

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université 20 Août 1955- Skikda

Faculté des Sciences

Département d'Informatique



Mémoire de fin d'étude

Pour l'obtention du diplôme de MASTER en informatique

Option : Réseaux et Systèmes Distribuées (RSD)

Thème :

**Conception et réalisation d'un système domotique
intelligent basé sur l'IoT**

Réalisé par :

BOUGUERRA Aya Imen

BOUBETTA Ilham

Encadré par :

Dr.SEDDARI Nouredine

Année universitaire 2024 - 2025

Remerciements

Nous remercions **ALLAH** le tout puissant d'avoir nous donner le courage, la force et la patience de mener à terme le présent travail.

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui nous voulons témoigner toutes nos reconnaissances.

Nous offrons nos sincères et chaleureux remerciements à notre encadreur de recherche, **Dr. Noureddine Seddari**. Nous la remercions de nous avoir encadré, orienté, aidé et conseillé, son ouverture d'esprit, et ses qualités scientifiques exceptionnelles.

Nous remercions également nos parents pour leur soutien et leurs encouragements.

Nous remercions aussi les membres du jury d'avoir accepté de présider le jury de notre soutenance, et pour avoir accepté de juger ce modeste travail.

Nous désirons aussi remercier les professeurs de notre faculté, qui nous avons fourni les outils nécessaires à la réussite de nos études universitaires.

Dédicaces

À mes très chers parents, vous êtes ma source de force et d'inspiration.
Votre amour inconditionnel, vos sacrifices et vos encouragements m'ont permis
d'arriver jusqu'ici.

Ce mémoire est le fruit de votre soutien, et je vous en serai éternellement
reconnaissante. Je vous aime plus que tout.

À mes frères, **Islem**, **Tarek** et **Wail**, vous êtes bien plus que des frères, vous êtes mes
piliers, mes complices et ma motivation au quotidien.

Merci pour votre présence et votre affection.

À mon binôme **Ilham**, plus qu'une coéquipière, tu es comme une sœur pour moi.
Ce parcours n'aurait pas été le même sans toi. Merci pour ton soutien, ta bienveillance et
ces précieux moments de travail et de complicité.

À mon chère amie Imene et à toutes les personnes qui me sont chères

Arya

Dédicaces

À mes chers parents, aucun mot ne saurait exprimer toute ma reconnaissance envers vous. Votre amour, vos prières et vos sacrifices m'ont guidé et soutenu à chaque étape de ma vie. Ce travail est le reflet de tout ce que vous m'avez inculqué. Que Dieu vous protège et vous comble de bienfaits.

À mes sœurs **Maroua, Chaïma, Amina, Rahma, Inas et Khadidja**,
Vous êtes bien plus que des sœurs, vous êtes mes amies, mon réconfort et ma source d'inspiration. Merci pour votre amour inconditionnel et votre présence précieuse dans ma vie.

À mes frères **Ayoub et Walid**,
Votre soutien et votre affection ont toujours été une force pour moi.
Merci d'être là, de croire en moi et de m'encourager à aller toujours plus loin.

À mon binôme **Aya**,
Tu es bien plus qu'une simple coéquipière, tu es une amie chère à mon cœur. Ce parcours a été enrichissant grâce à toi, et je te remercie pour ta bienveillance, ton soutien et tous ces souvenirs partagés.

À mon professeur **Hazmoun Fatima Zohraa**,
Vous avez joué un rôle inestimable dans mon apprentissage et dans mon parcours spirituel. Grâce à votre patience et votre engagement, j'ai eu l'honneur de mémoriser le Saint Coran. Que Dieu vous récompense pour tout le bien que vous transmettez.

À mes précieuses amies,
Votre présence à mes côtés a rendu ce chemin plus beau et plus léger. Merci pour votre amitié sincère, votre encouragement et les moments de joie partagés.

Ilham

Résumé :

Ce mémoire présente le développement d'un système domotique basé sur l'IoT, destiné à automatiser certaines fonctions de la maison. Il permet un contrôle à distance, depuis une application Android, développée avec MIT App Inventor, de plusieurs éléments : gestion d'éclairage, contrôle de température et l'humidité à l'intérieur du logement, contrôle des fenêtres, contrôle de niveau d'eau dans le réservoir, sécurisation de la porte d'entrée avec un code PIN, détection de gaz

Le but est d'offrir un confort accru pour les occupants, d'assurer une meilleure sécurité ainsi qu'une gestion optimum de l'énergie, à travers une solution simple et accessible. C'est pourquoi nous avons conçu une application qui transmet les commandes données par l'utilisateur à la carte ESP 32 via un smartphone.

Mots Clés : *Système domotique, IOT, application Android, carte ESP32.*

Abstract:

This thesis presents the development of a home automation system based on IoT, designed to automate certain household functions. It enables remote control, via an Android application developed with MIT App Inventor, of various elements: lighting management, temperature and humidity control inside the home, water level control in the tank, curtain control, front door security with a PIN code, and gas detection.

The aim is to provide enhanced comfort for the occupants, ensure better security, and achieve optimal energy management through a simple and accessible solution. That is why we designed an application that transmits user commands to the ESP32 board via a smartphone.

Keywords: home automation, IOT, Android application, ESP32 board.

ملخص:

تتناول هذه المذكرة تطوير نظام منزلي ذكي قائم على انترنت الاشياء (IOT)، تهدف الى اتمتة بعض وظائف المنزل. يتيح هذا النظام التحكم عن بعد، عبر تطبيق اندرويد تم تطويره باستخدام MIT App Inventor، في عدة عناصر مثل : إدارة الإضاءة ، التحكم في الحرارة و الرطوبة داخل المنزل ، التحكم في الستائر ، مراقبة منسوب المياه في الخزان ، تأمين باب الدخول برمز PIN ، وكشف تسربات الغاز .

الهدف من هذا المشروع هو توفير راحة أكبر للسكان، وضمان مستوى أمان أعلى، بالإضافة إلى تحقيق إدارة مثلى للطاقة، من خلال حل بسيط وسهل الاستخدام. ولهذا قمنا بتصميم تطبيق يرسل الأوامر التي يدخلها المستخدم إلى لوحة ESP32، عبر الهاتف الذكي.

الكلمات المفتاحية: نظام أتمتة المنزل، IOT، لوحة ESP32، تطبيق أندرويد.

Table des matières

Remerciements

Dédicaces

Résumé

Abstract

INTRODUCTION GENERALE :	1
-------------------------------	---

Chapitre 1 : Généralités sur la domotique

1.1. Introduction :	4
1.2. Définition :	4
1.3. Historique et évolution de la domotique :	5
1.3.1. Historique :	5
1.3.2. Évolution :	5
1.4. Fonctionnement de la domotique :	7
1.5. Domaines d'application :	8
1.5.1 Les lumières intérieures et extérieur :	8
1.5.2 Le chauffage et la climatisation :	8
1.5.3 Le système de sécurité :	9
1.5.4 Appareils intelligent :	9
1.5.5 Le système d'arrosage :	10
1.6. Avantages et inconvénients de la domotique :	11
1.7. Conclusion :	12

Chapitre 2 : L'internet des objets

2.1. Introduction :	13
2.2. Définition :	14
2.3. Historique :	14
2.4. Principe clés de l'IOT :	16
2.5. Fonctionnement de l'IOT :	19
2.6. Domaines d'applications de l'IoT :	20
2.7. Couches de l'architecture d'un système IOT :	23
2.7.1. La couche de détection (ou de la perception) :	23
2.7.2. La couche réseau (ou de la connectivité) :	23
2.7.3 Couche traitement des données :	23
2.7.4 Couche application (ou interface utilisateur) :	24

2.8.	Technologie de communication IOT :.....	24
2.9.	Protocole de IOT :.....	26
2.10.	Avantages et inconvénients de l'IOT aux maisons intelligentes :.....	30
2.11.	Conclusion :.....	32

Chapitre 3 : Android et les environnements de développements mobiles

3.1.	Introduction :.....	33
3.2.	Définition :.....	33
3.3.	Historique :.....	34
3.4.	Les différents systèmes d'exploitation mobiles :.....	36
3.4.1	Android :.....	36
3.4.2	IOS :.....	36
3.4.4	Autres systèmes d'exploitation :.....	37
3.5.	Caractéristiques d'un système Android :.....	38
3.6.	Architecture du système Android :.....	39
3.7.	Les avantages du Système Android :.....	40
3.8.	Environnements de développement :.....	41
3.8.1.	L'outil App Inventor :.....	41
3.8.1.1.	Pourquoi choisir App Inventor :.....	42
3.8.1.2.	Historique du logiciel App Inventor :.....	43
3.8.1.3.	Structure d'application Inventor :.....	43
3.8.1.4.	Avantages et inconvénients d'App Inventor :.....	46
3.8.2.	La plateforme Thinger.io :.....	47
3.8.2.1.	Fonctionnalités principales de Thinger.io :.....	48
3.8.2.2.	Connexion de Thinger.io avec les appareils :.....	49
3.9.	Conclusion :.....	50

Chapitre 4 : Conception et Implémentation

4.1.	Introduction :.....	51
4.2.	Description du projet :.....	51
4.3.	Diagramme prévisionnel d'avancement de travail :.....	52
4.4.	Les fonctionnalités du système :.....	54
4.5.	Simulation sous Fritzing :.....	64
4.6.	Implémentation :.....	66
4.6.1.	Environnement matériel (Hardware) :.....	66
4.6.1.1.	Module ESP 32 :.....	66

4.6.1.2.	Capteur de gaz MQ2 :.....	69
4.6.1.3.	Détecteur de mouvement :.....	70
4.6.1.4.	Capteur de température et d'humidité :.....	70
4.6.1.5.	Capteur AJ_SR04M :.....	71
4.6.1.6.	Servomoteur SG90 :.....	72
4.6.1.7.	Buzzer :.....	73
4.6.1.8.	LCD 1602 I2C :.....	74
4.6.1.9.	Clavier matriciel :.....	75
4.6.1.10.	Ventilateur :.....	76
4.6.1.11.	Autres Composants :.....	76
4.6.2.	Environnement logiciel (Software) :.....	80
4.6.2.1.	Environnement de programmation Arduino IDE :.....	80
4.6.2.2.	L'outil App Inventor :.....	84
4.6.2.3.	Plateforme Thinger.io :.....	86
4.7.	Fonctionnement du système :.....	91
4.8.	Le prototype de la maison intelligente :.....	93
4.9.	Conclusion :.....	95
	Conclusion générale.....	93

Bibliographie

Références

Annexe

Liste des figures

Figure 1.1 : Installation domotique.....	5
Figure 1.2 : Evolution de la domotique.....	7
Figure 1.3 : Gestion d'éclairage.....	8
Figure 1.4 : Gestion de climatisation.....	9
Figure 1.5 : Gestion de chauffage.....	9
Figure 1.6 : Gestion de sécurité.....	9
Figure 1.7 : Réfrigérateur intelligent.....	10
Figure 1.8 : Four intelligent.....	10
Figure 1.9 : Arrosage domotique.....	10
Figure 1.10 : Avantages de la domotique.....	11
Figure 2.1 : Internet des objets.....	13
Figure 2.2 : Evolution de l'Internet des objets.....	15
Figure 2.3 : Nombre d'appareils connectés à l'Internet des objets (IoT) installés dans le monde entre 2015 et 2025 (en milliards).....	15
Figure 2.4 : Fonctionnement d'un système IOT.....	19
Figure 2.5 : Domotique.....	20
Figure 2.6 : Appareils portables.....	20
Figure 2.7 : IOT en industrie.....	21
Figure 2.8 : Ville intelligente.....	21
Figure 2.9 : Avantages des voitures connectés.....	22
Figure 2.10 : Smart irrigation.....	22
Figure 2.11 : Suivi médicale à distance.....	23
Figure 2.13 : Couches de l'architecture d'un système IOT.....	24
Figure 2.14 : Les différents technologies de communication IOT.....	26
Figure 2.15 : Protocole HTTP.....	27
Figure 2.16 : Protocole MQTT.....	28
Figure 2.17 : Protocole CoAP.....	29
Figure 2.18 : Protocole Rest Api.....	30
Figure 2.19 : Avantages de l'IoT au maison intelligente.....	31
Figure 3.1 : Application Android.....	34
Figure 3.2 : Pourcentage de l'utilisation des version Android en 2025.....	36
Figure 3.3 : Architecture du système Android.....	39
Figure 3.4 : Logo du MIT App Inventor.....	42
Figure 3.5 : L'interface graphique d'App Inventor.....	44
Figure 3.6 : La fenêtre Scratch d'App Inventor.....	45

Figure 3.7 : Interface principale de Thinger.io.....	47
Figure 3.8 : Fonctionnalités de Thinger.io.....	48
Figure 4.1 : Schéma globale du système.....	52
Figure 4.2 : Diagramme prévisionnel d'avancement de travail.....	53
Figure 4.3 : Fonctionnement de la notification de Gaz.	54
Figure 4.4 : Fonctionnement de capteur DHT11.....	55
Figure 4.5 : Organigramme de fonctionnement de contrôle de température et d'humidité. ..	55
Figure 4.6 : Fonctionnement de la gestion de fuite de Gaz.....	56
Figure 4.7 : Diagramme de flux du système de détection de gaz avec alerte.	56
Figure 4.8 : Fonctionnement de l'allumage des lampes par application	58
Figure 4.9 : Fonctionnement de l'ouverture de la porte d'entrée.....	58
Figure 4.10 : Organigramme de fonctionnement de la porte d'entrée.....	59
Figure 4.11 : Fonctionnement de gestion des fenêtres.....	60
Figure 4.12 : Fonctionnement de contrôle de niveau d'eau.....	60
Figure 4.13 : Organigramme de fonctionnement de contrôle de niveau d'eau de réservoir. .	61
Figure 4.14 : Fonctionnement de PIR sensor.....	62
Figure 4.15 : Organigramme de fonctionnement de capteur de mouvement.....	63
Figure 4.16 : Interface principale de Fritzing.....	64
Figure 4.17 : Simulation des composants du premier ESP32.	65
Figure 4.18 : Simulation des composants du deuxième ESP32.....	65
Figure 4.19 : Microcontrôleur ESP32 DEV KIT V1.	66
Figure 4.20 : Détecteur de gaz MQ2.....	69
Figure 4.21 : Détecteur de mouvement.....	70
Figure 4.22 : Capteur de température et d'humidité DHT11.....	70
Figure 4.23 : Capteur HC RS04.	71
Figure 4.24 : Servomoteur SG90.....	72
Figure 4.25 : Configuration de la broche du buzzer.....	74
Figure 4.26 : Buzzer.....	74
Figure 4.27 : Schéma de câblage de la carte ESP32 avec l'afficheur LCD I2C 1602.	75
Figure 4.28 : Ecran LCD I2c.	75
Figure 4.29 : Fonctionnement du clavier matriciel avec la carte ESP32.	75
Figure 4.30 : Ventilateur 5V.....	76
Figure 4.31 : Bouton poussoir.	77
Figure 4.32 : Branches de LED.	77
Figure 4.33 : LEDS	77
Figure 4.34 : Résistances.....	78

Figure 4.35 : Jumper wires.	79
Figure 4.36 : Plaque d'essai.	79
Figure 4.37 : Interface de logiciel Arduino IDE.....	80
Figure 4.38 : Structure d'un programme Arduino.....	82
Figure 4.39 : Les étapes de téléchargement du code.....	84
Figure 4.40 : Les blocs du premier écran de notre application.....	84
Figure 4.41 : Les blocs du deuxième écran de notre application.	85
Figure 4.42 : Blocs du troisième écran de notre application.....	85
Figure 4.43 : Notre deux appareils ESP32 dans Thingier.io.	86
Figure 4.44 : Notre interface statistiques dans Thingier.io	86
Figure 4.45 : Tableau de bord du premier ESP32 "ESP32Control"	87
Figure 4.46 : Tableau de bord dans le cas où une fuite de gaz est détectée	87
Figure 4.47 : Interface de suivi des données du premier ESP32 sur Thingier.io.....	88
Figure 4.48:Statistiques de transmission de données de l'ESP32 sur 30 jours à Thingier.io.	88
Figure 4.49 : Tableau de bord du deuxième ESP32 " ESP32WaterLED"	89
Figure 4.50 : Tableau de bord dans le cas où le niveau d'eau inférieur à 25%.	89
Figure 4.51 : Interface de suivi des données du deuxième ESP32 sur Thingier.io.	90
Figure 4.52 : Statistiques de transmission de données de l'ESP32 sur Thingier.io.	90
Figure 4.53 : Interface d'authentification utilisateur	91
Figure 4.54 : L'interface principale de notre application.	91
Figure 4.55 : Alerte dans le cas où une fuite de gaz détectée et le niveau d'eau bas.....	92
Figure 4.56 : Troisième écran de notre application.....	92
Figure 4.57 : Design de notre prototype.....	93
Figure 4.58 : Vue réelle de notre maquette.....	93
Figure 4.59 : Structure externe de la maison connectée (vue de côté)	94
Figure 4.60 : Disposition des composants matériels dans la maquette.....	94
Figure 4.61 : Câblage des composants matériels situés sous la maquette.....	95

Liste des tableaux

Tableau 1 : L'évolution des version Android à travers le temps.	35
Tableau 2 : Caractéristiques du module ESP32.	67

Liste des abréviations

IOT	Internet of things
IDO	Internet des objets
OS	Operating System
IOS	Iphone Operating System
MIT	Massachusetts Institute of Technology

INTRODUCTION GENERALE :

De nos jours, la domotique représente un grand pas vers le progrès des habitations. Elle a été d'un large regroupement de technologies permettant d'automatiser, de programmer et de contrôler à distance les équipements de la maison afin d'en améliorer le confort, la sécurité, la gestion énergétique. Avec la montée en puissance de l'Internet des objets.

L'Internet des Objets (IDO), ou Internet of things (IoT), se définit comme un réseau mondial de services interconnectés, et d'objets intelligents de toutes natures, ces dispositifs ont beaucoup évolué en permettant aux objets dits connectés de communiquer entre eux et avec l'utilisateur au moyen de réseaux intelligents. Ce couplage entre domotique et Internet des objets permet de créer des habitats plus intelligents, plus interactifs, plus proches des attentes quotidiennes, tout en offrant un pilotage facilité via des interface web ou mobile.

Problématique :

Dans un contexte de transformation progressive des modes de vie et des attentes en matière de confort, de sécurité et de qualité de l'environnement intérieur, l'habitat moderne se confronte à de multiples enjeux. Les occupants de l'habitat (personnes âgées, personnes à mobilité réduite ou usagers souhaitant améliorer leurs conditions ou qualité de vie) s'imposent encore aujourd'hui des contraintes, voire parfois des souffrances, en raison de la gestion quotidienne du logement. Les gestes les plus simples (allumer les surfaces lumineuses, ouvrir des volets ou contrôler des températures) s'avèrent compliqués, lents, parfois pesants et même parfois en inconfort. Les aléas de la sécurité domestique (fuites de gaz, etc.) constituent également des menaces pour les habitants en l'absence de toute détection. Les enjeux des risques de pollution des environnements intérieurs et des ambiances dans le cadre du maintien de la santé sont également présents, notamment par températures fortes ou humidité. Qui plus est, ces différentes problématiques interrogent également la configuration/adaptation du logement en fonction des besoins variables des occupants, dans une optique de performance des matériaux et, de qualité de l'habitat, et de qualité de vie.

Objectif :

Cette étude s'intéresse à l'amélioration du confort, de la sécurité et de l'autonomie au sein de l'habitat, en ouvrant la voie à la conception d'un système IOT intelligent intégrant le contrôle de la lumière, la gestion des volets, la détection de gaz avec un système de sécurité automatique, l'ouverture de porte à l'aide d'un code PIN, et la surveillance des conditions environnementales (température et humidité), contrôle de niveau d'eau du réservoir. L'application de contrôle à distance sera réalisée à l'aide de MIT App Inventor permettant d'accéder à une interface simple et intuitive. L'application sera connectée à la plateforme Thingier.io permettant d'envoyer les données vers le cloud pour visualiser les données de manière simultanée avec les données des capteurs.

La conjonction de deux cartes ESP32 avec Thingier.IO offre pour le projet IoT une architecture distribuée, souple et évolutive. En raison de la connectivité Wifi intégrée aux ESP32, et du protocole de communication MQTT/HTTP, les données émises par l'ESP32 sont transmises en temps réel vers Thingier.IO, lequel fournit une interface de visualisation de données claire avec alertes et sécurisation. La solution est économique, modulable, appropriée aux systèmes embarqués connectés, elle permet aussi la collecte, l'analyse et la supervision à distance, et est donc particulièrement bien adaptée au projet académique.

Le mémoire sera organisé comme suit :

Chapitre 1 : est consacré à définir la domotique, son historique et évolution, ensuite nous allons voir le fonctionnement de la domotique. Puis, nous citons ses domaines d'applications tel que les lumières intérieurs et extérieurs, chauffage et climatisation, système de sécurité. Finalement, nous présentons ses avantages et inconvénients.

Chapitre 2 : est dédié pour l'Internet des objets IOT, dont nous abordons la définition de l'IOT, historique, principe clé, fonctionnement. Ensuite nous allons présenter ses domaines d'applications, l'architecture en couches d'un système IOT. Puis nous détaillerons les protocoles de communication IOT (MQTT, CoAP, etc.). En fin de compte, nous abordons ses avantages et ses inconvénients.

Chapitre 3 : est réservé à Android et l'environnement de développement. Nous allons définir un système Android ainsi que son historique, puis nous allons voir les différents systèmes d'exploitation mobile (Android, IOS, Windows Phone, ect.), caractéristiques d'un système Android. Ensuite nous allons mettre la lumière à l'architecture du système Android, et ses avantages. Puis nous allons exposer l'environnement de développement MIT App Inventor et la plateforme cloud Thingier.io

Chapitre 4 : est dédié à la conception et l'implémentation, en premier temps, nous allons présenter la conception de notre système ainsi que la simulation des deux modules ESP32.

Ensuite nous allons aborder la partie implémentation dont nous présenterons l'environnement hardware et software. Puis nous exposerons les fonctionnalités du système avec des captures d'écran, ainsi que le prototype de la maison.

Nous terminerons ce mémoire par une conclusion générale et quelques perspectives pour des travaux futures.

Chapitre 1 :

Généralités sur la domotique

1.1. Introduction :

La technologie domotique est progressivement mise en œuvre au sein de nos maisons grâce à la possibilité d'améliorer le confort, la sécurité, ainsi que la consommation d'énergie. La domotique c'est également la possibilité de contrôler à distance, de manière automatique, l'éclairage, le chauffage, l'électroménager, la sécurité, etc., par l'intermédiaire de l'ordinateur, de la tablette ou du smartphone. La température peut être ajustée et les lumières éteintes si les occupants ne sont pas d'ores et déjà dans les lieux, afin d'effectuer des économies d'énergie et d'offrir un cadre de vie agréable. De plus, la domotique permet un suivi en temps réel des usages d'énergie permettant ainsi de maîtriser les coûts énergétiques.

La première partie s'attachera donc à définir la domotique, puis à donner les principales données historiques et évolutives, à présenter le fonctionnement et les principales applications, ainsi que les avantages et inconvénients.

1.2. Définition :

La domotique englobe les diverses méthodes qui permettent de gérer, d'organiser et d'automatiser une maison, comme le suggère son nom dérivé du terme latin 'domus' (domicile) et du suffixe '-tique' (technique). Elle fusionne et exploite donc les secteurs de l'électronique, de l'informatique, des télécommunications et des systèmes automatisés.

L'automatisation domestique fonctionne dans un domaine technique et informatique extrêmement étendu. Elle autorise la programmation de presque tous les appareils et appareils électriques domestiques, allant de l'éclairage et du chauffage aux équipements audiovisuels et électroménagers, sans oublier l'ouverture des fenêtres. Elle permet aussi de superviser la maison en régulant les systèmes d'alarme, les mesures de prévention des incendies, ou encore la température dans les différentes pièces. [1]



Figure 1.1 : Installation domotique.

1.3. Historique et évolution de la domotique :

1.3.1. Historique :

Les premières innovations en matière de domotique ont vu le jour au début des années 1980. Elles étaient orientées vers l'amélioration du confort et de la sécurité, résultant de la miniaturisation des dispositifs électroniques et informatiques. Cependant, les consommateurs n'ont pas montré un grand intérêt, ce qui a conduit à un échec notable sur le marché. Certains experts estiment que cette technologie a été introduite trop tôt. Ce n'est qu'au début des années 2000 que la domotique commence à séduire un large public.

Depuis le début des années 2000 jusqu'à aujourd'hui, avec les avancées technologiques du début du 21ème siècle, la domotique devient de plus en plus accessible. Elle commence progressivement à s'intégrer dans nos foyers et nos entreprises. Cela résulte naturellement des chercheurs qui ont su innover dans ce domaine pour capter l'attention des consommateurs. [2]

1.3.2. Évolution :

➤ *Les années 80 :*

Ce n'est que le commencement de l'ère de l'ordinateur personnel qui a du mal à s'établir dans les ménages. La maison intelligente est une idée plutôt idéaliste qui n'est pas forcément

séduisante. Pourtant, le désir de rendre son domicile plus intelligent incite de nombreux ingénieurs à explorer ce domaine qui, avec les avancées technologiques, devient concret. [3]

➤ **Les années 90 :**

L'évolution de la domotique est caractérisée par la croissance des réseaux de communication, qui deviennent non seulement plus accessibles mais également plus rapides, notamment grâce à l'émergence du Wifi. L'informatique avance également à un rythme soutenu, et on observe l'apparition sur le marché des premiers systèmes d'exploitation. Internet commence à se populariser, et l'on prend conscience que cette technologie va jouer un rôle majeur dans notre existence future et transformer nos comportements. À cette époque, la domotique était réalisée via un ordinateur connecté à un logiciel comme Homeseer, qui est encore disponible aujourd'hui. [3]

➤ **Les années 2000 :**

Explosion retentissante de la bulle internet. Le nombre d'individus qui choisissent de l'adopter ne cesse d'augmenter, et nous entrons dans une période d'interopérabilité grâce à des partenariats entre fabricants permettant aux produits de diverses marques d'interagir et d'échanger des données.

En 2007, l'évolution de la domotique prend une ampleur significative avec l'arrivée de l'iPhone, marquant le début de l'ère des smartphones. On observe un intérêt massif des consommateurs pour ce type de dispositif. Ainsi, de nombreuses applications voient le jour, offrant la possibilité de contrôler les installations domestiques à distance. De magnifiques projets de domotique commencent à émerger. [3]

➤ **Les années 2010 :**

Nous entrons dans une époque où les réseaux sociaux s'étendent largement. Une part significative de la population fréquente Facebook ou Twitter régulièrement. Les utilisateurs partagent de plus en plus du contenu, et les premiers dispositifs connectés au cloud se manifestent. Les initiatives en matière de domotique se multiplient, en particulier celles orientées vers une approche communautaire. De plus, des boxes domotiques font leur apparition sur le marché, offrant tout le confort de la domotique à un prix abordable. [3]

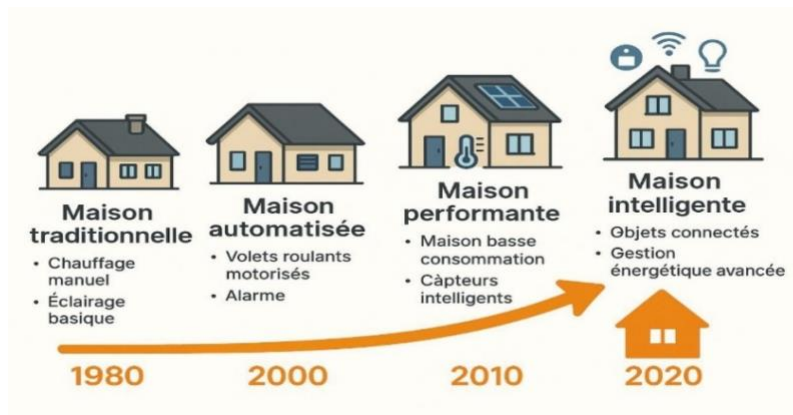


Figure 1. 2 : Evolution de la domotique.

1.4. Fonctionnement de la domotique :

La domotique représente un ensemble de dispositifs, de moyens de communication et d'interfaces électroniques qui interconnectent des appareils quotidiens via Internet. Chaque appareil est équipé de capteurs et se connecte par Wi-Fi, vous permettant de les contrôler depuis votre smartphone ou tablette, que vous soyez chez vous ou à des kilomètres. Cela vous permet d'allumer les lumières, de verrouiller la porte d'entrée ou de diminuer le chauffage, où que vous soyez.

Un système domotique est constitué de trois éléments essentiels : les capteurs, les contrôleurs et les actionneurs.

- Les capteurs ont la capacité de surveiller les variations de la lumière ambiante, de la température ou des mouvements. Les systèmes domotiques peuvent alors modifier ces paramètres (et bien d'autres) selon vos souhaits ;
- Les contrôleurs désignent les dispositifs – ordinateurs, tablettes ou smartphones – utilisés pour échanger des informations sur l'état des fonctions automatisées de votre domicile ;
- Les actionneurs peuvent inclure des interrupteurs de lumière, des moteurs ou des vannes motorisées qui régulent le mécanisme ou la fonctionnalité d'un système domotique. Ils sont conçus pour être actionnés par une commande à distance d'un contrôleur.[4]

1.5. Domaines d'application :

La domotique s'étend sur cinq principaux domaines d'application :

1.5.1 Les lumières intérieures et extérieur :

L'éclairage représente l'une des premières utilisations de la domotique, très souvent mise en œuvre en raison de sa simplicité à la pose, sans nécessiter l'intervention d'un professionnel. Il permet d'allumer et d'éteindre les lampes, éventuellement programmées pour s'adapter à nos horaires. Un capteur de mouvement peut également être utilisé pour automatiser l'éclairage. Les volets, connectés à ce système, pourront être ouverts ou fermés en fonction de l'heure de la journée. Grâce à ces dispositifs automatiques, il n'est plus nécessaire de s'en soucier lorsque l'on quitte son lieu de vie. [5]



Figure 1. 3 : Gestion d'éclairage.

1.5.2 Le chauffage et la climatisation :

Le chauffage et la climatisation constituent des éléments essentiels de la domotique, offrant la possibilité de diminuer la consommation d'énergie. De nombreuses personnes surchauffent leur habitation, mais en utilisant un thermostat approprié et une programmation adéquate, on peut activer le chauffage ou la climatisation uniquement lorsque les occupants sont présents. Les applications peuvent également les activer avant l'arrivée à la maison. Étant donné que le chauffage compte pour 50 % de la consommation énergétique, l'automatisation peut engendrer des économies significatives. [5]



Figure 1.5 : Gestion de chauffage.



Figure 1.4 : Gestion de climatisation.

1.5.3 Le système de sécurité :

La domotique renforce notre sécurité en permettant un meilleur contrôle grâce à des applications. Nous avons la possibilité de gérer les verrouillages et déverrouillages, d'utiliser des capteurs pour les portes et fenêtres, ainsi que des détecteurs de mouvement. Grâce aux caméras de surveillance, nous pouvons observer en temps réel les événements et sauvegarder les images sur une carte SD. En cas de problème, il est possible de contacter la police à distance. Ce système nous offre la tranquillité d'esprit lors de nos sorties de la maison.

[5]



Figure 1.6 : Gestion de sécurité.

