الجمهـــوريـــــة الجزائـــريـــــة الديمقـــراطيـــــة الشـعبيـــة
République Algérienne Démocratique et Populaire
وزارة التـــعلـــيـــــم العـــــالـــــــي والبـــحـــــث العـلمـــــي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
جامعة وهران 2 محمد بن أ حمد
Université d'Oran 2 Mohamed Ben Ahmed
معهد الصيانة والأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité :

Maintenance en Automatisation et Instrumentation

Thème

Système Domotique Intelligent Prédictif Et Contextuelle Pour La Sécurité Autonome Du Domicile

Présenté et soutenu publiquement par :

Hocine Sara

Laoudjane Yasmine

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Chennoufi Mohamed	Professeur	IMSI	**Président**
Hadj djelloul Kheira	MAA	IMSI	**Encadreur**
S.Mouffak	MCB	IMSI	**Examinateur**

Année 2025/2026


بسم الله الرحمن الرحيم

Remerciments

Ce travail a été réalisé a l'institut de maintenance et sécurité industrielle de l'Université de d'oran 2 Ahmed Ben Ahmed.

Nous remercions Dieu Tout-Puissant de nous avoir accordé la patience, la santé et la volonté de mener à bien ce projet.

Nous exprimons notre gratitude à nos parents qui ont tant sacrifié pour nous et nous ont inculqué l'importance de prendre soin de nous et de nous faire confiance. Leur amour et leur soutien inconditionnels ont été pour nous une source d'inspiration.

Nous sommes également très reconnaissants envers notre professeur, M Hadj djelloul Kheira., qui a généreusement donné son temps et son expertise pour superviser nos travaux. Ses conseils et ses encouragements nous ont été précieux.

Nous tenons à remercier Mr Chennoufi Mohamed, qui nous a fait l'honneur de présider le jury de ce mémoire.

Enfin, nous remercions les membres du jury d'avoir accepté d'examiner notre travail et pour leur temps et leur expertise pour évaluer notre projet.

Sara et Yassmine.

Dédicaces

﴿ وَأَنْ لَيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَى ﴾

This graduation is not just an ending, but a dream that learned how to become real. It carries the weight of silent battles, patient prayers, tired nights, and steps taken with hope, even when the path was unclear. I grew through every lesson, every fall, every fear, and every small victory. Today, I do not only celebrate a degree; I celebrate the woman I became while reaching it.

I dedicate this moment to my mother, my paradise on earth, my first love, and the light that never left my way. Her prayers were my protection, her tenderness my strength, and her smile the sweetest reward of every success.

To my father, whose place in my heart is deep, quiet, and irreplaceable. Some love is too great to be spoken, yet too present to ever be forgotten.

To my brother, my support and my silent strength, who made me feel that I was never alone.

To my dear aunt, who was second mother to me and a heart that held me with love.

To my cousins Amel and Aya, who filled my life with affection, and to my Baby girl Hayat who got a piece of my heart

To my Jasmin, my companion of the journey, my partner in seats, dreams, laughter, and memories. We began this chapter together, and today we close it together, while greater paths wait ahead.

To my friends Maghfira and Sourour, who turned friendship into family, and ordinary days into memories that will stay with me forever.

To the generous soul who believed in me when I grew tired of believing in myself, whose quiet presence lit my path without needing a name.

To my inspiring teacher, who became a beautiful example from the very first meeting.

To everyone who taught me, guided me, encouraged me, prayed for me, or wished me well in silence.

And finally, to myself.

To the girl who was tired and continued, who fell and rose, who carried fear in her heart but never let it stop her feet. To the girl who chose patience, effort, and faith.

Today I stand beneath the mercy of Allah, watching my dream bloom, and I am so proud

الحمد لله على بداية الحلم ..و الحمد لله على تمامه

Sara.

Dédicaces

À cet instant qui marque l'aboutissement de plusieurs années d'efforts, de sacrifices, de doutes, mais aussi d'espoir et de persévérance, je prends quelques lignes pour exprimer ma profonde gratitude à ceux qui ont rendu ce parcours plus beau et plus léger.

Je dédie ce mémoire à la fille que j'étais hier. Celle qui pleurait en silence, celle qui doutait et celle qui se demandait si elle allait tenir. Tu as tenu! Tu t'es relevée à chaque fois. Et aujourd'hui, ce mémoire, c'est ta réponse à chaque moment où tu as failli lâcher.

À mon père. Mon premier exemple, mon ancre, mon refuge. Tu m'as transmis sans le dire le goût de l'effort, la dignité du travail bien fait et la certitude que les rêves méritent qu'on se batte pour eux. Chaque fois que j'ai voulu abandonner, c'est ton visage qui m'est revenu. Merci d'avoir toujours cru en moi, même quand moi je n'y croyais plus.

À ma mère. Comment mettre en mots ce que tu représentes ? Tu es la prière murmurée avant l'examen, la voix qui rassure quand tout s'effondre, les sacrifices silencieux que tu n'as jamais comptés. Tu t'es dépensée pour mon bonheur sans jamais rien demander en retour, et chaque page de ce mémoire porte un peu de ta douceur, de ta force et de ton amour infini. Je ne serai jamais assez pour te remercier. Mais sache que tout ce que je suis, c'est toi.

À mes chères sœurs, Khadidja et Farah, mes complices, mes confidentes, celles qui ont partagé mes silences et compris mes angoisses sans que j'aie besoin de parler. Et à ma petite princesse Ritadj, dont le sourire a suffi, certains jours, à tout réparer.

À toute ma famille, ma copine Selma et mes amis, chaque mot d'encouragement, chaque dua, chaque regard fier a été du carburant pour continuer. Vous avez été ma chaleur dans les moments froids.

À mon repère infini. Mon binôme. Ma meilleure amie Sara. On s'est assises côte à côte depuis le premier jour, on a traversé ensemble les examens, les projets, les doutes, les fous rires et les moments où on se demandait si on allait y arriver. Et aujourd'hui, on y est. On devient ingénieures ensemble. Ce n'est pas juste un diplôme qu'on obtient, c'est une promesse qu'on s'est tenue. Ce mémoire est à nous deux, et cette fierté, je veux la partager avec toi pour toujours.

À notre chère encadrante. Elle aurait pu faire le minimum, mais elle a choisi de donner bien plus. Sa porte était toujours ouverte, sa patience n'a jamais faibli, et sa bienveillance a été un cadeau inattendu et précieux. Dans les moments de doute, ses conseils ont remis les choses en place. Dans les moments de blocage, sa guidance a rouvert le chemin. Ce mémoire lui doit beaucoup, et cette gratitude ne s'arrêtera pas à la soutenance.

À tous ceux qui ont croisé ce chemin et y ont laissé quelque chose, un conseil, un mot juste au bon moment, une présence réconfortante. Vous faites partie de cette réussite, même si vous ne le savez pas.

Ce mémoire n'est pas une fin. C'est la preuve que j'en étais capable et le début de tout ce qui m'attend.

À vous tous, ma plus sincère et ma plus profonde gratitude.

Yasmina.

Résumé

Ce mémoire porte sur la conception et la réalisation d'un système domotique intelligent destiné à renforcer la sécurité résidentielle. L'objectif principal est de développer une solution capable d'adapter son comportement en fonction du type de danger rencontré, contrairement aux systèmes classiques qui reposent sur des seuils fixes et n'interviennent généralement qu'une fois la situation devenue critique.

Pour atteindre cet objectif, le système combine trois approches complémentaires. La première repose sur un réseau de neurones LSTM capable d'analyser des séries temporelles afin de détecter précocement les signes annonciateurs d'une fuite de gaz ou d'une surcharge électrique. La deuxième s'appuie sur des informations contextuelles, telles que l'heure et la présence ou non des occupants, afin d'améliorer la pertinence de la détection d'intrusion. La troisième approche concerne la détection directe des incendies, permettant une réaction immédiate et ciblée dès l'apparition d'un départ de feu.

Le prototype développé utilise un microcontrôleur ESP32 qui communique via le protocole MQTT avec un serveur Flask chargé de l'exécution des modèles d'intelligence artificielle. Une interface web assure également la supervision et le suivi en temps réel de l'ensemble du système.

Les résultats expérimentaux obtenus confirment le bon fonctionnement de l'architecture proposée et montrent qu'il est possible d'intégrer efficacement des mécanismes de prédiction, d'analyse contextuelle et de réaction instantanée au sein d'une même solution de sécurité résidentielle**.**

Mots clé : domotique intelligente, sécurité résidentielle, Internet des objets (IoT), ESP32, MQTT, Flask, LSTM, prédiction, détection contextuelle, systèmes embarqués

Summary

This thesis focuses on the design and implementation of an intelligent home automation system aimed at enhancing residential security. The main objective is to develop a solution capable of adapting its behavior according to the type of threat encountered, unlike conventional systems that rely on fixed thresholds and generally react only when a situation becomes critical.

To achieve this objective, the proposed system combines three complementary approaches. The first is based on a Long Short-Term Memory (LSTM) neural network capable of analyzing time-series data to detect early warning signs of gas leaks and electrical overloads. The second relies on contextual information, such as the time of day and the occupants' presence status, to improve the relevance of intrusion detection. The third approach focuses on direct fire detection, enabling an immediate and targeted response as soon as a fire outbreak is detected.

The developed prototype uses an ESP32 microcontroller that communicates via the MQTT protocol with a Flask server responsible for running the artificial intelligence models. A web-based interface is also provided to ensure real-time monitoring and supervision of the entire system.

The experimental results confirm the proper functioning of the proposed architecture and demonstrate the feasibility of effectively integrating prediction, contextual analysis, and instant response mechanisms within a single residential security solution.

Keywords: Intelligent Home Automation, Residential Security, Internet of Things (IoT), ESP32, MQTT, Flask, LSTM, Prediction, Context-Aware Detection, Embedded Systems.

الملخص

تتناول هذه المذكرة تصميم وإنجاز نظام ذكي لأنظمة المنازل الذكية يهدف إلى تعزيز الأمن السكني. ويتمثل الهدف الرئيسي في تطوير حل قادر على تكييف سلوكه وفقًا لنوع الخطر المكتشف، على عكس الأنظمة التقليدية التي تعتمد على عتبات ثابتة ولا تتدخل عادةً إلا بعد وصول الوضع إلى مرحلة حرجة.

ولتحقيق هذا الهدف، يجمع النظام المقترح بين ثلاث مقاربات متكاملة. تعتمد المقاربة الأولى على شبكة عصبية من نوع LSTM قادرة على تحليل السلاسل الزمنية للكشف المبكر عن المؤشرات الدالة على تسرب الغاز أو الأحمال الكهربائية الزائدة. أما المقاربة الثانية فتعتمد على المعلومات السياقية، مثل الوقت وحالة وجود السكان من عدمه، بهدف تحسين دقة وملاءمة كشف التسلل. في حين تركز المقاربة الثالثة على الكشف المباشر عن الحرائق، مما يتيح استجابة فورية وموجهة بمجرد ظهور بوادر اندلاع حريق.

يعتمد النموذج الأولي المطور على متحكم دقيق من نوع ESP32 يتواصل عبر بروتوكول MQTT مع خادم Flaskالمسؤول عن تشغيل نماذج الذكاء الاصطناعي. كما توفر واجهة ويب إمكانية الإشراف والمتابعة الآنية لمختلف مكونات النظام.

وتؤكد النتائج التجريبية المتحصل عليها سلامة عمل البنية المقترحة، كما تُبرز إمكانية الدمج الفعال لآليات التنبؤ والتحليل السياقي والاستجابة الفورية ضمن حل موحد ومتكامل للأمن السكني.

الكلمات المفتاحية: منزل ذكي، أمن سكني، LSTM، تنبّؤ، إنترنت الأشياء، Flask، MQTT، ESP32، كشف سياقي، نظام مدمج.

SOMMAIRE

Introduction générale ..... *i*

Chapitre I : Fondements théoriques et cadres conceptuels des systèmes domotiques intelligents

I.1 Introduction **02**
I .2 Domotique intelligente et Internet des objets **02**
I.2.1 Évolution et conceptualisation de la domotique **02**
I.2.2 Internet des objets : infrastructure de la maison intelligente **03**
I.2.3 Architecture fonctionnelle des systèmes IoT **03**
I.2.4 Protocoles de communication dans l'IoT **06**
I.2.4.1 MQTT **07**
I.2.4.2 CoAP **08**
I.2.4.3 Bluetooth **08**
I.2.4.4 ZigBee **09**
I.2.5 Analyse comparative des protocoles de communication IoT **10**
I.3 Sécurité domotique et analyse des risques **11**
I.3.1 Enjeux contemporains de la sécurité résidentielle **11**
I.3.2 Typologie des risques domestiques **12**
I.4 Systèmes contextuels (Context-Aware Systems) **13**
I.4.1 Définition et portée du concept de contexte **13**
I.4.2 Approches de modélisation du contexte **14**
I.5 Intelligence artificielle et prédiction dans les smart homes **14**
I.5.1 Détection vs prédiction **15**
I.5.2 Modèles d'apprentissage automatique **16**
I.6 Synthèse critique et positionnement scientifique **18**
I.7 Conclusion **19**

Chapitre II : Analyse des besoins et conception de l'architecture d'une maison intelligente

II.1 Introduction **21**

SOMMAIRE

II.2 Analyse des scénarios de risques 21

II.2.1 Intrusion 22

II.2.2 Incendie 22

II.2.3 Fuite de gaz 23

II.2.4 Surcharge électrique 25

II.3 Cahier des charges 26

II.3.1 Besoins fonctionnels 26

II.3.2 Besoins non fonctionnels 28

II.4 Modélisation de la maison intelligente 28

II.4.1 Découpage en zones fonctionnelles 28

II.4.2 Stratégie de placement des capteurs et contraintes physiques 29

II.4.3 Synthèse globale du système de capteurs 30

II.5 Conception de l'architecture globale 31

II.5.1 Conception logicielle 32

II.5.1.1 Module Acquisition (Edge) 32

II.5.1.2 Module Traitement 32

II.5.1.3 Module IA prédictif 32

II.5.1.5 Module Moteur Contextuel 32

II.5.1.6 Module Décision 32

II.5.1.7 Module Interface 32

II.5.2 Choix technologiques 33

II.5.2.1 ESP32 : acquisition et action physique (couche Edge) 33

II.5.2.2 MQTT (Mosquitto) : communication temps réel 34

II.5.2.3 Python : intelligence et traitement des données 34

II.5.2.4 MySQL : persistance des données 35

II.5.2.5 Flask : couche serveur web et API 35

II.5.2.6 HTML / CSS / JavaScript : interface utilisateur 35

II.6 Modélisation UML 36

II.6.1 Diagramme de cas d'utilisation 36

II.6.2 Diagrammes de séquence 37

II.6.3 Diagramme de déploiement 39

II.7 Conclusion 40

SOMMAIRE

Chapitre III : Modélisation contextuelle et intelligence artificielle pour la prédiction des risques

III.1 Introduction 42

III.2 Vue d'ensemble des trois approches 42

III.3 Approche prédictive (Cas du gaz et du courant) 44

III.3.1 Acquisition et structuration des données 44

III.3.1.1 Dispositif expérimental et protocole — cas du gaz 44

III.3.1.2 Structure des données 44

III.3.1.3 Labellisation et définition des seuils 45

III.3.1.4 Conversion et calibration ADC 45

III.3.1.5 Distribution des classes 46

III.3.1.6 Cas du courant 46

III.3.2 Prétraitement des données 46

III.3.3 Modèle prédictif LSTM 47

III.3.3.1 Justification de l'approche 47

III.3.3.2 Justification du LSTM 48

III.3.3.3 Principe de prédiction anticipée 49

III.3.3.4 Logique décisionnelle hybride 50

III.3.3.5 Architecture du modèle 50

III.3.3.6 Paramètres d'entraînement 52

III.3.3.7 Analyse des choix 52

III.4 Approche contextuelle — Cas de l'intrusion 53

III.4.1 Pourquoi une logique contextuelle et non un modèle d'apprentissage ? 53

III.4.2 Capteurs et architecture matérielle 54

III.4.3 Modes de fonctionnement 54

III.4.4 Logique de décision complète 56

III.5 Cas particulier (Détection et extinction de flamme) 57

III.5.1 Pourquoi une action directe ? 57

III.5.2 Capteurs et matériel 57

III.5.3 Disposition physique des voies 58

III.5.4 Logique de décision 59

III.5.5 Avantages de cette approche 60

SOMMAIRE

III.6 Conclusion 61

Chapitre IV : Implémentation Du Prototype, Intégration Et Validation Expérimentale

IV.1 Introduction 63
IV.2 Architecture matérielle du prototype 63
IV.2.1 Inventaire du matériel utilisé 63
IV.2.2 Conception et fabrication de la maquette 66
IV.2.2.1 Choix du matériau 66
IV.2.2.2 Géométrie et dimensionnement 67
IV.2.2.3 Plan de découpe 67
IV.2.2.4 Fabrication et assemblage de la maquette 67
IV.2.3 Brochage de l'ESP32 et plan de câblage global 69
IV.2.3.1 Règles suivies pour l'affectation des broches 69
IV.2.4 Présentation détaillée des composants et de leur câblage 71
IV.2.4.1 Microcontrôleur ESP32-WROOM-32^{E} 71
IV.2.4.2 Breadboard et fils de connexion (jumpers) 72
IV.2.4.3 Capteur de gaz MQ-5 72
IV.2.4.4 Capteur ultrason HC-SR04 73
IV.2.4.5 Capteur de flamme infrarouge 5 voies 74
IV.2.4.6 Servomoteur SG90 75
IV.2.4.7 Module relais (ventilateur et pompe) 76
IV.2.4.8 Module horloge RTC DS1302 77
IV.2.4.9 Buzzer actif 79
IV.2.4.10 LEDs indicatrices 79
IV.3 Architecture logicielle du prototype 81
IV.3.1 Firmware embarqué sur l'ESP32 81
IV.3.2 MQTT 82
IV.3.3 Lecture du MQ-5 et sécurité locale 82
IV.3.4 Détection d'intrusion avec HC-SR04 83
IV.3.5 Détection de flamme et orientation du servo 84
IV.3.6 Réception des commandes du dashboard 84
IV.3.7 setup () et loop () 85
IV.4 Base de données MySQL et préparation du dataset 85
IV.4.1 Création de la base MySQL 85

IV.4.2 Collecte des mesures par MQTT 85
IV.4.3 Export MySQL vers CSV 86
IV.4.4 Structure du dataset obtenu 86
IV.5 Entraînement du modèle LSTM 86
IV.5.1 Chargement et labellisation des données 86
IV.5.2 Normalisation et fenêtres temporelles 87
IV.5.3 Séparation entraînement / test 87
IV.5.4 Architecture et entraînement du modèle 87
IV.6 Serveur Flask, MQTT et intégration de l'IA 89
IV.6.1 Configuration et chargement du modèle 89
IV.6.2 Buffer gaz et prédiction en temps réel et traitement des messages MQTT 90
IV.6.3 Routes Flask 90
IV.6.4 Architecture thread du serveur 90
IV.7 Dashboard web de supervision 91
IV.7.1 Principe de fonctionnement 91
IV.7.2 Affichage des résultats 91
IV.7.3 Envoi des commandes 93
IV.8 Tests et validation expérimentale 93
IV.8.1 Protocole et scénarios testés 93
IV.8.2 Résultats observés 93
IV.8.3 Synthèse et validation 94
IV.9 Conclusion 94

Conclusion générale 97

Bibliographie 101

Table des Figures

Figure I.1 : Évaluation de la domotique et rôle de l'Internet des objets dans l'infrastructure d'une maison intelligente 03
Figure I.2 : Architecture fonctionnelle des systèmes IoT 05
Figure I.3 : Architecture en couches d'un système Internet des Objets appliqué à la maison intelligente. Vers une maison intelligente connectée et autonome. 06
Figure I.4 : Analyse comparative des protocoles de communication IoT 11
Figure II.1 : Chaîne de traitement du scénario d'intrusion 22
Figure II.2 : Logique de détection et d'extinction incendie 23
Figure II.3 : Chaîne de traitement fuite de gaz 24
Figure II.4 : Détection et réaction surcharge électrique 25
Figure II.5 :Architecture globale du système 31
Figure II.6 : Le brochage du microcontrôleur 33
Figure II.7 : Échanges publish/subscribe de l'ESP32 via le broker MQTT Eclipse Mosquitto 34
Figure II.8 : Modélisation des interactions utilisateur–système (diagramme de cas d'utilisation) 37
Figure II.9 : Diagramme de séquence : prédiction de fuite de gaz 38
Figure II.10 : Diagramme de séquence : détection d'intrusion 39
Figure II.11 : diagramme de déploiement du système de sécurité intelligent. 40
Figure III.1 : Architecture interne d'un bloc LSTM 48
Figure III.2 : Illustration du mécanisme de prédiction anticipée en zone d'anomalie 49
Figure III.3 : Architecture du modèle LSTM, description couche par couche. 51
Figure III.4 : Arbre de décision pour la détection d'intrusion 56
Figure III.5 : Disposition des 5 voies du capteur de flamme 59
Figure III.6 : Arbre de décision pour la détection et l'extinction de flamme 60
Figure IV.1 : Vue d'ensemble : l'ESP32 est le hub central, relié aux capteurs à gauche, aux actionneurs à droite, au serveur via WiFi/MQTT en haut, et à l'alimentation en bas. 65
Figure IV.2 : Plaque de plexiglas (PMMA) utilisée pour la fabrication de la maquette.66

Table des Figures

Figure IV.3 : Plan de découpe des panneaux en PMMA et organisation structurelle de la maquette 68
Figure IV.4 : Étapes successives de fabrication de la maquette (découpe, perçage et assemblage) 69
Figures IV.5 : Brochage détaillé de l'ESP32 (38 pins, ESP32-WROOM-32E) avec l'affectation exacte de toutes les broches utilisées dans le projet 71
Figures IV.6 : : microcontrôleur ESP32-WROOM-32E 72
Figure IV.7 : Schéma des connexions internes d'une breadboard et des fils de liaison (jumpers) 72
Figure IV.8 : Câblage du capteur MQ-5 vers l'ESP32. Seule la sortie analogique AO est utilisée. 73
Figure IV.9 : Câblage du HC-SR04. TRIG est une sortie ESP32, ECHO une entrée. Les deux sont en logique 5V mais l'ESP32 (3.3V) tolère sans problème en lecture 74
Figure IV.10 : Câblage du module flamme 5 voies. Une alimentation de 5 V commune et 5 lignes numériques distinctes vers les GPIO d'entrée. 75
Figure IV.11 : Câblage du servomoteur SG90 76
Figure IV.12 : Module relais 76
Figure IV.13 : Câblage du module relais. Les bornes IN sont commandées par l'ESP32. Les bornes COM/NO/NC commutent la charge externe 77
Figure IV.14 : Le DS1302. 78
Figure IV.15 : Câblage du DS1302. Trois lignes de communication + alimentation 3,3 V (le module accepte aussi 5 V). 78
Figure IV.16 : Câblage du buzzer actif. Commande directe par GPIO, masse commune avec l'ESP32 79
Figure IV.17 : Câblage des LEDs avec l'ESP32 80
Figure IV.18 : Câblage de tout le matériel 80
Figure IV.19 : Vue d'ensemble de la maquette en plexiglas (PMMA) après assemblage et intégration des composants 81
Figure IV.20 : Lancement de Mosquito 82
Figure IV.21 : Détection du gaz sur le serial moniteur d'Arduino IDE 83
Figure IV.22 : Détection d'intrusion sur le serial moniteur d'Arduino IDE 84

Table des Figures

Figure IV.23 : Détection de flamme sur serial monitor d'Arduino IDE 84
Figure IV.24 : Aperçu du fichier DB_MQ5.csv dans Excel (24 667 mesures) 86
Figure IV.25 : Déroulement de l'entraînement du modèle LSTM (30 epochs + 20 epochs de fine-tuning) 87
Figure IV.26 : Résultats d'évaluation du modèle LSTM sur l'ensemble de test (précision globale : 93 %) 88
Figure IV.27 : Courbes d'apprentissage du modèle LSTM 88
Figure IV.28 : Fichiers générés après l'entraînement : modèle, scaler 89
Figure IV.29 : Démarrage du serveur Flask : chargement du modèle LSTM et connexion MQTT 89
Figure IV.30 : Réception des mesures et prédiction LSTM en temps réel (transition Normal → Anomalie) 90
Figure IV.31 : Réponse JSON de l'Endpoint /api/état dans le navigateur 90
Figure IV.32 : Interface du Dashboard en cas d'anomalie de gaz détectée 91
Figure IV.33 : Interface du Dashboard en cas de flamme détectée 92
Figure IV.34 : Interface du Dashboard en cas d'intrusion détectée 92
Figure IV.35 : Interface du Dashboard partie commandes 93

Liste des Tableaux

Tableau I.1 : Synthèse des caractéristiques et de l'architecture du protocole MQTT 07
Tableau I.2 : Synthèse des caractéristiques et de l'architecture du protocole CoAP. 08
Tableau I.3 : Synthèse des caractéristiques et de l'architecture du protocole Bluetooth. 09
Tableau I.4 : Synthèse des caractéristiques et de l'architecture du protocole ZigBee. 09
Tableau I.5 : Synthèse des principales différences entre détection et prédiction.....16
Tableau II.1 : Modélisation fonctionnelle des zones et profils de risque 29
Tableau II.2 : Déploiement des capteurs MQ-5 29
Tableau II.3 : Déploiement des capteurs PIR.....30
Tableau II.4 : Déploiement des capteurs de flamme.....30
Tableau II.5 : Surveillance électrique par ACS712 30
Tableau II.6 : Synthèse multi-zones.....30
Tableau III.1 : Synthèse des trois approches méthodologiques 43
Tableau III.2 : Structure du dataset gaz (n = 24 667).....44
Tableau III.3 : Sources de définition des seuils 45
Tableau III.4 : Seuils finaux de labellisation 45
Tableau III.5 : Distribution des classes 46
Tableau III.6 : Structure des données après fenêtrage.....47
Tableau III.7 : Répartition train / test.....47
Tableau III.8 : Logique combinant état actuel et prédiction 50
Tableau III.9 : Paramètres d'entraînement du modèle LSTM.....52
Tableau III.10 : Composants matériels de la détection d'intrusion 54
Tableau III.11 : Modes de surveillance contextuelle.....55
Tableau III.11 : Architecture matérielle du système anti-incendie.....57
Tableau IV.1 : liste complète du matériel utilisé dans le prototype. 64
Tableau IV.2 : Tableau récapitulatif des affectations.....70
Tableau IV.3 : Cartographie voie de détection ↔ angle du servomoteur.....75
Tableau IV.4 Organisation du Firmware ESP3 81
Tableau IV.5 : Synthèse des tests fonctionnels du prototype 93

INTRODUCTION GÉNÉRALE

Introduction Générale

1. Contexte et motivation

Depuis les premières expérimentations d'automatisation résidentielle dans les années 1970, époque dominée par des dispositifs rudimentaires tels que les minuteries mécaniques ou les thermostats programmables , la domotique n'a cessé de se redéfinir, de se complexifier, puis de se recomposer autour de paradigmes technologiques successifs. Les années 1990 introduisent la couche réseau, locale d'abord, structurante ensuite. Puis survient, dans les années 2010, l'irruption massive de l'Internet des Objets, véritable rupture épistémologique dans la manière de concevoir l'habitat : non plus un espace statique, mais un système dynamique, instrumenté, réactif.

Dans cette trajectoire, une convergence s'impose progressivement : capteurs miniaturisés, protocoles de communication légers, microcontrôleurs à très faible coût rendant désormais possible l'émergence de systèmes domotiques complets, y compris dans des environnements académiques contraints. Pourtant, cette maturité apparente masque une asymétrie persistante : la sécurité résidentielle demeure en retrait, souvent réduite à des mécanismes simplifiés, linéaires, voire purement seuil-dépendants.

2. Enjeux de la sécurité résidentielle

Les environnements domestiques sont exposés à plusieurs risques critiques tels que les incendies, les fuites de gaz, les surcharges électriques et les intrusions, qui diffèrent fortement par leur nature et leur dynamique. Cependant, les systèmes existants reposent principalement sur une approche réactive basée sur des seuils fixes, déclenchant une alarme uniquement après dépassement d'une valeur prédéfinie.

Cette méthode, bien que simple et robuste, ne permet ni l'anticipation ni l'adaptation aux spécificités de chaque phénomène. En effet, certaines situations évoluent progressivement et peuvent être détectées en amont par analyse des tendances, tandis que d'autres dépendent fortement du contexte ou exigent une réaction immédiate sans délai d'analyse.

Ainsi, une approche uniforme de la sécurité s'avère insuffisante face à la diversité des risques, qui nécessitent des stratégies de traitement différenciées selon leur nature et leur temporalité.

3. Problématique

À partir de ce constat, la question centrale peut être formulée ainsi : comment concevoir une architecture domotique capable d'adapter sa logique de décision à la nature du risque observé, en intégrant simultanément prédiction, contextualisation et détection directe dans un

même cadre unifié ?

Autrement dit, peut-on construire un système capable d'anticiper une fuite avant qu'elle ne devienne critique, d'interpréter une intrusion à travers son contexte d'occurrence, et de déclencher une réaction immédiate face à un incendie sans passer par une homogénéisation des processus décisionnels ?

4. Objectifs du projet

Ce travail propose la conception et la mise en œuvre d'un système domotique intelligent orienté sécurité résidentielle, articulé autour de trois paradigmes de traitement complémentaires, appliqués à quatre scénarios de risque.

Les phénomènes progressifs : fuites de gaz et surcharges électriques relèvent d'une logique prédictive. Celle-ci s'appuie sur l'apprentissage profond et l'exploitation de séries temporelles issues des capteurs, afin d'identifier des dérives statistiques avant l'atteinte du seuil critique.

Les intrusions, quant à elles, sont traitées selon une logique contextuelle. Les données des capteurs y sont interprétées conjointement avec des variables situationnelles : temps, présence, occupation, permettant de distinguer le comportement normal et l'événement suspect.

Enfin, les incendies mobilisent une logique de détection directe. Ici, la contrainte temporelle devient absolue : la décision est locale, immédiate, exécutée directement au niveau du microcontrôleur, sans couche d'abstraction supplémentaire.

L'objectif global est de démontrer la viabilité d'une architecture unifiée capable de superposer ces trois logiques sans les fusionner, mais en les orchestrant selon la nature du phénomène observé.

5. Méthodologie adoptée

La démarche méthodologique s'organise en trois niveaux complémentaires, formant une chaîne cohérente allant de l'abstraction théorique à la validation expérimentale.

Le premier niveau est conceptuel : il s'appuie sur l'étude de l'état de l'art en domotique intelligente, architectures IoT, protocoles de communication et modèles d'apprentissage temporel.

Le deuxième niveau est structurel : il consiste à traduire les exigences fonctionnelles en une architecture matérielle et logicielle formalisée, appuyée par une modélisation UML et un cahier des charges rigoureux.

Le troisième niveau est expérimental : il repose sur la réalisation d'un prototype physique instrumenté, l'entraînement de modèles d'intelligence artificielle sur des données hybrides (réelles et synthétiques), ainsi que la validation intégrale de la chaîne de traitement, depuis l'acquisition des signaux jusqu'à leur supervision via une interface web.

6. Organisation du mémoire

Pour une bonne présentation de notre travail, ce mémoire s'articule autour de quatre chapitres complémentaires :

Nous présentons dans le premier chapitre les fondements théoriques en abordant l'évolution de la domotique, les architectures IoT, les protocoles de communication, ainsi que les concepts d'intelligence artificielle et de systèmes contextuels sur lesquels repose cette étude

Dans le deuxième chapitre, nous traitons l'analyse des besoins à travers l'étude des quatre scénarios de risque, la définition du cahier des charges et la conception de l'architecture globale du système, appuyée par une modélisation UML.

Le troisième chapitre est consacré à la modélisation contextuelle et à l'intelligence artificielle. Il détaille le processus d'acquisition et de prétraitement des données, l'architecture du modèle prédictif, ainsi que la logique décisionnelle hybride.

Le dernier chapitre est destiné à présenter l'implémentation matérielle et logicielle du prototype, depuis la réalisation de la maquette et le câblage des capteurs jusqu'à l'intégration du firmware, du serveur et de l'interface de supervision. Il se conclut par la validation expérimentale du système.

CHAPITRE I :

Fondement théorique et cadres conceptuels des systèmes domotiques Intelligents

I.1 Introduction

La création d'un système domotique intelligent, qui soit autonome, contextuel et capable de prédictions, se trouve à la croisée de plusieurs domaines d'études : l'Internet des objets, l'intelligence artificielle, les systèmes distribués ainsi que la modélisation du contexte. Cette combinaison technologique requiert une organisation méticuleuse des fondements théoriques.

L'objectif de ce chapitre est de poser ce socle scientifique. Il ne s'agit pas simplement d'une présentation descriptive. C'est un développement progressif, presque similaire à une construction architecturale, des concepts nécessaires pour comprendre et justifier le système proposé.

Dans cette perspective, l'analyse est organisée autour de quatre axes majeurs :

- les **fondements de la domotique et de l'IoT**,
- les **enjeux de la sécurité domestique**,
- les **approches de modélisation contextuelle**,
- et enfin les **méthodes d'intelligence artificielle orientées vers la prédiction**.

L'ensemble converge vers une synthèse critique permettant de situer clairement notre contribution.

I .2 Domotique intelligente et Internet des objets

I.2.1 Évolution et conceptualisation de la domotique

La domotique, issue du terme *domus*, ne se limite plus à une simple automatisation de l'habitat. Elle constitue aujourd'hui un système socio-technique complexe, intégrant technologies numériques, comportements humains et logiques d'interaction [1], [2].

Historiquement, son évolution s'est faite par paliers technologiques successifs. D'abord rudimentaire, minuteries et automatismes isolés dans les années 1960, elle devient communicante avec l'apparition du protocole X10 dans les années 1980. Puis, elle s'accélère. Rapidement [3], [4].

Les années 2000 introduisent une rupture décisive : la connectivité Internet. Depuis 2010, l'intégration de l'intelligence artificielle et des assistants vocaux transforme la maison en un environnement adaptatif, capable non seulement de réagir, mais d'anticiper [4].

Ainsi, la domotique contemporaine ne se définit plus uniquement par ses équipements. Elle se définit par sa capacité à produire du sens à partir des données [5].

I.2.2 Internet des objets : infrastructure de la maison intelligente

L'Internet des Objets constitue le **substrat technologique fondamental** de la domotique moderne. Il s'agit d'un réseau distribué d'objets physiques interconnectés, capables de percevoir, transmettre et exploiter des données en temps réel [6], [7].

Introduit conceptuellement à la fin des années 1990, l'IoT a évolué vers une infrastructure globale, auto-configurable, reposant sur des protocoles standardisés. Aujourd'hui, il permet une gestion proactive et distribuée des environnements domestiques. L'habitat devient alors un système cyber-physique. Un système vivant, en quelque sorte [8], [9].

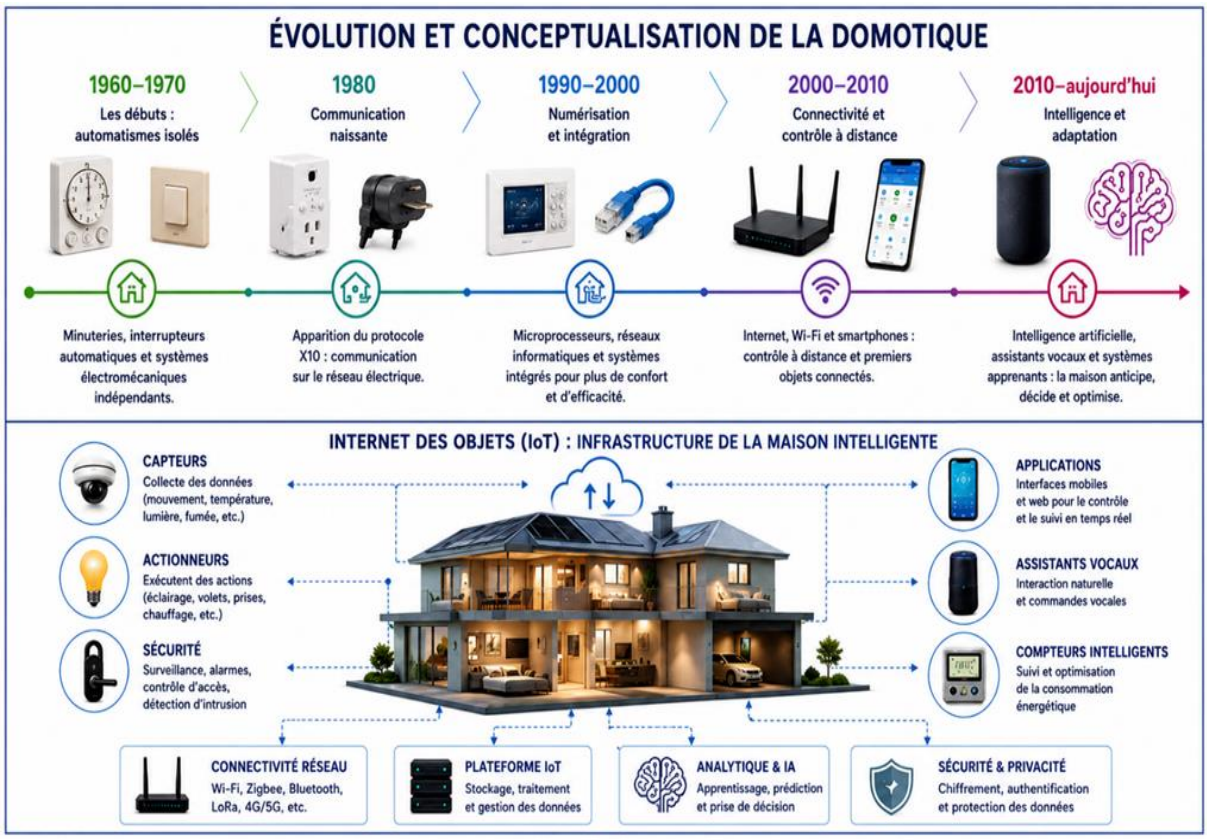


Figure I.1 : Évaluation de la domotique et rôle de l'Internet des objets dans l'infrastructure d'une maison intelligente

I.2.3 Architecture fonctionnelle des systèmes IoT

L'architecture de l'Internet des Objets constitue le socle organisationnel fondamental garantissant la cohérence, l'interopérabilité et la robustesse des systèmes connectés. Elle ne se limite pas à une simple juxtaposition de composants techniques. Elle formalise, de manière rigoureuse, les interactions entre des entités hétérogènes : dispositifs physiques, infrastructures réseau, plateformes logicielles et utilisateurs finaux [10].

