المدرسة الوطنية للعلوم التطبيقية - مراكش
ECOLE NATIONALE DES SCIENCES APPLIQUEES - MARRAKECH
جامعة القاضي عياض
UNIVERSITÉ CADI AYYAD

Ecole Nationale des Sciences Appliquées de Marrakech

Réseaux, Systèmes et Services Programmables

Projet de Fin de Semestre

Modélisation et détection d'objets connectées dans un smart building

Réalisé par:

DIHIA Hiba
YAMMOUN Mariam
TIHADI Naoual
BENKEROUM Ikram
KIMISSI Imane

Encadré par:

Mme El HALOUI Loubna

Année universitaire : 2024-2025

Dedication

Nous dédions ce projet à toutes les personnes qui œuvrent pour un avenir où la technologie et l'innovation améliorent notre qualité de vie. À tous ceux qui travaillent sans relâche pour rendre nos espaces de vie plus intelligents, durables et connectés, votre passion et votre engagement sont une source d'inspiration. Que ce projet soit une contribution collective à la réalisation de cette vision partagée.

Remerciements

Nous souhaitons exprimer notre sincère gratitude envers plusieurs personnes qui ont rendu ce projet possible.

Tout d'abord, nous tenons à remercier notre encadrante, Loubna Elhaloui, pour son soutien indéfectible, sa patience et ses conseils précieux tout au long de cette aventure. Votre expertise et votre vision ont été des atouts essentiels dans la réalisation de ce projet. Vous avez su nous guider et nous encourager à donner le meilleur de nous-mêmes.

Nous remercions également nos collègues de groupe pour leur collaboration et leur esprit d'équipe. Chacun d'entre vous a apporté des idées, des compétences et une motivation qui ont enrichi notre travail commun. Votre engagement et votre enthousiasme ont été des éléments clés dans l'aboutissement de ce projet.

Enfin, nous souhaitons exprimer notre gratitude envers nos familles et amis pour leur soutien moral et leur compréhension durant les périodes de travail intense. Votre présence à nos côtés a été une source de motivation constante.

Merci à toutes et à tous pour votre contribution à cette réalisation collective.

Résumé

Le projet "Modélisation et Détection des Objets Connectés dans un Smart Building" vise à intégrer les technologies de l'Internet des Objets (IoT) pour optimiser la gestion des ressources et améliorer la sécurité et le confort des occupants dans les bâtiments intelligents. En utilisant des outils comme Cisco Packet Tracer et la programmation Python, le projet propose de modéliser et simuler les interactions entre divers dispositifs connectés, tels que des capteurs et des thermostats.

La problématique centrale est de concevoir un système efficace pour détecter et gérer automatiquement ces objets connectés, permettant ainsi une surveillance en temps réel et une gestion proactive des ressources. Les enjeux de ce projet incluent une meilleure efficacité énergétique, une sécurité renforcée et une amélioration du confort des occupants, tout en contribuant à la réduction de l'empreinte carbone des bâtiments.

Les objectifs principaux consistent à créer un modèle virtuel des dispositifs IoT, développer un mécanisme de détection automatique et simuler divers scénarios d'utilisation pour analyser les interactions des objets connectés. En conclusion, ce projet démontre comment les technologies IoT peuvent transformer les bâtiments en environnements intelligents et efficaces, tout en soulignant l'importance de la sécurité des données et des systèmes pour l'avenir des maisons connectées.

Abstract

The project "Modeling and Detection of Connected Objects in a Smart Building" aims to integrate Internet of Things (IoT) technologies to optimize resource management and enhance the security and comfort of occupants in smart buildings. By utilizing tools such as Cisco Packet Tracer and Python programming, the project proposes to model and simulate the interactions between various connected devices, such as sensors and thermostats.

The central issue is to design an effective system for automatically detecting and managing these connected objects, thereby enabling real-time monitoring and proactive resource management. The stakes of this project include improved energy efficiency, enhanced security, and increased occupant comfort, all while contributing to the reduction of the carbon footprint of buildings.

The main objectives consist of creating a virtual model of IoT devices, developing an automatic detection mechanism, and simulating various usage scenarios to analyze the interactions of connected objects. In conclusion, this project demonstrates how IoT technologies can transform buildings into intelligent and efficient environments, while highlighting the importance of data and system security for the future of connected homes.

Sommaire

Liste des Figures

1 Introduction

Sommaire

1.1 Introduction générale

Dans un contexte de transformation numérique, l'intégration des technologies de l'Internet des Objets (IoT) dans les bâtiments est devenue essentielle pour répondre aux défis modernes d'efficacité énergétique, de sécurité, et de confort des occupants. Un bâtiment intelligent, ou Smart Building, représente une nouvelle approche où divers dispositifs connectés, comme les capteurs, caméras, et thermostats, communiquent pour optimiser la gestion des ressources. Cette interconnexion permet une surveillance en temps réel et une automatisation de multiples fonctions du bâtiment, de l'éclairage à la gestion de la température, en passant par la sécurité des espaces.
Le projet Modélisation et Détection des Objets Connectés dans un Smart Building vise à modéliser et simuler ces interactions grâce à des outils comme Cisco Packet Tracer et la programmation Python. Il s'agit de créer un environnement où chaque objet connecté est identifié et surveillé de manière automatique, afin d'optimiser la gestion énergétique et de renforcer les capacités de sécurité du bâtiment. En concevant des scénarios basés

sur des cas d'utilisation concrets, le projet permet d'analyser comment les dispositifs IoT répondent aux changements environnementaux et besoins des utilisateurs.
Ce projet s'inscrit dans un domaine d'innovation stratégique qui redéfinit le rôle des bâtiments dans nos sociétés et explore les moyens de rendre les infrastructures plus intelligentes, durables, et adaptées aux défis contemporains.

1.2 Contexte de projet

1.2.1 Problématique et Justification

La problématique centrale de ce projet est la suivante : Comment concevoir un system efficace for modéliser and detecter automatiquement les objets connected in an intelligent building, to optimize the management of resources and to ensure a surveillance in real time?

- **J**ustification du Projet : Ce projet est pertinent dans un contexte où la gestion énergétique et la sécurité des bâtiments sont des priorités croissantes. L'adoption de technologies IoT dans les bâtiments intelligents permet d'atteindre des niveaux d'efficacité inédits, tout en offrant un meilleur confort et une sécurité renforcée pour les occupants.

- **I**ntérêt Technologique et Sociétal : Sur le plan technique, ce projet met en œuvre des avancées technologiques significatives en matière d'automatisation et de gestion en temps réel. Sur le plan sociétal, il contribue à la réduction de l'empreinte carbone des bâtiments, à l'optimisation des ressources, et à la création de nouveaux standards de confort et de sécurité pour les bâtiments modernes

1.3 Présentation de projet

Le projet "Modélisation et Détection des Objets Connectés dans un Smart Building" vise à modéliser et simuler ces interactions grâce à des outils comme Cisco Packet Tracer et la programmation Python. Il s'agit de créer un environnement où chaque objet connecté est identifié et surveillé de manière automatique, afin d'optimiser la gestion énergétique et de renforcer les capacités de sécurité du bâtiment. En concevant des scénarios basés sur des cas d'utilisation concrets, le projet permet d'analyser comment les dispositifs IoT répondent aux changements environnementaux et besoins des utilisateurs.

1.3.1 Contexte et Enjeux

- **L'Internet des Objets (IoT)** connaît une croissance exponentielle, avec une application de plus en plus répandue dans divers secteurs, notamment celui des bâtiments. Les dispositifs IoT permettent de transformer les espaces physiques en environnements intelligents, où les objets connectés interagissent et communiquent pour optimiser l'utilisation des ressources. Dans le secteur des bâtiments, cela implique une gestion plus efficace de l'énergie, une sécurité accrue, et un confort optimisé pour les occupants.
- **Les Enjeux des Smart Buildings :**Les Smart Buildings répondent aux enjeux modernes de la gestion des bâtiments, en permettant une meilleure efficacité énergétique, une sécurité renforcée, et un confort des occupants optimisé. Cette approche présente des avantages écologiques et économiques en réduisant la consommation énergétique et en favorisant une utilisation plus durable des ressources.
- **Importance de la Modélisation et de la Détection Automatique :** La modélisation des dispositifs IoT dans les Smart Buildings permet une gestion proactive et une détection automatique des anomalies. Cela se traduit par une surveillance en temps réel de l'état du bâtiment et une capacité à anticiper les défaillances potentielles, garantissant ainsi un fonctionnement optimal et sécurisé.

1.3.2 Objectifs du Projet

Ce projet a pour objectif principal de concevoir et de mettre en œuvre un système capable de modéliser et de détecter automatiquement les objets connectés au sein d'un Smart Building. Cette modélisation doit permettre de gérer ces objets de manière autonome et efficace, en vue d'optimiser la gestion des ressources du bâtiment.

- **Modélisation des Objets Connectés:** Utiliser Cisco Packet Tracer pour créer un modèle virtuel des dispositifs connectés, en représentant fidèlement leur position, interaction et comportement dans le bâtiment.
- **Détection Automatique des Dispositifs IoT :** Développer un mécanisme de détection qui identifie et enregistre les objets connectés en temps réel, permettant ainsi une gestion proactive de ces dispositifs.
- **Création de Scénarios IoT Dynamiques:** Mettre en place différents scénarios IoT pour simuler des situations courantes dans un bâtiment intelligent, comme la gestion de la température, l'éclairage en fonction de la présence, et d'autres conditions environnementales.

2

Concepts et Technologies Utilisées

Sommaire

2.1 Introduction

Les bâtiments intelligents, ou Smart Buildings, incarnent une révolution dans la gestion et l'utilisation des infrastructures modernes grâce à l'intégration des technologies de

l'Internet des Objets (IoT). Ces édifices connectés visent à améliorer le confort, la sécurité et l'efficacité énergétique, tout en offrant des solutions innovantes pour optimiser les ressources et anticiper les besoins des occupants.
Ce chapitre explore les concepts fondamentaux des bâtiments intelligents et leur interaction avec l'IoT. Il s'intéresse également aux outils technologiques qui permettent de modéliser, simuler et gérer ces environnements complexes, tels que Cisco Packet Tracer et Python. L'objectif est de comprendre comment ces technologies contribuent à concevoir des systèmes connectés capables d'interagir harmonieusement dans un bâtiment intelligent.
En étudiant ces concepts et technologies, nous posons les bases nécessaires pour appréhender les défis et opportunités liés à l'intégration de l'IoT dans des environnements intelligents.

2.2 Bâtiments Intelligents et IoT

2.2.1 Définition d'un Smart Building

Qu'est-ce qu'un bâtiment intelligent? Un bâtiment intelligent, ou Smart Building, désigne un édifice qui intègre des technologies avancées pour automatiser, surveiller, et optimiser diverses fonctions de gestion et de maintenance. Ces bâtiments sont conçus pour améliorer la performance énergétique, maximiser le confort des occupants, et renforcer la sécurité, tout en utilisant des systèmes de gestion intégrés. Grâce à l'Internet des Objets (IoT), les bâtiments intelligents sont dotés de dispositifs connectés, tels que des capteurs, caméras, et thermostats, qui communiquent en temps réel pour adapter les fonctionnalités du bâtiment aux besoins des utilisateurs et aux conditions environnementales.
L'objectif d'un Smart Building est d'offrir une gestion autonome des ressources et des services, minimisant l'intervention humaine et optimisant la consommation d'énergie et le confort. Ces bâtiments constituent une composante essentielle des villes intelligentes en permettant une infrastructure durable et connectée, capable de répondre aux défis contemporains de gestion des ressources.

Principes fondamentaux et caractéristiques des Smart Buildings: Les bâtiments intelligents reposent sur plusieurs principes et caractéristiques clés, qui permettent une gestion avancée et proactive des infrastructures :

1. **Automatisation des Systèmes :** Les bâtiments intelligents utilisent des technologies d'automatisation pour contrôler et optimiser en temps réel des systèmes essentiels, comme le chauffage, la ventilation, la climatisation (HVAC), l'éclairage, et la sécurité. Ces systèmes s'ajustent automatiquement en fonction de divers paramètres, comme l'occupation, la température extérieure, ou la luminosité, afin de maintenir un environnement optimal.

2. **Interconnectivité et Communication entre Dispositifs :** Au cœur des bâtiments intelligents se trouve un réseau de dispositifs interconnectés (IoT) capables de communiquer entre eux. Cette connectivité permet aux différents systèmes et objets connectés de travailler de manière coordonnée, d'échanger des données, et de s'adapter aux conditions changeantes.

3. **Collecte et Analyse des Données :** Les bâtiments intelligents reposent sur la collecte massive de données provenant de leurs dispositifs connectés. Ces données sont ensuite analysées pour optimiser les opérations, prédire les besoins d'entretien, et identifier les opportunités d'économie d'énergie. Par exemple, en analysant les données d'occupation, le bâtiment peut adapter l'éclairage ou la température pour réduire la consommation d'énergie lorsque les zones sont inoccupées.