1.5.4 Appareils intelligent :

De multiples appareils domestiques incluent aujourd'hui des technologies avancées. Par exemple, un réfrigérateur intelligent peut établir une liste de courses en fonction de vos achats antérieurs, alerter sur les denrées alimentaires près de l'expiration et recommander des recettes en fonction de son contenu. Dans la cuisine, un four intelligent permet de le

préchauffer à distance grâce à un smartphone, ce qui procure un gain de temps et de commodité. [5]



Figure 1.7 : Réfrigérateur intelligent.



Figure 1.8 : Four intelligent.

1.5.5 Le système d'arrosage :

L'irrigation automatisée contribue à diminuer la consommation d'eau, qui constitue environ 30 % des usages domestiques. Grâce à la technologie domestique, elle peut être programmée et ajustée en fonction des conditions climatiques. Par exemple, elle se déclenche automatiquement lors des périodes de chaleur et se désactive en cas d'humidité. De plus, elle peut être connectée aux données météorologiques pour un arrosage efficace et économe. [5]



Figure 1.9 : Arrosage domotique.

1.6. Avantages et inconvénients de la domotique :

➤ Avantages :

- **Le confort** : Le contrôle, à distance ou automatique, d'appareils domestiques grâce à la domotique permet de se détendre davantage, et d'avoir du temps pour des travaux à réaliser. Ainsi, il est agréable, assis sur un sofa, d'agir depuis son fauteuil pour la lumière ou la fermeture des volets.
- **L'économie d'énergie** : En régulant la chaleur, l'éclairage, l'usage des appareils électroménagers, il est possible de diminuer sa consommation d'électricité, et d'avoir toutes les fins de mois des factures d'électricité bien allégées.
- **La sécurité** : En assurant la surveillance des accès à son domicile, en détectant les fuites de gaz ou d'eau, en alertant le résident sur les dangers, la domotique alors pourra assurer la sécurité des biens, des personnes.
- **L'adaptabilité** : Conçus pour être modulables et évolutifs, les systèmes de domotique permettront d'ajouter, au fil du temps, des appareils, des fonctionnalités, selon ses besoins, ses souhaits, ses moyens, mais aussi selon l'évolution technique. [6]



Figure 1.10 : Avantages de la domotique.

➤ Les inconvénients :

- **Le coût** : L'un des plus grands inconvénients de la domotique est le frais d'investissement initial. En effet, les appareils et systèmes domotiques coûtent généralement plus cher que les autres types de matériels, ce qui peut nécessiter une dépense importante selon les attentes du client.
- **La cybersécurité** : Tout comme tout appareil connecté, les systèmes domotiques peuvent être déployés, une cyberattaque peut avoir lieu. C'est un projet à prévoir pour lors de l'installation de notre système domotique.
- **La fiabilité** : Les systèmes domotiques peuvent avoir parfois des pannes, des anomalies qu'elles peuvent engendrer une interruption de service dans le fonctionnement de la maison. Il est donc nécessaire de faire le choix d'équipements et systèmes de qualité et, éventuellement, d'alternatives en cas de problème.
- **L'installation** : L'installation et la configuration des systèmes domotiques peuvent être complexes et nécessiter des compétences techniques. Il est parfois nécessaire de recourir à un professionnel pour réaliser cela. [6]

1.7. Conclusion :

En synthèse, nous avons caractérisé la domotique en retraçant son historique et son développement. Nous avons par la suite examiné son mode de fonctionnement et ses domaines d'application essentiels. En outre, nous avons souligné ses avantages ainsi que ses inconvénients. Pour conclure, cette analyse nous a aidés à mieux appréhender l'impact de la domotique sur le confort, la sécurité et l'efficacité énergétique des habitas.

CHAPITRE 2 :
L'Internet des Objets (IoT)

2.1. Introduction :

À l'heure actuelle, l'Internet des objets (IoT) occupe une place de plus en plus croissante dans notre quotidien, en s'harmonisant avec des technologies et des réseaux en perpétuelle évolution. En permettant la connexion et l'interaction entre des objets de différents types, l'IoT crée des opportunités dans de nombreux secteurs (domotique, santé, industrie et villes intelligentes). Ainsi, dans le cadre de l'IoT, les objets deviennent plus autonomes en cernant, traitant et en échangeant des données rapidement afin d'optimiser l'efficacité, le confort et la sécurité.

Dans ce chapitre, nous comprendrons l'IoT que nous placerons dans son contexte historique suivi d'une définition des principes et fondamentaux théoriques régissant cette technologie. Au chapitre suivant les particularismes et mouvements du développement dans ce secteur techno professionnel. Dans le chapitre suivant l'architecture et les composants d'un système IoT. Dans le chapitre suivant les principaux protocoles de communication pouvant être utilisés en IoT. Dans la section suivante les plateformes et outils de développement IoT, puis se sera le tour des applications IoT dans les maisons intelligentes. Pour conclure ce chapitre nous analyserons avantages et inconvénients dans la domotique IoT avant d'un dernier point récapitulatif des points majeurs abordés.



Figure 11 : Internet des objets.

2.2. Définition :

De manière générale, l'Internet des Objets englobe tous les objets capables d'être reliés à un réseau Internet. Cependant, actuellement, l'IoT se concentre surtout sur les objets connectés munis de capteurs, de logiciels et d'autres technologies qui leur permettent d'échanger des données entre eux, à des fins d'information ou d'automatisation. Jusqu'à présent, la connectivité était principalement assurée par le Wi-Fi, tandis qu'aujourd'hui, la 5G et d'autres types de plateformes en réseau permettent de traiter d'énormes volumes de données presque partout, de manière rapide et fiable. [7]

2.3. Historique :

➤ La création du concept d'internet des objets :

En 1999, Kevin Ashton, ingénieur britannique exceptionnel de la recherche en réseaux RFID, crée, à la suite de ses travaux, une nouvelle expression désormais incontournable : Internet of Things ou IoT, devenue "internet des objets" ou IDO. Rapidement, le concept d'Internet des objets a pris son essor en raison des révolutions technologiques majeures en information et communication que nous venons de citer :

La dynamique exponentielle de l'augmentation du parc des objets connectés, le développement rapide des plateformes IoT, l'élargissement des champs d'application de l'intelligence artificielle, l'apparition du Big Data ou explosion des volumes de données numériques. Actuellement, ce terme désigne un vaste réseau d'objets interconnectés et communicants via internet, ayant la capacité de collecter et d'analyser les données en toute autonomie afin d'accomplir des tâches. Tout objet physique capable d'émettre des informations grâce à des capteurs pouvant communiquer avec un réseau internet, devient un objet connecté. L'IoT est donc l'ensemble des technologies pouvant fournir des données pour alimenter le monde virtuel du Big Data, grâce à l'information fournie par des milliards d'objets physiques connectés aux réseaux. [7]

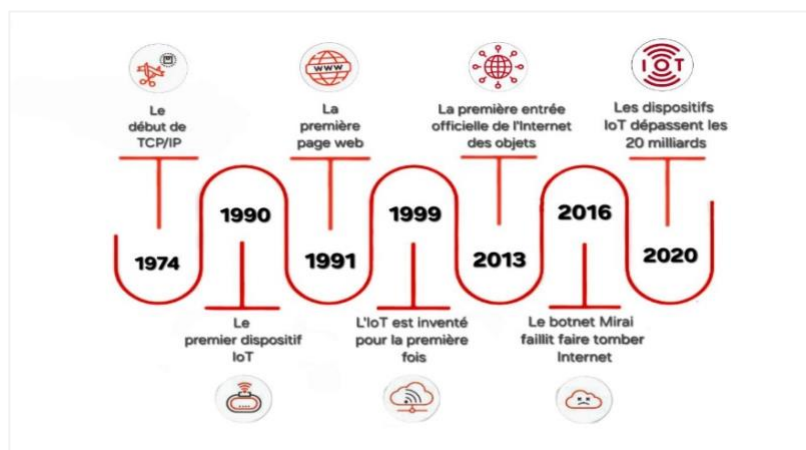


Figure 12 : Evolution de l'Internet des objets.

➤ L'IOT en chiffre :

Selon le Cisco Internet Business Solutions Group (IBSG), l'émergence de l'Internet des objets a été constatée lorsque le nombre d'objets reliés à Internet a dépassé celui des individus sur la planète, c'est-à-dire entre les années 2008 et 2009. En 2010, on dénombrait déjà 12,5 milliards d'appareils connectés pour 6,8 milliards de personnes.

En 2020, le total des objets connectés dans le monde dépassait les 21,7 milliards, comprenant 11,7 milliards d'objets connectés, à cela s'ajoutent 10 milliards de smartphones, tablettes et ordinateurs, qui, bien qu'étant connectés, ne sont pas considérés comme faisant partie des objets IoT.

De plus, il est envisageable que les investissements mondiaux dans l'Internet des objets atteignent 1 100 milliards de dollars d'ici 2023. [8]

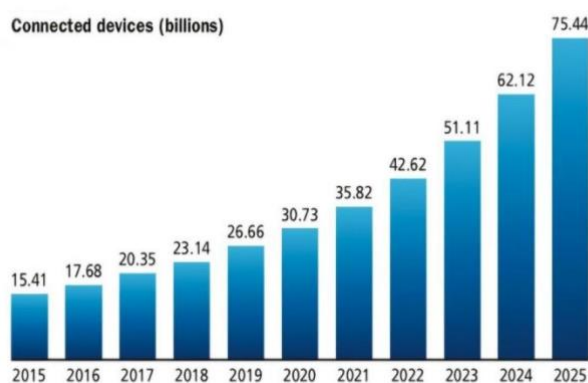


Figure 13 : Nombre d'appareils connectés à l'Internet des objets (IoT) installés dans le monde entre 2015 et 2025 (en milliards).

2.4. Principe clés de l'IOT :

- **Le développement du Cloud :**



Si les technologies de l'IoT existent depuis un certain temps, en particulier dans l'industrie, il leur manquait une infrastructure disponible en tout lieu et à tout moment pour pouvoir enfin se déployer.

C'est le Cloud qui a apporté la réponse adéquate, couplé avec les différents réseaux qui permettent d'assurer les interconnexions. Aujourd'hui, l'IoT ne peut être envisagée sans le Cloud, et les fournisseurs de ce dernier, les 'providers' l'ont bien compris. Ils sont plusieurs milliers à répondre à l'appel et proposent des solutions logicielles pour accueillir les données fournies ou à fournir aux objets connectés.

Une profusion qui participe à créer un secteur concurrentiel où les entreprises sauront profiter de coûts de plus en plus bas et d'un « Time to Market » de plus en plus accéléré. [9]

- **Le prix en baisse des objets connectés :**



Il est difficile de donner un chiffre exact sur le nombre d'objets connectés qui nous entourent. En effet, les données des analystes présentent deux caractéristiques communes : elles se chiffrent en dizaines de milliards et

elles varient énormément ! Ce constat dans le secteur a une conséquence intéressante : que ce soient des capteurs pour l'industrie ou des articles destinés au grand public, les objets connectés deviennent de moins en moins onéreux, surtout ceux qui sont des modèles standards issus de catalogues. Dans le domaine industriel, le seuil psychologique de 100 euros pour un capteur a été franchi. Actuellement, ce sont les coûts liés à l'installation et à la maintenance des objets connectés qui représentent la plus grande dépense. [9]

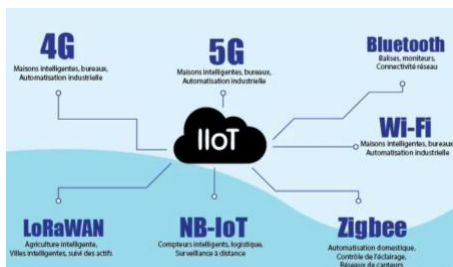
- **Des batteries plus performantes et moins chers :**



Voici un élément crucial que nous saisissons tous : la durabilité des batteries. En vérité, à l'exception de quelques dispositifs comme certaines étiquettes RFID qui tirent leur énergie des ondes des appareils qui les lisent, un objet connecté nécessite une source

d'alimentation pour opérer et pour échanger des informations. Et alors que l'innovation continue de stimuler ce domaine, les batteries deviennent de plus en plus efficaces en termes de puissance et de durée de vie de la charge. Garantir une autonomie toujours plus importante représente un défi majeur pour abaisser les frais de maintenance. [9]

- **L'omniprésence des réseaux sans fil :**



Un autre défi crucial pour l'évolution de l'IoT concerne les communications. Le WiFi, le Bluetooth, le Lora, le SigFox, le 3G, le 4G, et bientôt le 5G, pour n'en nommer que quelques-uns, les réseaux sans fil et leurs protocoles sont partout. Il semble presque impensable

aujourd'hui d'imaginer un capteur incapable de se connecter, à condition bien sûr d'utiliser la bonne technologie et le protocole adéquat. Le choix du réseau est aussi déterminé par les performances souhaitées, le type de batterie et le volume des données à transmettre. Chaque besoin a sa propre solution ... [9]

- **Le smartphone, une passerelle idéale pour l'IoT :**



Aujourd'hui, la connexion à internet et les discussions en ligne à travers les smartphones surpassent les modes d'utilisation traditionnels de l'internet. En plus de faire partie intégrante de notre quotidien, le smartphone est devenu l'appareil le plus efficace que nous possédons.

Cet appareil est désormais au cœur de notre quotidien, un dispositif connecté qui ne nous abandonne jamais. Il n'est donc pas surprenant que le smartphone soit devenu une passerelle pour l'Internet des objets, un point de connexion pour les dispositifs connectés, facilitant la

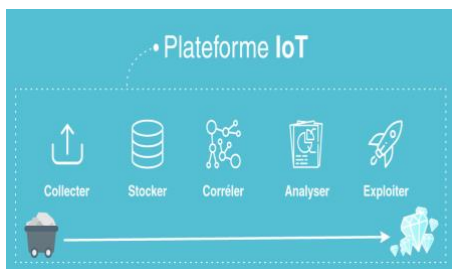
transmission d'informations vers une plateforme IoT, ou même leur traitement localement.[9]

- **Le Big Data et l'Intelligence Artificielle en renfort :**



Les dispositifs intelligents sont des capteurs qui transmettent ou reçoivent des informations pour évaluer et contrôler des appareils. Ces informations sont traitées sous forme de données gérées dans des ensembles de données structurées ou non structurées, mises en œuvre en quelques minutes dans le Cloud. Lorsque les quantités de données augmentent considérablement, ce qui se produit rapidement avec la prolifération des dispositifs intelligents, les bases de données traditionnelles sont remplacées par des Data Warehouses ou des solutions de Big Data, également sur le cloud. De plus, l'IA (Intelligence Artificielle) et ses algorithmes d'analyse de données deviennent de plus en plus accessibles.[9]

- **Des plateformes IoT :**



Les plateformes IoT intègrent toutes les fonctionnalités techniques mentionnées dans les neuf premiers éléments. Elles apportent une valeur ajoutée croissante car elles soutiennent de plus en plus le modèle commercial des fournisseurs d'appareils connectés, qui doivent faire face à la baisse des marges sur les appareils en eux-mêmes. Ils ont deux options : créer leur propre plateforme ou opérer leurs services dans le Cloud. La désintermédiation technologique, la confidentialité des entreprises et la sécurité des données sont des critères décisifs qui exigent une stratégie claire et bien pensée. [9]

2.5. Fonctionnement de l'IOT :

L'Internet des Objets peut être défini comme un ensemble de technologies interconnectées sur trois niveaux :

- ✓ **Objets** : ce sont les Choses. Ce sont des objets physiques, munis de capteurs rattachés au réseau Internet, qui fournissent des données de télémétrie (on peut penser à la voiture qui passe sur une autoroute, grâce à des capteurs de mouvement ou de pression) ;
 - ✓ **Informations** : ce sont les résultats, issus du traitement et de l'analyse des données brutes qu'envoient les Choses. Par exemple : en tenant compte de l'effectif de voitures qui passent par minute, on peut dire où en est le trafic d'une autoroute ;
 - ✓ **Actions** : il s'agit de réponses automatiques ou manuelles aux informations fournies par l'Insight. Par exemple : si l'Insight déclare que l'on est dans les bouchons, on peut envoyer des alertes à plusieurs systèmes comme des panneaux d'affichage autoroutiers ou bien diverses applications utilisées par les chauffeurs.
- [10]



Figure 14 : Fonctionnement d'un système IOT.

2.6. Domaines d'applications de l'IoT :

L'Internet des Objets (IoT) est employé avec efficacité dans de nombreux secteurs :

- **Domotique** : Le contrôle vocal permet de diriger les appareils électroménagers via des différentes voix d'assistants (Alexa, Google Home, etc.) et donne un réel confort et interconnexion des lieux de vie.



Figure 15 : Domotique.

- **Appareils portables** : Montres, bracelets connectés, lunettes connectés ... les objets connectés sont des dispositifs technologiques avancés de surveillance de la santé, de la pratique sportive et même de la réalité augmentée.



Figure 16 : Appareils portables.

- **Industrie** : L'Internet des objets (IoT) joue un rôle déterminant dans la transformation de l'industrie, le système de fabrication intelligent est un système intégré et collaboratif qui réagit en temps réel aux besoins variables et aux exigences de l'usine, du système

d'approvisionnement et des besoins des clients. Cela est permis par des capteurs et actuateurs évolués et interconnectés grâce aux technologies IoT dans le processus de fabrication et à la conduite des actions de fabrication par intelligence artificielle. Par ailleurs, grâce à l'interconnexion avec des systèmes extérieurs, les machines peuvent se mettre à produire dès la réception, par l'usine, d'un bon de commande.[11]



Figure 17 : IOT en industrie.

- **Villes intelligentes** (Smart city) : Une ville intelligente est une ville qui utilise les technologies de l'information et de la communication (TIC) pour améliorer la qualité des services urbains ou le coût des services urbains. La ville intelligente repose souvent sur des outils numériques pour l'amélioration de la qualité de vie des citoyens. La technologie est au service d'un développement intelligent de la zone urbaine tant au niveau de la mobilité qu'au niveau de l'environnement, de la participation citoyenne... Il n'est donc pas surprenant que la ville numérique semble souvent l'étape préliminaire à la ville intelligente pour une gestion urbaine.[11]



Figure 18 : Ville intelligente.

- **Véhicules connectés** : véhicules autonomes ou assistés, ces voitures peuvent se faire un diagnostic et se programmer pour leur entretien (comme les Tesla).

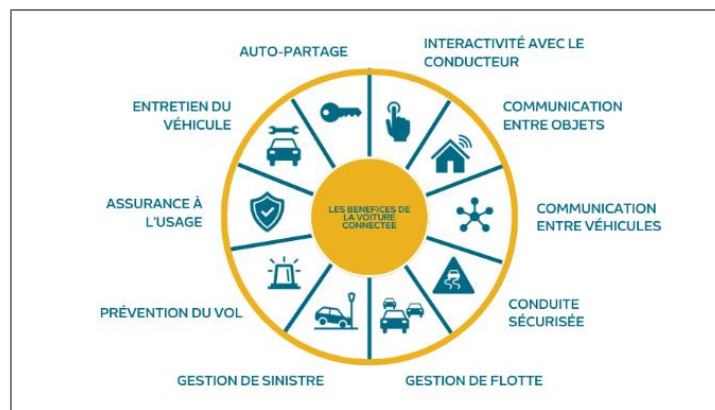


Figure 19 : Avantages des voitures connectées.

- **Agriculture** : des systèmes d'irrigation intelligents basés sur des données sur les conditions météorologiques, sur le sol et sur les cultures sont utilisés par les agriculteurs dans le cadre de la culture de certains légumes dans le but d'optimiser la plantation, l'usage des herbicides et la gestion de l'eau, avec l'avantage important d'empêcher par exemple les arrosages avant la pluie.



Figure 2.10 : Smart irrigation.

- **Santé** : Le secteur médical est transformé par l'Internet des objets, qui permet d'améliorer la surveillance des patients, d'utiliser des dispositifs intelligents et d'accroître la prévention des incidents, ce qui va de pair avec une meilleure qualité des soins dispensés et une plus grande efficacité des établissements hospitaliers.



Figure 20 : Suivi médicale à distance.

2.7. Couches de l'architecture d'un système IOT :

2.7.1. La couche de détection (ou de la perception) :

Constitue la base de tout système IoT. Elle se compose de capteurs (permettant notamment de mesurer la température, le mouvement, etc.) et d'actionneurs (permettant de réaliser des actions, par exemple, pour éclairer une pièce, etc.). Son rôle est de collecter les données du monde physique afin que le système puisse appréhender son environnement. Sans lui, le système ne pourrait tout simplement pas interagir avec son environnement. [12]

2.7.2. La couche réseau (ou de la connectivité) :

Elle est la couche du système qui assure la transmission entre capteurs, dispositifs et le reste du système. Elle repose sur différents protocoles : HTTP, MQTT ou AMQP ; des systèmes de chiffrement avec clé (publique et privée) et des passerelles, points d'accès ou réseaux privés 5G simples mais efficaces, permettant de faciliter la communication vers l'extérieur de façon sécurisée.[12]

2.7.3 Couche traitement des données :

Les données collectées sont analysées de manière objective, filtrées à travers un ensemble d'outils d'intelligence artificielle et de machine learning, dont l'objectif est de leur faire délivrer de l'information intelligible ; on peut prendre comme exemple la reconnaissance

anticipée d'une panne dans un système. Cette étape est cruciale pour convertir des données en décisions intelligentes, voire automatisées.[12]

2.7.4 Couche application (ou interface utilisateur) :

C'est ce qui est visible par l'utilisateur, ce qui lui permet d'interagir avec le système via une application mobile, un site Internet ou un tableau de bord ; on peut ici allumer les lumières de sa maison ou encore, à distance, faire fonctionner un capteur en réglant sa température. Elle rend le système accessible, facile d'utilisation et utile au quotidien. [12]

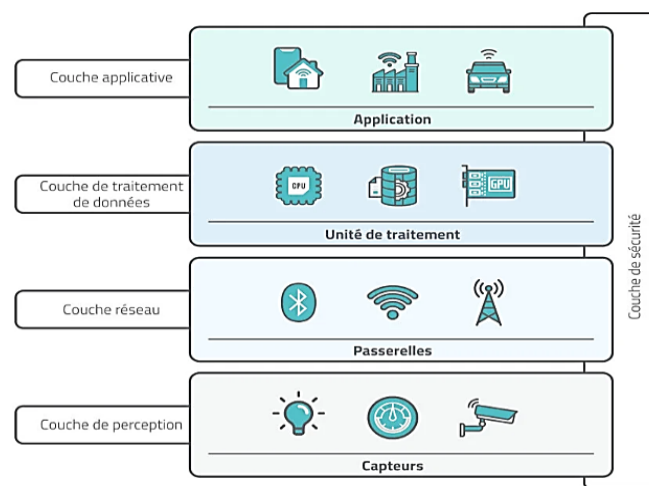


Figure 21 : Couches de l'architecture d'un système IOT.

2.8. Technologie de communication IOT :

Se profile ainsi un système informatique omniprésent dans lequel les objets numériques non seulement peuvent s'identifier collectivement, mais également penser et interagir entre eux dans une logique de collecte de données visant à automatiser les actions, ce qui est rendu possible par la combinaison de nouveaux outils technologiques que seules permettent d'opérer divers nouveaux outils technologiques désormais capables d'identifier les objets de leurs communications respectives et de les échanger.[13]

➤ **Wi-Fi :**

Le réseau IEEE 802.11 est un réseau sans fil local. En bande basse, IEEE 802.11 (b, g, n) transmet des données de 11 Mbps et jusqu'à 54 Mbps, jusqu'à 32 mètres à l'intérieur et 95 mètres à l'extérieur. La norme IEEE 802.11n est capable d'utiliser le double du spectre radio que la norme 802.11a ou 802.11g. Toutefois la norme IEEE 802.11a, lorsque à transmission permet d'atteindre jusqu'à 1 Gbps et peut aller au-delà des plages des normes b et g, où le Wi-Fi en bande basse transmet dans la bande ISM 2,4 GHz tandis que le Wi-Fi en bande haute transmet dans la bande 5 GHz. [14]

➤ **Bluetooth :**

Bluetooth est une technologie permettant de faire communiquer sans fil deux appareils à courte portée, sur la base de la proximité entre les appareils, alors qu'un mot de passe n'est pas exigé pour gérer la connexion. La norme Bluetooth fonctionne aux ondes radio UHF (ultra haute fréquence) sur des longueurs d'onde comprises entre 2.400 et 2.485 GHz et n'est donc pas forcément limitée à 50 mètres au maximum, mais est tributaire d'un rayon de 164 pieds entre deux appareils. [15]

Le Bluetooth est capable de transmettre des débits de 1 Mbps (1600 échanges/sec en full-duplex) en portée d'une dizaine de mètres environ avec un émetteur de classe II, mais alors d'une portée d'un peu moins de 100 mètres avec un émetteur de classe I.

➤ **Zigbee :**

ZigBee est développé par ZigBee Alliance, qui compte des centaines de sociétés membres (Ember, Freescale, Chipcon, Invensys, Mitsubishi, CompXs, AMI Semi-conducteurs, ENQ Semi-conducteurs), fabricants de semi-conducteurs et développeurs de logiciels de l'origine équipementiers. ZigBee et 802.15.4 ne sont pas à confondre. ZigBee est un protocole réseau basé sur des standards soutenus uniquement par l'alliance ZigBee, qui utilise les services de transports de la spécification de réseau IEEE 802.15.4.[16]

Caractéristiques de ZigBee : coût avantageux, débit : 10kbps-115.2kbps, portée radio : 10-75m, jusqu'à 65k noeuds par réseau, jusqu'à 100 réseaux co-localisés, jusqu'à 2 ans de durée de vie de batterie standards Alkaline.

➤ LORA :

Le terme LoRa désigne la technologie de communication Low Power Wide Area Network (LPWAN). Il s'agit d'un système de transmission sans fil ultra longue en se basant sur une technique à spectre étalé. Il privilégie la communication longue portée avec sa très haute sensibilité de réception, lui permettant de fonctionner dans un environnement sonore ou sous des niveaux de bruit ambiant élevés. LoRa est une technologie de transmission WAN basse consommation, essentiellement orientée vers l'Internet des objets. Lora est l'abréviation du mot anglais longrange. La longue portée est également l'un des nombreux atouts de Lora, avec une distance de transmission importante. [17]

Caractéristiques de LoRa : longue distance de transmission : sensibilité -148dBm, distance de communication jusqu'à 15 kilomètres, faible consommation d'énergie de travail, de nombreux nœuds de réseau, forte capacité anti-interférence, et à petit prix.

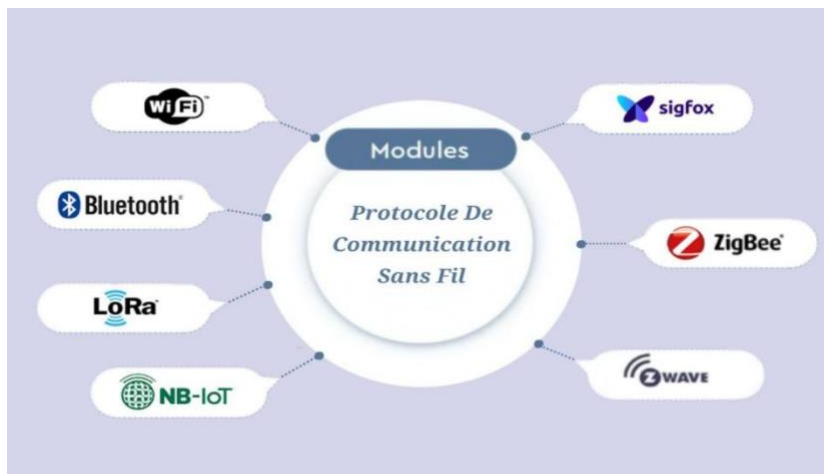


Figure 22 : Les différents technologies de communication IOT.

2.9. Protocole de IOT :

Les protocoles de l'IoT gèrent l'échange de données entre les appareils et les serveurs ou applications. Voici les protocoles les plus utilisés :

➤ HTTP :

Le protocole de transfert hypertexte (HTTP) est un protocole de communication au niveau applicatif destiné aux systèmes d'informations hypermédias collaboratifs et distribués qui est générique dans la mesure où il peut être utilisé pour beaucoup d'autres fonctions que celle qui consiste à transférer de l'hypertexte (système de gestion d'objets distribués ou serveur de noms). Il permet donc de transférer des données sur le Web en étendant ses méthodes de requêtes, ses codes d'erreur et ses en-têtes pour faire de cet outil un protocole aussi adapté à l'hypermédia qu'au Web. [18]

HTTP repose sur l'architecture client/serveur. Le client peut être un ordinateur personnel, un ordinateur portable, un ou plus généralement un périphérique mobile et le serveur HTTP est généralement situé sur un hôte Web qui exécute un logiciel de serveur Web (Apache ou IIS, par exemple). Lorsque vous accédez à un site Web, votre navigateur (client) va adresser une demande au serveur Web correspondant et celui-ci répondra en délivrant un code d'état HTTP. Si l'URI demandée est valide que la connexion est établie le serveur enverra au navigateur la page Web demandée et le fichier qui lui est associé. [19]

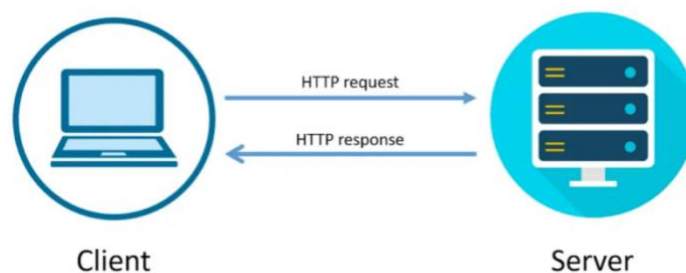


Figure 23 : Protocole http.

➤ MQTT :

MQTT est un protocole de publication-abonnement s'appuyant sur TCP développé par IBM et plus tard libéré sous licence ouverte dans le cadre de la messagerie applications. Dans un mode de publication-abonnement, des clients peuvent être à la fois (publieurs) d'informations sur un sujet particulier sur le serveur ou (abonnés) à un sujet sur lequel le serveur fera parvenir automatiquement les nouvelles informations au moment de leur

publication une fois ce dernier enregistré. MQTT combine donc le coût relativement élevé et la qualité de service apportée par TCP avec les caractéristiques un-à-un, un-à-plusieurs et plusieurs-à-un d'un format de publication-abonnement. De plus, il permet aux clients de spécifier les sujets de télémétrie qui les intéressent, n'ayant accès qu'aux informations publiées sur ces sujets. [20]

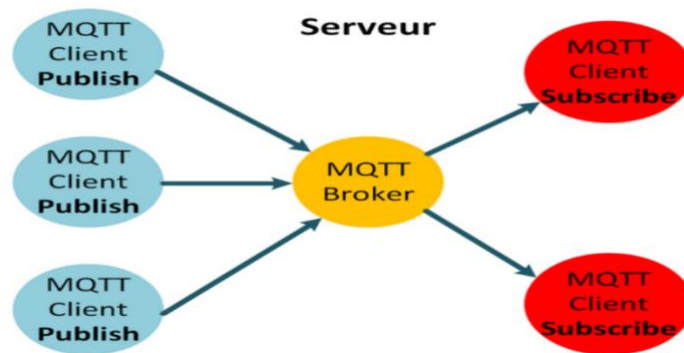


Figure 24 : Protocole MQTT.

