctionnement de XMPP	29
Figure 14 : Aperçu sur l'infrastructure mise en place.....	33
Figure 15 : Aperçu sur le traitement et le stockage de données	35
Figure 16 : Aperçu de l'interface WEB du broker	36
Figure 17 : Automate de fonctionnement de système IoT	39
Figure 18 : Automate de fonctionnement de système IoT - surveillance de distances	40
Figure 19 : Aperçu de l'interface WEB du broker - historique de distances.....	41
Figure 20 : Application mobile de système IoT de surveillance de distances.....	42
Figure 21 : Automate de fonctionnement de système IoT - surveillance de températures	42
Figure 22 : Aperçu de l'interface WEB du broker - historique de températures.....	43
Figure 23 : Application mobile de système IoT de surveillance de températures	44

Figure 24 : Schéma de câblage de capteurs..... 49

Figure 25 : Schéma de câblage d'actionneurs 50

Introduction Générale

Les recherches et les développements dans le domaine de l'internet ne cessent d'accroître et de se transformer progressivement en un réseau étendu, appelé « Internet des objets ». L'internet des objets (Internet of Things en anglais, abrégée par IoT) est un concept et non pas une technologie ou des appareils spécifiques. L'IoT est la volonté d'étendre le réseau internet et les échanges de données aux objets du monde physique, tels que : une voiture, une montre, etc.

L'expression de l'internet des objets (IoT), a été mise en place par Kevin Ashton dans une présentation en 1999 pour caractériser une architecture de l'information basée sur Internet [12]. Après, le terme est devenu populaire et largement utilisé. L'IoT est assez proche du M2M, c'est une version plus étendue du Machine to Machine dans la mesure où le traitement et la transmission des informations passent par un réseau beaucoup plus élargi. En 2003, la population mondiale a frôlé les 6 milliards d'individus et un demi-milliard d'appareils connectés à Internet. Avec l'apparition des smartphones, le nombre d'appareils connectés à Internet a atteint 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards. Les experts estiment que 50 milliards d'appareils seront connectés à la fin du 2020. Ces estimations ne prennent pas en considération l'évolution rapide d'Internet ni des avancées technologiques, mais uniquement les faits de l'heure actuelle. Le nombre de capteurs connectés à Internet pourrait augmenter de plusieurs millions, voire de plusieurs milliards du fait que tout ce qui existe se connecte (lampes, maisons, personnes, chaussures...).

Actuellement, le développement des applications à base d'IoT est intégré dans de nombreuses tâches de notre vie quotidienne tels que: l'industrie, la santé, le transport, la domotique, l'agriculture, etc. Les objets connectés servent à réduire des éléments de dépenses, tels que: le temps et la quantité d'effort.

Dans ce contexte, notre projet consiste à étudier l'architecture des applications à base d'IoT, afin de concevoir et mettre en place une infrastructure qui inclut les différentes couches d'un modèle IoT. Ensuite nous utilisons cette infrastructure pour développer et déployer des exemples de systèmes IoT qui permettent de contrôler à distance des objets connectés.

Le reste de ce manuscrit est organisé en trois chapitres et une conclusion générale. Le premier chapitre décrit les concepts fondamentaux en l'IoT. Le second chapitre présente notre première

contribution qui consiste à la mise en place d'une infrastructure d'internet des objets. Le troisième chapitre décrit notre deuxième contribution qui consiste à utiliser l'infrastructure mise pour développer et déployer deux exemples de système IoT: un exemple de surveillance de distance et un exemple de surveillance de température. Enfin la conclusion générale récapitule les différents points abordés dans ce manuscrit.

Chapitre 1 :

Concepts

fondamentaux en IoT

1. Introduction

Dans ce chapitre, nous présentons le concept général d'IoT. Dans la section 2 nous définissons les notions de l'IoT ainsi que de l'objet connecté. Dans les sections de 3 à 7 nous détaillons les différentes couches d'un modèle IoT. Nous enchainons par la définition de quelques protocoles utilisés dans la mise en place d'une architecture à base d'IoT dans la section 8. Nous terminons ce chapitre avec une présentation de quelques domaines d'applications et une conclusion dans respectivement les sections 9 et 10.

2. Définitions

2.1. IoT

Selon l'union internationale des télécommunications, l'Internet des objets (IoT) désigne une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication [5].

Selon les auteurs de [4], l'internet des objets désigne des objets ayant des identités virtuelles opérant dans des espaces intelligents utilisant des interfaces intelligentes pour se connecter et communiquer.

D'un point de vue conceptuel, l'Internet des objets affecte, à chaque objet une identification unique sous forme d'une étiquette lisible par des dispositifs mobiles sans fil, ceci afin que chaque objet puisse communiquer. Elle permet alors de créer une passerelle entre le monde physique et le monde virtuel [12].

D'un point de vue technique, l'IoT consiste à l'identification numérique directe et normalisée (adresse IP, protocole http, etc.) d'un objet physique grâce à un système de communication sans fil (puce RFID, Bluetooth ou Wifi) [12].

Notre étude de ce domaine nous a fait comprendre que l'internet des objets est l'interconnexion entre l'internet et les objets, les lieux et les environnements physiques grâce aux technologies de la communication afin de rassembler des données et avoir de nouvelles connaissances et visions sur le monde extérieur.

2.2. Objet connecté

Un objet connecté est un objet physique qui peut être relié à un capteur ou un actionneur. Il est équipé d'une puce qui lui permet de transcender son usage initial pour proposer de nouveaux services. L'objet connecté est un matériel électronique capable de communiquer avec un ordinateur, un smartphone ou une tablette via un réseau sans fil tels que : Wi-Fi, Bluetooth, réseaux de téléphonie mobile, réseau radio à longue portée de type Sigfox ou LoRa, etc., qui le relie à Internet ou à un réseau local [13].

3. Les couches d'un modèle IoT

Aujourd'hui il n'existe pas encore une architecture standardisée et spécifique pour l'IoT. Parmi les architectures proposées dans la littérature, nous trouvons l'architecture en quatre couches présentées dans la Figure 1. Les couches sont : la couche de perception, la couche réseau, la couche middleware et la couche d'application [6]. Chaque couche utilise des composants et des technologies spécifiques afin d'assurer des fonctionnalités prédéterminées.

La Table 1 récapitule les fonctionnalités, les composants matériels et logiciels des différentes couches. Chaque couche sera détaillée séparément dans les sections suivantes.

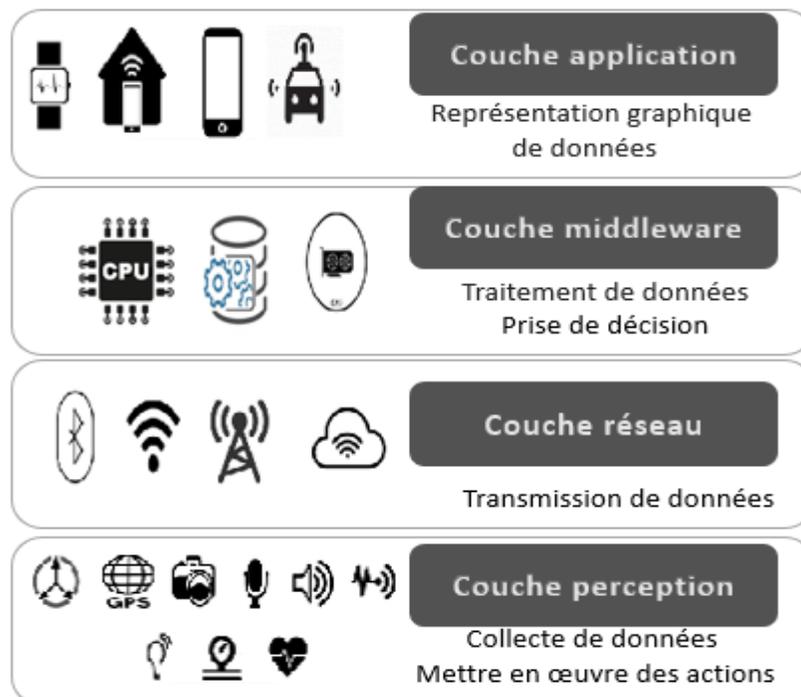


Figure 1 : Aperçu sur l'architecture IoT en 4 couches

	Fonctionnalités	Composants matériels	Composants logiciels
Couche perception	<ul style="list-style-type: none"> - Collecter les informations - Déclencher des actions 	<ul style="list-style-type: none"> - Capteurs/actionneurs - Cartes électroniques - WSN (Wireless Sensor Network) - Puce RFID (Radio Frequency IDentification) - Etc. 	
Couche réseau	<ul style="list-style-type: none"> - Transférer les données 	<ul style="list-style-type: none"> - Station de base 	<ul style="list-style-type: none"> - Réseau de communication - 3G, 4G, 5G. - Bluetooth - Zigbee - Etc.
Couche middleware	<ul style="list-style-type: none"> - Traiter les données - Prendre des décisions selon ces données 	<ul style="list-style-type: none"> - Serveur 	<ul style="list-style-type: none"> - Cloud - Big Data - BDD
Couche application	<ul style="list-style-type: none"> - Afficher les données 	<ul style="list-style-type: none"> - Dépend du domaine d'application (Mobile, web, etc.) [5] 	

Table 1 : Architecture IoT en 4 couches : fonctionnalités et composants

4. La couche perception

La couche perception est la couche responsable de la collection des données à partir de l'environnement physique telles que la température, l'humidité, etc., par divers dispositifs de détection. Cette couche est aussi responsable d'appliquer les décisions prises après le traitement de données en actionnant des objets physiques. Les objets de cette couche peuvent avoir des capacités de détection et/ou d'actionnement.

Les composants principaux de la couche perception sont:

4.1. Les capteurs

Les capteurs détectent les propriétés physiques et convertissent ces propriétés en signaux numériques. Ci-dessous quelques exemples de capteurs :

- **Capteur de mouvement PIR (PIR sensor)** est un capteur qui permet de détecter des mouvements dans un rayon de 5 mètres grâce aux radiations infrarouges (voir Figure 2 (a)).
- **Capteur de pulsations (pulse sensor)** est un capteur qui permet de détecter les pulsations du cœur lorsqu'il est placé au bon endroit sur le corps (voir Figure 2 (b)).
- **Capteur de flamme (flame sensor)** est un capteur sensible aux rayons infrarouges de 750 à 1100 nm, ce qui lui permet de détecter une flamme devant lui (voir Figure 2 (c)).
- **Capteur de température DHT11** [14] est capteur constitué d'un senseur d'humidité résistif, complété par une thermistance à coefficient négatif de température. Un microcontrôleur intégré s'occupe de faire les mesures, de les convertir en pourcentage d'humidité et de température (voir Figure 2 (d)).
- **Capteur de distance Hc-sr04** [15] est un capteur qui utilise les ultrasons pour déterminer la distance d'un objet. Il offre une plage de détection sans contact. Il fonctionne en mesurant le temps de retour d'une onde sonore, inaudible par l'homme, émise par le capteur. Il contient deux parties essentielles :
 - L'émetteur : émet un son à une fréquence définie (généralement autour de 40kHz)
 - Le récepteur : collecte le son répercuté par les obstacles

La distance aux objets est calculée par le temps mis par le son pour revenir au récepteur (voir Figure 2 (e)).

