



UNIVERSITE HASSIBA BENBOUALI DE CHLEF

Faculté de Technologie

Département d'Electronique

MEMOIRE DE MASTER

Domaine : SCIENCES ET TECHNOLOGIES

Filière : : Electronique

Spécialité : Electronique des Systèmes Embarqués

Conception et réalisation d'un système de sécurité intelligent pour une
maison.

Par

Affaf MEKRELOUF

Encadreur :

Mme. KESSAISSIA Fatma Zohra

Maitre de Conférences B à l'UHBC

Co-Encadreur :

M. ARAB Mustapha

Maitre de Conférences B à l'UHBC

Chlef, Juin 2025

Dédicace

À mes chers parents,

Pour leur amour inconditionnel, leur patience, leurs sacrifices et leur soutien indéfectible tout au long de notre parcours. Vous êtes les piliers de notre réussite, et c'est à vous que nous devons ce que nous sommes aujourd'hui. Nous espérons être à la hauteur de vos attentes et de votre confiance. Que Dieu vous protège et vous comble de ses bienfaits.

Je dédie également ce travail à nos frères et sœurs, pour leur affection sincère, leur présence constante et leurs encouragements, qui nous ont toujours été d'un grand réconfort.

Enfin, je souhaite adresser une pensée particulière à mon amie Hassani Aya, pour son soutien fidèle, sa bienveillance, et sa présence à mes côtés dans les moments les plus difficiles. Merci d'avoir partagé avec moi chaque étape de ce chemin, jusqu'à la réussite.

REMERCIEMENTS

Avant tout, je rends grâce à Dieu le Tout-Puissant, qui m'a offert la force, la patience et l'opportunité d'entreprendre ce parcours universitaire et de mener à bien ce travail de fin d'études.

Je tiens à exprimer ma profonde gratitude à ' Madame KESSAISSIA Fatma Zohra', ma promotrice, pour sa disponibilité, ses conseils avisés, et l'attention bienveillante qu'elle a portée à chaque étape de ce projet. Son accompagnement rigoureux, son sens du devoir, ainsi que sa rigueur scientifique ont été pour moi une source d'inspiration et de motivation. Je la remercie sincèrement pour sa patience, sa générosité et pour le temps qu'elle a consacré à l'encadrement de ce travail.

Je tiens à remercier vivement Dr. M. ARAB pour son soutien et sa présence durant ce travail de mémoire

Je tiens également à remercier Dr. A. BENBOUALI pour ses orientations, ses conseils et son suivi

Je suis très honorée que le Dr. B. BENICHOU a accepté la présidence du jury, et le Dr. I. Hadjadj a accepté la charge de juger et d'examiner ce mémoire.

Mes vifs remerciements s'adressent également à l'ensemble des enseignants et aux staffs administratifs de département d'Électronique et d'Électrotechnique, pour leurs contributions à ma formation et pour l'environnement pédagogique qu'ils ont su offrir tout au long de mon cursus.

J'exprime aussi ma reconnaissance à toutes les personnes qui, de près ou de loin, ont participé à la réalisation de ce mémoire, par leur aide, leurs conseils ou leur soutien moral, notamment dans les moments de difficulté.

Enfin, je remercie chaleureusement mes amis et collègues, pour leur soutien constant, leurs encouragements et leurs présences tout au long de cette aventure académique.

Merci à tous.

Résumé

Dans un contexte où la sécurité domestique devient une priorité face aux risques tels que les intrusions, les fuites de gaz ou les incidents sonores inhabituels, ce travail propose la conception et la réalisation d'un système de sécurité intelligent pour l'habitat. Le système repose sur l'utilisation du microcontrôleur ESP32-WROOM-32, qui permet l'acquisition et le traitement des données issues de plusieurs capteurs (gaz, mouvement, bruit, ouverture porte/fenêtre) et la commande d'actionneurs (LEDs, buzzer, servo-moteur) pour assurer une réponse immédiate aux situations critiques. La programmation a été effectuée via l'environnement Arduino IDE, tandis que les données sont stockées dans la base de données en temps réel Firebase, permettant une consultation à distance. Une application mobile, développée avec MIT App Inventor, offre à l'utilisateur une interface intuitive pour surveiller l'état du système, recevoir des alertes et effectuer des actions à distance. Ce projet illustre l'intégration efficace de l'Internet des Objets (IoT) dans un cadre domestique, en combinant fiabilité, accessibilité et interactivité pour renforcer la sécurité des habitations modernes.

Mots clés : système de sécurité intelligent, base de données en temps réel Firebase, application mobile, Internet des Objets

Abstract

In a context where home security is becoming increasingly important due to risks such as intrusions, gas leaks, or unusual noise disturbances, this work presents the design and implementation of an intelligent home security system. The system is based on the ESP32-WROOM-32 microcontroller, which enables the acquisition and processing of data from various sensors (gas, motion, sound and door/window opening) and the control of actuators (LEDs, buzzer, servo motor) to ensure an immediate response to critical events. The system was programmed using the Arduino IDE, while real-time data is stored in Firebase, allowing remote monitoring. A mobile application, developed with MIT App Inventor, provides users with an intuitive interface to monitor the system status, receive alerts, and perform remote actions. This project demonstrates the effective integration of Internet of Things (IoT) technologies in a domestic setting, combining reliability, accessibility, and interactivity to enhance the safety of modern homes.

Keywords : intelligent home security, Firebase real-time data, mobile application, Internet of Things

الملخص

في سياقٍ أصبح فيه أمن المنازل أولويةً في مواجهة مخاطر مثل الاقتحامات وتسربات الغاز وحوادث الضوضاء غير الاعتيادية، يقترح هذا المشروع تصميم وتنفيذ نظام أمن منزلي ذكي. يعتمد النظام على استخدام متحكم دقيق -ESP32-WROOM-32. يُمكن من جمع ومعالجة البيانات من أجهزة استشعار متعددة (غاز، حركة، ضجيج، فتح باب/نافذة) والتحكم في المشغلات (صمام ثنائي باعث للضوء، طنان، محرك مؤازر) لضمان استجابة فورية للحالات الحرجة. تمت البرمجة عبر البيئة **Arduino IDE**، بينما تُخزّن البيانات في قاعدة البيانات في الوقت الفعلي (**Firebase**)، مما يسمح بالوصول عن بُعد. يوفر تطبيق الجوال، تم تطويره باستخدام (**MIT App Inventor**)، واجهة سهلة الاستخدام لمراقبة حالة النظام، وتلقي التنبيهات، وتنفيذ الإجراءات عن بُعد. يوضح هذا المشروع التكامل الفعال لإنترنت الأشياء (**Internet of Things (IoT)**) في البيئة المنزلية، حيث يجمع بين الموثوقية وسهولة الوصول والتفاعلية لتعزيز أمن المنازل الحديثة.

الكلمات المفتاحية : نظام أمن منزلي ذكي، قاعدة البيانات في الوقت الفعلي، تطبيق الجوال، إنترنت الأشياء

Table de matières

Dédicace	I
Remerciements.....	II
Résumé.....	IV
Abstract.....	V
المخلص.....	VI
Liste des figures.....	XI
Liste des tableaux.....	XIV
Abréviations.....	XV
Introduction générale	2
CHAPITRE I : Etat de l’art sur la sécurité domestique intelligente	4
I.1. Introduction.....	5
I.2. Sécurité domestique	5
I.3. Importance de la sécurité dans les habitations modernes	6
I.4. Classification des risques domestiques	7
I.5. Classification des systèmes de sécurité.....	8
I.5.1. Systèmes de sécurité traditionnels	8
I.5.2. Caractéristiques des systèmes traditionnels	9
I.5.3. Limites des approches classiques	9
I.5.4. Évolution vers les systèmes de sécurité intelligents	10
I.6. Système de sécurité intelligent.....	11
I.6.1. Avantages des systèmes connectés et autonomes.....	11
I.6.2. Domaines d’application des systèmes de sécurité intelligents	12
I.7. Technologies utilisées dans la sécurité intelligente	14
I.7.1. Internet des Objets (IoT)	14
I.7.2. Composant de l’internet des objets	14
I.7.3. Fonctionnement de l’internet des objets.....	15
I.7.4. Internet des Objets (IoT) dans la sécurité domestique.....	15
I.8. Conclusion	16
Bibliographie chapitre 1	17
CHAPITRE II : Composantes matérielles d’un système de sécurité intelligent	19
II.1. Introduction.....	20
II.2. Microcontrôleur ESP32-WROOM-32	20
II.2.1. Historique	20
II.2.2. Différents types de microcontrôleur ESP32.....	21

Table de matière

II.2.3.	Définition de l'ESP32-WROOM-32	22
II.2.4.	Caractéristiques techniques	22
II.2.5.	Interfaces de commutation.....	23
II.2.6.	Architecture interne	24
II.2.7.	Principe de fonctionnement	24
II.3.	Capteurs	25

II.3.1. Capteur de Gas MQ-9	25
II.3.2. Capteur de bruit KY-037.....	28
II.3.3. Capteur de mouvement PIR	30
II.3.4. Switch	31
II.4. Actionneurs.....	32
II.4.1. Servomoteur	32
II.4.2. BUZZER	34
II.4.3. Diode électroluminescente LED.....	34
II.5. Alimentation des composants	35
II.5.1. Régulateur 3V3.....	35
II.5.2. Régulateur 5V.....	36
II.6. Circuit Imprimé PCB	36
II.7. Conclusion	40
Bibliographie chapitre 2	41
CHAPITRE III : Composantes logicielles d'un système de sécurité intelligent	42
III.1. Introduction.....	43
III.2. Présentation de logiciel Arduino IDE.....	43
III.2.1. Définition	43
III.2.2. Langage	44
III.2.3. Interfaces	44
III.2.4. Critères de choix.....	46
III.2.5. Configuration et initialisation des composant.....	47
III.2.5.1. Installation de la carte ESP32.....	47
III.2.5.2. Configurations du capteur de Gas MQ-9.....	49
III.2.5.3. Configurations du capteur de mouvement PIR.....	50
III.2.5.4. Configurations de capteur de bruit KY-037	51
III.2.5.5. Configurations de SWITCH.....	52
III.2.5.6. Configurations de Servomoteur.....	53
III.3. Présentation de la plateforme Fire base.....	54
III.3.1. Définition	54
III.3.2. Principales fonctionnalités de Fire base	55
III.3.3. Type de base de données de Fire base	55
III.3.4. Critères de choix	55
III.3.5. Configuration d'un compte Fire base	56
III.4. Cycle de fonctionnement	61

Table de matière

III.5.	Présentation MIT APP inventor	66
III.5.1.	Définition	66
III.5.2.	Interface de MIT APP Inventor.....	67
III.5.3.	Développement d'une application mobile par MIT APP.....	68
III.6.	Conclusion	72
	Bibliographie chapitre 3	73
	CHAPITRE IV : Réalisation d'un système de sécurité intelligent pour une maison	75
IV.1.	Introduction.....	76
IV.2.	Modélisation de la maison	76
IV.3.	Disposition des composants	77
IV.4.	Architecture globale du système	77
IV4.1.	Partie utilisateur.....	78
IV4.2.	Partie ESP32.....	79
IV4.3.	Partie matérielle.....	79
IV.5.	Tests fonctionnels des sous-systèmes.....	79
IV5.1.	Test du capteur de Gas MQ-9	79
IV5.2.	Test des switches de porte et de fenêtres.....	82
IV5.3.	Test des capteurs de bruit KY-037 et de mouvement PIR.....	84
IV.6.	Conclusion	86
	Conclusion générale	88

Liste des figures

Chapitre I

Figure I.1. Différents systèmes de sécurité d'une maison	6
Figure I.2. Modernisation des habitations.....	7
Figure I.3. Différents risques domestiques.....	8
Figure I.4. Systèmes de sécurité traditionnels.....	9
Figure I.5. Sécurité dans les Bureaux et entreprises.....	13
Figure I.6. Sécurité dans les établissements de santé.....	13

Chapitre II

Figure II.1. Brochage de la carte ESP32-WROOM.....	22
Figure II.2. Architecture interne d'une carte ESP32-WROOM-32	24
Figure II.3. Composants du capteur de Gas.....	26
Figure II.4. Schéma de câblage de MQ-9 avec ESP32.....	27
Figure II.5. Capteur de bruit KY-037.....	28
Figure II.6. Composants du capteur de bruit KY-037.....	28
Figure II.7. Schéma de câblage du capteur KY-037 de bruit.....	29
Figure II.8. Capteur de mouvement PIR.....	30
Figure II.9. Composant externe du capteur de mouvement	30
Figure II.10. Fonctionnement d'un détecteur de mouvement PIR	31
Figure II.11. Switch NO/NC	31
Figure II.12. Servomoteur SG90	32
Figure II.13. Composants de servomoteur à l'intérieur et extérieur	33
Figure II.14. Simulation de l'ESP32 et le servomoteur	34
Figure II.15. BUZZER	34
Figure II.16. LED's	35
Figure II.17. Régulateur de tension 3.3V AUK626 S1117.....	35
Figure II.18. Module Régulateur DC-DC Step Down LM2596	36
Figure II.19. Composant de circuit imprimé dans le logiciel ARES	37
Figure II.20. Routage des éléments du circuit imprimé.....	37
Figure II.21. Vérification des erreurs de conception.....	38
Figure II.22. Visualisation 3D de la face avant.....	38
Figure II.23. Visualisation 3D de la face arrière.....	39
Figure II.24. Carte électronique face avant.....	39
Figure II.25. Carte électronique face arrière.....	40

Chapitre III

Figure III.1. Arduino IDE Icon.....	44
Figure III.2. Câble USB pour ESP32.....	44
Figure III.3. Interface de logiciel Arduino IDE.....	45
Figure III.4. Première barre d'interface d'Arduino IDE.....	45
Figure III.5. Deuxième barre d'interface d'Arduino IDE.....	46
Figure III.6. Configuration de l'ESP32 sur l'IDE Arduino	47
Figure III.7. Installation du logiciel pour la carte ESP32.....	48

Figure III.8. Sélection de la carte ESP32.....	48
Figure III.9. Sélection du port de communication.....	49
Figure III.10. Programme de configuration du capteur de Gas MQ-9.....	49
Figure III.11. Communication et essai du capteur de Gas MQ-9.....	50
Figure III.12. Programme de configuration de capteur de mouvement PIR.....	50
Figure III.13. Communication et essai du capteur de mouvement PIR.....	51
Figure III.14. Programme de configuration de capteur de bruit KY-037.....	51
Figure III.15. Communication et essai du capteur de bruit KY-037.....	52
Figure III.16. Programme de configuration des Switches.....	52
Figure III.17. Communication et essai des switches.....	53
Figure III.18. Programme de configuration de servomoteur.....	53
Figure III.19. Communication et essai de servomoteur.....	54
Figure III.20. Fire base.....	54
Figure III.21. Interface de Fire base.....	55
Figure III.22. Nouveau projet de Fire base.....	56
Figure III.23. Assistance par l'IA.....	56
Figure III.24. Google Analytics.....	57
Figure III.25. Projet est prêt.....	57
Figure III.26. Authentification du Fire base.....	58
Figure III.27. Méthode d'authentification.....	58
Figure III.28. Compte email et mot de passe.....	59
Figure III.29. Liste d'utilisateurs.....	59
Figure III.30. Options de la base de données.....	60
Figure III.31. Règles de sécurité.....	60
Figure III.32. URL de la base de données.....	61
Figure III.33. Clé API.....	61
Figure III.34. Schéma fonctionnel de Fire base et l'ESP32.....	62
Figure III.35. ESP32 comme une station.....	62
Figure III.36. Configuration de l'ESP32 comme une station.....	63
Figure III.37. Configuration de Fire base.....	64
Figure III.38. Transmission de la valeur de capteur de Gas vers Fire base.....	64
Figure III.39. Contrôle des actionneurs par Fire base.....	65
Figure III.40. Interface de Fire base (Realtime Database).....	66
Figure III.41. MIT APP Inventor.....	66
Figure III.42. Interface de Designer.....	67
Figure III.43. Interface des blocs.....	67
Figure III.44. Nouveau projet de MIT APP inventor.....	68
Figure III.45. Composants non-visuels.....	68
Figure III.46. Composants visuels sur l'interface.....	69
Figure III.47. Configuration de Fire base dans MIT APP Inventor.....	69
Figure III.48. Lecture des valeurs de Fire base.....	70
Figure III.49. Control des actionneurs.....	70
Figure III.50. Réinitialisation des données depuis Fire base.....	71
Figure III.51. Application MIT AI2 Companion.....	71

Figure III.52 le QR code	72
Figure III.53. Interface de téléphone Android.....	72

Chapitre IV

Figure IV.1. Structure du plan de la maison.....	76
Figure IV.2. Photo du prototype réalisé.....	77
Figure IV.3. Téléversement du programme.....	77
Figure IV.4. Architecture globale du système	78
Figure IV.5. Résultat avant la détection de fuite de Gas.....	79
Figure IV.6. Etat de maison au niveau normal de Gas	80
Figure IV.7. Niveau normal de Gas sur Fire base.....	80
Figure IV.8. Niveau normal de Gas sur l'application Android.....	80
Figure IV.9. Résultat après la détection de fuite de Gas.....	81
Figure IV.10. Etat de la maison après la fuite de Gas.....	81
Figure IV.11. Niveau de Gas après la détection sur Fire base	82
Figure IV.12. Niveau de Gas sur l'application Android.....	82
Figure IV.13. Résultat test accès non autorisé.....	83
Figure IV.14. Ouverture de la fenêtre et fermeture de la porte.....	83
Figure IV.15. Etat de la fenêtre et la porte et sur Fire base.....	83
Figure IV.16. Etat de la porte et de la fenêtre sur l'application Android.....	84
Figure IV.17. Résultat du test de détection combinée du KY-037 et PIR	84
Figure IV.18. Détection d'intrusion.....	85
Figure IV.19. Résultat de réinitialisation du compteur	86

Liste des tableaux

Chapitre II

Tableau II.1. Série de capteurs de Gas MQ.

Abréviations

IoT : Internet of Things.
GPL : Gaz de Pétrole Liquéfié
LCD : Liquid Crystal Display
Wi-Fi : Wireless Fidelity
GSM: Global System for Mobile communications
LED : Light Emitting Diode
LPWAN : Low Power Wide Area Network
SoC : System on Chip.
BLE : Bluetooth Low Energy.
GPIO : General Purpose Input/Output
SDK : Software Development Kit.
ADC : Analog Digital Converter.
PCB : Printed Circuit Board.
ARES : Advanced Routing and Editing Software.
DRC : Design Rule Checker.
GND : Ground.
IDE : Integrated Development Environment.
USB : Universal Serial Bus.
MQ : Capteur de gaz.
PIR : Passive Infra Red - détecteur infrarouge passif
PWM : Pulse width modulation
URL : Uniform Resource Locator
API : Application Programming Interface
NoSQL : Not Only Structured Query Language
IA : Intelligence Artificielle
UID : User Identifier
RTDB: Realtime Database

IP : Internet Protocol

SSID : Service Set Identifier

MIT : Massachusetts Institute of Technology QR : Quick Response

INTRODUCTION GÉNÉRALE

Introduction générale

Face à l'augmentation des risques domestiques tels que les fuites de gaz, les départs d'incendie ou les intrusions, la sécurité des habitations est devenue une préoccupation majeure dans les sociétés modernes. En Algérie, plusieurs incidents dramatiques liés à des fuites de gaz ont été rapportés au cours des dernières années, causant des pertes humaines et matérielles considérables. Cette réalité met en évidence la nécessité d'adopter des dispositifs de sécurité plus performants, réactifs et autonomes. Grâce aux avancées technologiques récentes, notamment dans le domaine de l'Internet des Objets (IoT), il est désormais possible de concevoir des systèmes intelligents capables de surveiller, détecter et alerter en temps réel.

Le présent chapitre a pour but d'établir les bases théoriques relatives aux systèmes de sécurité domestique, en comparant les solutions traditionnelles aux approches intelligentes, et en présentant les concepts fondamentaux liés à la sécurité, à la détection de risques et aux technologies de communication modernes. Ce cadre conceptuel permettra de mieux comprendre l'intérêt et la portée du projet de sécurité intelligent que nous avons entrepris.

Les habitations modernes sont confrontées à des risques multiples comme les fuites de gaz, intrusions, départs de feu dont la détection tardive expose les résidents à des dangers graves et aux pertes matérielles. En Algérie, l'absence de systèmes automatisés de surveillance et d'alerte a rendu ces incidents particulièrement critiques. Il devient donc essentiel de développer une solution résidentielle intégrée, capable de détecter en continu la présence de gaz, l'ouverture non autorisée des portes et fenêtres, ainsi que les prémices d'un incendie, puis de lancer immédiatement des actions préventives (envoi de notifications, déclenchement d'alarme...) et d'informer les occupants à distance pour garantir la sécurité des personnes et des biens.

Notre projet de fin d'études a pour objectif de développer un système de sécurité intelligent pour la maison, en intégrant des capteurs de présence, de gaz et un switch connecté. Nous visons à créer une solution permettant de détecter les intrusions et les fuites de gaz, tout en envoyant des notifications instantanées à l'utilisateur via une application mobile. Grâce à cette application, l'utilisateur pourra également gérer à distance les différents dispositifs de sécurité, assurant ainsi une surveillance et une protection optimales de son domicile.

Afin de mieux comprendre la conception et la réalisation d'un système de sécurité domestique intelligent, ce mémoire est structuré en quatre chapitres principaux :

Le premier chapitre a pour but d'établir les bases théoriques relatives aux systèmes de sécurité domestique, en comparant les solutions traditionnelles aux approches intelligentes, et en présentant les concepts fondamentaux liés à la sécurité, à la détection de risques et aux technologies de communication modernes. Ce cadre conceptuel permettra de mieux comprendre l'intérêt et la portée du projet de sécurité intelligent que nous avons entrepris.

Le deuxième chapitre traite des composantes matérielles utilisées pour la réalisation d'un système de sécurité domestique intelligent. Plus précisément, il explique le rôle, le fonctionnement et l'intégration des capteurs, des actionneurs et du microcontrôleur qui composent le système.

Le troisième chapitre est consacré à la partie logicielle du projet. Il décrit la programmation du microcontrôleur ESP32 à l'aide de l'Arduino IDE, l'envoi et le stockage des données dans la base de données Firebase, ainsi que le développement d'une application mobile conçue avec MIT (Massachusetts Institute of Technology) App Inventor pour l'affichage des informations en temps réel et le contrôle à distance du système. Ce chapitre met en lumière l'importance de l'intégration logicielle pour assurer l'intelligence, l'autonomie et l'interactivité du système de sécurité proposé.

Le quatrième chapitre expose les résultats de la réalisation pratique du système, en présentant les tests effectués et l'évaluation des performances. Il met en avant le fonctionnement de l'application mobile pour l'affichage des données en temps réel et le contrôle à distance, illustrant ainsi l'efficacité et la fiabilité du système de sécurité développé.

CHAPITRE I

Chapitre I

Etat de l'art sur la sécurité domestique intelligente

I.1. Introduction

Dans un contexte de modernisation croissante des habitations et face à la multiplication des risques domestiques (fuites de Gas, incendies, vol, etc.), la sécurité résidentielle est devenue un enjeu majeur. La protection des biens et des personnes ne repose plus uniquement sur des dispositifs classiques, mais s'oriente vers des solutions intelligentes, capables de détecter automatiquement les situations critiques et de réagir en temps réel.

Ce chapitre a pour objectif d'explorer les fondements théoriques liés aux systèmes de sécurité domestique. Il présente tout d'abord les définitions clés et les types de risques auxquels sont exposés les logements modernes. Ensuite, il propose une comparaison entre les systèmes traditionnels et les systèmes dits « intelligents », en mettant en lumière les avantages de l'Internet des Objets (IoT) dans ce domaine. Enfin, il aborde les principales technologies, concepts et modes de communication utilisés dans les systèmes de sécurité connectés.

I.2. Sécurité domestique

La sécurité domestique peut se définir, dans le cadre de la domotique et des systèmes connectés, comme l'ensemble des dispositifs et méthodes permettant de prévenir, détecter et réagir face aux risques qu'ils soient d'origine humaine (intrusion, cambriolage) ou non humaine (incendie, fuite de Gas, inondation...) au sein d'une habitation.

Elle vise en particulier :

- **La protection des personnes et des biens** à travers des capteurs (mouvement, ouverture, fumée, Gas...) et des moyens d'alerte (sirènes, notifications, appels...).
- **La surveillance centralisée** et la coordination des différents éléments du système (capteurs, contrôleurs, interfaces utilisateur) via un microcontrôleur (ESP32, Arduino, etc.) et des protocoles IoT (MQTT, HTTP...).
- **L'automatisation des réponses** : comme la coupure automatique d'une vanne de Gas ou le verrouillage à distance d'une porte où pour limiter le délai de réaction et réduire les dommages potentiels [1].



Figure I.1. Différents systèmes de sécurité d'une maison

I.3. Importance de la sécurité dans les habitations modernes

La modernisation des habitations s'accompagne d'une densification des équipements électroniques, d'une interconnexion croissante des objets et d'une mobilité accrue des occupants. Dans ce contexte, la **sécurité** apparaît comme un enjeu central pour garantir :

- **La protection des personnes** : assurer la sûreté des résidents face aux intrusions, incendies, fuites de Gas ou défaillances électriques.
- **La préservation des biens** : éviter les vols, les dégradations ou les dommages matériels liés à des incidents domestiques.
- **La continuité de service** : maintenir un environnement habitable (électricité, chauffage, ventilation) et éviter les coupures longues dues à un sinistre.
- **La quiétude et le confort** : permettre aux occupants de se sentir en sécurité, même à distance, via une supervision et des alertes en temps réel.
- **La valorisation immobilière** : un logement équipé d'un dispositif de sécurité intelligent est perçu comme plus attractif sur le marché, notamment par les locataires ou acheteurs soucieux de leur bien-être [2].



Figure I.2. Modernisation des habitations

L'intégration de solutions domotiques et IoT dans les systèmes de sécurité permet non seulement de détecter précocement les anomalies, mais aussi de réagir automatiquement (coupure de Gas, déclenchement d'une alarme, verrouillage à distance) et de fournir des rapports d'activité détaillés [3].

I.4. Classification des risques domestiques

Les habitations modernes sont exposées à divers risques, qu'il est essentiel d'identifier pour concevoir un système de sécurité adapté :

- **Intrusions et cambriolages** : forçage de serrures, effraction de portes ou fenêtres, déplacement illégal dans le logement.
- **Incendies domestiques** : départ de feu lié à un dysfonctionnement électrique, à un appareil de cuisson ou à un défaut d'installation de chauffage.
- **Fuites de Gas** : émanation de Gas naturel ou de GPL due à une canalisation endommagée ou à une vanne mal fermée, créant un risque d'asphyxie et d'explosion [4].
- **Inondations et dégâts des eaux** : rupture de canalisation, remontée d'eaux pluviales ou débordement d'appareils sanitaires entraînant des dommages matériels importants.
- **Défaillances électriques** : surtensions, court-circuit ou isolations défectueuses pouvant provoquer des incendies ou des électrocutions [5].
- **Présence de monoxyde de carbone** : dégagement de ce Gas inodore et mortel par des appareils de combustion mal entretenus (chaudière, chauffe-eau).



Figure I.3. Différents risques domestiques

I.5. Classification des systèmes de sécurité

I.5.1. Systèmes de sécurité traditionnels

Les systèmes de sécurité traditionnels reposent principalement sur des dispositifs mécaniques ou électromécaniques, sans connectivité ni automatisation avancée. Ils offrent une protection de base, mais présentent des limites importantes en termes de surveillance à distance, d'analyse des données et de réactivité [6].

- **Serrures mécaniques et verrous renforcés** : principal moyen de protection contre les intrusions, reposant sur la résistance du cylindre et du pêne. Leur efficacité dépend de la qualité du matériau et de l'installation.
- **Portes et fenêtres blindées** : renforcement structurel des points d'accès pour retarder l'effraction, sans système d'alerte intégré.
- **Alarme filaire à déclenchement local** : capteurs d'ouverture ou de bris de vitre connectés par fils à une centrale qui déclenche une sirène interne ou externe ; ne permet ni l'envoi de notifications à distance ni l'enregistrement des événements.
- **Barrières et grilles de sécurité** : installations fixes pour dissuader l'entrée non autorisée, souvent utilisées en complément des serrures ; coût élevé et impact esthétique non négligeable.
- **Services de gardiennage** : présence humaine ou animale (chien de garde, veilleur) pour surveiller l'habitation ; coûteux et limité à la présence physique des intervenants.



Figure I.4. Systèmes de sécurité traditionnels

I.5.2. Caractéristiques des systèmes traditionnels

- **Absence de supervision à distance** : tout déclenchement d'alarme reste local, sans information en temps réel pour le propriétaire absent.
- **Pas de corrélation entre capteurs** : chaque détecteur fonctionne en silo, sans analyse croisée des événements.
- **Maintenance manuelle** : les vérifications et la remise en service après déclenchement sont effectuées sur place.
- **Évolutivité réduite** : difficile d'ajouter de nouveaux capteurs ou de modifier le comportement de façon logicielle [7].

I.5.3. Limites des approches classiques

Les systèmes de sécurité traditionnels, bien que robustes, souffrent de plusieurs lacunes majeures pour répondre aux exigences des habitations modernes :

- **Communication locale uniquement**

Les dispositifs filaires ou radio basiques ne transmettent l'alerte qu'au travers d'une sirène ou d'un panneau de contrôle placé dans l'habitation, sans possibilité d'envoyer une notification vers un smartphone, un serveur Cloud ou un centre de télésurveillance externe.

- **Aucune automatisation des scénarios**

Chaque événement détecté (ouverture de porte, mouvement, fumée) déclenche uniquement une alarme locale ; il n'existe pas de logique embarquée pour couper automatiquement l'arrivée de Gas, verrouiller les accès, ou déclencher un éclairage de sécurité selon un scénario préprogrammé.

- **Interface utilisateur minimaliste**

Les systèmes classiques se limitent souvent à un simple clavier ou un petit écran LCD pour l'armement/désarmement ; ils ne proposent pas de tableau de bord centralisé, de rapports d'événements horodatés, ni de paramétrage fin des seuils de déclenchement depuis une application ou une interface web.