➤ CoAP :

Le protocole CoAP, abréviation de Constrained Application Protocol, a une architecture reposant sur des requêtes et des réponses, comme le protocole HTTP, et a également une architecture REST. Il est effectivement conçu, comme son nom l'indique, pour des dispositifs et réseaux à ressource limitée, au premier chef les réseaux de capteurs sans fil adoptant une architecture Web of Things de type Web pour objets connectés.

Par le choix d'une architecture fondée sur le modèle REST, CoAP donne accès aux ressources d'objets de l'Internet par l'intermédiaire d'Uniform Resource Identifiers (URI) selon un modèle client-serveur, partageant ainsi un grand nombre de caractéristiques de HTTP, mais ne pouvant pas y être cependant assimilé comme une version réduite ou alternative de celui-ci, ne serait-ce qu'en raison des nombreuses distinctions qui existent aussi avec ce dernier. [21]

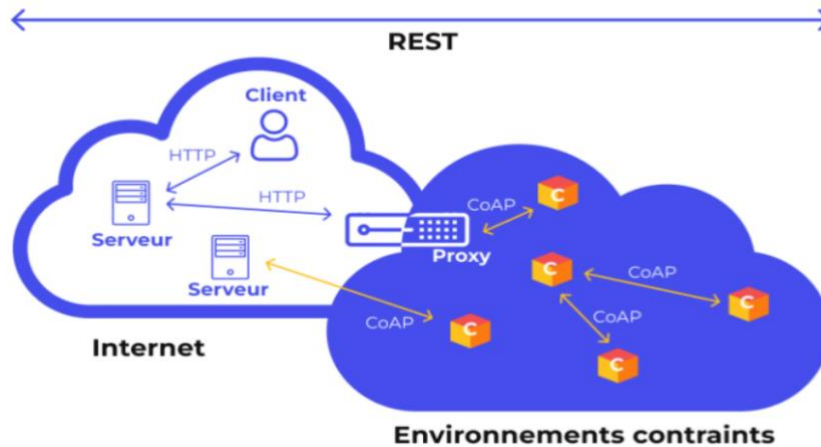


Figure 25 : Protocole CoAP.

➤ REST API :

Le terme REST API (Representational State Transfer Application Program Interface) désigne un style architectural de logiciels interopérables sur réseau (ou sur même machine), qui est d'ailleurs le style principalement utilisé par les développeurs pour lesquels les API REST sont particulièrement populaires pour créer des service web, souvent appelés services web RESTful. REST est un terme qui désigne un ensemble de méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur.

En s'appuyant sur le protocole HTTP, les REST API permettent aux logiciels présents sur un appareil de communiquer avec les logiciels d'un autre appareil (ou du même appareil) lorsque ceux-ci sont issus de systèmes d'exploitation et d'architectures différentes. Par exemple, le client des ressources acheminées dans un langage que le serveur comprend, et le serveur renvoie la ressource dans un langage que le client accepte. Le serveur renvoie la ressource en JSON (JavaScript Object Notation), XML (Extensible Markup Language) ou texte ou toute autre langue prise en charge par l'API. [22]

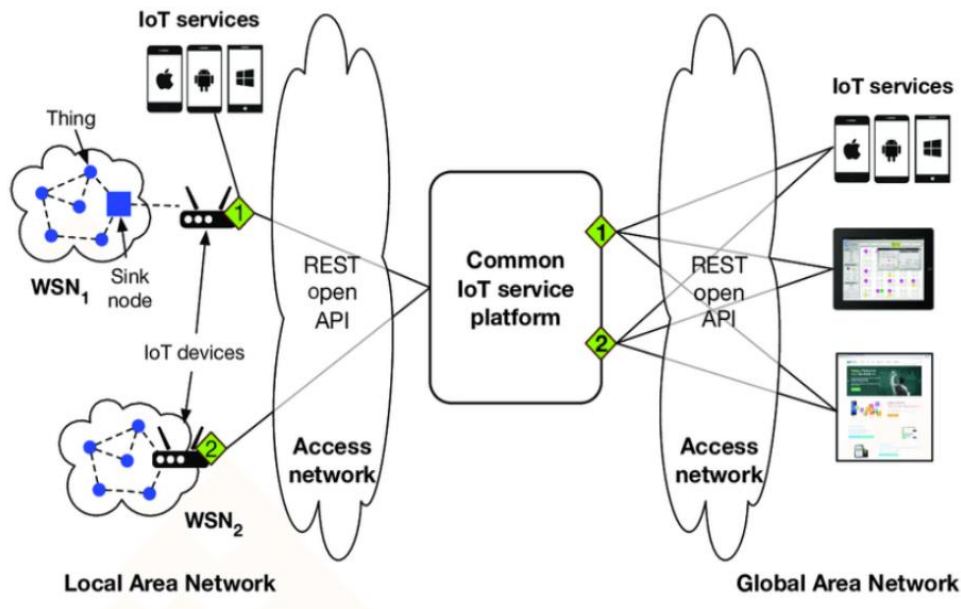


Figure 26 : Protocole Rest Api.

2.10. Avantages et inconvénients de l'IOT aux maisons intelligentes :

➤ Avantages :

- **Télécommande** : Puisque les appareils domestiques intelligents se connectent tous à votre réseau à domicile, il est possible d'accéder et de contrôler ces dispositifs à distance via les smartphones, tant que votre appareil mobile dispose d'une connexion Internet.
- **Commodité** : Les appareils intelligents pour la maison existent en différentes tailles, grandes ou petites, et apportent un peu de facilité dans notre quotidien. Ces dispositifs personnalisés sont conçus pour rendre nos maisons plus efficaces et mieux adaptées à nos besoins.
- **Surveillance et conservation de l'énergie** : Avec les maisons intelligentes, vous avez un meilleur contrôle sur votre consommation d'énergie. Que ce soit pour ajuster automatiquement la température, gérer l'éclairage ou réguler l'irrigation selon les conditions climatiques, vous pouvez rester informé sur ces données de consommation.

- **Sécurité** : L'ensemble des caméras de surveillance et divers capteurs, tels que ceux des portes ou des fenêtres, se réunissent pour offrir une protection supplémentaire à votre domicile. Vous pouvez surveiller votre maison à tout moment et en tout lieu, et recevoir des alertes lorsqu'un mouvement est détecté.

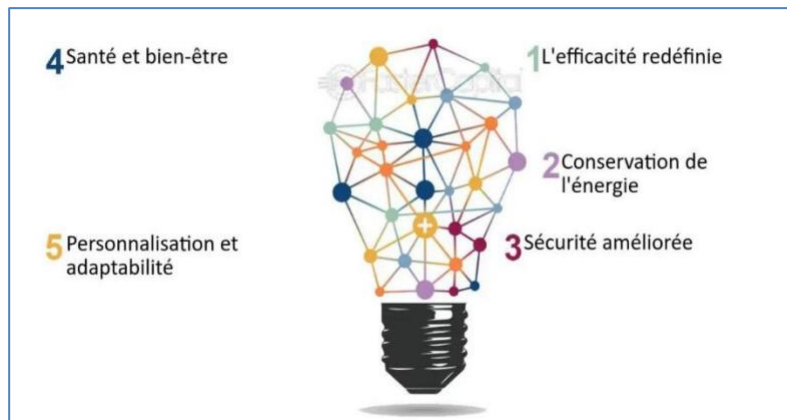


Figure 27 : Avantages de l'IoT au domicile intelligent.

➤ **Inconvénients :**

- **Coût élevé** : Les technologies intelligentes peuvent être coûteuses. Le coût d'équipement d'un système complet de maison intelligente peut être très élevé en raison de la quantité d'équipements et d'appareils nécessaires, d'autant plus que ces appareils intelligents ont toujours des prix assez élevés.
- **Installation et configuration complexes** : La pierre d'achoppement pour s'adapter à une maison intelligente est sa configuration et sa configuration complexes, ce qui n'est pas si convivial pour les novices. Si vous êtes une personne férue de technologie, l'adoption peut être assez rapide. Si vous n'êtes pas très technique, ce n'est peut-être pas une expérience agréable.
- **Risques pour la sécurité à domicile** : À condition que vos appareils soient connectés à Internet, il y a de fortes chances que vous ayez une sorte de problème de sécurité. Si les pirates trouvent un moyen d'accéder à votre système de maison intelligente, ils

pourraient potentiellement déverrouiller les portes, et éteindre les alarmes et les lumières, laisser votre maison sans défense pour entrer par effraction.

- **Filtrage des données** : La confidentialité des données des appareils domestiques intelligents est un autre élément essentiel que de nombreux consommateurs prennent au sérieux. Plus vous appliquez d'appareils domestiques intelligents, plus les fabricants de plates-formes auront de chances de collecter des données à votre sujet. En outre, les pirates auront accès à certaines données précieuses des appareils domestiques intelligents.

2.11. Conclusion :

Ce chapitre a été l'occasion de faire découvrir l'Internet des objets (IoT) via sa définition, son historique, ses principes fondamentaux, son mode de fonctionnement, ses différents domaines d'application et les différentes couches de l'architecture d'un système IoT, ainsi que les protocoles essentiellement de communication qui touchent aux systèmes IoT. Nous avons enfin pu mettre en avant les avantages et inconvénients que présente l'IoT en tant qu'application particulière qui s'institutionnalise ces dernières années : la maison intelligente.

CHAPITRE 3 :
Android et les environnements de
développement mobile

3.1. Introduction :

Actuellement plus de 3 milliards d'outils dans le monde utilise Android, une donnée qui ne laisse pas indifférent dans le monde de la technologie mobile. Android est produit par Google. Son innovation et sa capacité à mettre en place un environnement propice à l'innovation autour de l'application ont fait de lui un acteur incontournable pour le portable, pour le matériel électronique grand public dans sa globalité et pour l'ouverture d'horizons nouveaux de type « application ».

Android a considérablement modifié notre quotidien, en matière de localisation GPS comme de sécurité dans le domaine des paiements mobiles en passant par la manière d'interagir avec les outils numériques en donnant plus de liberté. Les applications ont donc considérablement augmenté, et peuvent être produites par tout un chacun dans un environnement « favorable » au développement d'applications.

Dans la première partie de ce travail, nous allons proposer une définition d'Android suivie d'un historique. Nous verrons les différents systèmes d'exploitation mobiles (on peut penser à Android, IOS, Windows Phone), leurs caractéristiques et architecture. Nous présenterons ensuite l'environnement d'exécution Android et quelques avantages de son utilisation.

Dans la deuxième partie, nous intégrerons l'étude de l'environnement de développement MIT App Inventor (historique, structure, avantages et inconvénients).

3.2. Définition :

Android est un système d'exploitation mobile développé par Google, basée sur Linux, et distribuée sous licence open source Apache v2. Il est conçu pour faire fonctionner une grande variété d'appareils, notamment les smartphones et les tablettes. Sa particularité réside dans son caractère open-source : son code source est librement accessible, permettant aux fabricants et aux développeurs de le modifier et de l'adapter selon leurs besoins. [23]

Android est développé par l'OHA (Open Hanset Alliance), une alliance internationale de compagnie.



Figure 3.1 : Application Android.

3.3. Historique :

En juillet 2005, Google acquiert Android Inc., une petite société développant des applications pour téléphones mobiles, rendant ainsi compte de l'intérêt croissant que le géant du web porte à ce marché en pleine explosion, et relayant notamment des spéculations sur la volonté de Google de se lancer dans la téléphonie mobile, bien que les objectifs de ce projet restent encore flous.

Dans ce cadre, une équipe sous la direction d'Andy Rubin, co-fondateur de la société, travaille à la conception d'un système d'exploitation mobile basé sur Linux. On a vu fleurir durant les deux années suivantes de multiples informations contradictoires rapportées par la presse, alimentant les rumeurs : Google ne travaillait-il pas de fait uniquement sur des logiciels mobiles pour son moteur de recherche ou encore sur un téléphone Google ?

Finalement, le 5 novembre 2007, l'OHA (Open Handset Alliance) est créée, avec comme objectif premier de définir des standards open source pour les terminaux mobiles, et le premier de ces standards est le système Android, qui repose sur le noyau Linux 2.6. [24]

Depuis sa première apparition en 2008 Android a connu un cheminement rapide, marqué par plusieurs mises à jour apportant une multitude de corrections de dysfonctionnements d'ordre fonctionnelles et sécuritaires.

Le tableau ci-après présente les différentes versions majeures d'Android :

Nom	Version	Date de sortie
Android	1.0	Septembre 2008
Cupcake	1.5	Avril 2009
Donut	1.6	Septembre 2009
Eclair	2.0 - 2.1	Octobre 2009 – Janvier 2010
Froyo	2.2	Mai 2010
Gingerbread	2.3	Décembre 2010
Honeycomb	3.0 - 3.2	Février 2011
Ice Cream Sandwich	4.0	Octobre 2011
Jelly Bean	4.1 - 4.3	Juin 2012 – Juillet 2013
KitKat	4.4	Octobre 2013
Lollipop	5.0 - 5.1	Novembre 2014
Marshmallow	6.0	Octobre 2015
Nougat	7.0 - 7.1	Aout 2016
Oreo	8.0 - 8.1	Aout 2017
Pie	9	Aout 2018
Android	10	Septembre 2019
Android	11	Septembre 2020
Android	12	Octobre 2021
Android	13	Aout 2022
Android	14	Octobre 2023
Vanilla Ice Cream	15	Attendu en 2024

Tableau 1 : L'évolution des version Android à travers le temps.

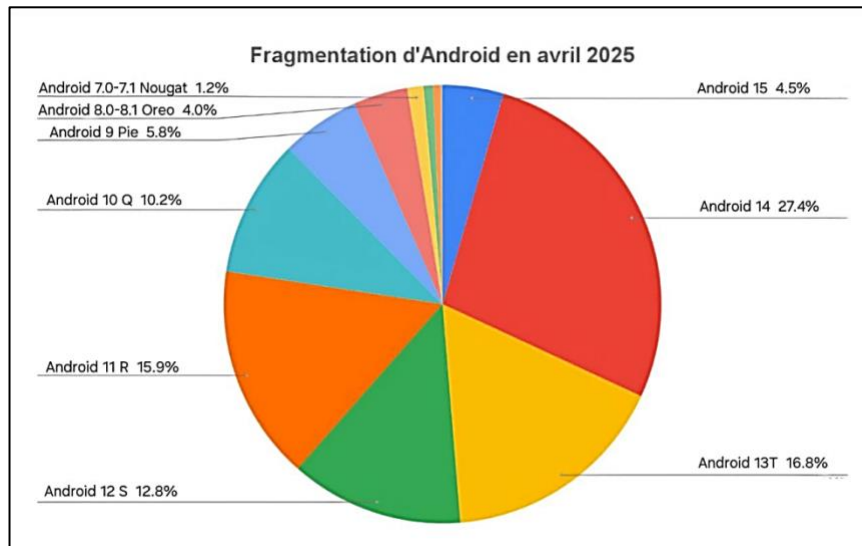


Figure 3.2 : Pourcentage de l'utilisation des version Android en 2025.

3.4. Les différents systèmes d'exploitation mobiles :

Trois systèmes d'exploitation se disputent les grosses parts du marché. Il s'agit d'Android, d'iOS et de Windows Phone :

3.4.1 Android :

Conçu par Google, Android est le système d'exploitation pour smartphones le plus répandu dans le monde car c'est aussi l'une des bases de la conception et du développement du métier de développeur mobile, notamment dans le domaine des systèmes Android, réputés pour leurs larges possibilités de personnalisation tant chez les fabricants de téléphones que chez les utilisateurs. Ce système est également associé à une large et pléthorique boutique d'applications qui répond à tous les besoins et toutes les préférences. Il faut aussi mentionner le large choix d'appareils sous Android qui est accessible à un grand nombre d'utilisateurs quel que soit son budget ou ses goûts. [25]

3.4.2 IOS :

Élaboré par Apple, iOS (pour Internetwork Operating System) fait partie des systèmes d'exploitation mobiles les plus répandus au monde. Exclusivement réservé aux produits de la marque et notamment l'iPhone, l'iPad ou l'iPod Touch, iOS propose une interface agréable et

intuitive du point de vue de l'expérience utilisateur. Techniquement, il repose sur une architecture solidement sécurisée via des mises à jour fréquentes, un app store duquel les applications sont contrôlées et un chiffrement des données toujours plus performant. L'ensemble des spécificités de sécurité offertes permet une fiabilité accrue et une sécurité de plus en plus renforcée, gage d'une parfaite continuité de toute la gamme des produits Apple au sein de l'univers (également nommé écosystème) de la marque à la pomme via des services tels que Handoff, AirDrop ou iCloud. [25]

3.4.3 Windows Phone :

Le système d'exploitation mobile Windows Phone, développé sous l'égide de Microsoft, est en quelque sorte le « dernier-né » des trois systèmes en présence (Android et iOS sont plus matures), il s'en distingue par l'originalité de son interface à tuiles dynamiques et par une intégration facilitée avec les services Microsoft. Sa sécurité est améliorée par le système de « sandboxing » des applications. En revanche, son store d'applications manquant largement de contenu et sa personnalisation peu développée ne favorisent clairement pas son adoption. [25]

3.4.4 Autres systèmes d'exploitation :

En plus des trois principaux systèmes d'exploitation présentés précédemment, il existe d'autres alternatives, moins connues ou désormais abandonnées. Parmi elles, on peut citer :

- **Tizen** : basé sur Linux, ce système se montre performant et personnalisable. Pas assez pour faire parler de lui au-delà de quelques produits Samsung (montres connectées, smartphones pour le marché indien) malgré ses qualités.
- **BlackBerry OS** : développé par BlackBerry, ce système reconnu pour son haut niveau de sécurité et son chiffrement des données, mais il n'a pas pu, avec la montée de l'iOS et de l'Android, résister à la disparition progressive de la plateforme. [25]

3.5. Caractéristiques d'un système Android :

Les principales caractéristiques du système d'exploitation Android, développé par Google, sont les suivantes :

- **Bibliothèques logicielles** : Android est constitué de divers types de bibliothèques et de programmes que les applications utilisant Android peuvent appeler pour bénéficier d'une fonctionnalité particulière.
- **Applications intégrées** : Le système est fourni avec des applications par défaut comme le carnet d'adresses, l'agenda, le web browser, et le programme permettant de passer des coups de téléphone.
- **Framework de développement** : Un kit de développement (SDK) orienté pour les développeurs souhaitant réaliser des applications pour la plate-forme Android.
- **Gestion de fichiers multimédias** : Le système est sensible aux différents formats du multimédia : de nombreux formats audio, vidéo ou image sont gérés comme MPEG4, H.264, MP3 pour le son, JPG, PNG, GIF pour l'image.
- **Services intégrés** : Android est riche d'un environnement de nombreux services permettant par exemple d'accéder aux capteurs de mouvement, à la caméra, au GPS, à l'écran tactile, aux fichiers du système de stockage, au réseau Internet, d'eux même envoyer des SMS, de faire du multitâche (c'est à dire avoir plusieurs applications en fonction simultanément).
- **Connectivité** : Le système permet de se connecter à différents réseaux de communication (Bluetooth, Wifi, GSM, UMTS). [26]

3.6. Architecture du système Android :

La Figure ci-dessous montre les différentes couches qui composent le système d'exploitation (OS) Android. Le système d'exploitation Android est divisé en cinq sections réparties sur quatre couches principales :

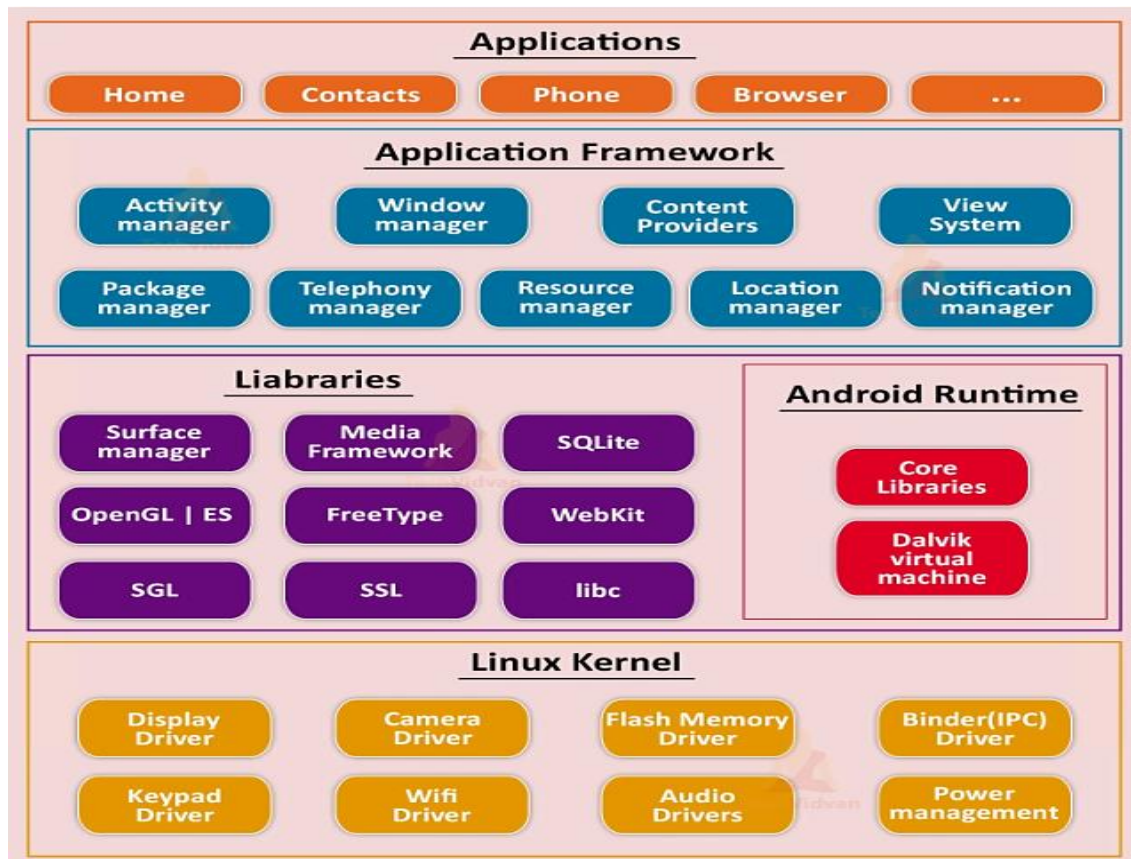


Figure 3.3 : Architecture du système Android.

1. Noyau Linux :

C'est la base du système Android. Dans cette couche se trouvent les pilotes de très bas niveau permettant de faire fonctionner les composants matériels (écran, clavier, Wi-Fi, audio, etc.) d'un terminal Android. [27]

2. Bibliothèques natives :

Cette couche réunit le code de base pour l'implémentation des fonctionnalités essentielles du système Android. Par exemple la bibliothèque SQLite permet d'accéder aux bases de données locales, alors que WebKit permet d'accéder au web. [27]

3. Environnement d'exécution Android (Android Runtime) :

Au même niveau que les bibliothèques, l'environnement fournit les API de base qui doivent être accordées par les programmes chargés de l'implémentation des applications Android en Java, en particulier il fournit la machine virtuelle Dalvik, spécifiquement écrite pour Android. Chaque application est pourvue d'un processus distinct et d'une instance de la machine virtuelle Dalvik, qui est une machine virtuelle optimisée pour faire fonctionner des applications dans une plate-forme mobile à faible mémoire et à faible puissance. [27]

4. Framework d'application :

Cette couche fournit l'interface à travers les organisations de services et composants développés et intégrés dans l'environnement Android et utilisés par les développeurs dans le développement d'applications comme la gestion des activités, des ressources, des notifications ou des services en arrière-plan. [27]

5. Applications :

C'est le plus élevé du système. Et elle est la plus visible car elle regroupe les applications préinstallées (telles que Téléphone, Contacts, Navigateur...) ainsi que toutes celles que l'utilisateur peut télécharger et installer depuis le Google Play Store. [27]

3.7. Les avantages du Système Android :

Les principaux avantages du système Android sont les suivants :

- ✓ **Open source** : Le système d'exploitation Android est construit sur une base rendue accessible aux programmeurs à travers un noyau Linux open source, inscrit dans une démarche d'ouverture, autorisant ainsi une modification et un développement commun du code source, favorisant innovation et coopération.
- ✓ **Personnalisation avancée** : Sur Android, les utilisateurs ont la possibilité de personnaliser leur appareil depuis l'écran d'accueil, en choisissant leurs fonds d'écran jusqu'aux thèmes, en passant par les widgets, et les lanceurs. Ils peuvent donc opter

pour le style qui leur ressemble, ce qui n'est pas le cas d'un autre concurrent direct, l'iOS.

- ✓ **Multitâche** : Le système d'exploitation mobile Android permet d'exécuter plusieurs tâches simultanément. L'OS permet d'exécuter plusieurs applications à la fois. Les utilisateurs peuvent passer très facilement d'une tâche à l'autre, les applications peuvent fonctionner en simultané au besoin. D'autre part, ces applications peuvent également être affichées en mode écran partagé, si l'on souhaite les utiliser côte à côte.
- ✓ **Écosystème riche en applications** : Les utilisateurs du système d'exploitation Android se tournent vers le Google Play Store, un magasin d'applications dans lequel ils peuvent lire, télécharger et installer des nombreuses applications, jeux, films, musiques, livres, et autres contenus proposés. Le Play Store s'inscrit dans cette dynamique avec sa multitude d'options. Peut ainsi facilement trouver les outils correspondant à ses besoins.
- ✓ **Accessibilité** : Android met à la disposition des utilisateurs en situation de handicap ou ayant des besoins spécifiques une variété étendue de fonctionnalités d'accessibilité proposées par le système d'exploitation, dont un lecteur d'écran, des gestes d'agrandissement, l'inversion des couleurs, un synthétiseur vocal... [28]

3.8. Environnements de développement :

3.8.1. L'outil App Inventor :

App Inventor est une application développée par Google. Elle est actuellement entretenue par le *Massachusetts Institute of Technology (MIT)*.

App Inventor est un IDE qui permet aux utilisateurs qui possèdent un compte google la création d'applications destinées à des systèmes Android (téléphones, tablettes, ...).

L'IDE d'App Inventor se présente sous la forme d'une application en ligne, accessible depuis le site web <https://appinventor.mit.edu/> d'App Inventor, et permettant de créer un projet :

- Création de la page graphique de la future application : mise en place des objets nécessaires à l'application sur une page de travail.

- Création d'un programme : association des blocs liés aux objets précédemment définis. Ce programme permet également le téléchargement de l'application vers le média désiré (tablette, téléphone...). [29]



Figure 3.4 : Logo du MIT App Inventor.

3.8.1.1. Pourquoi choisir App Inventor :

MIT App Inventor représente une plateforme de conception et développement destinés à des applications mobiles qui se veut simple et accessible au plus grand nombre, aux novices en programmation notamment, permettant de créer des applications Android grâce à une interface graphique à base de blocs logiques impossible à appréhender sans une maîtrise des langages informatiques requis d'emblée par la programmation informatique traditionnelle.

Son environnement de développement destiné à un usage pédagogique par le MIT devient le support d'une approche pédagogique progressive à l'apprentissage des concepts de programmation en permettant un niveau de complexité suffisant pour des projets aussi concrets que fonctionnels y compris des projets d'IoT via des services comme Thingier.io.

De plus MIT App Inventor est gratuit, accessible en ligne, il bénéficie en effet d'un large public et d'une communauté bien vivante, riches en tutoriels et autres ressources, le positionne en cible des projets étudiants et en prototypages rapides dans le cadre d'activités pédagogiques ou expérimentales.

3.8.1.2. Historique du logiciel App Inventor :

- En 2009, Google lance le développement d'un logiciel App Inventor sur les recherches du MIT sur l'enseignement de l'informatique.
- En 2011, Google rend App Inventor open source et confie la suite du projet au MIT.
- En 2012, la version bêta de MIT App Inventor 2 (AI2) est disponible.
- En 2013 : Lancement officiel d'App Inventor 2 avec une nouvelle interface sur la base de Blockly. [30]

3.8.1.3. Structure d'application Inventor :

Inventor est l'un des outils les plus prisés pour son environnement d'initiation au développement mobile grâce à une double interface de design et de programmation :

✓ L'interface graphique :

Pour réaliser la partie graphique de l'application, App Inventor affiche un écran de téléphone, présentant plusieurs options, dans lequel nous pouvons mettre les éléments prédéfinis que nous allons utiliser. Il existe des éléments graphiques comme des boutons, des labels et des éléments non graphiques comme des capteurs de géolocalisation ou encore des fonctions nous permettant d'effectuer des actions comme la lecture d'une piste audio ou la prise d'une photo. [31]

L'interface graphique d'App Inventor comporte 4 parties principales :

1. **Les palettes** : à savoir, l'écran de téléphone qui va apparaître dans le site et qui va permettre, en glissant les éléments que nous allons faire servir.
2. **La charte graphique** : qui se fait dans la zone "Viewer" du site : c'est la zone où apparaît l'assemblage des composants et les outils que nous allons utiliser pour construire notre application.

- 3. Les propriétés :** cela en est la zone où nous allons préparer chacun des composants de notre application en choisissant par exemple la taille ou encore les couleurs de fonds des boutons, ou le style des cartes que nous souhaitons.
- 4. La partie Média :** permettant d'inclure les images et les fichiers requis à partir d'une autre interface, exemple : aller mieux au niveau de l'interface graphique en tant que design. [31]



Figure 3.5 : L'interface graphique d'App Inventor.

✓ L'interface de programmation (Fenêtre Scratch) :

Comme pour Scratch, l'interface de programmation est très facile à utiliser.

Pour App Inventor, cette interface comporte 3 parties : Les blocs, les visualisations de chaque composante et l'interface de commande et de traitement.

Il existe différents blocs répondant à différents scénarios ou différentes actions que nous souhaitons créer comme :

- **Blocs de contrôle** : qui permettent des schémas conditionnels et récurrents dans la programmation.
- **Blocs de logique** : pour des comparaisons entre toute type de variables.
- **Blocs mathématiques** : pour toute instruction mathématique simple ou complexe.
- **Blocs de textes** : morceaux permettant de traiter le texte. Assimilables au type *char* et à la classe *String* en JAVA.
- **Blocs de listes** : pour créer des listes de données.
- **Blocs "dictionaries"** : qui représentent des structures de données qui associent une valeur à une autre valeur.

Et des blocs couleurs, des blocs de variables et des blocs procédures qui sont un ensemble d'instructions qui effectuent une ou des tâches spécifiques. [31]

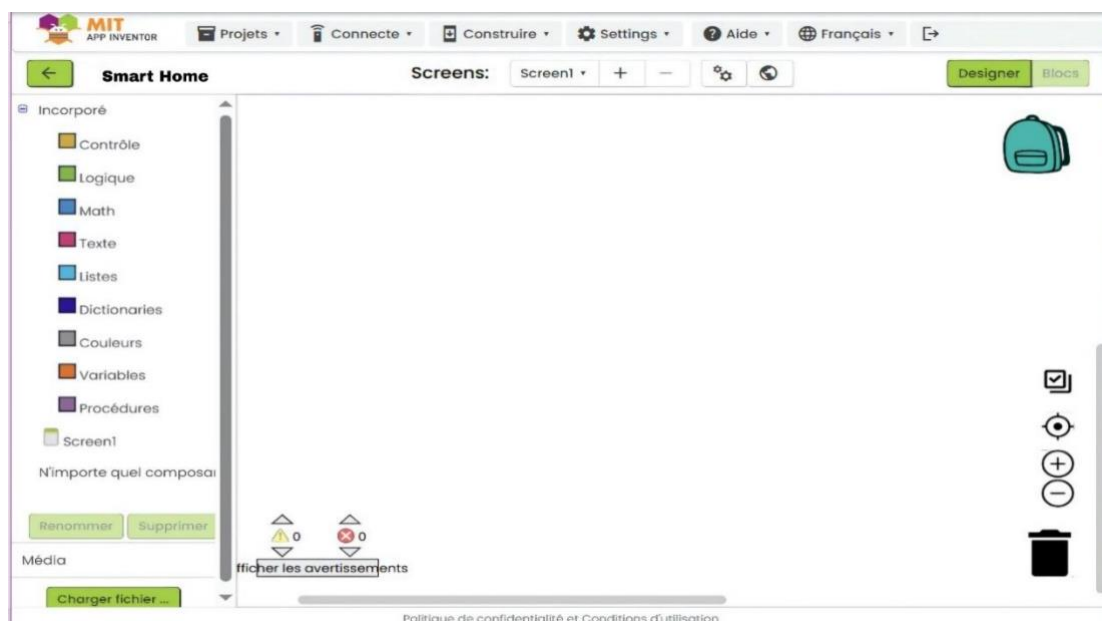


Figure 3.6 : La fenêtre Scratch d'App Inventor.

