

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou
Faculté de Génie Electrique et d'Informatique
Département d'informatique



Mémoire de Master

Spécialité

Informatique

Option

Réseaux, Mobilité et Systèmes Embarqués

Thème

« Transformation d'équipements classiques télécommandés en objets connectés et intelligents »

Présenté par

HALLI Thiziri

AMROUCHE Ramdane

Proposé et Encadré par

M^{rs} DAOUL.M

Devant le jury

M^{me} BELKADI.M Présidente du jury

M^{me} OUKFIF.K Examinatrice

Promotion : 2019 / 2020

Remerciements

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et béni par sa volonté durant toute cette période.

Notre profonde gratitude et sincères remerciements vont à notre promoteur M. DAOUI Mehammed pour sa présence continuelle, son encouragement et sa patience tout au long de ce travail.

Nous adressons nos remerciements aux membres du jury, devant qui, nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu.

Nous remercions les membres du jury, devant qui, nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu .

Nous réservons ici une place particulière pour remercier vivement tous ceux qui, de près ou de loin, nous ont aidés et encouragés tout au long de réalisation de ce modeste travail.

Un grand merci à nous pour avoir cru en nous et pour notre persévérance.

A la mémoire de notre regretté enseignant « M. HAMDANI Chabane »

Dédicaces

A mon père et à ma mère mes deux piliers dans la vie.

Ainsi qu'à toute ma famille et proches qui m'ont toujours soutenu et qui n'ont jamais cessé de croire en moi.

*A la mémoire des deux personnes que j'ai perdu au cours de cette année « **Yemma Hebou** » et « **Vava Mouh** » j'espère que de là où vous êtes, vous êtes fiers de moi.*

A notre petite famille de la spécialité RMSE promotion (2019/2020)

Et pour finir, une dédicace particulière à mon binôme Ramdane.

Thiziri.

Dédicaces

A mon père et à ma mère mes deux piliers dans la vie.

Ainsi qu'à toute ma famille et proches qui m'ont toujours soutenu et qui n'ont jamais cessé de croire en moi.

A notre petite famille de la spécialité RMSE promotion (2019/2020)

Et pour finir, une dédicace particulière à ma binôme Thiziri.

Ramdane.

Table des matières

Introduction Générale	1
1 Généralités sur les objets connectés et la domotique	3
1.1 Introduction :	3
1.2 Internet des objets (IoT) :	3
1.2.1 Définition :	3
1.2.2 Les composants d'un système IoT [Nic11],[1] :	4
1.2.3 Les caractéristiques d'un système IoT :	7
1.3 Domaines d'application de l'IoT :	8
1.3.1 Grand public :	8
1.3.2 Industriel :	10
1.3.3 Militaire :	11
1.4 Les avantages et inconvénients :	13
1.4.1 Avantages :	13
1.4.2 Inconvénients :	14
1.5 Protocoles de transfert de données IoT :	14
1.6 Les smart –homes :	15
1.6.1 Domaines d'application des maisons connectées [2] :	16
1.7 Conclusion :	18
2 Conception du système de mise à niveau d'équipements classiques.	19
2.1 Introduction :	19
2.2 L'objectif du projet :	19
2.3 Description du fonctionnement global :	21
2.4 Description des schémas internes et processus au niveau des cartes ESP32 :	22
2.4.1 Au niveau de l'objet :	22
2.4.2 Au niveau du détecteur de mouvement :	25
2.5 Description du fonctionnement interne de la Raspberry Pi 3 :	27
2.6 Description de fonctionnement de l'application Android :	29
2.7 Le reverse engineering :	31
2.8 Conclusion :	31
3 Réalisation du système	32
3.1 Introduction	32
3.2 Présentation de l'environnement de développement :	32
3.2.1 Présentation de l'environnement matériel :	32
3.2.2 Caractéristiques : [Fra14]	33
3.2.3 Raspberry pi 3 modèle b+ :	33
3.2.4 Présentation de l'environnement logiciel :	36

3.3	Présentation des étapes de réalisation :	40
3.3.1	Reverse engineering :	41
3.3.2	Préparation du la carte Arduino UNO :	41
3.3.3	Préparation du la Raspberry :	44
3.3.4	Le détecteur de mouvement :	47
3.3.5	La lampe multicolor :	48
3.3.6	L'application Android :	49
3.3.7	Création de la base de données Firebase :	53
3.4	Conclusion	54
	Conclusion générale	55

Table des figures

1.1	Tableau résumant les technologies de réseaux	5
1.2	Les composants d'un système IoT	7
1.3	Google Home Voice	8
1.4	WeMo Light Switch	9
1.5	WeMo Light Switch	9
1.6	Les AMR VirtualConveyor	11
1.7	L'IoT dans le domaine militaire	12
1.8	Fonctionnement du protocole MQTT	15
2.1	Situation des objets classiques	20
2.2	Situation future des objets	20
2.3	Situation finale des objets	20
2.4	Schéma du fonctionnement global	21
2.5	L'architecture interne de l'objet	22
2.6	Le fonctionnement de la première partie de la tâche 1	23
2.7	Le fonctionnement de la deuxième partie de la tâche 1	24
2.8	Le mécanisme de synchronisation des deux taches de notre système	24
2.9	Le fonctionnement de la tâche 2 de notre système.	25
2.10	Schéma du fonctionnement au niveau de capteur	25
2.11	Illustration du fonctionnement de la tâche 1	26
2.12	L'illustre de fonctionnement de la tâche2	27
2.13	Schéma du fonctionnement au niveau de la raspberry.	28
2.14	Diagramme de l'activité de l'application	29
2.15	L'illustration de principe de reverse engineering	31
3.1	Les différentes Pins de l'ESP32	33
3.2	le raspberry pi 3 modèle b+	34
3.3	Arduino UNO	35
3.4	L'émetteur IR	35
3.5	Le Récepteur IR	36
3.6	PIR motion Sensor	36
3.7	Plateforme IO	37
3.8	Logo Android	37
3.9	Logo freeRTOS	38
3.10	Logo FireBase	38
3.11	Logo MQTT	40
3.12	La lampe utilisée dans le cadre de notre travail.	40
3.13	Le schéma synoptique des commandes IR.	41
3.14	L'illustration des commandes infrarouges dans le moniteur série arduino.	42

3.15	Le schéma de câblage de récepteur infrarouge avec Arduino UNO	42
3.16	Le branchement de récepteur infrarouge avec Arduino UNO	43
3.17	La commande de récepteur infrarouge a distance.	43
3.18	La configuration d'adresse IP sur la raspberry	44
3.19	illustration d'installation de mosquitto	45
3.20	Illustration d'installation de PyreBase	46
3.21	le câblage entre la carte ESP32 et le détecteur de mouvement.	47
3.22	le schéma de câblage entre la carte ESP32 et le détecteur de mouvement sous fritzing.	47
3.23	La lampe conectée	48
3.24	L'authentification avec un compte google	49
3.25	L'interface d'authentification	50
3.26	Interface d'accueil.	51
3.27	L'interface dédiée à la Lampe.	52
3.28	L'enchainement des interfaces de l'application android.	53
3.29	la base de données Firebase	53

Introduction générale

Depuis son apparition l'IoT ne cesse de fasciner, de grandir et de s'améliorer. C'est un domaine en pleine expansion qui ne cesse d'attirer les gens de par sa valeur ajoutée. Il apporte des solutions à la vie quotidienne et contribue à l'amélioration de la qualité de vie des gens.

L'internet des objets est l'extension d'internet à des choses, objets et lieux du monde physique. Alors qu'auparavant internet ne s'étendait pas au-delà du monde électronique, l'avènement de l'IOT lui a permis d'entrer dans une nouvelle dimension celle du monde réel. Ce nouveau paradigme permet les échanges d'information et de données provenant de dispositifs et équipements physiques avec des plateformes et objets technologiques tels que les smartphones ou ordinateurs via le réseau internet.

L'IoT se trouve dans plusieurs domaines, dans l'industrie, le commerce, le marketing, dans les maisons, les villes, les immeubles, dans le domaine militaire, dans les transports et bien d'autres. On parle alors de smart homes, smart cities, smart cars, smart buildings, etc.

Sans qu'on s'en rende compte la technologie est présente partout dans notre vie quotidienne dans nos téléphones nos montres intelligentes, glucomètres et bien d'autres. Elle met à notre disposition une grande quantité d'informations qui améliorent nos vies, nous orientent lors des prises de décisions en mettant à notre disposition le maximum d'informations disponibles, elle nous permet d'être à jour et de facilement actualiser nos informations. Actuellement le marché de smart home regorge d'équipements domestiques qui sont « smart » dès leur fabrication. Mais dans ce cas si une personne, ayant déjà des équipements domestiques classiques, veut avoir des équipements connectés, elle se verra dans l'obligation de les abandonner et d'acheter des nouveaux, ce qui n'est pas très bon ni pour son porte-monnaie ni pour notre planète.

Notre projet vise à apporter une solution aux faits exposés ci-dessus. Nous souhaitons mettre à niveau les équipements domestiques déjà existants et les rendre « smart » afin de pouvoir les utiliser dans une maison intelligente ou système IOT, et qu'ils puissent communiquer et interagir avec leur environnement. Notre travail consiste à équiper un objet physique de capteurs, d'un émetteur infrarouge et d'une carte électronique de type ESP32 afin de le rendre intelligent et lui permettre de communiquer avec une Smart Home. La Smart Home est réalisée à base d'une carte Raspberry Pi embarquée avec une application de Smart Home basée sur le protocole MQTT.

La suite de notre mémoire est organisé en trois chapitre et une conclusion générale.

Le premier chapitre donne des généralités sur le domaine d'application qui est les objets connectés en général et les smartHome en particulier.

Le deuxième chapitre présente les détails de la conception de notre système.

Le troisième chapitre donne les outils de réalisation qui nous ont permis d'aboutir au produit final.

- **Chapitre 1 : Généralités sur les objets connectés et la domotique :** Dans ce chapitre nous présenterons le concept de l'IoT, ses domaines d'application, ses avantages et inconvénients ainsi que le protocole de communication MQTT. Par la suite nous verrons quelques définitions et généralités concernant les maisons intelligentes.
- **Chapitre 2 : Conception du système de mise à niveau d'équipements classiques :** Dans ce chapitre nous présenterons les différentes étapes de conception de notre application. Dans un premier temps, nous aborderons le schéma global de notre solution. Après cela, nous détaillerons chaque schéma interne de chaque entité.
- **Chapitre 3 : Réalisation :** et tests. Dans ce dernier, nous présenterons les différents outils de réalisation de notre solution. Enfin, nous présenterons les différentes étapes suivies pour concrétiser notre travail.

Chapitre 1

Généralités sur les objets connectés et la domotique

1.1 Introduction :

Internet est l'un de moyens technologiques les plus utilisés actuellement, en raison de ses différents avantages, notamment son accès à tous les domaines. Cette technologie a été en mesure d'unir le monde et permet la transmission d'informations d'une manière rapide et fluide à travers les quatre coins du globe. Contrairement au Web, l'Internet se développe et s'améliore continuellement, mais il n'a pas connu de transformation fondamentale. L'IoT

(Internet des Objets) est la première véritable évolution de l'Internet. Celle-ci donnera lieu à des applications révolutionnaires capables de transformer profondément notre mode de vie, et notre façon d'apprendre, de travailler et de nous divertir. L'IoT commence déjà à doter l'Internet de capacités sensorielles (température, pression, vibration, luminosité, humidité, tension).

L'apparition du terme « Internet des Objets » est récente. Ce dernier désigne la nouvelle génération d'Internet et des réseaux. Ce paradigme donne la possibilité aux appareils d'être connectés et de communiquer entre eux via un protocole Internet. Ces dispositifs comprennent des instruments, des capteurs, des actionneurs et divers outils d'intelligence artificielle. [Eva11]

Ce chapitre comportera l'étude générale des systèmes d'IoT. L'objectif visé est de comprendre les caractéristiques et spécificités des objets connectés, particulièrement la communication entre eux, afin de pouvoir développer notre système qui permet de transformer des équipements classiques télécommandés en objets connectés et intelligents.

1.2 Internet des objets (IoT) :

1.2.1 Définition :

Le terme IOT (Internet Of Things) porte plusieurs définitions. La plus simple serait de dire qu'il sert à désigner un système où des objets physiques connectés à un réseau, sont capables de créer des données et les transmettre, ils sont aussi en mesure d'interagir entre eux, avec l'utilisateur et leur environnement.

Selon l'UIT (Union Internationale des Télécommunications), l'IOT est défini comme « 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 interoperables existantes ou en évolution ».[[Nic11](#)]

En effet, depuis son apparition il y a 10 ans, l'Internet des Objets prend de plus en plus d'ampleur avec le développement des technologies. De plus en plus de domaines commencent à percevoir les nouvelles opportunités. Dès lors, il est difficile d'en donner une définition stable.[[Keu16](#)]

1.2.2 Les composants d'un système IoT [[Nic11](#)],[[1](#)] :

Un système IoT est composée de plusieurs éléments parmi eux nous distinguons :

Les objets (capteurs) :

Les objets sont dotés d'éléments capteurs qui permettent d'observer l'environnement dans lequel ils sont déployés. Ils offrent la possibilité de récolter des données en temps réel qui sont aussi diverses que les métiers.

Les capteurs sont des dispositifs permettant de transformer une grandeur physique observée (température, luminosité, mouvement etc...) en une grandeur digitale utilisable par des logiciels. Il existe une très grande variété de capteurs de tous types, les objets connectés ont souvent la fonction de captage de ces grandeurs physiques sur leur lieux d'utilisation.

Nous pouvons avoir des capteurs de température, d'humidité, de position, de temps de fonctionnement, de niveau d'alerte, de lumière, de présence, de proximité, de déplacement, d'accélération, de rotation, de son, de vibration, de gaz, de pression, etc.

Le réseau (connectivité) :

Les réseaux sont le maillon pondérant d'un projet IoT. Sans réseau aucun transfert de données n'est possible ni à l'interne ni à l'externe. Le choix de la technologie à adopter pour ce dernier se fait en suivant certains critères qui diffèrent d'un système à un autre tel que la taille de zone de couverture, l'énergie ... Parmi les différentes technologies nous avons :

- Les LPWAN (Low Power Wide Area Network) : Dans cette catégorie nous trouvons des technologies qui sont à longue portée et à faible consommation d'énergie, notamment les technologies SIGFOX et LoRaWAN.
- Le M2M (Machine to Machine) : Il s'agit tout simplement d'utiliser des réseaux télécom existants à savoir de la 2G à la 4G. Majoritairement c'est des abonnements souscrits auprès des opérateurs de téléphonie donnant accès à des volumes de données.
- La RFID (Radio Frequency Identification) : C'est une méthode d'identification à distance à l'aide de marqueurs (Tags) et de lecteurs de radiofréquences. Il existe deux technologies de RFID, la RFID passive où le Tag est une simple antenne qui est activée dès qu'elle se trouve à portée d'un émetteur. Quant à la RFID active, il s'agit d'objets comprenant un Tag capables d'émettre un signal à proximité d'un autre objet RFID. Cette technologie ne se suffit pas à elle-même et doit absolument être couplée à une autre pour collecter et stocker les données.

1.2. INTERNET DES OBJETS (IOT) :

- Le Bluetooth : Il s'agit d'une technologie radio de moyenne portée qui permet d'envoyer des messages de grande taille et en grande quantité. Cependant, cette connectivité ne se suffit pas à elle-même car elle nécessite une tierce technologie pour transférer et stocker les données. D'autre part, il s'agit d'un moyen de communication disposant d'un grand débit, puisqu'il repose sur la bande de fréquence 2,4GHz tout comme le wifi.

Voici la figure 1.1 ci-dessous résumant les technologies de réseaux :

Technologie	Points forts	Points Faibles	Type de cas d'usage
LoRa	<ul style="list-style-type: none"> Faible consommation énergétique Longue portée Un standard opérable 	<ul style="list-style-type: none"> Couverture mondiale encore faible Taille et volume de données 	<ul style="list-style-type: none"> Relève de compteurs d'énergie Envoi d'informations ponctuelles (géolocalisation,
Sigfox	<ul style="list-style-type: none"> Faible consommation énergétique Longue portée Un opérateur unique 	<ul style="list-style-type: none"> Couverture mondiale encore faible Taille et volume de données 	<ul style="list-style-type: none"> Relève de compteurs d'énergie Envoi d'informations ponctuelles (géolocalisation,
M2M	<ul style="list-style-type: none"> Couverture mondiale importante 	<ul style="list-style-type: none"> Consommation d'énergie Portée 	<ul style="list-style-type: none"> Terminaux de paiement Application avec de gros volumes de données ou d'envoi de données sur incident
RFID	<ul style="list-style-type: none"> Pas d'émission d'ondes Pas besoin d'énergie 	<ul style="list-style-type: none"> Faible portée Nécessité d'une passerelle 	<ul style="list-style-type: none"> Géolocalisation de zone Identification (contrôle d'accès, autorisation,...)
Bluetooth	<ul style="list-style-type: none"> Taille et volume des messages Débit 	<ul style="list-style-type: none"> Faible portée Nécessité d'une passerelle 	

FIGURE 1.1 – Tableau résumant les technologies de réseaux

Les données : (La valeur à l'état brut) :

Nous pouvons considérer les données comme des diamants bruts. Il s'agit d'informations brutes récoltées et captées depuis les objets. Afin de créer de la valeur pour les utilisateurs de ces données il est nécessaire de les stocker, archiver, trier, et sauvegarder dans des bases de données et de correctement structurer cette dernière, cela améliorera la performance des services IoT d'exploitation.

Les informations :

Les informations sont les résultantes de données traitées, corrélées, triées et analysées, un diamant bien taillé. Ces informations doivent elles-aussi être stockées, archivées et sauvegardées dans des bases de données.

Par exemple :

- Le capteur d'une porte d'entrée fournit une donnée : PORTE OUVERTE ou PORTE FERMÉE.
- Nous disposons d'une donnée universelle : L'HEURE.

L'information sera de nous alerter : « votre porte est ouverte et à cette heure ce n'est pas normal ». Deux données ont été corrélées, le capteur de porte et l'heure, pour générer une information.

Les applications d'exploitation :

Les applications d'exploitation sont en principe les interfaces Homme-machine (IHM) dans lesquelles nous pouvons visualiser les données sous forme de tableau de bord, il s'agit de notre Diamant montée en bijou.

Nous retrouvons des outils tels que des graphiques, des tableaux de données, des rapports, etc.

Cependant, ces bibliothèques d'objets ne suffisent pas à faire une bonne application d'exploitation. Aussi, des mécanismes complémentaires doivent être développés, tels qu'un protocole d'alerte par mail ou SMS à des équipes d'exploitation afin d'intervenir sur des incidents ou pour des dépassements de seuils. Pour cela, des mécanismes de comparaison entre les mesures reçues et des constantes ou entre un jeu de mesure doivent être mis en place.

Enfin les applications d'exploitation sont accessibles à travers des appareils mobiles tels que les smartphones et les tablettes.

Les actionneurs :

Les actionneurs sont des dispositifs qui transforment une donnée digitale en phénomène physique pour créer une action, ils sont en quelque sorte l'inverse du capteur. Exemple d'actionneurs : Alarmes, Haut-parleurs, relais, Lampes, Moteurs, Pompes, Serrures, Vannes, Ventilateur, Vérins, etc.

Les sources d'énergie pour les IoTs :

Les sources d'énergie sont de 4 types :

- Alimentation filaire pour les objets ayant accès à une prise de courant.
- Piles ou batteries pour ceux qui n'y ont pas accès ou de manière occasionnelle (recharge).
- Capteurs d'énergie ou « energy harvesting » (photovoltaïque, piezoélectrique, thermoélectrique, cinétique...) pour rallonger la durée de vie des objets à très faible consommation.
- Les objets passifs sans piles qui sont alimentés par les ondes électromagnétiques des lecteurs (RFID, NFC...).

L'énergie est un des grands défis des objets connectés, tant pour garantir la plus grande durée de vie possible sans maintenance, que pour garantir un respect environnemental malgré la multiplication des objets connectés énergivores qui envahissent nos espaces personnels et professionnels



FIGURE 1.2 – Les composants d'un système IoT

1.2.3 Les caractéristiques d'un système IoT :

Les caractéristiques fondamentales d'un système IoT sont les suivantes :

- Inter-connectivité : Tout peut être interconnecté avec une infrastructure globale d'information et de communication.
- Services liées aux objets connectés : L'IoT est capable de fournir des services liés aux objets dans les limites des contraintes, telles que la protection de la vie privée et la cohérence sémantique entre les choses physiques et leurs objets virtuels associés. Afin de fournir des services liés aux objets dans les contraintes sur les choses, les technologies dans le monde physique et le monde de l'information vont changer.
- Hétérogénéité : les périphériques de l'IoT sont hétérogènes en fonction des plateformes matérielles et des réseaux. Ils peuvent interagir avec d'autres appareils ou plateformes de services via différents réseaux.
- Changements dynamiques : l'état des dispositifs change dynamiquement, par exemple dormir et se réveiller, être connecté et/ou déconnecté ainsi que le contexte des dispositifs, y compris l'emplacement et la vitesse. De plus, le nombre d'appareils peut changer de façon dynamique.
- Enorme échelle : le nombre de périphériques qui doivent être gérés et qui communiquent entre eux sera d'au moins un ordre de grandeur supérieur à celui des périphériques connectés à Internet. La gestion des données générées et leur interprétation est complexe. Cela concerne la sémantique des données, ainsi que la gestion efficace de ces données.

