

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة عمّار ثليجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كلية العلوم
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE



Mémoire de MASTER

Domaine:Mathématiques et Informatique

Filière:Informatiques

Option:Systems d'Information et de Décision

Par: Nouioua Siham

Bait Marwa

THEME:

Création d'un réseau d' IOT

Soutenu publiquement le 27-06-2023 devant le jury composé de:

Dr sahlaoui atika	MC(A)	Président
Dr oulad djedid lakhdar	MC(A)	Examineur
Dr guibadj fatna	MC(A)	Examineur
Dr hamini narjes	MC(A)	Encadreur

Remerciements

Nous tenons à remercier mes conseillers, le Dr Hamini Nargis. Nous la remercions de nous guider, de nous aider et de nous donner des conseils.

Nous remercions sincèrement tous les professeurs et tous ceux qui ont guidé notre réflexion par leurs paroles, écrits, conseils et critiques, et qui ont accepté de nous rencontrer lors de nos recherches et de répondre à nos questions.

Enfin, merci aux amis qui nous ont toujours accompagnés. Leur soutien inconditionnel et leurs encouragements nous ont beaucoup aidés. Nos remerciements, notre respect et notre gratitude à tous.

Dédicace

Tout d'abord, je remercie ALLAH qui m'aime et m'a donnée la patience et le courage durant ces longues années d'étude, de m'avoir donné la force et le courage de mener à bien ce modeste travail.

Je tiens à dédier cet humble travail à :

A ma tendre mère H.Dikeche et mon très cher père M.nouioua.

A mon oncle Dikeche Ibrahim.

A mes sœurs et mes frères.

A mon binôme .

Et tout les professeur, je vous remercie tous.

Tout ceux qui m'aiment et que j'aime.

Dédicase

A mes très chers parents , source de vie,
d'amour et d'affection

A mes chers freres et leurs enfants,source de
joie et de bonheur

A toute ma famille,source d'espoir et de
motivation

A tous mes amis, chere amie avant d'être binome
A vous cher lecteur

Résumé

L'objectif de ce travail, est de présenter une étude sur la conception et la mise en œuvre de réseaux IoT pour automatiser et connecter des appareils domestiques afin de les rendre intelligents. Cette étude explore l'architecture des réseaux IoT, les technologies et les protocoles utilisés. Il se concentre sur les avantages de la maison intelligente et les applications pratiques. Cette recherche fournit une base solide pour ceux qui s'intéressent à la conception de maisons intelligentes et fournit une compréhension approfondie des défis et des opportunités posés par cette nouvelle technologie.

abstract

The objective of this work, is to present a study on the design and implementation of IoT networks to automate and connect home devices to make them smart. This study explores the architecture of IoT networks and the technologies and protocols used. It focuses on smart home benefits and practical applications. This research provides a solid foundation for those interested in smart home design and provides an in-depth understanding of the challenges and opportunities posed by this new technology.

ملخص

الهدف من هذا العمل، هو تقديم دراسة حول تصميم وتنفيذ شبكات إنترنت الأشياء لأتمتة وتوصيل الأجهزة المنزلية لجعلها ذكية. تستكشف هذه الدراسة بنية شبكات إنترنت الأشياء والتقنيات والبروتوكولات المستخدمة. يركز على فوائد المنزل الذكي والتطبيقات العملية. يوفر هذا البحث أساسًا متينًا للمهتمين بتصميم المنزل الذكي ويوفر فهمًا عميقًا للتحديات والفرص التي تطرحها هذه التكنولوجيا الجديدة.

Table de matiere

Table de matiere	5
Liste des figures	6
Liste des tableaux	8
1 Introduction	9
2 Concepts de base	11
2.1 Internet des objets	11
2.1.1 Définition	11
2.1.2 Les objets connecté	11
2.2 Architecture d'un réseaux IOT [1]:	11
2.3 Caractéristiques d'Internet des objets	12
2.4 Domaines d'application de l'internet des objets	13
2.4.1 La domotique	13
2.4.2 La santé	13
2.4.3 Les transports	14
2.5 Le protocoles de communication	14
3 Conception et Simulation	19
3.1 Analyse et Conception	19
3.1.1 Le langage UML	19
3.1.2 Le diagramme Cas d'utilisation	19
3.1.3 Diagramme de séquence	21
3.1.4 Diagramme d'activité	25
3.2 Simulation	28
3.2.1 Présentation de Tinkercad	28
3.2.2 Présentation de notre projet sur Tinkercad	28
3.2.3 Présentation de Cisco Packet Tracer	34
3.2.4 Présentation de notre projet sur Cisco Packet Tracer	34
4 Réalisation et test	43
4.1 Environnement de développement	43
4.1.1 Environnement matériel :	43
4.1.2 Environnement logiciel :	49
4.2 Réalisation :	52
5 conclusion	55
Bibliographie	56

Liste des figures

1	Architecture de l'IOT [1].	11
2	La domotique [2].	13
3	Publication MQTT [3].	15
4	Abonnement MQTT [3].	16
5	Quality Of Service (QOS) [3].	17
6	séquence d'échanges [3].	18
7	Diagramme de cas d'utilisation global.	20
8	Diagramme de cas d'utilisation global.	21
9	Diagramme de séquence ajouter un nouveau périphérique IoT.	22
10	Diagramme de séquence contrôler un périphérique IoT.	23
11	Diagramme de séquence créer un scénario .	24
12	Diagramme d'activité pour ajouter un périphérique IoT.	25
13	Diagramme d'activité pour contrôler un périphérique IoT.	26
14	Diagramme d'activité pour créer un scénario automatisé pour ce périphérique IoT.	27
15	Tinkercad.	28
16	PIR capteur.	29
17	Serial monitor.	29
18	Serial monitor.	29
19	Capteur d'humidité .	30
20	Serial monitor.	30
21	Serial monitor.	30
22	Porte verrouillée	31
23	Porte verrouillée	31
24	Porte verrouillée	32
25	Capteur de gaz.	32
26	Capteur de gaz.	33
27	Capteur de gaz.	33
28	représentation de Cisco Packet Tracer 8.	34
29	l'ouverture de la porte à l'aide de RFID	35
30	Programmation de la carte MCU.	36
31	Détection de mouvement	37
32	Programmation de la carte SBC.	38
33	la carte SBC.	38
34	Détection de gaz	39
35	Connexion des peripheries iot au réseau sans fil de la passerelle et activation du DHCP.	39
36	IoT Server.	40
37	smartphone.	40
38	IoT Server.	41
39	les conditions pour le scénario détection de CO2.	41
40	L'accès externe de la maison intelligente.	42
41	NodeMCU ESP8266 [4].	43
42	ESP8266 [5]	44
43	Arduino Uno [6].	45
44	Le capteur de mouvement [7].	46
45	Le capteur de gaz MQ2 [8].	46
46	Capteur d'humidité-Température DHT22 [9].	47
47	Capteur De Lumière LDR [10].	47
48	relais [11].	48
49	Un servomoteur [12].	48
50	clavier 3x4 [13]	49
51	Arduino ide	50
52	Arduino ide	50
53	programme platformIO.	52
54	test une LED.	53
55	Programme yaml.	53

56 Programme yaml. 54

Liste des tableaux

1	Comparaison des technologies sans fil et de leurs cas d'utilisation idéaux	12
2	Comparaison des protocoles de communication pour l'Internet des Objets	14
3	MQTT et modèle OSI	17
4	Format des messages	17
5	Equipements et Localisation dans Packet Tracer	42

1 Introduction

La création d'un réseau d'objets connectés (IoT) revêt une importance capitale en raison de sa capacité à connecter et à interagir avec des appareils et des systèmes, ouvrant ainsi de nouvelles perspectives en matière de collecte de données, d'automatisation et d'efficacité. Ce réseau facilite l'échange d'informations en temps réel, favorise l'innovation et offre des opportunités pour améliorer notre qualité de vie.

Nous avons choisi ce sujet en raison de son potentiel d'innovation et de son impact positif sur divers domaines tels que la santé, l'agriculture, les transports, l'énergie, etc. aussi la création d'un réseau d'objets connectés (IoT) est actuellement une tendance majeure dans le domaine de la technologie et de l'innovation, on est intéressé par ce domaine due a ces dernieres raisons et on aimeront bien travaillé sur ce sujet.

Le problème que nous avons rencontré est la capacité réduite des cartes électroniques et la sécurisation du réseau. Nous avons donc choisi de réaliser un réseau d'IoT basé sur le protocole MQTT et géré par une application réseau. MQTT est un protocole de messagerie léger, ouvert et basé sur le modèle de publication/abonnement (publish/subscribe).

Les IOT sont des ensembles de capteurs et d'actionneurs contrôlés par une carte électronique Wifi de type Nodemcu.

Dans la suite de ce chapitre, nous détaillons tous d'abord la problématique, ensuite, nous citons le contenu de ce document.

Problématique :

La problématique de création d'un réseau d'objets connectés (IoT) repose sur :

- Sécurité : Comment garantir la confidentialité, l'intégrité et la disponibilité des données échangées entre les appareils connectés et la gestion du réseau ? Quelles mesures de sécurité doivent être mises en place pour se prémunir contre les attaques potentielles sur le réseau domestique ?
- Évolutivité : Comment concevoir un réseau d'IoT qui peut s'adapter à l'ajout de nouveaux appareils intelligents au fil du temps ? Quelles sont les meilleures pratiques pour assurer la flexibilité et la scalabilité du réseau afin de répondre aux besoins futurs des utilisateurs ?
- Interopérabilité : Comment permettre une communication transparente entre les différents dispositifs connectés provenant de différents fabricants et utilisant différentes technologies de communication ?
Comment surmonter les problèmes d'incompatibilité pour assurer une interopérabilité maximale ?
- Gestion du réseau : Comment développer des mécanismes de gestion efficaces pour surveiller et contrôler le réseau d'IoT domestique ? Comment fournir des interfaces utilisateur conviviales pour permettre aux utilisateurs de gérer facilement leurs appareils connectés et d'effectuer des actions de contrôle ?

Contenu :

D'abord, nous commencerons par définir, et détailler les aspects généraux du monde L'Internet des Objets (IoT), les protocoles de communication en particulier le protocole MQTT. Ensuite, nous allons avoir l'architecture d'un réseau d'iot et les caractéristiques d'iotics, et certains domaines d'application (la santé, le transport, la domotique...).

Le second chapitre a été consacré à la domotique, on a l'analyse et la conception de l'application de la maison intelligente de manière simple pour améliorer le confort et assurer la sécurité. Ensuite, la

simulation de la maison intelligente avec objets connectés est également effectuée par le logiciel Packet Tracer de Cisco et Tinkercad.

Enfin, le quatrième chapitre est consacré à la réalisation matérielle (pratique, et montages). La réalisation virtuelle sera réalisée grâce à Arduino IDE. La réalisation pratique se fera avec les composants qui seront utilisés avec la carte node MCU.

Nous terminerons notre étude par une conclusion générale qui sera une synthèse globale de notre réalisation ainsi que des résultats obtenus.

Nous avons rencontré de nombreuses difficultés, notamment Manques des matériels et de temps.

2 Concepts de base

2.1 Internet des objets

2.1.1 Définition

L'IoT (Internet des objets) est un système d'interconnexion entre des dispositifs informatiques des machines ,des objets ,munies d'identifiants uniques (UID) avec la capacité de transférer des données sur un réseau [14].

2.1.2 Les objets connecté

sont des appareils reliés à Internet par un protocole de communication pour informer l'utilisateur de leur état ou pour réaliser des tâches [15].

2.2 Architecture d'un réseaux IOT [1]:

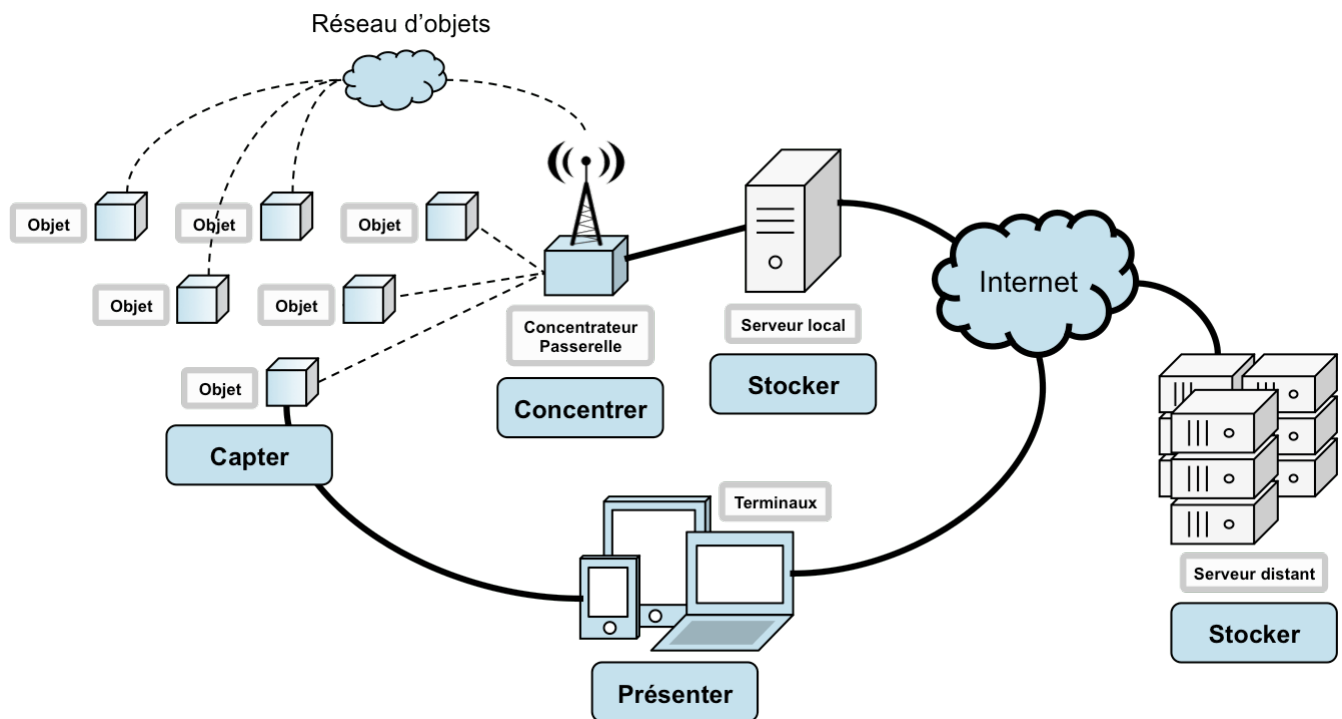


Figure 1: Architecture de l'IOT [1].

Précisons le rôle des différents processus présentés sur la figure [1] :

Capter: désigne l'action de transformer une grandeur physique analogique en un signal numérique.

Concentrer : permet d'interfacier un réseau spécialisé d'objet à un réseau IP standard (e.g. WiFi) ou des dispositifs grand public.

Stocker : qualifie le fait d'agréger des données brutes, produites en temps réel, méta taguées, arrivant de façon non prédictible.

Préserver :indique la capacité de restituer les informations de façon compréhensible par l’Homme, tout en lui offrant un moyen d’agir et/ou d’interagir.

Le traitement des données :est un processus qui peut intervenir à tous les niveaux de la chaîne, depuis la capture de l’information jusqu’à sa restitution. Une stratégie pertinente, et commune quand on parle d’Internet des objets, consiste à stocker l’information dans sa forme intégrale. On collecte de manière exhaustive, « big data », sans préjuger des traitements qu’on fera subir aux données. Cette stratégie est possible aujourd’hui grâce à des architectures distribuées type NoSQL, capables d’emmagasiner de grandes quantités d’information tout en offrant la possibilité de réaliser des traitements complexes en leur sein (Map/Reduce par exemple).

La transmission des données :est un processus qui intervient à tous les niveaux de la chaîne. Deux réseaux, supports des transmissions, cohabitent généralement :

Réseau local de concentration : on utilise alors des technologies comme ANT, ZigBee, Z-wave, NFC ou Bluetooth LE.

Réseau WAN: permettant d’interconnecter les réseaux spécialisés et de les interfacer avec des fermes de serveur. On utilise alors WiFi, les réseaux cellulaires (GSM, UMTS, LTE) ou encore les connexions physiques standard (Ethernet, fibre optique). Ces réseaux sont généralement connectés à Internet.

Table 1: Comparaison des technologies sans fil et de leurs cas d’utilisation idéaux

Technologie sans fil	Vitesse de transmission	Portée maximale	Consommation d’énergie	Exemples d’utilisation
Bluetooth	Faible à moyenne	Courte	Faible	Casques sans fil, enceintes Bluetooth, montres connectées
Wi-Fi	Élevée	Moyenne à longue	Élevée	Réseaux locaux, streaming de médias, domotique
ZigBee	Faible	Courte à moyenne	Faible	Capteurs IoT, éclairage intelligent, contrôle de température
Z-Wave	Faible	Moyenne à longue	Faible	Domotique, sécurité à domicile
LoRaWAN	Faible	Longue	Très faible	Capteurs IoT, ville intelligente, suivi de flotte
NB-IoT	Faible	Longue	Très faible	Réseaux de capteurs IoT, suivi de flotte, télémétrie

2.3 Caractéristiques d’Internet des objets

Connectivité :

La connectivité est une exigence importante pour l’infrastructure IoT. Les objets IoT doivent être connectés à l’infrastructure IoT. N’importe qui, n’importe où, n’importe quand peut se connecter, et cela doit être garanti à tout moment.

Intelligence et identité :

L’extraction des connaissances à partir des données générées est très importante. Par exemple, les

capteurs génèrent des données, mais ces données ne sont utiles que si elles sont correctement interprétées. Chaque appareil IoT a une identité unique.

Changement dynamique :

le changement dynamique de l'état des appareils .

Sécurité :

Les dispositifs IoT doivent être conçus pour être sécurisés et protégés contre les attaques.

2.4 Domaines d'application de l'internet des objets

Plusieurs domaines d'application sont touchés par l'IoT, Parmi ces principaux domaines nous citons: le domaine du sante , le domaine du domotique, le domaine du transport. . . .etc.

2.4.1 La domotique

La domotique fait référence à l'intégration de divers appareils et systèmes intelligents dans la maison pour fournir un contrôle, une automatisation et une surveillance centralisés de diverses fonctions telles que l'éclairage, le CVC, la sécurité, le divertissement, etc. L'objectif de la Domotique est de créer un milieu de vie confortable, sécuritaire, économe en énergie et pratique pour les occupants de la maison.