Autrement dit, elle agit comme un schéma directeur, orchestrant l'ensemble du cycle opérationnel du système : de la captation des données à leur valorisation décisionnelle, en passant par leur transmission et leur traitement. Cette structuration est essentielle. Sans elle, aucun système IoT ne peut atteindre un niveau de performance ou de fiabilité acceptable.

Cependant, il convient de souligner qu'il n'existe pas de modèle architectural universellement standardisé. Plusieurs approches coexistent. Néanmoins, dans la littérature scientifique comme dans les implémentations industrielles, une architecture en quatre couches s'impose comme le modèle de référence, en raison de sa clarté conceptuelle et de sa modularité [10].

Ces couches ne sont pas indépendantes. Elles sont fortement couplées, interdépendantes, et doivent fonctionner en parfaite synergie pour produire de la valeur.

a) Couche de perception (Sensing Layer)

La couche de perception constitue la fondation physique de l'écosystème IoT. Elle regroupe l'ensemble des dispositifs capables d'interagir directement avec l'environnement réel : capteurs, actionneurs, modules RFID, et objets connectés.

Son rôle est double. D'une part, elle assure la capture continue des données issues du monde physique (température, mouvement, luminosité, gaz, etc.). D'autre part, elle convertit ces grandeurs physiques en signaux numériques exploitables par les systèmes informatiques.

On peut la considérer comme les *organes sensoriels* du système. Sans elle, aucune perception. Sans perception, aucune intelligence.[11] ,[10]

b) Couche réseau (Network Layer)

La couche réseau intervient comme un mécanisme de transport, assurant l'acheminement des données collectées vers les niveaux supérieurs du système.

Elle s'appuie sur une diversité de technologies de communication, chacune répondant à des contraintes spécifiques :

- Wi-Fi pour les environnements domestiques à haut débit.
- Bluetooth pour les communications de proximité.
- réseaux cellulaires (4G/5G) pour la connectivité étendue.
- LPWAN pour les applications à longue portée et faible consommation.

Son exigence principale est claire : garantir une transmission fiable, sécurisée et à faible latence, même dans des environnements contraints ou instables.

Un réseau défaillant compromet l'ensemble du système. C'est un point critique. [10], [11]

c) Couche de traitement – Middleware (Processing Layer)

Cette couche représente le cœur analytique et décisionnel de l'architecture IoT. C'est ici que les données acquièrent leur véritable valeur.

Son fonctionnement repose sur deux processus fondamentaux :

- **Accumulation des données** : les flux issus des capteurs sont collectés, filtrés, puis stockés dans des infrastructures adaptées (bases de données, data lakes). Cette étape permet de stabiliser des données initialement volatiles.
- **Abstraction et structuration** : les données hétérogènes (IoT et non-IoT) sont ensuite harmonisées, enrichies et consolidées afin de constituer un référentiel cohérent et exploitable.

Mais son rôle ne s'arrête pas là. Il est également bidirectionnel. Les flux montants (capteurs → applications) sont transformés en informations intelligibles, tandis que les flux descendants (applications → actionneurs) sont traduits en instructions opérationnelles.

Ainsi, cette couche agit comme un intermédiaire intelligent, masquant la complexité technique tout en assurant l'interopérabilité globale du système. [10], [11]

d) Couche application (Application Layer)

Située au sommet de l'architecture, la couche applicative constitue le point de contact direct avec l'utilisateur final.

Elle transforme les données traitées en services concrets : supervision, contrôle à distance, alertes, visualisation, automatisation. C'est à ce niveau que l'abstraction technique devient expérience utilisateur.

Les applications peuvent être développées :

- soit directement sur des plateformes IoT intégrées,
- soit via des interfaces programmatiques (API), permettant une interconnexion avec des systèmes externes.

En définitive, cette couche matérialise la finalité du système : convertir la donnée en décision, puis la décision en action. [10], [11]

L'architecture IoT ne doit pas être perçue comme une simple organisation technique. Elle constitue une chaîne de transformation continue :

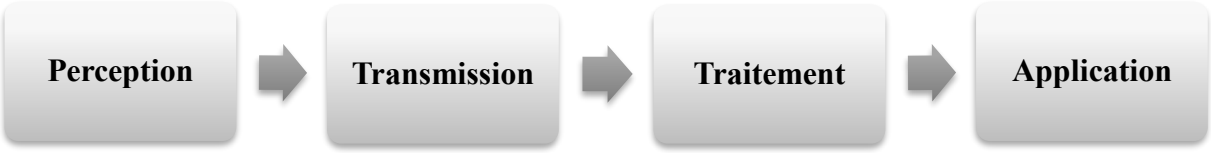


Figure I.2 : Architecture fonctionnelle des systèmes IoT

Chaque rupture dans cette chaîne compromet l'ensemble. Chaque couche est critique. Et c'est précisément leur coordination dynamique qui conditionne la performance globale du système.

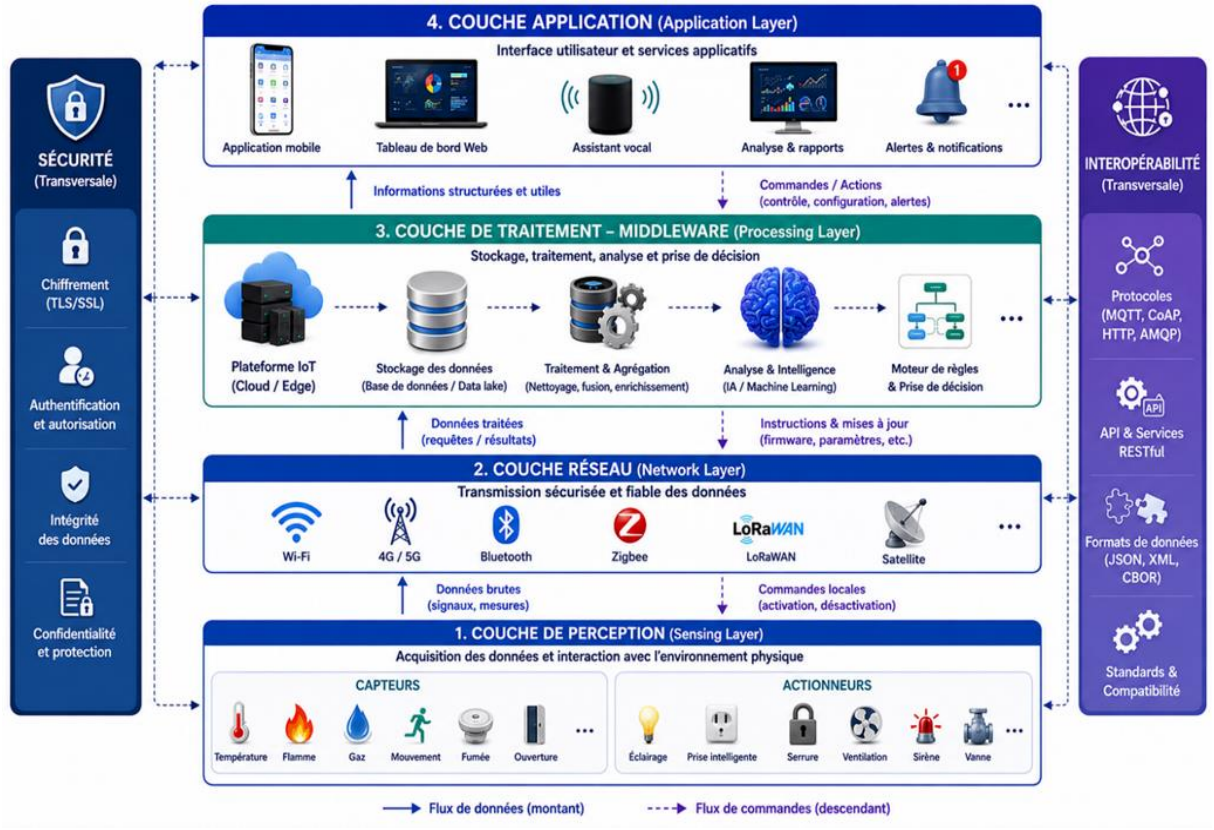


Figure I.3 : Architecture en couches d'un système Internet des Objets appliqué à la maison intelligente. Vers une maison intelligente connectée et autonome.

Cette figure illustre (Figure I.3) l'architecture en couches d'un système IoT appliqué à une maison intelligente. Elle met en évidence le flux bidirectionnel des données entre les capteurs, les infrastructures réseau, les plateformes de traitement et les interfaces utilisateur, ainsi que le rôle central du middleware dans la transformation des données en décisions exploitables.

I.2.4 Protocoles de communication dans l'IoT

Dans l'Internet des objets (IoT), les dispositifs connectés doivent échanger des données afin d'assurer leur fonctionnement et leur coordination. Pour permettre ces communications, plusieurs protocoles sont disponibles. Le choix du protocole constitue une étape essentielle dans la conception d'une solution IoT, car il influence directement ses performances et sa fiabilité [12], [13].

Chaque protocole possède des caractéristiques spécifiques, notamment en termes de portée de communication, de consommation énergétique, de latence et d'architecture. Ces propriétés déterminent son adéquation à un scénario d'utilisation donné, qu'il s'agisse d'environnements contraints, de systèmes temps réel ou de réseaux à grande échelle ,[12] . [14]

I.2.4.1 MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de communication conçu pour l'Internet des objets (IoT). Il se distingue par sa **légèreté**, sa **simplicité** et sa **faible consommation de bande passante et d'énergie**, ce qui le rend idéal pour les environnements contraints (réseaux instables, appareils sur batterie). [15]

Tableau I.1 : Synthèse des caractéristiques et de l'architecture du protocole MQTT.

Catégorie	Description
Origine	Créé en 1999 par Andy Stanford-Clark (IBM) et Arlen Nipper. Conçu pour la surveillance de pipelines via satellite, avec des contraintes de bande passante et d'énergie. [15]
Architecture	Modèle client–serveur composé de deux éléments :
Client MQTT	Peut agir comme publisher (éditeur), subscriber (abonné) ou les deux.[16]
Broker MQTT	Serveur central qui : reçoit les messages, filtre les données, gère les abonnements et distribue les messages aux clients concernés.[16]
Modèle de communication	Basé sur le modèle Publish/Subscribe (pub/sub), contrairement au modèle classique client-serveur. [17]
Publisher	Envoie (publie) des messages vers un topic. [17]
Subscriber	S'abonne à un ou plusieurs topics pour recevoir les messages. [17]
Broker (rôle dans pub/sub)	Assure la liaison entre publisher et subscriber en redistribuant les messages. [17]
Topics	Structure hiérarchique permettant d'organiser les messages (ex. : maison/cuisine/capteur). [16], [17]
Avantage des topics	Permettent une gestion flexible et fine des flux de données. [17]
Points clés	Communication asynchrone, découplage entre émetteur et récepteur, idéal pour les systèmes IoT distribués.

I.2.4.2 CoAP (Constrained Application Protocol)

CoAP (Constrained Application Protocol) est un protocole de transfert web léger conçu pour les environnements contraints de l'Internet des objets (IoT). Il est particulièrement adapté aux dispositifs à ressources limitées et aux réseaux à faible bande passante ou peu fiables. CoAP adopte une architecture simple et efficace basée sur le modèle REST, permettant l'accès aux ressources des objets connectés avec une faible surcharge de communication.[14], [18]

Tableau I.2 : Synthèse des caractéristiques et de l'architecture du protocole CoAP.

Catégorie	Description
Origine	Standardisé par l'IETF (groupe CoRE) en 2014 dans le RFC 7252.[18]
Architecture	Modèle client–serveur sans broker central. [18]
Modèle de communication	Requête/Réponse basé sur REST (GET, POST, PUT, DELETE). [18], [19]
Transport	UDP (protocole léger et rapide). [19]
Confirmable (CON)	Nécessite un accusé de réception (ACK) pour garantir la livraison. [19]
Non-confirmable (NON)	Aucun ACK requis, optimisé pour la rapidité et la faible consommation. [19]
Observe	Mécanisme permettant un comportement proche du pub/sub avec des notifications automatiques. [20]
Blockwise Transfer	Découpage des messages en blocs pour les données volumineuses.[20]
Message ID	Identifiant unique pour éviter les doublons et gérer la fiabilité. [19]
Points clés	Léger, faible consommation, adapté aux objets contraints, communication efficace sur les réseaux IoT. [18], [20]

I.2.4.3 Bluetooth

Le Bluetooth est une technologie de communication sans fil à courte portée permettant l'échange de données entre appareils proches. Il est largement utilisé dans les systèmes IoT, les périphériques mobiles et les équipements

domotiques grâce à sa simplicité, son faible coût et sa faible consommation énergétique.[21], [22].

Tableau I.3 : Synthèse des caractéristiques et de l'architecture du protocole Bluetooth.

Catégorie	Description
Origine	Créé en 1998 par le Bluetooth SIG (Ericsson, IBM, Intel, Nokia, Toshiba). [21]
Architecture	Radio (RF), baseband, link manager, couche applicative. [21]
Topologie réseau	Piconet (1 maître + jusqu'à 7 esclaves) et scatternet (interconnexion de piconets). [23]
Bande de fréquence	2,4 GHz (bande ISM). [23]
Technique de transmission	FHSS (Frequency Hopping Spread Spectrum) sur 79 canaux. [23]
Portée	De 10 à 100 mètres selon la puissance. [23]
Débit	Jusqu'à 721 kbps (Basic Rate). [23]
Points clés	Faible coût, faible consommation, robuste face aux interférences, adapté aux communications à courtes distances. [21], [23]

I.2.4.4 ZigBee

ZigBee est un protocole de communication sans fil conçu pour les réseaux à faible consommation d'énergie. Il est principalement utilisé dans les applications IoT nécessitant une longue autonomie, telles que les capteurs intelligents et les systèmes domotiques. ZigBee se distingue par sa capacité à former des réseaux maillés permettant une communication étendue et fiable. [22], [24], [25]

Tableau I.4 : Synthèse des caractéristiques et de l'architecture du protocole ZigBee.

Catégorie	Description
Origine	Basé sur la norme IEEE 802.15.4, conçu pour les réseaux IoT à faible consommation. [22], [24]
Architecture	Coordinateur (ZC), routeur (ZR), terminal (ZED). [22], [25]
Topologie réseau	Réseau maillé (mesh networking). [25]
Bande de *fréquences*	2,4 GHz, 915 MHz (USA), 868 MHz (Europe). [25]
Débit	Jusqu'à 250 kbps. [25]
Portée	Étendue grâce au routage multi-sauts entre nœuds. [25]

Consommation énergétique	Très faible, optimisé pour une longue durée de vie des batteries. [25]
Points clés	Réseau maillé, faible consommation, faible débit, adapté aux capteurs distribués et à la domotique. [24], [25]

I.2.5 Analyse comparative des protocoles de communication IoT

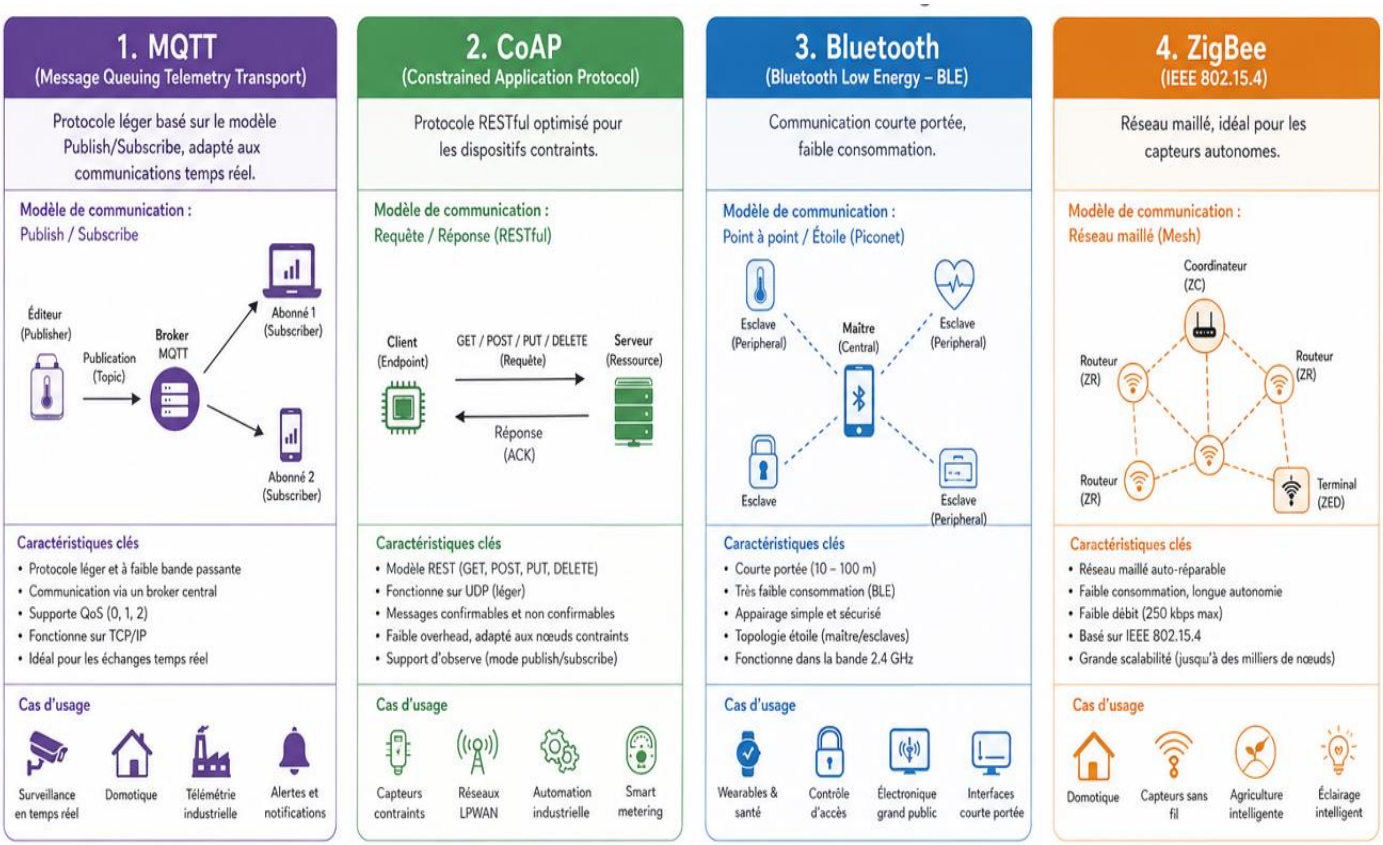
Les protocoles de communication présentés répondent à des exigences techniques différenciées au sein de l'écosystème Internet des Objets, en fonction des contraintes liées aux ressources matérielles, aux besoins applicatifs et aux caractéristiques des environnements de déploiement.[14], [26]

Dans ce contexte, MQTT s'impose comme le protocole de référence pour les architectures IoT orientées temps réel, en raison de sa légèreté, de son faible overhead et de son modèle de communication publish/subscribe, particulièrement adapté aux échanges asynchrones entre capteurs et serveurs. [17]

De son côté, CoAP constitue une alternative pertinente dans les environnements fortement contraints. Son approche RESTful, combinée à l'utilisation d'UDP, lui permet d'assurer une communication efficace pour les dispositifs à ressources limitées ne supportant pas une connexion TCP persistante. [18]

Par ailleurs, ZigBee est privilégié pour les applications nécessitant une autonomie énergétique prolongée et une topologie en réseau maillé (mesh), garantissant une meilleure robustesse et une extensibilité du réseau de capteurs. [24]

Enfin, Bluetooth se distingue par sa simplicité de déploiement et sa faible consommation énergétique, ce qui le rend particulièrement adapté aux communications de courte portée et aux interactions directes entre dispositifs.Ainsi, chaque protocole occupe une position spécifique et complémentaire, contribuant à la conception d'architectures IoT hybrides, optimisées selon les contraintes fonctionnelles et énergétiques du système. [23]



Protocole	MQTT	CoAP	Bluetooth (BLE)	ZigBee
Modèle de communication	Publish / Subscribe	Requête / Réponse (RESTful)	Point à point / Étoile	Réseau maillé (Mesh)
Couche transport	TCP/IP	UDP	Bluetooth LE	IEEE 802.15.4
Portée typique	Réseaux étendus (Internet)	Réseaux étendus (IP)	Courte portée (10 – 100 m)	Moyenne portée (10 – 100 m par saut)
Consommation	Faible	Très faible	Très faible	Très faible
Débit	Faible à moyen	Faible	1 Mbps (BLE 5)	250 kbps max
Cas d'usage principal	Supervision et contrôle temps réel	Dispositifs contraints, IoT léger	Connexions courtes, accessoires	Réseaux de capteurs, automatisation

Figure I.4: Analyse comparative des protocoles de communication IoT

I.3 Sécurité domotique et analyse des risques

I.3.1 Enjeux contemporains de la sécurité résidentielle

Le domicile, autrefois espace strictement protecteur, devient aujourd'hui un environnement à la fois sécurisé et exposé en raison de la multiplication des équipements électriques et des systèmes connectés. Cette évolution s'accompagne d'une diversification des risques domestiques (incendies, intrusions, fuites de gaz, pannes électriques), dont les

manifestations peuvent être soudaines ou progressives et souvent difficiles à détecter sans dispositifs adaptés. [27]

Les systèmes de sécurité traditionnels restent principalement réactifs, se limitant à la détection et à l'alerte après la survenue de l'incident, ce qui limite leur efficacité dans des environnements complexes et dynamiques. [27]

En réponse à ces limites, les technologies domotiques intelligentes introduisent une approche plus avancée, fondée sur l'analyse en temps réel, la prise en compte du contexte et des capacités prédictives, permettant de passer progressivement d'une logique réactive à une logique proactive de prévention des risques.

I.3.2 Typologie des risques domestiques

L'analyse des risques domestiques dans un environnement domotique intelligent permet d'identifier quatre catégories principales de menaces, différenciées selon leur nature, leur dynamique d'évolution et leur degré de prévisibilité.

a) Risque d'intrusion

Le risque d'intrusion correspond à une menace intentionnelle d'origine humaine, caractérisée par son imprévisibilité totale. Il s'agit d'un événement discret, sans phase précurseur exploitable, ce qui rend toute tentative de prédiction particulièrement limitée. Le traitement de ce risque repose essentiellement sur des mécanismes de détection immédiate, notamment via des capteurs infrarouges passifs (PIR) et des contacts magnétiques installés sur les ouvertures. Toutefois, ces dispositifs présentent une limitation majeure : l'absence de prise en compte du contexte d'usage, ce qui peut engendrer des ambiguïtés d'interprétation selon les situations (présence ou absence des occupants, temporalité, activité normale du foyer). [28]

b) Risque de fuite de gaz

La fuite de gaz constitue un risque progressif, insidieux et potentiellement létal. Son danger réside dans sa faible perceptibilité sensorielle et dans sa capacité d'accumulation dans des espaces confinés. La détection repose sur des capteurs de type semi-conducteur (série MQ), mesurant les variations de résistance d'un matériau sensible en présence de gaz spécifiques. Ce type de phénomène présente une dynamique évolutive exploitable, permettant d'envisager des approches d'analyse temporelle et, dans une certaine mesure, des modèles de prédiction basés sur l'évolution des concentrations mesurées. [29]

c) Risque d'incendie

Le risque d'incendie se distingue par sa rapidité de propagation et son caractère hautement destructeur. Dans de nombreux cas, l'évolution d'un départ de feu est quasi instantanée, laissant

un temps de réaction extrêmement limité. La détection est généralement assurée par des capteurs de flamme infrarouges capables d'identifier le rayonnement caractéristique des combustions. Dans ce contexte, la contrainte principale réside dans l'exigence d'une réaction immédiate, où toute latence de traitement peut entraîner une aggravation critique de la situation. [30]

d) Risque de surcharge électrique

La surcharge électrique constitue un risque latent, souvent sous-estimé dans les environnements domestiques. Elle résulte d'un dépassement des capacités nominales des circuits électriques, entraînant une hausse progressive de la température et, potentiellement, des défaillances matérielles critiques. La mesure repose sur des capteurs à effet Hall permettant d'estimer l'intensité du courant à partir du champ magnétique généré. Contrairement à d'autres risques, celui-ci présente une évolution progressive et continue, ce qui lui confère un fort potentiel d'analyse prédictive et de détection anticipée via des modèles de surveillance temporelle. [31]

I.4 Systèmes contextuels (Context-Aware Systems)

I.4.1 Définition et portée du concept de contexte

Le contexte en informatique ne se limite pas à une simple variable, mais représente une structure informationnelle complexe et multidimensionnelle. Depuis les années 1990, sa définition a évolué : d'abord centrée sur des éléments comme la localisation et l'identité (Schilit et al.), puis élargie aux facteurs temporels et environnementaux (Brown et al.). [32], [33]

La définition la plus acceptée est celle de Dey (2000), qui considère le contexte comme toute information permettant de caractériser la situation d'une entité. Cette vision est complétée par d'autres travaux intégrant des dimensions dynamiques telles que l'activité humaine, l'historique et les interactions sociales. [32], [33]

Ainsi, le contexte regroupe un ensemble cohérent d'informations (lieu, temps, activité, historique, interactions) décrivant une situation donnée. Un système contextuel exploite ces informations pour adapter son comportement en fonction de la situation. Dans la domotique, cela permet une adaptation automatique selon la présence des occupants ou le moment de la journée, afin d'améliorer le confort, la qualité de service et la sécurité. [34], [35]

I.4.2 Approches de modélisation du contexte

La diversité des systèmes sensibles au contexte a conduit au développement de plusieurs approches de modélisation, chacune répondant à des contraintes spécifiques en termes de complexité, de flexibilité et d'expressivité. [33]

a) Modèles logiques

- Représentation du contexte sous forme de faits et règles d'inférence
- Basés sur la logique formelle (McCarthy, systèmes multi-contextuels de Giunchiglia)
- Forte rigueur et cohérence
- Limite : complexité élevée de conception et de gestion

b) Modèles ontologiques

- Structuration sémantique des connaissances (concepts et relations)
- Utilisation de standards (ex. OWL) pour l'interopérabilité et le raisonnement automatique
- Applications dans les systèmes intelligents et la domotique
- Intègrent aussi sécurité et partage des connaissances (ex. SOUPA)

c) Modèles orientés objet

- Représentation du contexte sous forme d'objets autonomes (attributs + comportements)
- Favorisent modularité, flexibilité et réutilisabilité
- Adaptés aux environnements dynamiques (IoT, domotique)
- Permettent des interactions en temps réel entre entités

d) Modèles clé-valeur

- Représentation simple via paires clé-valeur
- Accès rapide et implémentation facile
- Adapté aux systèmes simples
- Limite : faible expressivité et absence de structuration complexe. [33], [35]

I.5 Intelligence artificielle et prédiction dans les smart homes

Le développement des maisons intelligentes a profondément modifié les interactions entre les systèmes de sécurité, de confort et de gestion énergétique au sein de l'environnement résidentiel. Contrairement aux architectures traditionnelles, fondées sur des règles déterministes de type *if-then-else*, ces systèmes évoluent désormais vers des approches plus dynamiques reposant sur l'intelligence artificielle et le machine learning.

Ces technologies introduisent une capacité nouvelle d'adaptation, permettant aux systèmes non seulement de réagir à des événements, mais également d'apprendre à partir des données collectées, d'ajuster leur comportement en fonction des usages réels et, dans certains cas, d'anticiper les situations futures. [36], [37]

I.5.1 Détection vs prédiction

Dans les systèmes de maisons connectées, la distinction entre détection et prédiction constitue un élément structurant de l'architecture fonctionnelle. Elle illustre le passage progressif d'un environnement simplement instrumenté vers un système réellement intelligent, capable non seulement de réagir à son environnement, mais également d'en anticiper les évolutions.

La **détection** correspond à l'identification immédiate d'un état ou d'un événement à partir des données issues des capteurs. Elle s'inscrit dans une logique essentiellement réactive, fondée sur des règles explicites ou des seuils prédéfinis appliqués en temps réel. [38] Cette approche permet de transformer des signaux bruts en informations exploitables, telles que la détection d'une intrusion, d'une fuite de gaz ou d'un incendie. Elle répond ainsi à une problématique centrée sur le présent : *« que se passe-t-il actuellement ? ».*

À l'inverse, la **prédiction** repose sur une logique anticipative visant à estimer la survenue d'événements futurs. Elle exploite les données historiques combinées aux données en temps réel afin d'identifier des tendances, des corrélations et des comportements évolutifs. [39]

Dans les systèmes de maisons intelligentes, elle permet notamment d'anticiper une panne, une dérive de consommation énergétique ou un comportement à risque avant leur apparition effective. Elle s'inscrit ainsi dans une perspective tournée vers le futur : *« que risque-t-il de se produire ? »*

Dans cette architecture, l'intelligence artificielle (IA) et le machine learning (ML) assurent une fonction de transition entre la perception des données et la prise de décision. Contrairement aux approches classiques basées sur des règles fixes de type *if-then-else*, ces méthodes introduisent une capacité d'adaptation continue fondée sur l'apprentissage et l'évolution des comportements utilisateurs. Elles permettent d'améliorer la robustesse des systèmes, de réduire les erreurs de classification et d'augmenter la pertinence des décisions automatisées.

Le tableau suivant synthétise les principales différences entre ces deux approches :

Tableau I.5 : Synthèse des principales différences entre détection et prédiction

Critere	Detection	Prediction
Definition	Identification d'un événement ou état **présent** à partir de signaux observables	Anticipation d'un événement **futur** à partir de données passées et présentes
Temporalite	Temps réel (instantané)	Anticipative (projection dans le futur)
Nature de l'approche	Réactive	Proactive
Objectif principal	Signaler ce qui se produit actuellement	Prévoir ce qui pourrait se produire
METHODE principale	Seuils, règles fixes, capteurs (logique déterministe)	Modèles statistiques, IA, machine learning
Exemples en domotique	Détection d'intrusion, fuite de gaz, incendie	Prévision de panne, surconsommation énergétique, comportement à risque
Dependance aux donnees	Données instantanées uniquement	Données historiques + données en temps réel
Resultat produit	Alerte ou action immédiate	Estimation, probabilité ou scénario futur
Niveau d'intelligence	Basique à intermédiaire	Avancé (apprentissage et adaptation)
Limite principale	Faible prise en compte du contexte global	Incertitude liée aux modèles et aux données
ROLE dans un SYSTEME intelligent	Perception immédiate du système	Anticipation et projection du système

I.5.2 Modèles d'apprentissage automatique

Le machine learning (ML) constitue une branche centrale de l'intelligence artificielle qui permet à un système d'apprendre directement à partir des données, sans programmation explicite de règles métier. L'idée fondamentale est de remplacer la logique déterministe par des modèles capables d'identifier des régularités, de généraliser et de prendre des décisions à partir

d'exemples. Dans ce cadre, trois grandes familles de modèles sont généralement distinguées : l'apprentissage supervisé, l'apprentissage non supervisé et l'analyse des séries temporelles. [40], [41]

a) Apprentissage supervisé

L'apprentissage supervisé repose sur l'exploitation de données étiquetées, c'est-à-dire des ensembles pour lesquels la sortie attendue est déjà connue. [42] Le modèle apprend ainsi à établir une relation entre les entrées et les sorties afin de généraliser sur de nouvelles données. Deux types de problèmes sont principalement traités :

- **Classification** : attribution d'une classe (ex. normal/anormal, intrusion/non-intrusion).
- **Régression** : estimation d'une valeur continue (ex. température, consommation, prix). [42]

Parmi les approches les plus utilisées figurent les arbres de décision, qui structurent la prise de décision sous forme hiérarchique de conditions successives, ainsi que le Random Forest, qui améliore la robustesse en combinant plusieurs arbres. Les SVM (Support Vector Machines) permettent quant à eux de déterminer une frontière optimale de séparation entre classes, tandis que certains modèles probabilistes comme Naive Bayes exploitent les probabilités conditionnelles pour des tâches rapides de classification. [43]

b) Apprentissage non supervisé

L'apprentissage non supervisé s'applique à des données non étiquetées, où aucune réponse prédéfinie n'est disponible. L'objectif n'est plus de prédire une sortie connue, mais de découvrir des structures cachées au sein des données. [40], [41]

Deux approches principales sont généralement utilisées :

- **Clustering (regroupement)** : organisation automatique des données en groupes homogènes. Des algorithmes comme k-means partitionnent les données en clusters, tandis que DBSCAN identifie des regroupements basés sur la densité et permet également de détecter des points aberrants. [41]
- **Réduction de dimensionnalité** : simplification de jeux de données complexes en conservant l'information essentielle. Des méthodes comme l'ACP (Analyse en Composantes Principales) ou les auto-encodeurs permettent de compresser les données tout en facilitant leur analyse et la détection d'anomalies. [40], [41]

c) Séries temporelles

Les séries temporelles correspondent à des données ordonnées dans le temps, où chaque observation dépend partiellement des précédentes. Ce type de données est particulièrement

fréquent dans les systèmes intelligents (consommation énergétique, température, activité des capteurs).

L'analyse repose généralement sur la décomposition en trois composantes : tendance globale, variations saisonnières et bruit aléatoire. [44]

Plusieurs modèles sont utilisés :

- **ARIMA**, modèle statistique classique combinant mémoire des valeurs passées et correction des erreurs, adapté aux prévisions à court terme. [44]
- **LSTM (Long Short-Term Memory)**, réseau de neurones récurrent capable de mémoriser des dépendances longues, particulièrement utile pour détecter des évolutions progressives ou des anomalies temporelles. [41]
- **XGBoost et LightGBM**, modèles de boosting performants qui transforment le problème temporel en variables supervisées classiques. [43]
- **Transformers**, approches plus récentes basées sur un mécanisme d'attention, permettant de pondérer les instants temporels les plus pertinents. [43]

I.6 Synthèse critique et positionnement scientifique

L'analyse des systèmes de sécurité domestique existants, qu'ils soient issus de solutions commerciales ou de travaux académiques, met en évidence un ensemble de limites structurelles récurrentes qui freinent leur évolution vers une véritable intelligence domotique. Ces insuffisances ne sont pas simplement techniques, mais traduisent également des choix de conception qui restent encore largement fragmentés et insuffisamment intégrés.

L'analyse de la littérature révèle quatre limites majeures :

1. **Approche mono-risque** : La majorité des systèmes actuels sont conçus pour traiter une menace unique à la fois (intrusion, incendie, fuite de gaz, etc.), ce qui entraîne une segmentation des dispositifs et empêche une vision globale de la sécurité domestique. Cette absence d'unification limite fortement la cohérence fonctionnelle des solutions
2. **Absence de prédiction** : Les systèmes reposent essentiellement sur des mécanismes de type seuil-action, où l'alerte est déclenchée uniquement après détection d'un événement. Si cette approche est efficace pour des incidents soudains, elle montre ses limites face à des phénomènes progressifs tels que la surcharge électrique ou l'accumulation de gaz, pour lesquels une anticipation serait plus pertinente.
3. **Manque de contextualisation** : Les systèmes actuels appliquent des règles statiques sans tenir compte de la situation réelle d'usage (présence ou absence des occupants, horaires, habitudes de vie). Cette rigidité entraîne des erreurs d'interprétation

fréquentes, notamment des faux positifs, ce qui peut réduire la confiance des utilisateurs et conduire à une désactivation partielle des dispositifs.

4. **Dépendance au réseau** : De nombreuses solutions dépendent d'une connexion internet permanente, ce qui les rend vulnérables en cas de coupure, y compris dans des situations critiques où la continuité du service est essentielle.