- **Sécurité** : comme nous gagnons des avantages de l'IoT, nous ne devons pas oublier la sécurité. En tant que créateur et concepteurs de l'IoT, nous devons concevoir des mécanismes assurant la sécurité. Cela inclut la sécurité de nos données personnelles et la sécurité de notre bien-être physique. La sécurité demeure parmi d'un des points faibles d'un système IoT. La sécurisation des points de terminaison des réseaux et des données qui le traversent signifie la création d'un paradigme de sécurité qui va évoluer.
- **Connectivité** : La connectivité permet l'accessibilité et la compatibilité du réseau. L'accessibilité permet de se mettre sur un réseau alors que la compatibilité fournit la capacité commune de consommer et de produire des données.

1.3 Domaines d'application de l'IoT :

Nous pouvons classer les domaines d'application selon trois axes :

1.3.1 Grand public :

Cet axe comprend les dispositifs portables tel que les smart phones les montres Intelligentes, l'équipement domotique (cafetières, téléviseurs...), les voitures intelligentes ou même les bracelets médicaux ou puces intégrées qui permettent aux personnes ainsi qu'à leurs médecins de contrôler leurs états de santé en temps réel. Ici les exigences sont assez sommaires, la quantité de données est petite ainsi que leurs valeur et exploitation.

Exemples [3]

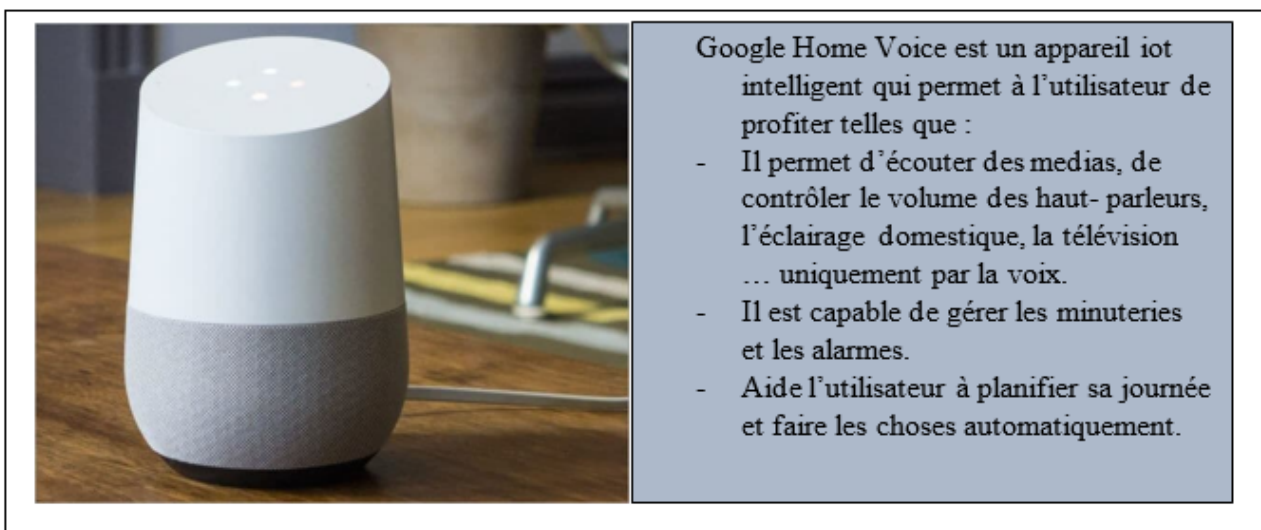


FIGURE 1.3 – Google Home Voice

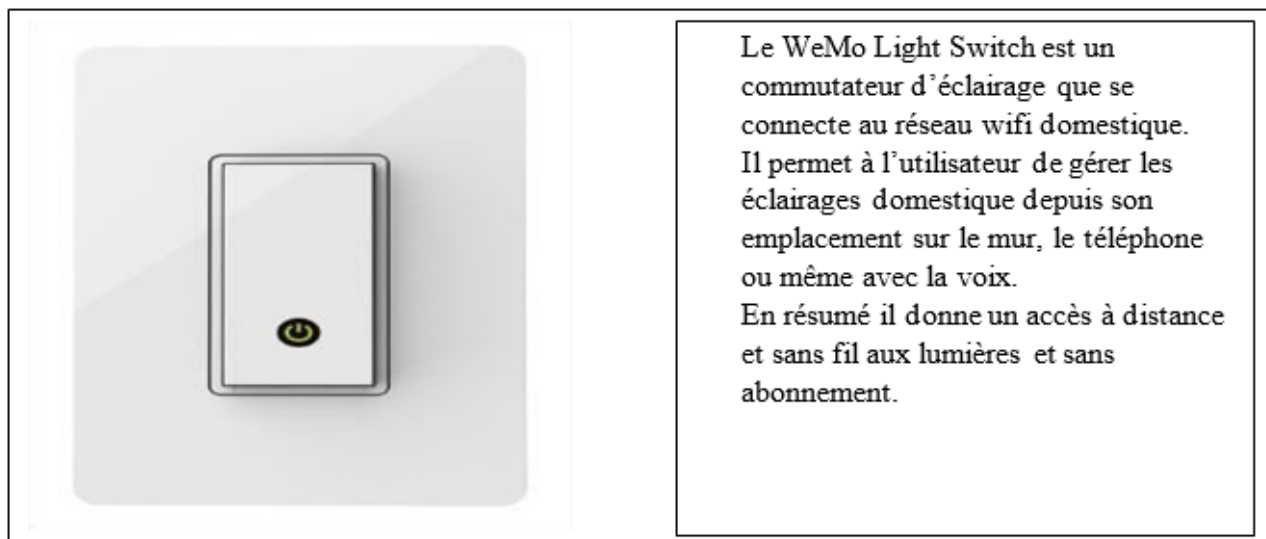


FIGURE 1.4 – WeMo Light Switch

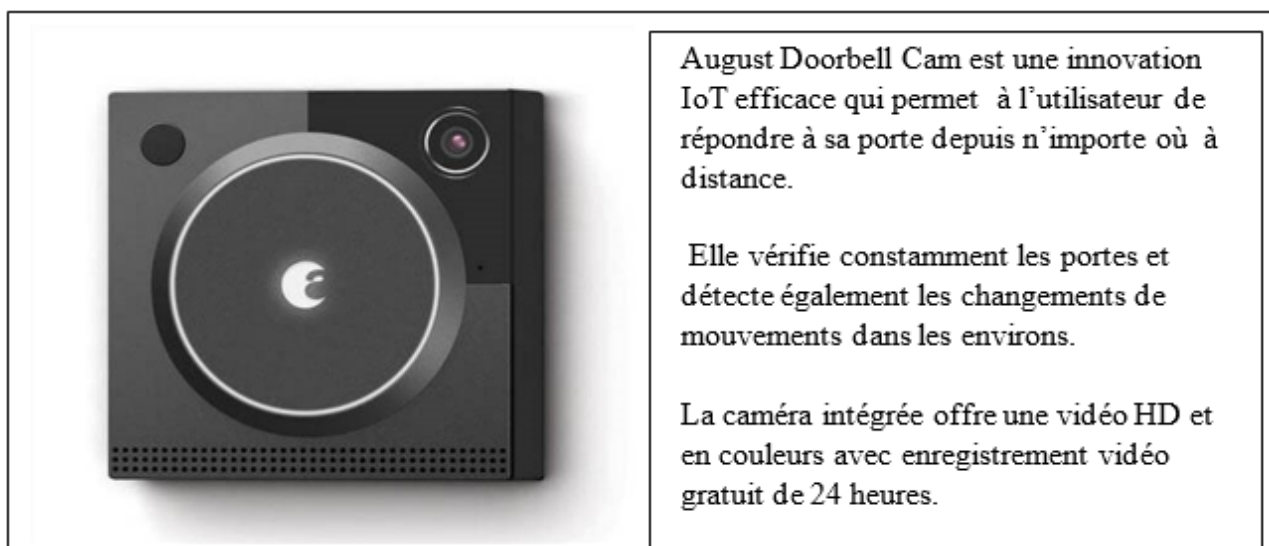


FIGURE 1.5 – WeMo Light Switch

1.3.2 Industriel :

Le terme « industrie 4.0 » annonce l'avènement d'une nouvelle révolution industrielle grâce à l'industrie intelligente. Le terme « Internet des objets industriels » a une promesse plus discrète de stimuler l'efficacité opérationnelle grâce à l'automatisation, la connectivité et l'analyse. Cependant l'impact de l'IoT – sur l'industrie dans son ensemble – est plus large.

Cet axe concerne les machines complexes dans le domaine du médicale, de l'aérospatial, du commerce, de l'industriel même d'autres équipements plus simples tel que le compteur intelligent qui s'occupe de la gestion des stocks ainsi que dans l'agriculture des capteurs qui permettent de vérifier la température, l'humidité, l'état des sols et bien d'autres qui permettent à l'agriculteur de s'occuper de ses plantations sans avoir à se déplacer.

Contrairement à l'axe grand public, ici la quantité de données peut être colossale, leur valeur très élevée, plus que celle du matériel physique, elles peuvent être exploitées sur l'ensemble de la chaîne de valeur d'une entreprise.

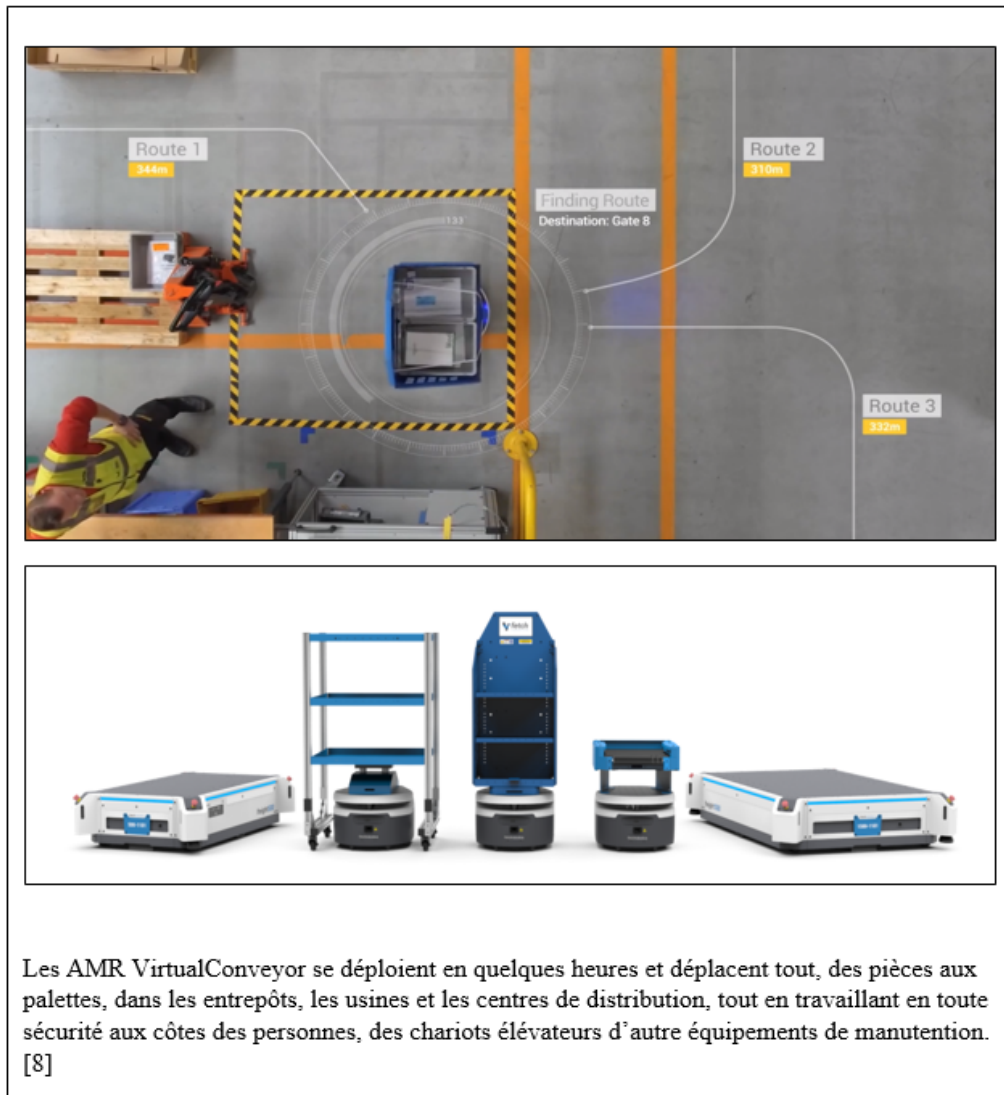


FIGURE 1.6 – Les AMR VirtualConveyor

1.3.3 Militaire :

Bien qu'elles ne soient plus le moteur principal en termes de d'innovation de l'IOT, les armées sont, dans une certaine mesure, à l'origine et au cœur du développement des objets connectés.

Dans l'Internet of Military Things (IoT) ou Internet of Battelfied Things (IoBT), les appareils de détection et de calcul portés par les soldats et intégrés dans leurs combinaisons de combat, casques systèmes d'armes et autres équipements sont capables d'acquérir une variété de la biométrie dynamique comme le visage, l'iris, l'espace péri oculaire, les empreintes digitales la fréquence cardiaque, la démarche, les gestes et les expressions faciales. [Som16]

On distingue 3 sortes d'objets connectés utilisés par les armées :

- a. Des objets connectés spécifiquement développés pour des besoins militaires (des capteurs de renseignement, plateformes de liaison et combat...).

1.3. DOMAINES D'APPLICATION DE L'IOT :

- b. Des objets connectés civils, achetés et utilisés dans un cadre militaire (logistique, transport, maintenance, santé, smart buildings, etc.).
- c. Des objets connectés civils achetés et utilisés individuellement par des militaires sans réel contrôle (une montre connectée qu'un militaire porte pendant une opération...).

Pour cet axe, la quantité de données est importante et leur valeur très élevée plus que dans n'importe quel autre axe.

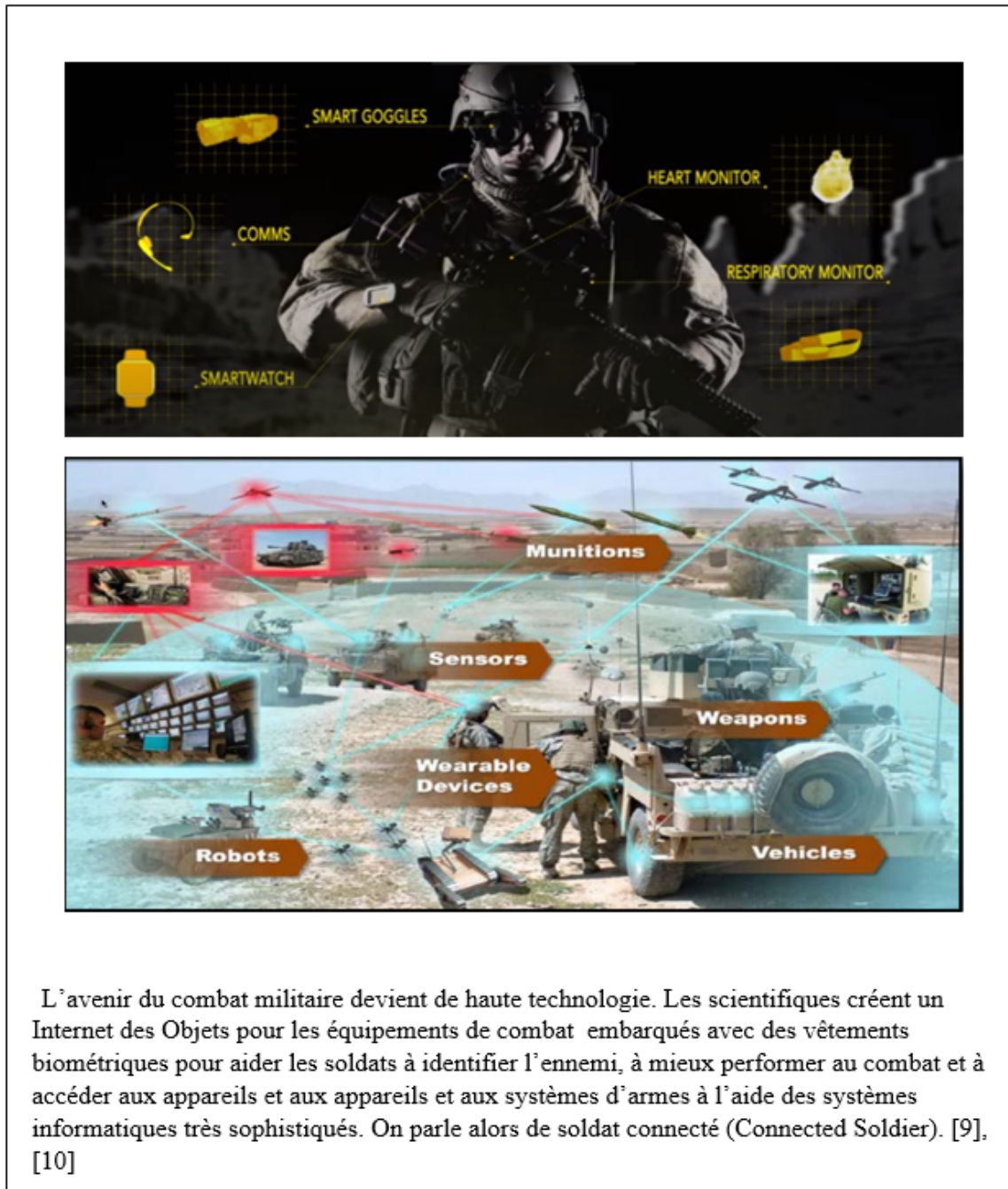


FIGURE 1.7 – L'IoT dans le domaine militaire

1.4 Les avantages et inconvénients :

L'Internet des Objets comporte plusieurs avantages et inconvénients en voici quelques-uns :

1.4.1 Avantages :

- Facilite la circulation de l'information et des échanges. -Booste la productivité au sein des entreprises.
- La collecte de données qui s'y fait à travers permet aux professionnels du marketing d'améliorer la qualité de leurs services.
- Permet d'améliorer la qualité de vie et apporte plus de confort au quotidien.
- Le suivi des patients permet un gain de temps et d'argent (et au patient et au médecin traitant).
- Offre une bonne automatisation et un bon contrôle de l'environnement.
- Permet d'économiser, notamment pour les particuliers, plus de temps d'argent en réduisant les tâches manuelles faites habituellement par des personnes.
- L'automatisation des tâches de la vie quotidienne procure une meilleure qualité de vie.
- Les dispositifs sont très exposés aux cyber-attaques.
- Sensible aux pannes qui dans certains cas peut engendrer des dangers.
- Violation de la vie privée des utilisateurs, à savoir que leurs données peuvent être exploitées. -Portent beaucoup de failles de sécurité.
- La réduction des tâches manuelles entraîne des suppressions d'emplois ce qui fait qu'un bon nombre de personnes se retrouve au chômage.
- Les appareils connectés n'ont pas de norme de compatibilité internationale.
- Les objets connectés peuvent devenir très complexes et entraîner une défaillance ou devenir incontrôlables.

Cependant, l'intérêt des utilisateurs pour l'IIoT surpasse les risques que cela engendre. Et à ce jour le domaine de l'IOT ne cesse de s'agrandir et ne connaît aucune limite hormis celle juridique qui vise à protéger les données personnelles, la traçabilité et l'identité numérique de ses utilisateurs.

1.4.2 Inconvénients :

- Les dispositifs sont très exposés aux cyber-attaques.
- Sensible aux pannes qui dans certains cas peut engendrer des dangers.
- Violation de la vie privée des utilisateurs, à savoir que leurs données peuvent être exploitées.
- Portent beaucoup de failles de sécurité.
- La réduction des tâches manuelles entraîne des suppressions d'emplois ce qui fait qu'un bon nombre de personnes se retrouve au chômage.
- Les appareils connectés n'ont pas de norme de compatibilité internationale.
- Les objets connectés peuvent devenir très complexes et entraîner une défaillance ou devenir incontrôlables.