4. **Amélioration de l'Efficacité Énergétique :** Un des objectifs majeurs des Smart Buildings est de réduire la consommation d'énergie et d'améliorer l'efficacité énergétique. En exploitant les données en temps réel, le bâtiment peut ajuster les systèmes de chauffage, de ventilation et de climatisation, gérer les consommations électriques, et utiliser des énergies renouvelables de manière optimale.

5. **Confort et Expérience des Occupants :** Les bâtiments intelligents visent à offrir un environnement de travail ou de vie plus confortable pour les occupants. Grâce à des systèmes intelligents, le bâtiment peut ajuster la température, l'éclairage, et même la qualité de l'air en fonction des préférences des occupants. Cette personnalisation contribue à un meilleur confort et à une productivité accrue.

6. **Sécurité et Surveillance Avancées :** Les bâtiments intelligents intègrent des systèmes de sécurité automatisés, comme la détection de mouvements, la surveillance vidéo, et la reconnaissance faciale, pour renforcer la sécurité des occupants. En cas d'urgence, ces bâtiments peuvent coordonner des actions automatiques, telles que l'évacuation ou l'alerte des services de sécurité, pour une réactivité accrue.

7. **Gestion Durable et Responsable des Ressources :** Enfin, les Smart Buildings contribuent à un développement durable en réduisant l'empreinte carbone et en minimisant le gaspillage de ressources. Ils intègrent souvent des solutions d'énergie renouvelable, comme le solaire ou la géothermie, et permettent un suivi précis de l'utilisation des ressources en eau, en électricité, et en chauffage.

2.2.2 Avantages et Enjeux de l'IoT dans les Bâtiments Intelligents

1. **Amélioration de l'Efficacité Énergétique:**

   L'Internet des Objets (IoT) joue un rôle fondamental dans le développement des bâtiments intelligents en permettant une gestion connectée et automatisée des différents systèmes au sein de ces infrastructures. Cette section explore les principaux avantages offerts par l'IoT dans les Smart Buildings, notamment en termes d'efficacité énergétique, de sécurité, de confort des occupants, ainsi que d'optimisation des coûts de maintenance et de gestion en se basant sur:

   - **Suivi et Réduction de la Consommation Énergétique :** Les capteurs IoT détectent la présence des occupants et adaptent l'éclairage ou la climatisation en conséquence, réduisant ainsi le gaspillage d'énergie dans les zones inoccupées.
   - **Utilisation Optimale des Énergies Renouvelables :** Grâce à des capteurs météorologiques et des systèmes intelligents de gestion d'énergie, les bâtiments peuvent maximiser l'utilisation des énergies renouvelables, comme le solaire et l'éolien, en fonction des conditions extérieures et de la demande interne.
   - **Analyse des Données pour l'Efficacité :** Les données recueillies par les dispositifs IoT sont analysées pour identifier des opportunités d'économie. Par exemple, le système peut détecter une augmentation inhabituelle de la consommation d'énergie et déclencher une inspection pour vérifier l'intégrité des équipements.

2. **Augmentation de la Sécurité et du Confort des Occupants:**

   Les dispositifs IoT permettent une surveillance continue et une gestion proactive de la sécurité, contribuant ainsi à un environnement plus sûr et plus confortable pour les occupants.

   - **Sécurité Renforcée :** Les bâtiments intelligents peuvent intégrer des caméras, détecteurs de mouvement, et systèmes d'alerte connectés pour surveiller en permanence les activités à l'intérieur et autour du bâtiment. Ces dispositifs IoT permettent des réponses immédiates en cas d'incident, comme une intrusion ou un incendie.
   - **Amélioration du Confort des Occupants :** L'IoT permet de personnaliser l'environnement intérieur pour répondre aux préférences individuelles. Par exemple, les occupants peuvent contrôler la température, l'éclairage, et même la qualité de l'air dans leurs espaces via des applications connectées, offrant ainsi un confort personnalisé.

- **Gestion de l'Environnement Intérieur :** Les capteurs de qualité de l'air, de température, et d'humidité permettent d'assurer un environnement intérieur sain, en maintenant automatiquement les niveaux optimaux pour le confort et la santé des occupants.

3. **Optimisation des Coûts de Maintenance et de Gestion** Les bâtiments intelligents connectés à l'IoT peuvent réduire significativement les coûts liés à la maintenance et à la gestion, en adoptant une approche proactive et prédictive.

   - **Maintenance Prédictive :** Grâce aux données en temps réel fournies par les capteurs, le bâtiment peut détecter les signes d'usure ou de défaillance des équipements avant qu'ils ne tombent en panne. Cela permet de planifier la maintenance de manière proactive, évitant ainsi les interruptions coûteuses et prolongeant la durée de vie des équipements.
   - **Réduction des Interventions Manuelles :** L'automatisation des tâches de gestion quotidienne, comme l'ajustement de l'éclairage ou de la température, réduit les interventions humaines, ce qui limite les erreurs et optimise les ressources en personnel.
   - **Réduction des Coûts d'Opération :** Les analyses de données IoT permettent d'optimiser les coûts d'opération en ajustant les processus et en identifiant les dépenses superflues. Par exemple, les données peuvent révéler des pics de consommation d'énergie et suggérer des stratégies pour les aplanir, réduisant ainsi les coûts associés aux heures de pointe.

2.2.3 Enjeux et Défis de l'Intégration de l'IoT dans les Bâtiments

L'Internet des Objets (IoT) dans les bâtiments intelligents apporte une série de défis qui doivent être relevés pour assurer une intégration sécurisée, efficace, et interopérable. Cette section aborde les principaux enjeux liés à la sécurité et à la confidentialité, la compatibilité entre dispositifs IoT, et la complexité de la gestion des vastes quantités de données collectées.

1. **Problématiques de Sécurité et de Confidentialité** La sécurité et la confidentialité sont des préoccupations majeures dans les systèmes IoT, en particulier dans les bâtiments intelligents où une grande quantité de données sensibles (informations sur les occupants, sécurité du bâtiment, données de surveillance, etc.) est collectée et échangée.

- **Risque de Piratage et Cyberattaques** : Les dispositifs IoT sont souvent vulnérables aux attaques, car beaucoup d'entre eux ne sont pas conçus avec des mécanismes de sécurité avancés. Un piratage de ces dispositifs pourrait permettre un accès non autorisé aux réseaux du bâtiment, compromettant non seulement la sécurité des données, mais aussi la sécurité physique des occupants.
- **Protection des Données Personnelles :** La collecte de données personnelles sur les habitudes des occupants, leur localisation, et leurs préférences soulève des questions de confidentialité. Pour respecter les réglementations comme le RGPD (Règlement Général sur la Protection des Données), il est crucial d'assurer un contrôle strict sur les données collectées et leur utilisation, en veillant à l'anonymisation et à la protection contre les accès non autorisés.
- **Gestion des Accès :** Il est essentiel de mettre en place des systèmes d'authentification robuste et de gestion des accès pour les dispositifs IoT afin de limiter les accès à des utilisateurs autorisés uniquement et prévenir les tentatives de manipulation des données.

2. **Compatibilité et Interopérabilité des Dispositifs** L'un des défis majeurs de l'IoT dans les bâtiments intelligents réside dans la diversité des équipements et des protocoles de communication, qui peuvent compliquer leur interopérabilité.

   - **Hétérogénéité des Protocoles de Communication :** Les dispositifs IoT utilisent divers protocoles (Wi-Fi, Zigbee, Z-Wave, Bluetooth, etc.), rendant difficile l'intégration des appareils provenant de différents fabricants dans un même système. Cette fragmentation des protocoles peut limiter l'efficacité de la communication entre les dispositifs, nuisant à la fluidité de l'ensemble du système.
   - **Standardisation Insuffisante :** En raison de l'absence de normes communes pour les dispositifs IoT, l'interopérabilité peut être restreinte, obligeant les entreprises à se tourner vers des solutions propriétaires. Cela augmente les coûts d'intégration et peut limiter la capacité d'un bâtiment intelligent à évoluer avec de nouveaux dispositifs à l'avenir.
   - **Maintenance et Mise à Jour :** La gestion d'un grand nombre de dispositifs, avec des configurations et des mises à jour distinctes, peut devenir complexe et fastidieuse. Les mises à jour logicielles sont essentielles pour corriger les vulnérabilités, mais elles nécessitent souvent des efforts de compatibilité pour que le système reste opérationnel.

3. **Complexité de la Gestion des Données Collectées** L'IoT génère un flux massif de données, et la gestion, l'analyse, et la sécurisation de ces données posent des défis importants pour les bâtiments intelligents.

    - **Volume et Vélocité des Données :** La collecte continue de données en temps réel (température, mouvements, consommation d'énergie, etc.) entraîne une grande quantité de données à traiter et à stocker. Gérer ce volume tout en maintenant la rapidité du traitement pour fournir des réponses en temps réel est un défi majeur.
    - **Stockage et Analyse des Données :** Le stockage des données nécessite des infrastructures adaptées (bases de données locales ou cloud) et peut engendrer des coûts importants. De plus, pour tirer des informations utiles, il est nécessaire d'utiliser des techniques avancées d'analyse de données, souvent basées sur le Big Data et l'Intelligence Artificielle.
    - **Protection des Données Sensibles :** La gestion des données doit aussi inclure des solutions de chiffrement et de gestion d'accès afin de protéger les informations critiques et éviter les fuites de données. L'analyse de données doit également respecter la confidentialité des utilisateurs, ce qui peut être complexe lorsqu'on traite de grands ensembles de données.

2.3 Outils et Technologies

2.3.1 Cisco Packet Tracer

Figure 2.1: Packet Tracer

Cisco Packet Tracer est un outil de simulation de réseaux développé par Cisco, largement utilisé pour l'enseignement et l'expérimentation en réseau informatique. Il permet de créer des environnements de simulation réalistes pour tester la connectivité entre des dispositifs et comprendre le comportement des réseaux dans des situations variées, comme celles rencontrées dans les projets IoT. L'une des fonctionnalités clés de Packet Tracer pour

les projets de bâtiments intelligents est sa capacité à simuler des objets connectés en intégrant divers capteurs et actionneurs, permettant de représenter des scénarios de Smart Building dans un environnement de réseau. Les utilisateurs peuvent ainsi programmer des interactions entre les objets, en visualiser le fonctionnement, et anticiper les problématiques liées à la communication des dispositifs IoT dans un réseau.

2.3.1.1 Processus de Modélisation des Dispositifs IoT dans Packet Tracer

Le processus de modélisation des dispositifs IoT dans Cisco Packet Tracer est une étape cruciale pour créer une simulation réaliste des objets connectés dans un Smart Building. Ce processus se déroule en plusieurs étapes, chacune étant essentielle pour garantir que les objets IoT fonctionnent de manière optimale dans un environnement connecté.

1. **Création de l'architecture du réseau IoT :** La première étape consiste à définir l'architecture du réseau. Cela implique de déterminer la topologie du réseau en incluant les dispositifs IoT nécessaires, tels que les capteurs de température, de mouvement, les caméras de sécurité, etc. Il est essentiel d'attribuer des adresses IP uniques à chaque dispositif, permettant ainsi une communication fluide et efficace entre eux. Cette étape permet de visualiser l'ensemble du réseau et de planifier comment les objets IoT interagiront dans le cadre du Smart Building.

2. **Modélisation des objets IoT :** Dans cette étape, on sélectionne les objets à simuler (capteurs, actionneurs, caméras, etc.) et on les configure pour qu'ils fonctionnent selon des scénarios spécifiques. Par exemple, pour un capteur de présence, on définit son comportement pour qu'il puisse activer automatiquement un éclairage lorsqu'une personne entre dans une pièce. Les objets peuvent également être configurés pour envoyer des données à un serveur ou effectuer des actions spécifiques en fonction des événements dans l'environnement.

3. **Définition des scénarios d'interconnexion:** Une fois que les objets sont configurés, il est nécessaire de créer des scénarios d'interconnexion. Ces scénarios définissent comment les dispositifs IoT interagiront les uns avec les autres. Par exemple, un capteur de température pourrait ajuster le thermostat ou les fenêtres en fonction des conditions climatiques détectées, ou des capteurs de présence pourraient gérer les éclairages et la sécurité. Les scénarios permettent de simuler des événements réels et d'étudier les réponses des dispositifs IoT.

4. **Simulation et test :** Après avoir configuré les objets et défini les scénarios, l'étape suivante consiste à lancer la simulation. Cette phase permet d'observer le comportement de tous les dispositifs dans l'environnement simulé. Les tests

peuvent inclure la vérification de la connectivité entre les objets IoT, l'analyse de la consommation d'énergie, la gestion des données générées, et l'identification de problèmes potentiels comme des latences ou des déconnexions de certains objets. L'objectif est de s'assurer que le système fonctionne correctement avant de le mettre en œuvre dans un environnement réel.