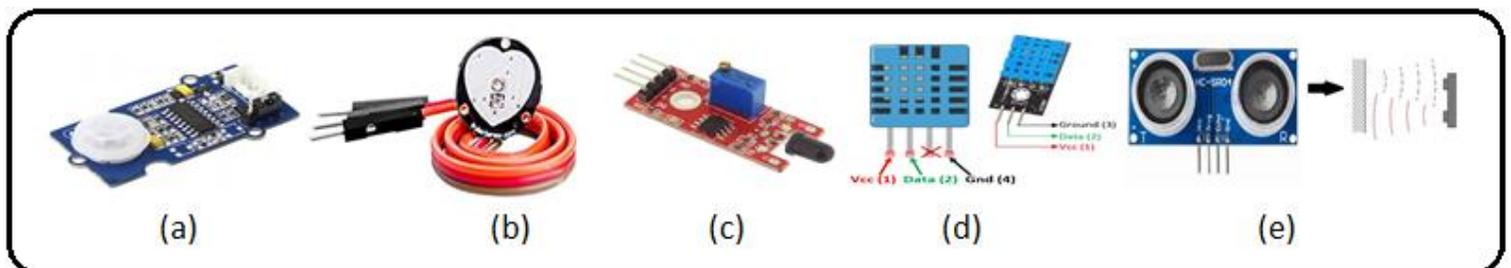


Figure 2 : Exemples de capteurs

4.2. Les actionneurs

Les actionneurs reçoivent des commandes pour effectuer des actions sur des objets physiques tels que des portes et des rideaux électriques à des moments spécifiques. Ci-dessous quelques exemples d'actionneurs :

- **Ventilateur** est un actionneur qui permet de faire tourner une hélice. Il est utilisé généralement pour refroidir un système (voir Figure 3 (a)).
- **Moteur pas à pas** est un actionneur qui permet de faire tourner pas à pas un mécanisme. Il est utilisé dans les imprimantes, scanners, disque dur d'un ordinateur, etc. (voir Figure 3 (b)).
- **Electrovanne** est un actionneur qui permet de contrôler le débit d'un liquide. Il est utilisé dans un système d'arrosage automatique (voir Figure 3 (c)).
- **Voyant** est un actionneur qui permet de produire un signal lumineux. Il est utilisé pour avertir l'utilisateur de l'état de fonctionnement d'une machine (voir Figure 3 (d)).
- **Afficheur LCD** est un actionneur possédant une interface visuelle entre le système et l'utilisateur. Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il affiche les données susceptibles d'être exploiter par l'utilisateur d'un système. Il est utilisé généralement dans les radios réveils, les montres, les calculatrices, etc. (voir Figure 3 (e)).



Figure 3 : Exemples d'actionneurs

4.3. Les cartes électroniques

Une carte électronique est une interface programmable qui sert à effectuer des tâches diverses. Ces cartes électroniques de petite taille sont équipées de microcontrôleurs (μC), et des entrée/sortie pour les communications sans fil. Comme elles peuvent être reliées avec des capteurs et/ou des actionneurs (voir Figure 4).

- **Les microcontrôleurs (μC)** : sont des très petits ordinateurs qui contiennent une unité de traitement programmable et des broches d'entrée / sortie.
- **Radio Tranceiver** : est un émetteur-récepteur qui permet de transmettre et recevoir les données. Il existe plusieurs types d'émetteur-récepteur, avec des protocoles de communication spécifiques.

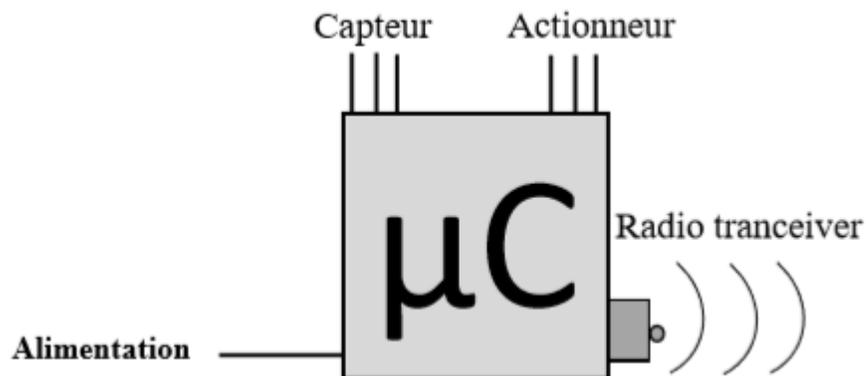


Figure 4 : Les composants d'une carte électronique

Ci-dessous quelques exemples des cartes électroniques.

- **Le module nRF24L01** [16] est un émetteur-récepteur adapté aux applications sans fil à faible puissance. Il utilise le protocole Enhanced ShockBurst (ESB) qui supporte la communication série et qui garantit une transmission fiable. Ce module fonctionne dans la bande des fréquences [2.400 - 2,4835] GHz qui inclut le Wifi ou le Bluetooth, par contre il n'est pas compatible avec les deux (voir Figure 5 (a)).

- **Le module Wi-Fi ESP8266** [8] est un module qui sert à faire la liaison avec un réseau Wi-Fi disponible. Il contient un microcontrôleur interne qui donne l'accès sans fil aux réseaux. L'ESP8266 est un système-sur-puce (Soc) qui fonctionne dans la bande des fréquences de 2,4 GHz. Ce module utilise le protocole de communication WIFI 802.11 b/g/n, qui permet au microcontrôleur interne de se connecter à un réseau Wi-Fi et de faire des connexions TCP/IP simples, ou d'être détectable par d'autres périphériques qui utilisent le même protocole WIFI (voir Figure 5 (b)).

- **La carte Arduino** est un microcontrôleur programmable qui permet de contrôler des éléments mécaniques, tels que : les systèmes d'alarmes, les lumières, les moteurs, etc. Cette carte électronique permet donc à son utilisateur de rendre intelligent des systèmes électroniques et de créer des mécanismes automatisés, sans avoir de connaissances particulières en programmation. L'intérêt principal des cartes ARDUINO est leur facilité de charger le programme compilé dans la mémoire du microcontrôleur simplement via un port USB. Cela à partir d'un environnement de développement (Arduino IDE), s'appuyant sur des outils open-source.

Le modèle UNO [17] de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. L'ATmega328 est un microcontrôleur 8 bits de la famille AVR dont la programmation peut être réalisée en langage C/C++. (Voir Figure 5 (c)).

Le modèle NANO [18] possède les mêmes caractéristiques qu'UNO sauf qu'il peut avoir aussi l'ATmega168 comme cœur (voir Figure 5 (d)).

- **La carte NodeMCU** [9] est un micro logiciel à source ouverte et un kit de développement. Il est basé sur le module Wifi ESP8266 cadencé à 80 MHz et exécutant le firmware open source NodeMCU. Cette carte se programme via l'IDE Arduino et est compatible avec les scripts LUA. Ce microcontrôleur dispose d'une interface Wifi idéale pour les objets connectés. Il est équipé de connecteurs latéraux mâles et femelles permettant de relier le module avec une plaque de montage rapide. L'interface sans fil Wifi permet la création des points d'accès sans fil, la connexion à internet et le partage des données (voir Figure 5 (e)).

- **Raspberry Pi** [11] est un petit ordinateur mono carte. En lui connectant des périphériques tels qu'un clavier, une souris et un écran, il agira comme un mini-ordinateur personnel. Raspberry Pi est couramment utilisé pour le traitement d'images et de vidéos en temps réel, les applications basées sur l'IoT et les applications robotiques. Le Raspberry Pi est moins performant qu'un ordinateur portable ou de bureau, mais il peut fournir toutes les fonctionnalités ou capacités attendues, à une faible consommation d'énergie (voir Figure 5 (f)).

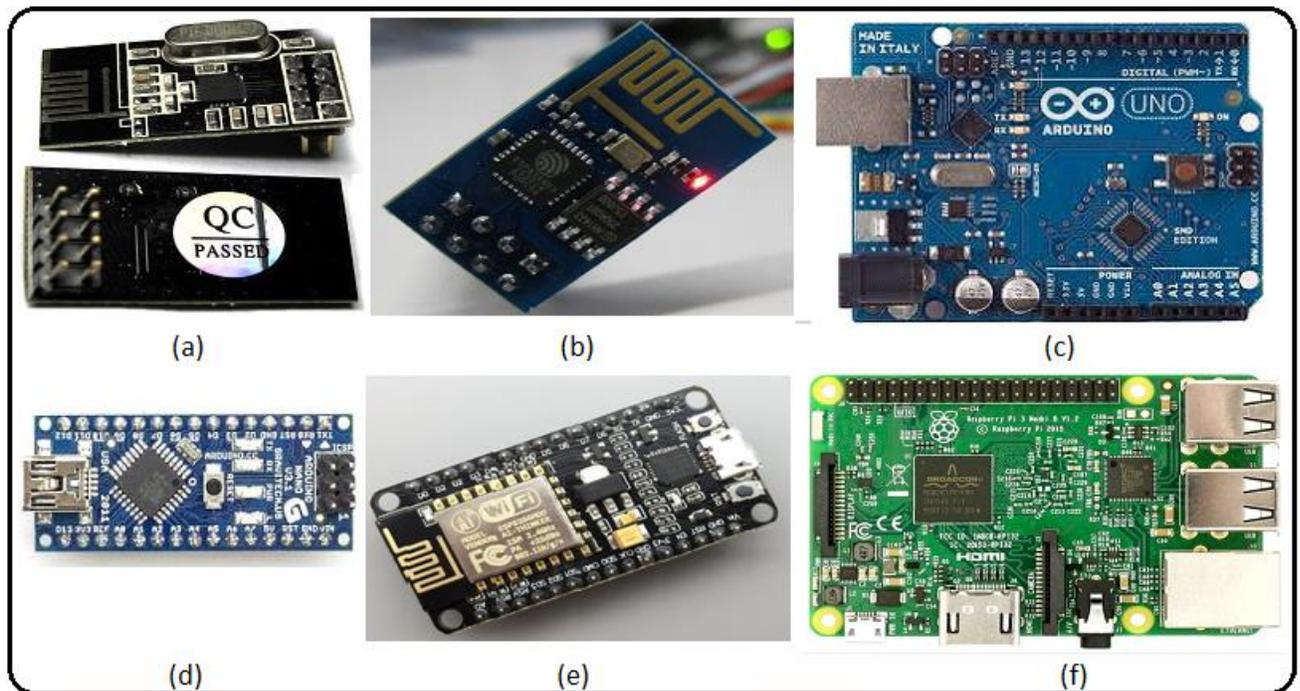


Figure 5 : Exemples de cartes électroniques

5. La couche réseau

La couche réseau est la couche responsable de la transmission des données reçues de la couche perception à une base de données et un serveur qui se trouvent sur la couche middleware. Les principales technologies utilisées pour réaliser cette opération sont : les technologies cellulaires et les réseaux LAN. Cette couche donne une valeur aux données captées et une identité aux objets connectés dans l'infrastructure.

Les principaux composants de la couche réseau sont :

- Les réseaux de communication utilisés pour les objets connectés.
- Les passerelles IoT pour contrôler le passage des informations entre les objets connectés et internet.

5.1. Les réseaux de communication

Les réseaux de communication permettent aux appareils intelligents de faire transiter leurs données. Il existe des dizaines de technologies différentes, qui ont chacune leurs particularités en termes de consommation d'énergie, de portée, et de prix. Pour choisir le réseau de communication le plus adapté pour relier les objets connectés au Net, il faut déterminer si ses objets connectés seront ou non situés loin du portail de réception de leurs données (voir Figure 6). On distingue deux grandes catégories de réseaux qui existent sur le marché : les réseaux longue portée et les réseaux courte portée.