Cependant, l'intérêt des utilisateurs pour l'ilot surpasse les risques que cela engendre. Et à ce jour le domaine de l'IOT ne cesse de s'agrandir et ne connaît aucune limite hormis celle juridique qui vise à protéger les données personnelles, la traçabilité et l'identité numérique de ses utilisateurs

1.5 Protocoles de transfert de données IoT :

Les protocoles de transfert de données IoT[Sen10] sont un ensemble de règles définissant le mode de communication utilisés pour connecter des appareils IoT à faible consommation.

Ces protocoles fournissent une communication point à point entre le matériel et l'utilisateur via un réseau. Parmi ces protocoles nous distinguons le MQTT.

Message Queuing Telemetry Transport (MQTT)

MQTT (standard pour l'IoT depuis 2015) est un protocole de messagerie de publication et d'abonnement (publish/subscribe) basé sur le protocole TCP/IP. Un client, appelé publisher, établit dans un premier temps une connexion de type 'publish' avec le serveur MQTT, appelé broker. Puis, le publisher transmet les messages au broker sur un canal spécifique, appelé topic.

Par la suite, ces messages peuvent être lus par des abonnés, appelés subscribers, qui au préalable ont établi une connexion de type 'subscribe' avec le broker. Ainsi, la transmission et la consommation des messages se font de manière asynchrone.

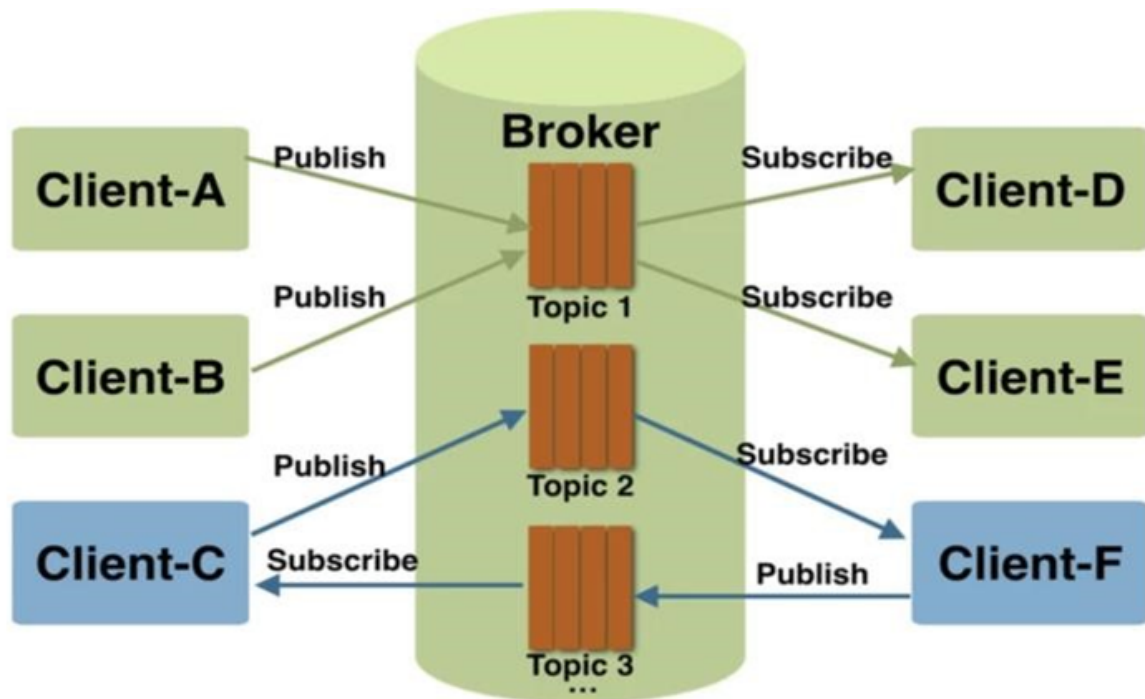


FIGURE 1.8 – Fonctionnement du protocole MQTT

MQTT a la particularité d'être un protocole léger parce que le nombre de messages est restreint et ont des tailles faibles.

En effet, chaque message se compose d'un en-tête fixe de 2 octets (spécifiant le type de message et le niveau de qualité de service employée), d'un en-tête variable facultatif, d'une payload (charge utile) limitée à 256 Mo. Il existe trois niveaux de qualité de service (QoS) qui déterminent la façon dont le protocole MQTT gère le contenu. Les clients abonnés peuvent spécifier le niveau de QoS maximal qu'ils souhaitent recevoir. Toutefois, plus le niveau de qualité est élevé et plus cela est gourmand en termes de latence et de bande passante, à cause des répétitions et des accusés de réception supplémentaires

1.6 Les smart –homes :

Les termes « domotiques », « maison connectée » et « maison intelligente » peuvent souvent être mélangés et incompris, voici quelques définitions permettant de mieux les comprendre :

La domotique :

La domotique, venant du mot latin « domus » (maison en latin) et du suffixe « tique » (pour informatique), la domotique est un ensemble de technologies qui permettent de contrôler et d'effectuer la centralisation et l'automatisation des différents systèmes de la maison tels que la sécurité, la gestion d'énergie et la communication afin de la rendre intelligente et connectée [8].

Maison connectée :

Une maison connectée est un domicile doté d'un système domotique visant à faciliter et à améliorer la qualité de vie de ses occupants, elle leur facilite les tâches quotidiennes puisqu'elle

peut être pilotée à distance à l'aide d'un ordinateur ou d'un simple smart phone et tablette. La seule chose requise est une connexion Internet. Ainsi il est possible de gérer son chauffage, la sécurité le verrouillage et la fermeture des portes de la maison et ce en restant derrière son écran [8].

Maison intelligente :

Une maison intelligente est dotée de toutes les fonctionnalités d'une maison connectée sauf qu'à l'exception de cette dernière une maison intelligente est entièrement autonome et n'a pas besoin d'être pilotée, elle est programmée de sorte à s'adapter aux exigences de ces occupants sans que ces derniers aient à la commander à chaque fois. Elle peut entreprendre certaines Actions tel l'allumage du chauffage, l'ouverture des portes et leur verrouillage peut suivre un certain timing ou peut se faire par événement selon la configuration [8].

1.6.1 Domaines d'application des maisons connectées [2] :

1. La sécurité :

La maison intelligente permet d'améliorer la sécurité des utilisateurs et ce par exemple via la mise en place des détecteurs de fumées pour prévenir les incendies, installation des détecteurs de présence et des caméras pour réduire les risques d'intrusion. En effet, plusieurs dispositions peuvent être avantageuses pour leur sécurité.

2. La santé :

Cette application de domotique est essentiellement dédiée au suivi des personnes à besoins spécifiques et fragiles (par exemple les personnes à mobilité réduite, âgées, malentendants... etc.).

Il peut s'agir d'équipements installés sur une personne et/ou dans le domicile qui contrôle un certain nombre de paramètres tels que le rythme cardiaque, taux de glycémie, température corporelle ou autres. Il peut aussi y avoir un capteur sensoriel qui permet de détecter les pertes de connaissances. De ce fait, en fonctions des résultats obtenues par les capteurs, il sera possible d'alerter les organismes compétents et/ou les contacts d'urgence.

3. Le confort :

La maison intelligente permet d'améliorer le confort des gens en mettant en place des scénarios en rapport avec leurs modes de vie. Cela permettra non seulement de régler leurs consommations, mais aussi de mettre en place des dispositions en fonction de leur présence dans leur maison.

4. Les économies d'énergie :

Tout le monde rêve de réduire au maximum sa consommation en énergie. C'est pour cette raison que chaque maison met en place quelques dispositions qui permettent aux utilisateurs de consommer moins. C'est désormais possible grâce à la domotique.

Grâce aux automatismes et aux capteurs, les équipements électriques inter reliés pilotent au plus juste la consommation énergétique (chauffage, éclairage, eau, ventilation, etc.), tout en gardant sous contrôle le confort des zones occupées. Le thermostat connecté peut leur permettre :

- de régler la température de chaque pièce,
- d'éteindre ou d'allumer un chauffage à distance,
- de suivre leur consommation par appareil.
- réguler l'éclairage de chaque pièce à partir de leur Smartphone.

5. L'autonomie :

Une maison intelligente doit être en mesure de répondre aux besoins des individus qui pourraient tirer profit de ces innovations qui peuvent être [10] :

- Des personnes âgées ou fragiles.
- Personnes qui vivent seuls et souffrent de maladies chroniques et ont besoin d'une surveillance continue.
- Les personnes vivant dans des zones urbaines où les services de santé sont inadéquats.
- Les personnes impliquées dans les soins de télésanté entreprenant des soins à distance ou par télémedecine avec des médecins pratiquant des visites virtuelles.

Pour but de faciliter leur vie quotidienne : faire des rappels lorsqu'ils doivent prendre les médicaments, alerter les services d'urgences en cas d'accident, faciliter la mise en contact avec leurs proches et le monde extérieur, etc. En effet, ces personnes ont besoin de vivre de façon autonome et en bon état de santé dans le confort et la sécurité, sans devoir quitter leurs demeures et se déplacer dans des institutions spécialisées à coût de séjours élevé ou limité généralement en nombre de places.

Contraintes des maisons intelligentes :

Une maison intelligente demande un investissement conséquent. L'achat des équipements, leur mise en place, leur entretien et leur gestion nécessitent généralement un coût très élevé qui constitue un inconvénient. Par ailleurs, une maison connectée dépend entièrement de l'outil internet et la sécurité accrue qu'elle offre peut-être remise en cause. Un hacker peut en effet prendre le contrôle de la maison de l'habitant ou voler ses données, s'il réussit à infiltrer sa maison. Autre inconvénient, une panne de réseau sur le Smartphone, une coupure de courant, le vol de son téléphone, et c'est la catastrophe assurée.

1.7 Conclusion :

Ce chapitre a fait l'objet d'une étude sur le concept IoT. Nous avons présenté quelques définitions se rapportant à ce dernier, ainsi que les caractéristiques et composants des systèmes IoT leurs domaines d'application ainsi que leurs avantages, inconvénients et limites.

En outre, nous avons pu étudier la communication avec le protocole de transfert de données MQTT. Enfin nous avons abordé certains points sur la domotique.

Chapitre 2

Conception du système de mise à niveau d'équipements classiques.

2.1 Introduction :

Dans tout projet, la phase de conception est indispensable. Elle permet de fixer les méthodes ainsi que les choix techniques à adopter lors de la réalisation. Elle décrit la solution apportée et sert de support à l'implémentation et la maintenance du projet.

Dans ce chapitre nous présenterons les objectifs de notre projet ainsi que l'architecture globale du système.

2.2 L'objectif du projet :

Au jour d'aujourd'hui, de plus en plus de sociétés d'objets connectés voient le jour, même les anciennes font une mise à niveau et se mettent, elles aussi, à fabriquer de produits de manière à ce qu'ils soient connectés et/ou intelligents dès leur sortie de l'usine. Par conséquent, le marché actuel est rempli d'objets connectés en tous genres.

De ce fait, une personne possédant des équipements domestiques classiques se verra dans l'obligation d'acheter de nouveaux équipements connectés, dans le cas où elle veut bénéficier des avantages des objets connectés.

A travers ce projet, nous avons pour mission d'apporter une solution permettant à l'utilisateur de profiter des atouts et confort qu'offrent les objets intelligents/connectés sans qu'il ait à changer les équipements déjà acquis.

Notre travail consiste à concevoir et réaliser une solution permettant de mettre à niveau les équipements classiques, disposants de moyens de contrôle rudimentaires (télécommande infrarouge par exemple) pour qu'ils soient connectés/ intelligents afin de pouvoir communiquer avec une centrale domotique. Cela est réalisé, en équipant les objets d'émetteur infrarouge contrôlé par un module IoT ESP32. En plus de cela nous utiliserons une carte Raspberry Pi 3 connectée à internet qui assurera le rôle d'une centrale domotique, cette dernière sera connectée à une base de données temps réel Firebase à partir de laquelle elle pourra gérer les données liées aux objets. Par ailleurs, nous concevrons une application Android connectée elle aussi à Firebase, ce sera à travers cette dernière que l'on pourra commander les objets en ayant accès à leur état actuel. Nous ajouterons également de nouveaux équipements intelligents conçus à base de capteurs (un capteur de mouvements par exemple) interfacés à des ESP32 qui vont s'intégrer et fonctionner en harmonie avec les équipements existants.

La figure 2.1 représente la situation des objets classiques qu'on souhaite mettre à niveau. Les figure 2.2 et 2.3 représentent la situation future.

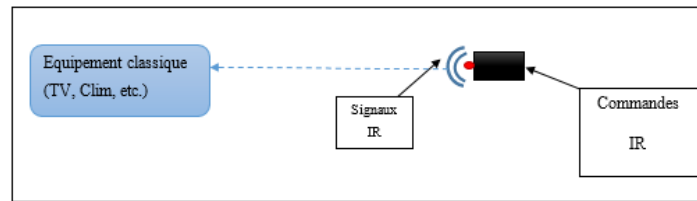


FIGURE 2.1 – Situation des objets classiques

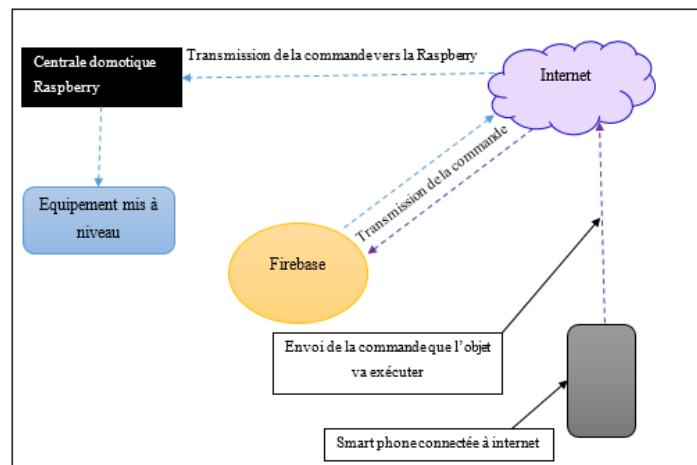


FIGURE 2.2 – Situation future des objets

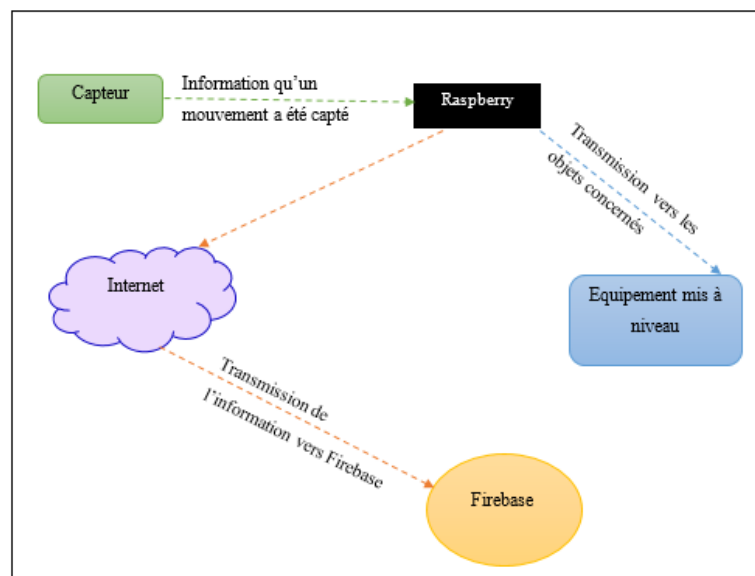


FIGURE 2.3 – Situation finale des objets

2.3 Description du fonctionnement global :

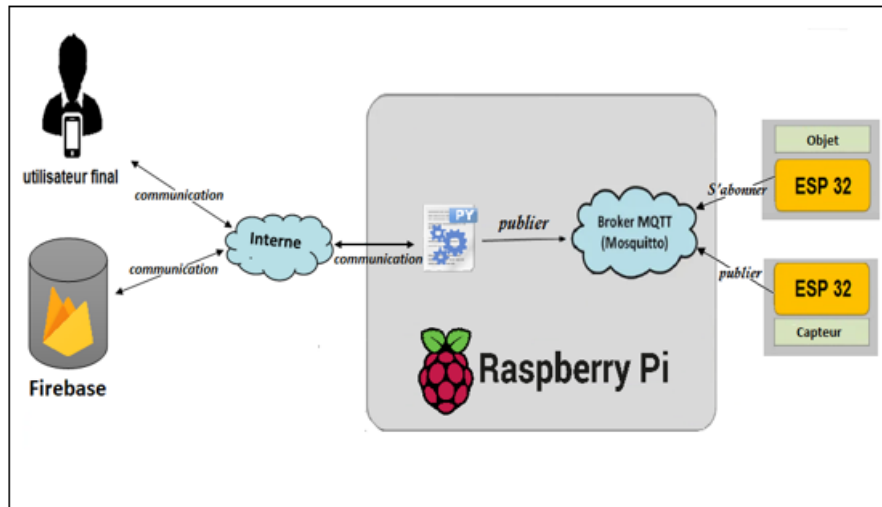


FIGURE 2.4 – Schéma du fonctionnement global

La figure 2.4 représente l'architecture globale de notre système.

Pour pouvoir commander les objets de la maison, l'utilisateur n'aura qu'à ouvrir une application Android à partir de son smartphone. A travers l'application il aura accès à l'état de chaque objet en temps réel et au moment où il envoie une commande à un objet elle est directement transmise vers la base de données temps réel de Firebase via un réseau Internet. Sachant qu'au niveau de l'application, chaque objet aura une interface dédiée simulant les boutons de la commande infrarouge associée et les commandes de cette dernière sont implémentées au niveau des cartes ESP32 liées aux objets.

Dès que le message arrive à Firebase, elle le transmet vers la Raspberry toujours via internet. Une fois que la Raspberry a réceptionné le message elle le dirige vers l'objet concerné.

En interne, la communication entre les cartes ESP32 et la Raspberry se fait suivant le protocole MQTT où les cartes ESP32 sont des Subscribers ou Publishers et la Raspberry est le Broker MQTT.

En ce qui est des capteurs, au moment où un mouvement est détecté une publication est faite dans un des topics du broker par l'ESP32 du détecteur. L'ESP32 de l'objet qui est en écoute de ce topic récupérera l'information publiée et une copie de cette information est envoyée vers Firebase qui la transmet vers l'appareil de l'utilisateur afin que ce dernier soit mis au courant de l'état actuel des objets.

2.4 Description des schémas internes et processus au niveau des cartes ESP32 :

2.4.1 Au niveau de l'objet :

Schéma interne :

Les objets qu'on souhaite mettre à niveau seront équipés d'un module de carte ESP32. Chacune de ces cartes est un Subscriber MQTT abonné à deux Topics au niveau du Broker MQTT à travers lequel il récupère son état « voulu ». Le premier Topic est celui où son état « voulu » par l'utilisateur est publié. Le deuxième est celui où les détecteurs de mouvement associés publient l'information qu'un mouvement a été détecté. De ce fait, au moment où un message est publié dans un des Topics l'abonné, qui est en écoute sur ce dernier, y aura directement accès.

Une fois qu'un message est récupéré depuis le Broker, un traitement est effectué au niveau de l'ESP32. Ce dernier consiste à faire correspondre le « nom » de l'« état voulu » de l'objet avec le code hexadécimal de la commande infrarouge associée. A l'issue de ce traitement, un code hexadécimal est transféré vers l'émetteur infrarouge qui, à son tour, l'envoie sous forme de signaux IR vers le récepteur contenu dans l'objet. Les codes hexadécimaux des commandes infrarouges sont récupérés à l'aide du Reverse Engineering que nous expliquerons un peu plus loin.

La figure 2.5 ci-dessous représente l'architecture interne de l'objet

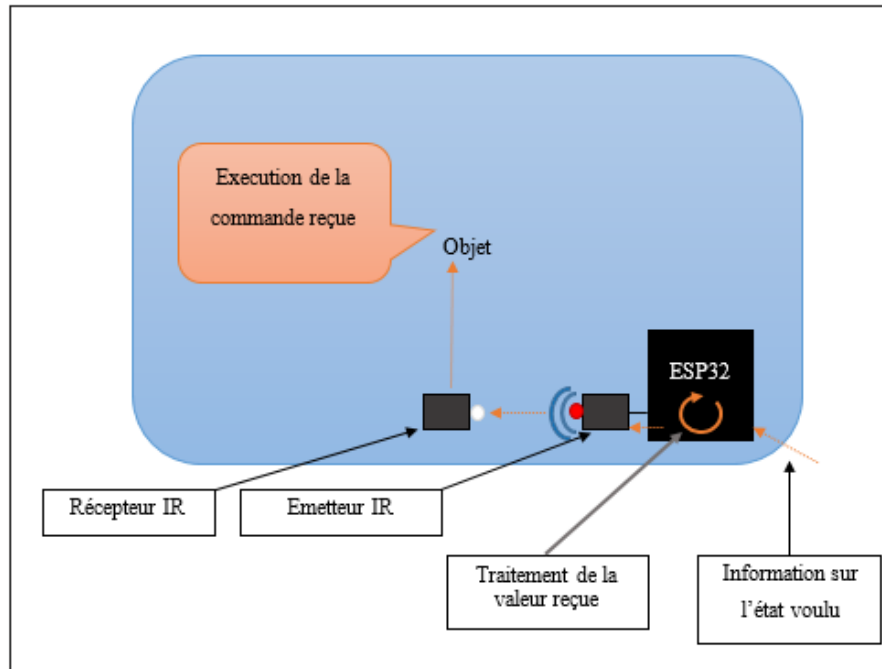


FIGURE 2.5 – L'architecture interne de l'objet

Les processus :

Au niveau des ESP32 nous avons implémenté trois processus sous forme de tâches FreeRTOS.