5. **Analyse des données et optimisation :** Après la simulation, les données collectées permettent d'analyser les performances du réseau IoT. Cela inclut la vérification de l'efficacité des processus automatisés, la gestion de l'énergie, ainsi que la réponse des objets aux scénarios définis. En fonction des résultats de l'analyse, des ajustements peuvent être apportés pour améliorer la réactivité des dispositifs, optimiser la consommation d'énergie, ou corriger d'éventuels problèmes détectés dans le réseau.

2.3.1.2 Simulation des Objets Connectés dans Packet Tracer

Dans le cadre de ce projet, Cisco Packet Tracer sera utilisé pour modéliser et simuler divers objets connectés présents dans un bâtiment intelligent, tels que des capteurs de température, de présence, et des caméras de surveillance, Cette simulation permet de :

- **Configurer et Interconnecter les Dispositifs IoT :** Grâce à Packet Tracer, il est possible de configurer chaque objet IoT avec des adresses IP, de définir leur comportement et de simuler des événements pour tester leurs réactions. Par exemple, on peut simuler l'activation automatique des lumières lorsqu'un capteur détecte la présence d'un occupant.

- **Analyser les Interactions et la Connectivité :** En simulant la connectivité entre les dispositifs, on peut visualiser comment les objets IoT interagissent entre eux et détecter des problèmes de réseau, comme des latences ou des interruptions.

- **Créer des Scénarios Personnalisés :** Packet Tracer permet de créer des scénarios de fonctionnement, comme la gestion énergétique (ex. : ajustement de la température en fonction de la présence des occupants), ce qui est essentiel pour concevoir des Smart Buildings efficaces et automatisés.

2.3.2 Python pour la Gestion des Données IoT

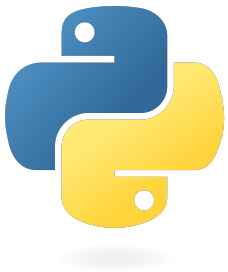

Figure 2.2: Python

Python est devenu un langage de programmation incontournable dans les projets liés à l'Internet des Objets (IoT), en raison de sa simplicité, de sa flexibilité et de son vaste écosystème de bibliothèques. Dans le contexte des bâtiments intelligents, Python offre une panoplie d'outils pour traiter, analyser et visualiser les données générées par les dispositifs IoT.

2.3.2.1 Introduction à Python et Ses Avantages dans le Projet

- **Pourquoi utiliser Python pour les projets IoT ?**

  Python se distingue par sa courbe d'apprentissage rapide, ce qui en fait un choix idéal pour les développeurs travaillant sur des projets IoT complexes. Sa syntaxe claire et lisible permet une implémentation rapide des fonctionnalités nécessaires pour collecter, analyser et interpréter les données transmises par des capteurs et autres objets connectés.

- **Caractéristiques de Python qui facilitent l'analyse et l'automatisation:**

- Support étendu pour les protocoles IoT tels que MQTT et CoAP.

- Richesse de bibliothèques pour le traitement des données (Pandas, NumPy).

- Capacités d'automatisation pour orchestrer les processus, comme l'analyse en temps réel ou la génération d'alertes.

- Intégration facile avec des outils IoT et des bases de données pour le stockage des informations.

2.3.2.2 Utilisation de Python pour le traitement et la visualisation des données collectées

Python se distingue par sa facilité d'apprentissage et sa capacité à traiter de grandes quantités de données générées par les capteurs IoT. Son utilisation permet une implémentation rapide et efficace des fonctionnalités nécessaires, telles que :

- **La collecte des données :** Python peut être utilisé pour collecter les données en temps réel à partir des capteurs IoT, grâce à des protocoles comme MQTT ou HTTP.
- **Le traitement des données :** Grâce à des outils comme Pandas et NumPy, qui offrent des moyens robustes pour structurer, nettoyer, et analyser les informations collectées.
- **La visualisation des données:** à l'aide de bibliothèques comme Matplotlib et Seaborn, permettant de présenter les données sous forme de graphiques ou de tableaux de bord compréhensibles.
- **L'automatisation des processus :** Python permet d'écrire des scripts pour automatiser des actions, comme déclencher des alertes en cas de défaillance ou ajuster les paramètres d'un bâtiment (température, éclairage, etc.).

2.4 Scénarios IoT et Leur Importance

Les scénarios IoT jouent un rôle fondamental dans la conception et l'évaluation des bâtiments intelligents. Ils permettent de modéliser et de simuler des cas pratiques d'utilisation des objets connectés, en tenant compte des besoins des occupants et des objectifs d'optimisation des performances du bâtiment.

2.4.1 Développement de Scénarios IoT dans un Smart Building

Pour développer des scénarios IoT, une approche méthodique est nécessaire. Celle-ci inclut l'identification des besoins spécifiques du bâtiment et de ses utilisateurs, la sélection des dispositifs IoT pertinents, et la définition des interactions entre ces dispositifs. Par exemple, un scénario peut consister à gérer automatiquement l'éclairage en fonction de la présence détectée dans une pièce, ou à ajuster la température pour maximiser le confort tout en minimisant la consommation énergétique. Ces scénarios peuvent également inclure des fonctions avancées telles que la surveillance de la sécurité via des caméras ou la gestion à distance des appareils connectés.

2.4.2 Importance des Scénarios dans l'Évaluation des Bâtiments Intelligents

2.4.2.1 Simulation des Conditions Réelles de Fonctionnement

Les scénarios IoT permettent de recréer des conditions réelles pour tester le comportement des dispositifs connectés et évaluer leur efficacité. En simulant des interactions comme la réaction des capteurs de température à des variations soudaines ou la gestion d'alertes de sécurité, il devient possible d'anticiper les besoins des occupants et d'améliorer la conception des systèmes IoT pour répondre à ces besoins. Cela offre également une base pour détecter des problèmes potentiels, comme des retards dans les communications ou des pannes de dispositifs.

2.4.2.2 Optimisation de la Gestion Énergétique et de la Sécurité

En intégrant des scénarios IoT, il est possible de réduire significativement la consommation énergétique du bâtiment grâce à des actions automatisées telles que l'extinction des lumières dans les zones inoccupées ou l'ajustement dynamique de la climatisation. De plus, ces scénarios renforcent la sécurité des occupants en surveillant constamment les accès et en émettant des alertes en cas de comportement suspect. Ainsi, les scénarios IoT ne se limitent pas à un rôle technique : ils apportent une réelle valeur ajoutée en améliorant l'efficacité globale et la qualité de vie dans les bâtiments intelligents.

3

Modélisation des Objets Connectés dans un Smart Building

Sommaire

3.1 Introduction

Avant d'explorer les composants majeurs et la structure du réseau de notre maison intelligente, il est crucial de comprendre l'architecture globale de la solution que nous avons conçue. Notre approche intègre une combinaison réfléchie de composants et de connexions réseau pour garantir une gestion fluide et intelligente des équipements domestiques. Les détails suivants offrent un aperçu clair de la manière dont notre solution est construite pour créer une maison intelligente fonctionnelle et interconnectée.

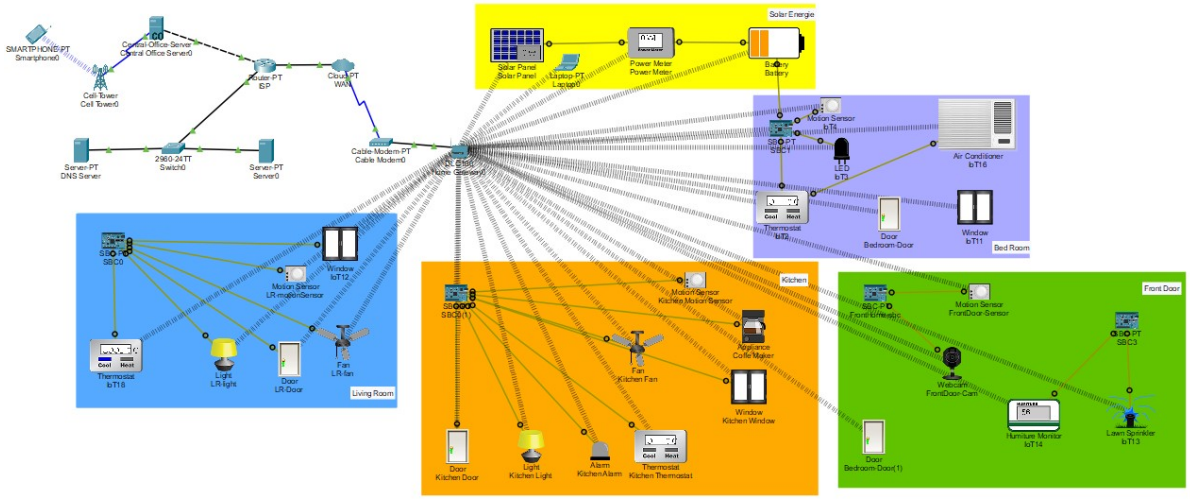

Figure 3.1: Architecture globale

Cette figure illustre visuellement comment les composants clés, tels que les capteurs, les actionneurs, le serveur DNS, le routeur, et les dispositifs de contrôle, interagissent au sein de notre maison intelligente. Elle offre une représentation graphique de la manière dont ces éléments convergent pour créer un écosystème domestique intelligent.

3.2 Équipements Utilisés

- Motion sensor

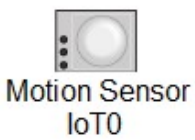

Figure 3.2: Motion sensor

Un capteur de mouvement est un dispositif virtuel utilisé pour détecter des mouvements dans un espace donné. Il peut être intégré dans des scénarios de domotique

pour automatiser des actions, comme l'allumage d'une lumière lorsque quelqu'un entre dans une pièce. Ce choix est pertinent pour simuler des systèmes intelligents en réseau et tester l'automatisation dans un environnement virtuel sans matériel physique.

- SBC (Single-Board Computer)

Figure 3.3: SBC (Single-Board Computer)

Un SBC (Single-Board Computer) est un ordinateur compact et autonome, souvent utilisé pour des projets de réseautage ou d'automatisation. Il permet de simuler des applications IoT ou des systèmes embarqués dans des réseaux virtuels. Le choix d'un SBC est justifié par sa capacité à exécuter des applications légères tout en restant économique, facilitant ainsi les tests dans des environnements réseau sans nécessiter un matériel complexe.

- Routeur-PT ISP

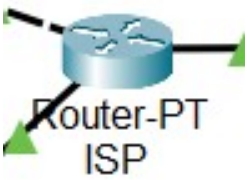

Figure 3.4: Routeur-PT ISP

Le routeur-PT ISP simule un routeur de fournisseur d'accès à Internet (ISP), permettant de connecter un réseau local à l'Internet simulé. Ce dispositif est choisi pour sa capacité à gérer l'acheminement du trafic entre un réseau privé et l'extérieur, en appliquant des protocoles de routage comme RIP ou OSPF et en fournissant une passerelle pour l'accès à des ressources externes, tout en simulant une connexion Internet dans des environnements de test.

- Switch

Figure 3.5: Switch

Un switch est un appareil de réseau qui connecte plusieurs dispositifs au sein d'un même réseau local (LAN) et dirige les paquets de données vers les ports corrects. Le switch est choisi pour sa capacité à créer des réseaux locaux évolutifs, à optimiser la communication interne grâce à la commutation de paquets, et à réduire la congestion du réseau.

- Serveur PT

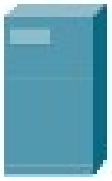

Figure 3.6: Serveur PT

Un serveur PT est une machine virtuelle dans Cisco Packet Tracer qui simule des services réseau essentiels tels que DHCP, DNS, HTTP, FTP, et d'autres services de gestion du réseau. Dans Packet Tracer, un serveur PT est utilisé pour tester et configurer des scénarios de réseau afin de simuler des environnements réels. Le serveur PT est particulièrement adapté pour offrir des services comme l'attribution d'adresses IP via DHCP ou la résolution de noms de domaine avec DNS. Ce choix est pertinent pour simuler la gestion centralisée des ressources et tester l'intégration des services dans un réseau sans matériel physique.

- Serveur PT DNS

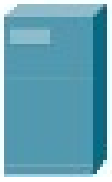

Figure 3.7: Serveur PT

Un serveur PT DNS est un serveur configuré pour fournir des services de résolution de noms de domaine. Ce serveur permet aux clients du réseau de convertir des noms de domaine en adresses IP, facilitant ainsi la navigation et la communication sur le réseau. Le choix d'un serveur DNS PT est justifié par sa capacité à simuler un environnement réseau réel où la résolution des noms de domaine est cruciale pour le bon fonctionnement des services et applications. Ce serveur DNS est essentiel pour tester l'intégration et la gestion des noms dans des scénarios réseau complexes.