Face à ces constats, une problématique centrale se dégage **: *comment concevoir un système domotique de sécurité capable d'intégrer plusieurs risques, d'anticiper les événements, de s'adapter au contexte et de fonctionner de manière autonome en cas de perte de connectivité ?***

C'est dans ce cadre que s'inscrit la contribution de ce travail, qui ne vise pas une simple amélioration progressive des systèmes existants, mais une évolution de paradigme vers une architecture domotique plus intelligente et intégrée. Cette contribution repose sur quatre axes principaux :

- **Intégration multi-risques :** conception d'un système unifié capable de gérer simultanément plusieurs types de menaces, afin de dépasser les approches fragmentées actuelles.
- **Capacité d'anticipation :** introduction de mécanismes prédictifs permettant d'identifier des comportements ou évolutions anormales avant la survenue des incidents.
- **Adaptation contextuelle :** prise en compte dynamique des conditions d'usage (présence, temporalité, environnement) afin d'améliorer la pertinence des décisions et de réduire les déclenchements erronés.
- **Autonomie hors ligne :** maintien du fonctionnement local du système afin d'assurer la continuité des services critiques même en l'absence de connectivité réseau.

Ainsi, ce travail s'inscrit dans une démarche de transition vers des systèmes domotiques plus robustes, intelligents et autonomes. Bien que ces apports ne couvrent pas l'ensemble des problématiques du domaine, ils constituent une base cohérente pour la conception de solutions de sécurité résidentielles plus intégrées et évolutives.

I.7 Conclusion

À travers ce chapitre, nous avons posé les fondements théoriques indispensables à la réalisation de notre projet en explorant les notions clés liées à la domotique, à l'Internet des objets (IoT), aux systèmes intelligents ainsi qu'aux différents risques domestiques. Cette étude a permis d'analyser l'existant afin d'en comprendre les apports, mais surtout d'en identifier les

limites principales, notamment en termes d'intégration globale des systèmes, de prise en compte du contexte et de capacités prédictives.

L'ensemble de cette analyse met en évidence un décalage significatif entre les solutions actuelles et les exigences d'un environnement résidentiel moderne, qui doit être à la fois intelligent, sécurisé et autonome. C'est dans cette perspective que s'inscrit la suite de ce travail, qui vise la conception et le développement d'un système domotique de sécurité répondant à ces insuffisances, à travers une approche plus intégrée, adaptative et évolutive.

CHAPITRE II :

Analyse des besoins et conception de l'architecture d'une maison Intelligente

II.1 Introduction

Dans ce chapitre, nous abordons la phase de conception du système domotique intelligent. Après avoir établi les bases théoriques nécessaires dans le chapitre précédent, il s'agit maintenant de traduire ces connaissances en une architecture concrète et structurée.
Nous commencerons par définir le cahier des charges du système, en identifiant les besoins fonctionnels et non fonctionnels ainsi que les besoins matériels qui orientent sa conception. Ensuite, nous analyserons les différents scénarios de risques domestiques afin de comprendre les comportements attendus du système face à chaque situation critique.

Nous proposerons par la suite une modélisation de la maison intelligente sous forme de zones fonctionnelles, accompagnée d'une justification du placement optimal des capteurs. Cette étape permettra de structurer l'environnement physique du système de manière cohérente et efficace.

Nous présenterons également la conception de l'architecture globale, en décrivant le flux de traitement des données depuis les capteurs jusqu'aux actionneurs, ainsi que les différentes couches de traitement, incluant l'intelligence artificielle et la prise de décision.
Enfin, nous détaillerons la conception logicielle du système, les diagrammes UML associés, ainsi que les choix technologiques retenus pour assurer la mise en œuvre du système.

Ce chapitre constitue ainsi une étape essentielle de transition entre l'étude théorique et la réalisation pratique du projet.

II.2 Analyse des scénarios de risques

Le système prend en charge plusieurs scénarios de risques domestiques, chacun caractérisé par une dynamique physique différente et une logique de traitement spécifique. L'objectif n'est pas uniquement la détection, mais également l'anticipation, la localisation et l'action adaptée selon la gravité et la nature du danger.

II.2.1 Intrusion :

L'intrusion est le scénario le plus contextuel de notre système. Un mouvement détecté peut représenter une situation parfaitement normale ou une menace sérieuse selon le moment et les circonstances. C'est pourquoi ce scénario repose entièrement sur le contexte pour prendre ses décisions, et c'est également le seul dans lequel l'utilisateur est intégré dans la boucle de décision.

Le système fonctionne selon deux modes distincts. En mode absence, toute détection déclenche immédiatement une notification à l'utilisateur et l'alarme s'active.En mode présence, la surveillance active est limitée à la plage horaire nocturne, entre minuit et six heures du matin. En dehors de cette plage, les détections sont ignorées. Durant la nuit, toute détection déclenche directement

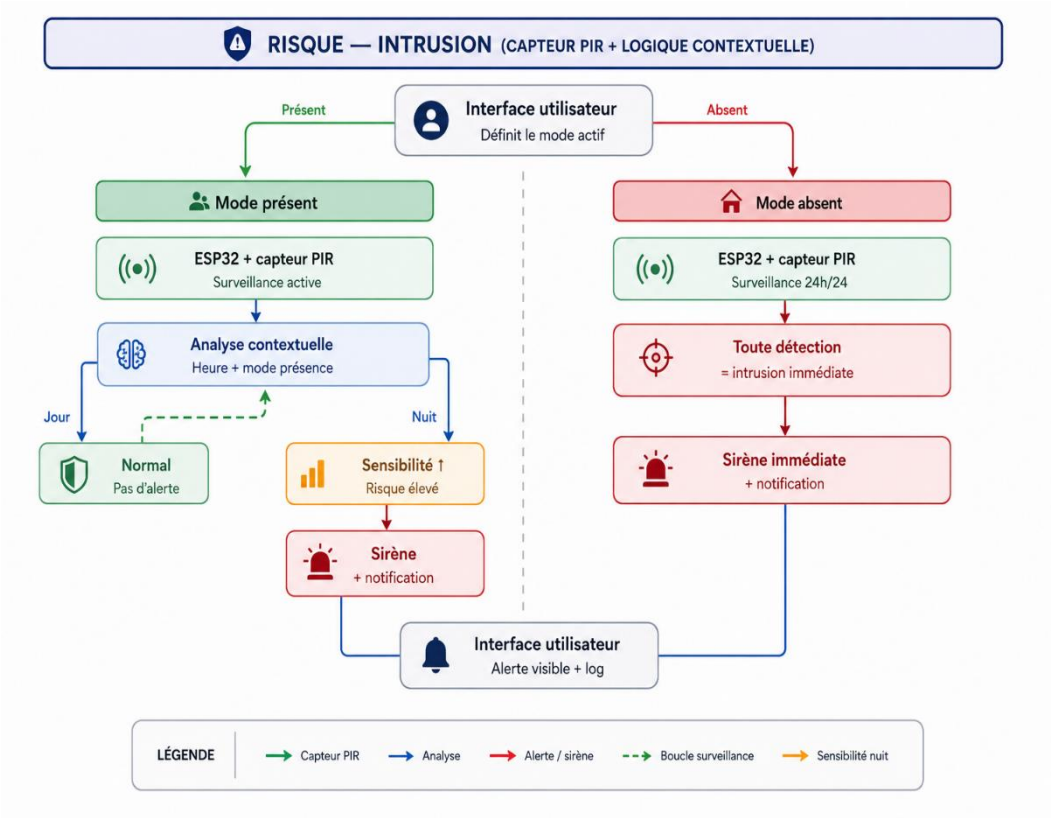


Figure II.1 : Chaîne de traitement du scénario d'intrusion

II.2.2 Incendie

Le scénario d'incendie est basé sur un capteur de flamme directionnel à 5 voies, permettant non seulement la détection du feu mais également la localisation approximative de sa source.

Contrairement aux systèmes classiques, la réponse ne se limite pas à une alerte sonore. Le système adopte une stratégie d'intervention active :

- Détection immédiate de la flamme via un capteur infrarouge.
- Estimation de la direction dominante de la source du feu.
- Activation d'un servomoteur permettant d'orienter le dispositif d'extinction.
- Déclenchement du système d'extinction (pompe ou agent extincteur).

Cette approche permet une extinction **ciblée et automatique**, réduisant le temps de propagation du feu.

Le traitement est entièrement local au niveau du **microcontrôleur embarqué**, sans dépendance réseau, afin de garantir une réaction instantanée.

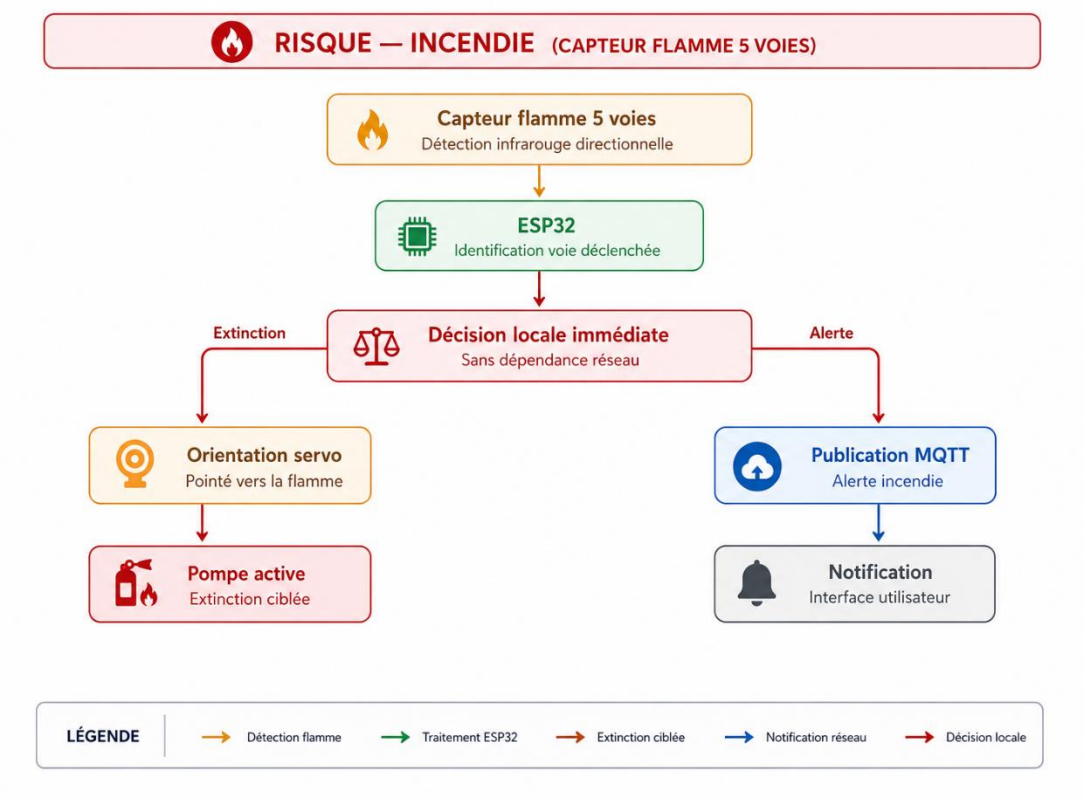


Figure II.2 : Logique de détection et d'extinction incendie

II.2.3 Fuite de gaz

La fuite de gaz constitue un scénario critique à évolution progressive, détecté via le capteur MQ-5.

Le système ne se limite pas à un seuil fixe, mais exploite une analyse temporelle des mesures pour détecter les tendances anormales.

a) Détection progressive

- Surveillance continue de la concentration en gaz.
- Analyse de la dérive temporelle (augmentation lente mais continue).
- Utilisation d'un modèle d'intelligence artificielle pour anticiper l'atteinte du seuil critique.

b) Actions correctives

Lorsqu'un risque est confirmé ou anticipé :

- Fermeture automatique de l'électrovanne de gaz.
- Activation du système de ventilation pour dilution du gaz.
- Notification utilisateur via dashboard.

c) Mode dégradé

En cas de perte de connexion :

- Le microcontrôleur applique une logique locale basée sur un seuil critique
- Les actions de sécurité sont exécutées sans intervention du serveur.

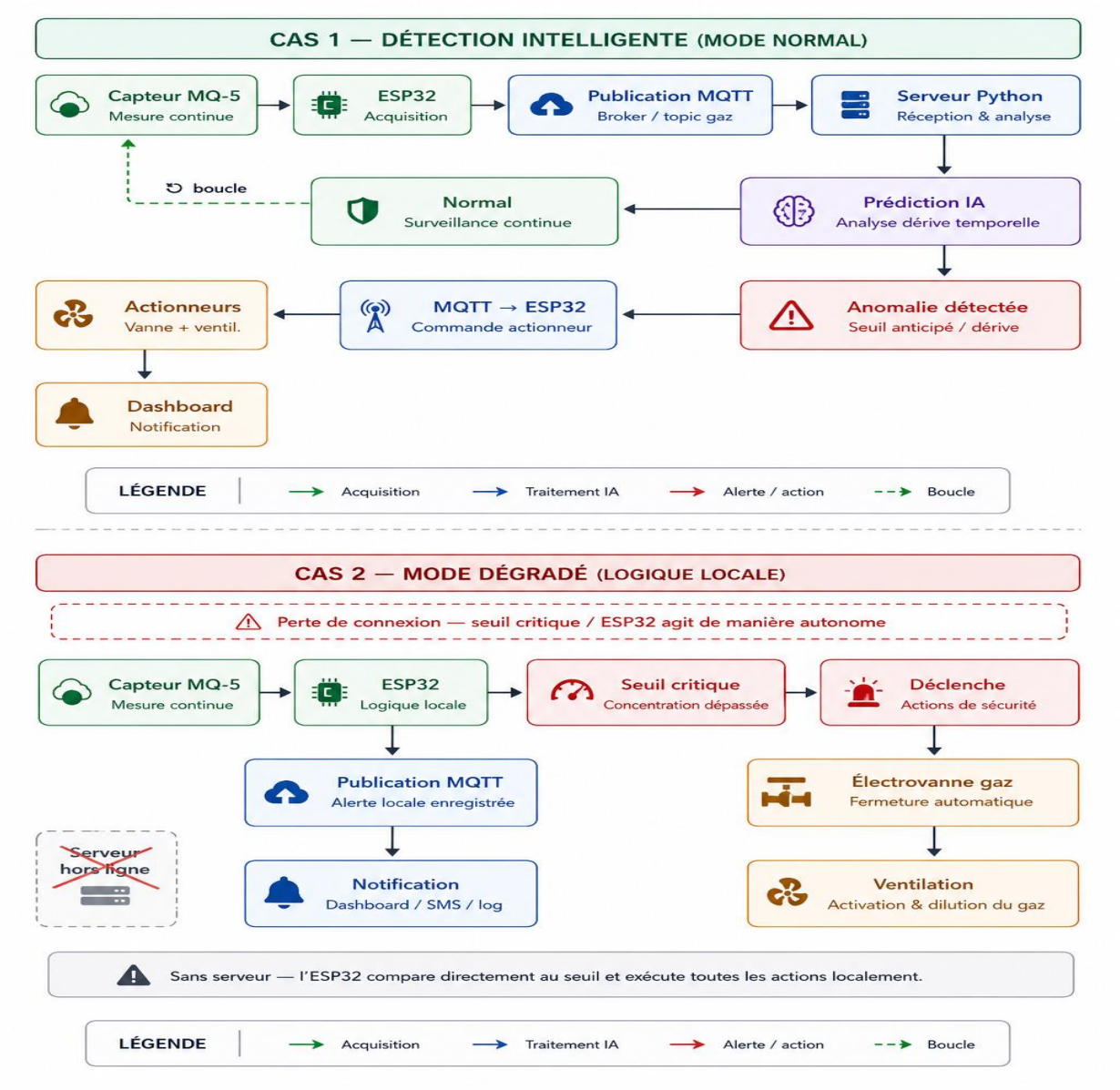


Figure II.3 : Chaîne de traitement fuite de gaz

II.2.4 Surcharge électrique

La surcharge électrique est caractérisée par deux comportements distincts : évolution progressive et défaut brutal.

a) Surcharge progressive

• Surveillance continue du courant via un capteur de type ACS712.

• Détection des dérives de consommation.

• Analyse des tendances pour anticiper les dépassements.

Dans ce cas, le système coupe automatiquement le circuit concerné via un relais.

b) Court-circuit (cas critique)

• Augmentation brutale et instantanée du courant.

• Détection immédiate sans analyse temporelle.

• Coupure instantanée du circuit sans communication réseau.

c) Isolation des circuits

Chaque circuit est surveillé indépendamment :

• Cuisine, salon, chambres, garage.

• Protection des circuits vitaux (communication, systèmes critiques).

Cela permet une coupure localisée et non globale, garantissant la continuité du fonctionnement du système.

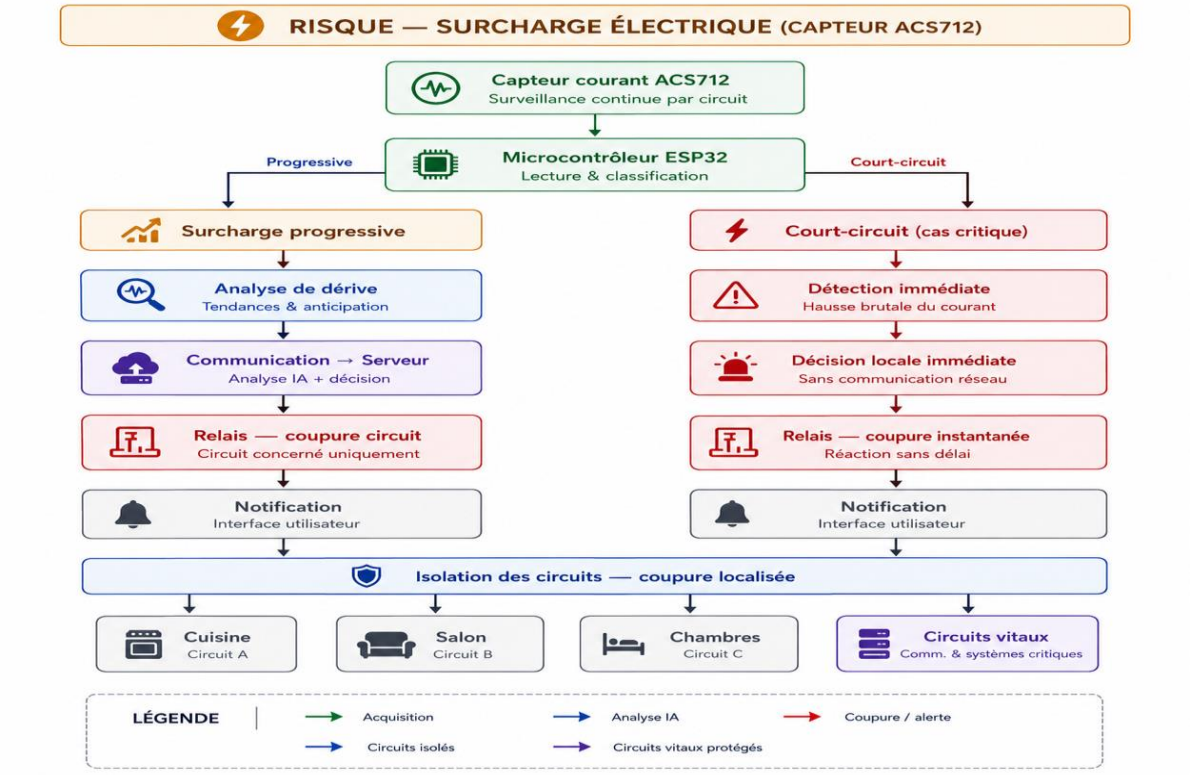


Figure II.4 : Détection et réaction surcharge électrique

II.3 Cahier des charges

Avant de concevoir le système, il est nécessaire de définir précisément ses objectifs et les conditions dans lesquelles il doit fonctionner. Cette étape constitue le cahier des charges. Elle permet de formaliser les attentes fonctionnelles et non fonctionnelles du système sans entrer encore dans les choix matériels ou logiciels.

On distingue deux catégories principales d'exigences :

• **Les besoins fonctionnels**, qui décrivent les services que le système doit assurer.
• **Les besoins non fonctionnels**, qui définissent les contraintes de qualité et de performance associées à ces services. [45]

En complément, certaines contraintes techniques doivent être prises en compte, notamment en termes de **ressources matérielles**, de **consommation énergétique** et de **connectivité**, afin de garantir la faisabilité du système dans un environnement domestique réel. [45]

II.3.1 Besoins fonctionnels

Les besoins fonctionnels d'un système domotique intelligent ne peuvent pas se limiter à une simple liste de fonctionnalités. Ils doivent refléter les situations réelles d'usage, les attentes des occupants et les contraintes d'un environnement domestique dynamique. [45]

Dans ce projet, quatre fonctions principales ont été identifiées :

- la surveillance continue de l'environnement
- la détection et la prédiction des anomalies
- le déclenchement automatique des actions correctives
- la supervision à distance et la traçabilité des événements

Ces fonctions sont interdépendantes et forment une chaîne logique complète où chaque étape conditionne la suivante.

a) Surveillance continue de l'environnement

La surveillance constitue la base du système. Elle repose sur la collecte continue des données issues des capteurs. [46] Cette couche relève de l'instrumentation et conditionne directement la fiabilité globale du système.

Le système doit assurer la mesure en continu des grandeurs physiques suivantes :

- concentration de gaz combustible
- détection de flamme
- intensité du courant électrique
- détection de mouvement

La fréquence d'échantillonnage est adaptée à la nature du phénomène : lente pour le gaz

et le courant, rapide pour la flamme et l'intrusion.

Le système doit également fonctionner en mode local en cas de perte de connexion réseau afin de garantir la continuité de la surveillance.

b) Détection des anomalies et prédiction des risques

Cette fonction constitue le cœur intelligent du système. Elle repose sur des techniques d'intelligence artificielle et d'analyse de séries temporelles.

Le système combine deux approches :

- une détection par seuil pour les situations critiques immédiates
- une approche prédictive basée sur l'apprentissage automatique

Les modèles analysent l'évolution temporelle des données afin d'anticiper les situations dangereuses avant leur apparition.

Le système intègre également une dimension contextuelle (présence, absence, horaires), permettant de réduire les faux positifs et d'adapter les décisions à la réalité de l'environnement.

c) Déclenchement automatique des actions correctives

Le système doit être capable d'agir automatiquement après détection ou prédiction d'un risque. Cette fonction relève de l'automatisme et du contrôle-commande.

Les actions varient selon le type de risque :

- coupure de gaz via électrovanne
- extinction directionnelle en cas d'incendie
- coupure de circuit électrique via relais
- activation d'alarme en cas d'intrusion

Le système applique une hiérarchie de décision entre prévention et réaction immédiate. En mode hors ligne, les décisions critiques sont exécutées localement sans dépendance au serveur.

d) Supervision, notification et traçabilité

Cette fonction concerne l'interaction avec l'utilisateur. Elle garantit la compréhension et la transparence du système.

Le système doit :

- notifier les événements importants
- afficher l'état en temps réel des zones
- permettre la consultation de l'historique
- enregistrer toutes les actions et mesures

Ces données sont stockées afin d'assurer la traçabilité et d'améliorer les performances du système.

II.3.2 Besoins non fonctionnels

Les besoins non fonctionnels définissent les conditions de fonctionnement du système. [45]

a) Temps réel

Le système doit traiter les données en temps réel sans délai perceptible. Cela permet une réaction immédiate face aux situations critiques comme une flamme ou une intrusion.

b) Robustesse

Le système doit continuer à fonctionner malgré les perturbations (coupure réseau, instabilité serveur). Une logique locale embarquée garantit cette continuité.

c) Fiabilité

Le système doit minimiser les faux positifs tout en garantissant la détection des vrais dangers. L'équilibre entre précision et sensibilité est assuré par l'intelligence artificielle et le contexte.

II.4 Modélisation de la maison intelligente

Avant de définir l'architecture logicielle, il faut d'abord modéliser la maison comme un système. Dans une approche d'ingénierie embarquée, elle n'est pas vue comme un simple ensemble de pièces, mais comme un système cyber-physique structuré en zones interconnectées, chacune ayant ses usages et ses risques.

La maison est donc représentée comme un graphe fonctionnel, où chaque zone constitue un nœud associé à des contraintes physiques et à des scénarios de danger (gaz, incendie, intrusion, surcharge électrique). Les interactions entre zones permettent aussi de prendre en compte la propagation des incidents.

Cette modélisation sert de base à toutes les décisions du système, notamment le placement des capteurs, la détection des événements et les réponses automatisées.

II.4.1 Découpage en zones fonctionnelles

La modélisation de la maison commence par une segmentation basée sur les usages, et non sur une simple découpe spatiale. L'espace domestique est donc divisé selon les activités qui s'y déroulent et les risques associés.

Chaque zone est considérée comme une unité fonctionnelle de risque, décrite par son activité principale, son niveau de consommation énergétique et matérielle, les dangers auxquels elle est exposée (gaz, incendie, intrusion, surcharge électrique) ainsi que sa capacité à propager un incident vers d'autres zones.

Cette représentation permet d'attribuer à chaque zone un profil de risque complet, utilisé ensuite pour adapter le placement des capteurs et guider les décisions du système.

Tableau II.1 : Modélisation fonctionnelle des zones et profils de risque

Zone	Fonction dominante	Profil de risque principal
Cuisine	Production thermique et usage gaz	Fuite de gaz, incendie, surcharge électrique
Salon	Espace de vie et multimédia	Intrusion, surcharge électrique, incendie
Chambres	Repos et faible réactivité humaine	Intrusion nocturne, incendie
Couloir	Axe de circulation et interface inter-zones	Intrusion, propagation de fumée
Garage	Stockage et énergie mécanique	Fuite de gaz, incendie, surcharge électrique, intrusion

Ce partitionnement n'est pas uniquement descriptif : il est **opérationnel**. Il conditionne la capacité du système à localiser un événement avec précision et à déclencher une réponse ciblée. Sans cette granularité, le système se limiterait à une détection globale non actionnable.

II.4.2 Stratégie de placement des capteurs et contraintes physiques

Une fois le modèle zonal établi, l'étape suivante consiste à définir une stratégie de déploiement spatial optimal des capteurs. Cette étape relève d'un problème d'ingénierie des systèmes distribués sous contraintes physiques (diffusion de gaz, propagation thermique, champs infrarouges, etc.).

Chaque capteur est positionné selon des critères de :

- couverture géométrique optimale,
- réduction des angles morts,
- réactivité physique du phénomène mesuré,
- minimisation du bruit et des fausses détections.

Tableau II.2 : Déploiement des capteurs MQ-5

Zone	Position	Justification technique
Cuisine	30–50 cm du sol, proche des installations gaz	Source primaire de gaz domestique
Garage	20–30 cm du sol, zone de stationnement	Accumulation possible en espace peu ventilé
Couloir	~1 m, position centrale	Capteur sentinelle de propagation inter-zones

Ce dispositif introduit une logique de **détection hiérarchique** : source (cuisine/garage) → propagation (couloir).

Tableau II.3 : Déploiement des capteurs PIR

Zone	Position	Rôle fonctionnel
Couloir	2 m, orienté entrée	Point de contrôle primaire
Salon	Angle haut, couverture fenêtres	Surveillance accès secondaires
Chambres	Au-dessus des portes	Intrusion en période de repos
Garage	Zone haute, porte principale	Accès secondaire critique

Tableau II.4 : Déploiement des capteurs de flamme

Zone	Position	Justification
Cuisine	Plafond central	Zone à forte densité énergétique
Salon	Plafond central	Charge électrique élevée
Couloir	Plafond central	Détection propagation incendie
Garage	Plafond orienté stockage	Présence de matériaux inflammables

Les chambres sont exclues volontairement : optimisation coût/risque et faible probabilité d'ignition.

Tableau II.5 : Surveillance électrique par ACS712

Circuit	Justification
Cuisine	Charges thermiques élevées et simultanées
Salon	Usage multimédia intensif
Chambres	Consommation modérée mais continue
Garage	Charges variables et équipements motorisés

II.4.3 Synthèse globale du système de capteurs

Le système complet constitue une architecture de surveillance distribuée hybride intégrant quatre classes de capteurs complémentaires : gaz, mouvement, incendie et courant.

Tableau II.6 : Synthèse multi-zones

Zone	Gaz	Mouvement	Flamme	Courant	Total
Cuisine	1	—	1	1	3
Salon	—	1	1	1	3
Chambres	—	1/chambre	—	1	≥2

Couloir	1	1	1	—	3
Garage	1	1	1	1	4

Le système déploie un minimum de 16 capteurs, assurant une redondance fonctionnelle contrôlée et une couverture multi-risque cohérente.

II.5 Conception de l'architecture globale

L'architecture du système repose sur une chaîne de traitement complète allant de l'acquisition des données jusqu'à l'exécution des actions correctives et à la supervision utilisateur. Elle adopte une approche hybride combinant un traitement local embarqué et un traitement centralisé sur serveur.

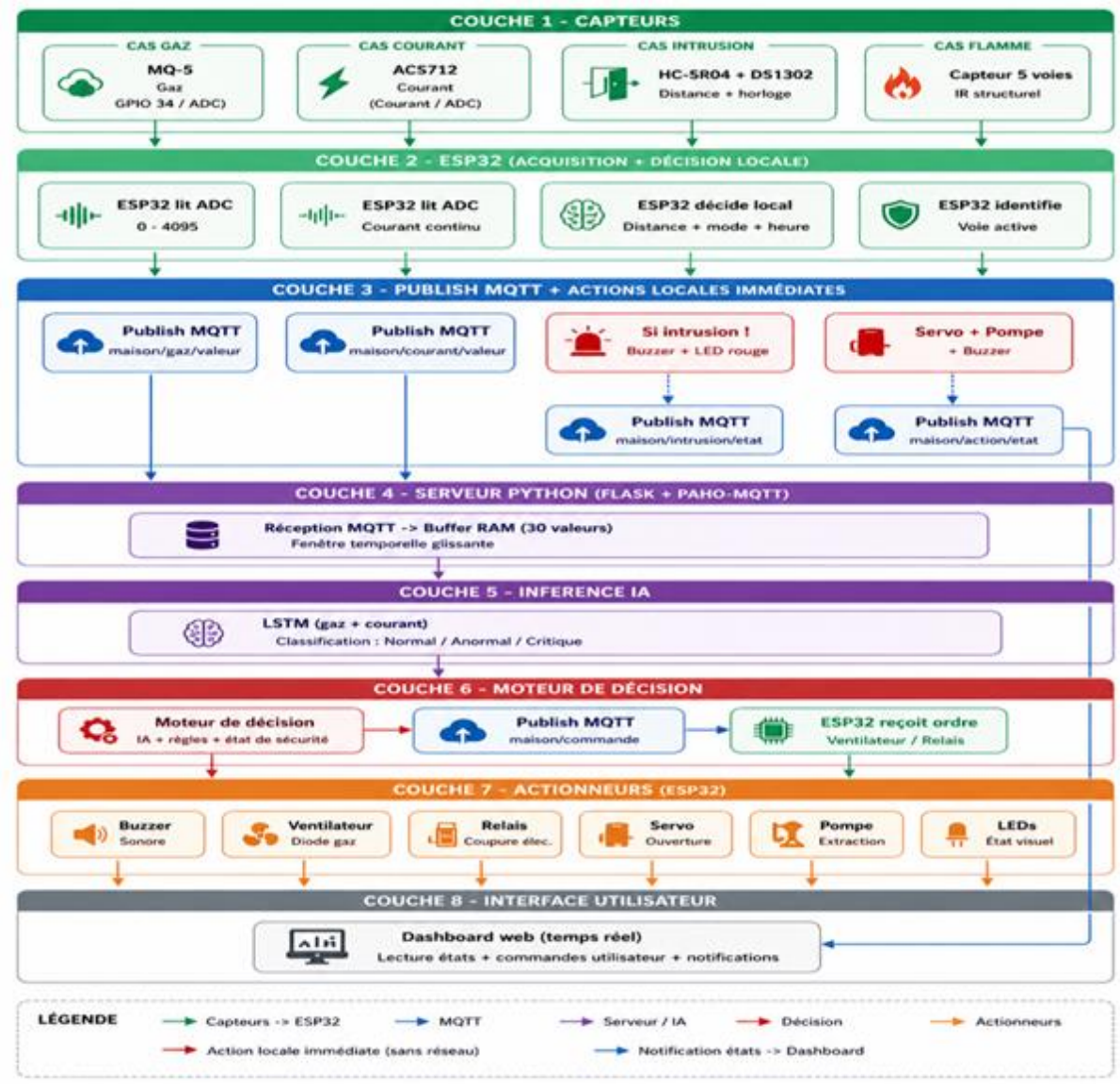


Figure II.5: Architecture globale de systeme

II.5.1 Conception logicielle

Le système est organisé selon une architecture modulaire distribuée, séparant les traitements embarqués exécutés sur le microcontrôleur et les traitements centralisés exécutés sur un serveur. Cette séparation est motivée par les contraintes de ressources des systèmes embarqués et la nécessité de réaliser des traitements complexes (analyse temporelle, inférence de modèles) côté serveur.

L'architecture repose sur six modules fonctionnels interconnectés, communiquant via le protocole MQTT.

II.5.1.1 Module Acquisition (Edge)

Ce module assure l'acquisition des données issues des capteurs et leur transmission vers le serveur. Il gère les signaux numériques (détection de mouvement, flamme) ainsi que les signaux analogiques (gaz, courant) via conversion ADC. En cas de perte réseau, un mode autonome permet l'exécution de règles de sécurité locales. [47]

II.5.1.2 Module Traitement

Ce module centralise le prétraitement des données. Il transforme les mesures brutes en séries temporelles exploitables, en extrayant des caractéristiques telles que la tendance et la variation. Les données sont ensuite stockées dans la base de données pour analyse et apprentissage. [48]

II.5.1.3 Module IA prédictif

Ce module réalise l'estimation des risques à partir des données historiques et en temps réel. Des modèles d'apprentissage supervisé et séquentiel (ex. : LSTM pour les phénomènes temporels) sont utilisés pour détecter les comportements anormaux et anticiper les situations critiques. [49]

II.5.1.4 Module Moteur Contextuel

Ce module embarqué introduit une dimension contextuelle au système. Il combine l'état temporel et l'entrée utilisateur (mode présence/absence) afin d'adapter la logique de détection d'intrusion. Les décisions critiques peuvent être prises localement sans dépendance au serveur. [50]

II.5.1.5 Module Décision

Ce module constitue le centre de décision du système. Il fusionne les résultats du module IA, les données des capteurs et le contexte utilisateur afin de déclencher les actions appropriées. Il applique une logique hybride combinant règles déterministes et prédiction. [50]

II.5.1.6 Module Interface

Ce module assure la supervision du système via une interface web. Il permet la

visualisation en temps réel des alertes, de l'état des capteurs et des actions déclenchées, offrant une transparence complète sur le fonctionnement du système. [51]

II.5.2 Choix technologiques

Les technologies retenues dans ce projet ont été sélectionnées selon une approche d'ingénierie système, en privilégiant la cohérence architecturale, la robustesse, la performance en temps réel et la compatibilité avec un environnement IoT distribué.

L'architecture globale repose sur une organisation en couches distinctes : couche embarquée (Edge), couche communication, couche intelligence, couche persistance et couche supervision.

II.5.2.1 ESP32 : acquisition et action physique (couche Edge)

ESP32 constitue le nœud embarqué du système.

Il assure deux fonctions principales :

- acquisition des données issues des capteurs (gaz, mouvement, flamme, courant électrique)
- activation des actionneurs (sirène, relais, pompe, ventilateur, électrovanne)

Le choix de l'ESP32 est justifié par :

- intégration native du Wi-Fi
- faible consommation énergétique
- compatibilité avec les systèmes embarqués IoT
- capacité de traitement suffisante pour des applications temps réel

Il représente la couche d'interaction directe avec l'environnement physique. [52]

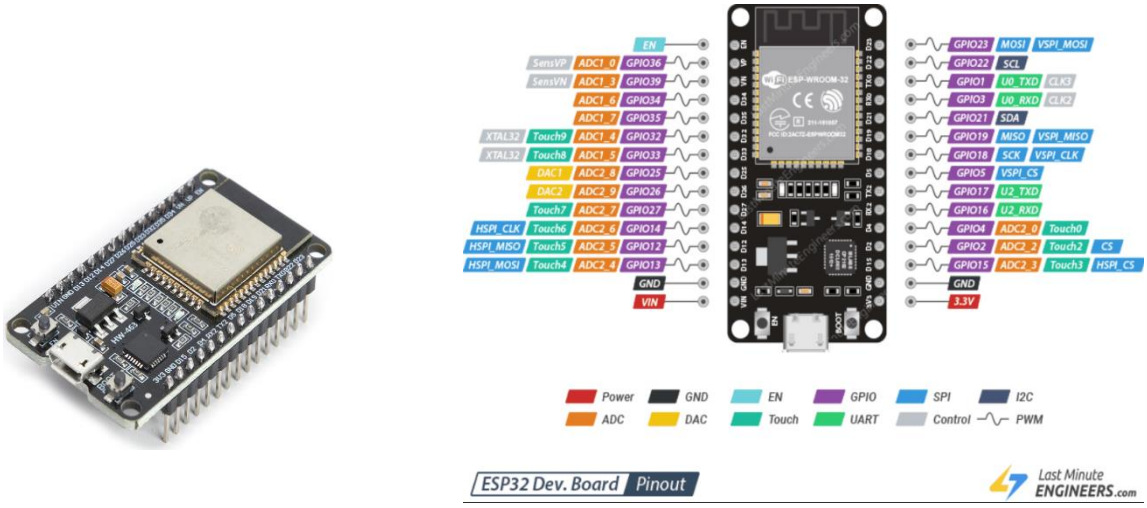


Figure II.6 : Le brochage du microcontrôleur (Source : lastminuteengineers.com)

II.5.2.2 MQTT (Mosquitto) : communication temps réel

MQTT, implémenté via Eclipse Mosquitto, est utilisé comme protocole de communication central.