3.8.1.4. Avantages et inconvénients d'App Inventor :

➤ **Avantages :**

- Environnement simple et efficace
- Pas de langage à apprendre, ni de lignes de code à écrire : pas de risque d'erreur de syntaxe.
- Programmation graphique : des blocs à définir, à déposer et à assembler, comme un puzzle.
- Accès à toutes les ressources de la tablette : écran tactile, multimédia, microphone avec reconnaissance vocale, les capteurs (accéléromètre, boussole, GPS, ...), le WiFi, le bluetooth, connexion internet, mémoire interne pour la gestion de bases de données.
- Test de l'application pendant son développement, par émulation, directement sur la tablette en Wifi (live testing). Test possible également sur le PC.
- Création d'applications fonctionnant sur tous les appareils Android.
- Possibilités d'échanges entre développeurs : application exécutable (.apk) ou code source pour App Inventor (.zip).
- Logiciel libre, gratuit, et multi-plateforme (Windows XP, Vista, Seven, Mac, Linux).
- App Inventor fonctionne en mode Cloud : Les fichiers sont tous sauvegardés sur les serveurs de Google et accessibles à partir de votre compte Google.
- Logiciel de plus en plus utilisé et qui continue d'évaluer grâce au MIT. [32]

➤ **Inconvénients :**

- App Inventor fonctionne en mode Cloud : Des temps de latence peuvent être assez longs. La compilation se fait sur les serveurs du MIT. Une application complexe nécessitera un temps assez long de compilation. Les serveurs peuvent être surchargés voire indisponibles.
- Pas de code Java modifiable : il faut utiliser une passerelle vers le SDK (pour les développeurs en Java).

- Ne permet pas toutes les possibilités offertes par une programmation en Java avec Eclipse (SDK de Google).
- L'anglais peut être un défaut pour certains. [32]

3.8.2. La plateforme Thinger.io :

Thinger.io est une plateforme gratuite et open-source pour l'Internet des objets (IoT). Elle permet de connecter et de gérer des dispositifs IoT avec une interface utilisateur simple et intuitive. Nous pouvons facilement gérer tous vos dispositifs à travers une seule interface intuitive. L'application prend en charge plusieurs systèmes d'exploitation (OS), tels que Windows, Linux, macOS et Android. Elle prend également en charge une variété de dispositifs, tels que les smartphones, les tablettes, les ordinateurs portables et les PC.

Thinger.io fournit une plateforme cloud ouverte, évolutive et sécurisée pour l'IoT. Elle est conçue pour être facilement évolutive et maintenir ses performances. Elle est également sécurisée et fiable, et son architecture est conçue pour protéger les données des dispositifs et du cloud. Elle dispose d'une interface utilisateur simple et intuitive, ce qui la rend adaptée aux débutants.

L'application est open-source, ce qui signifie qu'elle est disponible pour toute personne souhaitant l'utiliser et la modifier. [33]

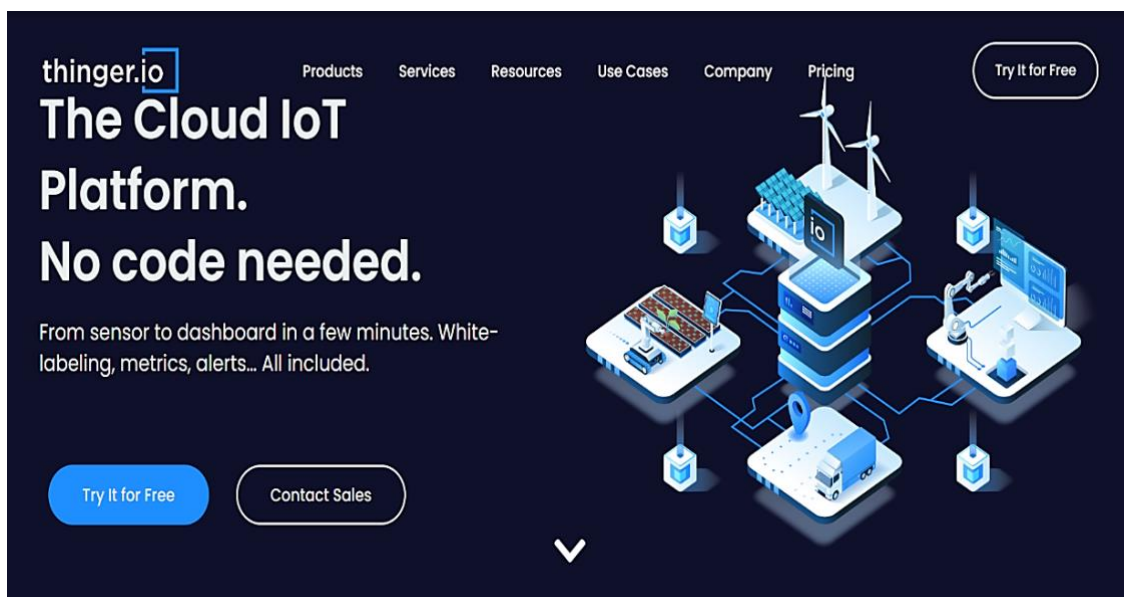


Figure 3.7 : Interface principale de Thinger.io.

3.8.2.1. Fonctionnalités principales de Thinger.io :

La plateforme Thinger.io se compose de deux parties principales : un serveur (backend) IoT et un front-end ou tout autre appareil permettant d'accéder aux différentes fonctionnalités depuis un ordinateur ou un smartphone.

La figure ci-dessous présente les principales fonctionnalités dont elle dispose pour la création de projets IoT.

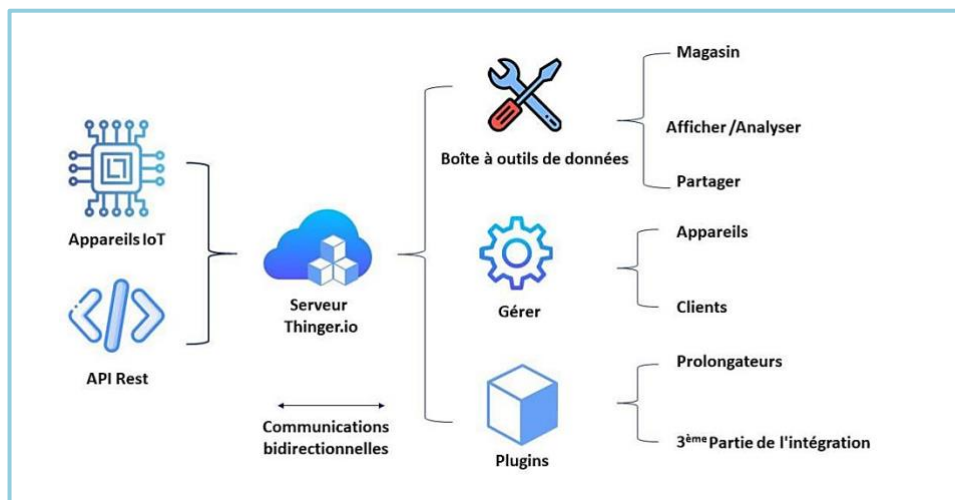


Figure 3.8 : Fonctionnalités de Thinger.io.

- **Connecter les appareils** : tous les appareils sont compatibles, quel que soit le processeur, le réseau, le fabricant : les communications bidirectionnelles des appareils sont ainsi possibles sous Linux, Arduino, Raspberry Pi ou MQTT et même avec des technologies émergentes telles que Sigfox, LoRaWAN ou d'autres ressources de données API Internet.
- **Stocker les données de l'appareil** : en quelques clics, un compartiment de données est versé pour stocker les données IoT de manière scalable, efficace et peu coûteuse, mais la solution permet également l'agrégation de données en temps réel.
- **Afficher les données en temps réel** ou les données enregistrées dans plusieurs widgets tels que des chronologies, des graphiques en anneau, des compteurs ou même des représentations personnalisées créant des tableaux de bord incroyables en quelques minutes.

- **Déclencher les événements et les valeurs de données** avec l'aide d'un moteur de règles intégré Node-RED.
- **Étendre ses fonctions individuelles** avec plusieurs plugins afin d'intégrer les projets IoT à votre logiciel d'entreprise ou à tout autre service Internet tiers.
- **Personnaliser l'apparence** avec un frontend totalement rebranding permettant d'introduire ses couleurs de marque, son logo, son domaine Web. [34]

3.8.2.2. Connection de Thingier.io avec les appareils :

Thingier.io offre un support intégral de nombreux types d'appareils et protocoles, d'applications IoT simples aux systèmes industriels. Pour connecter votre appareil, choisissez la technologie la plus appropriée selon ses capacités et son protocole de communication.

- **Appareils Arduino compatibles** : tout Arduino connecté à Internet, donc les appareils Espressif ESP8266 et ESP32, connectés via le protocole IOTMP qui active une API d'appareil avec connexion OTA (Over The Air) à distance et console distante.
- **Appareils Linux/Raspberry Pi** : tout appareil sous Linux avec protocole IOTMP, y compris l'API d'appareil, SSH distant, proxy pour sockets et services Web distants.
- **Appareils HTTP** : connecter n'importe quel appareil au moyen de requêtes HTTP vers Thingier.io. HTTP (Hypertext Transfer Protocol)
- **Appareils MQTT** : connecter n'importe quel appareil via le protocole MQTT. MQTT (Message Queuing Telemetry Transport)
- **Appareils LoraWAN** : il est possible d'interconnecter tous les appareils, via LoraWAN. [34]

3.9. Conclusion :

Dans ce chapitre, nous avons tout d'abord défini Android ainsi que son historique. Nous avons ensuite présenté les différents systèmes d'exploitation mobiles tels qu'Android, iOS, Windows Phone, etc. Ensuite, nous avons étudié ses caractéristiques et son architecture. Après cela, nous avons exposé l'environnement d'exécution d'Android et ses avantages.

Dans la deuxième partie, nous avons étudié l'environnement de développement MIT App Inventor, où nous avons vu son historique, sa structure, ainsi que ses avantages et inconvénients. Ensuite nous avons présenté la plateforme thinger.io, ses fonctionnalités et sa connexion avec les différents appareils.

CHAPITRE 4 : Conception et implémentation

4.1. Introduction :

Dans ce chapitre, nous abordons en détail le processus de conception et d'implémentation de notre système de maison intelligente. Nous commençons par une description globale du projet et des fonctionnalités proposées, accompagnée d'un diagramme prévisionnel d'avancement.

La suite du chapitre est consacrée à l'implémentation, en distinguant les environnements matériels et logiciels utilisés. Nous présentons les différents composants électroniques intégrés, tels que l'ESP32, les capteurs, les actionneurs, et l'interface utilisateur. Ainsi nous expliquent l'utilisation des outils comme Arduino IDE, App Inventor et Thinger.io. Enfin, nous détaillons le fonctionnement du système et présentons le prototype final de la maquette.

4.2. Description du projet :

Le projet que nous menons a pour but de réaliser un système de logement « intelligent », permettant par Internet le contrôle et la surveillance de différents composants de la maison par une application mobile Android, réalisée par MIT App Inventor, connectée à Thinger.io à partir d'objets connectés.

Ce système repose sur l'emploi de deux cartes ESP32, chacune attribuée à des fonctions spécifiques. Le premier ESP32 assure la gestion de l'environnement intérieur à savoir le réglage de la température et de l'humidité, la détection de gaz, l'ouverture et la fermeture automatiques des fenêtres et de la porte ; le second ESP32 est, pour sa part, chargé de la gestion du niveau d'eau du réservoir, du contrôle des lampes (chambre, salon, cuisine) et de la détection de mouvement. Cette répartition des tâches constitue une organisation distribuée, susceptible d'apporter une plus grande réactivité.

La communication entre l'ESP32 et l'utilisateur se fait par le protocole HTTP sur la base des requêtes API REST mises à sa disposition par Thinger.io en tant qu'interface intermédiaire entre le matériel et l'application Android.

La réalisation de ce projet sera divisée en deux parties principales :

La partie matériels (Hardware) : représente le choix et brochage des composants électroniques (le module ESP32, les capteurs, les servomoteurs...).

La partie logicielle (Software) : expose le développement de l'application mobile, configuration de Thingier.io, programmation de l'ESP32 via le logiciel Arduino IDE et assemblage de l'ensemble au sein d'une maquette en état de fonctionnement.

La figure ci-dessus 3.1 représente l'architecture globale de notre système

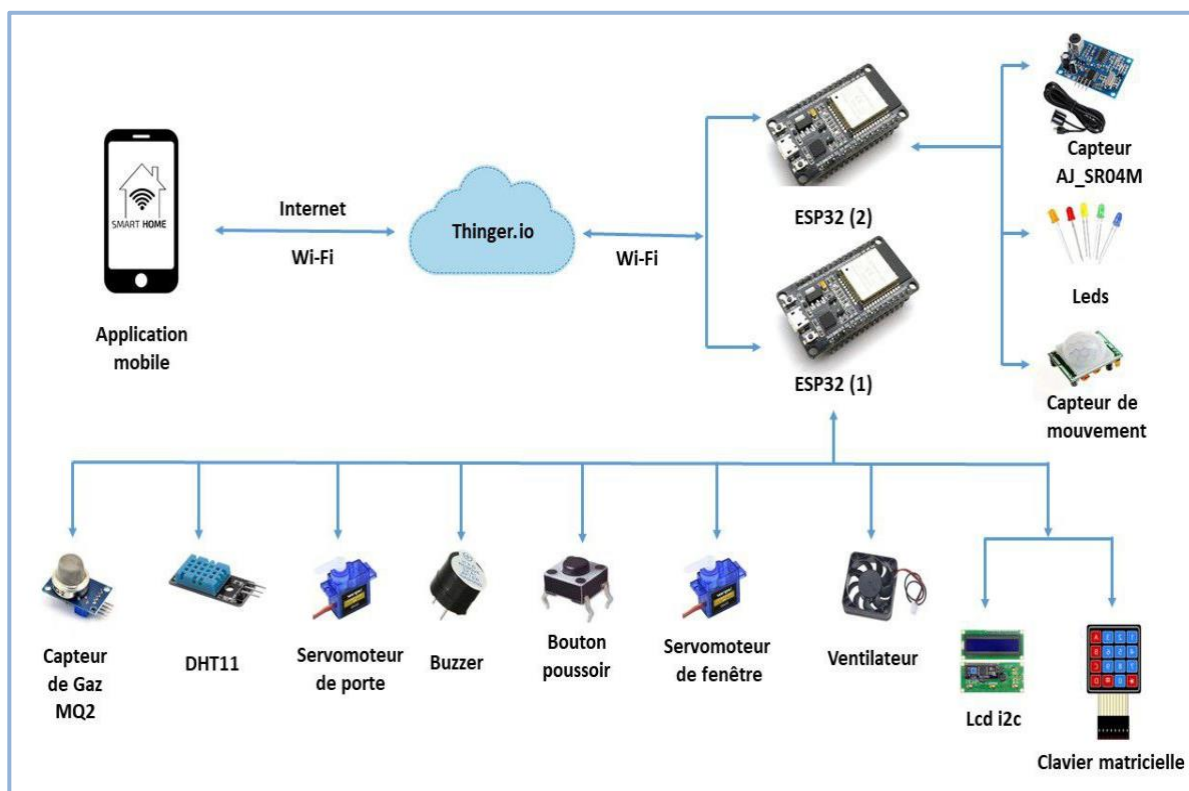


Figure 4.1 : Schéma globale du système.

4.3. Diagramme prévisionnel d'avancement de travail :

Durant l'élaboration de ce travail, nous avons réalisé un diagramme prévisionnel dans lequel nous avons divisé le travail en plusieurs sous objectifs à réaliser. Ce diagramme est présenté dans la figure ci-dessous :

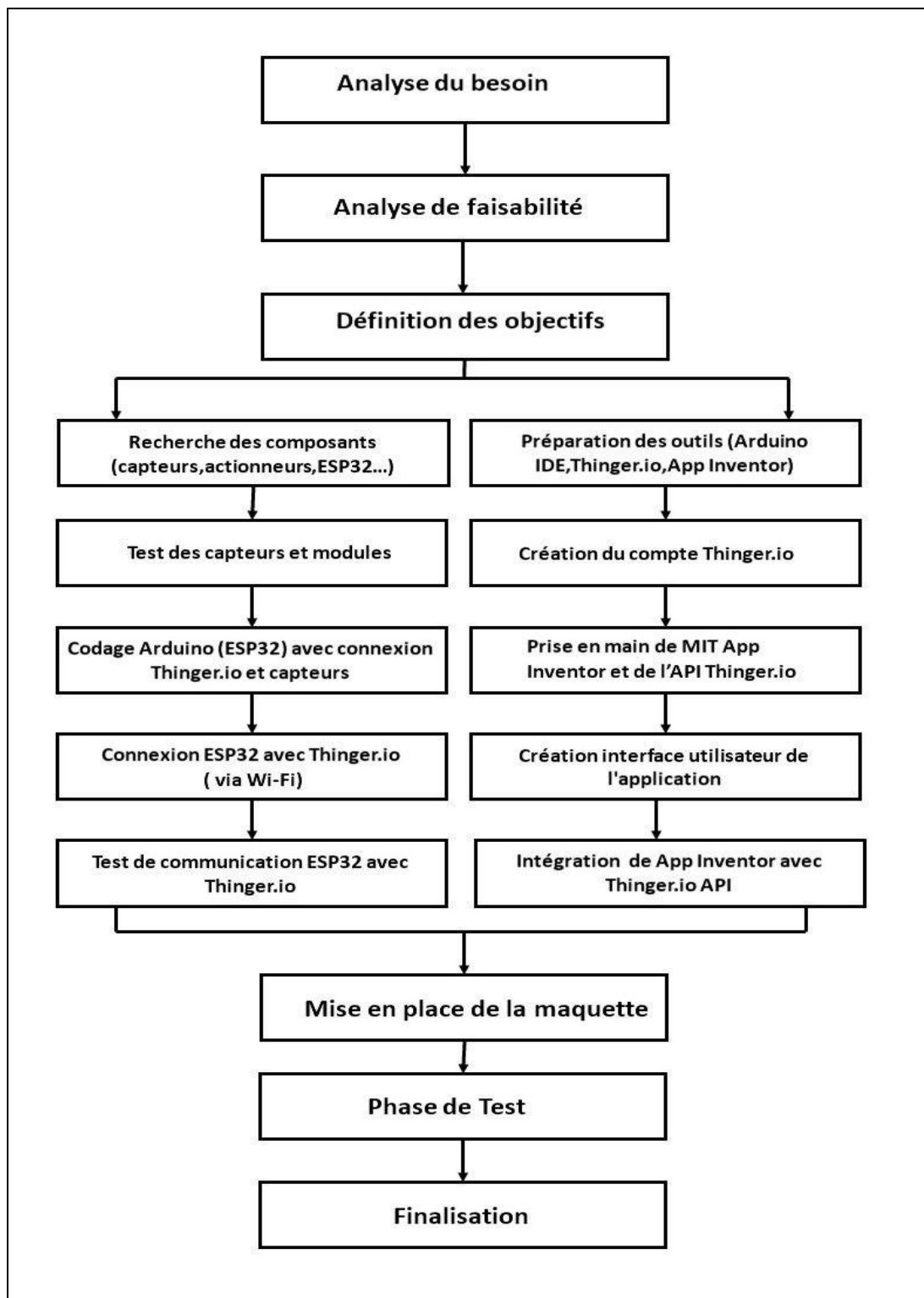


Figure 4.2 : Diagramme prévisionnel d'avancement de travail.

4.4. Les fonctionnalités du système :

Nous aborderons l'ensemble des fonctions offertes par notre système.

1. La notification :

L'utilisateur reçoit une notification sur Telegram s'il y a un problème dans la maison (fuite de gaz).

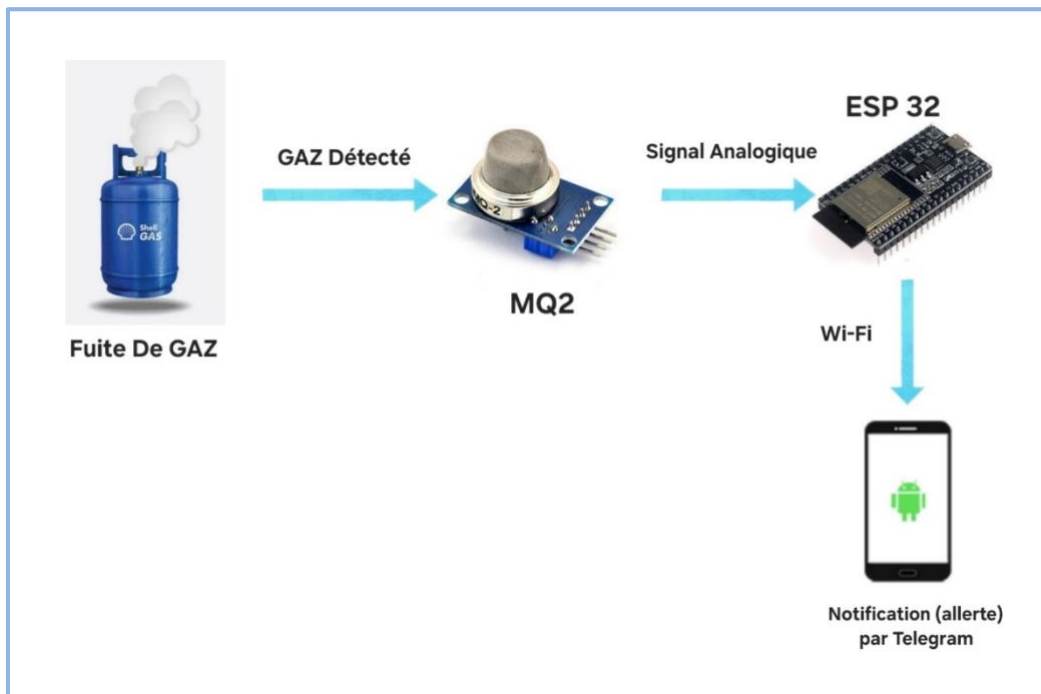


Figure 4.3 : Fonctionnement de la notification de Gaz.

2. Contrôle de température et l'humidité :

Notre système aide à régler la température et l'humidité au sein de la smart home.

Ces paramètres seront exploités pour réaliser les actions automatiques suivantes :

- Afficher de la température de la maison sur le smartphone en utilisant un capteur de température
- Activation automatique du ventilateur dès que la température dépasse 50°C ou si l'humidité dépasse 95 pourcents.

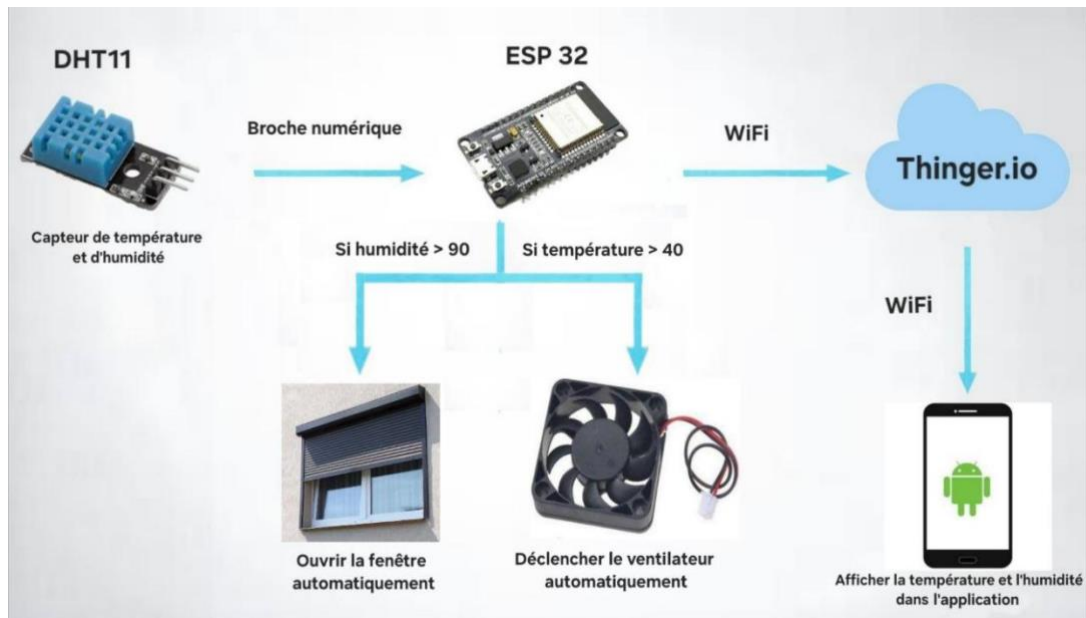


Figure 4.4 : Fonctionnement de capteur DHT11.

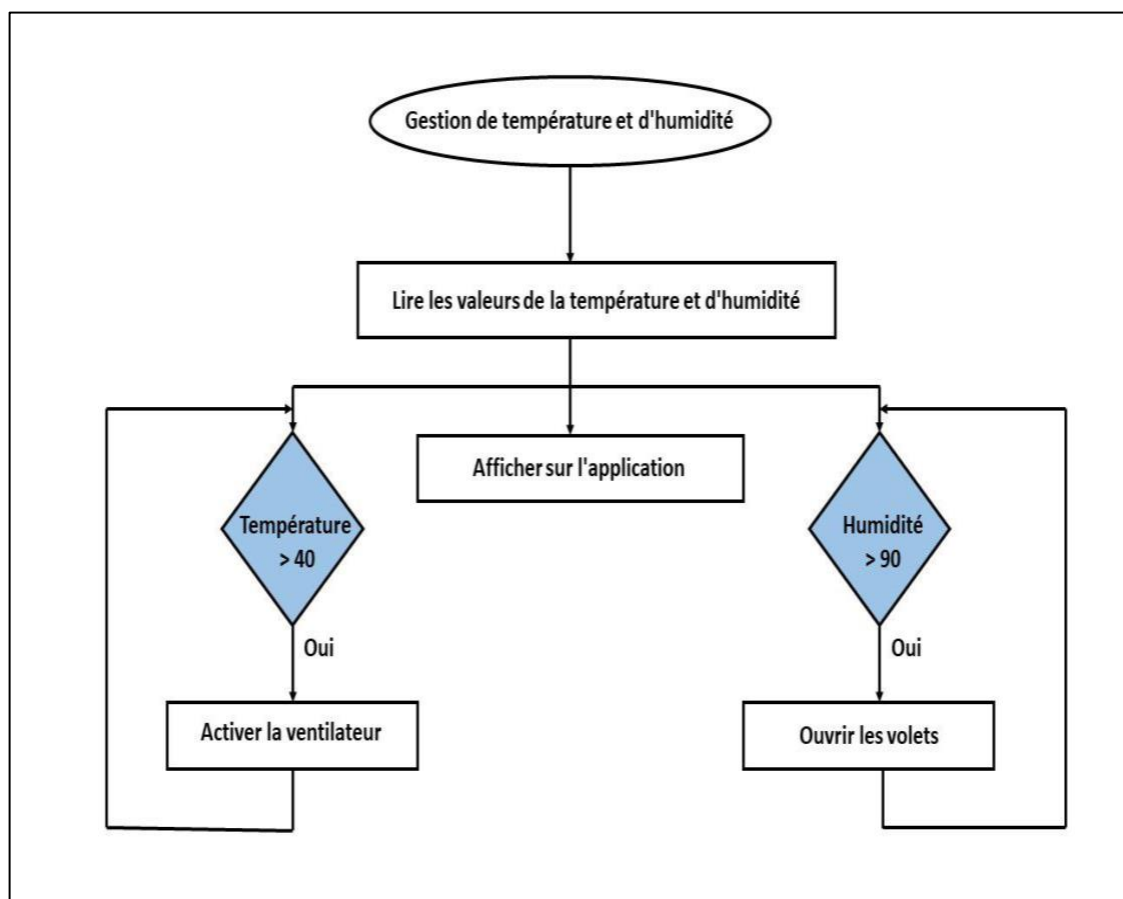


Figure 4.5 : Organigramme de fonctionnement de contrôle de température et d'humidité.

3. Détection du gaz :

Le capteur de gaz pour détecter et répondre à la présence d'une fuite de gaz.

- Une notification aux utilisateurs est envoyée dès qu'une fuite de gaz est détectée, indiquant qu'il y a une fuite de gaz dans la chambre.
- Une alarme sonore est déclencher.
- Activation automatique du ventilateur de la cuisine.

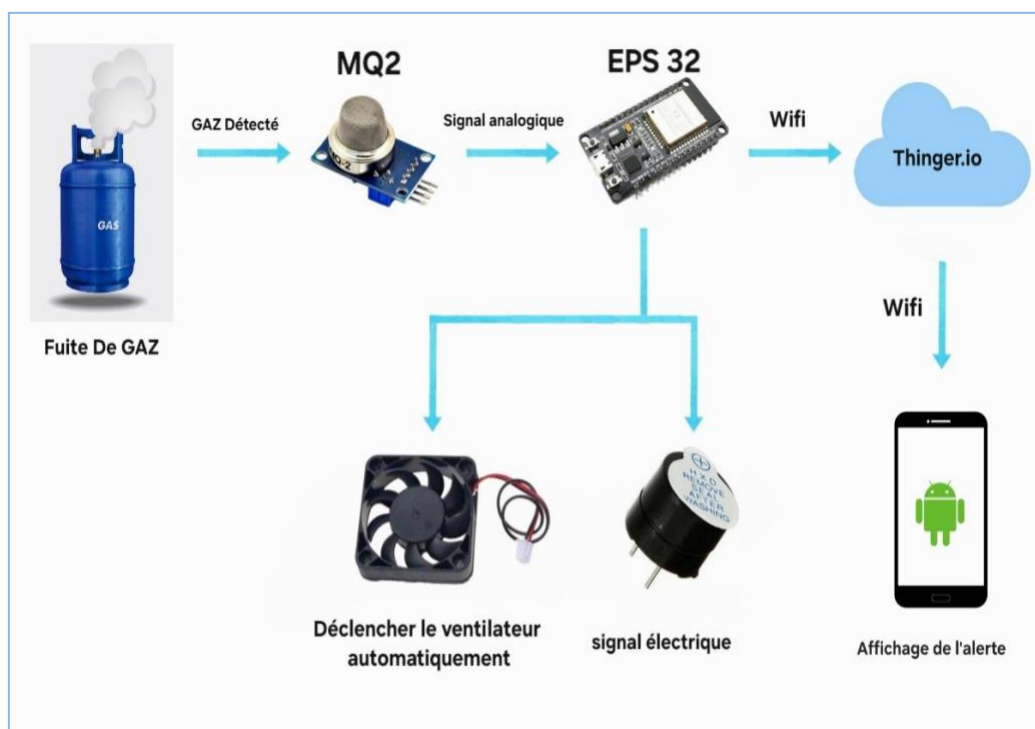


Figure 4.6 : Fonctionnement de la gestion de fuite de Gaz.

