

Rapport de Projet de Fin d'études

Etude et réalisation d'un système IoT



Réalisé par :
Salma Nakkache
Kaluba Mutale
Karima Messaoudi

Encadré par :
Mme Nadia Afifi
Mr Hicham Belhadaoui

Table des matières

Remerciement.....	4
Introduction générale	5
Chapitre 1 : Etude théorique de l'IOT.....	6
I. Introduction	6
II. Définition de l'IOT	6
III. Notion de l'objet connecté	8
IV. Domaine application de l'IOT	9
V. Architecture de l'IOT.....	12
VI. Technologies utilisées dans l'IOT	13
VII. Les protocoles de communication IoT	19
VIII. Conclusion	22
Chapitre 2 : Spécification des besoins	23
I. Introduction.....	23
II. Objectif du projet	23
III. Cahier de charge	24
IV. Les étapes de réalisation.....	25
V. Conclusion	25
Chapitre 3 : Analyse et conception	26
I. Introduction.....	26
II. Présentation d'UML	26
III. Modèle des cas d'utilisations	27
1. Diagramme de cas d'utilisation	27
2. Diagramme séquence.....	28
IV. Diagramme de structure	32
Diagramme de déploiement	33
V. Conclusion	33
Chapitre 4 : Matériels et logiciels	34
I. Introduction.....	34
II. Partie matériel	34
a. Les cartes électronique	34
b. Les capteurs	38
c. Les actionneurs	40
d. Autre composants	45
III. Partie logiciel	49
IV. Conclusion	35

Chapitre 5: Réalisation et mise au point	54
I. Introduction	54
II. Préparation de la maquette	54
III. Schéma globale du prototype	55
IV. Installation des composants	55
V. Préparation de la carte Raspberry	56
VI. Configuration du Camera et Raspberry.....	58
VII. Développement d'application	59
VIII. Conclusion	64
Conclusion et perspective	65
Reference.....	66

Remerciement

Nous dédions nos plus vifs remerciements à Mme Nadia Afifi et Monsieur Hicham Belhadaoui pour leur soutien, leur aide, leur disponibilité, leurs remarques pertinentes et leur précieux conseils qu'ils nous ont prodigués, tout au long de la période de notre PFE et de nous avoir donné tout le matériel dont nous avons besoin pour réaliser ce travail.

Nous tenons également à exprimer nos profondes reconnaissances à l'égard de, notre encadrant, Monsieur Hicham Belhadaoui, qui nous a offert la salle de recherche pour le bon déroulement de notre stage.

Un grand merci à nos familles, pour leurs conseils ainsi que leur soutien inconditionnel.

Enfin, que toute personne ayant contribué de près ou de loin à la préparation de ce travail, trouve ici l'expression de notre plus profonde reconnaissance.

Introduction

L'évolution de la technologie et du mode de vie nous permet aujourd'hui de prévoir des logements mieux adaptés. Il semble très important d'améliorer le confort et la sécurité de la maison. L'informatique a été appliquée pour créer des maisons intelligentes afin d'améliorer les conditions de vie des personnes à la maison et de leur permettre la piloter par un smartphone, ce qui permet de contrôler l'éclairage, la température ou divers appareils électroniques, améliorant ainsi le confort des occupants. En outre, un autre objectif fondamental est de protéger les individus grâce à un système capable de prévoir des situations potentiellement dangereuses ou de réagir à des événements mettant en danger l'habitant.

Dans ce contexte et dans le cadre de notre projet de fin d'étude nous proposons de concevoir un modèle réduit d'une « SMART HOME ». Ce prototype permettrait d'implémenter des fonctions de domotique à savoir : La gestion d'éclairage, l'acquisition de la température à l'intérieur de la maison, la détection de fuite de gaz et de la fumée. Ces scénarios seront automatisés via des cartes électroniques exécutant des programmes informatiques et aussi via une application mobile.

Tout au long de ce rapport, nous allons exposer les différentes étapes de réalisation de notre projet qui sera divisé en cinq chapitres :

Le premier chapitre « **Etude théorique de l'IOT** » se focalise à définir, et détailler les concepts théoriques des objets connectés.

Le deuxième chapitre « **Spécification des besoins** » est dédié à la description du projet ; la présentation du cahier des charges et les enjeux envisagés pour notre système smart-home.

Le troisième chapitre « **Analyse et conception** » sera consacré pour la modélisation du projet en utilisant le Langage de Modélisation Unifiée (UML).

Le quatrième chapitre « **Matériels et logiciels** » est consacré à la description de la partie matérielle et logicielle du projet ; les composantes de notre solution vont être détaillées.

Finalement, le cinquième chapitre « **Réalisation et mise au point** » est consacré à la conception et la réalisation de notre projet. Nous détaillerons les phases de la mise en place de notre système smart home.

Chapitre 1 : Etude théorique de l'IOT

I. Introduction :

L'internet des objets **IOT** est l'interconnexion avec l'internet de toutes les choses qui nous entourent, et comprend les sciences de l'électronique et les sciences des télécommunications, et avec toutes les innovations possibles dans ce domaine, l'IoT devient encore plus puissant et sa définition devient aussi plus souple pour suivre le développement technologique. Tout au long du présent chapitre, nous présenterons en détail les concepts théoriques de l'Internet des objets.

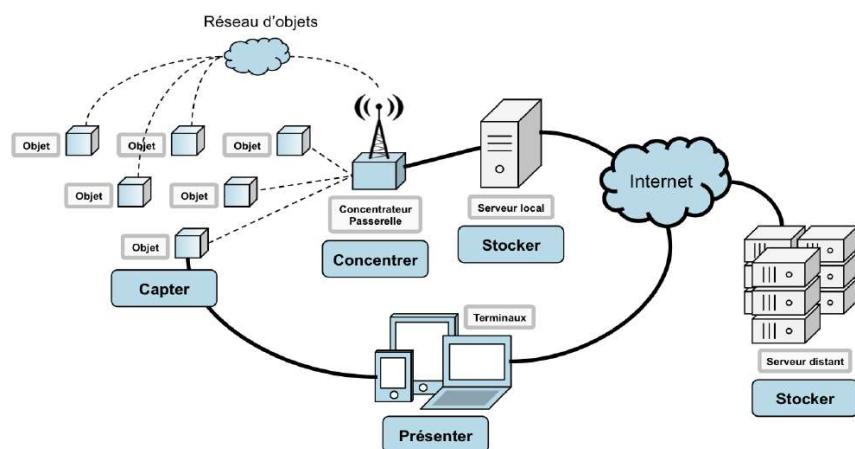
II. Définition de l'IOT :

L'Internet des objets, tel qu'il est donné syntaxiquement par son nom, est composé de deux termes : "Internet" et "Objets".

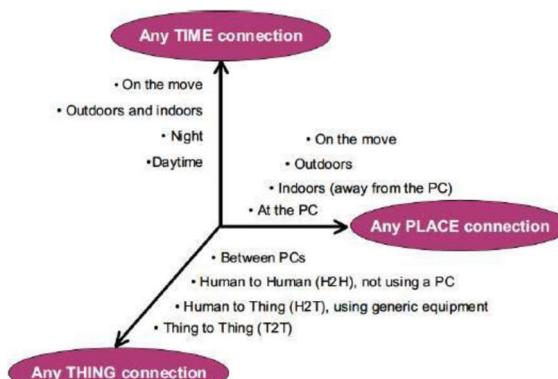


L'internet des objets se situe à l'intersection des domaines de l'informatique et des communications électroniques, où tout objet communique, peut être interrogé, envoie de l'information et interagit avec son environnement.

L'IoT peut être défini comme étant un réseau qui permet, via des systèmes d'identification électronique normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant.



Cette vision de l'Internet des objets introduira une nouvelle dimension aux technologies de l'information et de la communication : en plus des deux dimensions temporelles et spatiales qui permettent aux personnes de se connecter de n'importe où à n'importe quel moment, une nouvelle dimension donc est ajoutée « *objet* », qui leur permettra de se connecter à n'importe quel objet.



III. Notion de l'objet connecté :

L'IoT repose avant tout sur les objets connectés. On peut dire qu'un objet connecté a la capacité de capter une donnée et de l'envoyer, via le réseau Internet ou d'autres technologies, pour que celle-ci soit analysée et visualisée sur des tableaux de bord dédiés. Les objets connectés interagissent avec leur environnement par le biais de capteurs (température, vitesse, humidité, vibration...).

Généralement, un objet connecté est caractérisé par :

- **Identité** : pour que les objets soient gérables, il est essentiel que chaque objet connecté possède une identité unique qui lui propre et qui le distingue des autres objets du système.
- **Interactivité** : les progrès technologiques ont permis de connecter une grande variété d'objets et de dispositifs. Un objet n'a pas besoin d'être connecté à un réseau à tout moment. Pour des objets dits passifs tels que des livres ou des DVD, des étiquettes RFID doivent seulement être en mesure de signaler leur présence, de temps en temps, comme le moment de quitter le magasin.
- **Programmable** : l'objet connecté doit être programmé et piloté à distance via un ordinateur.
- **Sensibilité** : un objet a la capacité de percevoir son environnement et peut collecter ou transmettre des informations à celui-ci.
- **Autonomie** : cette caractéristique est, peut-être, la caractéristique la plus importante pour l'objet connecté. On désigne par cette caractéristique la capacité de l'objet d'agir sans l'intervention d'un tiers.



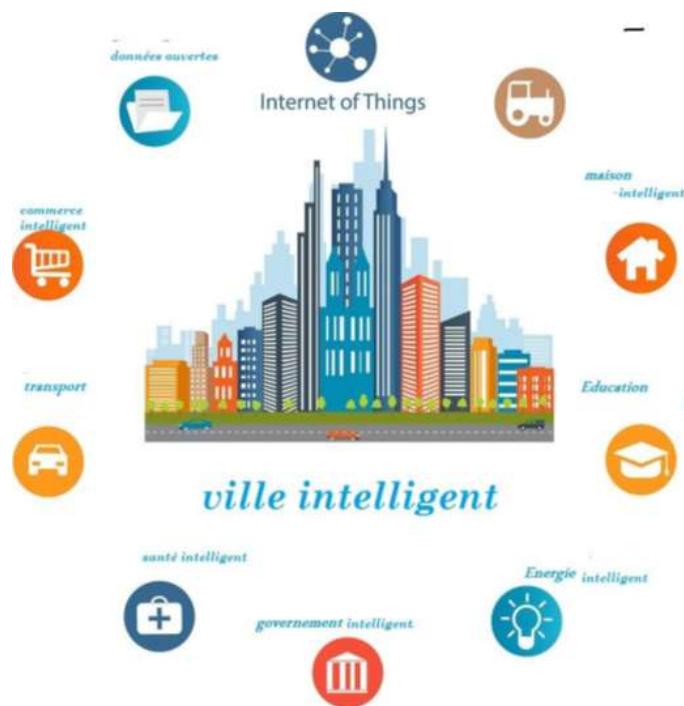
IV. Les domaines d'application de l'IOT :

Plusieurs domaines d'application sont touchés par l'IoT, Parmi ces principaux domaines nous citons: le domaine de la sécurité, le domaine du transport, l'environnement et l'infrastructure et les services publics...etc.

- **Villes Intelligentes :**

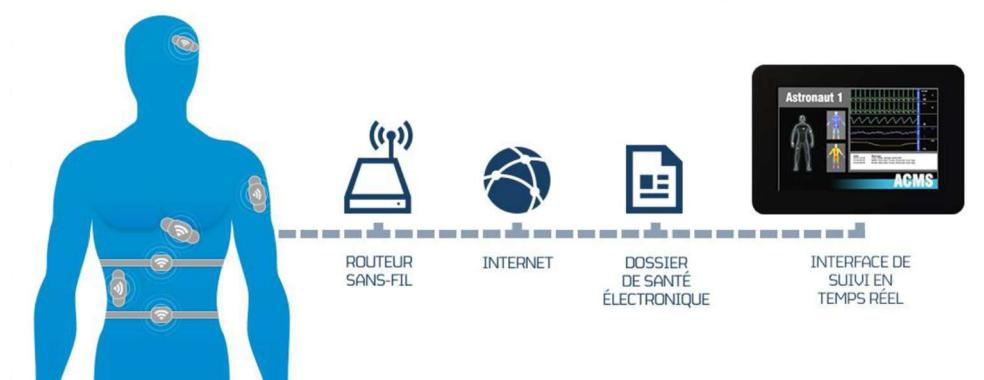
Efficace, innovant et participatif, le concept smart cities offre à ses habitants une qualité de vie maximale avec une consommation de ressources minimale grâce à une combinaison intelligente des infrastructures aux différents niveaux hiérarchiques.

Des capteurs peuvent être utilisés pour l'économie de l'eau et pour améliorer la gestion des parkings et du trafic urbain et diminuer les embouteillages et les émissions en CO₂. Ils peuvent être ainsi réutilisées par la population pour développer de nouveaux services et participer à l'amélioration de la vie urbaine.



- **Le Système de la santé électronique :**

L'internet des objets offre de nombreux avantages au secteur de la santé. Ce concept a rapidement transformé la prestation de soins en permettant le déploiement de réseaux personnels pour le contrôle et le suivi des signes cliniques, notamment pour des personnes âgées, les objets connectés permettant de suivre la tension, le rythme cardiaque, la qualité de respiration ou encore la masse graisseuse. Ceci permettra ainsi de faciliter la télésurveillance des patients à domicile, et apporter des solutions pour l'autonomie des personnes à mobilité réduite.



- **Agriculture intelligente :**

Dans le monde agricole, les solutions IoT prennent la forme de capteurs reliés à Internet pour collecter des mesures environnementales et mécaniques. Leur déploiement permet aux agriculteurs de prendre des décisions éclairées et améliore presque tous les aspects de leur travail, de l'élevage à l'agriculture.

Grâce aux nouveaux outils IoT, les agriculteurs peuvent surveiller les conditions de leur champ depuis n'importe où. En utilisant divers outils agricoles intelligents, les agriculteurs ont ainsi acquis un meilleur contrôle sur le processus d'élevage et de culture, ce qui améliore son efficacité.

Les applications IoT les plus populaires dans l'agriculture sont certainement les stations météorologiques intelligentes, les systèmes d'automatisation des serres, les drones agricoles qui collectent les données partout et en tout temps facilitant la surveillance à distance.



- **Alerte éruption volcanique :**

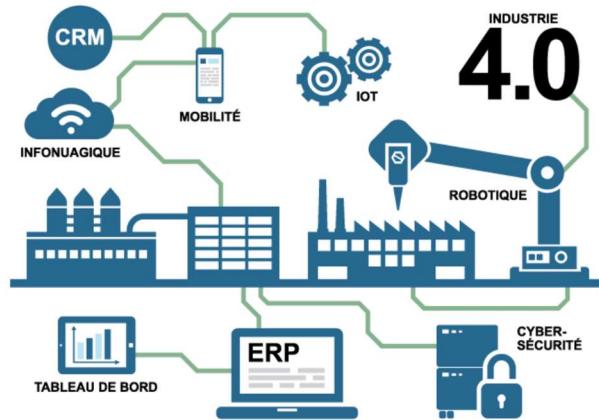
L'IOT peut jouer un rôle efficace dans la prévention des catastrophes naturelles. Plusieurs pays ont déjà mené des actions intéressantes sur ce sujet. Un exemple intéressant est celui de *Popocatepetl*, le volcan le plus actif d'Amérique du Nord. Situé à moins de 40 kilomètres de Mexico City, une éruption massive représenterait une catastrophe naturelle inimaginable. Le gouvernement avec des chercheurs a donc mis en place tout un réseau de capteurs permettant de prévenir les habitants en cas de séisme, un signe qui peut annoncer une éruption.



- **Industrie:**

La technologie IoT permettra un suivi total des produits, de la chaîne de production, jusqu'à la chaîne logistique et de distribution en supervisant les conditions d'approvisionnement. Cette traçabilité de bout en bout permet aux usines d'améliorer l'efficacité de ces opérations, d'optimiser la production et d'améliorer la sécurité des employés.

