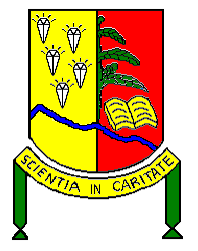
**UNIVERSITE DE MBUJIMAYI**

***Fondation Cardinal J.A. MALULA***

**CAMPUS DE TSHIKAMA**

**B.P.255 – MBUJIMAYI**

****

**Faculté des Sciences Appliquées**

**Département d’informatique**

**DEVELLOPEMENT D’UN SYSTEME SURVEILLANCE DOMOTIQUE**

***Présenté par : KABEYA MULEBA Arnauld***

***Travail de fin de cycle présenté et défendu en vue de l’obtention du grade de licencié en Informatique***

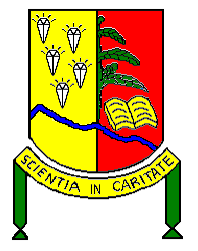
***ANNEE ACADEMIQUE 2023-2024***

**UNIVERSITE DE MBUJIMAYI**

***Fondation Cardinal J.A. MALULA***

**CAMPUS DE TSHIKAMA**

**B.P.255 – MBUJIMAYI**

****

**Faculté des Sciences Appliquées**

**Département d’informatique**

**DEVELLOPEMENT D’UN SYSTEME A SURVEILLANCE DOMOTIQUE**

***Travail de fin de cycle présenté et défendu en vue de l’obtention du grade de licencié en Informatique***

***Présenté par :* *KABEYA MULEBA Arnauld***

***Dirigé par :* *KATULUMBA MBIYA NGANDU Marcel***

***Chef de travaux***

***Encadré par :* *NGELEKA MUSANGU Steve***

***Assistant***

***ANNEE ACADEMIQUE 2023-2024***

***EPIGRAPHE***

« La connaissance n’est pas une simple accumulation de faits, mais la capacité de les interpréter et de les transformer pour résoudre les défis qui se dressent devant nous. » [1]

***Albert Einstein***

***DEDICACE***

A mes très chers parents Floribert Kabeya et Jolie Mujinga pour leur amour, soutien et conseil

***Kabeya Muleba Arnauld***

***REMERCIEMENTS***

Nous adressons nos plus profonds remerciements à notre Père céleste, Jésus-Christ, pour sa protection et sa bienveillance. C’est grâce à Lui que nous avons pu parvenir à ce stade de notre parcours. Honneur, gloire et puissance lui reviennent, Lui qui a créé toutes choses et soutient chacune de nos démarches.

Nos sincères remerciements vont également aux autorités académiques de notre université, ainsi qu’à nos professeurs, chefs de travaux et assistants, pour leur encadrement, leur dévouement et leurs précieux enseignements.

Nous exprimons une gratitude particulière à notre directeur de travail, le chef de travaux Katulumba Mbiya Ngandu Marcel, et à notre encadreur, l’assistant Ngeleka Musangu Steve, pour leur patience, leur disponibilité et leurs conseils avisés dans la réalisation de ce travail.

À mes très chers parents, oncles et frères, pour votre soutien moral et matériel tout au long de ce parcours, je vous adresse mes plus sincères remerciements.

Enfin, à mes amies et proches : Rachel Mbuyi, Plamedie Ndaya, Benitha Kalanga, Pierrome Mwamare, Tegra Tshibanda, Armel Kalala, Sephora Kabeya et particulièrement Maman Nana, pour votre soutien moral, votre amour inestimable et votre présence constante qui m’ont porté durant tout mon cursus universitaire.

À vous tous, je vous suis profondément reconnaissante. Que ce travail soit le fruit de votre impact positif sur ma vie.

***LISTE DES FIGURES***

Fig1 La domotique…………………………………………………………………………...6

Fig2 Les Piliers de la domotique……………………………………………………………...7

Fig3 Types de capteurs IoT…………………………………………………………………..11

Fig4 Capteur de température…………………………………………………………………11

Fig5 Capteur de fumée……………………………………………………………………….12

Fig6 Capteur de mouvement…………………………………………………………………12

Fig7 Capteur d'humidité……………………………………………………………………. .13

Fig8 Microcontrôleur………………………………………………………………………...13

Fig9 Capteur de force..………………………………………………………………………14

Fig10 Réalisation…………………………………………………………………………….21

Fig11 Test et Validation………………………………………………………………… ..…24

***LISTE DES TABLEAUX***

Tableau 1: Comparaison des microcontrôleurs populaires …………………………………14

***LISTE DES ABREVIATIONS ET ACRONYMES***

IoT : Internet of Things (Internet des Objets)

IDO : Internet des Objets (équivalent français d’IoT)

RASPBERRY : Raspberry Pi (Ordinateur à faible coût et taille réduite, utilisé pour des projets IoT et domotiques)

PROTEUS : Logiciel de simulation électronique (utilisé pour simuler les circuits et les systèmes embarqués)

TINKERCAD : Outil en ligne pour la modélisation 3D et la simulation de circuits électroniques

CLOUD : Cloud Computing (Informatique en Nuage)

MIT : Massachusetts Institute of Technology (Institut de Technologie du Massachusetts)

RFID : Radio Frequency Identification (Identification par Radiofréquence)

M2M : Machine to Machine (Communication entre machines)

COAP : Constrained Application Protocol (Protocole d'Application Contraint)

BIG DATA : Données Massives (Traitement de grandes quantités de données)

GPS : Global Positioning System (Système de Positionnement Global)

IA : Intelligence Artificielle

BLE : Bluetooth Low Energy (Bluetooth à faible énergie)

BLUETOOTH : Technologie de communication sans fil à courte portée

IFTTT : If This Then That (Si ceci, alors cela, un service de déclenchement d'actions automatiques)

NFC : Near Field Communication (Communication en champ proche)

IR : Infrared (Infrarouge)

PIR : Passive Infrared Sensor (Capteur Infrarouge Passif, utilisé pour détecter les mouvements)

ESP8266 : Module Wi-Fi à faible coût pour projets IoT

ESP32 : Microcontrôleur avec Wi-Fi et Bluetooth intégré

AT MEGA : Famille de microcontrôleurs utilisés dans des projets électroniques

RAM : Random Access Memory (Mémoire à Accès Direct)

ROM : Read-Only Memory (Mémoire en Lecture Seule)

KB : Kilobyte (Unité de mesure de la mémoire équivalente à 1024 octets)

BIT : Binary Digit (Chiffre binaire, valeur de 0 ou 1)

MHZ : Megahertz (Unité de fréquence égale à un million de cycles par seconde)

GPIO : General Purpose Input/Output (Entrée/Sortie à usage général)

DUAL-CORE : Processeur à double cœur, utilisé pour améliorer les performances du traitement des données

I2C : Inter-Integrated Circuit (Protocole de communication série entre composants électroniques)

SPI : Serial Peripheral Interface (Interface périphérique série)

UART : Universal Asynchronous Receiver/Transmitter (Récepteur/Transmetteur Asynchrone Universel)

DHT11 : Capteur de température et d'humidité

DHT22 : Capteur de température et d'humidité plus précis que le DHT11

BME280 : Capteur de température, humidité et pression atmosphérique

ARDUINO : Plateforme de prototypage open-source pour les projets électroniques

C++ : Langage de programmation orienté objet

PYTHON : Langage de programmation très utilisé dans le développement IoT et l’analyse de données

LCD : Affichage a Cristaux Liquides (C’est une technologie d’affichage qui utilise des cristaux liquides pour contrôler la lumière et afficher des images ou du texte)

### ***Résumé***

Ce mémoire porte sur le **développement d'un système de surveillance domotique** basé sur l'intégration de capteurs IoT. L'objectif principal est de concevoir et de mettre en place un système permettant de surveiller l'intérieur d'une maison en temps réel, en détectant toute activité suspecte ou toute condition anormale (intrusion, fumée, température anormale, etc.).

Pour la mise en œuvre, nous avons utilisé **Tinkercad** afin de modéliser une maison et tester l’intégration des différents composants électroniques. Le système repose sur des **capteurs** (mouvement, température, fumée, etc.), un **buzzer** pour les alertes sonores, des **LEDs** pour les notifications visuelles, et un **écran LCD** pour l'affichage des informations essentielles. Contrairement aux solutions classiques qui s’appuient sur une application mobile ou un site web, notre approche centralise toutes les interactions et notifications sur l’écran LCD du dispositif.

Ce travail s’inscrit dans une démarche visant à améliorer la **sécurité domestique**, en offrant une solution accessible, autonome et efficace. À travers les tests réalisés sur **Tinkercad**, nous avons pu valider la faisabilité technique du système et démontrer son potentiel pour une intégration réelle dans un environnement domestique.

### ***Abstract***

This thesis focuses on the **development of a home automation surveillance system** based on IoT sensor integration. The main objective is to design and implement a system that monitors a house in real-time, detecting any suspicious activity or abnormal conditions (intrusion, smoke, abnormal temperature, etc.).

For implementation, we used **Tinkercad** to model a house and test the integration of various electronic components. The system relies on **sensors** (motion, temperature, smoke, etc.), a **buzzer** for sound alerts, **LEDs** for visual notifications, and an **LCD screen** to display essential information. Unlike traditional solutions that rely on a mobile application or a website, our approach centralizes all interactions and notifications on the system’s LCD screen.

This work aims to enhance **home security** by providing an accessible, autonomous, and efficient solution. Through tests conducted on **Tinkercad**, we validated the technical feasibility of the system and demonstrated its potential for real-world integration in a domestic environment.

1. ***INTRODUCTION GENERALE***

Aujourd'hui, avec l'évolution des technologies, les maisons deviennent de plus en plus connectées et intelligentes. Grâce à l'Internet des objets (IoT), il est possible de rendre nos foyers plus sûrs et plus confortables. Ce projet de mémoire s'intéresse au développement d'un système de surveillance domestique utilisant différents capteurs. L'objectif principal est de créer un système capable de surveiller différents paramètres dans une maison, comme la température, la présence de fumée, les mouvements, ou encore l'humidité.

Le système sera composé de plusieurs capteurs reliés à un microcontrôleur. Les données collectées par ces capteurs seront affichées via à un écran LCD où elles pourront être surveillées en temps réel. Ce système permettra aussi d'envoyer des alertes automatiques en cas d'anomalies, comme la détection de fumée ou un changement soudain de température.

1. ***ETAT DE LA QUESTION***

L’évolution rapide de l’**Internet des Objets (IoT)** et des **systèmes embarqués** a considérablement transformé le domaine de la domotique et de la surveillance résidentielle. Aujourd’hui, les maisons intelligentes ne se contentent plus de proposer des fonctionnalités de confort, mais elles intègrent également des dispositifs avancés de sécurité pour protéger les occupants et leurs biens.

Dans ce contexte, de nombreux systèmes de surveillance domotique ont vu le jour, combinant capteurs, caméras, alarmes et applications mobiles pour offrir un contrôle en temps réel. Cependant, plusieurs défis persistent, notamment :

* **La fiabilité des capteurs** : certains capteurs peuvent être sujets à des erreurs de lecture ou à des défaillances techniques.
* **La vitesse de transmission des données** : les systèmes doivent être capables de réagir en temps réel aux événements détectés.
* **L’accessibilité et l’ergonomie des interfaces** : la gestion du système doit être intuitive et simple à utiliser, même pour des utilisateurs non technophiles.

De nombreuses solutions existantes reposent sur des plateformes en ligne (applications mobiles, sites web) pour afficher et gérer les informations de surveillance. Cependant, notre projet adopte une approche différente : **toutes les interactions et notifications seront affichées sur un écran LCD intégré au système, sans dépendre d’une plateforme externe**.

1. ***PROBLEMATIQUE***

Avec l'augmentation des besoins en sécurité dans les maisons, les systèmes de surveillance intelligents utilisant des capteurs sont de plus en plus utilisés. Ces capteurs permettent de surveiller en temps réel des paramètres comme la température, la fumée, les mouvements ou l'humidité. Cependant, plusieurs questions se posent :

* Comment concevoir un système de surveillance domotique autonome intégrant des capteurs IoT pour détecter et signaler efficacement les situations anormales dans un environnement domestique ?
* En quoi l’utilisation d’un écran LCD comme interface principale de notification permet-elle d’améliorer l’accessibilité et l’efficacité du système de surveillance domotique ?
* Quels sont les avantages et les limites techniques de la modélisation et des tests d’un système de surveillance domotique sur Tinkercad avant une intégration réelle ?