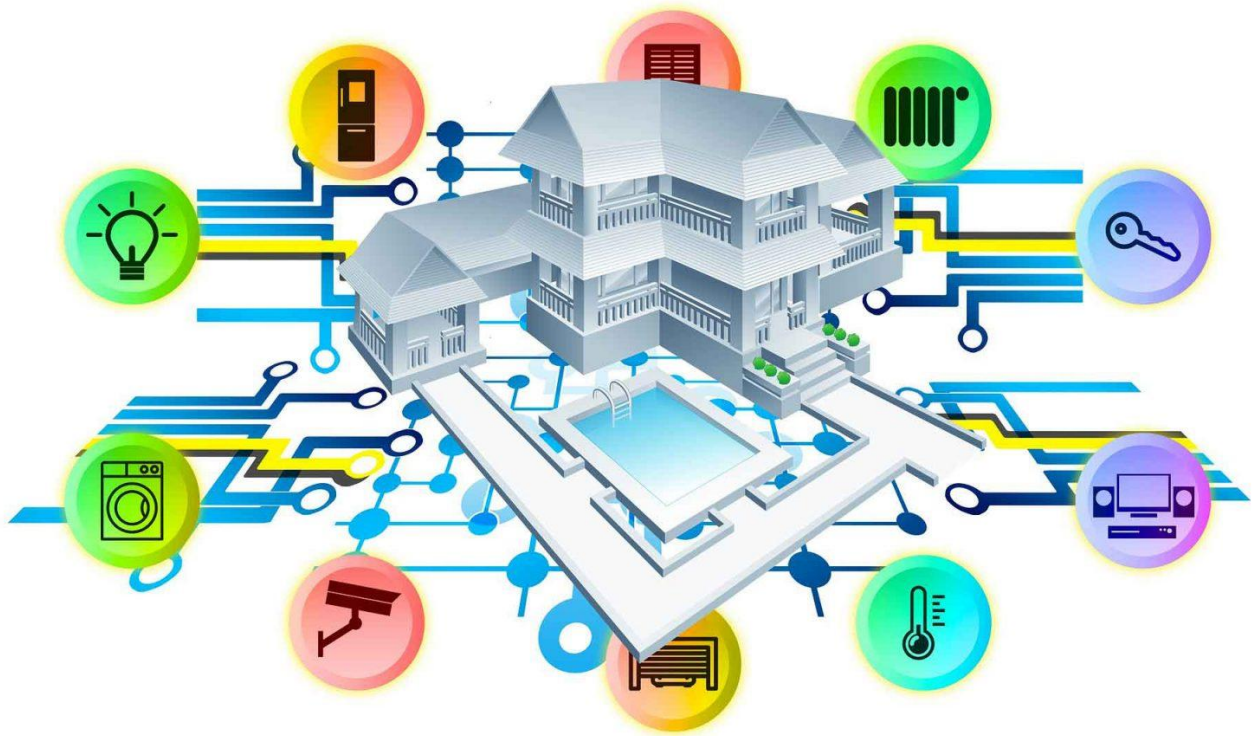


Figure 2: La domotique [2].

2.4.2 La santé

Les appareils IoT peuvent être utilisés pour surveiller à distance la santé des patients, suivre les traitements médicaux et collecter des données sur l'état de santé général.

2.4.3 Les transports

Les véhicules connectés peuvent être utilisés pour surveiller les performances du véhicule, la consommation de carburant, la localisation, la sécurité et l'assistance au conducteur.

2.5 Le protocoles de communication

Les protocoles de communication sont des ensembles de règles et de normes qui permettent aux dispositifs de communiquer entre eux. Les protocoles de communication définissent la manière dont les données sont transférées, formatées et contrôlées lors de la transmission. Il existe plusieurs protocoles applicatifs pour l'Internet des Objets (IoT), qui sont utilisés pour la communication entre les différents dispositifs IoT.

Table 2: Comparaison des protocoles de communication pour l'Internet des Objets

Protocoles	Conçu pour	Modèle de communication	de	QoS	Sécurité	Découverte automatique	Adapté pour
HTTP	Transfert de données entre serveurs et clients	Requêtes/réponses HTTP		Moyenne	Oui	Non	Applications web et mobiles IoT
CoAP	Réseaux de capteurs et IoT à faible consommation d'énergie	Requêtes/réponses HTTP		Limitée	Oui	Oui	Appareils IoT avec une faible consommation d'énergie
MQTT	Réseaux à faible bande passante et haute latence	Publication/abonnement		Oui	Oui	Non	Surveillance et contrôle de dispositifs IoT
DDS	Systèmes distribués, y compris les systèmes IoT	Publication/abonnement		Oui	Oui	Oui	Applications nécessitant une transmission de données en temps réel, tels que les réseaux de capteurs industriels
XMPP	Communication en temps réel entre les appareils IoT et les serveurs	Publication/abonnement		Oui	Oui	Non	Applications nécessitant une communication en temps réel, telles que la télémédecine
AMQP	Systèmes de messagerie distribués, y compris les systèmes IoT	Publication/abonnement		Oui	Oui	Non	Applications nécessitant une communication fiable et sécurisée entre les appareils IoT

MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie léger et basé sur un modèle de publication / abonnement, conçu pour les communications machine à machine (M2M) et l'Internet des objets (IoT).

MQTT a été créé dans le but de fournir une méthode simple et efficace de communication entre des dispositifs connectés à des réseaux à faible bande passante et à haute latence, comme les réseaux de capteurs sans fil ou les réseaux mobiles.

Contrairement à HTTP (Hypertext Transfer Protocol), qui est conçu pour les communications entre clients et serveurs, MQTT est conçu pour les communications de périphérie à périphérie et permet une communication bidirectionnelle. Il utilise un modèle de publication / abonnement dans lequel les

dispositifs peuvent publier des messages sur un canal ou un sujet (topic) et d'autres dispositifs peuvent s'abonner à ces canaux pour recevoir les messages publiés.

Avantages de MQTT [3] :

Flexibilité: MQTT est sur la couche "session" au-dessus de la couche réseau TCP/IP sur le modèle OSI, utilisée par les protocoles comme HTTP. MQTT est donc routable sur Internet. Il est possible de transmettre n'importe quel message sur les topics (sujets), de l'ASCII, du binaire ou du JSON.

Légereté : Les échanges MQTT sont beaucoup plus légers que HTTP.

Sécurité : La sécurisation est possible avec un chiffrement SSL/TLS, mais cela a un coût non négligeable en termes de performances sur de l'embarqué.

MQTT sécurisé ne peut être déployé sans consommation excessive qu'avec du matériel puissant ce qui exclut les Arduino 8bits par exemple.

Intégrité des données : MQTT introduit la notion de qualité de service (QOS) qui permet à un client de s'assurer qu'un message a bien été transmis, avec différents niveaux de fiabilité [16].

fonctionnement de MQTT

Une session MQTT est divisée en quatre étapes : connexion, authentification, communication et terminaison. Un client commence par créer une connexion TCP/IP vers le broker.

Les ports standards sont : 1883 pour la communication non chiffrée et 8883 pour la communication chiffrée utilisant SSL/TLS.

Publication MQTT

le client possède des capteurs et transmet leurs valeurs Le client se connecte au broker avec son identifiant et mot de passe (CONNECT).

Le broker accepte la connexion (CONNACK).

Le client peut publier des données (PUBLISH).

Les données sont regroupées dans des TOPICS, (ex température, pression, humidité , GPS, etc...).

Abonnement MQTT

le client va recevoir les messages transmis au broker par les capteurs afin de les exploiter (prise de décision, affichage, etc...).

Le client se connecte au broker avec son identifiant et mot de passe (CONNECT).

Le broker accepte la connexion (CONNACK).

Le client souscrit à des TOPICS (ex température, pression, humidité , GPS ... (SUBSCRIBE)).

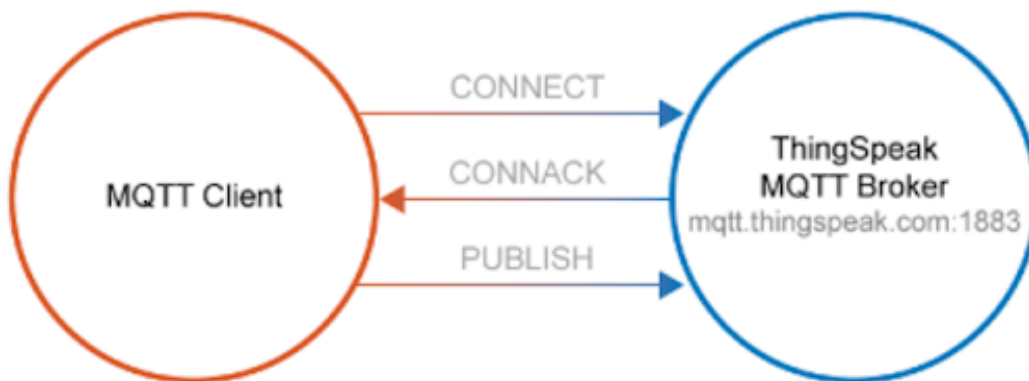


Figure 3: Publication MQTT [3].

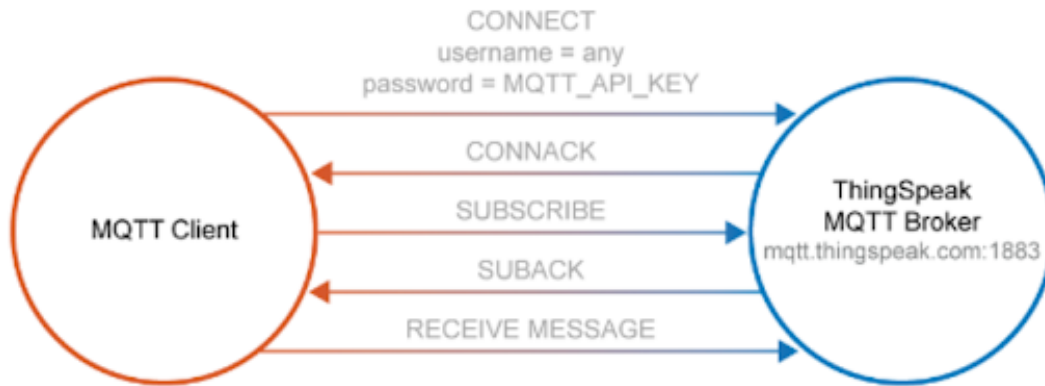


Figure 4: Abonnement MQTT [3].

A chaque publication le broker transmet les données des TOPICS au client. (SUBACK).
Un acquittement peut être transmis (RECEIVE MESSAGE).

Quality Of Service (QoS)

Envoyer et oublier (QoS 0) Le client ne recevra aucune confirmation du broker à la réception. De même, un message remis à un client par le broker ne sera pas acquitté. C'est le moyen le plus rapide de publier et de recevoir des messages, mais aussi celui où la perte de messages est la plus susceptible de se produire.

Au moins une fois (QoS 1) Chaque message du client vers le broker ou du broker vers le client doit être acquitté. Si l'acquittement n'arrive pas dans le temps imparti, le message est réexpédié.

Exactement une fois (QoS 2) QoS 2 est le niveau de service le plus élevé de MQTT. Ce niveau garantit que chaque message n'est reçu qu'une seule fois par les destinataires prévus. QoS 2 est le niveau de qualité de service le plus sûr et le plus lent. La garantie est fournie par au moins deux flux de demande / réponse (une négociation en quatre parties) entre l'expéditeur et le récepteur. L'expéditeur et le destinataire utilisent l'identifiant de paquet du message PUBLISH d'origine pour coordonner la remise du message.

Lorsqu'un récepteur obtient un paquet QoS 2 PUBLISH d'un expéditeur, il traite le message de publication en conséquence et répond à l'expéditeur avec un paquet PUBREC qui reconnaît le paquet PUBLISH. Si l'expéditeur n'obtient pas de paquet PUBREC du récepteur, il envoie à nouveau le paquet PUBLISH avec un drapeau dupliqué (DUP) jusqu'à ce qu'il reçoive un accusé de réception. Exemple de séquence d'échanges :

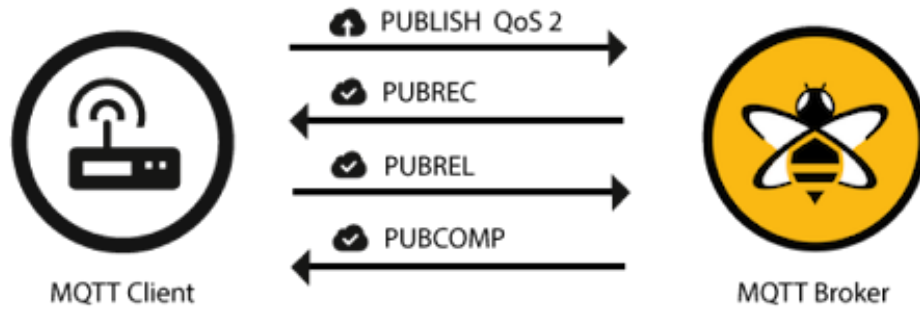


Figure 5: Quality Of Service (QOS) [3].

MQTT et modèle OSI

MQTT se situe sur la couche 5 du modèle OSI.

Dans la couche 6 le message est codé. (binaire, JSON, ASCII ou autre) et transmis à la couche 5 (MQTT) qui se charge du protocole. Les données MQTT sont alors encapsulées dans une trame TCP-IP (couche 4)

Table 3: MQTT et modèle OSI

Layer	Dénomination	Protocole	Description
7	Application		Acquisition de données/contrôle processus
6	présentation	Binaire, JSON, ASCII	Codage du message
5	Session	MQTT	Message IOT
4	Transport	TCP	Transmission Control Protocol
3	Réseau	IP	Internet Protocol
2	Liaison	802.11	Adressage MAC
1	Physique	802.11	Ondes radio (Wi-Fi)

Exemple de message MQTT

Table 4: Format des messages

Control Header	Packet Length	Variable length header	Payload
Type de message	Longueur du message	Dépend du type de message	données
1 octet	1 à 4 octets	0 à x octets	0 à 256 MOctets

La taille du message dépend du type de message :

2 octets : Control header + Length : par exemple "CONNACK" pour un acquittement de connexion.

3 octets : Control header + Length + Variable header : par exemple "PUBACK" pour un acquittement de publication.

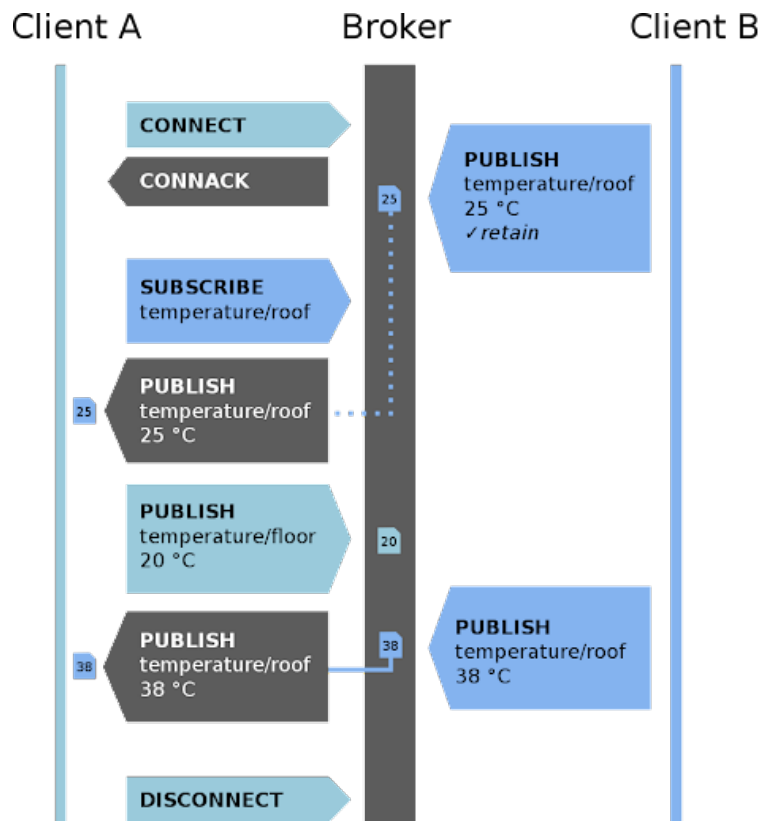


Figure 6: séquence d'échanges [3].

n octets : Control header + Length + Variable header+ Payload (données) : par exemple "CONNECT" avec ID et pass.

La taille minimum d'un paquet est 127 octets, Paket length contient alors un octet codé sur 7bits.

La taille maximum d'un paquet est de 256 Moctets.

3 Conception et Simulation

3.1 Analyse et Conception

3.1.1 Le langage UML

UML est un langage de modélisation capable d'exprimer et de développer des modèles d'objets indépendamment de tout langage de programmation. UML est sous la seule responsabilité d'OMG (Object Management Group) et vise à supporter l'analyse basée sur des concepts d'objet. Il est défini comme un langage de modélisation graphique et textuel conçu pour comprendre et décrire les exigences, spécifier et documenter les systèmes, concevoir des solutions et communiquer des idées.

UML unifie la notation et les concepts orientés objet. Il ne s'agit pas simplement de symboles graphiques, car les concepts véhiculés par les schémas ont une sémantique précise et ont autant de sens que les mots d'une langue.

UML utilise différents types de diagrammes pour représenter les différents aspects d'un système. Nous allons représenter :

- Le diagrammes de cas d'utilisation
- Le diagrammes de séquence
- Le diagrammes d'activité : ils montrent le flux de contrôle entre les activités d'un système.

3.1.2 Le diagramme Cas d'utilisation

Le diagramme de cas d'utilisation est l'un des types de diagrammes UML les plus couramment utilisés. Il est utilisé pour modéliser les exigences fonctionnelles d'un système à travers les interactions entre les utilisateurs du système, identifie les acteurs du système et identifie les scénarios d'utilisation.

Le diagramme de cas d'utilisation de la (Figure 7) représente les différentes fonctionnalités que les utilisateurs peuvent effectuer dans l'application:

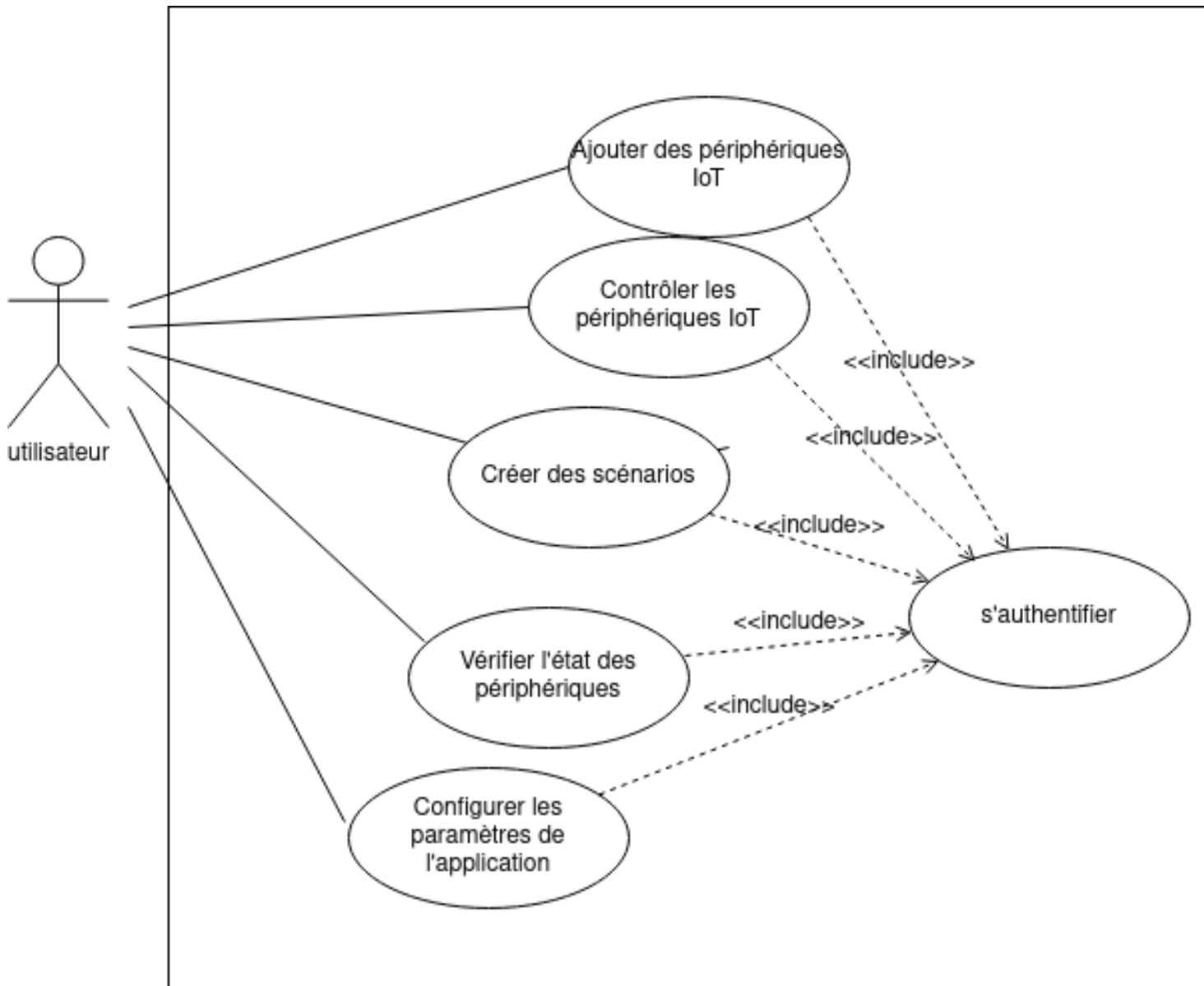


Figure 7: Diagramme de cas d'utilisation global.