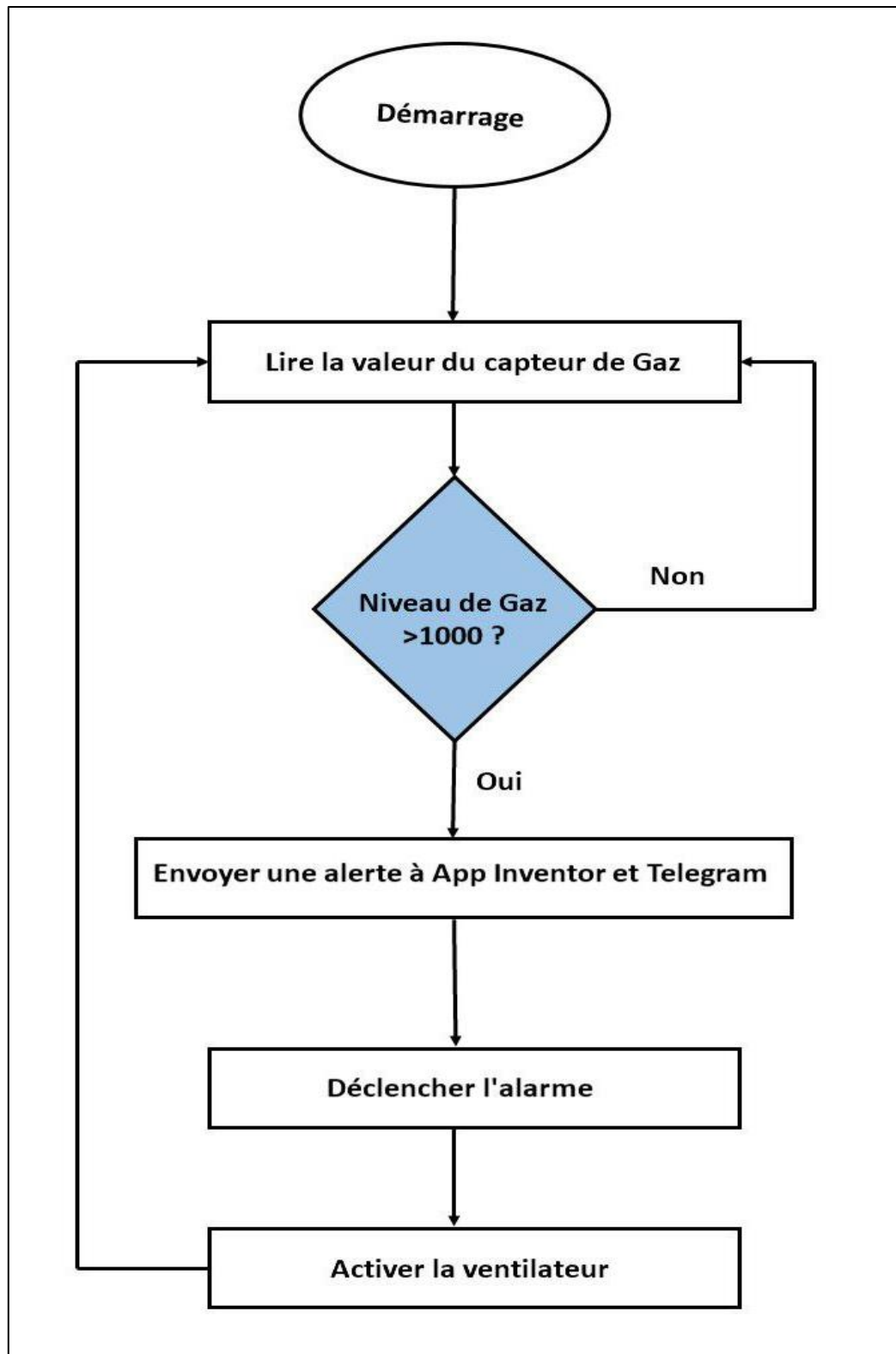


Figure 4.7 : Diagramme de flux du système de détection de gaz avec alerte.

4. Contrôle des lumières :

Le contrôle de la lumière permette aux utilisateurs de faire la gestion des lumières. Nous contrôlons les lampes via un smartphone de n'importe où.

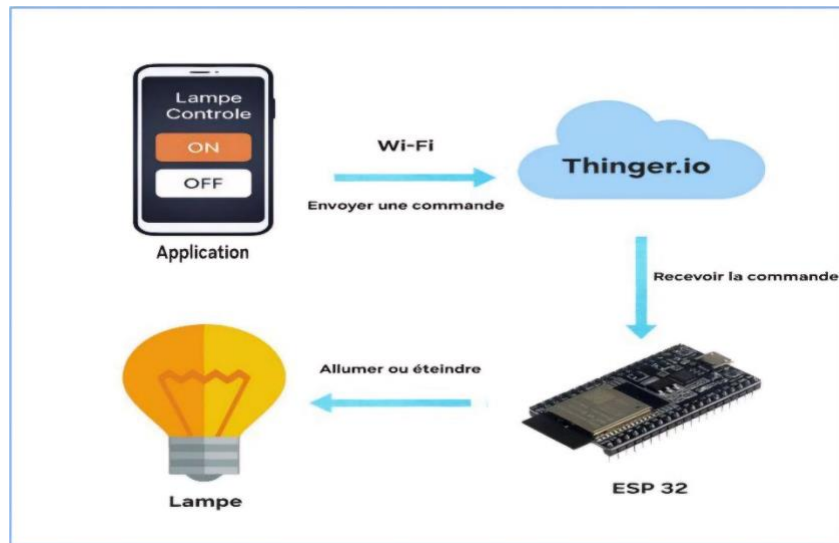


Figure 4.8 : Fonctionnement de l'allumage des lampes par application

5. Contrôle de la porte :

Cette fonction assure l'ouverture de la porte d'une manière plus sécurisé avec un code PIN.

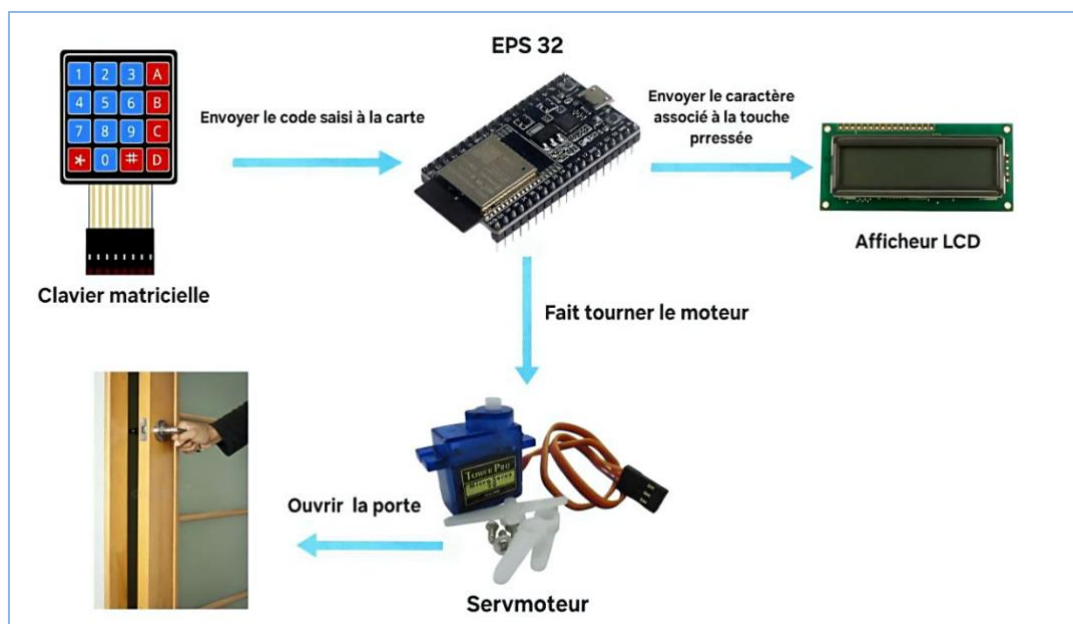


Figure 4.9 : Fonctionnement de l'ouverture de la porte d'entrée.

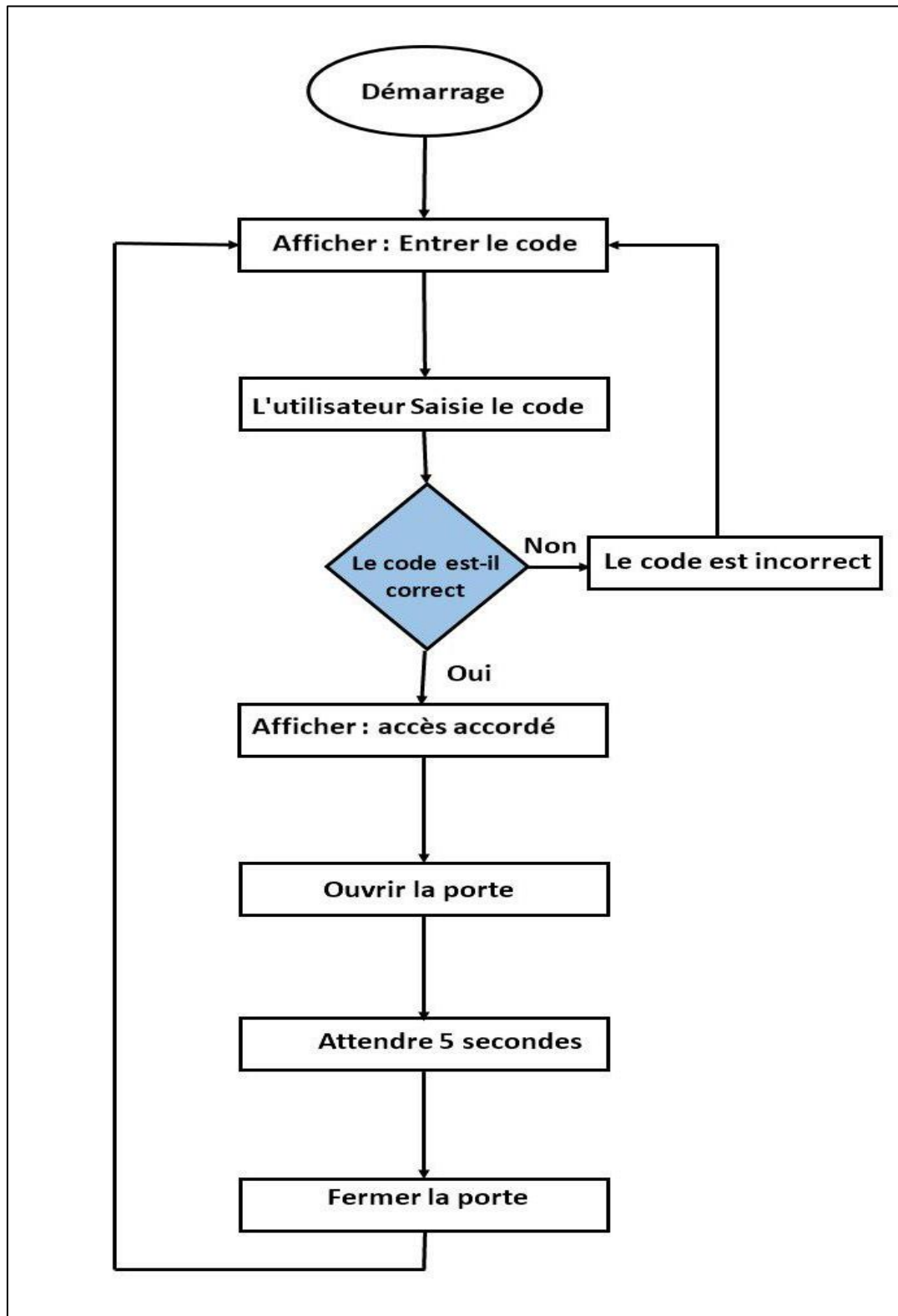


Figure 4.10 : Organigramme de fonctionnement de la porte d'entrée.

6. Contrôle des fenêtres :

L'utilisateur peut contrôler plus aisée l'ouverture des fenêtres d'une manière plus sécurisé en adoptant un système d'accès par l'application mobile.

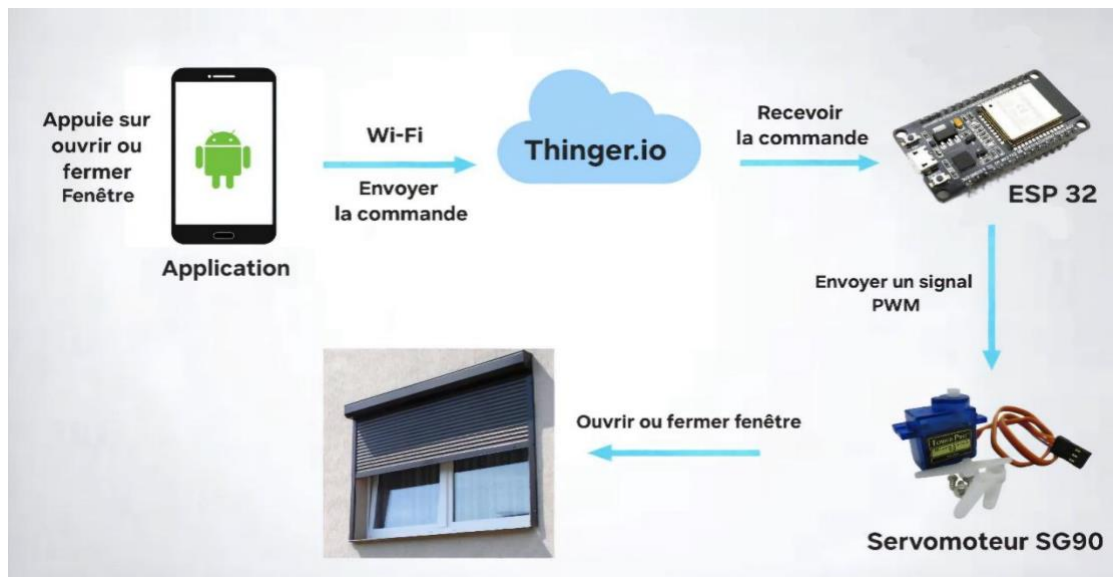


Figure 4.11 : Fonctionnement de gestion des fenêtres.

7. Contrôle de niveau de l'eau dans le réservoir :

L'utilisateur peut contrôler plus facilement le niveau d'eau dans son réservoir et reçoit une alerte sur son téléphone si la quantité d'eau devient inférieure à 20 %.

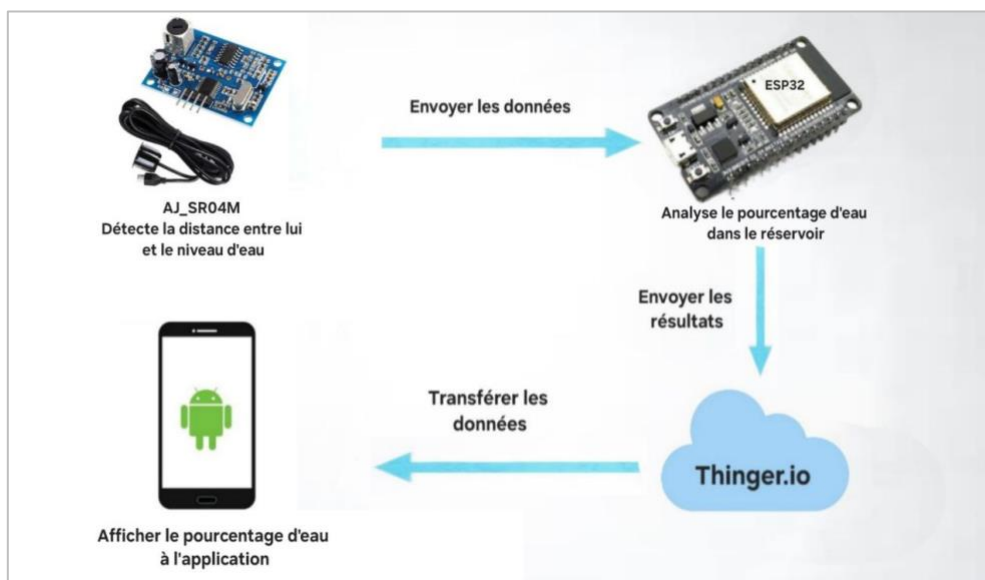


Figure 4.12 : Fonctionnement de contrôle de niveau d'eau.

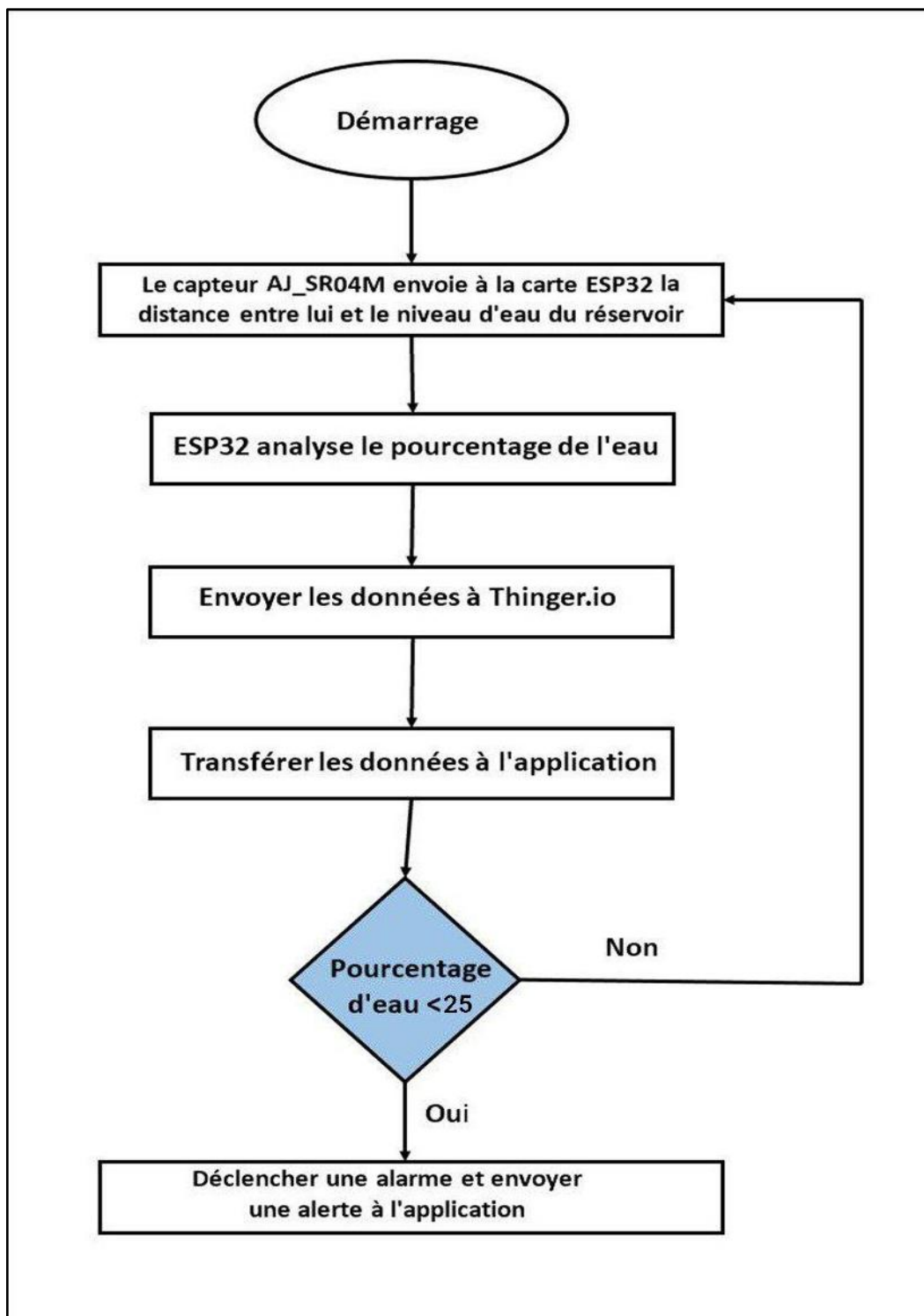


Figure 4.13 : Organigramme de fonctionnement de contrôle de niveau d'eau de réservoir.

8. Détection de mouvement :

Lorsque le capteur de mouvement détecte une présence, il active automatiquement l'allumage de la lampe.

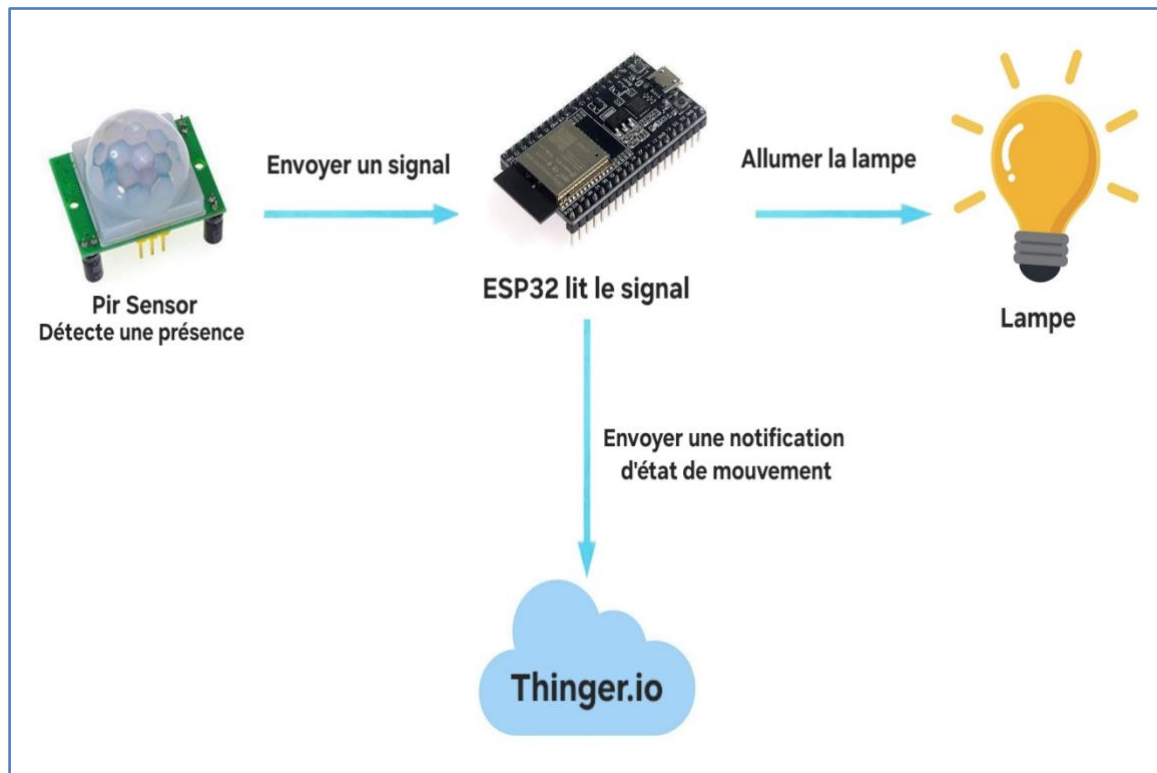


Figure 4.14 : Fonctionnement de capteur PIR.

La figure ci-dessous représente l'organigramme de fonctionnement de ce capteur en temps réel.

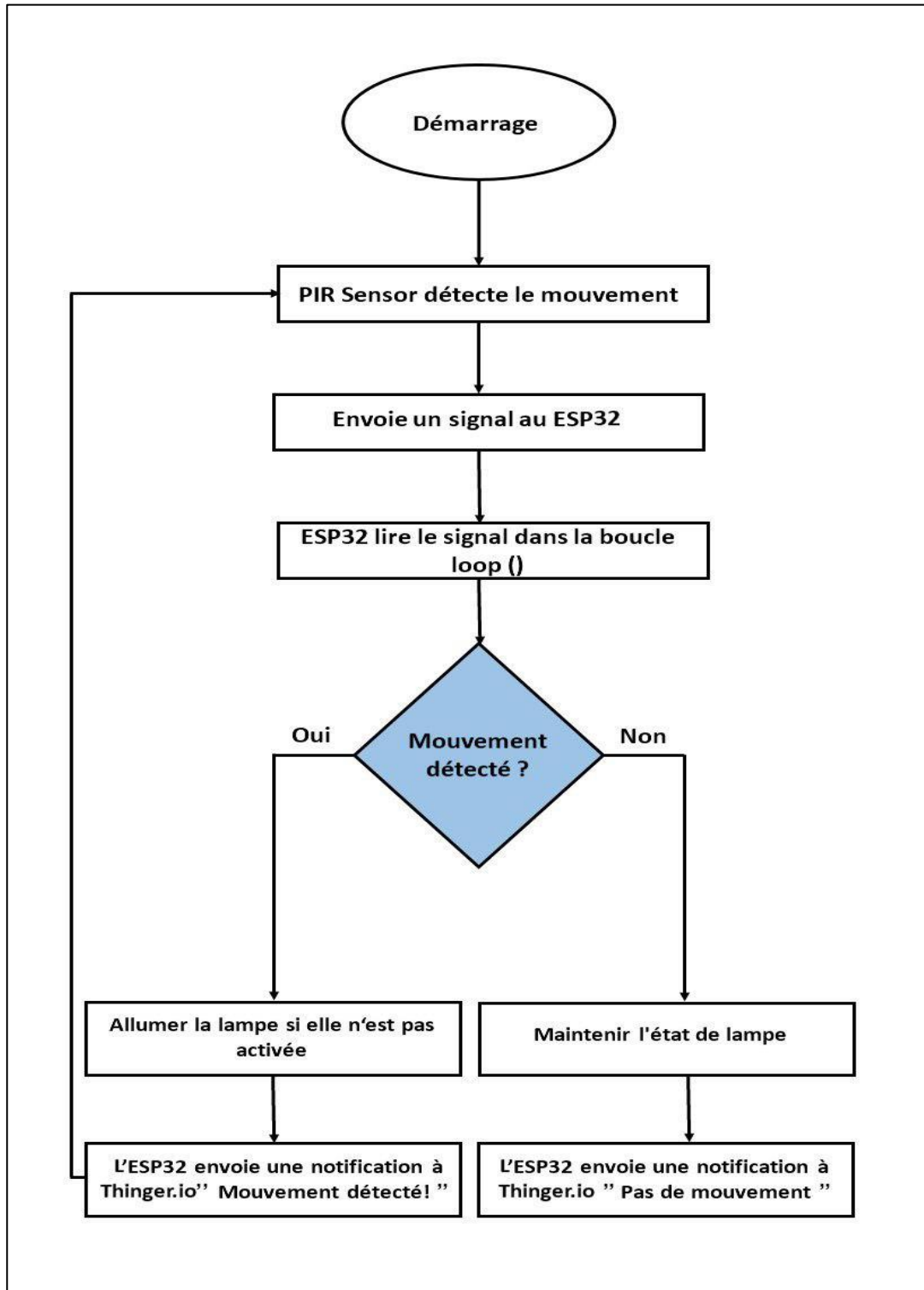


Figure 4.15 : Organigramme de fonctionnement de capteur de mouvement.

4.5. Simulation sous Fritzing :

Avant de passer à la réalisation pratique de notre projet, nous avons utilisé une simulation avec le logiciel Fritzing afin de vérifier le bon fonctionnement du circuit.

Fritzing est un logiciel gratuit de conception, simulation et documentation de circuits électroniques open source pour un public non-professionnel, ainsi que pour les étudiants, les makers et les enseignants. Son cœur de fonction est constitué par la création de circuits sur une breadboard virtuelle, la génération automatique de schémas électriques, la conception de PCB, la génération de fichiers destinés à la fabrication de PCB, le partage du projet de l'utilisateur au sein d'une communauté en ligne.

« FRITZING » disponible et téléchargeable sur ce lien : <https://fritzing.org/>

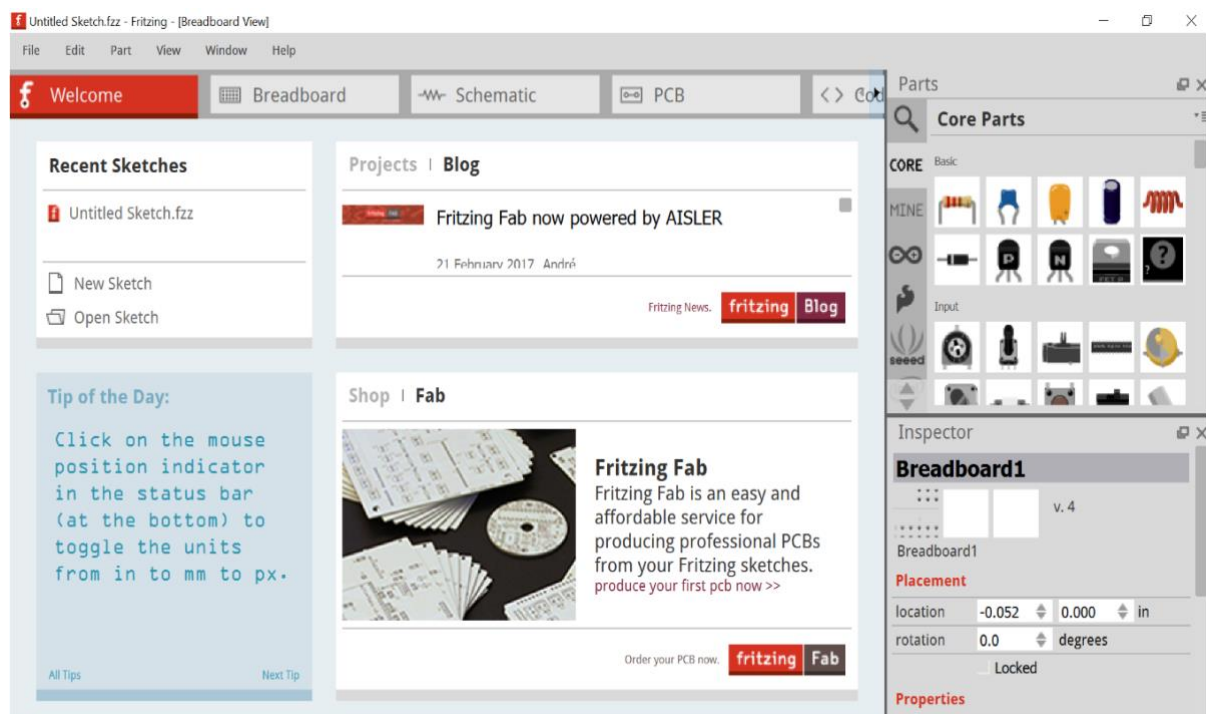


Figure 4.16 : Interface principale de Fritzing.