1. ***HYPOTHESES***

**L'intégration de capteurs IoT dans un système de surveillance domotique autonome, avec l'affichage des notifications sur un écran LCD, peut offrir une solution efficace, accessible et fiable pour la surveillance domestique, en éliminant la dépendance aux infrastructures réseau et en centralisant l'ensemble des interactions sur un dispositif local.**

L’objectif de cette hypothèse est de démontrer que la combinaison de capteurs IoT (mouvement, température, fumée, etc.) avec un affichage local sur un écran LCD peut permettre de concevoir un système de surveillance domestique performant, capable de détecter en temps réel les événements anormaux dans un environnement domestique, et d'alerter l'utilisateur de manière autonome et efficace. Ce système, en étant totalement indépendant des applications mobiles ou des plateformes web, pourrait résoudre plusieurs problèmes liés à l'accessibilité, à la sécurité des données et à la dépendance à l'infrastructure internet.

* L’une des principales difficultés rencontrées par les systèmes de surveillance actuels réside dans la nécessité d’une **connexion internet stable** et d'un **accès à une plateforme externe**, ce qui peut poser problème dans des zones rurales ou des environnements où la couverture réseau est limitée. De plus, ces solutions exigent souvent des **compétences techniques** pour gérer les applications mobiles ou les interfaces web, ce qui peut rendre leur utilisation complexe pour une large portion de la population.
* En revanche, l’utilisation d’un **écran LCD comme interface unique** permettrait de simplifier l’interaction avec le système. L’utilisateur n’aurait qu’à consulter les informations sur l’écran pour être informé des alertes en temps réel (intrusion, fumée, température anormale, etc.), sans avoir besoin d'une connexion à Internet ou de compétences avancées en technologie. Le système pourrait également être conçu de manière à inclure des **indicateurs visuels (LEDs)** et **sonores (buzzer)**, renforçant l'alerte et assurant une notification immédiate en cas de problème détecté. Ces **mécanismes de notification multimodaux** rendraient l'alerte plus intuitive et rapide à percevoir, même dans un environnement bruyant ou dans des situations où l’utilisateur serait éloigné de l’écran.
* L’utilisation de **Tinkercad** dans le processus de développement permet de valider la faisabilité technique de ce système avant de passer à une intégration physique réelle. Cette plateforme de simulation permet de tester les interactions entre les composants électroniques, de vérifier leur compatibilité, et de simuler leur fonctionnement dans un environnement contrôlé, réduisant ainsi les risques d'erreurs lors de la fabrication réelle. En simulant des comportements de capteurs et d'affichages sur Tinkercad, il est possible d'optimiser le design et de repérer des éventuels problèmes techniques ou de performance.

*Conclusion*

Cette hypothèse repose sur l'idée que **l'autonomie, l'efficacité et la simplicité d’utilisation d’un système de surveillance domestique basé sur des capteurs IoT et un affichage LCD peuvent offrir une solution robuste et accessible**, même dans des environnements où l’accès à Internet est limité ou où les utilisateurs ne possèdent pas de compétences techniques avancées. Ce système pourrait ainsi contribuer à renforcer la **sécurité domestique** en offrant une solution de surveillance **en temps réel** tout en étant **indépendant des plateformes en ligne**.

1. ***METHODES ET TECHNIQUES UTILISÉS***

Dans le cadre de ce travail, nous avons utilisé une méthode expérimentale pour créer et tester un système de surveillance domotique avec des capteurs. Cette méthode nous a permis de vérifier si le système fonctionne bien dans des situations réelles et d’apporter des améliorations si nécessaires.

Voici les étapes de cette méthode, expliquées simplement :

***1. Choix des composants :***

***Objectif :*** Trouver les capteurs et le microcontrôleur nécessaires pour le système.

***Action :*** Nous avons choisi des capteurs pour mesurer la température, la fumée, les mouvements et l'humidité, ainsi qu'un microcontrôleur pour gérer les données. Nous avons sélectionné ces composants en fonction de leur compatibilité et de leurs performances.

***2. Montage du système :***

***Objectif :*** Assembler tous les éléments du système.

***Action :*** Nous avons connecté les capteurs au microcontrôleur. Ensuite, nous avons programmé le microcontrôleur pour qu'il puisse collecter et envoyer les données des capteurs en temps réel.

***3. Test du système en conditions réelles :***

***Objectif :*** Vérifier que le système fonctionne bien dans un environnement réel.

***Action :*** Nous avons installé le système et simulé des situations comme l’augmentation de la température ou la détection de fumée. Ces tests nous ont permis de voir comment le système réagit et s’il envoie les alertes correctement.

***4. Collecte des données :***

***Objectif :*** Enregistrer les informations collectées par les capteurs.

***Action :*** Pendant les tests, nous avons enregistré les valeurs mesurées par les capteurs (comme la température ou l'humidité) et noté si les alertes étaient envoyées rapidement et correctement

***5. Analyse des résultats :***

***Objectif :*** Évaluer si le système est efficace et réactif.

***Action :*** Nous avons comparé les résultats obtenus avec ce que nous attendions. Cela nous a permis de vérifier la précision des capteurs et la rapidité avec laquelle le système envoyait les alertes en cas de problème.

***6. Améliorations du système :***

***Objectif :*** Améliorer le système en fonction des résultats des tests.

***Action :*** En analysant les résultats, nous avons fait des ajustements pour améliorer la précision des capteurs et la rapidité des alertes. Par exemple, nous avons modifié la programmation du microcontrôleur ou ajusté certains capteurs.

***7. Validation du système :***

***Objectif :*** Vérifier que tout fonctionne bien après les modifications.

**Action :** Après les ajustements, nous avons effectué des tests finaux pour nous assurer que le système fonctionne comme prévu, en envoyant des alertes fiables et en surveillant les données en temps réel.

***TECHNIQUE UTILISEES : La Simulation***

Dans le cadre de ce travail, une des techniques utilisées est la simulation. Cette méthode permet de tester le système de surveillance domotique dans un environnement virtuel avant de le construire physiquement. Voici comment nous avons utilisé cette technique :

* ***Choix de l’outil de simulation :***

Nous avons choisi un logiciel de simulation, comme Tinkercad, pour simuler le comportement des capteurs (température, fumée, mouvement, humidité) et du microcontrôleur (Arduino) sans avoir besoin de les assembler physiquement dès le départ.

* ***Modélisation des capteurs :***

Dans la simulation, nous allons créer un modèle de chaque capteur. Cela nous permet de voir comment chaque capteur réagit dans différentes conditions (par exemple, une augmentation de température ou la détection de fumée).

* ***Programmation du microcontrôleur :***

Nous allons simulée la programmation du microcontrôleur en créant des scripts qui collectent les données des capteurs et Cela permet de tester la transmission des données et la communication entre les capteurs.

* ***Test des scénarios :***

Nous avons utilisé la simulation pour créer plusieurs scénarios, comme la détection de fumée ou la variation de température. Cela nous permet de vérifier si le système réagit correctement et envoie les alertes comme prévu.

* ***Analyse des résultats de simulation :***

Les données collectées à partir de la simulation sont ensuite analysées pour s'assurer que le système fonctionne bien. Cela permet de corriger d’éventuelles erreurs dans le programme ou d'ajuster les paramètres des capteurs avant de passer à la construction réelle. [2]

1. ***CHOIX ET INTERET DU SUJET***

Le choix du développement d’un système de surveillance domotique est motivé par la nécessité croissante de sécuriser les maisons. Grâce aux capteurs, ce système permet de surveiller en temps réel plusieurs paramètres comme la température, la fumée, ou les mouvements, la lumière offrant ainsi une réponse rapide aux situations dangereuses.

Ce sujet est particulièrement intéressant car Il permet non seulement de renforcer la sécurité des foyers, mais aussi d'améliorer le confort en automatisant certaines tâches. Pour un étudiant, ce projet offre une opportunité d'acquérir des compétences techniques en travaillant avec des capteurs, des microcontrôleurs.

1. ***DELIMITATION DU SUJET***

Cette étude se concentre sur l'évaluation de capteurs spécifiques, notamment ceux de température, de fumée, de mouvement et d'humidité, dans un cadre de test contrôlé. Nous avons choisi d'examiner des scénarios typiques de surveillance domestique, tout en évitant des situations plus complexes ou extrêmes qui pourraient être abordées dans de futures recherches. Les expérimentations ont été réalisées sur une période s'étendant du 08/07/2024 jusqu'au 30/03/2025

1. ***SUBDIVISION DU TRAVAIL***

Outre l'introduction générale et la conclusion générale, notre travail est subdivisé en 4 chapitres, à savoir :

CHAPITRE I. LA DOMOTIQUE : Dans ce chapitre, nous aborderons la Définition de la domotique ainsi que son historique ses piliers, confort et sécurité et sa structure.

CHAPITRE II. INTRODUCTION AUX CAPTEURS IOT ET MICROCONTROLEURS : Dans ce chapitre, nous présenterons les principes fondamentaux des capteurs IoT, leur fonctionnement et les différents types utilisés dans notre système. Nous aborderons également le rôle des microcontrôleurs dans l'intégration des capteurs.

CHAPITRE III. CONCEPTION DU SYSTÈME DE SURVEILLANCE : Ce chapitre décrira les étapes de conception du système, y compris l'analyse des besoins, le choix des capteurs et des composants matériels, ainsi que l'architecture du système et les protocoles de communication utilisés.

CHAPITRE IV. RÉALISATION DU SYSTEME DOMOTIQUE : Dans ce chapitre, nous parlerons de l'intégration des données des capteurs et les fonctionnalités d'alerte en temps réel.

# LA DOMOTIQUE

## *2.1 Introduction*

Aujourd'hui, la domotique est au cœur de l'habitat que ce soit en termes de confort, d'énergie ou de sécurité. Il est bien entendu indispensable qu'à son installation elle soit programmée par un expert, mais ensuite ce dispositif d'automatisation boosté fonctionne en toute autonomie. Avec la domotique, on ne voit plus la maison du même ciel. Grâce à elle, on vit désormais dans un univers de rêve, délicieusement confortable, hautement sécurisé, très économique. [2]



*Fig1 La domotique*

## *2.2 Définition de la domotique*

La domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans le bâtiment, plus ou moins interopérables et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prise électrique, etc.) [3]

La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut trouver dans les maisons, les hôtels, les lieux publics...etc. [4]

A l'origine, la domotique avait donc pour but d'automatiser sa maison: ouverture et fermeture automatiques des volets, ouverture du portail électrique, gestion du chauffage, gestion de l'éclairage, etc. Ainsi avant l'ère des smartphones, il était par exemple possible d'activer son chauffage à distance en passant un coup de téléphone à sa maison, ou encore en lui envoyant un SMS. C'était tout à fait réalisable. Seulement une telle installation était relativement compliquée à mettre en place elle était coûteuse. [5]

Cette époque a malheureusement laissé des traces, puisque pour beaucoup encore aujourd'hui, domotique rime avec cher et compliqué. Pourtant, ce domaine a énormément évolué et il existe de nombreuses solutions simples à mettre en place et tout à fait abordables pour le grand public. [6]

## *2.3 Historique de la domotique*

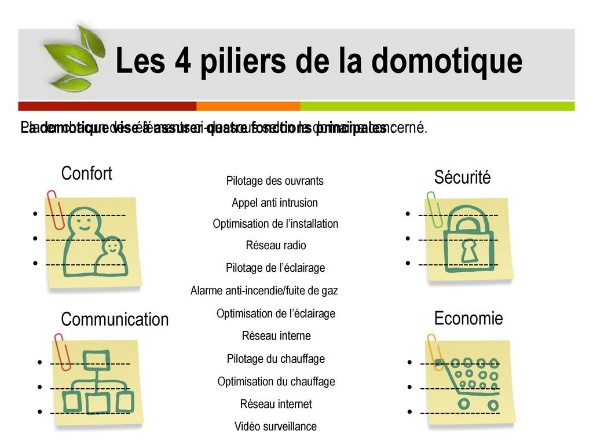
Les premiers travaux de domotique sont apparus dans les années 70 avec les problématiques énergétiques dues aux crises pétrolières.