- Central-Office Server

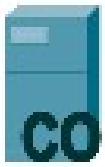

Figure 3.8: Central-Office Server

Le Central-Office Server dans Packet Tracer représente un serveur dans un bureau central ou un datacenter qui fournit des services réseau à un réseau étendu ou des succursales distantes. Il est utilisé pour centraliser les services, comme les applications, la gestion des utilisateurs ou les bases de données, et est souvent choisi pour sa capacité à offrir une gestion centralisée des ressources et des services dans des scénarios de réseau d'entreprise ou de campus.

- Cloud-PT

Figure 3.9: Cloud-PT

Un Cloud-PT dans Packet Tracer représente une connexion à un service cloud, qui peut simuler l'accès à des ressources distantes telles que des serveurs de stockage ou des services en ligne. Le choix du Cloud-PT est essentiel pour simuler l'accès aux services cloud dans un environnement réseau, permettant de tester la connectivité et la gestion des ressources via des plateformes distantes tout en offrant une vue réaliste des architectures modernes d'infrastructure IT.

- Cable-Modem-PT

Figure 3.10: Cable-Modem-PT

Un Cable-Modem-PT est un appareil qui simule la connexion à Internet via une connexion câble. Ce dispositif est utilisé pour fournir une connectivité Internet à un réseau local en utilisant une connexion par câble coaxial. Il est essentiel dans Packet Tracer pour simuler les connexions réseau haute vitesse utilisées dans les foyers ou petites entreprises, offrant une interface entre un réseau local et l'Internet.

- DLC Home Gateway

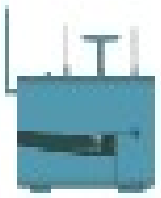

Figure 3.11: DLC Home Gateway

Un DLC Home Gateway est un dispositif qui agit comme une passerelle entre le réseau domestique et le fournisseur de services Internet (ISP). Dans Packet Tracer, il est utilisé pour gérer les connexions réseau dans un environnement domestique, comme les accès à Internet, les configurations de routeur domestique, et la gestion des périphériques locaux. Il est choisi pour simuler un réseau domestique ou un réseau de petite entreprise, en offrant une passerelle pour la communication avec l'Internet.

- Smartphone-PT

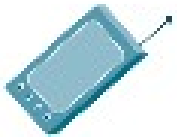

Figure 3.12: Smartphone-PT

Un Smartphone-PT est un dispositif mobile qui simule un smartphone dans Packet Tracer. Il permet de tester et de simuler l'utilisation de services réseau sur un appareil mobile, comme la navigation sur Internet, l'accès aux applications ou la gestion des connexions sans fil. Le Smartphone-PT est utilisé pour valider la connectivité sans fil dans des environnements réseaux et tester l'interaction des périphériques mobiles avec des infrastructures de réseau plus larges.

- Cell Tower-PT

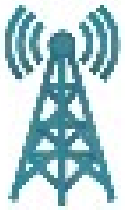

Figure 3.13: Cell Tower-PT

Une Cell Tower-PT dans Packet Tracer simule une station de base dans un réseau mobile, permettant aux appareils mobiles de se connecter au réseau. Elle est choisie pour tester les connexions sans fil et simuler la couverture réseau, essentiel pour l'analyse des performances d'un réseau mobile et des connexions sans fil.

- Battery-PT

Figure 3.14: Battery-PT

Une Battery-PT dans Packet Tracer simule une source d'alimentation de secours, fournissant de l'énergie lorsque l'alimentation principale est coupée. Elle est utilisée pour tester la gestion de l'énergie dans des scénarios où une alimentation continue est cruciale, comme dans les tours cellulaires ou autres équipements critiques.

- Power Meter-PT

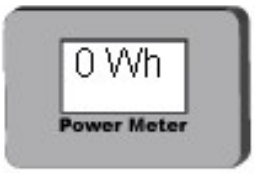

Figure 3.15: Power Meter-PT

Un Power Meter-PT dans Packet Tracer mesure la consommation d'énergie des équipements connectés au réseau. Il est utilisé pour surveiller l'efficacité énergétique, essentiel pour optimiser la consommation dans des réseaux alimentés par des batteries ou des systèmes solaires, permettant de mieux gérer les ressources énergétiques..

- Solar Panel-PT

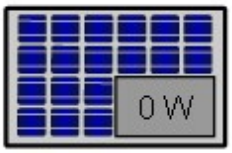

Figure 3.16: Solar Panel-PT

Un Solar Panel-PT dans Packet Tracer capte l'énergie solaire et la convertit en électricité pour alimenter des équipements. Il est choisi pour simuler des systèmes d'énergie renouvelable dans des projets de réseaux autonomes ou à faible empreinte carbone, permettant de tester la gestion de l'alimentation à partir d'une source d'énergie verte.

- Thermostat

Figure 3.17: Thermostat

Un thermostat est un dispositif connecté qui régule automatiquement la température d'un espace. Il est utilisé pour tester les scénarios de gestion climatique dans des environnements intelligents, validant l'interaction entre capteurs et systèmes de contrôle. Son choix s'explique par sa pertinence dans la régulation thermique automatisée.

- Ventilateur

Figure 3.18: ventilateur

Un ventilateur est un appareil contrôlé à distance qui assure le refroidissement d'un espace. Il permet de simuler la gestion énergétique et d'évaluer la réactivité des réseaux domotiques face aux variations climatiques. Il a été choisi pour son rôle clé dans la gestion de la ventilation intelligente.

- Alarme

Figure 3.19: Alarme

Une alarme est un système de sécurité qui détecte les intrusions et envoie des alertes. Elle est utilisée pour valider les scénarios de sécurité dans des environnements IoT, assurant une protection efficace et connectée. Ce choix s'impose par son importance dans les systèmes de surveillance.

- Webcam

Figure 3.20: Webcam

Une webcam est un dispositif de capture vidéo en temps réel. Elle permet de simuler des scénarios de surveillance et de communication visuelle, testant l'intégration des

flux multimédias dans un réseau connecté. Elle est choisie pour sa capacité à fournir une surveillance visuelle en temps réel.

- moniteur d'humidité

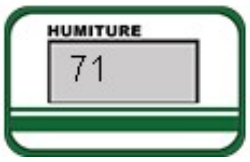

Figure 3.21: moniteur d'humidité

Un moniteur d'humidité est un capteur qui mesure la température et le taux d'humidité d'un espace. Il est utilisé pour valider la performance des réseaux environnementaux dans des scénarios domotiques ou industriels. Son choix s'explique par son utilité dans le suivi des paramètres environnementaux.

- Arroseur

Figure 3.22: Arroseur

Un arroseur automatique est un système connecté qui gère l'irrigation de manière intelligente. Il permet de tester la gestion des ressources en eau dans des scénarios d'agriculture connectée ou d'entretien paysager. Il a été retenu pour son rôle essentiel dans l'optimisation de l'irrigation.

- Climatiseur

Figure 3.23: Climatiseur

Un climatiseur est un appareil domotique qui ajuste la température et le confort d'un espace. Il est utilisé pour simuler des scénarios de régulation thermique et d'efficacité énergétique dans des environnements connectés. Il a été choisi pour son impact direct sur la gestion énergétique des espaces.

3.3 Création de l'architecture du réseau IoT

Dans le cadre de cette section, nous explorons les interactions dynamiques entre les composants de notre système de maison intelligente. En simulant une requête à distance, nous démystifions le processus complexe orchestré par notre architecture, mettant en lumière son efficacité et sa polyvalence.

3.3.1 Requête à Distance

L'utilisateur ouvre le web Browder sur le smartphone

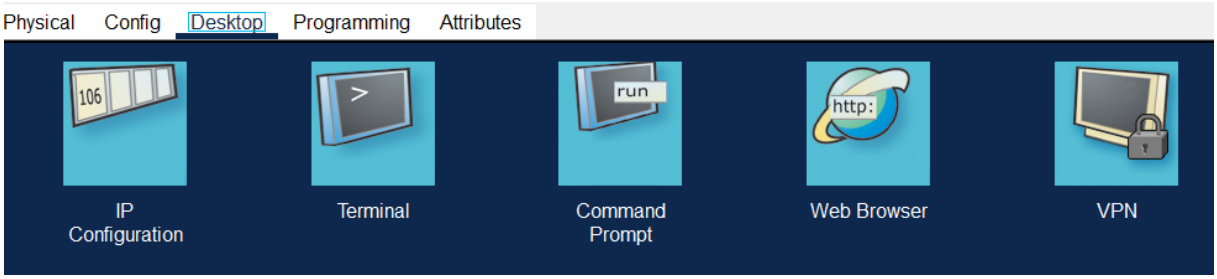

Figure 3.24: web Browder sur le smartphone

Accède à l'interface web 'iotserver', Ensuite il saisit son login et mot de passe

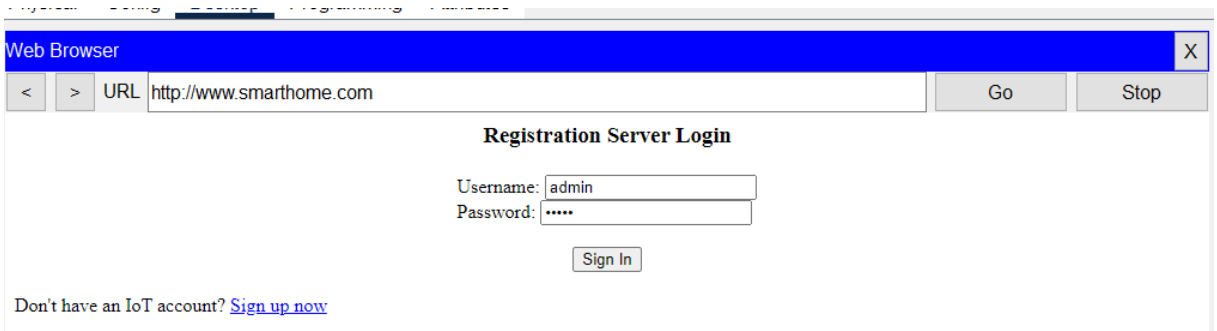

Figure 3.25: login

Le serveur DNS reçoit la requête de « http://www.smarthome.com » et traduit le nom de domaine en adresse IP du serveur IoT (10.0.0.253) et afficher l'interface graphique de l'application web dans le navigateur de smartphone.

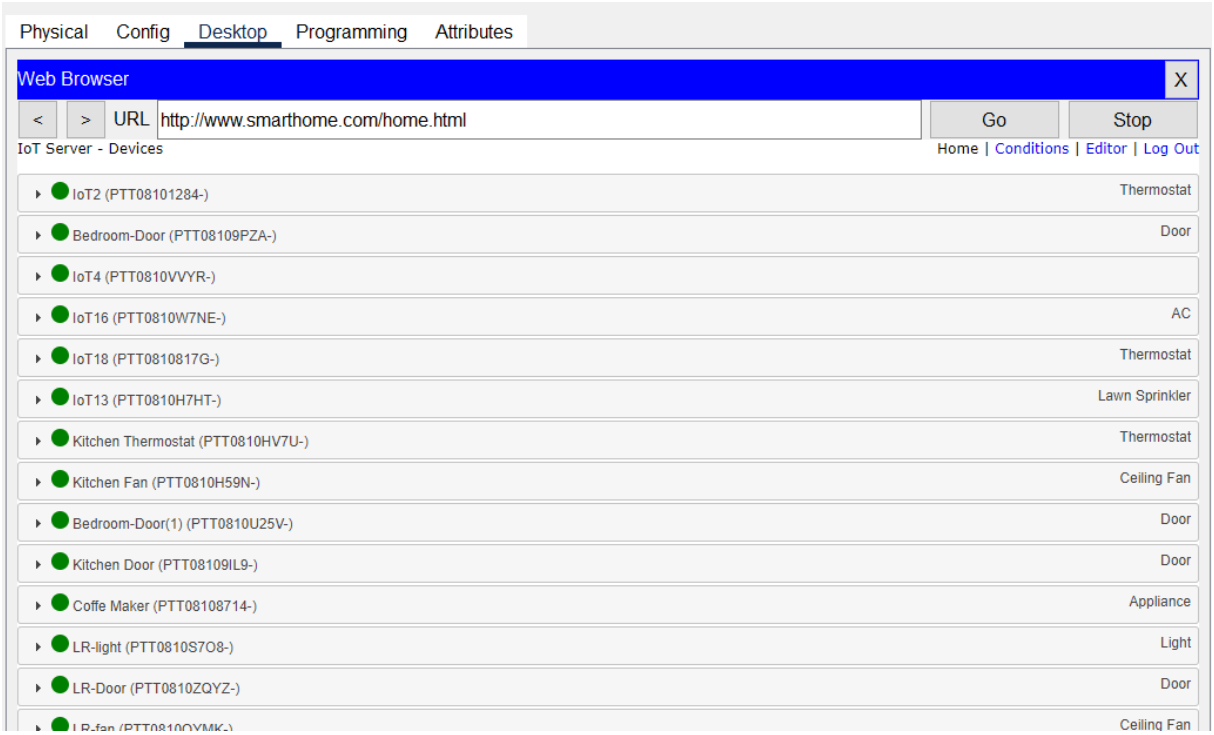

Figure 3.26: Effectuer une opération sur l'application

Si l'utilisateur effectuer une opération la requête est dirigée vers le serveur IoT via le routeur ISP et le Cloud PT.
Le serveur IoT traite la demande, identifie les dispositifs à contrôler, et envoie les commandes correspondantes à la passerelle domestique.
La passerelle domestique, reliée aux équipements de la maison, exécute les actions spécifiées à distance, influençant les dispositifs depuis n'importe où dans le monde.