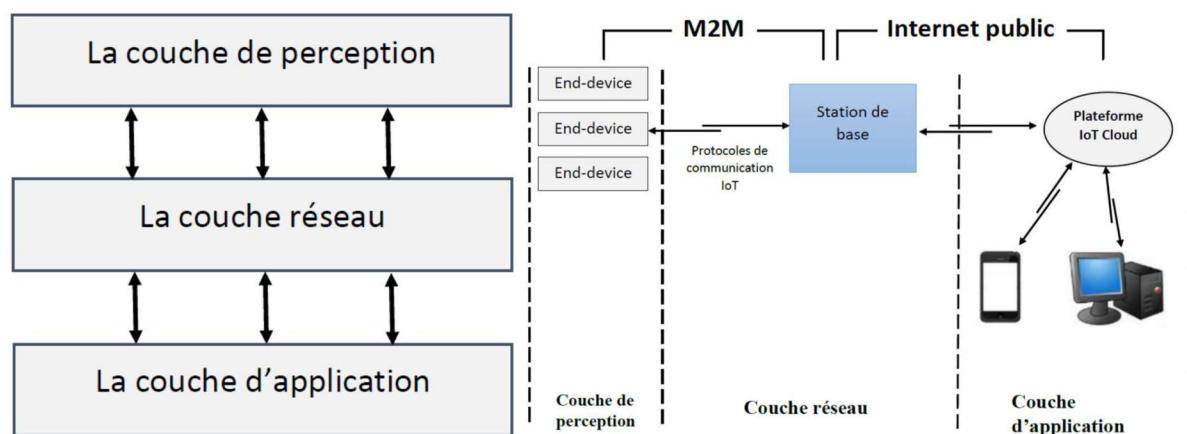
L'IoT est appelé quatrième vague de la révolution industrielle, ou Industrie 4.0. Elle apporte de nombreuses solutions et promesses en termes de productivité, de qualité et de sécurité.



V. Architecture de l'IOT :

L' architecture d'une solution IoT varie d'un système à l'autre en se basant sur le type de la solution à mettre en place.

Plusieurs normes architecturales IoT ont été proposées par plusieurs chercheurs et organismes de recherche. L'architecture la plus élémentaire est une architecture à trois couches:



- **La couche de perception :**

La tâche principale de la couche de perception est de reconnaître les propriétés physiques telles que la température, l'humidité, le niveau de la lumière, la vitesse, etc., par divers dispositifs de détection, et de convertir ces informations en signaux numériques. Les objets de cette couche peuvent avoir des capacités de détection et/ou des capacités d'actionnement.

- **La couche réseau :**

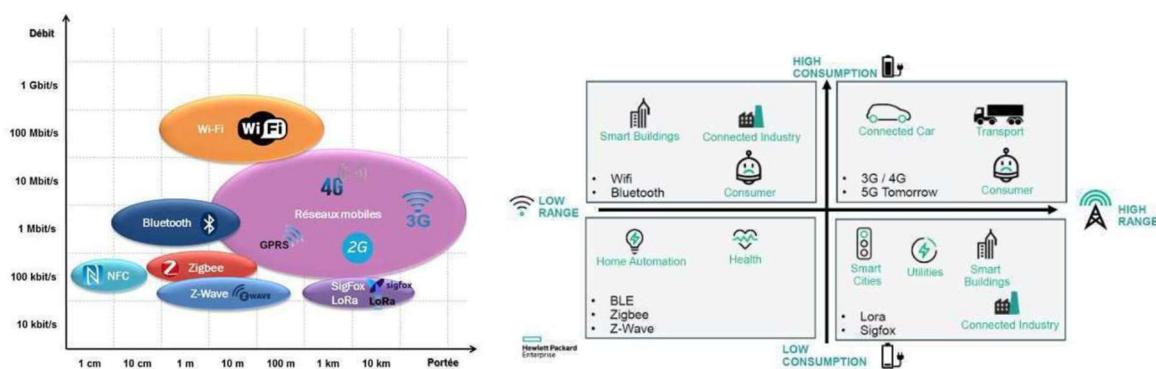
La couche réseau est la couche responsable de la transmission des données reçues de la couche de perception à une base de données, serveur, ou d'un centre de traitement.

Les principales technologies utilisées pour réaliser cette couche sont : les technologies cellulaires 2G / 3G / LTE, Wi-Fi, Bluetooth, Zigbee ou Ethernet....

- **La couche d'application :**

La couche application analyse les informations reçues de la couche réseau. Cette couche est chargée de fournir à l'utilisateur des services spécifiques et applications intelligentes.

VI. Les Technologies utilisées dans l'IOT :



1. RFID :

C'est une technologie utilisée pour l'identification des objets. Elle aide à étiqueter explicitement les objets pour faciliter leur identité à l'aide des appareils informatiques.

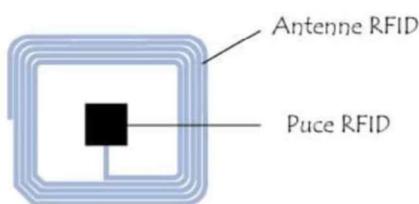
Un **système RFID** standard se compose de trois parties :

- Une étiquette RFID transpondeur qui transporte les données.
- Un lecteur/interrogateur qui envoie un signal radio à une étiquette RFID et lit la réponse de l'étiquette.
- Un logiciel middleware qui reçoit et traite les informations envoyées par le lecteur RFID.

Les données portées **sur une étiquette RFID** :

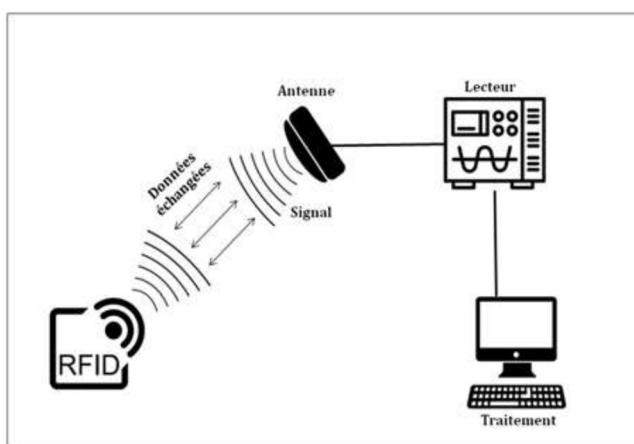
- Un identification unique (ID) telle que l'EPC (Electronic Product Code).
- Certaines informations liées au produit.

Les **étiquettes RFID** sont composées de deux composants de base :



- ✓ Un circuit intégré (puce RFID) pour stocker et traiter les informations, moduler et démoduler un signal radiofréquence (RF), et éventuellement pour récolter de l'énergie à partir du signal RF d'un lecteur
- ✓ Une antenne pour recevoir et transmettre des signaux RF.

Le principe de fonctionnement d'un système RFID :



Le lecteur RFID initie les transferts de données en diffusant un signal de requête. Plus d'une étiquette RFID répondra au lecteur en envoyant ses données stockées (par exemple, le code EPC).

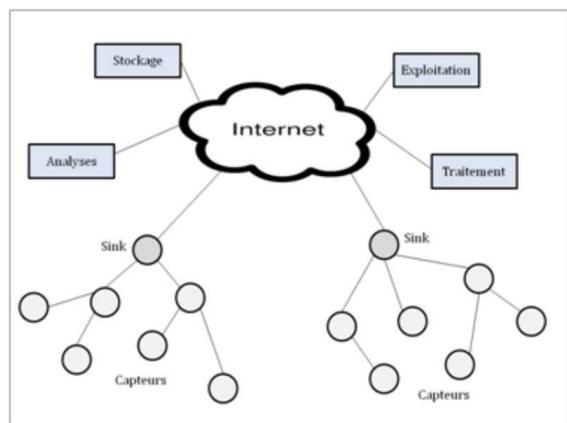
2. NFC :



Le NFC est un standard de communication RF (radio fréquence) sans-contact à courte distance basé sur la technologie RFID, permettant une communication simple, rapide, intuitive et facilement sécurisée entre deux dispositifs électroniques. Les deux premières lettres NF (Near Field ou champ proche) du sigle NFC correspondent à des notions physiques de propagation des champs électromagnétiques.

Le NFC est appliqué dans diverses applications : le paiement mobile, la billetterie, les carte de visite électroniques, le contrôle d'accès, les échanges peer-to-peer...

3. Réseaux de capteurs sans fil (WSN) :



Un WSN se compose généralement d'un certain nombre de nœuds de capteurs de petite taille, qui peuvent détecter et réagir à l'environnement, traiter les informations et communiquer entre eux sans fil; et un ou plusieurs récepteurs d'informations qui reçoivent et traitent les informations rapportées par les nœuds de capteurs, ainsi, ils contrôlent le fonctionnement du WSN.

Un WSN peut être déployé dans l'environnement pour interagir avec le monde physique, c'est-à-dire pour nous fournir les états réels du monde physique en temps réel via de riches informations sensorielles ainsi que la capacité de contrôler le monde physique via des actionneurs. Les réseaux de capteurs jouent un rôle crucial dans l'IoT.

4. Bluetooth :



La technologie Bluetooth est un acteur incontournable pour les réseaux de courte portée (WPAN) basée sur la norme IEEE 802.15.1. Elle se caractérise par une communication à courte portée, qui nécessite une faible puissance et un faible coût. La portée de transmission peut aller jusqu'à 10 mètres (par exemple, les Smartphones) avec une consommation électrique inférieure à 30 mW et une fréquence de 2,4 GHz (ISM).

5. Bluetooth Low Energy (BLE) :



C'est une technologie basée sur la norme IEEE 802.15.1 sous classe de la famille Bluetooth 4.0, commercialisée sous le nom Smart Bluetooth. Cette norme incompatible avec Bluetooth offre une consommation réduite d'énergie (Tx 2.9mW, Rx 2.3mW). Elle prend en charge d'iOS, Android, Windows et GNU / Linux. Elle est utilisée dans les smartphones, tablettes, montres intelligentes, appareils de surveillance de la santé et de la condition physique. Elle se caractérise par une portée de 50-150m avec des temps de latence 15 fois plus courts que Bluetooth, une vitesse de transmission de 1 Mbit/s, elle est utilisée dans les applications envoyant un volume réduit de données.

6. Wi-Fi :



Le Wi-Fi fournit une connexion sans fil à la fois pour Internet et pour les réseaux locaux (LAN). Aujourd'hui de nombreux appareils se connectent à Internet et aux points réseau pour accéder aux ressources. Le Wi-Fi est considéré donc comme une technologie permettant l'IoT, car il est un moyen de communication des objets.

7. ZigBee :



C'est une technologie créée par Zigbee Alliance et basée sur la norme IEEE 802.15.4. ZigBee est un protocole non IP cible les applications nécessitant des échanges de données relativement peu fréquents à de faibles vitesses de transmission (250 Kbit/s) sur un espace restreint et dans une portée de 100 m.

Les objets connectés avec Zigbee prennent trois rôles : Coordinateur, routeur, client.

Le coordinateur : est responsable de la gestion des clients, la formation et la maintenance du réseau. Chaque coordinateur peut se connecter à 8 objets (clients & routeurs).

Les routeurs : jouent le rôle d'un pont de données entre le client et le coordinateur.

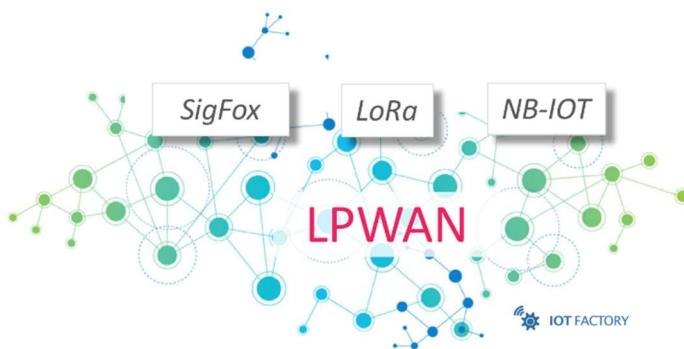
8. Z – Wave :



Z-Wave est une technologie télécoms RF à faible consommation, principalement conçue pour la domotique et les produits tels que les contrôleurs de lampe ou les capteurs.

Cette technologie est basée sur la norme Z-Wave Alliance ZAD12837/ITU-T G.9959, elle est caractérisée par une fréquence de 900MHz, une portée de 30m et un débit de 100 Kbit/s.

9. Technologies non cellulaires (LPWAN) :



Les réseaux longue portée, basse consommation (Low-power Wide-Area Network – LPWAN) sont une catégorie de réseaux très importante de l'IoT, ce sont les réseaux permettant à des capteurs de communiquer sur de longues distances, en consommant un minimum d'énergie.

LoRa, Sigfox et NB-IoT se démarquent dans les réseaux LPWAN.

- **LORA :**



LoRa correspond à la couche physique d'un réseau étendu à faible puissance et longue distance (LPWAN). LoRa utilise pour cela la technologie de modulation à spectre étalée CSS (Chirp Spread Spectrum) ; qui offre une communication sans fil longue portée et une faible consommation d'énergie. A la différence des réseaux cellulaires standards 3G, 4G ou 5G qui vous permet d'envoyer de grande quantité de données, LoRa envoie des messages de petites tailles, avec un débit modulable de 0,3 à 50 Ko par seconde.

- **LORAWAN :**



LoRaWAN (**Low Range Wide Area Network** - Réseau Étendue Longue Portée) est un protocole de communication radio qui constitue l'architecture du système et qui vous permet d'avoir une transmission de données bas débit mais surtout longue portée pour tous les objets connectés .

Le protocole réseau basse consommation LoRaWAN utilise la technologie de modulation LoRa afin de communiquer et gérer les dispositifs IoT. Le réseau LoRaWAN est composé de deux éléments essentiels : les **passerelles** : pour l'envoi et la réception des données aux nodes, et les **nodes** (nœuds) en tant que dispositifs finaux (capteurs IoT, etc...) qui reçoivent et envoient des informations à la passerelle.

10. Technologies cellulaires :



Les technologies de réseaux cellulaires auxquelles nous sommes habitués, telles que GSM (2G), UMTS(3G) et LTE(4G), ne sont pas particulièrement adaptées aux applications IoT. Pour cette raison, en 2016, le projet de partenariat de troisième génération 3GPP (l'organisation qui génère et maintient les normes pour les réseaux cellulaires) a publié une série de spécifications pour

les normes spécifiques à l'IoT: LTE-M (LTE for Machine Type Communication) et NB-IoT (Narrowband IoT).

LTE-M (ou e-MTC) : extension logicielle de la 4G LTE, dédiée au trafic M2M. LTE-M peut offrir un débit allant à 1Mbps.

NB-IoT : intégrée dans la 4G LTE et utilisant une interface radio distincte. NB-IoT offre un débit, adaptée aux applications IoT, de quelques dizaines de Kbps.

	Technologies non cellulaires		Technologies cellulaires		
	SigFox	LoRa	Nb-IoT	LTE-M	5G
Portée à l'extérieur	< 13 km	< 11 km	< 15 km	< 11 km	< 15 km
Bande Passante	100 Hz	< 500 kHz	200 kHz	1.4 MHz	
Débit de données	< 100 bps	< 10 kbps	< 150 kbps	< 1 Mbps	< 1 Mbps
Duré de vie de la batterie	> 10 ans	> 10 ans	> 10 ans	> 10 ans	> 10 ans
Disponibilité	2009	2009	2016	2016	2020

VII. Les protocoles de communication IoT :

Application Protocol		DDS	CoAP	AMQP	MQTT	MQTT-NS	XMPP	HTTP REST
Service Discovery		mDNS			DNS-SD			
Infrastructure Protocols	Routing Protocol	RPL						
	Network Layer	6LoWPAN			IPv4/IPv6			
	Link Layer	IEEE 802.15.4						
	Physical/Device Layer	LTE-A	EPCglobal	IEEE 802.15.4	Z-Wave			
Influential Protocols		IEEE 1888.3, IPsec				IEEE 1905.1		

1. Service Web Rest

REST (Representational State Transfer) ou RESTful est un style d'architecture permettant de construire des applications (Web, Intranet, Web Service).

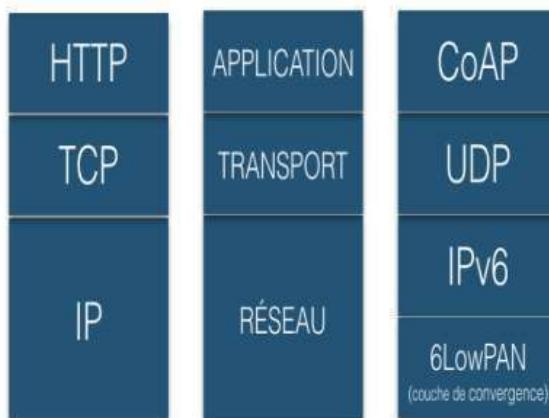
L'architecture REST utilise les spécifications originelles du protocole HTTP :

- l'URI (Uniform Resource Identifier) comme identifiant des ressources
- les verbes HTTP comme identifiant des opérations
- les réponses HTTP comme représentation des ressources. Ainsi, une ressource peut avoir plusieurs représentations dans des formats divers : HTML, XML, JSON, etc.