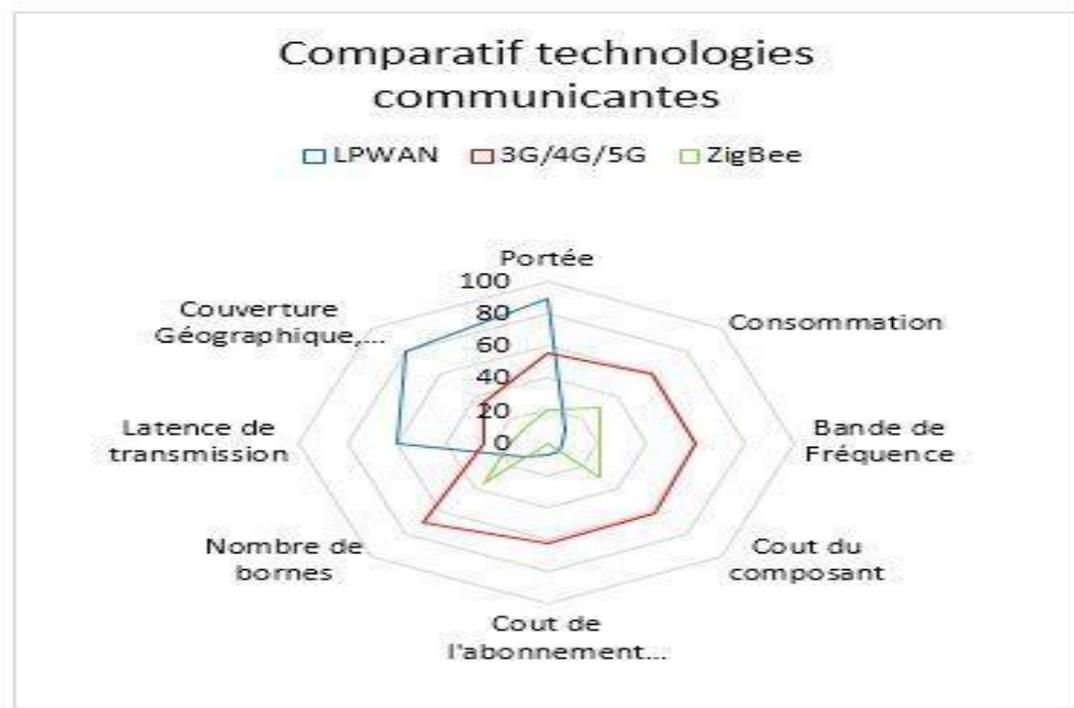


Figure 6 : Comparaison des technologies de communication

5.1.1. Réseau longue portée

Les réseaux longue portée sont capables de faire transiter des données d'un appareil à un autre sur de vastes distances comme les réseaux LPWAN tels que : Sigfox et LoRa ou encore les réseaux cellulaires tels que : GSM, 2G, 3G, etc. qui sont utilisés par les entreprises pour interconnecter des kilomètres d'infrastructures à Internet ou dans des projets de smart [19].

- Les réseaux LPWAN (voir Figure 7) ont une portée qui va de 5 à 40 km et une faible consommation énergétique. Ces réseaux se chargent d'émettre ou de recevoir des paquets vers les différents équipements IoT se trouvant à proximité à un faible coût. Ils sont constitués d'antennes passerelles qui permettent de faire la correspondance entre le monde Hertzien et le monde d'Internet.

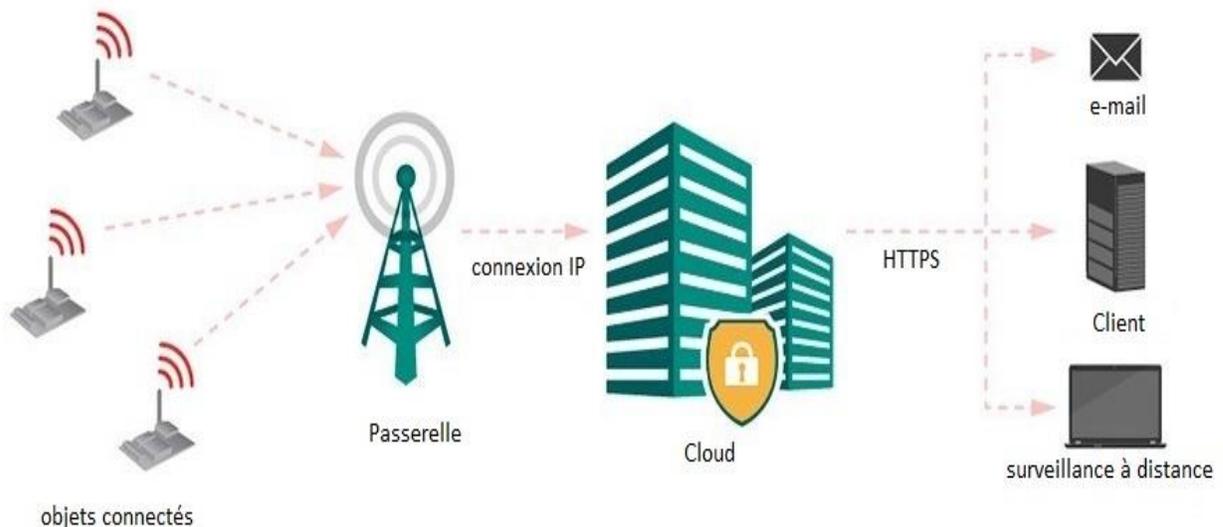


Figure 7 : Architecture d'un réseau LPWAN

- **Sigfox** a une portée qui peut être supérieure à 10 km et qui peut atteindre les 30 à 50 km à la campagne. Les objets connectés ne consomment que très peu d'énergie pour envoyer leurs données sur ce réseau. Par contre, il ne permet de transporter que de très faibles quantités de données, entre 10 et 100 bits par seconde (bps) [3] [19].
- **LoRa** est un réseau qui permet de transmettre des données à des distances de 2 à 5 kilomètres en milieu urbain, et jusqu'à 45 kilomètres en milieu rural avec une très basse

consommation d'énergie. Seuls 0,3 à 50 kilobits par seconde (kbps) peuvent transiter sur ce réseau. Comme Sigfox, il est idéal pour des capteurs émettant périodiquement une faible quantité de données tels que : la température, la géo-localisation, ou la pression [3] [19].

- Les réseaux cellulaires (GSM, 2G, 3G, 4G) sont fournis par les opérateurs télécoms traditionnels. Ils sont très gourmands en énergie et permettent de transférer d'importantes quantités de données. Par ailleurs, il faut prévoir un espace sur l'objet connecté pour insérer une carte SIM [3] [19].

5.1.2. Réseau courte portée

Les réseaux courte portée permettent de transférer des données sur de courtes distances comme les réseaux LAN tel que Wifi, le Z-Wave, le Zigbee, ou encore le Bluetooth. Ils sont beaucoup utilisés dans la domotique ou sur le marché des appareils intelligents [19].

- Les réseaux LAN (Local Area Network) ont une courte portée (entre 1 m et 100 m) (voir Figure 8) et sont peu consommateurs d'énergie. Ces réseaux sont très utilisés par les objets connectés du grand public. Par exemple les box domotique pour le smart home ou les bracelets connectés pour le smart health.



Figure 8 : Réseaux de communication courte portée

- **Wifi** a une portée de 100 mètres en terrain dégagé. Il est très énergivore et permet de transférer un grand nombre de données rapidement, jusqu'à 600 mbps. Il est bidirectionnel, ce qui permet de mettre facilement à jour les appareils [19].
- **Bluetooth** a une portée moyenne de 10 mètres et consomme environ 20 fois moins d'énergie que le Wifi. Ce réseau permet de transporter 1 Mbps seulement [19].
- **Z-Wave** est un réseau de communication dédié à la domotique qui peut faire circuler des données jusqu'à 100 kbps. Il est sans fil et facile à installer dans la maison. Il a une portée de base de 30 mètres. C'est un réseau maillé, où chaque appareil connecté au système est émetteur de données mais peut aussi relayer celles qui sont émises par ses voisins. Cela permet d'élargir sa portée [19].
- **Zigbee** permet de faire circuler jusqu'à 250 kbps de données. Il a une portée moyenne de 10 mètres. Il est également plus facile à implémenter pour les fabricants d'objets connectés que le Z-Wave ou le Bluetooth [19].
- **RFID** (Radio Frequency IDentification) est un système de radio-identification, activé par un transfert d'énergie électromagnétique qui se compose d'un émetteur « puce de géo localisation radio » et d'un lecteur RFID. Lorsque l'émetteur est dans la zone de détection du lecteur, la communication entre la puce de géo localisation et le lecteur peut être activée à distance de manière automatique. Cette distance peut aller de 1 mètre jusqu'à 50 mètres [20].

5.2. Les passerelles IoT

Les passerelles jouent le rôle d'intermédiaire entre l'objet connecté et l'internet. Elles permettent à un objet de se connecter et envoyer ses données vers internet (voir Figure 9). Ces passerelles fournissent ce qui est nécessaire en termes de connectivité, de sécurité et de management des appareils. Des exemples de ces passerelles sont les routeurs domestiques et les téléphones mobiles [21].

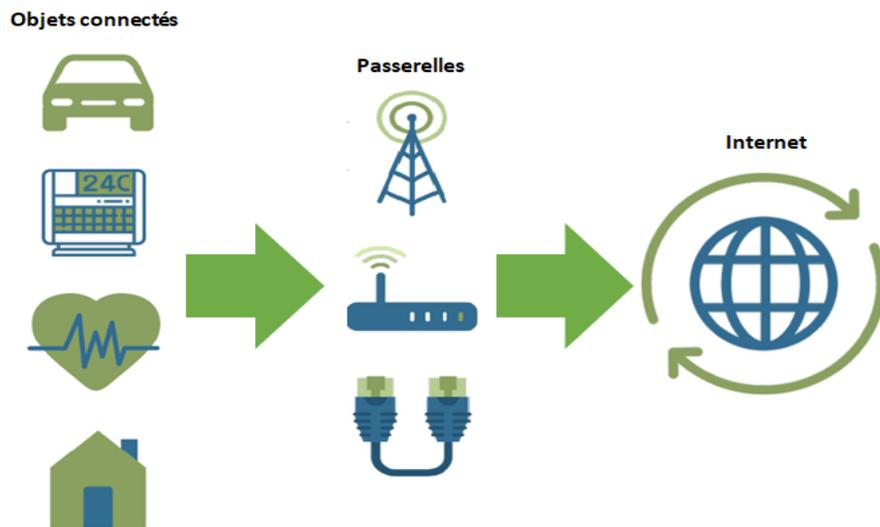


Figure 9 : Aperçu sur les passerelles IoT

6. La couche middleware

La couche middleware est la couche responsable de stockage et de traitement des données. Elle contient le serveur, la base de données et tout ce qui est lié à l'analyse et la prise de décisions. Cette couche reçoit les données des deux couches « réseau » et « application » pour faire son traitement et prendre des décisions. Des données ou des décisions peuvent éventuellement être transmises à la couche application pour notifier l'utilisateur ou perception (pour déclencher une action). Parmi les composants de la couche middleware on trouve la base de données, le Cloud et le Big Data.