Il assure :

- la transmission des données de l'ESP32 vers le serveur (publish)
- la diffusion des commandes du serveur vers l'ESP32 (subscribe)

Les raisons de ce choix sont :

- protocole léger adapté aux systèmes embarqués
- faible consommation réseau
- communication asynchrone efficace
- robustesse en cas de réseau instable (QoS + reconnexion automatique)

MQTT constitue donc le pilier de communication du système IoT. [17]

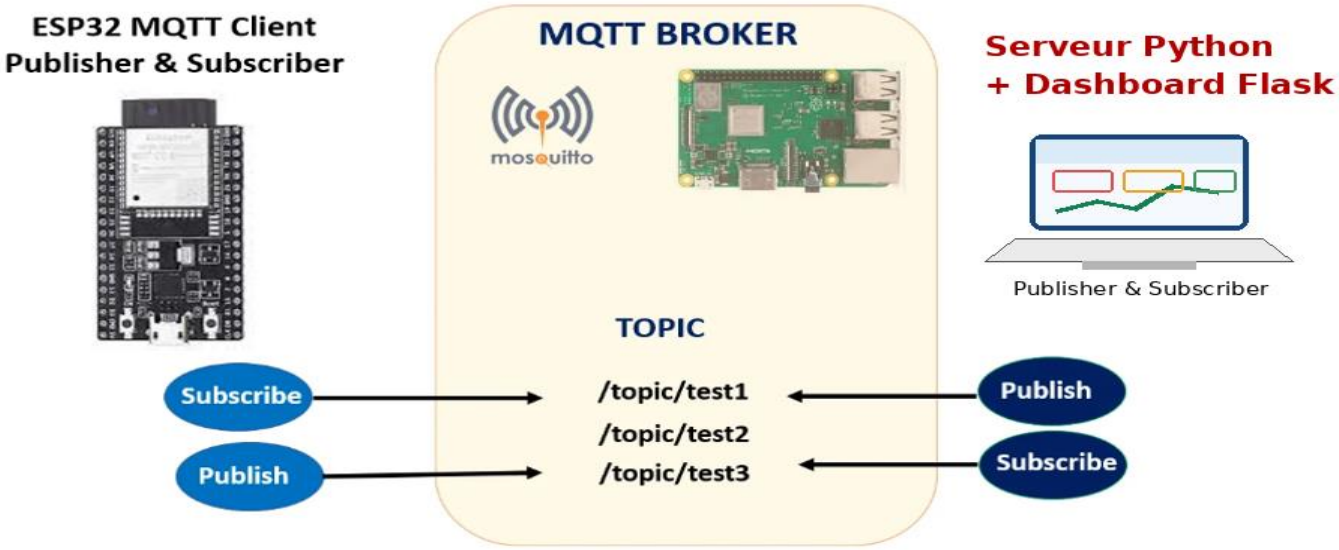


Figure II.7: Échanges publish/subscribe de l'ESP32 via le broker MQTT Eclipse Mosquitto

II.5.2.3 Python : intelligence et traitement des données

Python est utilisé côté serveur pour la logique de traitement et l'intelligence artificielle.

Il permet :

- le traitement des flux de données en temps réel
- l'intégration du modèle LSTM pour la prédiction des risques
- la mise en place d'un pipeline d'analyse des données capteurs

Les avantages principaux sont :

- Écosystème riche en IA et data science

- compatibilité avec MQTT et FastAPI
- support de l'asynchronisme pour le temps réel
- portabilité sur différents environnements (PC, serveur local)

Python constitue donc la couche d'intelligence décisionnelle du système. [53]

II.5.2.4 MySQL : persistance des données

MySQL est utilisé pour la gestion des données structurées. Il stocke :

- les mesures capteurs horodatées
- les alertes générées
- les prédictions du modèle IA
- l'historique des événements

Les raisons de ce choix :

- robustesse et maturité industrielle
- modèle relationnel adapté aux données structurées
- intégration simple avec Python (SQLAlchemy / connectors)
- performance suffisante pour un système IoT local

MySQL assure la traçabilité et l'historisation complète du système. [54]

II.5.2.5 Flask : couche serveur web et API

Flask est utilisé pour servir le dashboard et exposer les données du système. Il permet :

- le service de la page HTML via le moteur de templates Jinja2
- la communication entre backend et interface utilisateur via API REST
- la récupération de l'état du système en temps réel (endpoint /api/état)
- la réception des commandes utilisateur (endpoint /api/commande)

Ses avantages :

- framework Python léger et minimaliste, simple à mettre en œuvre
- réponses JSON natives via jsonify
- intégration directe avec le pipeline Python (LSTM, MQTT, MySQL)
- suffisant pour la charge d'un système IoT local

Flask constitue la couche d'interconnexion logicielle du système. [55]

II.5.2.6 HTML / CSS / JavaScript : interface utilisateur

Le dashboard de supervision est développé en HTML, CSS et JavaScript vanilla, avec Chart.js pour les graphiques en temps réel. Il permet :

- la visualisation en temps réel des capteurs
- l'affichage des alertes
- la consultation de l'historique

- une interaction avec le système (mode présence/absence, commandes)

Ses avantages :

- contrôle total sur le design de l'interface
- graphiques temps réel légers avec Chart.js
- aucune dépendance lourde côté client

Cette stack représente la couche de supervision utilisateur, légère et entièrement personnalisable. [56]

II.6 Modélisation UML

La modélisation UML (Unified Modeling Language) constitue une étape essentielle dans la conception du système, permettant de représenter de manière formelle et structurée les interactions, les comportements dynamiques et l'architecture globale. Elle facilite la compréhension du système, tant au niveau fonctionnel que technique, et sert de support de communication entre les différentes phases de développement. [57]

Dans ce projet, trois types de diagrammes UML complémentaires ont été retenus : le diagramme de cas d'utilisation, les diagrammes de séquence et le diagramme de déploiement. Chacun répond à une problématique spécifique de la conception.

II.6.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet d'identifier les différents acteurs du système ainsi que les fonctionnalités principales auxquelles ils ont accès. Il fournit une vision globale des interactions entre l'utilisateur et le système. [57]

Dans notre cas, trois acteurs principaux ont été identifiés :

- **L'utilisateur** : il interagit avec le système via une interface de supervision. Il peut consulter l'état du système, recevoir des alertes et configurer certains paramètres.
- **Le microcontrôleur (ESP32)** : il assure l'acquisition des données issues des capteurs et l'exécution des actions sur les actionneurs.
- **Le serveur** : il traite les données, exécute les modèles d'intelligence artificielle et prend les décisions.

Les principaux cas d'utilisation sont les suivants :

- Consulter l'état des capteurs en temps réel
- Recevoir des notifications en cas de danger
- Configurer le mode de fonctionnement (présence/absence)
- Analyser les données et prédire les risques

- Déclencher des actions correctives (alarme, coupure, extinction)
- Stocker les données et événements

Ce diagramme met en évidence la répartition des responsabilités entre les différents composants du système.

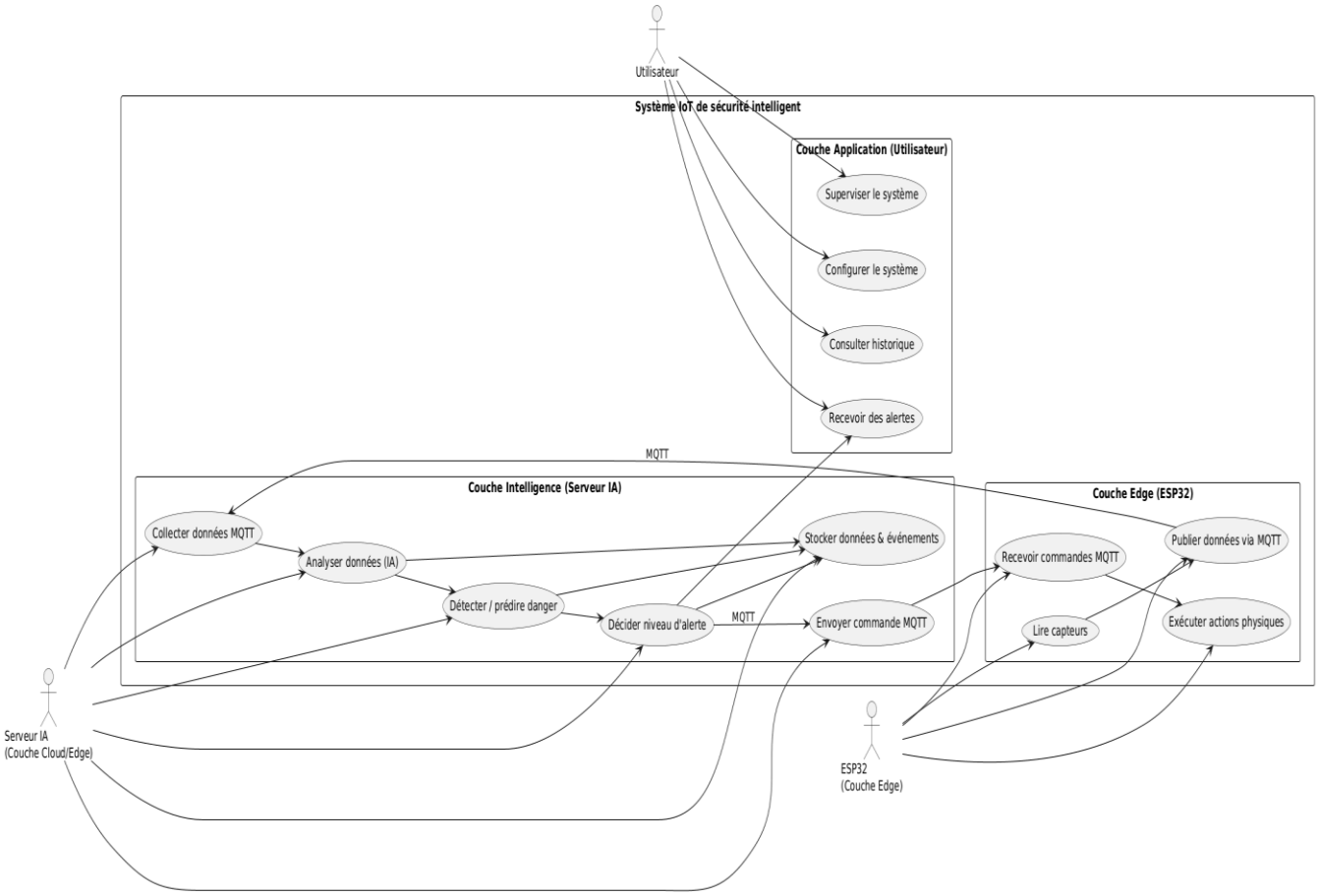


Figure II.8 : Modélisation des interactions utilisateur–système (diagramme de cas d'utilisation)

II.6.2 Diagrammes de séquence

Les diagrammes de séquence décrivent le déroulement temporel des interactions entre les composants du système pour un scénario donné. Ils permettent de visualiser précisément le flux d'informations et la logique de traitement. [57]

Deux scénarios représentatifs ont été modélisés.

a) Scénario de fuite de gaz (approche prédictive)

Dans ce scénario, le capteur de gaz mesure en continu la concentration et transmet les données à l'ESP32. Ces données sont ensuite envoyées via le protocole MQTT vers le serveur, où elles sont traitées et analysées par le modèle d'intelligence artificielle.

Le modèle identifie une tendance anormale et prédit un risque imminent. Le serveur

prend alors la décision de déclencher une action préventive et envoie une commande à l'ESP32. Celui-ci active l'électrovanne pour couper le gaz et met en marche la ventilation. Une notification est simultanément envoyée à l'utilisateur.

Ce diagramme met en évidence le rôle central du serveur dans la prise de décision prédictive.

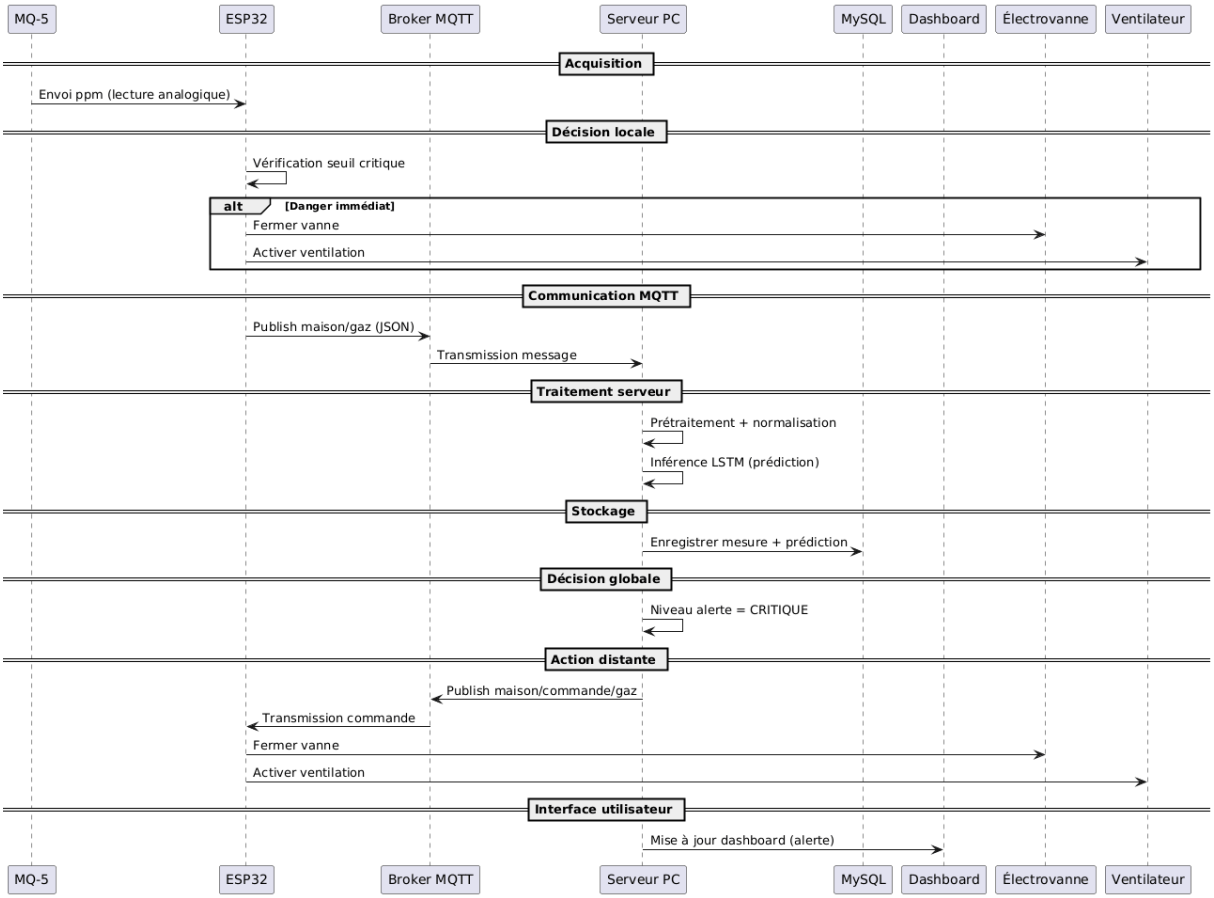


Figure II.9 : Diagramme de séquence : prédiction de fuite de gaz

b) Scénario d'intrusion (logique contextuelle)

Dans ce scénario, depuis le dashboard, l'utilisateur sélectionne le mode de fonctionnement : en mode **absence**, toute détection déclenche immédiatement une notification et active l'alarme. En mode **présence**, la surveillance est limitée à la plage nocturne (minuit – 6 h), et toute détection durant cette fenêtre déclenche directement l'alarme ; en dehors, les détections sont ignorées. Ce diagramme illustre l'importance de la contextualisation dans la réduction des fausses alertes.

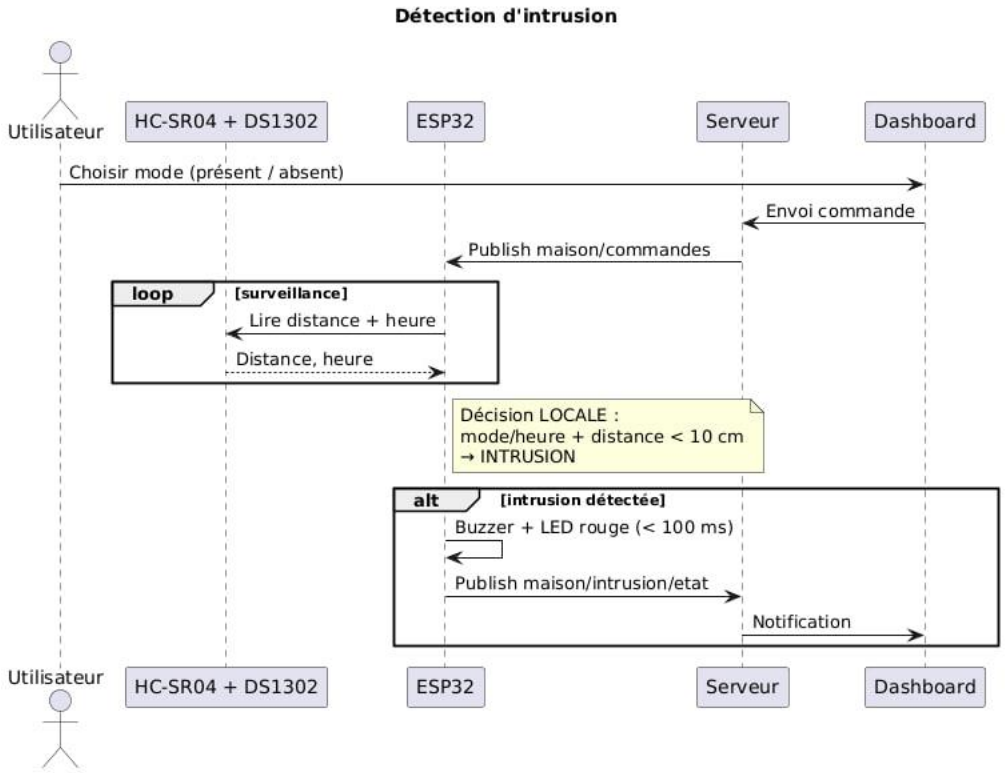


Figure II.10 : Diagramme de séquence : détection d'intrusion

II.6.3 Diagramme de déploiement

Le diagramme de déploiement décrit la répartition physique des composants du système sur les différents nœuds matériels. [57]

Le système est structuré autour de trois éléments principaux **:**

- **Le microcontrôleur ESP32 :** Il constitue le nœud embarqué du système. Il est responsable de l'acquisition des données des capteurs et du contrôle des actionneurs. Il communique avec le serveur via le réseau WiFi en utilisant le protocole MQTT.
- **Le serveur local :** Il héberge le broker MQTT, les modules de traitement, le modèle d'intelligence artificielle, ainsi que la base de données. Il joue le rôle de centre décisionnel du système.
- **L'interface utilisateur (PC ou smartphone) :** Elle permet la visualisation des données, la réception des alertes et l'interaction avec le système via une interface web.

Cette architecture distribuée permet de séparer clairement les responsabilités : traitement local pour la réactivité, traitement centralisé pour l'intelligence et la supervision.
L'utilisation des diagrammes UML dans ce projet permet de formaliser de manière rigoureuse la conception du système, en mettant en évidence à la fois les interactions fonctionnelles, les flux d'information et l'architecture physique.

Cette modélisation constitue une base solide pour la phase de réalisation, en garantissant une compréhension claire et partagée du fonctionnement global du système.

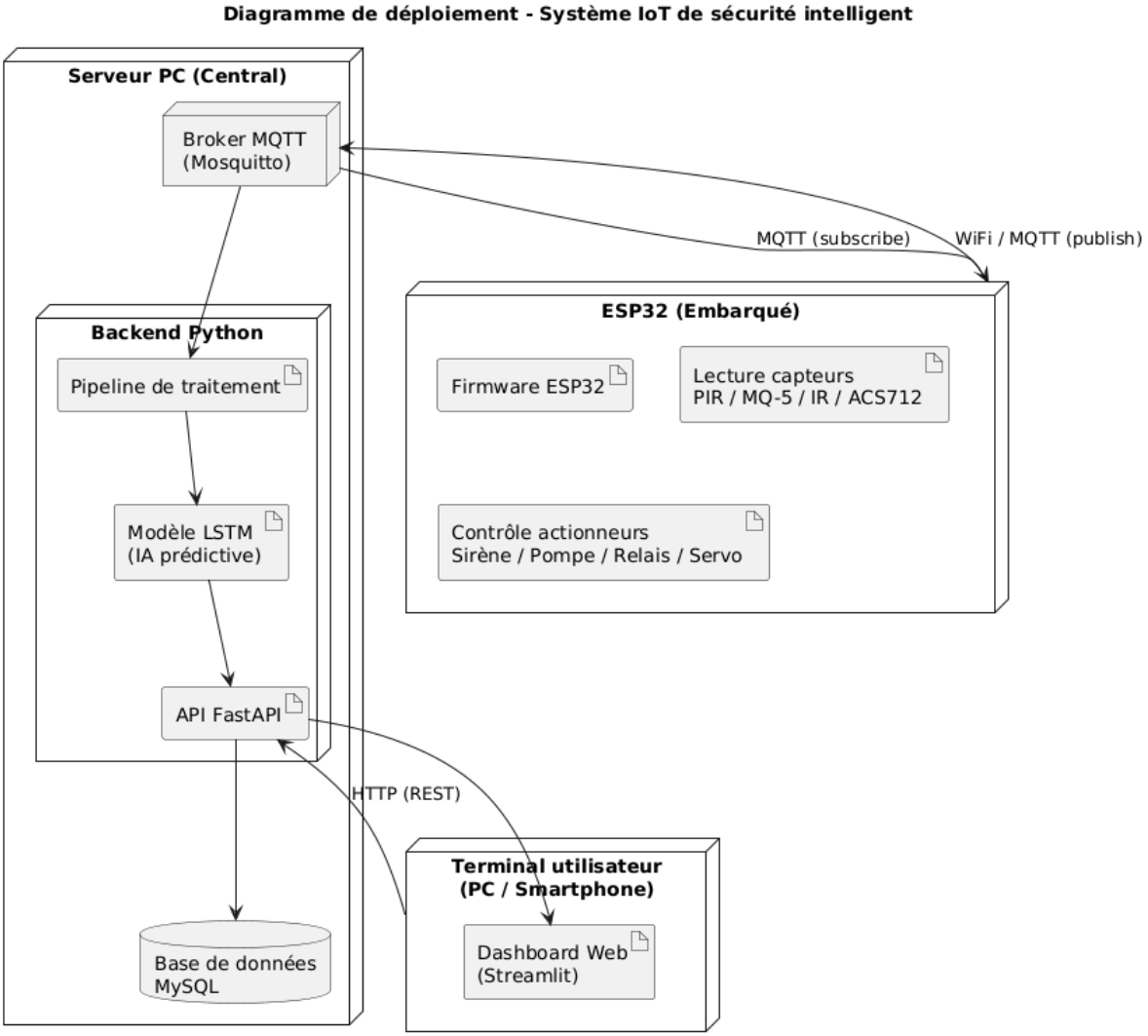


Figure II.11 : diagramme de déploiement du système de sécurité intelligent.

II.7 Conclusion

Ce chapitre a permis de définir de manière structurée l'ensemble des éléments nécessaires à la conception du système domotique intelligent. À partir du cahier des charges, nous avons analysé les scénarios de risques, modélisé l'environnement domestique, puis conçu l'architecture matérielle et logicielle du système.

L'utilisation de la modélisation UML et le choix de technologies adaptées garantissent une base solide pour la mise en œuvre. L'ensemble de ces travaux constitue une conception complète et cohérente, prête à être concrétisée dans le chapitre suivant consacré à la réalisation et à la validation du prototype.

CHAPITRE III :

Modélisation contextuelle et Intelligence artificielle pour la prédiction des risques

III.1 Introduction

Ce chapitre présente la méthodologie de transformation des données issues des capteurs en décisions de classification et de réaction, selon trois approches distinctes adaptées à la nature des dangers considérés.

Dans les cas de fuite de gaz et de surcharge électrique, les signaux présentent une évolution continue dont les variations progressives constituent un indicateur pertinent de dérive du système. Une approche prédictive basée sur l'apprentissage profond, notamment les réseaux LSTM (Long Short-Term Memory), est adoptée afin de modéliser les dépendances temporelles et d'anticiper l'évolution du signal avant l'atteinte d'un état critique.

À l'inverse, la détection d'intrusion repose sur un signal de nature binaire, correspondant à une présence ou une absence détectée. Toutefois, son interprétation est fortement dépendante du contexte d'occurrence, notamment de l'heure et de l'état d'occupation des lieux. Une logique déterministe contextuelle est ainsi mise en œuvre afin d'intégrer ces variables exogènes dans le processus de décision et de réduire les ambiguïtés d'interprétation.

Enfin, dans le cas du risque d'incendie, la contrainte dominante est celle de la latence décisionnelle, où la rapidité de réaction constitue un critère critique. Toute latence liée à un traitement complexe étant incompatible avec les exigences de sécurité, une réponse directe et locale est implémentée sur le microcontrôleur ESP32.

Ces trois paradigmes complémentaires traduisent un principe méthodologique fondamental : l'adéquation entre la nature du problème et l'outil de traitement mobilisé, plutôt qu'une approche uniforme appliquée à l'ensemble des situations

III.2 Vue d'ensemble des trois approches

Le système proposé ne repose pas sur une homogénéisation des traitements appliqués aux différents risques étudiés (fuite de gaz, surcharge électrique, intrusion et incendie).

Une telle uniformisation, bien qu'attrayante sur le plan de l'implémentation, ne serait pas adaptée aux différences fondamentales entre les dynamiques des signaux considérés. En effet, chacun de ces phénomènes se distingue par une signature physique spécifique, une dynamique d'évolution propre, ainsi que des contraintes temporelles distinctes.

Ainsi, certains dangers se caractérisent par des variations progressives et continues, tandis que d'autres se manifestent de manière discrète ou brutale, nécessitant des stratégies de traitement différenciées.

Le tableau III.1 synthétise les choix méthodologiques retenus en mettant en

correspondance, pour chaque type de risque, la nature du signal, l'approche adoptée et la justification associée.

Tableau III.1 : Synthèse des trois approches méthodologiques

Danger	Signal observé	Approche retenue	Justification
Fuite de gaz	Concentration ADC continue, montée progressive	Prédictive (LSTM)	L'analyse de la tendance permet d'agir avant le seuil critique
Surcharge électrique	Courant ACS712 continu, hausse progressive	Prédictive (LSTM)	Même logique que le gaz — la tendance précède le danger
Intrusion	Distance binaire (présence / absence)	Contextuelle (règles)	L'interprétation dépend de l'heure et du mode de présence d'utilisateur
Incendie / flamme	Détection IR directe (5 voies)	Action directe locale	Apparition brutale — toute latence serait inacceptable

Les signaux associés à la fuite de gaz et à la surcharge électrique présentent une évolution continue caractérisée par des variations progressives exploitables dans une modélisation temporelle. Cette nature justifie le recours à une approche prédictive basée sur des réseaux de neurones de type LSTM, capables d'anticiper l'évolution du système avant l'atteinte d'un état critique.

À l'inverse, la détection d'intrusion repose sur un signal binaire dont l'interprétation dépend fortement du contexte, notamment de l'heure et de l'état d'occupation des lieux. Une logique déterministe intégrant ces variables permet d'assurer une prise de décision robuste et interprétable.

Enfin, le risque d'incendie constitue un cas particulier dans lequel la contrainte dominante est la réactivité immédiate. La détection infrarouge multi-capteurs permet une identification quasi instantanée de la flamme, rendant toute latence de traitement incompatible avec les exigences de sécurité. Une réaction locale et embarquée sur ESP32 est donc retenue.

Ces trois approches — prédictive, contextuelle et réactive — sont complémentaires et reflètent un principe central du système : adapter la méthode de traitement à la nature du problème, tout en garantissant l'indépendance fonctionnelle des modules.

III.3 Approche prédictive (Cas du gaz et du courant)

Cette section présente le protocole d'acquisition, de structuration et de modélisation des données relatives à la fuite de gaz et à la surcharge électrique. Ces deux cas partagent une même méthodologie d'apprentissage supervisé fondée sur l'analyse de séries temporelles via des réseaux de neurones récurrents de type LSTM.

Le cas de la fuite de gaz, déjà entièrement collecté, constitue le scénario de référence détaillé. Le cas du courant électrique suit une méthodologie strictement identique .

III.3.1 Acquisition et structuration des données

III.3.1.1 Dispositif expérimental et protocole (cas du gaz)

Le dispositif expérimental est volontairement simplifié afin de garantir la reproductibilité et de limiter les sources de bruit. Il est composé d'un capteur MQ-5, d'un microcontrôleur ESP32 assurant l'acquisition via un convertisseur analogique-numérique 12 bits (plage 0–4095), ainsi que d'un système d'enregistrement des mesures au format CSV.

Le capteur MQ-5 est retenu pour sa sensibilité aux gaz combustibles domestiques (propane, butane, GPL, méthane) et sa réponse relativement limitée face à certains interférents environnementaux. Afin de stabiliser ses caractéristiques, il a été maintenu sous tension pendant 24 heures avant la phase de collecte.[58], [59]

La collecte s'est déroulée en continu pendant 37 h 49 min, avec une durée d'échantillonnage moyenne de 5,5 secondes. Des scénarios de fuite contrôlés ont été introduits afin de générer des dynamiques représentatives des variations réelles du signal.

III.3.1.2 Structure des données

Les données sont organisées sous forme de série temporelle composée de trois attributs principaux : un identifiant, un horodatage et une mesure brute issue de l'ADC.

Tableau III.2 : Structure du dataset gaz (n = 24 667)

Champ	Type	Description	Exemple
Id	INT	Identifiant unique auto-incrémenté	1, 2, 3...
Timestamp	DATETIME	Horodatage de la mesure	2026-03-11 13:47:00
Gaz	INT (0–4095)	Valeur brute ADC du MQ-5	597

III.3.1.3 Labellisation et définition des seuils

Les données sont étiquetées selon une classification supervisée à trois classes : normale, anomalie et critique. Les seuils de décision ne sont pas définis empiriquement, mais dérivés de sources techniques et normatives.

Tableau III.3 : Sources de définition des seuils

Source	Donnée clé	Valeur
Datasheet MQ-5 [58]	Plage de détection officielle	200 – 10 000 ppm
Datasheet MQ-5 [58]	Concentration de calibration	1 000 ppm GPL
NIOSH / NOAA [60], [61]	LEL du propane	21 000 ppm
Norme EN 50194 [62]	Seuil légal d'alarme domestique	10 % LEL ≈ 2 100 ppm
Pratique	Seuil de préalarme (notre cas)	5 % LEL ≈ 1 050 ppm

La LEL (Lower Explosive Limit) correspond à la concentration minimale à partir de laquelle un mélange gazeux devient inflammable. Le ppm (parts per million) représente une concentration volumique relative dans l'air.

III.3.1.4 Conversion et calibration ADC

La conversion des concentrations physiques en valeurs ADC dépend de la résistance interne du capteur R_0, qui varie selon chaque composant. Celle-ci est estimée à partir des mesures en air propre (164 ADC), correspondant au percentile 25 du dataset, permettant d'obtenir $R_0 \approx 36{,}9\ k\Omega$.

À partir de cette calibration et des courbes de sensibilité fournies par la datasheet, les seuils théoriques sont estimés à environ 660 ADC et 1850 ADC. Afin d'assurer une distribution suffisante des classes pour l'apprentissage supervisé, ces seuils sont ajustés à :

- 500 ADC
- 1500 ADC

Tableau III.4 : Seuils finaux de labellisation

Classe	Seuil ADC	Interprétation
0 — Normal	< 500	Air sain / hors détection
1 — Anomalie	500 – 1499	Début de variation significative
2 — Critique	≥ 1500	Zone proche du seuil de danger

III.3.1.5 Distribution des classes

Dans le cadre d'un apprentissage supervisé, la distribution des classes constitue un indicateur fondamental de la qualité statistique du jeu de données. Elle conditionne directement la capacité du modèle à généraliser correctement sur l'ensemble des états représentés. La répartition obtenue après labellisation est présentée ci-dessous.

Tableau III.5 : Distribution des classes

Classe	Lignes	Proportion
0 — Normal	23 535	95,41 %
1 — Anomalie	845	3,43 %
2 — Critique	287	1,16 %

Cette distribution fortement déséquilibrée reflète la rareté des situations de fuite dans un environnement réel. Afin de préserver la représentativité des données, aucune augmentation artificielle n'a été appliquée. Le déséquilibre est traité par pondération des classes lors de l'entraînement.

III.3.1.6 Cas du courant

Le traitement du signal de courant suit un protocole identique basé sur le capteur ACS712, exploité via l'ADC de l'ESP32. L'ensemble du pipeline (acquisition, structuration et labellisation) repose sur la même méthodologie.

III.3.2 Prétraitement des données

Les données brutes doivent être transformées afin d'être compatibles avec un modèle séquentiel de type LSTM. Cette étape comprend trois opérations principales :

- nettoyage,
- normalisation
- transformation en séquences temporelles.

A. Nettoyage

Le processus de nettoyage vise à garantir l'intégrité du dataset en supprimant les doublons, valeurs manquantes et incohérences liées à l'acquisition. Après traitement, 24 655 échantillons sont conservés sur 24 667, soit un taux de qualité de 99,95 %.

B. Normalisation

Les valeurs ADC, comprises entre 0 et 4095 (une plage trop large pour un réseau de neurones qui apprend mieux entre 0 et 1) sont normalisées afin d'améliorer la stabilité numérique et la convergence du réseau de neurones. La transformation MinMax est définie comme suit :

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

La normalisation MinMax ramène chaque valeur dans l'intervalle [0, 1], ainsi 0 devient 0, 4095 devient 1, et 1500 devient ≈ 0,37, etc. Les paramètres de normalisation sont sauvegardés (scaler.pkl) afin de garantir la cohérence entre entraînement et déploiement.

C. Fenêtrage temporel

Les données sont transformées en séquences temporelles afin de permettre au LSTM d'exploiter les dépendances temporelles.

Chaque échantillon d'entrée correspond à une fenêtre de 30 valeurs consécutives, associée à une sortie correspondant à la classe à prédire à un horizon de 2 pas de temps.

Tableau III.6 : Structure des données après fenêtrage

Élément	Description
X	24 623 séquences de 30 valeurs normalisées
y	Classe associée (0, 1 ou 2)

D. Division train/test

La séparation des données est effectuée de manière chronologique afin d'éviter toute fuite d'information temporelle (data leakage). Le dataset est divisé comme suit :

Tableau III.7 : Répartition train / test

Ensemble	Proportion	Taille	Rôle
Train	80 %	~19 698	Apprentissage
Test	20 %	~4 925	Évaluation

III.3.3 Modèle prédictif LSTM

III.3.3.1 Justification de l'approche

Les approches basées sur des seuils constituent des méthodes de détection efficaces pour l'identification d'états critiques, mais elles présentent une limitation structurelle majeure : elles sont intrinsèquement réactives et non anticipatives. Autrement dit, elles interviennent uniquement lorsque la valeur mesurée franchit un niveau prédéfini, ce qui correspond déjà à une situation de danger avéré ou imminent.

Dans le contexte de ce travail, une telle stratégie est insuffisante, car l'objectif n'est pas uniquement de détecter une situation dangereuse, mais d'anticiper son apparition. En effet, dans les cas de fuite de gaz ou de surcharge électrique, la phase la plus critique du point de vue de la

prévention correspond à la zone de transition, où le signal évolue progressivement vers un état critique sans l'avoir encore atteint.

Ainsi, l'enjeu principal consiste à exploiter les dynamiques temporelles du signal afin d'identifier des motifs précurseurs d'anomalie. Cette exigence justifie le recours à une approche d'apprentissage automatique capable de modéliser l'évolution temporelle des données.