Ces crises marquent le début du développement de l'électronique pour les bâtiments. Au départ, la domotique contrôle seulement les prises, l'éclairage et les volets roulants grâce à une télécommande. Au fur et mesure, de nouveaux objets se mettent en réseau comme les thermostats et les alarmes. Mais c'est véritablement à partir de la fin du 20e siècle, que la domotique va se démocratiser. [7]

Deux raisons expliquent ce développement: L'arrivée de l'ordinateur et des technologies de communication dans la maison au début des années 1990; notamment, le déploiement d'Internet qui permet aux ordinateurs de communiquer entre eux. Le coût de l'énergie qui augmente suite aux deux crises pétrolières survenues dans les années 70. Désormais, de nouvelles normes forcent les constructeurs privilégiés des bâtiments bien mieux isolés pour limiter leur utilisation chauffage. [8]

La domotique intervient donc avec des appareils capables de communiquer entre eux pour surveiller et gérer cette énergie. Depuis les années 2000, avec le développement des technologies sans fil comme le wifi ou le Bluetooth, la miniaturisation des composants électroniques, l'avènement des appareils mobiles, l'invasion des écrans tactiles et des télévisions connectées, les ingénieurs peuvent désormais proposer au public des produits objets connectés ou systèmes domotique bien plus puissants et simples d'utilisation. [9]

## *2.4 Les Piliers de la domotique*



*Fig2 Les Piliers de la domotique*

Cette nouvelle approche efficace est conçue pour des objectifs très varié présentant de vifs intérêts qui répondent aux attentes des populations et particulièrement aux besoins primordiaux des handicapés et offrant des opportunités pour les gens en difficultés :

**Le confort :** Actuellement, les logements sont correctement équipés au niveau appareillages électriques ce qui offre un certain confort mais s'accompagne de multitudes contraintes car rien n'a été pensé pour connecter ces appareils. La domotique vous permet de transformer votre habitation en un outil intelligent, à votre service, qui évolue avec vous, le système simplifie les activités quotidiennes, La centralisation des contrôles permet de gagner du temps, de connaître instantané ment l'état de tous les systèmes de la maison (éclairages, températures... etc.) et de les modifier à votre convenance. [10]

Contrôle par la voix la domotique possède permet une technique qui est plus sophistiquer et le contrôle des appareilles de l'habitas par la voix humaine. L'habitat doit répondre aux requêtes d'un utilisateur concernant un ensemble d'actions pouvant être automatisées (contrôle des lumières, des volets, des dispositifs multimédias, etc.). [11]

**La sécurité :** La domotique vise aussi un objectif de sécurisation des maisons. Vous pouvez en effet équiper vos logements de systèmes de sécurité performants grâce aux techniques de la domotique. Les systèmes de sécurité peuvent aussi être automatisés pour offrir une bonne protection aux personnes qui vivent dans la maison ainsi qu'à leurs biens. Elle est assurée par des capteurs telle que: [12]

Détection d'une présence ou d'une intrusion (détecteur de mouvement ou d'accès);

Surveillance de l'accès par des pièces du logement;

Transmission et visualisation des alarmes sur incident.

Les Alarmes technique: Le système domotique peut vous alerter en local ou à distance en cas d'anomalies des équipements suivants:

* Ventilateur à l'arrêt
* Détection fuite de gaz
* Fuite d'eau;
* Incendie
* Fumée
* Intrusion par force

**Économie d'énergie :** Un aspect essentiel de la domotique est la réduction de la consommation énergétique. Des capteurs intelligents permettent d’optimiser le chauffage, la climatisation, et l’éclairage en fonction de la présence des occupants et des conditions climatiques, contribuant ainsi à diminuer les factures énergétiques. [13]

**Communication :** Les systèmes domotiques permettent la communication entre différents appareils, souvent via l'Internet des objets (IoT). Cette inter-connectivité permet aux appareils de fonctionner de manière synchronisée et de fournir des informations en temps réel aux utilisateurs. [14]

## *2.5 Structure d'un système domotique*

Le système domotique regroupe l'ensemble des outils permettant d'actionner nos dispositifs électroniques. Alors il comprend de nombreux équipements tels que: - Unité de gestion. Capteurs. Actionneurs - Protocol de communication. [15]

**Unité de gestion :** L'unité de gestion joue le rôle d'une intelligence centralisé, elle pilote diverses unités de la maison qui peuvent fonctionne d'une manière autonome. Elle est composée de deux principaux éléments : Unité centrale C'est une carte de développement qui centralise tous les Information et déclenche des actions comme: Arduino, Raspberry pi. Le logiciel domotique: toutes les cartes de développement nécessitent un logiciel pour leur fonctionnement, c'est lui qui fait le traitement des informations afin d'obtenir le résultat voulu. [16]

**Capteurs :** Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable, mesurable et interprétable. Ils ont pour objectif de recueillir des informations présentes dans l'environnement. Ils détectent des phénomènes physiques tels que chaleur, lumière, son, pression, mouvement, position ou champ magnétique. Ces informations sont en- suite transmises à une unité centrale qui va définir le type d'action à effectuer. [17]

**Actionneurs :** Les actionneurs sont des dispositifs qui font des actions, en répondant à des commandes programmées, ces actions peuvent agir sur certains éléments (les portes, les volets...).

**Protocol de communication :** Le protocole domotique est le langage utilisé pour établir la communication entre les différents appareils connectés. Pour pouvoir communiquer entre eux, deux appareils doivent donc avoir le même langage de communication, c'est-à-dire le même protocole. [18]

## *2.6 Les différentes Technologies de la domotique*

Généralement, une installation domotique peut être conçue sur quatre principaux types de technologie. Ces technologies sont employées pour la transmission d'informations entre les équipements d'un système d'automatisation ou de communication :

Les technologies de la domotique, ou systèmes de maisons intelligentes, utilisent divers outils pour automatiser, contrôler et surveiller les appareils et les environnements domestiques. Voici un aperçu des principales technologies employées en domotique : [19]

## *1. IoT (Internet des Objets)*

L'IoT est la base de la domotique moderne, permettant aux appareils de se connecter et de communiquer via Internet. Les appareils IoT recueillent, échangent et réagissent aux données en temps réel, facilitant le contrôle à distance des équipements via des applications mobiles. [20]

## *2. Protocole Z-Wave*

Z-Wave est un protocole de communication sans fil utilisé dans de nombreux appareils domotiques. Sa faible consommation d'énergie et sa portée étendue (jusqu'à 100 mètres) en font un choix populaire pour les systèmes de sécurité, de chauffage, et de surveillance. [21]

## *3. Protocole Zigbee*

Zigbee est une technologie sans fil à faible consommation d'énergie adaptée aux réseaux en maillage, où chaque appareil agit comme un relais pour transmettre les données. Cela permet un réseau solide et extensible pour contrôler l'éclairage, les capteurs de mouvement, et les serrures intelligentes. [22]

## *4. Bluetooth ET BLE (Bluetooth Low Energy)*

Le Bluetooth classique et BLE sont utilisés pour connecter des appareils de domotique de proximité. BLE, en particulier, consomme très peu d’énergie, idéal pour les capteurs, les serrures intelligentes, et les systèmes d'alarme. [23]

## *5. Wi-Fi*

Le Wi-Fi est couramment utilisé pour la domotique, en particulier pour les appareils nécessitant un transfert de données élevé (comme les caméras de sécurité). Son avantage principal est qu'il utilise le réseau domestique existant, bien que cela puisse engendrer une consommation d'énergie plus élevée. [24]

## *6. IFTTT (If This Then That)*

IFTTT est une technologie qui permet de créer des automatisations personnalisées entre différents services et appareils. Par exemple, elle peut allumer les lumières lorsqu'un mouvement est détecté, ou activer le thermostat selon l'heure de la journée. [25]

## *7. Reconnaissance vocale (assistants vocaux)*

Des assistants vocaux comme Alexa d'Amazon, Google Assistant, ou Siri d'Apple permettent de contrôler des appareils via des commandes vocales. En intégrant des appareils IoT, ils ajoutent un niveau de confort et de contrôle mains libres pour l’utilisateur. [26]

## *8. Technologies RFID et NFC*

RFID et NFC permettent une identification et un contrôle d’accès sans contact. En domotique, ils sont souvent utilisés pour les systèmes de sécurité et d'ouverture de portes par reconnaissance de badge ou de smartphone. [26]

## *9. Infrarouge (IR)*

La technologie IR permet le contrôle d'appareils comme les téléviseurs ou les climatiseurs via des télécommandes universelles connectées, offrant un moyen simple d'intégrer les appareils traditionnels dans le réseau domotique. [27]

## *10. Systèmes de gestion de l’énergie*

Les technologies de gestion énergétique surveillent la consommation en temps réel et optimisent l’utilisation de l'énergie. Les compteurs intelligents, thermostats connectés, et systèmes de panneaux solaires sont souvent intégrés pour réduire la consommation. [28]

## ***Conclusion***

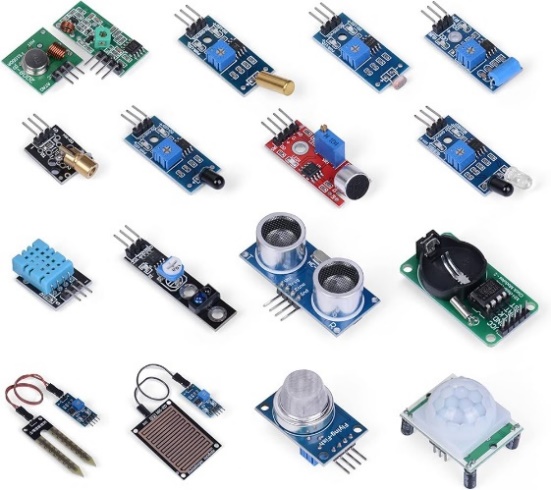
La domotique transforme notre manière de vivre en rendant nos maisons plus intelligentes, pratiques, et adaptées à nos besoins. En combinant électronique, automatisation, informatique et télécommunications, elle permet de centraliser le contrôle des différents appareils de la maison, améliorant ainsi le confort, la sécurité, et l’efficacité énergétique.

Son évolution, depuis les premiers systèmes coûteux et complexes des années 70 jusqu’aux solutions modernes abordables et simples à utiliser, reflète les progrès des technologies sans fil, des objets connectés, et de l’intelligence artificielle. Aujourd’hui, grâce à des systèmes comme IoT, Z-Wave, Zigbee, et Bluetooth, la domotique offre des outils puissants pour répondre aux besoins des utilisateurs, notamment ceux des personnes en difficulté ou à mobilité réduite.

Les piliers de la domotique, que sont le confort, la sécurité, les économies d’énergie et la communication, montrent comment cette technologie révolutionne notre quotidien, tout en proposant des solutions adaptées aux enjeux environnementaux et énergétiques actuels.