3.3.2 Configuration des Équipements

3.3.2.1 Configuration de routeur

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface GigabitEthernet6/0
Router(config-if)#
Router(config-if)#int g6/0
Router(config-if)#ip add 209.165.201.225 255.255.255.224
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet6/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet6/0, changed state to up

Router(config-if)#int f0/0
Router(config-if)#ip add 209.165.200.225 255.255.255.224
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#int f1/0
Router(config-if)#ip add 10.0.0.1 255.255.255.0
Router(config-if)#no sh

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up

Router(config-if)#
Router(config-if)#exit
```

Figure 3.27: configuration des interfaces

Cette figure illustre la configuration des interfaces d'un routeur Cisco via le mode `configure terminal` :

- **Interface GigabitEthernet6/0** : Adresse IP `209.165.201.225/27`, activée avec `no shutdown`.
- **Interface FastEthernet0/0** : Adresse IP `209.165.200.225/27`, activée avec `no shutdown`.
- **Interface FastEthernet1/0** : Adresse IP `10.0.0.1/24`, activée avec `no shutdown`.

```
Router(config)#ip dhcp exclude-address 209.165.201.225 209.165.201.229
                                       ^
% Invalid input detected at '^' marker.

Router(config)#ip dhcp excluded-address 209.165.201.225 209.165.201.229
Router(config)#ip dhcp pool CELL
Router(dhcp-config)#network 209.165.201.224 255.255.255.224
Router(dhcp-config)#def
Router(dhcp-config)#default-router 209.165.201.225
Router(dhcp-config)#dns-server 10.0.0.254
Router(dhcp-config)#
```

Figure 3.28: DHCP

La figure montre la configuration du DHCP sur le routeur:

- **Exclusion d'adresses** : La commande `ip dhcp excluded-address 209.165.201.225 209.165.201.229` réserve une plage non attribuée aux clients.
- **Création du pool DHCP** :
  - `ip dhcp pool CELL` : Crée un pool nommé `CELL`.
  - `network 209.165.201.224 255.255.255.224` : Définit la plage d'adresses DHCP.
  - `default-router 209.165.201.225` : Spécifie la passerelle par défaut.
  - `dns-server 10.0.0.254` : Configure le serveur DNS.

3.3.2.2 Configuration de CENTRAL_OFFICE_SERVER

Il gère les connexions entre les antennes (cell towers) et le réseau principal.
Nous distinguons deux interfaces : la première, appelée "Backbone".

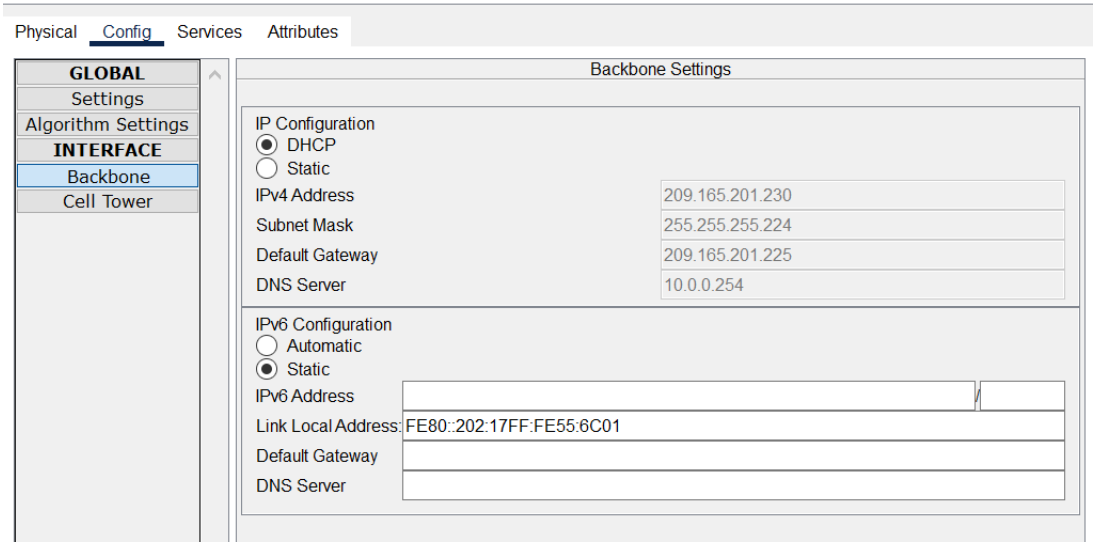

Figure 3.29: Backbone Interface

Elle est connectée au routeur de l'ISP et reçoit une adresse à partir de la plage configurée dans le serveur DHCP. En revanche, pour l'interface "Cell Tower", reliée à la station de base, une adresse spécifique est assignée, contrairement à la méthode DHCP, pour des raisons de configuration spécifiques.

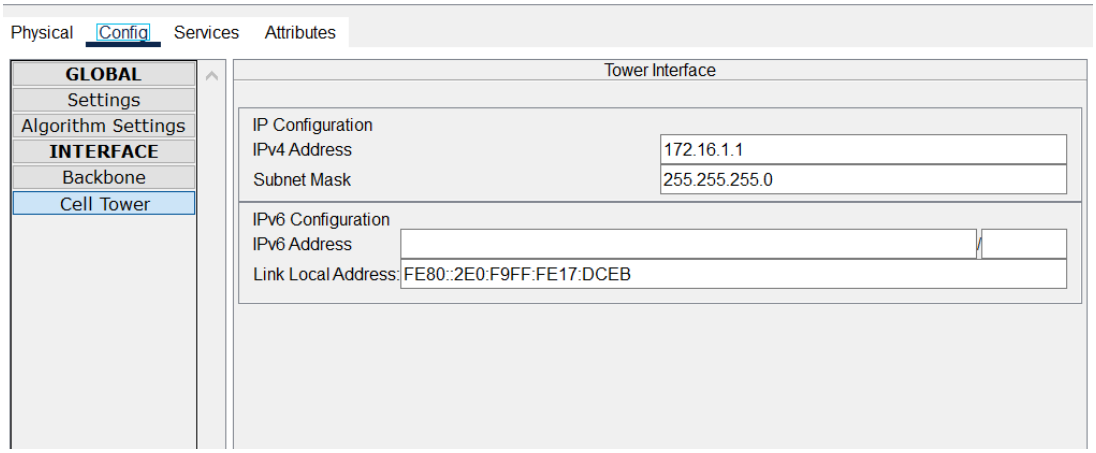

Figure 3.30: interface "Cell Tower"

3.3.2.3 La configuration des serveurs

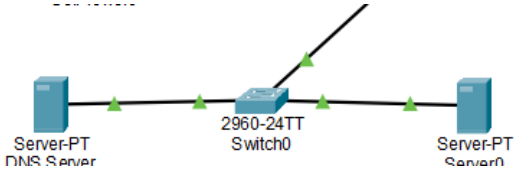

Figure 3.31: configuration des serveurs

- **le serveur IoT:**

  Pour le serveur IoT, une configuration statique est essentielle, car l'adresse du serveur doit être fixée. Cela garantit une stabilité d'adressage et une connectivité fiable pour les appareils IoT qui dépendent de cette adresse prédéfinie.

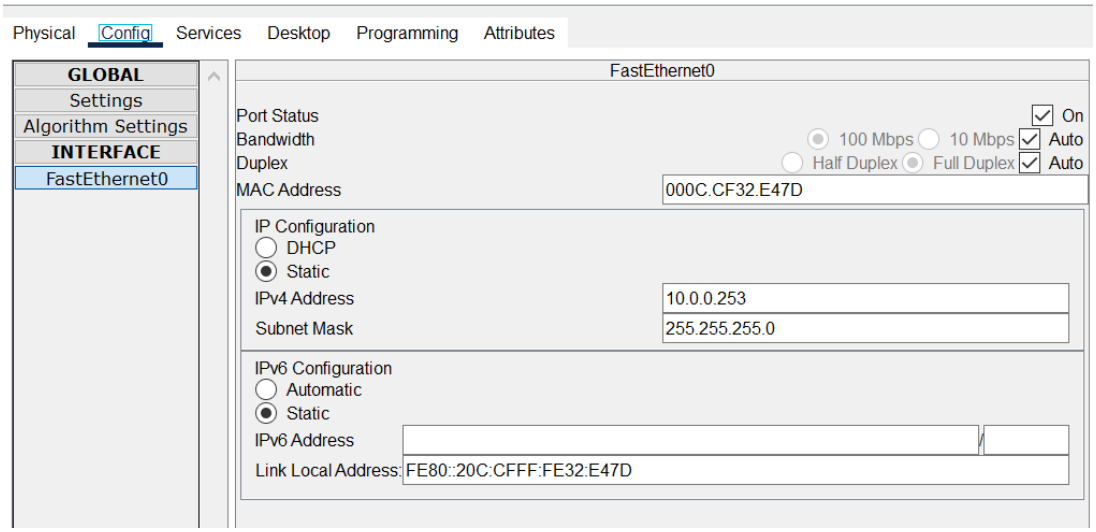

Figure 3.32: Configuration d'adressage pour le serveur IOT