Le cas d'utilisation 'Ajouter un périphérique' permet d'ajouter un nouveau périphérique à l'application en accédant aux paramètres et en sélectionnant l'option 'Ajouter un périphérique'. Quand au cas 'Contrôler un périphérique' il permet d'inclure des actions telles que l'allumage/éteignage, le réglage des paramètres, le changement de couleur, etc., en fonction des fonctionnalités du périphérique, en cas de 'Créer un scénario' il permet de définir les déclencheurs, les conditions et les actions à exécuter lorsque le scénario est activé. Par exemple, il peut configurer un scénario pour allumer automatiquement les lumières lorsque le capteur de mouvement détecte une présence, le cas d'utilisation 'Vérifier l'état d'un périphérique' l'utilisateur peut vérifier l'état actuel d'un périphérique iot. Cela peut inclure des informations telles que l'état de l'appareil (allumé/éteint), la température actuelle, l'humidité, etc. Cela permet à l'utilisateur de surveiller l'état de ses appareils connectés en temps réel, et le cas d'utilisation 'Configurer les paramètres de l'application' permet à l'utilisateur de personnaliser l'interface, de gérer les préférences, les notifications, la sécurité.

3.1.3 Diagramme de séquence

Le diagramme de séquence utilisé pour modéliser les interactions entre les objets dans un scénario particulier. Il est composé d'objets, de messages et de durées, et est utilisé pour identifier les séquences d'actions, les collaborations et pour vérifier les exigences fonctionnelles.

Le diagramme de séquence de la (Figure 8) montre comment l'utilisateur peut s'authentifier dans l'application web :

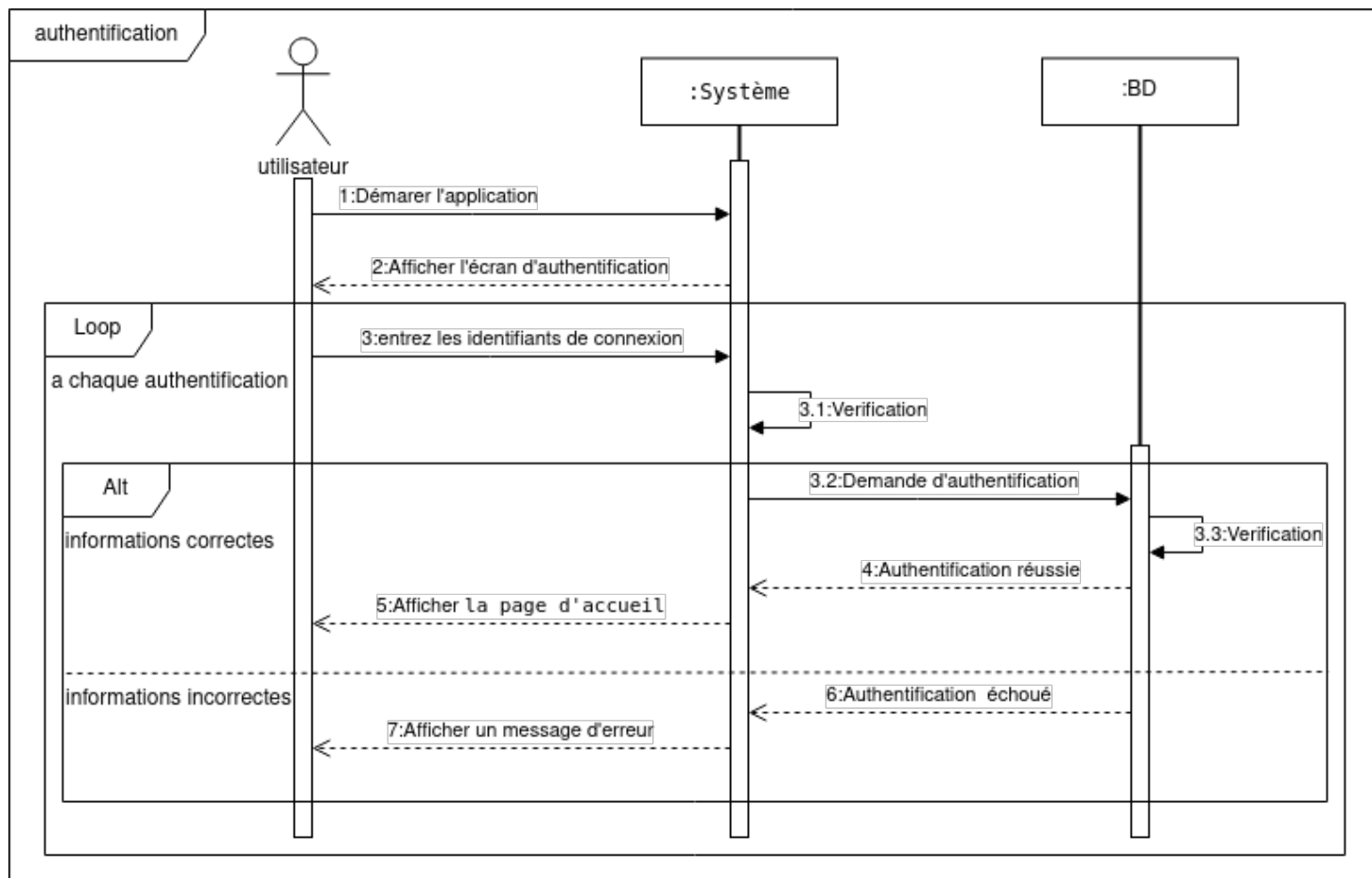


Figure 8: Diagramme de cas d'utilisation global.

Le diagramme de séquence de la (Figure 9) montre comment l'utilisateur peut ajouter un nouveau périphérique IoT à l'application :

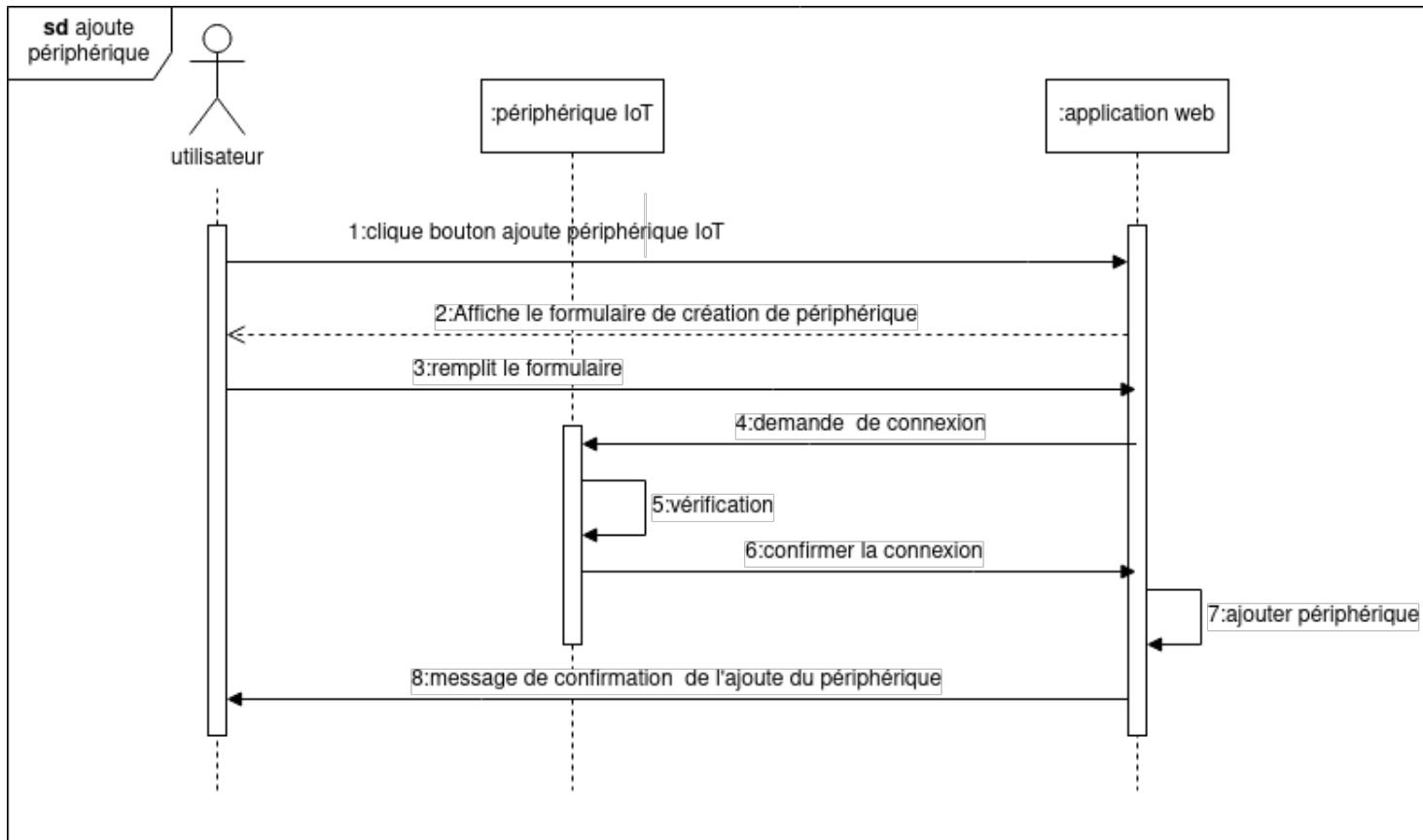


Figure 9: Diagramme de séquence ajouter un nouveau périphérique IoT.

Le diagramme de séquence de la (Figure 10) montre comment l'utilisateur peut contrôler un périphérique IoT :

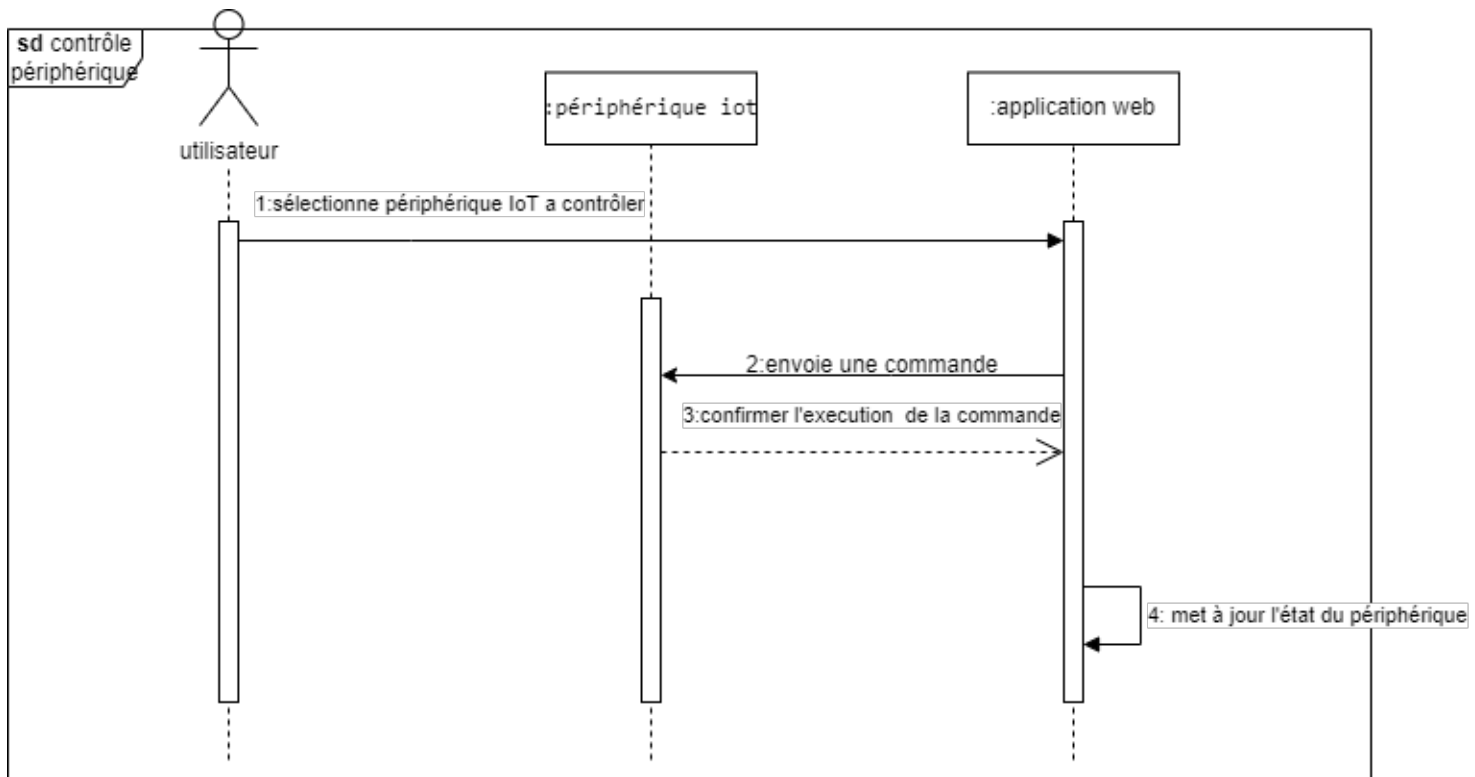


Figure 10: Diagramme de séquence contrôler un périphérique IoT.

Le diagramme de séquence de la (Figure 11) montre comment l'utilisateur peut créer un scénario pour automatiser les actions des périphériques IoT :

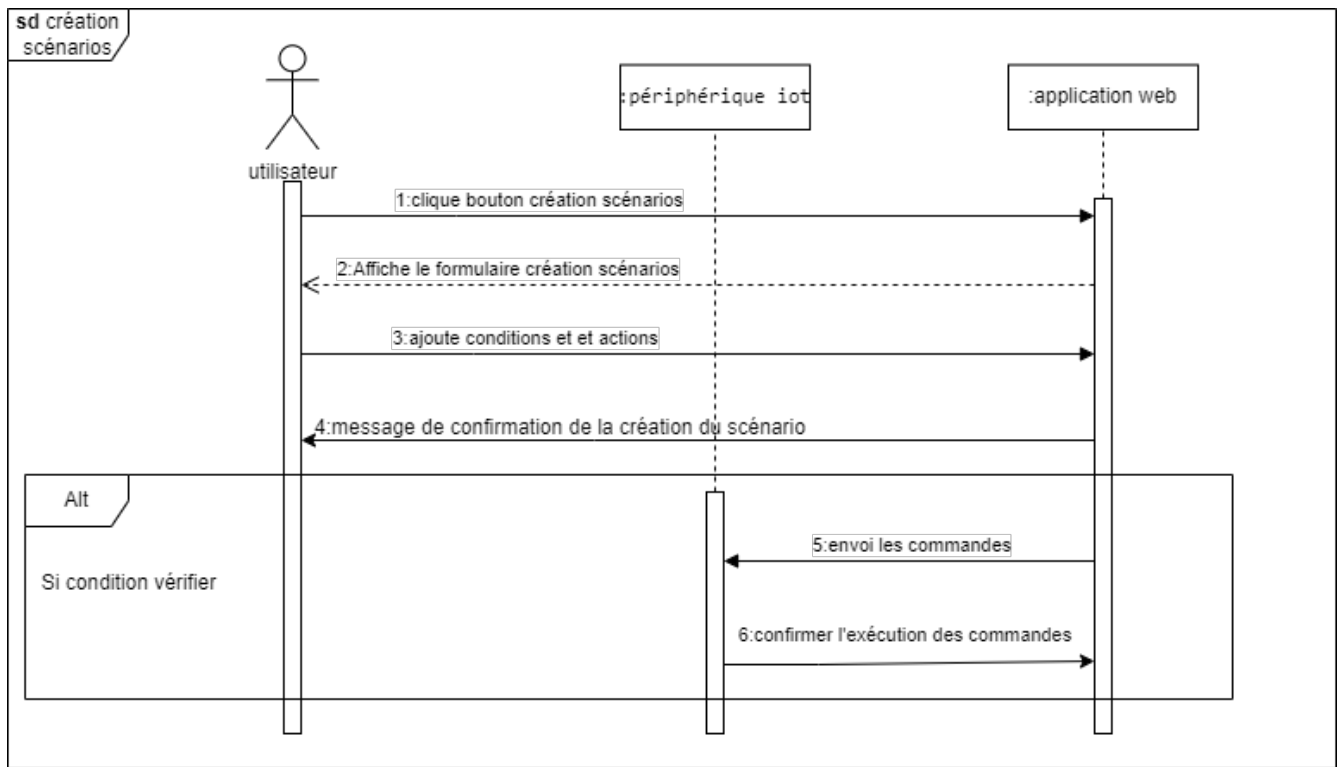


Figure 11: Diagramme de séquence créer un scénario .

3.1.4 Diagramme d'activité

Les diagrammes d'activité UML vous permettent de visualiser un cas d'utilisation à un niveau plus détaillé. Il s'agit d'un diagramme de comportement illustrant le flux d'activités dans un système [].

Un diagramme d'activité se compose de nœuds qui représentent les étapes d'un processus, tels que les actions, les décisions, les forks (bifurcations), les joins (réunions), les délais, les boucles et les points de départ et d'arrivée. Les arcs reliant ces nœuds représentent les transitions entre les étapes et les contraintes qui régissent le flux d'activités [].