L'importance du REST découle de la simplicité de la communication et du fait qu'il est supporté par toutes les plateformes M2M Cloud.

2. CoAP (Constrained Application Protocol) :

C'est un protocole web basé sur une architecture client/serveur, est une version légère de REST conçu pour des communications UDP. CoAP est destiné à l'utilisation sur des appareils électroniques à faible consommation d'énergie. Il utilise les URI pour identifier les ressources.



3. MQTT (Message Queuing Telemetry Transport):



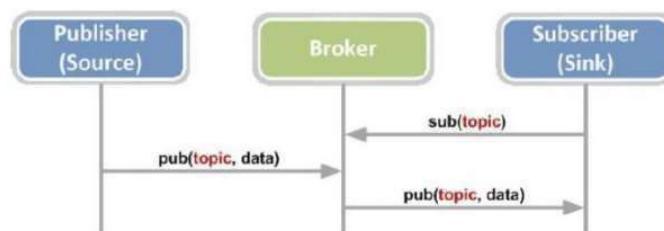
C'est un protocole de messagerie de type **publication/souscription** basé sur le protocole TCP/IP. Ce protocole spécialement dédié au monde du M2M et aux IOT. Il permet à deux équipements distants de communiquer via des messages de manière asynchrone avec une faible bande passante et une faible consommation en énergie. Il est utilisé pour faire communiquer des objets connectés qui collectent les différentes informations issues de capteurs intégrés et les envoyés via MQTT.

L'approche **publish/subscribe** classifie les messages par catégories (**Topics**) auxquelles les destinataires s'abonnent (**subscribe**).

Pour mettre en place une communication via MQTT, on aura d'une part le Broker ou serveur MQTT et d'autre part le client MQTT.

Le « client » MQTT peut être un dispositif matériel (un microcontrôleur, un nano ordinateur, un serveur...). Il utilise une librairie logicielle MQTT qui le permet de se connecter à un « Broker » MQTT. Le client qui envoie un message (topic) est nommé **publisher**, celui qui reçoit le message est nommé **subscriber**.

Le broker, connu par le publisher et le subscriber, identifie et authentifie les clients, réceptionne tous les messages, filtre les messages reçus et les distribue aux abonnées.



MQTT est caractérisé par :

- Une faible consommation d'énergie.
- Entêtes compressées MQTT topics sont structurées d'une façon hiérarchique.
- Les topics sont sensibles à la casse, codées en UTF-8 et doivent comporter au moins un caractère.
- Les topics peuvent être génériques : possibilité de faire des souscriptions à des topics qui ne sont pas encore définies.



VIII. Conclusion :

Dans ce chapitre nous avons présenté l'internet des objets, ses domaines d'applications et les principales technologie utilisées. Dans le chapitre suivant nous donnons une présentation générale de notre projet, les objectifs et les fonctionnalités de Smart-Home proposées.

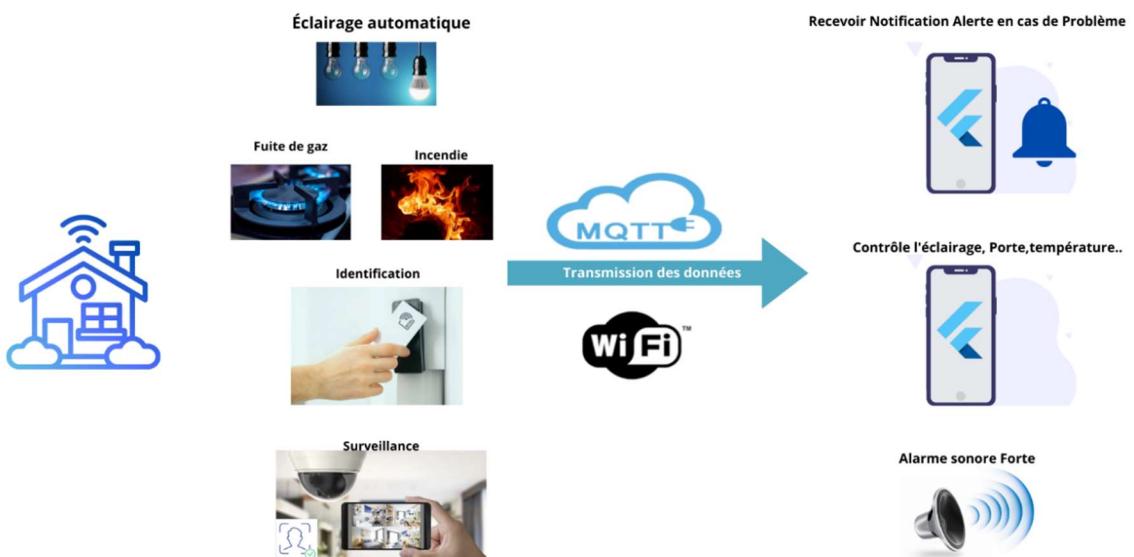
Chapitre 2 : Spécification des besoins

I. Introduction :

Les progrès technologiques de ces dernières années ont permis d'accroître les possibilités de conception de réseaux de capteurs dans plusieurs domaines que nous avons mentionnés précédemment, à savoir **la domotique**. Il est possible aujourd'hui de gérer la plupart des fonctions à partir d'un Smartphone, en intervenant sur toute la partie électronique d'une habitation : sécuriser la porte, ouvrir et fermer des volets à distance, contrôle d'éclairage, réglages des alarmes, vues des caméras de sécurité, contrôle du chauffage, reconnaissance vocale et autres.

Par définition, une maison intelligente est un ensemble d'objets connectés entre eux et contrôlés par un système de base : microcontrôleur. Aujourd'hui, ce concept de domotique a permis d'ajouter de l'intelligence à notre vie quotidienne, d'où l'orientation vers la maison intelligente. Cette dernière consiste en l'automatisation et la centralisation des différents objets de la maison afin d'assurer le contrôle, la surveillance et autres.

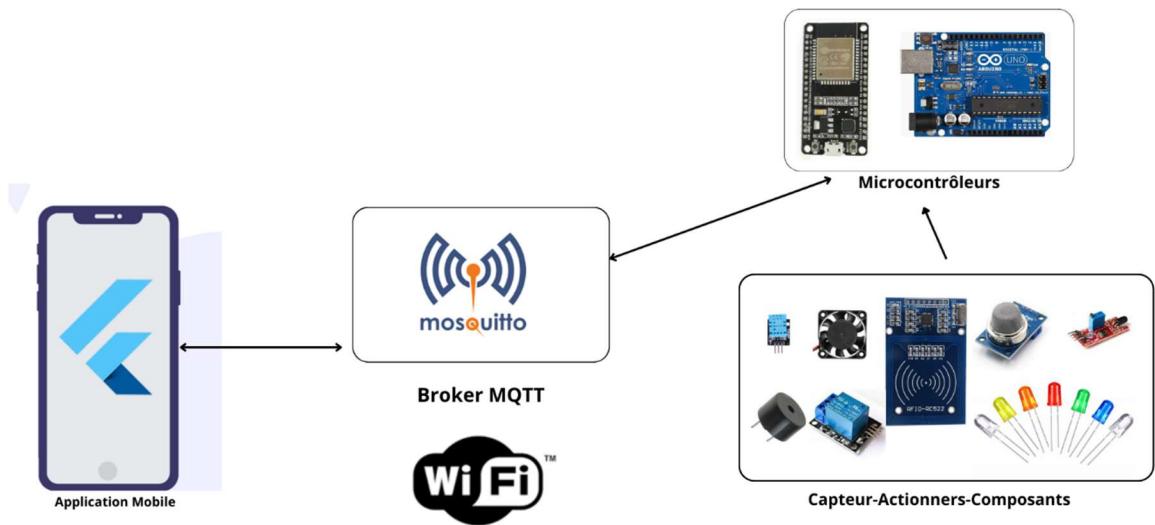
II. L'objectif du projet :



Notre mission consiste à réaliser une maison intelligente en implémentant des fonctions de domotique grâce à des modules et des capteurs moins chers, plus efficaces et plus faciles à utiliser conçus pour :

- Assurer un grand confort
- Assurer la sécurité qui est une priorité absolue pour la domotique.
- Assurer une économie d'énergie.

III. Cahier de charge :



Notre cahier des charges nous a redirigés vers les points suivants :

- Gestion d'automatisation d'éclairage.
- Acquisition de la température et d'humidité par un capteur de température DHT11.
- Automatisation d'ouverture et de fermeture des portes par un système RFID.
- La surveillance de fuites de gaz par un capteur MQ2 et envoie des alertes.
- Système de détection d'incendie par un capteur de flamme et envoie des alertes.
- Assurer la communication M2M par le protocole MQTT.
- La surveillance de la maison par une webcam.
- Développement d'une application mobile pour contrôler la maison.

IV. Les étapes de la réalisation :

La mise en œuvre de notre système a franchi six étapes principales :

- **Première étape :** C'est l'étape la plus importante car elle consiste en la recherche des informations théoriques et techniques à utiliser pour aboutir à notre objectif.
- **Deuxième étape :** Test du montage et implémentation du code sur le simulateur en ligne "wokwi" puis programmation de l'Arduino Uno et ESP32 avec L'IDE pour chaque montage réalisé.
- **Troisième étape :** L'installation et la configuration du broker MQTT sur la carte Raspberry Pi.
- **Quatrième étape :** réalisation du montage complet sur une plaque d'essai puis sur maquette afin d'assembler par la suite les composants en fonction de notre montage.
Et enfin la programmation du système par l'IDE.
- **Cinquième étape :** Streaming par une webcam sur la carte Raspberry Pi.
- **Sixième étape :** la création d'une application mobile avec flutter.

V. Conclusion :

Dans ce chapitre nous avons présenté les fonctionnalités qu'on va implémenter et la structure générale de notre SMART HOME. Le chapitre suivant portera sur l'analyse et la conception de notre système.

Chapitre 3 : Analyse et conception

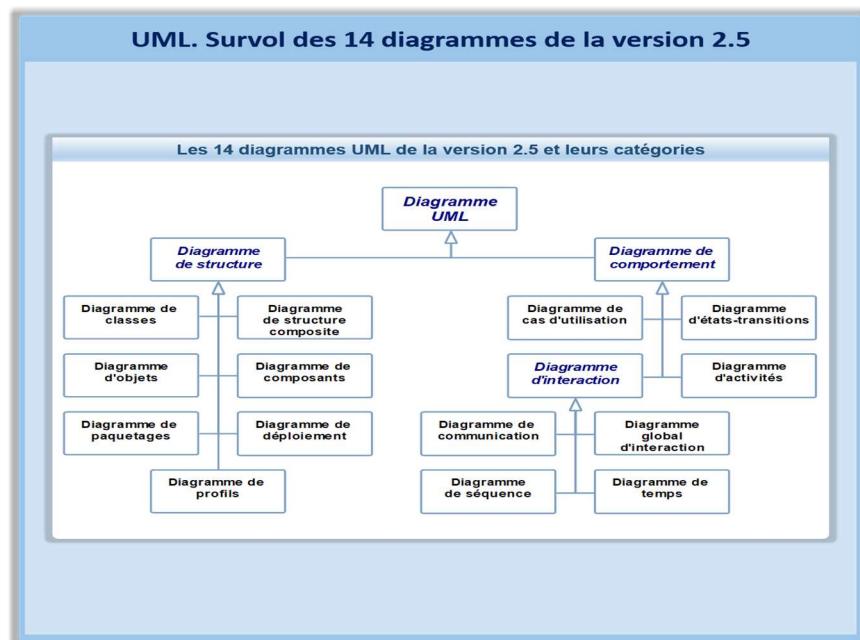
Introduction :

Dans cette partie, nous allons présenter des diagrammes qui permettent la modélisation et la description du fonctionnement du travail réalisé, pour faciliter la compréhension des concepts et visions que nous avons essayé d'appliquer à ce projet, en se basant sur le langage UML.

I. Présentation d'UML :

UML (Unified Modeling Language) a été développé pour établir un langage de modélisation visuel commun pour l'architecture, la conception et la mise en œuvre de la structure et du comportement des grands systèmes logiciels. En plus il est considéré comme un support de communication qui facilite la représentation et la compréhension de solutions.

En effet UML est un langage avec une syntaxe et des règles bien définies qui tentent à réaliser les buts décrits grâce à une représentation graphique formée de diagrammes et une modélisation textuelle qui vient enrichi la représentation graphique.



La conception de notre projet a été élaborée en suivant la démarche suivante :

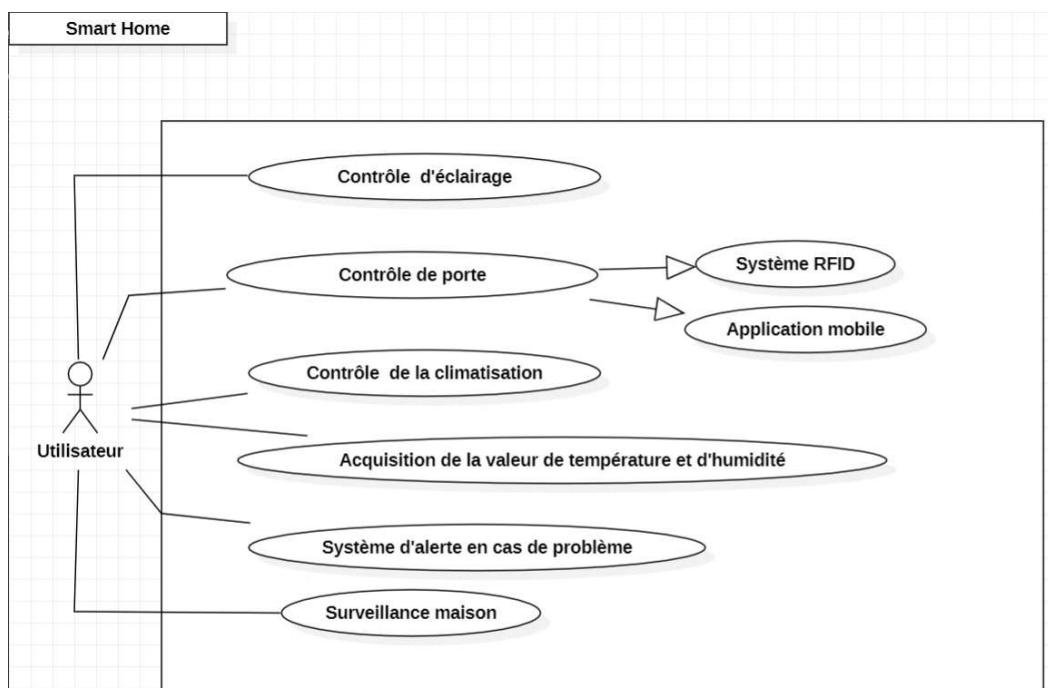
- ✓ Elaboration du diagramme de cas d'utilisation. Cette étape a été réalisée suite à la spécification fonctionnelle du cahier de charge.
- ✓ Elaboration des diagrammes de séquences pour mettre en évidence les interactions entre les différents objets du système.
- ✓ Elaboration du diagramme de déploiement qui présente l'architecture du système contenant du matériel et du logiciel.