Après l'adressage, la configuration du service IoT dans le serveur est réalisée de la manière suivante:

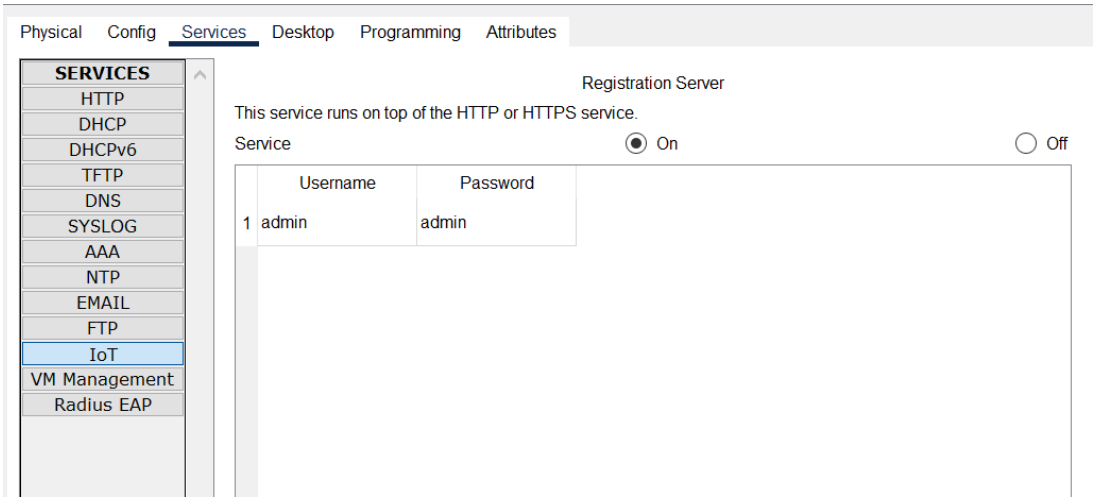

Figure 3.33: Configuration du service IoT

- **le serveur DNS**
  Pour configurer le DNS, une approche statique est nécessaire, car l'adresse du serveur DNS est définie en tant que 10.0.0.254.

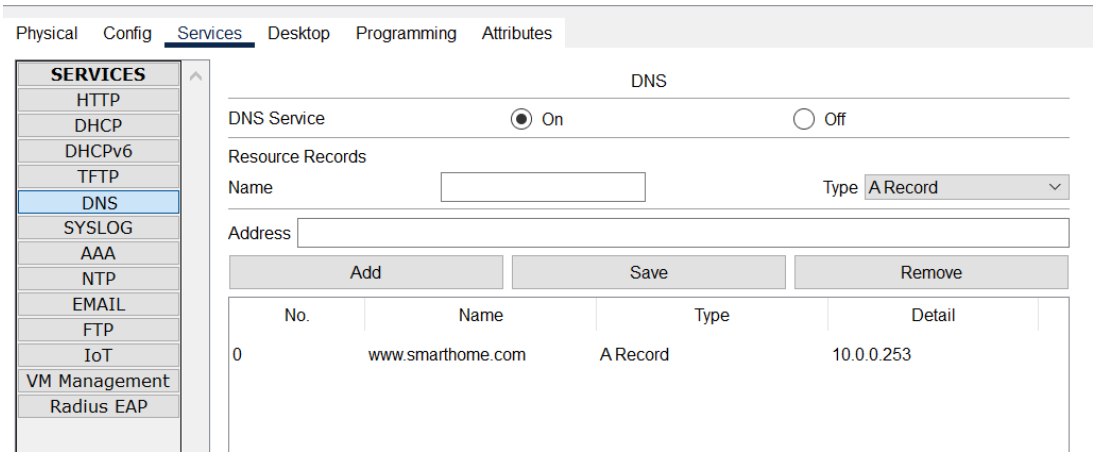

Figure 3.34: Dns config

On attribue le nom de domaine "www.smarthome.com" à l'adresse de notre serveur IoT, 10.0.0.253. Cette association simplifie l'accès au serveur en utilisant un nom plus convivial plutôt qu'une adresse IP

3.3.2.4 Configuration de SMARTPHONE

Le smartphone joue un rôle central en permettant le contrôle à distance des équipements domestiques. Grâce à une interface conviviale, il offre la possibilité de gérer divers

dispositifs tels que l'éclairage, la porte, et les fenêtres, offrant ainsi une expérience utilisateur pratique et accessible.

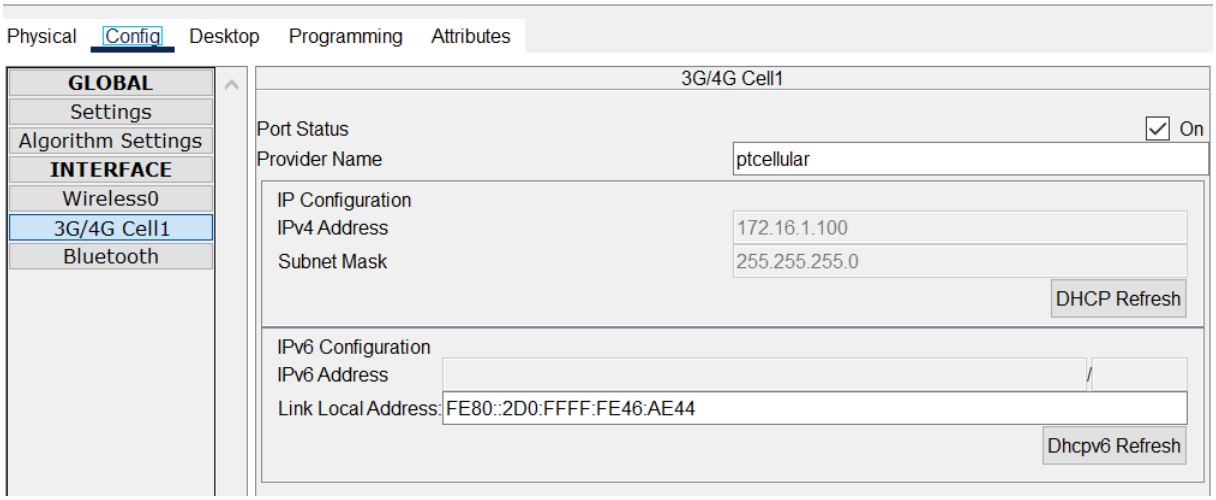

Figure 3.35: Configuration de SMARTPHONE

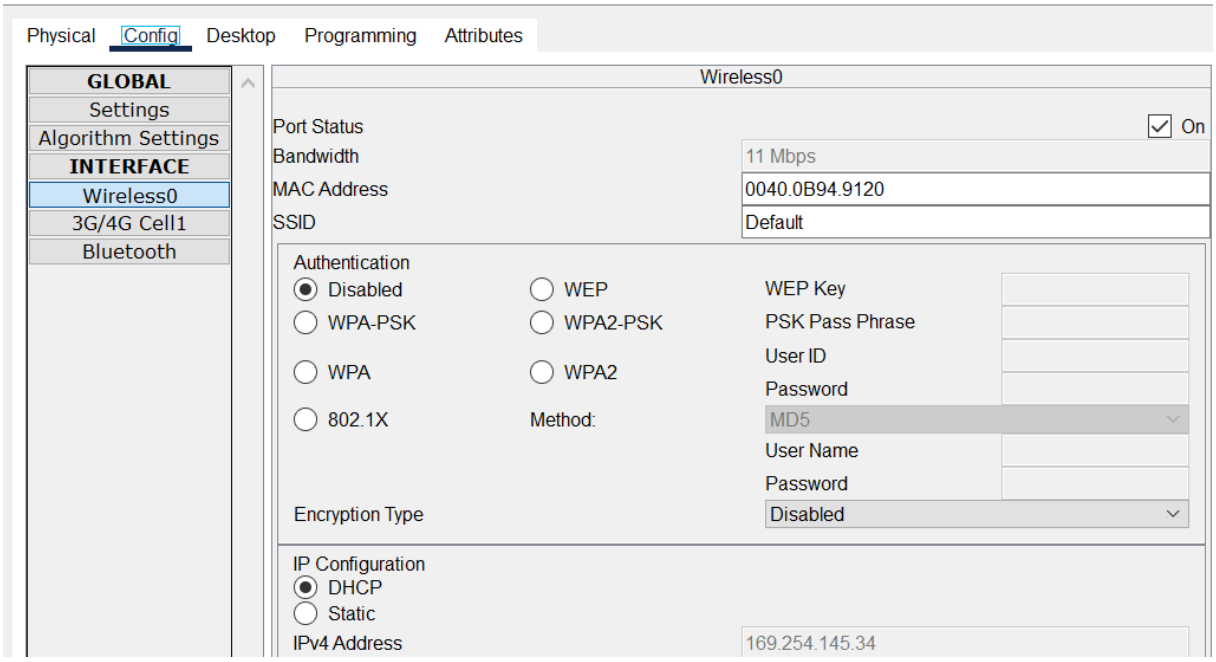

Figure 3.36: Configuration de DHCP pour le SMARTPHONE

La configuration du smartphone est simple en activant le serveur DHCP, permettant au smartphone d'obtenir une adresse IP de manière dynamique. Cela simplifie l'expérience utilisateur.

3.3.2.5 Configuration de CLOUD

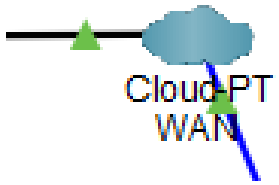

Figure 3.37: Configuration de CLOUD

Le Cloud, souvent assimilé à Internet, nécessite une configuration relativement simple. Elle implique essentiellement la redirection des paquets entrants du port eth6 vers le câble coaxial, ainsi que l'inverse

3.3.2.6 Configuration de HOME GATEWAY

Le DLC 100 Home Gateway agit comme un élément central, établissant la connectivité entre toutes les composantes de la maison, que ce soit par WiFi ou via des câbles physiques. Cette passerelle facilite la communication entre les dispositifs, offrant ainsi une intégration transparente et une gestion centralisée du réseau domestique.

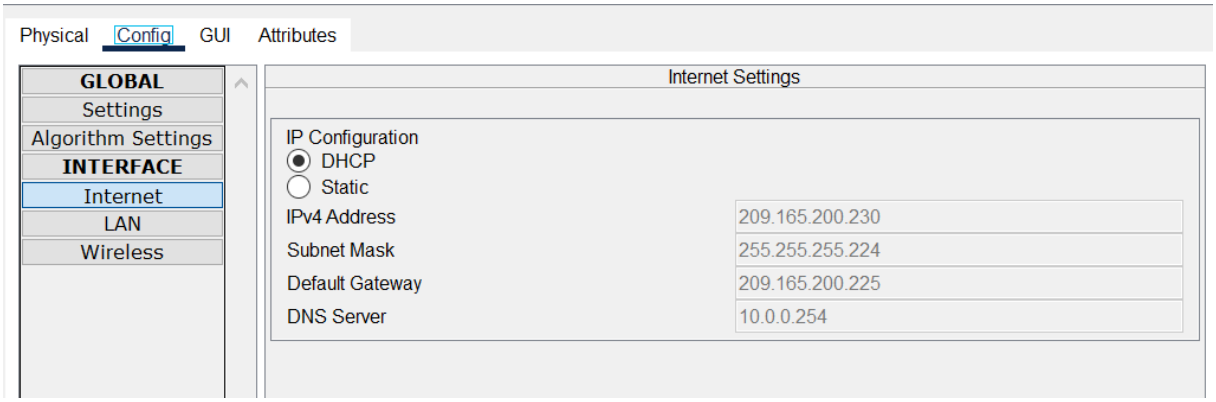

Figure 3.38: HOME GATEWAY

3.3.2.7 La configuration des équipements de la maison (l'exemple de la porte)

Pour configurer les équipements de notre maison, prenons l'exemple de la porte. Initialement, nous activons le protocole DHCP pour permettre à la porte d'obtenir une adresse du réseau, telle que 192.168.25.112/24, simplifiant ainsi la gestion des dispositifs connectés. Et nous configurons également la communication avec le serveur IoT de la manière suivante(config => settings)

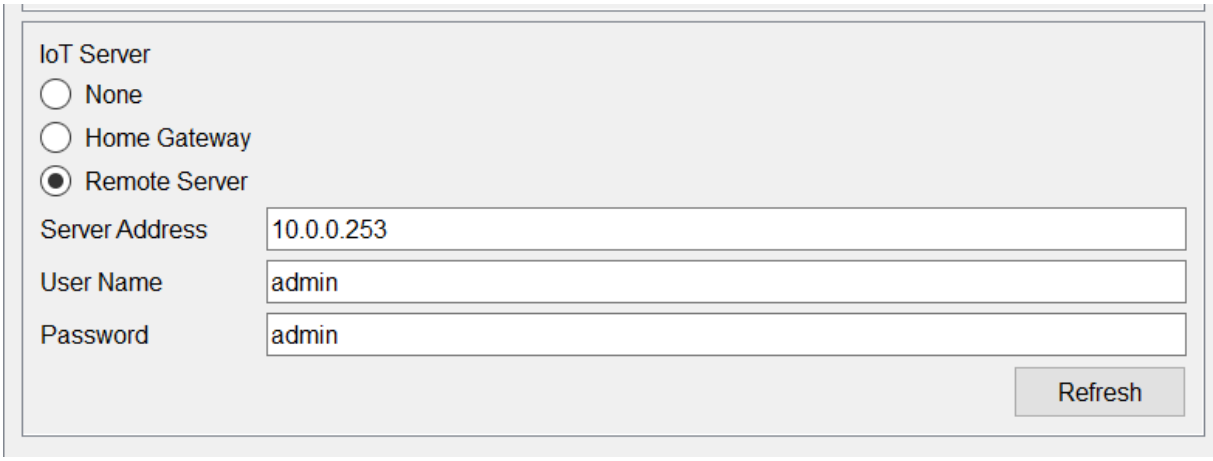

Figure 3.39: Configuration du Serveur IOT

Nous répétons ces étapes pour chaque équipement de la maison, garantissant une communication globale et centralisée. Ceci permet un contrôle efficace de tous les équipements via le serveur IoT, simplifiant ainsi la gestion globale du système domestique.

3.4 Simulation Packet Tracer : Cuisine Intelligente

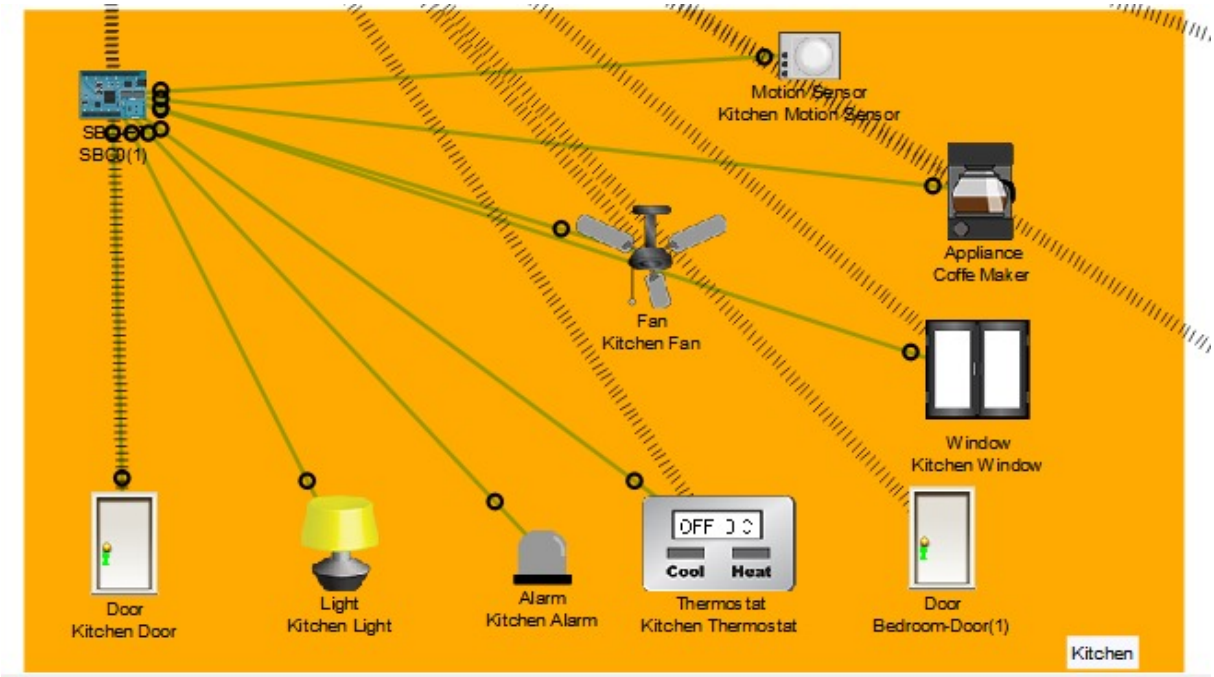

Figure 3.40: Enter Caption

Cette simulation repose sur un SBC (Single Board Computer) connecté à plusieurs capteurs et actionneurs dans une cuisine intelligente. Le code suivant permet de gérer

la détection de mouvement, la préparation de café, le contrôle de la température, ainsi que la gestion des alertes de fumée.