- **Absence de supervision et d'accès à distance**

En l'absence de connectivité réseau, le propriétaire ne peut ni vérifier l'état en temps réel de son système lorsqu'il est absent, ni armer/désarmer à distance ; toute modification ou diagnostic requiert un déplacement sur site [8].

- **Temps de réaction long**

En cas d'alerte, il faut généralement faire appel à un service de gardiennage, à un voisin ou aux forces de l'ordre ; le délai d'intervention peut être de plusieurs dizaines de minutes, multipliant les risques de vol, d'incendie ou de dégât des eaux avant arrivée des secours.

I.5.4. Évolution vers les systèmes de sécurité intelligents

L'émergence des systèmes de sécurité intelligents résulte de la convergence de plusieurs innovations technologiques ; elle répond aux limites des approches classiques en offrant :

- **Connectivité permanente**

Intégration de modules Wi-Fi, Bluetooth ou GSM permettant la supervision en temps réel et l'échange bidirectionnel de données entre le logement et l'utilisateur [9].

- **Architecture IoT et Cloud**

Hébergement des données et des règles de gestion sur des plateformes Cloud, facilitant l'analyse, la mise à jour et l'extension du système à distance.

- **Analyse et corrélation multi-capteurs**

Utilisation d'algorithmes embarqués (ou Edge compute) pour croiser les informations issues du PIR, du MQ-9, du KY-037 et du switch magnétique, réduisant les fausses alertes et améliorant la précision de la détection.

- **Automatisation avancée des scénarios**

Programmation de séquences d'actions conditionnelles : coupure de Gas en cas de détection simultanée de fuite et d'absence de mouvement, déclenchement d'un éclairage LED et enregistrement vidéo à la première intrusion détectée, etc.

- **Interface utilisateur riche et personnalisable**

Applications mobiles et portails Web offrant des tableaux de bord interactifs, des historiques d'événements horodatés, ainsi que des réglages de seuils et de notifications selon le profil de l'utilisateur [10].

I.6. Système de sécurité intelligent

Un système de sécurité intelligent est une solution intégrée qui s'appuie sur les principes de la domotique et de l'Internet des Objets pour assurer, de façon automatisée et coordonnée, la prévention, la détection et la gestion des menaces dans un environnement résidentiel ou professionnel.

Il combine une variété de capteurs connectés, un traitement local par microcontrôleur et une infrastructure Cloud, permettant la collecte continue de données, l'application de règles logiques et l'optimisation des réponses aux incidents. Grâce à cette architecture, l'utilisateur bénéficie d'une supervision à distance en temps réel, d'une maintenance prédictive et de la possibilité de faire évoluer dynamiquement les scénarios de sécurité en fonction des besoins et du contexte [11].

I.6.1. Avantages des systèmes connectés et autonomes

Les systèmes de sécurité modernes, reposant sur des architectures IoT et des traitements embarqués, présentent plusieurs bénéfices clés par rapport aux solutions traditionnelles :

- **Surveillance et réaction en temps réel**

Les données issues des capteurs sont transmises instantanément au Cloud ou à une application mobile, permettant une détection rapide et une réponse immédiate en cas d'anomalie.

- **Contrôle à distance généralisé**

L'utilisateur peut armer, désarmer ou modifier les paramètres du système depuis n'importe où, sans déplacement, via une interface web ou mobile sécurisée.

- **Automatisation sophistiquée**

La possibilité de déployer des scénarios conditionnels et des logiques d'action réduit la charge manuelle et assure une réaction cohérente et adaptée aux événements.

- **Réduction des fausses alertes**

La corrélation des données multi-capteurs (mouvement, Gas, son, ouverture) et l'apprentissage des comportements permettent de distinguer plus efficacement les incidents réels des perturbations sans danger.

- **Développement et flexibilité**

L'ajout de nouveaux capteurs ou services (vidéo-surveillance, gestion énergétique) s'effectue sans refonte de l'infrastructure, grâce à l'architecture modulaire et aux protocoles standards (MQTT...) [12].

I.6.2. Domaines d'application des systèmes de sécurité intelligents

Les systèmes de sécurité intelligents trouvent aujourd'hui des usages variés, grâce à leur flexibilité, leur connectivité et leur capacité d'automatisation :

- **Habitations résidentielles**

Dans un cadre domestique, les systèmes de sécurité intelligents assurent une surveillance continue (24 h/24, 7 j/7) en collectant et analysant en temps réel les données issues de capteurs de mouvement, d'ouverture ou de fumée. En cas d'anomalie, ils déclenchent immédiatement des alertes locales (sirènes) et distantes (notifications sur smartphone), tout en enregistrant un historique horodaté des événements.

La gestion à distance permet à l'utilisateur de consulter l'état de son système, d'armer ou désarmer la protection, et d'ajuster les paramètres (seuils, heures d'activation) via une application mobile ou un portail Web. L'intégration à la domotique de la maison facilite la coordination avec d'autres services (éclairage, chauffage, volets), renforçant la sécurité et le confort au quotidien [13].

- **Bureaux et entreprises**

Dans un environnement professionnel, les dispositifs de sécurité intelligents offrent un contrôle d'accès granularité (badges, codes, biométrie) et la supervision des zones sensibles (salles de serveurs, laboratoires).

Les flux de personnes sont suivis en temps réel, et toute anomalie (tentative d'accès non autorisé, présence hors horaires définis) déclenche des alertes centralisées vers un poste de sécurité ou un service de télésurveillance.

Les administrateurs peuvent générer des rapports d'activité détaillés et configurer des scénarios automatiques (verrouillage des accès, déclenchement de caméras) pour répondre rapidement aux incidents, garantissant la confidentialité et la continuité des opérations de l'entreprise [14].



Figure I.5. Sécurité dans les Bureaux et entreprises

- **Établissements de santé**

Les hôpitaux et cliniques requièrent un niveau de sécurité renforcé pour protéger à la fois les personnes et des équipements critiques (pharmacies, blocs opératoires, salles d'imagerie). Les systèmes intelligents contrôlent l'accès par profils d'utilisateurs (personnel médical, visiteurs, techniciens), enregistrent les passages et signalent immédiatement toute tentative d'intrusion ou d'accès non autorisé. Ils peuvent aussi surveiller l'état des patients (détection de chutes, chambres verrouillées) et déclencher des alertes médicales. L'historisation des accès et l'interface de gestion centralisée renforcent la traçabilité et la coordination entre services, tout en respectant les contraintes réglementaires en vigueur dans le secteur de la santé [15].



Figure I.6. Sécurité dans les établissements de santé

- **Bâtiments publics et collectivités**

Dans les écoles, administrations et lieux culturels, les systèmes de sécurité intelligents assurent un contrôle d'accès rigoureux (cartes d'identité, badges, systèmes biométriques) pour différencier visiteurs, personnel et intervenants externes, tout en conservant une traçabilité des passages. La vidéosurveillance exploitée via des caméras connectées permet une couverture en continu des zones critiques, avec fonctions de détection d'anomalies (mouvements suspects, attroupements) et enregistrement horodaté pour les enquêtes ultérieures. En cas d'événements exceptionnels intrusion, incendie, malveillance, le système déclenche immédiatement des alertes centralisées vers le service de sécurité interne ou les services d'urgence, tout en affichant des plans d'évacuation et en coordonnant l'ouverture des issues de secours pour optimiser la réponse et garantir la sûreté des usagers [16].

I.7. Technologies utilisées dans la sécurité intelligente

I.7.1. Internet des Objets (IoT)

L'Internet des Objets (IdO), souvent abrégé IoT (*Internet of Things*), désigne l'ensemble des objets physiques connectés à Internet et capables de collecter, transmettre, recevoir et parfois traiter des données, sans intervention humaine directe. Ces objets, appelés dispositifs intelligents, intègrent des capteurs, des microcontrôleurs, des modules de communication et parfois des actionneurs. Grâce à leur connectivité, ils peuvent interagir entre eux ou avec des plateformes centralisées, permettant une surveillance en temps réel, une prise de décision automatisée et une optimisation des processus dans divers domaines.

L'IoT repose sur une infrastructure numérique interconnectée, combinant le matériel (capteurs, réseaux, serveurs), les logiciels (applications, algorithmes, intelligence artificielle) et les réseaux de communication (Wi-Fi, 4G/5G, Bluetooth, etc.). Il s'applique à de nombreux secteurs : santé, agriculture, industrie, transport, domotique, et bien sûr, la sécurité [17].

I.7.2. Composant de l'internet des objets

Comme on a dit précédemment l'internet des objets permet la connectivité des différents objets via un réseau pour fonctionner efficacement et à distance, pour faire cette connectivité on a besoin de :

- **Objet physique** : c'est un objet permet de transcender des nouveaux services en lui fournissant des capteurs.

- **Capteurs** : ils sont installés sur l'objet connecté pour capter toutes les informations nécessaires sur cet objet. Les capteurs connus sont : Capteurs de température et thermostats, Capteurs de pression, Capteurs de Gas, Détecteurs d'intensité lumineuse.

- **Plateforme** : elle est considérée comme un type d'inter-logiciel utilisé pour connecter les composants IoT (objets, personnes, services, etc.) à l'environnement d'IoT. Elle fournit de nombreuses fonctions : accès aux appareils, assurer une installation / un comportement correct

de l'appareil, analyse des données et mieux connexion avec le réseau local, le Cloud ou d'autres périphériques.

- Réseaux : les composants IoT relie entre eux et avec le serveur par un réseau.
- Prestations de service : peuvent être utilisés pour traiter les Big Data et les transformer en informations précieuses, construire et exécuter des applications innovantes et optimiser les processus métier en intégrant les données de l'appareil [18].

I.7.3. Fonctionnement de l'internet des objets

Un système IdO comprend des capteurs/dispositifs qui communiquent avec le Cloud via une certaine forme de connectivité. Une fois que les données atteignent le Cloud, le logiciel les traite et décide d'effectuer une action, telle que le réglage des capteurs/dispositifs, sans l'intervention humaine. Un système IdO est composé de quatre éléments différents :

- Capteurs/Appareils : Sert à collecter des données à partir de leur environnement. Un appareil peut contenir plusieurs capteurs : par exemple, un smartphone contient un GPS, un appareil photo, un accéléromètre, etc.

- Connectivité : Une fois que la collecte des données est effectuée, l'appareil doit les envoyer dans le Cloud. Il le fait de différentes manières : Wi-Fi, Bluetooth, satellite, réseaux étendus à faible puissance (LPWAN) ou connexion directe à Internet via Ethernet. L'option de connectivité spécifique dépendra de l'application IoT.

- Traitement des données : Une fois que les données ont atteint le Cloud, le logiciel les traite et peut décider d'effectuer une action. Il peut s'agir d'envoyer une alerte ou de régler automatiquement les capteurs ou l'appareil sans intervention de l'utilisateur. Cependant, l'utilisateur doit parfois intervenir, et c'est là que l'interface utilisateur entre en jeu.

- Interface utilisateur : Si l'utilisateur a besoin d'intervenir ou s'il veut vérifier le système, une interface utilisateur lui permet de le faire. Toute action effectuée par l'utilisateur est renvoyée en sens inverse par le système. De l'interface utilisateur au Cloud et retour aux capteurs/appareils pour effectuer la modification demandée. Les protocoles précis de connectivité, de mise en réseau et de communication utilisés par les appareils connectés au Web varient avec chaque application IoT. De plus en plus, l'IoT utilise l'intelligence artificielle (IA) et l'apprentissage automatique pour faciliter et accélérer les processus de collecte de données [19].

I.7.4. Internet des Objets (IoT) dans la sécurité domestique

L'Internet des Objets (IoT) et la sécurité entretiennent une relation de plus en plus étroite dans le contexte des technologies modernes. L'IoT transforme profondément les méthodes traditionnelles de protection des personnes, des biens et des infrastructures, en rendant les systèmes de sécurité plus intelligents, autonomes et interconnectés. Cette convergence repose

sur l'intégration de capteurs intelligents, de microcontrôleurs, de plateformes de traitement de données et d'interfaces utilisateur accessibles à distance.

L'IoT apporte à la sécurité une capacité de détection en temps réel des menaces ou incidents (intrusion, fuite de Gas, incendie, ouverture non autorisée, etc.) grâce à des dispositifs déployés sur le terrain et connectés à Internet. Ces capteurs transmettent les informations collectées à un système central (souvent un microcontrôleur comme l'ESP32) qui les analyse et déclenche, si nécessaire, des actions automatisées : activation d'une alarme, verrouillage de portes, notification immédiate à un utilisateur, ou enregistrement vidéo.

En retour, la sécurité fournit un cadre fonctionnel et organisationnel dans lequel l'IoT peut être déployé efficacement. Elle impose des exigences telles que la fiabilité, la flexibilité aux attaques informatiques, la protection des données personnelles, et la réactivité en cas d'urgence. Par conséquent, tout système IoT de sécurité doit non seulement être opérationnel sur le plan technique, mais aussi conforme à des normes strictes de cybersécurité, pour éviter les intrusions numériques ou la manipulation malveillante à distance.

La relation entre IoT et sécurité est donc bidirectionnelle et complémentaire : l'IoT renforce les capacités des dispositifs de sécurité en leur donnant une intelligence distribuée, tandis que la sécurité structure l'utilisation de l'IoT pour qu'elle soit fiable, efficace et protectrice des utilisateurs. Dans les habitations, les entreprises, les espaces publics et les villes intelligentes, cette synergie favorise une surveillance active, une prévention accrue des incidents, et une gestion centralisée des alertes, tout en permettant une supervision à distance et une personnalisation des scénarios de réponse [20].

I.8. Conclusion

Ce premier chapitre a posé les bases théoriques indispensables à la compréhension des systèmes de sécurité domestique intelligente. Nous avons défini les concepts clés de la sécurité résidentielle et la typologie des risques (intrusion, fuites de Gas, mouvement, ouverture non autorisée), comparé les approches traditionnelles et intelligentes, et souligné le rôle central de l'Internet des Objets (IoT) dans l'évolution des dispositifs de protection. Nous avons également présenté les modèles de communication et les principes d'interopérabilité qui sous-tendent la collecte, le traitement et la diffusion des alertes en temps réel.

Fort de ce socle, nous sommes désormais en mesure d'aborder, dans le chapitre 2, la conception concrète de notre solution : choix des composants (ESP32-WROOM-32, capteurs MQ-9, PIR, switch magnétique, détecteur de bruit), architecture fonctionnelle, et description des protocoles de communication et de l'application mobile de supervision.

Bibliographie chapitre 1

[1] Otmani, Yacine. *Simulation d'un réseau domotique à base d'Internet des Objets*. Mémoire de Projet de Fin d'Études, Université Badji Mokhtar – Annaba, 2021. <https://biblio.univ-annaba.dz/ingeniorat/wp-content/uploads/2022/04/PFE-OTMANI-YACINE-2021.pdf>

[2] S. Panda, "Importance of Information Security Awareness," LinkedIn, 2014. [En ligne]. Disponible sur : <https://www.linkedin.com/pulse/importance-security-creating-awareness-highlighting#:~:text=Security%20measures%20enhance%20safety%2C%20prevent,for%20everyone%20to%20thrive%20in>. (consulté en mai 2025).

[3] ABGi France, "L'Internet des Objets au service de la domotique : la technologie révolutionne nos maisons pour les rendre plus écologiques et intelligentes," ABGi, 29 avril 2025. [En ligne]. Disponible sur : <https://abgi-france.com/internet-des-objets-au-service-de-la-domotique/> (consulté en mai 2025).

[4] "Gas naturel : quels dangers ? Les risques directs et indirects," Connaissance des Énergies, [En ligne]. Disponible sur : <https://www.connaissancedesenergies.org/fiche-pedagogique/Gas-naturel-quels-dangers> (consulté en mai 2025).

[5] INRS, "Accidents d'origine électrique : causes, prévention et risques," INRS, [En ligne]. Disponible sur : <https://www.inrs.fr/risques/electriques/accidents-origine-electrique.html> (consulté en mai 2025).

[6] "Système de sécurité électronique : définition, composants et installation," *Samsic Facility*, [En ligne]. Disponible sur: <https://www.samsic.fr/services/securite/securite-electronique/systeme/> (consulté en mai 2025).

[7] "Systèmes de sécurité électroniques : types et composants," *Electrolomas*, [En ligne]. Disponible sur: <https://electrolomas.com/fr/systemes-de-securite-electroniques-types-et-composants/> (consulté en mai 2025).

[8] Splashtop, "What is Unattended Access?," *Splashtop Blog*, [En ligne]. Disponible sur : https://www.splashtop.com/fr/blog/whatisunattendedaccess?srsltid=AfmBOoqbX8O9zNg7xhqbh4Cjsd1cITeGAXJCJ2WDAG-NRrLpvtl20_MZ (consulté en mai 2025).

[9] TechTarget, "GSM (Global System for Mobile Communications)," *SearchMobileComputing*, [En ligne]. Disponible sur : <https://www.techtarget.com/searchmobilecomputing/definition/GSM> (consulté en mai 2025).

[10] Aquilapp, "Qu'est-ce qu'une application mobile ?," *Aquilapp Ressources*, [En ligne]. Disponible sur : <https://www.aquilapp.fr/ressources/projet-mobile/quest-ce-quune-application-mobile> (consulté en mai 2025).

[11] « Domotique », Wikipédia, <https://fr.wikipedia.org/wiki/Domotique> (consulté en mai 2025) [Wikipédia](#)

[12] “Maison intelligente et connectée : avantages et inconvénients,” *Le Designer Français*, [En ligne]. Disponible : <https://www.ledesignerfrancais.com/conseil/maison-intelligente-et-connectee-avantages-et-inconvenients/> (consulté en mai 2025).

[13] TD Assurance, “Surveillance résidentielle,” *TD Assurance – Assurance habitation*, [En ligne]. Disponible : <https://www.tdassurance.com/produits-et-services/assurance-habitation/infocentre/surveillance-residentielle> (consulté en mai 2025).

[14] « Internet des objets », Wikipédia, https://fr.wikipedia.org/wiki/Internet_des_objets (consulté en mai 2025).

[15] « Hôpital », Wikipédia, <https://fr.wikipedia.org/wiki/H%C3%B4pital> (consulté en mai 2025).

[16] Wikipédia. *École ; Administration publique ; Vidéosurveillance ; Incendie*. Disponible sur : <https://fr.wikipedia.org> (consulté en mai 2025).

[17] Internet des objets – Wikipédia. Disponible sur : https://fr.wikipedia.org/wiki/Internet_des_objets (consulté en mai 2025).

[18] Ephikas, “SP IoT – Solutions et services pour l’Internet des Objets,” *Ephikas*, [En ligne]. Disponible : <https://www.ephikas.fr/sp%20iot.htm> (consulté en mai 2025).

[19] “Introduction à l’IoT : comment fonctionne l’Internet des Objets ?,” *Cellenza Blog*, [En ligne]. Disponible : <https://blog.cellenza.com/data/introduction-a-liot-comment-fonctionne-linternet-des-objets/> (consulté en mai 2025).

[20] Wikipédia. *Internet des objets*. Disponible sur : https://fr.wikipedia.org/wiki/Internet_des_objets (consulté en mai 2025).

CHAPITRE II

Chapitre II

Composantes matérielles d'un système de sécurité intelligent

II.1. Introduction

La conception est une étape importante qui précède toute étape de réalisation d'un système. La conception d'un système de sécurité intelligent pour une maison repose essentiellement sur l'analyse des besoins, par ailleurs, une évaluation des risques est indispensable pour identifier les zones sensibles de la maison et de définir les différents types de sécurité à examiner. Elle combine des composantes matérielles et des composantes logicielles pour offrir une protection efficace, automatisée et connectée. Les composantes matérielles d'un système de sécurité intelligent constituent l'infrastructure physique essentielle permettant la détection, l'analyse et la réponse aux menaces potentielles. Cependant, les composantes logicielles analysent en temps réel les flux de données provenant de capteurs, assurent l'échange d'informations entre différentes composantes et peuvent automatiser des réponses.

Ce chapitre est consacré à l'étude de l'ensemble des composantes matérielles utilisées pour la réalisation d'un système de sécurité domestique intelligent. Il décrit les différents capteurs choisis pour détecter les situations critiques : capteur de Gas pour les fuites, capteur PIR pour les mouvements, capteur sonore pour les bruits anormaux, et interrupteur magnétique pour l'ouverture ou la fermeture d'accès. Il détaille également les actionneurs intégrés au système, tels que les LEDs pour les alertes visuelles, le buzzer pour les signaux sonores et le servo-moteur pour les actions mécaniques. Enfin, le chapitre met en avant le rôle central du microcontrôleur ESP32-WROOM-32, qui assure la collecte, le traitement des données et la communication à distance grâce à sa connectivité Wi-Fi.

II.2. Microcontrôleur ESP32-WROOM-32

II.2.1. Historique

L'ESP32 est un microcontrôleur développé par la société chinoise Espressif Systems. Il s'agit de la deuxième génération de microcontrôleurs WiFi et Bluetooth de l'entreprise, succédant à l'ESP8266. L'ESP32 a été annoncé pour la première fois en septembre 2016 et est entré en production en décembre 2016. Le développement de l'ESP32 a débuté en 2013, avec l'objectif de créer un microcontrôleur à faible consommation d'énergie avec des fonctionnalités avancées de connectivité sans fil, telles que le WiFi et le Bluetooth. Espressif Système a également cherché à améliorer les performances de traitement du microcontrôleur, en intégrant un processeur plus puissant que celui de l'ESP8266. Le développement de l'ESP32 a été

largement basé sur l'expérience acquise avec l'ESP8266. L'ESP8266 avait été un grand succès pour Espressif System en raison de sa faible consommation d'énergie, de sa connectivité WiFi et de son faible coût, ce qui avait permis de développer de nombreux projets IoT. Cependant, il avait également quelques limitations, telles que des performances de traitement limitées et une capacité de stockage limitée.

Avec l'ESP32, Espressif System a cherché à résoudre ces problèmes en intégrant un processeur double cœur, une plus grande quantité de mémoire flash et de SRAM, ainsi que des fonctionnalités de connectivité WiFi et Bluetooth améliorées. L'ESP32 a également introduit de nouvelles fonctionnalités, telles que des interfaces Ethernet, des interfaces de caméra et des capacités de traitement de signal numérique avancées. Depuis sa sortie, l'ESP32 est devenu un choix populaire pour les projets IoT, les appareils connectés et d'autres applications nécessitant des fonctionnalités de connectivité sans fil et une faible consommation d'énergie. Espressif Systems continue de mettre à jour et d'améliorer l'ESP32, avec des mises à jour régulières du firmware et des bibliothèques logicielles pour améliorer les performances et ajouter de nouvelles fonctionnalités [1].

II.2.2. Différents types de microcontrôleur ESP32

Il existe de nombreux types de cartes ESP32 disponibles sur le marché, chacune avec ses propres spécifications et fonctionnalités. On cite :

- ESP32-WROOM-32 : Cette carte est une puce système intégrée qui combine un processeur à double cœur à 32 bits, une connectivité WiFi et Bluetooth, des interfaces Ethernet, des interfaces de caméra et des capacités de traitement de signal numérique avancées. C'est la référence de l'ESP32.

- ESP32 DevKitC : Cette carte de développement est équipée d'un connecteur micro-USB pour la programmation et le débogage, ainsi que d'un module ESP32- WROOM-32 intégré. Elle est compatible avec l'IDE Arduino et MicroPython.

- NodeMCU-32S : Cette carte de développement dispose d'un module ESP32-WROOM-32 intégré, ainsi que d'un support pour la connectivité WiFi, Bluetooth et Ethernet. Elle est également compatible avec l'IDE Arduino et MicroPython.

- ESP32-PICO-KIT : Cette carte de développement compacte est équipée d'un module ESP32-PICO, qui est une puce système intégrée similaire à l'ESP32- 23 WROOM-32. Elle dispose d'une interface micro-USB pour la programmation et le débogage, ainsi que d'un ensemble de broches GPIO pour l'interface avec d'autres périphériques.

- ESP32-S2-Kaluga-1 : Cette carte de développement est basée sur l'ESP32-S2, qui est une version plus récente de l'ESP32 avec des fonctionnalités de connectivité améliorées. Elle dispose d'un écran LCD couleur intégré, d'un capteur de lumière ambiante, d'un capteur de

température et d'humidité, ainsi que de diverses interfaces de connectivité telles que le WiFi, Bluetooth et Ethernet [1].

II.2.3. Définition de l'ESP32-WROOM-32

Le microcontrôleur ESP32-WROOM-32 est un module de communication sans fil développé par Espressif Systems, conçu pour les systèmes embarqués et les applications IoT (Internet of Things). Le cœur du module ESP32-WROOM-32 est le SoC (System on Chip) ESP32-D0WDQ6, qui est un microcontrôleur à double cœur basé sur l'architecture Xtensa LX6 32 bits, capable de fonctionner jusqu'à 240 MHz. Ce module combine des fonctionnalités avancées de connectivité Wi-Fi 802.11 b/g/n et Bluetooth v4.2 (Classic et BLE), ce qui lui permet de servir de passerelle entre les capteurs, les actionneurs et les réseaux distants [2].

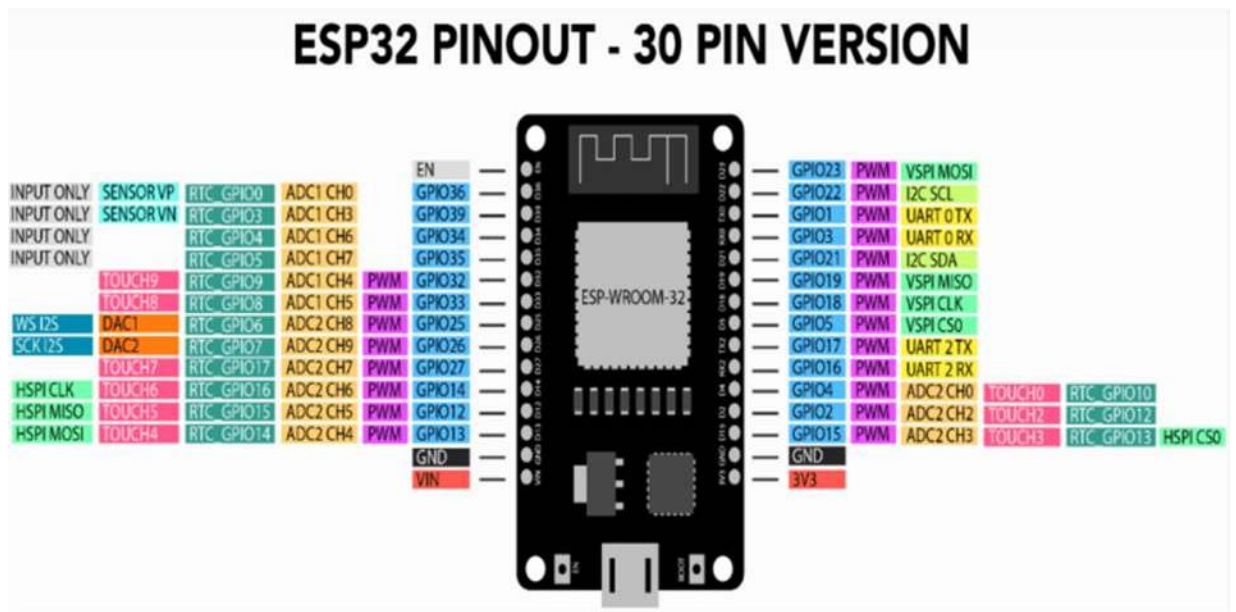


Figure II.1. Brochage de la carte ESP32-WROOM.

En plus de sa puissance de calcul, le module dispose d'une mémoire intégrée (ROM, SRAM, Flash) et de nombreuses interfaces de communication (UART, SPI, I2C, PWM, ADC, DAC, etc.) facilitant l'interfaçage avec une large gamme de composants électroniques. Grâce à sa compacité, sa faible consommation énergétique et sa flexibilité logicielle (ESP-IDF, Arduino, MicroPython), il est largement utilisé dans des domaines tels que la domotique, les systèmes de sécurité, la télésurveillance, ou encore les dispositifs portables intelligents.

II.2.4. Caractéristiques techniques

- Microcontrôleur : Processeur dual-core Tensilica Xtensa LX6 (fréquence d'horloge de 240 MHz).
- Connectivité sans fil : Connectivité Wi-Fi intégrée 802.11 b/g/n. Connectivité Bluetooth intégrée (Bluetooth Low Energy, BLE).

- Mémoire : 520 Ko de SRAM, 448 Ko de ROM (mémoire de programme). Support en option pour une mémoire flash externe SPI.
- Périphériques : Périphériques Pins GPIO (Entrée/Sortie à usage général). UART, SPI, I2C, I2S, ADC, DAC. PWM (Modulation de Largeur d'Impulsion). Capteur à effet Hall.
- Sécurité : Fonctionnalités de sécurité basées sur le matériel, y compris le démarrage sécurisé et le chiffrement de la mémoire flash.
- Gestion de l'alimentation : Modes de faible consommation multiples pour une efficacité énergétique. Régulateur programmable pour l'alimentation du cœur.
- Tension de fonctionnement : 2.2V à 3.6V.
- Température de fonctionnement : "-40°C à +85°C"
- Facteur de forme : Module compact avec des bords à créneaux pour une soudure facile.
- Horloges et minuteries : Horloge temps réel (RTC) intégrée. Minuteries et watchdogs pour les applications sensibles au temps.
- Antenne intégrée : Antenne intégrée pour la communication Wi-Fi et Bluetooth.
- Certifications : Certifications FCC, CE, IC, et autres pour la conformité.
- Support de développement : Support de l'IDE Arduino pour la programmation. Support MicroPython. Kit de développement logiciel (SDK) fourni par Espressif Systems.
- Options d'emballage : Disponible dans différentes options d'emballage, y compris les populaires ESP32 WROOM 32D et ESP32 WROOM 32U.
- Dimensions : Les dimensions typiques du module sont d'environ 18 mm x 25,5 mm [3].

II.2.5. Interfaces de commination

UART (Universal Asynchronous Receiver-Transmitter) : Les broches dédiées à l'UART facilitent la communication série.

- UART 0 (TX/RX) : Numéro de pin 1 (TX), Numéro de pin 3 (RX)
- UART 1 (TX/RX) : Numéro de pin 17 (TX), Numéro de pin 16 (RX)

I2C (Inter-Integrated Circuit) : Des broches sont réservées pour la communication série à deux fils, permettant la connexion à des périphériques I2C.