II. Modèle des cas d'utilisations :

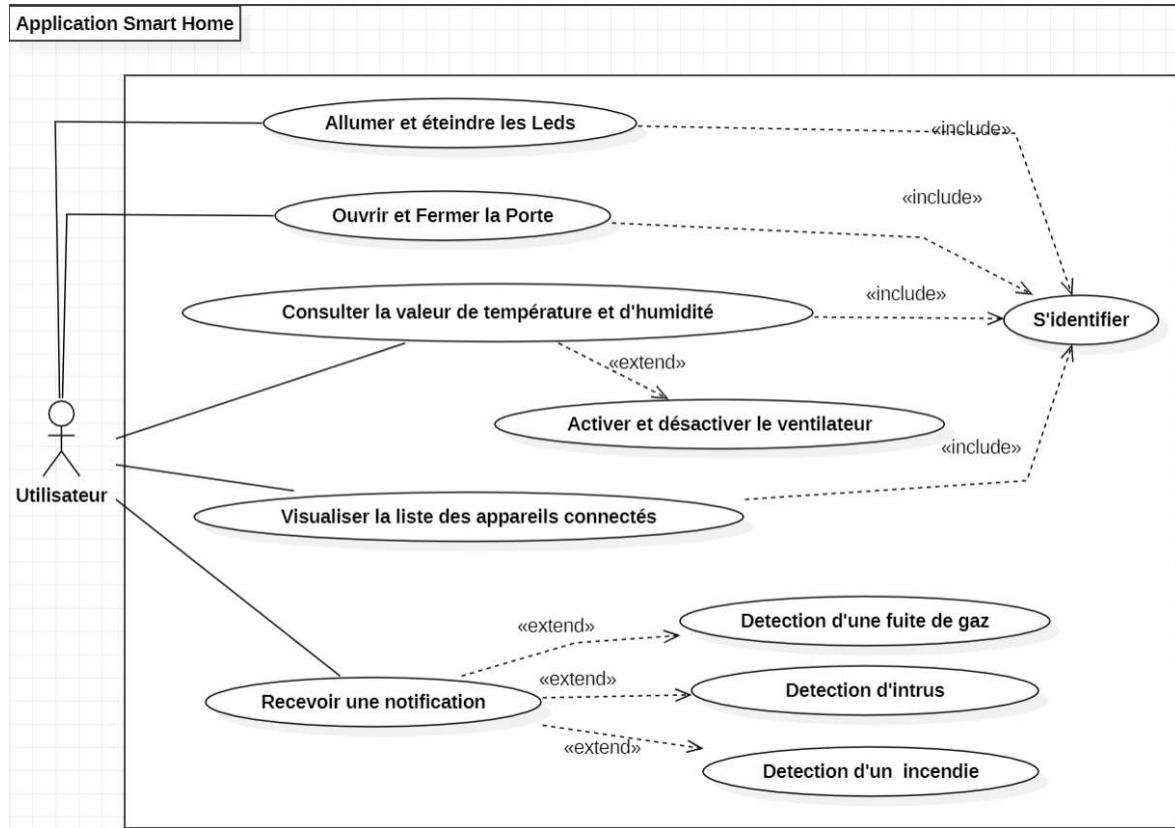
1. Diagramme de cas d'utilisation :

C'est un diagramme qui décrit les différents contextes dans lesquels un système peut être appliqué. Les diagrammes de cas d'utilisation nous fournissent un aperçu de haut niveau de ce qu'un système ou un élément d'un système se comporte sans plonger dans les détails de la mise en œuvre.

Le diagramme ci-dessous montre les principales fonctionnalités de notre smart home.



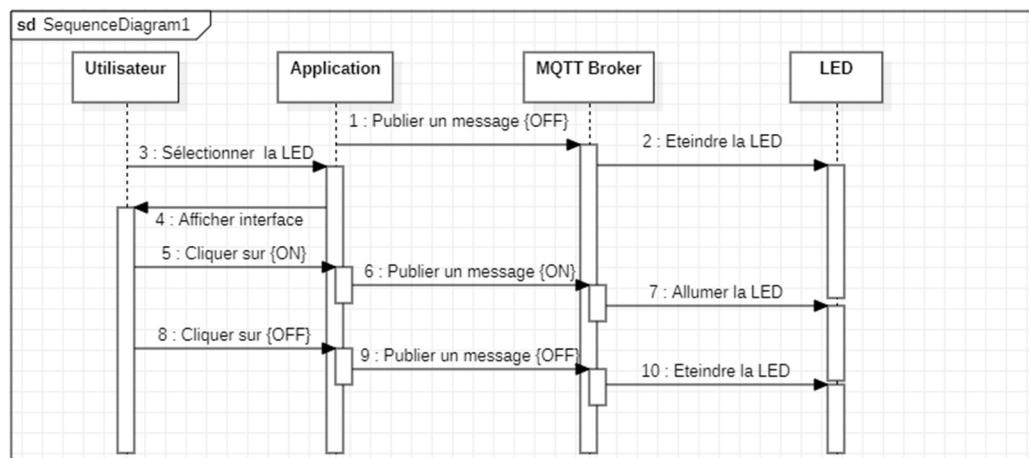
Ce diagramme illustre les fonctionnalités de l'application mobile qui contrôle notre smart home.



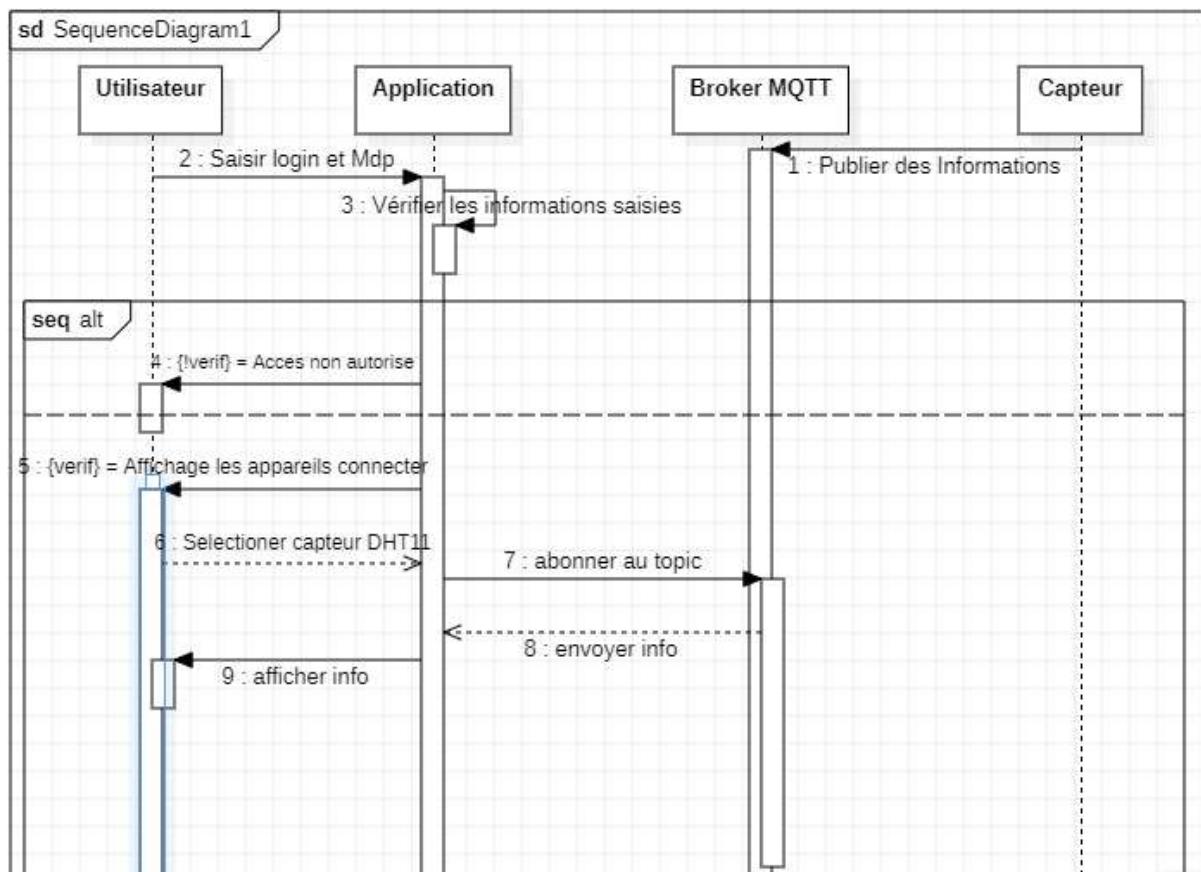
2. Diagramme de séquence :

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

- Diagramme de séquence de contrôle d'éclairage :



- Diagramme de séquence d'acquisition de température et d'humidité :



- Diagramme de séquence de contrôle de climatisation :

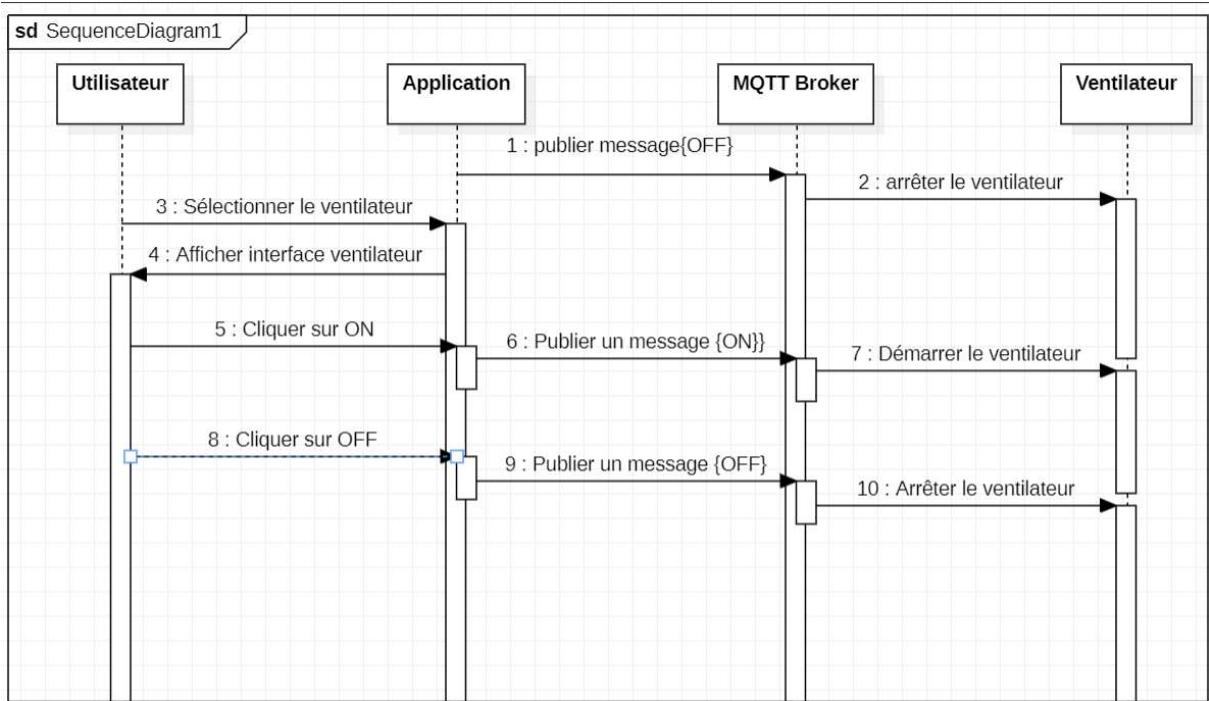
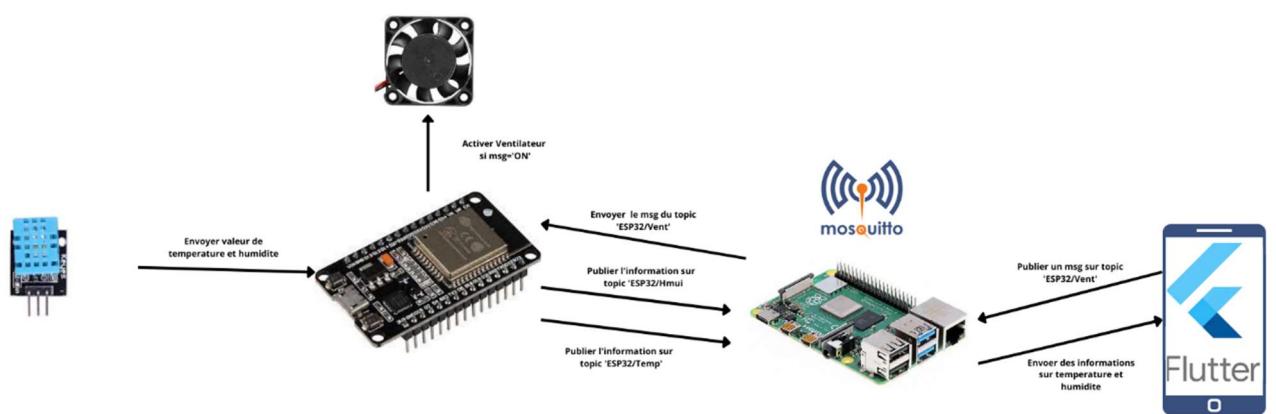


Schéma explicatifs d'un système de contrôle de climatisation :



- **Diagramme de séquence du système RFID :**

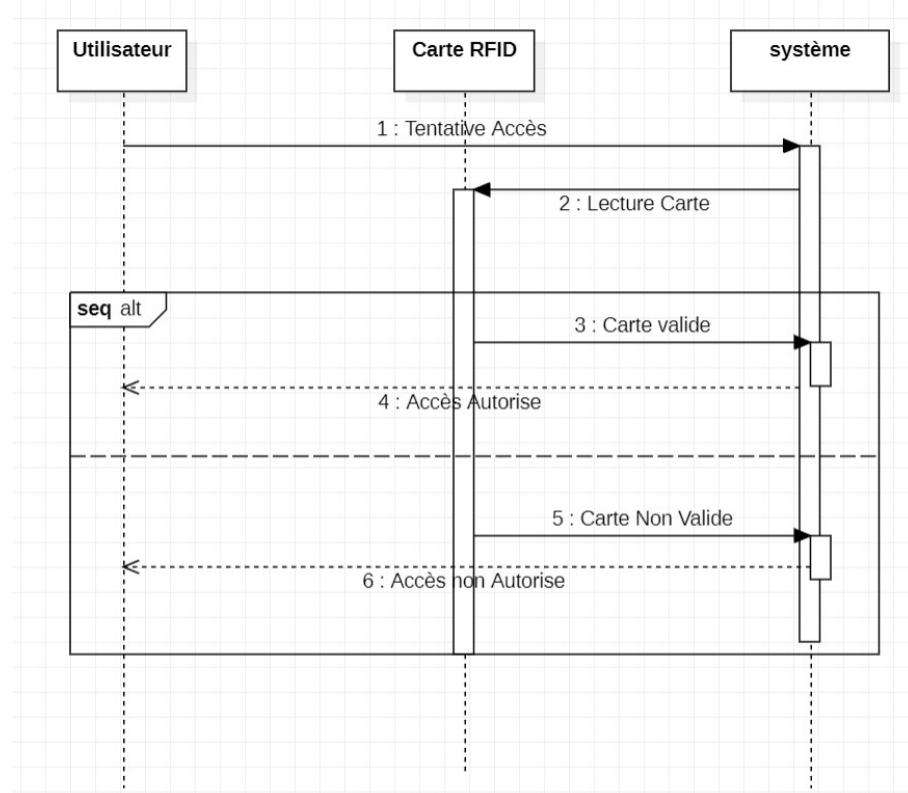
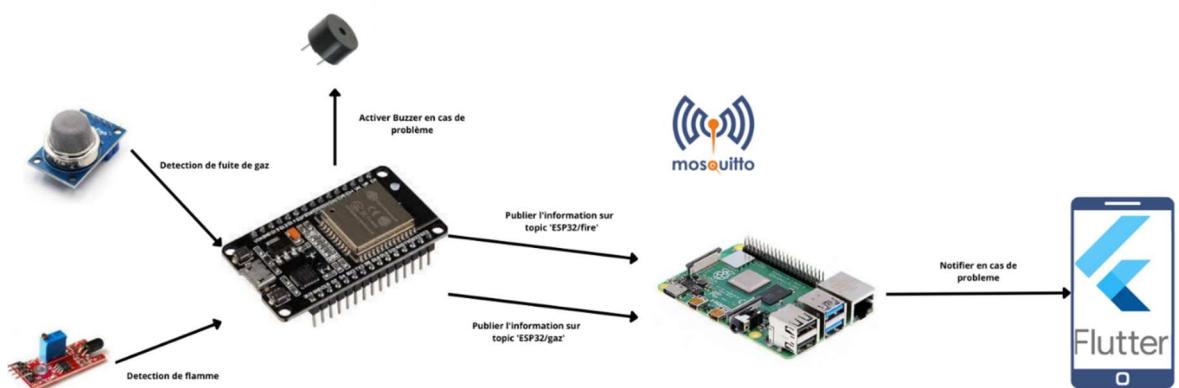