La première tâche se présente en deux parties. La première est chargée d'établir les connexions requises par la carte, à savoir la connexion wifi ainsi que la connexion au serveur MQTT contenu dans le Raspberry (le broker). Elle s'exécute en premier, à chaque démarrage de la carte et vérifie périodiquement s'il y a eu déconnexion auquel cas elle se reconnectera.

La figure 2.6 illustre le fonctionnement de la tâche 1 décrite ci-dessus :

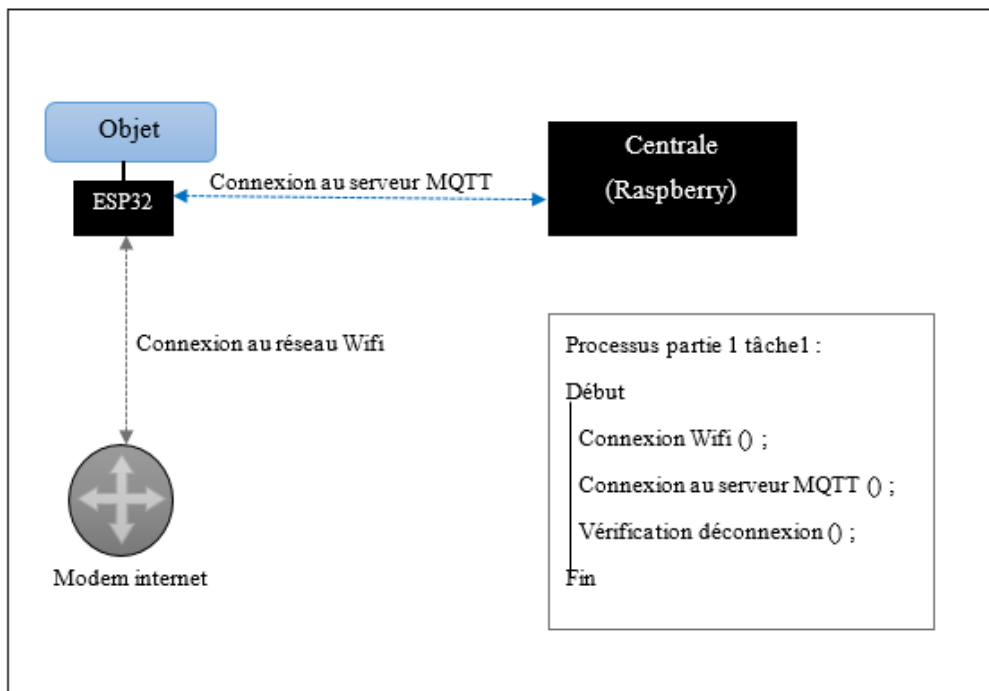


FIGURE 2.6 – Le fonctionnement de la première partie de la tâche 1

Une fois les connexions établies, elle se met en écoute sur les deux canaux (Topics) auxquels la carte ESP32, qui est un subscriber MQTT, est abonnée. Dès qu'un message est publié sur un de ces derniers, elle le prend et le transmet à la deuxième tâche qui prend le relais.

La figure ci-dessous représente un schéma explicatif illustrant le fonctionnement de la deuxième partie de la tâche 1.

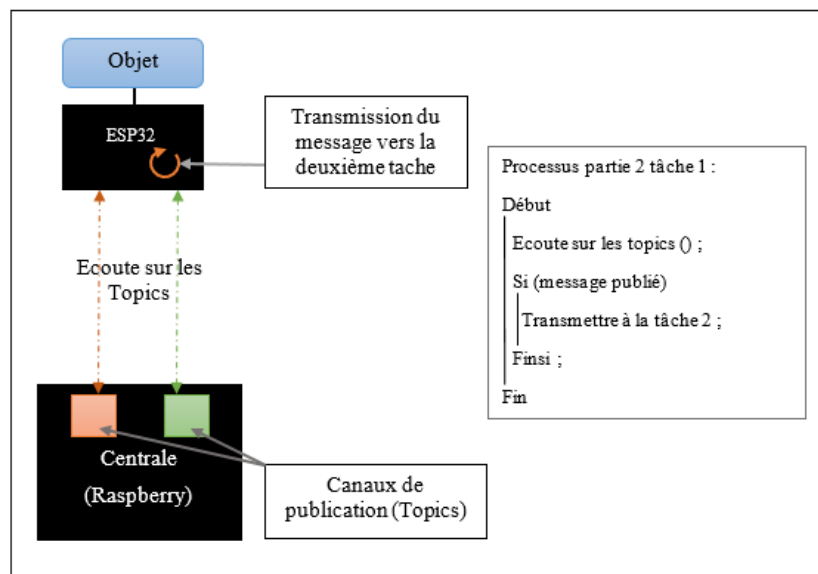


FIGURE 2.7 – Le fonctionnement de la deuxième partie de la tâche 1

Pour la deuxième tâche, comme elle est en collaboration avec la première, elle nécessite une synchronisation afin que tout marche d'une manière fluide. Cette synchronisation est obtenue à l'aide d'une structure de données de type Queue. De ce fait la tâche 1 sera un « Producteur » et la tâche 2 un « Consommateur ». La tâche « Producteur » dépose une donnée représentant l'état voulu de l'objet, par la suite la tâche « consommateur » récupère l'état, le traite et le transmet vers l'émetteur infrarouge. Dans le cas où la queue est vide la tâche 2 est mise en attente et est débloquée au moment où un élément est inséré. Quant au moment où la queue a atteint sa capacité maximal, la tâche 1 est, elle aussi, mise en attente jusqu'à ce qu'au moins une place se libère.

La figure 2.8 illustre le mécanisme de synchronisation décrit ci-dessus.

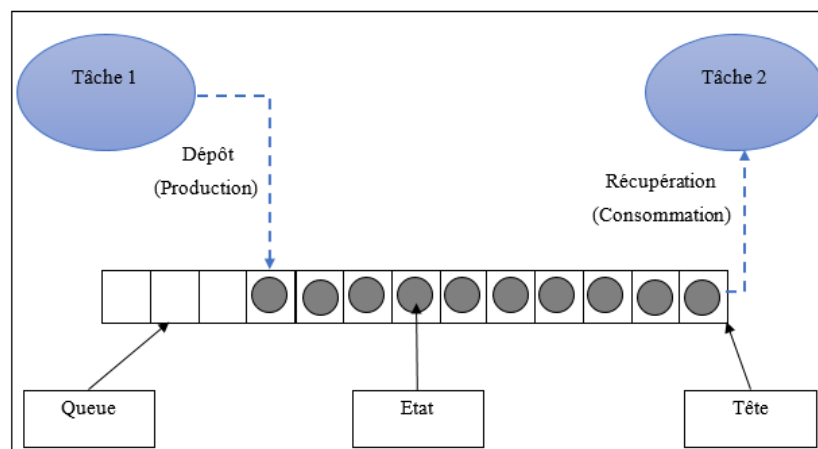


FIGURE 2.8 – Le mécanisme de synchronisation des deux tâches de notre système

Une fois qu'une valeur de l'état voulu est récupérée depuis la queue de synchronisation, un traitement visant à faire correspondre le nom de l'état avec le code hexadécimal associé est

2.4. DESCRIPTION DES SCHÉMAS INTERNES ET PROCESSUS AU NIVEAU DES CARTES ESP32 :

effectué par la tâche. Par la suite, ce code est transmis vers l'émetteur infrarouge lié à la carte qui prend le relais et l'envoie sous forme de signaux vers le récepteur contenu dans l'objet.

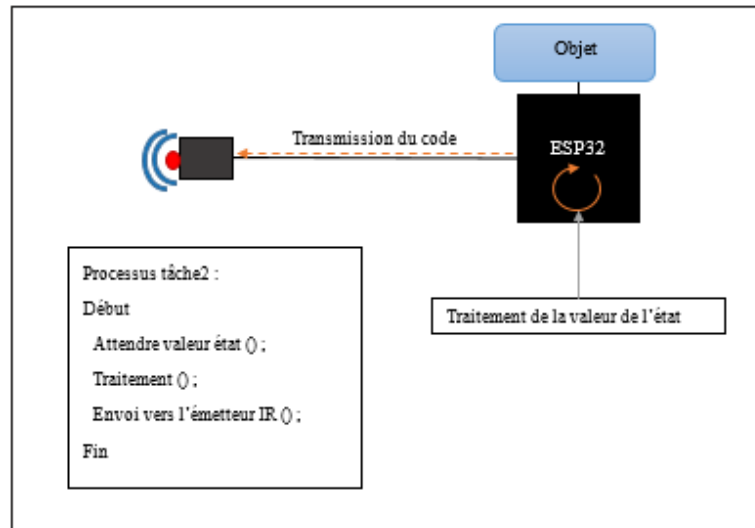


FIGURE 2.9 – Le fonctionnement de la tâche 2 de notre système.

2.4.2 Au niveau du détecteur de mouvement :

Schéma interne :

En ce qui est des capteurs, ils seront équipés d'un module de carte ESP32. Chacune de ces cartes est un Publisher MQTT qui va publier sur un Topic spécifique au niveau du Broker MQTT. A savoir qu'une pièce de la maison peut contenir un à plusieurs capteurs et que l'association entre les objets et les capteurs se fait selon la pièce de la maison où ils sont placés.

Le fonctionnement est relativement simple, au moment où un mouvement est détecté par le capteur, ce dernier envoie un signal à son ESP32. A chaque réception de signal une publication est faite dans le Broker MQTT.

La figure ci-dessous 2.10 illustre le fonctionnement que nous venons de décrire.

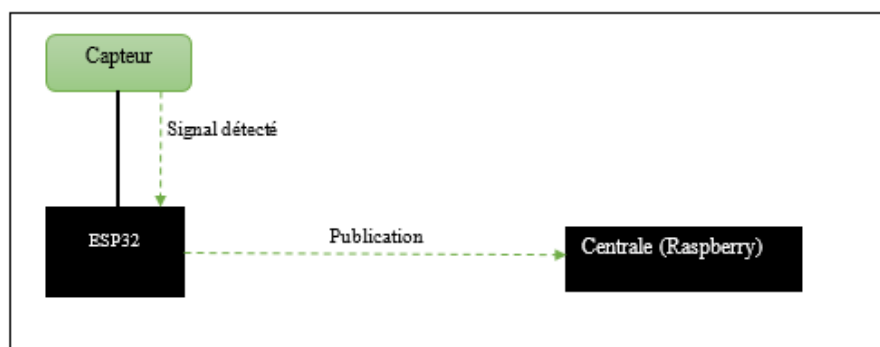


FIGURE 2.10 – Schéma du fonctionnement au niveau de capteur .

Processus :

Au niveau des ESP32 nous avons implémenté deux processus sous forme de tâches FreeRTOS.

Comme pour les ESP32 des objets, la première tâche implémentée est celle chargée d'établir les connexions requises par la carte, à savoir la connexion wifi ainsi que la connexion au serveur MQTT contenu dans le Raspberry (le broker). Elle s'exécute en premier, à chaque démarrage de la carte et vérifie périodiquement s'il y a eu déconnexion auquel cas elle se reconnectera.

La figure qui suit illustre le fonctionnement de la tâche 1 décrite ci-dessus.

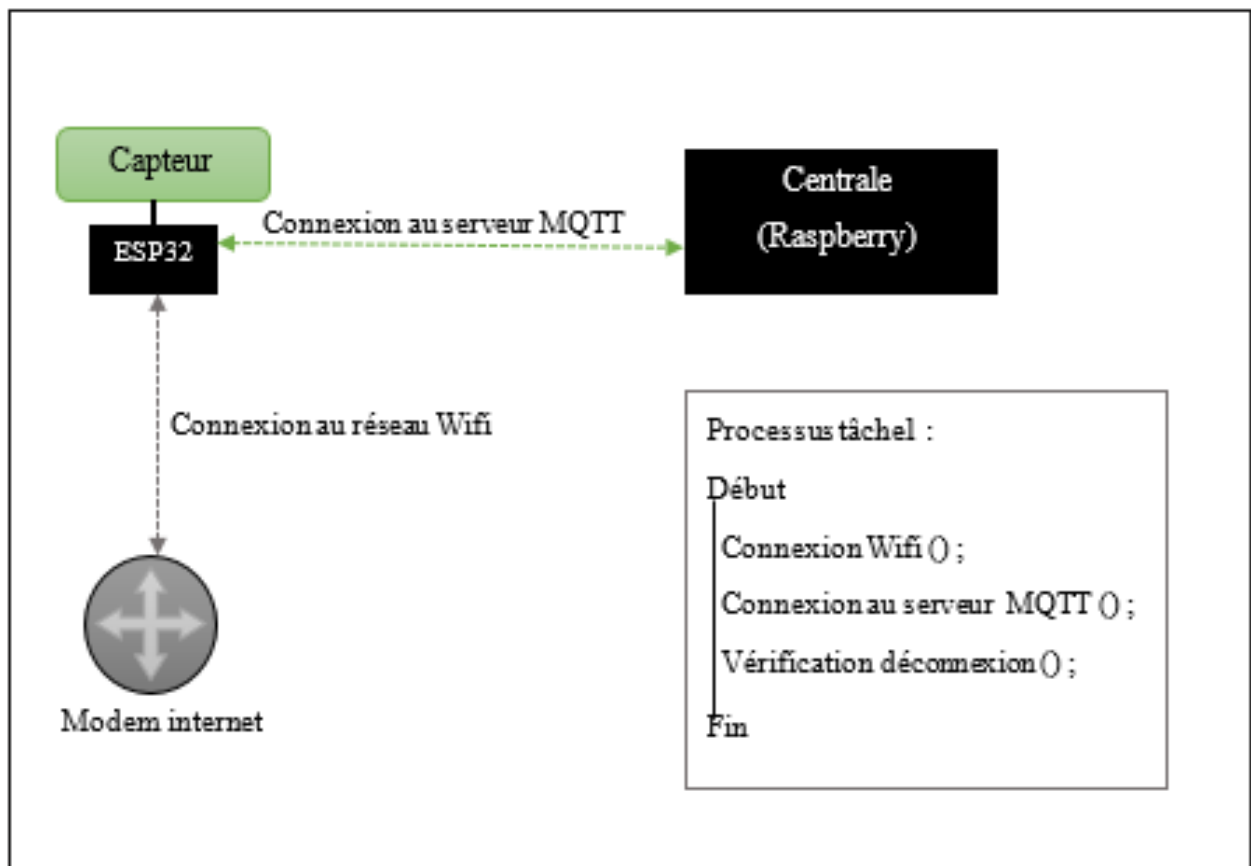


FIGURE 2.11 – Illustration du fonctionnement de la tâche 1

La deuxième tâche est chargée d'effectuer la publication un Topic spécifique au niveau du Broker. A son lancement elle est mise en attente d'un évènement, cette synchronisation est réalisée grâce à un sémaphore binaire. De ce fait, au moment où un mouvement est détecté le signal reçu par le capteur enclenche une interruption, cette dernière débloque le sémaphore grâce auquel la tâche 2 est libérée ce qui donne lieu à une publication.

La figure suivante illustre le fonctionnement de la tâche2 que nous venons de décrire

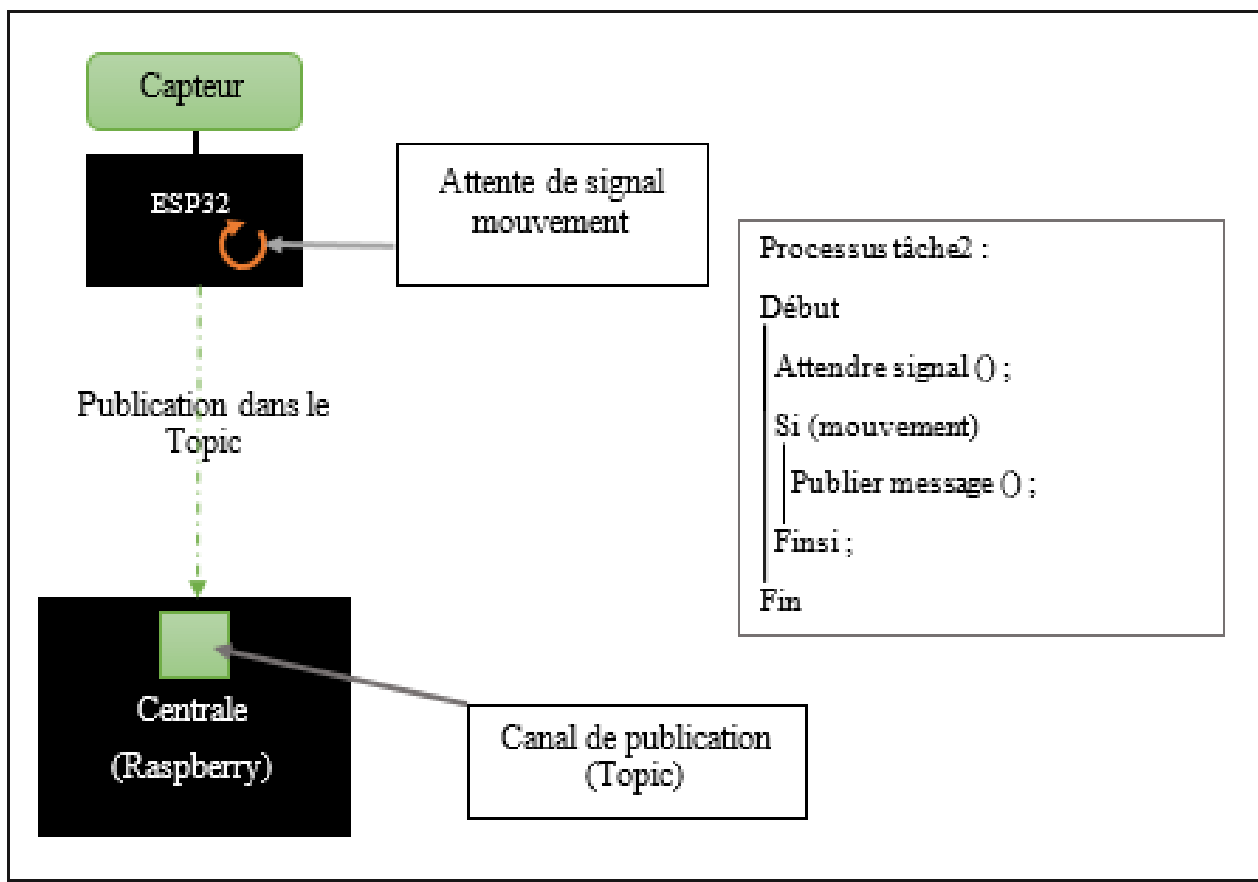


FIGURE 2.12 – L’illustre de fonctionnement de la tâche2

2.5 Description du fonctionnement interne de la Raspberry Pi 3 :

Au niveau de la carte Raspberry Pi 3, nous avons installé un broker MQTT (Mosquitto) qui permet de réceptionner les données provenant des publier dans leur topics correspondants et de les transmettre aux subscribers qui sont à l’écoute de ces derniers.

La Raspberry communique avec Firebase par le biais d’un script python qui est un publisher/subscriber. D’une part, il permet de réaliser une lecture à partir de la base de données et de la publier sur le topic lié à l’objet concerné. D’une autre part, il est à l’écoute des topics liés aux détecteurs, de ce faite au moment où une information de capture de mouvement est publiée il la transmet vers Firebase.

2.5. DESCRIPTION DU FONCTIONNEMENT INTERNE DE LA RASPBERRY PI 3 :

La figure suivante illustre ce que nous venons d'expliquer.