```
from gpio import *
from time import *

def main():
    while True:
        # Read temperature from analog pin 0
        temperature = analogRead(0)

        # Read motion sensor status from digital pin 1
        motion_sensor = digitalRead(1)

        if motion_sensor == HIGH:
            print("Motion detected. Turning on Lamp...")
            customWrite(2, 1)  # Turn on Lamp (pin 2)
            # Open door
            print("Opening the door.")
            customWrite(7, 1)
            print("Someone's awake, making coffe...")
            customWrite(8, 1)

            # Determine fan speed based on temperature thresholds
            if temperature < 20:
                fan_speed = 0  # Fan off
                print("Fan Off (temperature below 20C)")
            elif 20 <= temperature < 25:
                fan_speed = 1  # Low speed
                print("Fan Low Speed (temperature 20-24C)")
            else:
                fan_speed = 2  # High speed
                print("Fan High Speed (temperature 25C and
                    above)")

            # Set fan speed on pin 3
            customWrite(3, fan_speed)

            delay(6000)
            print("Closing the door.")
            customWrite(7, 0)
            print("Turning off Lamp.")
            customWrite(2, 0)  # Turn off Lamp
            print("Turning off Fan.")
            customWrite(3, 0)  # Turn off Fan
            print("Turning Off Coffe Maker")
            customWrite(8, 0)

            delay(500)  # Small delay to avoid rapid toggling
```

```
46
47          # Read smoke level from analog pin 4
48          smokeLevel = analogRead(4)
49          # smokeLevel = 60
50
51          # Check smoke levels and set alarm and fan states
52          if smokeLevel >= 10 and smokeLevel < 50:
53              print("Moderate smoke level detected. Activating
                    alerts.")
54              digitalWrite(5, LOW)
55              customWrite(6, 1)
56              customWrite(7, 1)
57          elif smokeLevel >= 50:
58              print("High smoke level detected! Activating
                    alarms.")
59              digitalWrite(5, HIGH)
60              customWrite(6, 1)
61              customWrite(7, 1)
62          else:
63              # print("Smoke level normal. Deactivating all
                    alerts.")
64              digitalWrite(5, LOW)
65              customWrite(6, 0)
66              customWrite(7, 0)
67
68          delay(500)  # Delay to avoid excessive sensor reading
69
70  if __name__ == "__main__":
71      main()
```

Listing 3.1: Code Python pour le contrôle domotique

- Détection de mouvement et actions associées
  - Le capteur de mouvement (connecté au pin digital 1) détecte la présence d'une personne.
  - Lorsqu'un mouvement est détecté :
    - Allumage de la lampe (pin digital 2) : Une lumière s'allume pour éclairer la cuisine.
    - Ouverture de la porte (pin digital 7) : Une commande est envoyée pour déverrouiller et ouvrir la porte.
    - Préparation du café (pin digital 8) : La machine à café se met en marche automatiquement pour préparer une tasse de café.
- Régulation de la température et gestion du ventilateur

  - Un capteur de température (connecté au pin analogique 0) mesure la température ambiante.
  - Le système ajuste la vitesse du ventilateur en fonction de seuils prédéfinis :
    * Température $< 20°C$: Ventilateur éteint.
    * Température entre 20°C et 24°C : Ventilateur à faible vitesse (Low Speed).
    * $Température \geq 25°C$: Ventilateur à haute vitesse (High Speed).

- Gestion des alertes en cas de fumée
  - Un capteur de fumée (connecté au pin analogique 4) mesure les niveaux de fumée dans la cuisine.
  - Trois scénarios sont possibles :
    * Niveau de fumée normal : Aucune action, les alertes sont désactivées.
    * Niveau de fumée modéré (10-50) :
      · Activation d'alertes légères (pin digital 5 en LOW).
      · Le ventilateur (pin digital 6) s'active pour dissiper la fumée.
      · La porte (pin digital 7) s'ouvre automatiquement.
    * Niveau de fumée élevé (≥ 50) :
      · Activation d'alarmes fortes (pin digital 5 en HIGH).
      · Ventilateur et porte restent actifs pour une ventilation et une évacuation maximales.

- Temporisation et remise à l'état initial
  - Après 6 secondes, les éléments suivants reviennent à leur état initial :
    * La porte se ferme (pin digital 7 en LOW).
    * La lampe s'éteint (pin digital 2 en LOW).
    * Le ventilateur s'éteint (pin digital 3 en LOW).
    * La machine à café s'éteint (pin digital 8 en LOW).

3.5 Simulation Packet Tracer : Chambre Intelligente

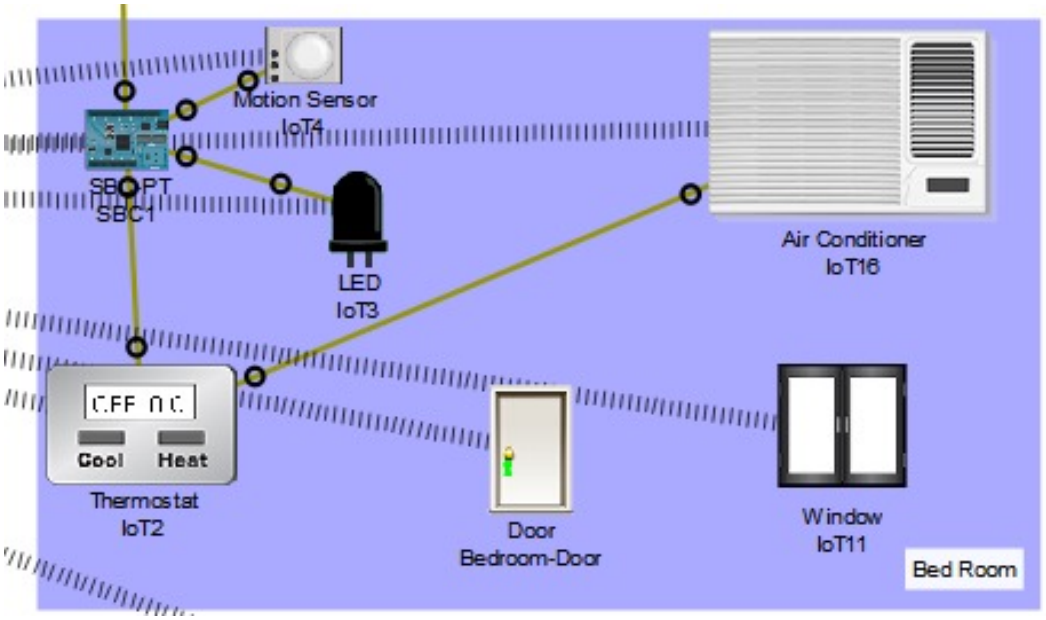

Figure 3.41: Enter Caption

Cette simulation repose sur un Single Board Computer (SBC) connecté à plusieurs capteurs et actionneurs dans une chambre intelligente. Le système gère la détection de mouvement pour activer automatiquement le thermostat et ajuster la température ambiante. Lorsqu'un mouvement est détecté, il déclenche également l'ouverture de la porte et peut activer un air conditionné pour réguler la température. En parallèle, la fenêtre se ferme ou s'ouvre automatiquement en fonction de la température mesurée. Enfin, un LED fournit une indication visuelle de l'état des différents dispositifs dans la chambre.

```
from gpio import *
from time import *

def main():
    pinMode(0, IN)

    while True:
        if digitalRead(0) == HIGH:
            digitalWrite(1, HIGH)
        else:
            digitalWrite(1, LOW)

        # Read the digital signal from pin 9 and store it in the
           motion_sensor variable.
        motion_sensor = digitalRead(0)

        if motion_sensor == HIGH:  # Check if motion is detected.
```

```
17              print("Motion detected...")
18              print("Turning on Thermostat...")
19
20              customWrite(3, 1)  # Send value 1 to pin 1 (turn on
                   Thermostat).
21              delay(6000)  # Keep thermostat on for 6000 ms (6
                   seconds).
22
23              print("Turning off Thermostat.")
24              customWrite(3, 0)
25
26              delay(500)
27
28  if __name__== "__main__":
29      main()
```

Listing 3.2: Code Python pour la gestion du thermostat

- Détection de mouvement et activation du thermostat
  - Le capteur de mouvement (connecté au pin digital 0) détecte la présence d'une personne dans la chambre.
  - Lorsqu'un mouvement est détecté :
    - Activation du thermostat (pin digital 3) : Le thermostat est activé pour ajuster la température ambiante et offrir un confort optimal.
    - Affichage du message : Un message s'affiche pour indiquer que le mouvement a été détecté et que le thermostat est activé.
  - Le thermostat reste allumé pendant 6 secondes (temps suffisant pour ajuster la température).
  - Extinction automatique du thermostat : Le thermostat s'éteint automatiquement après cette période pour économiser de l'énergie.
- Fonctionnement global du thermostat
  - Le thermostat connecté à la broche numérique 3 est utilisé pour réguler la température ambiante de la chambre.
  - Activation : Le thermostat est activé uniquement lorsqu'un mouvement est détecté, garantissant qu'il ne fonctionne que lorsque la chambre est occupée.
  - Extinction : Le thermostat s'éteint après 6 secondes d'inactivité pour réduire la consommation d'énergie lorsque la chambre n'est plus occupée.
- Temporisation et remise à l'état initial

– Le système insère des temporisations afin de ne pas envoyer des commandes trop fréquemment :

– Temporisation après détection de mouvement : Une pause de 500 ms est introduite avant chaque nouvelle vérification du capteur de mouvement pour éviter une sollicitation excessive du SBC.

– Temporisation d'arrêt du thermostat : Le thermostat reste actif pendant 6 secondes pour ajuster la température avant de s'éteindre.

3.6 Simulation Packet Tracer : Salon Intelligent

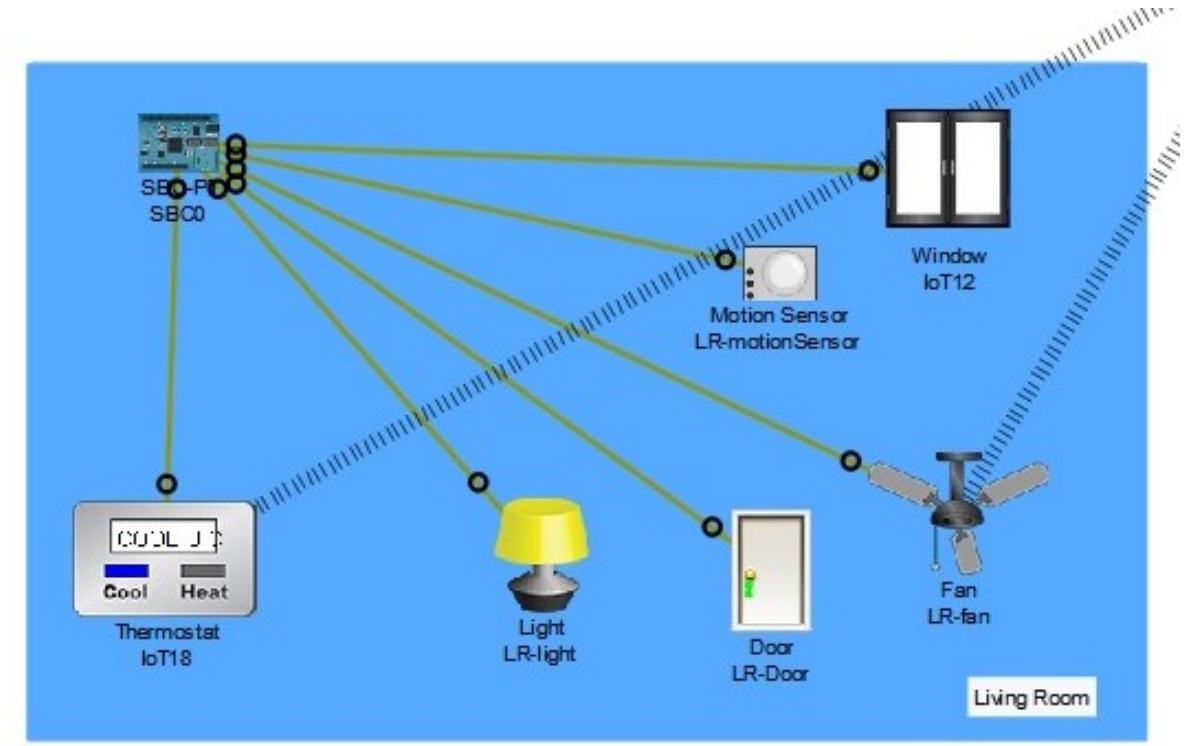

Figure 3.42: Enter Caption

Cette simulation repose sur un Single Board Computer (SBC) connecté à plusieurs capteurs et actionneurs pour automatiser la gestion du salon intelligent. Le système détecte la présence des occupants grâce à un capteur de mouvement, régule la température à l'aide d'un ventilateur, et contrôle l'éclairage ainsi que l'ouverture de la porte. Ces automatisations visent à améliorer le confort et l'efficacité énergétique du salon.