# INTRODUCTION AUX CAPTEURS IOT ET MICROCONTROLEURS

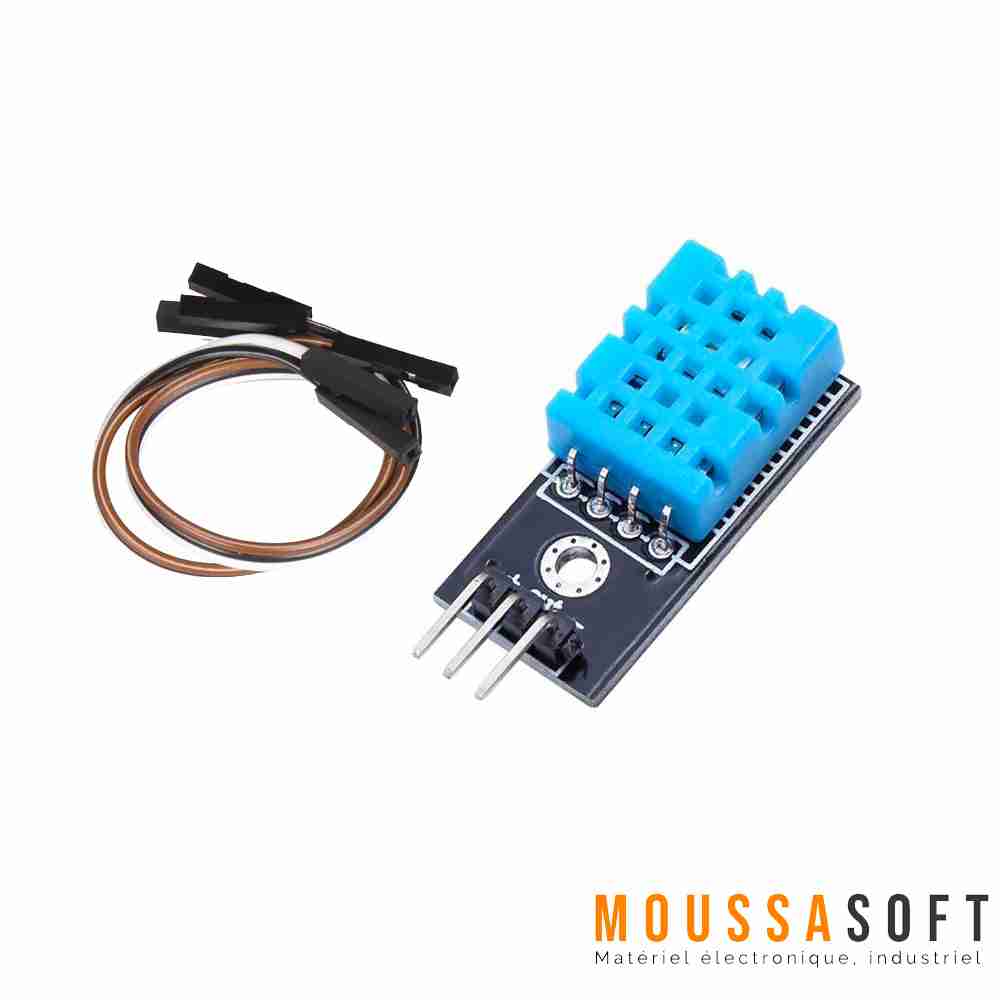
## *3.1 Types de capteurs IoT*



*Fig3 Types de capteurs IoT*

Dans le cadre de ce travail de mémoire sur le système de surveillance domotique, nous utiliserons quatre types de capteurs IoT principaux : capteurs de température, capteurs de fumée, capteurs de mouvement et capteurs d'humidité. Chacun de ces capteurs joue un rôle crucial dans la surveillance des paramètres environnementaux à l'intérieur d'une maison. Voici un aperçu de leur fonctionnement et principes respectifs. [29]

## *3.1.1 Capteur de température*



*Fig4 Capteur de température*

## Principe de fonctionnement

Un capteur de température IoT mesure la chaleur ou le froid dans un environnement donné. Il fonctionne généralement à l'aide d'une thermistance (capteur à résistance variable) ou d'un thermocouple. Dans le cas de la thermistance, la résistance électrique change en fonction de la température. Les données sont ensuite envoyées au microcontrôleur, qui les interprète et affiche la température en temps réel. [30]

## *Application dans notre sujet*

Dans un système de surveillance domestique, un capteur de température permet de surveiller les variations de température dans une pièce et de déclencher des alertes si celle-ci dépasse des seuils prédéfinis, ce qui peut signaler un incendie ou une panne de système de climatisation/chauffage. [31]

## *3.1.2 Capteur de fumée*



*Fig5 Capteur de fumée*

## *Principe de fonctionnement*

Les capteurs de fumée utilisent généralement un capteur optique ou un capteur à ionisation. Dans un capteur optique, un faisceau lumineux est réfléchi par des particules de fumée, ce qui active un signal d'alerte. Les capteurs à ionisation détectent des particules de combustion en perturbant le courant entre deux électrodes. [32]

## *Application dans notre sujet*

Le capteur de fumée, connecté à un système IoT, permet de détecter la présence de fumée dans une habitation, ce qui est essentiel pour prévenir des incendies. En cas de détection de fumée, une alerte est automatiquement envoyée à l’utilisateur via l’application mobile ou web. [33]

## *3.1.3 Capteur de mouvement*



*Fig6 Capteur de mouvement*

## *Principe de fonctionnement*

Les capteurs de mouvement fonctionnent généralement avec des technologies infrarouges passives (PIR) ou à ultrasons. Les capteurs PIR détectent les variations de rayonnement infrarouge émises par des objets en mouvement. Lorsqu’un objet chaud, comme un être humain, passe devant le capteur, celui-ci enregistre une modification dans le rayonnement infrarouge, déclenchant ainsi un signal. [34]

## *Application dans notre sujet*

Le capteur de mouvement est essentiel pour la surveillance de la sécurité domestique, car il détecte toute présence inhabituelle dans les zones surveillées. Le système peut envoyer une alerte instantanée à l'utilisateur lorsqu'un mouvement est détecté dans une zone où personne n'est censé se trouver. [35]

## *3.1.4 Capteur d'humidité*



*Fig7 Capteur d'humidité*

## *Principe de fonctionnement*

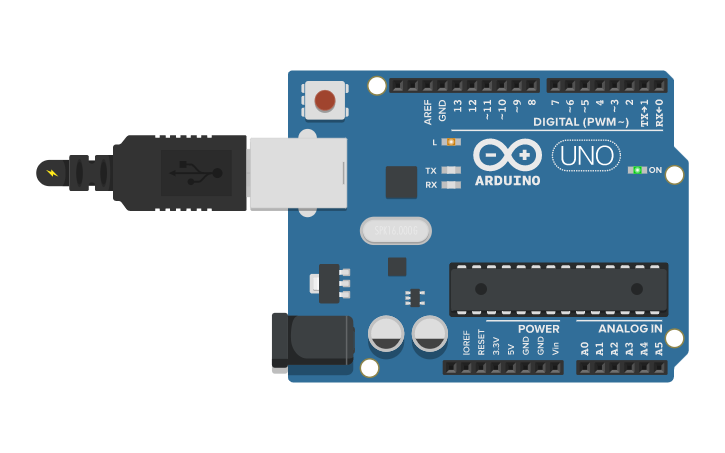
Les capteurs d'humidité mesurent la quantité de vapeur d'eau présente dans l'air. Ils sont basés sur un élément sensible dont la résistance électrique change en fonction du niveau d'humidité. Ces capteurs utilisent soit un capteur capacitif soit un capteur à résistance pour détecter l'humidité relative dans l'air. [36]

## *Application dans notre sujet*

Un capteur d'humidité intégré dans un système de surveillance domotique permet de détecter une augmentation d'humidité, ce qui pourrait indiquer une fuite d'eau ou des conditions propices à la formation de moisissures. Cela permet à l’utilisateur d’agir rapidement pour prévenir d’éventuels dégâts matériels. [37]

## *4.1 Microcontrôleurs dans les systèmes Iot*

## *4.1.2 Rôles des microcontrôleurs dans les systèmes Iot*



*Fig8 microcontrôleur*

Les microcontrôleurs jouent un rôle central dans les systèmes IoT (Internet of Things).   
Ils sont chargés de contrôler les capteurs, de collecter les données, de les traiter et de les transmettre à une plateforme centrale, souvent via Internet.

## *Principe de fonctionnement*

Ils servent de "cerveau" du système IoT, assurant la coordination et l'interaction entre différents éléments du système. [38]

## *4.1.3 Capteur de force*

**

*Fig9 Capteur de force*

Un **capteur de force** est un dispositif électronique qui mesure la force appliquée sur une surface et la convertit en un signal électrique. Il est largement utilisé en **robotique, industrie, biomécanique et domotique** pour détecter des pressions, des impacts ou des charges.

*Principe de fonctionnement*

Un capteur de force fonctionne généralement selon l'un des principes suivants :

* **Variation de résistance** (résistif)
* **Déformation mécanique** (jauges de contrainte)
* **Effet piézoélectrique** (génération de tension sous pression)

Lorsque la force appliquée augmente, la sortie électrique change, permettant une mesure proportionnelle.

## *Principaux rôles des microcontrôleurs dans un système IoT :*

* **Collecte de données :** Les microcontrôleurs reçoivent les informations des capteurs (température, mouvement, fumée, humidité, etc.).
* **Traitement des données :** Ils analysent ou traitent les données avant de les transmettre, selon le besoin.
* **Transmission des données :** Ils communiquent les données traitées vers un serveur ou une plateforme cloud via des protocoles de communication comme Wi-Fi, Bluetooth, ou MQTT.
* **Automatisation et contrôle :** Ils peuvent également déclencher des actions automatisées en fonction des données recueillies, comme activer une alarme en cas de détection de fumée. [39]

## *4.1.4 Comparaison des microcontrôleurs populaires*

Voici une comparaison des microcontrôleurs les plus utilisés dans les systèmes IoT, notamment l'Arduino, l'ESP8266 et l'ESP32 : [40]

|  |  |  |  |
| --- | --- | --- | --- |
| Microcontrôleur | Caractéristiques principales | Avantages | Inconvénients |
| Arduino (UNO) | 8-bit ATmega328P, 16 MHz, RAM : 2 KB, Pas de Wi-Fi intégré | Facile à utiliser, Large communauté, Nombreuses bibliothèques disponibles | Pas de Wi-Fi ou Bluetooth intégré, Puissance de calcul limitée |
| ESP8266 | 32-bit, 80 MHz, RAM : 160 KB, Wi-Fi intégré | Wi-Fi intégré pour un prix abordable, Compact, Consommation d'énergie réduite | Mémoire RAM limitée, Moins de ports GPIO que l'Arduino |
| ESP32 | 32-bit dual-core, 240 MHz, RAM : 520 KB, Wi-Fi + Bluetooth intégrés | Puissance de calcul élevée, Wi-Fi et Bluetooth intégrés, Nombreux ports GPIO | Configuration plus complexe pour les débutants, Moins de support qu'Arduino |

## *4.1.5 Interfaces et Communication avec les Capteurs*

Pour que les microcontrôleurs puissent interagir avec les capteurs, ils utilisent différentes interfaces de communications telles que GPIO, I2C, SPI, ou UART.   
Les microcontrôleurs comme l'Arduino, l'ESP8266 ou l'ESP32 peuvent utiliser ces interfaces pour communiquer avec les capteurs. [41]

## *Types d'interfaces couramment utilisées*

* **GPIO (Entrée/Sortie générale) :** Utilisée pour lire des données simples des capteurs, connectés directement aux broches du microcontrôleur.
* **I2C (Inter-Integrated Circuit)** : Utilisée pour connecter plusieurs capteurs sur les mêmes broches du microcontrôleur avec seulement deux fils (données et horloge).
* **SPI (Serial Peripheral Interface) :** Une interface rapide utilisée pour des capteurs nécessitant des transferts rapides comme les capteurs de pression.
* **UART (Universal Asynchronous Receiver-Transmitter) :** Utilisée pour la communication série, avec des capteurs comme le GPS ou les modules GSM. [42]

## *Exemple de communication avec un capteur*

* **Capteur de température (DHT11) :** Communique via une interface GPIO, où le capteur envoie des données numériques au microcontrôleur sous forme d’impulsions électriques.
* **Capteur d'humidité (BME280) :** Utilise l'interface I2C pour envoyer des données relatives à l'humidité et la température à un microcontrôleur. [43]

## *5.1 Protocoles des Communications Iot*

## *5.1.2 Protocoles Courantes (MQTT, HTTP, CoAP)*

## *5.1.3 MQTT (Message Queuing Telemetry Transport)*

MQTT est un protocole léger basé sur la publication/abonnement, conçu pour les appareils ayant des ressources limitées et nécessitant des communications fiables dans des réseaux à faible bande passante. [44]

## ***Avantages***

* Très léger, idéal pour les environnements à faible bande passante.
* Fonctionne sur le modèle de publication/abonnement, permettant une communication efficace entre les appareils.
* Fiabilité avec trois niveaux de QoS (Quality of Service). [45]

*Inconvénients* ***:*** Moins adapté aux transferts de gros volumes de données.