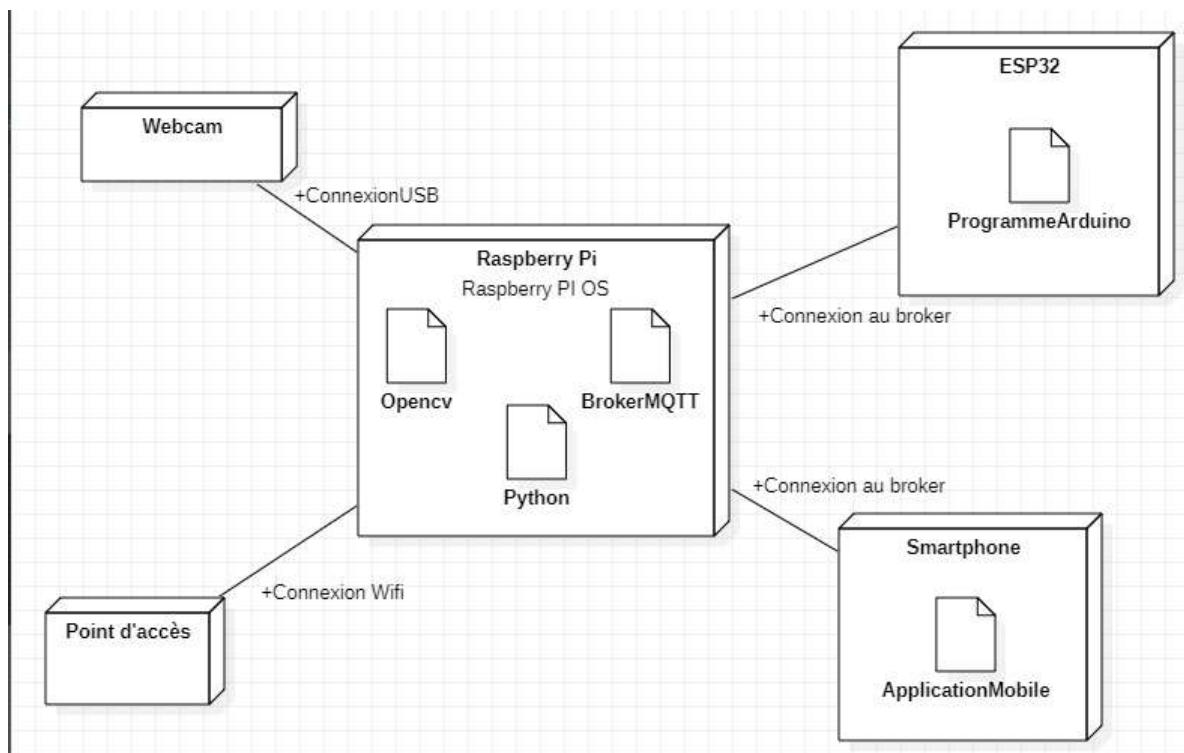
Schéma explicatif du système d'alerte d'incendie :

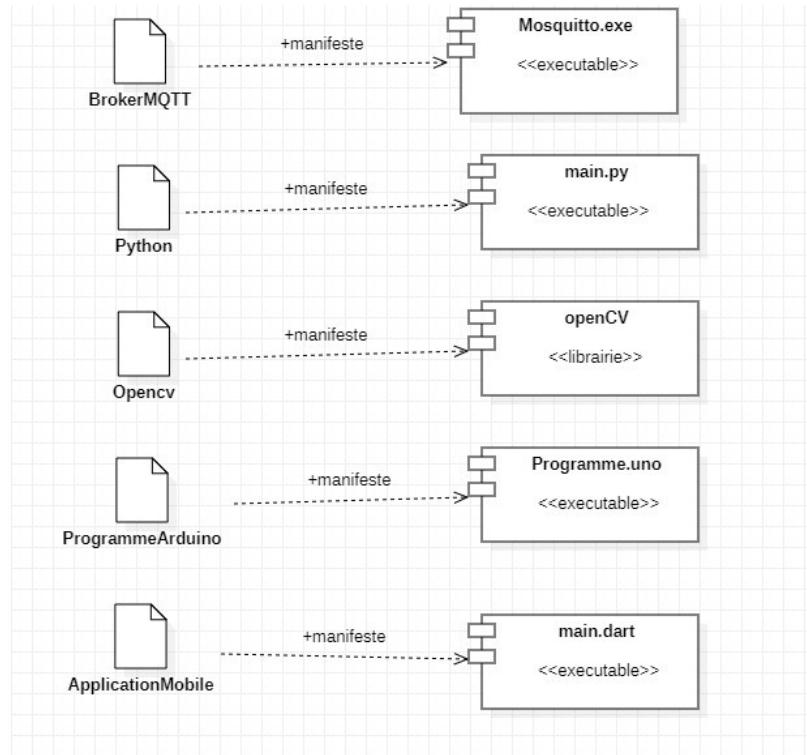


III. Diagramme de structure :

1. Diagramme de déploiement :

Un diagramme de déploiement se rapproche encore plus de la réalité physique, puisqu'ils identifient les éléments matériels (PC, Modem, Station de travail, Serveur, etc.), leur disposition physique (connexions) et la disposition des exécutables (représentés par des composants) sur ces éléments matériels.





VI. Conclusion :

Dans ce chapitre nous avons présenté la partie analyse et conception de notre système. Nous allons décrire dans le chapitre suivant, les composants logiciels et les matériels essentielles pour réaliser notre projet.

Chapitre 4 : Matériels et logiciels utilisés

I. Introduction :

Dans ce chapitre nous présenterons les outils matériels et les environnements logiciels utilisés pour développer notre système et ainsi que les différentes plateformes d'exécution de ses différentes parties.

II. Partie Matériel :

1. Cartes Electroniques :

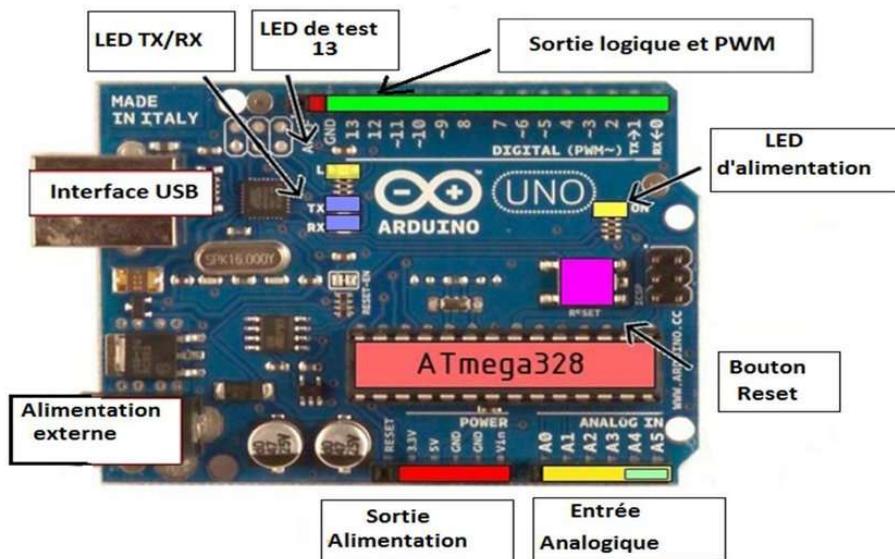
a. Arduino Uno :

L'Arduino Uno est un microcontrôleur programmable qui permet, comme son nom l'indique, de contrôler des éléments mécaniques : systèmes, lumières, moteurs, etc. Cette carte électronique permet donc à son utilisateur de programmer facilement des choses et de créer des mécanismes automatisés. Cette carte se programme en C++ à l'aide d'un logiciel de programmation gratuit et open-source fourni par Arduino (Arduino IDE).



- **Caractéristique de la carte Arduino :**

- Microcontrôleur ATmega328 : c'est le cœur de la carte Arduino. Il est responsable de la plupart des fonctions de la carte, telles que la gestion des entrées et sorties, la communication avec l'ordinateur et l'exécution des programmes que vous téléversez.
- Tension d'alimentation interne 5V
- Tension d'alimentation (recommandée) 7 à 12V, limites = 6 à 20 V
- Entrées/sorties numériques 14 dont 6 sorties PWM
- Entrées analogiques 6
- Courant max par broches E/S 40 mA
- Courant max sur sortie 3,3V 50mA
- Mémoire Flash 32 KB dont 0.5 KB utilisée par le bootloader
- Mémoire SRAM 2 KB
- Mémoire EEPROM 1 KB
- Fréquence horloge 16 MHz



- b. **ESP 32 :**

L'ESP32 est une carte électronique permettant de réaliser des projets «IoT» assez facilement. Elle possède en effet une connectivité assez complète. L'ESP32 est assez simple à prendre en main comme la carte Arduino. Il est donc possible de l'utiliser en utilisant le même langage de

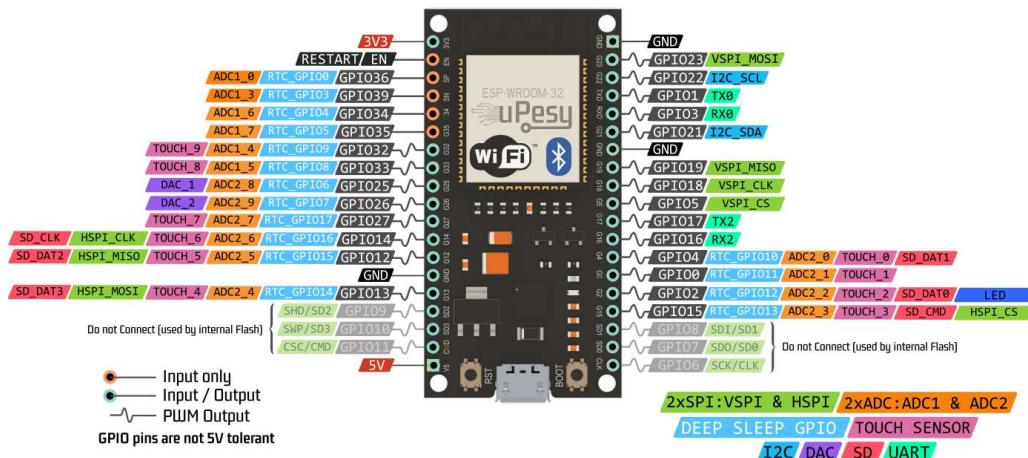
programmation, les mêmes modules complémentaires et surtout le même logiciel de compilation (Arduino IDE).



- **Caractéristiques :**

- Processeurs :
 - CPU : Xtensa double-cœur , microprocesseur LX 32 bits, fonctionnant à 240 MHz .
 - Coprocesseur ultra basse consommation (ULP) ;
- Mémoire :
 - 520 KiO SRAM ;
 - Flash : 4000 kB
 - RAM : 520 kB
 - EEPROM : 448 kB
- Connectivité sans-fil :
 - Wi-Fi, Bluetooth intégré et Bluetooth Low Energy (BLE)
- Broches de la carte ESP32 Wroom :

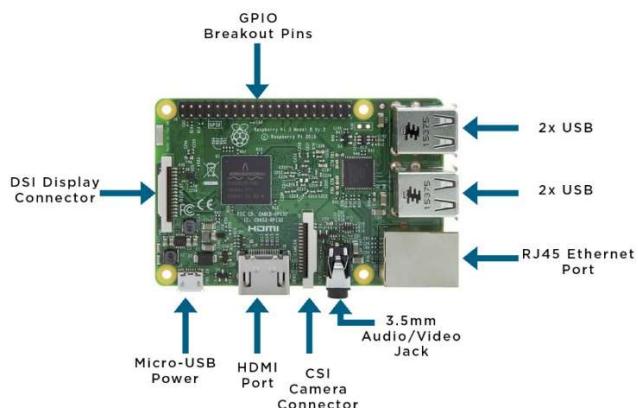
ESP32 Wroom DevKit Full Pinout



c. Raspberry Pi 3 Model B :

Le Raspberry Pi est un mini-ordinateur muni d'un processeur ARM. Les Raspberry sont de réels ordinateurs qui évoluent sur un système d'exploitation Raspberry Pi OS (dérivé du populaire Debian de la plateforme Linux). Le langage principal pour programmer sur Raspberry Pi est Python.

Elle peut être utilisée comme serveur, passerelle multimédia, Elle comporte ainsi plus de 512 Mb de mémoire RAM. Un port HDMI, une entrée pour caméra, des ports USB, Bluetooth, WiFi, port Webcam et parfois port Ethernet et carte son... Elle contient aussi des ports GPIO entrée/sortie permettant de piloter de nombreux capteurs et moteurs.



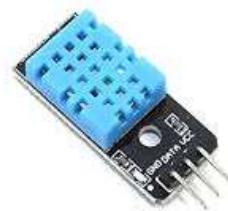
2. Capteurs :

Les capteurs sont des dispositifs utilisés pour détecter un événement ou une grandeur physique, tels que température, humidité, pression, etc. et qui fournit un signal électrique correspondant. Ce signal sera traité par un microcontrôleur pour l'interprétation, l'analyse et la prise de décision.

Les capteurs conçus pour les systèmes IoT sont généralement de petite taille, ont un faible coût et consomment moins d'énergie.

Dans notre projet on a utilisés les capteurs suivants :

a. Capteur de température et d'humidité :



Le capteur DHT11 est un capteur permettant d'acquérir une température et une humidité ambiante d'une manière numérique. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer la température et l'humidité de l'air et la transmettent d'une manière numérique sur un bus série.

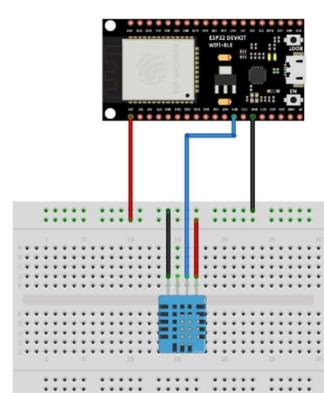
Les plages de mesures de ce capteur sont :

Température : entre 0 et 50 °c avec une tolérance de 2%

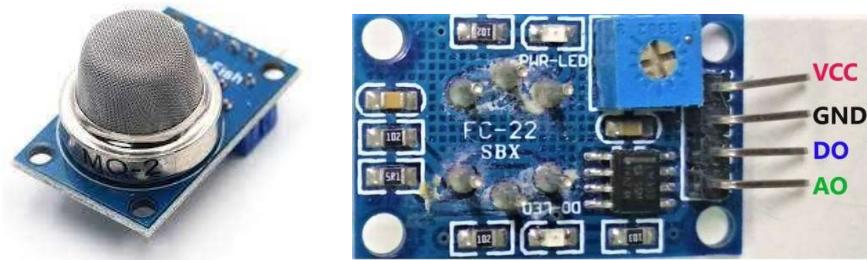
Humidité : entre 20 et 90 % avec une tolérance de 5 %

- Schéma:

Module	ESP32
DHT11	
GND	GND
VCC	3V3
DATA	14



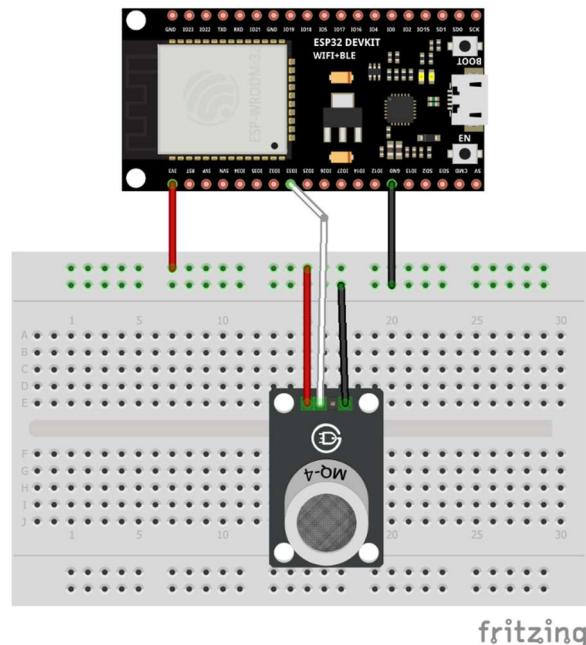
b. Capteur de détection du gaz MQ2 :



Le capteur de gaz inflammable et de fumée MQ2 détecte la concentration des gaz combustibles dans l'air et renvoie sa lecture comme tension analogique. Ce capteur est utilisé pour la détection des fuites de gaz dans les maisons, les environnements industriels, etc.. Le MQ-2 mesure des concentrations de gaz de 100 à 10000 ppm.