```
1 from gpio import *
2 from time import *
3
```

```
def main():
    while True:
        # Read temperature from analog pin 0
        temperature = analogRead(0)

        # Read motion sensor from digital pin 1
        motion_sensor = digitalRead(1)

        # Check motion sensor status
        if motion_sensor == HIGH:
            print("Motion detected. Opening the door...")
            customWrite(4, 1)  # Open the door
            print("Turning on Lamp...")
            customWrite(2, 1)
            delay(6000)  # Delay of 6 seconds

            print("Turning off Lamp.")
            customWrite(2, 0)
            customWrite(4, 0)
            delay(6000)  # Delay of 6 seconds

            # Determine fan speed based on temperature
            if temperature < 20:
                fan_speed = 0  # Fan off
                print("Fan Off (temperature below 20C)")
            elif 20 <= temperature < 25:
                fan_speed = 1  # Low speed
                print("Fan Low Speed (temperature 20-24C)")
            else:
                fan_speed = 2  # High speed
                print("Fan High Speed (temperature 25C and
                   above)")

            # Set fan speed
            customWrite(3, fan_speed)

            delay(6000)  # Delay to allow fan to run at set speed
            print("Turning off Fan.")
            customWrite(3, 0)  # Turn off fan

            delay(500)

if __name__ == "__main__":
    main()
```

Listing 3.3: Code Python pour la gestion du mouvement et de la température

- Détection de mouvement et contrôle des dispositifs

  - Capteur de mouvement (connecté au pin digital 1) :
    - Détecte la présence d'une personne dans le salon.
    - Lorsqu'un mouvement est détecté :
      - Ouverture de la porte (pin digital 4) : Une commande est envoyée pour ouvrir la porte.
      - Allumage de la lampe (pin digital 2) : Une lumière est allumée pour éclairer le salon.
    - Les dispositifs (porte et lampe) restent actifs pendant 6 secondes avant de revenir à leur état initial.
    - Extinction automatique de la lampe et fermeture de la porte pour économiser l'énergie.

- Régulation de la température et gestion du ventilateur
  - Capteur de température (connecté au pin analogique 0) :
    - Mesure la température ambiante pour ajuster la vitesse du ventilateur en fonction de seuils prédéfinis :
      - Température < 20°C : Ventilateur éteint.
      - $20°C \leq Température < 25$°C : Ventilateur à faible vitesse (Low Speed).
      - $Température \geq 25°C$: Ventilateur à haute vitesse (High Speed).
    - Une fois la vitesse réglée, le ventilateur fonctionne pendant 6 secondes avant de s'éteindre automatiquement.

- Temporisation et remise à l'état initial
  - Temporisations intégrées pour éviter une sollicitation excessive du SBC :
    - Une pause de 500 ms est insérée entre chaque vérification du capteur de mouvement.
    - Chaque dispositif retourne à son état initial après un délai de 6 secondes.
  - Ce fonctionnement garantit une optimisation des ressources tout en assurant une réactivité efficace.

3.7 Simulation Packet Tracer : Entrée Intelligente

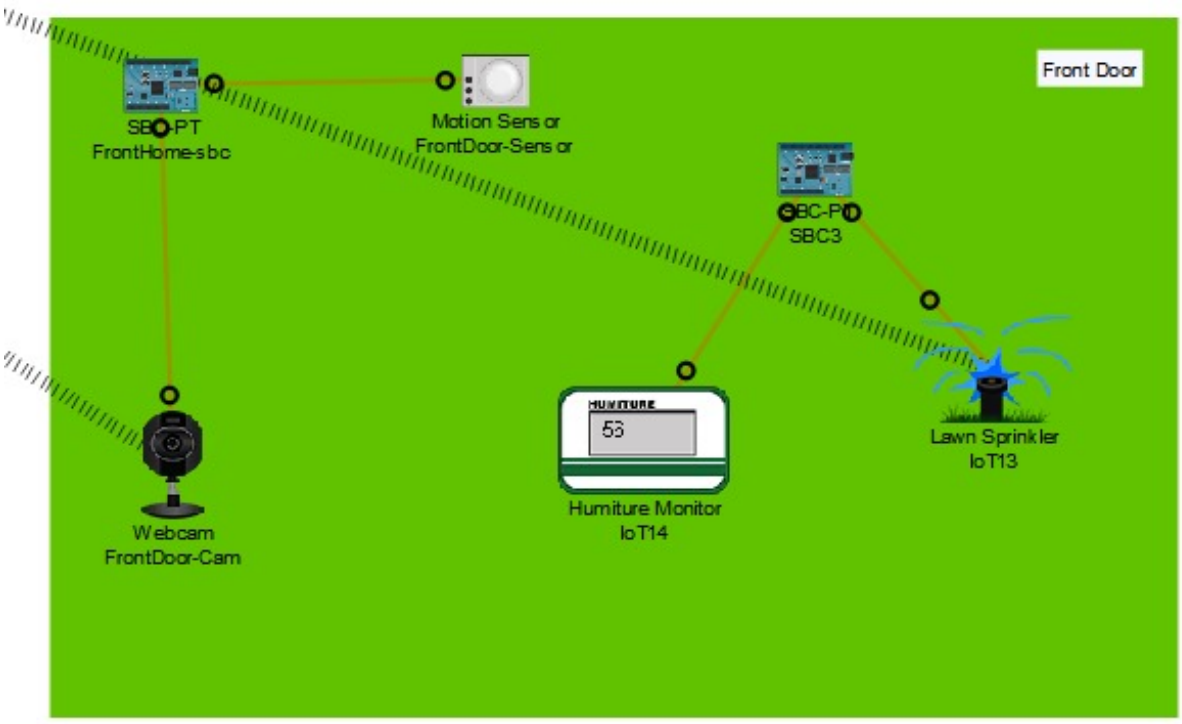

Figure 3.43: Enter Caption

Cette simulation repose sur un Single Board Computer (SBC) connecté à plusieurs capteurs et actionneurs pour automatiser les interactions à la porte d'entrée. Le système utilise un capteur de mouvement pour détecter la présence, active une webcam pour surveiller l'activité, contrôle un moniteur d'humidité et déclenche un arroseur automatique selon les conditions.

```
from gpio import *
from time import *

def main():
    pinMode(0, IN)  # Configure pin 0 as input

    while True:
        # Check if motion is detected
        if digitalRead(0) == HIGH:
            digitalWrite(1, HIGH)  # Turn on connected device
        else:
            digitalWrite(1, LOW)  # Turn off connected device

        # Store the digital signal from pin 0 in the
           motion_sensor variable
        motion_sensor = digitalRead(0)
```

```
16
17          if motion_sensor == HIGH:  # If motion is detected
18              print("Motion detected...")
19
20              customWrite(1, 1)  # Activate the camera
21              delay(6000)
22
23              print("Turning off camera.")  # Deactivate the camera
24              customWrite(1, 0)
25
26              delay(500)  # Wait 500 ms before checking again
27
28  if __name__ == "__main__":
29      main()
```

Listing 3.4: Code Python pour la détection de mouvement et activation de caméra

- Détection de mouvement et activation de la webcam
  - Capteur de mouvement (connecté au pin digital 0) :
    - Détecte la présence d'une personne ou d'un mouvement devant la porte.
    - Lorsqu'un mouvement est détecté :
      - Activation de la webcam (pin digital 1) : La webcam s'allume pour capturer ou transmettre des images en temps réel.
      - La webcam reste active pendant 6 secondes pour assurer une surveillance adéquate avant de s'éteindre automatiquement.
- Surveillance de l'humidité et gestion de l'arroseur
  - Moniteur d'humidité (connecté à un pin analogique) :
    - Mesure le niveau d'humidité ambiante pour évaluer les besoins en irrigation.
    - Conditions de fonctionnement :
      - Si l'humidité est inférieure à un seuil défini, l'arroseur automatique (lawn sprinkler) s'active pour arroser les environs.
      - Lorsque le seuil est atteint ou dépassé, l'arroseur reste désactivé pour économiser l'eau.
    - Arroseur automatique (connecté au pin digital 2) :
      - Fonctionne en tandem avec le moniteur d'humidité pour une gestion écologique.
- Temporisation et fonctionnement global

- Temporisation :
  - Une pause de 500 ms est appliquée entre chaque cycle de détection pour éviter une sollicitation excessive des composants.
  - L'arroseur et la webcam sont temporisés pour s'éteindre après leurs tâches respectives, assurant une efficacité énergétique.

4
Conclusion

Pour conclure, ce projet sur la modélisation et détection des objets connectés dans un bâtiment intelligent montre comment les technologies IoT peuvent améliorer la gestion des ressources et la sécurité des bâtiments. En utilisant des outils comme Cisco Packet Tracer et Python, nous avons pu simuler des scénarios réels et comprendre comment les objets connectés peuvent interagir pour rendre un bâtiment plus intelligent et plus efficace.

Cependant, un aspect essentiel à améliorer pour l'avenir est la sécurisation de la maison intelligente. Il est important de mettre en place des systèmes de sécurité solides pour protéger les données des utilisateurs et empêcher les intrusions. Cela inclut la gestion des accès et la protection contre les cyberattaques. En optimisant la sécurité, ce projet pourra être une base pour des maisons encore plus connectées, sûres et intelligentes à l'avenir.

Webliographie

[1] Cisco Packet Tracer Labs. "Cisco Packet Tracer Tutorial." *YouTube*. https://www.youtube.com/watch?v=DImMM-AgiQ4&ab_channel=CiscoPacketTracerLabs
Consulté le 26 novembre 2024.

[2] BigBangScience. "How to use Cisco Packet Tracer for IoT." *YouTube*. https://www.youtube.com/watch?v=qBMUKP5sdJI&ab_channel=BigBangScience
Consulté le 26 novembre 2024.

[3] Benard Otom. "Packet Tracer Tutorial for Beginners." *YouTube*. https://www.youtube.com/watch?v=A3bfsLExPzk&ab_channel=BenardOtom
Consulté le 26 novembre 2024.

[4] Wikipedia. "Ville intelligente." *Wikipedia*. https://fr.wikipedia.org/wiki/Ville_intelligente
Consulté le 20 octobre 2024.

[5] Cisco Networking Academy. "Cisco Packet Tracer." *Cisco*. https://www.netacad.com/fr/courses/packet-tracer
Consulté le 26 novembre 2024.

[6] Adafruit. "Learning Python for IoT." *Adafruit*. https://learn.adafruit.com/category/python
Consulté le 26 novembre 2024.

[7] MOKOSmart. "IoT in Smart Home." *MOKOSmart*. https://www.mokosmart.com/fr/iot-in-smart-home/
Consulté le 26 novembre 2024.

[8] Adafruit Industries. "Introduction to IoT and Smart Homes." *YouTube*. https://youtu.be/HoV7wOVjH40?feature=shared
Consulté le 26 novembre 2024.