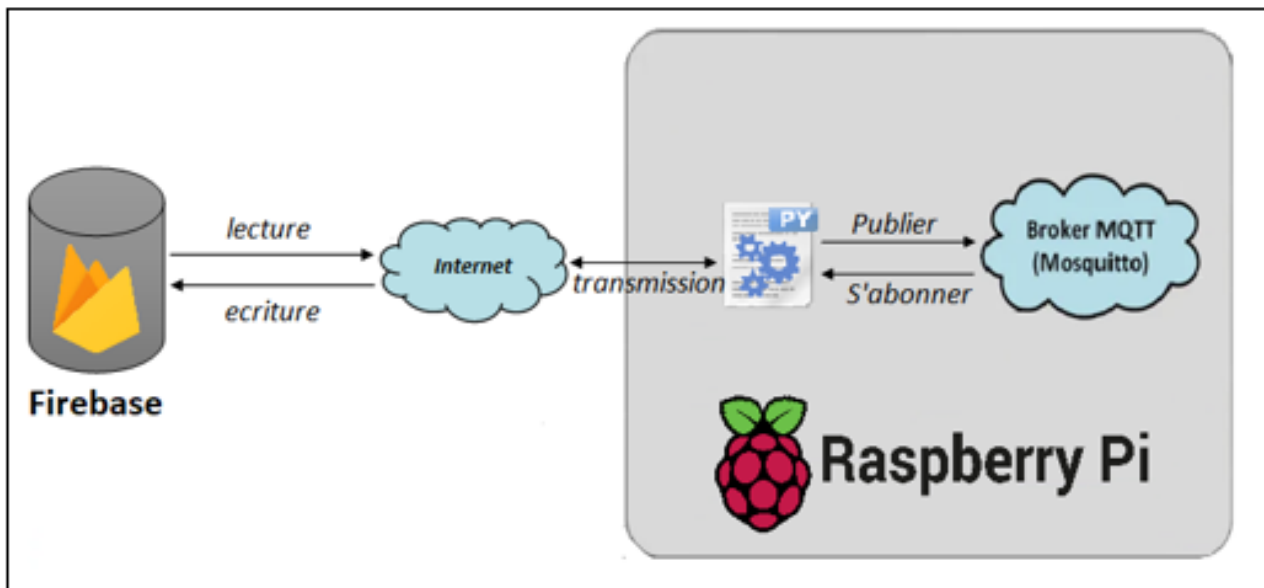


FIGURE 2.13 – Schéma du fonctionnement au niveau de la raspberry.

2.6 Description de fonctionnement de l'application Android :

La figure suivante représente le diagramme d'activité de notre application Android.

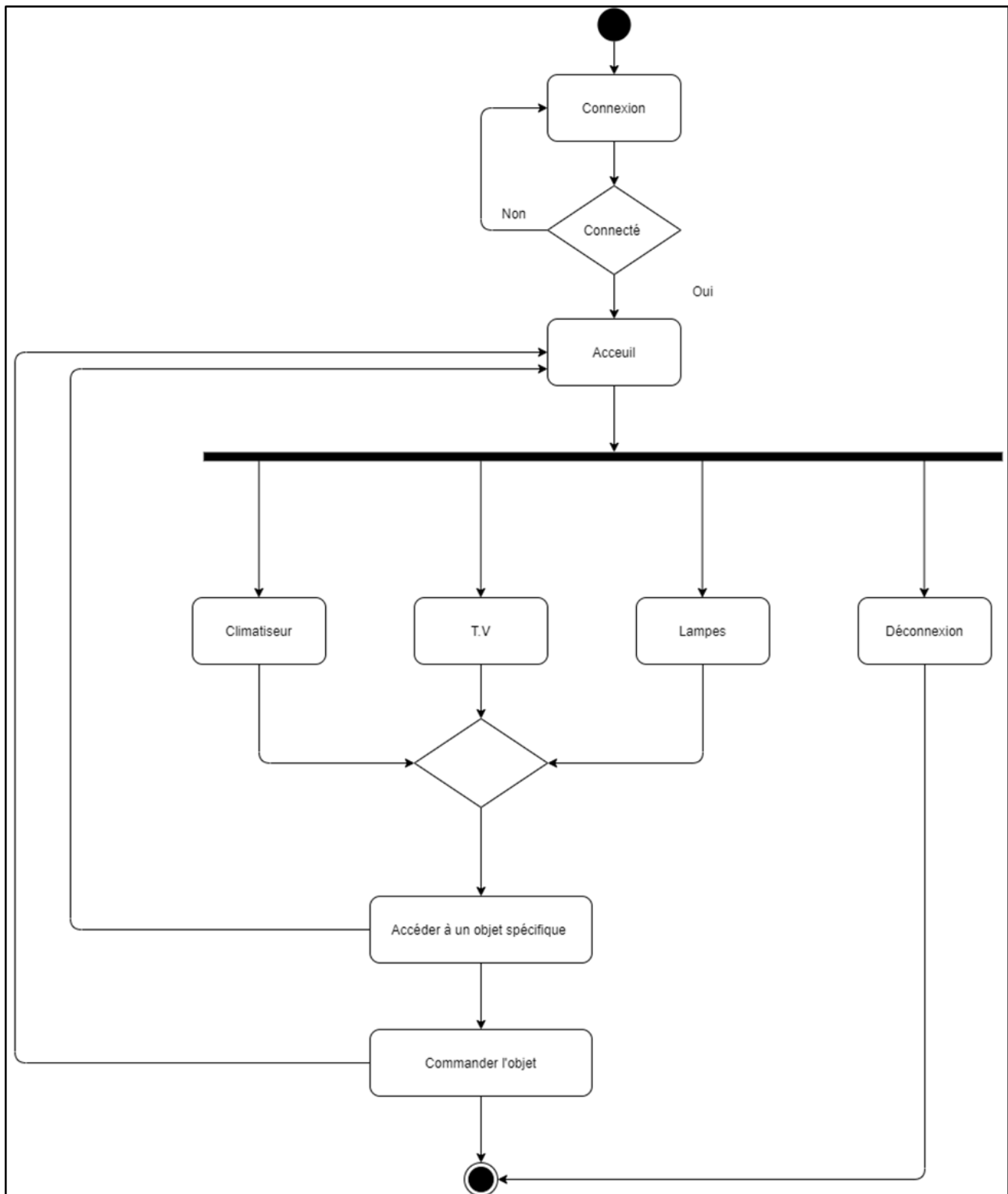


FIGURE 2.14 – Diagramme de l'activité de l'application .

Interprétation de la figure :

L'application comportera un nombre indéfini d'activités puisqu'elle est faite suivante le nombre et types d'équipements que l'utilisateur possède.

Nous pouvons classer ces activités en quatre catégories.

- La première est l'activité d'authentification.
- La seconde est celle qui représente l'accueil.
- La troisième est celle qui est dédiée à chaque type d'équipement par exemple pour le type « Lampes », l'interface de cette dernière affichera tous les équipements contenus dans ce type.
- La quatrième et dernière catégorie est celle dédiée à chaque équipement spécifique par exemple la « Lampe 1 » située dans la cuisine.

Dans un premier temps, l'utilisateur doit s'authentifier pour pouvoir accéder à l'interface d'accueil de l'application.

Une fois l'authentification faite il peut choisir d'entrer dans une catégorie d'objets et ainsi pouvoir accéder à un objet spécifique par exemple accéder à la lampe de sa chambre contenue dans la catégorie « Lampes ».

A savoir, que les catégories présentes dans le schéma ne sont pas les seules auxquelles la solution de notre projet s'applique, elle s'étend à n'importe quel objet qui est contrôlable via une commande infrarouge tel que les volets électriques, les portails, système de chauffage, etc.

A partir de là, il peut commander l'état de l'objet choisi et/ou consulter son état actuel tout en sachant qu'à tout moment il peut revenir vers l'interface d'accueil. Aussi, il peut choisir de directement se déconnecter et ainsi sortir de l'application.

2.7 Le reverse engineering :

Le reverse engineering (ingénierie inverse ou retro-ingénierie en français) est une méthode ou processus par lequel une technologie ou un objet (matériel ou logiciel) est décomposé et déconstruit dans le but de pouvoir l'étudier, comprendre son fonctionnement interne, son architecture, son code ou pour extraire certaines informations voulues et ce en faisant le chemin inverse de sa construction.

La figure suivante illustre le principe du reverse engineering.

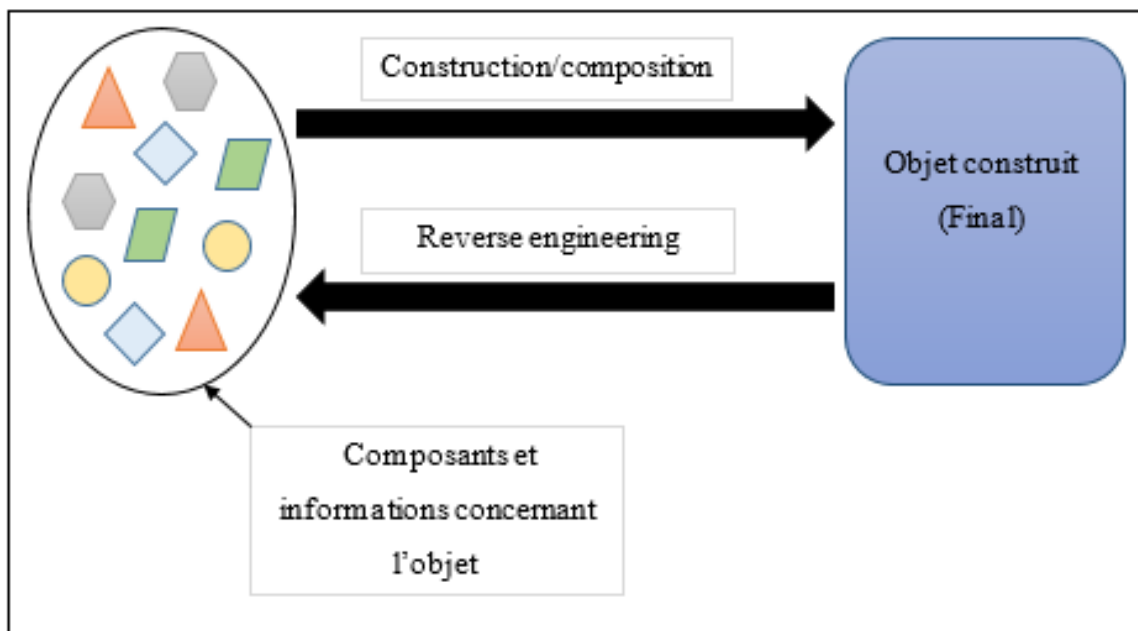


FIGURE 2.15 – L'illustration de principe de reverse engineering

Dans notre cas il s'agira d'intercepter et décoder les signaux infrarouges émis par la commande vers l'objet correspondant, afin de pouvoir extraire les commandes et informations contenues dans ces derniers dans le but de pouvoir les implémenter par la suite sur une carte électronique ESP32.

2.8 Conclusion :

Dans ce chapitre, nous avons présenté tous les détails de la conception de notre système. Dans un premier lieu, Nous avons présenté le principe de notre solution et l'architecture globale du système. Par la suite, nous avons expliqué le détail de chaque partie du projet.

Dans le chapitre suivant, nous présenterons les détails de la réalisation.

Chapitre 3

Réalisation du système

3.1 Introduction

La réalisation est la dernière étape qui clôture notre projet et c'est sure cette dernière que porte ce chapitre. Dans la première partie de ce chapitre nous allons voir l'environnement de développement et les différents outils matériels et logiciels utilisés pour réaliser notre projet.

Dans la deuxième partie nous présenterons les différentes étapes de réalisation avec des photos et des captures d'écran.

3.2 Présentation de l'environnement de développement :

3.2.1 Présentation de l'environnement matériel :

Pour la réalisation de notre projet nous avons utilisé les équipements suivants :

ESP32 :

Présentation générale :

ESP32-WROOM-32 est un module puissant fabriqué par Espressif qui cible une grande variété d'applications, allant des réseaux de capteurs à faible puissance aux tâches les plus exigeantes, telles que l'encodage vocal, la diffusion de musique et le décodage MP3.

Au cœur de ce module se trouve la puce ESP32-D0WDQ6 * qui est conçue pour être évolutive et adaptative. Il existe deux cœurs de processeur qui peuvent être contrôlés individuellement. L'ESP32 intègre un riche ensemble de périphériques, allant des capteurs tactiles capacitifs, interface de carte SD, Ethernet, SPI haute vitesse, UART, I²S et I²C. [[Y.S17](#)]

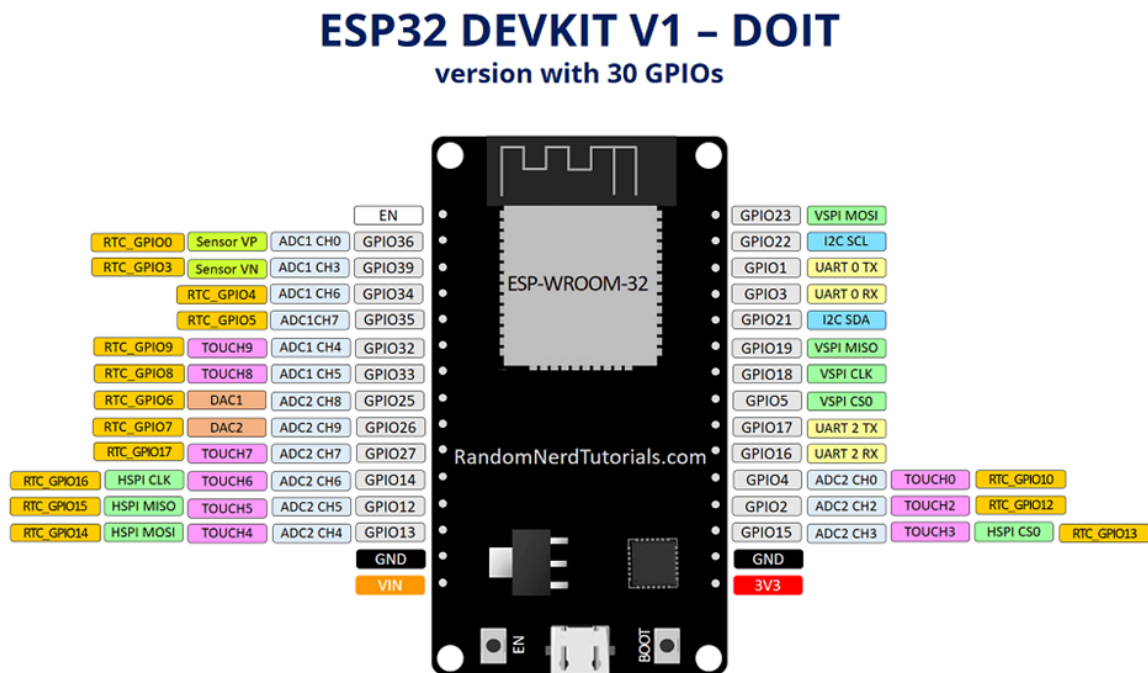


FIGURE 3.1 – Les différents Pins de l'ESP32

3.2.2 Caractéristiques : [Fra14]

- Dimensions : 18 mm x 25.5 mm x 3.1 mm.
- Catégorie du produit modules Wi-Fi (802.11).
- Protocole pris en charge pour le wifi 802.11 b/g/n.
- Fréquence de 2.4 GHz à 2.5 GHz.
- Débit de données de 150 Mb/s.
- Tension d'alimentation de fonctionnement entre 2.7 V et 3.6 V.
- Température de fonctionnement entre - 40 C et + 85 C.
- Sécurité : WPA, WPA2, WPA2-Enterprise, WPS.
- Sensibles à l'humidité.
- Fréquence d'horloge réglable de 80 MHz à 240 MHz.

3.2.3 Raspberry pi 3 modèle b+ :

Raspberry pi est un nano-ordinateur à processeur ARM qui peut être branchée à plusieurs périphériques tels que : la souris, le clavier, l'écran Il est capable d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian, Debian ...) et d'autres logiciels compatibles. Le but original était de redonner un souffle à l'apprentissage de l'informatique aux enfants avec l'IDE Scratch et Python. Mais actuellement, il est utilisé dans divers domaines tels que l'électronique et la robotique grâce aux GPIO, il peut aussi être utilisé comme un serveur ou un hébergeur web.

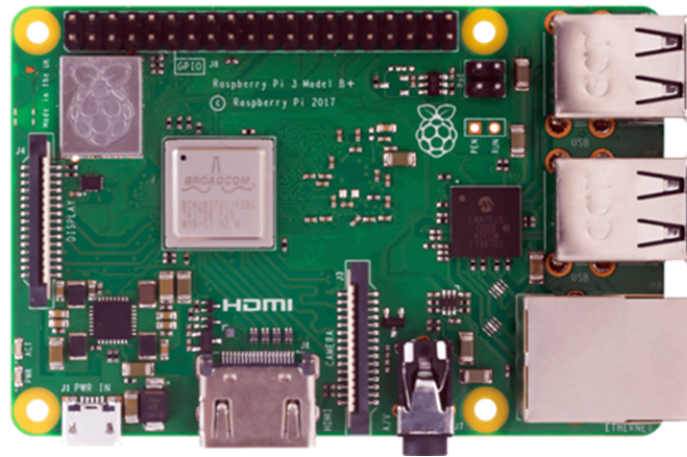


FIGURE 3.2 – le raspberry pi 3 modèle b+

Caractéristiques [4]

Voici les caractéristiques les plus importantes :

- Carte mère Raspberry Pi 3 type B+
- Processeur Broadcom BCM2837B0 Cortex-A53 64-bit SoC @ 1,4 GHz
- RAM : 1 Go LPDDR2 SDRAM
- Connexions sans fil : Bluetooth 4.2 BLE, Wi-Fi Dual Band b/g/n/ac
- Connexion filaire : Gigabit Ethernet
- Décodage et encodage H.264 1080p30
- Support PoE (PoE HAT non fourni)
- Lecteur de carte micro-SD
- Header GPIO 40 broches
- Support du boot PXE
- Port caméra CSI pour connecter la caméra Raspberry Pi
- Port d'affichage DSI pour connecter l'écran tactile Raspberry Pi
- 4 x USB 2.0
- Compatible avec les accessoires du Pi 3 Model B
- Alimentation 5V / 2.5A (transformateur non fourni)

Arduino UNO : [Gui14]

C'est une plateforme de prototypage électronique open-source, basée d'une part sur du matériel et d'autre part sur un ensemble de logiciels faciles à utiliser. Destiné aux inventeurs, artistes ou amateurs qui souhaitent créer leur propre système automatique en le codant de toute pièce.



FIGURE 3.3 – Arduino UNO

Emetteur IR :[5]

Ce composant permettant de transmettre un signal via une LED infrarouge, invisible à l'œil nu.

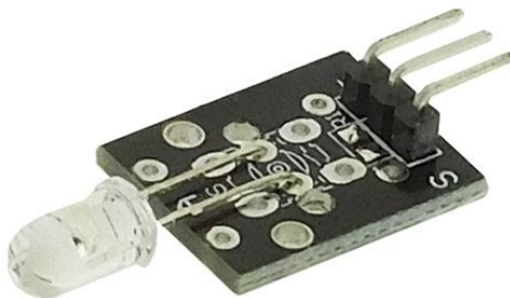


FIGURE 3.4 – L'émetteur IR

Récepteur IR [5] :

dispositifs conçus pour recevoir un signal infrarouge. Pour l'utiliser, il faut connecter la pin 3 une alimentation, la pin du milieu à la masse (GND) et la sortie sera sur la pin 1. Il ne réalise aucun décodage de signal, il transmettra les données brutes. Pour que le signal passe, la distance entre les deux ne doit pas dépasser 10 mètres.

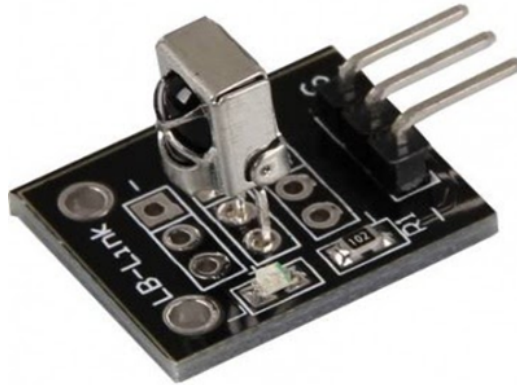


FIGURE 3.5 – Le Récepteur IR

PIR motion Sensor :

Connu également sous le nom de capteurs infrarouges passifs. Il est conçu à partir d'une gamme de capteurs à semi-conducteurs générant une tension quand ils sont exposés à la chaleur. Ils sont utilisés dans des applications de détection thermique, et comprennent des capteurs de sécurité et des détecteurs de mouvement.



FIGURE 3.6 – PIR motion Sensor

3.2.4 Présentation de l'environnement logiciel :

Visual Studio Code et Plateforme IO [9] :

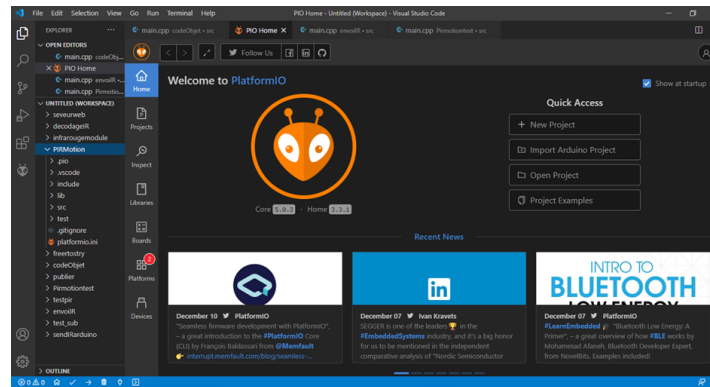


FIGURE 3.7 – Plateforme IO

1. Visual studio code :

Visual studio code est un éditeur de texte, développé par Microsoft, permettant d'écrire des codes. Il supporte plusieurs langages de programmation comme Java, JavaScript, Go, Node.js et C++.