6.1. Base de données

Une base de données est un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données) [22]. Le choix d'une base de données efficace pour les applications IoT pourrait être très difficile car l'environnement IoT diffère d'un domaine d'application à un autre. Ce choix doit inclure de nombreux facteurs, les plus importants sont : l'évolutivité, la capacité à gérer d'énormes quantités de données à des vitesses adéquates, un schéma flexible, la portabilité avec des outils analytiques variés, la sécurité et les coûts.

6.2. Cloud Computing

Le Cloud Computing est un terme général employé pour désigner la livraison de ressources et de services à la demande par internet. Une fois dans le Cloud, les données peuvent être protégées en appliquant une sécurité de haut niveau, et peuvent être directement accédées et visualisées depuis n'importe quel endroit. Il permet la collecte et le traitement des données, tout en maintenant de faibles coûts de traitement de données complexes [1].

6.3. Big Data

Le terme de Big Data ou méga données, désigne une large quantité de données structurées ou non structurées. A mesure que le nombre d'objets connectés augmente, le volume de données générées explose. C'est là que les Big Data interviennent en nous offrant la possibilité de prendre en charge de gros volume de données et les analyser en temps réel afin d'extraire des informations exploitables [2].

7. La couche application

La couche application fournit des services intelligents de haute qualité pour répondre aux besoins des clients. L'utilisateur peut surveiller les données capturées et peut également interagir avec le système pour effectuer des réglages.

Le composant principal de cette couche est une application logicielle pour smartphones, tablettes ou ordinateurs de bureau qui fournissent des interfaces graphiques (GUI) ou lignes de

commandes pour la surveillance et le contrôle des objets connectés. Parmi ces applications on trouve les applications bureau, les applications Web et les applications mobiles.

7.1. Application de bureau

Une application bureau est un logiciel applicatif qui affiche son interface graphique dans un environnement de bureau et doit être installé sur l'ordinateur de son utilisateur, afin de réaliser une ou plusieurs tâches selon les besoins.

7.2. Application WEB

Une application web est un logiciel applicatif, hébergée sur un serveur et accessible via les navigateurs Internet tels que : Internet Explorer, Mozilla Firefox et Google Chrome. Contrairement à un logiciel de bureau, les applications web n'ont pas besoin d'être installées. Il suffit en effet de se connecter à un navigateur pour pouvoir y accéder [23].

7.3. Application Mobile

Une application mobile est un logiciel applicatif développé pour un smartphone ou une tablette. Les applications mobiles peuvent être installées soit sur un système Android soit sur un système iOS. Les applications mobiles permettent généralement un accès plus confortable à des sites ou services accessibles en version mobile [24].

8. Protocoles d'application pour l'IoT

L'échange de données entre les différentes couches est l'un des principaux aspects de l'IoT. Il existe de nombreux protocoles disponibles pour permettre cette communication. Certains protocoles sont conçus en tenant compte des exigences spécifiques de l'IoT. Dans ce qui suit nous explorons les protocoles les plus utilisés dans les applications IoT, à savoir : MQTT, AMQP, COAP et XMPP [27].

8.1. MQTT (Message Queuing Telemetry Transport)

MQTT ou Transport de télémétrie Message Queuing est un protocole de messagerie léger basé sur le protocole TCP/IP, et sur le concept de publication-abonnement « publish/subscribe ».

Ce concept permet aux clients de s'abonner aux différents topics (canaux), afin d'être notifié une fois d'autres clients publient des données sur ces topics (voir Figure 10). Ce protocole est le mieux adapté aux communications d'internet des objets (IoT) en raison de son faible encombrement de code, de ses exigences en matière de bande passante et de sa consommation d'énergie sur l'appareil.

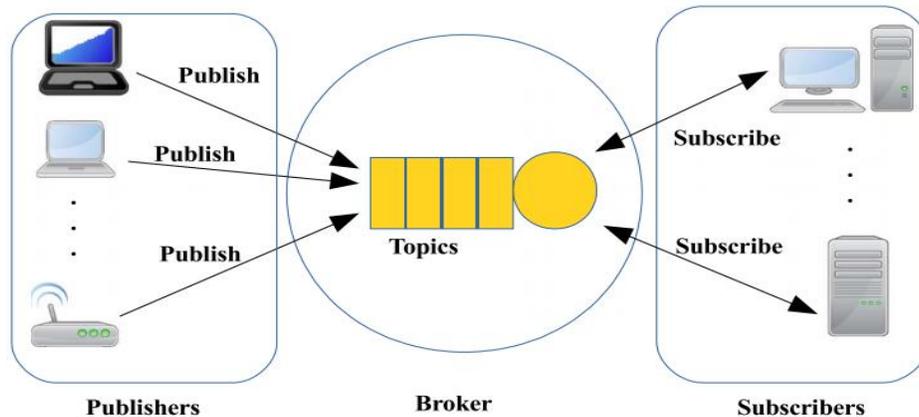


Figure 10 : Principe de fonctionnement de MQTT

8.2. AMQP (Advanced Message Queuing Protocol)

AMQP ou Protocole avancé de Message Queuing, comme le protocole MQTT, il se base sur le protocole TCP/IP. Avec ses modèles de messagerie flexibles, il permet l'échange de messages qui ne peut pas être effectué avec une faible bande passante (voir Figure 11).

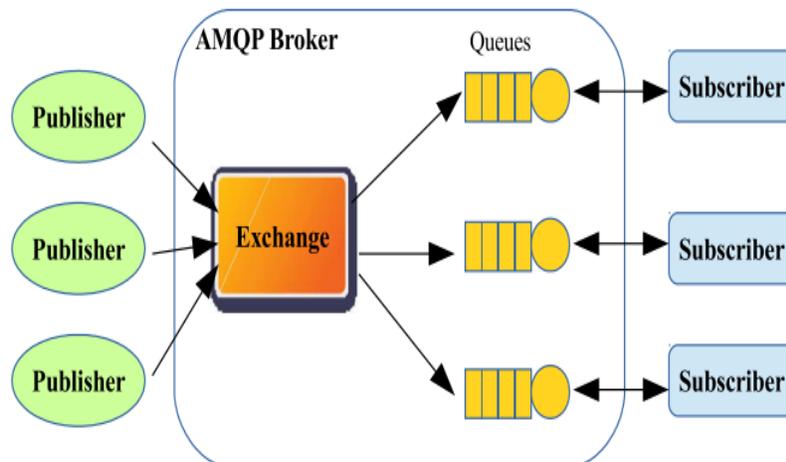


Figure 11 : Principe de fonctionnement d'AMQP

8.3. CoAP (Constrained Application Protocol)

CoAP ou Protocole d'application contrainte est un protocole de transfert Web qui s'exécute sur UDP (pas de surcharge TCP) pour maintenir la légèreté de l'implémentation globale. Ce protocole est basé sur l'architecture demande/réponse du HTTP (voir Figure 12).

Il utilise alors des commandes comme GET, POST, PUT et DELETE pour fournir des interactions orientées ressources.

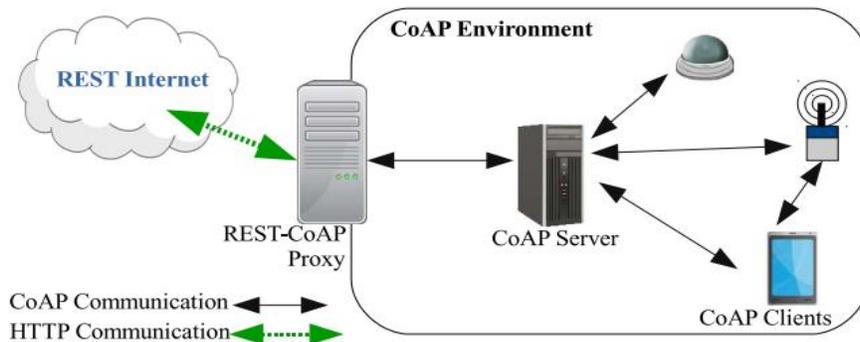


Figure 12 : Principe de fonctionnement de COAP

8.4. XMPP (Extensible Messaging and Presence Protocol)

XMPP est un protocole de messagerie et de présence extensible basé sur les deux concepts : publication/abonnement et demande/réponse (voir Figure 13). Il fonctionne avec TCP et permet les échanges décentralisés de messages, c'est-à-dire qu'il est composé de plusieurs serveurs reliés entre eux qui permettent la communication entre les clients. Il est largement utilisé pour la messagerie instantanée et les appels vocaux/vidéo.

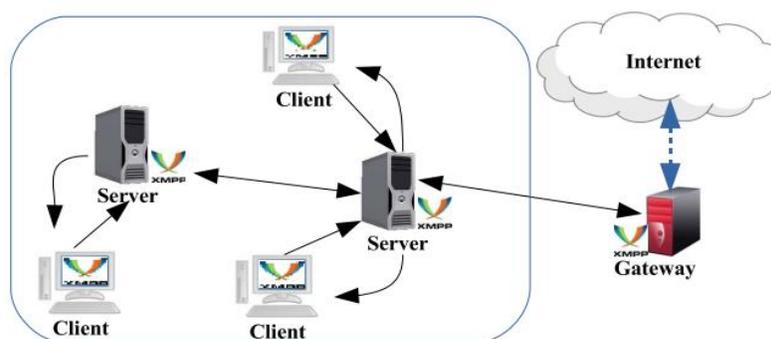


Figure 13 : Principe de fonctionnement de XMPP

9. Domaines d'applications de l'IoT

L'IoT permet le développement de plusieurs applications intelligentes qui touche essentiellement à la domotique, les villes, le transport, la santé et l'industrie. Dans ce qui suit nous citons quelques exemples de domaine d'applications de l'IoT.

9.1. La domotique

La domotique regroupe l'ensemble des technologies permettant l'automatisation des équipements d'un habitat. Elle vise à apporter du confort à savoir des systèmes IoT de commande à distance, de la sécurité par exemple des systèmes IoT d'alarmes et de la gestion d'énergie comme l'optimisation de l'éclairage. [10].

L'application de l'IoT à la domotique consiste à mettre les dispositifs domestiques accessibles à travers le réseau. Cela permet de contrôler les différents équipements d'une maison depuis une même interface sur une tablette ou un téléphone.

9.2. Le transport

Depuis la création de l'IoT, le nombre de véhicules intelligents est en croissance. Presque tous les véhicules vendus aujourd'hui dans le monde renferment des capteurs et des moyens de communication pour traiter la congestion du trafic, la sécurité, la pollution et le transport efficace des marchandises. L'IoT peut sauver des vies, réduire le trafic, minimiser l'impact des véhicules sur l'environnement et renforcer la sécurité routière [7].

9.3. La santé

Dans le domaine de la santé l'IoT peut assurer le suivi des signes cliniques des patients par la mise en place des réseaux personnels de surveillance. Ces réseaux sont constitués de biocapteurs posés sur les corps des patients ou dans leurs lieux d'hospitalisation. Cela facilitera la télésurveillance des patients et apportera des solutions pour l'autonomie des personnes à mobilité réduite [7].

9.4. L'industrie

La technologie IoT permet aux usines d'améliorer l'efficacité des opérations par la mise en place des réseaux de suivi. Ces réseaux permettent d'optimiser la production, d'améliorer la sécurité des employés, de faciliter la lutte contre la contrefaçon, la fraude et d'assurer un suivi total des produits. Par exemple l'IoT a permis chez Amazon de rendre la marchandise intelligente et les entrepôts entièrement automatisés [7].