III.3.3.2 Justification du LSTM

Le réseau de neurones de type Long Short-Term Memory (LSTM), introduit par Sepp Hochreiter et Jürgen Schmidhuber en 1997, constitue une extension des réseaux de neurones récurrents (RNN) conçue spécifiquement pour résoudre le problème de dépendances temporelles longues. [63]

Les RNN classiques souffrent en effet de limitations connues, notamment le phénomène de disparition ou d'explosion du gradient, ce qui limite leur capacité à apprendre des relations temporelles sur des séquences longues. Le LSTM introduit une architecture dotée de mécanismes de mémoire interne (cell state) et de portes de contrôle (input gate, forget gate, output gate), permettant de conserver, filtrer et exploiter l'information pertinente sur de longues périodes.

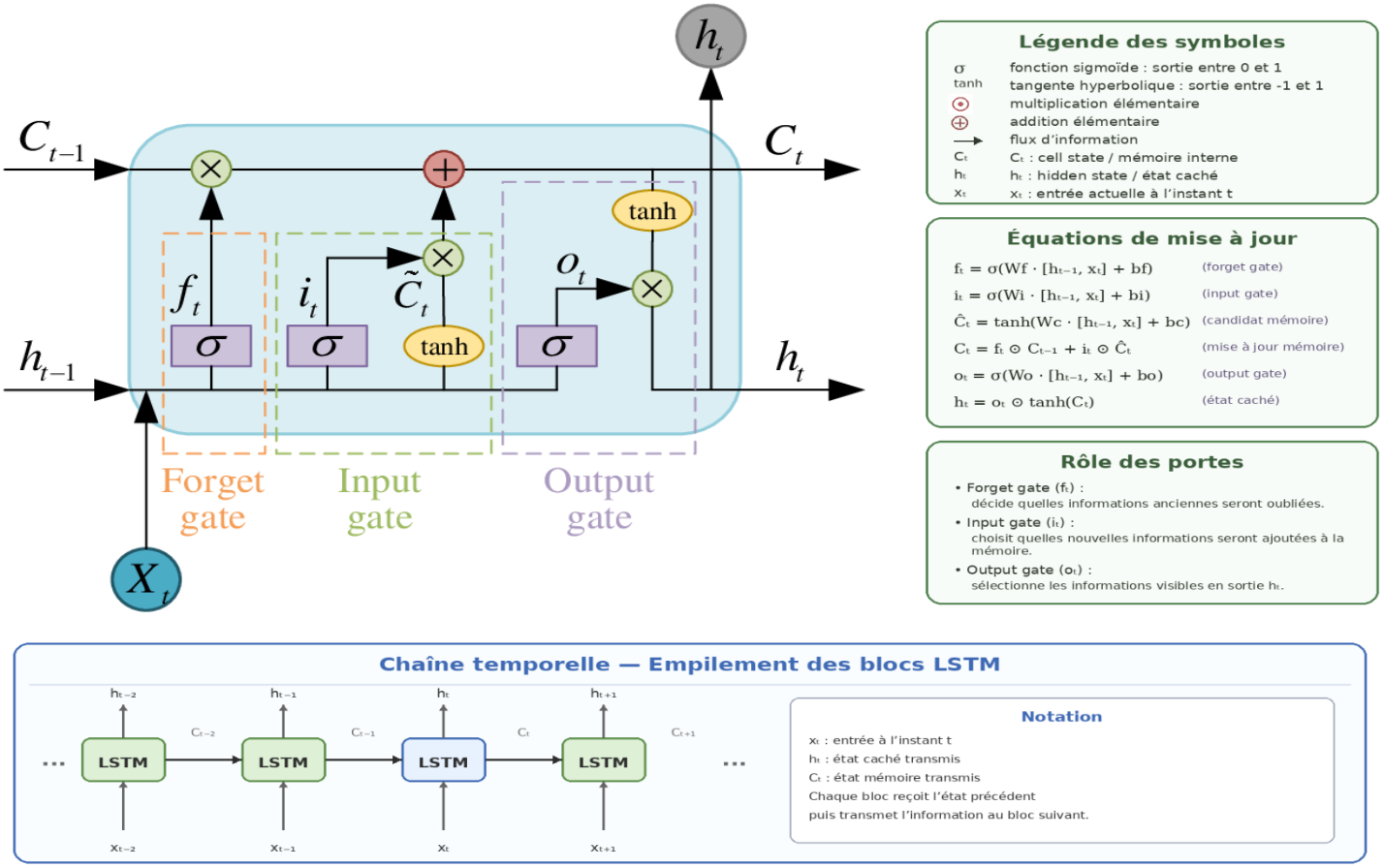


Figure III.1 : Architecture interne d'un bloc LSTM

Dans le cadre de ce travail, cette capacité est essentielle, car les signaux issus des capteurs (MQ-5 et ACS712) ne présentent pas une variation instantanée mais une évolution progressive dans le temps. Le modèle doit donc être capable de distinguer :

- une variation ponctuelle isolée,
- d'une tendance persistante indicative d'une dérive réelle.

Le LSTM est ainsi particulièrement adapté à la détection de motifs temporels tels que les montées progressives ou les fluctuations anormales répétées.

III.3.3.3 Principe de prédiction anticipée

Le modèle est conçu pour exploiter des fenêtres temporelles glissantes issues de la série temporelle, permettant une analyse contextuelle du comportement du signal sur une durée donnée.

Concrètement, chaque entrée du modèle correspond à une séquence de valeurs consécutives, à partir desquelles le réseau apprend à associer une dynamique temporelle à une classe de sortie (normale, anomalie, critique).

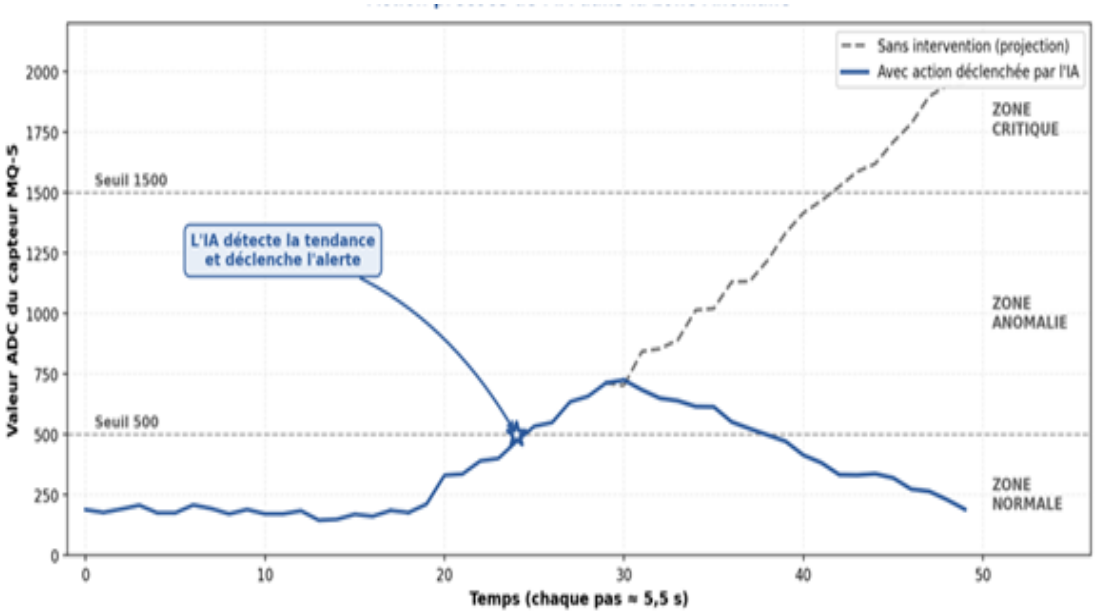


Figure III.2 : Illustration du mécanisme de prédiction anticipée en zone d'anomalie

L'objectif n'est pas uniquement de classifier l'état actuel du système, mais d'anticiper son évolution à court terme. Ainsi, lorsque le signal entre dans une phase de dérive progressive (zone d'anomalie), le modèle peut identifier des tendances caractéristiques d'une transition vers un état critique.

Cette capacité d'anticipation permet une détection précoce, c'est-à-dire avant l'atteinte du seuil critique, ce qui constitue un avantage majeur dans des systèmes de sécurité embarqués.

III.3.3.4 Logique décisionnelle hybride

Le système repose sur une architecture décisionnelle hybride combinant deux mécanismes complémentaires :

- **Détection par seuil** : elle assure une sécurité déterministe et immédiate en cas de dépassement d'un niveau critique. Cette approche garantit qu'aucune situation dangereuse n'est ignorée, même en cas de défaillance du modèle prédictif.
- **Prédiction par LSTM** : elle permet d'anticiper les évolutions du signal dans la zone intermédiaire (zone d'anomalie), où les variations ne sont pas encore critiques mais présentent une dynamique suspecte.

Cette combinaison permet de fusionner robustesse et anticipation :

- La couche seuil agit comme un **filet de sécurité (safety layer)**.
- Le modèle LSTM agit comme un **système d'anticipation (early warning system)**.

Cette redondance fonctionnelle réduit significativement le risque de faux négatifs critiques et améliore la fiabilité globale du système.

Tableau III.8 : Logique combinant état actuel et prédiction

Situation observée	Réaction du système
Valeur actuelle normale ET futur prédit normal	Aucune action — surveillance silencieuse
Valeur actuelle anomalie ET futur prédit anomalie	Action automatique : fermeture vanne + sirène
Valeur actuelle critique (filet de sécurité)	Action automatique immédiate sans attendre l'IA

III.3.3.5 Architecture du modèle

Le modèle implémenté est une architecture séquentielle de deux réseaux de type LSTM conçue pour l'analyse de séries temporelles multiclasse. Il est structuré de manière hiérarchique, permettant une extraction progressive des caractéristiques temporelles, depuis les motifs simples jusqu'aux représentations abstraites de haut niveau.

Le flux de traitement s'effectue de manière ascendante (de l'entrée vers la sortie).

La **couche d'entrée** reçoit une fenêtre temporelle composée de 30 valeurs ADC normalisées. Chaque fenêtre représente une portion locale de l'évolution du signal, permettant au modèle d'exploiter les dépendances temporelles à court et moyen terme.

La **première couche LSTM (64 unités)** constitue le cœur du modèle. Elle est chargée de l'extraction des dynamiques temporelles. Elle apprend à reconnaître des motifs caractéristiques tels que :

- les montées progressives associées à une fuite,

- les pics isolés liés au bruit,
- les oscillations aléatoires sans signification physique.

Une **couche Dropout (taux 0.2)** est ensuite appliquée afin de réduire le surapprentissage. Ce mécanisme consiste à désactiver aléatoirement 20 % des neurones pendant l'entraînement, ce qui force le réseau à apprendre des représentations plus robustes et généralisables.

Une **seconde couche LSTM (32 unités)** permet d'affiner les représentations temporelles extraites par la première couche. Elle agit comme un niveau de compression sémantique, en consolidant les informations pertinentes sur la séquence.

Un second **Dropout (0.2)** est appliqué afin de renforcer la régularisation du modèle.

Deux **couches fully connected (Dense)** assurent ensuite la transformation des représentations internes en espace de décision :

- une couche intermédiaire avec activation ReLU,
- une couche de sortie de 3 neurones.

La **fonction Softmax** est utilisée en sortie afin de transformer les scores en probabilités normalisées. Le modèle fournit ainsi une distribution du type :

- 95 % Normal
- 4 % Anomalie
- 1 % Critique

La classe retenue correspond à la probabilité maximale.

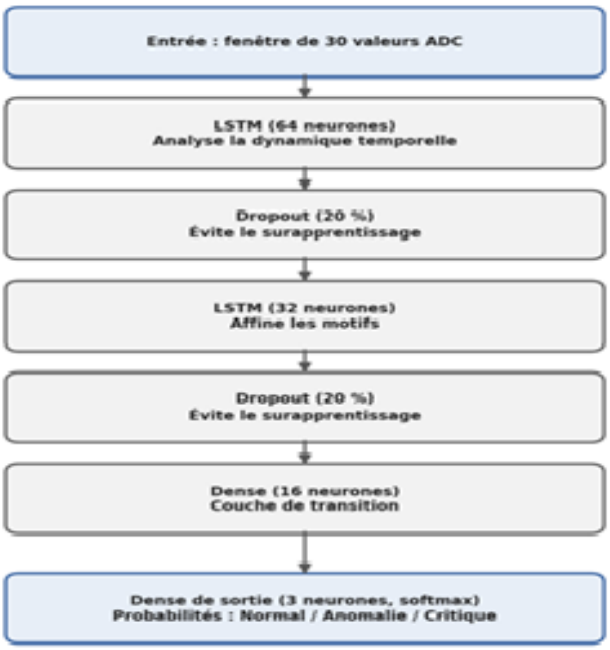


Figure III.3 : Architecture du modèle LSTM, description couche par couche.

III.3.3.6 Paramètres d'entraînement

L'entraînement du modèle consiste à optimiser ses paramètres internes à partir d'un ensemble d'exemples labellisés (fenêtres temporelles + classes associées). L'objectif est de minimiser l'erreur entre les prédictions du modèle et les valeurs réelles.

Tableau III.9 : Paramètres d'entraînement du modèle LSTM

ParamètreversionMémoire	Valeur retenue	Rôle
Optimiseur	Adam [64]	Ajuste dynamiquement le taux d'apprentissage
Fonction de perte	categorical_crossentropy	Mesure l'erreur sur classification multi-classes
Époques	30	Nombre de passes complètes sur le dataset
Batch size	64	Nombre d'échantillons par mise à jour
Pondération des classes	{0: 0.34, 1: 14.7, 2: 45}	Compensation du déséquilibre des classes

III.3.3.7 Analyse des choix

L'optimiseur Adam est largement utilisé en apprentissage profond en raison de sa convergence rapide et de sa robustesse face aux gradients bruités ou non stationnaires. Il combine les avantages des méthodes adaptatives (RMSProp) et des méthodes à moment (SGD avec momentum).

La fonction de perte categorical cross-entropy est adaptée aux problèmes de classification multi-classes où les sorties sont exprimées sous forme de distributions de probabilité.

La pondération des classes constitue un élément critique dans ce contexte. En effet, sans mécanisme de rééquilibrage, le modèle tendrait à privilégier la classe majoritaire (classe normale, ~95 %), ce qui conduirait à une performance trompeuse mais inutile en pratique. L'introduction de poids élevés pour les classes rares (anomalie et critique) force le modèle à accorder une importance équivalente aux situations de danger, malgré leur faible fréquence.

III.4 Approche contextuelle — Cas de l'intrusion

Après avoir traité deux scénarios reposant sur des modèles d'apprentissage profond, cette section introduit un mécanisme décisionnel de nature différente, conçu pour répondre aux contraintes spécifiques de la détection d'intrusion dans un environnement domotique intelligent. Contrairement aux cas précédents fondés sur l'apprentissage à partir des données, ce scénario adopte une logique contextuelle déterministe reposant sur un ensemble explicite de règles de décision.

L'objectif n'est pas de substituer les approches d'intelligence artificielle déjà utilisées dans le système, mais d'employer une méthode mieux adaptée à la nature du phénomène observé ainsi qu'aux exigences de réactivité, de traçabilité et de robustesse associées à la sécurité autonome du domicile.

III.4.1 Pourquoi une logique contextuelle et non un modèle d'apprentissage ?

Le choix d'une approche non fondée sur l'apprentissage automatique découle principalement des caractéristiques du signal exploité ainsi que du mode d'interprétation attendu.

Premièrement, le mécanisme de surveillance ne repose pas sur l'analyse d'un signal continu complexe nécessitant une modélisation prédictive, mais sur l'observation d'une variation spatiale détectée par un capteur. Contrairement au suivi de phénomènes analogiques évolutifs tels que les concentrations de gaz ou certaines variables environnementales, la décision recherchée consiste principalement à déterminer si un changement significatif est observé dans une zone de surveillance donnée.

Deuxièmement, l'incertitude associée au scénario d'intrusion ne provient pas directement de la mesure physique mais du contexte dans lequel celle-ci est interprétée. Une variation détectée à 14 h alors que les occupants sont présents constitue un comportement attendu. En revanche, le même événement détecté durant une plage nocturne ou lorsque le logement est déclaré inoccupé peut être interprété comme une situation anormale nécessitant une alerte.

Ainsi, la prise de décision dépend davantage de règles contextuelles explicites que d'une généralisation statistique issue d'un entraînement.

Troisièmement, l'utilisation d'une logique déterministe améliore l'interprétabilité du système et facilite sa validation fonctionnelle. Chaque décision peut être retracée à partir d'un ensemble identifiable de conditions temporelles et opérationnelles. Cette propriété est

particulièrement pertinente dans les applications de sécurité où la reproductibilité des comportements constitue une exigence importante.

III.4.2 Capteurs et architecture matérielle

Le sous-système de détection d'intrusion repose sur trois composants matériels principaux : un capteur ultrasonique HC-SR04, un module d'horloge temps réel DS1302 ainsi qu'un interrupteur physique permettant de sélectionner le mode d'occupation du logement.

Tableau III.10 : Composants matériels de la détection d'intrusion

Composant	Modèle	Rôle
Capteur de distance	HC-SR04 (ultrason)	Mesure la distance jusqu'au premier obstacle
Horloge temps réel	DS1302	Fournit l'heure fiable, même sans Wi-Fi
Interrupteur	Bouton physique 2 états	Mode présent / absent saisi par l'utilisateur

Le capteur HC-SR04 repose sur un principe de télémétrie ultrasonique. Une impulsion acoustique de fréquence nominale de 40 kHz est émise puis réfléchie par l'objet rencontré. Le temps écoulé entre l'émission et la réception de l'écho permet d'estimer la distance selon :

$$d = (v \times t) / 2$$

où d représente la distance estimée, v la vitesse du son dans l'air et t le temps total aller-retour de l'onde.

Dans cette architecture, le HC-SR04 n'est pas utilisé comme détecteur de présence au sens strict mais comme capteur de variation spatiale. Toute modification significative de la distance mesurée dans une zone supposée stable est interprétée comme un événement susceptible de correspondre à une intrusion.

Le module DS1302 garantit quant à lui la continuité temporelle grâce à son alimentation autonome par pile-bouton, permettant de conserver les informations horaires même en cas de coupure électrique ou d'absence de connectivité.

Note expérimentale : Un capteur PIR (Passive Infrared Sensor) avait initialement été envisagé pour cette fonction. Toutefois, les expérimentations ont montré des déclenchements insuffisamment reproductibles dans les conditions réelles de test, notamment une sensibilité variable selon l'environnement d'utilisation. Après plusieurs cycles de validation, le HC-SR04 a été retenu pour sa stabilité de mesure, sa reproductibilité expérimentale et sa capacité à fournir une information quantitative exploitable dans la logique contextuelle.

III.4.3 Modes de fonctionnement

Afin d'adapter le niveau de surveillance au contexte réel d'occupation, deux modes opérationnels ont été définis.

Tableau III.11 : Modes de surveillance contextuelle

Mode	Activé par	Plage de surveillance	Justification
Présent	Bouton sur PRÉSENT (défaut)	Nocturne (00 h – 06 h)	Le jour, les habitants se déplacent normalement
Absent	Bouton sur ABSENT	Active 24 h / 24	Personne ne devrait être à l'intérieur

Cette distinction introduit une couche de contextualisation permettant au système d'adapter automatiquement sa sensibilité décisionnelle. La décision finale ne dépend donc plus uniquement de la mesure instantanée du capteur mais également du contexte temporel et de l'état d'occupation du domicile.

III.4.4 Logique de décision complète

Le mécanisme de décision repose sur une séquence hiérarchisée de vérifications permettant de combiner les informations spatiales, temporelles et contextuelles avant tout déclenchement d'alerte.

Dans un premier temps, le système lit l'état du bouton de sélection afin d'identifier le mode actif (Présent ou Absent). Ce choix conditionne directement les règles temporelles appliquées au processus de surveillance.

Ensuite, le capteur HC-SR04 effectue des mesures périodiques de distance à intervalles réguliers. Tant que la distance observée demeure supérieure au seuil défini et que la scène reste considérée comme stable, le système poursuit son cycle de surveillance sans générer d'événement.

Lorsqu'une mesure inférieure au seuil est détectée, une seconde étape de validation est exécutée. Le système vérifie alors si la surveillance est effectivement active pour le contexte courant :

- En mode Présent, la détection n'est prise en compte que pendant la plage horaire nocturne définie ;
- En mode Absent, toute détection est considérée comme pertinente indépendamment de l'heure.

Si l'ensemble des conditions contextuelles est satisfait simultanément, le système interprète l'événement comme une intrusion potentielle et déclenche immédiatement les mécanismes d'alerte locaux, notamment l'activation de la sirène.

Cette approche permet de réduire les déclenchements inutiles tout en conservant un comportement décisionnel explicable et reproductible.

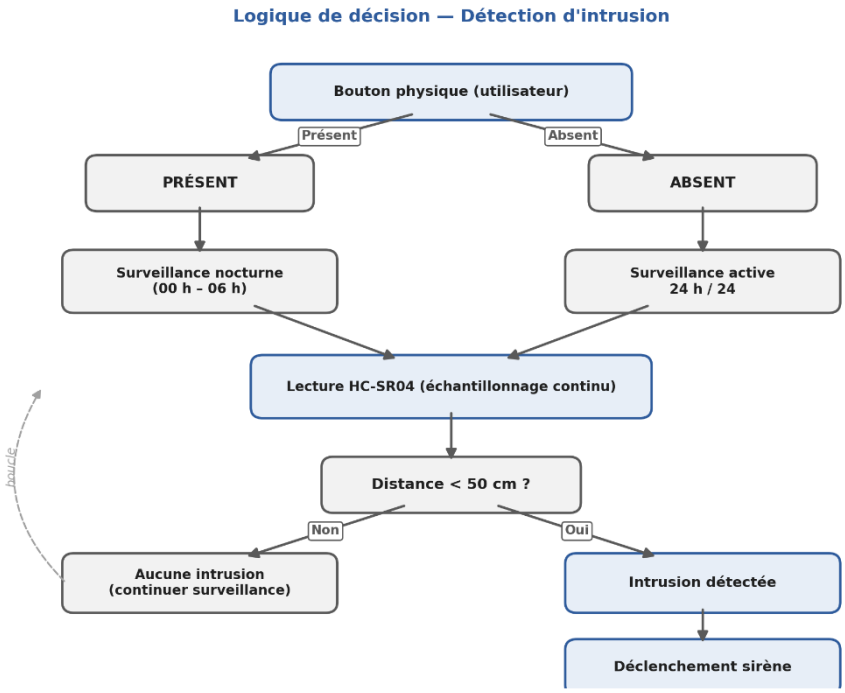


Figure III.4 : Arbre de décision pour la détection d'intrusion

Choix du seuil de distance : le seuil de détection fixé à 50 cm a été déterminé à partir de considérations à la fois géométriques, expérimentales et fonctionnelles.
Le capteur étant installé à proximité immédiate de la zone d'entrée, une intrusion réelle implique nécessairement qu'un objet ou une personne traverse un volume restreint devant le dispositif de mesure. Le seuil doit donc être suffisamment faible pour éviter les perturbations liées aux objets éloignés, tout en restant assez large pour garantir une détection précoce.

Dans la maquette expérimentale utilisée pour la validation (dimensions approximatives de 40 × 20 cm), plusieurs distances ont été observées afin d'identifier un compromis entre couverture spatiale et stabilité de mesure. Le seuil de 50 cm a présenté le meilleur comportement expérimental : il limite l'influence de l'arrière-plan, réduit les fluctuations parasites et assure une couverture complète de la zone de passage.

Il convient toutefois de souligner que cette valeur reste liée à la configuration physique retenue et devra être recalibrée dans le cas d'un déploiement à plus grande échelle ou dans un environnement réel présentant des dimensions différentes.

III.5 Cas particulier (Détection et extinction de flamme)

Le scénario de détection de flamme constitue une troisième stratégie de sécurité, distincte des approches prédictives et contextuelles précédemment présentées. Dans ce cas, la priorité n'est plus l'interprétation avancée des données mais la réduction maximale du temps de réaction.

Pour cette raison, l'architecture retenue repose sur un traitement local exécuté directement sur l'ESP32, sans dépendance fonctionnelle vis-à-vis du réseau, du serveur ou d'un traitement différé.

III.5.1 Pourquoi une action directe ?

Le choix d'un traitement embarqué est motivé par les caractéristiques dynamiques du phénomène surveillé.

Contrairement à certaines anomalies domestiques qui évoluent progressivement et peuvent être analysées dans le temps, une combustion ouverte peut évoluer rapidement et produire des effets critiques en quelques secondes. Dans ce contexte, l'introduction d'étapes supplémentaires de communication réseau, traitement distant ou synchronisation externe risque d'augmenter le temps de réponse.

L'architecture retenue privilégie donc une chaîne décisionnelle courte : détection locale → orientation → activation de l'extinction → alerte.

Il est important de préciser que cette hypothèse est valable dans le cadre du périmètre expérimental considéré, où toute détection de flamme dans la zone surveillée est interprétée comme un événement anormal nécessitant une réaction immédiate.

III.5.2 Capteurs et matériel

Le système anti-incendie repose sur quatre composants fonctionnels assurant successivement la détection, le positionnement du dispositif d'extinction, l'action corrective puis l'alerte locale.

Tableau III.12 : Architecture matérielle du système anti-incendie

Composant	Rôle
Capteur de flamme 5 voies (sortie numérique)	Détection orientée — chaque voie surveille un angle distinct
Servomoteur	Oriente l'extincteur vers la direction de la flamme
Pompe d'eau	Libère le produit d'extinction lorsqu'il est activé
Sirène	Alerte sonore locale immédiate

Le capteur de flamme fonctionne par détection du rayonnement infrarouge généralement associé aux phénomènes de combustion, principalement dans la bande spectrale

proche infrarouge. Lorsque l'intensité mesurée dépasse le seuil interne de déclenchement, la voie concernée passe à l'état actif.

L'utilisation d'un capteur multi-voies permet non seulement de détecter l'existence d'une flamme mais également d'estimer sa direction relative. Cette information est exploitée pour piloter le servomoteur afin d'orienter automatiquement le mécanisme d'extinction vers la zone détectée.

Le système développé constitue un prototype expérimental d'intervention automatique destiné à démontrer la faisabilité d'une réponse autonome localisée.

III.5.3 Disposition physique des voies

Le capteur de flamme utilisé dans ce système est constitué de cinq photodétecteurs infrarouges disposés selon une configuration angulaire permettant une couverture directionnelle de l'environnement surveillé.

Chaque voie est orientée vers une direction spécifique de l'espace, formant ainsi une structure en éventail couvrant un champ de vision global élargi. Cette disposition permet de segmenter l'espace de détection en cinq zones angulaires distinctes, chacune associée à une sortie binaire indépendante.

Concrètement, les capteurs sont agencés de manière à couvrir approximativement un arc frontal, chaque voie correspondant à une direction relative (gauche extrême, gauche, centre, droite, droite extrême). Cette organisation transforme une simple détection infrarouge ponctuelle en une information spatiale exploitable.

Lorsqu'une flamme est détectée, la voie activée indique la direction approximative de la source de rayonnement. Cette information est ensuite utilisée par le système de contrôle pour orienter le servomoteur vers la zone concernée, permettant une intervention ciblée du mécanisme d'extinction.

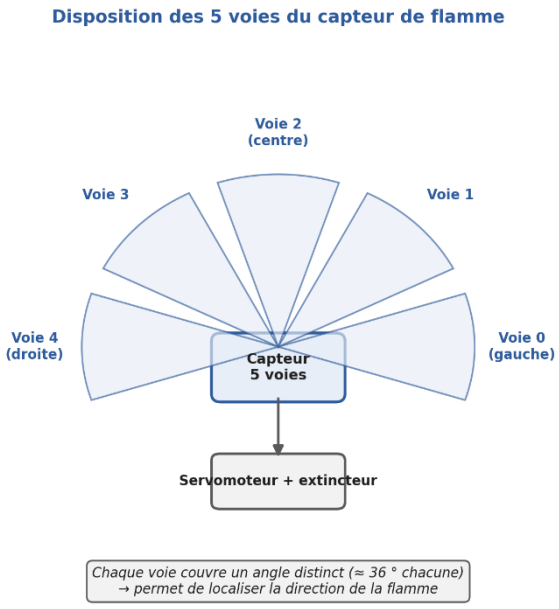


Figure III.5 : Disposition des 5 voies du capteur de flamme

III.5.4 Logique de décision

Le système embarqué basé sur ESP32 interroge en continu les cinq voies du capteur de flamme. Chaque voie fournit un signal binaire indiquant la présence ou l'absence de rayonnement infrarouge dans une direction donnée.

En l'absence d'activation sur l'ensemble des voies, le système reste dans une boucle d'attente active, sans déclenchement d'action.

Dès qu'au moins une voie passe à l'état actif (signal binaire 0/1 issu du seuil interne du capteur), le système identifie immédiatement la voie concernée. Cette information directionnelle est utilisée comme entrée de commande pour le servomoteur, afin d'orienter le dispositif d'extinction vers la zone détectée.

Une fois l'alignement mécanique effectué, deux actions sont déclenchées de manière quasi simultanée :

- l'activation de la pompe d'extinction ;
- le déclenchement de la sirène d'alerte locale.

L'ensemble du traitement repose sur une logique événementielle locale, sans filtrage statistique ni modèle prédictif. Le temps de réponse global dépend principalement du cycle d'échantillonnage de l'ESP32 et des temps mécaniques du servomoteur, et reste de l'ordre de quelques dizaines à quelques centaines de millisecondes selon la configuration matérielle.

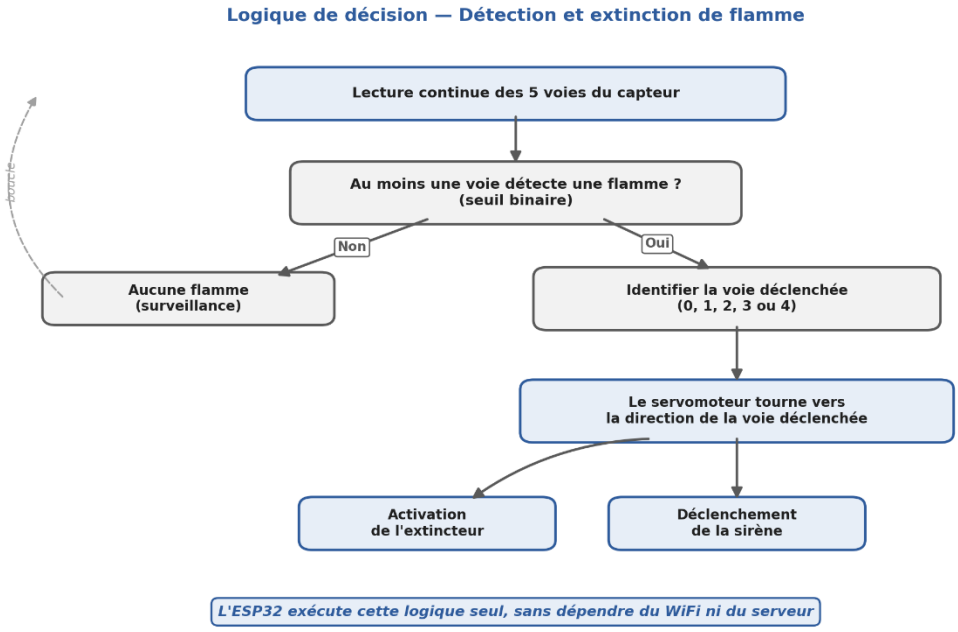


Figure III.6 : Arbre de décision pour la détection et l'extinction de flamme

III.5.5 Avantages de cette approche

Cette architecture présente trois avantages principaux par rapport aux mécanismes précédemment décrits dans les cas d'usage d'intrusion et de détection contextuelle.

Premièrement, la rapidité de réaction : L'absence d'intermédiaires (réseau, serveur ou modèle d'inférence) permet une prise de décision entièrement locale. Le délai entre la détection d'une flamme et le déclenchement de l'action corrective est ainsi réduit au minimum, dépendant uniquement du traitement embarqué et des actionneurs.

Deuxièmement, l'autonomie fonctionnelle : Le système reste opérationnel en cas de coupure de connexion WiFi, de défaillance serveur ou d'indisponibilité de ressources externes. Cette indépendance est essentielle dans un scénario critique où la continuité de service est prioritaire.

Troisièmement, la simplicité décisionnelle : La logique repose sur une comparaison binaire intégrée au capteur de flamme, sans phase d'apprentissage ni interprétation probabiliste. Cette simplicité réduit la complexité logicielle et améliore la prévisibilité du comportement du système.

Il convient toutefois de noter que cette approche privilégie la réactivité et la robustesse opérationnelle au détriment de toute sophistication algorithmique, ce qui la rend particulièrement adaptée aux systèmes embarqués critiques à réponse immédiate.

III.6 Conclusion

Ce chapitre a présenté la démarche méthodologique adoptée pour transformer les données issues des capteurs en décisions intelligentes adaptées aux différents types de risques. Il a mis en évidence le principe fondamental d'adéquation entre la nature des signaux et les stratégies de traitement associées.

Trois approches complémentaires ont été définies. Les phénomènes de fuite de gaz et de surcharge électrique, caractérisés par des signaux continus et évolutifs, sont traités par une approche prédictive basée sur un modèle LSTM, permettant d'anticiper les dérives avant l'atteinte des seuils critiques. La détection d'intrusion repose sur une logique contextuelle déterministe exploitant des règles explicites adaptées aux conditions d'utilisation. Enfin, le risque d'incendie est géré par une approche réactive embarquée sur microcontrôleur, garantissant une réponse immédiate compte tenu des contraintes strictes de latence.

L'ensemble de ces choix traduit une architecture hybride et modulaire, dans laquelle chaque danger est traité selon la méthode la plus adaptée à sa dynamique propre, assurant ainsi robustesse et efficacité globale.

Ce chapitre constitue la base scientifique du système décisionnel, en établissant le lien entre les phénomènes physiques, leur modélisation algorithmique et leur intégration dans une architecture globale. Le chapitre suivant présentera l'implémentation du prototype embarqué, l'intégration du système ainsi que la validation expérimentale des performances en conditions réelles.

CHAPITRE IV :

Implementation du Prototype, Integration et Validation Expérimentale

IV.1 Introduction

Ce chapitre présente la mise en œuvre pratique des travaux de conception et de développement exposés dans les chapitres précédents. Tandis que le chapitre 2 a défini l'architecture globale du système et que le chapitre 3 a décrit les modèles d'intelligence artificielle ainsi que les méthodes algorithmiques retenues, ce chapitre est consacré à la réalisation du prototype, à l'intégration des différents modules matériels et logiciels, ainsi qu'à la validation expérimentale de la solution proposée.

L'objectif est de transformer les concepts théoriques étudiés précédemment en une solution opérationnelle capable d'assurer la collecte des données, leur analyse en temps réel et le déclenchement d'actions adaptées en fonction de l'état du système. Cette phase comprend la sélection et l'intégration des composants matériels, le développement des applications embarquées sur les microcontrôleurs ESP32, la mise en œuvre de l'infrastructure de communication basée sur MQTT, le déploiement du serveur de traitement ainsi que la conception de l'interface de supervision destinée à l'utilisateur.

Dans une démarche orientée vers la reproductibilité, chaque composant est présenté de manière détaillée à travers ses caractéristiques techniques, son rôle dans le système, son schéma de câblage et les extraits de code associés à son fonctionnement. Les choix technologiques retenus sont justifiés au regard des contraintes fonctionnelles et techniques du projet afin de garantir la cohérence de l'ensemble de la solution développée.

À l'issue de ce chapitre, le lecteur disposera d'une vision complète de l'implémentation du système, depuis l'assemblage des composants matériels jusqu'au déploiement des services logiciels. Les expérimentations réalisées permettront également d'évaluer le comportement global du prototype et de vérifier la conformité de son fonctionnement avec les objectifs définis lors de la phase de conception.

IV.2 Architecture matérielle du prototype

IV.2.1 Inventaire du matériel utilisé

Le tableau suivant présente l'ensemble des composants matériels intégrés au prototype développé dans le cadre de ce projet. Pour chaque élément sont indiqués sa référence, sa quantité, sa tension d'alimentation ainsi que sa fonction au sein du système. Cette présentation permet d'avoir une vue globale de l'architecture matérielle mise en œuvre.

Tableau IV.1 : liste complète du matériel utilisé dans le prototype.