La figure suivante présente le schéma de simulation des composants matériel utilisé dans le premier ESP32 :

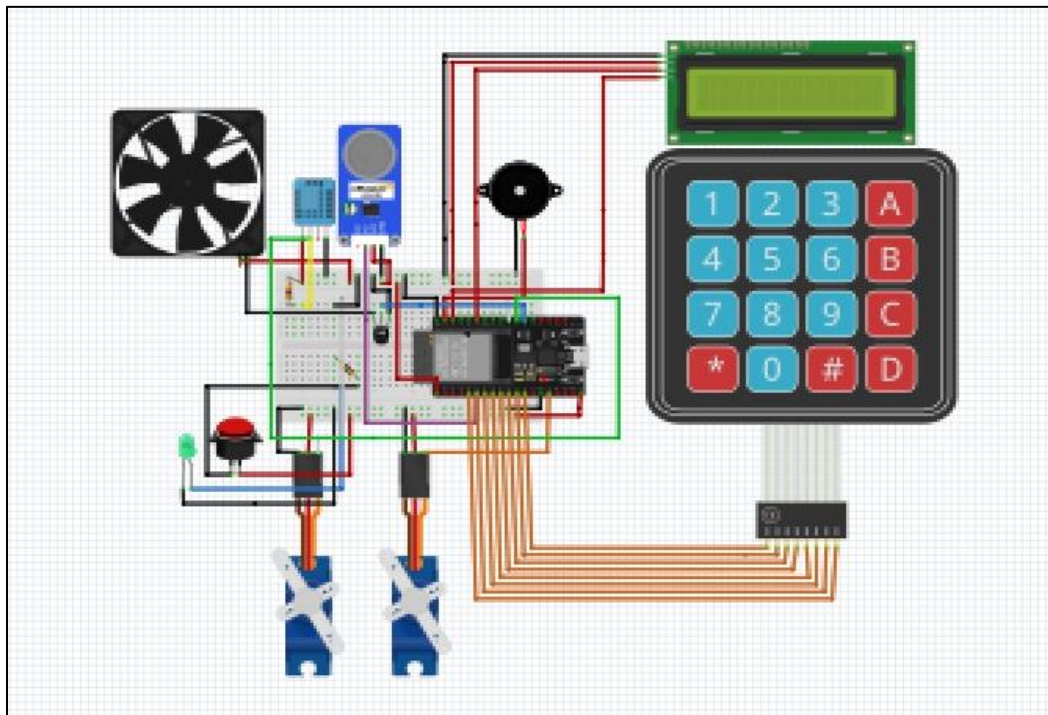


Figure 4.17 : Simulation des composants du premier ESP32.

La figure suivante présente le schéma de simulation des composants matériel utilisés dans le deuxième ESP32 :

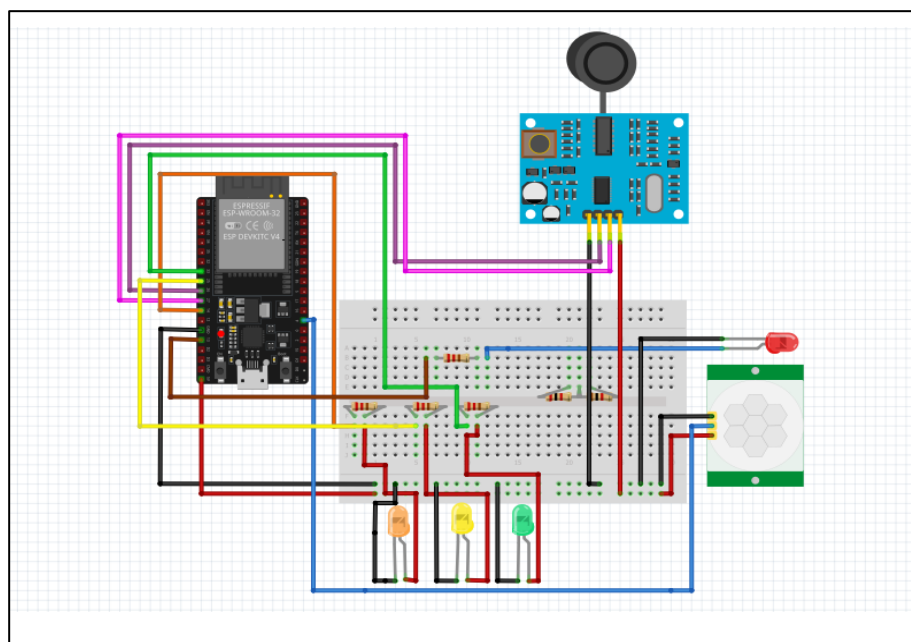


Figure 4.18 : Simulation des composants du deuxième ESP32.

4.6. Implémentation :

Après avoir présenté la conception de notre système, nous allons présenter notre application réalisée ainsi les différents outils nécessaires pour le développement.

4.6.1. Environnement matériel (Hardware) :

Pour mettre en œuvre notre système, nécessite des composants électroniques. Chaque élément a été choisi pour sa fonction au sein du fonctionnement global du système, qu'il soit pour le contrôle des appareils, pour la détection de l'environnement ou pour la communication entre les différents modules. Dans cette partie nous présenterons donc les différents matériels que nous avons utilisés.

4.6.1.1. Module ESP 32 :

La carte de développement ESP-32 DEV KIT V1 est l'une des cartes de développement créées par la carte de développement ESP-32 DEV KIT V1 pour évaluer le module ESP-WROOM-32. Il est basé sur le microcontrôleur ESP32 qui offre une prise en charge Wifi, Bluetooth, Ethernet et faible consommation d'énergie dans une seule puce. La carte de développement ESP-32 DEV KIT V1 utilise le langage de script Lua et est une plate-forme Internet des objets (IoT) open source. [35]

L'ESP32 peut être programmé à l'aide de l'environnement de développement intégré Arduino (IDE) ou d'autres langages de programmation tels que C ou Python.

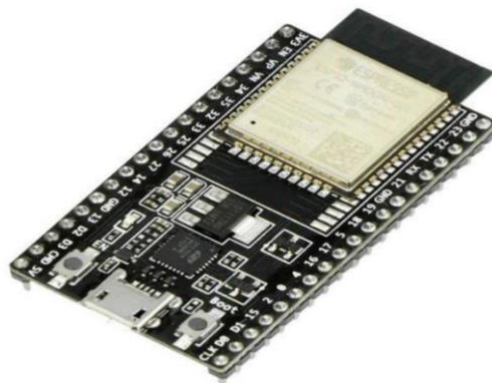


Figure 4.19 : Microcontrôleur ESP32 DEV KIT V1.

➤ **Caractéristique du module ESP 32 :**

Le tableau suivant représente les caractéristiques techniques du microcontrôleur ESP32 :

Caractéristique	ESP32
Microcontrôleur	ESP32-WROOM-32 (double cœur Xtensa LX6 32-bit)
Fréquence d'horloge	Jusqu'à 240 MHz
Mémoire Flash	4 Mo
Mémoire RAM	~520 Ko SRAM intégrée
Wi-Fi	IEEE 802.11 b/g/n (2.4 GHz)
Bluetooth	Bluetooth v4.2 BR/EDR et BLE
Nombre total de broches	38 broches
Nombre de GPIO utilisable	35 à 30
Tension de fonctionnement	3.3 V
Nombre d'entrées analogiques	18
Nombre de PWM	Support sur presque tous les GPIO
USB	Micro-USB
Consommation énergétique	Moyenne à faible (avec gestion d'énergie)
Connexion cloud possible	Oui (avec Thingier.io, Blynk, Firebase, MQTT...)
Capacités spéciales	Capteur tactile, cryptographie, capteur Hall, mode deep sleep

Tableau 2 : Caractéristiques du module ESP32.

➤ Pourquoi ESP32 ?

- **L'ESP32 est open source** : a fortiori les schémas électroniques et de développement (ESP-IDF) ainsi que les bibliothèques. Il peut donc être adapté sur le plan matériel, de tels cartes peuvent être librement produites, et commercialisées sans avoir à payer de licence. Le succès de ce standard a permis sa large diffusion, et a conduit à la constitution d'une grande communauté qui soutient, ou contribue à l'améliorer en permanence.
- **Le coût** : si l'ESP32 est open source, et présente par ailleurs l'avantage d'être très demandée, donc prisé par de multiple façonnier, il existe ainsi de nombreux modules ESP32 compatibles qui sont proposés à très bas prix. On se retrouve communément avec des cartes performantes ESP32 affichées à un prix toujours inférieur à 10 Euros, soit une solution indubitablement économique pour réaliser des projets connectés.
- **La facilité** : L'ESP32 est programmée avec l'IDE Arduino, MicroPython ou bien l'environnement natif ESP-IDF. D'où la possibilité, de la part d'un néophyte ou d'un utilisateur chevronné, de se créer rapidement un prototype fonctionnel en un rien de temps. On peut ainsi développer des applications connectées, même sans connaissance en électronique.
- **ESP32 est multiplateforme** : Le travail sur l'ESP32 peut être envisagé avec la grande majorité des systèmes d'exploitation. En effet, l'ensemble des outils de développement (IDE Arduino, VS Code, PlatformIO) tournent sur de tels systèmes : Windows, Linux, macOS. Ce caractère multiplateforme de l'ESP32 rend disponible le développement d'objets connectés pour un large public.
- **Un choix gigantesque** : Utiliser l'ESP32, c'est aussi « choisir de s'approprier une plateforme à l'échelle de l'imaginaire ». Avec une grande variété d'interfaces, une puissance de traitement efficace et une connectivité intégrée : le champ des possibles est immense pour réaliser une multitude de projets allant de simples objets connectés jusqu'aux solutions de systèmes domotiques complexes.

4.6.1.2. Capteur de gaz MQ2 :

Le dispositif de détection MQ-2 permet de mesurer la concentration de divers gaz ou fumées dans l'air ambiant, avec une sensibilité allant de 300 ppm jusqu'à 10 000 ppm. Puisqu'il est correctement calibré, il peut détecter plusieurs gaz à différents taux de concentration comme le GPL, le propane, le méthane, l'isobutane, l'alcool, l'hydrogène, ainsi qu'une concentration de fumeurs. Par contre, il est plutôt conçu pour être utilisé dans les logements, à la température ambiante. Énergisé par un courant 5V qui lui permet de chauffer son capteur à température de mesure, il utilise une sortie analogique et un petit potentiomètre pour régler la sensibilité à son besoin. [36]

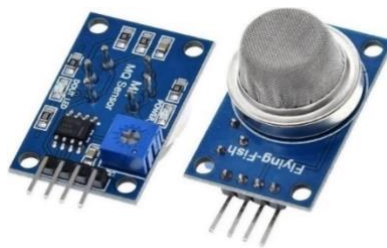


Figure 4.20 : Détecteur de gaz MQ2.

➤ Caractéristiques techniques :

- ✓ La tension de fonctionnement est de + 5V
- ✓ Peut être utilisé pour mesurer ou détecter le GPL, l'alcool, le propane, l'hydrogène, le CO et même le méthane
- ✓ Tension de sortie analogique : 0V à 5V
- ✓ Tension de sortie numérique : 0 V ou 5 V (logique TTL)
- ✓ Durée de préchauffage 20 secondes
- ✓ Peut être utilisé comme capteur numérique ou analogique
- ✓ La sensibilité de la broche numérique peut être modifiée à l'aide du potentiomètre

➤ Brochage :

- Vcc est connecté à la broche 3.3V de l'ESP32 (1).
- GND est connecté à la masse GND de l'ESP32 (1).
- La sortie A0 (analogique) est reliée à l'entrée analogique GPIO 35 de l'ESP32.

4.6.1.3. Détecteur de mouvement :

Ce capteur a pour fonction fondamentale la détection de toute anomalie de présence dans un environnement donné. Son fonctionnement, basé sur la technologie infrarouge, lui donne possibilité de repérer les mouvements, les formes ou encore les changements de volume des différents objets dans son champ (qui est encore limité) de vision. Il sera donc orienté, comme son nom l'indique, vers la protection, l'alerte contre l'intrusion ou les comportements suspects. Un bon choix de positionnement s'avère indispensable pour assurer son efficacité. Son fonctionnement est plutôt basé sur la détection de chaleur d'un corps vivant, de l'homme, étant entendu que c'est le corps qui émettrait la chaleur. [37]

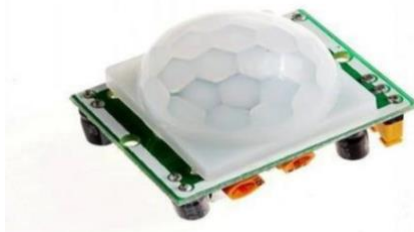


Figure 4.21 : Détecteur de mouvement.

➤ Brochage :

- Vcc est branché à la broche 3.3V de l'ESP32 (2).
- GND est relié à la broche GND de l'ESP32 (2).
- OUT est connecté à la broche GPIO 14 de l'ESP32 (2).

4.6.1.4. Capteur de température et d'humidité :

Le capteur DHT11 permet de mesurer l'humidité et la température, il est basique et peu cher, il est encore choisi aujourd'hui pour des projets où l'on peut s'en contenter pour faire le travail demandé.[38]

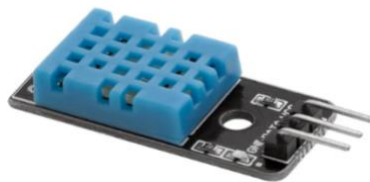


Figure 4.22 : Capteur de température et d'humidité DHT11.

➤ **Caractéristiques techniques :**

- ✓ Gamme de mesure de température : 0°C à 50°C
- ✓ Précision de mesure de température : $\pm 2^\circ\text{C}$
- ✓ Gamme de mesure d'humidité relative : 20% à 90%
- ✓ Précision de mesure d'humidité relative : $\pm 5\%$
- ✓ Alimentation : 3,3V à 5,5V
- ✓ Interface de communication : signal numérique
- ✓ Dimensions : 15 x 12 mm

➤ **Brochage :**

- Vcc est relié à la broche 3.3V de l'ESP32 (1).
- GND est relié à la broche GND de l'ESP32 (1).
- DATA est connecté à la broche GPIO 17 de l'ESP32 (1).

4.6.1.5. Capteur AJ_SR04M :

Le AJ-SR04M est un capteur ultrasonique étanche capable de mesurer la distance en émettant une onde sonore et en mesurant le temps de retour après réflexion. Il permet de mesurer le niveau d'eau, de détecter des obstacles en temps réel dans un milieu humide ou en extérieur.



Figure 4.23 : Capteur HC RS04.

➤ **Caractéristiques techniques :**

- ✓ Dimensions : 45 mm x 20 mm x 15 mm
- ✓ Plage de mesure : 2 cm à 400 cm
- ✓ Résolution de la mesure : 0.3 cm
- ✓ Angle de mesure efficace : 15 °
- ✓ Largeur d'impulsion sur l'entrée de déclenchement : 10 μ s

➤ **Brochage :**

- VCC est connecte à la broche Vin de l'ESP32(2)
- GND est connecte à la broche GND de l'ESP32(2)
- TRIG est connecte à la broche GPIO 27 de l'ESP32(2)
- ECHO est relie à la broche GPIO 26 de l'ESP32(2)

4.6.1.6. Servomoteur SG90 :

Le dispositif de Micro Servo est constitué d'un moteur accordé mécaniquement à un potentiomètre. On utilise pour le commander avec un ESP32 une bibliothèque à cet effet. Le circuit électronique intégré au servomoteur convertit la largeur des impulsions PWM, réceptionnées, en une position physique précise, ainsi, lorsque le servo commande, le moteur tourne jusqu'à correspondre à la position du potentiomètre au point de consigne. Le micro servo est livré avec un kit comptant trois bras en plastique (palonniers) et une vis pour l'insertion aisée dans un système mécanique. [39]



Figure 4.24 : Servomoteur SG90.

➤ Caractéristiques techniques :

- ✓ Dimensions : 22 x 11.5 x 27 mm.
- ✓ Poids : 9 gr. – Tension d'alimentation : 4.8v à 6v.
- ✓ Vitesse : 0.12 s / 60° sous 4.8v.
- ✓ Couple : 1.2 Kg / cm sous 4.8v.
- ✓ Amplitude : de 0 à 180°.

Le servo est équipé d'une prise de type Graupner à 3 fils. La correspondance des fils est la suivante :

- ✓ Marron : masse
- ✓ Rouge : + 5v
- ✓ Orange : commande

➤ Brochage :

Servomoteur de la porte :

- Le fil Vcc (rouge) est connecté à une alimentation 5V externe.
- Le fil GND (marron) est relié à la masse GND commune avec l'ESP32 (1).
- Le fil Signal (orange) est connecté à la broche GPIO 13 de l'ESP32 (1).

Servomoteur de la porte :

- Le fil Vcc (rouge) est connecté à une alimentation 5V externe.
- Le fil GND (marron) est relié à la masse GND commune avec l'ESP32 (1).
- Le fil Signal (orange) est connecté à la broche GPIO 5 de l'ESP32 (1).

4.6.1.7. Buzzer :

Un buzzer est un composant de l'électronique qui émet un son continu quand il est sous tension. Il est fréquemment employé dans les systèmes de signalisation permettant de signaler un état donné (alarme, témoin lumineux, etc.) et peut être mis en service de manière manuelle par exemple en appuyant sur un bouton, ou bien déclenché par un signal électrique. C'est un petit composant à 2 broches, piloté par un microcontrôleur, et qui entre dans la réalisation d'un montage sur planche à pain. [40]



Figure 4.25 : Buzzer.

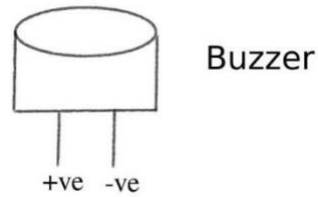


Figure 4.26 : Configuration de la broche du buzzer.

➤ Caractéristiques techniques

- ✓ Tension nominale : 5V DC
- ✓ Plage de tension de fonctionnement : 4V à 7V DC
- ✓ Courant nominal : <30 mA
- ✓ Type de son : Bip continu
- ✓ Fréquence de résonance : ~2300 Hz
- ✓ Niveau sonore : > 85 dB à 10 cm
- ✓ Dimensions : 19mm x 15mm

➤ Brochage :

- Fil positif (+) du buzzer est connectée à la broche GPIO 23 de l'ESP32.
- Fil négatif (-) est reliée à la masse GND.

4.6.1.8. LCD 1602 I2C :

L'afficheur LCD I2C 1602 est ici un écran LCD alphanumérique de 16 colonnes et 2 lignes, d'où son nom « 1602 ». Il est capable d'afficher jusqu'à 16 caractères par ligne, c'est-à-dire 32 caractères au total, répartis sur 2 lignes.

Pour afficher du texte sur un écran LCD 1602 par l'intermédiaire d'une carte ESP32, il faut forcément la bibliothèque de communication I2C, car l'écran LCD 1602 avec module I2C permet un raccordement réduit au niveau des broches de contrôle de celui-ci, donc permet une intégration facile au projet. [41]



Figure 4.27 : Ecran LCD I2C.

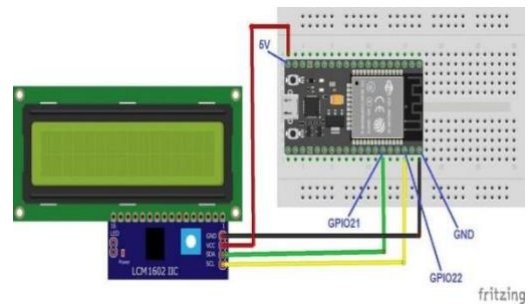


Figure 4.28 : Schéma de câblage de la carte ESP32 avec l'afficheur LCD I2C 1602.

➤ Brochage :

- Vcc du module I2C est connecté à la broche 5V de l'ESP32.
- GND est relié à la masse GND.
- SDA est connecté à la broche GPIO 21.
- SCL est connecté à la broche GPIO 22.

4.6.1.9. Clavier matriciel :

Un clavier matriciel 4x4 est un type de clavier composé de 4 colonnes et 4 lignes de touches. Chaque touche correspond à une combinaison unique de lignes et de colonnes, ce qui permet d'identifier la touche pressée en mesurant la connexion électrique entre ces lignes et colonnes. [42]

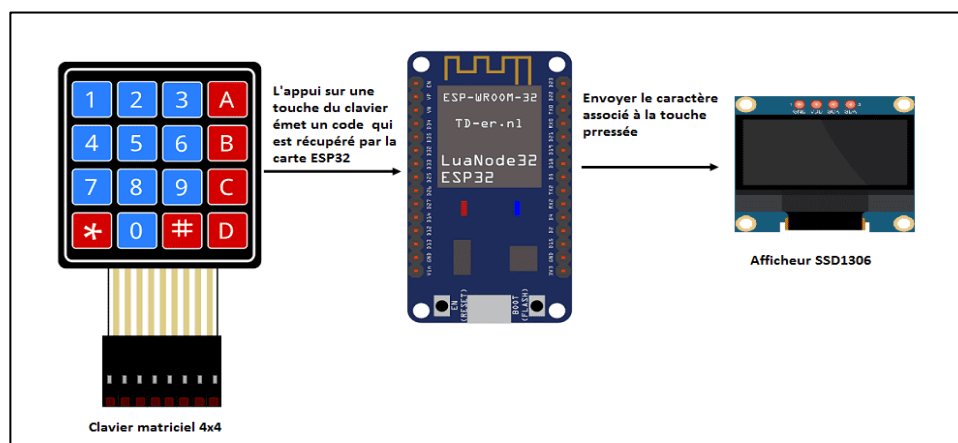


Figure 4.29 : Fonctionnement du clavier matriciel avec la carte ESP32.

➤ Brochage :

- Les lignes R1 à R4 sont respectivement connectées aux broches GPIO 14, 27, 26, 25 de l'ESP32 (1)
- Les colonnes C1 à C4 sont reliées aux broches GPIO 33, 32, 18, 19.

4.6.1.10. Ventilateur :

Un ventilateur 5V est un dispositif électromécanique conçu pour générer un flux d'air à l'aide de pales rotatives, fonctionnant sous une tension nominale de 5 volts (V) en courant continu (DC).



Figure 4.30 : Ventilateur 5V.

➤ Brochage :

- Commande (IN du relais ou base transistor) est reliée à GPIO 16
- VCC est connecté à une alimentation adaptée au ventilateur 5V
- GND est connecté à la masse GND de l'ESP32 (1) via relais.

4.6.1.11. Autres Composants :

✓ **Bouton poussoir :**

Le bouton poussoir se caractérise comme un interrupteur élémentaire, permettant à l'utilisateur soit de l'activer, soit de le désactiver, soit d'établir une relation avec la machine. Il a la particularité de fonctionner de manière transitoire, à savoir que lorsque la pression de l'utilisateur n'est plus là, il revient de lui-même à sa position d'origine. [43]



Figure 4.31 : Bouton poussoir.

➤ **Brochage :**

- Une borne du bouton est connectée à la broche GPIO 21.
- L'autre borne est reliée à la masse GND.
- Une résistance pull-up interne est activée dans le code pour assurer une lecture stable.

✓ **LEDs :**

La « diode électroluminescente » ou LED est un composant électronique qui émet de la lumière quand un courant électrique la traverse.

Actuellement, on peut trouver des LED de plusieurs couleurs : rouge, vert, bleu, jaune ou blanche. Des LED appelées LED RGB permettent d'afficher plusieurs couleurs (rouge, vert et bleu) en combinant ces trois couleurs.

La LED ne laisse passer le courant que dans un seul sens : de l'anode à la cathode. Dans le sens inverse (dit bloquant), le courant ne circule pas.[44]



Figure 4.32 : LEDS



Figure 4.33 : Branches de LED.

➤ **Brochage :**

- La LED du salon a son anode reliée à GPIO 32, et sa cathode à GND avec ESP32 (2).
- La LED de la chambre est branchée entre GPIO 33 (anode) et GND (cathode) avec ESP32 (2).
- La LED de la cuisine est connectée entre GPIO 25 et GND avec ESP32 (2).
- La LED de l'alarme est connectée entre GPIO 13 et GND avec ESP32 (2).

✓ **Résistances :**

Les résistances constituent des composants qui possèdent une valeur de résistance stable et qui ne variera pas. Elles servent en premier lieu à limiter le passage des électrons dans un circuit, autrement dit, à réduire le courant électrique. Ce sont des composants qui qualifient de passifs : ils ne produisent pas d'énergie, mais en consomment. Ils sont souvent utilisés en association avec des composants actifs tels que des amplificateurs, des microcontrôleurs ou d'autres circuits intégrés. Les résistances permettent principalement de limiter le courant, de diviser les tensions ou de tirer vers un niveau logique (haut ou bas) les lignes d'entrée et de sortie. [45]



Figure 4.34 : Résistances.

✓ **Câbles (Jumper wires) :**

Les jumpers désignent les petits connecteurs métalliques permettant d'ouvrir ou de fermer une portion d'un circuit électronique. En reliant deux ou plusieurs broches sur la carte, ils modifient certains comportements du circuit. Ils permettent souvent de configurer des périphériques informatiques, notamment les cartes mères. Par exemple, sur certaines cartes

mères proposant une fonction détection intrusion, la position d'un jumper détermine l'activation ou la désactivation de cette option.[46]



Figure 4.35 : Jumper wires.

✓ Plaque d'essai (BreadBoard) :

Sous le nom de plaque d'essai se désigne un équipement que l'on a coutume d'appeler "breadboard" ou "protoboard", en qualité d'outillage de montage temporaire, précisément dédié à l'assemblage des circuits électroniques. La structure de ce matériel spécialisé se présente comme un réseau d'orifices interconnectés, selon la disposition interne, en règle générale (mais pas uniquement) horizontale ou verticale, des liaisons internes.

C'est bien sûr dans cette organisation matérielle que viennent de façon non soudée s'insérer les composants électroniques et les fils de connexion, pour réaliser tous les essais, aussi bien d'un prototype qu'en phase de conception qu'en phase de tests. [47]

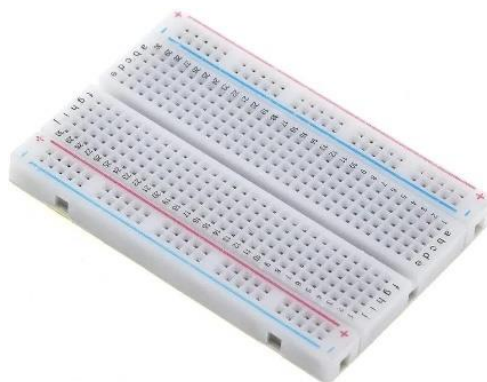


Figure 4.36 : Plaque d'essai.

4.6.2. Environnement logiciel (Software) :

4.6.2.1. Environnement de programmation Arduino IDE :

Pour notre carte ESP32, l'IDE Arduino a été sélectionné en tant que logiciel de programmation. En effet, il constitue l'éditeur du code saisi ou modifié, avant d'être enregistré sur la carte par connexion USB, voie de transfert de l'énergie et des données nécessaires.

L'IDE qui est un Integrated Development Environment, présente une interface graphique comportant un éditeur de texte ainsi que l'ensemble des outils nécessaires au développement, ce qui constitue la meilleure solution pour la saisie, l'enregistrement, la compilation, la vérification puis le transfert du programme au module ESP32. 8.2 Description de L'interface de L'IDE Arduino :

Comme tout langage de programmation, Arduino dispose d'une interface simple et flexible, pouvant être exécutée sur n'importe quel système d'exploitation.



Figure 4.37 : Interface de logiciel Arduino IDE.

➤ Langages de programmation pour Arduino :

Dans l'objectif de coder les algorithmes de tous les divers composants de cette applicative nous allons programmer en langage C, qui est surtout le langage adapté aux systèmes embarqués mais le langage de choix pour l'écriture des bibliothèques (drivers) fournies avec l'environnement de développement Arduino IDE. En effet, ce langage appartient à la famille des langages de bas niveau, car il permet le contrôle de matériel, (avec l'accès notamment aux registres).

➤ Structure générale du programme Arduino IDE :

Un programme est composé de 3 parties :

▪ **Partie déclarative :**

- ✓ Inclusion de **fichiers** (bibliothèques, ...)
- ✓ Déclaration des **variables globales** (accessibles depuis partout dans le programme)
- ✓ Déclaration de **constantes** (comme des variables, mais qui ne peuvent jamais changer)

▪ **Partie configuration (fonction setup ()) :**

- ✓ Initialisation de la communication série
- ✓ Initialisation des ports numériques en mode « entrée » ou « sortie ».
- ✓ Affichage d'informations

▪ **Boucle sans fin (fonction Loop ()) :**

- ✓ Située après la fonction setup () dans le code Arduino.
- ✓ Elle s'exécute en boucle de façon continue tant que la carte est alimentée.
- ✓ Contient les instructions à répéter (lecture de capteurs, contrôle d'actionneurs, etc.).
- ✓ Permet au système de rester actif et réactif en temps réel.

```
1  const int ledPin = 13;
2  const int knockSensor = A0;
3  const int threshold = 100;
4
5  int sensorReading = 0;
6  int ledState = LOW;
7
8  void setup() {
9      pinMode(ledPin, OUTPUT);
10     Serial.begin(9600);
11 }
12
13 void loop() {
14     sensorReading = analogRead(knockSensor);
15
16     if (sensorReading >= threshold) {
17         ledState = !ledState;
18         digitalWrite(ledPin, ledState);
19         Serial.println("Knock!");
20     }
21     delay(100);
22 }
```

partie déclarative

partie initialisation

boucle sans fin

Figure 4.38 : Structure d'un programme Arduino

➤ Configuration d'ESP32 DEV KIT V1 avec l'environnement Arduino :

Après avoir installer Arduino IDE dans notre système :

1. Ouvre l'Arduino IDE. Va dans Fichier > Préférences. Dans le champ "URL de gestionnaire de cartes supplémentaires", colle ce lien :
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json , Clique sur OK pour enregistrer.
2. Va dans Outils > Type de carte > Gestionnaire de cartes... Dans la barre de recherche, tape ESP32. Installe le paquet nommé "esp32 by Espressif Systems".
3. Va dans Outils > Type de carte, choisis ESP32 Dev Module. Connecte ta carte ESP32 avec un câble USB. Va dans Outils > Port et choisis le port COM détecté (par ex : COM3).

4. Dans Outils, vérifie ou modifie ces paramètres : Board : ESP32 Dev Module Upload Speed : 115200 Flash Frequency : 80MHz Partition Scheme : Default 4MB with spiffs.

➤ **Processus de téléversement d'un programme sur une carte ESP32 :**

Pour pouvoir envoyer un programme vers la carte ESP32 par l'intermédiaire d'une liaison USB, il faut d'abord :

1. Créer un nouveau sketch ou bien ouvrir un ancien sketch dans l'IDE Arduino : ce dernier est basé sur le langage C au sens large, pouvant générer peu de code ;
2. Compiler le programme pour vérifier qu'il n'y a pas d'erreurs ou de messages d'erreur survenant en cas de problème de code compilé ;
3. En cas d'erreurs, corriger le code avant de continuer ;
4. Envoyer le programme dans la carte ESP32 en choisissant le bon port et le bon modèle ;
5. Effectuer les connexions du circuit électronique permettant de réaliser le projet ;
6. Attendre quelques secondes, le programme doit commencer à s'exécuter automatiquement sur la carte.
7. Alimenter la carte via le (les) port(s) USB, ou éventuellement une source autonome (piles 9V, Li-ion, etc.) ;
8. Vérifier que le système fonctionne bien comme devant, pour vérifier toutes les fonctions du montage.