- I2C (SDA/SCL) : Numéro de pin 21 (SDA), Numéro de pin 22 (SCL)

SPI (Serial Peripheral Interface) : Des broches sont attribuées à l'interface SPI pour la communication avec des dispositifs externes.

- SPI (MISO/MOSI/SCK/SS) : Numéro de pin 19 (MISO), Numéro de pin 23 (MOSI), Numéro de pin 18 (SCK), Numéro de pin 5 (SS).

II.2.6. Architecture interne

Un microcontrôleur est généralement composé de trois éléments :

- Unité centrale de traitement.
- Mémoires (mémoire morte et mémoire vive),
- Ports d'entrée et de sortie.

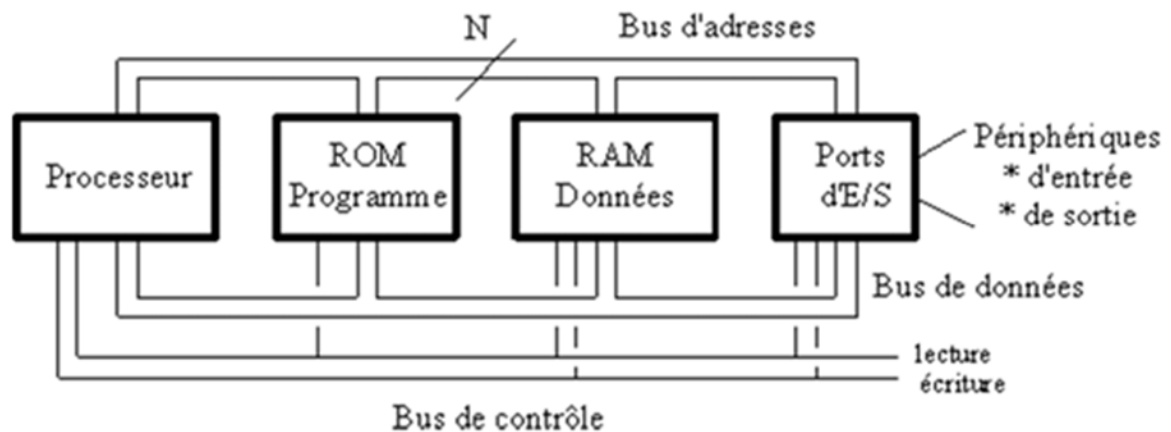


Figure II.2. Architecture interne d'une carte ESP32-WROOM-32.

Un microcontrôleur est un circuit électronique intégré qui comprend un processeur, une mémoire et des interfaces d'entrée/sortie. Il est programmable et est utilisé pour contrôler les opérations d'un système électronique. Les microcontrôleurs sont souvent utilisés dans des applications embarquées telles que les systèmes de contrôle de température, les systèmes d'éclairage, les systèmes de sécurité, les systèmes de surveillance et les appareils électroménagers. Les avantages des microcontrôleurs sont leur faible coût, leur faible consommation d'énergie et leur petite taille [1].

II.2.7. Principe de fonctionnement

Le principe de fonctionnement de la carte ESP32 est relativement simple. La carte est équipée d'un microcontrôleur ESP32, qui est un système sur puce (SoC) avec un processeur à double cœur cadencé à jusqu'à 240 MHz, une connectivité Wifi et Bluetooth intégrée, ainsi que de nombreuses interfaces périphériques pour les capteurs, les actionneurs et les autres composants électroniques. Le microcontrôleur ESP32 peut être programmé à l'aide de différents langages de programmation, tels que le langage Arduino, C++, Micro Python et d'autres langages de programmation. Les programmes peuvent être téléchargés sur la carte via un port USB ou via une connexion sans fil, telle que le Wifi ou le Bluetooth. Une fois programmée, la carte ESP32 peut alors contrôler différents périphériques électroniques, tels que des capteurs de température, des moteurs, des afficheurs, etc. Les données peuvent être échangées avec d'autres dispositifs via la connectivité Wifi ou Bluetooth intégrée. En résumé, la carte ESP32 permet de

développer des projets IoT avec une grande flexibilité et une grande puissance de traitement dans un format compact et facile à utiliser.

II.3. Capteurs

Un capteur électronique est un dispositif qui transforme une grandeur physique telle que la température, la pression, la lumière ou le mouvement en un signal électrique mesurable. Ce signal peut ensuite être utilisé pour contrôler un système ou pour enregistrer des données. Les capteurs électroniques sont largement utilisés dans de nombreux domaines, tels que l'automobile, la sécurité, la médecine, l'industrie et les télécommunications, pour mesurer des variables physiques et les convertir en données exploitables. Les exemples courants de capteurs électroniques incluent les capteurs de température, les capteurs de Gas, les capteurs de bruit, les capteurs de mouvement et les capteurs de position.

II.3.1. Capteur de Gas MQ-9

Un capteur de Gas est un dispositif qui détecte la présence d'un ou plusieurs types de Gas dans l'environnement. Ces capteurs ont de nombreuses applications, telles que les systèmes de sécurité des raffineries, des centres industriels et même des maisons. Ces capteurs peuvent détecter des Gas combustibles, des Gas toxiques, des Gas polluants, etc. Il existe plusieurs méthodes de détection des Gas, la plus couramment utilisée étant les capteurs électrochimiques. Ces capteurs mesurent la concentration d'un Gas spécifique en effectuant une réaction chimique sur leurs électrodes chauffées et en mesurant le courant électrique résultant [4].

La série de capteurs de Gas MQ est la plus courante disponible. Ces capteurs ont différents modèles pour détecter divers Gas, dont certains sont listés dans le tableau suivant :

Tableau II.1. Série de capteurs de Gas MQ.

Nom de capteur	Gas détectable
MQ-2	Méthane, Butane, LPG, Smoke
MQ-3	Alcool, Méthanol, Smoke
MQ-4	Méthane, CNG Gas
MQ-8	Hydrogène
MQ-9	Carbon monoxyde, Flammable Gas
MQ-131	Ozone
MQ-135	Air qualité
MQ-216	Natural Gas, Coal Gas
MQ-137	Ammonia

Le capteur de Gas MQ-9 appartient à la série MQ, qui comprend plusieurs capteurs spécialisés pour un Gas particulier. Dans le cas du capteur de Gas MQ-9, les Gas cibles sont le méthane, le propane et d'autres Gas combustibles qui peuvent représenter un danger mortel en cas de fuite. À mesure que la concentration ambiante des Gas cibles augmente, le capteur les absorbe avec diligence à un rythme plus élevé, transmettant leur présence par des valeurs analogiques sur sa broche analogique [5].

II.3.1.1. Caractéristiques techniques

- Gas détectés : CO, CH₄, LPG (Gas inflammable).
- Matériau sensible : SnO₂
- Tension d'alimentation (Vc) : 5,0 V ± 0,1 V DC
- Tension de chauffage (VH) : 5,0 V ± 0,1 V (haute T), 1,5 V ± 0,1 V (basse T)
- Courant de chauffage : 150 mA
- Plage de détection : 10 – 500 ppm (CO), 300–10 000 ppm (CH₄, LPG)
- Temps de réponse (T90) : ≤ 10 s
- Température de fonctionnement : -20 °C à +50 °C
- Sortie : Analogique 0 – 5 V
- Préchauffage initial : ≥ 24 h pour stabilisation

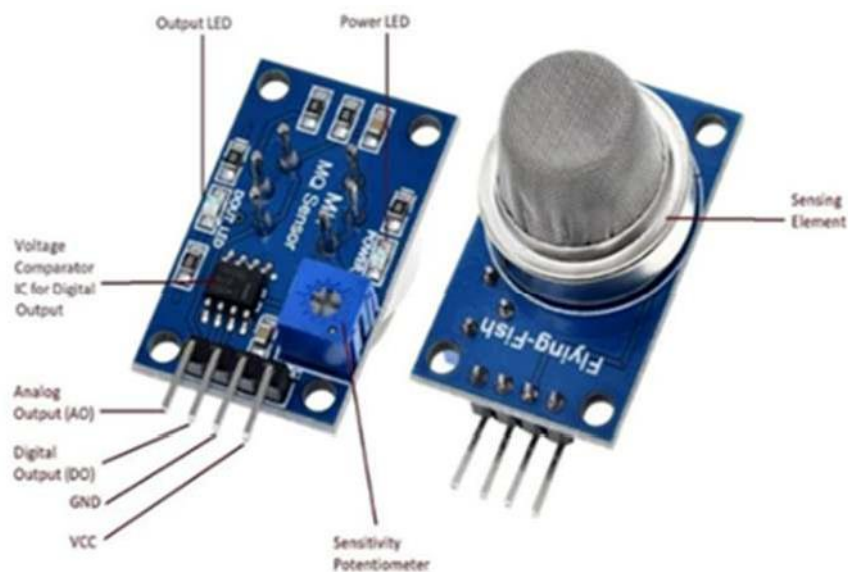


Figure II.3. Composants du capteur de Gas

- La résistance sensible change selon la concentration : en air pur, la résistance est élevée, elle diminue en présence de Gas réducteurs.

- Le module fournit généralement une sortie analogique linéaire en tension, facile à lire par un convertisseur analogique digital CAD (en anglais : ADC Analog Digital Converter) du microcontrôleur.

II.3.1.2. Principe de fonctionnement

1. Matériau sensible : une couche de SnO_2 dont la conductivité est faible en air propre et augmente en présence de Gas réducteurs.
2. Chauffage cyclique :
 - Haute température (5,0 V) : rapide détection de CO et régénération de la surface sensible en brûlant les polluants adsorbés.
 - Basse température (1,5 V) : détection sélective de CH_4 et LPG, évitant les interférences [6].
3. Variation de résistance : les molécules de Gas réducteurs réagissent avec l'oxygène adsorbé sur SnO_2 , libèrent des électrons dans le matériau, ce qui diminue la résistance.
4. Conversion en signal : un simple pont diviseur résistorique (ou circuit adapté) transforme cette variation de résistance en tension analogique proportionnelle à la concentration en Gas

II.3.1.3. Brochage de capteur MQ-9

Le capteur MQ-9 a quatre broches : VCC, GND, DO (sortie numérique), et AO (sortie analogique). Pour connecter le capteur à l'ESP32 :

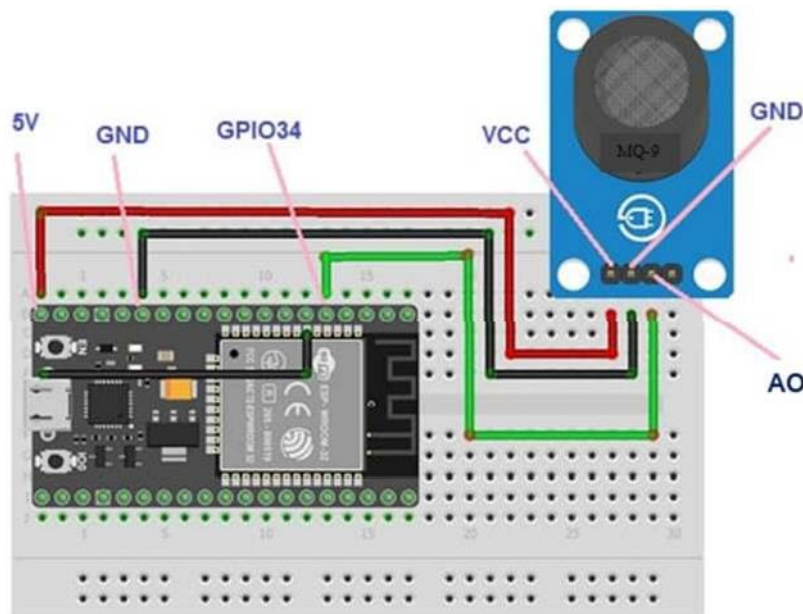


Figure II.4. Schéma de câblage de MQ-9 avec ESP32

VCC à 3.3V de l'ESP32.

GND à GND de l'ESP32.

AO à une entrée analogique (par exemple, 34) de l'Arduino pour lire les valeurs de concentration de Gas.

DO peut être connecté à une entrée numérique si vous voulez détecter un certain seuil de concentration de Gas [7].

II.3.2. Capteur de bruit KY-037

Le KY-037 est un module de détection sonore à haute sensibilité, équipé d'un microphone à électret, d'un amplificateur et d'un comparateur LM393. Il est conçu pour détecter les variations de niveau sonore dans l'environnement et fournir des sorties analogique et numérique exploitables par des microcontrôleurs tels qu'Arduino, ESP32 ou Raspberry Pi. Ce capteur est largement utilisé dans des projets de détection de sons, de commandes vocales, de systèmes d'alarme ou d'installations interactives [8].

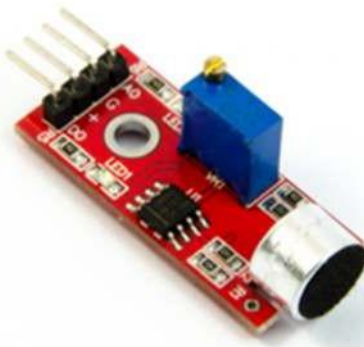


Figure II.5. Capteur de bruit KY-037

II.3.2.1. Caractéristiques techniques

- Tension d'alimentation : 3,3 V à 5 V DC
- Sorties : Analogique (AO) et numérique (DO)
- Microphone : Électret CMA-6542PF
- Comparateur : LM393
- Sensibilité : Ajustable via potentiomètre
- Consommation électrique : ~0,5 mA
- Fréquence de réponse : 50 Hz – 20 kHz
- Dimensions : 36 mm × 15 mm × 10 mm.

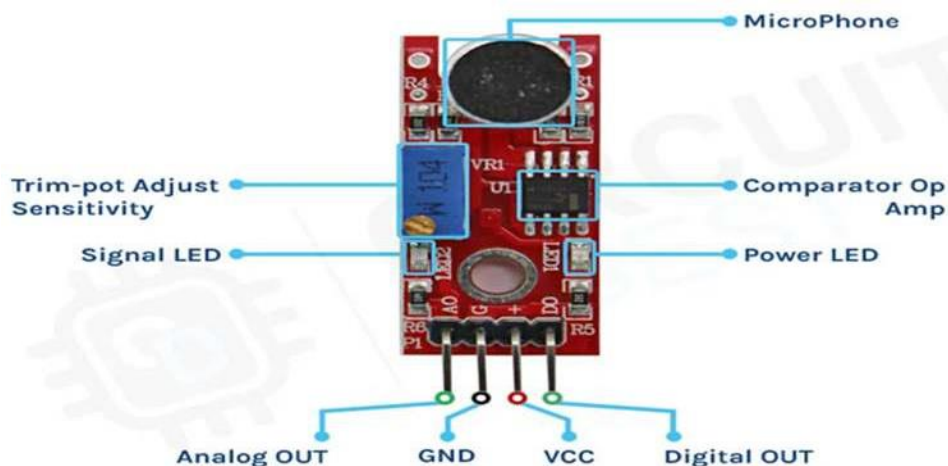


Figure II.6. Composants du capteur de bruit KY-037

II.3.2.2. Principe de fonctionnement

Le capteur KY-037 fonctionne selon le principe suivant :

- Détection sonore : Le microphone à électret capte les ondes sonores de l'environnement et les convertit en signaux électriques.
- Amplification : Le signal électrique est amplifié pour être exploitable par le comparateur et les sorties.
- Sortie analogique (AO) : Fournit une tension proportionnelle à l'intensité sonore détectée, permettant une mesure continue du niveau sonore.
- Sortie numérique (DO) : Le comparateur LM393 compare le signal amplifié à un seuil défini par le potentiomètre. Si le niveau sonore dépasse ce seuil, la sortie numérique passe à l'état haut (HIGH), sinon elle reste à l'état bas (LOW).
- **Indicateurs LED :**
 - **LED1** : Indique que le capteur est alimenté.
 - **LED2** : S'allume lorsque le niveau sonore dépasse le seuil défini [9].

II.3.2.3. Brochage de capteur KY-037

Le capteur de bruit KY-037 est connecté à la carte ESP32 selon le schéma suivant : la broche VCC du capteur au 3.3 V de l'ESP32. GND à la masse (GND).

la sortie numérique D0 est connectée à une entrée digitale (par exemple GPIO15) pour détecter les impulsions sonores.

la sortie analogique A0 peut être branchée à une entrée analogique telle que GPIO34 pour mesurer l'intensité du signal sonore.

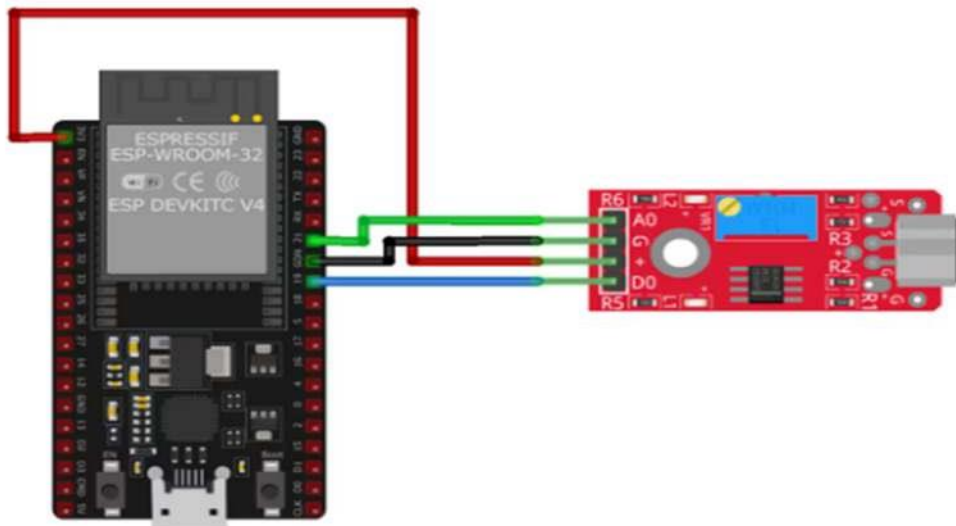


Figure II.7. Schéma de câblage du capteur KY-037 de bruit.

II.3.3. Capteur de mouvement PIR

Les détecteurs de mouvement PIR sont utilisés pour détecter les mouvements des humains et des animaux dans un rayon de 7 mètres. Ce modèle dispose d'un réglage de seuil de détection avant déclenchement de 0,3 à 18 secondes ainsi que d'un réglage de sensibilité [10].

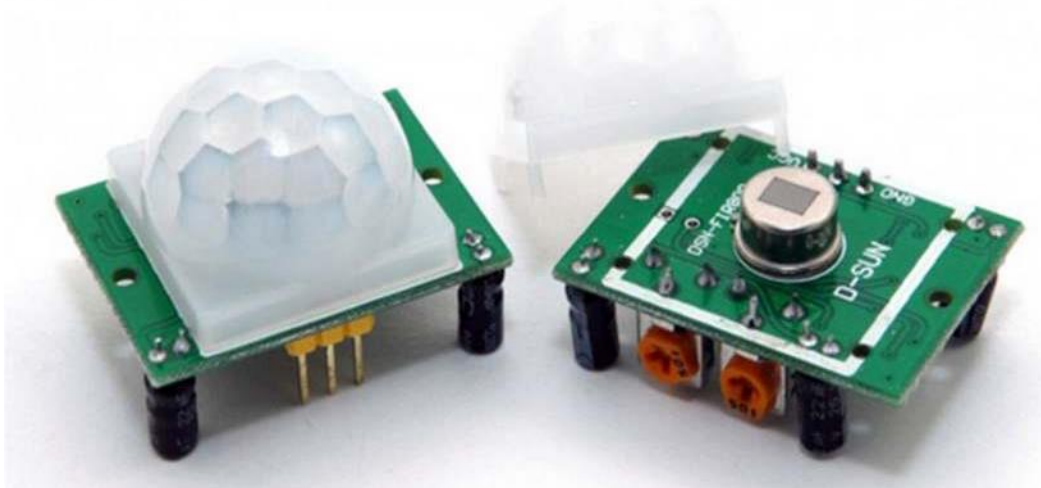


Figure II.8. Capteur de mouvement PIR

II.3.3.1. Caractéristiques techniques

On cite quelques caractéristiques du capteur PIR :

- Tension d'alimentation : 5-12V en fonction des modules (5V recommandé)
- Portée : Le module peut détecter un mouvement jusqu'à 7m
- Angle de mesure : ~120°
- Les Dimensions : 32mm * 24mm

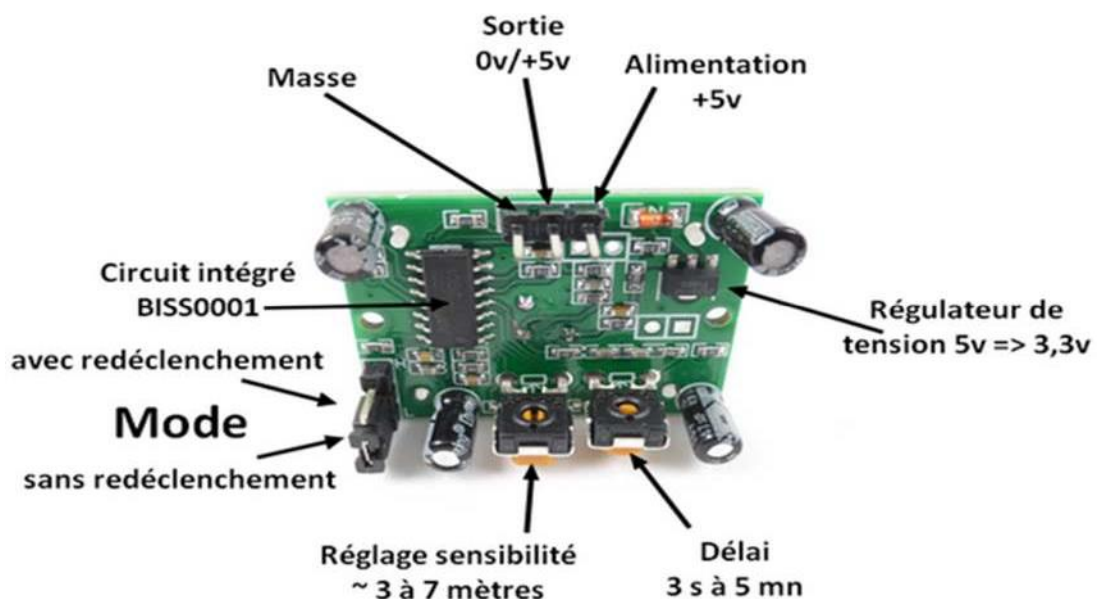


Figure II.9. Composant externe du capteur de mouvement

II.3.3.2. Principe de fonctionnement

Sous le dôme blanc en plastique, se trouve un capteur pyroélectrique qui détecte le rayonnement infrarouge émis par les objets environnants, c'est pourquoi il est appelé « capteur infrarouge passif ».

Il diffère des capteurs actifs qui émettent eux-mêmes un rayonnement (via une LED infrarouge) pour ensuite capter la partie transmise ou réfléchi (via une photorésistante par exemple).

Si un mouvement est détecté dans l'environnement du capteur, cela entraîne des variations du rayonnement infrarouge détecté. Le circuit électronique du module nous envoie ensuite l'information [11].

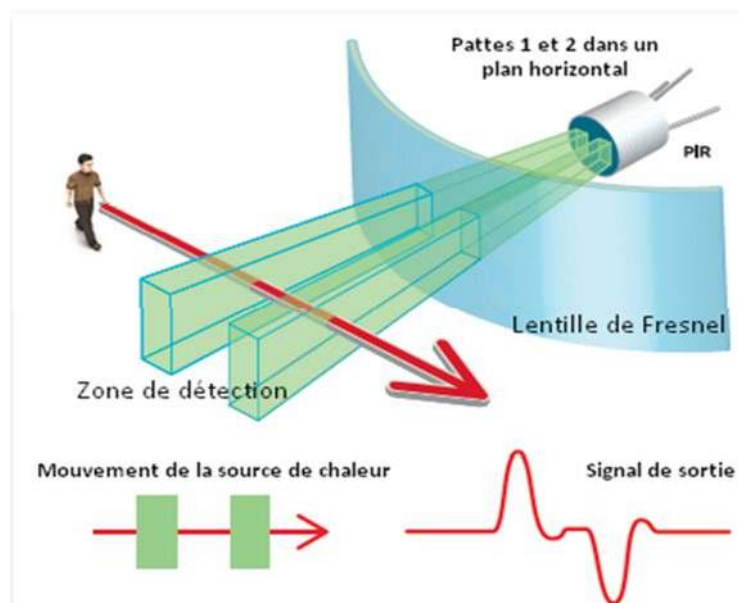


Figure II.10. Fonctionnement d'un détecteur de mouvement PIR

II.3.4. Switch

Un micro switch, appelé également commutateur à action rapide (snap action switch) ou micro rupteur, est un interrupteur très sensible qui permet de transmettre de faibles courants ou d'ouvrir un circuit pour stopper le fonctionnement d'une machine. Il est très souvent utilisé comme dispositif de sécurité.



Figure II.11. Switch NO/NC.

II.3.4.1. Caractéristiques techniques

Il se compose d'un actionneur (ex : un levier) et de 3 bornes (contacts) :

- COM pour « commun »,
- NO pour « Normalement Ouvert / Open »,
- NF ou NC pour « Normalement Fermé / Close ».

II.3.4.2. Principe de fonctionnement

L'alimentation en courant se fait via la borne commune (COM), qui va mettre un ressort sous tension. Le ressort touche la borne NF au repos et est relié à l'actionneur. Une fois l'actionneur enclenché, le ressort change de position et le courant est envoyé vers la borne NO [12].

II.4. Actionneurs

II.4.1. Servomoteur

Un servomoteur est un moteur ayant la capacité de maintenir une position à un effort statique. Si c'est en continu, la position du servomoteur est vérifiée, et en fonction de la mesure, elle est corrigée. Ainsi, le système du servomoteur est dit immobilisé ou motorisé du fait qu'il atteigne des positions prédéterminées.



Figure II.12. Servomoteur SG90.

II.4.1.1. Composant d'un servomoteur

Il est composé de plusieurs éléments visibles :

- Les fils, qui sont au nombre de trois (nous y reviendrons).
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal.
- Le boîtier qui le protège.

Mais aussi de plusieurs éléments que l'on ne voit pas :

- Un moteur à courant continu

- Des engrenages pour former un réducteur (en plastique ou en métal)
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent)
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

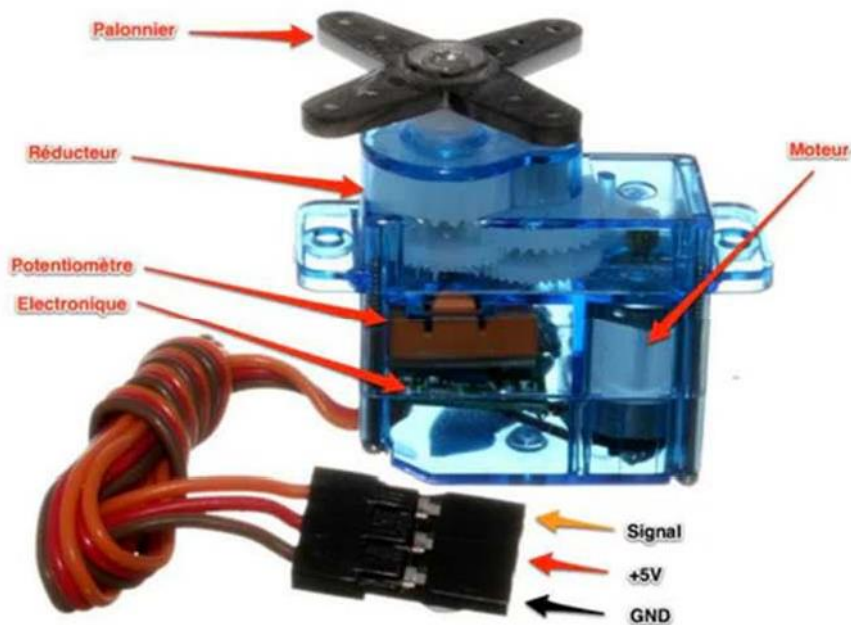


Figure II.13. Composants de servomoteur

II.4.1.2. Principe de fonctionnement

Un servomoteur se pilote par l'intermédiaire d'un câble à trois fils. Ce câble permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal. Contrairement à un moteur à courant continu (CC) simple, qui peut être piloté par des variations de tension, le servomoteur réagit en fonction d'une impulsion de durée variable. C'est la durée de ce signal qui détermine la rotation de l'axe donc la position de l'objet fixé dessus. L'impulsion détermine la position en absolu, pas en relatif : une durée précise correspond à une position précise de l'axe, toujours la même. Le signal doit être répété régulièrement (toutes les 20 ms).

Ces caractéristiques expliquent qu'il faille une électronique complexe pour piloter un servomoteur, et que cette électronique lui soit dédiée.

II.4.1.3. Brochage d'un servomoteur

Le servomoteur a besoin de trois fils de connexion pour fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

- Rouge : pour l'alimentation positive (4.5V à 6V en général)
- Noir ou marron : pour la masse (0V)
- Orange, jaune, blanc, ... : entrée du signal de commande

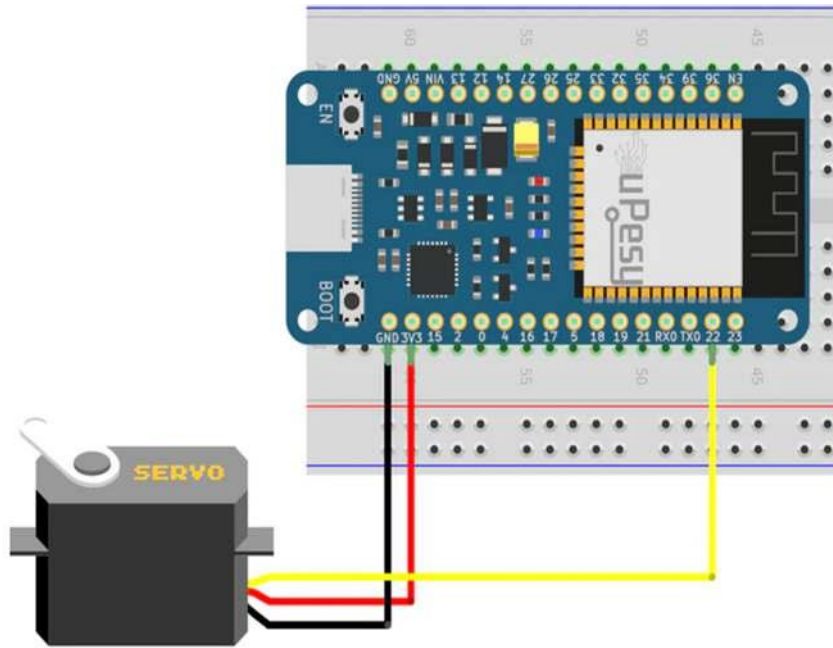


Figure II.14. Simulation de l'ESP32 et le servomoteur

II.4.2. BUZZER

Un BUZZER est un composant électronique utilisé pour produire des sons audibles. Il est souvent utilisé dans des applications d'alerte, de signalisation ou de notification sonore dans divers systèmes électroniques. Le son produit par un BUZZER peut être continu ou intermittent [13].