10. Conclusion

Dans ce chapitre nous avons décrit l'architecture en quatre couches du concept d'internet des objets. Nous avons défini l'IoT, l'objet connecté et présenté les différentes couches de cette architecture. Ensuite nous avons décrit quelques protocoles d'applications utilisés dans des projets IoT. Enfin nous avons présenté quelques domaines d'utilisations de l'IoT.

Dans le chapitre suivant, nous présentons la mise en place de notre propre infrastructure qui inclut les quatre couches de l'architecture IoT présentée dans ce chapitre.

Chapitre 2 :

Mise en place de

l'infrastructure IoT

1. Introduction

Dans ce chapitre, nous décrivons les technologies que nous avons utilisé pour réaliser notre infrastructure IoT qui inclut les quatre couches : perception, réseau, middleware et application. La Figure 14 représente un aperçu sur les différentes technologies que nous avons utilisées pour chaque couche, ainsi que les différentes opérations effectuées entre les couches. La couche perception permet de capter des données et/ou exécuter des actions. La couche réseau est responsable de la transmission des données. La couche middleware stocke et traite les données captées. Enfin la couche application offre à l'utilisateur la possibilité de surveiller les données et d'interagir avec le système IoT. L'opération « transmettre » désigne la transmission des données entre les différentes couches. Les opérations « capter », « publier », « collecter » et « stocker » expliquent comment les données sont acheminées de la couche perception vers la couche middleware. Les opérations « surveiller » et « interagir » permettent à l'utilisateur de visionner les données et ajuster les réglages selon ses besoins. Les opérations « traiter » et « déclencher » indiquent le traitement des données pour prendre des décisions, et l'envoi de cette décision à la couche perception via la couche réseau. Enfin l'opération « actionner » indique l'application de la décision et éventuellement l'exécution d'une action.

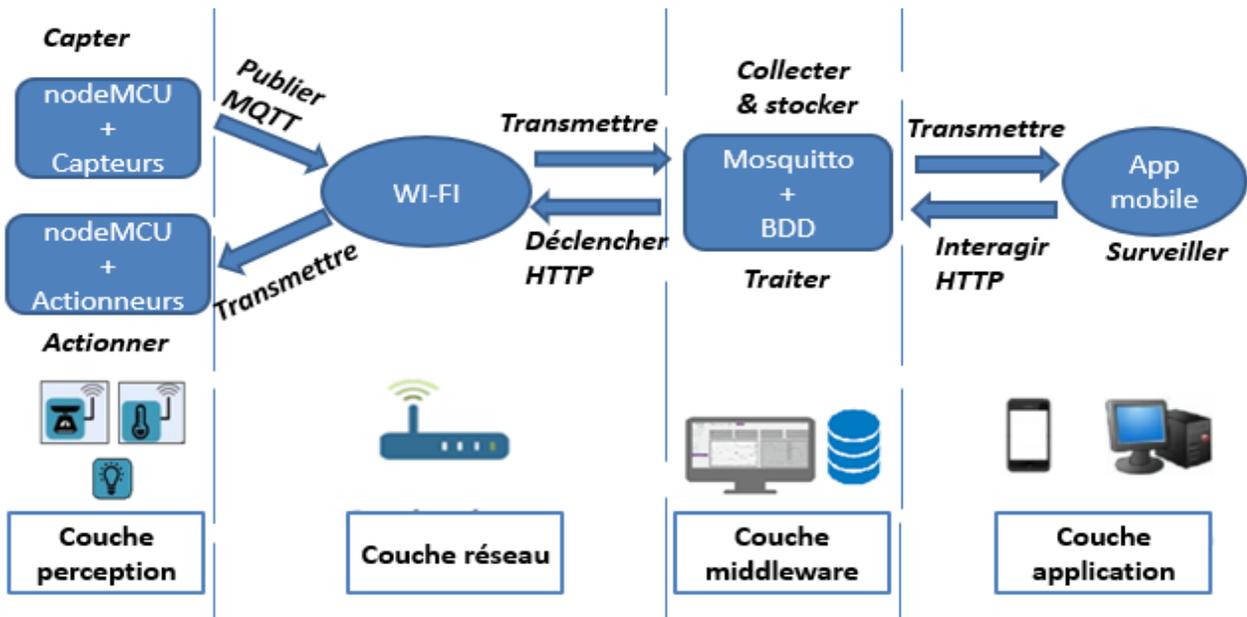


Figure 14 : Aperçu sur l'infrastructure mise en place

2. La couche perception

Nous avons utilisé deux cartes électroniques de type nodeMCU, parce que c'est les seules cartes électroniques équipées d'un module wifi intégré que nous avons trouvés disponibles sur le marché. Nous avons relié la première carte aux capteurs pour connecter un objet et capter des données à transmettre à la couche middleware. Le capteur est la partie responsable de transformer une grandeur physique analogique en un signal numérique, appelé donnée. Cette dernière sera ensuite publiée dans un broker local installé sur un serveur de la couche middleware. Pour assurer cet acheminement de données, nous avons utilisé le protocole MQTT qui est basé sur le modèle publish/subscribe.

Nous avons relié la deuxième carte aux actionneurs afin d'exécuter les événements reçus de la couche middleware. L'actionneur reçoit des événements de la couche middleware et déclenche des actions si nécessaire. Pour transmettre des requêtes depuis la couche middleware vers l'actionneur, nous avons utilisé le protocole HTTP.

Selon le besoin du système IoT qu'on veut mettre en place, il est possible d'utiliser plus ou moins de cartes électroniques.

3. La couche réseau

La couche réseau est responsable du transfert de données entre les différentes couches. Nous avons choisi d'utiliser un réseau de type LAN pour tester notre infrastructure car les exemples de systèmes IoT que nous avons réalisés ne nécessitent pas un réseau de longue portée.

4. La couche middleware

La couche middleware est responsable du stockage et du traitement des données. Nous avons utilisé des brokers en ligne tels que : AdaFruit IO et Google FireBase en passant par internet. Cependant ces services ne sont pas gratuits et limités en termes de quantités de données traitées. Voilà pourquoi nous avons développé notre propre broker appelé « Mosquito ». Ce dernier implémente le protocole MQTT et peut être installé sur une machine à faible consommation de ressources [25]. Nous avons renforcé la sécurité du broker en ajoutant la combinaison utilisateur/mot de passe pour pouvoir publier et/ou recevoir les données. Ce broker

va intercepter les données publiées à des topics spécifiques au système IoT pour les transmettre aux clients abonnés sans les stocker. Pour ne pas perdre les données, nous avons utilisé le protocole MQTT over Web Sockets car il est supporté par Mosquitto et assure le stockage de données. Avec ce protocole, les données sont transférées sur le réseau depuis Mosquitto vers un client JavaScript abonné aux topics et encapsulées par plusieurs trames Web Sockets (voir Figure 15). Une fois que le client reçoit les données du broker, il peut les stocker dans une base de données MySQL en utilisant la méthode POST du protocole HTTP [26].

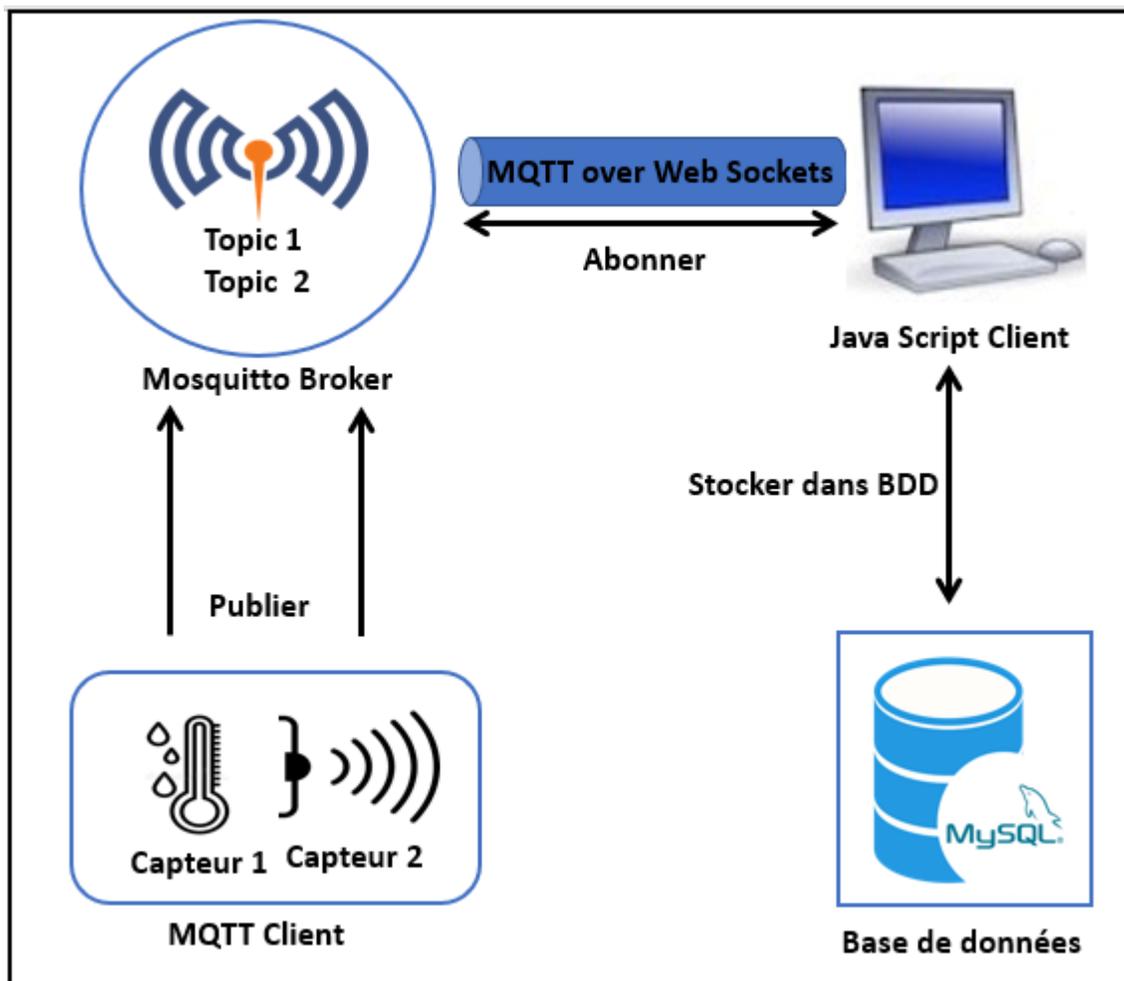


Figure 15 : Aperçu sur le traitement et le stockage des données

Nous avons utilisé le langage PHP pour traiter les données stockées, en comparant les réglages définis par l'utilisateur via la couche application avec les données captées dans la couche

perception. Cette comparaison permet de prendre une décision sur les évènements à déclencher, et la transmettre via le protocole HTTP à la couche perception.

La Figure 16 montre l'interface web graphique du broker que nous avons développé avec les langages « HTML, CSS et Bootstrap » pour consulter l'historique des données publiées. Ce qui permet à l'utilisateur d'avoir plus d'informations, et de s'en servir pour améliorer les services de son système.