- Branchements :**

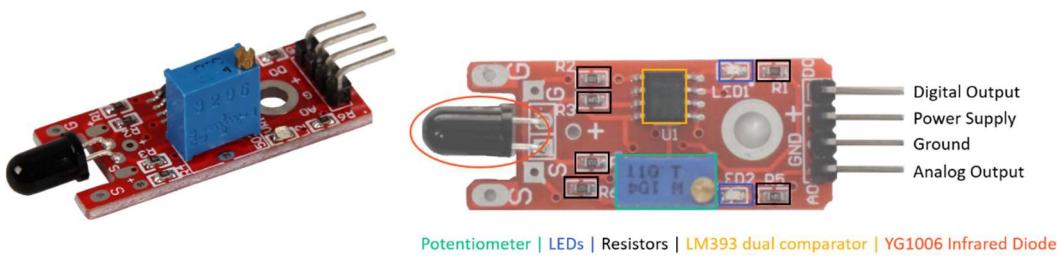
Module	ESP32
MQ2	
GND	GND
VCC	3V3
AO	34



c. Capteur de flamme :

Le module capteur de flamme KY-026 permet la détection de flamme à l'aide d'un récepteur infrarouge qui va capter les émissions lumineuses de sources de chaleur.

Toutes sources de chaleur (au-delà de 0 Kelvin) émettent une lumière infrarouge. Le module de détection de flamme KY-026 mesure l'intensité de la lumière infrarouge émise par le feu sur une plage de longueur d'onde comprise entre 760 à 1100 nm. Le module dispose de sorties numériques et analogiques et d'un potentiomètre pour régler la sensibilité. Utilisé couramment dans les systèmes de détection d'incendie.



Potentiometer | LEDs | Resistors | LM393 dual comparator | YG1006 Infrared Diode

Sortie numérique: Si une flamme est détectée, un signal est émis ici.

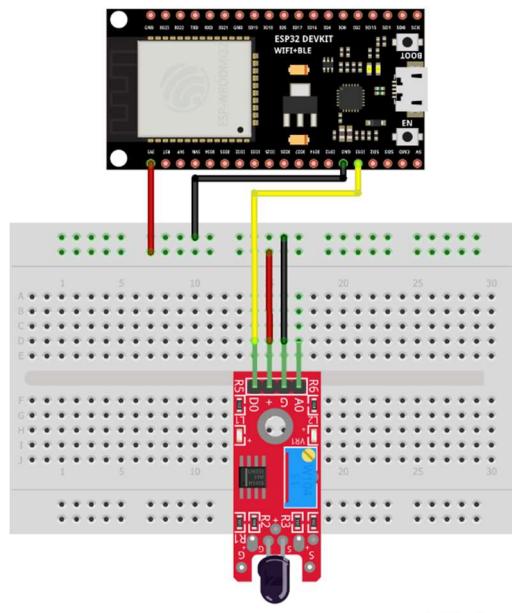
Sortie analogique: Valeur mesurée directe de l'unité de détection

LED1: Indique que le capteur est alimenté en tension.

LED2: Indique qu'une flamme a été détectée.

- **Branchements :**

Flamme Sensor	ESP32
GND	GND
VCC	3V3
DO	13



fritzing

3. Actionneur :

Il s'agit d'une technologie complémentaire aux capteurs, qui convertit l'énergie électrique en mouvement ou énergie mécanique.

Parmi les actionneurs utilisés dans notre projet sont :

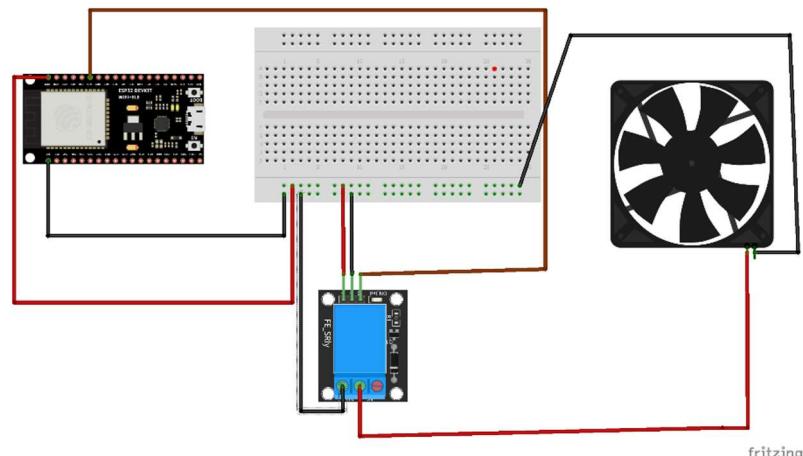
a. Ventilateur :



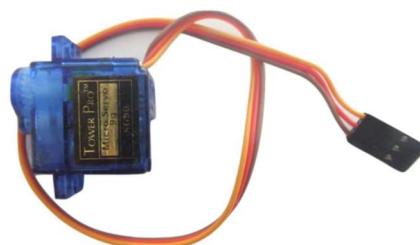
Un ventilateur de 5V est un dispositif de refroidissement électrique qui fonctionne à une tension de 5 volts. Il est souvent utilisé dans les projets électroniques pour refroidir des appareils ou des systèmes qui produisent de la chaleur.

- **Branchements :**

ESP32	Ventilateur	Relais
GND	Noire	---
--	Rouge	COM
GND	--	ON
GPIO 21	--	S
3V3	--	+
GND	--	-

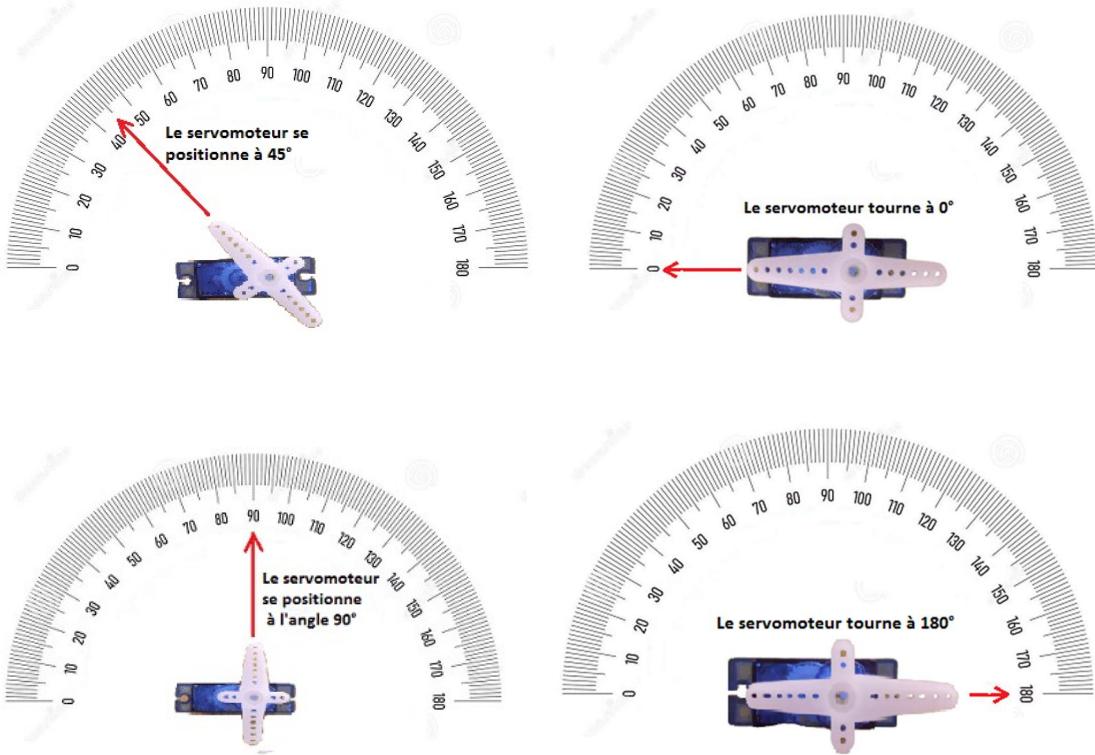


b. Servomoteur :

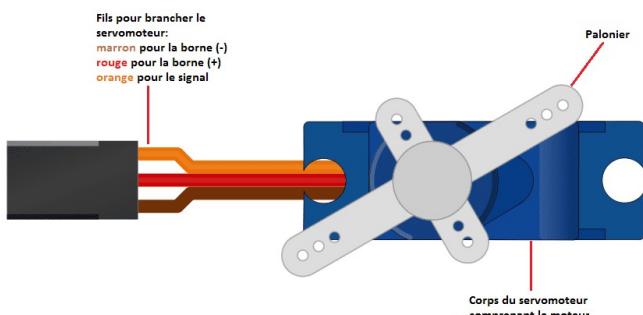


Un servomoteur est un type de moteur électrique qui peut être contrôlé de manière précise pour effectuer des mouvements de rotation limités. Ils sont souvent utilisés dans les robots et les appareils de contrôle de mouvement pour effectuer des tâches précises, telles que la manipulation d'objets ou la régulation de la vitesse.

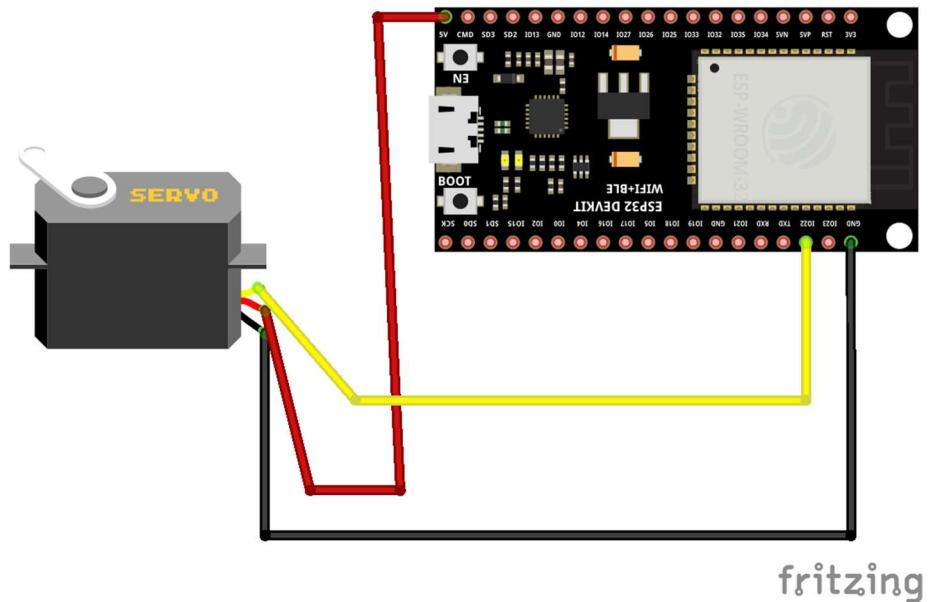
Exemples des positions du servomoteur :



- **Branchements :**



Correspondance des broches		
ESP32	Couleur du fil	Servomoteur SG90
GND	Marron	GND
5V ou 3V3	Rouge	5V
GPIO22	Orange	Signal PWM



c. Relais :



Un relais est un interrupteur électrique qui est utilisé pour mettre un circuit sous ou hors tension. Il est souvent utilisé pour contrôler un circuit à haute puissance avec un signal à faible puissance, comme un interrupteur ou un signal de commande informatique. Les relais sont utilisés dans une grande variété d'applications, notamment les systèmes de commande industrielle. Ils peuvent être utilisés pour contrôler des moteurs, des lumières et d'autres charges électriques. Certains relais sont conçus pour commuter rapidement un circuit sous et hors tension, tandis que d'autres sont conçus pour un contrôle plus précis et peuvent être utilisés pour moduler le courant qui traverse un circuit.

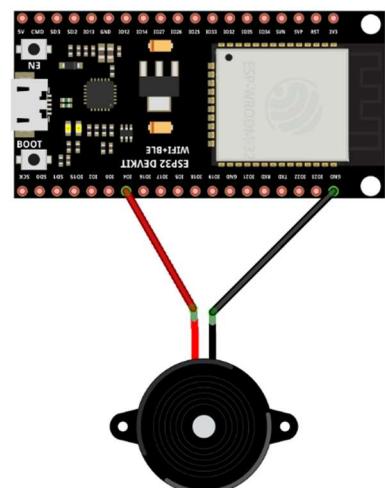
d. Buzzer :



Un buzzer ou un beeper est un dispositif de signalisation audio. Cet avertisseur sonore produit un son caractéristique quand on lui applique une tension. Ce son est principalement destiné à donner un signal ou à prévenir d'un danger. Le buzzer est bien audible grâce à sa puissance sonore.

- **Branchements :**

Buzzer	ESP32
+ (VCC)	5
- (GND)	GND



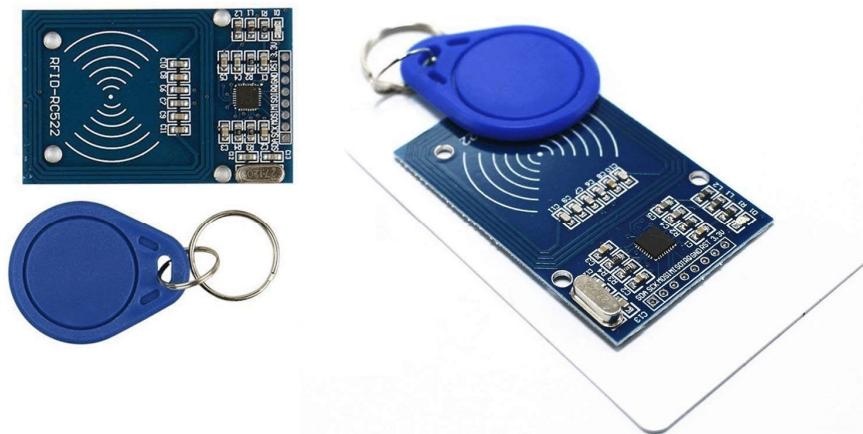
e. Leds:



Une diode électroluminescente (abrégé en DEL en français, ou LED, de l'anglais), est un dispositif capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.

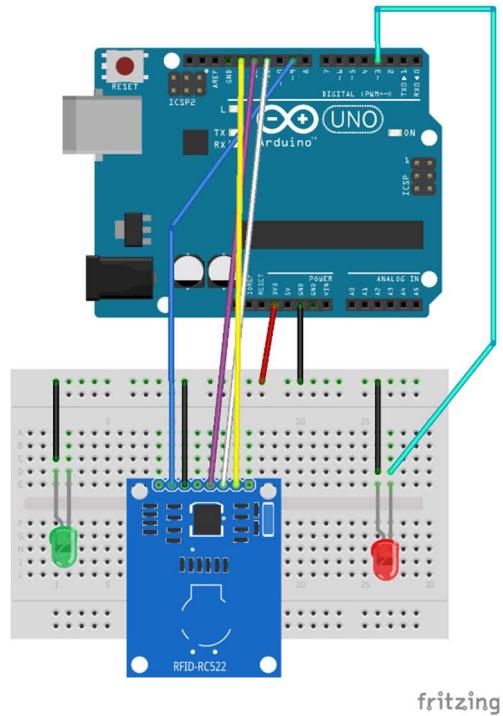
4. Autres Composants :

❖ Module RFID RC522:



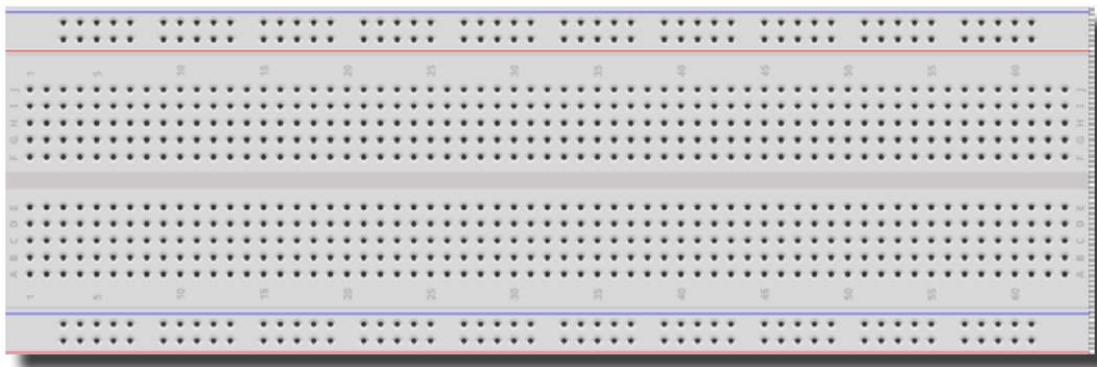
Le module RFID RC522 est un lecteur de carte à puce qui est utilisé pour l'identification des objets. Ce module permet entre autre, d'activer un mécanisme lorsque la bonne carte est présentée au lecteur.