Les diagrammes d'activité peuvent être utilisés pour modéliser une grande variété de processus, tels que les processus métier, les processus de fabrication, les processus de développement de logiciels, les processus de gestion de projets, etc. Ils peuvent être utilisés à diverses étapes du cycle de vie du système, depuis la conception jusqu'à la mise en œuvre et la maintenance.

Le diagramme d'activité de la (Figure 12) montre comment l'utilisateur peut ajouter un périphérique IoT :

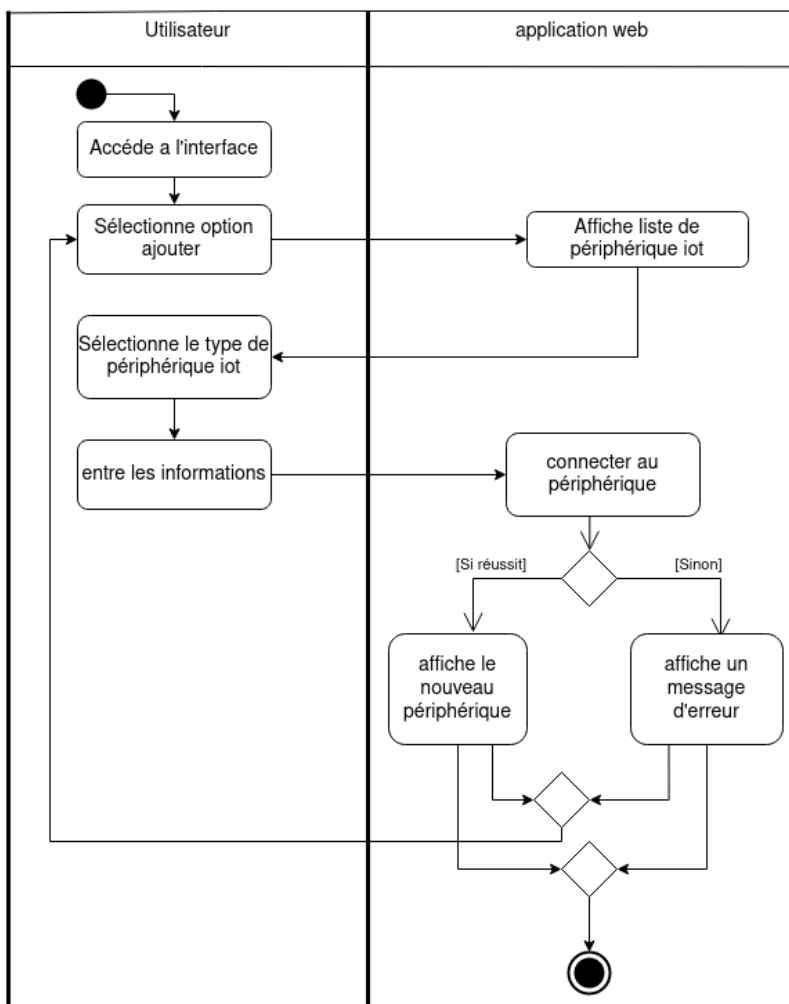


Figure 12: Diagramme d'activité pour ajouter un périphérique IoT.

Le diagramme d'activité de la (Figure 13) montre comment l'utilisateur peut contrôler un périphérique IoT :

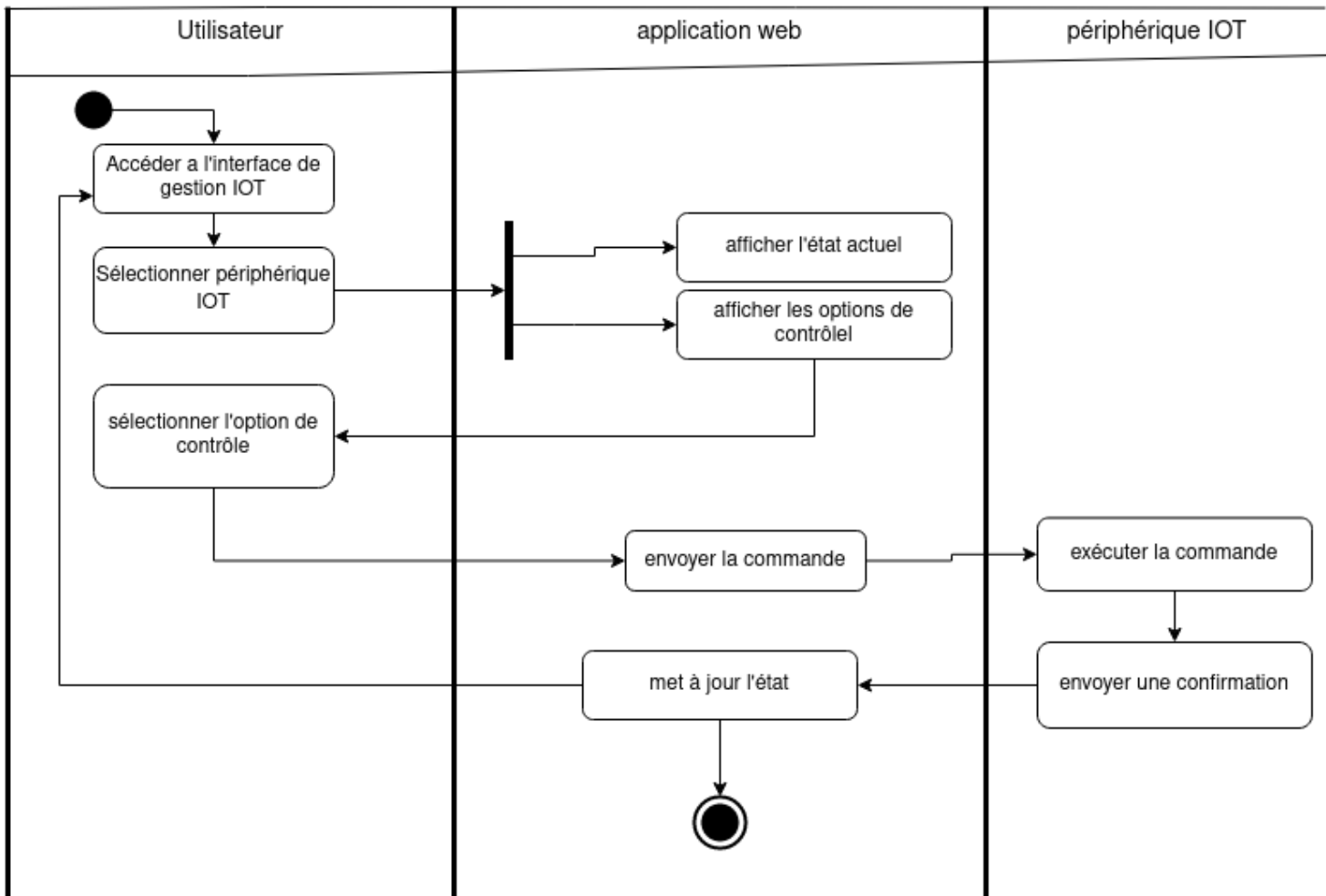


Figure 13: Diagramme d'activité pour contrôler un périphérique IoT.

Le diagramme d'activité de la (Figure 14) montre comment l'utilisateur peut créer un scénario automatisé pour ce périphérique IoT :

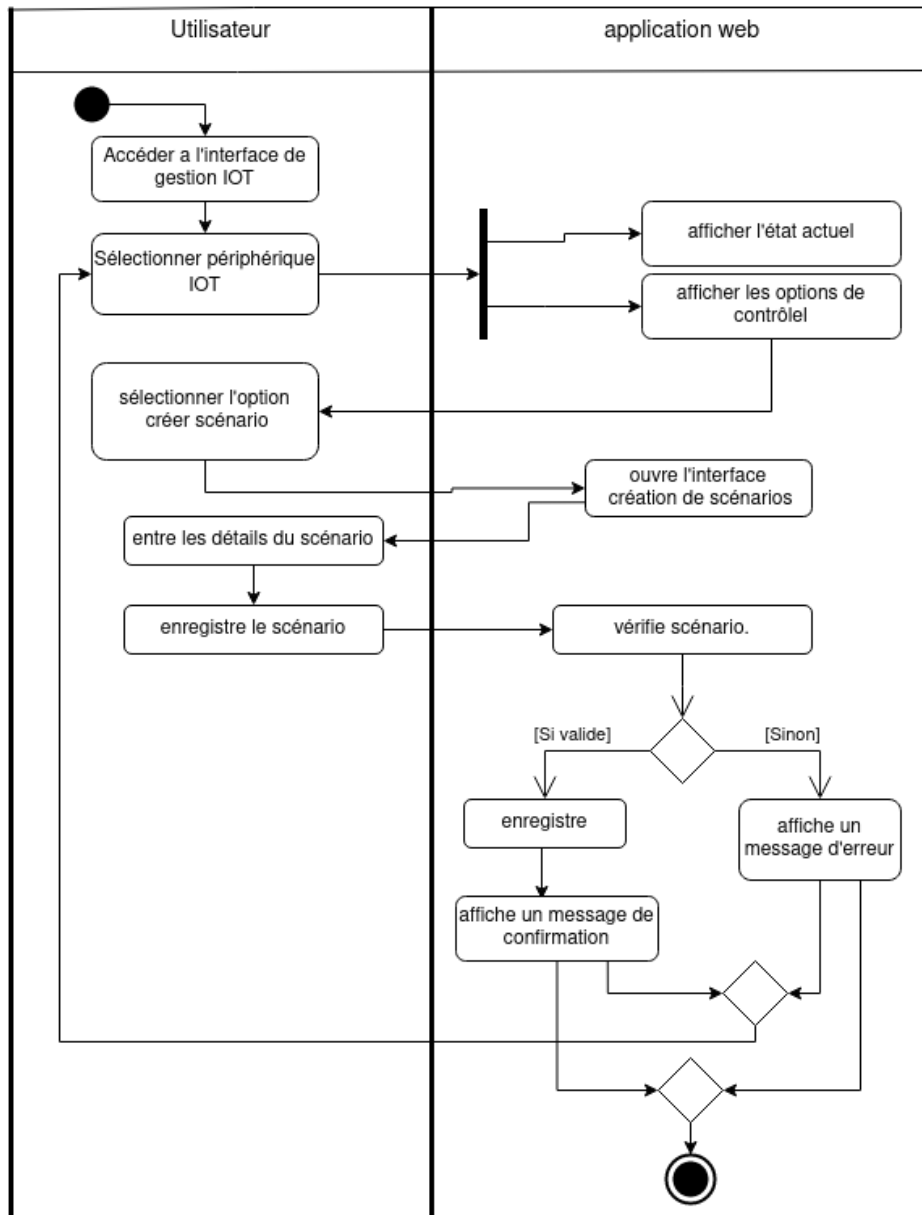


Figure 14: Diagramme d'activité pour créer un scénario automatisé pour ce périphérique IoT.

3.2 Simulation

3.2.1 Présentation de Tinkercad

Tinkercad est un logiciel de modélisation 3D en ligne développé par Autodesk. Il offre une interface utilisateur intuitive qui permet aux utilisateurs de créer des modèles 3D à partir de formes géométriques de base et qui peuvent intégrer des circuits électroniques. Tinkercad propose également une bibliothèque de modèles 3D prêts à l'emploi, ainsi que des outils de modification de forme, de rotation et de mise à l'échelle pour créer des modèles plus complexes. Les modèles créés dans Tinkercad peuvent être exportés sous forme de fichiers STL, qui peuvent ensuite être utilisés pour l'impression 3D ou la découpe laser.

Tinkercad est souvent utilisé dans les programmes d'éducation pour aider les étudiants à apprendre les concepts de la conception 3D et de l'impression 3D. Il est également utilisé par les amateurs, les designers et les ingénieurs pour créer des prototypes, des pièces de rechange et des projets créatifs.

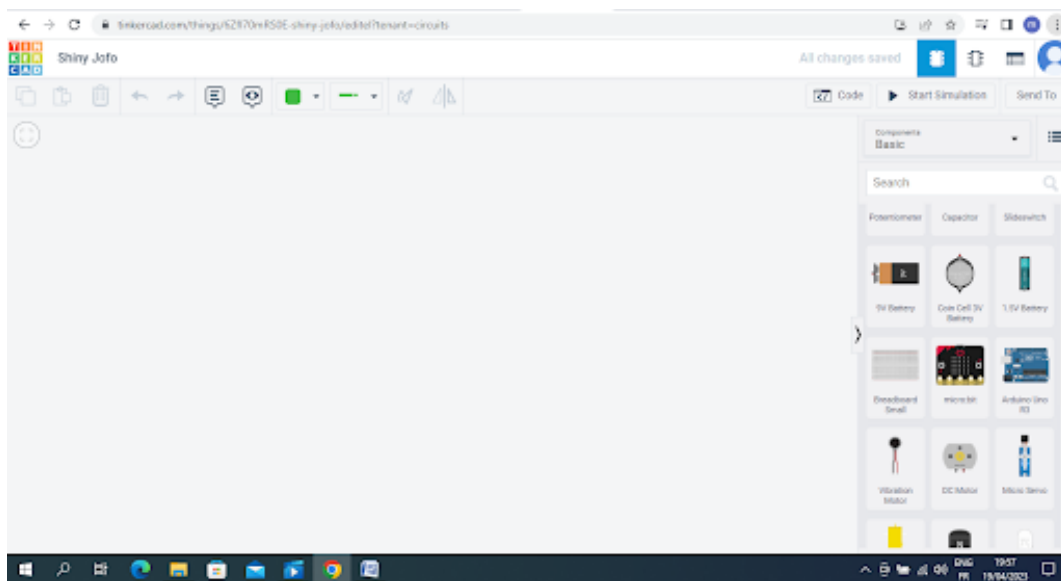


Figure 15: Tinkercad.

3.2.2 Présentation de notre projet sur Tinkercad

Les outils tester sur tinkercad :

- **PIR capteur:**

Ici nous avons relié un capteur de mouvement (PIR) avec la carte Arduino Uno.

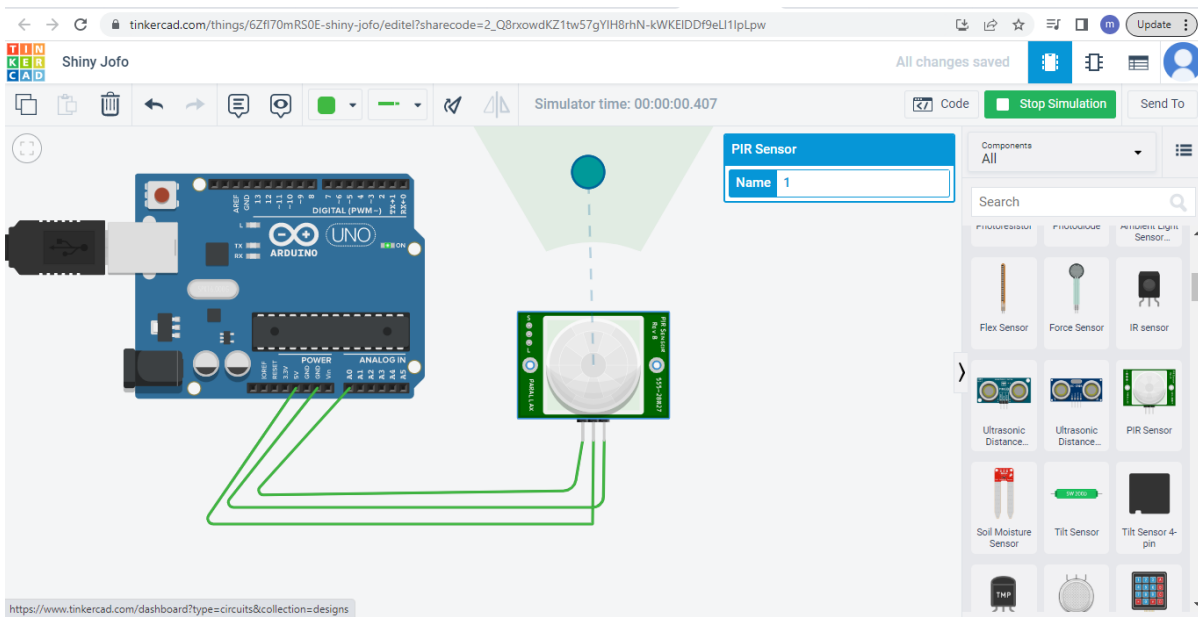


Figure 16: PIR capteur.

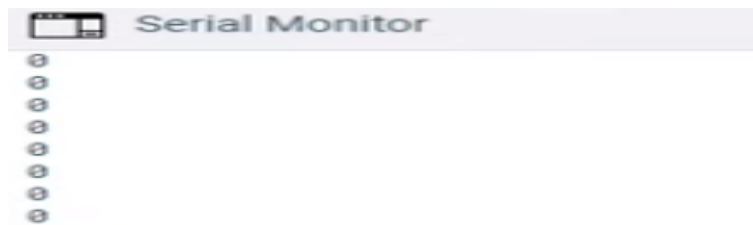


Figure 17: Serial monitor.

Notez ici quand il n'y a pas de corps à côté du capteur il affiche la valeur 0.

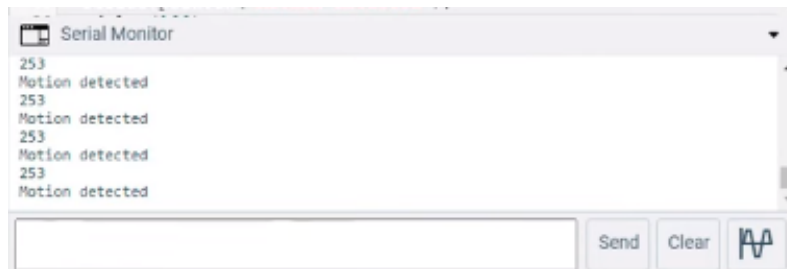


Figure 18: Serial monitor.

Par contre ici on a un corp a coté du capteur donc il affiche la valeur 253.

- 2/ Capteur d'humidité :

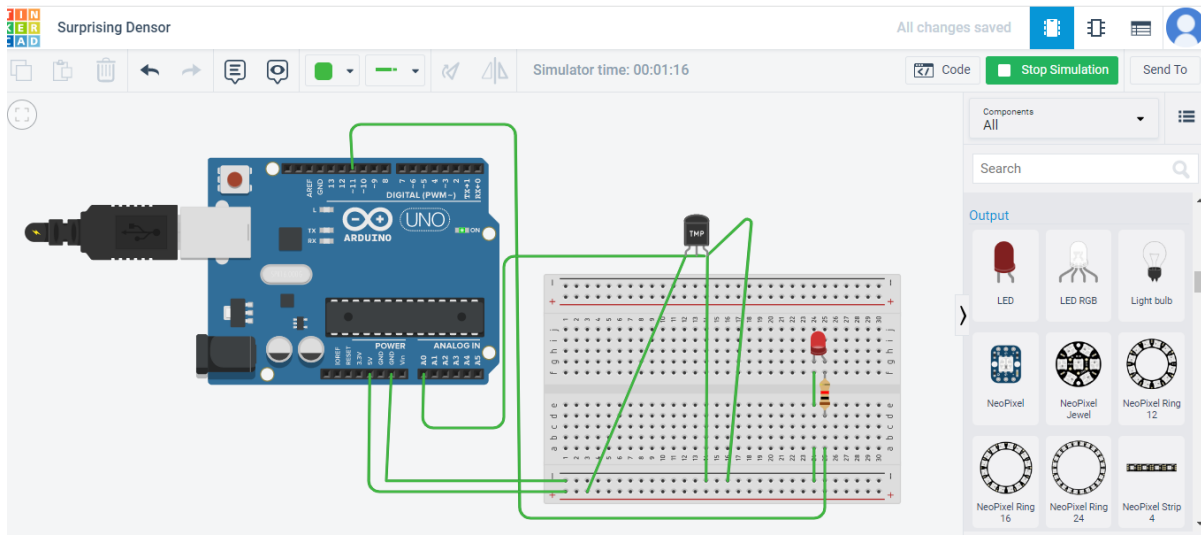


Figure 19: Capteur d'humidité .