Figure 16 : Aperçu de l'interface WEB du broker

5. La couche application

Dans la couche application, nous devons fournir à l'utilisateur une application qui lui permet de mieux interagir avec son système IoT. Pour notre infrastructure, Nous avons développé une application mobile qui affiche la dernière donnée captée, et offre à l'utilisateur la possibilité de faire des réglages selon ses choix et ses besoins.

6. Exemples de systèmes IoT

Dans cette section, nous décrivons quelques exemples de systèmes IoT qui peuvent être réalisés en utilisant l'infrastructure que nous avons mise en place.

6.1. Lumière intelligente

Avec un capteur de lumière, nous pouvons allumer des lampes d'extérieur connectées au système et situées aux alentours d'une maison. Une fois la nuit tombe, le niveau de lumière captée se baisse au-dessous de celui que nous avons réglé, alors les lampes s'allument.

6.2. Alarme fumée intelligente

En utilisant un capteur de fumée, l'utilisateur sera alerté sur son smartphone si le degré de fumée captée dépasse un seuil défini. Comme il peut interagir avec le système pour actionner le système anti-feu même s'il se trouve loin de l'incident. Le système anti-feu doit être connecté à internet via le cloud en utilisant un réseau de communication longue distance tel que « 4G ».

7. Conclusion

Dans ce chapitre nous avons présenté notre première contribution, qui consiste à la mise en place d'une infrastructure IoT qui inclut les quatre couches. Nous avons expliqué le fonctionnement de notre infrastructure, et nous avons présenté le choix des technologies et protocoles utilisées dans chaque couche. A la fin, nous avons cité quelques exemples de systèmes IoT qui peuvent être réalisés grâce à notre infrastructure.

Dans le chapitre suivant, nous présentons deux exemples de systèmes IoT que nous avons réalisés en utilisant l'infrastructure mise en place.

Chapitre 3 :

Exemples de systèmes

IoT implémentés

1. Aperçu

Dans ce chapitre nous présentons le fonctionnement des systèmes IoT que nous avons développés pour tester l'infrastructure que nous avons mise en place. Nous modélisons le fonctionnement de nos systèmes IoT à l'aide d'un automate à état finis représenté dans la figure 17 où les transitions sont structurées sous la forme événement[condition]/action.

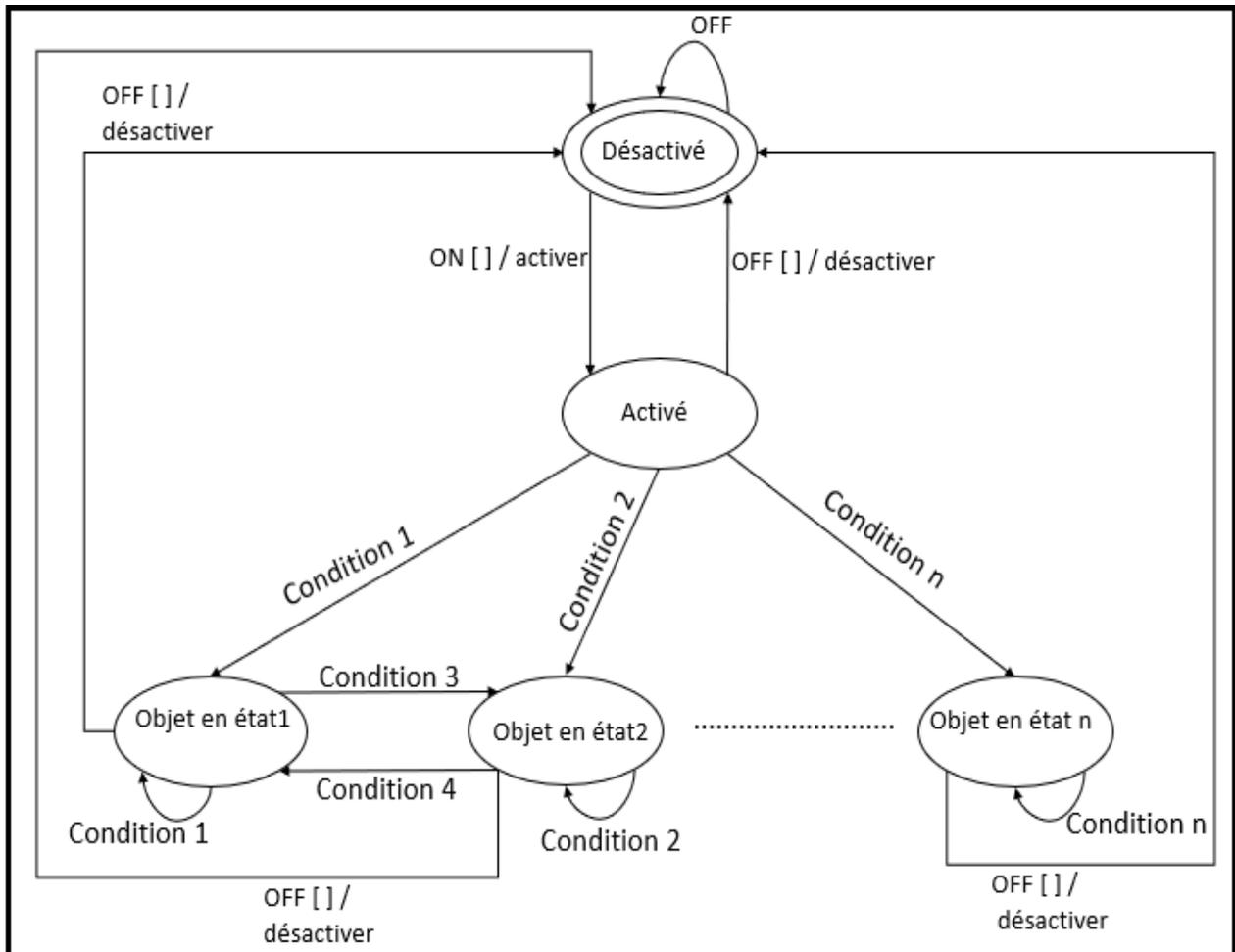


Figure 17 : Automate de fonctionnement de système IoT

Les états sont les suivants : « désactivé », « activé », « objet en état 1 », « objet en état 2 », ..., « objet en état n ». Sachant que le nombre d'états est $n+2$ et qu'un objet peut avoir au minimum deux états ($n \geq 2$).

Les états « désactivé » et « activé » désignent si notre système IoT est démarré ou pas. Lorsque le système IoT est « activé » l'utilisateur peut visualiser l'état de ses objets connectés et effectuer des réglages selon ses besoins.

Pour passer d'un état à un autre, la condition de la transition doit être vérifiée, sinon le système reste dans le même état. Nous pouvons arrêter carrément le système IoT sans vérifier les conditions en passant à l'état « désactivé ».

2. Système IoT de surveillance de distances

Nous surveillons les distances en utilisant un capteur de distance ultrason et une lampe comme actionneur. C'est un système qui peut être utilisé pour plusieurs exemples. Nous pouvons l'utiliser pour surveiller un enfant s'il se rapproche d'un endroit dangereux tels qu'une prise de courant ou des escaliers. La Figure 18 montre l'automate à état fini qui explique le fonctionnement de ce système.

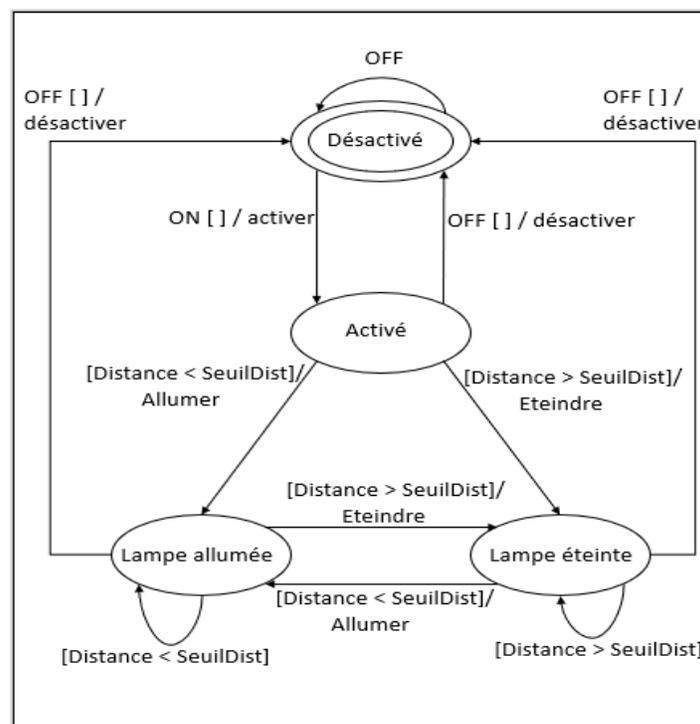
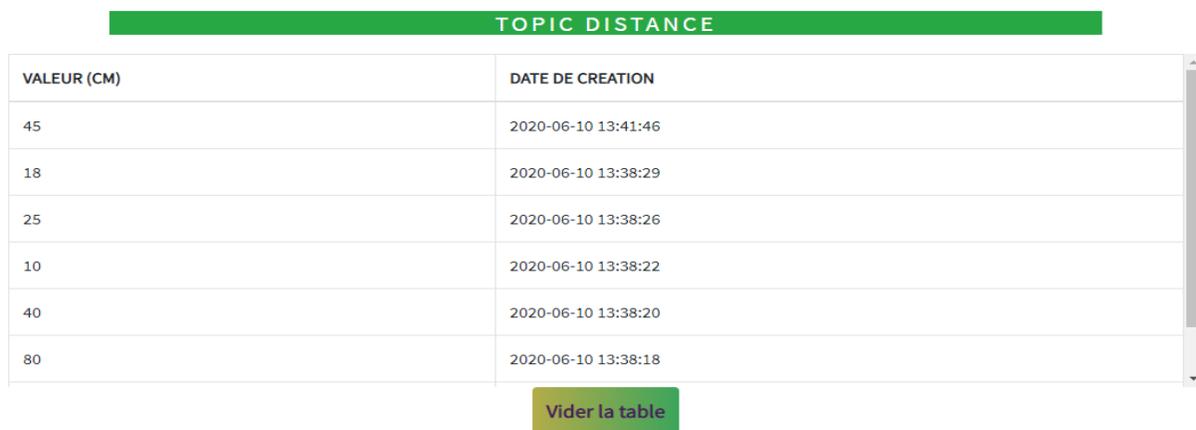


Figure 18 : Automate de fonctionnement de système IoT - surveillance de distances

L'état « désactivé » désigne que notre système IoT n'est pas démarré. Lorsqu'il passe à l'état « activé » le traitement commence. Si la distance captée est inférieure au seuil réglé par l'utilisateur et la lampe est éteinte, l'action « allumer » s'exécute et la lampe passe à l'état « lampe allumée ». Si la distance captée est supérieure au seuil, la lampe passe à l'état « lampe éteinte » si elle est allumée ou reste éteinte si elle l'est déjà. L'utilisateur peut aussi la forcer à s'éteindre s'il arrête l'application en revenant à l'état « désactivé ».

La figure 19 donne un aperçu sur l'interface WEB du broker que nous avons créé, nous pouvons voir l'historique des distances captées en temps réel.