## *4.1.4 HTTP (Hypertext Transfer Protocol)*

Protocole web standard utilisé pour la communication entre les appareils et les serveurs. Il est simple à utiliser et largement pris en charge, mais n’est pas optimisé pour les systèmes IoT.

## ***Avantages***

* Universellement utilisé et compris.
* Bien supporté par les plateformes web.
* Facile à implémenter.

## ***Inconvénients***

* Lourd en termes de bande passante et de ressources.
* Moins efficace pour des systèmes IoT nécessitant des communications fréquentes ou en temps réel. [46]

## *4.1.5 CoAP (Constrained Application Protocol)*

CoAP est un protocole de communication spécialement conçu pour les appareils contraints dans les systèmes IoT. Il est basé sur l'architecture REST, similaire à HTTP, mais avec une empreinte plus légère. [47]

Avantages

* Léger et optimisé pour les appareils avec peu de ressources.
* Prend en charge les communications machine-to-machine (M2M).
* Moins de surcharge que HTTP.

## ***Inconvénients***

* Moins utilisé et supporté que HTTP.
* Pas aussi flexible que MQTT en termes de QoS.

## ***Conclusion***

Dans ce chapitre, nous avons expliqué les composants importants d’un système de surveillance domotique : les capteurs IoT et les microcontrôleurs. Nous avons présenté les différents types de capteurs utilisés (température, fumée, mouvement et humidité), en précisant leur rôle et leur utilité pour surveiller les conditions dans une maison. Ces capteurs aident à détecter rapidement des problèmes comme un incendie, une fuite d’eau ou une intrusion.

Nous avons aussi parlé des microcontrôleurs, qui coordonnent tout le système en collectant, traitant et envoyant les données des capteurs. Une comparaison entre les microcontrôleurs populaires (Arduino, ESP8266, ESP32) nous a permis de choisir ceux qui conviennent le mieux à ce projet. Enfin, nous avons étudié les protocoles de communication et retenu MQTT comme le plus adapté, grâce à sa légèreté et sa fiabilité.

# CONCEPTION DU SYSTEME DE SURVEILLANCE

## *4.1. Architecture du Système*

L'architecture d'un système de surveillance domotique utilisant des capteurs IoT repose sur plusieurs éléments clés. Elle doit permettre de collecter les données des capteurs, de les transmettre de manière efficace et de les afficher aux utilisateurs en temps réel. Le système se compose de trois parties principales : les capteurs, le microcontrôleur, et le réseau de transmission des données. [48]

## *4.1.1. Description du Schéma Global du Système*

**Le schéma global du système de surveillance inclut :**

* **Les capteurs IoT :** Ces capteurs sont placés à différents endroits de la maison pour surveiller des paramètres comme les mouvements, la température, l'humidité, ou encore la fumée, la lumière.
* **Le microcontrôleur :** C’est l’élément central qui reçoit les données des capteurs. Il traite ces informations et les envoie à une plateforme centrale (comme un serveur ou une application). [49]

## *4.1.2. Intégration des Capteurs et des Microcontrôleurs*

L’intégration des capteurs et des microcontrôleurs est une étape cruciale. Chaque capteur doit être relié au microcontrôleur de façon à ce qu'il puisse transmettre ses données. Le microcontrôleur agit comme un “**cerveau**” du système. [50]

## *Les capteurs peuvent être de différents types*

* **Capteurs de mouvement :** Ils détectent les mouvements dans une pièce.
* **Capteurs de température :** Ils surveillent les variations de température.
* **Capteurs de fumée :** Ils détectent la présence de fumée ou de gaz dangereux.
* **Capteurs d'humidité :** Ils contrôlent le niveau d'humidité dans l'air.
* **Capteurs de lumière :** La photorésistance est un capteur qui mesure l'intensité lumineuse.
* **Capteur de force :** Mesure la force appliquer sur une surface ou un objet. [51]

Ces capteurs sont connectés au microcontrôleur à l’aide de fils ou de connexions sans fil comme le Wi-Fi ou le Bluetooth, selon le type de capteur. Une fois connectés, ils envoient continuellement des données au microcontrôleur qui, à son tour, les traite et les analyse. [52]

## *5. Développement de l'infrastructure de collecte des données*

## *5.1. Configuration des capteurs et des microcontrôleurs*

La configuration des capteurs et des microcontrôleurs est une étape essentielle pour assurer la bonne collecte des données. Voici comment procéder :

*1. Sélection des capteurs :* Chaque capteur doit être choisi en fonction du paramètre à surveiller. Par exemple :

**Capteurs de mouvement (PIR)** **:** pour détecter les déplacements dans les zones surveillées.

**Capteurs de température et d'humidité (DHT11 ou DHT22) :** pour contrôler l’environnement intérieur.

**Capteurs de fumée :** pour détecter d’éventuels incendies ou gaz nocifs.

**Capteurs de force** **:** pour détecter la force appliquer sur une surface.

*2. Connexion au microcontrôleur :* Les capteurs sont connectés à un microcontrôleur comme un Arduino. Cela peut se faire par des connexions filaires ou sans fil (par exemple, Wi-Fi ou Bluetooth).

## *5.2. Programmation pour la collecte et la transmission des données*

Une fois les capteurs connectés au microcontrôleur, il faut programmer ce dernier pour collecter les données et les transmettre. Voici les étapes clés :

*5.2 Récupération des données des capteurs :* Le microcontrôleur est programmé en C++ (pour Arduino) Chaque capteur envoie des données à intervalles réguliers. Par exemple, un capteur de température peut être programmé pour envoyer une lecture toutes les 5 minutes. [53]

## *6. Gestion des alertes et des anomalies*

## *6.1. Définition des critères d'alerte*

Les critères d'alerte sont essentiels pour déterminer quand une anomalie doit déclencher une action immédiate. Ces critères dépendent des valeurs limites ou seuils définis pour chaque capteur. Pour chaque capteur IoT installé (température, mouvement, fumée, etc.), des seuils spécifiques sont configurés pour signaler des anomalies : [54]

* **Température :** Une température supérieure à 30°C ou inférieure à 9°C déclenche une alerte pour signaler une surchauffe ou une température dangereusement basse.
* **Mouvement :** Un capteur de mouvement peut être configuré pour envoyer une alerte uniquement s’il détecte une présence dans une zone sensible, comme l'intérieur de la maison, durant des horaires définis (par exemple, la nuit).
* **Fumée/Gaz :** La détection de fumée ou de gaz dangereux comme le CO2 ou le méthane entraîne une alerte instantanée, car cela pourrait indiquer un incendie ou une fuite de gaz.
* **Intrusion :** La détection d’une force intrusion dans une domotique pourrait indiquer la présence des voleurs forçant la porte à enter dans une maison.

Ces seuils d'alerte sont ajustés selon le type de capteur et les préférences de sécurité des utilisateurs. [55]

## ***6.2. Détection des anomalies***

Pour la détection d'anomalies, plusieurs méthodes simples peuvent être appliquées sans recourir à des algorithmes complexes :

*1. Surveillance des seuils critiques :* Le système surveille constamment les données en provenance des capteurs pour détecter toute valeur dépassant ou n’atteignant pas les seuils définis. Lorsque cela se produit, une alerte est déclenchée. Par exemple, si la température dépasse 40°C, cela est immédiatement détecté comme une anomalie. [56]

*2. Surveillance des comportements anormaux :* En plus des seuils fixes, il est possible de surveiller les comportements anormaux à travers l’analyse des tendances des capteurs sur une période définie. Par exemple, une montée graduelle de l'humidité pourrait signaler une fuite d'eau avant même que l'alerte des seuils ne soit atteinte. [57]

*3. Vérification croisée :* En utilisant plusieurs capteurs pour mesurer des paramètres similaires (ex. : capteurs de fumée et de gaz), le système peut confirmer la véracité de l'anomalie détectée avant d'envoyer une alerte, réduisant ainsi les faux positifs. [58]

## *6.3. Mise en place du système d'alerte (notifications push, emails, etc.)*

Une fois une anomalie détectée, il est essentiel que le système envoie rapidement des alertes grâce au buzzer en affichant le danger via un écran LCD à utilisateurs pour qu'ils puissent réagir en temps réel. Voici les différentes méthodes pour la gestion des alertes :

* **Détection des anomalies** : Cela pourrait inclure des mouvements suspects (intrusion) ou des conditions environnementales (comme la température ou l'humidité extrêmes).
* **Alerte de l'utilisateur** : En cas de détection, le système doit immédiatement notifier l'utilisateur par des moyens visuels (via l'écran LCD) et sonores (via le buzzer).
* **Affichage des messages de danger** : Des messages d'alerte sont affichés sur un écran LCD pour fournir à l'utilisateur une vue d'ensemble des conditions dangereuses détectées. [59]

## ***Conclusion***

Dans ce chapitre, nous avons détaillé la conception d’un système de surveillance domotique basé sur des capteurs IoT. Nous avons décrit l'architecture globale du système, comprenant les capteurs, le microcontrôleur, et la plateforme de surveillance. Chaque élément joue un rôle précis : les capteurs collectent les données, le microcontrôleur les traite et les transmet, et la plateforme permet à l’utilisateur de suivre l'état de sa maison en temps réel.

## CHAPITRE IV. RÉALISATION DU SYSTEME DOMOTIQUE

## ***4.1 Introduction***

Ce chapitre détaille la mise en œuvre du système de surveillance domotique, en expliquant le choix des technologies, l’architecture du système, le développement du programme et l’affichage des informations sur un écran LCD. Il expose également les tests réalisés pour valider le bon fonctionnement du système. [60]

## ***4.2 Choix des technologies et outils***

### ***4.2.1 Matériels utilisés***

Le choix des composants matériels a été guidé par les exigences du projet en termes de fiabilité, d’efficacité et de compatibilité.

* **Microcontrôleur :**
  + Arduino Uno / ESP32
* **Capteurs :**
  + **Capteur PIR** : Détecte les mouvements pour signaler une intrusion.
  + **Capteur de température et d’humidité (DHT11/DHT22)** : Permet de surveiller l’environnement intérieur.
  + **Capteur de gaz (MQ-2/MQ-135)** : Détecte la présence de gaz toxiques ou inflammables.
  + **Capteur de lumière** : Sa résistance fournis varie en fonction de la lumière.
  + **Capteur de force**: mesure la force appliquée sur une surface et la convertit en un signal électrique.
* **Actionneurs :**
  + **LEDs** : Indiquent visuellement les alertes et l’état du système.
  + **Buzzer** : Émet un son en cas d’alerte (intrusion, fuite de gaz, etc.).
  + **Relais** : Active ou désactive un dispositif externe comme une alarme puissante.
  + **Lampe** : elle s’allume en fonction de l’intensité la lumière et commandé à travers un relais
  + **Moteur DC** : elle joue le rôle d’un ventilateur une fois que le seuil de la température est dépassé elle se met automatiquement en marche
  + **Potentiomètre** : utiliser pour augmenter ou diminuer le contraste ou la lumière de l’écran LCD
  + **BLOC D’ALIMENTATION** : fournit du courant continue pour alimenter le moteur DC ainsi que la lumière
  + **Bouton poussoir** : Envoie un signal ON ou OFF lorsqu’ils sont pressés
* **Écran LCD** :
  + Sert d’interface utilisateur en affichant les relevés des capteurs et les messages d’alerte en temps réel. [61]