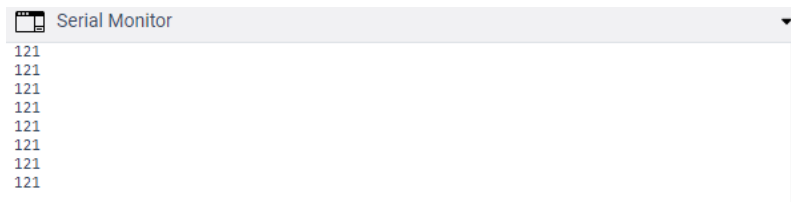


Figure 20: Serial monitor.

Notez ici que lorsque l'humidité est supérieure à 100, la lumière s'allume ,et afficher la valeur 121 .

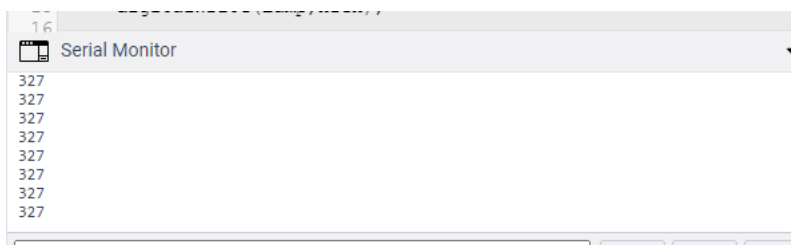


Figure 21: Serial monitor.

Par contre ici lorsque l'humidité est inférieure à 100, la lumière s'éteint et afficher la valeur 327.

- **Porte verrouillée :**

Ici nous avons relié un keypad ,LCD et servo motor avec de la carte arduino uno .

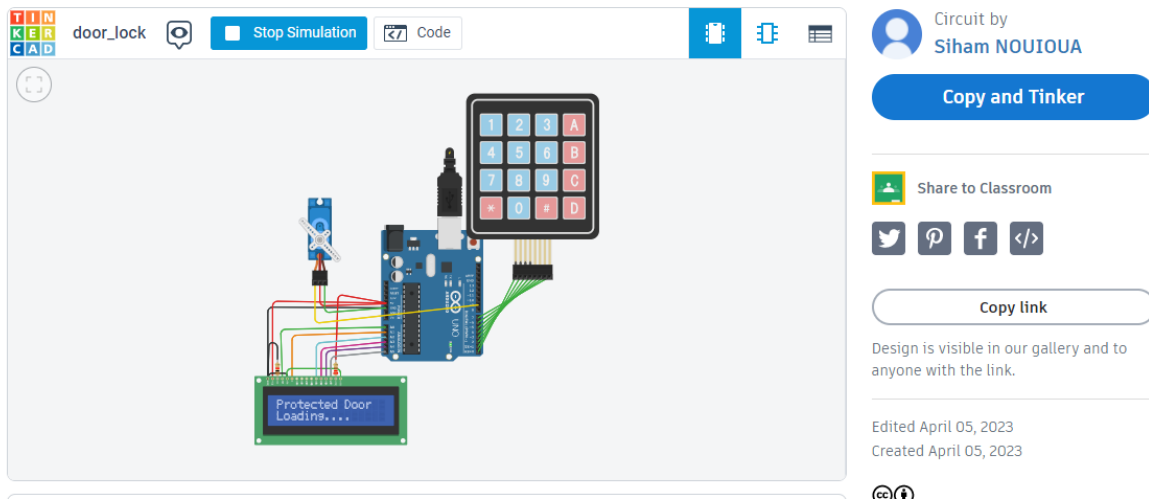


Figure 22: Porte verrouillée

On remarque ici que la porte est fermée comme indiqué sur l'écran LCD.

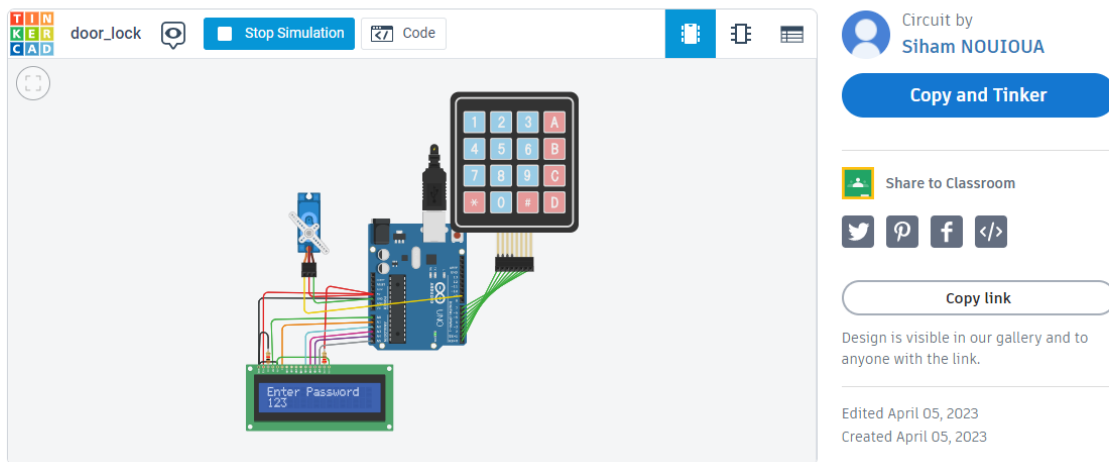


Figure 23: Porte verrouillée

Après, il faut entrer le mot de passe comme indiqué dans l'image .

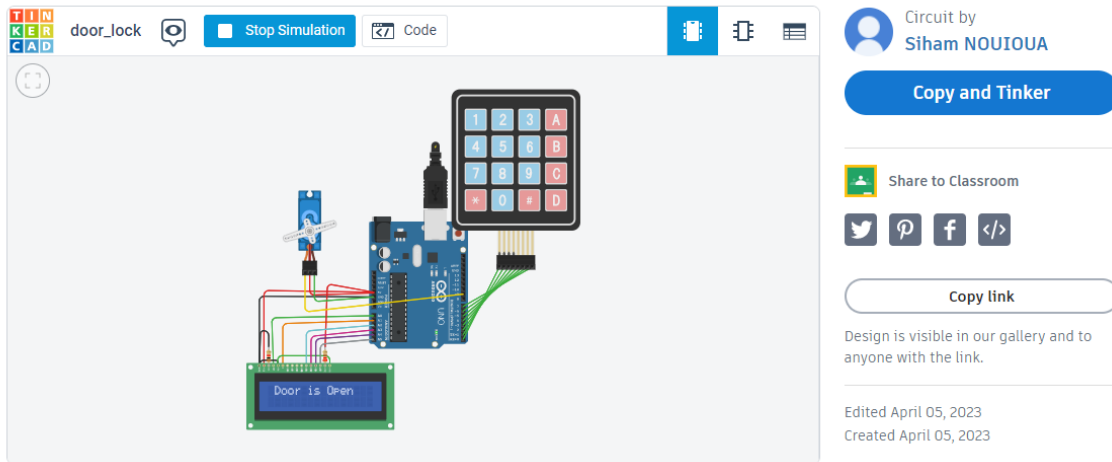


Figure 24: Porte verrouillée

Après la saisie de mot de passe la porte est ouverte.

- **Capteur de gaz :**

Ici nous avons relié un capteur de gaz, une alarme piezoélectrique, des leds, des résistances avec la carte Arduino Uno.

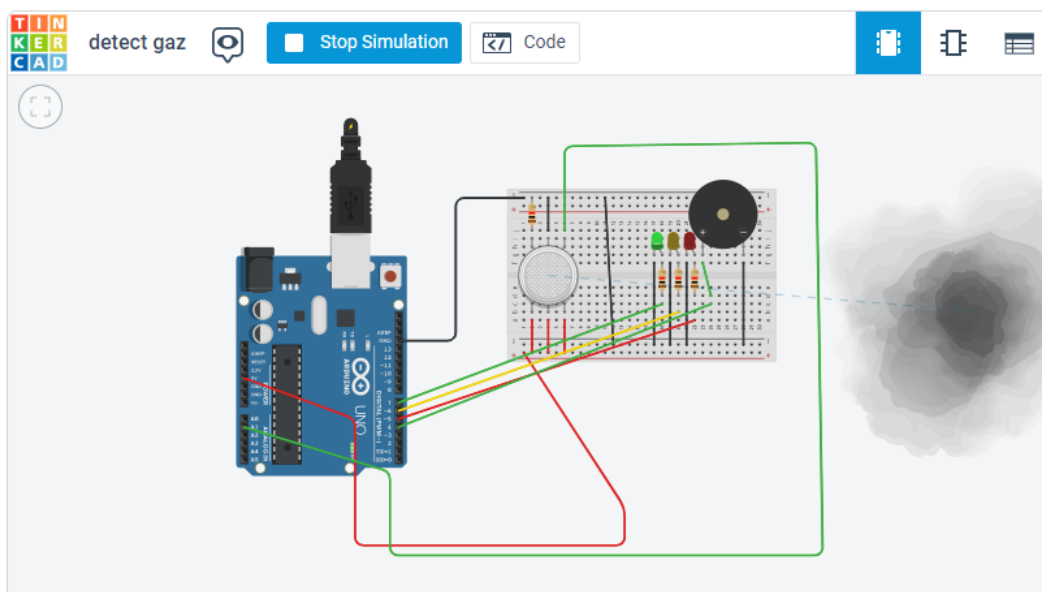


Figure 25: Capteur de gaz.

On remarque ici que lorsque le gaz est éteint, le voyant de la lumière verte est allumé.

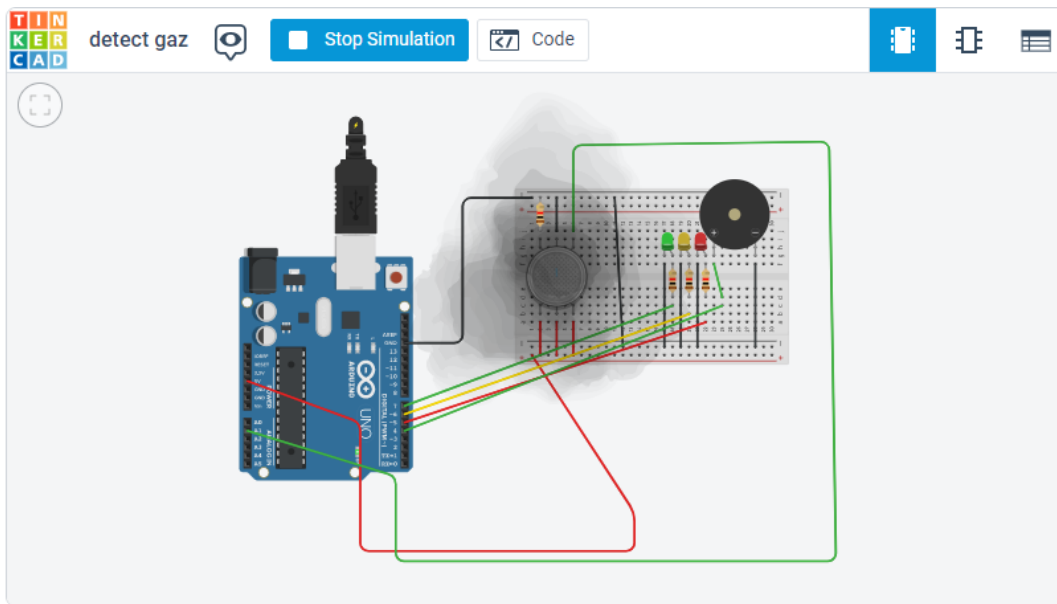


Figure 26: Capteur de gaz.

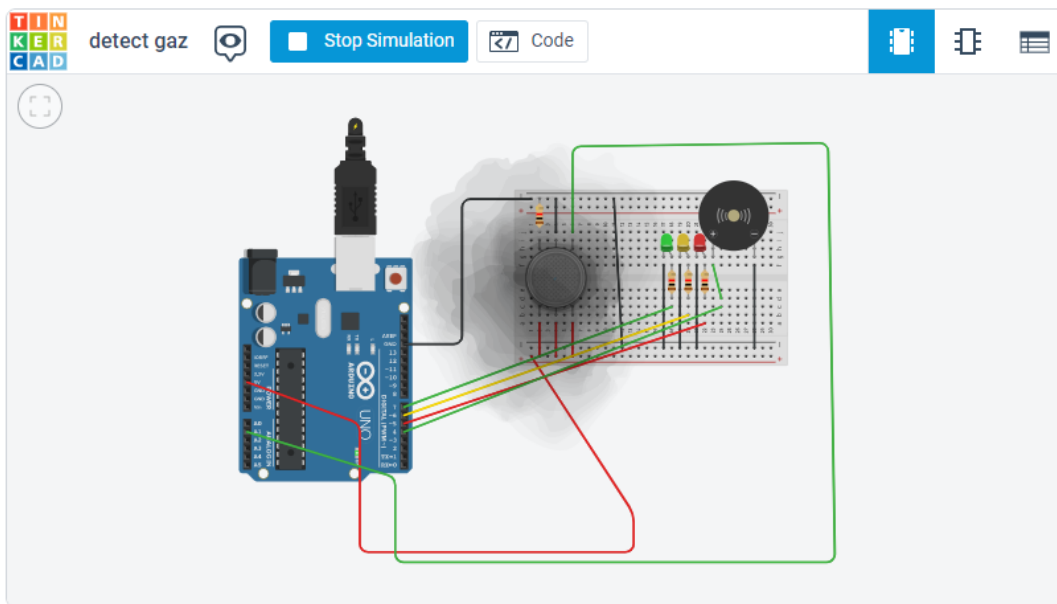


Figure 27: Capteur de gaz.

Notez qu'ici, quand on a le gaz allumé la lumière rouge avec l'alarme sont allumés.

3.2.3 Présentation de Cisco Packet Tracer

Packet Tracer est un logiciel de CISCO permettant de construire un réseau physique virtuel et de simuler le comportement des protocoles réseaux sur ce réseau.

L'utilisateur construit son réseau à l'aide d'équipements tels que les routeurs, les commutateurs ou des ordinateurs.

Ces équipements doivent ensuite être reliés via des connexions (câbles divers, fibre optique). Une fois l'ensemble des équipements reliés, il est possible pour chacun d'entre eux, de configurer les adresses IP, les services disponibles, etc . . . [17].

La version 8 de Packet Tracer est amélioré et contient des nouvelle fonctionnalité.

On a utiliser des équipements qui possèdent le service Internet Of Things.. Les capteurs et les langages de programmation ont été aussi améliorés dans cette version de Packet Tracer. Cet environnement nous permet de réaliser notre projet.

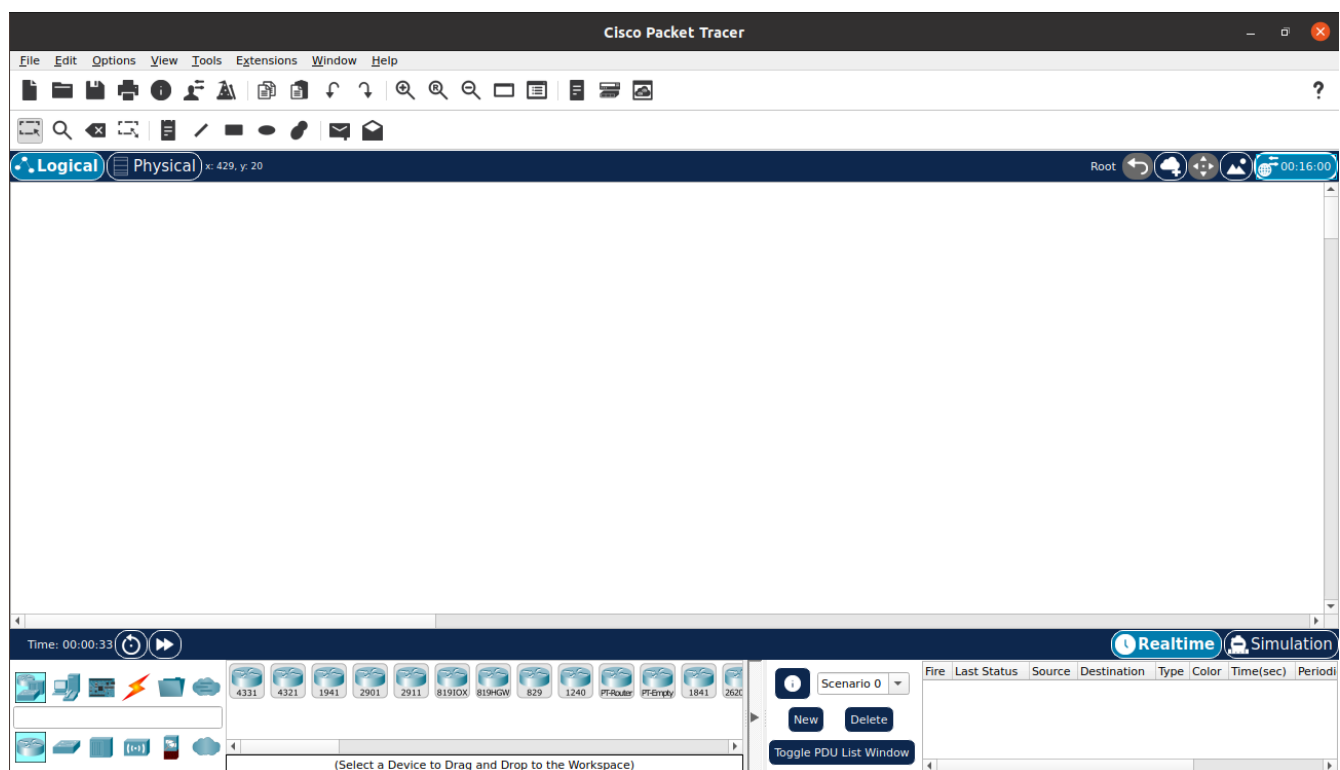


Figure 28: représentation de Cisco Packet Tracer 8.

3.2.4 Présentation de notre projet sur Cisco Packet Tracer

Dans notre projet, on simulé un réseau d'IOT domestique avec le protocole MQTT. Le but est de simuler les différentes conditions et scénarios que le matériel peut rencontrer dans des situations réelles. De tester le réseau: testez la connectivité des appareils dans un environnement domestique simulé.

Nous avons utilisé des cartes électroniques avec des microcontrôleurs. Tels que SBC (Single Board Computer) et MCU (Multiple Control Unit) dans Cisco Packet Tracer.