Il prend en charge le débogage, la correction des erreurs syntaxiques, la compilation. Nous pouvons modifier son thème et lui installer des extensions afin d'ajouter des fonctionnalités supplémentaires. Dans notre cas nous lui avons installé l'extension de Plateforme IO.

2. Platform IO :

Platform IO est un environnement de développement open source spécialisé dans l'IOT et qui permet de programmer des applications de systèmes embarqués. Il met à notre disposition une sélection d'outils nécessaires à l'élaboration de logiciels pour les objets. Il facilite l'import des bibliothèques et modules requis dans un programme donné ainsi que l'import des bibliothèques relatives à la carte électronique utilisée, l'utilisateur n'a qu'à choisir la carte et tout le nécessaire sera automatiquement téléchargé.

Android Studio [10] :



FIGURE 3.8 – Logo Android

Android Studio est un outil de développement qui met à notre disposition les outils requis à la création d'applications mobiles Android. Il permet principalement d'éditer les fichiers

Java/Kotlin et les fichiers de configuration XML d'une application Android. Dans notre cas, nous avons conçu les interfaces avec l'XML et nous les avons programmées avec le langage Java.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur d'appareil Android permettant de faire tourner un système Android virtuel sur un ordinateur et offre à l'utilisateur la possibilité de relier son application à une base de données Firebase.

Free RTOS [11] :



FIGURE 3.9 – Logo freeRTOS

Créé en 2003 par Richard Barry et la FreeRTOS Team, sa conception est volontairement minimaliste pour qu'il puisse être installé sur de petits systèmes.

FreeRTOS est généralement désigné comme un système d'exploitation temps réel utilisé dans les systèmes embarqués. En réalité, il s'agit d'une bibliothèque portable contenant des APIs (Application Programming Interface) permettant de développer des applications temps réel.

Elle nous offre des outils logiciels permettant de réaliser de multiples fonctions telles que :

- Encapsuler des traitements dans des tâches indépendantes.
- Ordonnancer les tâches, les faire communiquer entre elles et les synchroniser.
- Fournir des mécanismes de gestion des événements, du temps et des délais, etc.

Elle est de plus en plus utilisée car elle est Open source, libre de modification, de réutilisation, portable sur un grand nombre de processeurs, dispose d'une documentation très abondante, adaptée aux systèmes disposants d'une mémoire limitée. C'est la librairie la plus préconisée pour les projets embarqués temps réel de l'avenir

FireBase [CHE18] :



FIGURE 3.10 – Logo FireBase

Firebase est une base de données NoSQL, créée en 2011 par James Tamplin et Andrew Lee hébergée dans le cloud. Il s'agit d'une plateforme qui permet de stocker et de synchroniser des données entre plusieurs utilisateurs en temps réel, avec une forte sécurité basée sur l'utilisateur. Et de créer des applications sans serveur à l'aide des SDK mobiles et web fournis avec la base de données, mais également exécuter du code backend qui répond aux événements déclenchés par votre base de données à l'aide de Cloud Functions pour Firebase. Firebase optimise l'utilisation hors ligne, en se bénéficiant du cache local sur l'appareil pour diffuser et stocker les modifications, ces dernières sont synchronisées lorsque l'appareil est à nouveau en ligne. La plateforme met à la disposition de ses utilisateurs différents services pouvant être répartie en deux catégories :

- Les outils de développement et de tests de son application
- Les outils permettant d'augmenter et d'engager ses cibles.

Parmi ses outils nous trouvons « Realtime database » et « Firebase Authentification », expliqués dans ce qui suit :

- **Realtime Database :**[\[Alc19\]](#)

Firebase Realtime Database n'est autre qu'une base de données NoSQL, bénéficiant d'un hébergement (Cloud) et permettant le stockage et la synchronisation de données de vos utilisateurs. Les développeurs peuvent gérer cette base de données en temps réel.

Firebase Realtime Database est capable de fournir à votre application la valeur des données et les mises à jour appliquées sur ces dernières avec une simple API. Grâce à la synchronisation en temps réel, les utilisateurs de votre application peuvent consulter leurs données depuis n'importe quel terminal (sur le web ou depuis leur mobile). Notez que cette base de données est livrée avec des SDK mobiles et web et permet la création d'applications sans utiliser de serveurs.

Quand vos utilisateurs passent en mode hors ligne, les SDK de base de données en temps réel utilisent le cache pour enregistrer les modifications. Quand l'appareil est en ligne, les données locales connaissent une synchronisation automatique.

- **Firebase Authentification :**[\[Alc19\]](#)

Cet outil fournit des SDK faciles à exploiter, des services back-end ou encore des bibliothèques d'interface utilisateur. Ces bibliothèques vous permettent d'authentifier vos utilisateurs.

En général, la configuration manuelle d'un système d'authentification prend plusieurs mois. Par la suite, il faut engager une équipe pour la maintenance. Avec Firebase, les choses se déroulent autrement. La configuration du système ne prend que quelques heures même s'il faut prendre en charge des opérations délicates comme la fusion de comptes. Plusieurs méthodes s'offrent à vous pour authentifier vos utilisateurs notamment l'exploitation de :

- Leur e-mail et de leur mot de passe ;
- Google Cloud ;
- Twitter ;
- Facebook ;
- Numéro de téléphone ;

Broker MQTT Mosquitto [6] :



FIGURE 3.11 – Logo MQTT

Mosquitto est un serveur MQTT Open Source (souvent appelé Broker) qui peut être installé sur une carte électronique Raspberry Pi (généralement sous Raspbian) mais aussi sur presque tous les systèmes d'exploitations existants tels que MacOS, Windows, Linux ... etc. MQTT est un protocole de communication très rapide et léger particulièrement adapté à la domotique et aux objets connectés. Il assure une communication entre des objets (M2M) utilisant des technologies différentes tout en économisant la batterie.

Raspbian [12] :

Raspberry Pi OS (anciennement nommé Raspbian) est un OS open source et gratuit basé sur l'OS Debian, qui est une distribution de Unix, optimisé pour fonctionner sur les différents Raspberry Pi. Il s'agit d'une modification de Debian spécifiquement adaptée pour les processeurs de type ARM. Étant donné les ressources limitées des nano-ordinateurs pour lesquels Raspberry Pi OS est principalement destiné, il intègre des logiciels réputés pour être légers et peu gourmands en ressources.

3.3 Présentation des étapes de réalisation :

Dans la réalisation et les tests de notre application nous avons pris pour exemple une lampe qui offre une variété de couleurs qui est contrôlée avec une télécommande infrarouge. La figure suivante représente une photo de la lampe utilisée pour réaliser les tests.



FIGURE 3.12 – La lampe utilisée dans le cadre de notre travail.

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

Pour arriver à la réalisation de notre solution nous sommes passés par plusieurs étapes qui sont comme suit :

3.3.1 Reverse engineering :

Cette étape nous a permis d'intercepter et de décoder les signaux infrarouges émis par la commande vers l'objet correspondant, afin de pouvoir extraire les commandes et informations contenues dans ces derniers dans le but de pouvoir les implémenter par la suite sur une carte électronique ESP32.

Pour réaliser cela, nous nous sommes munis d'une carte électronique de type Arduino UNO à laquelle nous avons relié un équipement récepteur infrarouge. Après avoir écrit le code Arduino permettant de décoder les signaux infrarouge émis par plusieurs types de télécommandes nous l'avons téléversé sur la carte Arduino. Par la suite, nous avons émis les signaux des différents boutons affichés en hexadécimal sur le moniteur série de l'environnement de développement.

Une fois que cela est fait, les valeurs retournées seront ensuite stockées dans un fichier pour pouvoir les récupérer ultérieurement afin de les implémenter sur la carte ESP32 qui sera reliée à l'objet.

La figure suivante schématise comment le décodage des commandes IR est réalisé.

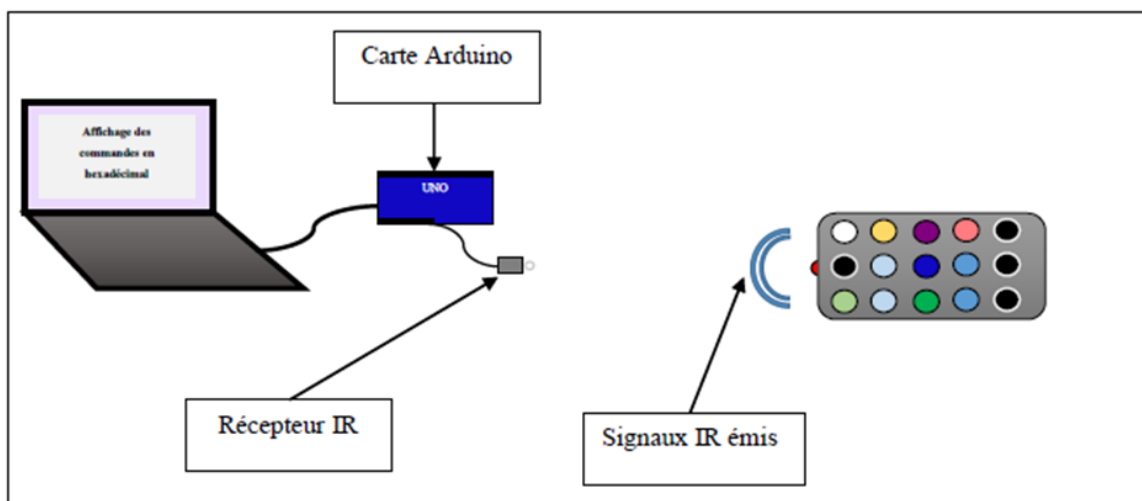


FIGURE 3.13 – Le schéma synoptique des commandes IR.

3.3.2 Préparation de la carte Arduino UNO :

Nous avons mis en place une carte Arduino UNO tel que expliqué auparavant afin de visualiser les données reçus d'un récepteur infrarouge tel que illustré dans la figure 3.14 qui représente le code hexadécimal des commandes infrarouges récupérées et affichées sur le moniteur série de l'environnement de développement Arduino.

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

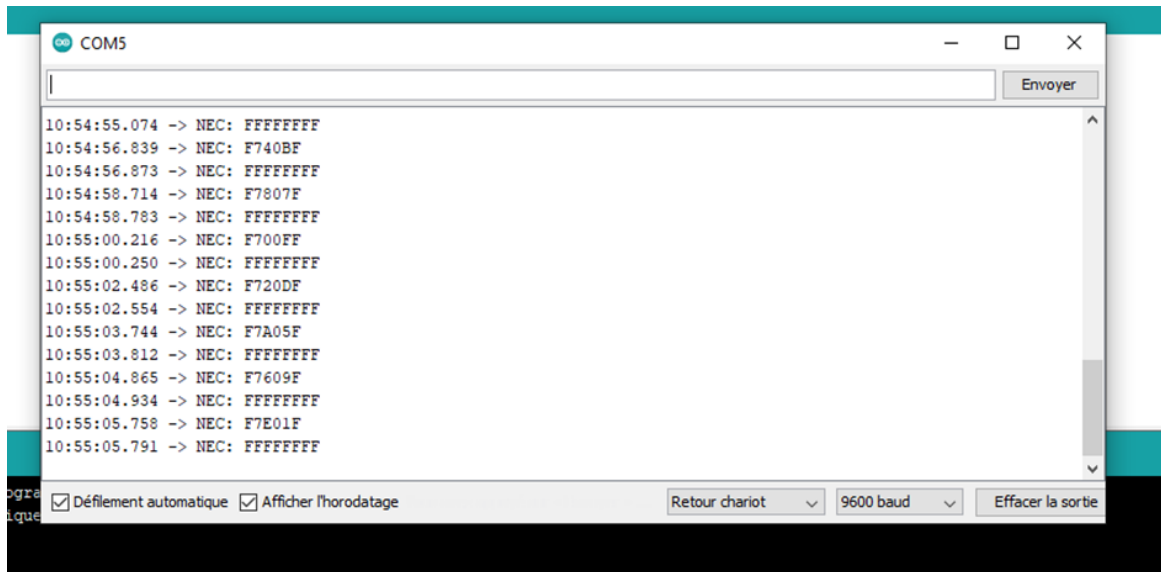


FIGURE 3.14 – L'illustration des commandes infrarouges dans le moniteur série arduino.

La figure suivante représente le schéma de câblage réalisé avec la carte Arduino UNO et le récepteur infrarouge sous fritzing.

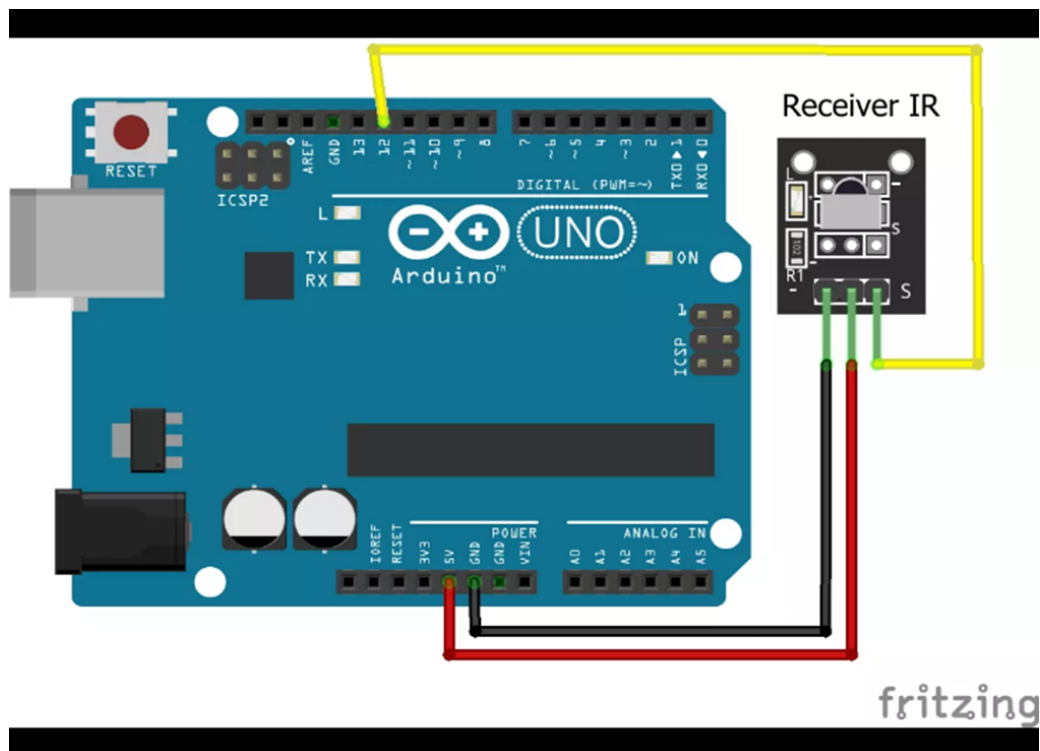


FIGURE 3.15 – Le schéma de câblage de récepteur infrarouge avec Arduino UNO

Les figures (figure 3.16) (figure 3.17) représentent le branchement réalisé avec la carte Arduino UNO et le récepteur infrarouge dans notre travail.

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

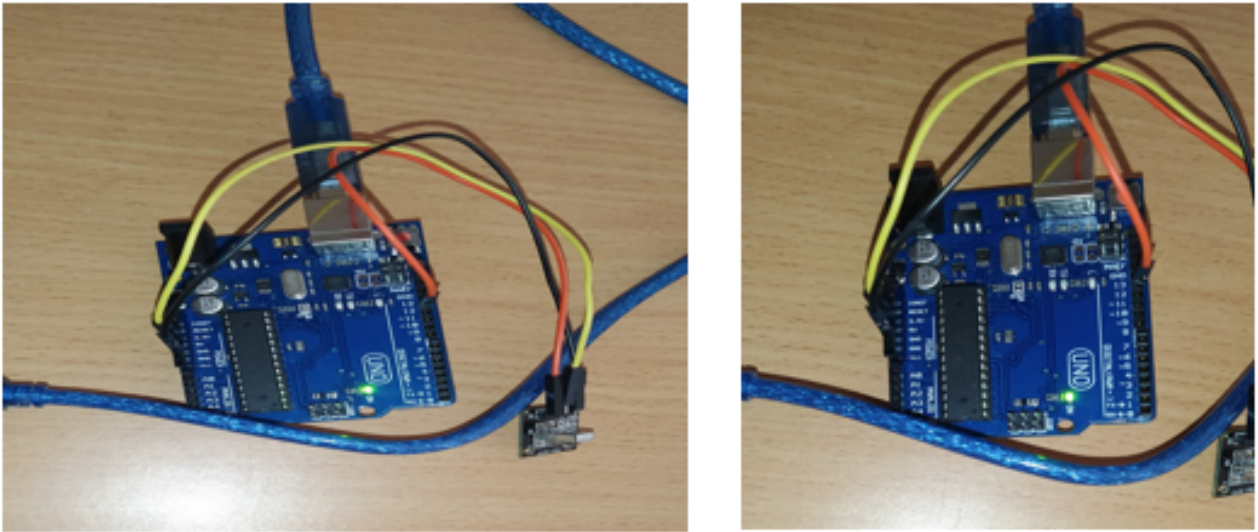


FIGURE 3.16 – Le branchement de récepteur infrarouge avec Arduino UNO .

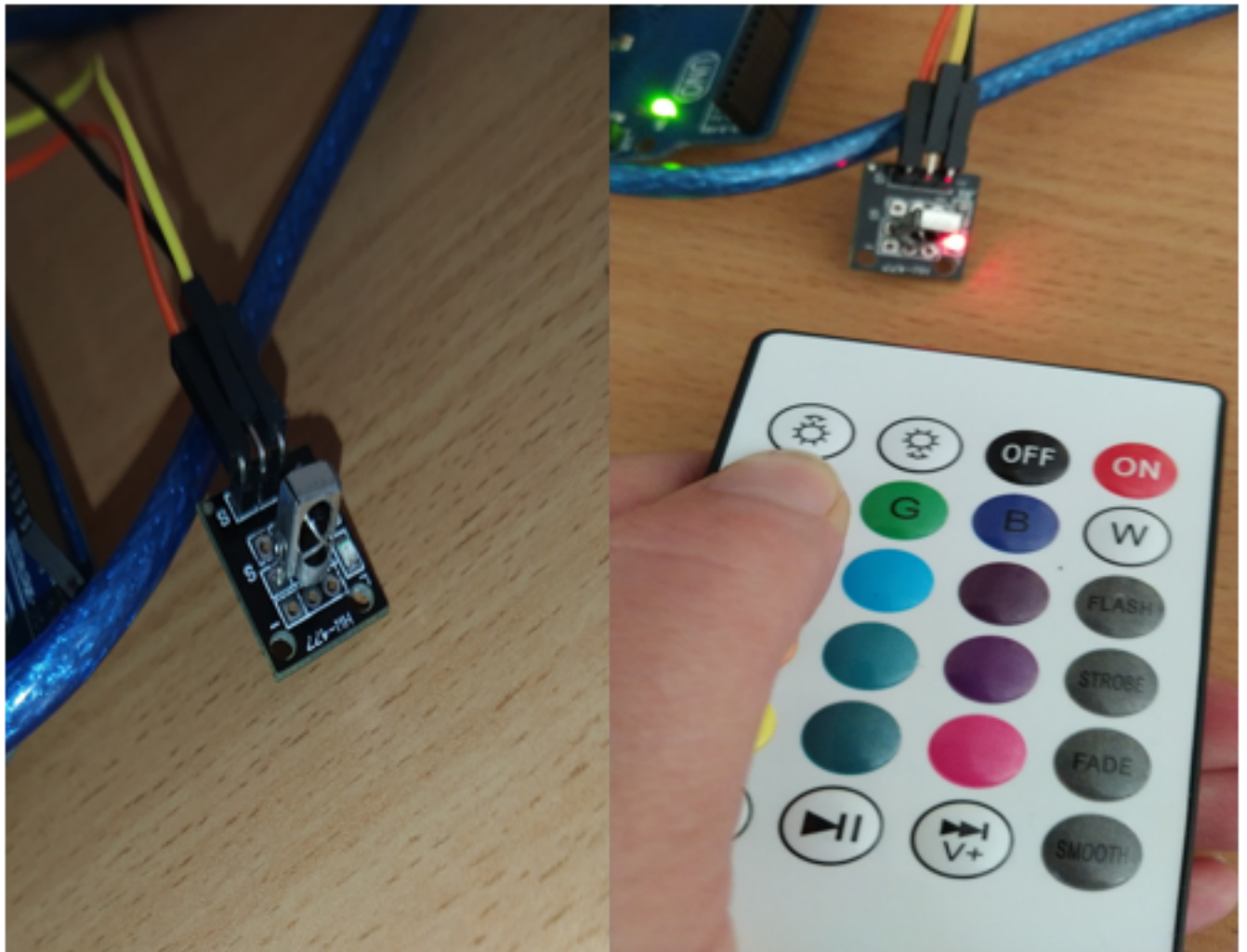


FIGURE 3.17 – La commande de récepteur infrarouge a distance.