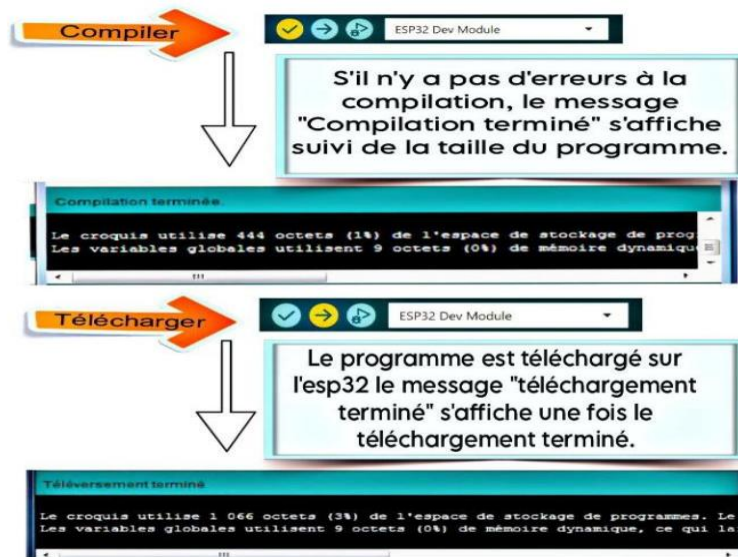


Figure 4.39 : Les étapes de téléchargement du code.

4.6.2.2. L'outil App Inventor :

Nous avons conçu trois interfaces (screen) dans MIT App Inventor.

La première interface est pour la connexion comme mesure de sécurité, permettant à l'utilisateur de s'authentifier en saisissant son nom d'utilisateur et son mot de passe avant d'accéder aux fonctionnalités du système.

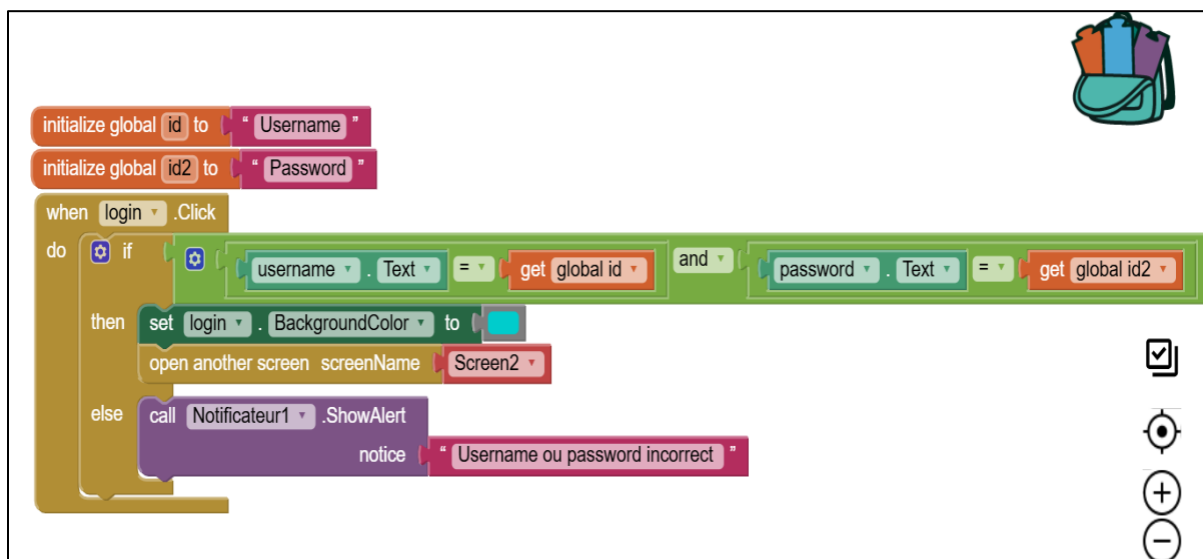


Figure 4.40 : Les blocs du premier écran de notre application.

Chapitre 4 : Conception et réalisation

La deuxième interface est dédiée aux différentes fonctionnalités de la maison intelligente, elle permet de contrôler la température et l'humidité, l'ouverture et la fermeture de fenêtre, comme elle intègre également un système de détection de gaz avec alarme en cas de fuite, et le niveau d'eau dans le réservoir.

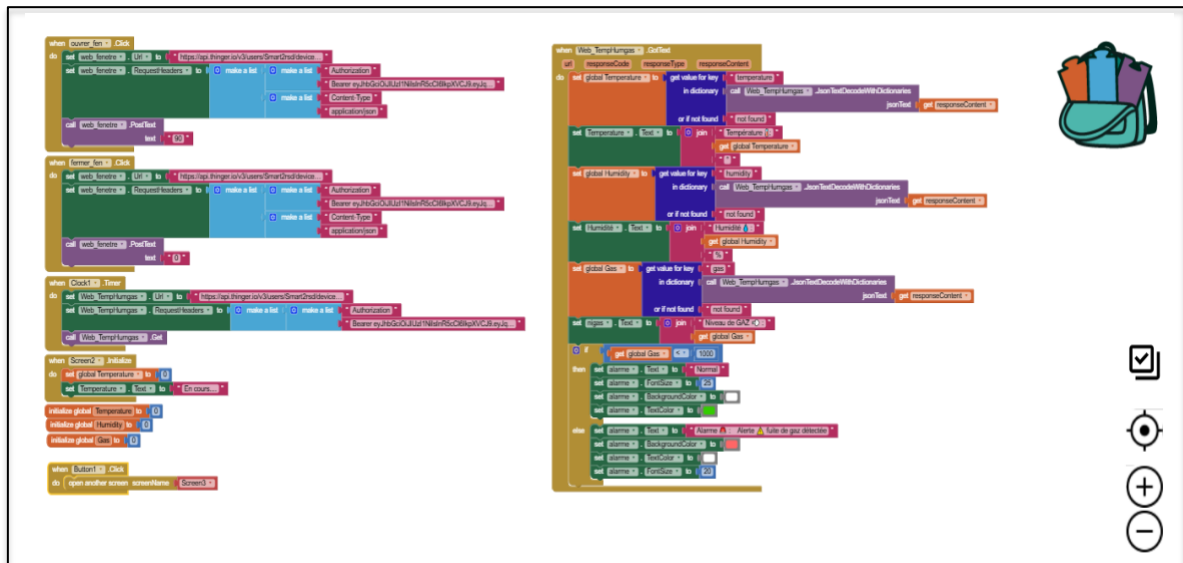


Figure 4.41 : Les blocs du deuxième écran de notre application.

La troisième interface est réservée pour la gestion d'éclairage ou l'utilisateur peut allumer ou éteindre les différentes lampes de sa maison (chambre, salon, cuisine).

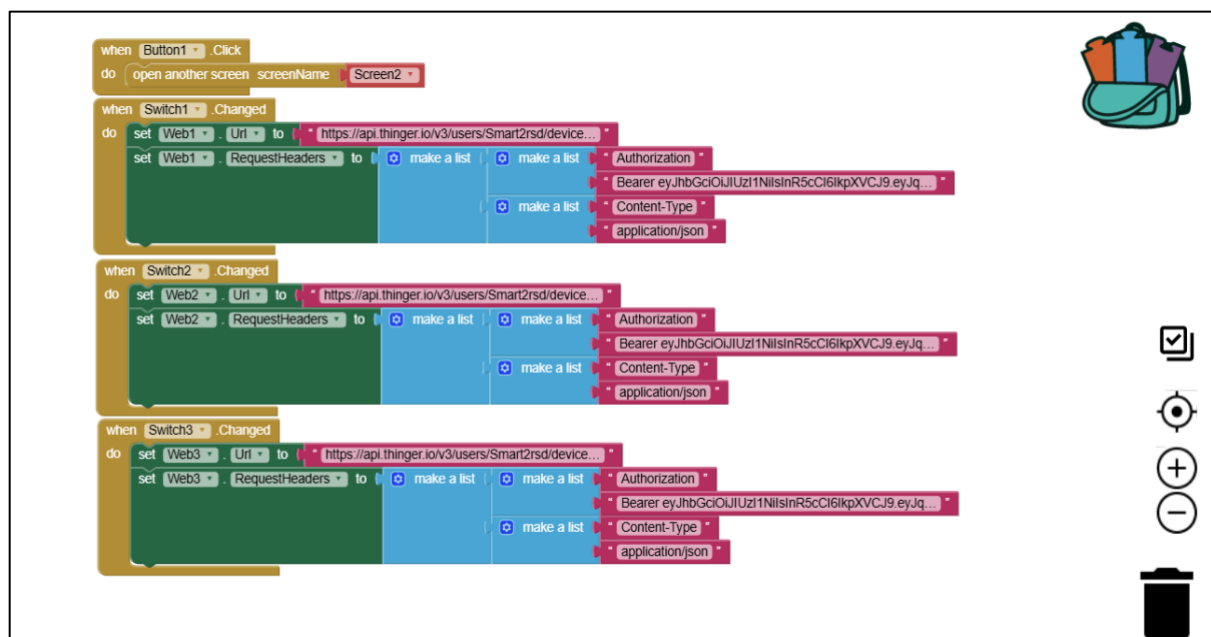


Figure 4.42 : Blocs du troisième écran de notre application.

4.6.2.3. Plateforme Thinger.io :

Notre plateforme Thinger.io contient deux appareils ESP32 et deux tableaux de bord.

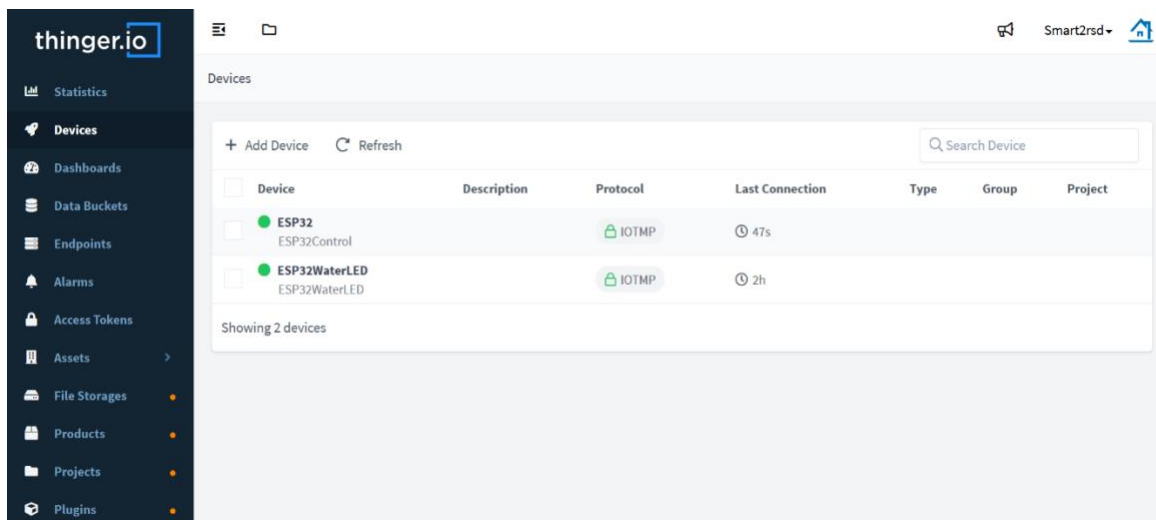


Figure 4.43 : Notre deux appareils ESP32 dans Thinger.io.

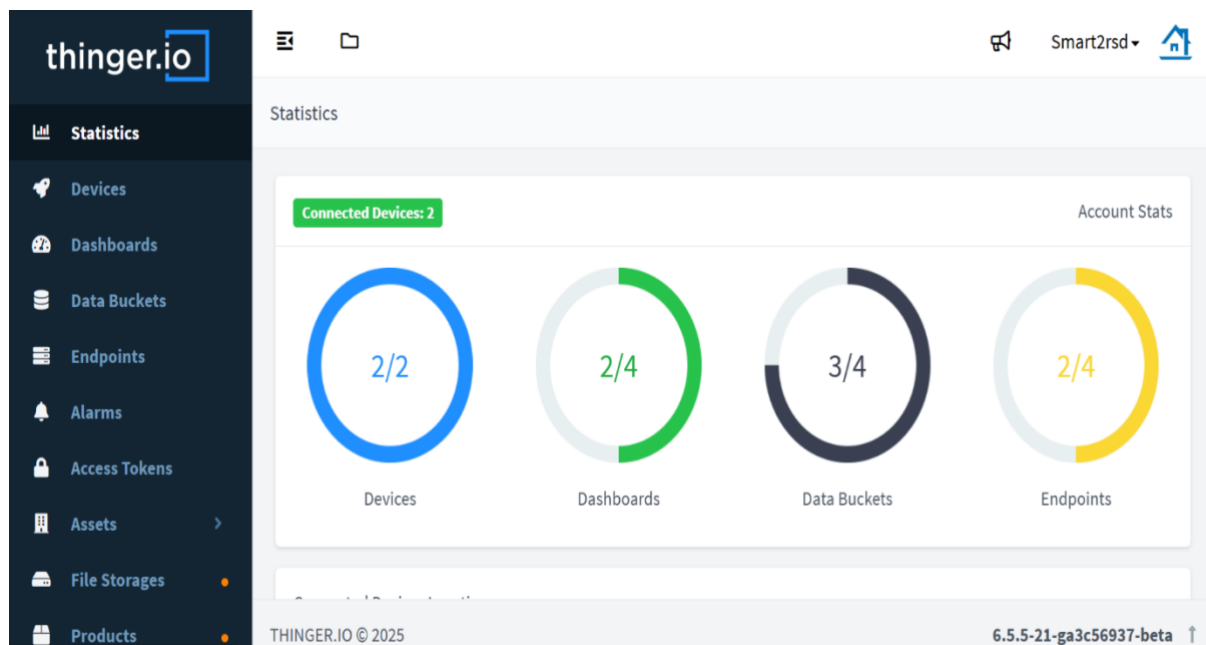


Figure 4.44 : Notre interface statistiques dans Thinger.io

Chapitre 4 : Conception et réalisation

Le premier ESP32 gère le contrôle de température et d'humidité, gestion des fenêtres, état de porte, contrôle de gaz avec une alarme si une fuite de gaz détecté.

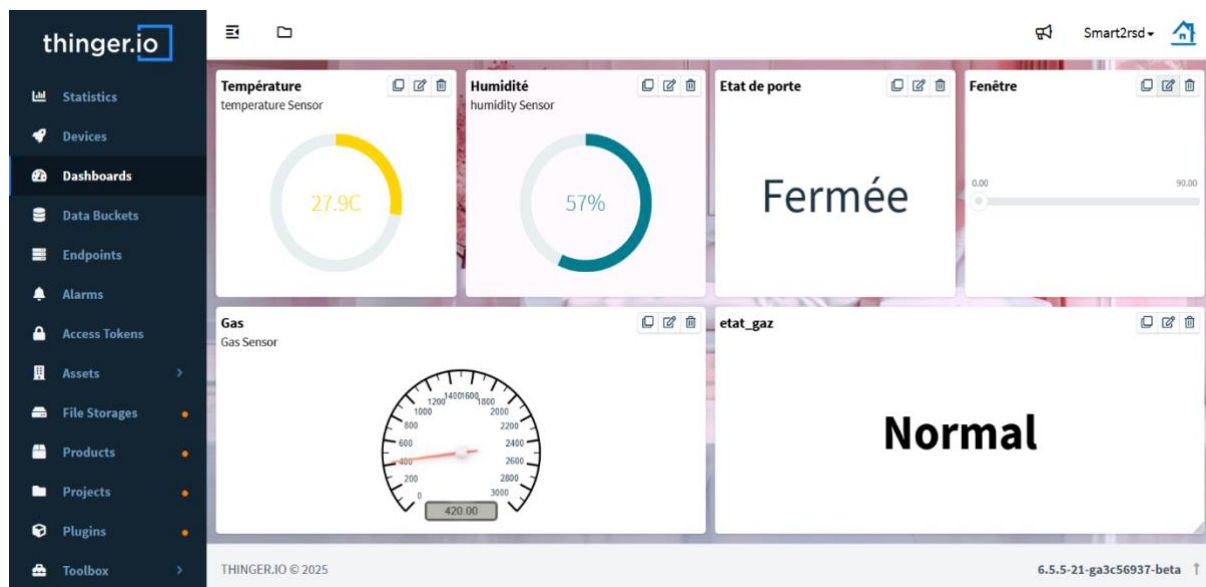


Figure 4.45 : Tableau de bord du premier ESP32 "ESP32Control"

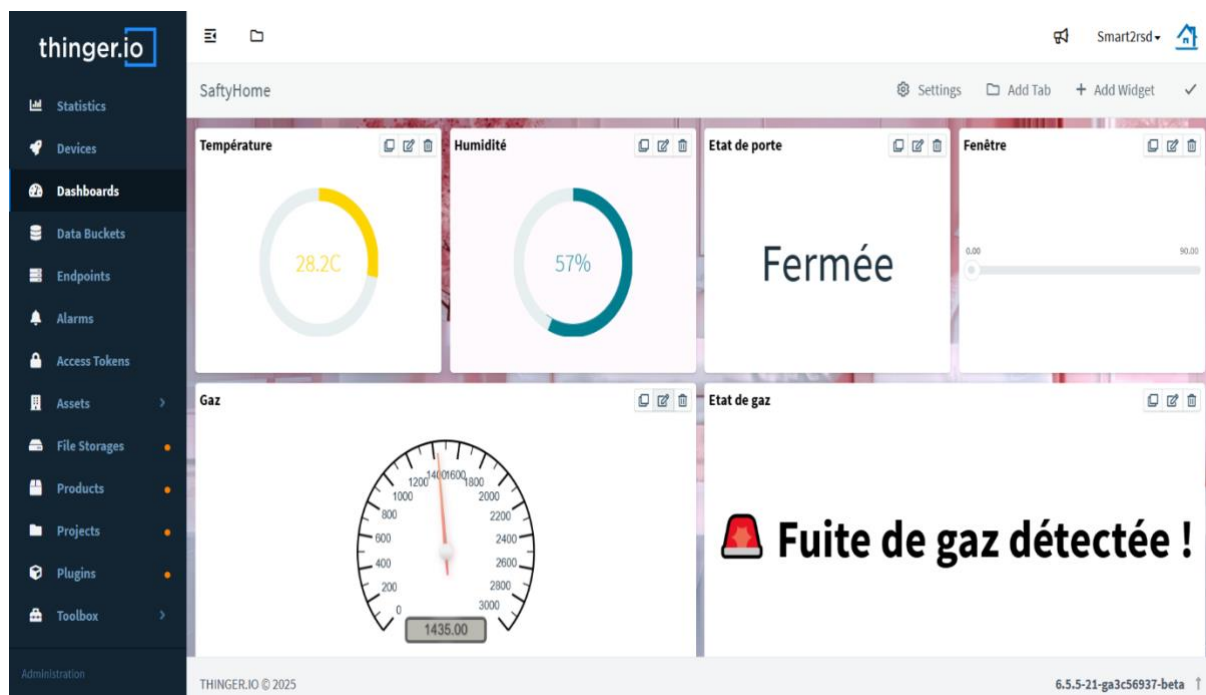


Figure 4.46 : Tableau de bord dans le cas ou une fuite de gaz est détectée

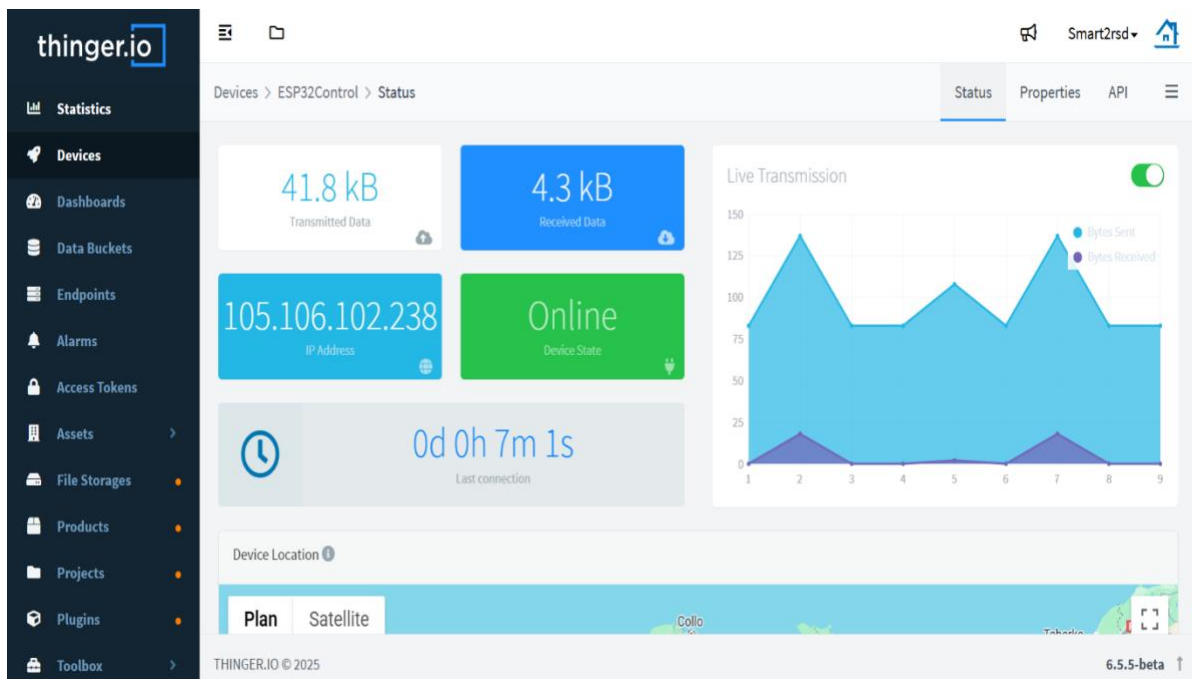


Figure 4.47 : Interface de suivi des données du premier ESP32 sur Thingier.io.

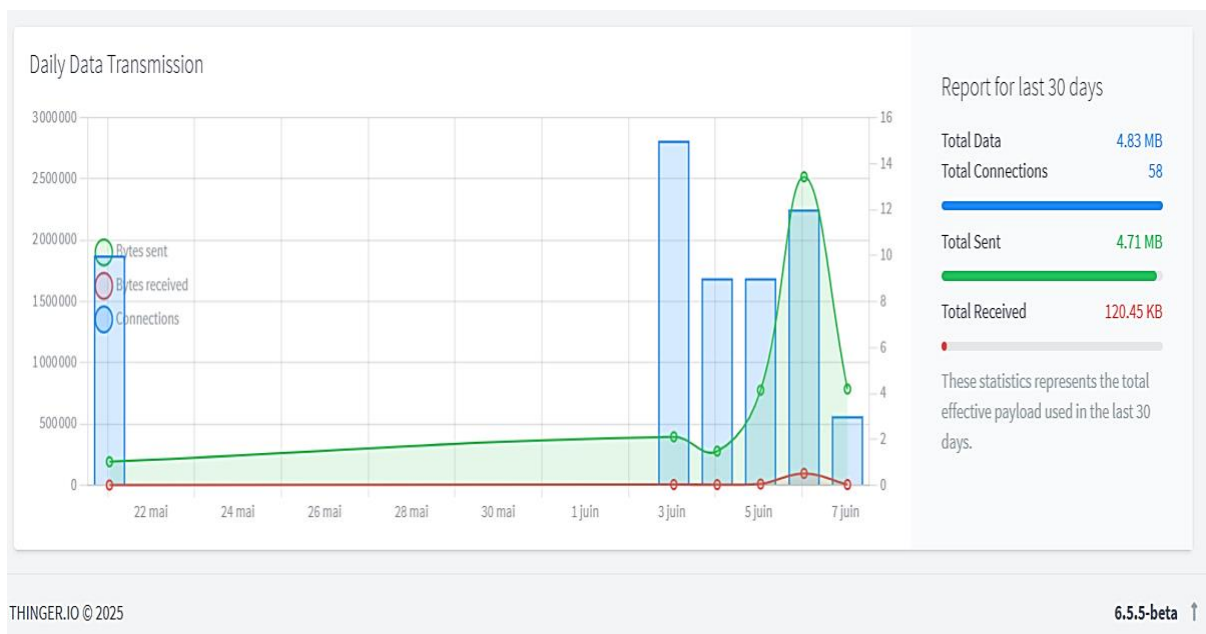


Figure 4.48 : Statistiques de transmission de données de l'ESP32 sur 30 jours à Thingier.io.

Le deuxième ESP32 est dédié pour la gestion des lampes et le contrôle de niveau d'eau a l'intérieure du réservoir.

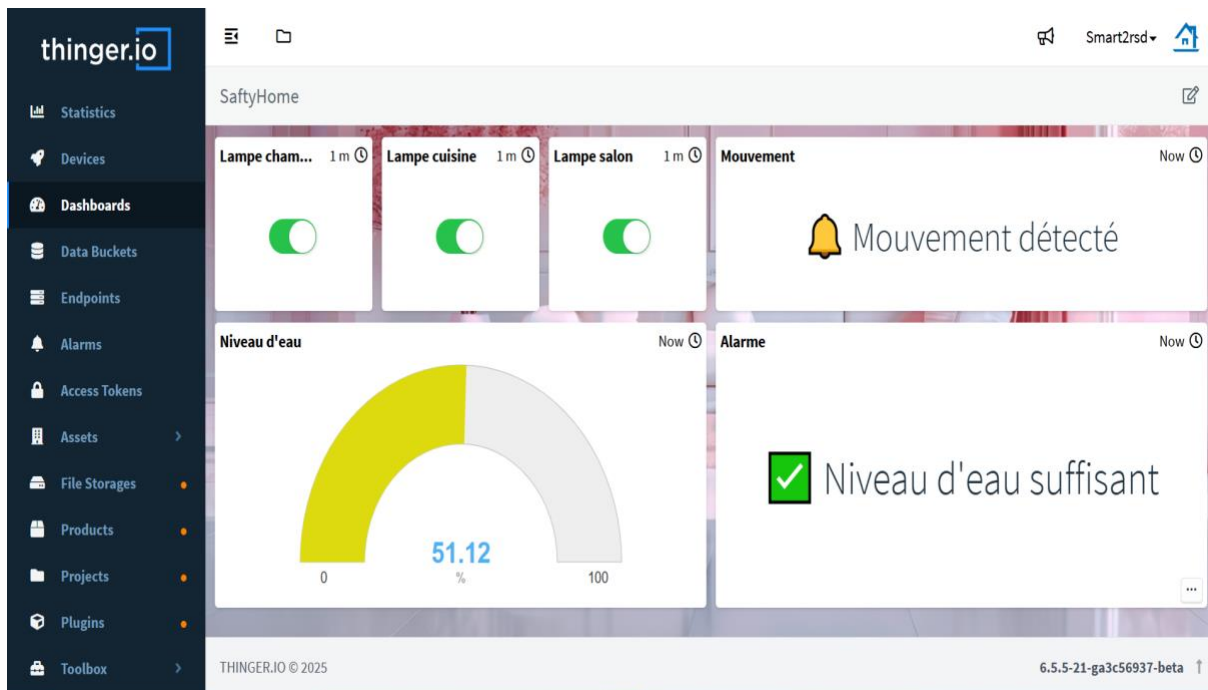


Figure 4.49 : Tableau de bord du deuxième ESP32 "ESP32WaterLED"



Figure 4.50 : Tableau de bord dans le cas ou le niveau d'eau inférieur à 25%.

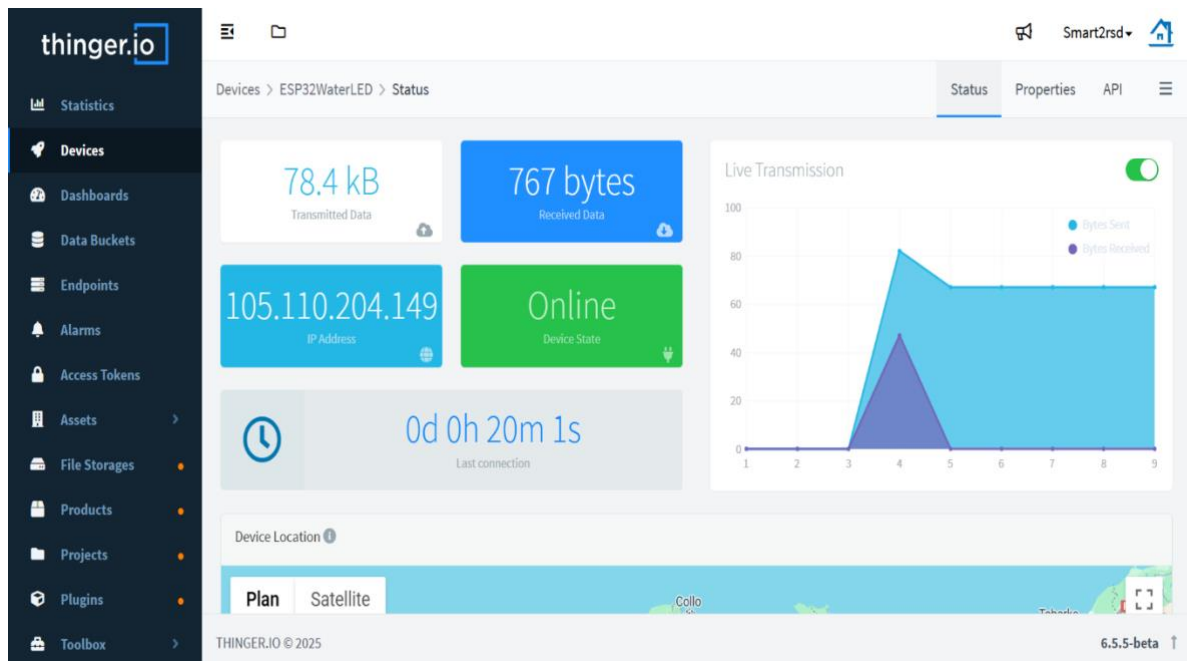


Figure 4.51 : Interface de suivi des données du deuxième ESP32 sur Thinger.io.

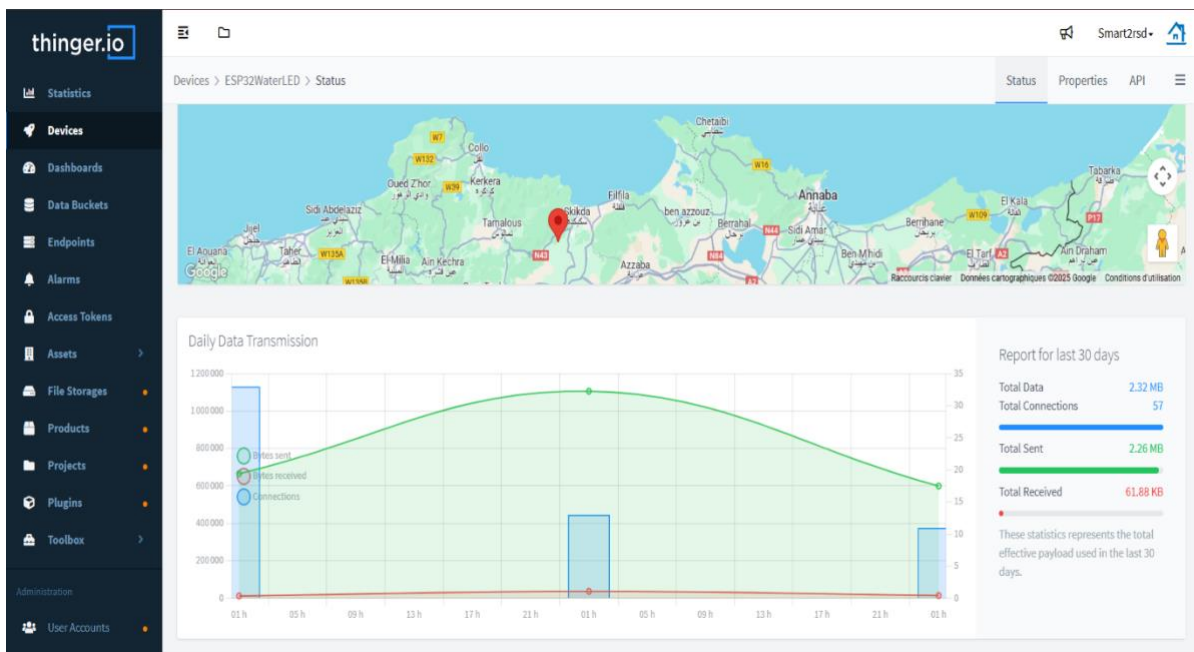


Figure 4.52 : Statistiques de transmission de données de l'ESP32 sur Thinger.io.