Figure II.15. BUZZER

Un BUZZER possède 2 broches :

- Le long : positive.
- Le petit : négative.

II.4.3. Diode électroluminescente LED

Une Diode électroluminescente DEL (en anglais LED : Light Emitting Diode) est un composant optoélectronique, sa facilité de montage sur un circuit imprimé, sa faible

consommation, sa résistance mécanique, sa petite taille, sa longue durée de vie et d'autres caractéristiques font la LED un composant de plus en plus inévitable [10].

La LED possède 2 broches :

- La longue broche : positive.
- La petite broche : négative.

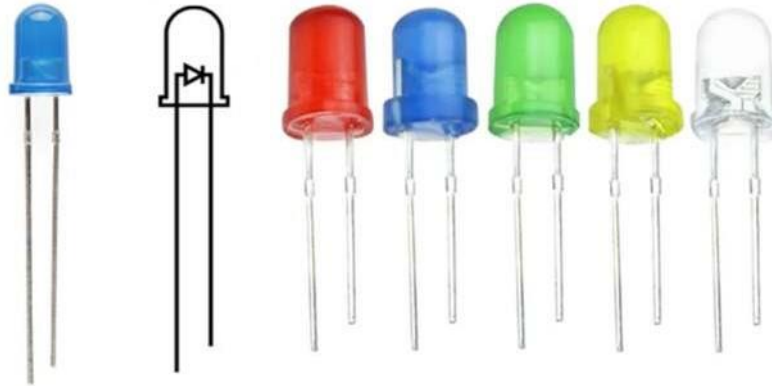


Figure II.16. LED's

II.5. Alimentation des composants

II.5.1. Régulateur 3V3

La série S1117 de régulateurs fixes et ajustables positifs est conçue pour fournir 1A avec une efficacité élevée. Tous les circuits internes sont conçus pour fonctionner avec un différentiel d'entrée à sortie de 1,3V. Le réglage sur puce ajuste la tension de référence à 2%.

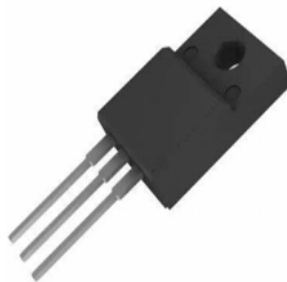


Figure II.17. Régulateur de tension 3.3V AUK626 S1117

Le régulateur possède les caractéristiques suivantes :

- Sortie ajustable ou fixe
- Courant de sortie de 1A.
- Chute de tension faible, 1.3V maximum à un courant de sortie de 1A
- Protection contre la surchauffe

- Réponse transitoire rapide [14].

II.5.2. Régulateur 5V

Le module Régulateur DC-DC Step Down LM2596 permet de générer une tension de sortie allant de 1.3V à 35V / 3A. Basé sur le régulateur LM2596 de chez Texas Instrument, il permet de générer une tension plus basse que la tension que vous lui fournissez en entrée. La tension de sortie s'ajuste via le potentiomètre présent sur le module.



Figure II.18. Module Régulateur DC-DC Step Down LM2596

Les caractéristiques générales du module DC-DC sont :

- Tension d'entrée : 3.2V à 40V
- Tension de sortie : 1.3V à 35V
- Courant maximum de sortie : 3A max
- Fréquence de l'hacheur : 65KHz
- Précision de la tension de sortie : +/-2.5%
- Taille du PCB (L x l x H) : 43mm x 21mm x 14mm [15].

II.6. Circuit Imprimé PCB

Un circuit imprimé PCB (Printed Circuit Board), est une plaque isolante permettant de connecter les différents composants électroniques entre eux, dans le but de réaliser un circuit électronique complexe. On le désigne aussi par le terme de carte électronique. Les PCB sont utilisés dans de nombreux appareils électroniques, tels que les ordinateurs, les téléphones portables, les téléviseurs, les équipements médicaux, etc. Il est constitué d'un assemblage d'une ou plusieurs fines couches de cuivre séparées par un matériau isolant. Les couches de cuivre sont gravées par un procédé chimique pour obtenir un ensemble de pistes Ils peuvent être fabriqués en utilisant différentes techniques, telles que la gravure chimique, la sérigraphie ou la photolithographie [16].

Pour réaliser le circuit imprimé il faut passer par logiciel ARES de Proteus (Advanced Routing and Editing Software) qui est conçu spécialement pour la conception de circuits

imprimés basée sur des listes de connexions, complétée par une suite d'outils d'automatisation de conception hautes performances.

II.6.1. Conception de PCB

La conception d'une carte PCB passe par les étapes suivantes :

- La 1^{ère} étape : création des broches de microcontrôleur ESP32, les composants, les modules d'alimentation.

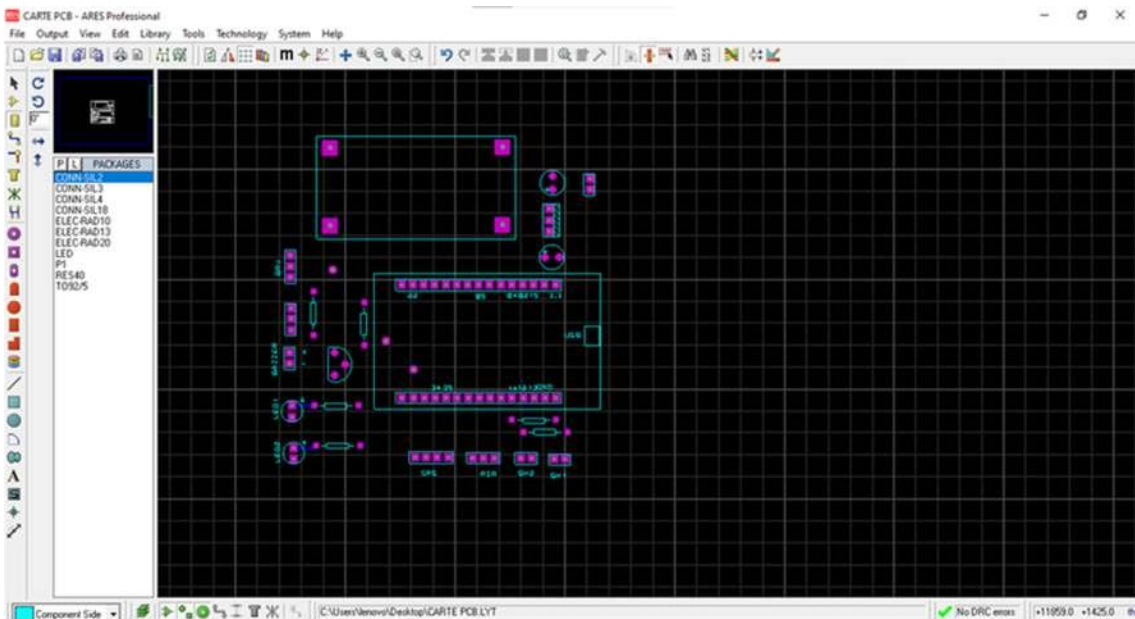


Figure II.19. Composant de circuit imprimé dans le logiciel ARES

- La 2^{ème} étape : Consiste à connecter tous les composants et les modules d'alimentation au microcontrôleur ESP32.

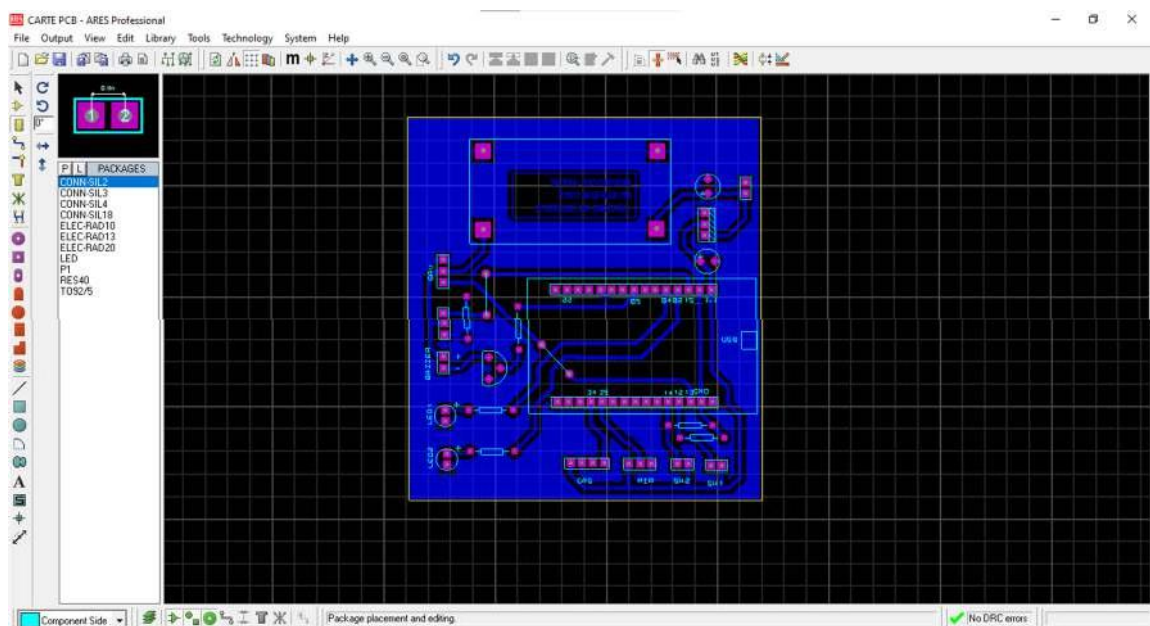


Figure II.20. Routage des éléments du circuit imprimé

➤ La 3^{ème} étape : Vérification s'il n'y a pas d'erreur de conception du circuit imprimée dans DRC (Design Rule Checker).



Figure II.21. Vérification des erreurs de conception.

II.6.2. Réalisation du circuit imprimé

Les résultats obtenus de la réalisation du circuit imprimé (PCB) sous l'environnement ARES de Proteus en visualisation 3D, est donné par la figure ci-dessous :

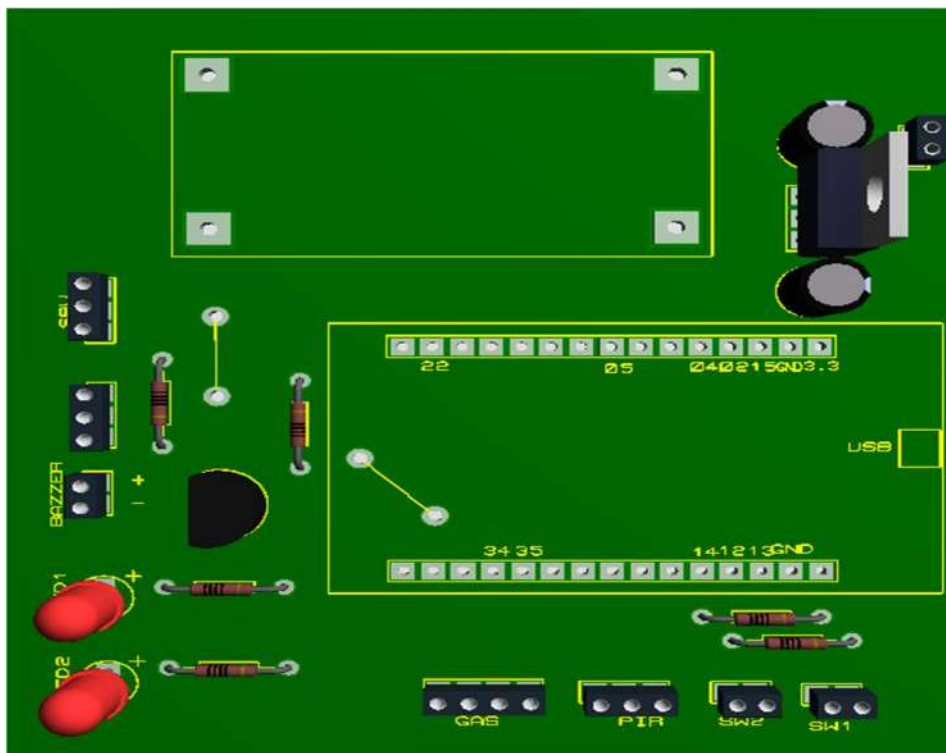


Figure II.22. Visualisation 3D de la face avant

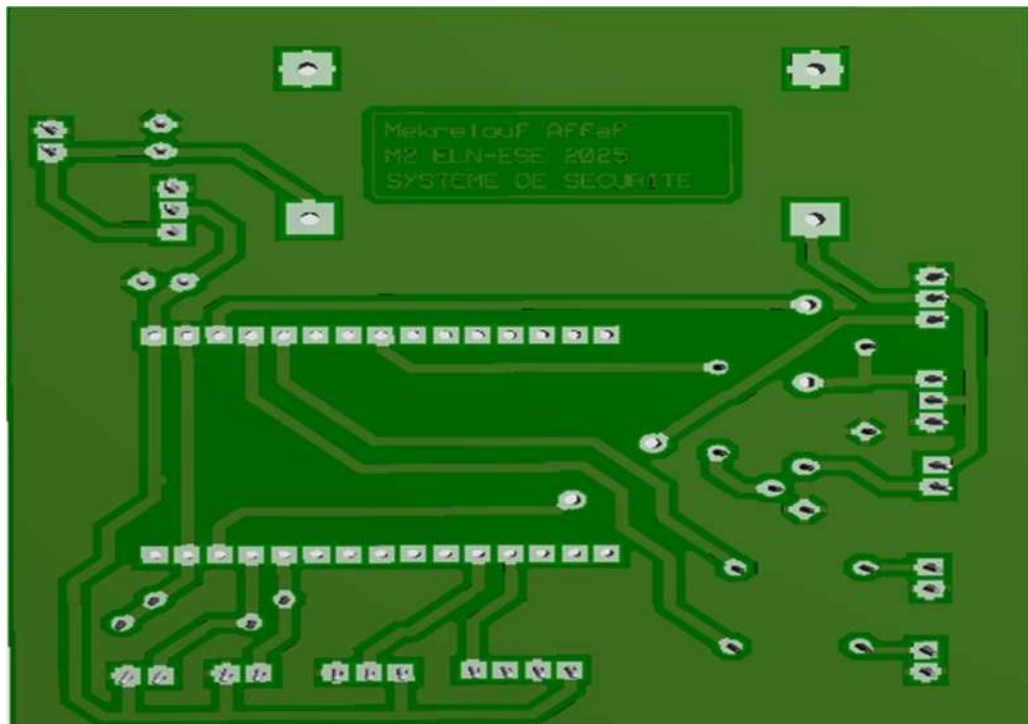


Figure II.23. Visualisation 3D de la face arrière

La carte électronique réelle du circuit imprimé après un processus de révélation et développement, tout en utilisant des produits chimiques, on obtient les résultats illustrés en face avant (coté composant) par la figure II.24 et en face arrière (coté circuit imprimé) par la figure II.25.

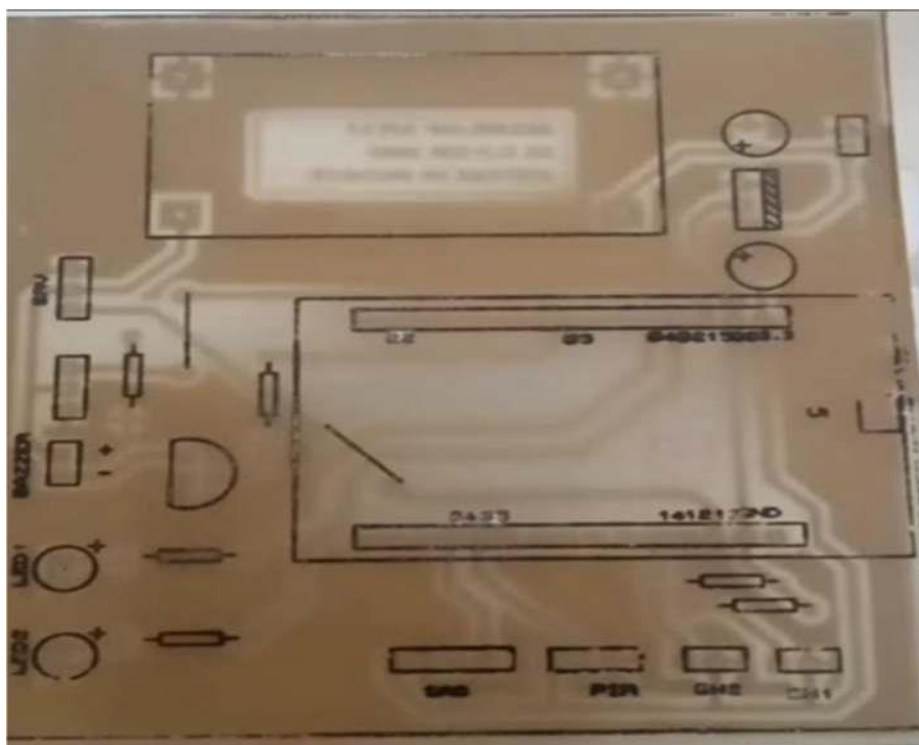


Figure II.24. Carte électronique face avant

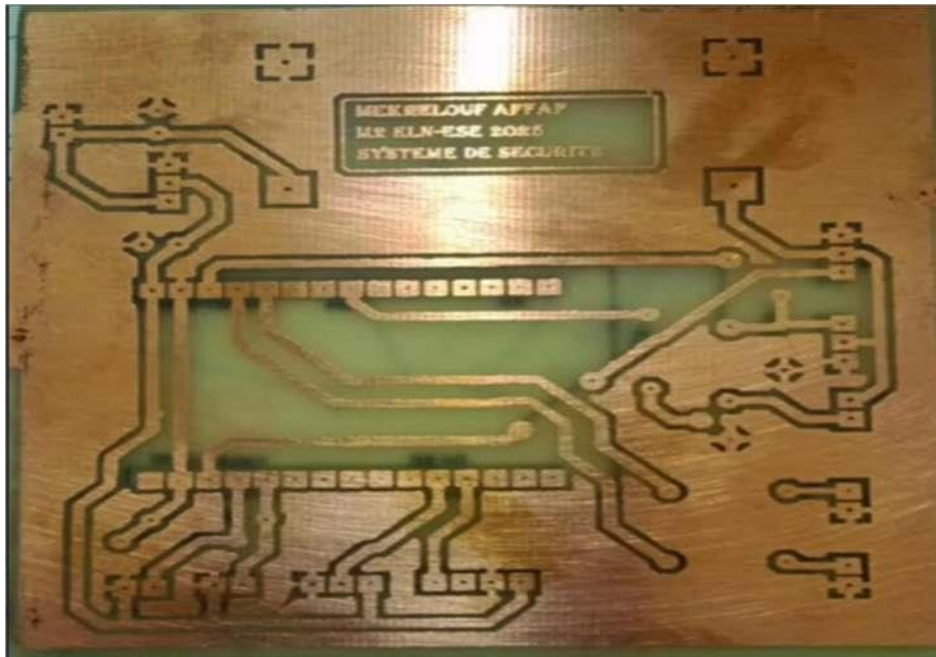


Figure II.25. Carte électronique face arrière

II.7. Conclusion

Le présent chapitre a été consacré à la présentation des composantes matérielles qui constituent notre système d'étude, dont on a expliqué la fiche technique de chaque composant, le principe de fonctionnement ainsi que le brochage et la liaison avec le microcontrôleur ESP32. Bien qu'on a expliqué le rôle principal du microcontrôleur ESP32-WROOM-32.

A la fin de ce chapitre, on a accédé à la simulation sous l'environnement ARES de Proteus, ensuite on a passé à la réalisation de la carte électronique à partir d'une carte à circuit imprimé PCB.

Ces éléments constituent la base physique du projet. Par ailleurs, le prochain chapitre traitera la deuxième partie de la conception d'un système de sécurité intelligent pour une maison, c'est la partie logicielle, en abordant la programmation des composants, le stockage des données sur Firebase, et la création de l'application mobile pour l'affichage et le contrôle du système.

Bibliographie chapitre 2

[1] Kessaissia Mohamed Fares & Ferhat Mohamed. « Conception d'un système de Gestion intelligente (IoT) de confort thermique des bâtiments », Mémoire de Master Filière Électronique Spécialité Instrumentation, 2023.

[2] Espressif Systems, “ ESP32-WROOM-32 Datasheet ”, Version 3.9, 2023. [En ligne]. Disponible :

https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

[3] DZduino, carte ESP32. Disponible sur : [ESP32,WROOM,32](#) (11/05/2025)

[4]electropeak, MQ9 Gas Sensor disponible sur: [How to Calibrate, Use, And Interface MQ9 Gas Sensor w/ Arduino](#) (06/05/2025)

[5] The Engineering Projects. *MQ-9 Gas Sensor: Datasheet, Pinout & Working*. Retrieved from: <https://www.theengineeringprojects.com/2024/03/mq-9-gas-sensor.html>

[6] <https://www.winsen-sensor.com/product/mq-9b.html?by=mq-sensor>

[7] <https://www.electrosks.com/electronique-detail/capteur-Gas-mq-9> (19/05/2025)

[8] Arduino Modules. *KY-037 High Sensitivity Sound Detection Module*. Disponible sur : <https://arduinomodules.info/ky-037-high-sensitivity-sound-detection-module/>(19/05/2025)

[9] Arduino Modules. *KY-037 High Sensitivity Sound Detection Module*. <https://arduinomodules.info/ky-037-high-sensitivity-sound-detection-module/>(19/05/2025)

[10] Conception d'une maison intelligente Réalisé par : MOSTEFAI Abir MESLI Sarah Rania

[11] esp32 pir sensor. Disponible sur : <https://www.upesy.com/blogs/tutorials/pir-infrared-sensor-module-esp32-whith-arduino-code> (01/06/2025)

[12] dealec micro switch NO/NC. Disponible sur : <https://www.dealec.fr/blog/2021/01/15/en-savoir-plus-sur-les-micro-switchs>) (01/06/2025)

[13] La carte ESP32 et le Buzzer. Disponible sur : <https://www.robotique.site/tutoriel/la-carte-esp32-et-le-buzzer/> (01/06/2025)

[14] Datasheet4U, régulateur 3.3v disponible sur : [S1117API Datasheet - AUK](#) (05/06/2025)

[15] ElectroProto, datasheet de Down LM2596 disponible sur : [Régulateur DC-DC Step Down LM2596 1.25V-32V / 3A](#) (05/06/2025)

[16] Wikipédia, Circuit imprimée, disponible sur : [Circuit imprimé — Wikipédia](#) (05/06/2025).

CHAPITRE III

Chapitre III

Composantes logicielles d'un système de sécurité intelligent

III.1. Introduction

Après avoir présenté les aspects matériels du système de sécurité intelligent, ce chapitre est consacré à la partie logicielle, essentielle au bon fonctionnement et à l'interconnexion des différents composants. Il s'agit notamment de la programmation du microcontrôleur ESP32-WROOM-32 à l'aide de l'environnement Arduino IDE, qui permet de lire les données des capteurs, de déclencher les actionneurs en cas d'alerte, et d'assurer la communication avec la base de données.

Le présent chapitre aborde également la connexion du système à la plateforme Fire base, utilisée pour le stockage en temps réel des données, facilitant l'accès à distance et la traçabilité des événements. Enfin, une application mobile développée avec MIT App Inventor est présentée, offrant à l'utilisateur une interface simple pour visualiser les états du système et interagir avec celui-ci à distance.

Ce chapitre met ainsi en lumière la liaison entre la programmation embarquée, le Cloud et l'application mobile, afin de réaliser un système de sécurité intelligent, autonome et connecté.

III.2. Présentation de logiciel Arduino IDE

III.2.1. Définition

L'Arduino IDE (Integrated Development Environment) est un logiciel open-source conçu pour faciliter la création et la programmation de projets électroniques basés sur la plateforme Arduino. L'IDE fournit un éditeur de texte pour écrire le code, un compilateur pour transformer le code en langage machine compréhensible par le microcontrôleur, ainsi qu'un environnement de programmation convivial pour téléverser le code sur la carte Arduino ou ESP32.

L'IDE Arduino dispose également d'une bibliothèque de fonctions pré-écrites qui peuvent être utilisées pour interagir avec des capteurs et des actionneurs couramment utilisés, ce qui facilite la création de projets électroniques sans nécessiter une connaissance approfondie de l'électronique ou de la programmation. L'IDE est disponible pour les systèmes d'exploitation Windows, Linux, ...etc. ce qui le rend accessible à un large éventail d'utilisateurs [1].



Figure III.1. Arduino IDE Icon

Le téléversement (upload) du programme dans la mémoire de l'ESP32 se passe via le port USB de l'ordinateur, une fois téléverser, le logiciel s'appelle un micrologiciel. La console donne des informations sur le déroulement de la compilation et affiche les messages d'erreur.



Figure III.2. Câble USB pour ESP32.

III.2.2. Langage

L'Arduino, basé sur le langage C++, est un langage de programmation simplifié conçu pour contrôler des circuits électroniques via des cartes Arduino. Accessible aux débutants, il offre une syntaxe claire et des fonctions préécrites facilitant l'utilisation de capteurs et d'actionneurs. Open-source et compatible avec d'autres langages, il permet le développement de projets électroniques variés, tout en favorisant l'extension de ses fonctionnalités par la communauté.

III.2.3. Interfaces

L'interface de l'IDE Arduino est plutôt simple, il offre une interface minimale et simple pour développer un programme sur les cartes Arduino.

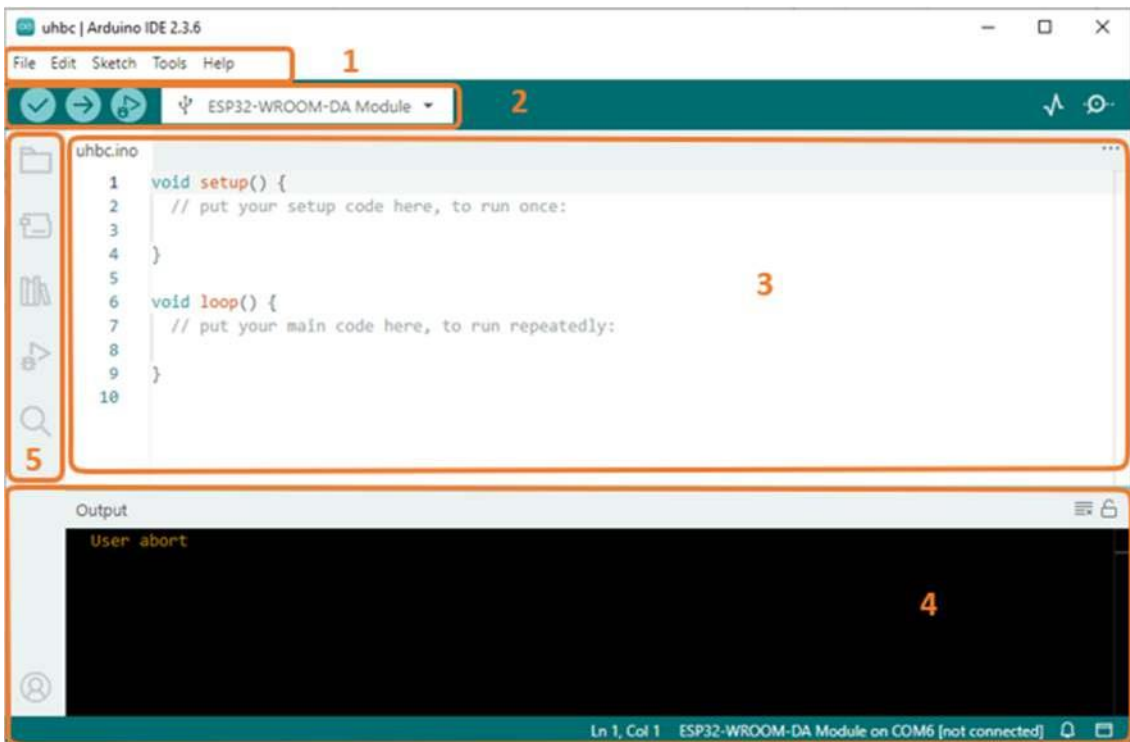


Figure III.3. Interface de logiciel Arduino IDE.

Menu 1 : Contient les options de configuration du logiciel :

- **Fichier** : Création, sauvegarde en spécifiant la destination.
- **Edit** : couper, copier, coller, supprimer, sélectionner, etc.
- **Sketch** : permet de compiler, téléverser le programme sur la carte, gérer les bibliothèques et accéder aux fichiers du projet.
- **Tools** : pour spécifier le type de la carte, le port série, formater, recharger et réparer l'encodage, graver la séquence d'initialisation, de la carte branchée sur l'ordinateur.

Menu 2 : Contient des boutons qui vont servir à programmer les carte Arduino :



Figure III.4. Première barre d'interface d'Arduino IDE

- **Bouton 1** : Vérification du programme.
- **Bouton 2** : téléverser le programme dans les cartes Arduino utilisé
- **Bouton 3** : lancer une session de débogage pour analyser et corriger le fonctionnement du programme en temps réel.
- **Bouton 4** : choix des cartes Arduino (ESP32-WROOM-DA Module) et le port de communication série.

Bloc 3 : Cette partie contient le programme.

Une configuration (setup) : cette partie est lue seulement une fois, elle inclut les fonctions à exécuter au démarrage (utilisation de pins comme entrées ou sorties).