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

Notons que nous devons effectuer la même opération pour chacun des équipements que nous souhaitons mettre à niveau tel qu'un téléviseur, un climatiseur, etc. Une alternative à ce procédé serait de consulter la documentation technique des équipements et récupérer les codes infrarouges utilisés. Toutefois, une large gamme d'équipements ne publie pas ces informations sur leurs documentation technique ce qui rend le recours au Reverse Engineering incontournable

3.3.3 Préparation du la Raspberry :

Configuration de l'adresse IP statique :

Avant d'installer le broker MQTT Mosquitto nous avons fixé l'adresse IP à l'interface ethernet eth0 du raspberry dans notre réseau internet (celui du département).

La figure ci-dessous représente l'étape où nous avons fixé l'adresse IP.

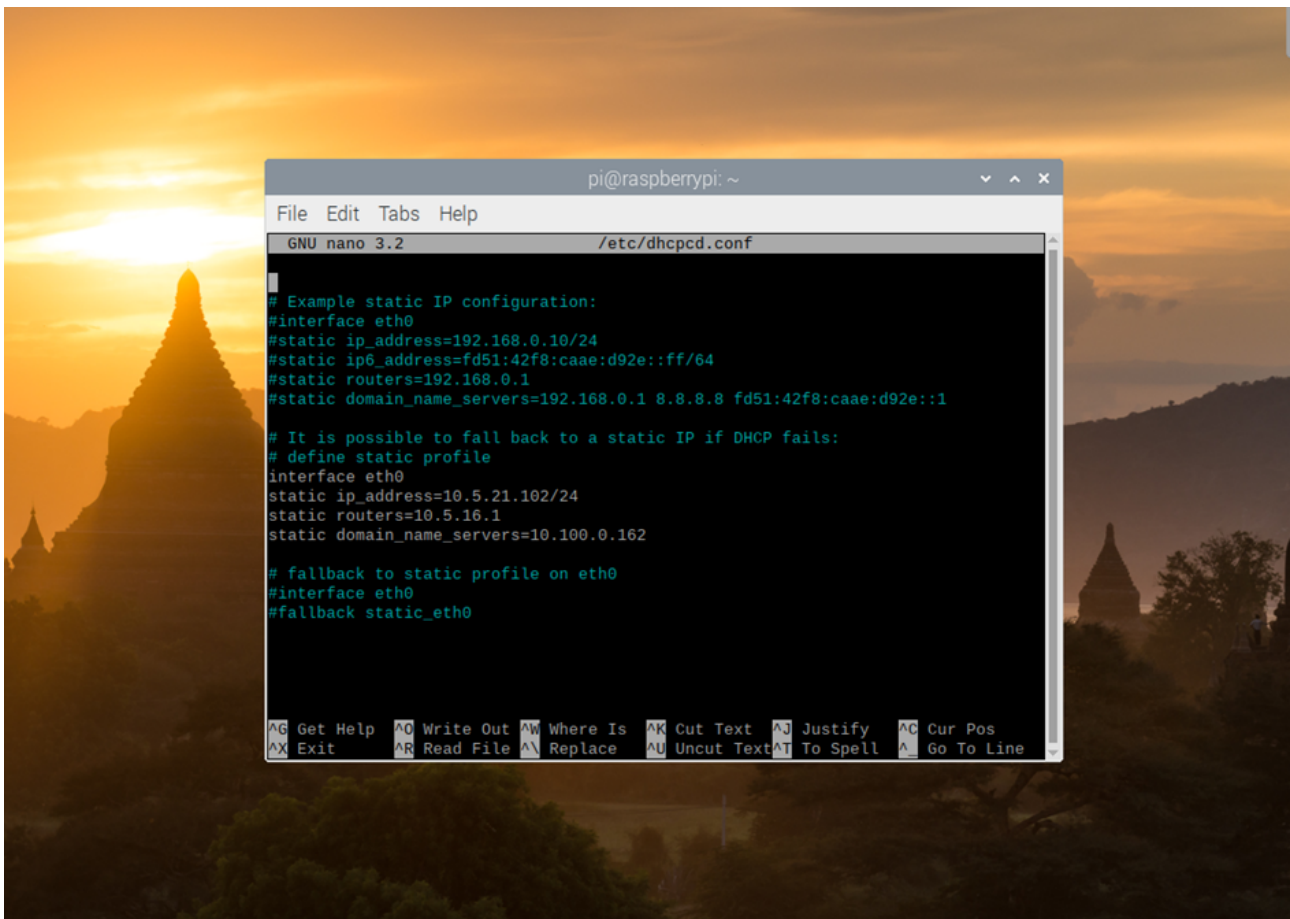


FIGURE 3.18 – La configuration d'adresse IP sur la raspberry

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

Instalation de Broker MQTT :

Par la suite nous avons installé le Broker MQTT mosquitto. sur la raspberry PI3 ,et la figure 3.19 représent les commandes utilisées pour l'installation de mosquitto et des éléments requis pour la communication mqtt .

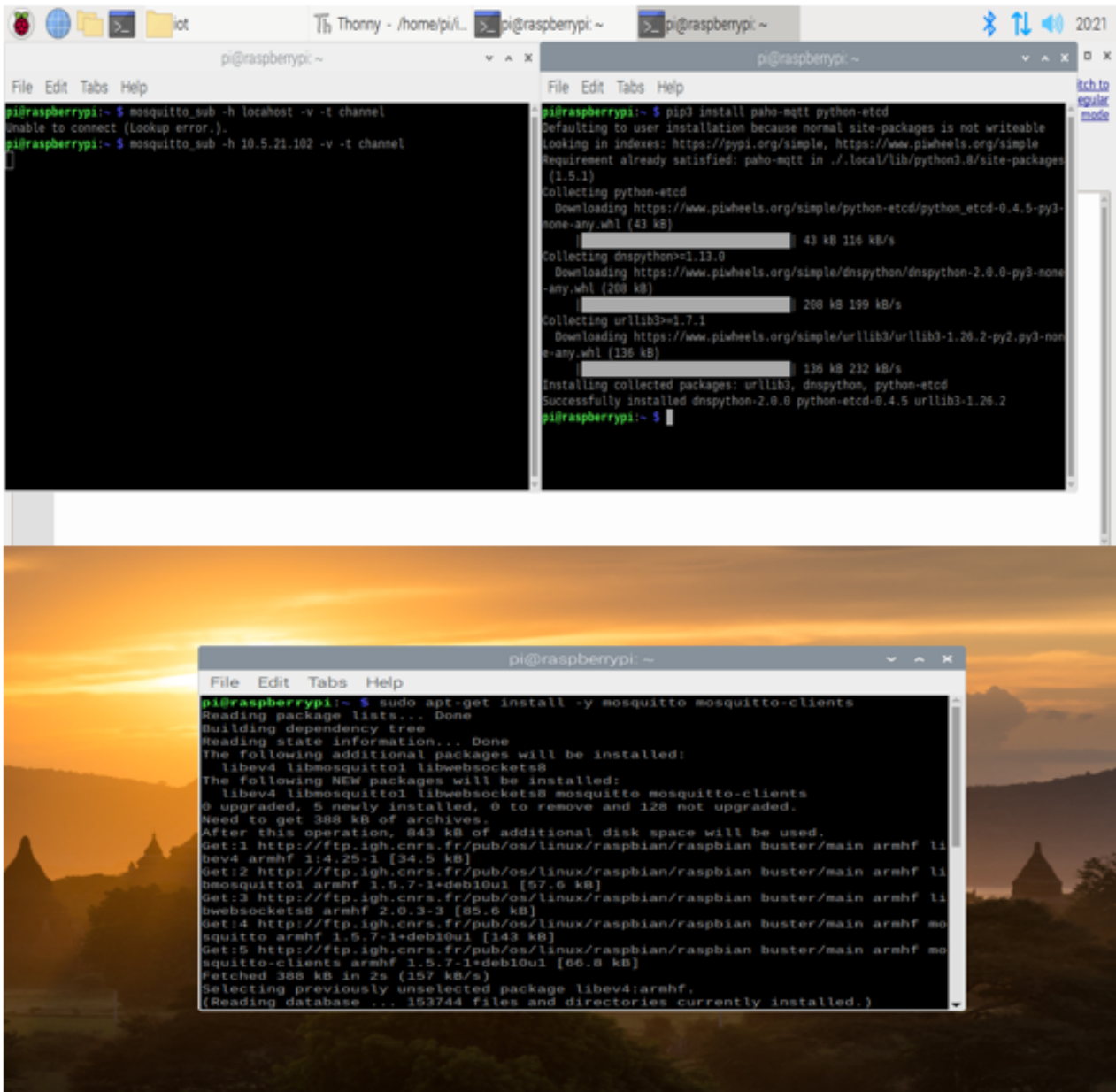
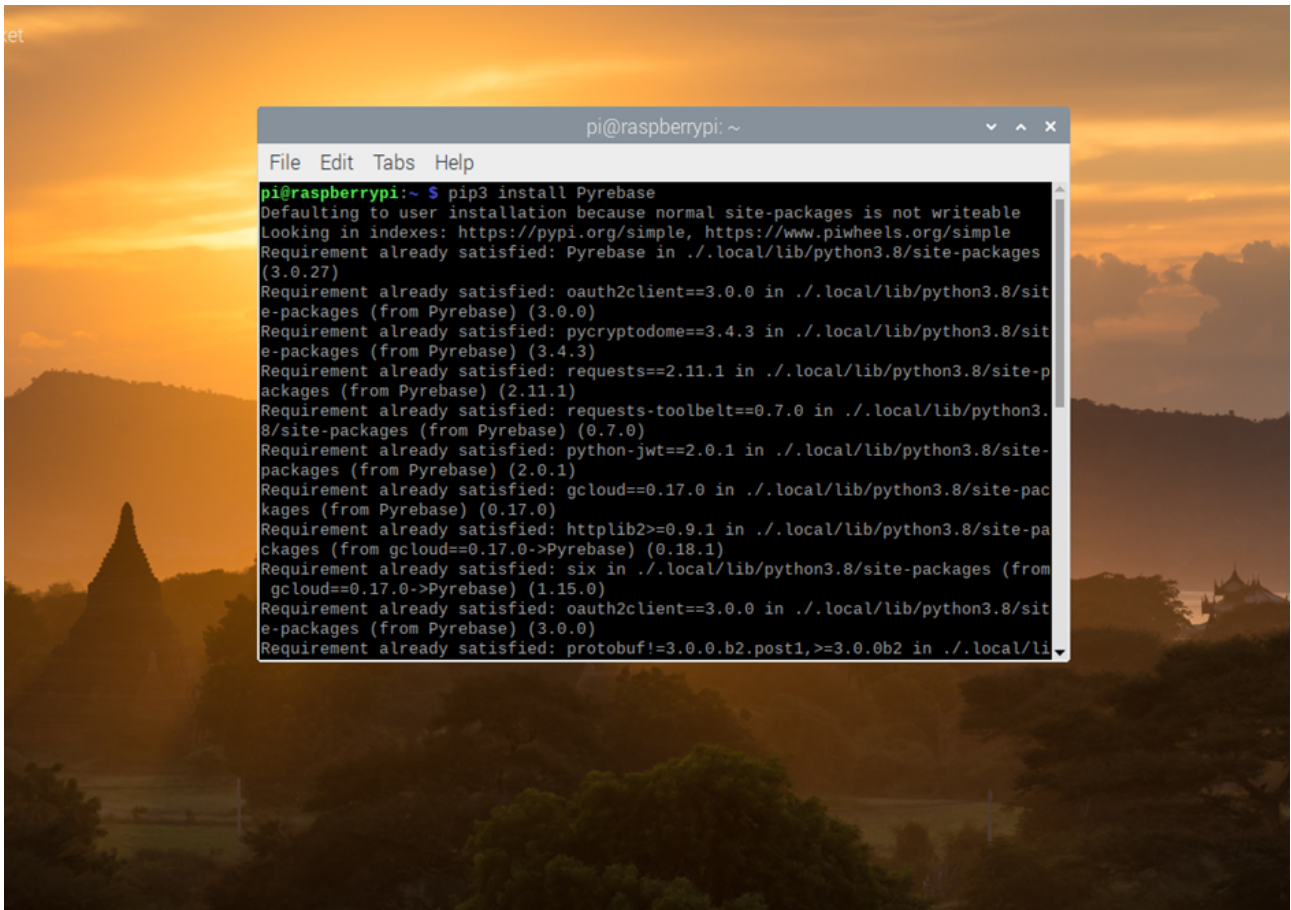


FIGURE 3.19 – illustration d'installation de mosquitto

Installation de PyreBase :