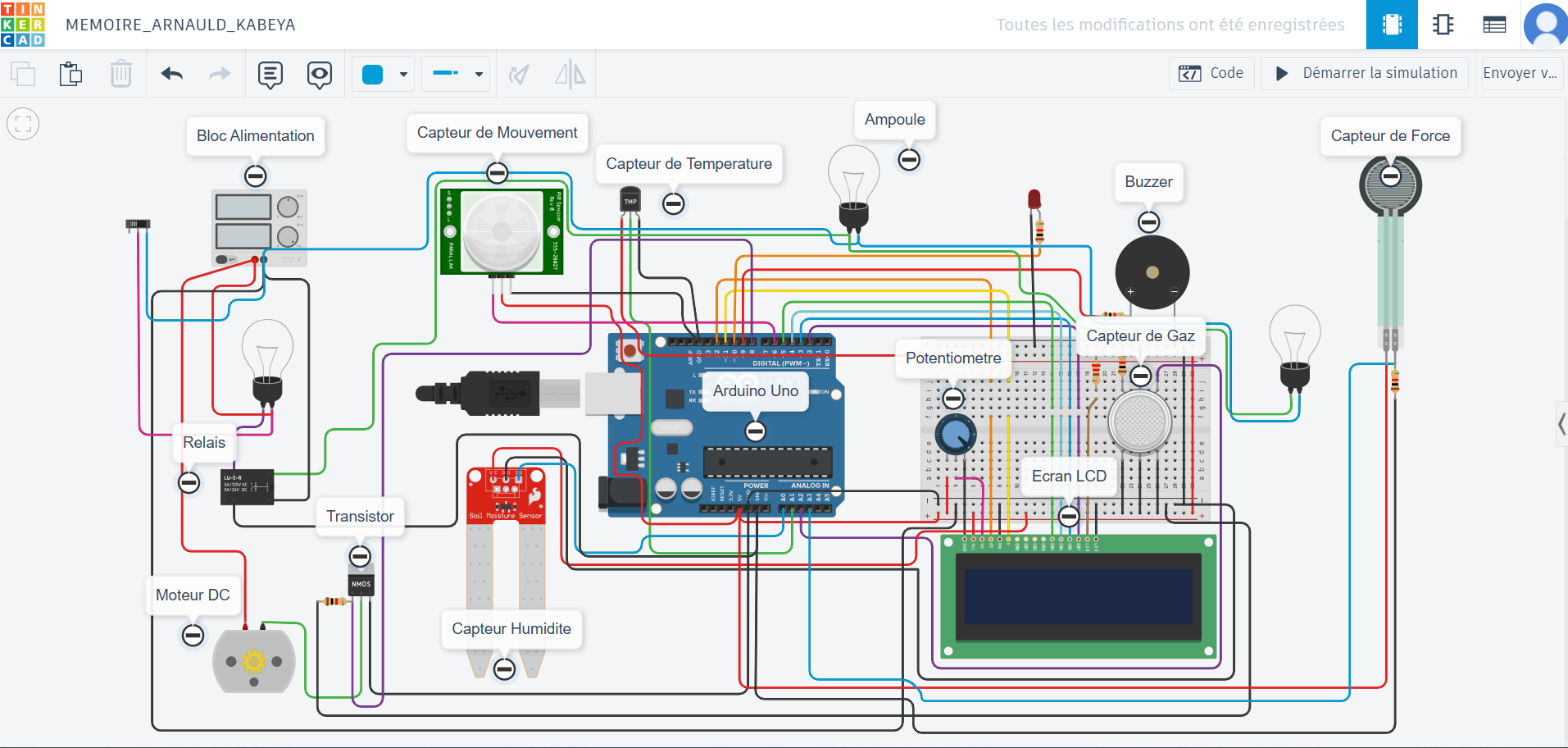
### ***4.2.2 Logiciels et outils de développement***

* **Tinkercad** : Utilisé pour simuler les circuits avant leur implémentation physique.
* **Bibliothèques Arduino** :
  + LiquidCrystal.h : Gestion de l’affichage sur l’écran LCD.
  + DHT.h : Gestion du capteur de température et d’humidité.
  + MQUnifiedsensor.h : Gestion du capteur de gaz.

## ***4.3 Conception et architecture du système***

### ***4.3.1 Architecture générale du système***

L’architecture repose sur une **communication directe entre les capteurs, les microcontrôleur et l’écran LCD**. Voici le schéma fonctionnel du système :



*Fig10 Réalisation*

### ***4.3.2 Fonctionnalités du système***

Le système domotique assure plusieurs fonctionnalités :

* **Détection d’intrusion** : Lorsqu’un mouvement est détecté par le capteur PIR, et une force sur le capteur de force un message d’alerte s’affiche sur l’écran LCD, et le buzzer se déclenche.
* **Surveillance de la température et de l’humidité et de Sécurité** : L’écran LCD affiche les relevés du capteur en continu.
* **Détection de gaz** : Si une concentration anormale est détectée, une alarme sonore est activée et un message d’alerte apparaît sur l’écran.
* **Activation automatique des actionneurs** :
  + Les LEDs et le buzzer se déclenchent en fonction des alertes détectées.

## ***4.4 Développement et programmation***

### ***4.4.1 Programmation du microcontrôleur***

Le programme est écrit en langage **C++**. Il gère la lecture des capteurs, l’analyse des données et l’activation des actionneurs.

#### **Exemple de code : Lecture du capteur PIR et affichage sur LCD**

#include <LiquidCrystal.h>

const int pirPin = 2;

const int ledPin = 13;

const int buzzerPin = 12;

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup() {

pinMode(pirPin, INPUT);

pinMode(ledPin, OUTPUT);

pinMode(buzzerPin, OUTPUT);

lcd.begin(16, 2);

lcd.print("Systeme Pret");

}

void loop() {

int mouvement = digitalRead(pirPin);

if (mouvement == HIGH) {

digitalWrite(ledPin, HIGH);

digitalWrite(buzzerPin, HIGH);

lcd.clear();

lcd.print("Intrusion!");

} else {

digitalWrite(ledPin, LOW);

digitalWrite(buzzerPin, LOW);

lcd.clear();

lcd.print("Aucun Mouvement");

}

delay(1000);

}

#### **Autres fonctionnalités programmées :**

* Lecture et affichage des relevés du capteur de température/humidité.
* Gestion des alertes de détection de gaz.
* Activation automatique des LEDs et du buzzer selon les seuils définis.

### ***4.4.2 Gestion de l’affichage sur l’écran LCD***

L’écran LCD affiche dynamiquement les informations en fonction des données des capteurs.  
**Exemples d’affichage :**

* Température et humidité :
* Temp: 28°C
* Humidité: 60%
* Détection de gaz :
* Alerte Gaz !
* Intrusion détectée :
* Mouvement !

## ***4.6 Conclusion***

Ce chapitre a présenté la mise en œuvre du système domotique, de la sélection des composants à la programmation du microcontrôleur. Les tests ont démontré que le système fonctionne correctement et atteint ses objectifs.

## CHAPITRE V. TEST ET VALIDATION DU SYSTEME

#### 4.1 Introduction

Ce chapitre est consacré aux tests et à la validation du système de surveillance domotique développé. Il vise à vérifier le bon fonctionnement des capteurs, du microcontrôleur, de l’affichage sur l’écran LCD ainsi que des différentes alertes. Les tests seront effectués dans un environnement simulé à l’aide de Tinkercad afin d’assurer la fiabilité du système avant une éventuelle implémentation réelle. [62]

#### 4.2 Objectifs des tests

Les tests effectués ont pour but de :

* Vérifier la connectivité et le bon fonctionnement des capteurs (mouvement, température, gaz, etc.).
* Tester l’interaction entre les capteurs et le microcontrôleur (Arduino).
* Valider l'affichage des informations sur l’écran LCD.
* Vérifier le déclenchement des alertes sonores et visuelles en cas d’anomalie.
* Analyser les performances du système en termes de rapidité de détection et d’affichage.

#### 4.3 Protocole de test

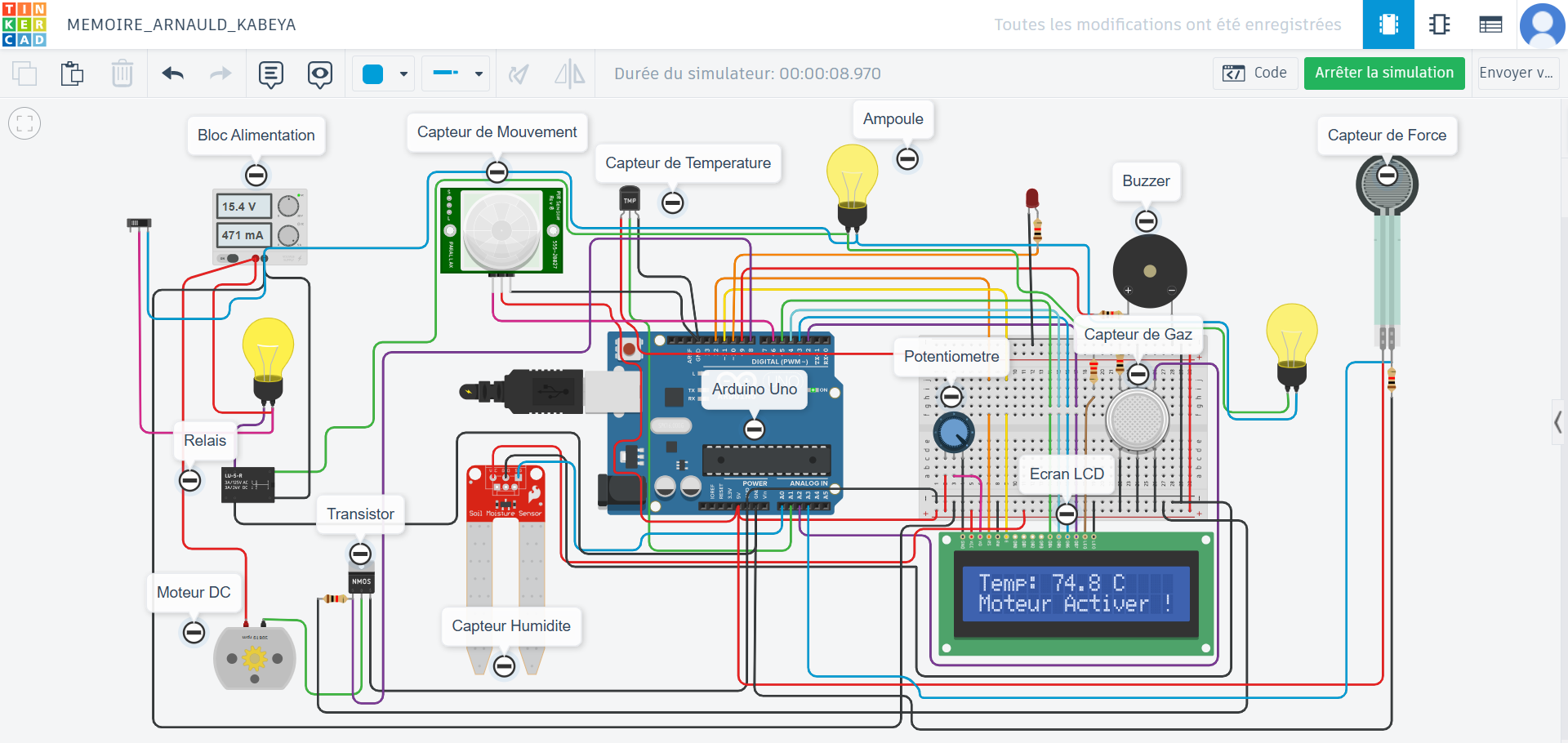
Les tests ont été réalisés en suivant un protocole bien défini :

1. **Connexion et initialisation des composants**
   * Vérification des connexions entre les capteurs et l’Arduino dans Tinkercad.
   * Exécution du code pour s’assurer que le microcontrôleur reconnaît correctement les capteurs.
2. **Test des capteurs individuels**
   * **Capteur de mouvement (PIR)** : Simulation d’une présence pour observer la réaction du capteur et l’affichage du message correspondant sur l’écran LCD.
   * **Capteur de température et d’humidité (DHT11/DHT22)** : Vérification des valeurs affichées et de la précision des mesures.
   * **Capteur de gaz (MQ-2/MQ-135)** : Simulation d’une fuite de gaz et test de la réaction du système (affichage d’un avertissement et déclenchement du buzzer).
3. **Test des alertes et affichage**
   * Observation de l’affichage des messages sur l’écran LCD en fonction des scénarios de test.
   * Vérification du fonctionnement du buzzer en cas d’alerte.
   * Test des LEDs indicatrices (rouge pour alerte, verte pour état normal).
4. **Validation de l’ensemble du système**
   * Exécution du programme complet avec plusieurs scénarios simulés.
   * Vérification de la synchronisation entre la détection des événements et l’affichage sur l’écran LCD.
   * Analyse du temps de réponse du système. [63]

#### 4.4 Résultats des tests

Les tests effectués ont permis d’évaluer le bon fonctionnement du système. Voici les principaux résultats :

* Tous les capteurs ont correctement détecté les événements simulés.
* L’écran LCD a affiché les informations en temps réel avec une mise à jour fluide.
* Les alertes sonores et visuelles se sont déclenchées conformément aux attentes.
* Le système a réagi en moins d’une seconde après la détection d’un événement.