Une boucle (loop) : cette partie est jouée dans une boucle, c'est à cet endroit que les fonctions sont exécutées. Outre cette structure minimaliste, nous pouvons ajouter des "sous-programmes" ou des "routines" qui peuvent être appelés à tout moment dans la boucle, très utile pour la production de pièces de code répétitives.

Bloc4 : c'est la console elle sert à mentionner les erreurs et aide pour la correction de ces erreurs.

Menu 5 : gestion des cartes et projets.



Figure III.5. Deuxième barre d'interface d'Arduino IDE

Bouton 1 : affiche la liste des projets récemment réalisés ou enregistrés.

Bouton 2 : permet de choisir le type de carte Arduino utilisée (comme UNO, Méga, Nano, ESP32, etc.).

Bouton 3 : donne accès aux bibliothèques des cartes et des composants (pour capteurs, modules, etc.).

Bouton 4 : lance le débogage pour analyser le programme en temps réel.

Bouton 5 : effectue une recherche dans le code du projet actuellement ouvert.

III.2.4. Critères de choix

On a choisi d'utiliser ce langage car il est fondé sur plusieurs caractéristiques, parmi lesquels on peut citer :

- Logiciel et matériel open source et extensible.

- Environnement de programmation clair et simple.
- Multiplateforme : tourne sous Windows, Linux....
- Nombreuses bibliothèques disponibles avec diverses fonctions implémentées.

III.2.5. Configuration et initialisation des composants

III.2.5.1. Installation de la carte ESP32

Pour installer la carte ESP32 dans l'IDE Arduino, on a suivi les instructions ci-dessous :

1. Dans l'IDE Arduino, allons dans **File > Préférences** ;
2. Saisissons les informations suivantes dans le champ "Additional Board Manager URLs" :

https://raw.githubusercontent.com/esp8266/arduino-esp32/gh-pages/package_esp32_index.json

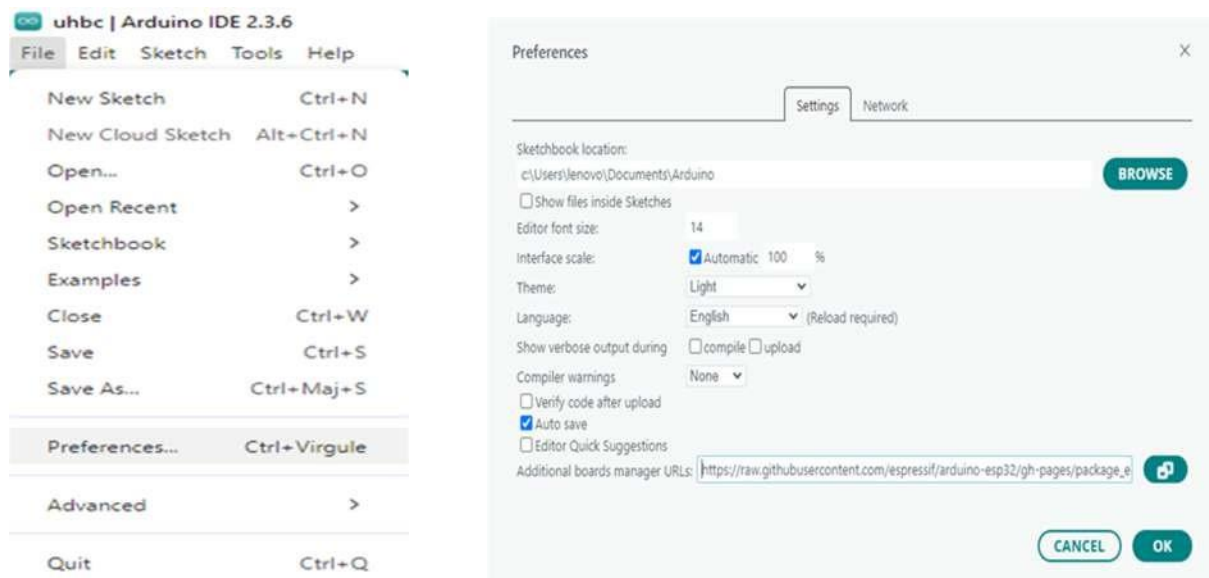


Figure III.6. Configuration de l'ESP32 sur l'IDE Arduino

3. Ensuite cliquer sur OK.
4. Ouvrez le gestionnaire des cartes. Accédez à **Tools > Board > Boards Manager**, Saisissons 'ESP32' dans le champ Type, puis appuyez sur Installer [2].

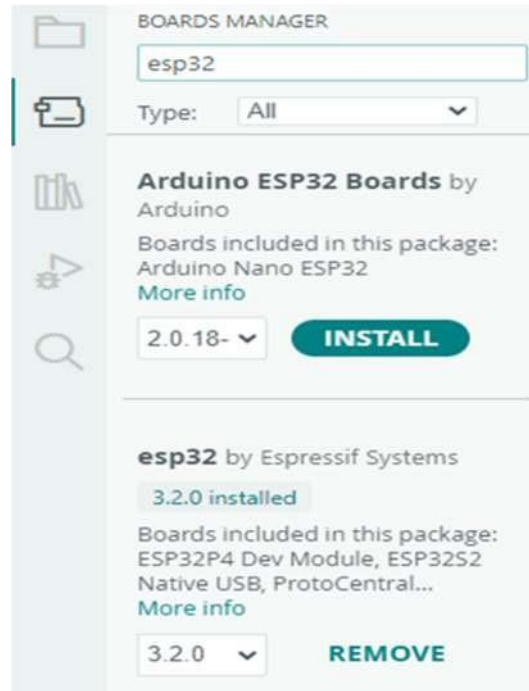


Figure III.7. Installation du logiciel pour la carte ESP32.

5. Sélectionnez la carte et le port
6. On choisit dans la barre Tools>Board >esp32> ESP32-WROOM-DA Module.

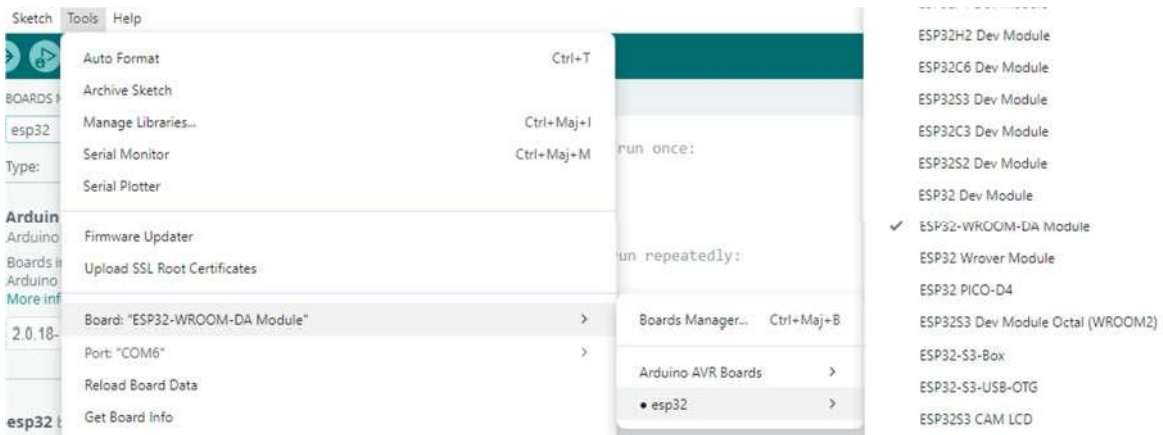


Figure III.8. Sélection de la carte ESP32.

Avant de brancher la carte ESP32 sur l'ordinateur en utilisant un câble USB, il doit sélectionner le port de communication [3].

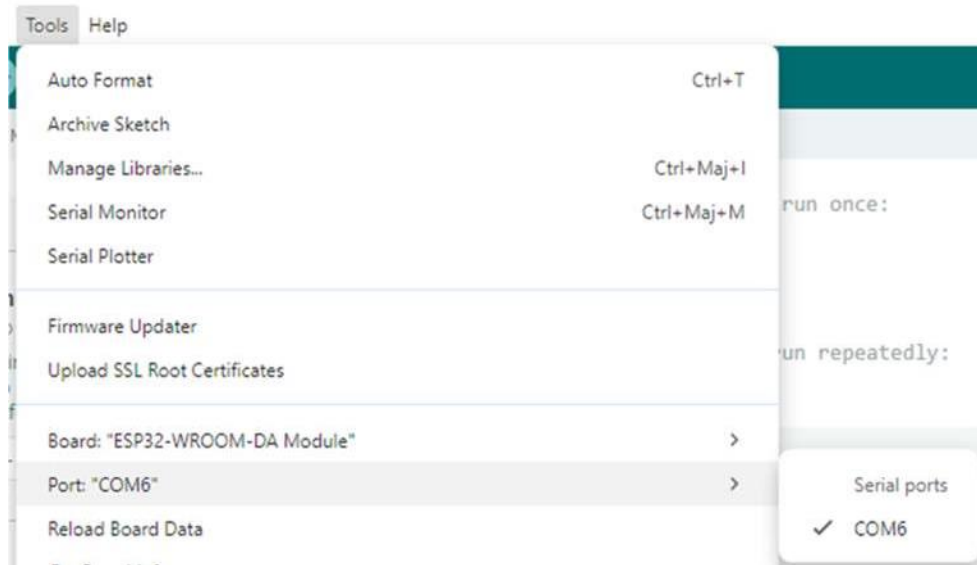


Figure III.9. Sélection du port de communication.

Après avoir suivi ces étapes, nous avons terminé l'installation de la carte ESP32.

III.2.5.2. Configurations du capteur de Gas MQ-9

Le code Arduino IDE utilisé pour configurer le capteur de Gas MQ-9 est illustré par la figure ci-après.

```
MQ2.ino
1 // Définir la broche analogique à laquelle le capteur de gaz est connecté
2 #define GAS_SENSOR_PIN 34 // GPIO34 est une entrée analogique sur l'ESP32
3 #define LED1_PIN 2
4 void setup() {
5     // Initialiser la communication série à 9600 bauds
6     Serial.begin(9600);
7     // Configurer la broche du capteur de gaz en tant qu'entrée
8     pinMode(GAS_SENSOR_PIN, INPUT);
9     pinMode(LED1_PIN, OUTPUT);
10 }
11
12 void loop() {
13     // Lire la valeur analogique du capteur de gaz
14     int sensorValue = analogRead(GAS_SENSOR_PIN);
15
16     // Afficher la valeur brute lue depuis le capteur
17     Serial.print("Gas Sensor Value: ");
18     Serial.println(sensorValue);
19
20     // Attendre 1 seconde avant la prochaine lecture
21     delay(1000);

```

Output Serial Monitor X

Not connected. Select a board and a port to connect automatically.

```
Gas Concentration Percentage: 54
% Gas Sensor Value: 2219
Gas Concentration Percentage: 54
% Gas Sensor Value: 2219
Gas Concentration Percentage: 54
% Gas Sensor Value: 2218
```

Figure III.10. Programme de configuration du capteur de Gas MQ-9.

Ce programme lit les données d'un capteur de Gas connecté à la broche GPIO34 de l'ESP32. Il convertit la valeur analogique lue en un pourcentage estimé de concentration de

Gas. Si ce pourcentage dépasse un seuil défini (52 %), une LED connectée à la broche GPIO2 s'allume pour signaler un danger. Les valeurs sont affichées sur le moniteur série toutes les secondes.

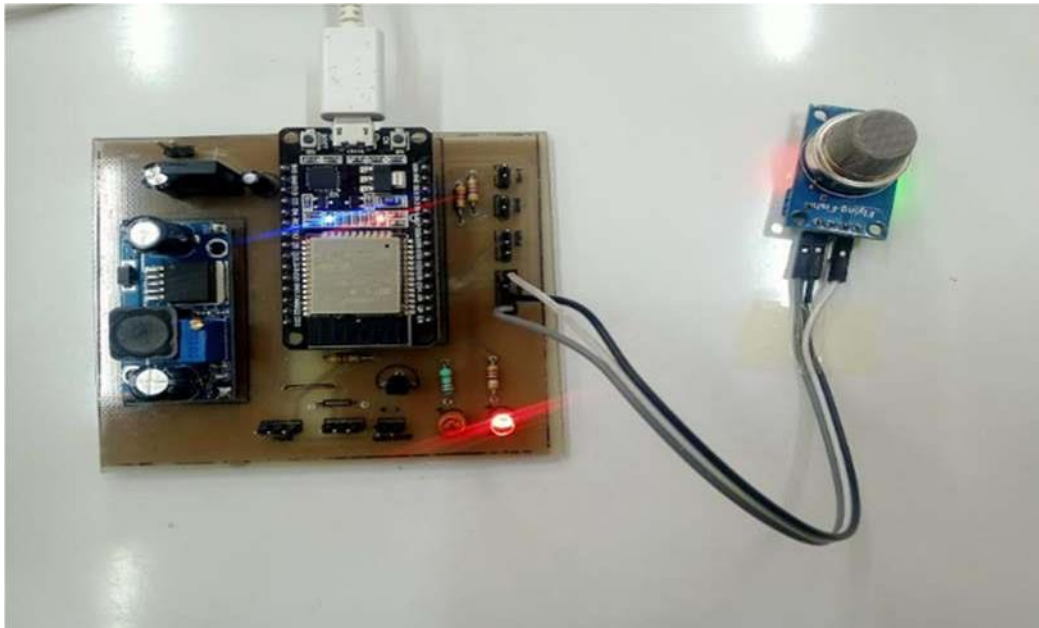


Figure III.11. Communication et essai du capteur de Gas MQ-9.

III.2.5.3. Configurations du capteur de mouvement PIR

Dans cette configuration du capteur de mouvement PIR, nous avons utilisé le code Arduino IDE suivant :

```
PIRmimoi.ino
1  #define PIR_PIN 33 // Définir la broche du capteur PIR (GPIO15)
2  #define LED_PIN 2  // Définir la broche de LED (allumer une LED lors de la détection de mouvement)
3
4  int pirState = 1;
5
6  void setup() {
7    pinMode(PIR_PIN, INPUT); // Définir la broche PIR comme une entrée
8    pinMode(LED_PIN, OUTPUT); // Définir la broche LED comme une sortie
9    Serial.begin(9600);      // Initialiser la communication série
10 }
11
12 void loop() {
13   pirState = digitalRead(PIR_PIN); // Lire l'état du capteur (HAUT ou BAS)
14
15   if (pirState == HIGH) { // Si un mouvement est détecté
16     digitalWrite(LED_PIN, HIGH); // Allumer la LED
17     Serial.println("Motion Detected!"); // Afficher un message en cas de détection de mouvement
18   }
19   else {
```

Output Serial Monitor X

```
Not connected. Select a board and a port to connect automatically.
Motion Detected!
No motion detected.
No motion detected.
No motion detected.
No motion detected.
Motion Detected!
No motion detected.
No motion detected.
```

Figure III.12. Programme de configuration de capteur de mouvement PIR.

Ce programme utilise un capteur PIR connecté à la broche GPIO33 pour détecter un mouvement. Lorsqu'un mouvement est détecté, une LED (sur la broche GPIO2) s'allume et un message est affiché sur le moniteur série. Si aucun mouvement n'est détecté, la LED reste éteinte. Le système vérifie l'état du capteur toutes les 500 millisecondes.

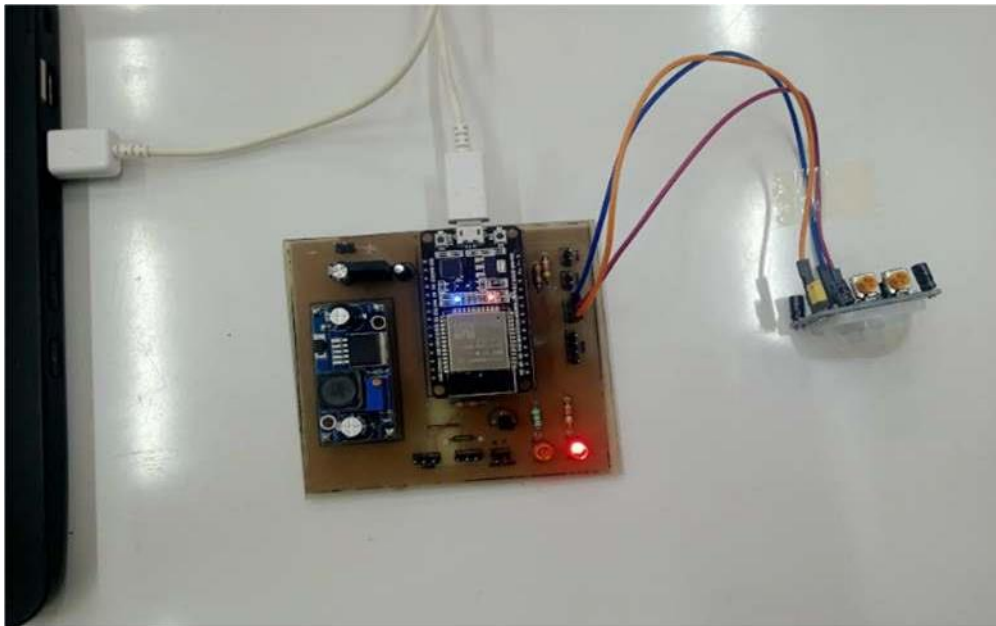


Figure III.13. Communication et essai du capteur de mouvement PIR.

III.2.5.4. Configurations de capteur de bruit KY-037

Pour configurer le capteur de bruit KY-037, nous avons opté pour le code Arduino IDE ci-dessous :

```
BRUIT.ino
1  #define pinBruit 15      // Capteur de bruit (sortie D0)
2  #define pinBuzzer 5     // Buzzer (peut être n'importe quel GPIO)
3
4  int etatPrecedent = LOW; // Stocke l'état précédent du capteur pour détecter les changements
5  int compteurCoups = 0;  // Compte le nombre de coups détectés
6  int seuilCoups = 3;    // Seuil de déclenchement (ex. : déclencher après 3 coups)
7
8  void setup() {
9      Serial.begin(9600); // Initialiser la communication série pour affichage
10     pinMode(pinBruit, INPUT); // Configurer la broche du capteur de bruit comme entrée
11     pinMode(pinBuzzer, OUTPUT); // Configurer la broche du buzzer comme sortie
12     digitalWrite(pinBuzzer, LOW); // S'assurer que le buzzer est éteint au démarrage
13     Serial.println("Détection des coups de bruit activée...");
14 }
15
16 void loop() {
17     int etatActuel = digitalRead(pinBruit); // Lire l'état actuel du capteur de bruit (0 ou 1)
18     // Détection d'un front montant
19     if (etatPrecedent == LOW && etatActuel == HIGH) {
20         compteurCoups++;
21         Serial.print("Coup détecté ! Total : ");
22     }
23 }

```

Output Serial Monitor X

Not connected. Select a board and a port to connect automatically.

```
Coup détecté ! Total : 1
Coup détecté ! Total : 2
Coup détecté ! Total : 3
Seuil atteint ! Activation du buzzer !
Coup détecté ! Total : 1
Coup détecté ! Total : 2

```

Figure III.14. Programme de configuration de capteur de bruit KY-037.

Ce programme détecte les coups sonores grâce à un capteur de bruit numérique. Lorsqu'un certain nombre de coups (par exemple 3) est atteint, un buzzer s'active pendant une seconde pour signaler une alerte, puis le système se réinitialise. Le tout est contrôlé via une lecture d'état sur GPIO15 et une sortie sonore sur GPIO5.

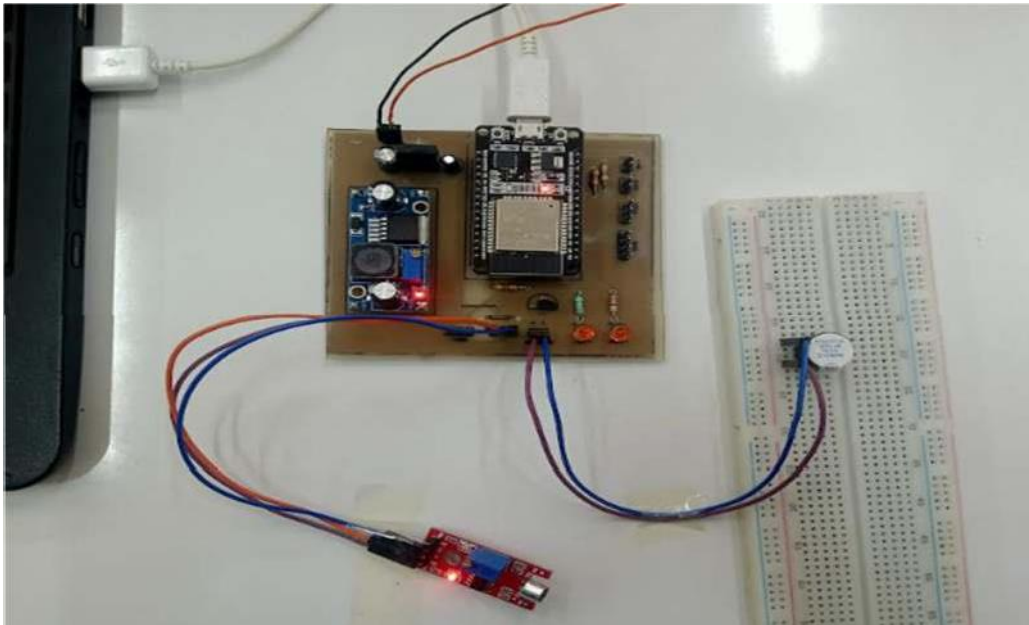


Figure III.15. Communication et essai du capteur de bruit KY-037

III.2.5.5. Configurations de SWITCH

Afin de mieux configurer les switches, on a utilisé le code Arduino IDE comme suit :

```
switch.ino
1  #define SWITCH_FENETRE 14 // Broche reliée à l'interrupteur magnétique de la fenêtre
2  #define SWITCH_PORTE 12 // Broche reliée à l'interrupteur magnétique de la porte
3  #define BUZZER_PIN 13 // Broche de sortie pour le buzzer
4
5  void setup() {
6    Serial.begin(9600); // Démarre la communication série
7
8    // Entrées avec pull-up interne
9    pinMode(SWITCH_FENETRE, INPUT_PULLUP);
10   pinMode(SWITCH_PORTE, INPUT_PULLUP);
11
12   pinMode(BUZZER_PIN, OUTPUT); //le buzzer comme une Sortie
13   digitalWrite(BUZZER_PIN, LOW); // buzzer éteint au départ
14 }
15
16 void loop() {
17   int etatFenetre = digitalRead(SWITCH_FENETRE);
18   int etatPorte = digitalRead(SWITCH_PORTE);
19
20   // Affichage dans le moniteur série
21   Serial.print("Fenêtre : ");

```

Output Serial Monitor ×

Not connected. Select a board and a port to connect automatically.

```
Porte : OUVERTE
BUZZER activé Fenêtre : FERMÉE
Porte : OUVERTE
BUZZER activé Fenêtre : FERMÉE
Porte : OUVERTE
BUZZER activé Fenêtre : OUVERTE
```

Figure III.16. Programme de configuration des Switches.

Ce programme vérifie en temps réel l'état d'un interrupteur magnétique sur une porte et sur une fenêtre. Si l'un des deux switches est ouvert, un buzzer s'active pour alerter. Le tout est surveillé via le moniteur série, et les broches sont configurées avec résistances de pull-up internes pour fiabiliser la lecture.

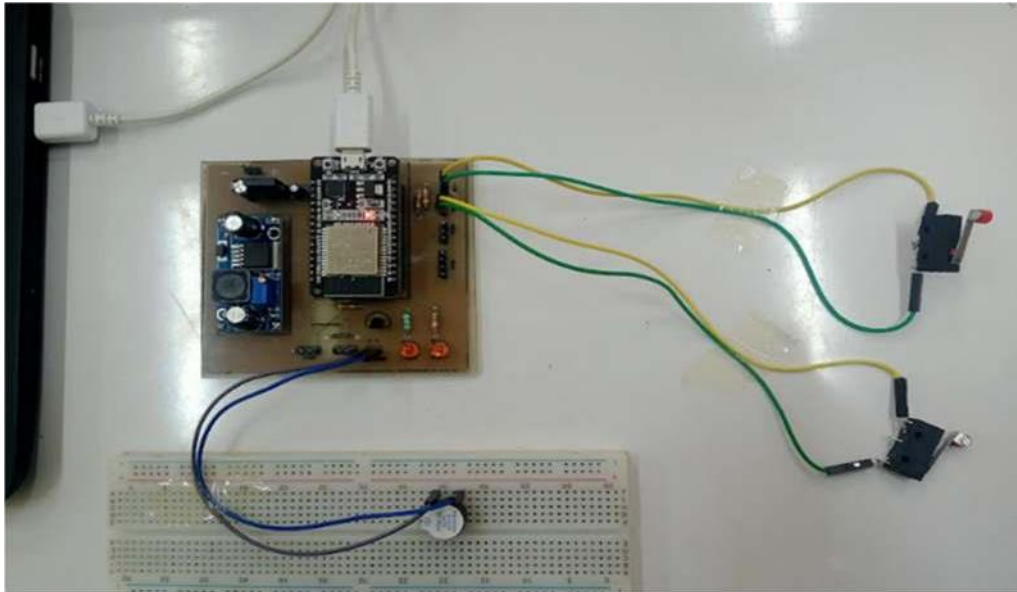


Figure III.17. Communication et essai des switches.

III.2.5.6. Configurations de Servomoteur

La configuration du servomoteur est achevée, en utilisant le code Arduino IDE suivant :

```
servomoteur.ino
1  #include <ESP32Servo.h> // Bibliothèque pour contrôler le servo avec ESP32
2  // Crée un objet "monServo" de type Servo
3  Servo monServo;
4
5  #define SERVO_PIN 13 // Définit la broche GPIO 13 utilisée pour le signal du servo
6
7  void setup() {
8
9      Serial.begin(9600); // Initialise la communication série
10     monServo.attach(SERVO_PIN); // Connecte le servomoteur à la broche définie (GPIO 13)
11     Serial.println("Test du servomoteur avec ESP32"); // Affiche un message de démarrage .
12 }
13 void loop() {
14
15     monServo.write(0); // Envoie le signal pour tourner le servo à 0° (position initiale)
16     Serial.println("Position : 0°");
17     delay(1000); // Attendre 1 seconde
18     monServo.write(90); // Envoie le signal pour tourner le servo à 90° (position centrale)
19     Serial.println("Position : 90°");
20     delay(1000); // Attendre 1 seconde
21     monServo.write(180); // Envoie le signal pour tourner le servo à 180° (position maximale)

```

Serial Monitor ×

Not connected. Select a board and a port to connect automatically.

```
Position : 0°
Position : 90°
Position : 180°
Position : 0°
Position : 90°
Position : 180°
```

Figure III.18. Programme de configuration de servomoteur.

On utilise la bibliothèque <ESP32Servo.h> au lieu de la bibliothèque classique <Servo.h> car elle est spécialement conçue pour l'ESP32, assurant une gestion stable et précise des signaux PWM (Pulse Width Modulation) nécessaires au bon fonctionnement des servomoteurs.

Ce programme permet de tester le fonctionnement d'un servomoteur contrôlé par un ESP32. Il utilise la bibliothèque ESP32Servo, spécialement adaptée à ce microcontrôleur, pour assurer une gestion correcte des signaux PWM. Le servo est successivement positionné à 0°, 90°, puis 180°, avec un délai d'une seconde entre chaque mouvement.

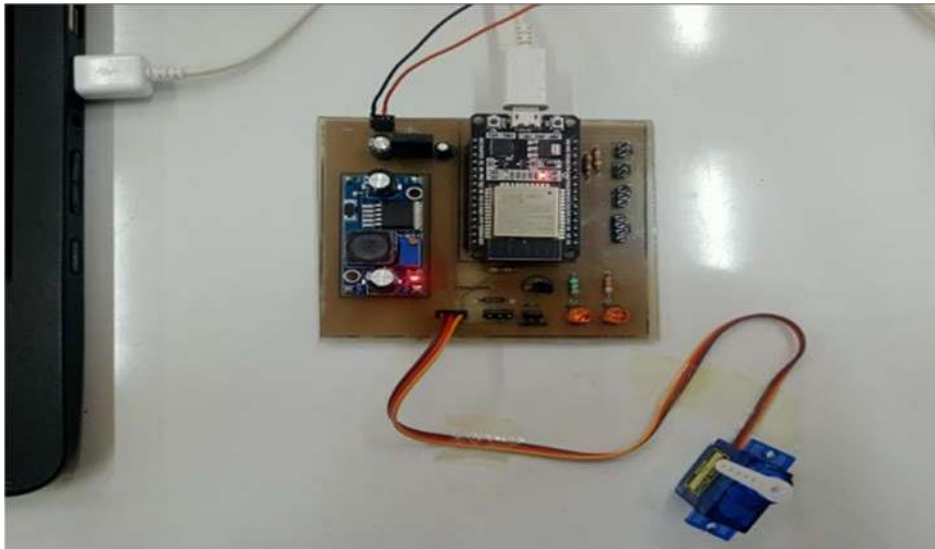


Figure III.19. Communication et essai de servomoteur.

III.3. Présentation de la plateforme Fire base

III.3.1. Définition

Fire base est une plateforme proposée par Google. Principalement utilisé pour développer des applications mobiles et web, Il est basé sur le Cloud, ce qui signifie que les applications créent avec Fire base utilisent les services de Google Cloud Platform pour stocker les données, traiter les demandes et effectuer d'autres opérations en arrière-plan [4].

Créée en 2011 comme startup nommée Envolv, elle devient Fire base en 2012, avec une base de données temps réel, avant d'être rachetée par Google en 2014 [5].



Figure III.20. Fire base.

III.3.2. Principales fonctionnalités de Fire base

Dans notre projet on a besoin de ces deux principales fonctionnalités seulement :

- Base de Données en Temps Réel (Realtime Database),
- Cloud Firestore.

Fire base propose deux solutions de base de données pour stocker et synchroniser les données en temps réel entre les clients. Realtime Database est une base de données NoSQL (Not Only Structured Query Language) qui permet la synchronisation instantanée, tandis que Cloud Firestore offre des capacités de requête plus avancées et une meilleure évolutivité [6].