- **Branchements :**

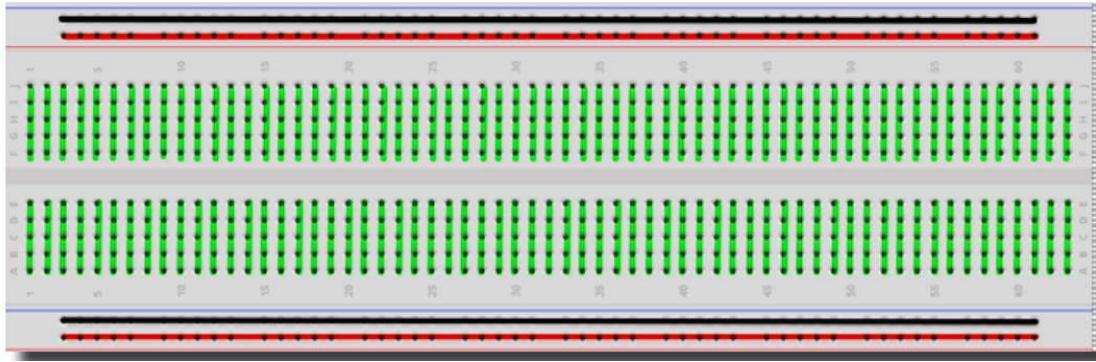


fritzing

- ❖ **Plaque d'essai :**



Une plaque d'essai, aussi connue comme breadboard, est un tableau composé d'orifices électriquement connectés entre eux de façon interne. Sur cette plaque on peut insérer les éléments électroniques et les fils pour le montage et prototypage de circuits électroniques. Elle sert à créer et à tester des prototypes de circuits électroniques. L'objectif est de pouvoir essayer sur elle nos projets de façon simple étant totalement fonctionnels, et pouvoir, aussi, les modifier facilement s'il s'avère nécessaire.



- En vert, les trous sont reliés entre eux. Il y a une continuité électrique.
- En rouge, les trous sont reliés entre eux. Il y a une continuité électrique. On attribue cette ligne pour le « + » (positif).
- En noir, les trous sont reliés entre eux. Il y a une continuité électrique. On attribue cette ligne pour le « - » (négatif ou masse)

❖ Des fils de connexion :



Les fils de connexion sont des fils électriques utilisés pour connecter des composants électroniques à une carte électronique. Ils sont généralement utilisés pour connecter des capteurs, des actionneurs, des afficheurs et d'autres composants à la carte afin de créer des circuits électroniques. Ils sont disponibles dans une variété de couleurs pour aider à identifier et organiser les différents fils dans un circuit.

Il existe trois types de fils de connexion: les fils de connexion mâle-mâle et les fils de connexion mâle-femelle. Les fils de connexion mâle-mâle sont utilisés pour connecter des composants qui ont tous deux des broches mâles, tandis que les fils de connexion mâle-

femelle sont utilisés pour connecter des composants avec une broche mâle et une broche femelle.



❖ Web Caméra :



C'est une caméra d'enregistrement vidéo USB Webcam HD 480P en direct.

III. Partie Logiciels :

Nous avons utilisé les logiciels suivants :

1. Fritzing :



Fritzing est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon. Il est adapté aux débutants ou confirmés en électronique pour faire rapidement des circuits simples, et également un bon outil didactique pour apprendre à bidouiller en électronique par la pratique.

Le logiciel comporte trois vues principales :

- La « **Platine d'essai** », où l'on voit les composants tels qu'ils sont dans la réalité et où l'on construit le montage.
- La « **Vue schématique** », qui représente le schéma fonctionnel du circuit.
- Le « **Circuit imprimé** », qui représente la vue du circuit imprimé tel qu'il sera sorti en PDF pour être imprimé.

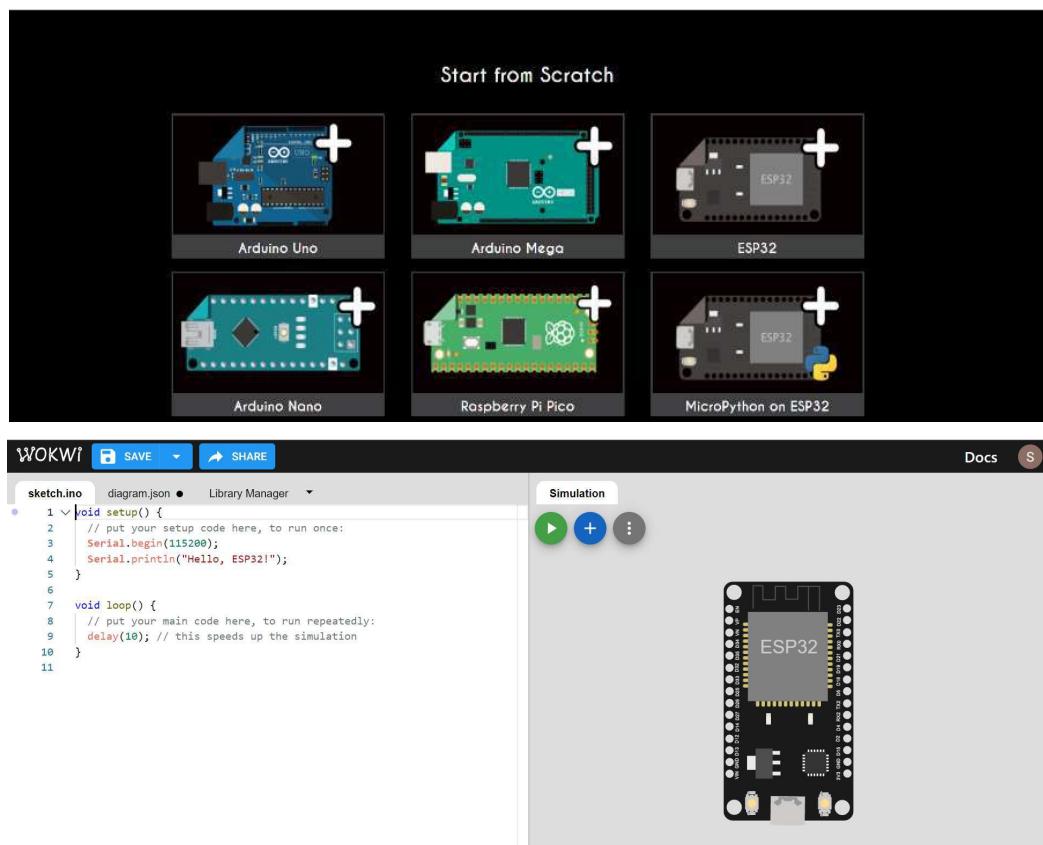
2. Plateforme de programmation de ESP32 et Arduino :



Pour programmer les cartes de notre projet ESP32 et Arduino on a utilisé le logiciel open source : Arduino Ide. C'est un environnement de développement intégré fonctionnant sur divers systèmes d'exploitation (Windows, Mac OS, Gnu/Linux) qui permet d'écrire le programme sur un ordinateur et de le transférer via le port USB. Le langage de programmation est très proche du C++, qui est l'un des langages les plus courants.

3. Wokwi :

Avant de tester en réalité des programmes avec des capteurs physiques, on a recouru à un simulateur en ligne wokwi qui nous permet de tester le code ainsi que le montage pour éviter les erreurs et endommager le matériel.



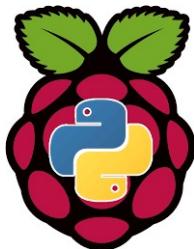
4. Broker MQTT: Mosquitto :



Eclipse Mosquitto est un courtier de messages open source qui implémente les versions 5.0, 3.1.1 et 3.1 du protocole MQTT. Mosquitto est léger et convient pour une utilisation

sur tous les appareils, des ordinateurs mono-carte à faible puissance aux serveurs complets. Il comprend également une bibliothèque client C et C++, ainsi que les utilitaires **mosquitto_pub** et **mosquitto_sub** pour la publication et l'abonnement.

5. Python



Python est un puissant langage de programmation dynamique qui est utilisé dans différents domaines (application, web, ...). On a utilisé ce langage pour lancer le stream sur la carte raspberry.

Les principaux points forts de Python sont :

- Langage adapté aux débutants
- Les grandes possibilités du langage. Il va aussi bien être adapté aux petits projets qu'aux gros projets.
- Peut-être cross-platform
- Stable, langage qui est assez vieux et qui a fait ses preuves
- Simple à comprendre

6. Opencv :



Initialement développée par Intel, OpenCV (Open Computer Vision) est une bibliothèque graphique. Elle est spécialisée dans le traitement d'images, que ce soit pour de la photo ou de la vidéo. Sa première version est sortie en juin 2000. Elle est disponible sur la plupart des systèmes d'exploitation et existe pour les langages Python, C++ et Java.

7. Flask



Flask est un micro framework open-source de développement web en Python. Il est classé comme micro framework car il est très léger. Il est utilisé pour développer des applications web sur un Raspberry Pi. On l'a utilisé pour lancer un flux

vidéo sur une page web.

8. Visual Studio code



Visual Studio Code est un éditeur de code source léger, mais puissant, qui s'exécute sur le bureau, il est disponible pour Windows, macOS et Linux. Inclut la prise en charge intégrée de JavaScript, TypeScript et

Node.js, et possède un écosystème enrichi d'extensions pour d'autres langages et runtimes tel que Dart, flutter...

9. Flutter :



Flutter est un framework d'application mobile open source qui est une solution multiplateforme pour créer des applications de haute qualité qui fonctionne sur iOS et Android. Il utilise un cadre réactif, ce qui permet de créer rapidement et facilement de belles applications mobiles. Il utilise un langage de programmation créé par Google : le Dart. C'est un langage de programmation orientée-objet comme Swift ou Kotlin.

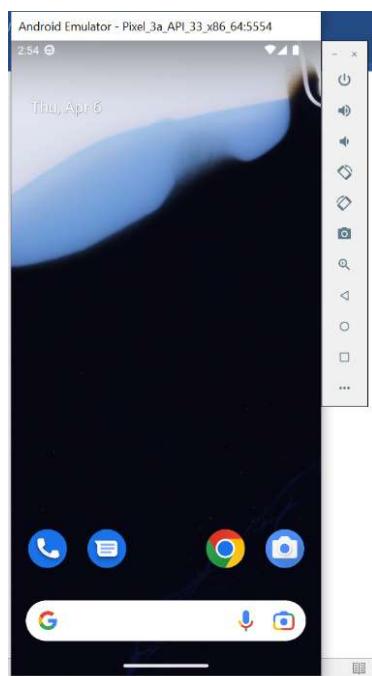
Ce framework se caractérise par sa performance rapides et fluides, il utilise un système de widgets personnalisables pour créer des interfaces utilisateur interactives et

dynamiques. Les widgets peuvent être personnalisés pour s'adapter à n'importe quelle conception ou thème.

10. Android Studio



L'émulateur Android est un périphérique virtuel Android (AVD, Android Virtual Device) qui représente un périphérique Android spécifique. On a utilisé cet émulateur Android en tant que plate-forme cible pour exécuter et tester l'application Android sur notre PC.



IV. Conclusion :

Ce chapitre a présenté l'ensemble des outils et des logiciels que le système requiert. Le chapitre suivant expose les phases de la mise en œuvre de notre smart home.

Chapitre 5 : Réalisation et mise au point

I. Introduction :

Après une étude approfondie des concepts théoriques des techniques à implémenter, et après la modélisation des fonctionnalités du système. Dans ce chapitre, nous mettons l'accent sur le processus de mise en œuvre, ce processus comprend 5 parties essentielles :

La première partie concerne la préparation de la maquette du prototype.

La deuxième partie concerne la réalisation du Montage globale qui englobe tous les composants.

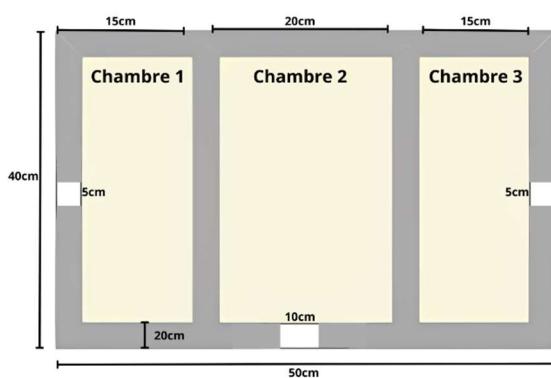
La troisième partie concerne l'installation du montage globale sur la maquette déjà préparée.

La quatrième partie concerne la préparation et la configuration de la carte Raspberry.

La cinquième et la dernière partie concerne le développement de l'application Mobile sur Flutter.

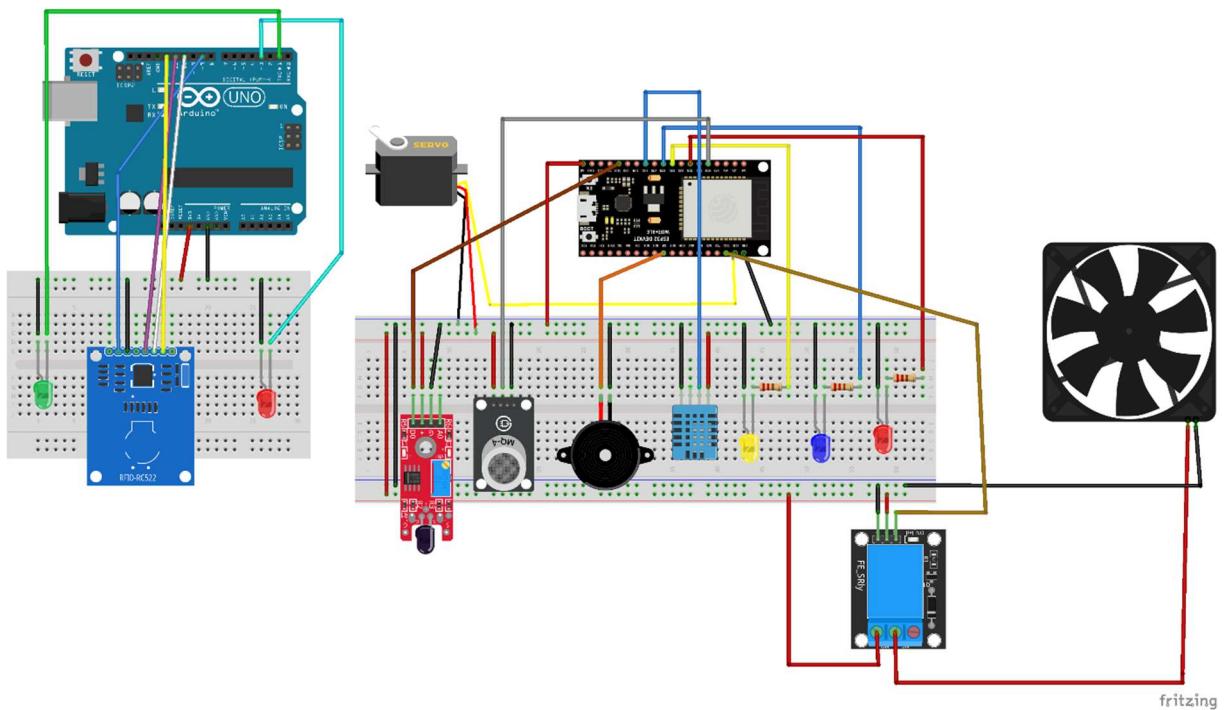
II. Préparation de la maquette :

La première étape consiste à créer une maison en bois pour le prototype. Pour cela nous avons préparés une maquette qui se constitue de trois chambres.



III. Schéma globale du Prototype :

Le schéma ci-dessous représente le montage global de notre système réalisé sur Fritzing.



IV. Installation des Composants :

Les figures ci-dessous montrent le prototype réalisé.