Les cartes électroniques utilisées : SBC (Single Board Computer):

L'ordinateur à carte unique (SBC) est un ordinateur complet et fonctionnel dans lequel le microprocesseur, les fonctions d'entrée/sortie, la mémoire et d'autres fonctionnalités sont tous intégrés sur une seule carte de circuit imprimé, avec de la RAM intégrée à une quantité prédéterminée et avec pas de connecteurs d'extension pour les périphériques, offrant une solution compacte et autonome pour des applications embarquées et des projets électroniques.

MCU (Multiple Control Unit):

Un microcontrôleur (MCU) est un petit ordinateur intégré dans un système sur puce (SoC). Il contient un cœur de processeur, une mémoire et des appareils d'entrée et de sortie programmables. Les microcontrôleurs sont conçus pour les applications intégrées ou les applications qui nécessitent peu de ressources informatiques. Les microprocesseurs utilisés dans les ordinateurs sont souvent utilisés pour prendre en charge d'autres applications générales qui requièrent un plus grand nombre de ressources.

Les scénarios domotique :

scénarios domotiques qu'on a réalisés avec Cisco Packet Tracer sont ::

- L'ouverture de la porte à l'aide de RFID.
- Détection de mouvement.
- Détection de gaz.
- Détection de l'humidity.

L'ouverture de la porte à l'aide de RFID : scénario :

Dans ce scénario, le lecteur RFID détecte si la carte est valide, si c'est le cas le MCU envoie un signal à la porte pour permettre son ouverture, sinon il n'ouvre pas la porte.

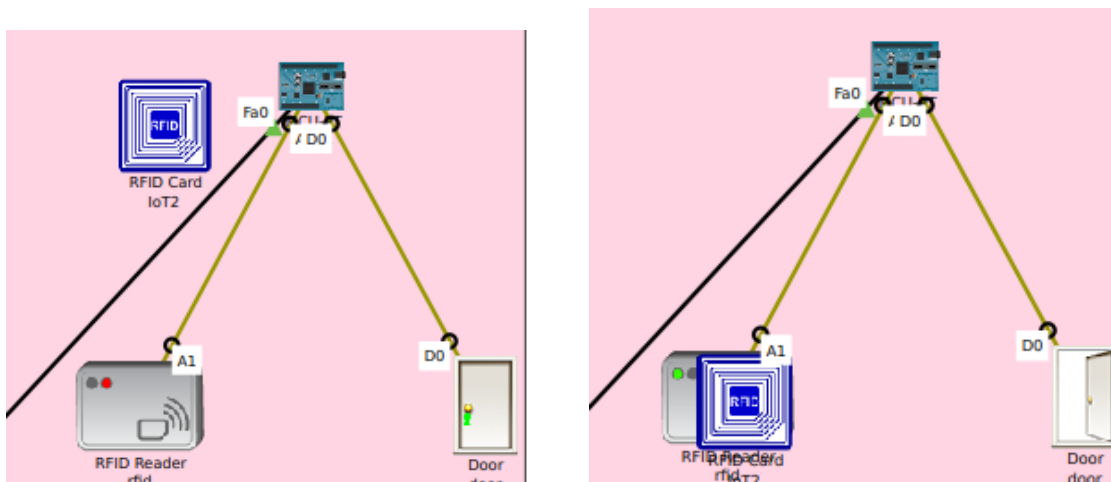


Figure 29: l'ouverture de la porte à l'aide de RFID

Connexion des périphériques iot:

- On ajoute le lecteur RFID ,la carte RFID se trouve sous **End Devices » Smart City** ,la porte se trouve sous **End Devices » Home**.

- ajoute la carte MCU se trouve sous **Components** » **Boards**.
- Nous avons utilisé un câble iot se trouve sous **Connections** » **Connections** pour nous connecter les peripherique avec la carte MCU.
- ce tableau montre les ports :

Périphérique	Port	Port du carte MCU.
Le lecteur RFID	A1	A0
La porte	D0	D0

Programmation de la carte MCU :

```

Specifications Physical Config Programming Attributes
Blink (JavaScript) - main.js
Open New Delete Rename Import Stop Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -
main.js
1 var door=0;
2 var reader=A0;
3
4
5 function setup() {
6   pinMode(door, OUTPUT);
7   pinMode(reader, INPUT);
8
9
10 }
11
12 function loop() {
13   if(analogRead(reader)===0){
14     customWrite(door,1);
15   }
16   else{
17     customWrite(door,0);
18   }
19
20 }
21 }
22

```

Figure 30: Programmation de la carte MCU.

Détection de mouvement : scénario :

Dans ce scénario, on ajoute un capteur de mouvement. Si il capte un mouvement, la port ouvre automatiquement et la lumière de la chambre est allumée, on utilise la carte SBC pour programmer ce scénario.

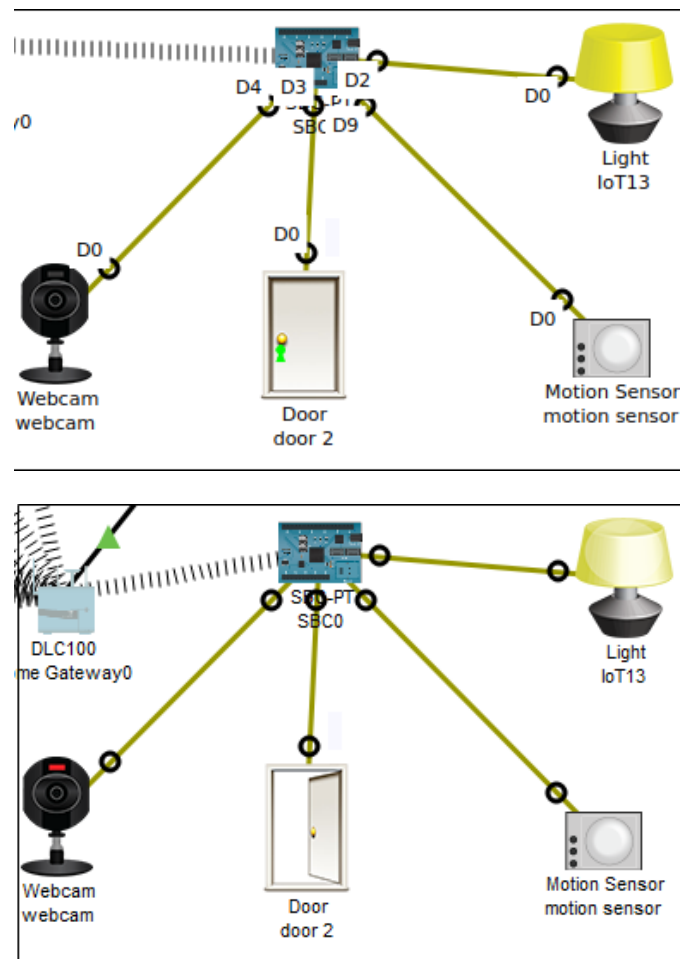


Figure 31: Détection de mouvement

Connexion des périphériques iot:

- On a connecté les périphériques IOT : la webcam, la porte, le led, le détecteur de mouvement avec la carte SBC à l'aide des câbles IOT.
- Ce tableau montre les ports :

Périphérique	Port	Port du carte SBC.
Detector de mouvement	D0	D9
Webcam	D0	D4
La porte	D0	D3
Le led	D0	D2

Programmation de la carte SBC :

Le programme python que nous avons écrit est le suivant :



Figure 32: Programmation de la carte SBC.

Pour connecter la carte SBC à la passerelle domestique (Home Gateway) allez sur la carte **SBC»Config»Wireless»SSID** et taper le SSID de la passerelle comme il est montré.

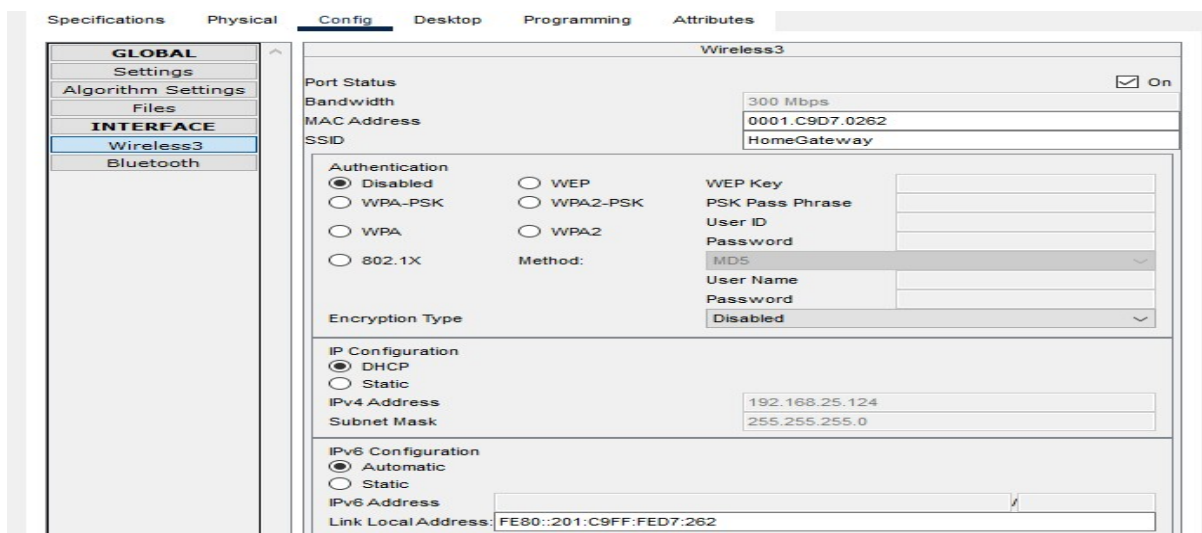


Figure 33: la carte SBC.

Détection de gaz : scénario :

Ce scénario permet d'ouvrir la fenêtre et le ventilateur si le taux de CO2 est supérieur à la norme sinon les objets connectés restent en positions de repos.

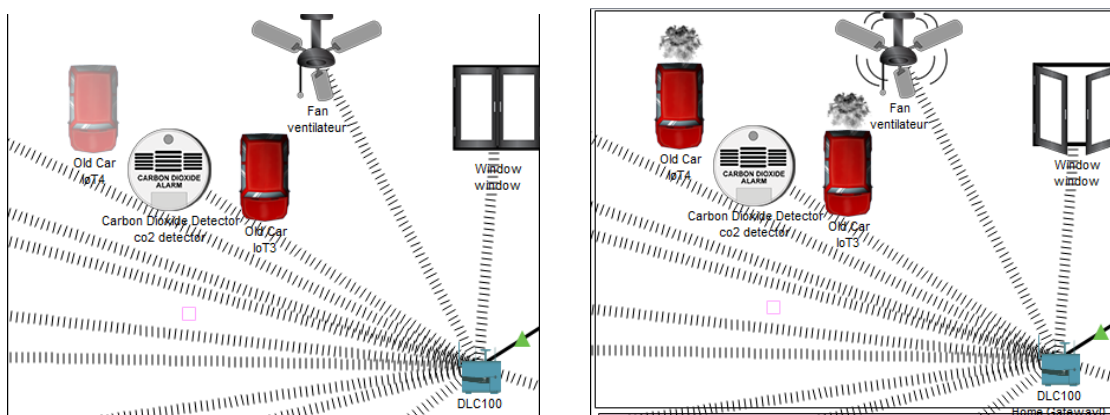


Figure 34: Détection de gaz

Connexion des périphériques iot:

- On a connecté les périphériques iot : le ventilateur, la fenêtre et le détecteur de dioxyde de carbone CO2 à la passerelle domestique (Home Gateway).
- Nous mettons SSID 'Home Gateway' dans les périphérique iot se trouve sous **Config»Wireless»SSID**.

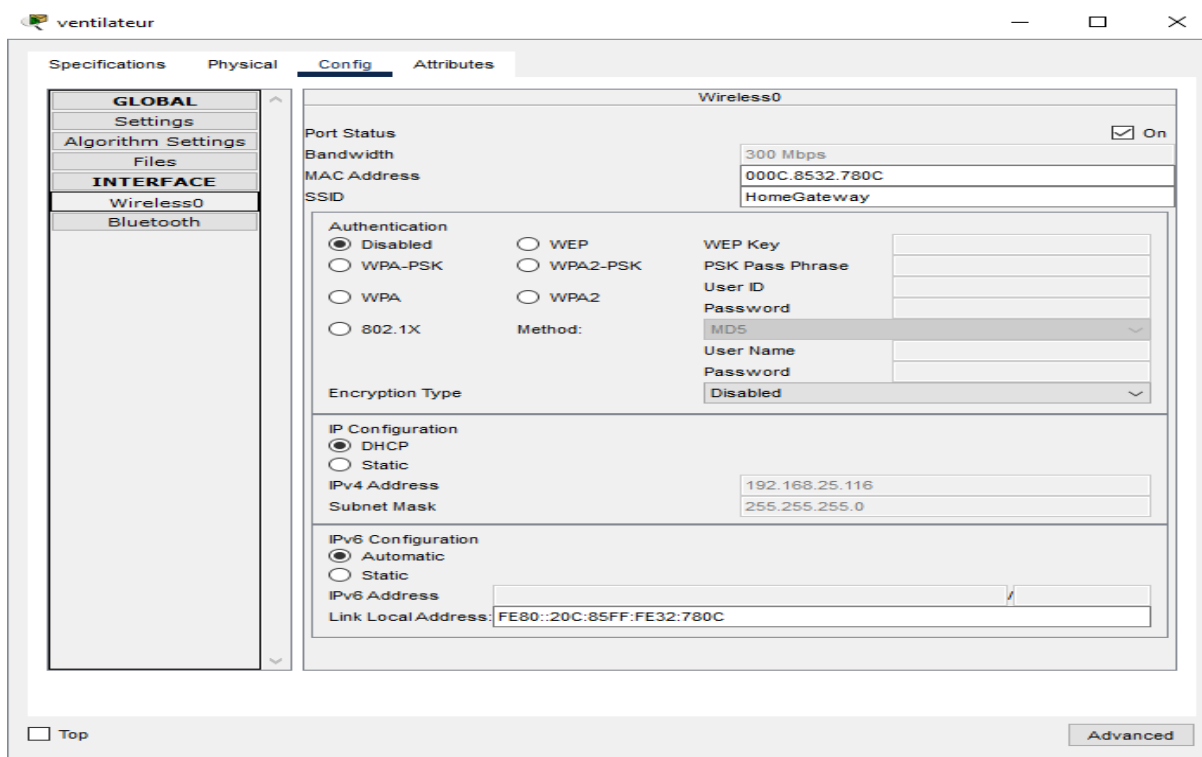
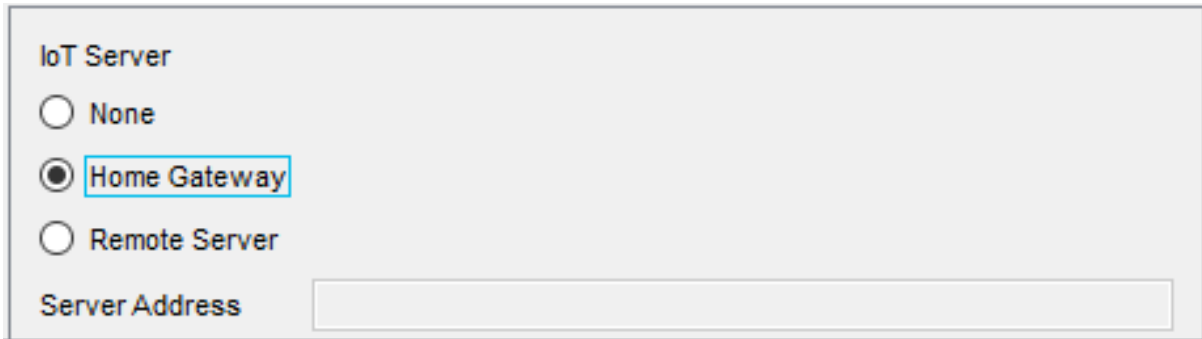


Figure 35: Connexion des peripheries iot au réseau sans fil de la passerelle et activation du DHCP.

- En change IoT Server en 'Home Gateway'.



IoT Server

None

Home Gateway

Remote Server

Server Address

Figure 36: IoT Server.

- Après on est allé vers smartphone Desktop»Web Browser» on a tapé l'adresse IP de la passerelle 'Home Gateway' et on a introduit le 'Username :admin' 'Password :admin'.



Smartphone0

Physical Config Desktop Programming Attributes

Web Browser X

< > URL http://192.168.25.1 Go Stop

Home Gateway Login

Username: admin

Password:

Submit

Figure 37: smartphone.

- Toutes les simulations IoT sont visibles depuis la page d'accueil principale de l'interface IOT se trouvant dans Home Gateway, comme le montre la figure:

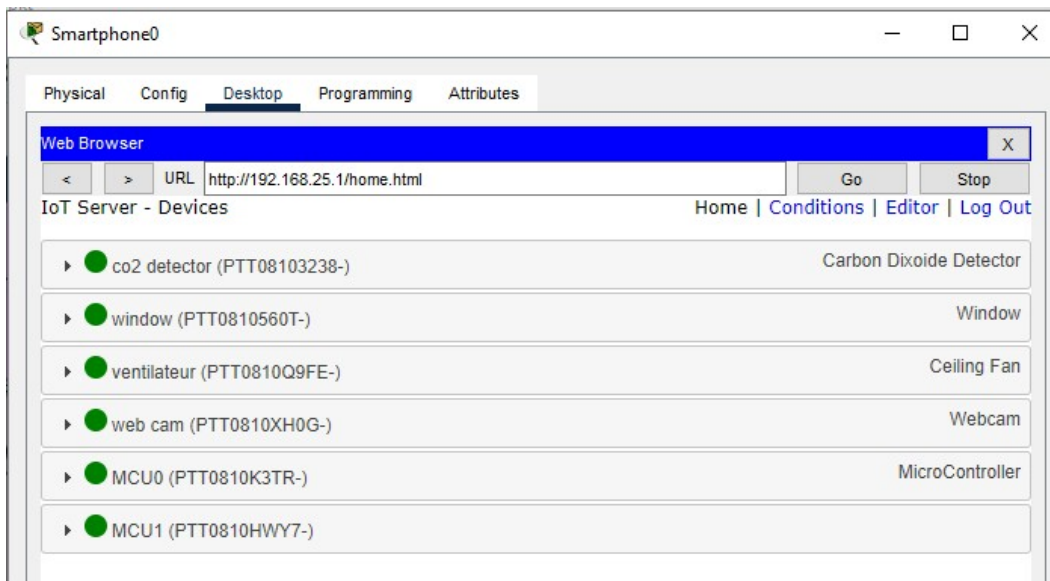


Figure 38: IoT Server.

- Voici les conditions de marche et d'arrêt de ce scénario:
Si le taux de CO2 > 0.25 alors le Ventilateur en marche, Fenêtre ouverte.
Sinon, Ventilateur à l'arrêt, Fenêtre fermée.