III.3.3. Type de base de données de Fire base

Les deux options de base de données Fire base sont Firestore et Firebase Realtime Database, offrent une structure NoSQL, et il n'est pas possible d'utiliser une base de données relationnelle. Le NoSQL est un type de bases de données, dont la spécificité est d'être non relationnelle. Il est utilisé pour le Big Data et les applications web en temps réel. Pour prendre en charge les immenses volumes de données, les stocker et les analyser, il est impératif de s'en remettre à de nouvelles solutions [7].

Authentification : Fire base Authentication permet de gérer facilement les processus de connexion et d'inscription des utilisateurs. Il prend en charge plusieurs méthodes d'authentification, y compris les connexions via Google, Facebook, Twitter, et les adresses email [8].

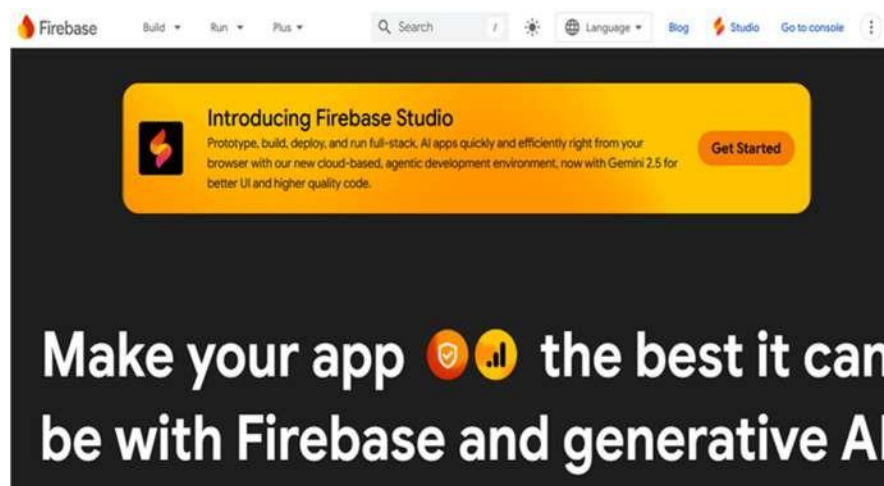


Figure III.21. Interface de Fire base.

III.3.4. Critères de choix

On a choisi d'utiliser cette plateforme de développement car elle est fondée sur plusieurs caractéristiques, parmi lesquels on peut citer [9] :

- Rapidité de développement.
- Traitement des données en temps réel.

- Facilité la création des applications.

III.3.5. Configuration d'un compte Fire base

La configuration d'un compte Fire base passe par les étapes suivantes :

- La 1^{ère} étape : créer un nouveau projet.
 - ✓ Allez sur Fire base et connectez-vous avec un compte Google.
 - ✓ Accédez à la console Fire base et créez un nouveau projet.
 - ✓ Donnez un nom à votre projet et cliquez sur Continuer.



Figure III.22. Nouveau projet de Fire base.

Ensuite, activez ou désactivez l'assistance par l'IA pour votre projet. Cette étape est facultative.



Figure III.23. Assistance par l'IA

Désactivez l'option, Activer Google Analytics pour ce projet, car elle n'est pas nécessaire. Ensuite, cliquez sur Créer un projet.

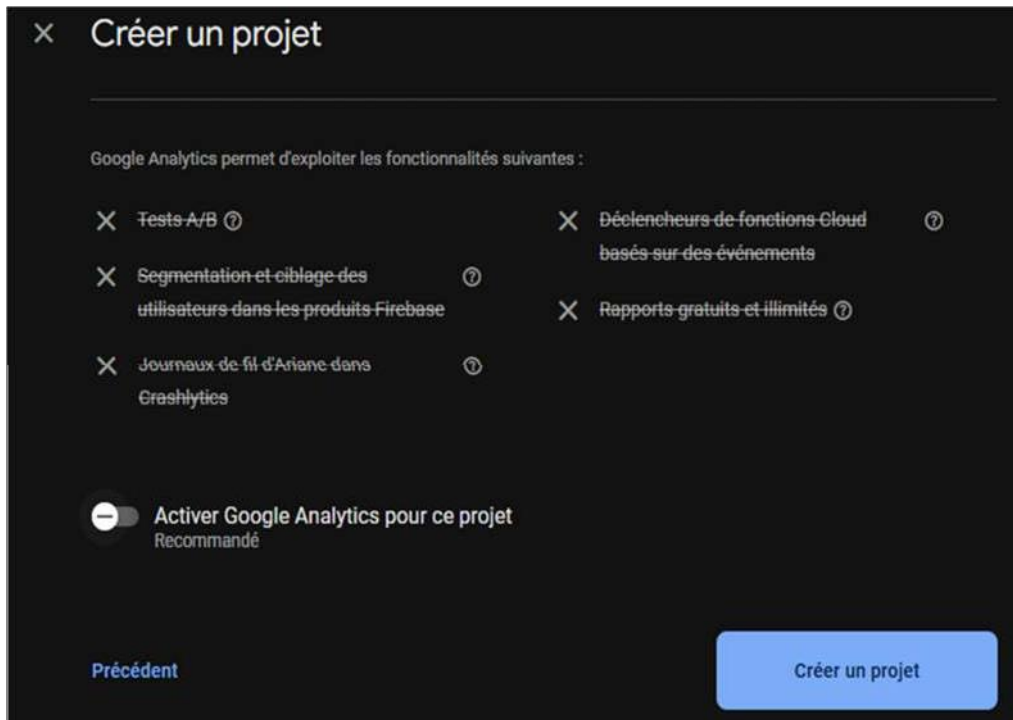


Figure III.24. Google Analytics.

La configuration de projet prendra quelques secondes. On clique sur Continuer lorsqu'il est prêt.



Figure III.25. Projet est prêt.

On sera redirigé vers la page de console de votre projet.

➤ La 2^{ème} étape : Configurer les méthodes d'authentification.

Dans la barre latérale gauche, on clique sur crée > Authentification, puis sur Commencer.

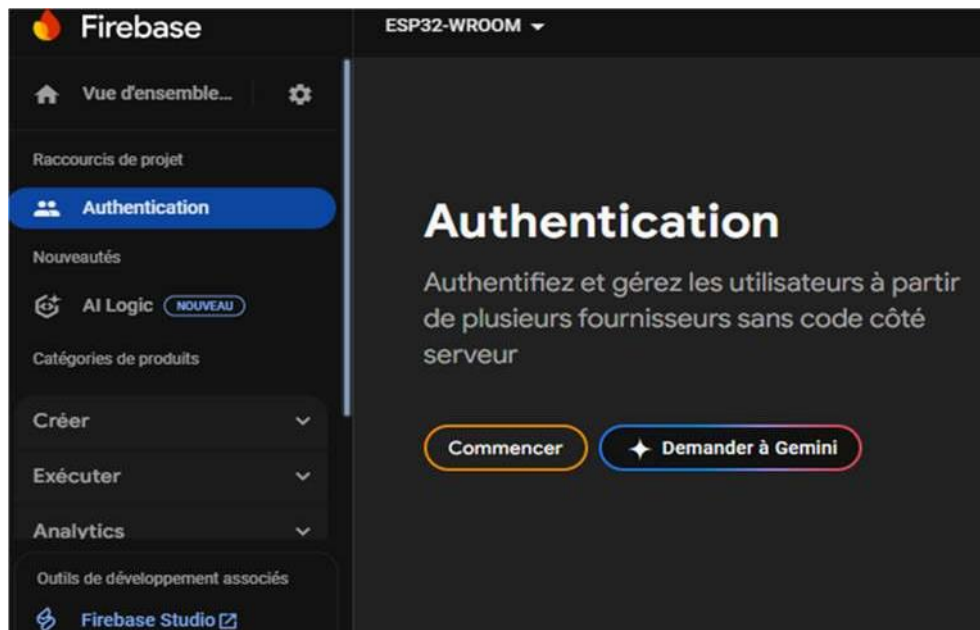


Figure III.26. Authentification du Fire base.

Il existe plusieurs méthodes d'authentification, comme l'authentification par e-mail et mot de passe, par compte Google, compte Facebook, et d'autres encore.

On Sélectionne E-mail/mot de passe et on active cette méthode d'authentification. Ensuite, on clique sur Enregistrer.

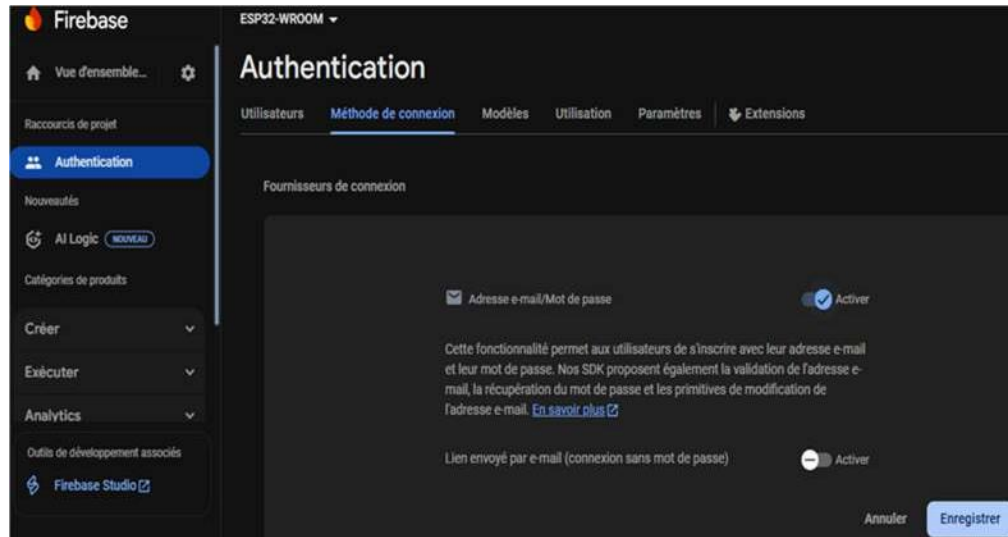


Figure III.27. Méthode d'authentification.

Ensuite, en haut de la page, on clique sur l'onglet Utilisateurs. Puis, on clique sur Ajouter un utilisateur.

On crée un nouvel utilisateur avec une adresse e-mail et un mot de passe. L'adresse e-mail peut être une adresse personnelle. On crée un mot de passe pour cet utilisateur. Enfin, on clique sur Ajouter un utilisateur.

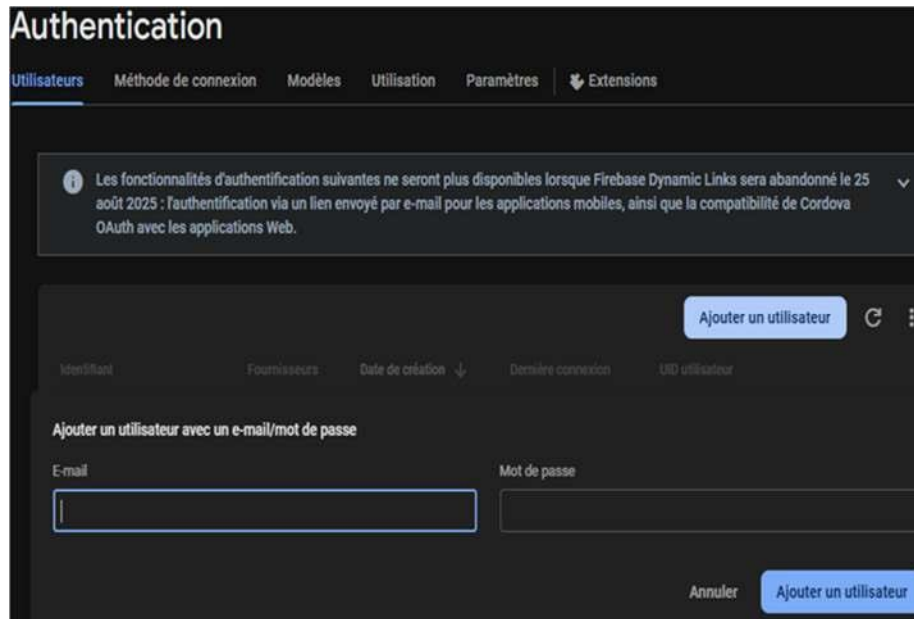


Figure III.28. Compte email et mot de passe.

L'utilisateur apparaîtra dans la liste des utilisateurs. Nous pouvons voir des informations le concernant, comme la date de création, la dernière connexion et son identification utilisateur.

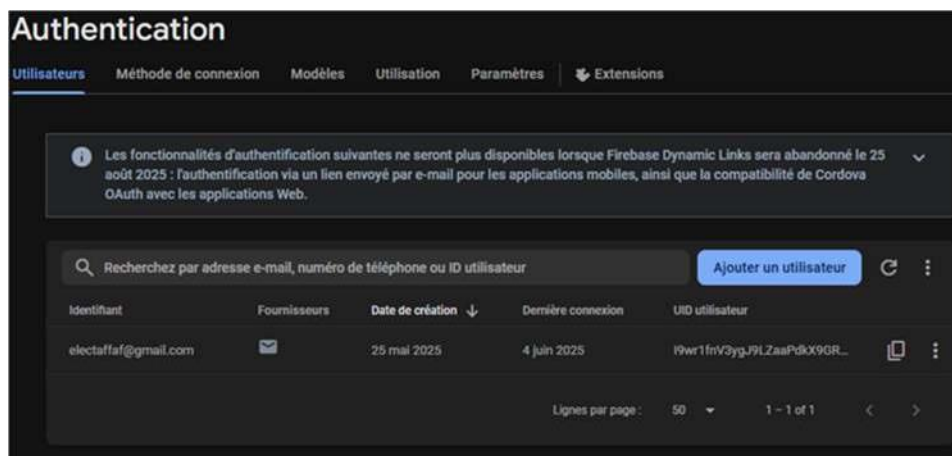


Figure III.29. Liste d'utilisateurs.

- La 3^{ème} étape : Créer une base de données en temps réel (Realtime Database)

Dans la barre latérale gauche, on clique sur crée > Realtime Database puis on clique sur Créer une base de données.

On sélectionne l'emplacement de la base de données. Il doit être le plus proche possible de la localisation.

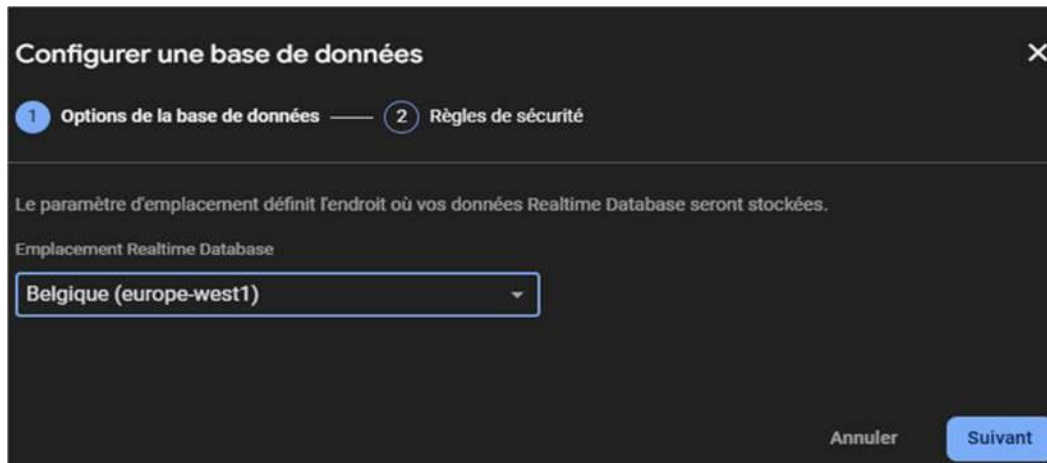


Figure III.30. Options de la base de données.

On configure les règles de sécurité pour la base de données. À des fins de test, on sélectionne Démarrer en mode test. Dans les tutoriels suivants, nous apprendrons à sécuriser notre base de données à l'aide de règles spécifiques.



Figure III.31. Règles de sécurité.

Notre base de données est maintenant créée. Nous devons copier et enregistrer l'URL de la base de données.

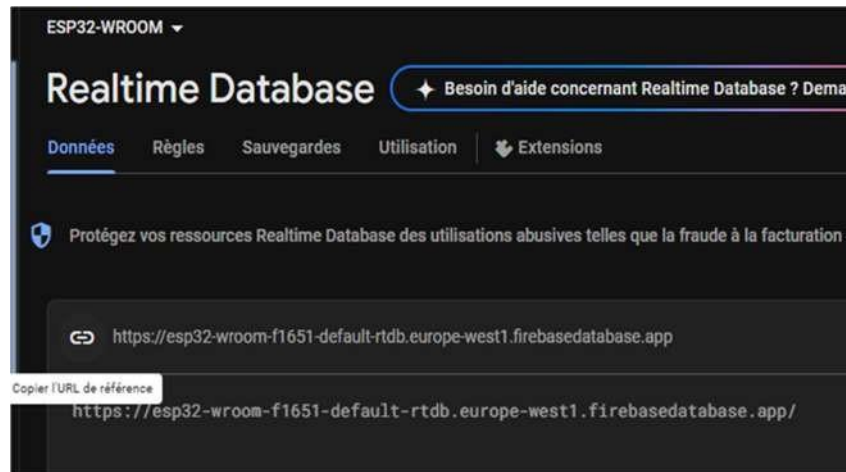


Figure III.32. URL de la base de données.

- La 4ème étape : Obtenir la clé API du projet.

Pour obtenir la clé API de votre projet, cliquez sur Paramètres du projet dans la barre latérale gauche.

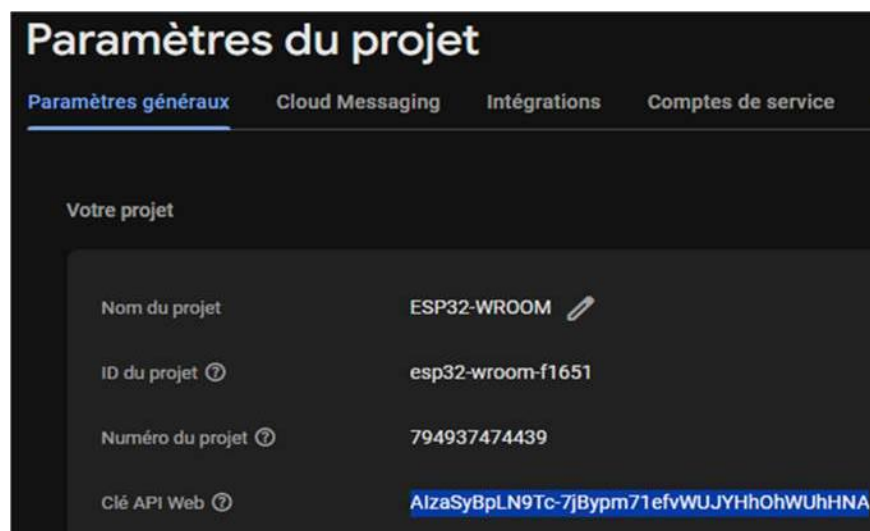


Figure III.33 Clé API.

Après avoir suivi ces étapes, nous avons terminé la configuration d'un compte Fire base [10].

III.4. Cycle de fonctionnement

Le fonctionnement entre le code Arduino IDE, l'ESP32 et Fire base repose sur une communication continue entre le microcontrôleur, les capteurs, et la base de données en ligne. Le code est d'abord écrit et téléversé dans l'ESP32 via l'environnement Arduino IDE. Ce code permet à l'ESP32 de se connecter au réseau Wi-Fi, de lire les données des capteurs connectés (comme un capteur de Gas ou un détecteur de mouvement), et d'envoyer ces informations vers Fire base. Fire base agit comme un serveur distant, permettant de stocker ces données en temps réel et de les consulter depuis une application mobile ou une interface web. En parallèle, l'ESP32 peut également lire des données provenant de Fire base [11].

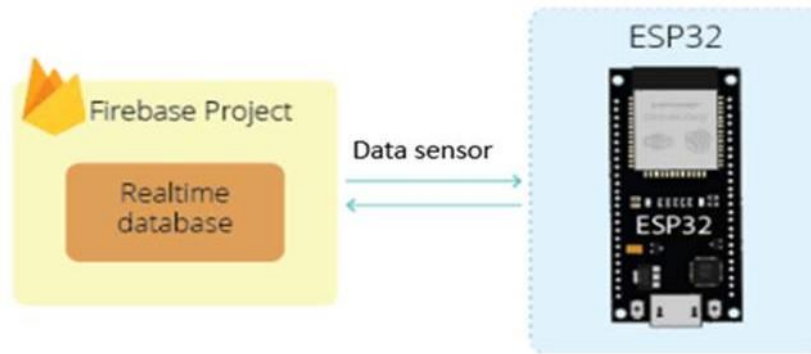


Figure III.34. Schéma fonctionnel de Fire base et l'ESP32.

Les étapes nécessaires pour assurer le fonctionnement :

- La 1^{ère} étape : installation des bibliothèques suivantes :

<WiFi.h> : permet à l'ESP32 de se connecter à un réseau Wi-Fi.

<Firebase_ESP_Client.h> : bibliothèque principale pour communiquer avec Fire base (Realtime Database).

<RTDBHelper.h> : la bibliothèque Fire base, utilisé pour simplifier la gestion des erreurs et des chemins de la Realtime Database (RTDB).

<TokenHelper.h> : la bibliothèque Fire base pour gérer l'authentification (API Key, user email/mot de passe, etc.).

<ESP32Servo.h> : permet de contrôler des servomoteurs avec l'ESP32.

- La 2^{ème} étape : Configuration de l'ESP32 comme une station [12].

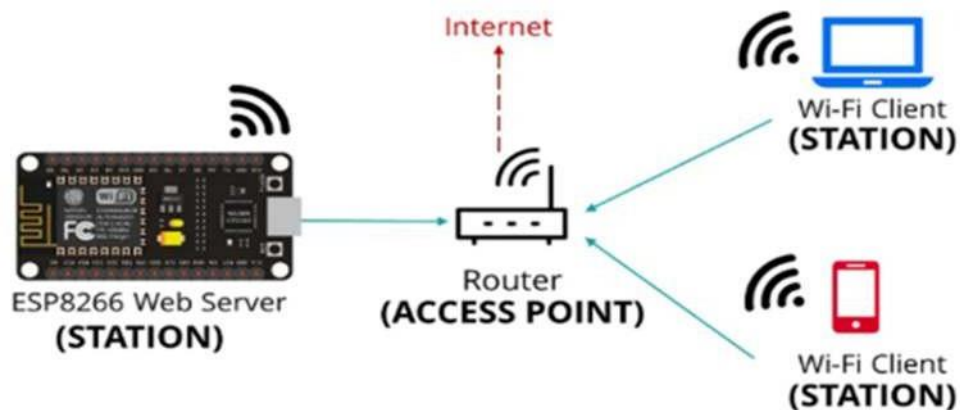
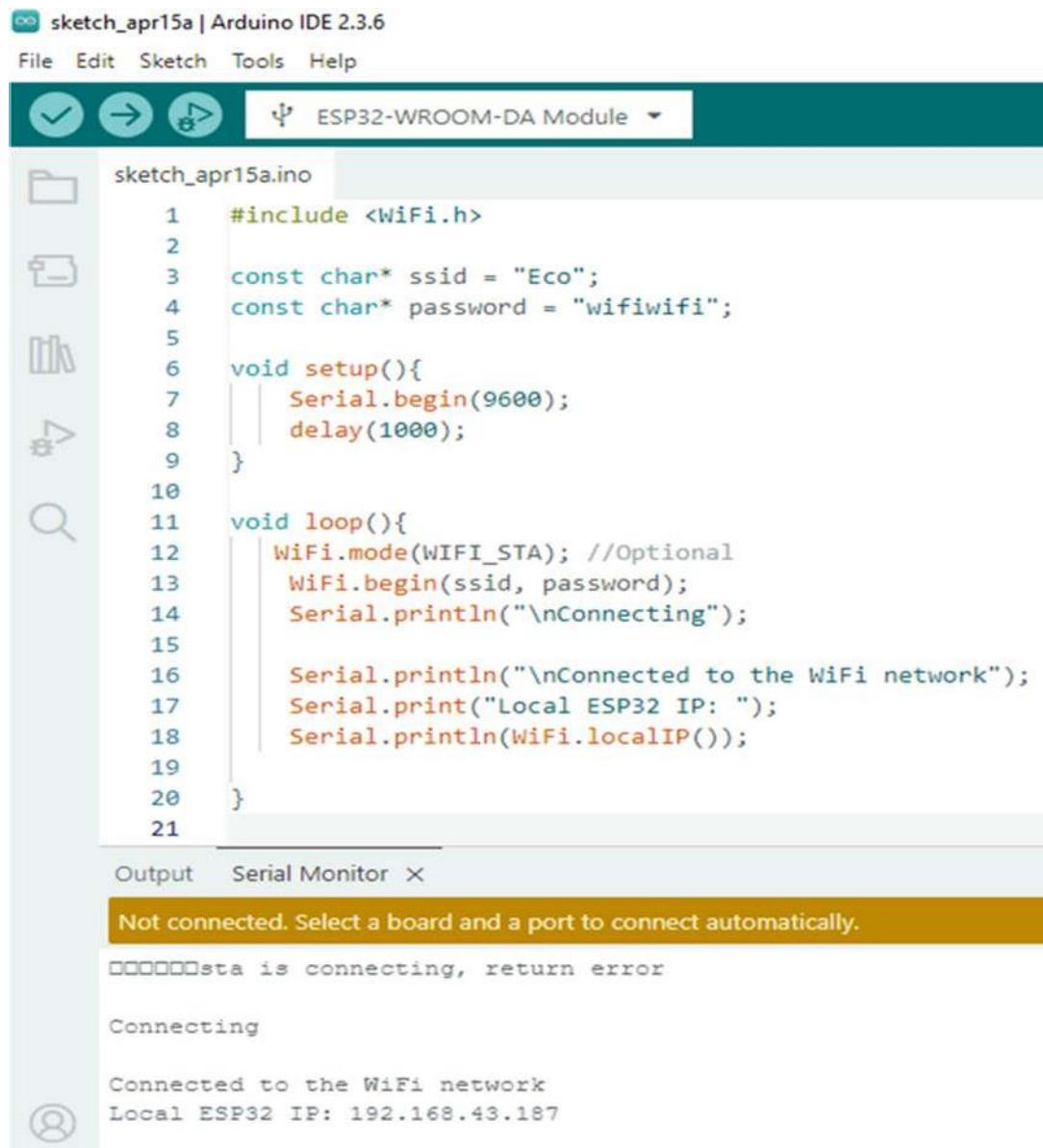


Figure III.35. ESP32 comme une station.

Un code Arduino IDE développé permettant de configurer l'ESP32 pour se connecter automatiquement à un réseau Wi-Fi au démarrage. Il commence par initialiser la communication série à 9600 bauds afin d'afficher des informations dans le moniteur série. Ensuite, il place l'ESP32 en mode station (WiFi_STA) et lance la connexion au réseau défini par l'SSID et le mot de passe. Dès que l'ESP32 obtient une adresse IP du routeur (statut

WL_CONNECTED), il sort de la boucle et affiche dans la console série l'adresse IP locale assignée.



```
sketch_apr15a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
ESP32-WROOM-DA Module
sketch_apr15a.ino
1  #include <WiFi.h>
2
3  const char* ssid = "Eco";
4  const char* password = "wifiwifi";
5
6  void setup(){
7      Serial.begin(9600);
8      delay(1000);
9  }
10
11 void loop(){
12     WiFi.mode(WIFI_STA); //Optional
13     WiFi.begin(ssid, password);
14     Serial.println("\nConnecting");
15
16     Serial.println("\nConnected to the WiFi network");
17     Serial.print("Local ESP32 IP: ");
18     Serial.println(WiFi.localIP());
19
20 }
21

Output Serial Monitor X
Not connected. Select a board and a port to connect automatically.
sta is connecting, return error

Connecting

Connected to the WiFi network
Local ESP32 IP: 192.168.43.187
```

Figure III.36. Configuration de l'ESP32 comme une station.

- La 3^{ème} étape : configuration de Fire base.

Le code Arduino IDE a pour objectif de connecter un ESP32 à un réseau Wi-Fi et à une base de données Fire base Realtime Database, Dans la fonction setup (), la connexion série est initialisée pour afficher des messages de débogage. Ensuite, le module Wi-Fi de l'ESP32 tente de se connecter au réseau identifié par le nom (SSID) et le mot de passe. En parallèle, les identifiants de l'utilisateur Fire base (email, mot de passe, URL de la base de données et clé API) sont configurés via les objets FirebaseAuth et FirebaseConfig. Le programme tente ensuite de s'inscrire ou de se connecter à Fire base à l'aide de Firebase.signUp(). Si la connexion est réussie, un message est affiché, et Fire base est initialisé avec Firebase.begin(), qui permet d'établir une communication sécurisée entre l'ESP32 et la base de données.

```
sketch_jun10a.ino
1  #include <Firebase_ESP_Client.h>
2  #include <WiFi.h>
3  #include <RTDBHelper.h>
4  #include <TokenHelper.h>
5  #include <ESP32Servo.h>
6
7  const char* ssid = "ECO";
8  const char* password = "WIFIWIFI";
9  // Identifiants Firebase
10 #define FIREBASE_HOST "URL de projet de Firebase"
11 #define FIREBASE_AUTH "API key"
12 // Utilisateur Firebase
13 #define USER_EMAIL "compte email"
14 #define USER_PASSWORD "mot de passe"
15 FirebaseConfig config;
16 FirebaseAuth auth;
17 FirebaseData fbdo;
18
19 void setup() {
20   Serial.begin(9600);
21
22   auth.user.email = USER_EMAIL;
23   auth.user.password = USER_PASSWORD;
24
25   config.database_url = FIREBASE_HOST;
26   config.api_key = FIREBASE_AUTH;
27
28   if(Firebase.signUp(&config, &auth, "", "")){
29     Serial.println("singUp OK");
30     signupOK = true;
```

Figure III.37. Configuration de Fire base.