*Fig11 Test et Validation*

#### 4.5 Limites et améliorations possibles

Bien que le système fonctionne correctement, certaines améliorations peuvent être envisagées :

* Optimisation du temps de réponse pour un affichage encore plus rapide.
* Intégration de capteurs supplémentaires pour une surveillance plus complète.
* Ajout d’une batterie de secours pour garantir le fonctionnement en cas de coupure d’électricité.

#### 4.6 Difficultés rencontrées

Un relais consomme généralement un certain courant pour fonctionner, en particulier pour activer son électro-aimant (appelé aussi bobine). Voici comment cela fonctionne et pourquoi un relais pourrait consommer plus que ce que la broche Arduino peut fournir :

### *1.* ***Structure du relais***

Un relais est composé de plusieurs éléments : une bobine (ou électro-aimant), des contacts, et souvent une diode de protection. C'est la bobine qui consomme le courant, et son but est de créer un champ magnétique lorsqu'elle est alimentée, ce qui permet de déplacer un contact et de fermer ou ouvrir un circuit.

### *2.* ***Consommation de courant de la bobine***

* La bobine du relais a une certaine résistance. Quand elle est alimentée, elle demande un certain courant pour créer le champ magnétique nécessaire à son fonctionnement.
* Le courant nécessaire dépend des caractéristiques du relais, notamment de sa résistance interne. Par exemple, si la résistance de la bobine est faible, elle nécessitera un courant plus élevé pour créer le champ magnétique. [65]

### *3.* ***Pourquoi 28 mA et non 20 mA ?***

* Connecté le relais directement à une broche de l'Arduino (par exemple D8), la broche de l'Arduino peut ne pas être capable de fournir tout le courant nécessaire pour alimenter correctement la bobine.
* Un relais standard peut consommer entre 30 mA et 100 mA en fonction de sa conception (parfois plus si la bobine est de grande taille). Cela dépasse largement la capacité des broches Arduino, qui sont limitées à environ 20 mA par broche. C’est probablement pour cette raison que vous voyez 28 mA sur la broche D8. [66]

#### **Conclusion**

Ce chapitre a présenté les tests réalisés sur le système de surveillance domotique, ainsi que leur validation. Les résultats montrent que le système fonctionne correctement et répond aux exigences initiales. Les performances obtenues sont satisfaisantes, mais des améliorations peuvent être apportées pour optimiser encore davantage le système.

#### **5. Code Source**

## // C++ code

## #include <LiquidCrystal.h>

## #define LM35\_PIN A1

## #define GAZ\_PIN A2

## #define HUMIDITY\_PIN A0

## #define FORCE\_SENSOR A3

## #define PIR\_PIN 6

## #define LED\_PIN 10

## #define BUZZER\_PIN 9

## #define MOTOR\_PIN 8

## #define RELAY\_PIN 7

## int seuilGaz = 85;

## float seuilTemperature = 30.0;

## int seuilHumidite = 50;

## int seuilForce = 250;

## LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

## void setup() {

## lcd.begin(16, 2);

## lcd.print("Temp: ");

## pinMode(LED\_PIN, OUTPUT);

## pinMode(BUZZER\_PIN, OUTPUT);

## pinMode(MOTOR\_PIN, OUTPUT);

## pinMode(RELAY\_PIN, OUTPUT);

## pinMode(PIR\_PIN, INPUT);

## Serial.begin(9600);

## }

## void afficherAlerte(const char\* message) {

## lcd.setCursor(0, 1);

## lcd.print(" ");

## lcd.setCursor(0, 1);

## lcd.print(message);

## }

## void activerAlerte() {

## for (int i = 0; i < 5; i++) {

## digitalWrite(LED\_PIN, HIGH);

## tone(BUZZER\_PIN, 1000);

## delay(2000);

## digitalWrite(LED\_PIN, LOW);

## noTone(BUZZER\_PIN);

## delay(2000);

## }

## }

## void loop() {

## int valeurGaz = analogRead(GAZ\_PIN);

## bool mouvement = digitalRead(PIR\_PIN);

## int force = analogRead(FORCE\_SENSOR);

## 

## int sensorValue = analogRead(LM35\_PIN);

## float voltage = sensorValue \* (5.0 / 1023.0);

## float temperature = voltage \* 100.0;

## 

## float humidite = analogRead(HUMIDITY\_PIN) \* (100.0 / 1023.0);

## lcd.setCursor(6, 0);

## lcd.print(" ");

## lcd.setCursor(6, 0);

## lcd.print(temperature, 1);

## lcd.print(" C");

## Serial.print("Gaz : ");

## Serial.println(valeurGaz);

## Serial.print("Mouvement : ");

## Serial.println(mouvement ? "Detecté" : "Aucun");

## Serial.print("Humidité : ");

## Serial.print(humidite);

## Serial.println("%");

## Serial.print("Force : ");

## Serial.println(force);

## bool alerteDeclenchee = false;

## if (force > seuilForce) {

## afficherAlerte("Porte Attaquer !");

## Serial.println("⚠️ ALERTE : Tentative de destruction détectée !");

## activerAlerte();

## alerteDeclenchee = true;

## }

## if (mouvement) {

## afficherAlerte("Mouvement !");

## Serial.println("⚠️ Mouvement détecté!");

## activerAlerte();

## alerteDeclenchee = true;

## }

## if (valeurGaz > seuilGaz) {

## afficherAlerte("Alerte: Gaz !");

## Serial.println("⚠️ ALERTE : Gaz détecté !");

## activerAlerte();

## alerteDeclenchee = true;

## }

## if (humidite > seuilHumidite) {

## afficherAlerte("Alerte: Humidite !");

## Serial.println("⚠️ ALERTE : Forte humidité détectée !");

## activerAlerte();

## alerteDeclenchee = true;

## }

## if (temperature > seuilTemperature) {

## digitalWrite(MOTOR\_PIN, HIGH);

## afficherAlerte("Moteur Activer !");

## Serial.println("🌀 Moteur activer !");

## alerteDeclenchee = true;

## } else {

## digitalWrite(MOTOR\_PIN, LOW);

## }

## if (!alerteDeclenchee) {

## afficherAlerte("Systeme normal");

## }

## 

## delay(100);

## }

## *CONCLUSION GENERALE*

Ce travail s’est inscrit dans le cadre du **développement d’un système de surveillance domotique**, visant à renforcer la sécurité d’un domicile grâce à l’intégration de capteurs et d’un microcontrôleur. L’objectif principal était de concevoir un système capable de détecter différents événements (mouvement, variation de température, présence de gaz, etc.) et d’afficher les informations en temps réel sur un écran LCD, tout en déclenchant des alertes visuelles et sonores en cas d’anomalie.

Pour atteindre cet objectif, nous avons d’abord étudié les principes de la **domotique et des systèmes embarqués**, puis défini l’architecture matérielle et logicielle du projet. Le développement et les tests ont été réalisés dans un environnement simulé à l’aide de **Tinkercad**, permettant de valider l’interconnexion des composants et leur bon fonctionnement. Les résultats des tests ont montré que le système réagit efficacement aux événements détectés, en affichant les informations pertinentes et en activant les alertes dans un délai très court.

Malgré ces performances satisfaisantes, certaines améliorations peuvent être envisagées, notamment l’ajout de nouvelles fonctionnalités comme la gestion à distance via un module de communication ou l’optimisation de la consommation énergétique du système. Ces perspectives d’évolution ouvrent la voie à un **système de surveillance plus avancé, connecté et intelligent**, capable de mieux répondre aux besoins de la sécurité domestique.

Ce projet a démontré la faisabilité et l’efficacité d’une solution de surveillance domotique basée sur des composants accessibles et une approche modulaire. Il constitue une base solide pour de futurs développements et s’inscrit dans une dynamique d’innovation visant à rendre les habitations plus sûres et intelligentes. [64]

## ***REFERENCES***

## ***WEBOGRAPHIE***

[1]<https://www.ibm.com/fr-fr/topics/internet-of-things>

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

[3]<https://www.ibm.com/fr-fr/topics/internet-of-things>

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

[5]<https://www.softwareag.com/fr_fr/resources/iot/guide/internet-of-things.html>

[6]<https://www.softwareag.com/fr_fr/resources/iot/guide/internet-of-things.html>

[7]<https://www.arduino.cc/>

[8]<https://www.raspberrypi.org/>

[9]<https://www.sparkfun.com/>

[10]<https://www.adafruit.com/>

[11]<https://www.arduino.cc/reference/en/>

[12]<https://www.espressif.com/en/products/socs/esp32/resources>

[13]<https://learn.adafruit.com/>

[14]<https://learn.sparkfun.com/tutorials/tags/internet-of-things>

[15]<https://mqtt.org/>

[16]<https://developer.mozilla.org/en-US/docs/Web/HTTP>

[17]<https://coap.technology/>

[18]<https://www.hivemq.com/mqtt-essentials/>

[19]<https://learn.adafruit.com/iot-protocols>

[20]<https://knowhow.distrelec.com>

[21]<https://demeter-fb.fr>

[22]<https://www.dusuniot.com>

[23]<https://fr.digi.com>

[24]<https://iotfactory.eu>

[25] <https://fr.wikipedia.org/wiki/Home_Assistant>

[26]<https://fr.wikipedia.org/wiki/Homey>

[27]<https://fr.wikipedia.org/wiki/LoRaWAN>

[28][<https://hal.archives-ouvertes.fr/hal-01234567>]

[29][<https://hal.archives-ouvertes.fr/hal-02345678>]

[30][<https://tel.archives-ouvertes.fr/tel-03456789>]

[31][<https://fr.wikipedia.org/wiki/Contiki>]

[32][<https://fr.wikipedia.org/wiki/CoAP>]

[33][<https://owasp.org/www-project-internet-of-things/>]

[34][<https://www.sciencedirect.com/science/article/pii/S0167739X15003018>]

[35]<http://www.home-assistant.io/>

[36]<http://homey.app/>

[37]<http://www.jeedom.com/>

[38]<http://www.domoticz.com/>

## ***Ouvrages***

[39] **Jean-Baptiste Waldner**, Nanocomputers and Swarm Intelligence, ISTE Ltd and John Wiley & Sons Inc, 2007.

[40] **Jean-Baptiste Waldner**, Nanoinformatique et intelligence ambiante, Hermes Science Publications, 2006.

[41] **Ovidiu Vermesan** et **Peter Friess** (Éds.), Internet of Things – From Research and Innovation to Market Deployment, River Publishers, 2014.

[42] **Daniel Giusto**, **Antonio Iera**, **Giacomo Morabito** et **Luigi Atzori** (Éds.), The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications, Springer, 2010.

[43] **Adrian McEwen** et **Hakim Cassimally**, Designing the Internet of Things, Wiley, 2013.

[44] **Samuel Greengard**, The Internet of Things, MIT Press, 2015.

[45] **Hannu Tenhunen** et **Arto Lehto** (Éds.), Internet of Things: Research and Development Towards a Connected World, Springer, 2017.

[46] **Pethuru Raj** et **Anupama C. Raman**, The Internet of Things: Enabling Technologies, Platforms, and Use Cases, CRC Press, 2017.

[47] **Olivier Hersent**, **David Boswarthick** et **Omar Elloumi**, The Internet of Things: Key Applications and Protocols, Wiley, 2012.