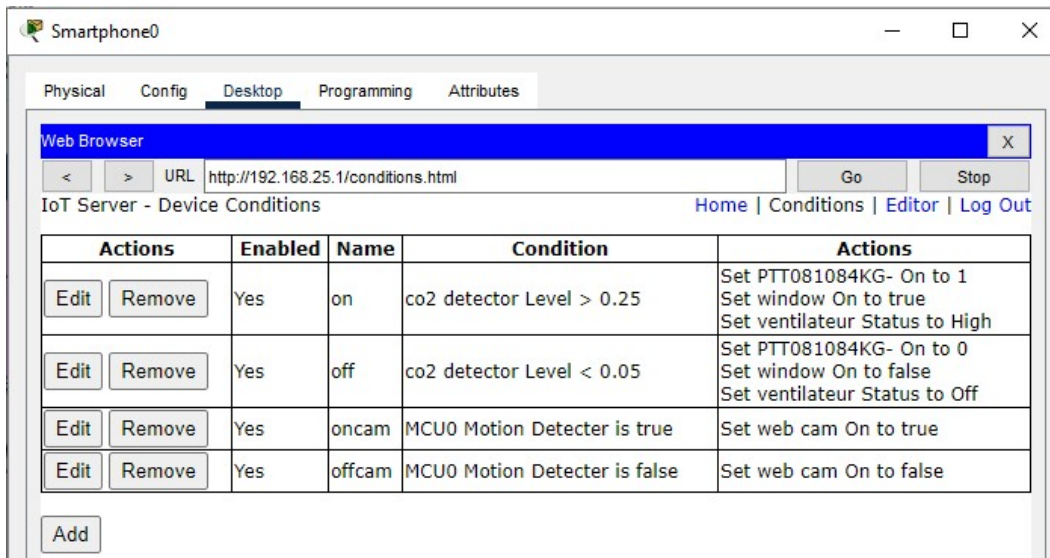


Figure 39: les conditions pour le scénario détection de CO2.

L'accès externe de la maison intelligente:

Pour l'accès externe on besoins d'ajouter d'autres équipements pour créer un réseau internet et facilite l'accès à la maison et pour gérer les iot.

on a divisé la tâche en quatre partie :configuration de réseau domestique via Gateway,configuration du réseau internet,configuration serveur central,configuration du serveur DNS et serveur IOT.

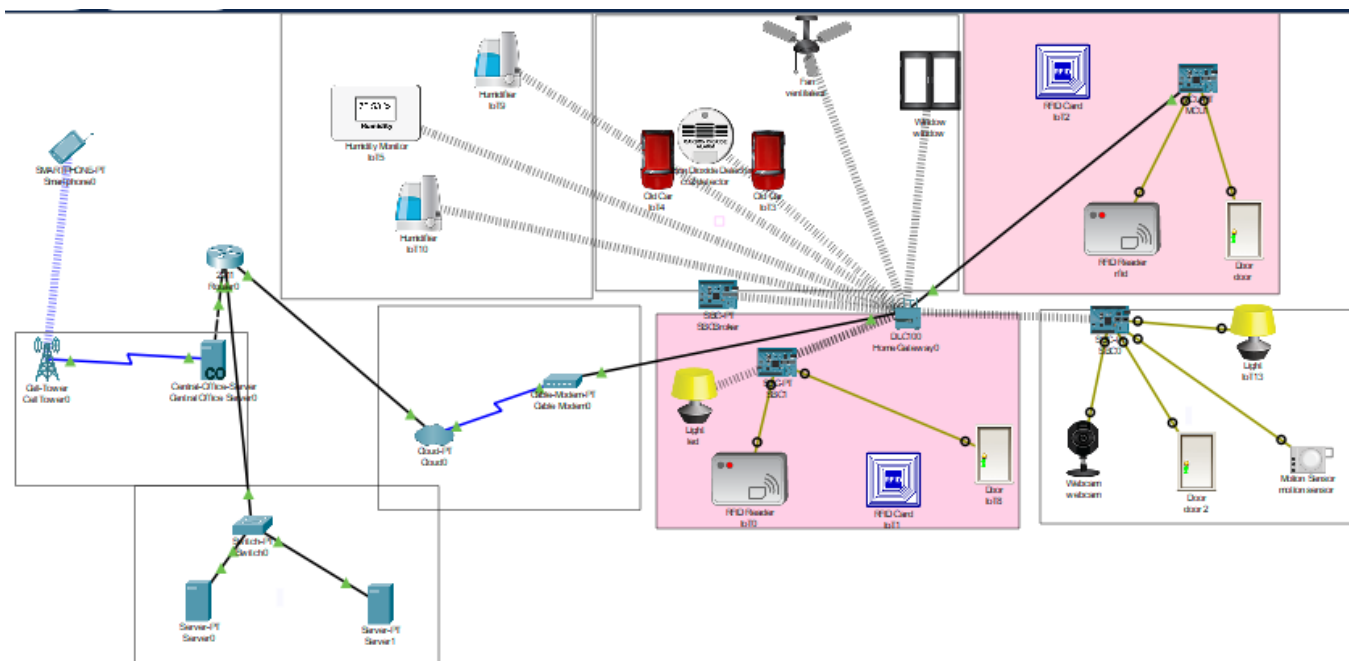


Figure 40: L'accès externe de la maison intelligente.

Les équipements utilisés pour cette réalisation:

Table 5: Equipements et Localisation dans Packet Tracer

Equipement	Localisation dans Packet Tracer
Routeur 2911	Network devices >> router >> 2911
Cell Tower	Network devices >> Wireless Devices >> Cell-Tower
Central Office Server	Network Devices >> Wireless Devices >> Central Office Server
Switch-PT	Network Devices >> Wireless Devices >> Switch-PT
Serveur-PT (serveur DNS)	End devices >> End Devices >> Serveur-PT
Cloud-PT	Network Devices >> WAN Emulation >> Cloud-PT
Cable-Modem	Network Devices >> WAN Emulation >> Cable-Modem

4 Réalisation et test

Dans ce chapitre nous illustrons les outils matériels et logiciels déployés et quelques détails pratiques

4.1 Environnement de développement

Afin de déployer notre système IoT, nous avons besoin de certains outils matériels et logiciels. Pour cela, nous avons sélectionné les modules, capteurs, logiciels et IDE suivants.

4.1.1 Environnement matériel :

On a tous d'abord, les cartes électroniques :

a) NodeMcu ESP8266 :

Le microcontrôleur ESP8266 NodeMCU est un petit circuit intégré à faible coût et à faible consommation. Alimentation électrique pour IOT (Internet des objets). Son unité centrale de traitement (CPU) 32 bits et sa connectivité Wi-Fi intégrée en font un choix populaire pour les projets de communications sans fil.

NodeMCU ESP8266 est basé sur la plate-forme de solution Espressif Systems ESP8266. WiFi abordable et performant. Compatible avec la plupart des langages de programmation y compris Lua, Arduino IDE, C++, MicroPython et plus encore.

NodeMCU ESP8266 est programmé pour communiquer avec d'autres appareils via des programmes, protocoles tels que TCP/IP, MQTT, HTTP. Il est également possible de connecter des capteurs, etc.actionneur. Parfait pour les projets d'automatisation et de domotique.

NodeMCU ESP8266 est également compatible avec une large gamme de capteurs et de modules. Capteurs de température, capteurs de lumière, relais, LCD, modules de contrôle, stockage de données, etc.

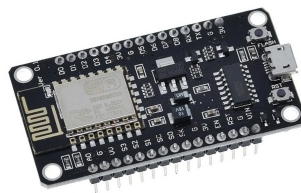


Figure 41: NodeMCU ESP8266 [4].

Caractéristiques ESP8266: [18]

- Alimentation:
 - 5 Vcc via micro-USB
 - 5 à 9 Vcc via broche Vin (régulateur intégré)
- Microcontrôleur: ESP8266

- Microprocesseur: Tensilica LX106
- Fréquence: 80 MHz
- Mémoire RAM: 64 kB
- Mémoire Flash: 96 kB
- 10 E/S digitales compatibles PWM
- Interfaces: I2C, SPI, UART
- Interface Wifi 802.11 b/g/n 2,4 GHz
- Antenne intégrée
- Température de service: -40 à 125 °C
- Dimensions: 58 x 31 x 12 mm

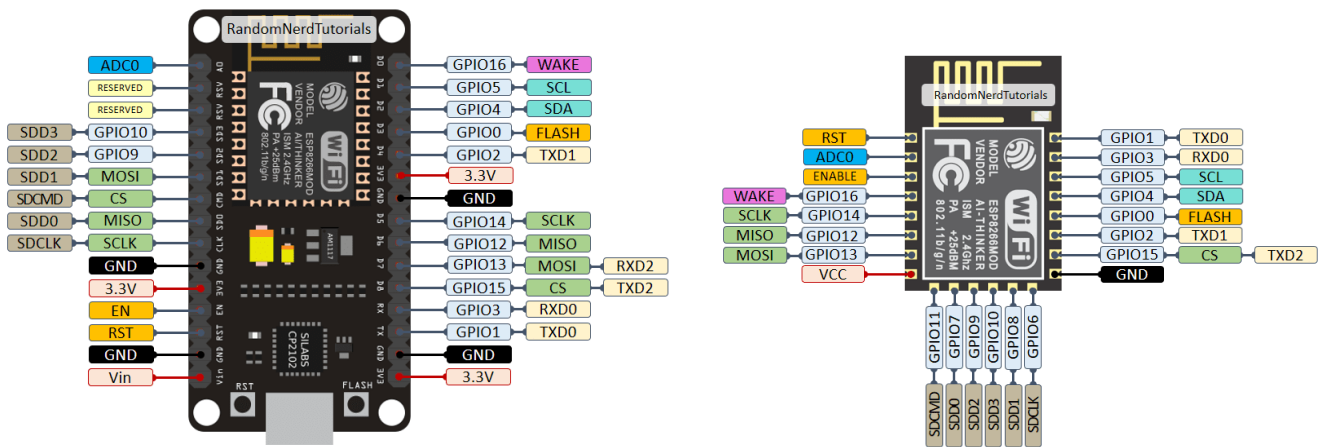


Figure 42: ESP8266 [5]

b) Arduino UNO:

La carte Arduino est une plate-forme de prototypage électronique open-source qui permet de créer des projets interactifs à base de microcontrôleurs programmables. Elle a été créée en 2003 par une équipe d'enseignants et d'étudiants italiens qui souhaitaient concevoir une carte conviviale pour leur projet de contrôle, de capteurs et d'actionneurs. Les cartes Arduino sont basées sur des microcontrôleurs Atmel AVR (le plus souvent ATmega328 P. modèle actuel), peut être programmé en C/C++ via un environnement de développement intégré (IDE) est inclus dans la carte. Arduino IDE est compatible avec Windows, Mac OS X et Linux. et permet de programmer la carte à l'aide d'une interface USB. Les cartes Arduino ont plusieurs entrées et sorties numériques et analogiques (E/S) qui vous permettent de le faire. Utilisé pour contrôler les LED, les moteurs, les capteurs, les LCD et d'autres composants électroniques. Également compatible avec divers modules et boucliers. Peut être connecté à la carte pour étendre ses fonctionnalités.

Caractéristiques Arduino UNO: [19]

- Microcontrôleur : ATmega328
- Tension de fonctionnement : 5V
- Tension d'alimentation (recommandée) : 7-12V



Figure 43: Arduino Uno [6].

- Tension d'alimentation (limites) : 6-20V
- Broches E/S numériques : 14 (dont 6 disposent d'une sortie PWM)
- Broches d'entrées analogiques : 6 (utilisables en broches E/S numériques)
- Intensité maxi disponible par broche E/S (5V) – 40 mA
- Intensité maxi disponible pour la sortie 3.3V – 50 mA
- Intensité maxi disponible pour la sortie 5V – 500 mA max
- Mémoire Programme Flash : 32 KB (ATmega328)
- Mémoire SRAM (mémoire volatile) : 2 KB (ATmega328)
- Mémoire EEPROM (mémoire non volatile) : 1 KB (ATmega328)
- Vitesse d'horloge : 16 MHz

Ensuite, nous avons :

c) Les capteurs:

Le capteur de mouvement (PIR motion sensor) :

Un capteur de mouvement PIR (capteur infrarouge passif) est un type de capteur qui détecte le mouvement en mesurant les changements de chaleur ambiante. Les capteurs PIR sont largement utilisés dans les systèmes de sécurité, l'éclairage automatique, les climatiseurs, etc.

Un capteur PIR utilise une lentille pour focaliser le rayonnement infrarouge émis par les objets dans son champ de vision. Lorsqu'un objet chaud, tel qu'un corps humain, passe devant le capteur, une partie du rayonnement infrarouge est absorbée, créant une différence de température. Des capteurs détectent cette différence de température et déclenchent des actions telles que l'allumage des lumières ou le déclenchement d'une alarme.

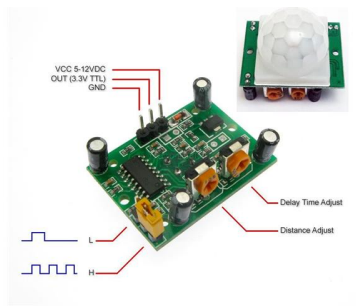


Figure 44: Le capteur de mouvement [7].

Le capteur de gaz MQ2 :

Le capteur de gaz MQ2 est un type de capteur de gaz qui peut détecter différents types de gaz, Substances inflammables et toxiques telles que le méthane, le propane, l'alcool, l'hydrogène et les vapeurs. Le capteur MQ2 est généralement utilisé dans les systèmes de détection de gaz domestiques et industriels, détecteur de gaz potentiellement dangereux.

Le fonctionnement du capteur de gaz MQ2 est basé sur les variations de la résistance électrique du capteur de gaz. Au fur et à mesure que le gaz s'accumule à la surface, il devient un matériau plus vulnérable. Le capteur est équipé d'un élément chauffant. Il chauffe les matériaux sensibles à la température de fonctionnement et chauffe les circuits électroniques. Mesure les changements de résistance des matériaux sensibles. Lorsque le capteur MQ2 détecte la présence de gaz, un signal électrique est envoyé au microcontrôleur, Ou il peut être envoyé à un système de contrôle pour déclencher des alarmes et des actions correctives.

Il est important de noter que le capteur MQ2 ne mesure pas la concentration exacte du gaz détecté. C'est plutôt la présence ou l'absence de gaz. La concentration de gaz peut être estimée comme suit : Cependant, la résistance mesurée dépend de nombreux facteurs tels que : B. Température, humidité, etc. Qualité de l'air ambiant. Par conséquent, il est important de calibrer le capteur pour obtenir des résultats précis .

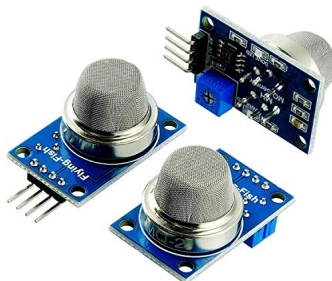


Figure 45: Le capteur de gaz MQ2 [8].

Caractéristiques du capteur de gaz MQ2 [20]:

- Alimentation : 5V.
- Sortie : une analogique et une numérique.
- Plage de détection : 300 10000 PPM.
- Gaz détectés : GPL, fumée, alcool, propane, monoxyde de carbone.
- Le temps de réponse : 1 s.

- Dimensions : 32x22x27mm.
- La durée de vie : 2 à 5 ans dans des conditions normales d'utilisation.

Capteur d'humidité-Température DHT22 :

Le capteur DHT22 est un capteur numérique de température et d'humidité. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer l'air ambiant et génère un signal numérique sur la broche de données [21].

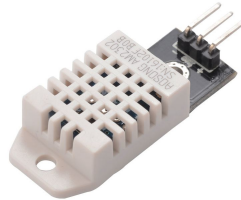


Figure 46: Capteur d'humidité-Température DHT22 [9].

Caractéristiques capteur d'humidité-Température DHT22 [9] :

- Alimentation: 3,3 à 6 Vcc
- Consommation maxi: 1,5 mA
- Consommation au repos: 50 μ A
- Plage de mesure: - température: -40 à +80 °C - humidité: 0 à 100
- Précision: - température: $\pm 0,5$ °C - humidité: ± 2
- Dimensions: 25 x 15 x 9 mm

Capteur LDR :

Un capteur à résistance dépendante de la lumière (LDR), également appelé photorésistance, est un appareil électronique utilisé pour détecter ou mesurer l'intensité de la lumière ambiante. Il est sensible à la quantité de lumière entrante et a une résistance qui change en fonction de l'intensité de cette lumière.

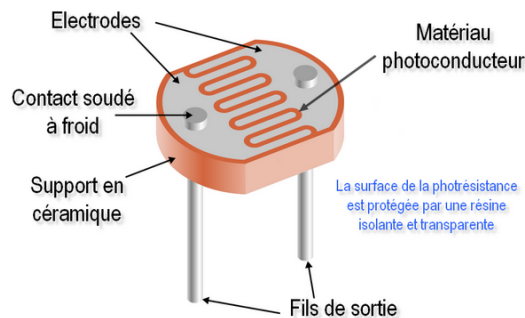


Figure 47: Capteur De Lumière LDR [10].

Enfin, nous avons :

d) Les actionneur:

Module relais 5V :

Un relais one channel (un canal) est un dispositif électronique utilisé pour contrôler l'activation ou la désactivation d'un circuit électrique à haute tension à l'aide d'un signal de faible tension.

Un relais one channel 5V 250V, est conçu pour fonctionner avec une tension de commande de 5 volts et peut commuter une tension maximale de 250 volts.

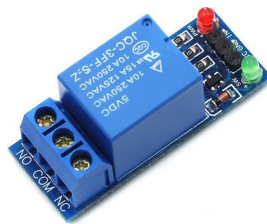


Figure 48: relais [11].

Servomoteur :

Un servomoteur est un système asservi par un retour de mesure .Il pourra automatiquement corriger sa position en fonction de celle-ci , ce que signifie qu'il sera capable de conserver sa position ou de la corriger si elle venait a bouger dans le temps [22].

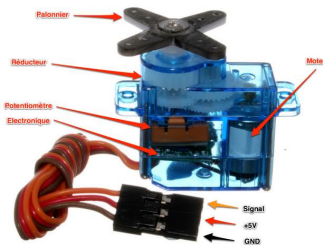


Figure 49: Un servomoteur [12].

keypad 3x4 :

Un clavier 3x4, également connu sous le nom de clavier matriciel 3x4, est un clavier numérique utilisé pour la saisie de données dans de nombreux appareils électroniques tels que les systèmes de sécurité, les systèmes de contrôle d'accès, etc.

Il y a trois rangées horizontalement et quatre colonnes verticalement, et un total de 12 touches forment une matrice. Ces touches sont généralement numérotées de 1 à 9, avec des touches

supplémentaires 0, "*" et "#". Selon l'utilisation, les touches peuvent également avoir des symboles spéciaux ou des fonctions supplémentaires.



Figure 50: clavier 3x4 [13]

4.1.2 Environnement logiciel :

a) **Arduino ide :**

L'IDE Arduino, également connu sous le nom d'Environnement de Développement Intégré Arduino, est un logiciel open source destiné à la programmation de la carte Arduino. Cet outil permet aux utilisateurs de créer, compiler et télécharger facilement du code sur leur carte Arduino. Grâce à son interface graphique conviviale, il est particulièrement apprécié des débutants en électronique et en programmation.

L'IDE Arduino est compatible avec le langage de programmation Wiring, une version simplifiée de C++ spécialement adaptée aux projets Arduino.