- La 4^{ème} étape : Lecture des capteurs et envoi des données.

```
newcode.ino TokenHelper.h
142 // 3. Écrire infos utilisateur toutes les 10s
143 if (millis() - sendDataPrevMillis > 10000 || sendDataPrevMillis == 0) {
144   sendDataPrevMillis = millis();
145   int gazValue = analogRead(GAZ_PIN);
146   int gazPercent = map(gazValue, 0, 4095, 0, 100); // Conversion en pourcentage
147
148   String path = "/users/" + String(auth.token.uid.c_str()); // UID de l'utilisateur
149   FirebaseJson json;
150
151   json.set("email", USER_EMAIL);
152   json.set("timestamp", Firebase.getCurrentTime());
153   json.set("gaz", gazPercent); // Valeur du capteur
154
155   if (Firebase.RTDB.setJSON(&fbdo, path.c_str(), &json)) {
156     Serial.println("Infos et gaz envoyés !");
157     Serial.print(gazPercent);
158     Serial.println(" %");
159
160
161   if (Firebase.RTDB.setJSON(&fbdo, path.c_str(), &json)) {
162     Serial.println("Email enregistré dans Firebase !");
163   } else {
164     Serial.print("Erreur : ");
165     Serial.println(fbdo.errorReason());
166   }
167
168   delay(10000); // Attendre 10s avant de réessayer
```

Figure III.38. Transmission de la valeur de capteur de Gas vers Fire base.

L'ESP32 lit les valeurs des capteurs connectés (comme un capteur de Gas MQ-9, un capteur de mouvement) à l'aide de ses broches analogiques ou numériques. Une fois la lecture effectuée, ces données sont envoyées vers Fire base Realtime Database en utilisant les fonctions de la bibliothèque `Firebase_ESP_Client`.

La fonction `Firebase.RTDB.setInt()` est utilisée pour envoyer (ou mettre à jour) une valeur entière (int) dans Fire base Realtime Database depuis une carte ESP32.

- La 5^{ème} étape : contrôle des actionneurs.

```
newcode.ino  TokenHelper.h
75 void loop(){
76   | if (Firebase.ready()) {
77
78   // 1. Lire la LED depuis la racine "/LED1_PIN"
79   | if (Firebase.RTDB.getString(&fbdo, "/LED1_PIN")) {
80     String stateStr = fbdo.stringData();
81     Serial.println("Firebase LED1_PIN = " + stateStr);
82
83     // Interpréter le texte (ex: "on" ou "off")
84     if (stateStr == "true") {
85       digitalWrite(LED1_PIN, HIGH);
86     } else if (stateStr == "false") {
87       digitalWrite(LED1_PIN, LOW);
88     } else {
89       Serial.println("Valeur non reconnue !");
90     }
91   } else {
92     Serial.println("Erreur Firebase : " + fbdo.errorReason());
93   }
```

Figure III.39. Contrôle des actionneurs par Fire base.

Cette étape permet à l'ESP32 de récupérer des données stockées dans Fire base, généralement des instructions envoyées par un utilisateur à distance via une application mobile ou une interface web. Si un utilisateur modifier l'état de LED dans Fire base, cela met à jour une valeur et l'ESP32 va lire cette valeur pour exécuter l'action correspondante (comme allumer une LED).

La fonction `Firebase.RTDB.getInt()` est utilisée pour lire (ou récupérer) une valeur entière (int) depuis Fire base Realtime Database avec une carte ESP32. Elle permet à l'ESP32 de consulter à distance une donnée stockée dans la base, généralement une commande envoyée depuis une application mobile ou une interface web.

- La 6^{ème} étape : récupération des données depuis la plateforme Fire base.

Une fois que nous avons téléchargé le programme Arduino IDE sur le microcontrôleur ESP32 et connecté au Wi-Fi, accédons au projet Fire base que nous avons créé et réinitialisons Realtime Database ensuite toutes les données apparaîtront automatiquement.

L'interface de la plateforme Fire base après lancement du programme est illustrée par la figure ci-dessous.

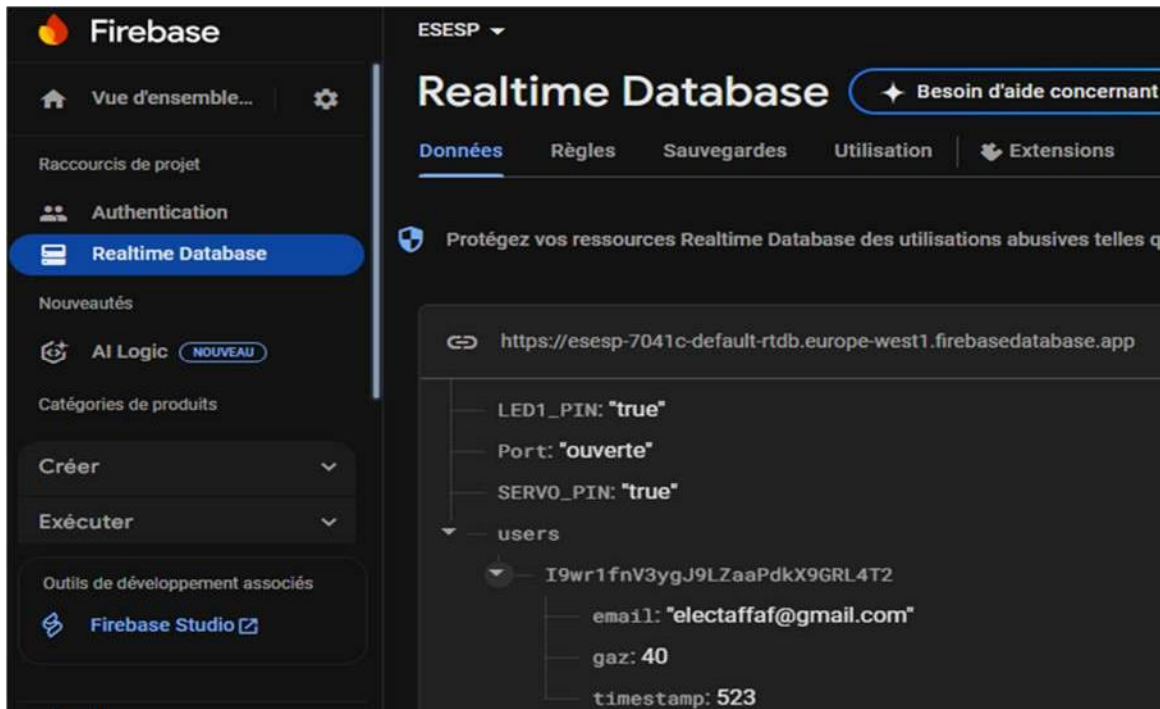


Figure III.40. Interface de Fire base (Realtime Database).

Après avoir suivi ces étapes, nous avons terminé d'assurer le fonctionnement entre le microcontrôleur ESP32 et plateforme Fire base.

III.5. Présentation MIT APP inventor

III.5.1. Définition

App Inventor est un environnement de développement initialement créé par Google, et désormais maintenu par le Massachusetts Institute of Technology (MIT). Il permet de concevoir des applications mobiles destinées aux appareils Android (comme les smartphones et les tablettes). Son principal avantage réside dans sa simplicité d'utilisation, rendant le développement accessible même aux débutants grâce à une interface graphique intuitive basée sur le glisser-déposer de blocs de code [13].



Figure III.41. MIT APP Inventor.

III.5.2. Interface de MIT APP Inventor

L'environnement de développement MIT App Inventor se compose principalement de deux parties :

Designer : C'est l'interface visuelle qui permet de créer l'apparence de l'application.

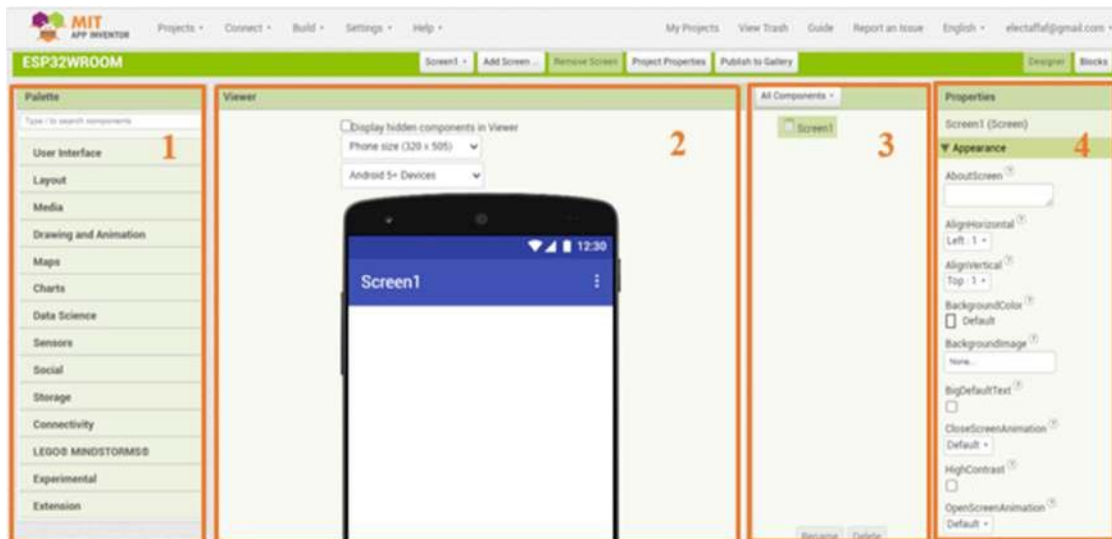


Figure III.42. Interface de Designer

Fenêtre1 : palette des objets à faire glisser sur l'écran.

Fenêtre 2 : écran ou l'on place les objets.

Fenêtre 3 : explorateur des objets, un arbre dont les racines est l'écran.

Fenêtre 4 : propriété de l'objet sélectionné, tout sur son apparence et son comportement [14].

Blocs : L'assemblage de blocs permet d'élaborer l'algorithme de l'application, Ils sont classés par familles, et on les met en place dans la zone de travail par un simple glissé-déposé.

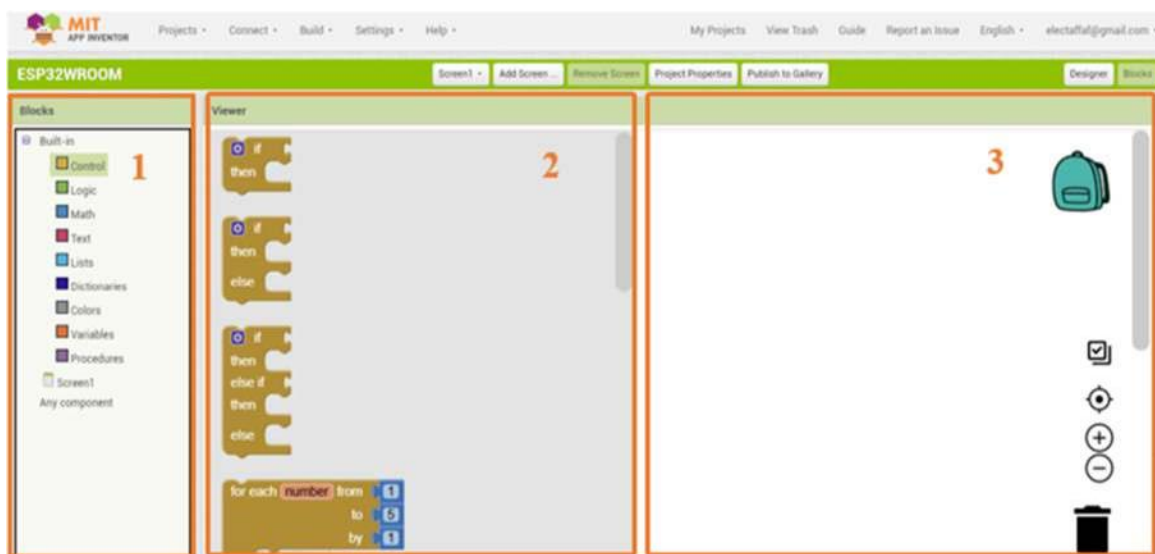


Figure III.43. Interface des blocs.

Fenêtre1 : famille de blocs de programmation.

Fenêtre2 : les blocs à faire glisser sur la zone de travail.

Fenêtre3 : zone de travail, les blocs y sont assemblés pour former l'algorithme [15].

III.5.3. Développement d'une application mobile par MIT APP

Les étapes nécessaires pour développer une application mobile avec MIT App Inventor sont décrits comme suit :

- La 1^{ère} étape : Créer un nouveau projet

Cliquer sur projets > commencer nouveau projet > Donner un nom significatif à votre projet (ex. : ESP32WROOM).

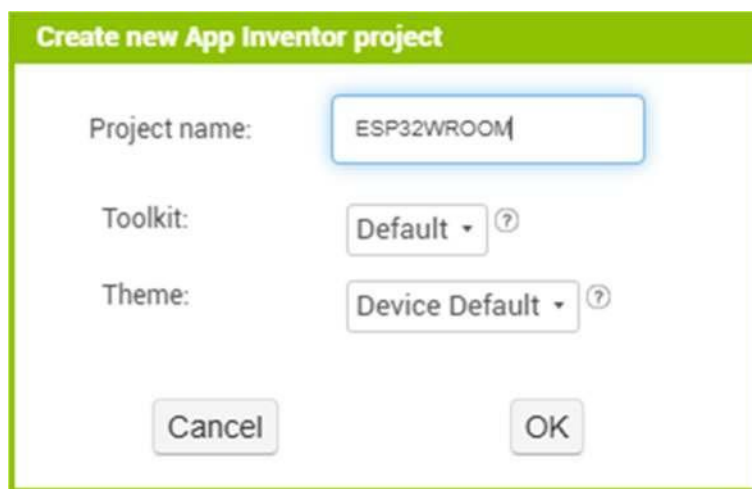


Figure III.44. Nouveau projet de MIT APP inventor.

- La 2^{ème} étape : Concevoir l'interface (Designer)

Glissons-déposons les composants visuels vers l'interface, ensuite Configurons leurs propriétés (bouton, label, champs texte, disposition...).

Ajoutons si nécessaire des composants non-visuels (clock, Fire base ...).

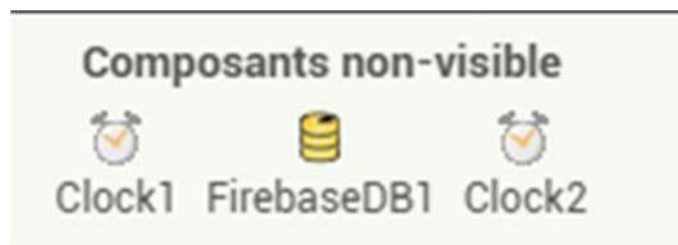


Figure III.45. Composants non-visuels

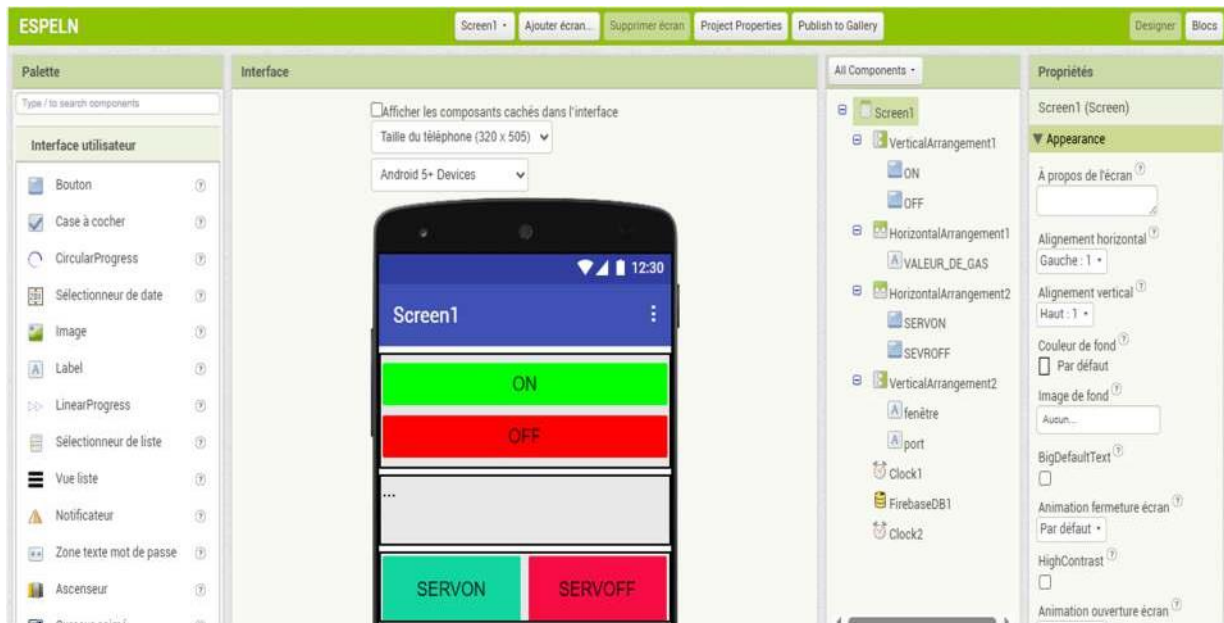


Figure III.46. Composants visuels sur l'interface.

Configuration de Fire base dans MIT APP Inventor : Copier Le lien URL et la clé API depuis Fire base, et les coller sur l'application.



Figure III.47. Configuration de Fire base dans MIT APP Inventor

- La 3^{ème} étape : Programmer le comportement (Blocks). Définir les actions à exécuter :
 - Lecture depuis Fire base :

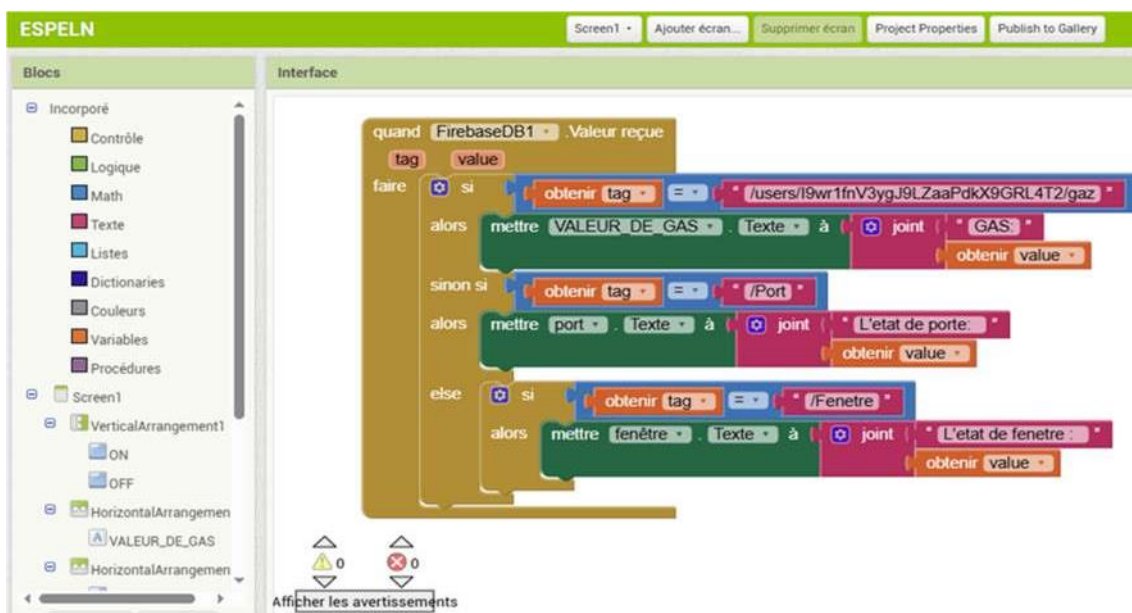


Figure III.48. Lecture des valeurs de Fire base.

La fonction FirebaseDB.valeur reçue est utilisée dans MIT App Inventor pour lire une donnée stockée dans Firebase Realtime Database. Elle prend en paramètre le chemin (tag) de la donnée que l'on souhaite récupérer.

Ces algorithmes faire la lecture de pourcentage de Gas (valeur en %) et l'état de la fenêtre ou de la porte (ouverte/fermée).

- Écriture vers Fire base :

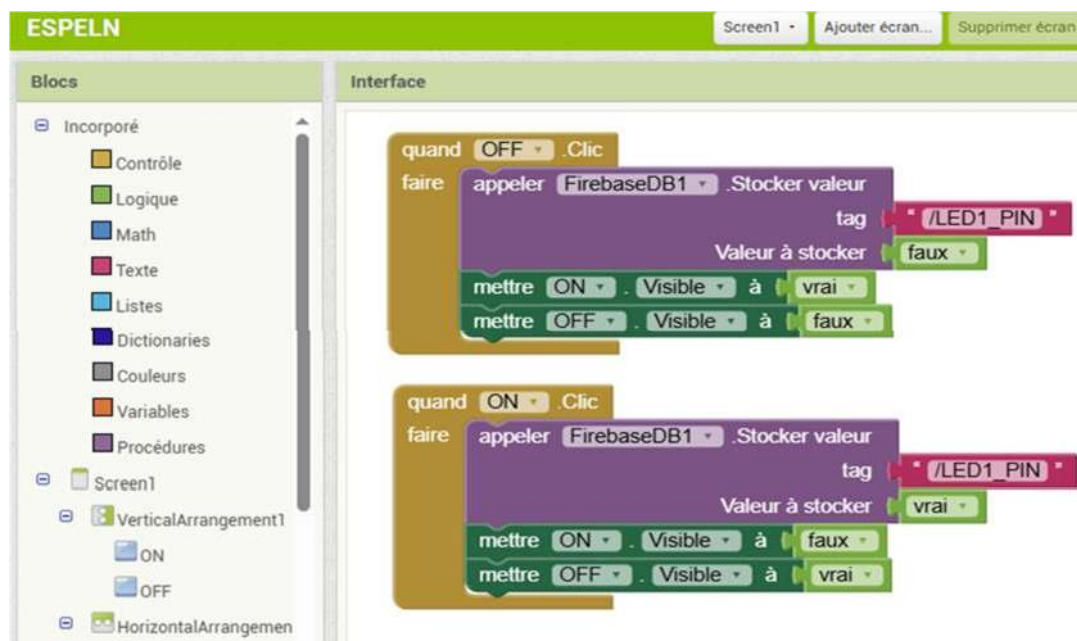


Figure III.49. Control des actionneurs.

La fonction FirebaseDB.stocker valeur dans MIT App Inventor permet d'enregistrer ou mettre à jour une donnée dans Fire base Realtime Database.

Ces algorithmes faire le contrôle d'une LED depuis deux boutons (ON/OFF).

- Modification des données :

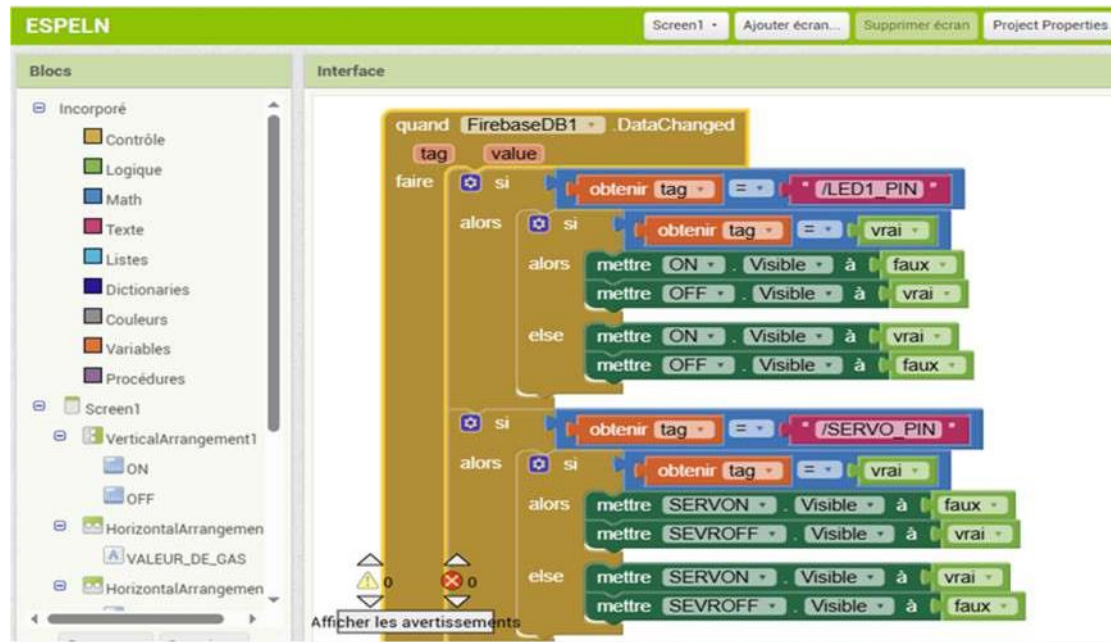


Figure III.50. Réinitialisation des données depuis Fire base.

La fonction FirebaseDB.DataChanged est un événement déclencheur dans MIT App Inventor, utilisé pour détecter en temps réel toute modification de données dans la Firebase Realtime Database.

- La 4^{ème} étape : Tester et envoyer l'application.
- Installer l'application MIT AI2 Companion sur un téléphone Android.

MIT AI2 Companion

MIT App Inventor

4,2★
29,1 k avis

5 M+
Téléchargements

3 ans et plus

Installer

Partager

Ajouter à la liste de souhaits



Figure III.51. Application MIT AI2 Companion.

- Scanner le code QR pour tester l'application en temps réel.



Figure III.52. le QR code

Après scanner le code QR par l'application MIT AI2 Companion sur notre téléphone Android, le résultat est obtenu est donné par la figure suivante :



Figure III.53. Interface de téléphone Android

Avec cette application nous pouvons modifier, contrôler et gérer notre donnée à distance et en temps réel.

III.6. Conclusion

Ce chapitre a traité la partie logicielle qui présente la deuxième partie de la conception d'un système de sécurité intelligent, incluant la programmation du microcontrôleur, la connexion à la plateforme Fire base, ainsi que le développement de l'application mobile avec MIT APP inventor. Ces éléments assurent une gestion efficace des données et une interaction facile à l'utilisateur.

Le chapitre suivant sera consacré à la présentation de la réalisation du système ainsi que les résultats obtenus après les essais pratiques du système avec des tests du programme Arduino et le contrôle par l'application correspondante, afin d'évaluer les performances et la fiabilité du système réalisé.

Bibliographie chapitre 3

- [1] TAIBAOUI Wassel BOUKHALFA Ahmed. « *Étude et réalisation d'un système de sécurité domestique pour une maison intelligente* », mémoire de fin d'études de Master en Electronique, 2023.
- [2] [Installing ESP32 in Arduino IDE \(Windows, Mac OS X, Linux\) | Random Nerd Tutorials](#) (03/06/2025)
- [3] Kessaissia Mohamed Fares & Ferhat Mohamed. « *Conception d'un système de Gestion intelligente (IoT) de confort thermique des bâtiments* », Mémoire de Master Filière Électronique Spécialité Instrumentation, 2023.
- [4] Yield Studio. (n.d.). *Firestore : plateforme de développement mobile*. Disponible sur : <https://www.yieldstudio.fr/glossaire/firebase-plateforme-developpement-mobile> (08/06/2025)
- [5] GeeksforGeeks. (2025, February 18). *Firestore – Introduction*. Disponible sur: https://www.geeksforgeeks.org/firebase-introduction/?utm_source=chatgpt.com (08/06/2025)
- [6] Sharma, D. (2023, October 10). *Google Firestore, Supabase, and Backend Development*. Medium. <https://medium.com/dhiwise/google-firebase-supabase-and-backend-development-96e9062dac97>
- [7] PhoenixNAP. (n.d.). *What Is NoSQL? – NoSQL Definition & Features*. Disponible sur: https://phoenixnap.com/kb/what-is-nosql?utm_source=chatgpt.com (08/06/2025)
- [8] Yield Studio. (n.d.). *Firestore : Les 8 principales fonctionnalités de Firestore*.disponible sur : <https://www.yieldstudio.fr/glossaire/firebase-plateforme-developpement-mobile> (08/06/2025)
- [9] Yield Studio. (n.d.). *Firestore :Les 5 avantages de Firestore*. Disponible sur : <https://www.yieldstudio.fr/glossaire/firebase-plateforme-developpement-mobile> (08/06/2025)
- [10] Silva, R. (n.d.). *ESP32: Data Logging to Firestore Realtime Database*. Random Nerd Tutorials.disponible sur : <https://randomnerdtutorials.com/esp32-data-logging-firebase-realtime-database/> (09/06/2025)
- [11] Megantoro, P., Prastio, R. P., Faqih, H. A. K., Kusuma, H. A., Abror, A., & et al. (2022). *Instrumentation system for data acquisition and monitoring of hydroponic farming using ESP32 via Google Firestore*. *Indonesian Journal of Electrical Engineering and Computer Science*, **27**(1), 52–61. Disponible sur: https://www.researchgate.net/publication/362996942_Instrumentation_system_for_data_acquisition_and_monitoring_of_hydroponic_farming_using_ESP32_via_Google_Firebase (09/06/2025)
- [12] PhysicsOpenLab. (2020, May 10). *DataLogging with ESP32 Module*. PhysicsOpenLab disponible sur : <https://physicsopenlab.org/2020/05/10/datalogging-with-esp32-module/> (09/06/2025)
- [13] Lycée Blaise Pascal. (n.d.). *App Inventor*.disponible sur : [App Inventor – L'Informatique, c'est fantastique !](#) (10/06/2025)
- [14] Lycée Blaise Pascal. (n.d.). *Le Designer*. In *Prise en main d'App Inventor*. L'Informatique, c'est fantastique ! disponible sur : [Le Designer – L'Informatique, c'est fantastique !](#) (10/06/2025)

[15] Lycée Blaise Pascal. (n.d.). *Les blocs de programmation*. In *App Inventor. L'Informatique, c'est fantastique !* disponible sur : [Les Blocs de programmation – l'Informatique, c'est fantastique !](#) (10/06/2025).

CHAPITRE IV

Chapitre IV

Réalisation d'un système de sécurité intelligent pour une maison

IV.1. Introduction

L'objectif principal de ce projet est de concevoir un système de sécurité intelligent capable de détecter différents types de risques domestiques (fuite de Gas, intrusion, bruit anormal, ouverture non autorisée) et de réagir automatiquement en temps réel, tout en permettant un suivi et un contrôle à distance via une application mobile.

Ce chapitre a pour but de valider la mise en œuvre pratique du système proposé. Il présente les tests réalisés sur les capteurs et les actionneurs, la communication avec la base de données Fire base, ainsi que le fonctionnement de l'application mobile développée à l'aide de MIT App Inventor. Les résultats obtenus sont ensuite analysés afin d'évaluer la fiabilité, la réactivité et les limites du système mis en place.