TOPIC DISTANCE	
VALEUR (CM)	DATE DE CREATION
45	2020-06-10 13:41:46
18	2020-06-10 13:38:29
25	2020-06-10 13:38:26
10	2020-06-10 13:38:22
40	2020-06-10 13:38:20
80	2020-06-10 13:38:18

Vider la table

Figure 19 : Aperçu de l'interface WEB du broker - historique de distances

Nous avons développé une application mobile pour ce système. Lorsque l'utilisateur tourne l'application vers le ON, il commence à recevoir la dernière distance captée en permanence tous les deux seconds sur son mobile. L'application envoie une requête HTTP à la couche middleware pour signaler qu'elle est en état de ON. Cette requête contient aussi le seuil fixé par l'utilisateur. Une fois que la couche middleware reçoit la requête, elle commence son traitement qui consiste à comparer les valeurs captées avec le seuil réglé par l'utilisateur. Si la distance entre le capteur et l'objet le plus proche est inférieure au seuil, un événement « allumer » est envoyé à l'actionneur pour allumer la lampe, et l'utilisateur sera averti par des bips sur son application. Dans le cas contraire la lampe s'éteint et les bips s'arrêtent (voir Figure 20).

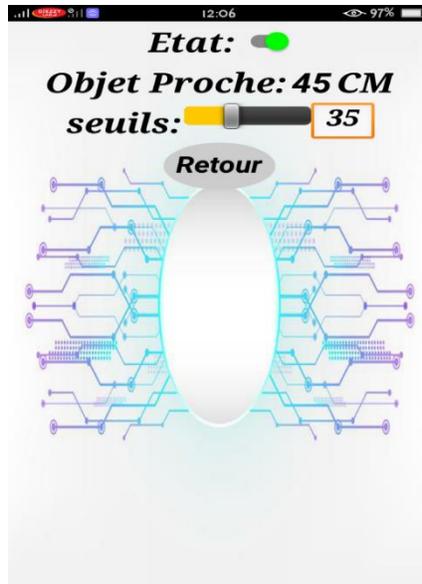


Figure 20 : Application mobile de système IoT de surveillance de distances

3. Système IoT de surveillance de températures

Nous surveillons les températures en utilisant un capteur de température DHT11 et un ventilateur comme actionneur. C'est un système qui peut être utilisé pour plusieurs exemples. Nous pouvons l'utiliser pour allumer le climatiseur de la maison automatiquement. La Figure 21 montre l'automate à état fini qui explique le fonctionnement de ce système.

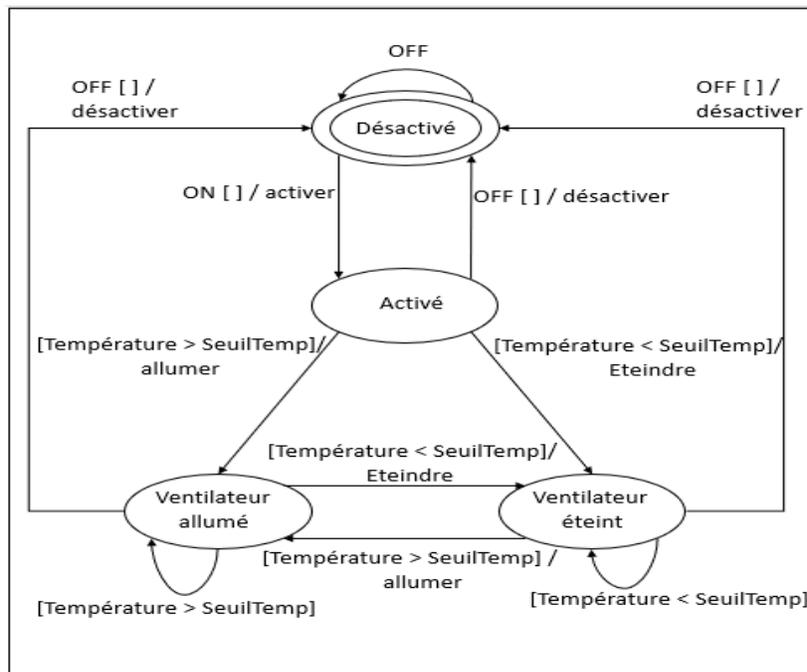
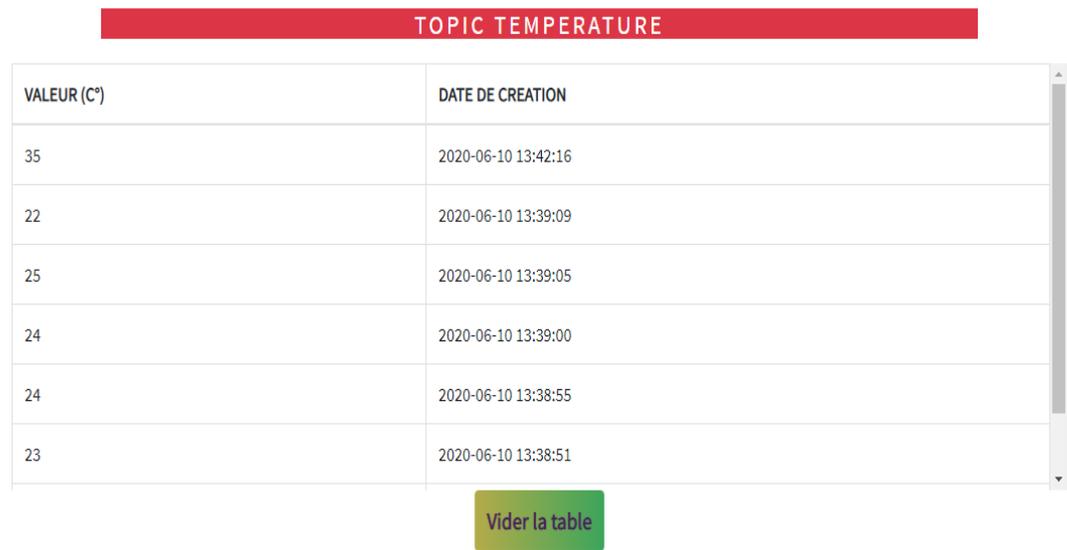


Figure 21 : Automate de fonctionnement de système IoT - surveillance de températures

Nous avons les mêmes états pour ce système, sauf que les conditions sont inversées. Si la température captée est supérieure au seuil réglé par l'utilisateur et le ventilateur est arrêté, l'action « allumer » s'exécute et le ventilateur passe à l'état « ventilateur allumé ». Si la température captée est inférieure, le ventilateur passe à l'état « ventilateur éteint » s'il est allumé ou reste éteint lorsqu'il l'est déjà. L'utilisateur peut aussi le forcer à s'éteindre s'il arrête l'application en revenant à l'état « désactivé ».

La figure 22 représente L'interface WEB du broker qui contient l'historique des températures captées.



The screenshot shows a web interface with a red header bar containing the text "TOPIC TEMPERATURE". Below the header is a table with two columns: "VALEUR (C°)" and "DATE DE CREATION". The table contains seven rows of data. Below the table is a green button with the text "Vider la table".

VALEUR (C°)	DATE DE CREATION
35	2020-06-10 13:42:16
22	2020-06-10 13:39:09
25	2020-06-10 13:39:05
24	2020-06-10 13:39:00
24	2020-06-10 13:38:55
23	2020-06-10 13:38:51

Figure 22 : Aperçu de l'interface WEB du broker - historique de températures

Nous avons mis les deux exemples de système IoT dans la même application mobile. L'utilisateur a les mêmes privilèges que l'exemple des distances, ce qui lui permet de recevoir la dernière température captée, comme il peut régler les seuils selon les besoins pour « tourner » ou « arrêter » le ventilateur (voir Figure 23).

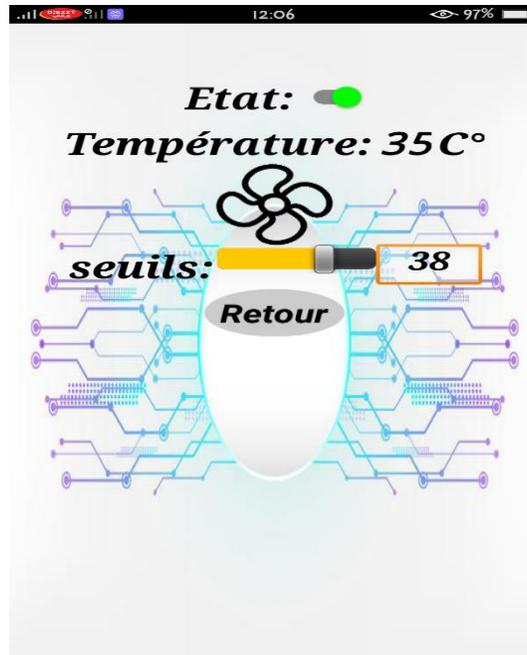


Figure 23 : Application mobile de système IoT de surveillance de températures

4. Conclusion

Dans ce chapitre, nous avons présenté nos exemples de systèmes IoT en expliquant comment ils utilisent l'infrastructure mise en place. Nous avons présenté les données captées sur notre l'interface WEB du broker et l'application mobile pour chaque exemple.

Conclusion Générale

La technologie d'internet des objets offre à l'être humain un confort supplémentaire qui facilite sa vie quotidienne, et cela grâce à une multitude d'applications et de protocoles de communications basés sur internet. Le travail réalisé dans ce mémoire s'inscrit dans ce contexte et il est composé en trois parties.

Dans la première partie nous avons fait une étude détaillée sur le concept de l'internet des objets. Grâce à cette étude nous avons constaté que différentes disciplines, technologies et techniques peuvent y figurer pour mettre en place une infrastructure IoT.

Dans la deuxième partie nous avons réalisé notre infrastructure basée sur l'architecture IoT en quatre couches. Nous avons présenté les composants matériels et logiciels que nous avons utilisé pour chaque couche.

Dans la troisième partie nous avons utilisé notre infrastructure pour développer et déployer deux exemples de systèmes IoT qui permettent de contrôler à distance des objets connectés. Le premier exemple consiste à surveiller les distances et le deuxième exemple à surveiller la température.

Ce travail étant une œuvre humaine, n'est pas un modèle parfait. Nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et les remarques tendant à améliorer d'avantage cette étude.

Bien évidemment, des améliorations futures peuvent être apportées à ce projet pour qu'il soit versatile, plus fiable, et adaptable. En perspective nous prévoyons :

- Développer d'autres versions pour le logiciel de la couche application, une version pour chaque type de terminal (tablette, smartphone, desktop, et pourquoi pas une Smart Watch).
- Ajouter d'autres fonctionnalités à l'application comme la représentation graphique de l'historique des données.
- Renforcer la sécurité du broker en ajoutant des protocoles de communication sécurisé tels que le SSL et le TLS.

Ce projet de fin d'études était une occasion précieuse pour mettre en œuvre les connaissances et les compétences acquises tout au long de notre formation et d'assumer la responsabilité qui nous été confiée. Cette expérience a aiguisée nos capacités d'analyse et de synthèse et a surtout fortifiée notre motivation, détermination et donner l'ambition de suivre une carrière dans le domaine d'informatique.

Annexes

1. Branchement des capteurs

La Figure 24 montre le branchement de nos deux capteurs sur la carte électronique NodeMCU. Le capteur de distance ultrason HC-SR04 est branché avec quatre fils : l'orange « écho » et le jaune « trig » sont reliés aux pins « d1 » et « d2 » respectivement pour la réception des données. Le câble bleu « GND » et le câble rouge « VCC » sont utilisés pour l'alimentation. Le capteur de température DHT11, est branché avec trois fils : le jaune « PIN » pour les données, le bleu « GND » et le rouge « VCC » pour l'alimentation.