L'interface du logiciel Arduino se présente de la façon suivante



Figure 51: Arduino ide

- 1) Permet de vérifier le programme.
- 2) Compiler et envoyer le programme vers la carte.
- 3) Créer un nouveau fichier
- 4) Charger un programme existant
- 5) Sauvegarder le programme en cours

Le menu File dispose d'un certain nombre de choses qui vont être très utiles :



Figure 52: Arduino ide

b) **Home Assistant :**

Home Assistant est une plate-forme domotique open source qui aide à contrôler et à automatiser divers appareils et services dans une maison intelligente.

Il prend en charge une variété de périphériques et de protocoles de communication, ce qui en fait une solution flexible et personnalisable pour les utilisateurs [23].

Il est principalement développé en Python, dispose également d'un système de configuration basé sur YAML (Yet Another Markup Language). Il permet de définir les appareils, les automatisations, les scènes, les groupes, etc., en utilisant une syntaxe simple et facile à comprendre.

Il peut être installé sur différents types de matériels, notamment :

- **Raspberry Pi :** C'est l'une des options les plus populaires pour installer Home Assistant.
- **Ordinateur personnel :** Home Assistant peut être installé sur un ordinateur personnel sous Windows, macOS ou Linux.
- **Service dédié.**
- **Machines virtuelles :** Vous pouvez créer une machine virtuelle (VM) et installer Home Assistant dessus. Telles que VirtualBox, VMware et Proxmox peuvent être utilisées.

c) **Oracle VM VirtualBox :**

VirtualBox est le logiciel de virtualisation gratuit, open source et multiplateforme d'Oracle. Celui-ci permet d'héberger une ou plusieurs machines virtuelles, avec des systèmes d'exploitation différents [24].

d) **Visual Studio Code :**

Visual Studio Code est un éditeur de code simplifié, gratuit et open source développé par Microsoft. Il est disponible pour Windows, macOS et Linux. Il fournit aux développeurs un environnement de développement intégré et des outils pour piloter des projets techniques de l'édition à la construction en passant par le débogage [25].

e) **PlatformIO :**

PlatformIO est un écosystème open source dédié au développement IoT. PlatformIO IDE est un environnement de développement C/C++ pour les systèmes embarqués. Il est multiplate-forme (Windows, Mac, GNU/Linux) et fournit une extension à votre éditeur de texte existant (Atom ou VSCode) [26, 27].

f) **Python :**

Python est un langage de programmation open source, multiplate-forme et orienté objet [28].

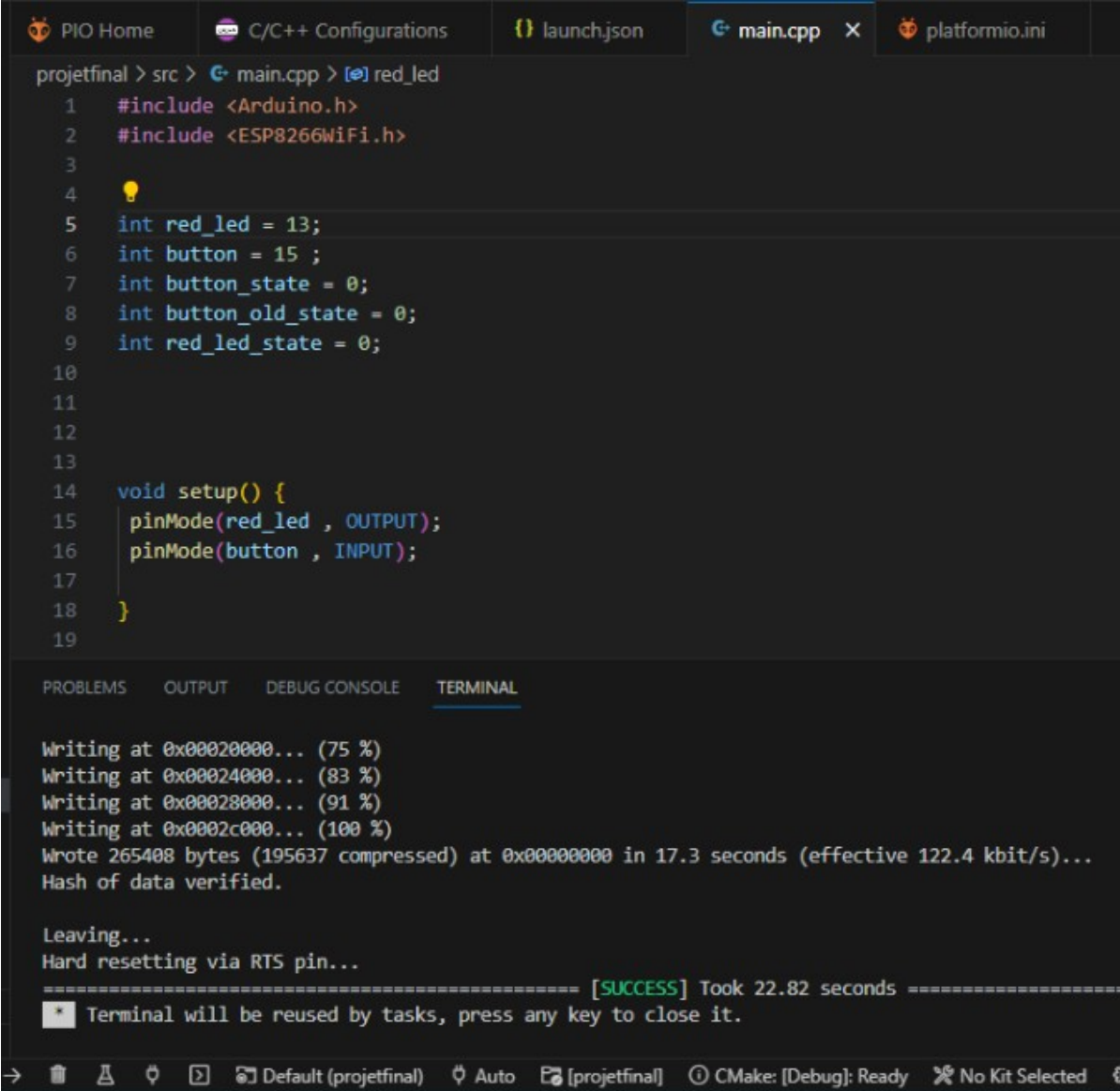
g) **Yaml :**

YAML est Data Sequence Format, YAML signifie un autre langage de balisage, et pour d'autres, c'est une abréviation fréquente pour YAML Ain't Markup Language (YAML n'est pas un langage de balisage), soulignant que YAML est utilisé pour représenter des données à la place de documents, et sa conception est assez lisible et compréhensible [29].

4.2 Réalisation :

a) programmation la carte esp8266:

voilà la programmation de la carte esp8266 pour allumer et éteindre une led à l'aide de PlatformIO.



```
pio Home C/C++ Configurations launch.json main.cpp x platformio.ini
projetfinal > src > main.cpp > red_led
1 #include <Arduino.h>
2 #include <ESP8266WiFi.h>
3
4
5 int red_led = 13;
6 int button = 15 ;
7 int button_state = 0;
8 int button_old_state = 0;
9 int red_led_state = 0;
10
11
12
13
14 void setup() {
15     pinMode(red_led , OUTPUT);
16     pinMode(button , INPUT);
17 }
18
19

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Writing at 0x00020000... (75 %)
Writing at 0x00024000... (83 %)
Writing at 0x00028000... (91 %)
Writing at 0x0002c000... (100 %)
Wrote 265408 bytes (195637 compressed) at 0x00000000 in 17.3 seconds (effective 122.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 22.82 seconds =====
Terminal will be reused by tasks, press any key to close it.
```

Figure 53: programme platformIO.

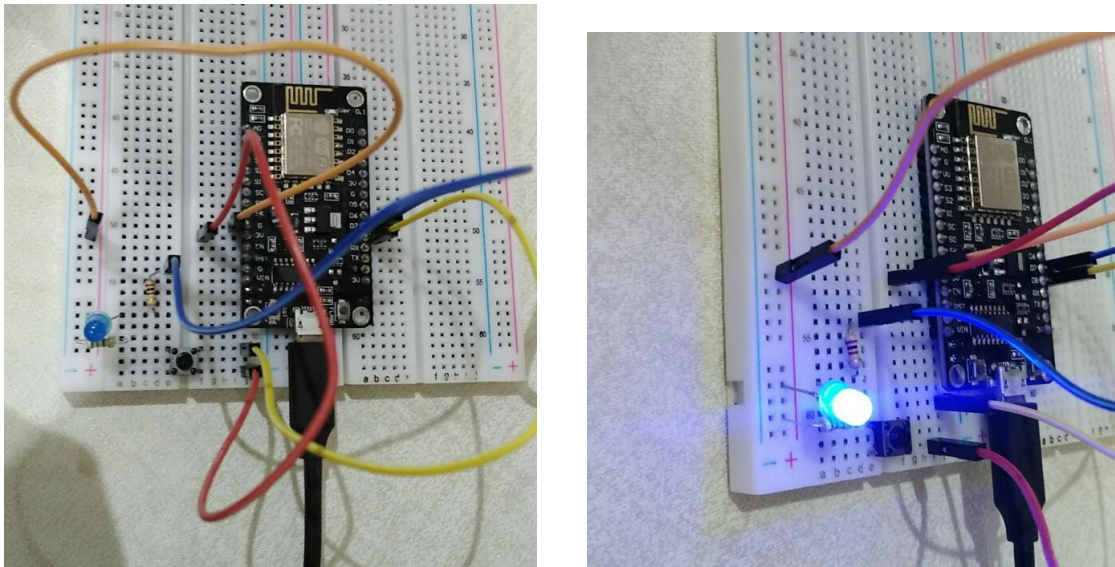


Figure 54: test une LED.

b) programmation Yaml:

Voilà le programme yaml dans le fichier configuration de Home Assistant. Ce fichier est utilisé pour configurer et personnaliser divers aspects.

On a ajouté des composants spécifiques au système, tels que la lumière, des capteurs.

```

/config/configuration.yaml
1 |
2 # Configure a default setup of Home Assistant (frontend, api, etc)-
3 default_config:-
4   -
5 # Uncomment this if you are using SSL/TLS, running in Docker container, etc.-
6 # http:-
7 #   base_url: example.duckdns.org:8123-
8   -
9 # Light-
10 light:-
11   - platform: mqtt
12     name: "Light"
13     command_topic: "office/rgb1/light/switch"
14     state_topic: "inlight"
15     payload_on: "0"
16     payload_off: "1"
17     retain: true
18   -
19 # Sensors-
20 sensor:-
21   - platform: mqtt
22     name: "Temperature"
23     state_topic: "outTemperature"
24     unit_of_measurement: '°C'
25   - platform: mqtt
26     name: "Window"
27     state_topic: "outWindow"
28     icon: mdi>window-closed
29   - platform: mqtt
30     name: "Motion"
31     state_topic: "outMotion"
32     icon: mdi>motion-sensor
33   - platform: mqtt
34     name: "Photores"
35     state_topic: "outPhotores"
36     icon: mdi>theme-light-dark

```

Figure 55: Programme yaml.

```
38 # MQTT broker-
39 mqtt:-
40 -
41 # Text to speech-
42 tts:-
43   -- platform: google_translate-
44 -
45 group: !include groups.yaml-
46 automation: !include automations.yaml-
47 script: !include scripts.yaml-
48 ↵
```



Figure 56: Programme yaml.

5 conclusion

La nouvelle ère de l'informatique se distingue par l'émergence de nouveaux domaines qui reflètent les tendances technologiques actuelles, parmi ces domaines figurent l'Internet des objets. L'Internet des objets est un concept qui se réfère à la connexion et à la communication entre des objets physiques grâce à des capteurs, des logiciels et des réseaux. Il offre de nombreuses possibilités dans divers domaines, tels que la domotique, la gestion des villes intelligentes, la santé... Grâce à cette technologie, les objets peuvent devenir plus intelligents, plus autonomes et plus efficaces dans leurs fonctionnalités, ce qui ouvre de nouvelles perspectives pour l'innovation et l'amélioration de notre quotidien.

Tout au long de cette mémoire, nous avons présenté les concepts de base de l'Internet des Objets, sa structure, ses caractéristiques, et architecture d'un réseaux IOT, ainsi que vu les protocoles de communication et fait un comparatif "HTTP, CoAP, MQTT, DDS, XMPP, AMQP" et on a détaillé dans mqtt c est un protocole de messagerie les communications entre appareils connectés.

Ensuite, nous avons fait une conception et analyse, et simulation à l'aide tikerCAD qui permet de tester et observer le fonctionnement de hardware et la programmation des prototypes d'IOT virtuel en temps réel, et de simuler un réseau d'IOT domestique sous Cisco Packet Tracer.

Enfin, nous montrons les outils matériels et logiciels qui nous ont aidé et nous montrons aussi le résultat final de notre mise en œuvre.

Lors de la réalisation de ce projet, nous avons rencontré quelques difficultés, principalement le manque de matériel, et le respect de la date de réalisation a constitué le grand défi que nous avons rencontré.

Dans le futur et comme perspectives de ce travail, nous envisageons d'améliorer et de commercialiser notre produit final, de nous développer et de découvrir d'autres produits domotiques. Et de développer une application Web pour un système domotique.

Bibliographie

- [1] “Modèles d’architectures de l’internet des objets,” <https://blog.octo.com/en/modeles-architectures-internet-des-objets/>, 2011.
- [2] “Domotique et maison connectée : le dossier complet,” <https://www.mesconseils.info/domotique-et-maison-connectee-le-dossier-complet/>.
- [3] “Protocole mqtt,” <https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/pedagogiques/13195/13195-1-protocole-mqtt.docx>.
- [4] “Nodemcu esp8266,” <https://einstronic.com/product/nodemcu-esp8266-ch340-narrow/>.
- [5] “Référence de brochage esp8266 : Quelles broches gpio devriez-vous utiliser ?” <https://www.raspberryme.com/referance-de-brochage-esp8266-queelles-broches-gpio-devriez-vous-utiliser/>.
- [6] “Arduino uno r3,” <https://www.azurefilm.com/en/arduino-uno-r3>.
- [7] “Pir sensor,” <https://nilambaraelectronics.com/product/pir-sensor-hc-sr501/>.
- [8] “Mq-2 capteur de gaz,” <https://www.amazon.fr/AZDelivery-Capteur-qualit%C3%A9-Arduino-compris/dp/B07RX5SXQ6>.
- [9] “Capteur d’humidité-température dht22,” http://www.wikidebrouillard.org/wiki/Item:Capteur_d%27humidit%C3%A9-Temp%C3%A9rature_DHT22.
- [10] “Module détecteur de lumière à ldr,” <https://www.framboise314.fr/scratch-raspberry-pi-composants/module-detecteur-de-lumiere-a-ldr/>.
- [11] “Relais module 5v,” <https://www.bol.com/nl/nl/p/otronic-relais-module-5v-ideaal-voor-arduino-esp32-esp8266-raspberry-pi/9300000007382505/>.
- [12] “le servomoteur sg90,” <https://www.upesy.fr/blogs/tutorials/esp32-servo-motor-with-arduino-code-sg90>.
- [13] “clavier 4x3,” <https://forum.arduino.cc/t/partage-clavier-4x3/339251>.
- [14] “Definition iot (internet of things),” <https://actualiteinformatique.fr/internet-of-things-iot/quest-ce-que-iot-internet-of-things-internet-des-objets>.
- [15] “Objets connectés : exemples d’appareil et usages en france,” <https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1440688-objets-connectes-exemples-d-appareil-et-usages-en-france/>.
- [16] “Qu’est-ce que le protocole mqtt,” 2021. [Online]. Available: <https://www.ip-systemes.com/details-comprendre-le-protocole+mqtt-795.html#>
- [17] “Packet tracer,” http://projet.eu.org/pedago/sin/tutos/packet_tracer.pdf.
- [18] “Esp8266: le module wifi pour arduino,” <https://www.hwlibre.com/fr/esp8266/>.
- [19] “Carte arduino uno caractéristiques, description,” <https://arduino-france.site/description-arduino-uno/>.
- [20] “Solution iot pour la détection des fuites de gaz à domicile,” <https://di.univ-blida.dz/xmlui/bitstream/handle/123456789/6695/TCHERAK%20Amine%20et%20OUZNADJI%20Younes.pdf?sequence=1&isAllowed=y>.
- [21] “Capteur de température et humidité dht22,” <https://info.blaisepascal.fr/rpi-capteur-de-temperature-et-humidite-dht22/>.
- [22] “Comment fonctionne les servomoteur?” <https://conseilsjudicieux.com/comment-fonctionne-les-servomoteur/>.
- [23] “Home assistant : l’outil domotique pour les gouverner tous,” https://www.frandroid.com/produits-android/maison-connectee/826727_home-assistant-loutil-domotique-pour-les-gouverner-tous.
- [24] “Oracle vm virtualbox,” <https://www.blogdumoderateur.com/tools/oracle-vm-virtualbox/>.
- [25] “Visual studio code,” <https://www.blogdumoderateur.com/tools/visual-studio-code/>.
- [26] “Vscodé et platformio,” <https://dle-dev.com/index.php/2020/10/02/vscodé-et-platformio/>.

- [27] “Platformio ide,” <https://docs.platformio.org/en/latest/integration/ide/pioide.html>.
- [28] “Python : qu’est-ce que c’est ?” <https://www.futura-sciences.com/tech/definitions/informatique-python-19349/>.
- [29] “Yaml, qu’est-ce que c’est ?” <https://www.redhat.com/fr/topics/automation/what-is-yaml>.
- [30] “Les 4 domaines d’application de la domotique.” [Online]. Available: <https://www.tafsquare.com/fr/blog/les-domaines-d-application-de-la-domotique/>
- [31] “Tot les protocoles de communication pour les réseaux sans-fil et filaires,” <https://blog.engineering.publicissapient.fr/2018/08/29/iot-les-protocoles-de-communication-pour-les-reseaux-sans-fil-et-filaires-comment-choisir/>.
- [32] “Les 4 briques d’une solution iot à succès,” <https://www.matooma.com/fr/s-informer/actualites-iot-m2m/architecture-solution-iot>.
- [33] A. Oussama, “Développement d’un système d’iot (internet of things) dans le cadre de smart university.” <http://bib.univ-ueb.dz:8080/jspui/bitstream/123456789/10499/1/Memoire%20finale%20kouah%20Version%20finale%20v3.pdf>, 2020.
- [34] A. Benfattoum, “Quels protocoles applicatifs pour l’internet des objets ?” <https://www.frugalprototype.com/quels-protocoles-applicatifs-pour-linternet-des-objets/>, décembre 2015.
- [35] “What is mqtt ?” https://www.sharetechnote.com/html/IoT/App_Protocol_MQTT.html.
- [36] “Unified modeling language (uml),” <https://helios2.mi.parisdescartes.fr/~bouzy/Doc/UML-NotesCours.pdf>.
- [37] “Formation esp8266 / esp32 - découverte,” <https://formation.tech/formations/internet-des-objets/esp8266-esp32-decouverte>.
- [38] “Module wifi node mcu- esp-8266,” <https://youpilab.com/components/product/module-wifi-node-mcu-esp-8266-12>.