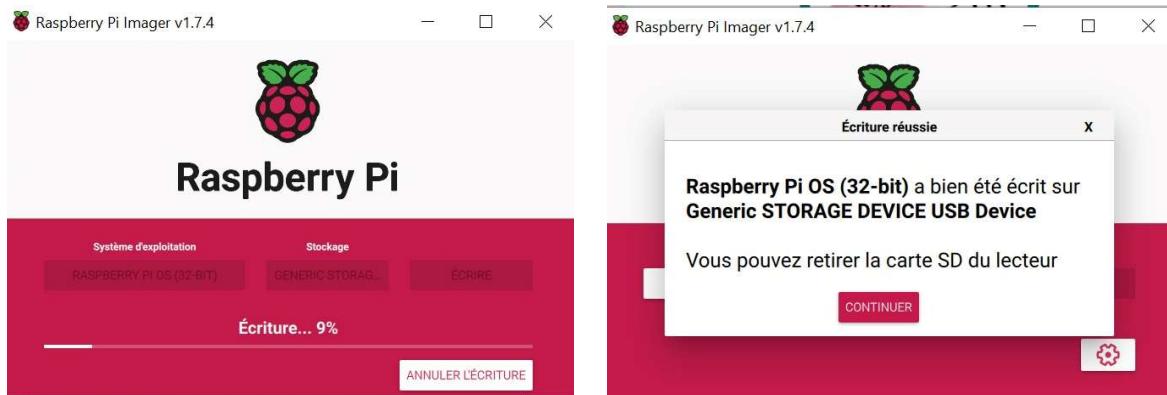


V. Préparation de Raspberry :

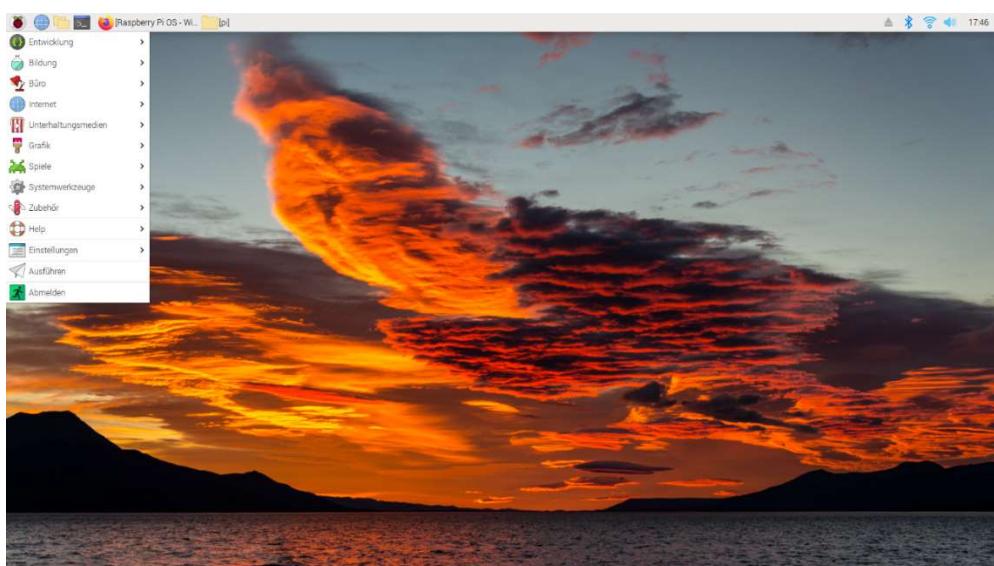
1. Installation du Système d'exploitation :

Avec Raspberry Pi, on peut pratiquement faire tout ce que l'on peut faire avec un ordinateur ou un serveur sous Linux. Le Raspberry Pi utilise une carte SD à la place d'un disque dur, elle a besoin d'un système d'exploitation pour fonctionner.

Pour faire l'installation du système d'exploitation, tout d'abord on a besoin d'une carte microSD avec un système d'exploitation (Raspberry PI OS 32-bit) gravé dessus.



Par ailleurs il nous faudra également un clavier, souris, et un vidéoprojecteur. Après l'insertion de la carte SD dans son support. On branche un clavier USB et une souris au Raspberry Pi, on connecte la sortie HDMI au vidéoprojecteur et on branche l'alimentation du Pi.



2. Installation du Broker MQTT : Mosquitto

- On commence par mettre à jour la liste de paquets et la mise à niveau des paquets :

```
sudo apt update && sudo apt upgrade
```

- On installe Mosquitto :

```
sudo apt install mosquitto mosquitto-clients -y
```

- Après l'installation, on active le service Mosquitto et on vérifie la version MQTT à l'aide des commandes suivantes.

```
sudo systemctl enable mosquitto.service
mosquitto -v
```

- On Crée un compte utilisateur pour permettre l'accès à distance authentifié en utilisant la commande suivante.

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd admin
```

- On ouvre le fichier mosquitto.conf dans l'éditeur nano en utilisant la commande suivante.

```
sudo nano /etc/mosquitto.conf
```

- On ajoute ces lignes dans le fichier mosquitto.conf

```
per_listener_settings true
pid_file /run/mosquitto/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
include_dir /etc/mosquitto/conf.d
allow_anonymous false
listener 1883
password_file /etc/mosquitto/passwd
```

- On redémarre le service :

```
sudo systemctl restart mosquitto
```

- On teste le status :

```
sudo systemctl status mosquitto
```

3. Configuration du Caméra et Raspberry :

Parmi les fonctionnalités qu'on doit réaliser c'est d'assurer la surveillance de la maison.

Pour cela on a utilisé notre carte raspberry et une webcam pour lancer le stream et l'afficher dans l'application.

Premièrement, on a installé la librairie **OpenCV** pour capturer le flux video et détecter les visages. Ce flux va être diffuser dans notre application en utilisant la librairie **Flask**.

Pour installer OpenCV, on exécute cette commande :

```
sudo apt-get install python3-opencv
```

Pour installer Flask, on exécute cette commande :

```
Pip install Flask
```

Maintenant, l'environnement est prêt. Alors pour démarrer le Stream il faut exécuter le script python.

VI. Développement de l'application :

Nous avons développé une application mobile pour piloter notre système en utilisant le framework **Flutter**, ce qui constitue une solution légère.

1. Connection au Broker :

Pour que notre application mobile se connecte au Broker MQTT il faut ajouter des packages de MQTT qui sont compatibles avec notre broker :

Pour cela on ajoute ces lignes dans le fichier *pubspec.yaml*. Ce fichier permet de spécifier les dépendances nécessaires pour notre projet Flutter. On peut ajouter des packages qui seront nécessaires pour notre application. On peut aussi spécifier la version souhaitée pour chaque dépendance.

```
dependencies:
  flutter:
    mqtt_client: ^9.7.4
    flutter_mqtt: ^1.0.0+5
```

Ces librairies vont nous aider à exploiter les fonctionnalités du Broker MQTT .

```
3 import 'package:mqtt_client/mqtt_client.dart';
4 import 'package:mqtt_client/mqtt_server_client.dart';
```

Cette ligne nous permet de créer un objet client MQTT en spécifiant l'adresse IP de notre serveur MQTT, le nom de client MQTT (App Flutter) et le numéro de port de notre serveur.

```
14     client = MqttServerClient.withPort('192.168.147.136', 'APP_flutter', 1883);
```

Le rôle de ces lignes de code est de se connecter à un serveur MQTT en utilisant les informations de connexion fournies dans connMessage, d'afficher une exception si la connexion échoue, puis de fermer la connexion MQTT en cas d'erreur.

```
--  
32     client.connectionMessage = connMessage;  
33     try {  
34         await client.connect();  
35     } catch (e) {  
36         print('Exception: $e');  
37         client.disconnect();  
38     }
```

On peut s'abonner à un topic grâce à la fonction **subscribe()**.

```
50     const topic1 = 'Home/Chambre1/temp';  
51     client.subscribe(topic1);
```

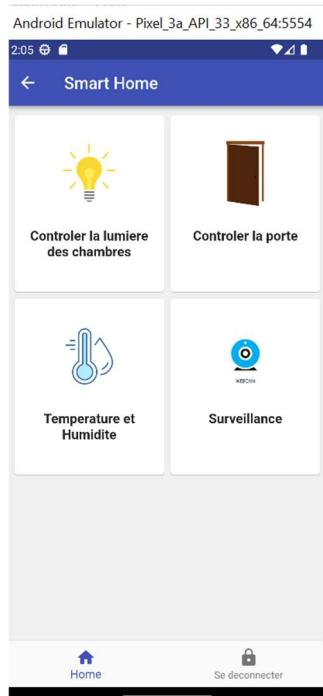
On publie des messages à un topic grâce à la fonction **PublishMessage()**.

```
126     void publishMessageVent(String message) {  
127         const pubTopic = 'Home/Chambre1/Vent';  
128         final builder = MqttClientPayloadBuilder();  
129         builder.addString(message);  
130  
131         if (client.connectionStatus?.state == MqttConnectionState.connected) {  
132             client.publishMessage(pubTopic, MqttQos.atMostOnce, builder.payload!);  
133         }  
134     }  
135 }
```

2. Présentation des interfaces :



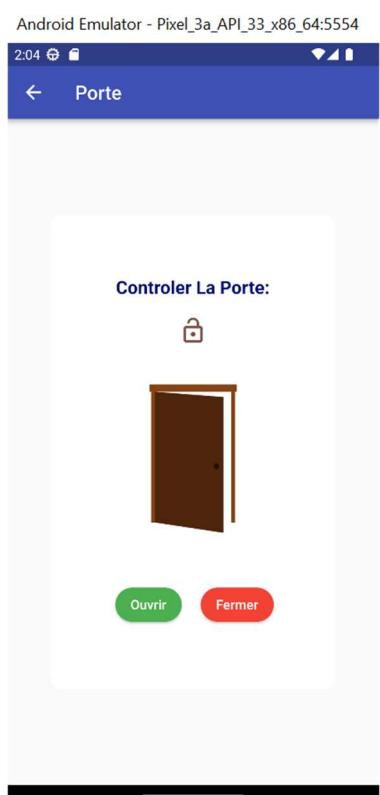
Via cette interface l'utilisateur peut se connecter à l'application en saisissant son login et mot de passe. Il va être dirigé ainsi vers la page d'accueil.



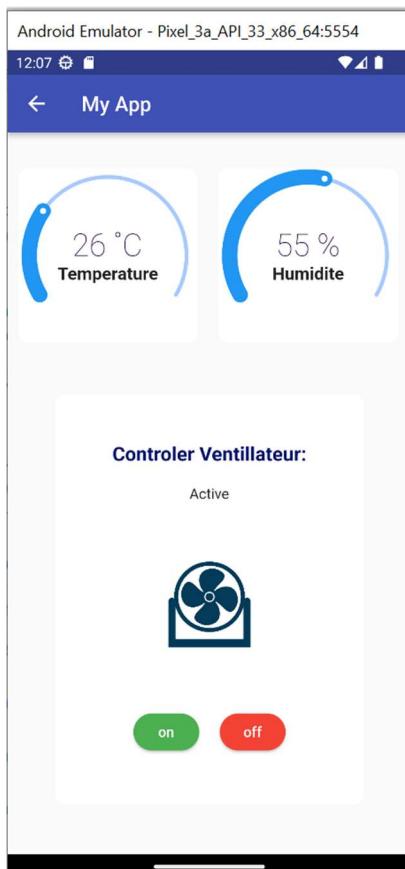
A travers cette interface l'utilisateur peut choisir la fonctionnalité qu'il désire : contrôler l'éclairage, ouverture et fermeture de porte, Contrôle de température et d'humidité et surveillance.



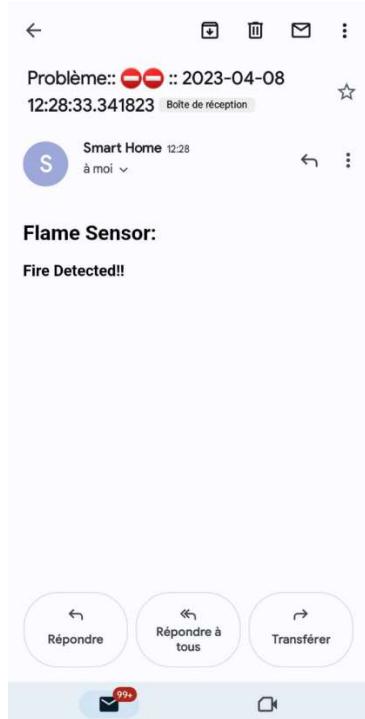
Cette interface permet à l'utilisateur d'allumer et d'éteindre les leds.



Cette interface permet à l'utilisateur d'ouvrir et de fermer la porte.



A partir de cette interface, l'utilisateur peut consulter la valeur de température et d'humidité à l'intérieur de sa maison comme il peut activer et désactiver le ventilateur selon le besoin.



En cas de problème : détection de flamme ou de fuite de gaz, l'utilisateur reçoit une notification par Gmail.



A travers cette interface, l'utilisateur peut accéder au flux vidéo pour surveiller sa maison à distance.

Conclusion :

En conclusion de ce chapitre dédié à la réalisation et à la mise au point de notre projet de fin d'études, nous pouvons affirmer que notre objectif principal a été atteint avec succès. Nous avons pu concevoir et développer un système IoT fonctionnel qui répond aux besoins identifiés dans notre cahier de charge.

Conclusion et perspective :

En conclusion, ce rapport de projet de fin d'études a présenté une étude approfondie et une réalisation réussie d'un système IoT. Nous avons conçu un prototype d'une maison automatisée. Elle est capable de gérer l'éclairage des chambres, de prévenir en cas d'incendie ou dans le cas de fuite de gaz aussi surveillé l'état de la température et l'humidité ainsi que la surveillance par une webcam. Ce projet nous a permis de nous amuser grâce à la manipulation du matériel, tout en acquérant de meilleures connaissances des applications de la domotique, ce qui pourrait nous être fortement utile pour notre vie professionnelle future. Bien sûr tout ce travail s'est déroulé dans les meilleures conditions possible, en effet une bonne cohésion et une bonne entente ont permis l'obtention d'un travail abouti et satisfaisant. Ce projet nous a fait découvrir un secteur que nous ne connaissons pas vraiment et qui nous a intéressés de plus en plus au fur et à mesure que nous approfondissions nos recherches. Le seul point « négatif », est sûrement le manque de temps pour pouvoir encore approfondir ce travail, car ce dernier ne s'arrête pas ici il a encore plusieurs tâches qui peut être amélioré. En effet, beaucoup de possibilités s'offrent aux passionnés de domotique, tant sur le matériel disponible que sur les actions à réaliser. Cependant rien ne nous empêche de continuer sur cette voie de notre propre côté. Ce projet a été vivant, entraînant et motivant pour la suite de nos études.

Références

- <https://www.ypesy.fr/blogs/tutorials/install-esp32-on-arduino-ide-complete-guide#d4de1d274691a51d956cd70e1a5a>
- <https://www.synox.io/actualites-sectorielle/4-choses-a-savoir-sur-linternet-des-objets/>
- <https://www.ip-systemes.com/quest-ce-que-lora-lorawan.html>
- <https://www.youtube.com/watch?v=OogIdLc9uYc>
- <https://www.youtube.com/watch?v=bB1RChYdR5A>
- <https://circuitdigest.com/microcontroller-projects/automatic-irrigation-system-using-arduino-uno>
- <https://tutoduino.fr/debuter/capteur-temperature/#:~:text=Le%20montage%20est%20tr%C3%A8s%20simple,l'Arduino%20Uno%20par%20exemple.>
- <https://www.youtube.com/watch?v=g9u6KleX7iU>
- <https://www.youtube.com/watch?v=eBePt0tAtSw>
- <https://www.makeuseof.com/install-mqtt-server-node-red-raspberry-pi-home-automation/>
- <https://medium.com/coreflux-blog/mqtt-firebase-a-coreflux-flux-asset-3d7746a7d5ac>
- <https://www.youtube.com/watch?v=Jwwx00hhztY>
- <https://microcontrollerslab.com/iot-fire-detection-esp32-flame-sensor-email-alert/>
- <https://www.robotique.tech/robotics/intelligent-flame-detection-system-with-esp32/>
- <https://tutoduino.fr/tutoriels/controler-ventilateur-brushless-arduino/#:~:text=Contr%C3%B4le%20par%20relais&text=Il%20est%20utilis%C3%A9%20pour%20la,appareils%20n%C3%A9cessitant%20une%20puissance%20importante.&text=Dans%20notre%20cas%20le%20module,num%C3%A9rique%20de%20l'Arduino.>
- <https://youtu.be/LmtmnKDrmFY>
- <https://www.allaboutcircuits.com/projects/control-a-motor-with-an-arduino/>
- <https://raspberry-pi.fr/connecter-wifi-raspberry-pi-3/#:~:text=Mettre%20%C3%A0%20jour%20Raspbian%20pour,WiFi%20ne%20soient%20pas%20install%C3%A9s.&text=Lorsque%20l'op%C3%A9ration%20sera%20termin%C3%A9e,de%20red%C3%A9marrer%20votre%20Raspberry%20Pi.>
- <https://youtu.be/mzX5oqd3pKA>