IV.2. Modélisation de la maison

Cette étape consiste à créer un modèle pour une maison (maquette). Pour cela, nous avons modélisé la structure principale (plan), contenant les murs extérieurs avec le sol. Le modèle consiste en une maquette de dimension de 35×40cm constituée d'une entrée, une cuisine, une chambre, un couloir. La Figure IV.1 illustre la structure du plan.

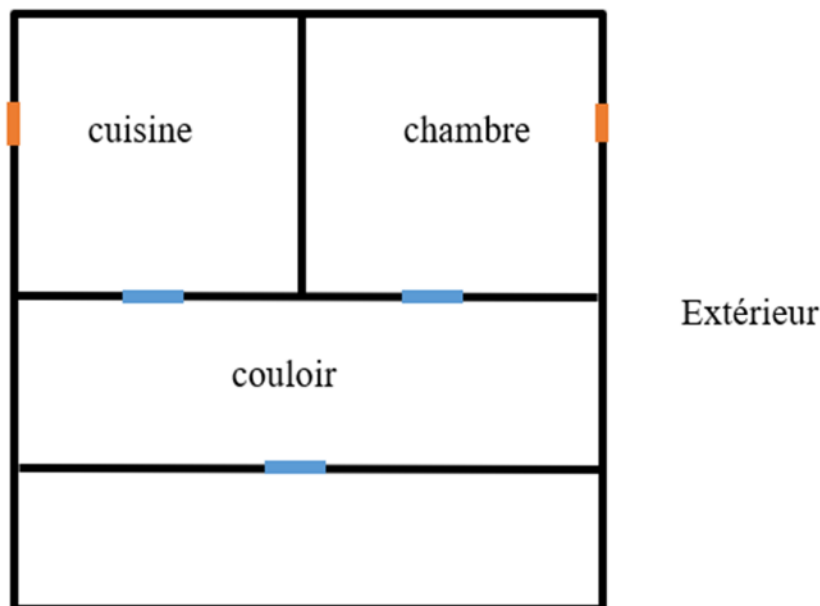


Figure IV.1. Structure du plan de la maison

La surface blanche illustre le sol, les lignes noires représentent les murs, les traits bleus décrivent les portes et les traits orange les fenêtres.

Un prototype réalisé du système d'étude est représenté par la figure IV.2 suivante :

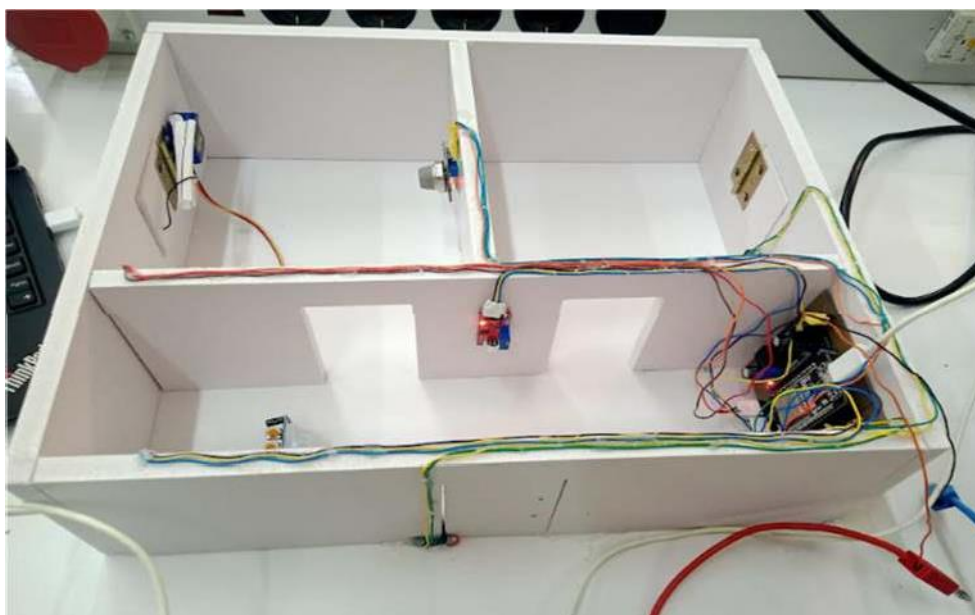


Figure IV.2. Photo du prototype réalisé.

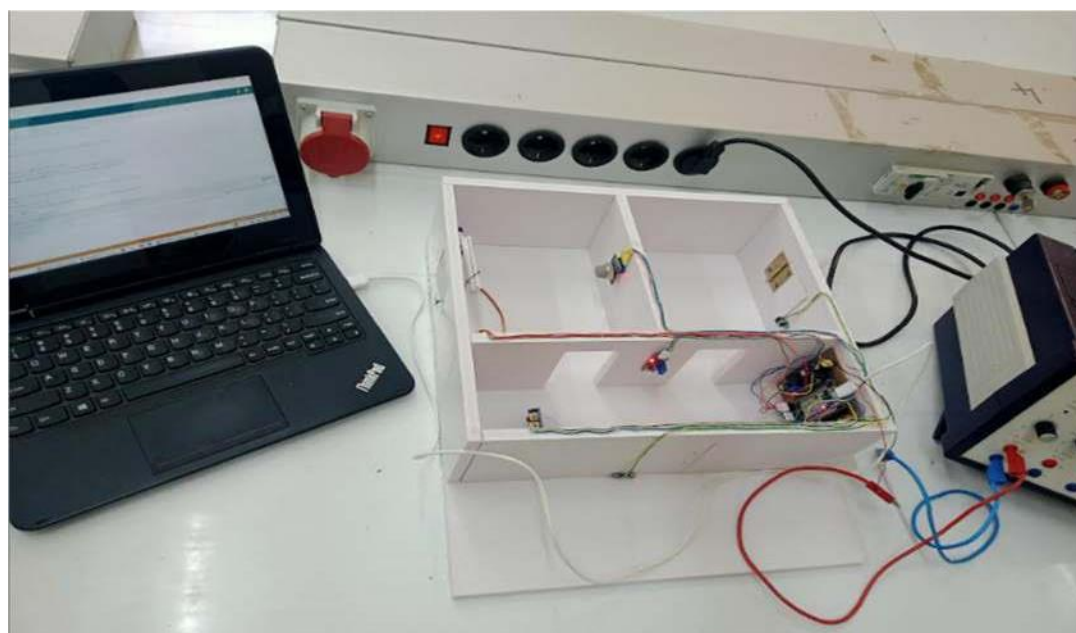


Figure IV.3. Téléversement du programme.

IV.3. Disposition des composants

Les composants sont disposés sur la maquette de manière à assurer certaines fonctionnalités de la domotique bien précisément la sécurité intelligente d'une maison.

La disposition des composants est faite comme suit :

Cuisine : un capteur de Gas MQ-9 et un servomoteur placé sur la fenêtre.

Porte d'entrée : switch placé sur le mur à côté de la porte.

Couloir : un capteur de bruit KY-037, un capteur de mouvement PIR et un BUZZER.

Chambre : switch à côté de la fenêtre.

IV.4. Architecture globale du système

Afin de réaliser un système de sécurité intelligent pour une maison, nous avons adopté une architecture globale simple et efficace, La Figure IV.4 illustre cette architecture, qui se compose de trois parties principales, interconnectées via les technologies de l'Internet des Objets (IoT) :

- 1- Partie utilisateur.
- 2- Partie ESP32.
- 3- Partie matérielle.

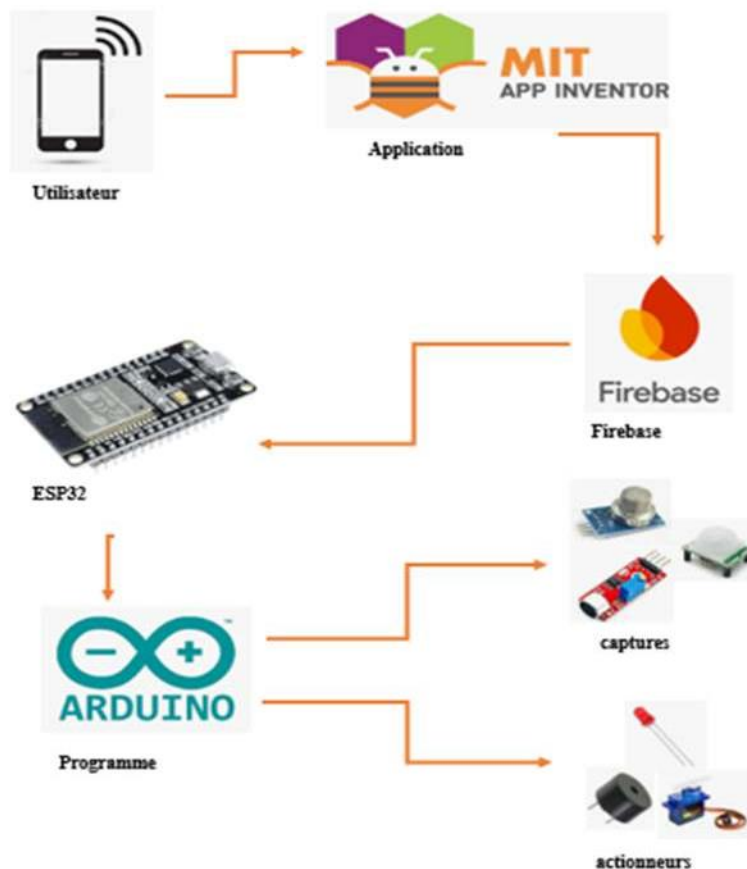


Figure IV.4. Architecture globale du système.

IV.4.1. Partie utilisateur

Elle comprend l'application mobile développée avec MIT App Inventor, qui permet à l'utilisateur de consulter les données en temps réel, recevoir des alertes, et interagir avec le système à distance.

IV.4.2. Partie ESP32

Cette section représente le cœur du traitement. Le microcontrôleur ESP32-WROOM-32, programmé avec l'environnement Arduino IDE, assure la lecture des capteurs, la prise de décision et la communication avec Fire base.

IV.4.3. Partie matérielle

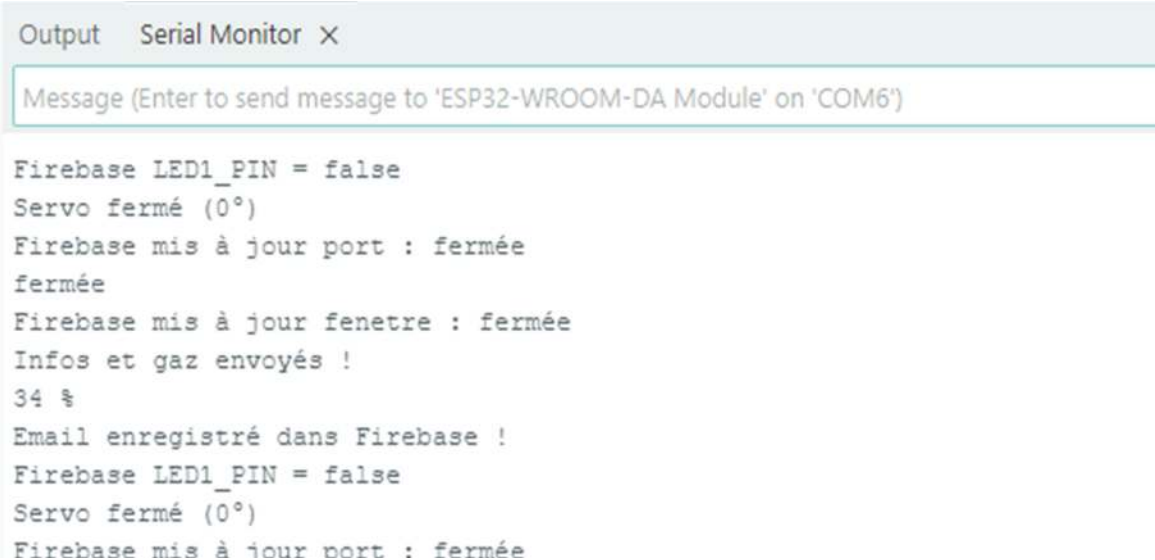
Elle regroupe l'ensemble des capteurs (Gas, mouvement, bruit, ouverture de porte/fenêtre) et des actionneurs (LEDs, buzzer, servo-moteur) utilisés pour détecter les événements critiques et réagir de manière automatique.

IV.5. Tests fonctionnels des sous-systèmes

Afin de valider le bon fonctionnement de l'ensemble du système de sécurité, chaque sous-système a été testé individuellement dans des conditions proches de la réalité. Ces tests ont permis de vérifier la réactivité des capteurs, le comportement des actionneurs ainsi que l'intégrité des communications entre les différents modules (ESP32, Fire base, application mobile). Les résultats obtenus sont présentés dans les sous-sections suivantes.

IV.5.1. Test du capteur de Gas MQ-9

Vérifions que le capteur MQ-9 détecte une fuite de Gas, envoie les données vers la base de données Fire base, et permette à l'utilisateur, via l'application mobile, de contrôler manuellement le servomoteur pour ouvrir ou fermer une fenêtre en réponse à cette détection. La Figure IV.5 illustre les résultats sur moniteur série de l'Arduino IDE avant la détection de fuite de Gas.



```
Output  Serial Monitor X
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM6')
Firebase LED1_PIN = false
Servo fermé (0°)
Firebase mis à jour port : fermée
fermée
Firebase mis à jour fenetre : fermée
Infos et gaz envoyés !
34 §
Email enregistré dans Firebase !
Firebase LED1_PIN = false
Servo fermé (0°)
Firebase mis à jour port : fermée
```

Figure IV.5. Résultat avant la détection de fuite de Gas.

La figure IV.6 présente l'état de la maison à un niveau normal de Gas.



Figure IV.6. Etat de maison au niveau normal de Gas.

La figure IV.7 illustre le niveau normal de Gas affiché sur Fire base.

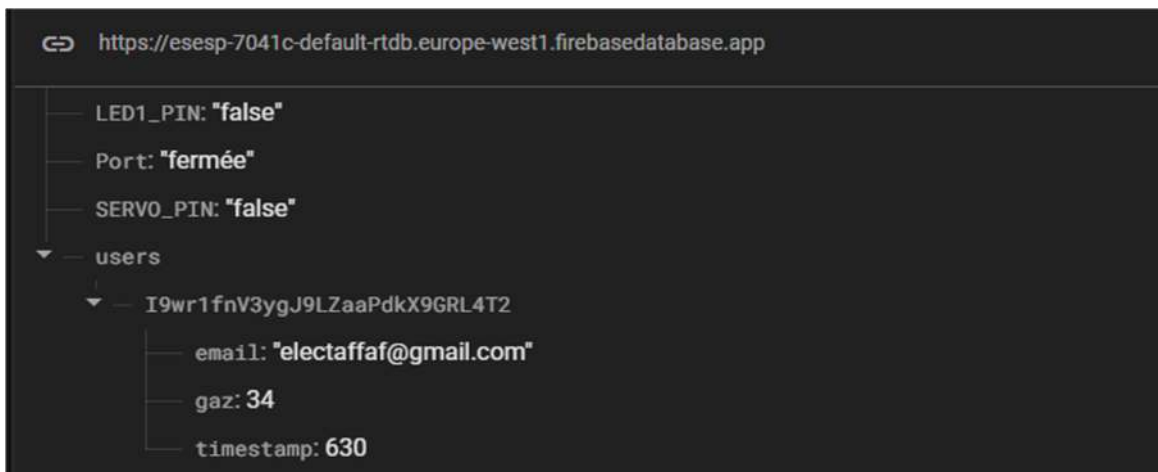


Figure IV.7. Niveau normal de Gas sur Fire base.

La figure IV.8 illustre l'affichage au niveau normal de Gas sur l'application Android.



Figure IV.8. Niveau normal de Gas sur l'application Android.

La Figure IV.9 illustre les résultats sur moniteur série de l'Arduino IDE après la détection de fuite de Gas (seuil de 47%).

La LED verte de capteur MQ-9 est allumée.

Nous envoyons un ordre pour l'ouverture de fenêtre (servomoteur à 90°).

```
Output  Serial Monitor X
Not connected. Select a board and a port to connect automatically.
Firebase mis à jour port : fermée
fermée
Firebase mis à jour fenetre : fermée
Infos et gaz envoyés !
49 %
Email enregistré dans Firebase !
Firebase LED1_PIN = false
Servo ouvert (90°)
Firebase mis à jour port : fermée
fermée
```

Figure IV.9. Résultat après la détection de fuite de Gas

La figure IV.10 montre l'état de maison après la fuite de Gas.



Figure IV.10. Etat de la maison après la fuite de Gas

La figure IV.11 illustre le niveau de Gas après la détection sur Fire base.

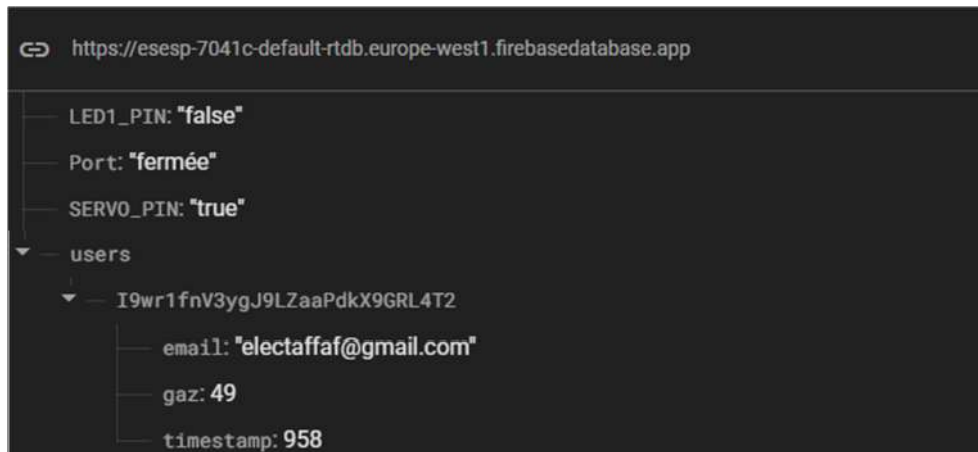


Figure IV.11. Niveau de Gas après la détection sur Fire base

La figure IV.12 illustre l'affichage de niveau Gas sur l'application Android.



Figure IV.12. Niveau de Gas sur l'application Android

Résultats observés :

- Le capteur MQ-9 a fonctionné correctement et les valeurs sont visibles dans l'application.
- La commande du servomoteur via le bouton "SERVO ON / SERVO OFF" est transmise avec succès.
- Le servomoteur a réagi instantanément :
 - Rotation à 90° pour ouvrir.
 - Retour à 0° pour fermer.
- Aucun bug ou retard n'a été observé durant le test.

IV.5.2. Test des switches de porte et de fenêtres

Après avoir vérifié la détection de l'état de l'ouverture ou de la fermeture d'une porte et d'une fenêtre à l'aide de deux interrupteurs (switches) magnétiques, le système envoie ces états

vers la base de données Fire base pour affichage sur l'application mobile. La Figure IV.13 montre les résultats obtenus sur moniteur série de l'Arduino IDE.

```
Output  Serial Monitor  X
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM6')

33  📧
Email enregistré dans Firebase !
Firebase LED1_PIN = false
Servo fermé (0°)
Firebase mis à jour port : fermée
ouverte
Firebase mis à jour fenetre : ouverte
Infos et gaz envoyés !
```

Figure IV.13. Résultat test accès non autorisé

La figure IV.14 montre l'ouverture de la fenêtre et fermeture de la porte.

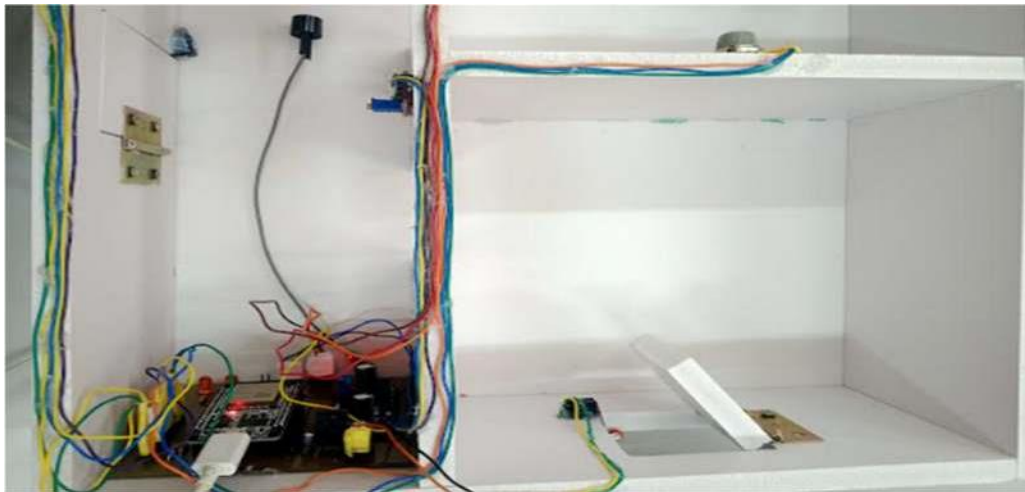


Figure IV.14. Ouverture de la fenêtre et fermeture de la porte.

La figure IV.15 illustre l'affichage de l'ouverture de la fenêtre et fermeture de la porte sur Fire base.

```
🔗 https://esesp-7041c-default-rtdb.europe-west1.firebaseio.com/app/
https://esesp-7041c-default-rtdb.europe-west1.firebaseio.com/app/
├── Fenetre: "ouverte"
├── LED1_PIN: "false"
├── Port: "fermée"
```

Figure IV.15. Etat de la fenêtre et la porte et sur Fire base.

La figure IV.16 illustre l'état d'ouverture de la fenêtre et fermeture de la porte sur l'application Android.



Figure IV.16. Etat de la porte et de la fenêtre sur l'application Android

Résultats observés :

- Les deux switches sont correctement détecté les changements d'état.
- Le temps de transmission vers Fire base est très petit (< 10 secondes).
- L'application mobile a bien affiché "Porte ouverte/fermée" et "Fenêtre ouverte/fermée".
- Le système est stable et fiable, même après plusieurs manipulations successives.

IV.5.3. Test des capteurs de bruit KY-037 et de mouvement PIR

Vérifier le bon fonctionnement combiné des capteurs de bruit KY-037 et de mouvement PIR dans le système de sécurité. Lorsqu'un bruit fort et un mouvement est détecté, le système doit afficher une alerte sur le moniteur série de l'Arduino IDE.

En cas de la détection d'intrusion, on obtient les résultats suivants :

- Activer le buzzer comme signal d'alerte sonore.
- Allume une LED.

La figure IV.17 illustre les résultats visibles sur le moniteur série de l'Arduino IDE.

```
Output  Serial Monitor X
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM6')
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 2
Coup de bruit détecté ! Total : 3
Coup de bruit détecté ! Total : 4
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 2
Coup de bruit détecté ! Total : 3
Coup de bruit détecté ! Total : 4
Coup de bruit détecté ! Total : 5
Coup de bruit détecté ! Total : 6
Coup de bruit détecté ! Total : 7
Coup de bruit détecté ! Total : 8
Coup de bruit détecté ! Total : 9
Intrusion détectée ! Déclenchement de l'alarme !
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 2
Coup de bruit détecté ! Total : 3
Intrusion détectée ! Déclenchement de l'alarme !
```

Figure IV.17. Résultat du test de détection combinée du KY-037 et PIR

La figure IV.18 montre la détection combinée du capteur de bruit KY-037 et du capteur de mouvement PIR.

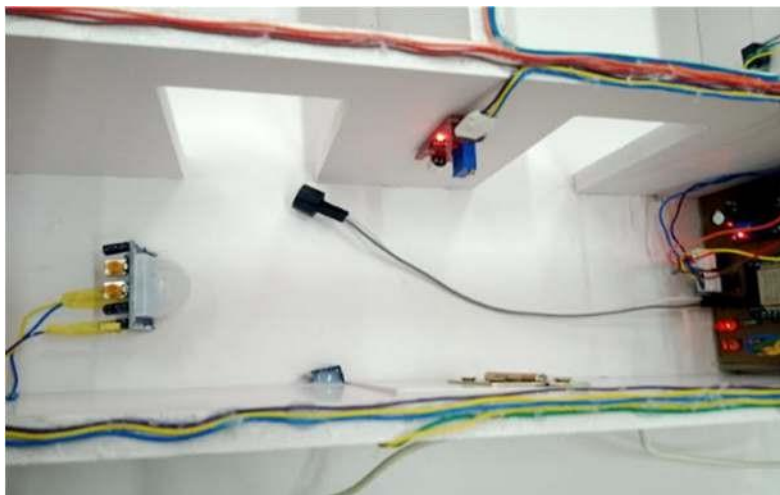
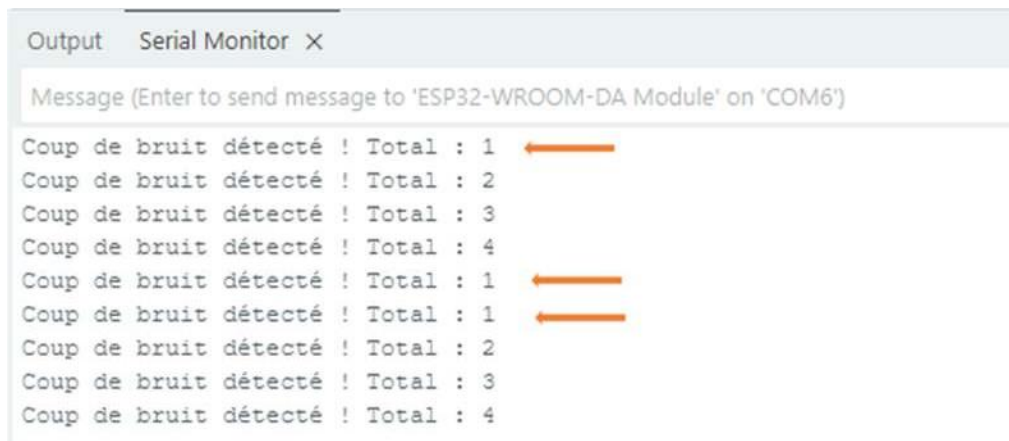


Figure IV.18. Détection d'intrusion.

En cas d'absence d'une intrusion, s'il n'y a pas de détection de mouvement. La figure IV.19 montre la réinitialisation du compteur de coup de bruit.



```
Output  Serial Monitor X
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM6')
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 2
Coup de bruit détecté ! Total : 3
Coup de bruit détecté ! Total : 4
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 1
Coup de bruit détecté ! Total : 2
Coup de bruit détecté ! Total : 3
Coup de bruit détecté ! Total : 4
```

Figure IV.19. Résultat de réinitialisation du compteur.

Résultats observés :

- Chaque coup de bruit est bien détecté par le capteur KY-037, et incrémente un compteur visible dans le moniteur série.
- Le compteur se réinitialise automatiquement toutes les 10 secondes, sauf si un mouvement est détecté dans l'intervalle déterminé.
- En cas de bruit répété sans détection de mouvement, aucune alarme n'est déclenchée, ce qui évite les fausses alertes dues à des bruits ambiants.
- Lorsque plusieurs coups de bruit sont détectés dans une période de 10 secondes et qu'un mouvement est enregistré simultanément par le capteur PIR, une alarme (BUZZER et LED) est déclenchée.

IV.6. Conclusion

Ce chapitre a permis de valider le bon fonctionnement du système de sécurité intelligent à travers une série de tests réalisés sur les différents sous-systèmes. Chaque capteur (Gas, mouvement, bruit, switch) a été examiné individuellement.

Les résultats observés ont démontré la fiabilité et la réactivité du système. L'intégration conditionnelle des capteurs, notamment la détection simultanée du bruit et du mouvement, a permis d'éviter efficacement les fausses alertes. De même, les actionneurs (BUZZER, LEDs, servomoteur) ont réagi de manière conforme aux situations détectées.

L'application mobile, quant à elle, a permis une surveillance et un contrôle à distance efficaces, renforçant ainsi l'interactivité et l'accessibilité du système.

Dans l'ensemble, les objectifs fixés au début de projet ont été atteints, validant la pertinence de l'approche adoptée pour assurer une sécurité domestique intelligente, autonome et connectée.

CONCLUSION GÉNÉRALE

Conclusion générale

Dans un contexte où la sécurité des habitations devient une nécessité face à la montée des risques tels que les intrusions, les fuites de Gas ou les incidents domestiques.

Ce travail de mémoire est consacré à la conception et la réalisation d'un système de sécurité intelligent basé sur les technologies modernes de l'Internet des Objets (IoT), assurant par ailleurs une sécurité optimale d'une maison, pour cette raison on a utilisé des capteurs, des switches et des actionneurs connectés à un microcontrôleur ESP32, qui sont contrôlés par une application Android.

Le premier chapitre a expliqué les bases théoriques en présentant les enjeux de la sécurité domestique, les types de risques et l'apport des systèmes intelligents.

Le deuxième chapitre est consacré à la présentation du matériel utilisé, notamment l'ESP32-WROOM-32, les capteurs (Gas, mouvement, bruit, ouverture) et les actionneurs (servo-moteur, buzzer, LEDs), en justifiant chaque choix technique.

Le troisième chapitre a détaillé l'implémentation logicielle à travers la programmation sur l'environnement Arduino IDE, la connexion à la base de données Fire base pour le stockage des données en temps réel, et le développement d'une application mobile via MIT App Inventor pour l'affichage et le contrôle à distance.

Enfin, le quatrième chapitre a permis de valider expérimentalement les fonctionnalités du système à travers des tests unitaires et globaux. Les résultats ont montré que le système fonctionne de manière fiable et cohérente, avec une capacité à détecter les situations critiques et à réagir automatiquement, tout en évitant les fausses alertes.

En conclusion, ce travail a permis de démontrer la faisabilité et l'efficacité d'un système de sécurité connecté pour l'habitat, combinant accessibilité, autonomie, et réactivité.

Comme perspective à ce travail sur la conception et la réalisation d'un système de sécurité intelligent pour une maison, nous favorisons de futures améliorations d'augmenter le nombre de capteurs installés tels que l'intégration de caméras IP, la reconnaissance vocale et d'améliorer le contrôle des capteurs avec l'application Android ou l'envoi d'alertes via SMS ou notification push pour une protection encore plus avancée.