[48] **Dietmar P.F. Möller**, Guide to Computing Fundamentals in Cyber-Physical Systems: Concepts, Design Methods, and Applications, Springer, 2016.

## ***Articles scientifiques***

[49] **Tiago M. Fernández-Caramés** et **Paula Fraga-Lamas**, « A Review on the Use of Blockchain for the Internet of Things », IEEE Access, vol. 6, pp. 32979-33001, 2018.

[50] **Hong-Ning Dai**, **Zibin Zheng** et **Yan Zhang**, « Blockchain for Internet of Things: A Survey », IEEE Internet of Things Journal, vol. 6, no 5, pp. 8076-8094, 2019.

[51] **Nacer Khalil**, **Mohamed Riduan Abid**, **Driss Benhaddou** et **Michael Gerndt**, « Wireless Sensors Networks for Internet of Things », Procedia Computer Science, vol. 52, pp. 210-217, 2015.

[52] **Charith Perera**, **Chi Harold Liu** et **Srimal Jayawardena**, « The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey », IEEE Transactions on Emerging Topics in Computing, vol. 3, no 4, pp. 585-598, 2015.

[53] **Rongxing Lu**, **Xiaohui Liang**, **Xu Li**, **Xiaodong Lin** et **Xuemin (Sherman) Shen**, « EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications », IEEE Transactions on Parallel and Distributed Systems, vol. 23, no 9, pp. 1621-1631, 2012.

[54] **Alessio Botta**, **Walter de Donato**, **Valerio Persico** et **Antonio Pescapé**, « Integration of Cloud computing and Internet of Things: A survey », Future Generation Computer Systems, vol. 56, pp. 684-700, 2016.

[55] **Li Da Xu**, **Wu He** et **Shancang Li**, « Internet of Things in Industries: A Survey », IEEE Transactions on Industrial Informatics, vol. 10, no 4, pp. 2233-2243, 2014.

[56] **Giancarlo Fortino**, **Wassim Gaia** et **Antonio Guerrieri**, « Modeling and Simulating Internet of Things Systems: A Hybrid Agent-Oriented Approach », Computing in Science & Engineering, vol. 19, no 5, pp. 68-76, 2017.

[57] **Surya Nepal**, **Shiping Chen** et **Jinjun Chen**, « Security and Privacy for the Internet of Things: A Survey », IEEE Internet of Things Journal, vol. 5, no 1, pp. 1-16, 2018.

[58] **Mohammad Ali Salahuddin**, **Kashif Ahmad**, **Erchin Serpedin** et **Mounir Ben Ghorbel**, « IoT Traffic Characterization: From Generation and Measurement to Classification », IEEE Network, vol. 32, no 6, pp. 18-26, 2018.

## ***Mémoires et Thèses***

[59] **Mohammed Islam Naas**, Placement des données de l'internet des objets dans le cloud, Thèse de doctorat, Université de Bretagne Occidentale, 2019.

[60] **Abdelkader Bouchareb**, Sécurisation de l'Internet des objets, Mémoire de maîtrise, Université du Québec à Trois-Rivières, 2021.

[61] **Saad Benkirane**, Modèle collaboratif pour l'Internet of Things (IoT), Mémoire de maîtrise, Université du Québec à Chicoutimi, 2014.

[62] **Nicolas Le Sommer**, Gestion de la mobilité et de la connectivité dans les réseaux de capteurs sans fil, Thèse de doctorat, Université de Bretagne Sud, 2008.

[63] **Julien Bouaziz**, Conception et optimisation de protocoles de communication pour les réseaux de capteurs sans fil, Thèse de doctorat, Université Paris-Est, 2010.

[64] **Yassine Hadjadj-Aoul**, Gestion de la qualité de service dans les réseaux de capteurs sans fil, Thèse de doctorat, Université de Rennes 1, 2008.

[65] **Hafid Haffaf**, Sécurité et qualité de service dans les réseaux de capteurs sans fil, Thèse de doctorat, Université de Béjaïa, 2011.

[66] **Amine Dhraief**, \*Conception et évaluation de protocol

Table des Matières

EPIGRAPHE…………………………………………………………………………………………....I

DEDICACE………………………………………………………………………………….………....II

REMERCIEMENTS…………………………………………………………………………………..III

LISTE DES FIGURES………………………………………………………………………………..IV

LISTE DES TABLEAUX……………………………………………………………….......................V

LISTES DES ABREVIATIONS ET ACRONYMES………………………………………………....VI

RESUMÉ……………………………………………………………………………………………..VII

ABSTRACT………………………..………………………………………………………………..VIII

INTRODUCTION GENERALE……………………………………………………………………...IX

ETAT DE LA QUESTION………………………………………………………………………….…..1

PROBLÉMATIQUE...…………………………………………………………………….....................1

HYPOTHESE …...…………………………………………………………………………………......2

METHODES ET TECHNIQUES UTILISÉS…………………………………………………………..2

PROBLEMATIQUE……………………………………………………………………………………2

CHOIX ET INTERET DU SUJET……………………………………………………………………..4

DELIMITATION DU SUJET ………………………..………………………………………………...4

SUBDIVISON DU TRAVAIL ……………...………………………………………………………….4

[**CHAPITRE I.** LA DOMOTIQUE 7](#_Toc193387676)

[*2.1 Introduction* 7](#_Toc193387677)

[*2.2 Définition de la domotique* 7](#_Toc193387678)

[*2.3 Historique de la domotique* 7](#_Toc193387679)

[*2.4 Les Piliers de la domotique* 8](#_Toc193387680)

[*2.5 Structure d'un système domotique* 9](#_Toc193387681)

[*2.6 Les différentes Technologies de la domotique* 10](#_Toc193387682)

[*1. IoT (Internet des Objets)* 10](#_Toc193387683)

[*2. Protocole Z-Wave* 10](#_Toc193387684)

[*3. Protocole Zigbee* 10](#_Toc193387685)

[*4. Bluetooth ET BLE (Bluetooth Low Energy)* 10](#_Toc193387686)

[*5. Wi-Fi* 10](#_Toc193387687)

[*6. IFTTT (If This Then That)* 11](#_Toc193387688)

[*7. Reconnaissance vocale (assistants vocaux)* 11](#_Toc193387689)

[*8. Technologies RFID et NFC* 11](#_Toc193387690)

[*9. Infrarouge (IR)* 11](#_Toc193387691)

[*10. Systèmes de gestion de l’énergie* 11](#_Toc193387692)

[***Conclusion*** 11](#_Toc193387693)

[**CHAPITRE II.** INTRODUCTION AUX CAPTEURS IOT ET MICROCONTROLEURS 12](#_Toc193387694)

[*3.1 Types de capteurs IoT* 12](#_Toc193387695)

[*3.1.1 Capteur de température* 12](#_Toc193387696)

[Principe de fonctionnement 12](#_Toc193387697)

[*Application dans notre sujet* 12](#_Toc193387698)

[*3.1.2 Capteur de fumée* 13](#_Toc193387699)

[*Principe de fonctionnement* 13](#_Toc193387700)

[*Application dans notre sujet* 13](#_Toc193387701)

[*3.1.3 Capteur de mouvement* 13](#_Toc193387702)

[*Principe de fonctionnement* 13](#_Toc193387703)

[*Application dans notre sujet* 13](#_Toc193387704)

[*3.1.4 Capteur d'humidité* 14](#_Toc193387705)

[*Principe de fonctionnement* 14](#_Toc193387706)

[*Application dans notre sujet* 14](#_Toc193387707)

[*4.1 Microcontrôleurs dans les systèmes Iot* 14](#_Toc193387708)

[*4.1.2 Rôles des microcontrôleurs dans les systèmes Iot* 14](#_Toc193387709)

[*Principe de fonctionnement* 14](#_Toc193387710)

[*4.1.3 Capteur de force* 15](#_Toc193387711)

[*Principe de fonctionnement* 15](#_Toc193387712)

[*Principaux rôles des microcontrôleurs dans un système IoT :* 15](#_Toc193387713)

[*4.1.4 Comparaison des microcontrôleurs populaires* 16](#_Toc193387714)

[*4.1.5 Interfaces et Communication avec les Capteurs* 16](#_Toc193387715)

[*Types d'interfaces couramment utilisées* 16](#_Toc193387716)

[*Exemple de communication avec un capteur* 16](#_Toc193387717)

[*5.1 Protocoles des Communications Iot* 17](#_Toc193387718)

[*5.1.2 Protocoles Courantes (MQTT, HTTP, CoAP)* 17](#_Toc193387719)

[*5.1.3 MQTT (Message Queuing Telemetry Transport)* 17](#_Toc193387720)

[***Avantages*** 17](#_Toc193387721)

[***Inconvénients*** 17](#_Toc193387722)

[*4.1.4 HTTP (Hypertext Transfer Protocol)* 17](#_Toc193387723)

[***Avantages*** 17](#_Toc193387724)

[***Inconvénients*** 17](#_Toc193387725)

[*4.1.5 CoAP (Constrained Application Protocol)* 17](#_Toc193387726)

[***Inconvénients*** 17](#_Toc193387727)

[***Conclusion*** 18](#_Toc193387728)

[**CHAPITRE III.** CONCEPTION DU SYSTEME DE SURVEILLANCE 19](#_Toc193387729)

[*4.1. Architecture du Système* 19](#_Toc193387730)

[*4.1.1. Description du Schéma Global du Système* 19](#_Toc193387731)

[*4.1.2. Intégration des Capteurs et des Microcontrôleurs* 19](#_Toc193387732)

[*Les capteurs peuvent être de différents types* 19](#_Toc193387733)

[*5. Développement de l'infrastructure de collecte des données* 19](#_Toc193387734)

[*5.1. Configuration des capteurs et des microcontrôleurs* 19](#_Toc193387735)

[***1. Sélection des capteurs :*** 19](#_Toc193387736)

[***2. Connexion au microcontrôleur :*** 20](#_Toc193387737)

[*5.2. Programmation pour la collecte et la transmission des données* 20](#_Toc193387738)

[*5.2 Récupération des données des capteurs* ***:*** 20](#_Toc193387739)

[*6. Gestion des alertes et des anomalies* 20](#_Toc193387740)

[*6.1. Définition des critères d'alerte* 20](#_Toc193387741)

[***6.2. Détection des anomalies*** 20](#_Toc193387742)

[***1. Surveillance des seuils critiques :*** 20](#_Toc193387743)

[***2. Surveillance des comportements anormaux :*** 20](#_Toc193387744)

[***3. Vérification croisée :*** 21](#_Toc193387745)

[*6.3. Mise en place du système d'alerte (notifications push, emails, etc.)* 21](#_Toc193387746)

[***Conclusion*** 21](#_Toc193387747)

[CHAPITRE IV. RÉALISATION DU SYSTEME DOMOTIQUE 22](#_Toc193387748)

[*4.1 Introduction* 22](#_Toc193387749)

[*4.2 Choix des technologies et outils* 22](#_Toc193387750)

[*4.2.1 Matériels utilisés* 22](#_Toc193387751)

[*4.2.2 Logiciels et outils de développement* 22](#_Toc193387752)

[*4.3 Conception et architecture du système* 23](#_Toc193387753)

[*4.3.1 Architecture générale du système* 23](#_Toc193387754)

[*4.3.2 Fonctionnalités du système* 23](#_Toc193387755)

[*4.4 Développement et programmation* 23](#_Toc193387756)

[*4.4.1 Programmation du microcontrôleur* 23](#_Toc193387757)

[*4.4.2 Gestion de l’affichage sur l’écran LCD* 24](#_Toc193387758)

[***4.6 Conclusion*** 25](#_Toc193387759)

[CHAPITRE V. TEST ET VALIDATION DU SYSTEME 26](#_Toc193387760)

[*CONCLUSION GENERALE* 32](#_Toc193387866)

[***REFERENCES*** 33](#_Toc193387867)

[***WEBOGRAPHIE*** 33](#_Toc193387868)

[***Ouvrages*** 34](#_Toc193387869)

[***Articles scientifiques*** 35](#_Toc193387870)

[***Mémoires et Thèses*** 35](#_Toc193387871)