4.7. Fonctionnement du système :

➤ Authentification :

L'utilisateur entre leur nom utilisateur et leur mot de passe pour accéder à l'application

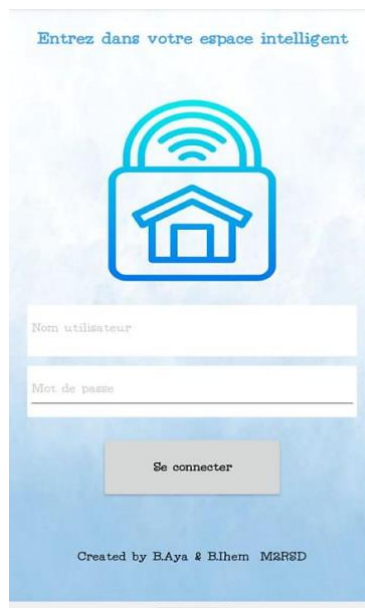


Figure 4.53 : Interface d'authentification utilisateur

Après l'authentification de l'utilisateur, l'application passe automatiquement à l'écran 2. Sur cet écran, l'utilisateur peut surveiller la température et l'humidité, contrôler l'ouverture et la fermeture de la fenêtre, ainsi que vérifier le niveau de gaz et le niveau d'eau du réservoir.



Figure 4.54 : L'interface principale de notre application.

Lorsqu'une fuite de gaz est détectée ou que le niveau d'eau du réservoir descend en dessous de 20 %, une alerte est automatiquement déclenchée pour avertir l'utilisateur.



Figure 4.55 : Alerte dans le cas où une fuite de gaz détectée et le niveau d'eau bas

En cliquant sur le bouton de gestion d'éclairage, l'utilisateur accède aux différentes lampes de la maison. Il peut aussi choisir d'allumer toutes les lampes en une seule action.

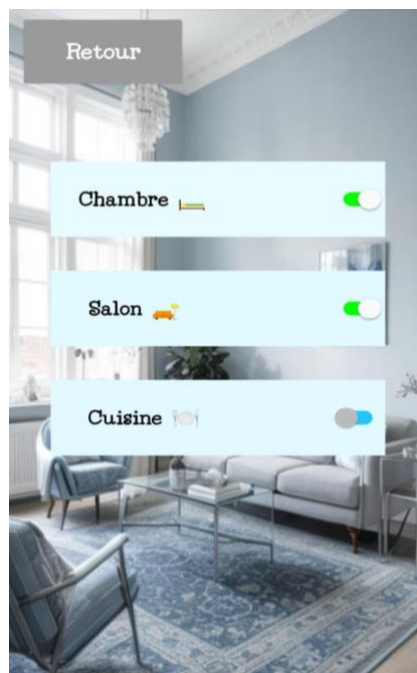


Figure 4.56 : Troisième écran de notre application.

4.8. Le prototype de la maison intelligente :

Avant de passer à la fabrication du prototype physique, nous avons réalisé une maquette 3D du projet en utilisant le logiciel Fusion 360 pour le design.



Figure 4.57 : Design de notre prototype.

À partir de ce design 3D, nous avons conçu notre modèle de maison intelligente. Les illustrations ci-dessous présentent le prototype tel qu'il existe en réalité.



Figure 4.58 : Vue réelle de notre maquette.

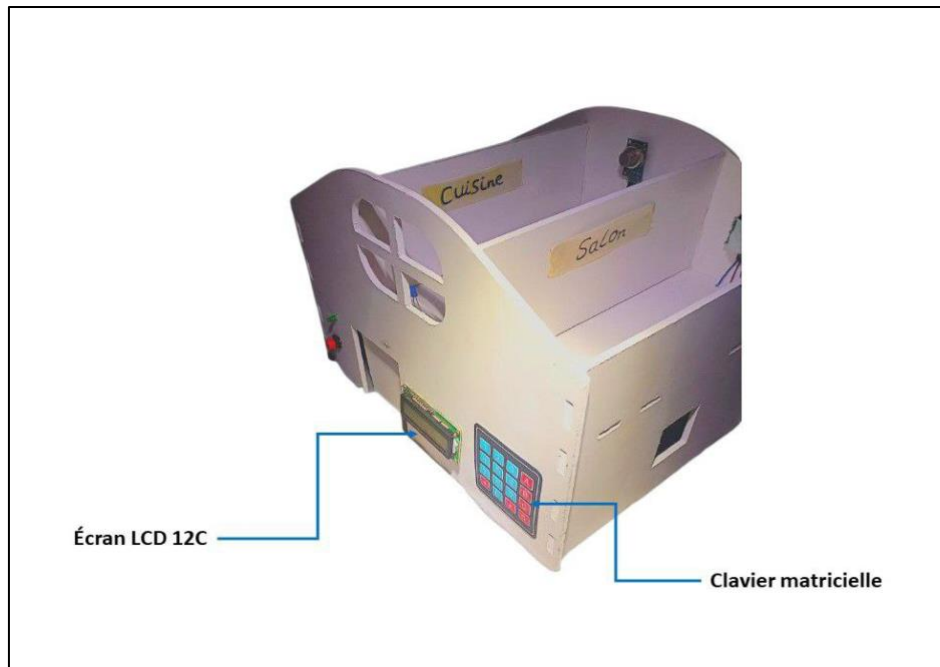


Figure 4.59 : Structure externe de la maison connectée (vue de côté).

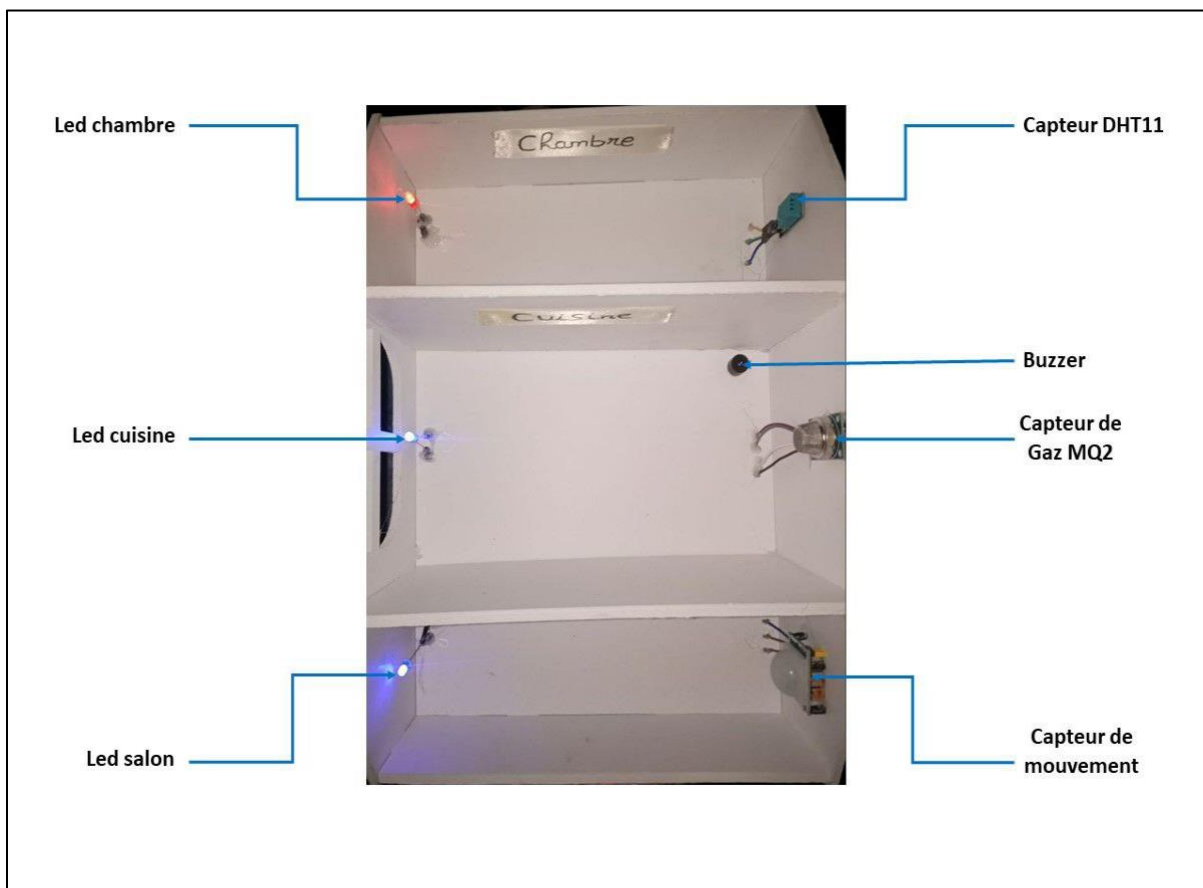


Figure 4.60 : Disposition des composants matériels dans la maquette.

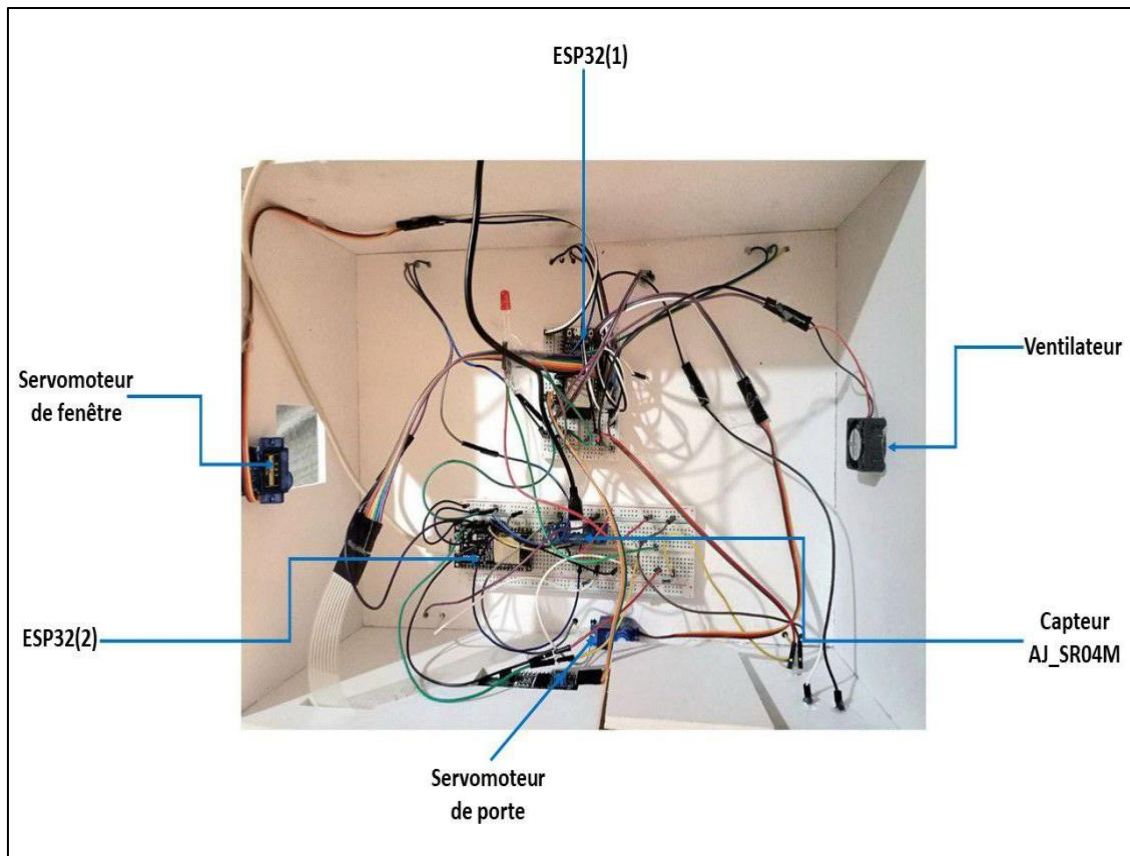


Figure 4.61 : Câblage des composants matériels situés sous la maquette.

4.9. Conclusion :

Dans ce chapitre, nous avons présenté la conception et la réalisation de notre système et les différents outils utilisés dans notre développement et la réalisation de ce travail.

Enfin, nous avons présenté les résultats obtenus.

Conclusion générale

Dans un contexte où la sécurité et le confort résidentiel deviennent des priorités majeures, les maisons intelligentes représentent une solution d'avenir. Dans ce mémoire, nous avons présenté SafetyHome, un système domotique conçu pour optimiser à la fois la sécurité et le bien-être des occupants. Ce système intègre plusieurs fonctionnalités clés : surveillance de la température et de l'humidité, gestion de l'éclairage et des fenêtres, contrôle du niveau d'eau dans un réservoir, détection de gaz, et mise en œuvre de protocoles de sécurité automatiques.

Pour piloter ce système, nous avons développé une application mobile intuitive à l'aide de la plateforme MIT App Inventor, permettant une interaction en temps réel avec les équipements connectés. L'application offre un accès simplifié à toutes les fonctionnalités, que ce soit depuis un smartphone Android ou via interface web.

Nous avons utilisé la plateforme Thinger.io non seulement comme base de données pour stocker les mesures et commandes en temps réel, mais aussi comme un outil central de supervision capable de gérer l'ensemble du système de manière efficace, via une interface accessible sur le web ou sur mobile Android.

Une maquette physique de la maison intelligente a été réalisée afin de valider le bon fonctionnement de l'ensemble.

Dans l'avenir, il sera envisageable d'intégrer des objets connectés, des équipements de santé connectés, des outils de reconnaissance vocale ou faciale dans le système. La gestion des comportements des usagers par l'intelligence artificielle permettra un fonctionnement du système « autonome », en fonction des besoins des utilisateurs. De tels apports pourraient offrir davantage de sûreté, de confort et de personnalisation au système au profit des utilisateurs.

En conclusion, nous espérons que ce mémoire pourra dégager un certain nombre de voies pour des travaux ultérieurs approfondis.

Bibliographie

[4] INNAL Rim Merabet Fatima el zohra, Réalisation d'un système antivol de voiture en Arduino, Mémoire de fin d'études En vue de l'obtention du diplôme de Master 2 en Informatique Option : Réseaux et Systèmes Distribués (RSD) /2020

[11] Mr BAYAT Mohammed Elaymene, Mr BAYAT Mohammed Elamine, Un système sécurisé pour les maisons intelligentes, *mémoire de master professionnel*. Spécialité : Administration et Sécurité des Réseaux

[13] B. Khoo, "Rfid as an enabler of the internet of things: Issues of security and privacy," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, pp. 709–712, IEEE, 2011.

[14] R. Chaloo, A. Oladeinde, N. Yilmazer, S. Ozcelik, and L. Chaloo, "An overview and assessment of wireless technologies and co-existence of zigbee, bluetooth and wi-fi devices," *Procedia Computer Science*, vol. 12, pp. 386–391, 2012.

[15] J. T. Vainio *et al.*, "Bluetooth security," in *Proceedings of Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Seminar on Internetworking: Ad Hoc Networking, Spring*, vol. 5, 2000.

[16] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, "Study on zigbee technology," in *2011 3rd International Conference on Electronics Computer Technology*, vol. 6, pp. 297–301, IEEE, 2011.

[17] A. Zourmand, A. L. K. Hing, C. W. Hung, and M. AbdulRehman, "Internet of things (iot) using lora technology," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 324–330, IEEE, 2019.

[18] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Rfc2616: Hypertext transfer protocol–http/1.1," 1999.

Bibliographie

[20] Y. Chen and T. Kunz, "Performance evaluation of iot protocols under a constrained wireless access network," in *2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*, pp. 1–7, IEEE, 2016.

[30] Inguel Malha, Conception et réalisation d'un système Domotique par GSM "Mémoire De Fin D'étude De MASTER ACADÉMIQUE" (Université Mouloud Mammeri de Tizi-Ouzou, 2017)

Références

[1] <https://www.mmatic.fr/lhistoire-de-la-domotique-de-la-fiction-a-la-realite/><https://www.lemagdeladomotique.com/dossier-1-domotique-definition-applications.html>

[2] <https://www.mmatic.fr/lhistoire-de-la-domotique-de-la-fiction-a-la-realite/>

[3] <https://domegy.be/on-en-parle/evolution-domotique-de-1980-a-nos-jours/>

[5] <https://www.tafsquare.com/fr/blog/les-domaines-d-application-de-la-domotique/>

[6] <https://annecy.lamaisondestravaux.com/renovation-interieur/installation-electrique-maison/info-conseils/la-domotique-avantages-et-inconvenients>

[7] <https://www.sap.com/suisse/products/artificial-intelligence/what-is-iot.html>

[8] <https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/>

[9] <https://itsocial.fr/contenus/tribunes/10-principes-cles-definissent-vague-iot-objets-connectes/>

[10] <https://blog.cellenza.com/data/introduction-a-liot-comment-fonctionne-linternet-des-objets/>

[12] <https://www.m2m.fr/iot/architecture-iot/>

[19] "waytolearnx." <https://waytolearnx.com/2019/06/protocole-http.html>. Accessed: 2022-06-2.

Bibliographie

- [21] <https://embedded-computing.pages.forge.hefr.ch/2023-il-embedded-computing/Lecture/Communication/coap/>
 - [22] <https://www.uptrends.fr/qu-est-ce-que/rest-api>
 - [23] <https://www.makeo.fr/lexique/android>
 - [24] <https://igm.univ-mlv.fr/~dr/XPOSE2008/android/>
 - [25] <https://blog.lesjeudis.com/actualite-des-systemes-d-exploitation-mobile>
 - [26] <https://www.blogdumoderateur.com/tools/android/>
 - [27] <https://w3r.one/fr/blog/mobile/android/architecture-android/comprendre-architecture-android-vie-ensemble-composants-modeles>
 - [28] <https://www.spiceworks.com/tech/tech-general/articles/android-os/>
 - [29] <https://info.blaisepascal.fr/langages/appinventor/>
 - [31] <https://gomycod.com/dz/fr/la-boite-a-outils-app-inventor/>
 - [32] <https://blogpeda.ac-poitiers.fr/lp2i-si/2013/01/23/developper-des-applications-android-avec-app-inventor/>
 - [33] <https://thinger-io.fr.softonic.com/android>
 - [34] <https://docs.thinger.io/>
 - [35] <https://sauveee.com/produit/carte-de-developpement-esp-32-dev-kit-v1/>
 - [36] <https://powertech-dz.net/products/single/mq-2-capteur-de-gaz-de-fumee-gpl-butane-vente-composants-electronique-blida-algerie-144>
 - [37] <https://www.futura-sciences.com/maison/definitions/maison-detecteur-mouvement-10637/>
 - [38] <https://www.robotique.tech/tutoriel/mesurer-la-temperature-avec-la-carte-esp32-et-un-capteur-dht11/>
-

Bibliographie

[39] <https://boutique.semageek.com/fr/104-micro-servo-tower-pro-sg90-3007447379574.html>

[40] https://www.robotique.tech/tutoriel/buzzer/#google_vignette

[41] <https://www.robotique.site/tutoriel/la-carte-esp32-et-lafficheur-lcd-i2c-1602/>

[42] <https://www.robotique.tech/tutoriel/utilisation-du-clavier-matriciel-4x4-avec-la-carte-esp32/>

[43] <https://fr.cdoepushbutton.com/news/push-button-switch-symbols-explained/>

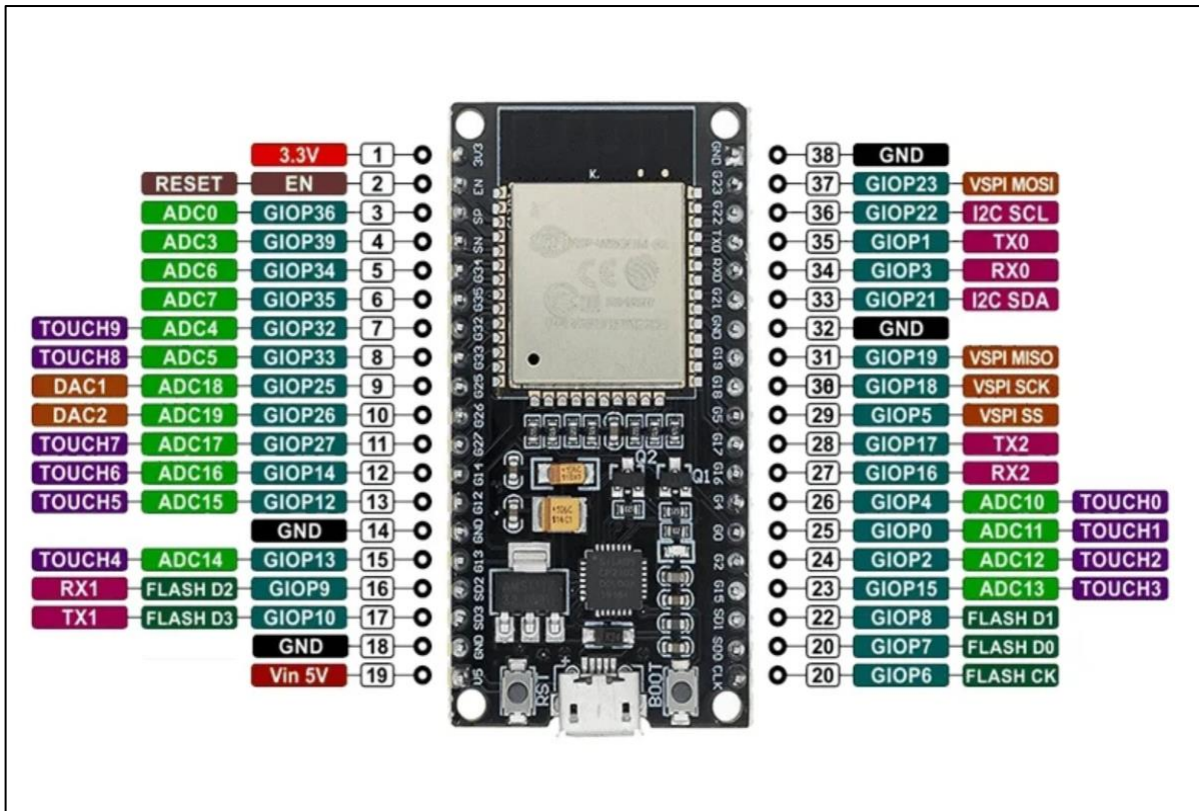
[44] <https://arduinofactory.fr/la-led-arduino/>

[45] <https://www.electronique-kit.com/les-r%C3%A9sistances-arduino>

[46] <https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/>

[47] <https://ledisrupteurdimensionnel.com/arduino/plaque-dessai-ou-breadboard-protoboard/>

Annexe



Nous avons développé le code suivant pour le premier ESP32 :

```
#define DHTPIN 17

#define DHTTYPE DHT11

#define GAS_PIN 35

#define SERVO_PIN 13

#define WINDOW_SERVO_PIN 5

#define BUZZER_PIN 23

#define FAN_PIN 16
```

```
DHT dht(DHTPIN, DHTTYPE);

Servo servo;

Servo windowServo;

LCD_I2C lcd(0x27, 16, 2);

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

  { '1', '2', '3', 'A' },

  { '4', '5', '6', 'B' },

  { '7', '8', '9', 'C' },

  { '*', '0', '#', 'D' }

};

byte rowPins[ROWS] = { 14, 27, 26, 25 };

byte colPins[COLS] = { 33, 32, 18, 19 };

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

String password = "1234";

String input = "";

String etat_porte = "Fermée";

String etat_gaz = "Normal";

TingerESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

void sendTelegramMessage(String message) {

  if (WiFi.status() == WL_CONNECTED) {

    HTTPClient http;

    String url = "https://api.telegram.org/bot" + String(TELEGRAM_BOT_TOKEN) +
```

```
"/sendMessage?chat_id=" + String(TELEGRAM_CHAT_ID) + "&text=" + message;

    http.begin(url);

    http.GET();

    http.end();

}

}

void setup() {

    Serial.begin(115200);

    dht.begin();

    lcd.begin();

    lcd.backlight();

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Enter Password:");

    pinMode(BUZZER_PIN, OUTPUT);

    pinMode(FAN_PIN, OUTPUT);

    digitalWrite(BUZZER_PIN, LOW);

    digitalWrite(FAN_PIN, LOW);

    servo.attach(SERVO_PIN);

    servo.write(0);

    windowServo.attach(WINDOW_SERVO_PIN);

    windowServo.write(0);

    thing.add_wifi(SSID, PASSWORD);

    thing["servo"]["pos"] << [(pson& in) {
```

```
    windowServo.write((int)in);

};

thing["etat_porte"] >> [](pson& out) {

    out = etat_porte;

};

thing["etat_gaz"] >> [](pson& out) {

    out = etat_gaz;

};

// Nuevo recurso con validación
thing["sensors_env"] >> [](pson& out) {

    float temp = dht.readTemperature();

    float hum = dht.readHumidity();

    int gas = analogRead(GAS_PIN);

    if (isnan(temp) || isnan(hum)) {

        out["error"] = "DHT error";

    } else {

        out["temperature"] = temp;

        out["humidity"] = hum;

        out["gas"] = gas;

    }

};

}

void loop() {

    thing.handle();

}
```

```
int gasLevel = analogRead(GAS_PIN);

float temp = dht.readTemperature();

float humidity = dht.readHumidity();

if (gasLevel >= 1000) {

    etat_gaz = "🚒 Fuite de gaz détectée !";

    digitalWrite(BUZZER_PIN, HIGH);

    digitalWrite(FAN_PIN, HIGH);

    if (!gasAlertSent) {

        sendTelegramMessage("🚒 Fuite de gaz !");

        gasAlertSent = true;

    }

} else {

    etat_gaz = "Normal";

    digitalWrite(BUZZER_PIN, LOW);

    digitalWrite(FAN_PIN, LOW);

    gasAlertSent = false;

}

if (temp > 40.0) {

    digitalWrite(FAN_PIN, HIGH);

} else if (gasLevel < 1000) {

    digitalWrite(FAN_PIN, LOW);

}

if (humidity > 90.0) {
```

```
    windowServo.write(90);

} else if (humidity < 80.0) {

    windowServo.write(0);

}

char key = keypad.getKey();

if (key) {

    if (key == '#') {

        if (input == password) {

            lcd.clear();

            lcd.print("Access Granted");

            servo.write(90);

            etat_porte = "Ouverte";

            delay(5000);

            servo.write(0);

            etat_porte = "Fermée";

        } else {

            lcd.clear();

            lcd.print("Wrong Password");

        }

        input = "";

        delay(2000);

        lcd.clear();

        lcd.print("Enter Password:");

    } else if (key == '*') {
```

```
    input = "";  
  
    lcd.clear();  
  
    lcd.print("Cleared");  
  
    delay(500);  
  
    lcd.clear();  
  
    lcd.print("Enter Password:");  
  
} else {  
  
    input += key;  
  
    lcd.clear();  
  
    lcd.print("Entered: ");  
  
    lcd.print(input);  
  
}  
  
}  
  
delay(200);  
  
}
```

Annexe

Pour le deuxième ESP32 nous avons développé le code suivant :

```
#define LED_CHAMBRE 33

#define LED_SALON 32

#define LED_CUISINE 25

#define LED_ALARME 13

#define TRIG_PIN 27

#define ECHO_PIN 26

#define PIR_PIN 12

#define TANK_HEIGHT_CM 30.0

#define TANK_FULL_DISTANCE_CM 20.0

ThingyESP32 thingy(USER_NAME, DEVICE_ID, DEVICE_CREDENTIAL);

bool etatChambre = false;

bool etatSalon = false;

bool etatCuisine = false;

bool etatAlarme = false;

String etatMouvement = "Pas de mouvement";

void setup() {

  Serial.begin(115200);

  pinMode(LED_CHAMBRE, OUTPUT);

  pinMode(LED_SALON, OUTPUT);

  pinMode(LED_CUISINE, OUTPUT);

  pinMode(LED_ALARME, OUTPUT);

  pinMode(TRIG_PIN, OUTPUT);

  pinMode(ECHO_PIN, INPUT);
```

```
pinMode(PIR_PIN, INPUT);

digitalWrite(LED_ALARME, LOW);

thing.add_wifi(SSID, PASSWORD);

thing["led_chambre"] = [(pson &in, pson &out) {

  if (in.is_empty()) out = etatChambre;

  else {

    etatChambre = in;

    digitalWrite(LED_CHAMBRE, etatChambre);

  }

}];

thing["led_salon"] = [(pson &in, pson &out) {

  if (in.is_empty()) out = etatSalon;

  else {

    etatSalon = in;

    digitalWrite(LED_SALON, etatSalon);

  }

}];

thing["led_cuisine"] = [(pson &in, pson &out) {

  if (in.is_empty()) out = etatCuisine;

  else {

    etatCuisine = in;

    digitalWrite(LED_CUISINE, etatCuisine);

  }

}];
```

```
thing["niveau_eau"] >> [](pson &out) {

    float distance = lireDistance();

    float hauteur_max = TANK_HEIGHT_CM - TANK_FULL_DISTANCE_CM;

    float hauteur_eau = TANK_HEIGHT_CM - distance;

    float pourcentage = constrain((hauteur_eau / hauteur_max) * 100.0, 0, 100);

    out["pourcentage"] = pourcentage;

    if (pourcentage < 25.0) {

        digitalWrite(LED_ALARME, HIGH);

        etatAlarme = true;

        out["etat"] = " ⚠ Niveau d'eau bas";

    } else {

        digitalWrite(LED_ALARME, LOW);

        etatAlarme = false;

        out["etat"] = " ✅ Niveau d'eau suffisant";

    }

    Serial.print("Distance: "); Serial.print(distance); Serial.print(" cm | ");

    Serial.print("Hauteur eau: "); Serial.print(hauteur_eau); Serial.print(" cm | ");

    Serial.print("Pourcentage: "); Serial.print(pourcentage); Serial.println(" %");

};

thing["alarme"] >> [](pson &out) {

    out = etatAlarme;

};

thing["mouvement"] >> [](pson &out) {

    out = etatMouvement;

};
```

```
}

void loop() {

  thing.handle();

  Serial.println(etatMouvement);

  if (digitalRead(PIR_PIN) == HIGH) {

    digitalWrite(LED_SALON, HIGH);

    etatMouvement = "🔔 Mouvement détecté";

  } else {

    if (!etatSalon) {

      digitalWrite(LED_SALON, LOW);

    }

    etatMouvement = "✅ Pas de mouvement";

  }

}

float lireDistance() {

  digitalWrite(TRIG_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);

  float distance = duration * 0.034 / 2.0;

  return distance; }
```