Catégorie	Composant	Qté	Alimentation	Fonction
Microcontrôleur	ESP32 DevKit (ESP32-WROOM-32E, 38 broches)	1	5 V USB	Acquisition des données, communication Wi-Fi et échanges MQTT
Capteur	MQ-5	1	5 V	Détection des gaz combustibles (GPL, méthane, butane)
Capteur	HC-SR04	1	5 V	Détection d'intrusion par mesure de distance
Composant passif	Résistances 10 kΩ et 20 kΩ	2	—	Diviseur de tension pour l'adaptation du signal écho du HC-SR04
Capteur	Module de détection de flamme infrarouge 5 voies	1	5 V	Détection et localisation d'un départ de feu
Horloge temps réel	DS1302 avec pile CR2032	1	3,3 V	Fourniture de l'heure et de la date pour le contexte temporel
Actionneur	Servomoteur SG90	1	5 V	Orientation du système de pulvérisation
Actionneur	Module relais 6 voies 5 V	1	5 V	Commande des charges externes
Actionneur	Buzzer actif	1	5 V	Signalisation sonore d'alarme
Actionneur	Ventilateur 12 V (40 mm)	1	12 V externe	Extraction des gaz détectés
Actionneur	Mini-pompe à eau 12 V	1	12 V externe	Système de pulvérisation d'eau
Signalisation	LEDs vertes, orange	3	5 V	Indication visuelle de l'état

	et rouges			du système
Composant passif	Résistances 220 Ω	3	—	Limitation du courant des LEDs
Câblage	Breadboard 830 points	1	—	Réalisation du prototype
Câblage	Fils Dupont (M-M, M-F)	60	—	Interconnexion des composants
Alimentation	Chargeur 12 V / 2 A et câble USB	1	5 V et 12 V	Alimentation du système et des charges externes
Structure	Plaques de plexiglas (PMMA) de 3 mm	12	—	Réalisation de la maquette
Structure	Colle spéciale PMMA et fixateur	2	—	Assemblage des éléments de la maquette

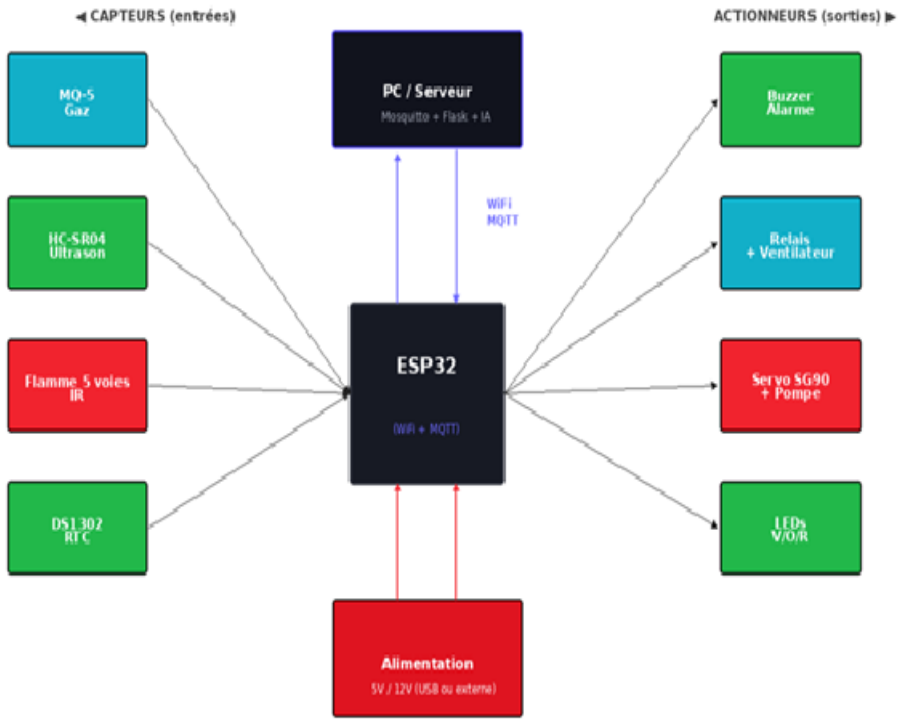


Figure IV.1 : Vue d'ensemble : l'ESP32 est le hub central, relié aux capteurs à gauche, aux actionneurs à droite, au serveur via Wifi/MQTT en haut, et à l'alimentation en bas.

IV.2.2 Conception et fabrication de la maquette

Afin de valider expérimentalement le fonctionnement du système développé dans des conditions proches d'un environnement résidentiel, une maquette physique a été conçue et réalisée. Celle-ci représente une habitation miniature dans laquelle sont intégrés les différents éléments du prototype, notamment les capteurs, les actionneurs, le microcontrôleur ESP32 ainsi que les dispositifs de communication.

Cette maquette constitue un environnement expérimental contrôlé permettant de reproduire les principaux scénarios étudiés dans ce projet, à savoir la détection de fuite de gaz, la détection d'intrusion et la détection d'incendie. Elle facilite également l'observation des interactions entre les différents composants du système et l'évaluation de leur comportement en situation réelle.

Au-delà de son rôle de démonstrateur, la maquette représente un support de validation permettant de vérifier l'intégration matérielle et logicielle de l'ensemble de la solution proposée.

IV.2.2.1 Choix du matériau

Le matériau retenu est le plexiglas (PMMA) d'une épaisseur de 3 mm, choix justifié par plusieurs avantages :

- **Transparence** : permet d'observer directement l'agencement interne des composants, facilitant la validation et la présentation du système.
- **Propriétés mécaniques** : rigidité suffisante pour la stabilité de la structure, tout en étant léger et facile à usiner.
- **Compromis optimal** : par rapport au bois, au carton ou au PVC, il offre un meilleur équilibre entre esthétique, robustesse et visibilité interne, ce qui le rend adapté à un prototype de démonstration.

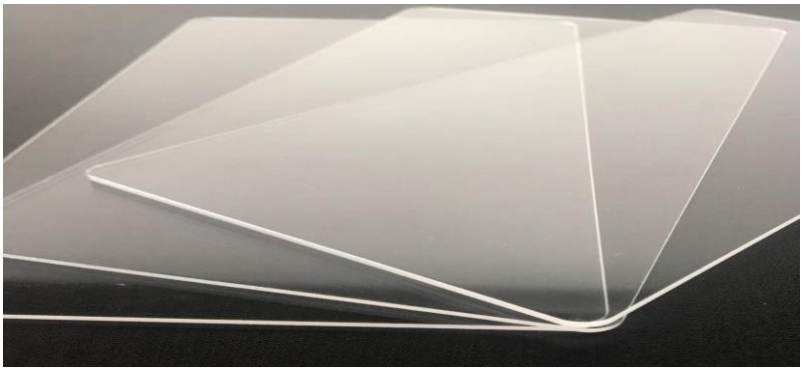


Figure IV.2 : Plaque de plexiglas (PMMA) utilisée pour la fabrication de la maquette.

IV.2.2.2 Géométrie et dimensionnement

La maquette représente une habitation simplifiée à deux niveaux, chacun composé de deux chambres, afin de simuler différents scénarios de détection dans des zones séparées.

Les dimensions globales de la structure sont fixées à 30 cm de largeur, 15 cm de profondeur et 30 cm de hauteur totale. Chaque étage possède ainsi une hauteur de 15 cm. Ce dimensionnement constitue un compromis entre compacité, facilité de manipulation et espace suffisant pour l'intégration des composants électroniques. Ce dimensionnement permet une organisation efficace du câblage et un accès facilité aux composants lors des tests, tout en maintenant une structure stable et représentative d'un environnement domestique réduit.

IV.2.2.3 Plan de découpe

La structure est constituée de douze pièces de plexiglas de 3 mm, réparties entre les deux niveaux selon une architecture modulaire. Chaque étage comprend :

- Une base de support.
- Deux façades (avant et arrière).
- Deux parois latérales
- Une cloison interne séparant les deux chambres.

Cette organisation assure la stabilité de la structure tout en facilitant l'intégration des composants électroniques et l'accès à l'intérieur lors des tests et de la maintenance.

Le plan de découpe (Figure IV.3) associé présente les dimensions précises de chaque pièce ainsi que les ouvertures prévues pour l'intégration des capteurs, des actionneurs et du passage des câbles. La découpe des plaques a été réalisée de manière à garantir un assemblage précis et une bonne rigidité structurelle de l'ensemble.

IV.2.2.4 Fabrication et assemblage de la maquette

La fabrication de la maquette a débuté par la découpe des plaques de plexiglas (PMMA). Les contours des différentes pièces ont été définis à partir du plan de découpe, puis réalisés à l'aide d'un outil de découpe adapté au matériau afin de garantir la précision dimensionnelle et la qualité des bords.

Pour les ouvertures telles que les fenêtres, la porte ainsi que les passages de câbles, des perçages ont été effectués aux angles afin de faciliter l'extraction des zones internes. Les découpes ont ensuite été réalisées progressivement en suivant les tracés définis.

Les passages de câbles ont été positionnés de manière stratégique. Au niveau du rez-de-chaussée, ils sont situés sur la façade arrière afin de limiter leur visibilité. Au premier étage, les ouvertures sont intégrées dans la base de la structure afin d'assurer la continuité du câblage

entre les deux niveaux.



Figure IV.3 : Plan de découpe des panneaux en PMMA et organisation structurelle de la maquette

L'assemblage a été réalisé progressivement. Dans un premier temps, la base a été fixée à la façade principale en vérifiant l'orthogonalité des angles (90°). Les parois latérales ainsi que la cloison intérieure ont ensuite été installées, suivies de la façade arrière. Le même procédé a été appliqué au premier étage.

L'assemblage des différentes pièces a été assuré à l'aide d'une colle spécifique au PMMA à base de dichlorométhane, permettant une soudure chimique des surfaces et garantissant une structure rigide après séchage.

Le toit de la structure a été fixé en dernier afin de permettre un accès facile aux composants internes durant la phase d'intégration et de test du système.

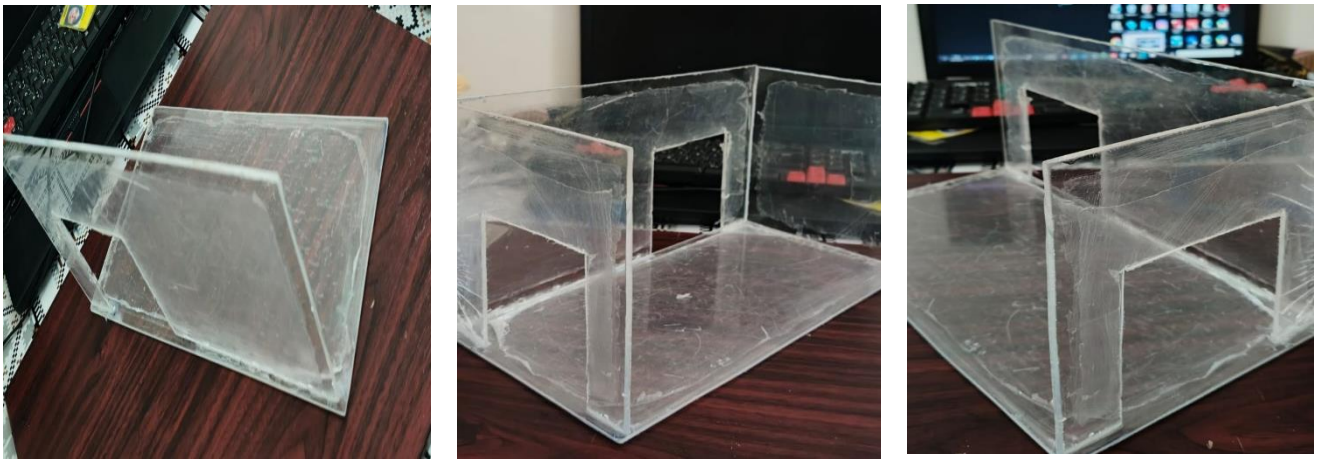


Figure IV.4 : Étapes successives de fabrication de la maquette (découpe, perçage et assemblage)

IV.2.3 Brochage de l'ESP32 et plan de câblage global

La carte utilisée est un module ESP32-WROOM-32E sur une carte de développement à 38 broches. Ce module représente une évolution du modèle ESP32-WROOM-32, offrant une meilleure stabilité et une gestion optimisée de la consommation énergétique, notamment en mode deep sleep, tout en conservant une compatibilité totale avec l'écosystème logiciel ESP32. [1]. Cependant, le choix des broches GPIO a été réalisé en tenant compte des contraintes matérielles et des recommandations de la documentation officielle d'Expressif ainsi que des bonnes pratiques issues de sources spécialisées. [2].

IV.2.3.1 Règles suivies pour l'affectation des broches

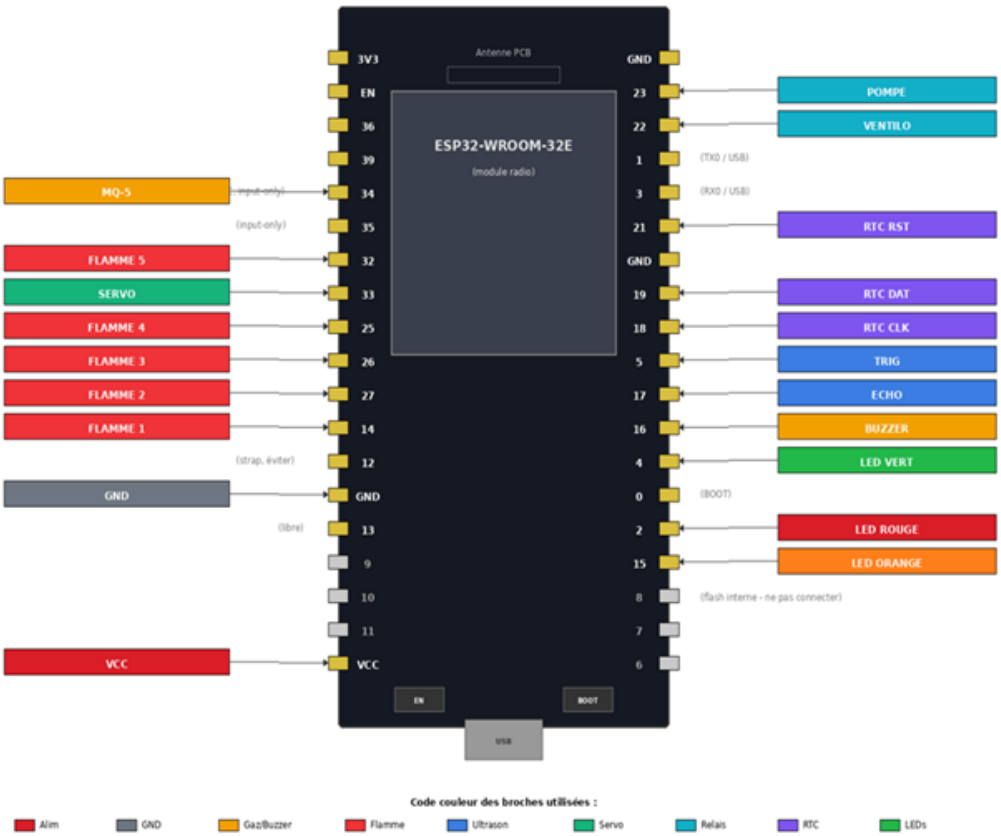
L'ESP32 expose 38 broches physiques, dont seulement 26 GPIO sont librement utilisables. Les contraintes suivantes ont guidé la conception du câblage :

- **GPIO 6–11** : réservées à la mémoire flash, inutilisables.
- **GPIO 34–39** : entrées uniquement, ne peuvent pas piloter des actionneurs.
- **ADC2** : non fiable lorsque le Wi-Fi est actif, les entrées analogiques sont donc exclusivement affectées aux broches ADC1.
- **GPIO 0, 2, 5, 12 et 15** : influencent le mode de démarrage et doivent être utilisés avec précaution.
- **PWM** : les signaux de contrôle du servomoteur doivent être affectés à des GPIO compatibles avec la génération matérielle de PWM.

Ces contraintes ont guidé la répartition des broches afin de garantir la stabilité du système.

Tableau IV.2 : Tableau récapitulatif des affectations

GPIO	Direction	Composant	Type de signal	Note
34	IN	MQ-5 (sortie AO)	Analogique	ADC — fonctionne avec Wifi actif
5	OUT	HC-SR04 TRIG	Numérique	Envoie une impulsion pour déclencher la mesure de distance.
17	IN	HC-SR04 ECHO	Numérique	Reçoit le signal de retour du capteur ultrasons.
14	IN	Flamme voie 1	Numérique	
27	IN	Flamme voie 2	Numérique	
26	IN	Flamme voie 3	Numérique	
25	IN	Flamme voie 4	Numérique	
32	IN	Flamme voie 5	Numérique	ADC mais utilisé en numérique
33	OUT	Servo SG90 (signal)	PWM 50 Hz	Envoie un signal PWM pour contrôler la position du servo.
16	OUT	Buzzer (+)	Numérique	Commande directe
22	OUT	Relais ventilateur	Numérique (inv.)	Branche en NO
23	OUT	Relais pompe	Numérique (inv.)	Branche en NO
4	OUT	LED verte	Numérique + 220 Ω	Mode présent
15	OUT	LED orange	Numérique + 220 Ω	Mode absent
2	OUT	LED rouge	Numérique + 220 Ω	Alarme intrusion
18	OUT	DS1302 CLK	Numérique	Envoie le signal d'horloge vers le module RTC
19	IN/OUT	DS1302 DAT	Numérique	Permet l'échange des données avec le module RTC.
21	OUT	DS1302 RST/CE	Numérique	Active la communication avec le module RTC.



Figures IV.5 : Brochage détaillé de l'ESP32 (38 pins, ESP32-WROOM-32E) avec l'affectation exacte de toutes les broches utilisées dans le projet

IV.2.4 Présentation détaillée des composants et de leur câblage

Cette section présente de manière détaillée les principaux composants utilisés dans le prototype. Pour chaque élément, sont décrits son rôle dans le système, ses caractéristiques techniques essentielles ainsi que son mode d'intégration au niveau de l'ESP32.

IV.2.4.1 Microcontrôleur ESP32-WROOM-32E

L'ESP32-WROOM-32E constitue l'unité centrale du système. Il intègre un microcontrôleur dual-core, des interfaces Wi-Fi et Bluetooth, ainsi qu'un ensemble de broches GPIO permettant l'interfaçage avec les capteurs et actionneurs.

Dans ce projet, il assure plusieurs fonctions essentielles :

- acquisition et traitement des données capteurs.
- exécution de la logique embarquée.
- communication via le protocole MQTT.
- commande des actionneurs.

Le port USB est utilisé pour l'alimentation et la programmation du module lors des phases de développement et de test.



Figures IV.6 : microcontrôleur ESP32-WROOM-32E

IV.2.4.2 Breadboard et fils de connexion (jumpers)

Une breadboard de 830 points a été utilisée pour faciliter le prototypage sans soudure, permettant l'interconnexion rapide des composants et la modification aisée du câblage lors des tests.

Les rails latéraux distribuent les tensions VCC et GND à l'ensemble des modules, tandis que la zone centrale assure le raccordement des composants via des fils Dupont, garantissant une connexion flexible entre l'ESP32, les capteurs et les actionneurs.

Ce mode de prototypage est particulièrement adapté aux phases expérimentales, car il permet des modifications rapides du circuit sans altération permanente du matériel.

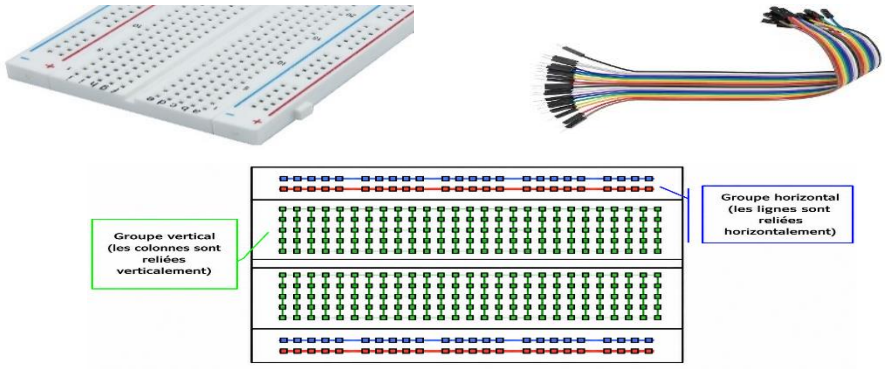


Figure IV.7 : Schéma des connexions internes d'une Breadboard et des fils de liaison (jumpers)

IV.2.4.3 Capteur de gaz MQ-5

Le capteur MQ-5 est un capteur de gaz basé sur une résistance sensible aux composés combustibles (GPL, propane, butane et méthane). Sa résistance varie en fonction de la concentration de gaz détectée, ce qui entraîne une variation de la tension de sortie analogique (AO).

L'ESP32 exploite son convertisseur analogique-numérique (ADC 12 bits), fournissant une valeur comprise entre 0 et 4095, proportionnelle à la concentration de gaz mesurée.

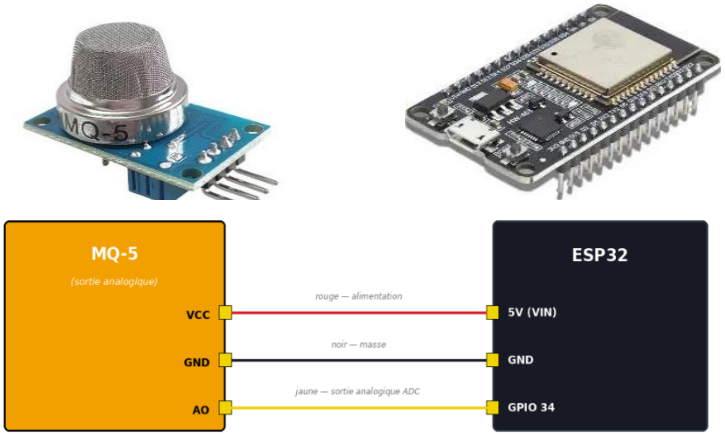


Figure IV.8 : Câblage du capteur MQ-5 vers l'ESP32. Seule la sortie analogique AO est utilisée.

IV.2.4.4 Capteur ultrason HC-SR04

Le HC-SR04 permet la mesure de distance par émission et réception d'ondes ultrasonores. Il dispose de quatre broches : VCC, GND, TRIG et ECHO.

Le principe de fonctionnement est le suivant :

- Une impulsion est envoyée sur la broche TRIG.
- Le capteur émet une onde ultrasonore.
- L'onde est réfléchie par un obstacle.
- La durée de l'écho est mesurée sur la broche ECHO.

La distance est ensuite calculée à partir du temps de propagation de l'onde selon la relation :

$$\boldsymbol{d = \frac{t \times 0{,}0343}{2}}$$

où :

- (d) représente la distance en centimètres.
- (t) le temps de vol en microsecondes.
- 0,0343 correspond à la vitesse du son en cm/µs.

Une forme simplifiée couramment utilisée est donnée par : $d(\text{cm}) = \frac{t}{58}$

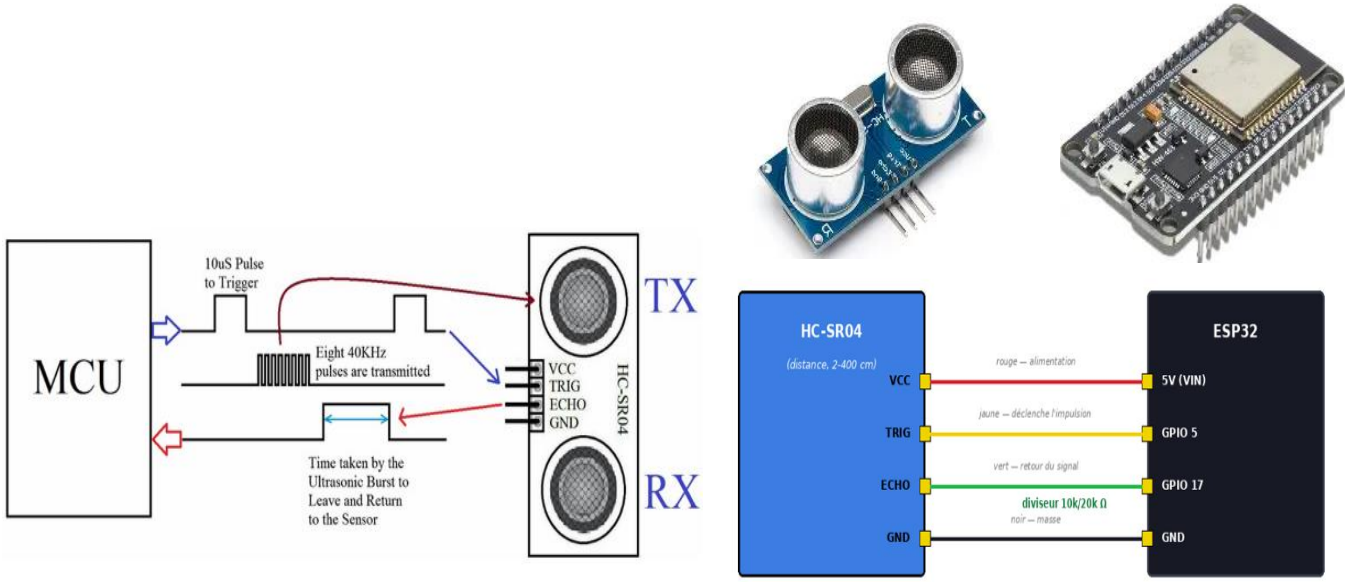


Figure IV.9 : Câblage du HC-SR04. TRIG est une sortie ESP32, ECHO une entrée. Les deux sont en logique 5 V mais l'ESP32 (3.3V) tolère sans problème en lecture

Remarque d'intégration électrique : Le signal ECHO du HC-SR04 est de niveau logique 5 V, tandis que les GPIO de l'ESP32 fonctionnent à 3,3 V. Afin de protéger l'entrée GPIO, un pont diviseur de tension a été mis en place à l'aide de deux résistances (10 kΩ et 20 kΩ), permettant d'adapter le niveau logique à l'entrée de l'ESP32.

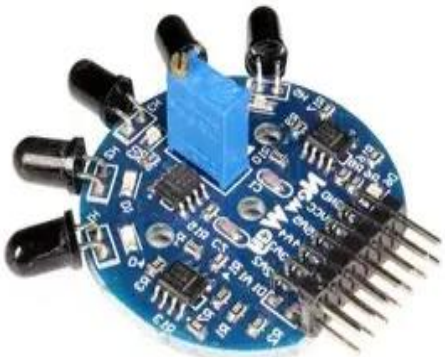
IV.2.4.5 Capteur de flamme infrarouge 5 voies

Le module de détection de flamme est constitué de cinq capteurs infrarouges directionnels, détectant les rayonnements émis par une flamme dans une plage typique de 760 nm à 1100 nm. Chaque voie fournit un signal binaire indiquant la présence ou l'absence de flamme dans sa direction.

Dans le cadre de ce projet, cette information est utilisée pour orienter un servomoteur associé au système d'extinction. Selon la voie détectée, un angle de rotation est appliqué afin de diriger l'action de suppression vers la zone concernée.

Tableau IV.3 : Cartographie voie de détection ↔ angle du servomoteur.

Voie active	Angle servo
Voie 1	155°
Voie 2	130°
Voie 3	90°
Voie 4	50°
Voie 5	20°



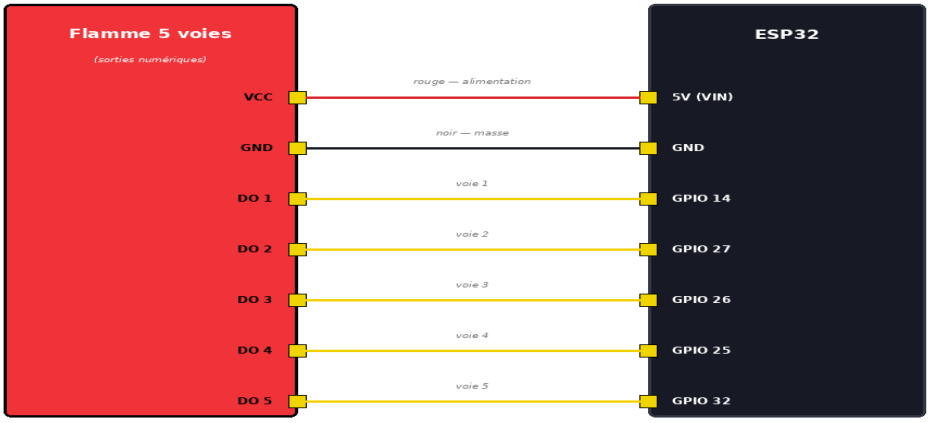


Figure IV.10 : Câblage *du module flamme 5 voies. Une alimentation de 5 V commune et 5 lignes numériques distinctes vers les GPIO d'entrée.*

Ce mécanisme permet une réponse directionnelle adaptée en fonction de la localisation du foyer détecté.

IV.2.4.6 Servomoteur SG90

Le servomoteur SG90 est utilisé dans ce projet pour assurer l'orientation du système d'extinction. Il est mécaniquement couplé à un support permettant de diriger le jet d'eau produit par la pompe vers la zone de détection de flamme.

Le SG90 fonctionne sous une alimentation de 5 V et est commandé par un signal de type PWM (Pulse Width Modulation) généré par l'ESP32. La fréquence du signal est fixée à environ 50 Hz, correspondant à une période d'environ 20 ms.

La position angulaire du servomoteur est déterminée par la largeur de l'impulsion :

- 1 ms correspond approximativement à 0°
- 1,5 ms correspond à 90°
- 2 ms correspondent à 180°

Ainsi, la variation de la largeur d'impulsion permet de contrôler précisément l'orientation du système d'extinction en fonction de la zone détectée.

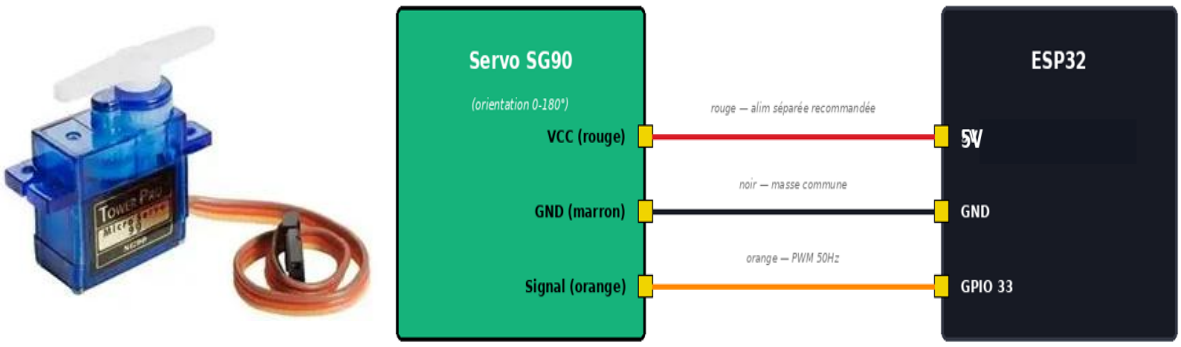


Figure IV.11 : Câblage du servomoteur SG90

IV.2.4.7 Module relais (ventilateur et pompe)

Le module relais à deux canaux est utilisé pour commander des charges alimentées en 12 V, notamment le ventilateur et la pompe à eau. Il joue le rôle d'interface entre la partie basse tension (ESP32) et la partie puissance.

Les sorties de l'ESP32 commandent les entrées IN du module relais. Lorsque l'entrée est activée, le relais établit la connexion entre les bornes COM et NO, permettant ainsi l'alimentation de l'actionneur correspondant.

Les charges sont câblées en mode normalement ouvert (NO), garantissant qu'elles restent hors tension en l'absence de commande. Cette configuration améliore la sécurité du système en cas de dysfonctionnement du microcontrôleur.



Figure IV.12 : Module relais.

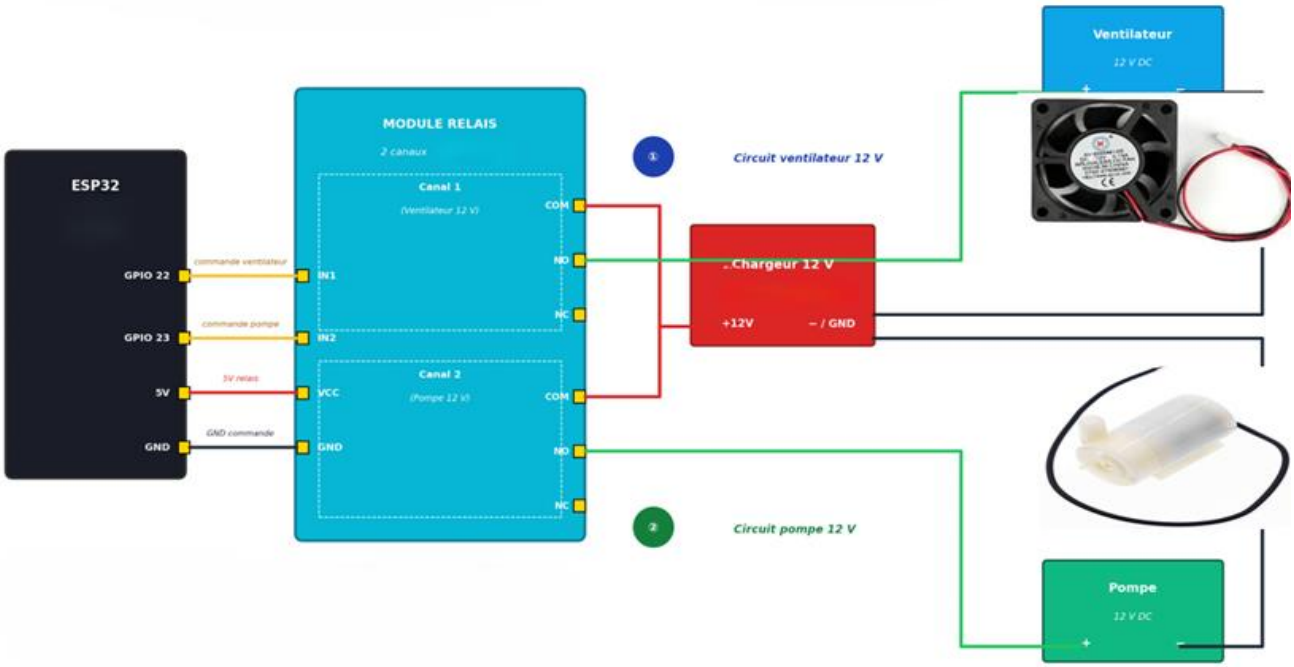


Figure IV.13 : Câblage du module relais. Les bornes IN sont commandées par l'ESP32. Les bornes COM/NO/NC commutent la charge externe

IV.2.4.8 Module horloge RTC DS1302

Le DS1302 est un module d'horloge temps réel (RTC) permettant de conserver l'heure et la date même en l'absence d'alimentation principale, grâce à une batterie de sauvegarde (CR2032).

La communication avec l'ESP32 s'effectue via une interface série synchrone composée de trois signaux :

- CE/RST : activation du module.
- SCLK : signal d'horloge.
- I/O : ligne de données bidirectionnelle.

Au démarrage du système, l'ESP32 interroge le module DS1302 afin de récupérer l'heure courante. Cette information est ensuite utilisée pour adapter la logique de surveillance temporelle, notamment la gestion des périodes d'activation des capteurs d'intrusion.

Le module intègre également une mémoire RAM interne permettant le stockage temporaire de données système.

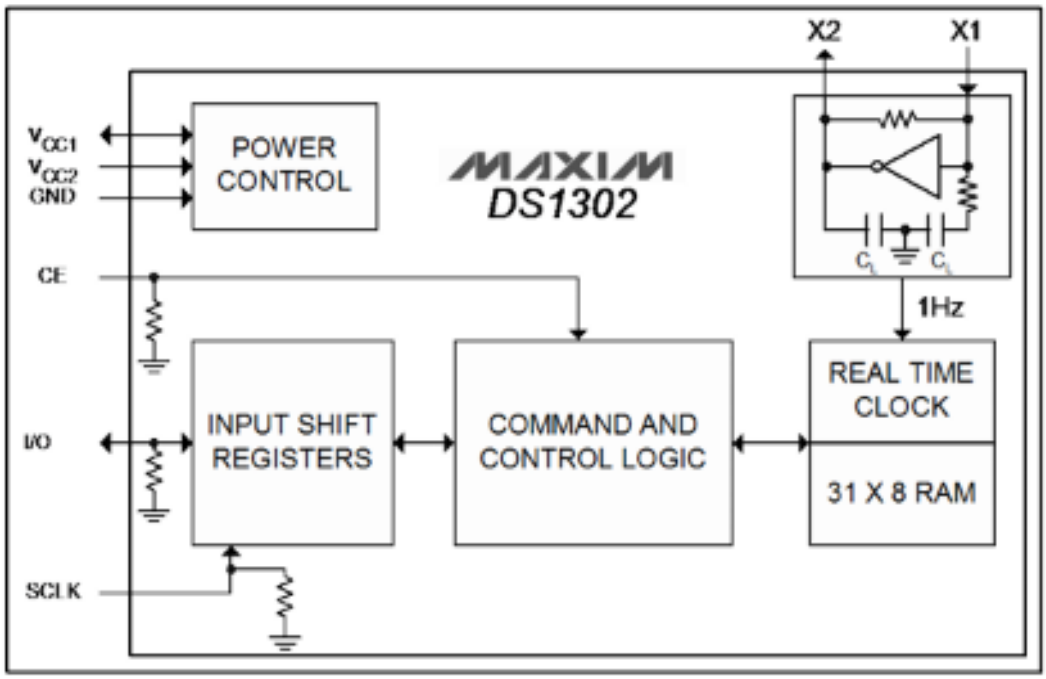


Figure IV.14 : Le DS1302.

Le DS1302 contient une horloge interne qui garde la date et l'heure. Le bloc Power Control gère l'alimentation normale et la pile de secours. Les broches CE/RST pour activer l'échange, SCLK pour synchroniser les données, et I/O pour envoyer ou recevoir l'information. Le module possède aussi une petite mémoire RAM 31×8 pour stocker quelques informations.