Lorsque la carte NodeMCU est branché avec une alimentation, les capteurs captent et publient régulièrement les données

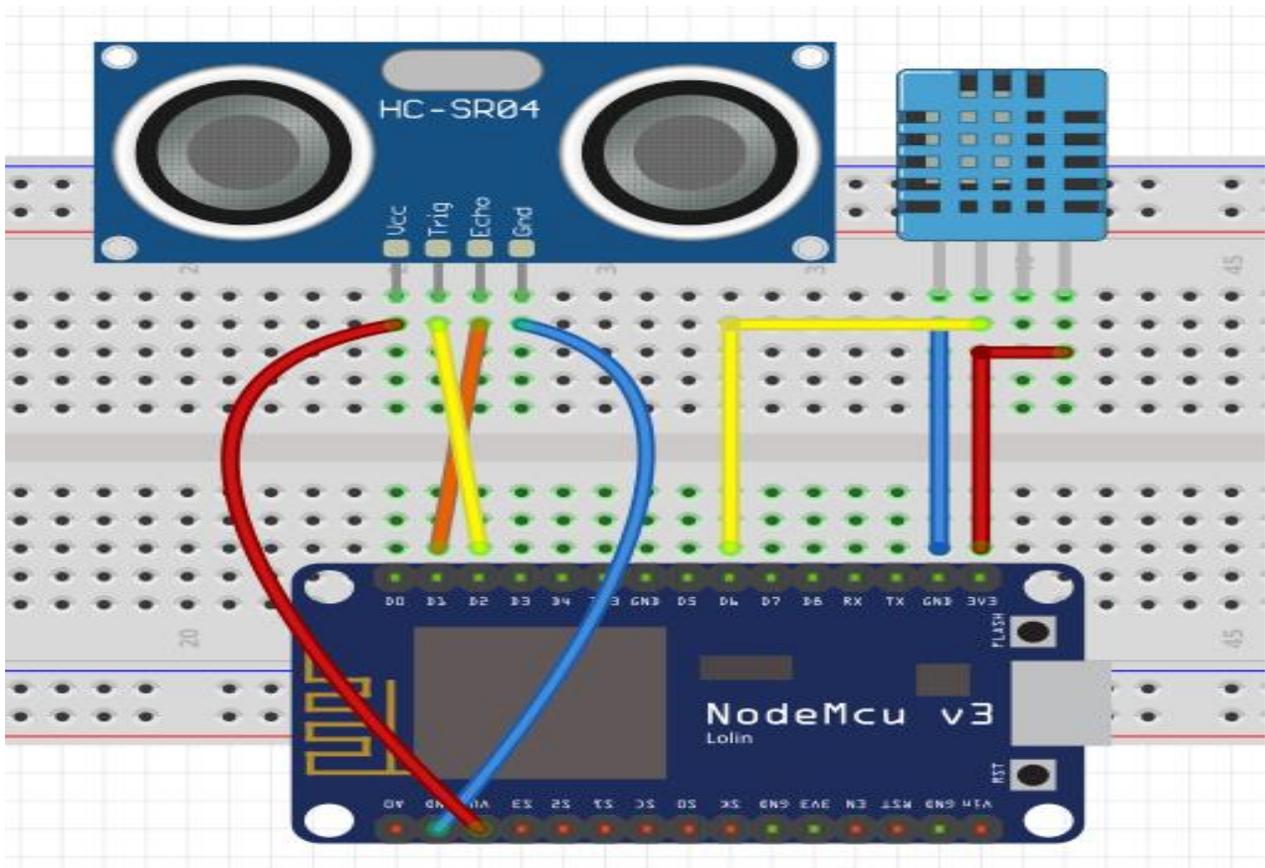


Figure 24 : Schéma de câblage des capteurs

2. Branchement des actionneurs

La Figure 25 montre le branchement de nos deux actionneurs « relais » sur la carte électronique NodeMCU. Chaque relais est branché avec trois fils : le bleu et le jaune pour le PIN de chaque relais, le noir « GND » et le rouge « VCC » pour passer le courant à la lampe ou le ventilateur.

Lorsque la carte NodeMCU est branché avec une alimentation, nos relais vont jouer le rôle des interrupteurs commandés

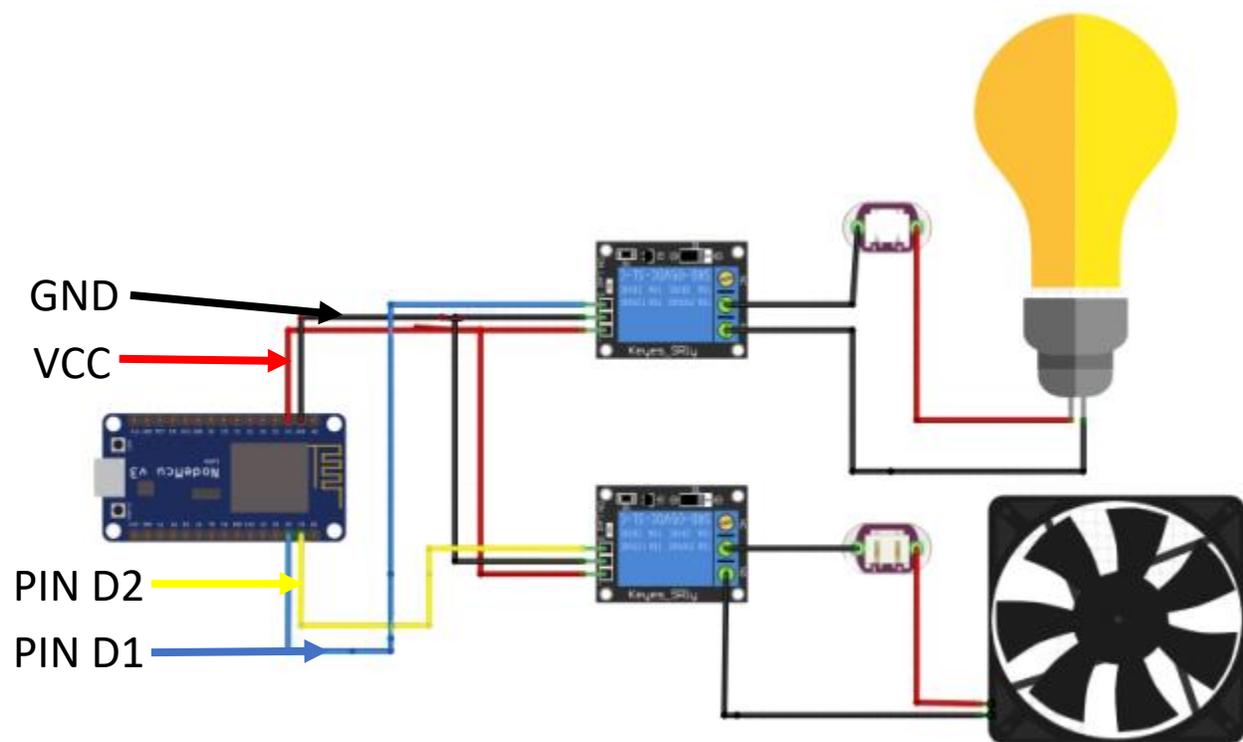


Figure 25 : Schéma de câblage des actionneurs

Bibliographie :

- [1] Botta, Alessio & Donato, Walter & Persico, Valerio & Pescapè, Antonio. (2015). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*. 56. 10.1016/j.future.2015.09.021.
- [2] Ge, Mouzhi & Bangui, Hind & Buhnova, Barbora. (2018). Big Data for Internet of Things: A Survey. *Future Generation Computer Systems*. 10.1016/j.future.2018.04.053.
- [3] Gilbert, Johann. (2018). Étude et développement d'un réseau de capteurs synchronisés à l'aide d'un protocole de communication sans fil dédié à l'Internet des objets.
- [4] INFISO D.4 Networked Enterprise & RFID INFISO G.2 Micro & Nanosystems (DG INFISO), in cooperation with the RFID Working Group of the European Technology Platform on Smart Systems Integration (EPOSS), *Internet of Things in 2020: A Roadmap for the Future* (Sept. 5, 2008)
- [5] ITU-T Study Group 20, [Présentation générale de l'Internet des objets \[archive\]](#) (rapport ITU-T Y.2060), ITU, juin 2012.
- [6] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.
- [7] Lampropoulos, Georgios & Siakas, Kerstin & Anastasiadis, Theofylaktos. (2018). Internet of Things (IoT) in Industry: Contemporary Application Domains, Innovative Technologies and Intelligent Manufacturing. 4. 109-118. 10.31695/IJASRE.2018.32910.
- [8] P. Srivastava, M. Bajaj and A. S. Rana, "Overview of ESP8266 Wi-Fi module based Smart Irrigation System using IOT," 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2018, pp. 1-5, doi: 10.1109/AEEICB.2018.8480949.
- [9] Parihar, Yogendra Singh. (2019). Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products. 6. 1085.

[10] Perwej, Dr. Yusuf & Haq, Kashiful & Parwej, Dr. Firoj & M., Mumdouh. (2019). The Internet of Things (IoT) and its Application Domains. International Journal of Computer Applications. 182. 36-49. 10.5120/ijca2019918763.

[11] Poyen, C Eng Faruk. (2019). Raspberry Pi and its Use in IoT Applications.

[12] Shawn DuBravac, Ph.D. Carlo Ratti, Ph.D. Internet des Objets : Évolution ou Révolution ?

Webographie :

- [12] Futura Tech. <https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/>
- [13] Stéphane Tanguy, Bernard Salha, Smart Grids-Cre, mars 2017. www.smartgrids-cre.fr/index.php?p=objets-connectes-definition
- [14] Robot Maker. <http://www.robot-maker.com/ouvrages/utiliser-picosynthe/une-pico-station-meteo/>
- [15] Génération Robots. <http://www.generationrobots.com/blog/fr/capteur-ultrason-capteur-a-ultrason-en-vente-chez-generation-robots/>
- [16] Muhammad Aqib Dutt, Electronics Hobbyists. <https://electronics hobbyists.com/nrf24l01-interfacing-with-arduino-wireless-communication/>
- [17] Arduino. <https://store.arduino.cc/arduino-uno-rev3>
- [18] Arduino. <https://store.arduino.cc/arduino-nano>
- [19] Célia Garcia-Montero, Journal Du Net. <https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1181267-les-reseaux-iot/>
- [20] Radio-identification, mai 2020. <https://fr.wikipedia.org/w/index.php?title=Radio-identification&oldid=170695162>
- [21] Global Sign, mai 2017. <https://www.globalsign.fr/fr/blog/qu-est-ce-qu-une-passerelle-iot/>
- [22] DEVELOPPEZ. <https://laurent-audibert.developpez.com/Cours-BD/?page=introduction-bases-de-donnees>
- [23] Ideematic. <https://www.ideematic.com/dictionnaire-digital/application-web/>
- [24] Application mobile, juin 2020. https://fr.wikipedia.org/wiki/Application_mobile

[25] Mosquito. <https://mosquitto.org/>

[26] HiveMQ Team, avril 2015. <https://www.hivemq.com/blog/mqtt-essentials-special-mqtt-over-websockets/>

[27] IOT BYTES, mai 2016. <https://iotbytes.wordpress.com/application-protocols-for-iot/>