Par défaut, Firebase ne supporte pas le langage de programmation Python. Et comme nous programmons notre Raspberry Pi avec ce dernier, nous avons dû lui installer le module Python “Pyrebase” grâce auquel nous avons pu faire communiquer le script python contenu dans notre carte avec notre base de données temps réel.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ pip3 install Pyrebase  
Defaulting to user installation because normal site-packages is not writeable  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: Pyrebase in ./local/lib/python3.8/site-packages (3.0.27)  
Requirement already satisfied: oauth2client==3.0.0 in ./local/lib/python3.8/site-packages (from Pyrebase) (3.0.0)  
Requirement already satisfied: pycryptodome==3.4.3 in ./local/lib/python3.8/site-packages (from Pyrebase) (3.4.3)  
Requirement already satisfied: requests==2.11.1 in ./local/lib/python3.8/site-packages (from Pyrebase) (2.11.1)  
Requirement already satisfied: requests-toolbelt==0.7.0 in ./local/lib/python3.8/site-packages (from Pyrebase) (0.7.0)  
Requirement already satisfied: python-jwt==2.0.1 in ./local/lib/python3.8/site-packages (from Pyrebase) (2.0.1)  
Requirement already satisfied: gcloud==0.17.0 in ./local/lib/python3.8/site-packages (from Pyrebase) (0.17.0)  
Requirement already satisfied: httplib2>=0.9.1 in ./local/lib/python3.8/site-packages (from gcloud==0.17.0->Pyrebase) (0.18.1)  
Requirement already satisfied: six in ./local/lib/python3.8/site-packages (from gcloud==0.17.0->Pyrebase) (1.15.0)  
Requirement already satisfied: oauth2client==3.0.0 in ./local/lib/python3.8/site-packages (from Pyrebase) (3.0.0)  
Requirement already satisfied: protobuf!=3.0.0.b2.post1,>=3.0.0b2 in ./local/lib/python3.8/site-packages (from Pyrebase) (3.0.0)
```

FIGURE 3.20 – Illustration d’installation de PyreBase

Le scripte effectue une écriture dans le champ contenant l’état de l’objet à chaque fois qu’un mouvement est détecté.

Quant à la lecture, elle est faite en temps réel avec le streaming de données, c’est-à-dire que notre scripte python se met en écoute sur le champ contenant l’état de l’objet et à chaque changement de valeur dans ce dernier il en est averti et récupère l’information.

Nous donnons dans ce qui suit les schémas de réalisation de deux objets intégrés, à savoir un détecteur de mouvement et une lampe multicolore

3.3.4 Le détecteur de mouvement :

Les figures suivantes représentent le câblage entre la carte ESP32 et le détecteur de mouvement dans le cadre de notre travail réalisé.

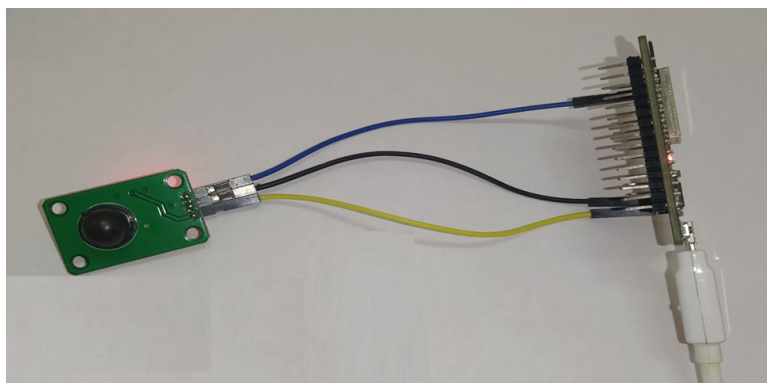


FIGURE 3.21 – le câblage entre la carte ESP32 et le détecteur de mouvement.

La figure suivante représente le schéma de câblage entre la carte ESP32 et le détecteur de mouvement sous fritzing.

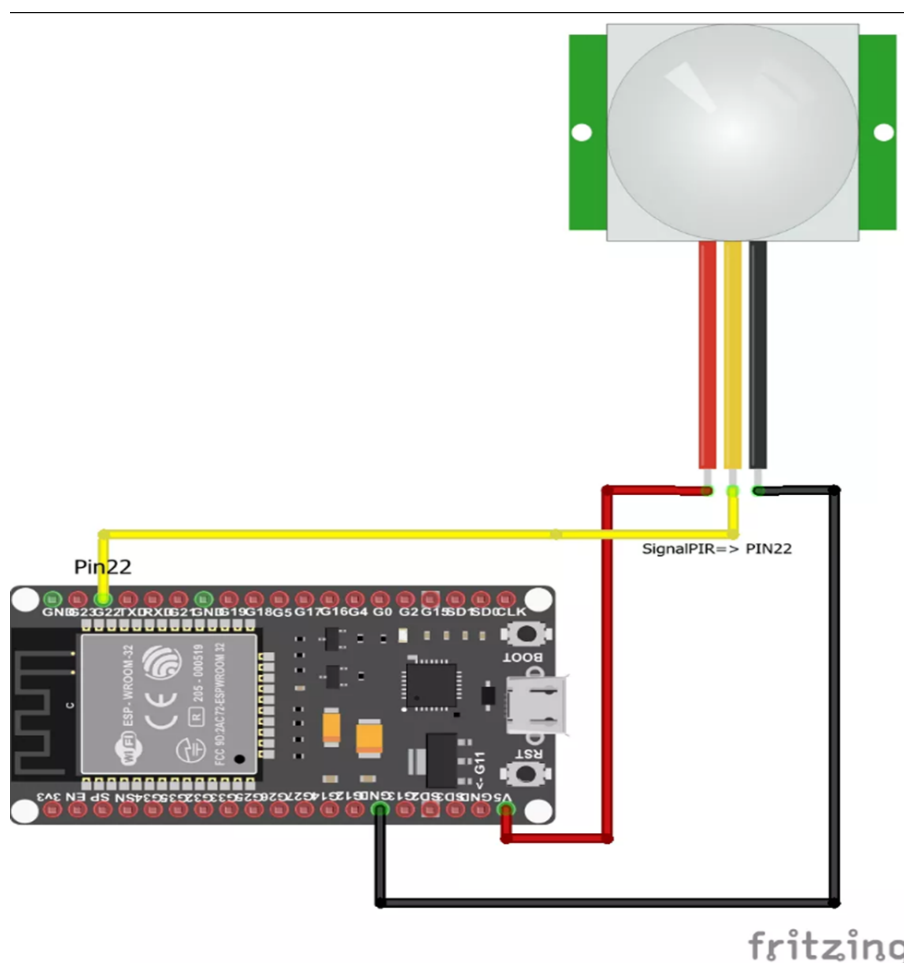


FIGURE 3.22 – le schéma de câblage entre la carte ESP32 et le détecteur de mouvement sous fritzing.

3.3.5 La lampe multicolor :

Les figures suivantes représentent la réalisation faite au niveau de la lampe. Ainsi que le câblage effectué entre l'émetteur infrarouge et l'ESP32.



FIGURE 3.23 – La lampe conectée

3.3.6 L'application Android :

Pour le moment notre application est composée de trois activités, par conséquent de trois interfaces.

La première est l'interface d'authentification.

Cette dernière est réalisée à l'aide Firebase UI Auth (User of Interface) qui est une bibliothèque construite sur la SDK de Firebase Authentication.

Elle permet de réaliser une connexion avec email/mot de passe, Google, Facebook, Twitter, GitHub ainsi que par téléphone et ce d'une manière simple.

Elle offre l'intégration automatique de Smart Lock for Passwords pour une connexion rapide et bien d'autres avantages. [7]

Dans notre cas, nous l'avons utilisé pour réaliser une authentification avec un compte Google.

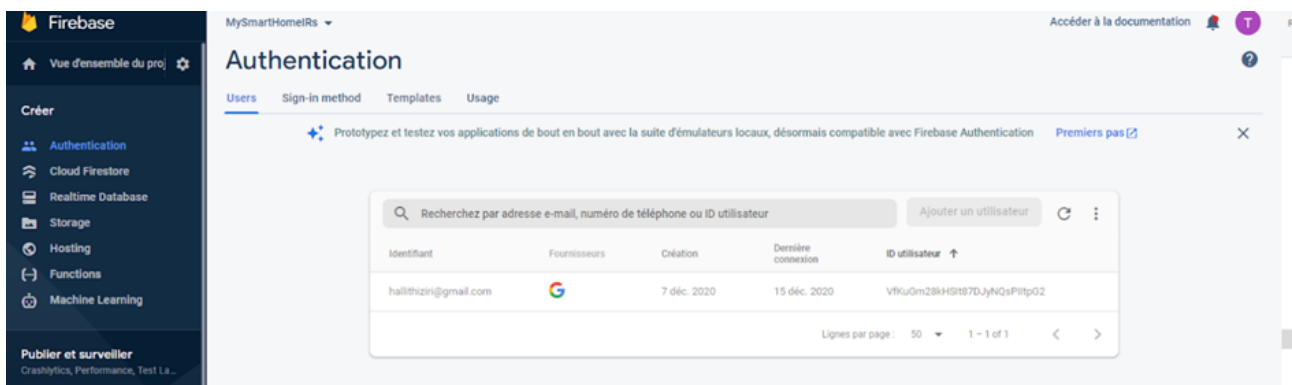


FIGURE 3.24 – L'authentification avec un compte google

Les interfaces de l'application android :

L'interface Authentification :

pour contrôler les équipements de la maison via l'application mobile nous devons d'abord s'authentifier via un compte mogole tels que décrit dans la figure 3.25.

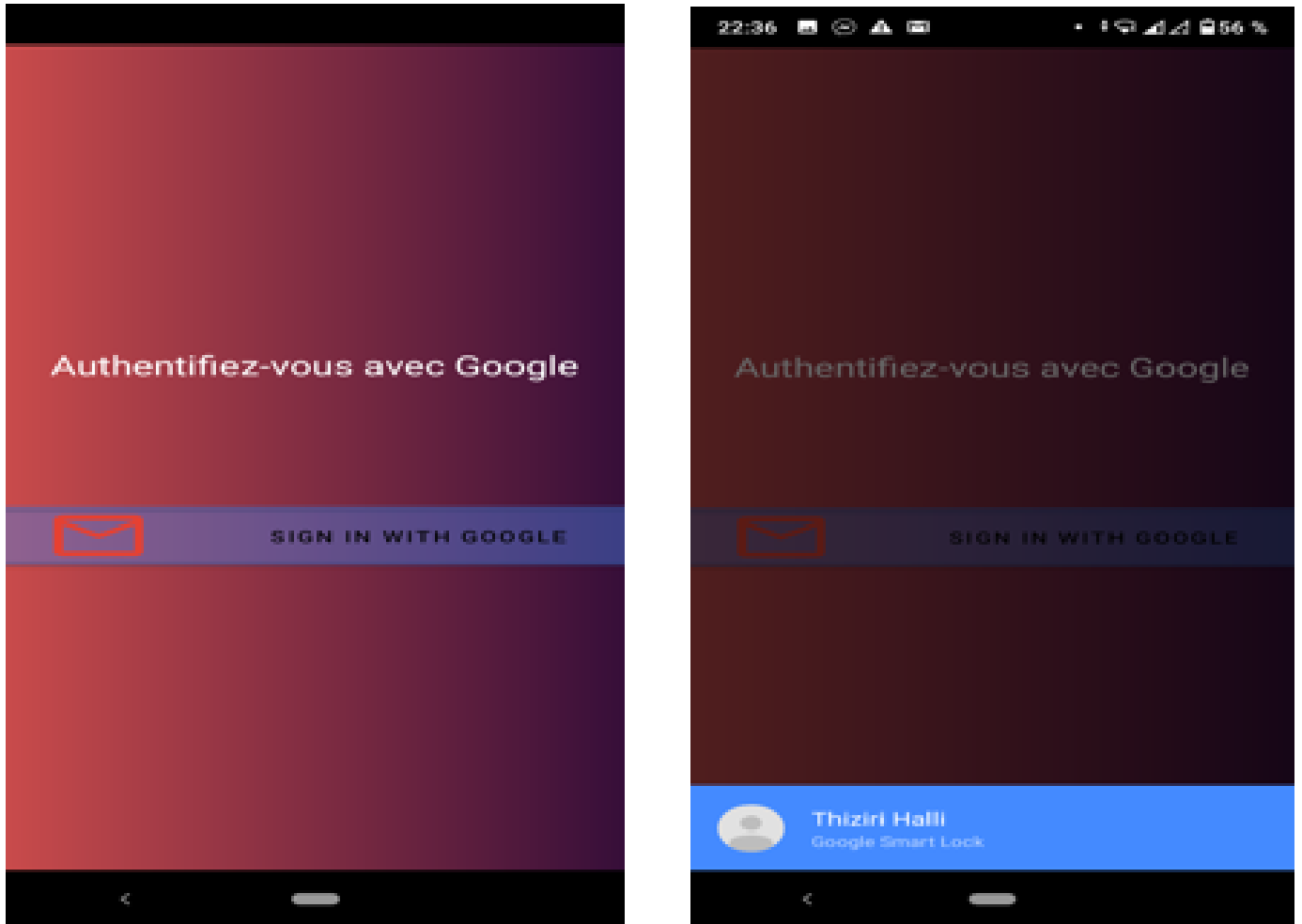


FIGURE 3.25 – L'interface d'authentification

Interface d'accueil :

Une fois l'authentification faite, l'utilisateur accède à l'interface d'accueil où il trouvera tous ses équipements triés par type. La figure suivante représente l'interface d'accueil.



FIGURE 3.26 – Interface d'accueil.

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

En cliquant sur les lampes ça nous dirige vers l'interface dédiée à la lampe avec laquelle le test est réalisé. Cette interface simule la télécommande de la lampe tout en donnant l'état de l'objet en temps réel. Au moment où l'utilisateur sélectionne un état voulu pour la lampe une modification est faite dans la base de données suite à cela une lecture de cette modification est faite et est affiché sur l'interface à l'utilisateur.

L'interface dédiée à la Lampe :

Dans cette interface nous pouvons changer l'état et contrôler la luminosité de notre lampe tel que expliqué auparavant , la figure suivante représente cette interface .

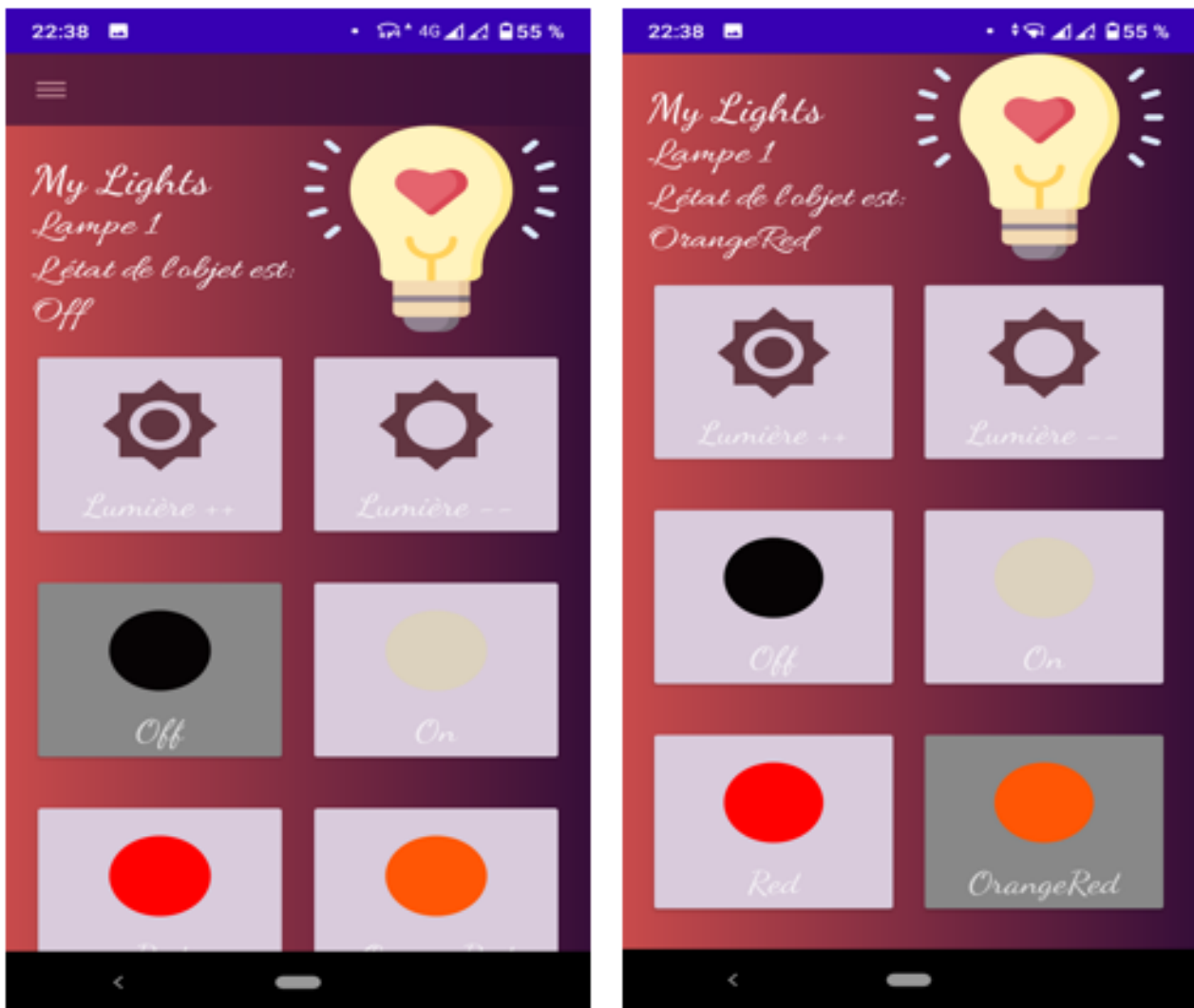


FIGURE 3.27 – L'interface dédiée à la Lampe.

3.3. PRÉSENTATION DES ÉTAPES DE RÉALISATION :

L'acheminement des interfaces de l'application android : Dans La figure ci-dessous nous avons illustré la manière dont les interfaces de l'application mobile sont enchainée

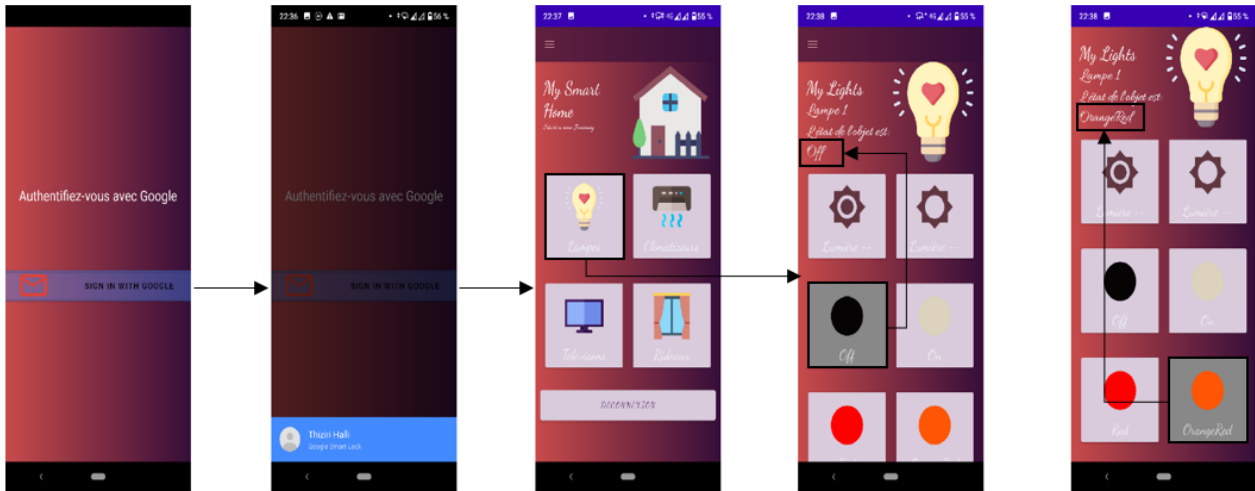


FIGURE 3.28 – L'enchainement des interfaces de l'application android.

3.3.7 Création de la base de données Firebase :

Nous avons créé une base de données dans l'onglet Realtime Database dans la console de Firebase.

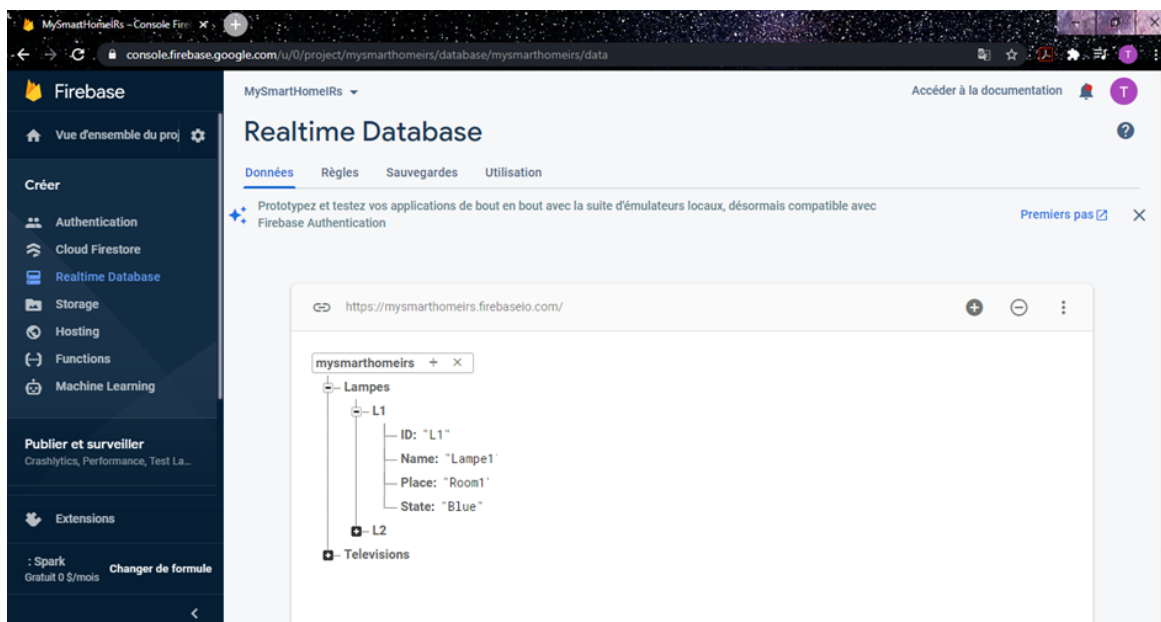


FIGURE 3.29 – la base de données Firebase

3.4 Conclusion

Dans ce chapitre nous avons commencé par présenter les outils et l'environnement de travail que nous avons utilisés. Ensuite nous avons présenté les différents schémas de câblage entre les différents capteurs utilisés pour la réalisation de notre système iot. Nous avons aussi présenté quelques captures d'écran illustrant notre application Android et les étapes de configurations utilisées sur notre centrale domotique et notre base de données Firebase.

Conclusion générale :

Le domaine des objets connectés prend de plus en plus d'ampleur et s'infiltrer pratiquement dans tous les domaines de la vie quotidienne, notamment dans les foyers. Plein de projets se sont penchés sur la réalisation de maisons intelligentes/connectées. Les maisons automatisées offrent plusieurs avantages, cependant elles ne sont pas très répandues dans le monde, ceci est dû en partie au prix exorbitant que l'automatisation d'un foyer coûte.

A travers notre travail nous nous sommes fixé comme objectif d'apporter une solution qui permet à l'utilisateur d'automatiser sa maison en gardant ses anciens équipements classiques et ce à un prix assez abordable.

Au début, nous avons présenté le concept de l'IoT, ses avantages et inconvénients ainsi que ses domaines d'application. Par la suite, nous nous sommes penchés sur les maisons automatisées tout en précisant la différence entre les termes "domotique" "maison intelligente" et "maison connectée". Après cela, nous avons présenté la conception de notre système en nous appuyant sur les schémas de fonctionnement internes. Enfin, nous avons présenté la réalisation de notre système et les différents tests appliqués sur ce dernier.

Avec ce travail, nous sommes arrivés à atteindre notre principal objectif qui est de concevoir et réaliser un système qui permet de mettre à niveau des équipements classiques télécommandés en leur introduisant des technologies qui les font entrer dans le monde de l'IoT. A savoir que notre solution est personnalisable, elle va différer d'un foyer à un autre selon le type et le nombre d'équipements contenus dans ce dernier.

Notre projet reste imparfait et peut être amélioré. pour cela nous proposons d'examiner et développer les points suivants :

- Ajouter plus de capteurs tels que des capteurs de température, de lumière et pleins d'autres de sorte à ce que notre système ne réagisse pas uniquement au mouvement mais à toutes les données de captures en parallèle.
- Faire bénéficier l'utilisateur d'une intelligence artificielle plus avancée et développée lui permettant un maximum de confort et plus d'autonomie pour la maison. De sorte à ce qu'elle puisse apprendre les habitudes de ses occupants et ce au cours de la journée, par exemple leurs habitudes au réveil, pendant la journée, au moment de dormir, etc.

Si les occupants de la maison sont des personnes âgées ou à besoins spécifiques la maison s'adaptera à leurs besoins en leur fournissant non seulement le confort mais aussi la sécurité

et surveillera leur état de santé, si la personne fait un malaise la maison contactera les pompiers.

Nous souhaitons vivement que ce projet ouvre un champ de possibilité d'extension et qu'il servira comme support de base pour d'autres travaux d'études plus approfondies.

Bibliographie

- [Alc19] Etienne Alcouffe. *A quoi sert Firebase, la plateforme mobile de Google ?* junto, Avr 25, 2019.
- [CHE18] MAXIME CHENOT. *introduction à firebase*. lesveilleursdenuit, 26 AVRIL 2018.
- [Eva11] Dave Evans. « *L'Internet des objets Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?* Cisco Internet Business Solutions Group (IBSG)., Avril 2011.
- [Fra14] IoT France. *ESP32 Tout sur le nouveau produit d'expressif qui va remplacer le fameux esp8266*. overblog, 8 novembre 2014.
- [Gui14] Jean-Luc Guillaume. *Qu'est ce qu'une carte Arduino ?* LOCODUINO, 8 novembre 2014.
- [Keu16] Dorian Keuller. « *Le secteur de la santé face à l'émergence de l'Internet des Objets : développement d'un outil d'aide à la décision*. UNIVERSITE CATHOLIQUE DE LOUVAIN, 2016.
- [Nic11] Nicolas. « *qu'est ce que l'IOT et pourquoi mener une strategie ?* digora, Avril 2011.
- [Sen10] Yassir Senoun. *Internet des Objets : Quels protocoles applicatifs utiliser ?* engeneering at PUBLICIS SAPIENT, 2010.
- [Som16] SAHRAOUI Somia. « *Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things)*. Université de Batna (thèse Doctorat), 2016.
- [Y.S17] Y.Senoun. *Comparative analysis and practical implementation of the ESP32 micro-controller module for the internet of things*. IEEE, 12-15 Sept 2017.

webographie

- [1] : <https://www.connectwave.fr/techno-appli-iot/iot/reseaux-et-infrastructures-iot/> (consulté le 15 juillet septembre 2020)
- [2] : https://igm.univ-mlv.fr/~dr/XPOSE2007/aessaidi-ndiop_LA-DOMOTIQUE/intro.htm?fbclid=IwAR0tqrldxKh1yciDou9fPixaTOeXfAkQIFoGv4C_YDKmNohj4DCvPh51puM (consulté le 15 juillet septembre 2020)
- [3] : <https://www.imagescreations.fr/qu-est-ce-que-assistant-vocal-google-home/> (consulté le 18 juillet septembre 2020)
- [4] : <https://www.ldlc.com/fiche/PB00205573.html#> (consulté le 20 juillet septembre 2020)
- [5] : <https://touch-ir.com/comment-marche-un-emetteur-recepteur-infrarouge/> (consulté le 24 juillet septembre 2020)
- [6] : <https://projetsdiy.fr/mosquitto-broker-mqtt-raspberry-pi/#> (consulté le 28 juillet septembre 2020)
- [7] : <https://firebase.google.com/docs/auth/android/firebaseui> (consulté le 30 juillet septembre 2020)
- [8] : <https://www.maison-travaux.fr/maison-travaux/domotique/domotique-maison-connectee-maison-intelligente-comparatif-fp-> (consulté le 2 septembre septembre 2020)
- [9] : <https://www.youtube.com/watch?v=-2NhjhzFQsM> (consulté le 3 septembre septembre 2020)
- [10] : <https://developer.android.com/studio> (consulté le 3 septembre septembre 2020)
- [11] : [BOOK_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf](#) (consulté le 5 septembre septembre 2020)
- [12] : <https://www.raspbian.org/> (consulté le 7 septembre septembre 2020)