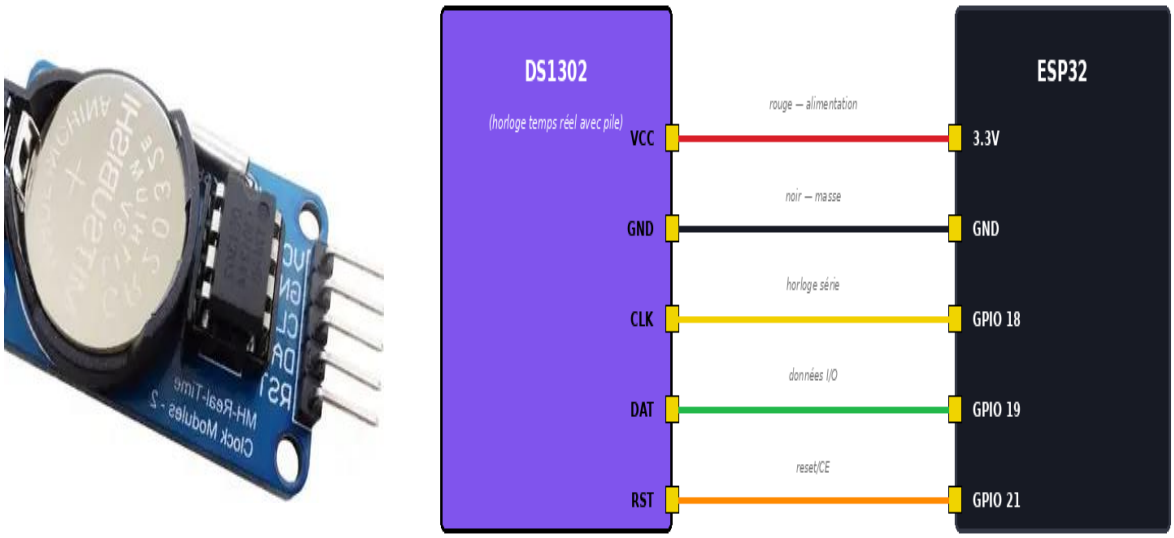


Figure IV.15 : Câblage du DS1302. Trois lignes de communication + alimentation 3,3 V (le module accepte aussi 5 V).

IV.2.4.9 Buzzer actif

Le buzzer utilisé dans ce projet est un buzzer actif intégrant un oscillateur interne. Lorsqu'il est alimenté, il génère automatiquement un signal périodique dans la plage de fréquence audible (environ 2 à 2,5 kHz), ce qui produit une alarme sonore sans nécessiter de génération de signal externe.

L'ESP32 commande le buzzer via une sortie numérique. L'état logique haut active l'alarme sonore, tandis que l'état bas l'arrête. Ce dispositif est utilisé comme signal d'alerte en cas de détection de gaz, de flamme ou d'intrusion.

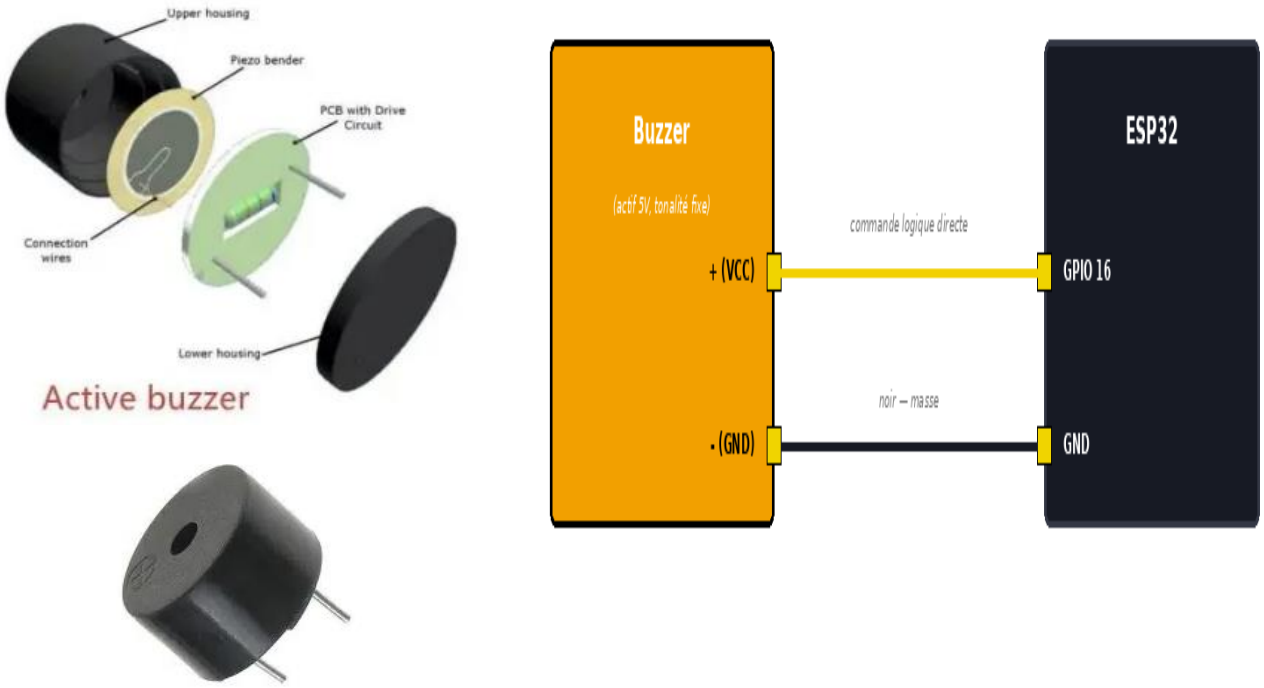


Figure IV.16 : Câblage du buzzer actif. Commande directe par GPIO, masse commune avec l'ESP32

IV.2.4.10 LEDs indicatrices

Le système utilise trois LEDs permettant de visualiser l'état global du dispositif :

- LED verte : fonctionnement normal (mode présence),
- LED orange : mode surveillance réduite (absence),
- LED rouge : état d'alarme.

Chaque LED est protégée par une résistance de 220 Ω afin de limiter le courant et d'éviter tout dommage aux broches GPIO de l'ESP32.

Les LEDs sont commandées directement par les sorties numériques du microcontrôleur.

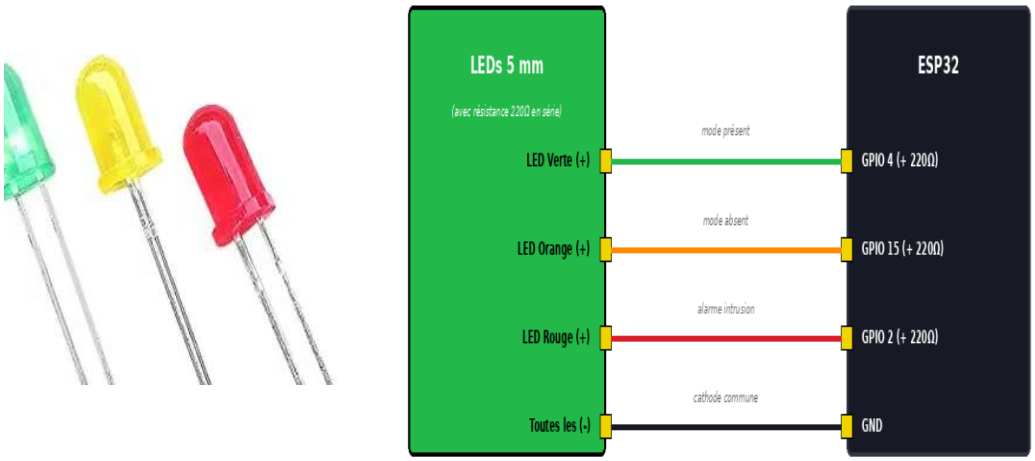


Figure IV.17 : Câblage des LEDs avec l'ESP32

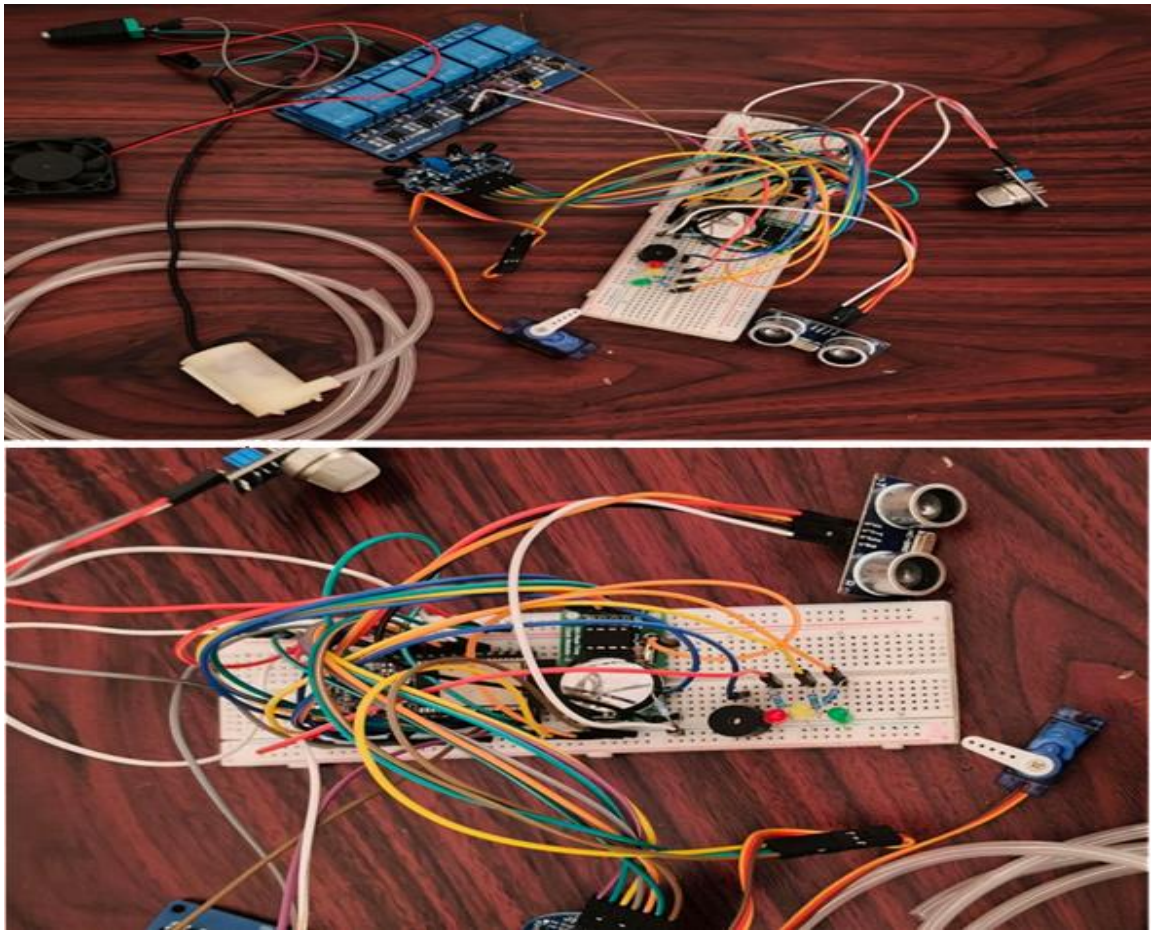


Figure IV.18 : Câblage de tout le matériel

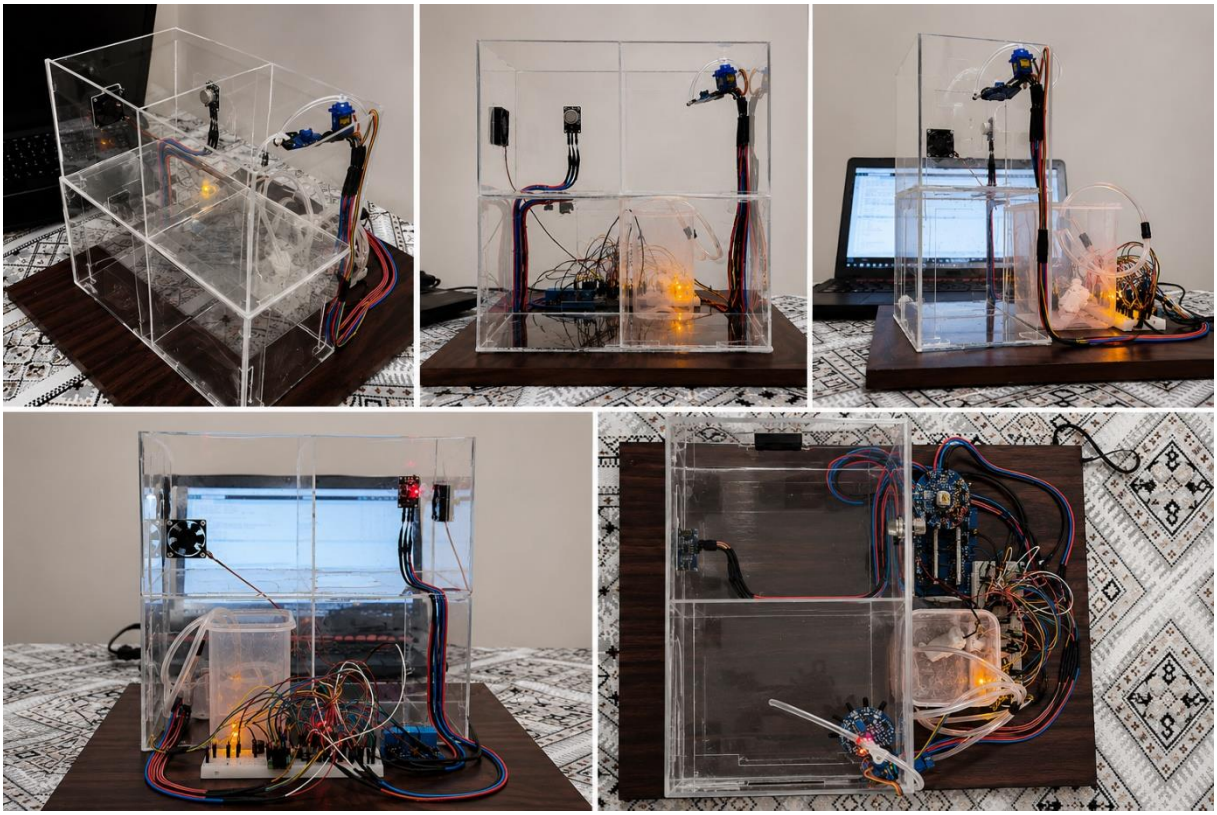


Figure IV.19 : Vue d'ensemble de la maquette en plexiglas (PMMA) après assemblage et intégration des composants

IV.3 Architecture logiciel du prototype

IV.3.1 Firmware embarqué sur l'ESP32

Le Firmware de l'ESP32 centralise l'ensemble des sous-systèmes du prototype, notamment l'acquisition des données issues des capteurs, la commande des actionneurs, la gestion des modes de fonctionnement ainsi que la communication via le protocole MQTT. Il est structuré en modules fonctionnels distincts afin de garantir une meilleure lisibilité du code, de faciliter le débogage et d'améliorer la maintenabilité globale du système embarqué.

Tableau IV.4 Organisation du Firmware ESP32

Bloc du code	Rôle
Configuration	WiFi, adresse du broker MQTT, topics utilisés
Pins	Broches des capteurs, LEDs, relais, buzzer, servo et RTC
Helpers	Fonctions simples : bip (), mesurerDistance (), publier ()
Modes	Présent, absent et reset de l'alarme
Traitements	Gaz, intrusion et flamme
Setup () / loop()	Initialisation puis exécution continue du système

IV.3.2 MQTT

Le protocole MQTT assure la transmission des données entre l'ESP32, le serveur et le tableau de bord (Dashboard). Le broker Mosquito est configuré pour écouter sur le port 1883 et joue le rôle d'intermédiaire de communication entre les différents composants du système.

Dans ce flux, l'ESP32 publie les mesures issues des capteurs ainsi que les événements système sur des topics dédiés. Le serveur s'abonne à ces topics afin de recevoir et de traiter les données en temps réel. Le Dashboard, quant à lui, interagit avec le serveur pour récupérer et afficher l'état du système de manière synchronisée.

Les commandes de contrôle suivent le chemin inverse : elles sont émises depuis le Dashboard, relayées par le serveur, puis transmises à l'ESP32 via le broker MQTT pour exécution.

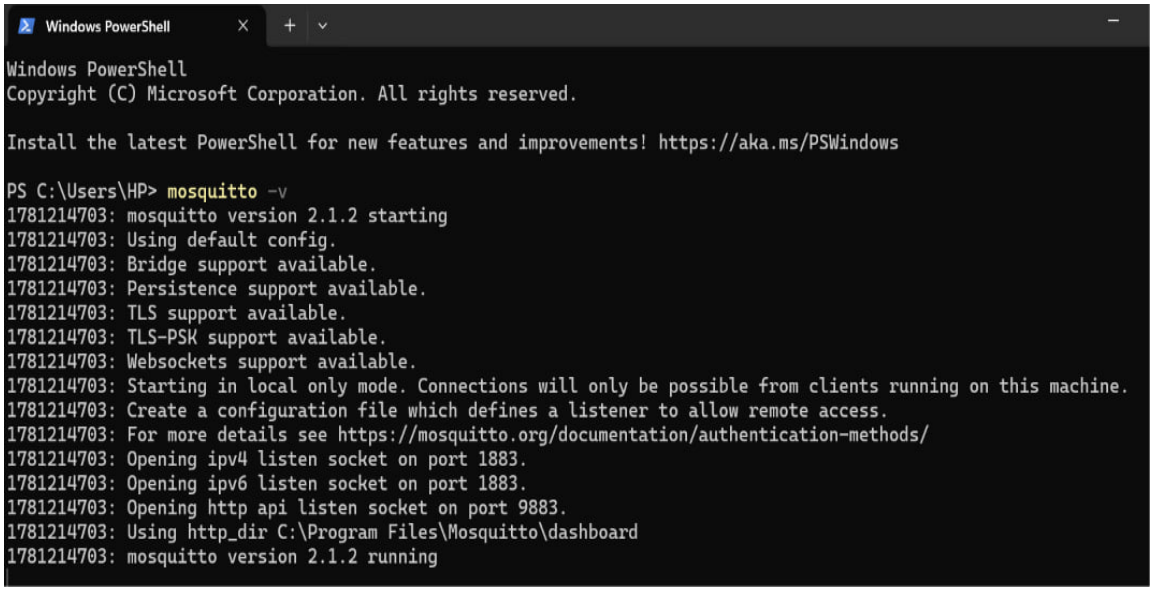


Figure IV.20: Lancement de Mosquito

Les échanges avec le serveur utilisent cinq topics MQTT :

```
const char* TOPIC_GAZ       = "maison/gaz/valeur";
const char* TOPIC_INTRUSION = "maison/intrusion/état";
const char* TOPIC_FLAMME    = "maison/flamme/état";
const char* TOPIC_STATUT    = "maison/statut/général";
const char* TOPIC_COMMANDES = "maison/commandes";
```

IV.3.3 Lecture du MQ-5 et sécurité locale

Le capteur MQ-5 est lu via la fonction analogRead (), fournissant une valeur analogique représentative de la concentration de gaz détectée. À intervalles réguliers de cinq secondes, l'ESP32 transmet cette valeur au serveur sous forme de message JSON afin d'assurer un suivi centralisé et une historisation des mesures.

Parallèlement à cette transmission, une logique de sécurité locale est implémentée directement sur l'ESP32. Celle-ci permet de déclencher une action immédiate lorsque la valeur mesurée dépasse un seuil critique prédéfini, sans attendre la validation ou le traitement côté serveur. Cette architecture hybride garantit à la fois la supervision distante et une réactivité en temps réel en cas de situation dangereuse.

```
[GAZ] 534 -> publie
[GAZ] 687 -> publie
[GAZ] 823 -> publie
[GAZ] 976 -> publie
[GAZ] 1134 -> publie
[GAZ] 1389 -> publie
[GAZ] 1567 -> publie
[GAZ] SEUIL CRITIQUE -> ventilo ON
[GAZ] 1723 -> publie
[GAZ] 1845 -> publie
MQTT recu [maison/commandes] : {"action":"alerte_gaz"}
>> ACTION SERVEUR : alerte gaz
[GAZ] 1654 -> publie
[GAZ] 1423 -> publie
[GAZ] 1198 -> publie
[GAZ] retour normal -> ventilo OFF
[GAZ] 876 -> publie
```

Figure IV.21 : Détection du gaz sur le serial moniteur d'Arduino IDE

IV.3.4 Détection d'intrusion avec HC-SR04

Le capteur ultrasonique est utilisé pour la mesure de distance. En l'absence de réception d'un écho valide, la fonction retourne une valeur par défaut de 400 cm afin d'éviter les comportements indéterminés et de limiter les fausses alertes liées aux erreurs de mesure.

Le comportement du système de surveillance dépend du mode de fonctionnement ainsi que du contexte temporel. Deux règles principales sont appliquées : en mode *Absent*, la surveillance est active en continu sur une plage de 24 heures ; en mode *Présent*, elle est restreinte à une fenêtre nocturne définie entre 00 h et 06 h, période durant laquelle les déplacements sont supposés être minimaux et plus susceptibles de correspondre à des anomalies.

```
[GAZ] 165 -> publie
[GAZ] 170 -> publie
[GAZ] 168 -> publie
MQTT recu [maison/commandes] : {"action":"absent"}
>> MODE : ABSENT
[GAZ] 172 -> publie
[GAZ] 169 -> publie
[GAZ] 174 -> publie
[INTRUSION] detectee !
[GAZ] 167 -> publie
[GAZ] 170 -> publie
MQTT recu [maison/commandes] : {"action":"reset"}
>> Alarme reset
[GAZ] 165 -> publie
```

Figure IV.22 : Détection d'intrusion sur le serial moniteur d'Arduino IDE

IV.3.5 Détection de flamme et orientation du servo

Dès qu'une voie du capteur de flamme est activée, le système déclenche immédiatement une réponse locale. Le servomoteur s'oriente alors vers la direction correspondant à la voie détectée, la pompe d'extinction est activée et le buzzer est déclenché afin de générer une alerte sonore.

L'ensemble du processus est exécuté entièrement en local sur l'ESP32, sans dépendance au réseau ni à une infrastructure externe, afin de garantir un temps de réponse minimal et une continuité de fonctionnement en situation critique.

```
=== Smart Home ESP32 ===
Connexion WiFi..... OK !
IP : 192.168.1.15
Connexion MQTT... OK !
>> MODE : PRESENT
=== Systeme pret ===

[GAZ] 164 -> publie
[GAZ] 168 -> publie
[GAZ] 170 -> publie
Flame3 Detected!
[GAZ] 172 -> publie
[GAZ] 165 -> publie
Flame3 Detected!
Flame2 Detected!
[GAZ] 169 -> publie
[GAZ] 174 -> publie
[GAZ] 168 -> publie
[GAZ] 170 -> publie
```

Figure IV.23 : Détection de flamme sur serial monitor d'Arduino IDE

IV.3.6 Réception des commandes du Dashboard

L'ESP32 est configuré pour écouter le topic MQTT maison/commandes. Le tableau de bord (Dashboard) y publie des instructions sous forme de messages JSON. À la réception d'une

commande, l'ESP32 analyse le contenu du message et exécute la fonction correspondante en fonction de l'action spécifiée.

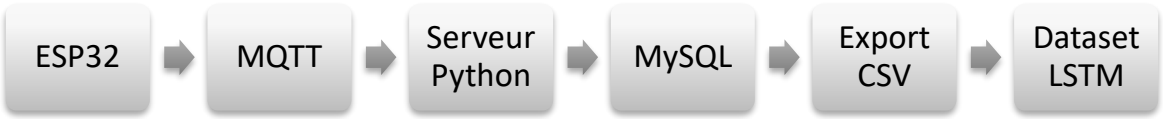
IV.3.7 Setup () et Loop ()

La fonction setup () assure l'initialisation de l'ensemble des périphériques et des modules du système embarqué : configuration des broches d'E/S, initialisation du servomoteur, du module RTC, ainsi que des connexions Wifi et MQTT. Par mesure de sécurité, les relais sont initialisés à l'état *OFF* au démarrage afin d'éviter toute activation non intentionnelle des actionneurs.

La fonction loop() constitue le cœur du cycle d'exécution. Elle maintient la connexion MQTT active et exécute de manière séquentielle les différents traitements du système, notamment l'acquisition des données des capteurs, l'évaluation des règles de décision et la gestion des actions associées.

IV.4 Base de données MySQL et préparation du dataset

Cette section décrit le processus de collecte, de stockage et de préparation des données issues du capteur MQ-5 en vue de leur utilisation pour l'entraînement du modèle LSTM. L'ensemble du pipeline suit une chaîne de traitement structurée allant de l'acquisition en temps réel jusqu'à la constitution d'un dataset exploitable en apprentissage automatique :



IV.4.1 Création de la base MySQL

Une base de données MySQL nommée maison intelligente est mise en place afin d'assurer le stockage structuré des données de capteurs. Une table dédiée, capteurs, est utilisée pour enregistrer les mesures issues du système embarqué, en particulier les valeurs analogiques du MQ-5 ainsi que, le cas échéant, les métadonnées temporelles associées.

IV.4.2 Collecte des mesures par MQTT

Le serveur Python agit comme intermédiaire entre le système embarqué et la base de données. Il est abonné au topic MQTT dédié au capteur de gaz.

À chaque message reçu, la valeur transmise par l'ESP32 est extraite puis insérée en temps réel dans la base MySQL. Ce mécanisme permet une acquisition continue des données, assurant la synchronisation entre la mesure physique et son enregistrement numérique.

IV.4.3 Export MySQL vers CSV

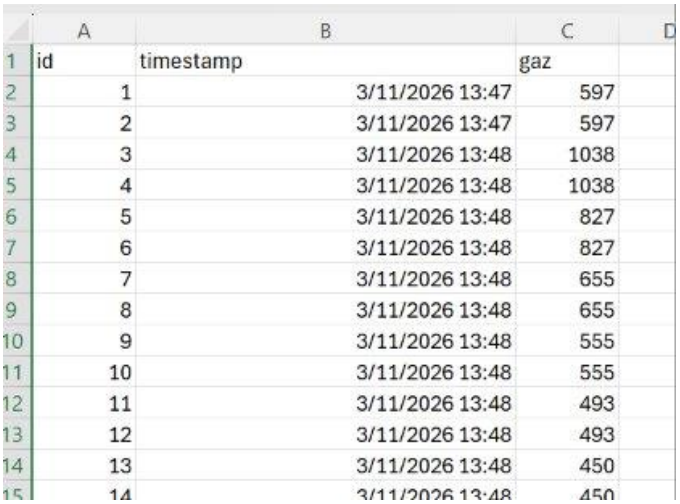
Une fois la phase de collecte terminée, les données stockées dans MySQL sont exportées vers un fichier au format CSV (DB_MQ5.csv). Ce fichier constitue le dataset final utilisé pour les étapes d'entraînement et de validation du modèle LSTM.

Cette étape assure une séparation claire entre la phase d'ingestion des données et la phase d'apprentissage automatique.

IV.4.4 Structure du dataset obtenu

Le dataset final (DB_MQ5.csv) contient un total de **24 667 enregistrements**, correspondant aux mesures successives du capteur MQ-5 sur la période d'acquisition.

Chaque ligne du fichier représente une observation temporelle associée à la valeur du capteur, permettant la construction de séquences temporelles nécessaires à l'apprentissage du modèle LSTM.



	A	B	C	D
1	id	timestamp	gaz	
2	1	3/11/2026 13:47	597	
3	2	3/11/2026 13:47	597	
4	3	3/11/2026 13:48	1038	
5	4	3/11/2026 13:48	1038	
6	5	3/11/2026 13:48	827	
7	6	3/11/2026 13:48	827	
8	7	3/11/2026 13:48	655	
9	8	3/11/2026 13:48	655	
10	9	3/11/2026 13:48	555	
11	10	3/11/2026 13:48	555	
12	11	3/11/2026 13:48	493	
13	12	3/11/2026 13:48	493	
14	13	3/11/2026 13:48	450	
15	14	3/11/2026 13:48	450	

Figure IV.24 : Aperçu du fichier DB_MQ5.csv dans Excel (24 667 mesures)

IV.5 Entraînement du modèle LSTM

Cette section décrit l'ensemble du processus d'entraînement du modèle LSTM utilisé pour l'analyse des séries temporelles issues du capteur MQ-5. Le pipeline couvre la labellisation des données, leur transformation en séquences temporelles, ainsi que la conception et l'apprentissage du réseau de neurones récurrent.

IV.5.1 Chargement et labellisation des données

Les données sont chargées à partir du fichier CSV généré lors de la phase de collecte. Une étape de labellisation est ensuite appliquée afin d'associer chaque observation à l'une des trois classes définies au chapitre III, sur la base des seuils de concentration du gaz.

Cette transformation permet de convertir un problème de régression (valeurs continues du capteur) en un problème de classification supervisée à trois classes.

IV.5.2 Normalisation et fenêtres temporelles

Les valeurs ADC (0–4095) sont normalisées entre 0 et 1 avec MinMaxScaler. Le dataset est ensuite découpé en fenêtres de 30 valeurs consécutives, chacune associée à la classe prédite 2 pas plus tard.

IV.5.3 Séparation entraînement / test

Le jeu de données est partitionné de manière chronologique afin de préserver la structure temporelle des observations. La séparation est effectuée selon un ratio de 80 % pour l'entraînement et 20 % pour le test.

Contrairement à un découpage aléatoire, cette approche évite toute fuite d'information temporelle (*data leakage*) et garantit une évaluation plus réaliste des performances du modèle en conditions séquentielles.

IV.5.4 Architecture et entraînement du modèle

Le modèle LSTM est composé d'une architecture empilée comprenant deux couches LSTM successives comportant respectivement 64 et 32 unités. Ces couches sont suivies de couches de régularisation *Dropout* afin de limiter le surapprentissage.

La sortie du réseau est ensuite traitée par deux couches entièrement connectées (*Dense*), dont la dernière utilise une fonction d'activation *softmax*. Cette dernière produit un vecteur de probabilités associé aux trois classes définies, permettant une classification multi-classes du signal temporel.

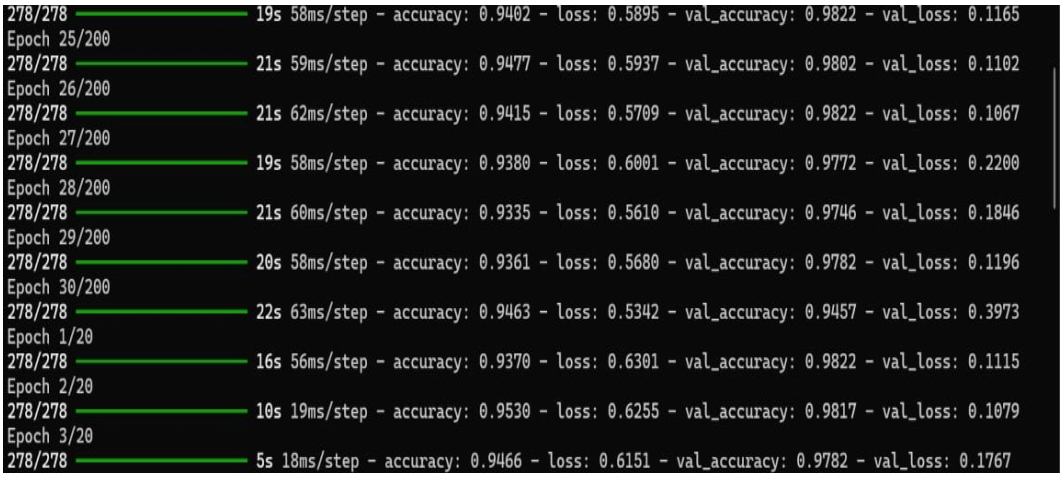


Figure IV.25 : Déroulement de l'entraînement du modèle LSTM (30 epochs + 20 epochs de fine-tuning)

```
278/278 ━━━━━━━━━━━━━━━━━━━━ 7s 25ms/step - accuracy: 0.9387 - loss: 0.5455 - val_accuracy: 0.9802 - val_loss: 0.1091
Epoch 17/20
278/278 ━━━━━━━━━━━━━━━━━━━━ 7s 25ms/step - accuracy: 0.9434 - loss: 0.5383 - val_accuracy: 0.9802 - val_loss: 0.0924
Epoch 18/20
278/278 ━━━━━━━━━━━━━━━━━━━━ 9s 31ms/step - accuracy: 0.9459 - loss: 0.5315 - val_accuracy: 0.9726 - val_loss: 0.3588
Epoch 19/20
278/278 ━━━━━━━━━━━━━━━━━━━━ 8s 22ms/step - accuracy: 0.9232 - loss: 0.5569 - val_accuracy: 0.9822 - val_loss: 0.0895
Epoch 20/20
278/278 ━━━━━━━━━━━━━━━━━━━━ 6s 23ms/step - accuracy: 0.9530 - loss: 0.5276 - val_accuracy: 0.9756 - val_loss: 0.1141
154/154 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step
              precision    recall  f1-score   support

      Normal       1.00      0.95      0.97      4397
    Anomalie       0.76      0.71      0.73       389
    Critique       0.30      0.79      0.44       141

    accuracy                           0.93      4927
   macro avg       0.69      0.82      0.71      4927
weighted avg       0.96      0.93      0.94      4927
```

Figure IV.26 : Résultats d'évaluation du modèle LSTM sur l'ensemble de test (précision globale : 93 %)

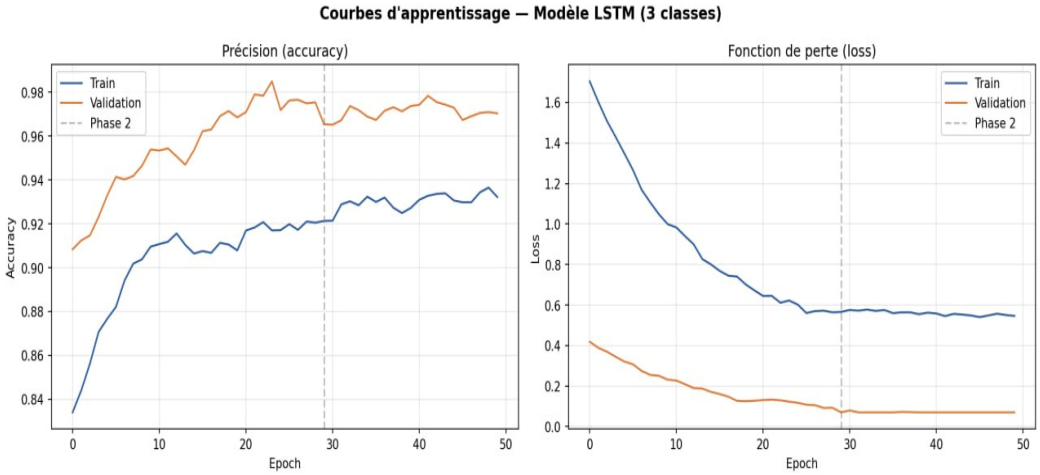


Figure IV.27 : Courbes d'apprentissage du modèle LSTM

La précision augmente progressivement au cours de l'entraînement avant de se stabiliser autour de 97 %, ce qui indique que le modèle parvient à apprendre les caractéristiques discriminantes des données. Toutefois, cette seule métrique ne suffit pas à conclure à l'absence de surapprentissage, qui doit également être évaluée à partir de la comparaison entre les courbes d'entraînement et de validation ainsi que d'indicateurs complémentaires.

Par ailleurs, l'observation d'une perte d'entraînement supérieure à la perte de validation n'est pas, en soi, un comportement attendu systématique et ne peut pas être directement attribuée au poids des classes rares. Ce phénomène peut plutôt s'expliquer par des mécanismes tels que la régularisation (Dropout), la nature du calcul de la fonction de coût, ou encore la variabilité des lots (*mini-batches*) durant l'apprentissage. Une analyse plus fine des métriques par classe serait nécessaire pour confirmer l'impact des classes déséquilibrées.

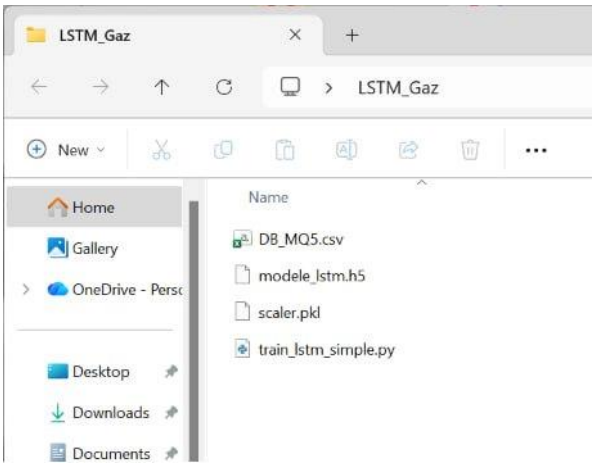


Figure IV.28 : Fichiers générés après l'entraînement : modèle, scaler

IV.6 Serveur , Flask, MQTT et intégration de l'IA

Après l'entraînement, le serveur joue un rôle d'orchestrateur entre l'ESP32, le modèle d'intelligence artificielle et le dashboard. Il reçoit les messages MQTT, maintient une fenêtre glissante contenant les 30 dernières valeurs du capteur de gaz, applique le modèle LSTM préalablement entraîné, puis renvoie l'état du système à l'interface web.

Le serveur n'effectue aucune phase de réentraînement en temps réel : il se limite au chargement du modèle sauvegardé (modele_lstm.h5) ainsi que du scaler (scaler.pkl), utilisés pour la normalisation et l'inférence. Cette architecture permet de séparer clairement la phase d'apprentissage hors ligne et la phase d'exécution en production, garantissant ainsi stabilité et reproductibilité des prédictions.

IV.6.1 Configuration et chargement du modèle

Les paramètres du système sont configurés puis le modèle LSTM et le scaler sont chargés. Une gestion d'exception est mise en place afin de traiter les éventuelles erreurs de chargement.

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                    Python

PS C:\Users\HP> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe c:/Users/HP/Desktop/SmartHome/server.py
I0000 00:00:1781215368.851911   24072 port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating
und-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
[LSTM] Chargement du modele...
I0000 00:00:1781215371.243044   24072 cpu_feature_guard.cc:227] This TensorFlow binary is optimized to use available CPU instructions in performance
 operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compile
[LSTM] Modele charge avec succes
================================================
  SMART HOME - Serveur etape 4
================================================
[OK] Thread MQTT lance
[OK] Dashboard : http://localhost:5000
c:\Users\HP\Desktop\SmartHome\server.py:180: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  mqtt_client = mqtt.Client()

 * Serving Flask app 'server'
 * Debug mode: off
[MQTT] Connecte au broker
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.1.10:5000
Press CTRL+C to quit
```

Figure IV.29 : Démarrage du serveur Flask : chargement du modèle LSTM et connexion MQTT

IV.6.2 Buffer gaz et prédiction en temps réel et traitement des messages MQTT

Un buffer circulaire de 30 valeurs est maintenu en mémoire. Dès qu'il est plein, la prédiction LSTM est exécutée. À chaque message gaz reçu, la valeur est ajoutée au buffer et la prédiction est déclenchée. Pour l'intrusion et la flamme, les états sont directement mis à jour.

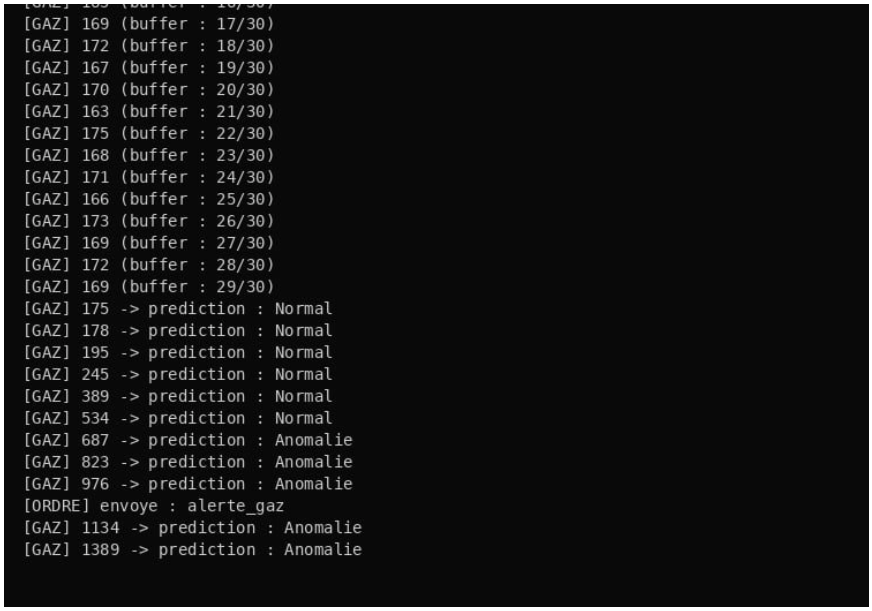


Figure IV.30 : Réception des mesures et prédiction LSTM en temps réel (transition Normal → Anomalie)

IV.6.3 Routes Flask

Flask expose deux routes principales : /api/état renvoie l'état complet du système en JSON, et /api/commande reçoit les actions de l'utilisateur (présent, absent, reset) et les transmet à l'ESP32 via MQTT.

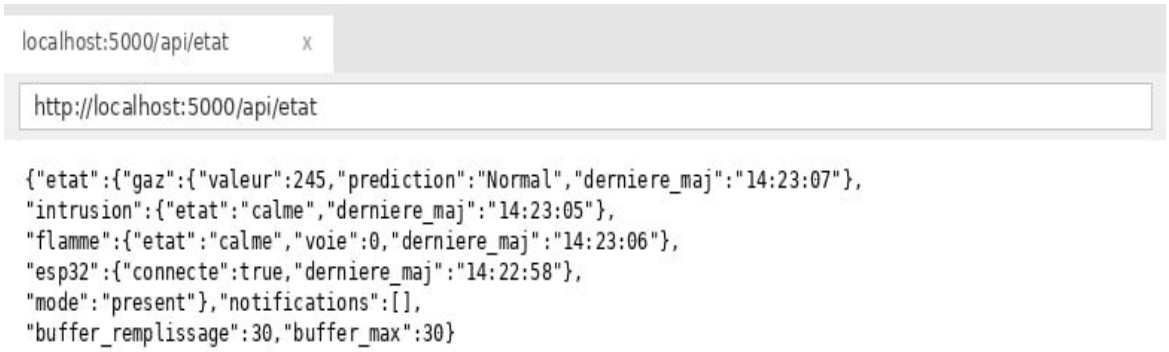


Figure IV.31 : Réponse JSON de l'Endpoint /api/état dans le navigateur

IV.6.4 Architecture thread du serveur

Au lancement, le client MQTT démarre dans un thread séparé. Flask sert le Dashboard pendant que MQTT reste en écoute en arrière-plan.

IV.7 Dashboard web de supervision

Le Dashboard est l'interface permettant de suivre l'état de la maquette et d'envoyer les commandes principales. La chaîne de communication est bidirectionnelle :

ESP32 → MQTT → Serveur → Dashboard / Dashboard → Serveur → MQTT → ESP32

IV.7.1 Principe de fonctionnement

Le navigateur ouvre dashboard.html. JavaScript interroge /api/état toutes les 2 secondes via fetch(). Flask renvoie les données en JSON. La fonction applyState () met à jour les cartes (gaz, intrusion, flamme, mode). Si un danger est détecté, la carte concernée passe en état d'alerte. Les boutons Présent, Absent et Reset envoient une commande à /api/commande.

IV.7.2 Affichage des résultats

1/ Cas de gaz

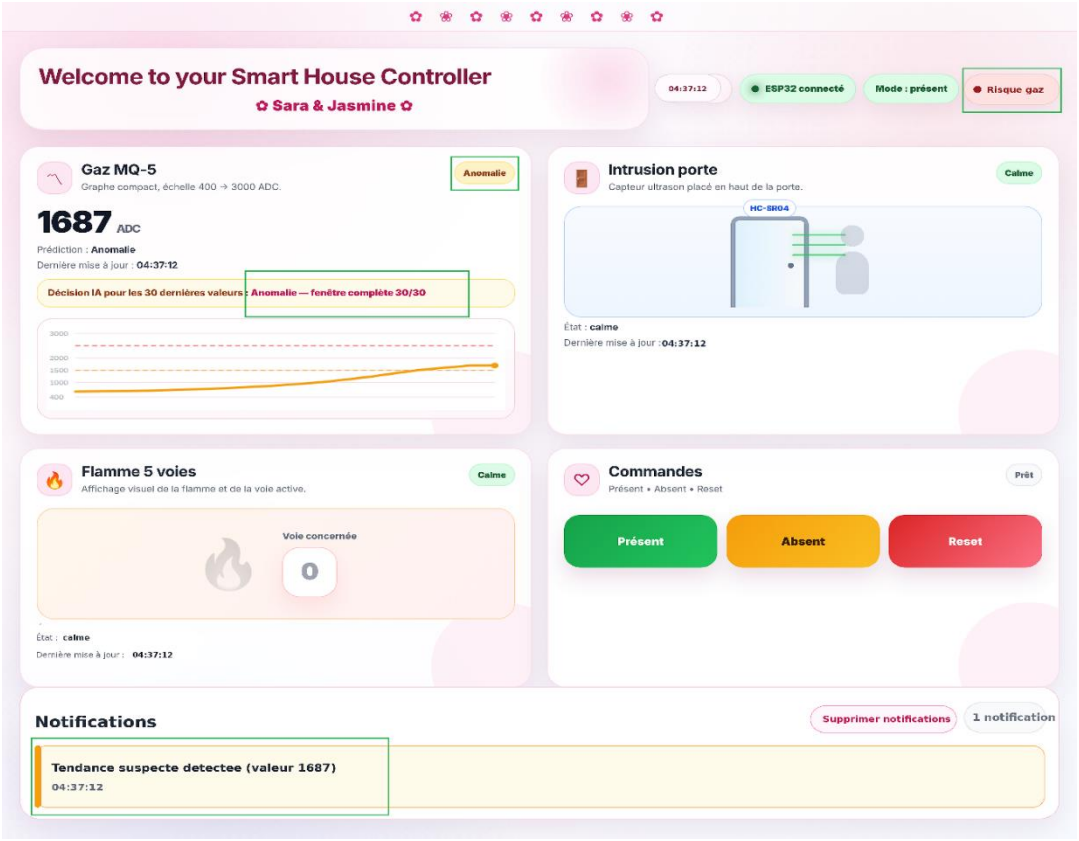


Figure IV.32 : Interface du Dashboard en cas d'anomalie de gaz détectée

2/ Cas de flamme

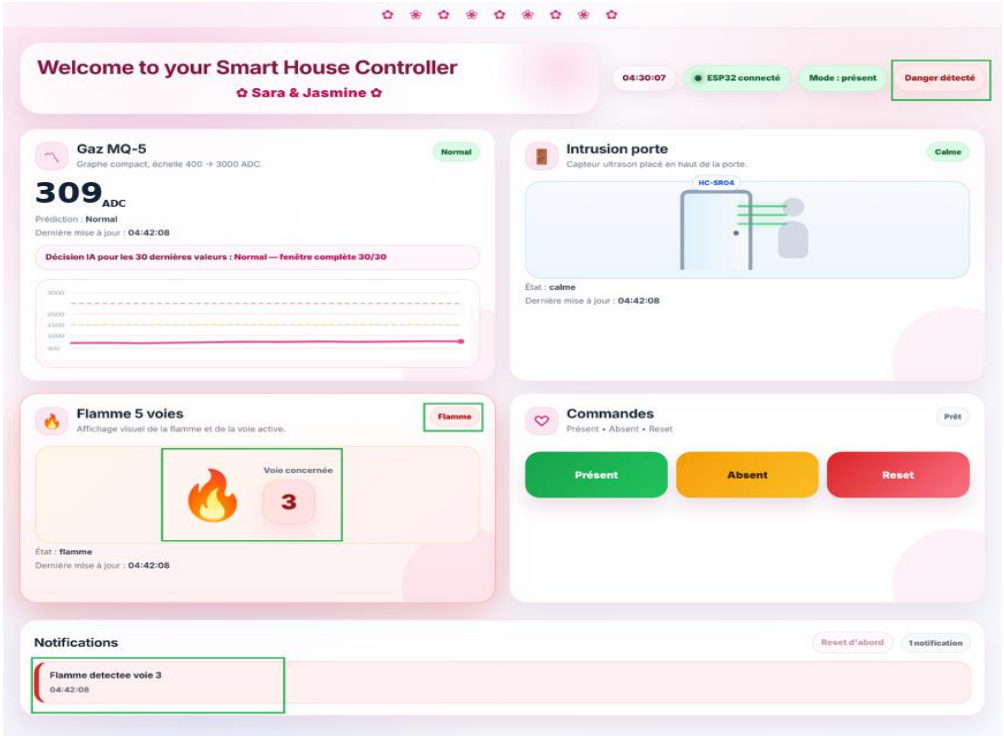


Figure IV.33 : Interface du Dashboard en cas de flamme détectée

3/ Cas d'intrusion

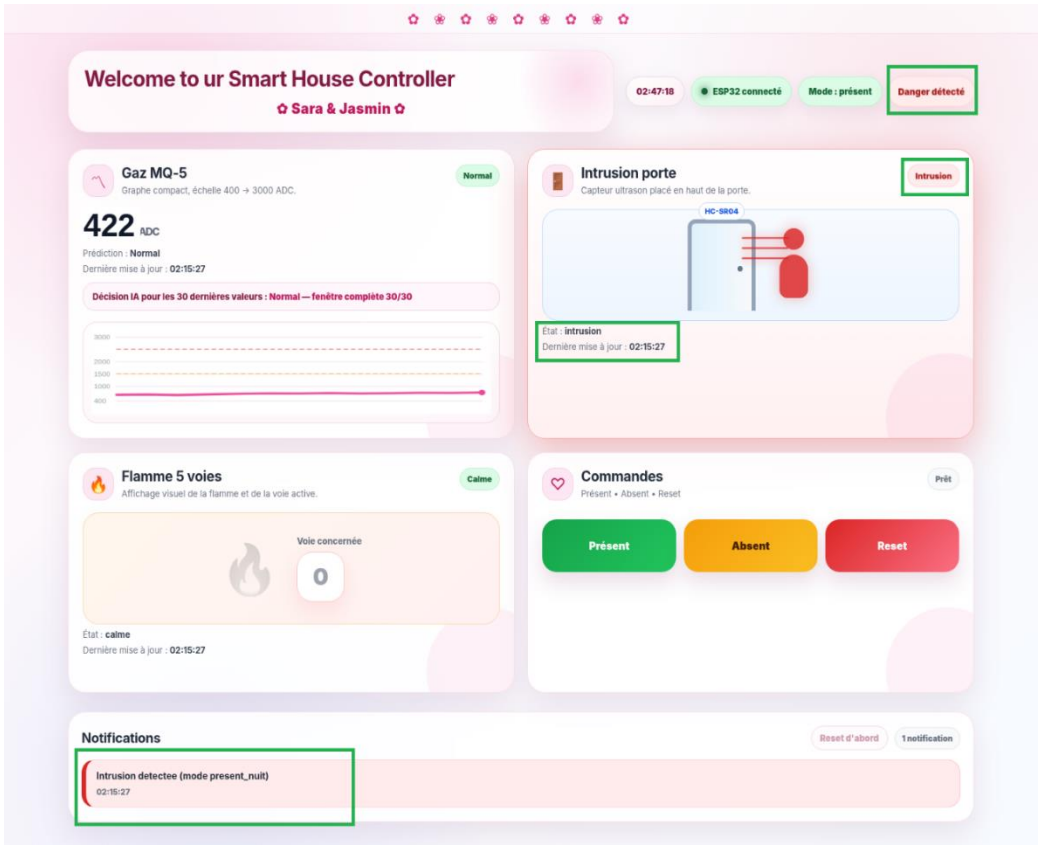


Figure IV.34 : Interface du Dashboard en cas d'intrusion détectée

IV.7.3 Envoi des commandes

Les trois boutons appellent la même fonction JavaScript avec une action différente.



Figure IV.35: Interface du Dashboard partie commandes

IV.8 Tests et validation expérimentale

Cette dernière étape vise à vérifier le bon fonctionnement de la chaîne complète du système, depuis l'acquisition des données par les capteurs jusqu'à l'affichage des résultats sur le Dashboard, en passant par le traitement embarqué sur l'ESP32, la communication MQTT, le serveur Flask et l'inférence du modèle LSTM.

IV.8.1 Protocole et scénarios testés

Les tests ont été conçus afin de valider le comportement du système dans différents scénarios représentatifs des cas d'usage définis.

Tableau IV.5 : Synthèse des tests fonctionnels du prototype

Test	Méthode	Résultat attendu	Résultat
Gaz MQ-5 normal	Valeur stable du capteur	Affichage normal, aucune alarme	OK
Gaz montant	Source de gaz approchée du MQ-5	Anomalie puis Critique si seuil dépassé	OK
Intrusion porte	Passage devant le HC-SR04	Notification + alarme selon le mode	OK
Flamme 5 voies	Flamme approchée d'une voie	Affichage voie + action pompe/servo	OK
Commandes	Boutons Présent, Absent, Reset	Changement de mode et remise à zéro	OK
Dashboard	Rafraîchissement automatique	Mise à jour des cartes et notifications	OK

IV.8.2 Résultats observés

Les essais expérimentaux ont permis d'observer le comportement global du système dans des conditions simulées de fonctionnement réel :

- Le niveau de gaz est affiché en temps réel sur le Dashboard, accompagné de la prédiction issue du modèle LSTM.

- Le serveur déclenche l'inférence uniquement lorsque le buffer temporel de 30 valeurs est complètement rempli, conformément à la logique de fenêtre glissante.
- La détection d'intrusion dépend du mode de fonctionnement actif et les événements sont correctement transmis à l'interface utilisateur.
- La détection de flamme permet d'identifier la voie activée et déclenche immédiatement les actionneurs locaux associés.
- La commande de réinitialisation (*Reset*) permet de remettre les états d'alarme à leur état nominal après traitement des événements.
- La chaîne de communication ESP32 → MQTT → Flask → Dashboard reste stable et fonctionnelle durant l'ensemble des tests.

IV.8.3 Synthèse et validation

Les résultats obtenus confirment que le prototype respecte les objectifs fonctionnels définis, à savoir la détection multi-capteurs, le traitement intelligent des données, la prise de décision locale et la supervision à distance via une interface web.

L'architecture hybride mise en place permet de distinguer clairement :

- Une logique prédictive basée sur le modèle LSTM pour le gaz,
- Une logique événementielle directe pour l'intrusion et la flamme,
- Et une supervision centralisée via le serveur et le Dashboard.

Par ailleurs, les mécanismes de sécurité embarqués garantissent une réponse locale immédiate en cas de situation critique (gaz ou flamme), indépendamment de l'état du serveur ou de la connectivité réseau.

Cette validation expérimentale démontre la cohérence globale du système (matériel et logiciel), qui intègre de manière conjointe perception, traitement, décision et action dans une architecture IoT distribuée fonctionnelle.

IV.9 Conclusion

Ce chapitre a présenté l'implémentation complète du système domotique intelligent, depuis la conception et la réalisation de la maquette physique en plexiglas jusqu'à sa validation expérimentale.

La maquette a permis de reproduire un environnement contrôlé dans lequel les différents scénarios de risque ont pu être testés de manière reproductible, notamment la détection de gaz, l'intrusion et la détection de flamme.

Sur le plan logiciel, le firmware embarqué sur l'ESP32 assure l'acquisition des données

capteurs, l'exécution des logiques de décision locales ainsi que la communication avec le reste du système via le protocole MQTT. Cette couche embarquée garantit une réponse autonome en temps réel pour les situations critiques.

Le serveur Flask joue quant à lui un rôle d'orchestrateur central en assurant la réception des données, l'intégration du modèle LSTM pour la prédiction du niveau de gaz, la gestion des événements système et la mise à disposition d'une interface de supervision via le Dashboard.
Les tests expérimentaux réalisés confirment le bon fonctionnement de l'ensemble de l'architecture. Ils valident également la complémentarité des trois approches développées au chapitre III prédictive pour le gaz, contextuelle pour l'intrusion et événementielle directe pour la flamme qui s'intègrent de manière cohérente dans un système unifié.

L'ensemble constitue ainsi un prototype fonctionnel de système domotique intelligent distribué, combinant traitement embarqué, intelligence artificielle et supervision web, et démontrant la faisabilité d'une architecture hybride orientée vers la sécurité autonome du domicile.

Conclusion Générale

Conclusion Générale

Au terme de ce travail, ce projet nous a permis de transformer une idée initiale, celle d'une maison plus sûre et plus intelligente — en un prototype concret et fonctionnel, capable de surveiller plusieurs risques domestiques et de réagir automatiquement en fonction du contexte. L'objectif n'était pas uniquement de détecter les dangers après leur apparition, mais de proposer une approche plus avancée reposant sur la prédiction, l'analyse contextuelle et l'autonomie locale. Ce mémoire vise ainsi à démontrer qu'un système domotique moderne doit non seulement détecter un danger, mais également en comprendre l'évolution et agir le plus tôt possible.

1. Bilan du travail réalisé

L'étude du contexte général a permis de mettre en évidence les limites des systèmes classiques, basés sur des seuils fixes et souvent incapables d'anticiper des phénomènes progressifs comme les fuites de gaz.

Le prototype développé repose sur une architecture cohérente. L'ESP32 joue le rôle de nœud central local, assurant la collecte des données, la communication via WiFi/MQTT et le contrôle des actionneurs. Le serveur centralise les informations, héberge le modèle LSTM et assure la supervision via un tableau de bord (dashboard).

Pour la détection de gaz, le capteur MQ-5 alimente un modèle LSTM qui analyse des fenêtres temporelles de mesures afin d'identifier des tendances anormales avant le dépassement du seuil critique. Pour l'intrusion, une approche contextuelle a été adoptée, combinant un contexte temporel (module RTC distinguant jour/nuit) et un contexte utilisateur (mode Présent/Absent défini via le dashboard). Enfin, pour l'incendie, la réaction est entièrement locale : un capteur de flamme à cinq directions permet d'estimer l'origine du feu, tandis qu'un servomoteur oriente le système d'extinction et active la pompe de manière immédiate.

2. Limites et difficultés rencontrées

Ce projet a été confronté à plusieurs contraintes techniques inhérentes à la mise en œuvre expérimentale d'un système domotique réel.

La collecte des données relatives aux fuites de gaz s'est révélée particulièrement délicate. L'utilisation d'une source ponctuelle telle qu'un briquet ne permettant pas de reproduire une évolution progressive du phénomène, une simulation contrôlée de fuites a été mise en place afin d'obtenir des données plus représentatives, tout en respectant les conditions de sécurité. Par ailleurs, la sensibilité du capteur MQ-5 aux variations environnementales (température, luminosité et circulation d'air) a rendu les seuils issus de la documentation constructeur

difficilement exploitables, ce qui a nécessité une recalibration directe sur la maquette expérimentale.

Concernant le scénario de surcharge électrique, son implémentation s'est heurtée à des contraintes matérielles et de sécurité. Les essais réalisés ont mis en évidence un risque de dégradation de certains composants, ce qui a conduit à limiter les expérimentations afin de préserver l'intégrité du système. De plus, la génération d'un profil de surcharge progressif et contrôlé nécessitait l'utilisation d'équipements spécifiques, notamment un rhéostat adapté, non disponible dans le cadre du projet. Cette limitation met en évidence la difficulté de reproduire fidèlement un environnement électrique domestique réel dans un contexte de maquette expérimentale.

Enfin, le capteur PIR initialement utilisé pour la détection d'intrusion a présenté une instabilité dans l'environnement de test, se traduisant par des variations non souhaitées et des déclenchements erronés. Après analyse, une solution alternative basée sur un capteur ultrasonique HC-SR04 a été adoptée. Ce choix a permis d'améliorer la stabilité de la détection et de garantir une mesure plus fiable de la présence dans une zone définie.

3. Perspectives et améliorations futures

Ce prototype constitue une base vers une maison véritablement intelligente. Plusieurs améliorations peuvent être envisagées.

Sur le plan matériel, l'intégration de capteurs complémentaires (température, humidité, fumée, qualité de l'air) permettrait une fusion de données plus robuste et une meilleure interprétation des risques.

Sur le plan logiciel, une approche d'apprentissage continu pourrait être mise en place, où le modèle est réentraîné périodiquement à partir des données collectées, afin de s'adapter aux spécificités de chaque environnement domestique.

Une comparaison des résultats de notre modèle avec d'autres modèles d'apprentissage doit également être réalisée , comme on peut utiliser un autre modèle d'apprentissage différents de LSTM.

L'interface peut également évoluer vers une application mobile intégrant notifications en temps réel, contrôle à distance, alertes SMS pour les zones sans Internet, et éventuellement des commandes vocales.

Enfin, ce travail ouvre des perspectives d'intégration avec les technologies modernes de la domotique, notamment l'Edge AI, la cybersécurité IoT et les assistants intelligents, vers des

systèmes domestiques capables non seulement de surveiller, mais aussi de comprendre, apprendre et anticiper leur environnement.

En conclusion, ce mémoire a permis de concevoir et de réaliser un prototype domotique intégrant détection, prédiction, analyse contextuelle et action automatique. Malgré les contraintes rencontrées — collecte de données, limitations matérielles et instabilité de certains capteurs — l'objectif principal a été atteint : démontrer la faisabilité d'un système de sécurité domestique plus intelligent et plus proactif.

Ce travail constitue ainsi une base solide pour des améliorations futures et représente une étape significative vers le développement de maisons réellement autonomes, sûres et intelligentes.

Bibliographie & Référence

[1] « Révolution de la domotique : les clés pour une maison plus intelligente en 2024 », Latour & Petit. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://latouretpetit.be/blog/revolution-de-la-domotique-les-cles-pour-une-maison-plus-intelligente-en-2024/Mémoire
[2] A. Reha *et al.*, « Conception et réalisation d'une maison intelligente », in *Colloque sur les Objets et systèmes Connectés*, CASABLANCA, Morocco: Ecole Supérieure de Technologie de Casablanca (Maroc), Institut Universitaire de Technologie d'Aix-Marseille (France), juin 2019. Consulté le: 28 mars 2026. [En ligne]. Disponible sur: https://hal.science/hal-02298836
[3] R. Duss et L. Salamolard, « La Domotique : La Maison du Futur ».
[4] « **Histoire de la domotique : des origines aux innovations modernes** », https://www.elecstore.fr/. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.elecstore.fr/histoire-fascinante-domotique-debuts-aujourd-hui/
[5] G. Noureddine, M. Y. Dalila, G. Hocine, et A. Salim, « Réalisé par : - MEGUEHOUT Walid - MEGHZILI Yasser ».
[6] A. Zanella, N. Bui, A. Castellani, L. Vangelista, et M. Zorzi, « Internet of Things for Smart Cities », *IEEE Internet Things J.*, vol. 1, n° 1, p. 22-32, févr. 2014, doi: 10.1109/JIOT.2014.2306328.
[7] S. Imad, « Internet des Objets (IdO) : Concepts, Enjeux, Défis et Perspectives », *Internet Obj.*, vol. 2, févr. 2018, doi: 10.21494/ISTE.OP.2018.0229.
[8] Admin, « That "Internet of Things" Thing », RFID JOURNAL. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.rfidjournal.com/expert-views/that-internet-of-things-thing/73881/
[9] « Why it is called Internet of Things: Definition, history, disambiguation ». Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://iot-analytics.com/internet-of-things-definition/
[10] F. Skandrani, « Architecture IoT : L'essentiel à savoir », IoT Industriel Blog. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/
[11] G. Phion, « IoT en 2026 : maîtrisez l'Internet des Objets de A à Z », Rotek. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://rotek.fr/iot-internet-des-objets/
[12] A. S. Abdul-Qawy, « The Internet of Things (IoT): An Overview », vol. 5, n° 12, 2015.
[13] E. Al-Masri *et al.*, « Investigating Messaging Protocols for the Internet of Things (IoT) », *IEEE Access*, vol. 8, p. 94880-94911, 2020, doi: 10.1109/ACCESS.2020.2993363.
[14] N. Naik, « Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP », in *2017 IEEE International Systems Engineering Symposium (ISSE)*, oct. 2017, p. 1-7. doi: 10.1109/SysEng.2017.8088251.
[15] « Introducing the MQTT Protocol – MQTT Essentials: Part 1 ». Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/
[16] « MQTT Protocol Explained: Publish-Subscribe, QoS Levels, Topics & Retained Messages ». Consulté le: 14 avril 2026. [En ligne]. Disponible sur: https://justprotocols.com/protocols/mqtt
[17] D. Soni et A. Makwana, *A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT)*. 2017.
[18] Z. Shelby, K. Hartke, et C. Bormann, « The Constrained Application Protocol (CoAP) », Internet Engineering Task Force, Request for Comments RFC 7252, juin 2014. doi: 10.17487/RFC7252.
[19] « What is the Constrained Application Protocol (CoAP)? », A1 Digital. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.a1.digital/knowledge-hub/what-is-the-constrained-application-protocol-coap/
[20] « CoAP: A Smart Solution Enabling M2M Applications », IoT For All. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.iotforall.com/coap-a-smart-solution-enabling-m2m-applications
[21] G. Kalanandhini, A. R. Aravind, G. Vijayalakshmi, J. Gayathri, et K. K. Senthilkumar, « Bluetooth technology on IoT using the architecture of Piconet and Scatternet », *AIP Conf. Proc.*, vol. 2393, n° 1, p. 020121, mai 2022, doi: 10.1063/5.0074188.

[22] « (PDF) A review on comparative study of Bluetooth and ZigBee », ResearchGate. Consulté le: 1 juin 2026. [En ligne]. Disponible sur: https://www.researchgate.net/publication/393515922_A_review_on_comparative_study_of_Bluetooth_and_ZigBee
[23] admin, « Bluetooth Classic Introduction and Architecture - ». Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://circuitlabs.net/bluetooth-classic-introduction-and-architecture/
[24] « Introduction of ZigBee », GeeksforGeeks. Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.geeksforgeeks.org/computer-networks/introduction-of-zigbee/
[25] « Zigbee Protocol - an overview | ScienceDirect Topics ». Consulté le: 3 avril 2026. [En ligne]. Disponible sur: https://www.sciencedirect.com/topics/engineering/zigbee-protocol
[26] « Comparative Study of IoT Protocols | Request PDF », *ResearchGate*, doi: 10.2139/ssrn.3186315.
[27] G. Vardakis, G. Hatzivasilis, E. Koutsaki, et N. Papadakis, « Review of Smart-Home Security Using the Internet of Things », *Electronics*, vol. 13, n° 16, p. 3343, janv. 2024, doi: 10.3390/electronics13163343.
[28] « Le guide ultime du système d'alarme anti-intrusion en 2024 : protéger complètement votre maison ». Consulté le: 15 avril 2026. [En ligne]. Disponible sur: https://www.roombanker.com/fr/blog/intrusion-alarm-system-complete-guide/
[29] B.-K. Chew, A. Mahmud, et H. Singh, « Autonomous Hazardous Gas Detection Systems: A Systematic Review », *Sensors*, vol. 25, n° 21, p. 6618, janv. 2025, doi: 10.3390/s25216618.
[30] « Statistiques mondiales sur les incendies | CTIF - International Association of Fire Services for Safer Citizens through Skilled Firefighters ». Consulté le: 15 avril 2026. [En ligne]. Disponible sur: https://ctif.org/fr/statistiques-mondiales-sur-les-incendies
[31] « Comment utiliser ACS712 capteur de courant Arduino - Moussasoft ». Consulté le: 15 avril 2026. [En ligne]. Disponible sur: https://www.moussasoft.com/acs712-capteur-de-courant-arduino/
[32] A. Zimmermann, A. Lorenz, et R. Oppermann, « An Operational Definition of Context », in *Modeling and Using Context*, B. Kokinov, D. C. Richardson, T. R. Roth-Berghofer, et L. Vieu, Éd., Berlin, Heidelberg: Springer, 2007, p. 558-571. doi: 10.1007/978-3-540-74255-5_42.
[33] C. Perera, A. Zaslavsky, P. Christen, et D. Georgakopoulos, « Context Aware Computing for The Internet of Things: A Survey », *IEEE Commun. Surv. Tutor.*, vol. 16, n° 1, p. 414-454, 2014, doi: 10.1109/SURV.2013.042313.00197.
[34] C. Yin, « Samcco : un Système d'Apprentissage Mobile Contextuel et Collaboratif dans des Situations Professionnelles », Theses, Ecole Centrale de Lyon, 2010. Consulté le: 1 juin 2026. [En ligne]. Disponible sur: https://theses.hal.science/tel-00564488
[35] J. Luo et H. Feng, « A Framework for NFC-based Context-aware Applications », *Int. J. Smart Home*, vol. 9, n° 1, p. 111-122, janv. 2015, doi: 10.14257/ijsh.2015.9.1.12.
[36] F. Charpillet, S. Ploix, et P. Reignier, « Introduction (FR) », *Rev. Ouverte Intell. Artif.*, vol. 4, n° 1, p. 1-9, mai 2023, doi: 10.5802/roia.48fr.
[37] M. I. I. Amezzane, « L'intelligence artificielle dans les Smart Homes ».
[38] « detection - Advanced search results in Entries | Oxford English Dictionary ». Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://www.oed.com/search/advanced/Entries?q=detection&sortOption=Frequency&tl=true
[39] « prediction ». Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://dictionary.cambridge.org/dictionary/english/prediction
[40] « Tous les modèles de Machine Learning expliqués brièvement », MonCoachData. Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://moncoachdata.com/blog/modeles-de-machine-learning-expliques/

[41] C. R. China, « Types de machine learning | IBM ». Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://www.ibm.com/fr-fr/think/topics/machine-learning-types
[42] « Machine Learning : Les grands types de modèles de données », https://www.clicdata.com/fr/. Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://www.clicdata.com/fr/blog/modeles-machine-learning/
[43] « 12 exemples de Machine Learning tirés du monde réel », MonCoachData. Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://moncoachdata.com/blog/12-exemples-de-machine-learning-tires-du-monde-reel/
[44] « Modèle ARIMA avec Python – Prévisions de séries temporelles », MonCoachData. Consulté le: 2 juin 2026. [En ligne]. Disponible sur: https://moncoachdata.com/blog/modele-arima-avec-python/
[45] « Distinguer besoins fonctionnels et non-fonctionnels | PDF | Application | Développement de logiciel », Scribd. Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://fr.scribd.com/document/681720002/OBJECTIFS-GLOBAL-ET-SPECIFIQUES-Comment-distinguer-l-expression-des-besoins-fonctionnels-et-non-fonctionnels
[46] « SURVEILLANCE : Définition de SURVEILLANCE ». Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://www.cnrtl.fr/definition/surveillance
[47] H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, et Z. Kastrati, « Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm », *Sensors*, vol. 21, n° 14, p. 4932, janv. 2021, doi: 10.3390/s21144932.
[48] D. Schneiderbauer, « Predictive Analytics for Smart Homes using Serverless Edge Computing », *Softw. Eng.*.
[49] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, et M. Salehi, « Deep Learning for Time Series Anomaly Detection: A Survey », arXiv.org. Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://arxiv.org/abs/2211.05244v3
[50] S. A. Z. Hassan et A. M. Eassa, « A Proposed Architecture for Smart Home Systems Based on IoT, Context-awareness and Cloud Computing », *Int. J. Adv. Comput. Sci. Appl. Ijacsa*, vol. 13, n° 6, juin 2022, doi: 10.14569/IJACSA.2022.0130612.
[51] S. Alami, « Real-Time Energy Monitoring and Anomaly Detection in IoT-Based Systems: A Hybrid Approach Combining Isolation Forest and Dynamic Baselines for Sustainable Energy Awareness ».
[52] R. Teja, « Getting Started with ESP32 | Introduction to ESP32 », ElectronicsHub. Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://www.electronicshub.org/getting-started-with-esp32/
[53] « Welcome to Python.org », Python.org. Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://www.python.org/
[54] « MySQL ». Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://www.mysql.com/
[55] « Welcome to Flask — Flask Documentation (3.1.x) ». Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://flask.palletsprojects.com/en/stable/
[56] « HTML CSS JavaScript », HTML CSS JavaScript. Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://html-css-js.com/
[57] Admin, « UML – Guide complet », Visual Paradigm Blog Français. Consulté le: 4 juin 2026. [En ligne]. Disponible sur: https://blog.visual-paradigm.com/fr/uml-a-comprehensive-guide/
[58] alldatasheet.com, « MQ-5 Datasheet(PDF) ». Consulté le: 12 juin 2026. [En ligne]. Disponible sur: http://www.alldatasheet.com/datasheet-pdf/pdf/1304543/WINSEN/MQ-5.html
[59] « MQ-5 Datasheet | HANWEI ELETRONICS ». Consulté le: 12 juin 2026. [En ligne]. Disponible sur: https://datasheet4u.com/datasheets/HANWEI-ELETRONICS/MQ-5/904638
[60] « Propane - IDLH | NIOSH | CDC ». Consulté le: 12 juin 2026. [En ligne]. Disponible sur: https://www.cdc.gov/niosh/idlh/74986.html
[61] « CAMEO Chemicals | NOAA ». Consulté le: 12 juin 2026. [En ligne]. Disponible sur:

https://cameochemicals.noaa.gov/search/results
[62] « CENELEC - EN 50194-1 - Electrical apparatus for the detection of flammable gases in household premises - Part 1: Test methods and performance requirements | GlobalSpec ». Consulté le: 12 juin 2026. [En ligne]. Disponible sur: https://standards.globalspec.com/std/14639159/en-50194-1
[63] S. Hochreiter et J. Schmidhuber, « Long Short-Term Memory », *Neural Comput.*, vol. 9, p. 1735-1780, nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[64] D. P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization », 30 janvier 2017, *arXiv*: arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980.